INOVANCE

Advancing industrial technology,
for a better world



Industrial Robot InoRobotLab

User Guide

Industrial Automation | New Energy Vehicle | Intelligent Elevator | Intelligent Robot | Digital Energy | Rail Transit

Data code 19012861A01

Legal and Disclaimer

Copyright

Copyright © 2026 Shenzhen Inovance Technology Co., Ltd. All rights reserved

This documentation is the exclusive property of Shenzhen Inovance Technology Co., Ltd. No individual or entity may excerpt, reproduce, modify, translate, or distribute any content herein without written consent from Inovance.

Legal action will be taken against infringement.

Trademarks

INOVANCE is a registered trademark of Shenzhen Inovance Technology Co., Ltd. and its affiliates. All other trademarks or registered trademarks mentioned in this documentation are the property of their respective owners. Unauthorized use of these trademarks by third parties for any purpose without written authorization could violate the rights of their owners.

Disclaimer of Liability

Due to continuous updates and improvements of products and technologies, the content of this documentation may not fully match the actual products. In the event of any discrepancies, the actual products shall prevail.

The contents are subject to change without notice due to product upgrade.

Waste Disposal

The storage, use, and disposal of this product (including optional accessories) must comply with local laws and regulations.

Qualified Personnel

The product/system described in this documentation may be operated only by personnel qualified for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel can identify the risks of the product/system and prevent possible dangers.

Proper Use of the Product

Proper transportation, storage, assembly, installation, commissioning, operation, and maintenance are required to ensure the safe operation of the product without any problems. The required ambient conditions must be met. All operations must follow the guidelines provided in this documentation.

Preface

Introduction

InoRobotLab is a software used for programming, debugging, setting, monitoring, and other operations of Inovance robots on Windows PC. This guide aims to provide you with comprehensive, detailed, and easy-to-understand instructions to help you quickly familiarize with and master the operation methods of various functions of Inovance robots.

More Data

Data Name	Data Code	Description
IR-TP200 Series Industrial Robot Teach Pendant User Guide	19012798	This guide introduces software-related operations of the robot teach pendant. Contact Inovance technical support to obtain this guide.
Industrial Robot Instruction Guide	19012825	This guide introduces the function descriptions, instruction formats and parameter explanations, usage examples, and application cases for project instructions of Inovance Technology industrial robots.
Industrial Robot Remote I/O Control Function User Guide	19012874	This guide introduces the interface configuration, bus read/write instruction formats, and bus invocation precautions for remote I/O control of Inovance Technology industrial robots.
Industrial Robot Remote Ethernet Control Function User Guide	19012860	This guide introduces the interface configuration, API call format, API function call examples, common issues and troubleshooting for remote Ethernet control of Inovance Technology industrial robots.
Industrial Robot Controller Software Update User Guide	19012875	This guide introduces the software update procedures and component update procedures for Inovance Technology industrial robots.

Data Name	Data Code	Description
Inovance Robot Selection Guide	19120306	This guide is designed to help users understand the product types of Inovance robots, providing selection logic, configuration examples, and parametric decision models to facilitate precise matching with production scenario requirements.
Industrial Robot Communication Guide	19012876	This guide introduces the comprehensive multi-protocol communication solution for Inovance industrial robots. It covers mainstream industrial standards including Modbus, EtherNet/IP, PROFINET, EtherCAT, and the MC protocols, serving as a complete technical guide for industrial automation system integration.
Industrial Robot InoRobotLab User Guide (This guide)	19012861	This guide introduces software-related operations of the robot InoRobotLab software. Contact Inovance technical support to obtain this guide.

Revision History

Revision Date	Data Version	Revision
2026-04	A01	Corrected some minor errors.
2026-03	A00	First issue.

Access to the Guide

This guide is not delivered together with the product. You can obtain the PDF version in the following ways.

- Do keyword search under Service and Support at https://www.inovance.com/global.
- Scan the QR code on the product with your smart phone.
- Scan the QR code below to install My Inovance app, where you can search for and download user guides.

Basic Safety Instructions

Safety Precautions

Safety Disclaimer

- This chapter presents essential safety instructions for a proper use of the equipment. Before operating the equipment, read through the guide and comprehend all the safety instructions. Failure to observe the safety instructions may result in death, severe personal injuries, or equipment damage.
- "DANGER", "WARNING", and "CAUTION" items in this guide do not indicate all safety precautions that need to be followed; instead, they just supplement the safety precautions.
- Use this product according to designated environment requirements. Damage caused by improper use is not covered by warranty.
- Inovance shall take no responsibility for any personal injuries or property damage caused by improper usage.

Safety Levels and Definitions



Indicates that failure to comply with the notice will result in severe personal injuries or even death.



Indicates that failure to comply with the notice may result in severe personal injuries or even death.



Indicates that failure to comply with the notice may result in minor or moderate personal injuries or equipment damage.

Safety Precautions

- Drawings in the guide are sometimes shown without covers or protective guards. Remember to install the covers or protective guards as specified first, and then perform operations in accordance with the instructions.
- The drawings in the guide are shown for illustration only and may be different from the product you purchased.
- Operators must take mechanical precautions to protect personal safety and wear protective equipment, such as safety shoes, safety clothing, safety glasses, protective gloves, and protective sleeves.

Unpacking



- Do not install the equipment if you find damage, rust, or signs of use on the equipment or accessories upon unpacking.
- Do not install the equipment if you find water seepage or missing or damaged components upon unpacking.
- Do not install the equipment if you find the packing list does not conform to the equipment you received.



- Before unpacking, check whether the outer package is intact, damaged, wet, damped, or deformed.
- Unpack the package in sequence. Do not hit the package violently.
- Check whether there is damage, rust, or injuries on the surface of the equipment and equipment accessories before unpacking.
- Check whether the package contents are consistent with the packing list before unpacking.

Storage and Transportation



- Large-scale or heavy equipment must be transported by qualified professionals using specialized hoisting equipment. Failure to comply may result in personal injuries or equipment damage.
- Before hoisting the equipment, ensure that components such as the front cover and terminal blocks are secured firmly with screws. Loosely-connected components may fall off and result in personal injuries or equipment damage.
- Never stand or stay below the equipment when the equipment is being hoisted by the hoisting equipment.
- When hoisting the equipment with a steel rope, ensure the equipment is hoisted at a constant speed without suffering from vibration or shock. Do not turn the equipment over or let the equipment stay hanging in the air. Failure to comply may result in personal injuries or equipment damage.

CAUTION

- Handle the product with care and mind your steps to prevent personal injuries or product damage.
- When carrying the equipment with bare hands, hold the equipment casing firmly with care to prevent parts from falling. Failure to comply may result in personal injuries.
- Store and transport the equipment based on the storage and transportation requirements. Failure to comply will result in equipment damage.
- Avoid storing or transporting the equipment in environments with water splash, rain, direct sunlight, strong electric field, strong magnetic field, and strong vibration.
- Avoid storing the product for more than 3 months. When the product needs to be stored for an extended period, take more strict protection and necessary inspection.
- Pack the equipment strictly before transportation. Use a sealed box for long-distance transportation.
- Never transport the equipment with other equipment or materials that may harm or have negative impacts on this equipment.

Installation

DANGER

- The equipment can only be operated by professionals with electrical knowledge. Non-professionals are not allowed to operate on the equipment.

WARNING

- Read through the guide and safety precautions before installation.
- Do not install the equipment in places with strong electric or magnetic fields.
- Before installation, check that the mechanical strength of the installation site can bear the weight of the equipment. Failure to comply will result in mechanical hazards.
- Do not wear loose clothes or accessories during installation. Failure to comply may result in an electric shock.
- When installing the equipment (such as controller) in a closed environment (such as a cabinet or casing), use a cooling device (such as a fan or air conditioner) to cool the environment down to the required temperature. Failure to comply may result in equipment over-temperature or a fire.
- Do not retrofit the equipment.
- Do not loosen the fixing bolts of the product parts and components.
- When the equipment (such as controller) is installed in a cabinet or final assembly, a fireproof enclosure providing both electrical and mechanical protections must be provided. The IP rating must meet IEC standards and local laws and regulations.
- Before installing devices with strong electromagnetic interference, such as a transformer, install a shielding device for the equipment to prevent malfunction.
- Install the product on an incombustible object such as metal and do not touch or attach the product to combustibles. Failure to comply can result in fire accident.



- Cover the top of the equipment (such as controller) with a piece of cloth or paper during installation. This is to prevent unwanted objects such as metal chippings, oil, and water from falling into the equipment and causing faults. After installation, remove the cloth or paper on top of the equipment to prevent over-temperature caused by poor ventilation due to blocked ventilation holes.

Wiring



- Equipment installation, wiring, maintenance, inspection, or component replacement must be performed only by professionals.
- Before wiring, disconnect all the power supplies of the equipment. Wait for at least the time designated on the equipment warning label before further operations because residual voltage still exists after power-off. Measure the direct voltage of the main circuit and ensure that the voltage is within the safety range. Failure to comply can result in an electric shock.
- Do not perform wiring, remove the equipment cover, or touch the circuit board with power ON. Failure to comply can result in an electric shock.
- Check that the equipment is grounded properly. Failure to comply can result in an electric shock.



- Do not connect the input power supply to the output side of the equipment. Failure to comply can result in equipment damage or even a fire.
- When connecting a drive to the motor, check that the phase sequences of the drive and motor terminals are consistent to prevent reverse motor rotation.
- Cables used for wiring must meet cross sectional area and shielding requirements. The shield of the cable must be reliably grounded at one end.
- Tighten the terminal screws with the tightening torque specified in the user guide. Improper tightening torque may result in overheat, damage to the connecting parts, or a fire.
- After wiring is done, check that all cables are connected properly and no screws, washers or exposed cables are left inside the equipment. Failure to comply may result in electric shock or equipment damage.



- Follow the electrostatic discharge (ESD) procedure and wear an anti-static wrist strap to perform wiring. Failure to comply may result in damage to the equipment or to the internal circuit.
- Use shielded twisted pairs for the control circuit. Connect the shield to the grounding terminal of the equipment for grounding purpose. Failure to comply can result in equipment malfunction.

Power-On



- Before power-on, check that the equipment is installed properly with reliable wiring and the motor can be restarted.
- Check that the power supply meets equipment requirements before power-on to prevent equipment damage or a fire.
- Do not open the cabinet door or protective cover of the product, touch any wiring terminal of the product, or remove any part of the product with power on. Failure to comply can result in an electric shock.



- After the completion of wiring and parameter setting, perform a robot test run to confirm that the robot can operate safely. Failure to comply may result in personal injuries or equipment damage.
- Before power-on, check that the rated voltage of the equipment is consistent with that of the power supply. Failure to comply may result in fire accident.
- Before power-on, check that no one is near the equipment, motor, or machine. Failure to comply may result in death or personal injuries.

Operation



- The equipment must be operated only by professionals. Failure to comply can result in death or personal injuries.
- Do not touch any connecting terminals or disassemble any unit or component of the equipment during operation. Failure to comply can result in an electric shock.



- Do not touch the equipment enclosure, fan, or resistor with bare hands. Failure to comply may result in burns.
- Prevent metal or other objects from falling into the equipment during operation. Failure to comply may result in a fire or equipment damage.

Maintenance



- Equipment installation, wiring, maintenance, inspection, or component replacement must be performed only by professionals.
- Do not maintain the equipment with power ON. Failure to comply can result in an electric shock.
- Before maintenance, cut off all the power supplies of the equipment and wait for at least the time designated on the equipment warning label.
- When a permanent magnet motor is used, do not touch the motor terminals immediately after power-off because the motor terminals can generate induced voltage during rotation even after the equipment is powered off. Failure to comply can result in an electric shock.



- Perform routine and regular inspection and maintenance on the equipment in accordance with maintenance requirements and keep a record.

Repair



- Equipment installation, wiring, maintenance, inspection, or component replacement must be performed only by professionals.
- Do not maintain the equipment with power ON. Failure to comply can result in an electric shock.
- Before inspection and repair, cut off all the power supplies of the equipment and wait for at least the time designated on the equipment warning label.



- Submit a request for repair services in accordance with the product warranty agreement.
- When the fuse is blown or the circuit breaker or earth leakage circuit breaker (ELCB) trips, wait for at least the time designated on the equipment warning label before power-on or further operations. Failure to comply may result in death, personal injuries or equipment damage.
- Troubleshooting and repair must be performed by professionals in accordance with the repair instructions, with repair records kept properly.
- Replace quick-wear parts of the equipment in accordance with the replacement instructions.
- Do not operate on damaged equipment. Failure to comply may result in death, personal injuries, or severe equipment damage.
- After the equipment is replaced, check the wiring and set parameters again.

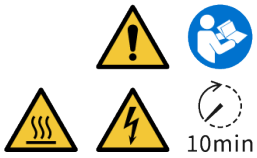
Disposal



- Dispose of retired equipment in accordance with local regulations and standards. Failure to comply may result in death, personal injuries, or property damage.
- Recycle retired equipment in accordance with industry waste disposal standards to avoid environmental pollution.

Safety Label

For safe operation and maintenance, comply with the safety labels on the equipment. Do not damage or remove the safety labels. See the following table for descriptions of the safety labels.

Safety labels	Description
	• Read through the safety instructions before operating the equipment. Failure to comply may result in death, personal injuries, or equipment damage. • Do not touch the terminals or remove the cover with power ON or within 10 min after power-off. Failure to comply can result in an electric shock.

Industrial Information Security

The product provides an interface to connect to the network and transmits data through the network interface. To protect factories, systems, machines, and networks from cyber attacks and ensure safe operation, a proper industrial information security protection mechanism must be implemented.
The customer is responsible for providing and continuously ensuring a secure connection between the product and the customer network or any other network to prevent unauthorized access to its factories, systems, machines, and networks. The system or machine can be connected to the corporate network or the Internet only when secure connections are ensured and appropriate security measures are taken (for example, anti-virus software or firewall installed).
Inovance continuously develops and improves products and solutions to improve safety. It is strongly recommended that you update the product promptly and always use the latest version.



Malware (such as viruses, Trojans, and worms) can put the product in an unsafe running state, which may result in death, serious injuries, and property damage. Observe the following precautions:

- Always use the latest software version. If the product version is no longer supported or the latest version is not used, customers will be exposed to increased risk of cyber attacks.
- Take appropriate protection measures (including but not limited to installing anti-virus software, firewall, WAF, IPS/IDS, and situation awareness system, and applying authentication, data encryption) to prevent files in mobile storage devices from being damaged by malware, and to prevent products, networks, systems, and interfaces from unauthorized access, interference, intrusion, data leakage, or information breach.
- Check all safety-related interfaces and settings after commissioning.

Contents

1 Overview 27
1.1 Product Introduction 27
1.2 Version Description 28
1.3 InoRobotLab Acquisition and Installation 32
1.3.1 How to get InoRobotLab 32
1.3.2 Installation Environment Requirements 32
1.3.3 Installing InoRobotLab 33
1.3.4 Updating InoRobotLab 35
1.4 InoRobotLab Interface 36
1.4.1 Main Interface 36
1.4.2 Quick Operation 36
1.4.3 Station 37
1.4.4 Work Pane 38
1.4.5 Information Pane 39
1.4.6 Status Bar 40
1.4.7 Control Panel 41
1.4.8 Menus 42
2 Basic Concepts 46
2.1 Classification of Robots 46
2.2 Frames 46
2.2.1 Joint Frame 47
2.2.2 Base Frame 47
2.2.3 World Frame 48
2.2.4 Tool Frame 48
2.2.5 Workobject Frame 49
2.2.6 Other Frames 50
2.3 Position Variables 50
2.3.1 Overview 50
2.3.2 Storage Format of Position Variables 50
2.3.3 Attributes of Position Variables 52
2.4 Offset 57
2.4.1 Offset Along the Current Tool Pose 57
2.4.2 Offset in a Cartesian Reference Frame 58
2.5 Interpolation and Transition 59
2.6 Singular Position 61
2.7 Motion Range and Interference Zone 62

3 Get Started 64
3.1 Programming Flowchart 64
3.2 Programming Examples 64
3.2.1 Introduction 64
3.2.2 Powering On the Robot System 65
3.2.3 Creating a Project 66
3.2.4 Connecting to the Controller 66
3.2.5 Editing the Main Task Program 67
3.2.6 Taking Points 68
3.2.7 Compiling the Project 70
3.2.8 Synchronizing to the Controller 71
3.2.9 Starting/Stopping the Debugging 72
3.2.10 Debugging the Breakpoint 73
3.2.11 Starting/Stopping Program Running 74
4 Basic Operations 75
4.1 Overview 75
4.2 Connecting/Disconnecting the Controller 77
4.3 User Login 79
4.3.1 User Login 79
4.3.2 Changing the Password 79
4.3.3 Retrieving the Password 80
4.4 Description of User Authorities 81
4.5 Switching Control Permission 85
4.6 Clearing Warnings and Alarms 87
4.7 Emergency Stop 87
4.8 Enabling/Disabling the Robot 88
4.9 Switching between Manual and Auto Modes 88
4.10 Configuring the Default Number of Axes Offline 89
4.11 Axis Operation 90
4.11.1 Coordination 90
4.11.2 Switching Mechanical Unit 93
4.11.3 Switching Frame 94
4.11.4 Switching Tool Number 95
4.11.5 Switching Workobject Number 95
4.11.6 Switching Load Number 96
4.11.7 Adjusting Speed 97
4.11.8 Axis Motion 97

4.11.9 Orientation Calibration 98
4.11.10 Homing 99
4.11.11 Mechanical Lock 100
4.11.12 Dynamic Brake 100
4.12 Viewing Current Position 101
4.13 Calibrating the Tools 103
4.14 Calibrating the Workobject 103
4.15 Robot Model Data Manager 103
4.16 Robot Software Upgrade 104
5 Project Management 105
5.1 Basic Concepts 105
5.1.1 Project 105
5.1.2 Task 107
5.1.3 Module 107
5.1.4 Function 108
5.2 Project Operations 110
5.2.1 Creating a Project 110
5.2.2 Opening a Local Project 112
5.2.3 Backing Up to Local Project 113
5.2.4 Backing Up the Current Project 114
5.2.5 Upgrading the Project 115
5.2.6 Saving the Project 118
5.2.7 Saving the Project As 119
5.2.8 Opening the Controller Project 120
5.2.9 Compiling the Project 121
5.2.10 Synchronizing to the Controller 122
5.2.11 Synchronizing the Project to Local 124
5.2.12 Renaming the Project 125
5.2.13 Deleting the Project 126
5.2.14 Exporting the Labels 126
5.2.15 Creating a File 127
5.2.15.1 Creating a Program File 127
5.2.15.2 Creating a Point File 129
5.2.16 Creating a Task 131
5.2.17 Setting the Task Type 133
5.2.18 Adding a Local File to the Project 134
5.2.18.1 Adding a Local Program File to the Project 134

5.2.18.2 Adding a Local Point File to the Project 136
5.2.19 Adding a Controller File to the Project 137
5.2.19.1 Adding a Controller Program File to the Project 137
5.2.19.2 Adding a Controller Point File to the Project 139
5.2.20 Upgrading a Single File 140
5.3 Program File Operations 140
5.3.1 Overview 140
5.3.2 Viewing the Point Data 141
5.3.3 Compiling 143
5.3.4 Saving Changes to the Program File 144
5.3.5 Saving the Program File As 146
5.3.6 Opening the Program file 147
5.3.7 Copying and Pasting the Program File 149
5.3.8 Renaming the Program File 150
5.3.9 Setting the Program File as Task 150
5.4 Point File Operations 152
5.4.1 Saving Changes to the Point File 152
5.4.2 Saving the Point File As 153
5.4.3 Copying and Pasting the Point File 153
5.4.4 Renaming the Point File 155
5.4.5 Deleting the Point File 155
6 Programming 157
6.1 Program Editing 157
6.1.1 Overview 157
6.1.2 Programming Rules 157
6.1.3 Syntax Help 158
6.1.4 Syntax Check 160
6.1.5 Shortcuts 161
6.1.6 Find and Replace 163
6.2 Point Variables Editing 165
6.2.1 Overview 165
6.2.2 Editing the Point Variables 166
6.2.3 Moving to the Selected Point 169
6.2.4 Using the Point Variables 170
6.3 Labels Editing 173
6.3.1 I/O Labels 173
6.3.2 AD Labels 174
6.3.3 BRD Labels 175

7 Debugging 176
7.1 Debugging Mode 176
7.2 Basic Debugging 176
7.2.1 Project Example 176
7.2.2 Starting and Stopping Debugging 179
7.2.3 Adding and Deleting Breakpoints 180
7.2.4 Enabling, Disabling, and Deleting All Breakpoints 181
7.2.5 Entering a Breakpoint 182
7.2.6 Running the Program Statement by Statement 183
7.2.7 Adding a Monitoring Variable 183
7.2.8 Modifying the Monitoring Variables 185
7.2.9 Viewing the Monitoring Variables 186
7.2.10 Returning to the Start Line 187
7.2.11 Setting the Start Line 188
7.2.12 Exiting Debugging 189
7.3 Multitasking Debugging 190
7.3.1 Task Management 193
7.3.2 Project Example 194
7.3.3 Displaying Multiple Tasks Calling the Same Function 197
7.3.4 Viewing Task Switches During Debugging 198
7.3.5 Entering a Breakpoint 198
7.4 I/O Monitor 199
7.4.1 Introduction of Robot Bus Addresses 199
7.4.2 Inputs 200
7.4.2.1 Overview 200
7.4.2.2 Bit Monitoring 201
7.4.2.3 Byte Monitoring 202
7.4.2.4 Word Monitoring 204
7.4.3 Outputs 205
7.4.3.1 Overview 205
7.4.3.2 Bit Monitoring 206
7.4.3.3 Byte Monitoring 207
7.4.3.4 Word Monitoring 209
7.4.4 System I/Os 210
7.4.5 Meaning of Colors in I/O Monitor 211
7.5 AD Monitor 212
7.6 DA Monitor 212

7.7 Global Variable Monitor 213
7.7.1 Overview 213
7.7.2 Adding a Monitoring Variable 213
7.7.3 Setting the Variable Value 216
7.8 Local Monitor 217
7.8.1 Overview 217
7.8.2 Adding a Monitoring Variable 217
7.8.3 Viewing the Monitoring Variables 219
7.9 Monitoring the Variable Using the Mouse 219
7.10 Connection Status Monitor 220
7.10.1 Device Connection Status 220
7.10.2 Bus Connection Status 222
7.11 Work Condition Monitor 223
7.12 Monitoring the Load Setting Value 232
8 Monitoring Permissions 234
8.1 Overview 234
8.2 Setting the Control Permission 238
9 Controller Parameter Configuration 243
9.1 Robot Settings 243
9.2 Zero Settings 243
9.2.1 Absolute Zero 243
9.2.2 Work Origin 244
9.2.2.1 Setting the Work Origin 244
9.2.2.2 Triggering Signal Output when Approaching Work Origin 246
9.2.3 Homing Calibration (Zero Point Calibration) 247
9.2.3.1 Zero Point Calibration for SCARA Robot 248
9.2.3.1.1 Preparation 248
9.2.3.1.2 Calibration Steps 249
9.2.3.1.3 Notes 252
9.2.3.2 Zero Point Calibration for 6-Axis Robot 252
9.2.3.2.1 Preparation 252
9.2.3.2.2 Calibration Steps 254
9.2.3.2.3 Notes 261
9.3 Installation Parameters 262
9.3.1 Tools 262
9.3.2 Workobject 270
9.3.3 Load 275
9.3.4 Arm Load 276

9.3.5 Installation Form 277
9.3.5.1 Installation Form Settings 277
9.3.5.2 Installation Form Detection 279
9.3.6 Environmental Parameters 280
9.3.7 Auto Warm-up 281
9.3.7.1 Steps 283
9.3.7.2 Notes 283
9.4 Peripheral Configuration 284
9.4.1 Bus Configuration 284
9.4.1.1 Switch Settings 284
9.4.1.2 Address Allocation Settings 290
9.4.1.3 I/O Mapping Settings 295
9.4.1.4 Project Number Configuration 297
9.4.1.5 Point File Configuration 297
9.4.2 IR-Link Settings 300
9.4.3 Mechanical Unit Configuration 304
9.4.4 Fieldbus Master Configuration 304
9.4.4.1 CANopen Master Configuration 304
9.4.4.1.1 Master Configuration 305
9.4.4.1.2 Slave Configuration 306
9.4.4.2 DeviceNet Master Configuration 313
9.4.4.2.1 Device Import 313
9.4.4.2.2 Configuration Settings 314
9.4.4.2.3 Master Configuration 314
9.4.4.2.4 Slave Configuration 315
9.4.4.2.5 Sync to/from Controller 316
9.4.4.2.6 Mutual Exclusion Between Bus Function and Arc Welding Plug-in 317
9.4.4.3 EtherCAT Master Configuration 317
9.4.4.3.1 Device Import 317
9.4.4.3.2 Master Configuration 318
9.4.4.3.3 Slave Status Management Function 319
9.4.4.3.4 Slave Configuration 320
9.4.4.3.5 Servo/Controller Compatibility Configuration 323
9.4.4.4 EtherNet/IP Master Configuration 324
9.4.4.4.1 Device Import 324
9.4.4.4.2 Configuration Settings 324
9.4.4.4.3 Master Configuration 325
9.4.4.4.4 Slave Status Management Function 325

9.5 Motion Parameters 326
9.5.1 Teaching Parameter Settings 326
9.5.2 Run Parameter Settings 327
9.5.3 Axis Parameter Settings 329
9.5.4 Collision Detection 330
9.5.5 Trajectory Recovery Settings 339
9.5.6 Advanced 341
9.5.6.1 Closed-loop Vibration Suppression 341
9.5.6.2 Self-learning Vibration Suppression 342
9.5.6.3 Torque Model Correction 348
9.5.6.4 Soft Float Function 349
9.6 Control Parameters 350
9.7 Workspace 351
9.7.1 Axis Limit Settings 351
9.7.2 Interference Zone Settings 351
9.8 System Settings 352
9.8.1 System Backup and Load 352
9.8.2 Control Permission 357
9.8.3 Flying Trigger I/O 357
9.8.4 Network Configuration 358
9.8.5 Time Settings 359
9.8.6 Program Quantity Settings 360
9.8.7 Start Button Work Mode 361
9.8.8 Line Number Reset Policy 363
9.8.9 Remote Stop Strategy 363
9.8.10 Robot Maintenance 364
9.8.11 Miscellaneous 371
9.9 Debug 372
9.9.1 Network Debug 372
9.9.2 COM Switch 372
9.9.3 System Diagnostics 373
9.9.3.1 Starting System Diagnostics and Exporting Diagnostic Results 373
9.9.3.2 Viewing Controller Status 375
9.9.3.2.1 Remote 378
9.9.3.2.2 Operation Log 378
9.9.3.2.3 Historical Alarm Log 379
9.9.3.2.4 Trajectory Information 380
9.9.4 Controller Universality 386

10 Communication Manager 388
10.1 Overview 388
10.2 Server Mode 388
10.2.1 Opening the Server 388
10.2.2 Closing the Server 389
10.2.3 Disabling the Connection 390
10.2.4 Refreshing the List 391
10.3 Client Mode 392
10.3.1 Creating a Connection 392
10.3.2 Disabling the Connection 393
10.3.3 Refreshing the List 394
11 System Resources 395
11.1 Vision Calibration Process 395
11.1.1 Overview 395
11.1.2 Vision Calibration of SCARA Robots 396
11.1.3 Vision Calibration of 6-Axis Robots 397
11.1.3.1 Steps 398
11.1.3.2 Result Verification 398
11.1.4 Eye-to-Hand Overlook Calibration 399
11.1.4.1 Installation Condition 399
11.1.4.2 Calibration Steps 399
11.1.4.3 Camera Communication Test 401
11.1.5 Eye-to-Hand Look Up Calibration 409
11.1.5.1 Installation Condition 409
11.1.5.2 Calibration Steps 410
11.1.6 Eye-in-Hand J2 Calibration 410
11.1.6.1 Installation Condition 411
11.1.6.2 Calibration Steps 411
11.1.7 Eye-in-Hand J4 Calibration 413
11.1.7.1 Installation Condition 414
11.1.7.2 Calibration Steps 414
11.1.8 Eye-in-Hand J5 Calibration 415
11.1.8.1 Installation Condition 415
11.1.8.2 Calibration Steps 416
11.1.9 Eye-in-Hand J6 Calibration 416
11.1.9.1 Installation Condition 416
11.1.9.2 Calibration Steps 417
11.1.10 Vision Process Calibration Examples and Result Verification 418

11.2 Conveyor Tracking 431
11.2.1 Overview 431
11.2.1.1 Workflow 431
11.2.1.2 Position Description System 432
11.2.1.3 Function Specifications 432
11.2.1.4 Configuration Flowchart 433
11.2.2 Hardware Configuration 433
11.2.2.1 Connecting the Encoder 433
11.2.2.2 Connecting the Photoelectric Sensor 434
11.2.2.3 Vision Device 434
11.2.3 Frame Settings 434
11.2.4 Parameter Settings 437
11.2.4.1 Basic Parameter Settings 438
11.2.4.2 Encoder Calibration 439
11.2.4.3 Workobject Height Calibration 441
11.2.4.4 Boundary Parameter Settings 441
11.2.4.5 Detection Parameter Settings 443
11.2.5 Notes 451
11.2.6 Application Cases 454
11.2.6.1 Dynamic Grasping 454
11.2.6.2 Dynamic Sorting 455
11.2.7 Auxiliary Functions 456
11.2.7.1 Accuracy Verification 456
11.2.7.2 One-Drive-Many Vision Function 458
11.2.7.3 Monitoring Interface 466
12 Function Applications 469
12.1 ECP Function 469
12.1.1 Overview 469
12.1.2 Notes 469
12.1.3 Calibration of the ECP Frame 469
12.1.4 Motion in ECP Mode 469
12.1.5 Jog Motion in ECP Mode 471
12.2 External Axis 472
12.2.1 External Axis Linked Motion 472
12.2.1.1 Prerequisite 472
12.2.1.2 Wiring 472
12.2.1.3 Configuring the Servo 473
12.2.1.4 Properties 474
12.2.1.5 Setting the Controller Parameters 474

12.2.1.6 Notes 477
12.2.2 External Axis Coordination 478
12.2.2.1 Prerequisite 479
12.2.2.2 Coordinated Mechanical Units 479
12.2.2.3 Setting Workobject for the Coordinated Mechanical Unit 500
12.2.2.4 Coordinated Motion Instructions 501
12.2.2.5 Precision of Coordinated Motion 502
12.2.3 External Axis Supported Servo Gantry Synchronization Function 503
12.2.3.1 Prerequisite 503
12.2.3.2 Wiring 503
12.2.3.3 Configuring the Servo 503
12.3 Optimal Trajectory 504
12.3.1 Debugging Steps 504
12.3.2 Application Example 505
12.4 Independent Axis 506
12.4.1 Overview 506
12.4.2 Description of Independent Axis Function 506
12.5 Load Auto-Tuning 513
12.5.1 Overview 513
12.5.2 6-Axis Robot Load Auto-Tuning 513
12.5.2.1 Opening Load Auto-Tuning 513
12.5.2.2 Tool Load Auto-Tuning 514
12.5.2.3 Grip Load Auto-Tuning 518
12.5.3 SCARA Robot Load Auto-Tuning 523
12.5.3.1 Opening Load Auto-Tuning 523
12.5.3.2 Tool Load Auto-Tuning 524
12.5.3.3 Grip Load Auto-Tuning 528
12.5.4 Notes 532
12.6 Collision Response 534
12.6.1 Overview 534
12.6.2 Plan to Stop 534
12.6.2.1 Description 534
12.6.2.2 Steps 534
12.6.3 Immediate Rollback 535
12.6.3.1 Description 535
12.6.3.2 Steps 535
12.6.3.3 Notes 535
12.6.4 Stop Immediately 536
12.6.4.1 Description 536

12.6.4.2 Steps 536
12.6.4.3 Notes 536
12.6.5 Priority of Actions Triggered by Instruction Settings and Interface Settings 536
12.7 Singularity Crossing 537
12.7.1 Steps 537
12.7.2 Notes 538
12.8 Unordered Matching 539
12.8.1 Functions and Operating Steps 540
12.8.2 Notes 542
12.9 Soft Float Function 543
12.9.1 Overview 543
12.9.2 Steps 543
12.9.2.1 Enabling the Soft Float Function 543
12.9.2.2 Disabling the Soft Float Function 543
12.9.2.3 Soft Follow Function 544
12.9.2.4 Soft Float Function Parameter Configuration 544
12.9.3 Notes 545
12.10 Free Curve 546
12.10.1 Overview 546
12.10.2 Steps 547
12.10.3 Configuration of Curve Parameters 547
12.10.4 Notes 550
12.10.5 Movs Instruction Usage Example 554
12.11 Interference Zone 554
12.11.1 Overview 554
12.11.2 Steps 555
12.11.3 Notes 562
12.12 Safety I/O Operation 563
12.12.1 Overview 563
12.12.2 Steps 563
12.13 Function Activation 566
12.13.1 Activating the Function 566
12.13.2 Generating the Activation File Using the Authorization Platform 567
12.14 Interrupt 570
12.14.1 Overview 570
12.14.2 Function Introduction 570
12.14.2.1 Registering an Interrupt 570
12.14.2.2 Special Notes 571
12.14.2.2.1 Parameter Class Limitations and Rules 571

12.14.2.2.2 Motion Instruction Processing Timing 571
12.14.2.2.3 Non-Motion Instruction Processing Timing 573
12.14.3 Instructions 573
12.14.4 Function Coupling 575
12.14.4.1 Breakpoint 575
12.14.4.2 Single Step 575
12.14.4.3 Set Start Line or Return to Start Line 575
12.14.4.4 Start/Stop Type 575
12.14.4.5 Project Synchronization 575
12.14.4.6 Trajectory Recovery 575
12.14.4.7 Function Type 576
12.14.5 Notes 576
13 Multi-language (Language Settings) 578
13.1 Overview 578
13.2 Multilingual Interface 578
13.2.1 Initial Setup on First Launch 578
13.2.2 Accessing the Function 578
13.2.3 Switching the Language 578
13.2.4 Importing the Language Pack 579
13.2.5 Downloading the Language Pack 582
13.2.6 Multilingual Guide Translation 582
14 Virtual Controller 583
14.1 Virtual Controller Applications 583
14.1.1 Overview 583
14.2 Virtual Controller Operation 583
14.2.1 Accessing the Virtual Controller 583
14.2.2 New Virtual Controller Wizard 584
14.2.2.1 System Name 585
14.2.2.2 Robot Selection 585
14.2.2.3 Summary 585
14.3 Recently Created Virtual Controller 586
14.3.1 Device Preview 586
14.3.2 Device Connection and Disconnection 587
14.3.3 Device Deletion 589
14.4 List of Supported Virtual Controller Functions and Instructions 590
14.4.1 List of Supported Functions 590
14.4.2 List of Supported Instructions 593
15 Predictive Maintenance 599
15.1 Overview 599

15.2 Software and Hardware Conditions 599
15.3 Usage Workflow 599
15.3.1 Motion Instruction Marking 600
15.3.2 Data Collection Configuration 600
15.3.3 Fault Warning Configuration 602
15.3.4 Fault Monitoring 602
15.3.5 Configuration for Continuous Running of the Functions 604
15.3.6 Configuration for Automatic Restart of the Functions 605
15.4 Quick Verification 606
16 Appendix 608
16.1 Historical Information Inquiry 608
16.1.1 Alarm History 608
16.1.2 Operation Log 609
16.2 Robot Alarms and Handling Method 610
16.3 Servo Debug 822
16.4 Simple Calculation of Load Parameters 836

1 Overview

1.1 Product Introduction

The robot system consists of human-machine interaction (HMI) software, robot controller, manipulator, etc. InoRobotLab is a human-machine interaction software that enables easy programming and commissioning of the robot. It can be used alone or in combination with the teach pendant that features easy operation and teaching, depending on the application.



You can use InoRobotLab for programming, debugging, state monitoring, and parameter setting of Inovance robots.

- It allows you to program the robots by text editing using the keyboard directly. The programming supports keywords, custom variables, custom functions, and labels. In addition, it supports the completion of code segments, secondary prompts, and the search and replacement of the contents of program files in the whole project, which effectively improves the programming efficiency.
- During debugging, InoRobotLab displays the current running line of each task and allows you to switch between open program files. The main tasks and dynamic tasks support breakpoint and statement-by-statement debugging. InoRobotLab supports adding monitoring variables during debugging, displaying the values of local and global variables, and also supports variable monitoring at mouse hover.
- In addition to variable monitoring, InoRobotLab also allows I/O monitoring, AD monitoring, global variable monitoring, connection status monitoring (device connection status, bus connection status), work condition monitoring, query (current values of tools, workobjects, and loads), and conveyor monitoring.
- InoRobotLab provides common functions, such as axis control panel, project management, task management, communication service management, servo debugging, etc.
- It also provides process modules including vision calibration and conveyor tracking. The process management supports the installation of process plug-ins.
- The controller parameter configuration provides robot settings, zero settings (absolute zero, work origin, homing calibration), installation parameters (tools, workobjects, loads, arm load, environment parameters, auto warm-up parameters), peripheral configuration (fieldbus configuration, IR-Link settings, mechanical unit settings, fieldbus master configuration, safety I/O action), motion parameters (teaching parameter settings, run parameter settings, axis parameter settings, collision detection, trajectory recovery settings, closed-loop vibration suppression, auto-learning vibration suppression, torque model correction, soft float), control parameters (servo parameter settings), workspace (axis limit settings, interference zone settings), system settings, debug, etc.

1.2 Version Description

Version number	Description
V4R24C4	[What's new] • Virtual controller. See the 14 Virtual Controller section. • EtherNet/IP master. See the 9.4.4.4 EtherNet/IP Master Configuration section. [What's enhanced] • Language. See the Multi-language (Language Settings) section. • Maintenance. See the 9.8.10 Robot Maintenance section. • External axis. See the 12.2 External Axis section. • EtherCAT. See the 9.4.4.3 EtherCAT Master Configuration section. • Diagnostic function. See the 9.9.3 System Diagnostics section. • Slave. See the 9.4.4.3.3 Slave Status Management Function section. • Emergency stop mode. See the Emergency Stop Mode section. • Load auto-tuning. See the 12.5 Load Auto-Tuning section. • Interrupt. See the 12.14 Interrupt section.
V4R24C3	[What's new] • EtherCAT master. See the 9.4.4.3 EtherCAT Master Configuration section. • DeviceNet master. See the 9.4.4.2 DeviceNet Master Configuration section. • Collision response function. See the 12.6 Collision Response section. • Robot maintenance function. See the 9.8.10 Robot Maintenance section. • Interrupt. See the 12.14 Interrupt section. [What's enhanced] • Added more external axis types. See the 12.2 External Axis section. • The allowable number of programs has been expanded to 256. See the 9.8.6 Program Quantity Settings section. • Load auto-tuning function. See the 12.5 Load Auto-Tuning section. • System diagnosis viewing function. See the 9.9.3 System Diagnostics section.

Version number	Description
V4R24C2	[What's new] • One-drive-many vision function for tracking process. See the 11.2 Conveyor Tracking section. • System diagnosis viewing function. See the 9.9.3 System Diagnostics section. • Trajectory recovery mode and adjustable threshold. See the 9.5.5 Trajectory Recovery Settings section. • Custom function return value. See the 5.1.4 Function section. [What's enhanced] • Increased the number of tasks. See the 5 Project Management and Debug sections. • Optimized the interference zone. See the 12.11 Interference Zone section. • Optimized remote I/O control permission. See the 4.5 Switching Control Permission and 9.8.2 Control Permission sections. • Optimized the work origin function. See the 9.2.2 Work Origin section. • Optimized the self-learning vibration suppression function. See the 9.5.6.2 Self-learning Vibration Suppression section.
V4R24C1	[Version Introduction] In the V4R24C1 version, the soft float function and free curve function are the key updated functions. Additionally, the multi-point recipe function, plug-in function, and TCP communication optimization are also the changes in this version. [What's new] • Soft float function. See the 9.5.6.4 Soft Float Function section. • Free curve function. See the 12.10 Free Curve section. • Multi-point recipe function. See the 5.4 Point File Operations section. • Plug-in function. See the "Plug-in Management" section in the "Teach Pendant User Guide" [What's enhanced] • TCP Communication. See the 10 Communication Manager section.  See the detailed description of each function.

Version number	Description
S03.23R	[Version Introduction] The S03.23R version is fully compatible with the S03.22R version in terms of projects. In this version, singularity crossing function, load auto-tuning function, move back function, independent axis, and unordered matching of the manipulator and controller are the key updated functions. [What's new] • Singularity crossing. See the 12.7 Singularity Crossing section. • Load auto-tuning. See the 12.5 Load Auto-Tuning section. • Independent axis. See the 12.4 Independent Axis section. • Unordered matching of robot and controller. See the 12.8 Unordered Matching section. • String API. See the "String Call" section in the "Teach Pendant User Guide". • Pulse instructions. See the "Pulse" section in the "Industrial Robot Instruction Guide". • Auto warm-up. See the 9.3.7 Auto Warm-up section. [What's enhanced] • Dynamic tracking. See the 11.2 Conveyor Tracking section. CAUTION See the detailed description of each function.

S03.22R	[Version Introduction] In the S03.22R version, the external axis and ECP functions are the key updated functions. To accommodate the introduction of these key functions, a new point system has been implemented. With the implementation of the new point system, the project structure also needs to change accordingly, including tools, workobjects, and loads no longer being treated as project resources but as system parameters, with certain adjustments made to the parameter settings and calibration methods for tools and workobjects. In version S03.22R, we have also introduced usability enhancements, such as find and replace across the entire project and variable monitoring at the mouse pointer on the PC platform. In addition, this version features physical I/Os increased to 128, support for mm/s speed unit, fast pickup during tracking process, and temperature rise protection. Upgrading from previous versions (19/20/21) to S03.22R requires a full device flashing. [What's new] • External Axis. See the 12.2 External Axis section. • ECP. See the 12.1 ECP Function section. • InoRobotLab software supports variable monitoring via mouse pointer. See the 7.9 Monitoring the Variable Using the Mouse section. • Added support for mm/s speed unit. See the "Industrial Robot Instruction Guide". • Temperature rise protection function. See the 7.11 Work Condition Monitor section. • InoRobotLab supports the update function. Go to "Tools" > "Robot software update" (Administrator permission is required. The steps are the same as those of InoRobUpdater software. Press F1 after entering the update wizard.) [What's enhanced] • Find and replace throughout the entire project. See the 6.1.6 Find and Replace section. • Increased physical I/Os to 128 • Back up the data before update. You can import and restore the backup file, including the following items: ◦ Project ◦ System backup ◦ Configuration of bus communication ◦ Backup of self-learning vibration suppression files • The project can be updated to version 22 using InoRobotLab software, but the following precautions must be taken: ◦ After the upgrade, there will still be a small number of compiling errors. You can correct them according to the compilation error information. (Tool is required for motion instructions.) ◦ Tools, workobjects, and loads parameters must be manually selected and synchronized to the controller. ◦ The MovAbsJ instruction is required to move the joint to the joint point (JP).

Version number	Description
	◦ After the upgrade, some points need to be reconfirmed and some points need to be corrected. • After update to version 22. If the following configurations are used, you need to manually reset them ◦ I/O mapping ◦ Vision calibration ◦ Conveyor tracking

1.3 InoRobotLab Acquisition and Installation

1.3.1 How to get InoRobotLab

You can get the InoRobotLab software via the following methods:

- Download the software installation package for free from the "Service and Support > Download Center" page on the Inovance Technology official website (www.inovance.com). Both x86 and x64 versions are available for selection. For users with a Windows x64 system, we recommend using the x64 version.
- Get the software installation package from the Inovance agents.

1.3.2 Installation Environment Requirements

Hardware requirements

Items	Minimum configuration	Recommended configuration
CPU	-	Intel Core i3 or higher and its equivalents
Memory	4 GB	8 GB and above
Hard disk capacity	60 GB (more than 5 GB free space)	200 GB and above
Resolution	1024 x 768	1920 x 1080
GPU	Supports OpenGL2.0	-

Software requirements for InoRobotLab that requires no installation

Item	Requirement
Operating system	Windows 7 and later
.Net Framework	.Net Framework 4.7.2 and later

Item	Requirement
VC++ Runtime	Microsoft Visual C++ 2008 Redistributable SP1 (x86 or x64) Microsoft Visual C++ 2010 Redistributable (x86 or x64) Microsoft Visual C++ 2012 Redistributable (x86 or x64) Microsoft Visual C++ 2015-2022 Redistributable (x86 or x64)
Others	Microsoft Excel Application

Software requirements for InoRobotLab that requires installation

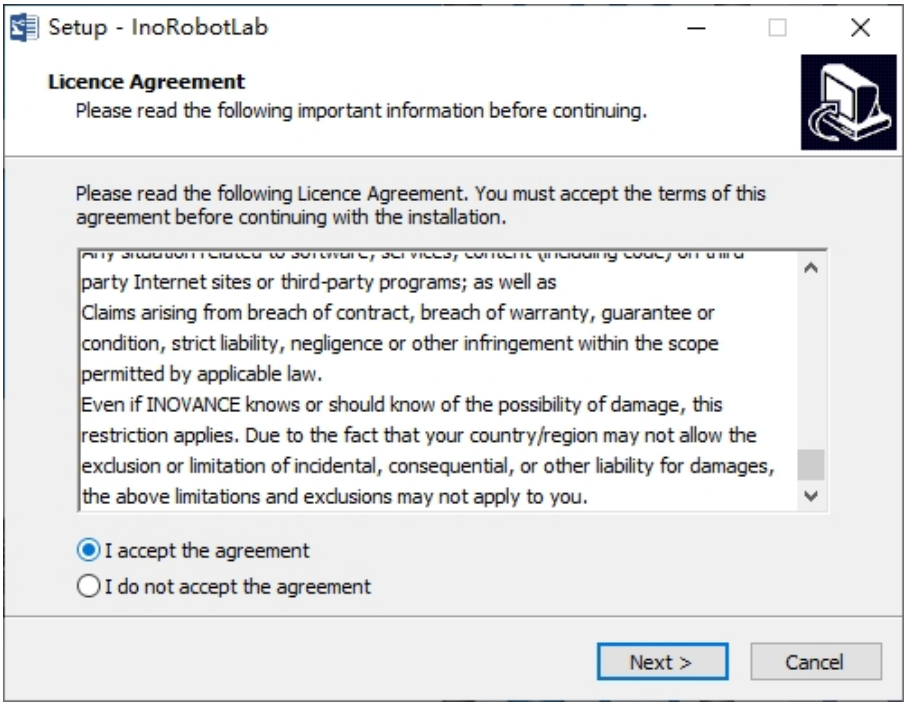
Item	Requirement
Operating system	Windows 7 and later
Others	Microsoft Excel Application

1.3.3 Installing InoRobotLab

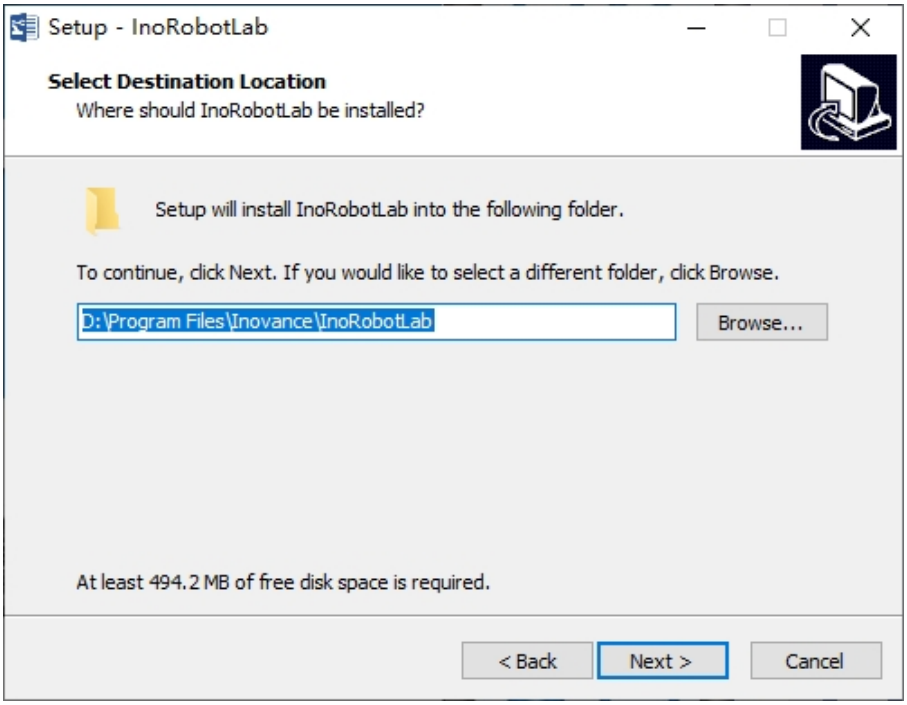
Before using InoRobotLab, install .Net Framework 4.7.2 and later.

After installing .Net Framework 4.7.2 and later, run InoRobotLab.exe in the software directory. (For example, the EXE file of version V4R24C4 is "InoRobotLabSetUp_ V4R24C4.exe". If the InoRobotLab software has been installed before, you will be prompted to uninstall the previous version.)

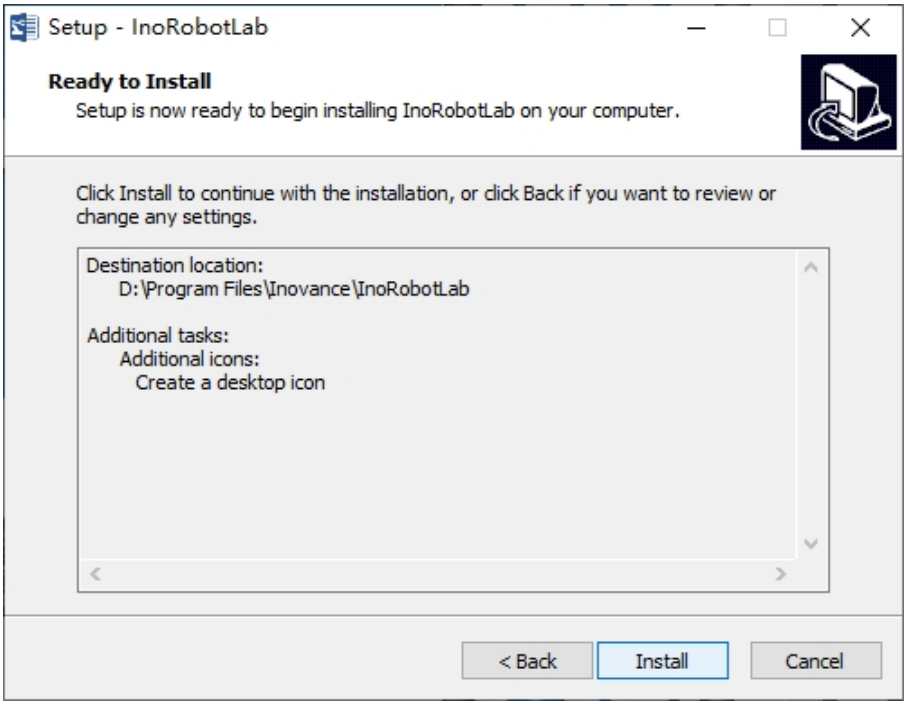
1. Run the InoRobotLabSetup.exe installer to enter the License Agreement page.



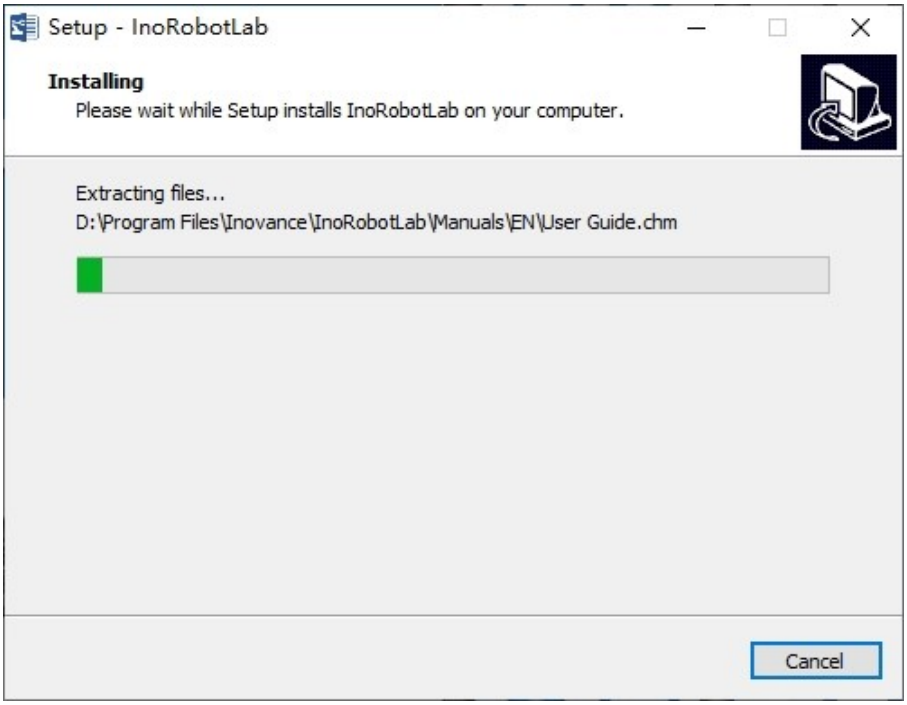
2. Select "I accept the agreement" and click "Next" to go to the "Select installation location" page.



3. The installer detects whether .NET Framework 4.7 is installed on the system. If not, .NET Framework 4.7 will be installed first. After installation of .NET Framework 4.7, select the installation directory of InoRobotLab, and then click "Next".



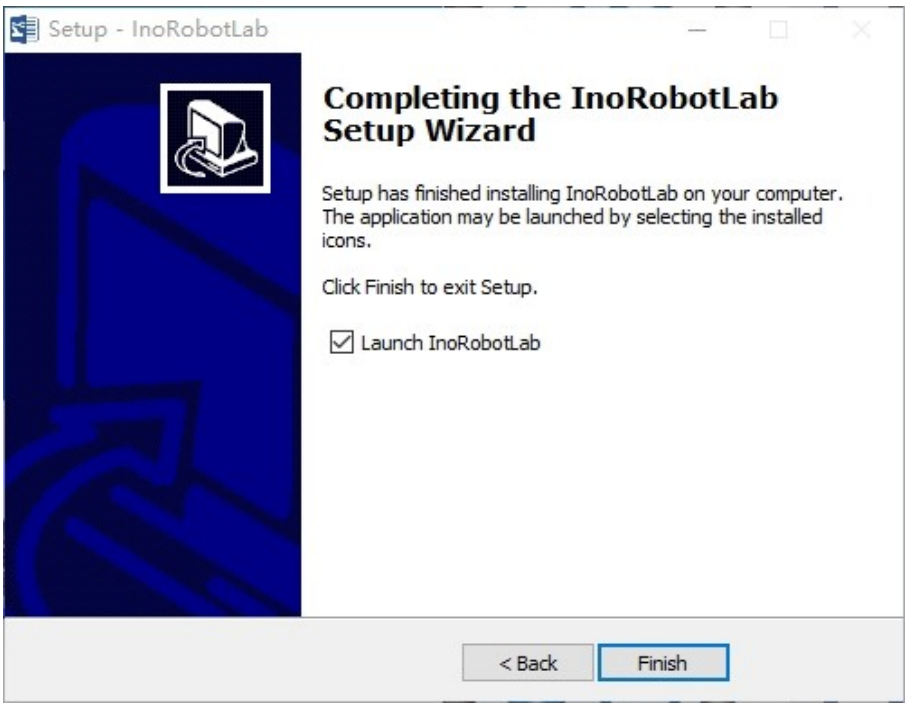
4. Click the "Install" button.



CAUTION

The installation takes a while. Please wait for the installation to complete.

5. Click "Finish" to complete the installation of InoRobotLab software.



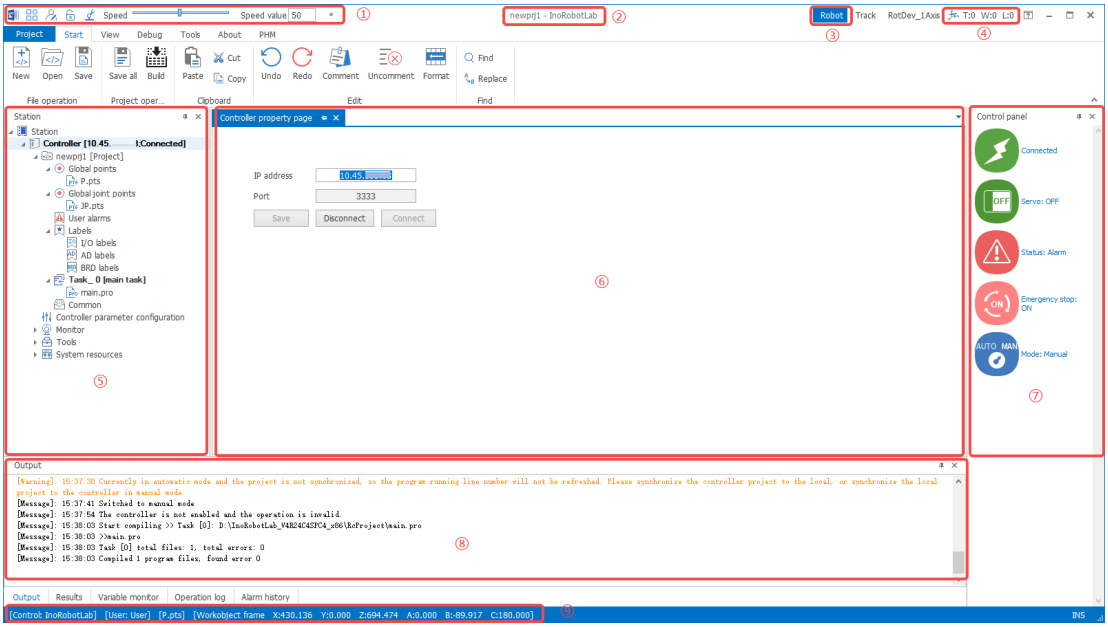
1.3.4 Updating InoRobotLab

To update InoRobotLab, uninstall the current version and then install the new version.

1.4 InoRobotLab Interface

1.4.1 Main Interface

The following figure shows the main interface of InoRobotLab.



No.	Description
①	Quick access toolbar
②	Current project path
③	Mechanical unit
④	Current frame
⑤	Station
⑥	Work pane
⑦	Control panel
⑧	Information pane
⑨	Status bar

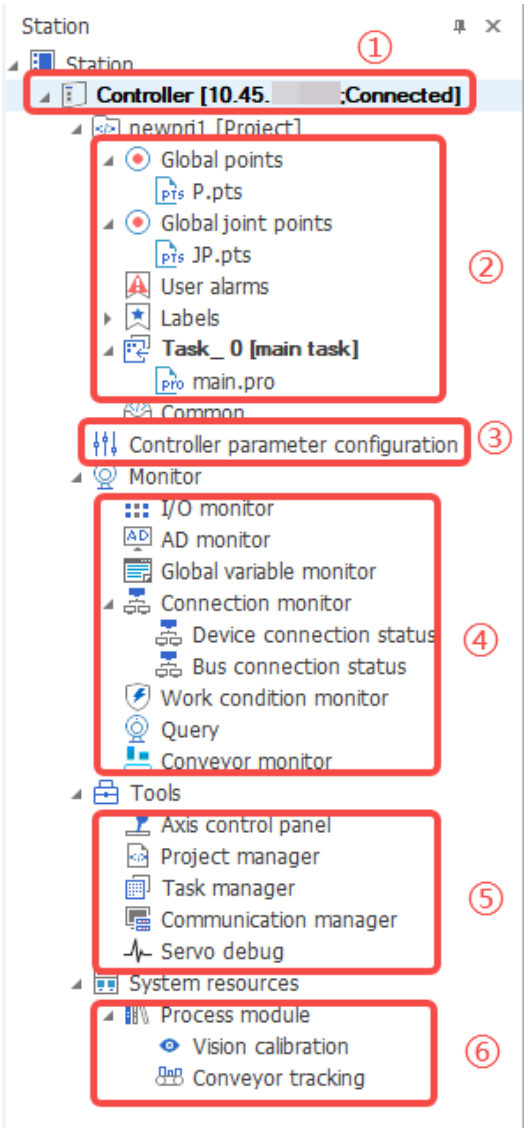
1.4.2 Quick Operation

Item	Description
	Minimizes or maximizes the window, closes the window etc.
	Opens the start page where you can create a project, open a project, import a project from the controller, or open the recently opened projects.
	Switches the user permission.
	Switches the control permission.

Item	Description
	Sets the default number of robot axes in offline status.

1.4.3 Station

The station adopts a tree structure including many operable nodes. Double-click on a node to open the corresponding page; or right-click on the node to open the right-click menu. The nodes on the tree control are the entry points for many operations. The tree structure is divided into six parts, as shown below.



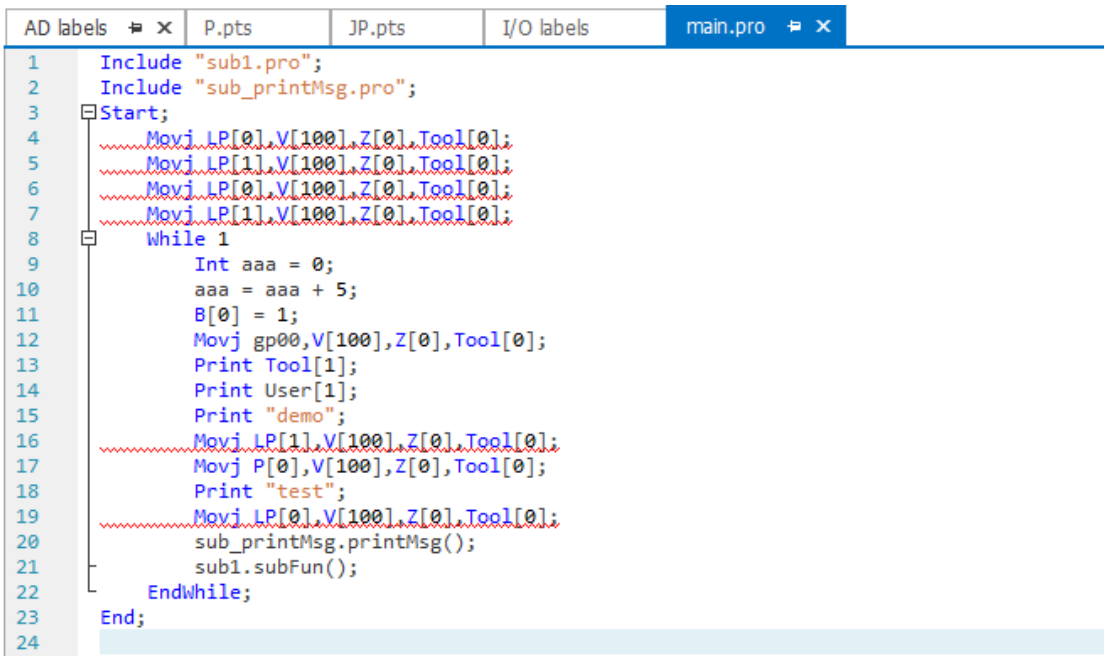
No.	Name	Description
①	Controller	• Displays the IP address of the target robot controller and its connection status. By right-clicking on "Controller", you can modify the IP address of target controller, connect to or disconnect from the controller. • Backs up some parameters (tools, workobjects, and loads) of the current controller to the local project. • Backs up the current project, that is, the project currently used in the controller to a local path.

No.	Name	Description
②	Project	Robot project, including global points file (P.pts by default), global joint points file (JP.pts by default), user-defined alarms, labels, task configuration information, and common programs. The project is the most frequently operated node by users. The series of operations on the project are called project management. The project facilitates programming, debugging, and operation of the robot, as well as program porting.
③	Controller parameter configuration	Enables setting, backup and restoring of the controller parameters, and upgrade of the system.
④	Monitor	Monitors the controller status.
⑤	Tools	Includes axis control panel, project manager, task manager, communication manager, and servo debug.
⑥	System resources	Offers access to the process modules.

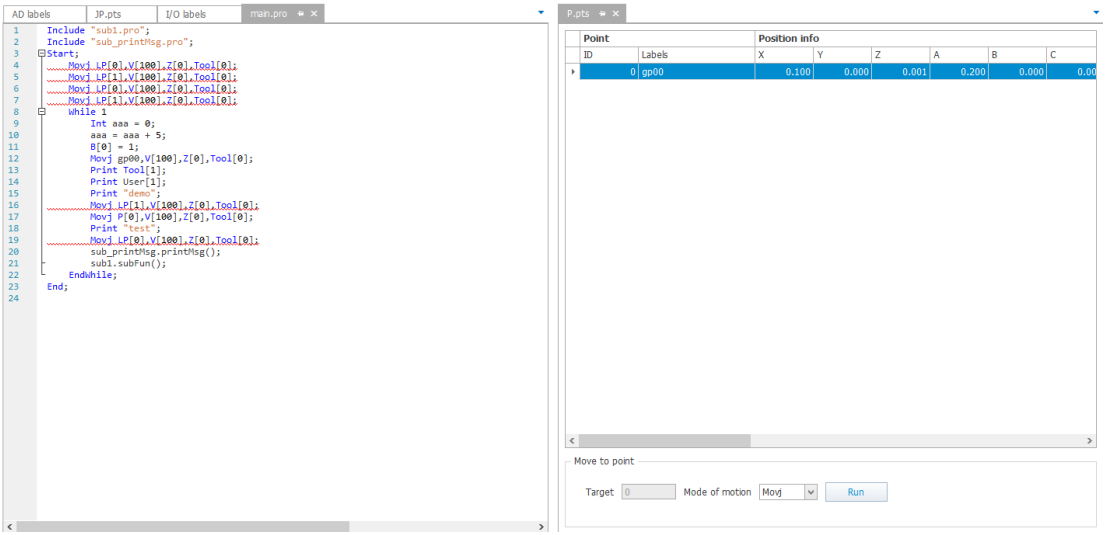
1.4.4 Work Pane

With TabbedView, most open pages can be embedded in this area as tabs. You can overlap these tabs, or group them horizontally or vertically. You can also float a tab to facilitate operation. To change the tab layout, right-click on the tab.

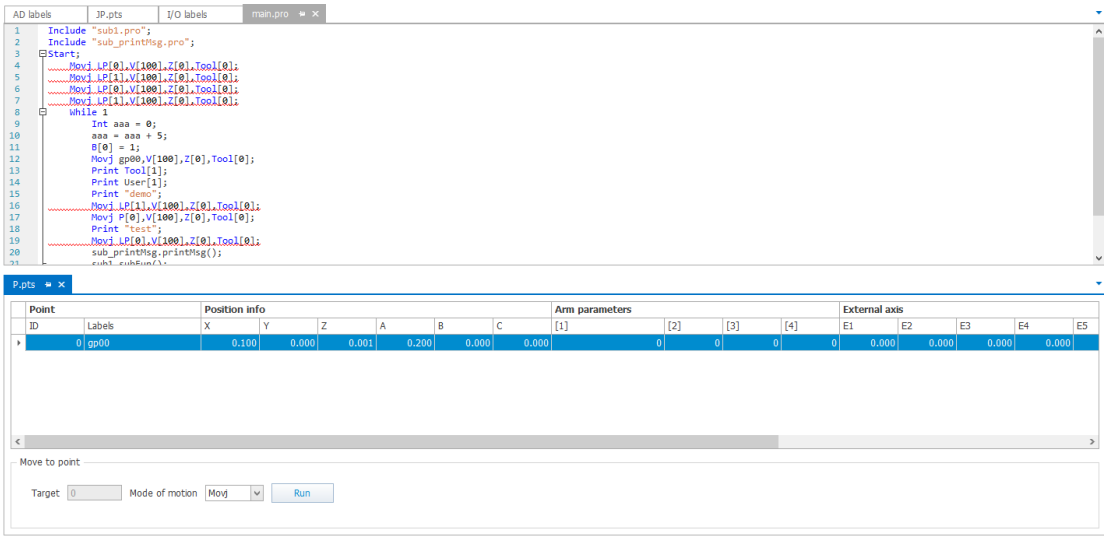
- Overlapping mode



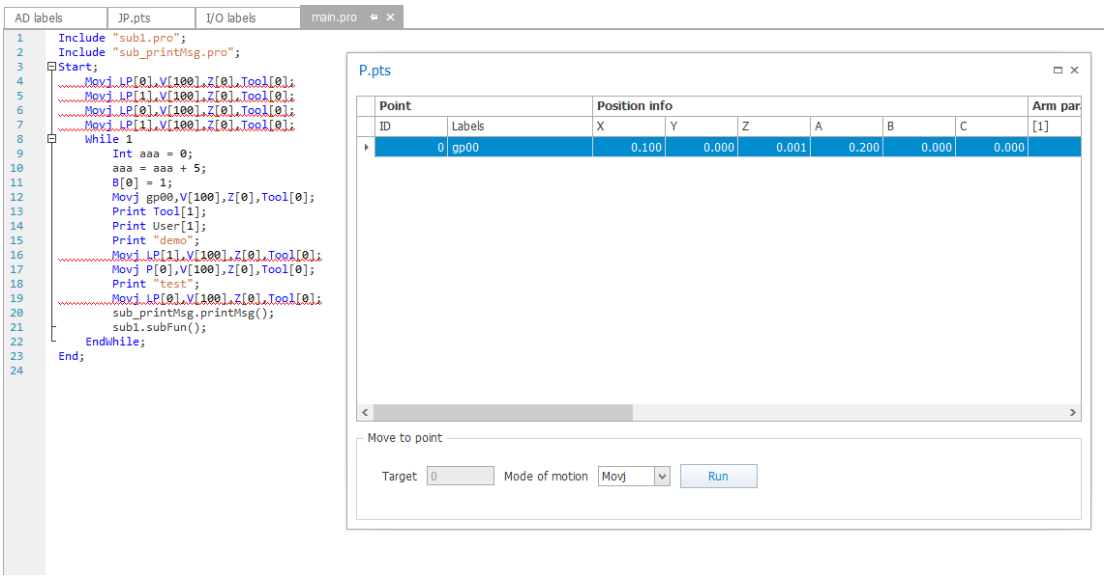
- Vertical group mode



- Horizontal group mode

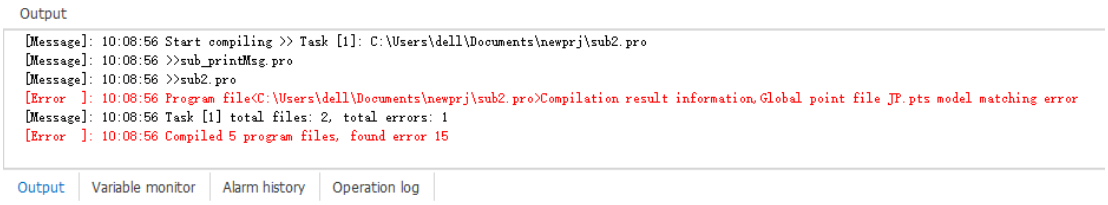


- Float mode



1.4.5 Information Pane

The information pane displays various types of information to the user. The information pane consists of five pages, as shown below.



Page Name	Description
Output	Outputs operation result information, exception information, state change, compilation information, controller warnings, alarms, debug information, and prompt information.
Results	Outputs the search result information of the program file.
Variable monitor	Monitors the variable values via a tree structure in the debug mode.
Alarm history	Displays historical alarms.
Operation log	Displays the operation log.

1.4.6 Status Bar

Displays the current basic status information, as shown below.

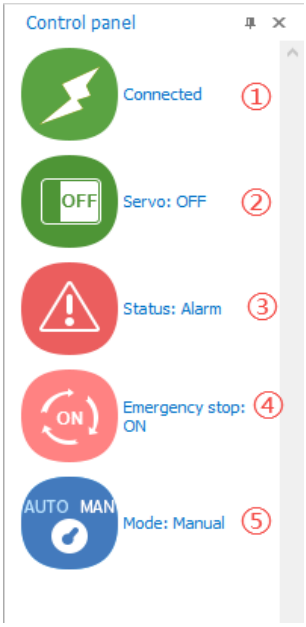


No.	Status name	Description
①	Debug/Auto	Displays the status of the main task or dynamic task during debug or auto mode. When a main task exists, the status of the main task is displayed; when only a dynamic task exists, the status of the dynamic task is displayed: • Run: The main task or dynamic task is running. • Stop: The main task or dynamic task is stopped. • End: The main task or dynamic task is ended. • Inactive: The main task or dynamic task is inactive. • Not synced: Displayed only in auto mode when the project is not synchronized. The display of the line number is not refreshed in this case. Displays the current debug or auto status. Not available when not in debug or auto mode.
②	Breakpoint	When a breakpoint is reached, the status is displayed in orange color. If a breakpoint is not reached, the status is not displayed.
③	Control permission	Displays the current control, including InoRobotLab, Remote Ethernet client, Remote I/O unit (effective in both auto and manual modes), and Remote I/O unit (only effective in manual mode).
④	User	Displays the current user authority, including User, Admin, and Manufacturer.

No.	Status name	Description
⑤	Safety door	Displays the safety door status in orange when the safety door is triggered. Not available when the safety door is not triggered.
⑥	Mechanical lock	Displays the mechanical lock status in orange when the mechanical lock is triggered.
⑦	Point file name	Name of the currently loaded point file
⑧	Frame and position information	Current frame type and position information
⑨	Cursor position	Position of the cursor during editing of the current program file
⑩	INS/OVR	Displays the edit form during program file editing, including insert or override.

1.4.7 Control Panel

The control panel controls and displays the basic status of the robot. See the figure below:



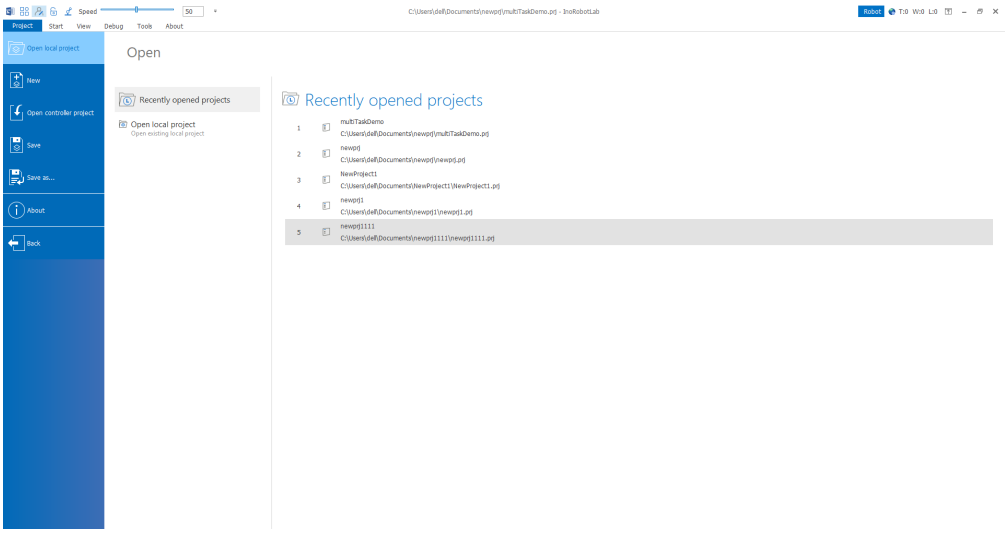
No.	Name	Description	Icon
①	Connect or disconnect	Displays the connection status.	Connected:  Disconnect: 
②	Servo status control	Servo status control and display	Servo OFF:  Servo ON: 

No.	Name	Description	Icon
③	Status	Warning/Alarm status display, and clears alarms and warnings	Normal:  Warning:  Alarm: 
④	Emergency stop	Turns on or off emergency stop, and displays emergency stop status.	Emergency stop OFF:  Emergency stop ON: 
⑤	Switch mode	Manual/Auto	Manual:  Auto: 

1.4.8 Menus

Project

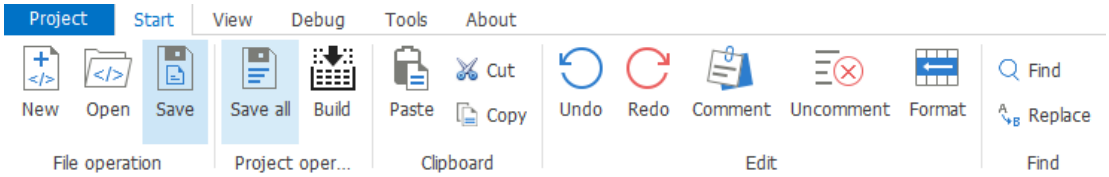
Click "Project" on the main page, as shown in the figure below:



- Project page: You can operate the project with "Open local project", "New", "Open controller project", "Save", "Save As".
- About: You can view the software version information.
- Return: Returns to the display of the main page.

Start

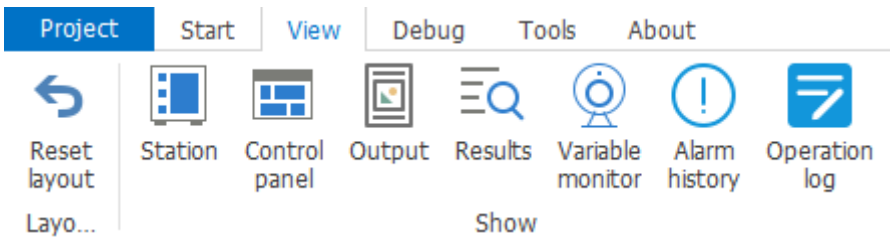
On the main page, click "Start". The "Start" menu includes menu items related to program file editing, as shown in the figure below.



Function	Shortcuts	Description
New	Ctrl+N	Creates a program file.
Open	Ctrl+O	Opens a local program file.
Save	Ctrl+S	Saves the current file or document, such as .pro and .pts.
Save All	Alt+S	Save all modified program files or documents, such as .pro and .pts.
Paste	Ctrl+C	Pastes the text or point variables from clipboard to the cursor position during program or point editing.
Cut	Ctrl+X	Cuts the selected text or point variables during program or point editing.
Undo	Ctrl+Z	Undoes the last editing in the code editor.
Redo	Ctrl+Y	Redoes the last editing you have undone.
Comment	Ctrl+#	Comments the cursor line by default, or lines selected.
Uncomment	Alt+#	Cancels comments of the cursor line by default, or lines selected.
Format	Ctrl+K	Aligns the program lines in the code editor.
Find	Ctrl+F	Opens the "Find" window.
Replace	Ctrl+R	Opens the "Replace" window.

View

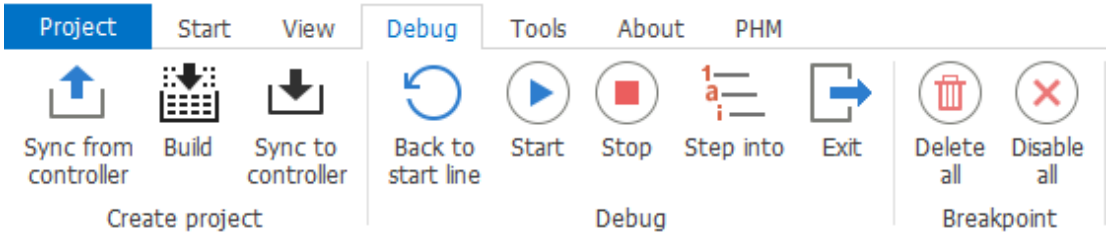
On the main page, click "View". The View menu allows you to switch among open windows on the main page, as shown in the figure below.



- Reset layout: Resets the layout to the default. (The current layout is saved automatically when you close the software, and the previous layout is loaded when you open the software again.)
- Station: Displays the "Station" window.
- Control panel: Displays the "Control panel" window.
- Output: Displays the "Output" window.
- Results: Displays the "Results" window.
- Variable monitor: Displays the "Variable monitor" window.
- Alarm history: Displays the "Alarm history" window.
- Operation log: Displays the "Operation log" window.

Debug

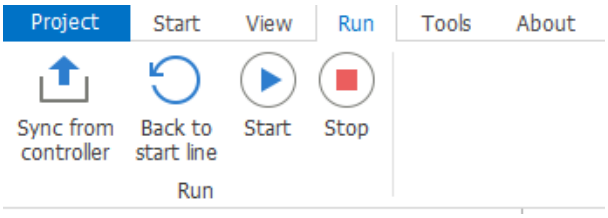
On the main page, click "Debug". The "Debug" menu includes menu items related to project debugging (which are automatically hidden when in auto mode), as shown in the figure below.



Function	Shortcuts	Description
Sync from controller	-	When connected to the controller, uploads the currently open controller project to local and opens it.
Build	F6	Compiles the current project.
Sync to controller	-	Synchronizes the currently open project to the controller. The project is automatically saved and compiled before synchronization.
Back to start line	F2	Returns the main task and dynamic task to the start line and enters the debug mode.
Start	F5	Starts debugging from the start line of the task. If not in debug mode, the system enters the debug mode.
Stop	Shift+F5	Stops debugging. In this case, you can debug step by step.
Step-into	F1	Single-step execution
Exit	Esc	Exits the debug mode.
Delete all	Ctrl+Shift+F9	Deletes breakpoints set in all program files.
Disable all	Alt+F9	Disables breakpoints set in all program files. Available only when breakpoints are enabled.

Run

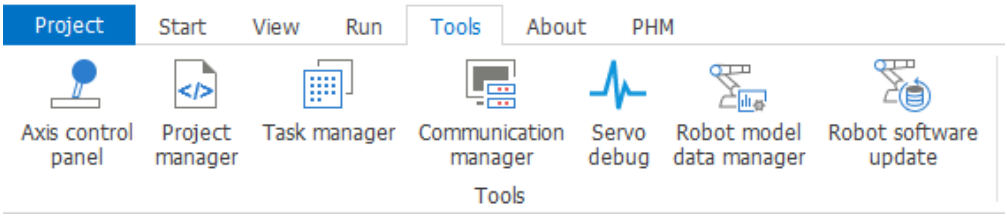
On the main page, click "Run". The "Run" menu includes menu items related to operations in auto mode (which are automatically hidden when in manual mode), as shown in the figure below.



Function	Shortcuts	Description
Sync from controller	-	Synchronizes project to local
Back to start line	-	Returns the main task and dynamic task to the start line. Effective only when the project has already been synchronized and the task is not running. You can view the task status in the status bar.
Start	F5	Starts robot operation.
Stop	Shift+F5	Stops robot operation. After stop, you can click "Start" to resume running, or "Back to start line" to return to the start line.

Tools

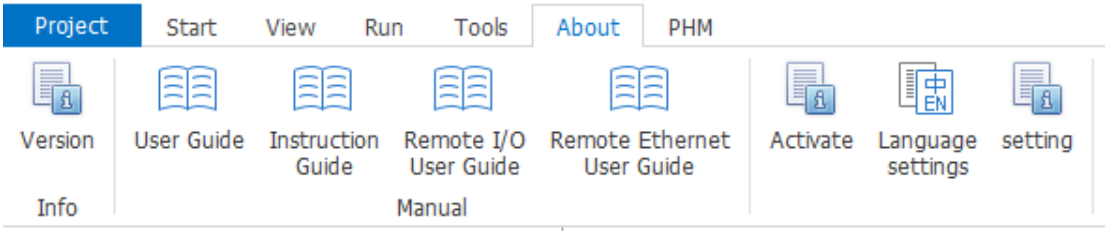
On the main page, click "Tools". The "Tools" menu allows you to switch among open windows on the tools, as shown in the figure below.



- Axis control panel: Opens the window for manually controlling the robot.
- Project manager: Opens the window for managing user programs and projects.
- Task manager: Opens the window of task management and task status monitoring.
- Communication manager: Opens the window of network communication management and status monitoring for external devices.
- Servo debug: Opens the window of servo debugging software.
- Robot model data manager: Opens the window of robot model and data management.
- Robot software update: Opens the window of robot controller and servo software update.

About

On the main page, click "About", as shown in the figure below.



- Version: Opens the window of version information.
- User Guide: Opens the window of "InoRobotLab Industrial Robot User Guide".
- Instruction Guide: Opens the window of "Industrial Robot Instruction Guide".
- Remote I/O Guide: Opens the window of "Industrial Robot Remote I/O Control Function User Guide".
- Remote EtherNet Guide: Opens the window of "Industrial Robot Remote EtherNet Control Function User Guide".
- Activate: Opens the window of optional module activation.
- Language settings: Opens the window of robot language settings.
- Settings: Opens the window of software minimization and connection lock interface settings.

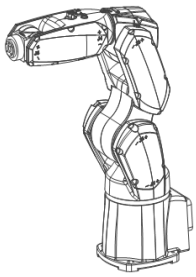
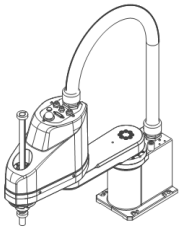
Shortcut F1

- When you are editing the program, you can press F1 to open the instruction online help. Search is performed automatically based on the key string included in the statement.
- When result is found, you will be directed to the corresponding description in the online help. If it fails to automatically direct to the corresponding position, you can try double-clicking on an instruction with the mouse and then pressing F1.

2 Basic Concepts

2.1 Classification of Robots

Robots can be classified based on the number of axes, serial/parallel characteristics, etc. The following table lists several common robots:

Name	Type	Number of axes	Serial/ Parallel	Description	Application
6-axis robot		6	Serial	Extremely flexible and suitable for working with virtually any trajectory or angle.	Loading, painting, measuring, arc welding, spot welding, packaging, assembly, forging, casting, etc.
SCARA robot		4	Serial	Lightweight structure and fast response	Mechanical assembly, material distribution, etc., assembly, labeling, patching, dispensing, etc. in 3C industry.

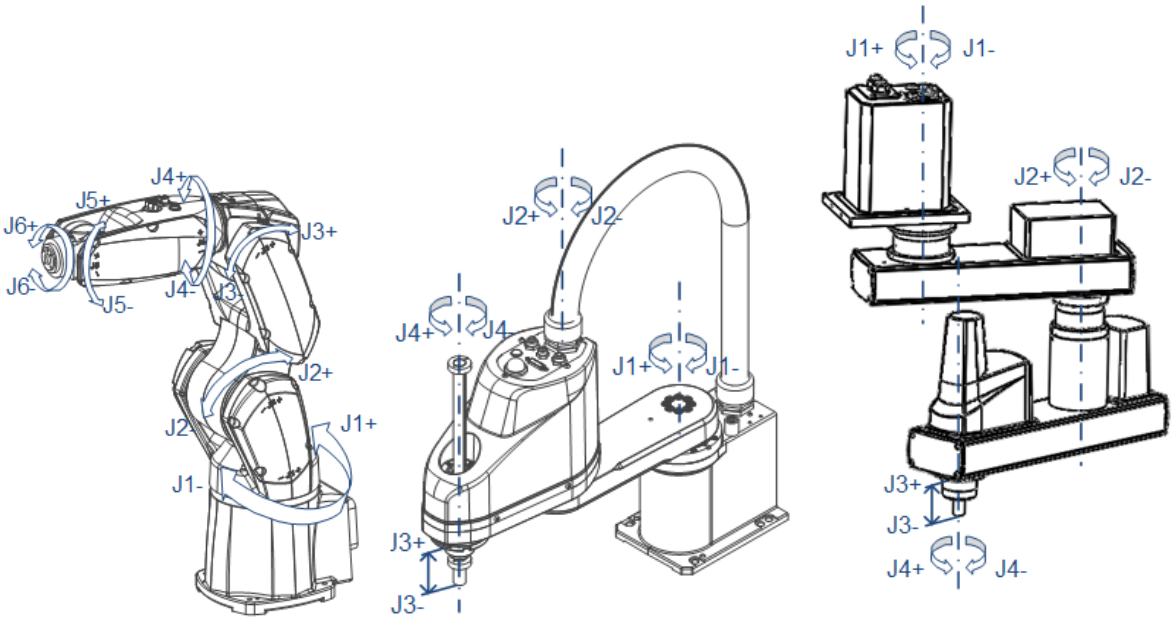
2.2 Frames

Five frames are available in Inovance robot system.

Frame No.	Frame name	Definition
1	Joint frame	A frame defined at each joint of the robot, which is a direct description of the motion of the robot joints.
2	Base frame	A frame defined on the base of the robot, often used as a reference for motion.
3	World frame	The absolute coordinate of the system and a reference position of other frames.
4	Tool frame	A frame defined on the tool, which can be customized.
5	Workobject frame	A frame defined on the workobject, which can be customized.

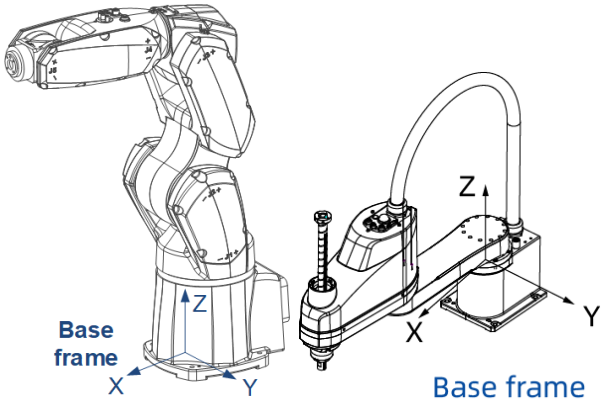
2.2.1 Joint Frame

The joint frame is located at the joints of the robot.



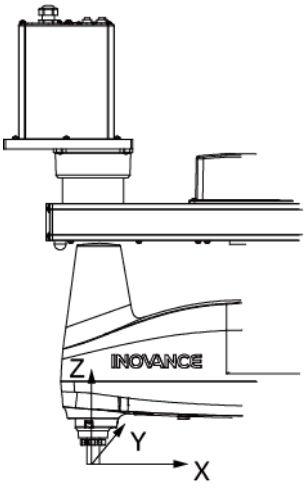
2.2.2 Base Frame

The base frame is also called the robot frame and is generally located at the base of the robot.



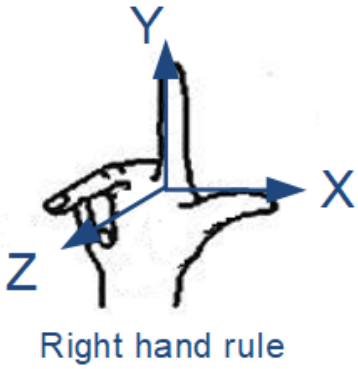
Base frame

The base frame of inverted SCARA robot is located on the flange at the zero point and does not move with the flange.



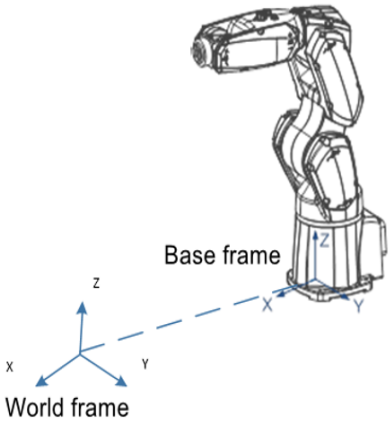
Base frame

The base frame is a Cartesian frame. (The world, tool and workobject frames are also Cartesian frames.) The X and Z directions can be determined first, and then the Y direction can be determined by the right-hand rule.



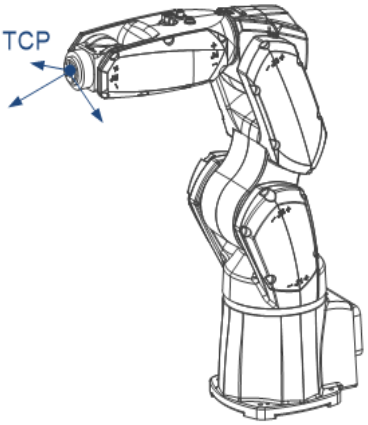
2.2.3 World Frame

The world frame does not depend on other references. It is often defined as the absolute origin and used as the reference position of other frames. When the robot is not fixed (i.e., mounted on the guide rail), the world frame cannot be coincident with the base frame. It is often necessary to describe the position of the robot in the base frame relative to the world frame.

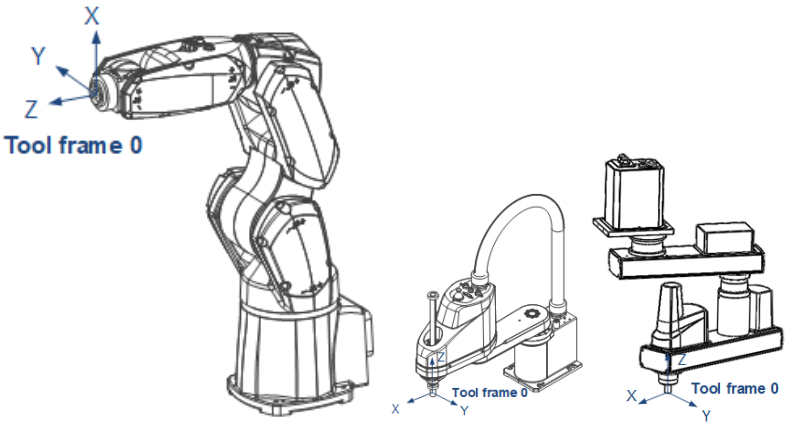


2.2.4 Tool Frame

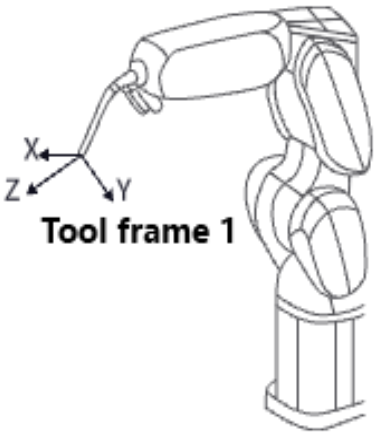
The tool frame is attached to a tool. The tool center point (TCP) is a reference point for a position where the robot reaches. The tool endpoint is generally taken as the TCP and the direction can be freely defined.



Up to 16 tool frames can be defined in Inovance robot control system. Tool 0 indicates that no tool is used, in which case, the tool frame is located at the end of the manipulator.

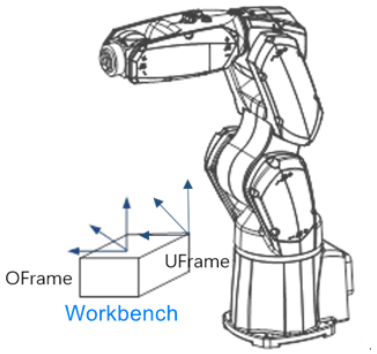


Tools 1-15 can be user defined.



2.2.5 Workobject Frame

The workobject frame is a user-defined frame. The workobject frame can be set arbitrarily and is generally set on specific objects, such as a workbench or a work object.



- Up to 16 workobject frames can be defined in Inovance robot control system.
- Workobject 0 indicates that no additional workobject frame is used. In this case, the workobject frame coincides with the world frame.
- Workobjects 1 to 15 can be defined by the user.
- A workobject includes two pose relationships (UFrame, OFrame). Parameters within UFrame indicate the relationship of the {UFrame} frame to the world frame or the flange end. Parameters within OFrame indicate the relationship of the {OFrame} frame to the {UFrame} (related to User Frame).

2.2.6 Other Frames

Some specialized frames are used for some special processes.

Frame number	Frames	Process application	Description
6	Fixed camera field of view frame	Specially used for the vision process	Point (X,Y,θ) within fixed camera field of view is defined in this frame.
7	Mobile camera field of view frame	Specially used for the vision process	Point (X,Y,θ) within mobile camera field of view is defined in this frame.
8	Object frame on the conveyor	Specially used for the tracking process	Point (X,Y,Z,A,B,C) relative to the object frame on the conveyor is defined in this frame.

2.3 Position Variables

2.3.1 Overview

- In Inovance robot control system, points in space are represented by "position variables". Points are divided into joint points and position points in terms of type.
  - A joint point stores the robot joint value and the external axis joint value.
  - A position point stores the coordinate value, arm parameter, and external axis value. In terms of scope, the position variables are divided into global position variables and local position variables.
- Global points:
  - The scope of the global position variables is a single project. That is, the global position variables can be used in all programs within a single project.
  - The global position points are denoted by P and stored in the global point file P.pts. One project contains up to 10,000 position points from P[0] to P[9999].
  - The global joint points are denoted by JP and stored in the global point file JP.pts. One project contains up to 10,000 joint points from JP[0] to JP[9999].
- Local points:
  - The scope of the local position variables is a single program file. That is, the local position variables can be used only in the current program file. The local points are denoted by LP[***].
  - The local position points are stored in the program file in the project. One program file contains up to 10,000 joint points from LP[0] to LP[9999].
  - There are no local joint points.

Usage of points: For convenience, the joint point JP is referred to as point J, and the position point P and LP are collectively referred to as point P.

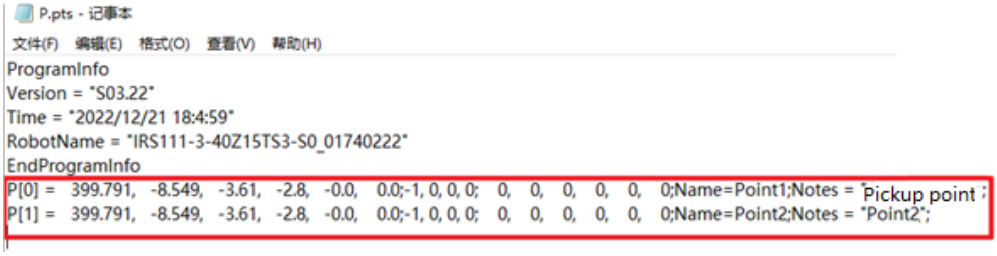
2.3.2 Storage Format of Position Variables

1) The position variable point are stored in the format as shown in the figure below:

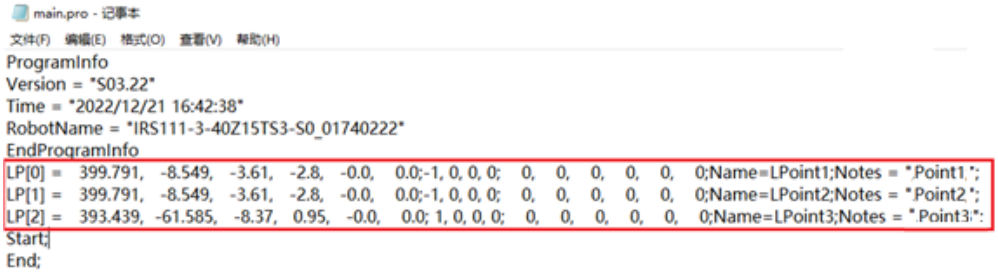
P[0] = 399.791, -8.549, -3.61, -2.8, -0.0, 0.0; -1, 0, 0, 0; 0, 0, 0, 0, 0, 0; Name=Point1; Notes = "Pickup point" ;
① ② ③ ④ ⑤

No.	Name	Delimiter	End character	Description
①	Coordinate data	English comma ","	English semicolon ";"	The number of data is always 6.
②	Arm parameter	English comma ","	English semicolon ";"	None
③	External axis value	English comma ","	English semicolon ";"	None
④	Labels	None	English semicolon ";"	The label should be less than 20 characters, start with letters and contain only letters, numbers and underscores. When there is no label, this item does not exist.
⑤	Remarks	None	English semicolon ";"	The remarks must not exceed 100 characters, with no restrictions on Chinese and English characters. Equal sign is not allowed. (Chinese input is not supported on the hand-held device, but display is supported). When there is no remark, this item does not exist.

The global position variables are stored in the project file P.pts and are common to all programs within the project.



The local position variables are stored in the robot program and are at the beginning of the file before the Start instruction. See the figure below:

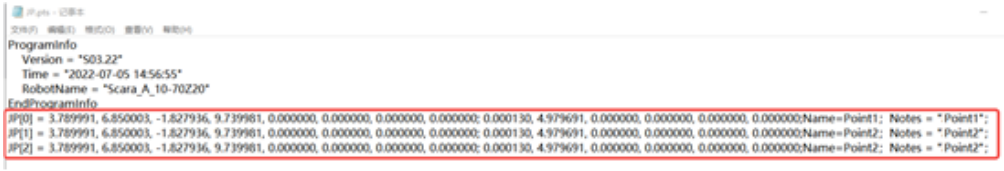


2) The joint points are stored in the format as shown in the figure below:

JP[0] = 300, 50, -3.61, -2.8, -0.0, 0.0; 0, 0, 0, 0, 0, 0; Name=Point1; Notes = "Pickup point" ;
① ② ③ ④

No.	Name	Delimiter	End character	Description
①	Coordinate data	English comma ","	English semicolon ";"	The number of data is 6 or 8 (extra axis values are 0)
②	External axis value	English comma ","	English semicolon ";"	None
③	Labels	-	English semicolon ";"	The label should be less than 20 characters, start with letters and contain only letters, numbers and underscores. No entry when no label is present.
④	Remarks	-	English semicolon ";"	The remarks must not exceed 100 characters, with no restrictions on Chinese and English characters. Equal sign is not allowed. (Chinese input is not supported on the hand-held device, but display is supported). When there is no remark, this item does not exist.

The global Joint position variables are stored in the project file JP.pts and are common to all programs within the project.



2.3.3 Attributes of Position Variables

Coordinate value

- For the JP point, always take the current joint value under the joint frame.
- For the P point, take the TCP point of the current tool under the current workobject frame.

CAUTION

In the P point taken, the tool number and workobject number are not included. Therefore, when using the P point, ensure that the tool number and workobject number are the same as those used for taking the point.

Arm parameters:

The robot can reach the same orientation in many ways. Arm parameters are used to distinguish these ways.

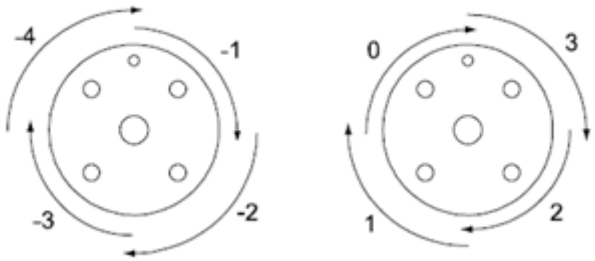
For a 6-axis serial robot:

- ArmType[0] represents multiple turns of J1.
- ArmType[1] represents multiple turns of J4.

- ArmType[2] represents multiple turns of J6.
- ArmType[3] represents a combination of different configurations of the waist joint, elbow joint, and wrist joint.

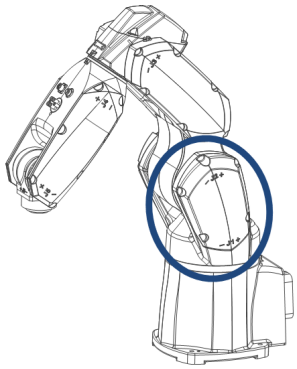
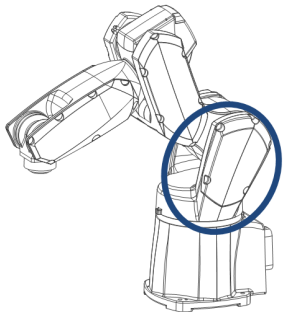
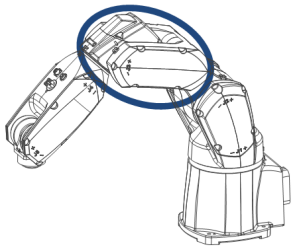
(This is only an illustration of the correspondence of the arm parameters. The range of arm parameter is defined separately.)

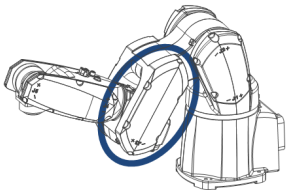
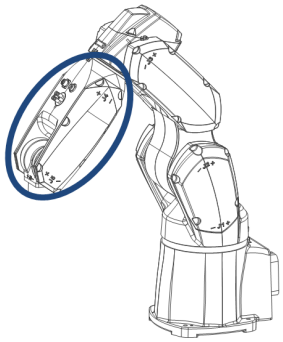
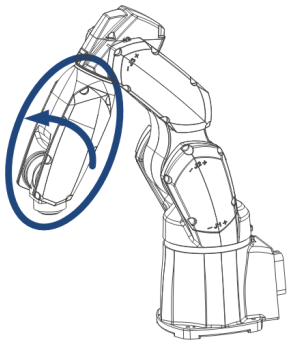
ArmType[0-2] (J1, J4, J6)

Arm parameter description								
Arm parameter value	-4	-3	-2	-1	0	1	2	3
Corresponding joint value range (unit: °)	[-360,-270)	[-270,-180)	[-180,-90)	[-90,0)	[0,90)	[90,180)	[180,270)	[270,360)
Increase or decrease by 1 every 90°, decrease clockwise, and increase counterclockwise. As shown in the figure below, "-1" is for -90° to 0°, "0" for 0° to 90°, "1" for 90° to 180°, and so on. 								

ArmType[3]

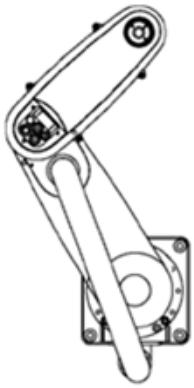
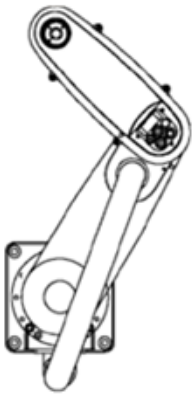
Arm parameter description								
Value of arm parameter 4	0	1	2	3	4	5	6	7
Waist arm type	Forward	Forward	Forward	Forward	Backward	Backward	Backward	Backward
Elbow arm type	Upward	Upward	Downward	Downward	Upward	Upward	Downward	Downward
Wrist arm type	Flipped	Not flipped	Flipped	Not flipped	Flipped	Not flipped	Flipped	Not flipped

Waist forward	Waist backward	Elbow upward
		

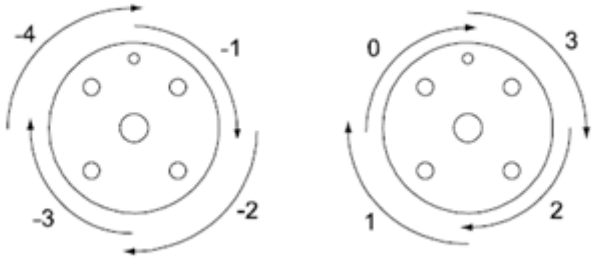
Elbow downward	Wrist not flipped	Wrist flipped
		

For SCARA robots, the values of the arm parameters 1, 4 are valid (This is only an illustration of the correspondence of the arm parameters. The range of arm parameter is defined separately.)

ArmType[0]

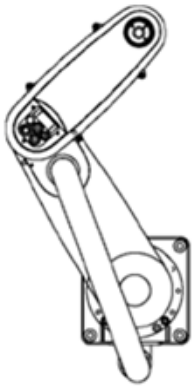
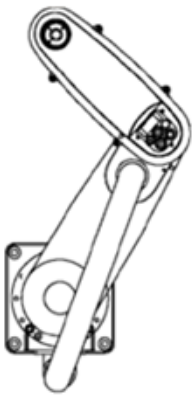
Arm parameter value	Corresponding joint	Diagram
-1	Left arm	
1	Right arm	

ArmType[3] (value of joint coordinate J4)

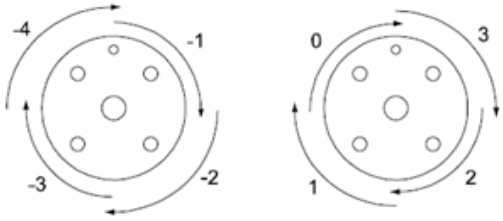
Arm parameter description								
Arm parameter value	-4	-3	-2	-1	0	1	2	3
Corresponding joint value range (unit: °)	[-360,-270)	[-270,-180)	[-180,-90)	[-90,0)	[0,90)	[90,180)	[180,270)	[270,360)
Increase or decrease by 1 every 90°, decrease clockwise, and increase counterclockwise. As shown in the figure below, "-1" is for -90° to 0°, "0" for 0° to 90°, "1" for 90° to 180°, and so on. 								

For inverted SCARA robots, arm parameters 1, 2, 3, 4 are all valid (This is only an illustration of the correspondence of the arm parameters. The range of arm parameter is defined separately.)

ArmType[0]

Arm parameter value	Corresponding joint	Diagram
-1	Left arm	
1	Right arm	

ArmType[1/2/3] (value of joint coordinate J1, J2, J4)

Arm parameter description								
Arm parameter value	-4	-3	-2	-1	0	1	2	3
Corresponding joint value range (unit: °)	[-360,-270)	[-270,-180)	[-180,-90)	[-90,0)	[0,90)	[90,180)	[180,270)	[270,360)
Increase or decrease by 1 every 90°, decrease clockwise, and increase counterclockwise. As shown in the figure below, "-1" is for -90° to 0°, "0" for 0° to 90°, "1" for 90° to 180°, and so on. 								

Definition of robot arm parameters:

Model	Arm parameter	Arm parameter range	Corresponding angle range (unit: °)
6-axis	ArmType[0]	[-5,4]	[-450, 450)
	ArmType[1]	[-5,4]	[-450, 450)
	ArmType[2]	[-∞, +∞]	[-∞, +∞]
	ArmType[3]	[0,7]	-
SCARA	ArmType[0]	[-1,1]	-
	ArmType[1]	[-5,4]	[-450, 450)
	ArmType[2]	[-5,4]	[-450, 450)
	ArmType[3]	[-9,8]	[-810, 810)
Inverted SCARA	ArmType[0]	[-1,1]	-
	ArmType[1]	[-5,4]	[-450, 450)
	ArmType[2]	[-5,4]	[-450, 450)
	ArmType[3]	[-9,8]	[-810, 810)

CAUTION

- Note: If arm parameters are modified after teaching, the robot will reach the same pose in an another arm type. This means a great change to the motion. Modify the arm parameters with caution!
- If you modify arm parameters directly by point editing or via instruction, the modified arm parameters may be inconsistent with the actual arm parameters the robot can reach. In this case, the robot will reach the position closest to the modified arm parameters.

For example, for the SCARA robots:

P[1] = (10,10, 0, 179,0,0); ## The arm parameter for P[1] are [1,0,0,1].

PR[1] = (0,-10,0,10,0,0);

P[2] = Offset(P[1], PR[1]); ## The arm parameters for P[2] are still [1,0,0,1], but the actual arm parameters corresponding to point P[2] can be [1,0,0,-2]. [1,0,0,2], [1,0,0,6]… When moving to target P[2], the robot will perform the motion according to the arm parameters [1,0,0,2] closest to [1,0,0,1].

2.4 Offset

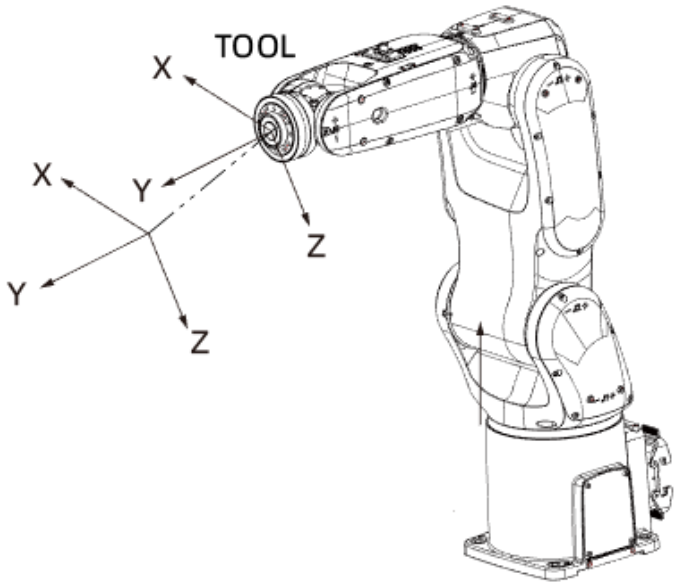
Offset is used to describe the motion in space, which can be the motion of joints J1 - J6 described in joint frame, or the spatial change of a certain pose XYZABC relative to a certain frame in three-dimensional space.

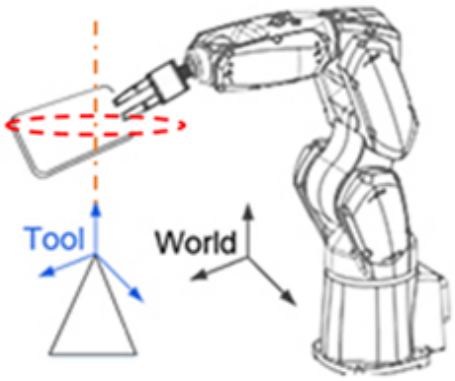
In the Inovance robot system, the offset is divided into the following three types.

2.4.1 Offset Along the Current Tool Pose

Function: Moves the robot along the current tool pose.

Instruction: OffsetT





2.4.2 Offset in a Cartesian Reference Frame

Function: Offset indicates offset of a target point in a Cartesian reference system. The target point may be the TCP or flange center point. The reference frame may be a user frame or a base frame.

In typical scenarios, for grip tool:

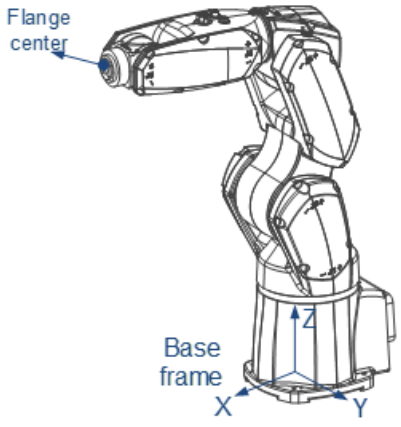
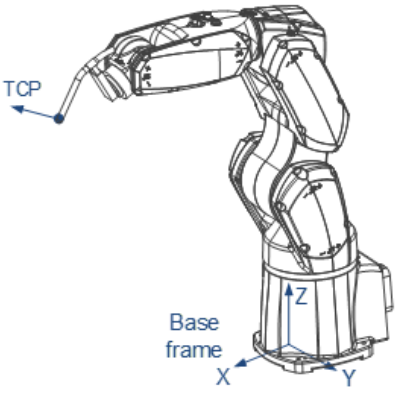
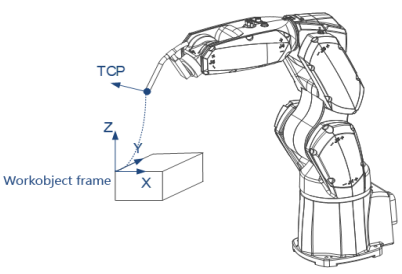
- Offset is the offset of TCP relative to the workobject frame.
- Movl Offset(P,PR) … Tool[2] ##Offset relative to wobj[0] (which is also the base frame).
- Movl Offset(P,PR) …Tool[2] Wobj[3] ##Offset relative to Wobj[3].

External tool (in the case of ECP)

This motion mode is based on the principle of TCP movement control relative to the workobject frame, and its spatial transformation exhibits the following characteristics:

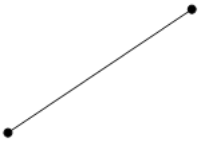
- Motion reference: Establishes a virtual reference frame with the external TCP as the origin, which remains spatially parallel to the workobject frame;
- Translation characteristics: The workobject undergoes linear translation along each axis of the workobject frame;
- Rotation characteristics: The workobject performs rotational transformations around the X/Y/Z axes of the virtual reference frame.

Instruction: Offset

Offset 1	Offset 2	Offset 3
		

2.5 Interpolation and Transition

Interpolation is the basic form of robot motion, and complex motion is made up of a series of interpolation motions. There are three types of interpolation, depending on the interpolation trajectory.

Interpolation type	Trajectory	Characteristics
Joint interpolation (Movj, MovAbsj)		Point-to-point interpolation, the fastest interpolation in which each joint moves at the fastest speed. The motion trajectory is unpredictable. It is often used for applications such as spot welding and transportation.
Linear interpolation (Movl)		The motion trajectory is straight. It is often used for applications such as track welding and surface mounting.
Arc Interpolation (Movc)		The trajectory is arc-shaped.

CAUTION

When executing Movl and Movc, the arm parameters of the robot are not allowed to change. If you need to change the arm parameters, insert the Movj and MovAbsj instruction to complete the transition of arm orientation.

In the actual continuous motion process, in order to accelerate rhythm, precise arrival is not required. At this time, the intermediate point of the motion will show the form of trajectory approximation, which is called transition.

In Inovance robot system, the transition level parameters are set as follows: Fine, Z[0] to Z[200], Z[CP], Z[CP], and ZR[m].

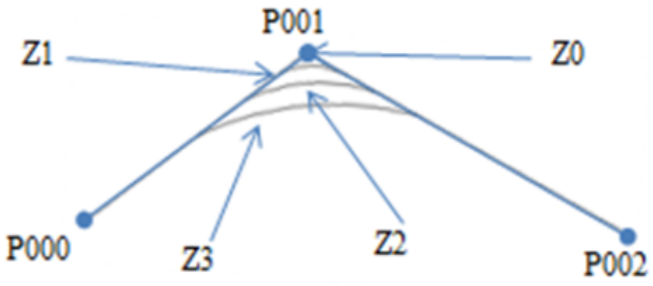
- Fine: Indicates precise arrival. The robot is considered to achieve precise arrival when it enters and stays within the set arrival error range.
- Z[0] to Z[200]: The robot does not stop at the target point, but smooths through the set transition area. Transition length = transition level x transition unit length. (To set the transition unit length, go to "Controller Parameter Configuration" > "Motion parameters" > "Run parameter settings" > "Transition settings" > "Transition feature").
- Z[CP]: The robot does not stay at the target point and smooths through the set transition area with the maximum transition length.
- ZR[m]: Relative transition level, integer, value range 0 to 200, represents the percentage of maximum allowable transition length.
  - If the current instruction is Movj/Movl/Movc, when the relative transition level m is equal to 100, it indicates that the maximum allowable transition length is half the motion length of the instruction,

which is consistent with the effect of Z[CP]. When the relative transition level m is greater than 100, the maximum allowable transition length will exceed half the motion length of the instruction. When the ZR parameter is 200, the maximum allowable transition length is equal to the full motion length of the instruction.

- If the current instruction is Jump/Jumpl and the additional parameter RH=0, the maximum allowable transition length will exceed half the motion length of the instruction when the relative transition level m is greater than 100. When the ZR parameter is 200, the maximum allowable transition length is equal to the full motion length of the instruction.
- If the current instruction is Jump/Jumpl and the additional parameter RH≠0, the maximum allowable transition length is the total length of RH segment. At this time, ZR[100] is the total length of RH segment. When the ZR value is greater than 100, it is consistent with the effect of ZR[100].

CAUTION

- When the robot completely transitions the current motion to the next motion, the robot may not pass the original trajectory at all. Because the path of motion in the transition area cannot be guaranteed, make sure that the robot transition path does not interfere with surrounding objects at low speeds.
- Since the ZR parameter allows for a transition length longer than half the motion length of the instruction, in rare cases, the transition areas of the start and end points of the instruction may overlap. In this case, the robot will reduce the transition area that is more than half the motion length of the instruction to ensure that the two transition areas no longer overlap. In particular, if both transition areas at the beginning and end of the instruction are larger than half the motion length of the instruction, they will be reduced to half the motion length of the instruction at the same time.
- Since the transition length specified by the ZR parameter is calculated by the percentage of the maximum allowable transition length of the instruction (usually half the motion length of the instruction), it is important to note that theoretical transition length of the two instructions may be inconsistent when the displacement difference between two adjacent motion instructions is large. The robot may transition in an asymmetric transition path within the transition area specified by the user.



Interpolation precision

For details,see the description of transition characteristics in the Industrial Robot Instruction Guide (for example, "Transition characteristics" under section "MovJ").

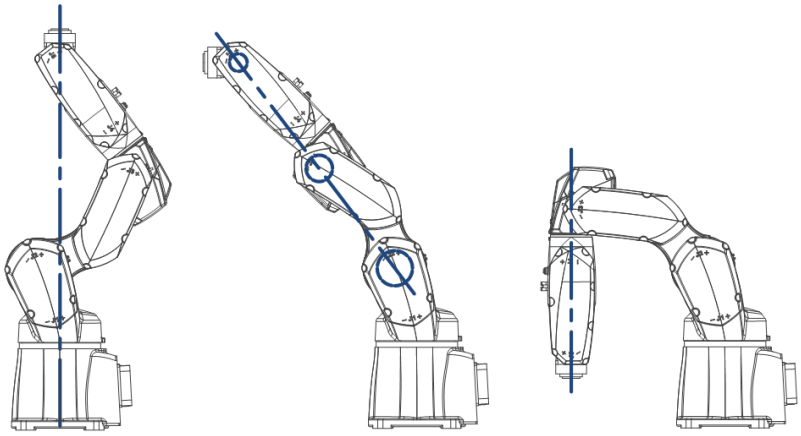
2.6 Singular Position

When moving in a non-joint frame, the robot may move to certain special positions where the robot loses some freedom of movement called singular positions.

In joint interpolation Movj and MovAbsj, the singular position does not affect normal motion. In the process of linear interpolation Movl and arc interpolation Movc, the singular position prevents the robot from moving properly.

> **CAUTION**
>
> When a singularity position alarm is triggered, you can switch to joint motion mode and manually adjust the pose of each axis through teaching to move the robot out of the singularity area and clear the alarm status.

There are three singular positions for the 6-axis robots, as shown in the figure below.

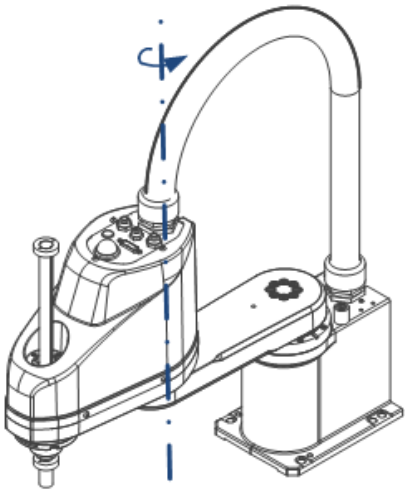


Top singularity Extended singularity Wrist singularity

There are three singular positions for the 6-axis robots, as shown in the table below.

Top singularity	Extended singularity	Wrist singularity
Intersection of J4, J5, and J6 axes	Centers of J2, J3 and J5 axes	J4 and J6 axes are collinear
Located directly above J1	Collinear with the center	-

The SCARA robot has only one singular position, at J2= 0°, when the 1st and 2nd arms are in a straight line.

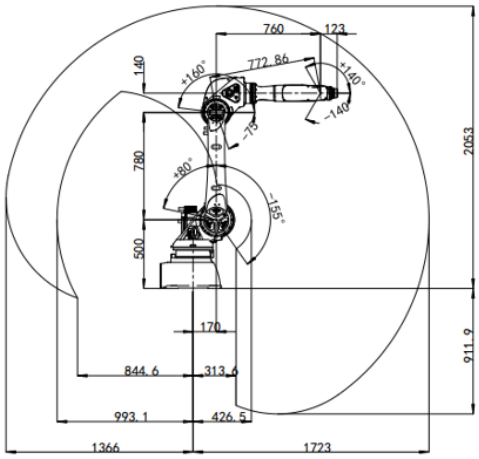


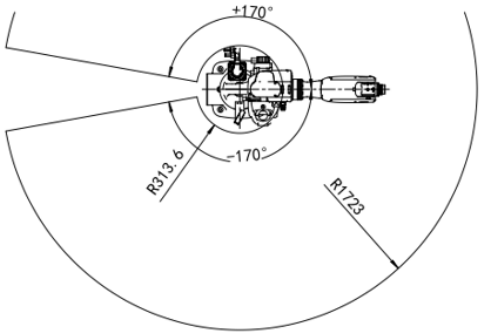
(SCARA singular position J2=0°)

For the description of singularity crossing feature, see 12.7 Singularity Crossing.

2.7 Motion Range and Interference Zone

The range of motion is a collection of all points that can be reached by the end of the robot's arm. The range of motion is related to the length of the robot arm and the range of motion of the robot joints.

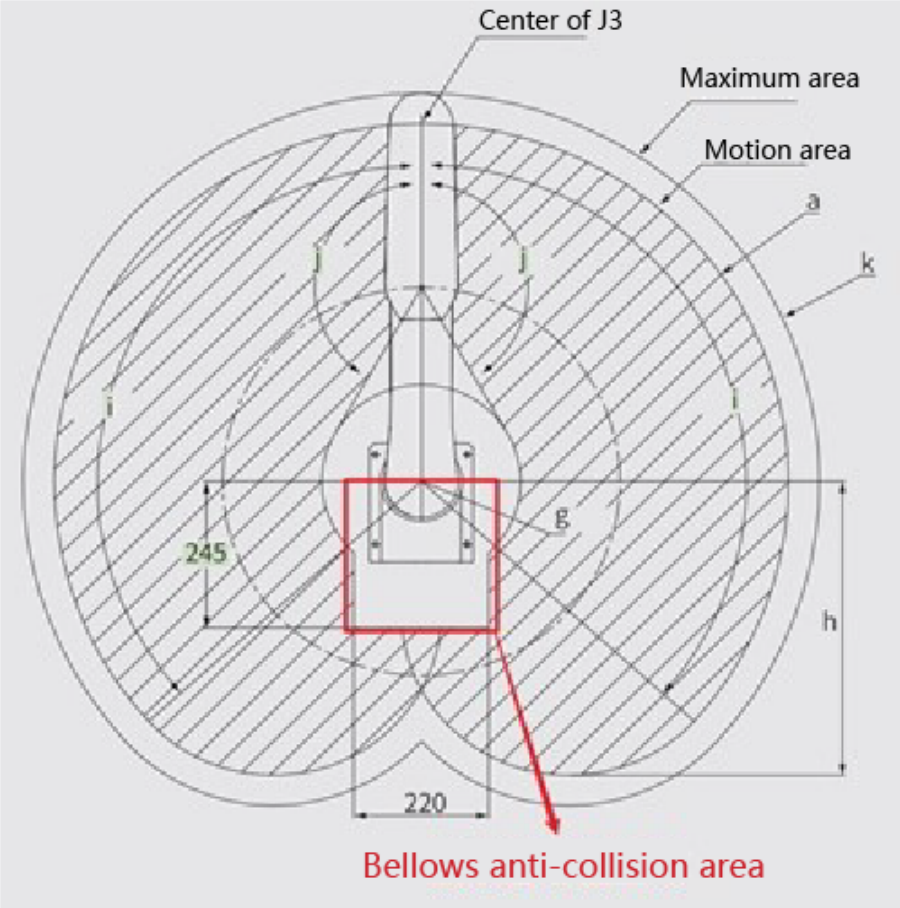




Interference zone: In the range of motion, there are often areas where the end gripper is prevented from reaching. When in these areas, the robot will collide with its own components or external devices. These areas are called interference zones.

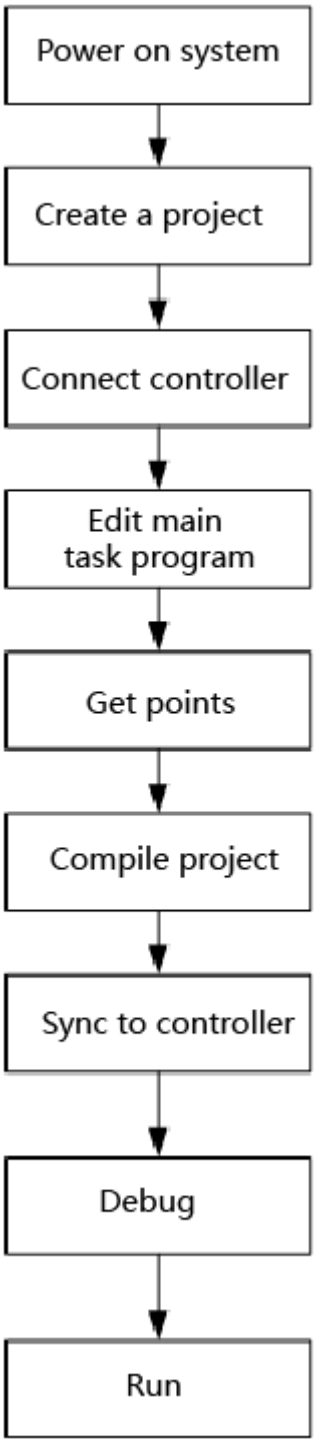
You can customize the interference zone. For details, see 12.11 Interference Zone.

For the IR-S4&S7&S10 series SCARA robots in the 4kg/7kg/10kg series, a bellows anti-collision function has been added: In the world frame, the base area of the SCARA robot is set as an interference zone. Whether in joint frame motion or Cartesian frame motion, it will detect whether the lead screw end will enter the interference zone defined in the world frame. If it enters, an alarm (0x2814 Robot entered non-movement areas) will be triggered and the robot will stop. The bellows anti-collision areas vary for different models. For details, see the "Inovance Robot Selection Guide".


Center of J3
Maximum area
Motion area
a
k
j
j
i
i
g
245
h
220
Bellows anti-collision area

3 Get Started

3.1 Programming Flowchart



3.2 Programming Examples

3.2.1 Introduction

After installing InoRobotLab on your PC, create a simple application as described below to get familiar with InoRobotLab programming software.

As an example, the following describes how to program the robot to realize material place and pickup.

```
Start;
   # Preparation
   Movj pStart,V[30],Z[0],Tool[0];
   Int count = 0;
   Set GasClaw,OFF;
   Wait GasClaw Open == ON, T[0];
   # Start the work cycle
L[1]:
Movl pBeforeTake,V[30],Z[0],Tool[0];
WaitInPos;
   Wait takeBegin == ON, T[0];
Movl pTake,V[30],Z[0],Tool[0];
WaitInPos;
   Set GasClaw,ON;
   Wait GasClawClose == ON, T[0];
   Movl pBeforePut,V[30],Z[0],Tool[0];
Movl pPut,V[30],Z[0],Tool[0];
WaitInPos;
   Set GasClaw,OFF;
   Wait GasClawOpen == ON, T[0];
   Print "Number of places";
   count = count + 1;
   Print count;
   Movl pBeforePut,V[30],Z[0],Tool[0];
   Goto L[1];
End;
```



Ensure that the project is always under InoRobotLab control.

3.2.2 Powering On the Robot System

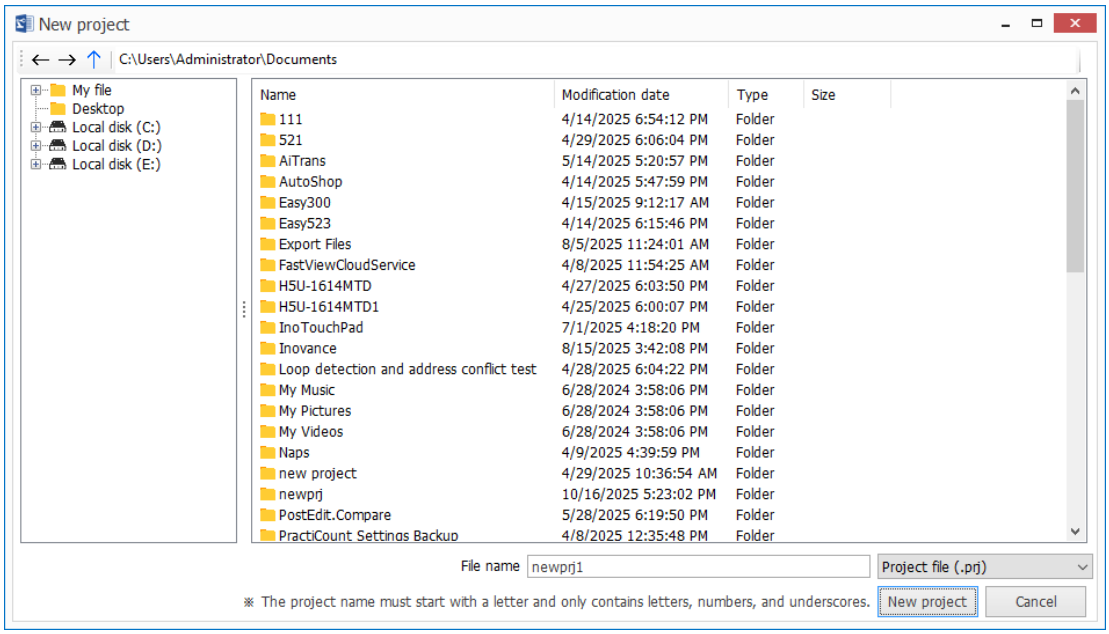
Before operation, check that:

- All safety protections are in place. No personnel are present in the operating area of the robot system.

- The power supply of the controller, PC, and I/O devices is connected properly and no alarm occurs.
- The InoRobotLab programming software is open on the PC.

3.2.3 Creating a Project

1. Double-click the InoRobotLab software icon to open the software.
2. Click "Project" > "New" to create a new project.
3. In the pop up window, enter the project name in the Project name field, such as "newprj", and then click the "New Project" button. The project name must start with a letter and contain only letters, numbers, and underscores.



By default, the file name of the main task program is "main.pro".

3.2.4 Connecting to the Controller

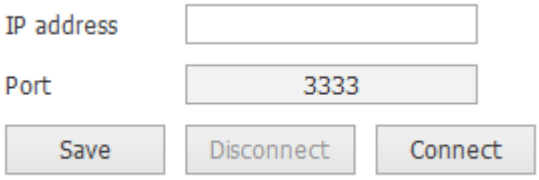
It is necessary to establish a connection with the controller to read the robot type and parameters from the controller.

Operations such as creation of the project are related to the robot type. If the project does not match the robot type, the project cannot be synchronized to the controller successfully.

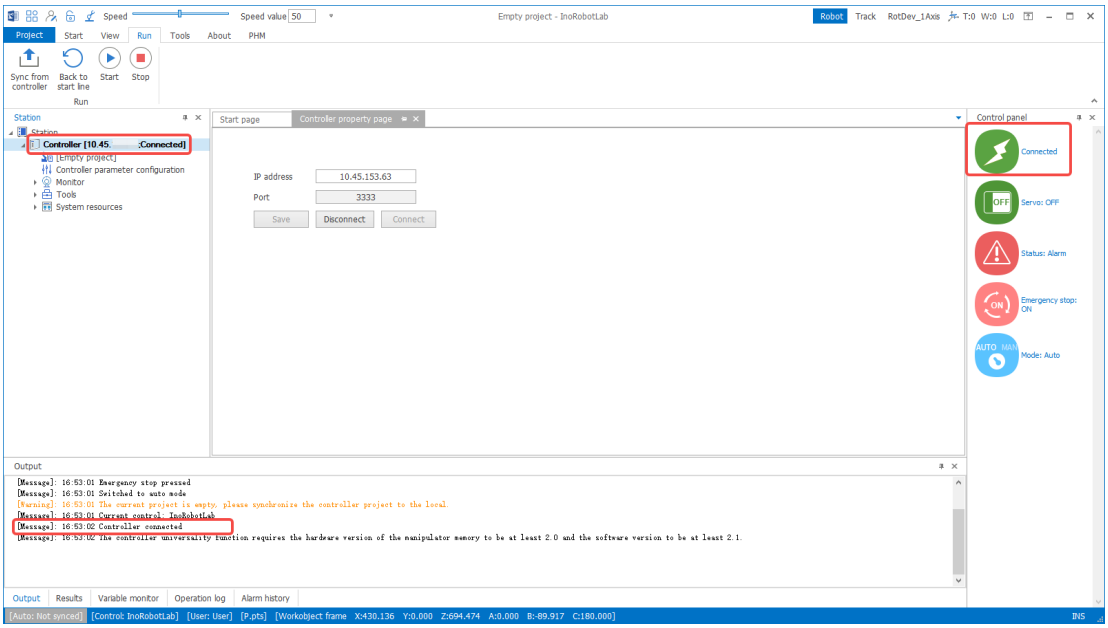


Before connecting InoRobotLab to the controller, ensure that the teach pendant (including PC-based virtual teach pendant) is disconnected from the controller.

1. Right-click on the controller node and select "Connect".
2. If the connection fails, right click the controller node and select "Properties". The controller property page is opened. Enter the IP address of the target controller and click "Connect".



3. After successful connection, the controller node status displays "Connected". The bottom Output pane prints the connection success information, and the connection status displayed on the control panel changes from "Disconnected" to "Connected".



CAUTION

- If the connection fails but the controller can be pinged successfully, check if:
  - The port 3333 of the controller is used, for example, by the teach pendant; or
  - Multiple InoRobotLab instances are connected to the same controller.
- If the 0x0080 alarm cannot be cleared, see Emergency Stop Trigger Method for settings.

3.2.5 Editing the Main Task Program

After connecting InoRobotLab to the controller and creating the new project, you can create a material place and pickup program.

Double-click the "main.pro" program file under the "newProj" project.

In the open program code editor, enter the following program.

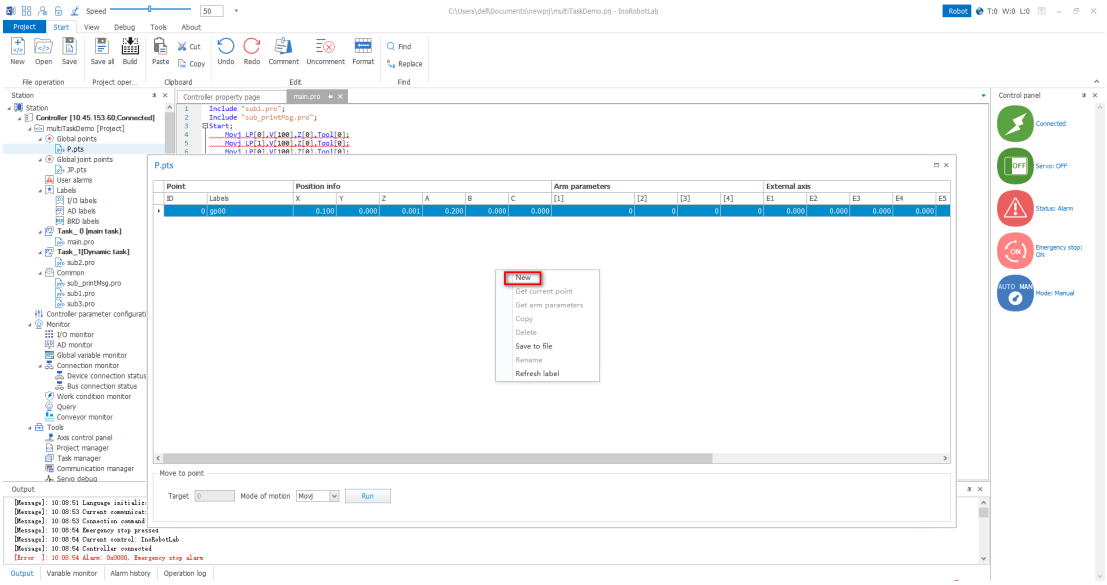
```
Start;                                          # Entry command for the task
   Print "MAIN Start";                         # Print statement MAIN Start
   Int intA=0;                                  # Define local variable intA
   While intA<3                                 # Perform loop execution under the condition of
local variable intA<3
    intA=intA+1;                                # Perform addition of local variables
    Movj P[0],V[30],Z[0],Tool[0],Wobj[0];       # Movj to P[0]
    Movj P[1],V[30],Z[0],Tool[0] ,Wobj[0];      # Movj to P[1]
    Movj P[2],V[30],Z[0],Tool[0] ,Wobj[0];      # Movj to P[2]
    Movj P[3],V[30],Z[0],Tool[0] ,Wobj[0];      # Movj to P[3]
EndWhile;                                       # End loop
CloseClaw(1);                                   # Call the function to turn off the gripper
Print "MAIN End";                               # Print statement MAIN End
End;                                            # Program end flag
Func CloseClaw(int ClawNo)                      # Define the function of turning off the gripper
Switch ClawNo                                   # Select the number of the gripper to be turned off
 Case 1:                                        # Gripper 1
   Set Out[13],ON;
   Break;
 Case 2:                                        # Gripper 2
   Set Out[14],ON;
   Break;
EndSwitch;
EndFunc;
```

3.2.6 Taking Points

Position variables are used for robot motion instructions. When motion instructions are included in the program, the user needs to take the position point. The points include global position points, global joint points, and local position points, which are represented by P[1], JP[1], and LP[1] respectively. Global points are take effect throughout the project, while local points take effect only in a single program file.

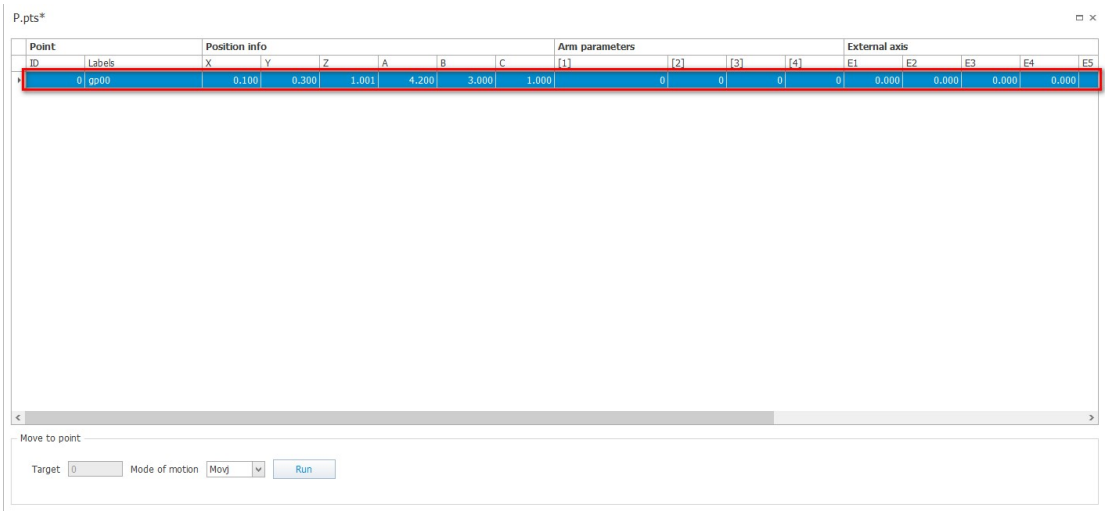
Steps

1. Double-click "P.pts" under the project directory to open the global points file.
2. On the "P.pts" page, when the robot reaches the target position, right-click on the list and select "New". The added point appears in the list. Add other points as per needs.



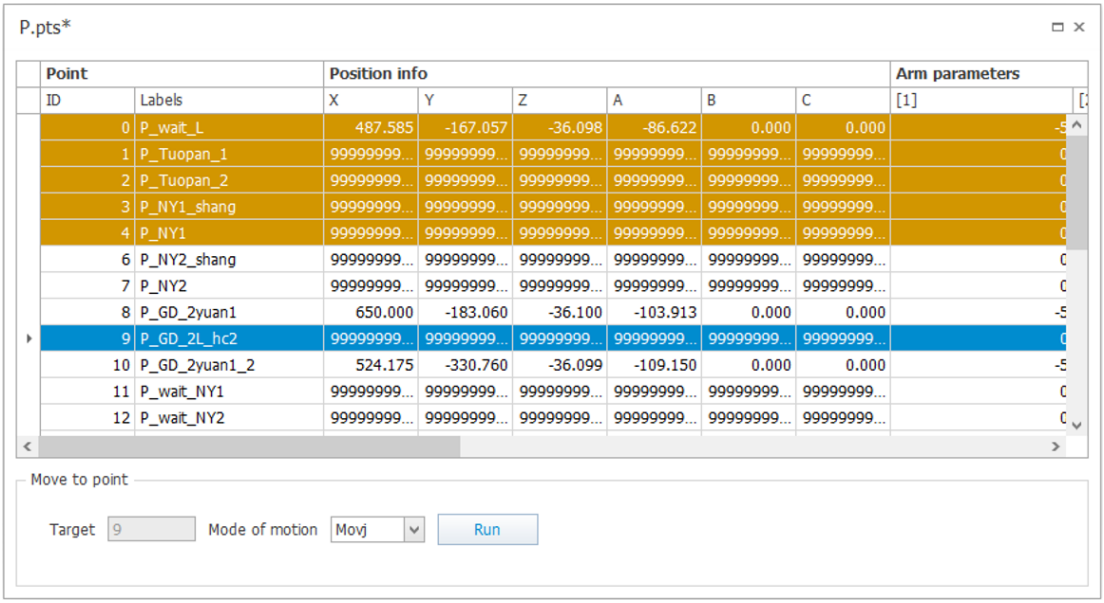
Results

After the point is added, it appears in the list, as shown below.



NOTE

- If multiple points are used in the program, they need to be added to the points list. For example, the "main.pro" program uses four points P[0], P[1], P[2], and P[3], as shown in the figure below.



- After completing the point selection, you need to save by right-clicking and selecting the "Save" button or using the shortcut key Ctrl + S.

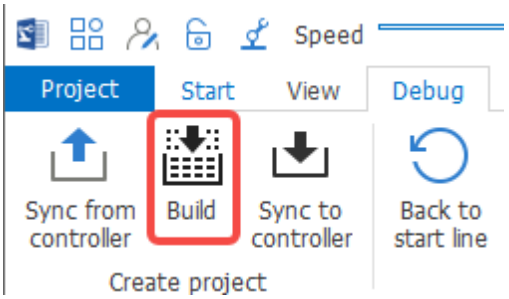
3.2.7 Compiling the Project

Project compilation can check for syntax errors in the project and prepare for downloading the project to the controller. In addition, function calls across program files also require compilation of the entire project. For more details, see the 5.2.9 Compiling the Project section.

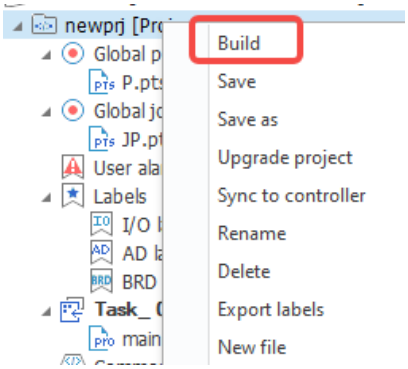
Steps

To enter the project compiling interface, use either of the following two methods:

- Method 1: In the menu bar, click "Debug" > "Build" to start compiling the project, as shown in the figure below.



- Method 2: Right-click on the project node and select "Build" to start compiling the project, as shown in the figure below.



Results

If an error is reported, solve the problem according to the error information. If no error is reported, you can synchronize the project to the controller.

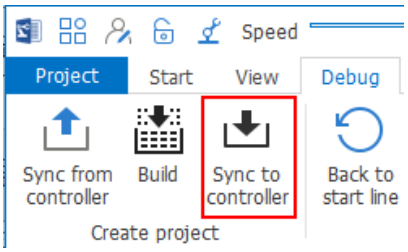
3.2.8 Synchronizing to the Controller

Synchronize the compiled project to the controller, so that the controller can execute the program in the project.

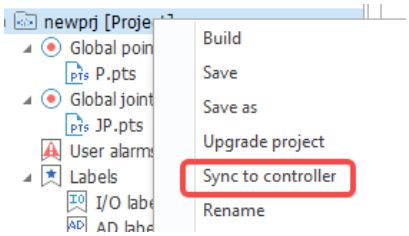
Steps

There are two methods to synchronize to the controller:

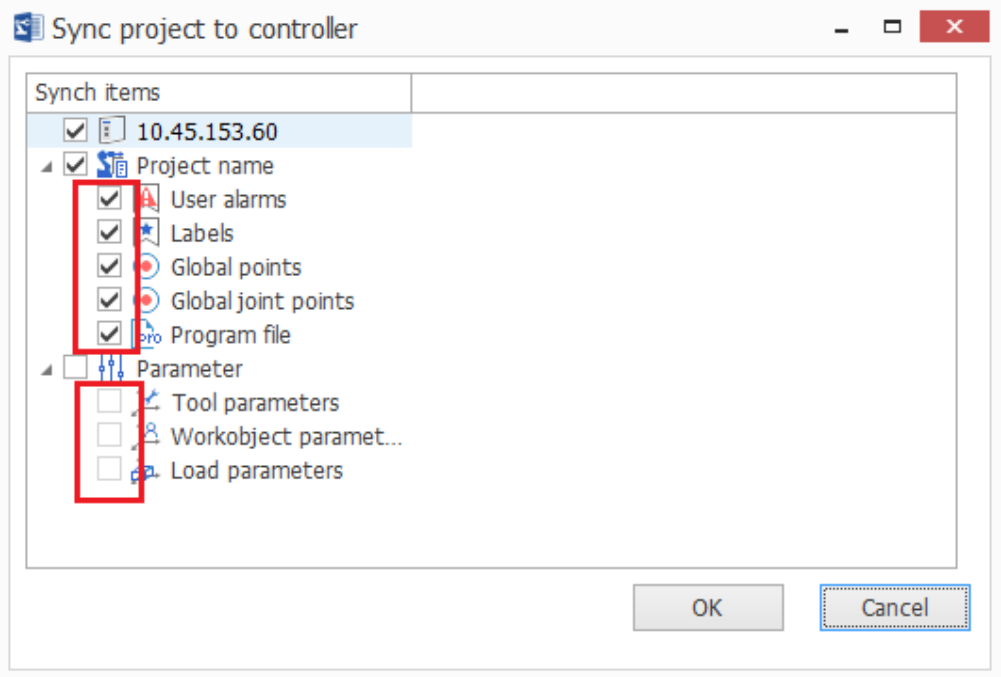
- Method 1: In the menu bar, click "Debug" > "Sync to controller", as shown in the figure below:



- Method 2: Right-click on the project node and select "Sync to controller", as shown in the figure below:



In the pop up window, select the project and click "OK".



Results

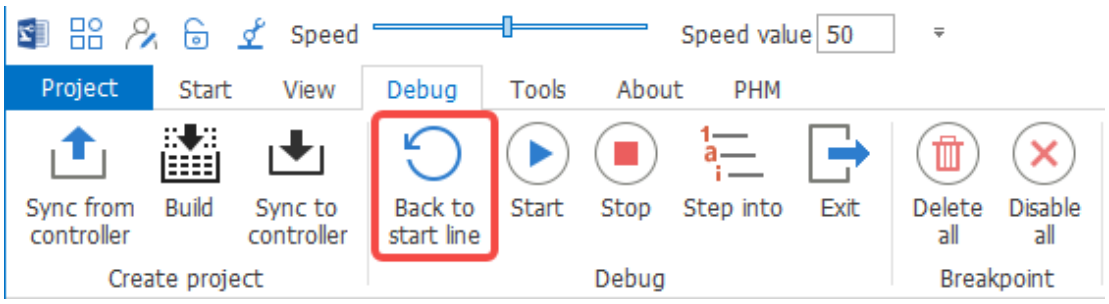
After the project is synchronized to the controller, the project will be the current active project in the controller. You can debug the program in the project.

3.2.9 Starting/Stopping the Debugging

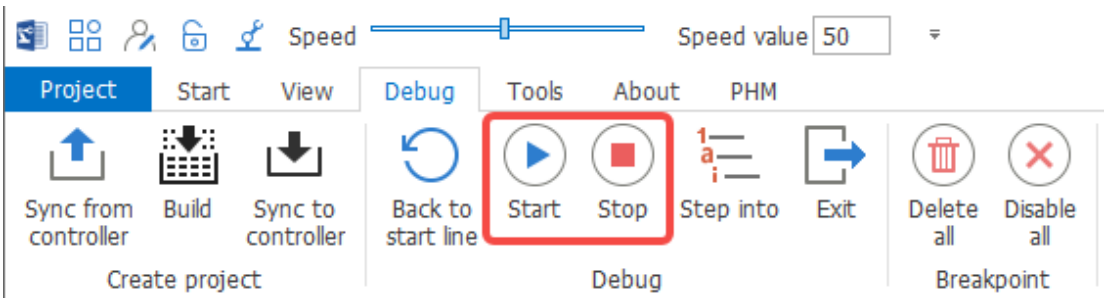
Without entering the automatic mode, you run the program in the active project in the controller by a number of auxiliary means to verify whether the program execution effect is consistent with the expected result.

Steps

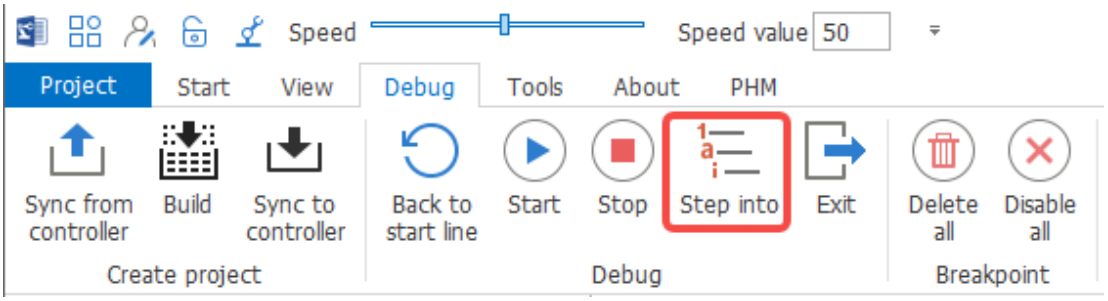
1. Debug the program in manual mode when the servo is enabled.
2. In the menu bar, click "Debug" > "Back to start line" to set the start line of the main program as the current line, as shown in the figure below.



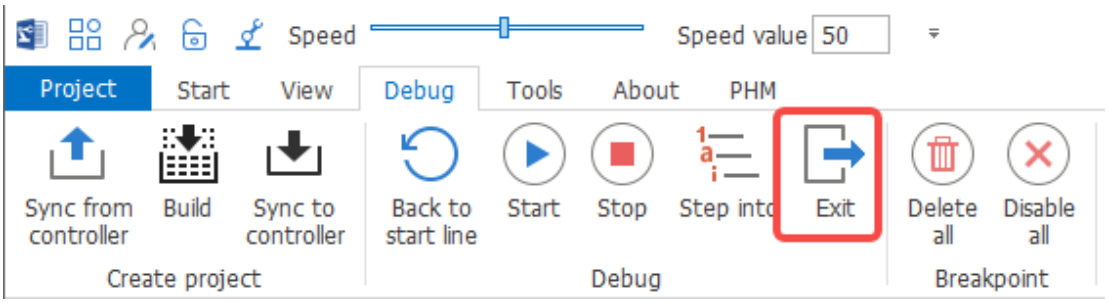
3. Click "Start" to run the program from the current line until a breakpoint is encountered.
   - During program running, you can click "Stop" to interrupt the program.



◦ Or click "Step into" to execute the program by one line before you click it again.



4. If you return to the start line or start the execution of the program, the InoRobotLab enters the debug mode. To exit the debug, click "Exit".



Results

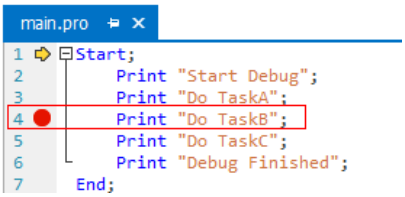
Compare the program execution effect with the expected result to adjust and optimize the program.

3.2.10 Debugging the Breakpoint

If you want to debug a certain program line and subsequent program lines in detail, you can set a breakpoint in the target line. The program stops execution at the breakpoint and waits for subsequent instructions.

Steps

Click on the right side of the program line number to set a breakpoint. Click again to cancel the breakpoint.



Results

After the program is started, it stops at the line where the breakpoint is set. The line is not executed. In this case, you can click "Start" or "Step into" to resume the program execution.

> **NOTE**
>
> For the use of breakpoint debug, see 7.2.3 Adding and Deleting Breakpoints, 7.2.4 Enabling, Disabling, and Deleting All Breakpoints, and 7.2.5 Entering a Breakpoint.

3.2.11 Starting/Stopping Program Running

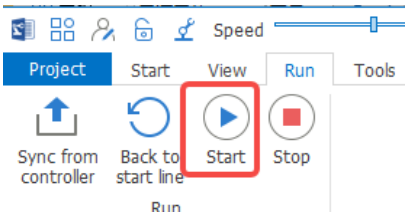
After the project is synchronized to the controller, you can switch to automatic mode and run the program in the project.

Steps

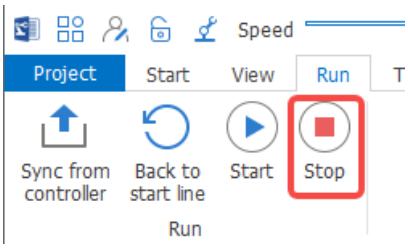
1. If there is an existing project in the controller, click the mode switch button on the control panel to switch from the manual mode to the auto mode, and the robot is automatically enabled.



2. In this case, the "Debug" option in the menu bar changes to "Run". Click "Start" to start the program in the current project, and "Stop" to stop the program.



Start running



Stop running

4 Basic Operations

4.1 Overview

Basic Operations		Icon	Description
Connect/Disconnect the controller			You can debug or run the program, and control the robot after connecting the InoRobotLab software to the controller. For details operations, see 4.2 Connecting/Disconnecting the Controller.
User login			InoRobotLab includes three types of login users: user, admin, and manufacturer. Each type of user has different access to the InoRobotLab operation functions. In terms of access, they are sorted in order of user < admin < manufacturer from low to high. For detailed operations, see 4.3.1 User Login.
Switch the control permission			The actions of the robot controller and the robot manipulator are controlled by the upper-layer devices including InoRobotLab, teach pendant, remote Ethernet client, and remote I/O unit. The InoRobotLab and the teach pendant share one port and have the same highest priority among all upper-layer devices. If you need to control the robot with an upper-layer device other than InoRobotLab or teach pendant, you can switch control permission from InoRobotLab or teach pendant to another target upper-layer device. For detailed operations, see 4.5 Switching Control Permission.
Basic control panel operations	Connection state		Displays the connection status of InoRobotLab and the robot controller. See 4.2 Connecting/Disconnecting the Controller.
	Clear warnings and alarms		After you have resolved the problem that generates warnings and alarms, click this icon to clear the robot warnings and alarms to restore the robot to normal state. For detailed operations, see 4.6 Clearing Warnings and Alarms.
	Emergency stop		Triggers/Cancels the robot emergency stop. For detailed operations, see 4.7 Emergency Stop.

Basic Operations		Icon	Description
	Enable/ Disable robot		Enables or disables the robot. For detailed operations, see 4.8 Enabling/Disabling the Robot.
	Switch mode		Switch between the manual mode and auto mode. For detailed operations, see 4.9 Switching between Manual and Auto Modes.
Configure the default number of robot axes in offline mode			Some InoRobotLab software interfaces displayed or program logic are related to the number of axes of the target robot (for example, programs written for a 4-axis robot should not involve axis #5 or #6. This option allows you to set the number of robot axes so that you can offline programming without connection to the controller. This option is used only for offline programming and is independent of the actual number of axes obtained after the controller is connected. For detailed operations, see 4.10 Configuring the Default Number of Axes Offline.
Axis operation	Switch the frame		The frames that can be used on InoRobotLab include joint, Cartesian, tool, and user frames. You can switch between different frames to make teaching and programming easier. For detailed operations, see 4.11.3 Switching Frame.
	Switch the tool number		If the tool is changed, you need to switch the tool number. For detailed operations, see 4.11.4 Switching Tool Number.
	Switch the workobject number		Multiple workobject frames may exist. You can switch the workobject frame as per needs. For detailed operations, see 4.11.5 Switching Workobject Number.
	Adjust speed		You can adjust the robot motion speed as needed. For detailed operations, see 4.11.7 Adjusting Speed.
	Axis motion (continuous motion and jogging)		You can control the robot to move continuously or jog. For detailed operations, see 4.11.8 Axis Motion.

Basic Operations		Icon	Description
	View current position		The current position is displayed on the axis control panel, as well as in the status bar. For detailed operations, see 4.12 Viewing Current Position section.
Installation parameters	Tool parameters		After tools such as grippers, nozzles, or welding gun are installed at the end of the robot, it is necessary to calibrate the tool frame before use to determine the relative relationship between the working position and orientation of the tool and the robot base frame. This allows you to program and control the position and orientation of the end tool. For detailed operations, see 9.3.1 Tools.
	Workobject parameters		The workobject frame is a user-defined frame. The workobject frame can be set arbitrarily and is generally set on specific objects, such as a workbench or a conveyor. Before using the workobject frame, calibration is required. For detailed operations, see 9.3.2 Workobject.
	Load parameters		You can set load parameters to improve the accuracy and stability of robot motion. For detailed operations, see 9.3.3 Load.

4.2 Connecting/Disconnecting the Controller

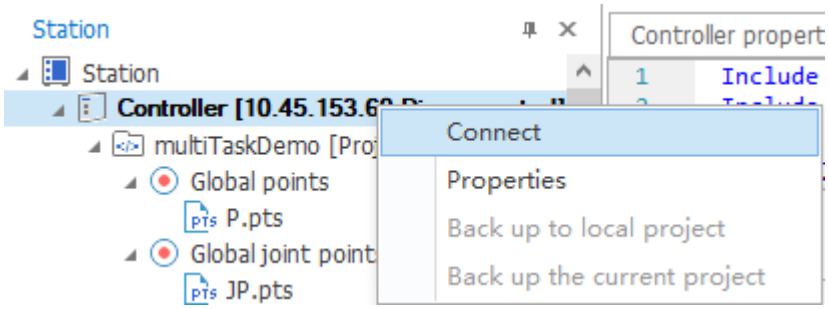
Connecting the controller

If the robot controller is not connected, you can only perform operations such as editing the local project file, without interaction with the robot. If interaction with the robot is required, connect the controller.

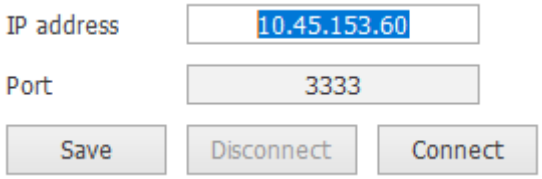
Steps

There are two methods to connect the controller:

- Method 1: Right-click on the controller node and select "Connect".



- Method 2: Right-click on the controller node and select "Properties", and click "Connect" in the dialog. In this mode, you can also modify the IP of the controller to be connected.



Results

- If the connection is successful, the controller node status displays "Connected", and the bottom Output pane prints "Controller connected".
- If the connection fails, the controller node status displays "Disconnected", and the bottom Output pane prints "Device offline". The user is prompted to check whether the controller is powered on or occupied by the teach pendant.

If the connection fails, check according to the following steps:

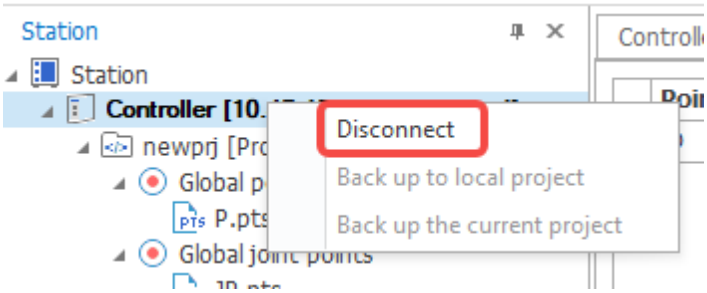
1. Check that the hardware wiring is correct.
2. Check whether the PC and the robot controller are in the same network and whether the robot controller can be pinged successfully.
3. Check whether the target robot controller is connected to another InoRobotLab instance or teach pendant. If connected, disconnect the controller from another InoRobotLab instance or teach pendant.

InoRobotLab software may disconnect from the controller when:

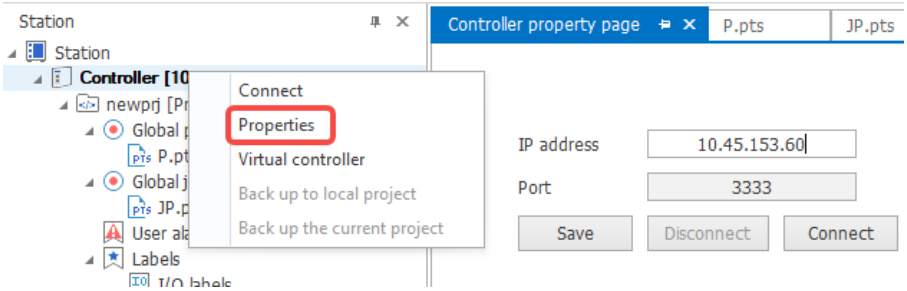
- Strong interference occurs on the network, for example, the FTP transfer is unstable.
- The CPU usage of the PC is too high (over 50%).
- The physical network connection is down due to, for example, the network cable is disconnected, and the router or switch is powered off.
- The InoRobotLab (or teach pendant) version differs too much from the controller version.
- The CPU usage of the controller is too high (over 95%).

Disconnecting the controller

- Right-click on the controller node and select "Disconnect", as shown in the figure below.



- Alternatively, right-click on the controller node and select "Properties", and click "Disconnect" in the open page, as shown in the figure below.



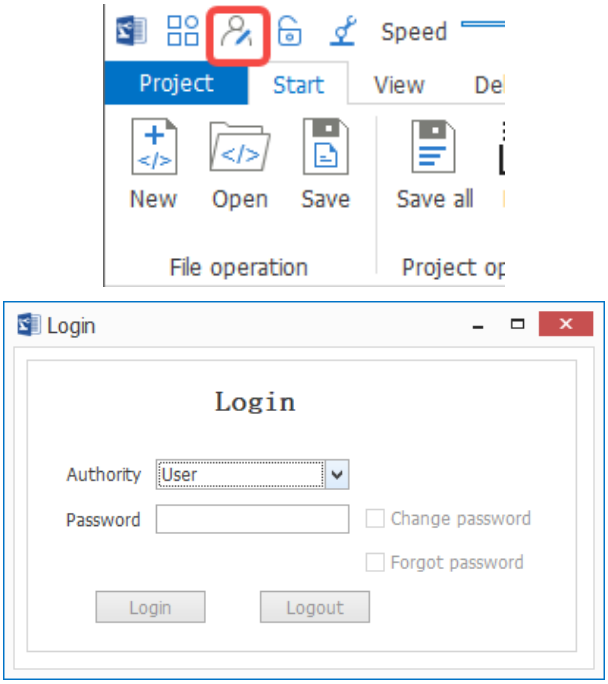
4.3 User Login

4.3.1 User Login

InoRobotLab includes three types of login users: user, admin, and manufacturer. Each type of user has different access to the InoRobotLab operation functions. In terms of access, they are sorted in order of user < admin < manufacturer from low to high.

Steps

In the upper left corner of the toolbar, click the login icon, select the user type in the pop-up window, enter the password, and click "Login".

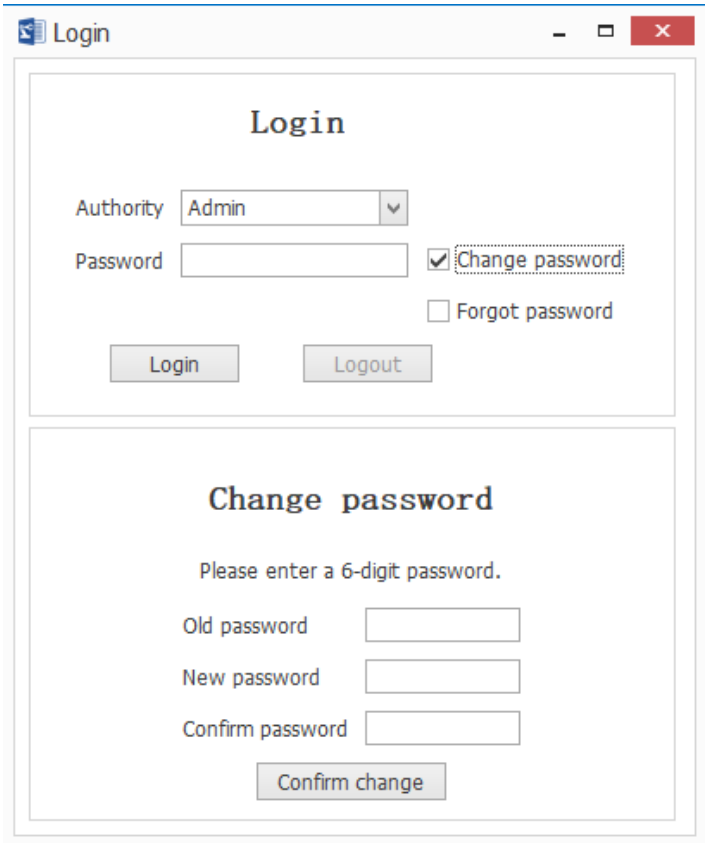


4.3.2 Changing the Password

InoRobotLab only allows you to change the password of administrator account. Before changing the password, log in to InoRobotLab.

Steps

After logging in with administrator account, open the user login page again, set "Authority" to "Admin" and click "Change password", enter the old password, new password, and confirm password, and then click "Confirm change".

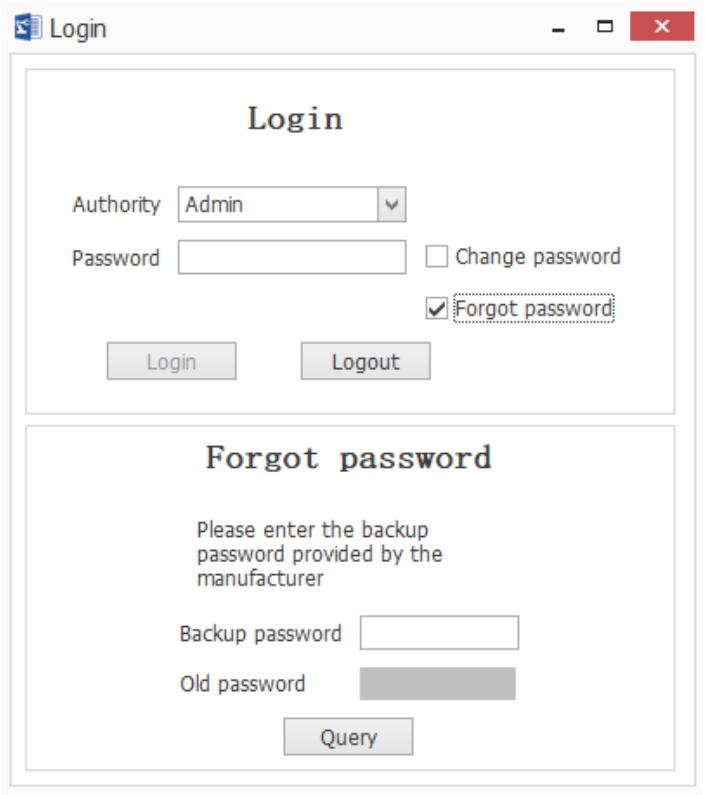


4.3.3 Retrieving the Password

InoRobotLab only allows the administrator password to be retrieved. If you forget the administrator password, contact Inovance for help.

Steps

Set "Authority" to "Admin" and click "Forgot Password", enter the backup password and click "Query". Then old password is displayed.



4.4 Description of User Authorities

Authority	Icon	Password	Applicable To
User		None	Ordinary users for program editing, saving, downloading, running, debugging and more
Admin		000000	Robot system maintenance personnel, with access to most system settings besides operation permissions granted to ordinary users.
Manufacturer		Factory reserved	Inovance technical support personnel, with the highest authority and can perform operations such as upgrading the system.

NOTE

The manufacturer has the highest authority and access to all software features.

The following table shows the correspondence between user type and operation permissions.

Function	Operation permission	Role
Axis control panel	All operations	User
Project manager	All operations	User
Task manager	Start/Stop	User
Communication manager	Switch between client and server	Admin
	Open, close and disconnection of "Server"	Admin
	Refreshing the list of "Server"	User
	Creating new connection, disconnection of "Client"	Admin
	Refreshing the list of "Client"	User
Servo debug	Open	User
Vision calibration	Setting and calibration	Admin
Conveyor tracking	Setting and calibration	Admin
One-drive-many vision function	All operations	Admin
Program lock	Locking, changing password	Admin
Real-time data interaction	Import, export, delete	Admin

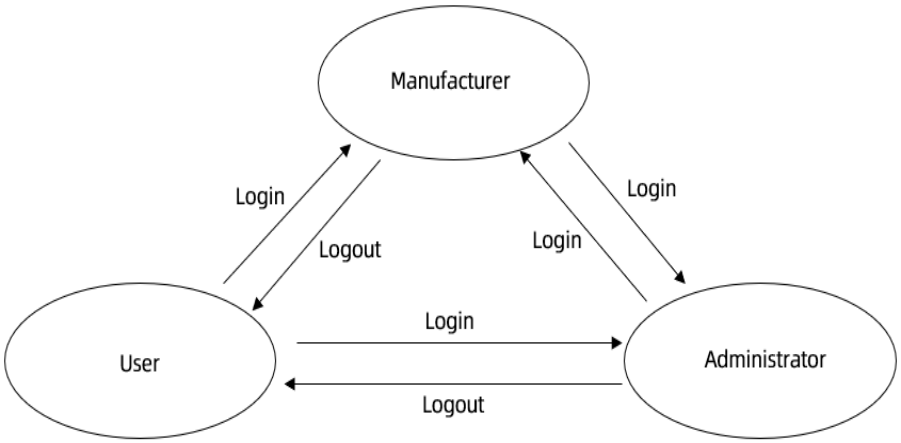
Function	Operation permission	Role
Lifetime prediction	All operations	Manufacturer
Program template	Create	Manufacturer
I/O monitor	All operations	User
AD monitor	All operations	User
Global variable monitor	All operations	User
Device connection status	Refresh	User
Bus connection status	All operations	User
Current protection	Reset max value	User
Load	All operations	User
Robot settings > Structure and kinematics	Save	Manufacturer
Robot settings > Arm load	Save	User
Robot settings > Installation method	Save	Manufacturer
Peripheral configuration > Fieldbus slave configuration > Switch settings	Switch and save	Admin
Peripheral configuration > Fieldbus slave configuration > Address allocation settings	Save	Admin
Peripheral configuration > Fieldbus slave configuration > I/O mapping settings	Save	Admin
Peripheral settings > IR-Link settings	Save	Admin
Motion parameters > Run parameter settings	Save	Admin
Motion parameters > Axis parameter settings	Save	Manufacturer
Motion parameters > Collision detection	Save	Admin
Motion parameters > Trajectory recovery settings	Save	Manufacturer
Motion Parameters > Advanced > Closed loop vibration suppression	Save	Admin

Function	Operation permission	Role
Motion Parameters > Advanced > Self-learning vibration suppression	Data backup, load data, restore data to defaults, and clear data	Admin
	Display "Clear data" option on the main page	User
Motion Parameters > Advanced > Torque model correction	All operations	Admin
Zero settings > Absolute zero	Save	Admin
Zero settings > Work origin	Save	User
Zero settings > Homing calibration	All operations	Admin
Workspace > Axis limit settings	Save	Admin
Workspace > Interference zone settings	Take points, save	Admin
Arm parameter offset mode	Save	Admin
System settings > System backup and load	Configuration file backup, load configuration file, and load point file	Admin
	System upgrade, restore factory settings, and format memory card	Manufacturer
System settings > Control permission	Save	Admin
System settings > Safety door-triggered stop mode	Save	Admin
System settings > Flying trigger I/O	Save	Admin
System settings > Network configuration	Save	Admin
System settings > Time settings	Sync time to controller	User
System settings > Emergency stop trigger method	Save	Admin
System settings > Emergency stop mode	Save	Admin
System settings > Program quantity settings	Save	Admin
System settings > Start button work mode	Save	Admin

Function	Operation permission	Role
System settings > Line number reset policy	Save	Admin
System settings > Remote stop strategy	Save	Admin
System settings > Robot maintenance	View	User
System settings > Robot maintenance	Maintenance	Manufacturer, Admin
Debug > Network debug	Ping	User
Debug > COM switch	Save	User
Debug > System diagnostics	Start, stop, and export	User
Login	All operations	User
Axes configuration (offline)	Save	User
Project file-related operations	All operations	User
Debugging-related operations	All operations	User
Main menu > View	Reset layout, station, control panel, output, results, variable monitor, alarm history, and operation log	User
About > Version	Basic information	User
	Detailed information	Manufacturer
Guide	User Guide, Instruction Guide, Remote I/O User Guide, Remote Ethernet User Guide	User
Control panel	Enable, clear alarm, emergency stop, and switch mode	User
System resources > Process module	All operations	Admin
Other unmentioned operations	All other unmentioned operations	User

NOTE

- The following figure shows the login and logout relationship of the three user types.



- By default, the InoRobotLab software is opened with user account.
- To log in with the admin or manufacturer account, a password is required.

4.5 Switching Control Permission

The actions of the robot controller and the robot manipulator are controlled by the upper-layer devices including InoRobotLab, teach pendant, remote Ethernet client, remote I/O unit (effective in both auto and manual modes), and remote I/O unit (only effective in auto mode).

The InoRobotLab and the teach pendant share one port and have the same highest priority among all upper-layer devices.

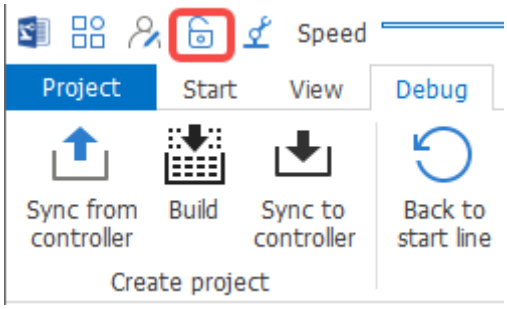
If you need to control the robot with an upper-layer device other than InoRobotLab or teach pendant, you can switch control permission from InoRobotLab or teach pendant to another target upper-layer device.

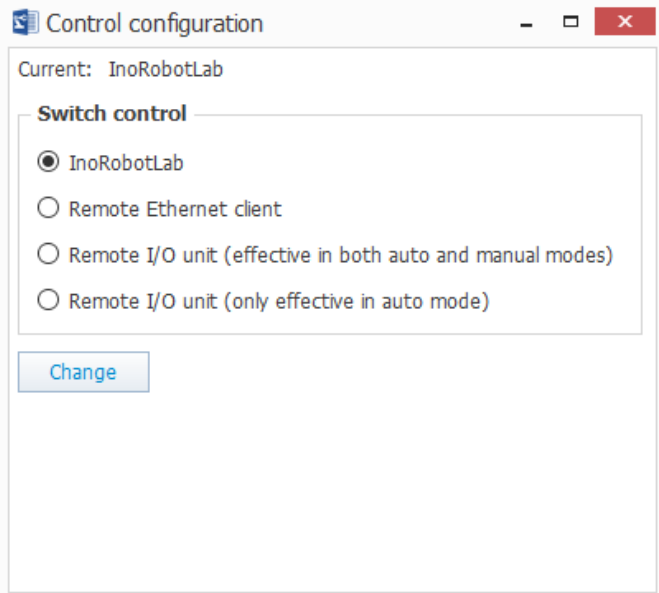
Steps

The current control permission is displayed in the bottom status bar.



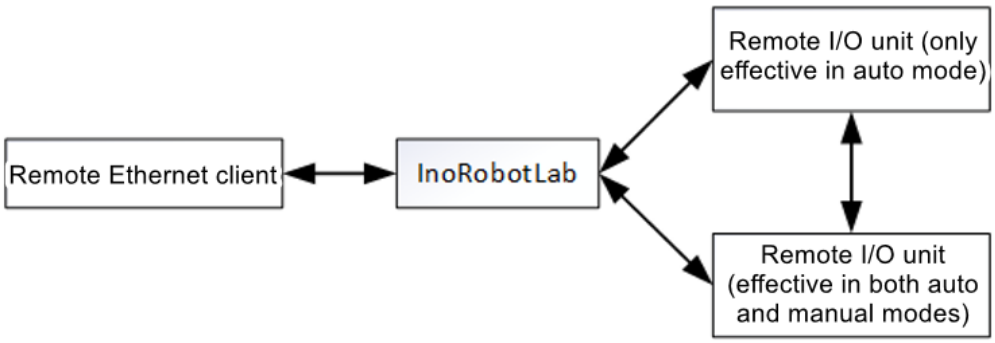
You can switch the control permission by clicking the control permission icon in the upper-left corner of the toolbar.





CAUTION

- The current control permission displayed after the software is opened is not necessarily InoRobotLab. It depends on your previous setting.
- The control permission can only be switched between InoRobotLab and other devices. However, there is a special case in which the control permission can be directly switched between remote I/O unit (effective only in auto mode) and the remote I/O unit (effective in both auto and manual modes). The switch scenarios are shown in the figure below.
  - Switching between a remote Ethernet client and a remote I/O unit (effective only in auto mode) is not supported.
  - Switching between a remote I/O unit (effective only in auto mode) and a remote I/O unit (effective in both auto and manual modes) is supported.



Description of control permissions

- When the control permission is switched from InoRobotLab to another device, the robot actions are controlled by another device. In this case, InoRobotLab only displays the robot status, such as the I/O status, running program line number, etc. When the control permission is switched from another device to InoRobotLab, the robot actions are controlled by InoRobotLab.
- When the control permission is granted to remote I/O unit, it is necessary to set the startup speed percentage for the first running. Normally, the startup speed is set to be lower than 100% for operation safety.

4.6 Clearing Warnings and Alarms

Steps

When a warning or alarm occurs, you can try to clear it by clicking the button on the control panel, as shown in the figure below.



Results

- If the fault is resolved and the warning or alarm no longer exists, the status changes to normal.
- If the fault is not resolved and the warning or alarm persists, the status will still be displayed as warning or alarm.

4.7 Emergency Stop

The control panel displays/allows you to operate the emergency stop status of the robot. For emergency stop, see Emergency Stop Mode.

Steps

Click the "Emergency stop" button on the control panel.

- When the "Emergency stop" button displays "OFF", clicking the button will trigger the emergency stop.
- When the button displays "ON", clicking it will cancel the emergency stop.

Results

- After the emergency stop is triggered, the "Output" pane prints "Emergency stop pressed" and the emergency stop button displays "ON".
- After the emergency stop is canceled, the "Output" pane prints "Emergency stop released" and the emergency stop button displays "OFF".

4.8 Enabling/Disabling the Robot

The control panel displays/allows you to operate the robot status.

Steps

Click the "Servo" button on the control panel.

- When the button displays "OFF", click to enable.
- When the button displays "ON", click to disable.



- If the robot is enabled in teaching mode and then InoRobotLab software is closed, the robot will be automatically disabled.
- When the robot is in the alarm state (except for position limit alarm in manual mode), it is not allowed to enable the robot.

Results

- After the robot is enabled, the Output pane prints "Controller enabled" and the enable button displays "Servo ON".
- After the robot is disabled, the Output pane prints "Controller disabled" and the enable button displays "Servo OFF".

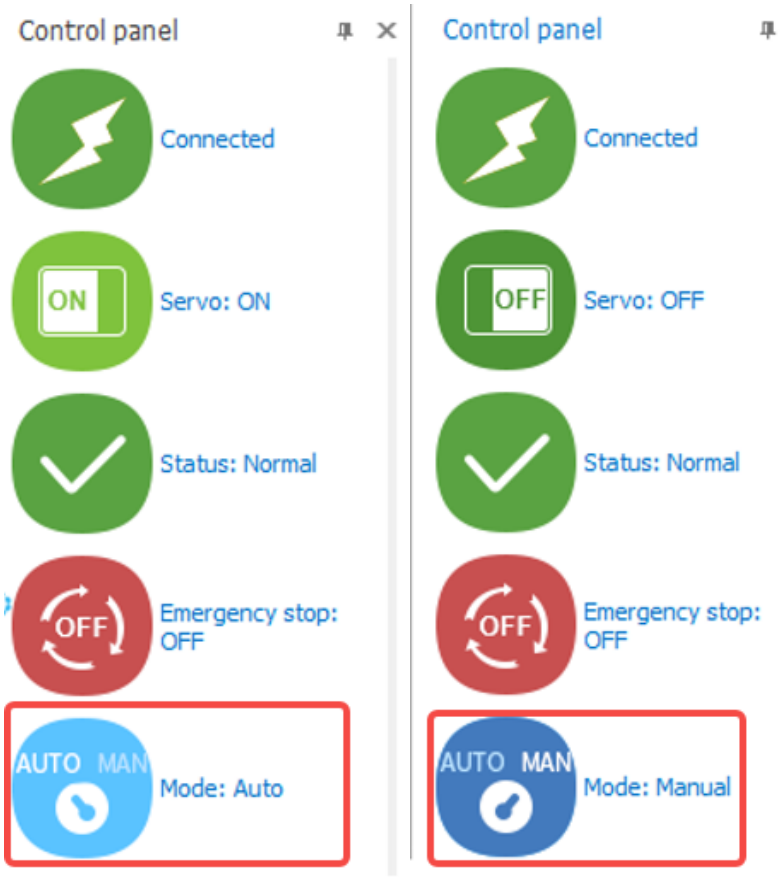
4.9 Switching between Manual and Auto Modes

The control panel displays/allows you to switch manual/auto mode. Manual mode is used for program editing and debugging. Auto mode is used for real-world high speed running.

Steps

Click the "Mode" button on the control panel.

- When the mode is "Manual", clicking the button will switch the mode to "Auto".
- When the mode is "Auto", clicking the button will switch the mode to "Manual".



Results

- After the mode is switched to "Manual", the Output pane prints "Switched to manual mode" and the mode switch button displays "Manual".
- After the mode is switched to "Auto", the Output pane prints "Switched to auto mode" and the mode switch button displays "Auto".

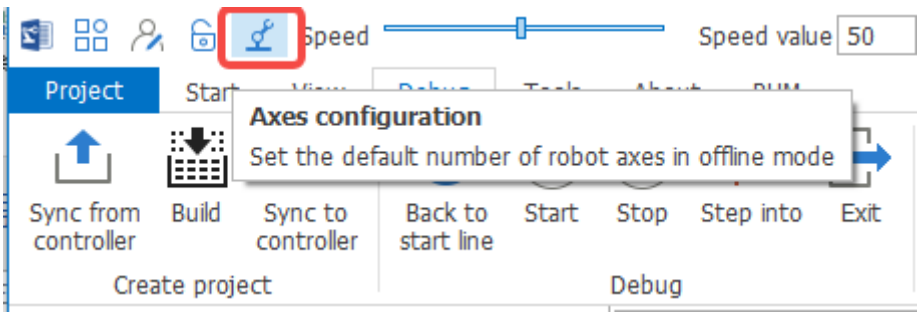
4.10 Configuring the Default Number of Axes Offline

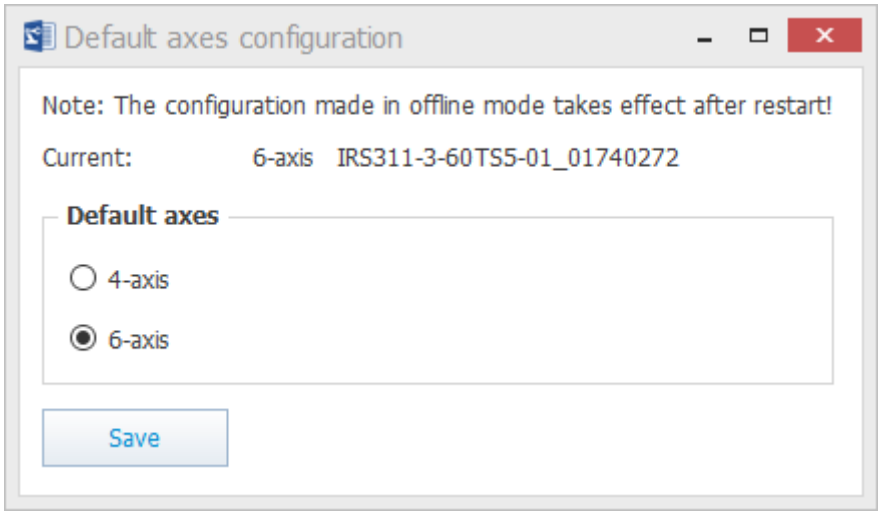
Some InoRobotLab software interfaces displayed or program logic are related to the number of axes of the target robot (for example, programs written for a 4-axis robot should not involve axis #5 or #6). You can set the number of robot axes so that you can offline programming without connection to the controller.

This option is used only for offline programming and is independent of the actual number of axes obtained after the controller is connected.

Steps

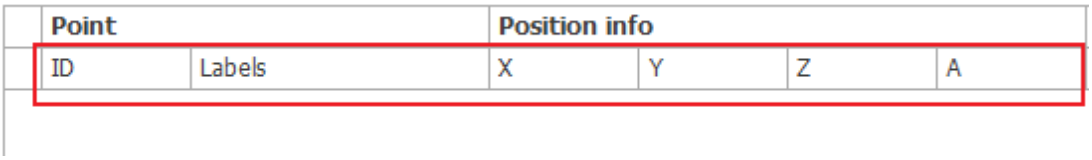
Click the axes configuration button, select the number of axes in the pop up window, and click "Save".





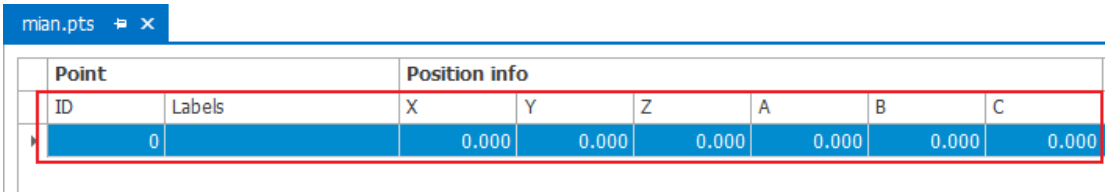
Results

After setting the default number of axes, restart the software to make the settings take effect. You can see that the number of axes shown in the point file matches the set number of axes.



Point		Position info			
ID	Labels	X	Y	Z	A

Point data for 4-axis robot



mian.pts

Point		Position info					
ID	Labels	X	Y	Z	A	B	C
0		0.000	0.000	0.000	0.000	0.000	0.000

Point data for 6-axis robot

4.11 Axis Operation

4.11.1 Coordination

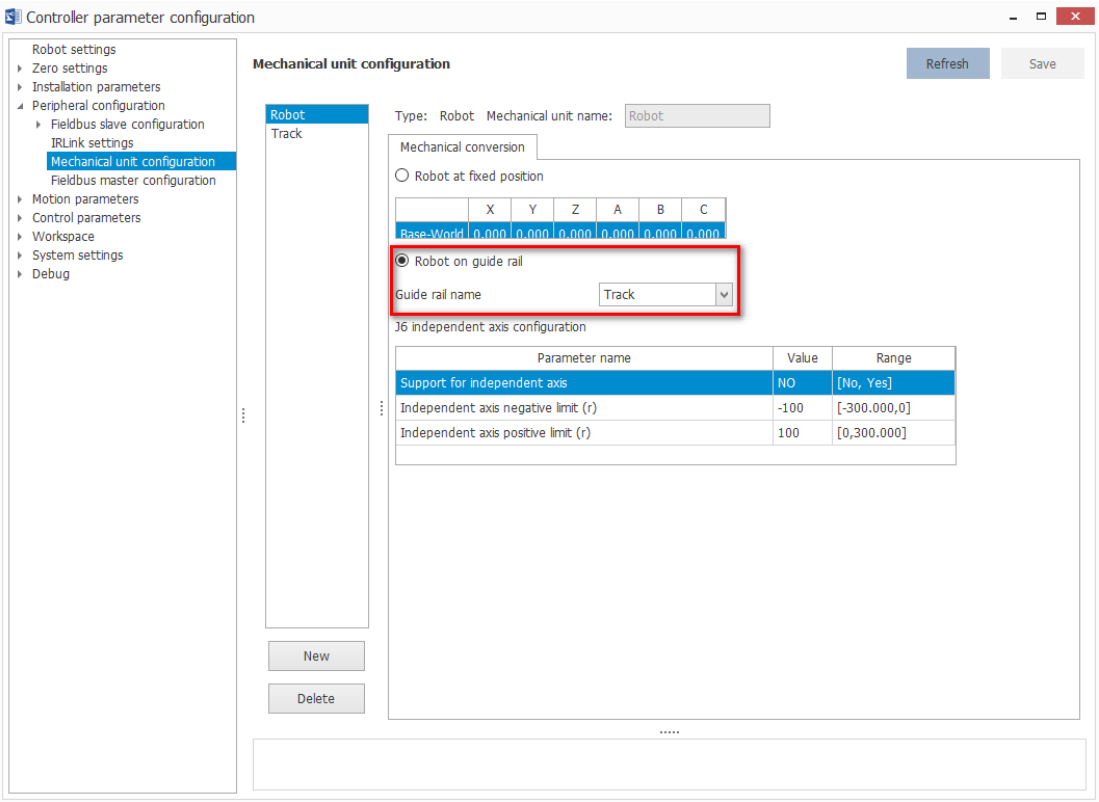
Currently, two types of coordinated movement are supported in the robot gripper scenario.

- Coordination between the robot and the base guide rail: When the guide rail is moved manually, the robot coordinates with the guide rail to keep the position of TCP in the world frame unchanged.
- Coordination between the robot and the positioner: When the positioner is moved manually, the robot coordinates with the positioner to keep the position of the TCP in the workobject frame unchanged.

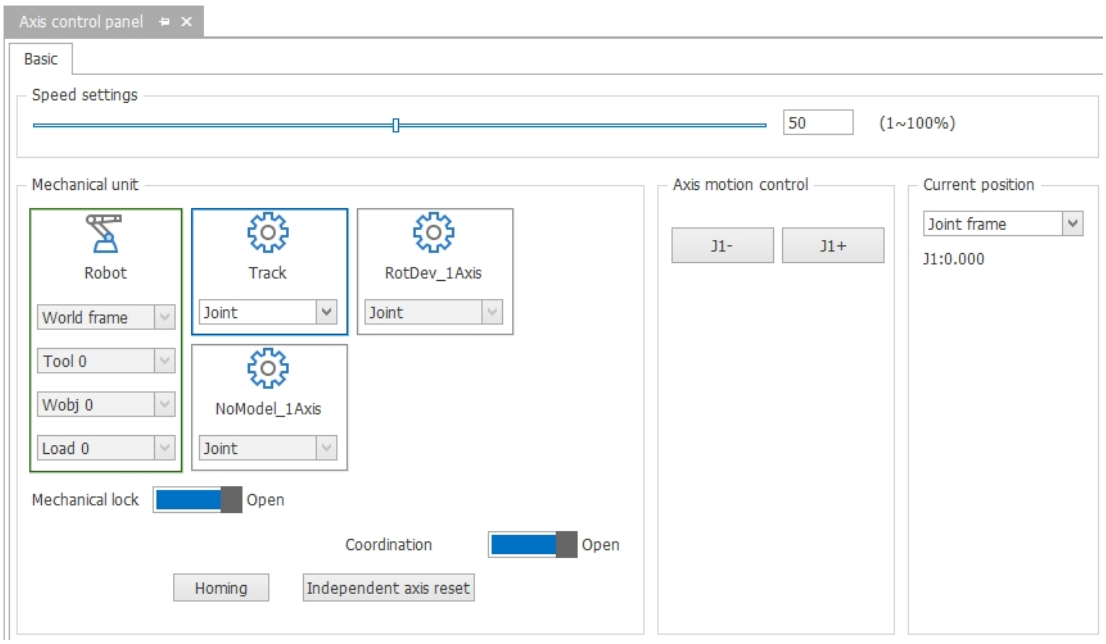
Steps

- Coordination between the robot and the base guide rail

1. Place the robot on guide rail and set the mechanical unit name of the guide rail.

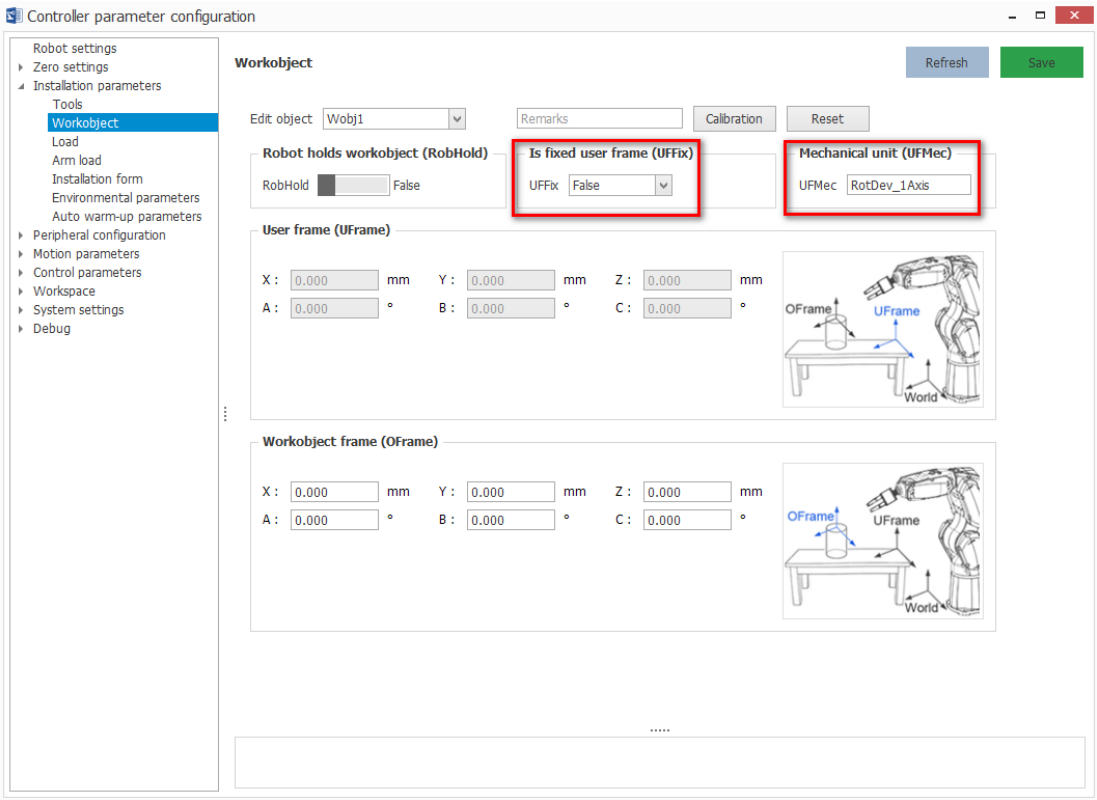


2. When the current mechanical unit is switched to the base guide rail in the axis control panel and the robot is a world frame or workobject frame, the robot automatically starts coordination, as shown below.

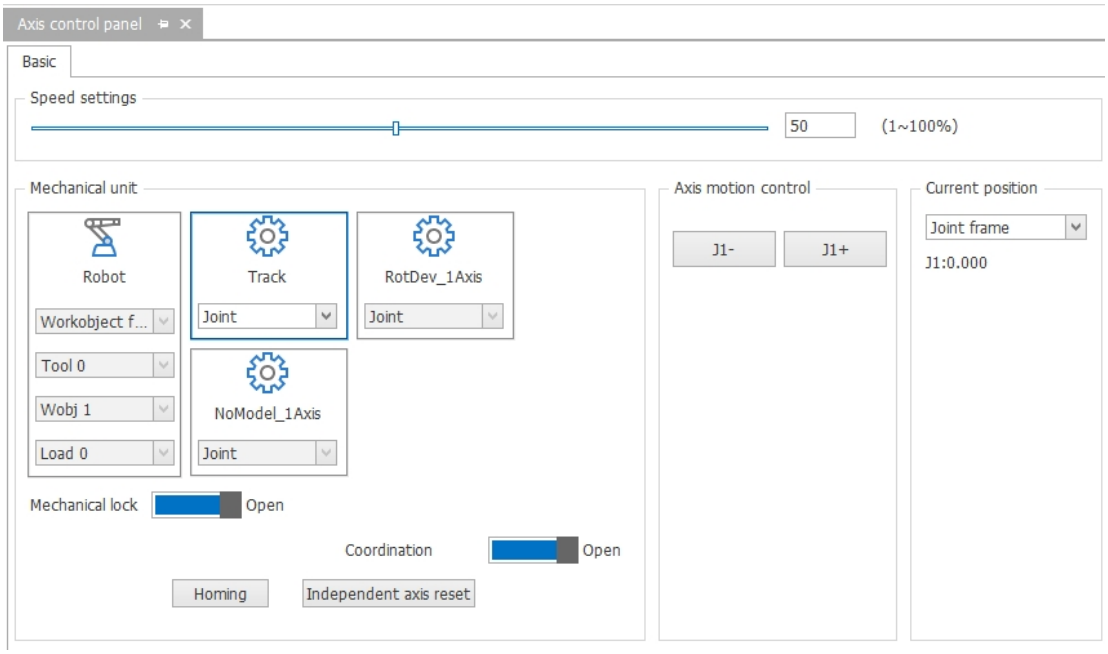


- Coordination between the robot and the positioner

1. Set UFFix to "False" and UFMec to the name of positioner coordinating with the robot.

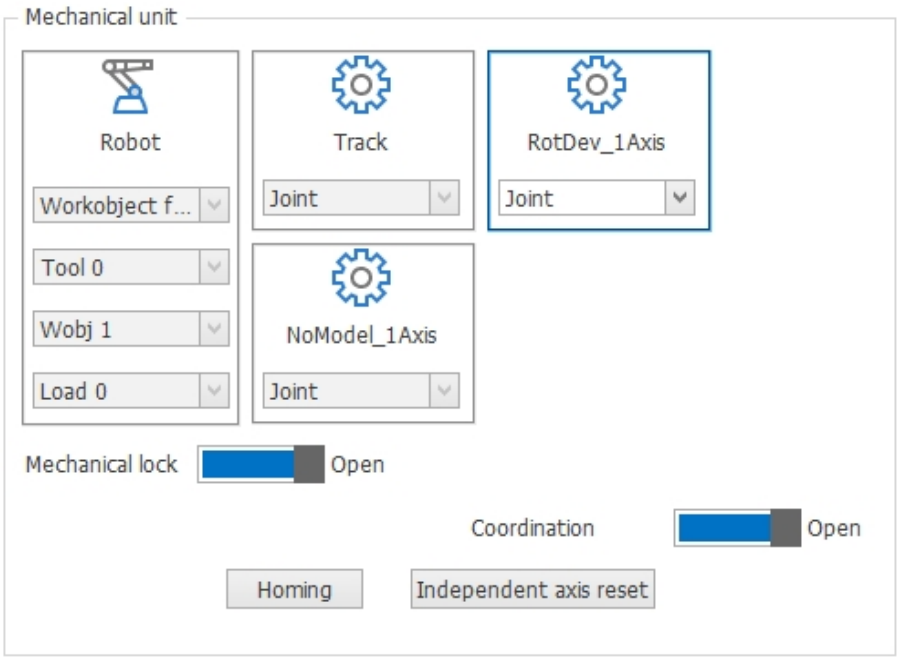


2. When the current mechanical unit is switched to the positioner in the axis control panel and the robot is a workobject frame and a configured workobject is selected, the robot automatically starts coordination, as shown below.

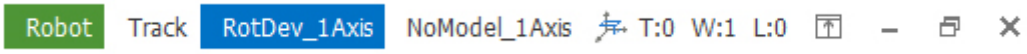


Results

When the robot is in the coordination state, the robot is displayed with a green border in the axis control panel.



Meanwhile, in the upper right corner of the software, the robot is displayed in white characters with a green background.

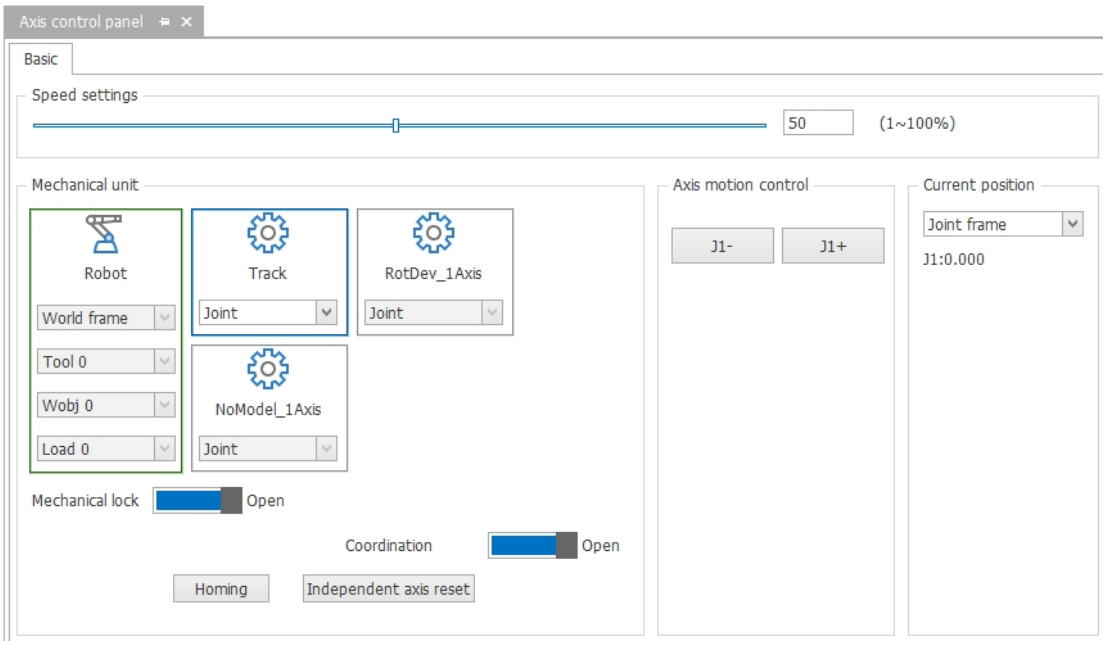


4.11.2 Switching Mechanical Unit

In the axis control panel, you can switch between different mechanical units to set and operate the currently selected mechanical unit.

Steps

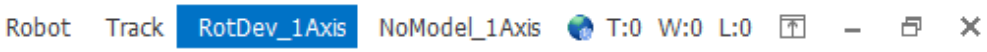
Click the visualization area of the mechanical unit to switch between different mechanical units.



Results

The active mechanical unit is displayed with a blue border in the axis control panel, and the inactive mechanical unit is displayed with a gray border.

Meanwhile, in the upper right corner of the software, the active mechanical unit is displayed in white characters with a blue background.



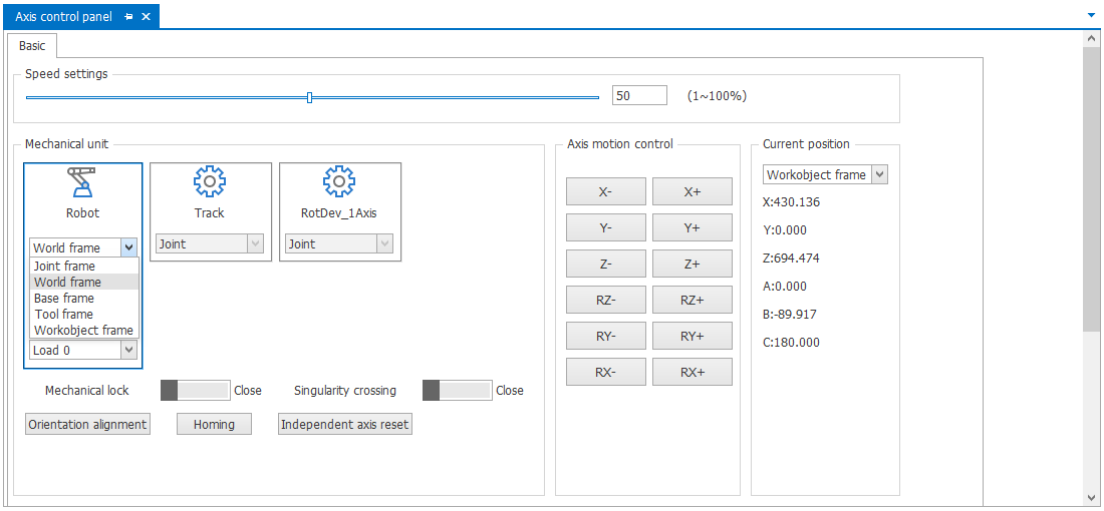
4.11.3 Switching Frame

Inovance robots contain multiple frames. The frames that can be used on InoRobotLab include joint, world, base, tool, and workobject frames. You can switch between different frames to make teaching and programming easier.

- When you switch to the joint frame on InoRobotLab, the coordinate values displayed on the interface are the joint values of the current mechanical unit. The joint values are the rotation angles of the robot arm relative to the zero point position.
- When you switch to a non-joint frame on InoRobotLab, the current position has different values under different frames. You can select a position format as needed.

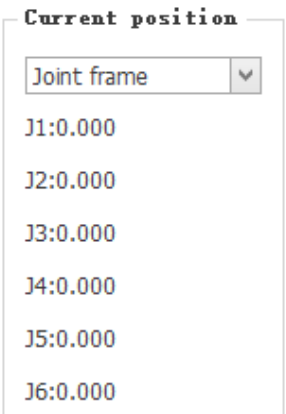
Steps

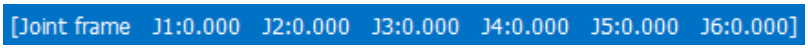
In the axis control panel, switch the frame currently used.



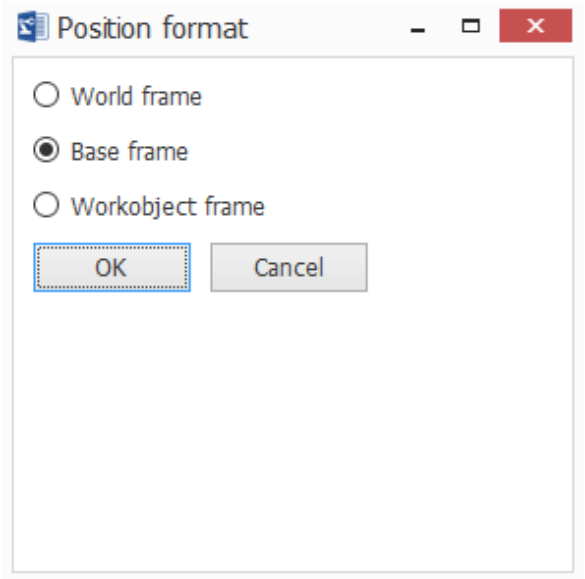
Results

- When you switch to a joint frame, the "Current position" in the axis control panel and the bottom status bar display the joint values.





- When you switch to a non-joint frame, you can switch the position format by clicking "Current position" in the axis control panel or the position in the bottom status bar.

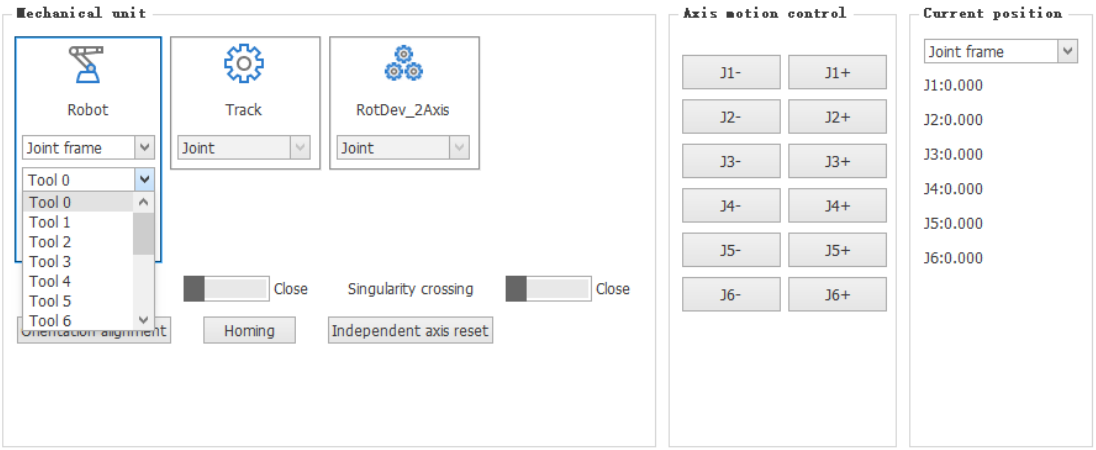


4.11.4 Switching Tool Number

If the tool is changed, you need to switch the tool number.

Steps

In the Robot mechanical unit on the axis control panel, switch the current tool number.



NOTE

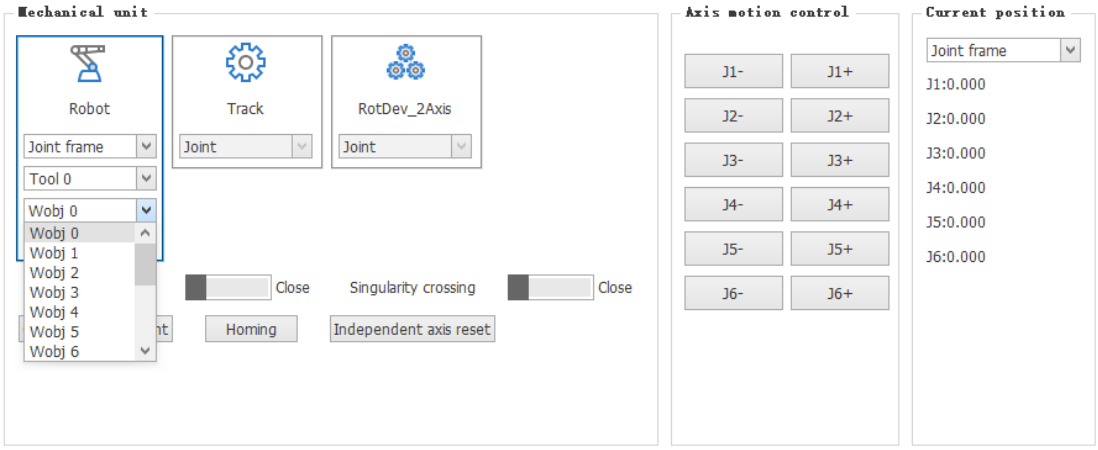
- You cannot switch the current tool number when the main task is in running, when InoRobotLab is in auto mode, or when the control permission is not granted.
- If the tool number to be switched is not Tool 0 and the load mass is 0, the switchover fails.

4.11.5 Switching Workobject Number

If the workobject is changed, you need to switch the workobject number.

Steps

In the Robot mechanical unit on the axis control panel, switch the current workobject number.



NOTE

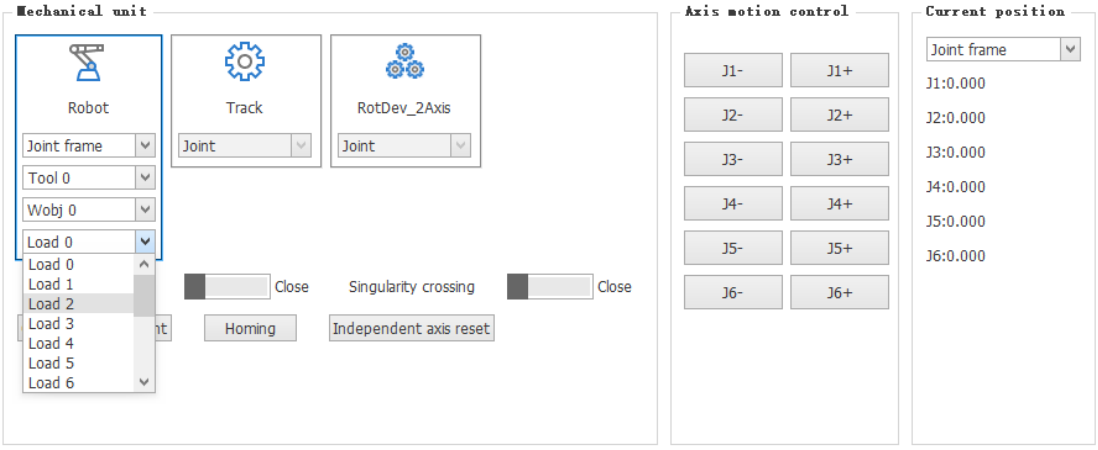
You cannot switch the current workobject number when the main task is in running, when InoRobotLab is in auto mode, or when the control permission is not granted.

4.11.6 Switching Load Number

If the grip load is changed, you need to switch the load number.

Steps

In the Robot mechanical unit on the axis control panel, switch the current load number.



NOTE

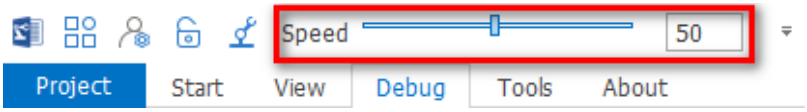
- You cannot switch the current load number when the main task is in running, when InoRobotLab is in auto mode, or when the control permission is not granted.
- If the load number to be switched is not Load 0 and the load mass is 0, the switchover fails.

4.11.7 Adjusting Speed

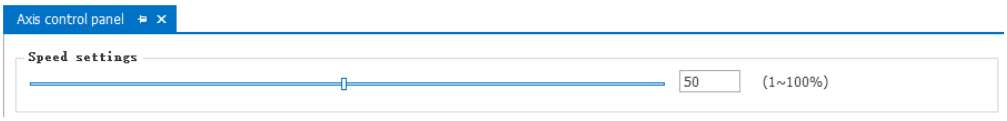
You can adjust the robot motion speed as needed.

Steps

- You can change the motion speed in the upper left corner of the software, as shown below.



- Also, you can change the motion speed on the "Axis control panel" page, as shown in the figure below.



Results

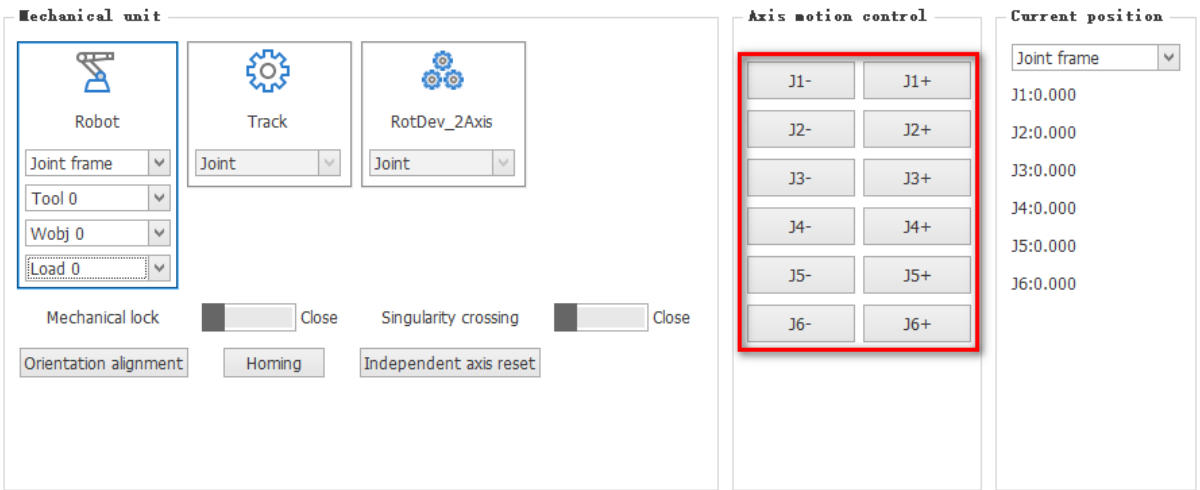
The speed change takes effect immediately. You can test whether the robot motion speed meets the setting value.

4.11.8 Axis Motion

Axis motion is used to control the continuous motion or jog movement of the robot.

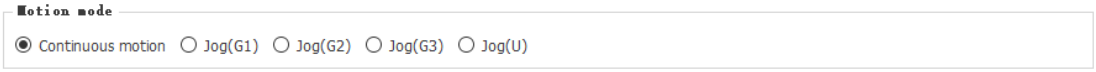
Steps

1. Open the "Axis control panel" page in Tools.
2. Select "Continuous motion" or "Jog".
3. Select the frame, tool number, workobject number, and load number, and set the speed and then click the button in the axis motion control panel to start the motion. Press the button to start motion and release the button to stop.



> **NOTE**
>
> - You can select "Continuous motion" or "Jog" for axis motion.
> - In case of continuous motion, the robot moves continuously when the button is pressed and stops immediately when the button is released.
> - In case of jog motion, the robot moves by a specified distance when the button is pressed and stops until the movement is complete regardless of whether the button is released or not.

Jog



The step size for jog motion includes G1, G2, G3, and user-defined step size.

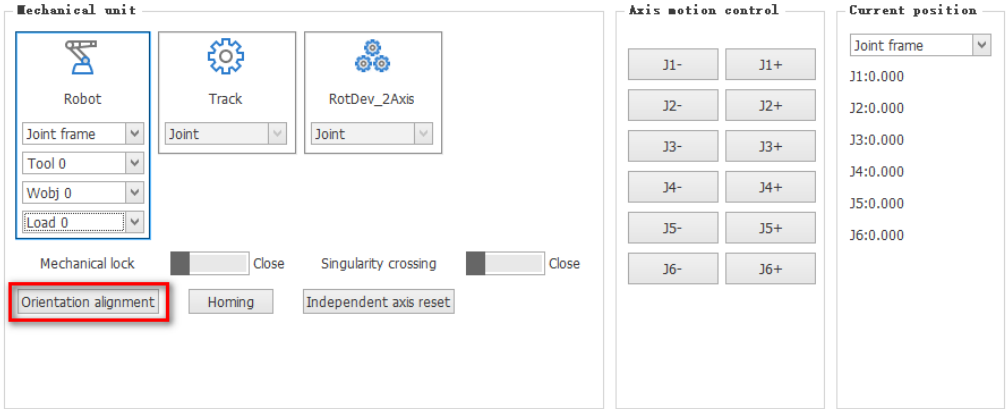
Name	Level	Joint step size (°)	Position step size (mm)	Orientation step size (°)
Jog (G1)	Short	0.05	0.05	0.05
Jog (G2)	Medium	0.5	0.5	0.5
Jog (G3)	Long	2	2	2
Jog (U)	User-defined	Max. 10 Min. 0.01	Max. 10 Min. 0.01	Max. 10 Min. 0.01

4.11.9 Orientation Calibration

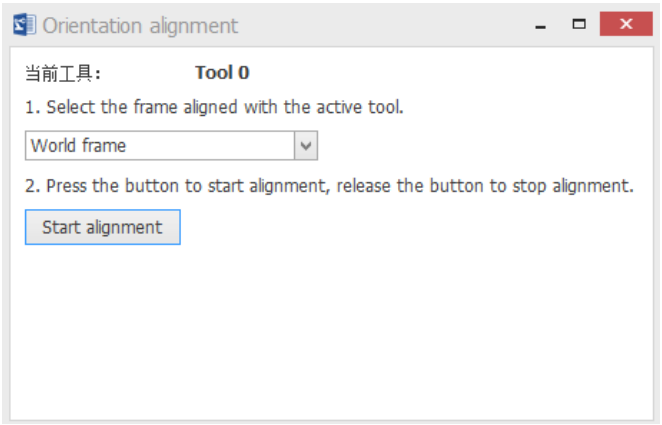
The orientation calibration function takes effect only for 6-axis robots. The principle is to adjust the orientation of the tool so that the Z axis of the current tool is aligned with the nearest axis of a certain frame, for the purpose of accurate control in real time.

Steps

1. Switch to the robot mechanical unit in the axis control panel, confirm the tool number, and click the "Orientation alignment" button.



2. In the pop up dialog box, select the frame to which the current tool is aligned, and then click and hold the "Start alignment" button until the alignment is complete.



Grip tool scenario	
World frame	Align the tool's Z axis to the world
Base frame	Align the tool's Z axis to the base
Workobject frame	Align the tool's Z axis to the workobject

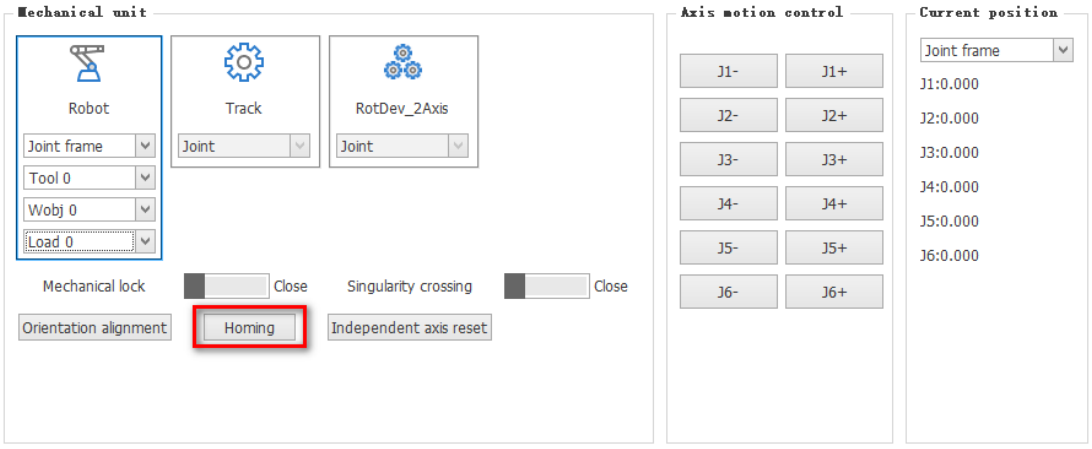
External tool scenario	
World frame	Align the flange's Z axis to the world
Base frame	Align the flange's Z axis to the base
Workobject frame	Align the tool's Z axis to the workobject
Flange frame	Align the tool's Z axis to the flange

4.11.10 Homing

Returns all joints of the selected mechanical unit to the zero value.

Steps

On the axis control panel, click the "Homing" button to start homing and release the button to stop.



Results

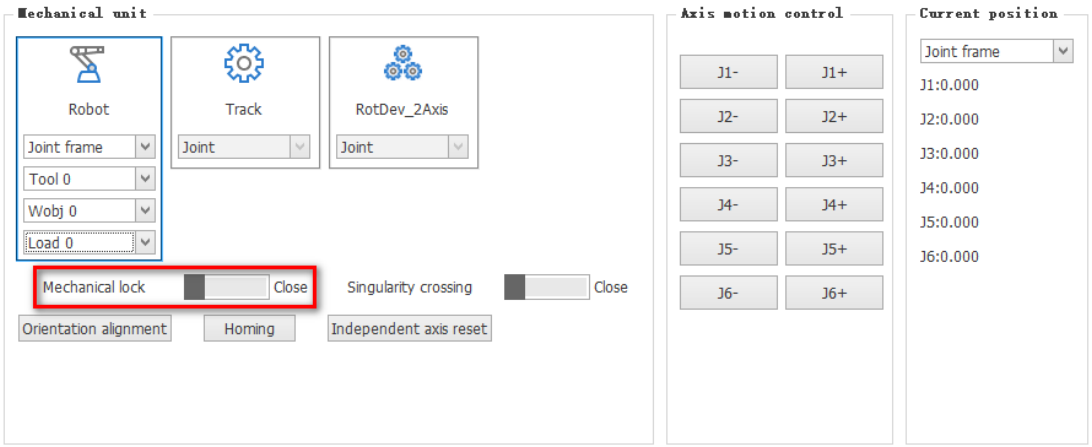
The homing stops automatically after all joints of the current mechanical unit return to the zero value.

4.11.11 Mechanical Lock

When the mechanical lock is enabled, the robot cannot move. However, other robot functions are normal. This helps to verify the logic of the program.

Steps

In the left directory tree of the software, click "Tools" and double click "Axis control panel" to open the axis control panel. Click "Mechanical lock" to toggle the open/close state of the mechanical lock, as shown in the figure below:



Results

Move the robot in axis motion control panel and verify whether the mechanical lock configuration takes effect by observing whether the robot actually moves.

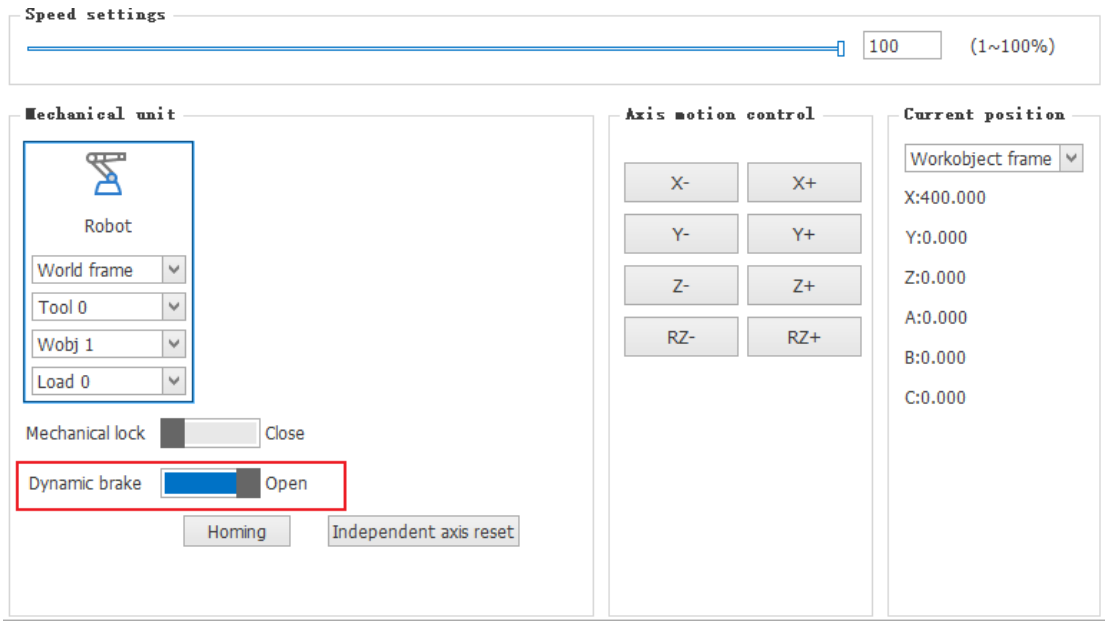
> **NOTE**
>
> It is not allowed to configure the mechanical lock when the robot is enabled and the main task is running, when InoRobotLab is in auto mode, or when the control permission is not granted.

4.11.12 Dynamic Brake

For the SCARA robots, the robot axis can be pushed more easily by turning off the dynamic brake switch after the robot is disabled.

Steps

Go to "Tools" > "Axis control panel". The dynamic brake switch is displayed for the SCARA robots.

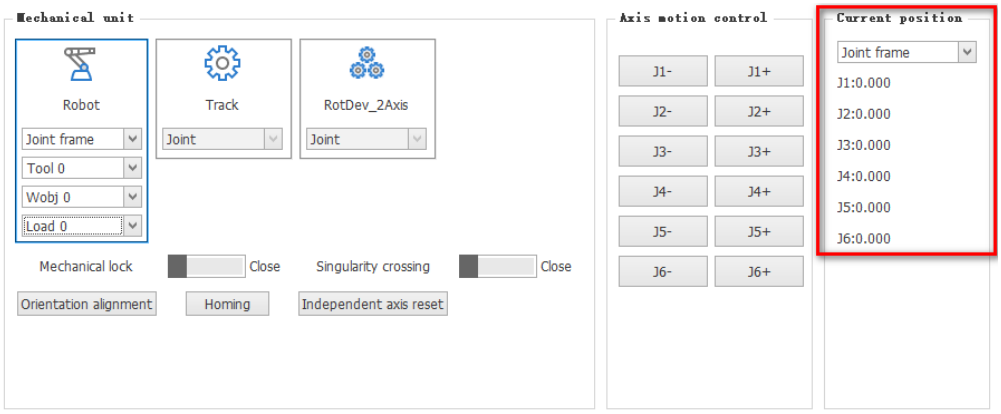


NOTE

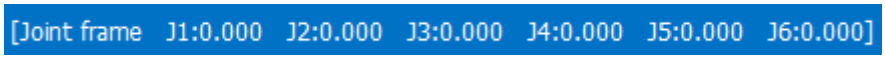
- This function applies to the J1, J2, and J4 axes of the SCARA robots.
- The dynamic brake switch can be turned on or off only when the robot is disabled.
- The dynamic brake switch is always ON when the robot is enabled.
- When the controller is powered on again, the dynamic brake switch is restored to the ON state.

4.12 Viewing Current Position

The "Axis control panel" page displays the frame and coordinate values of the active mechanical unit.

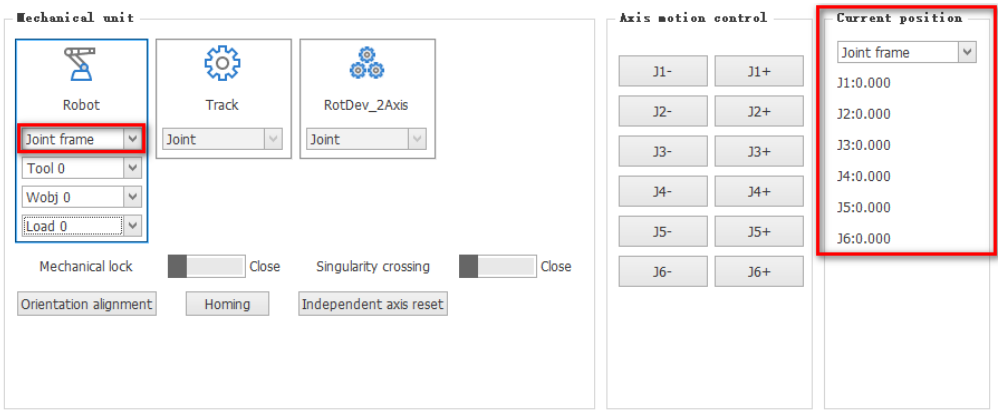


You can view the current frame and coordinate values in the status bar at the bottom of the software.

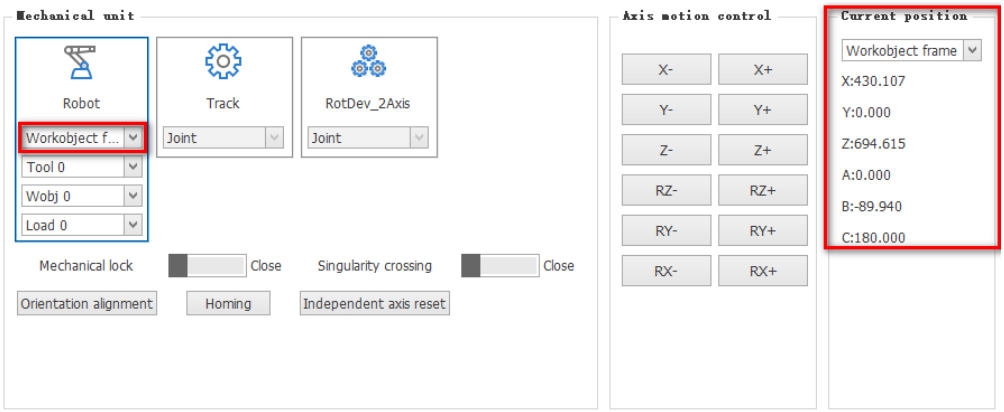


Steps

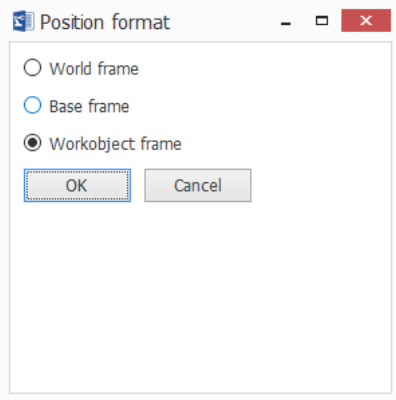
- When the active mechanical unit is a robot and the frame of the mechanical unit is selected as joint frame, the "Current position" pane always displays the joint frame and the coordinate value of each joint.



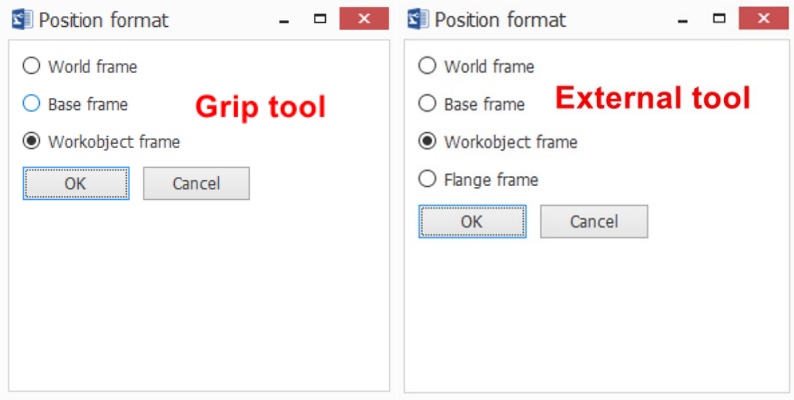
- When the active mechanical unit is a robot and the frame of the mechanical unit is selected as a non-joint frame, the "Current position" pane displays the workobject frame and coordinate values by default.



You can switch the display of the TCP values in world frame, base frame, and workobject frame, as shown in the figure below.



- For grip tools, you can switch between the world/base/workobject frames; and for external tools, you can switch between the world/base/workobject/flange frames, as shown in the figure below.



- When the tool conflicts with the workobject, for example, if both the tool and workobject are configured to be held by the robot at the same time, the message "The active tool does not match the workobject type and the current position cannot be displayed." will be displayed.



NOTE

The frame that can be switched for the current position has different meanings than the frame in the mechanical unit.

4.13 Calibrating the Tools

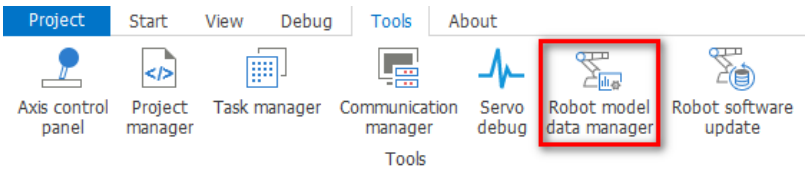
For details, see 9.3.1 Tools.

4.14 Calibrating the Workobject

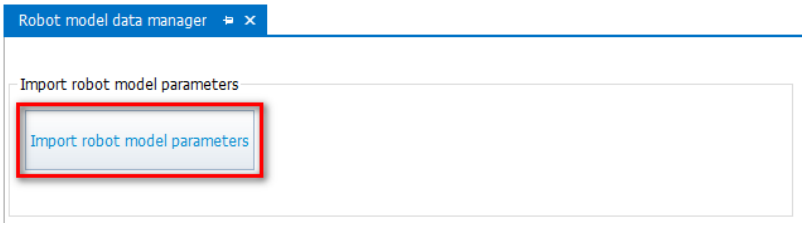
For details, see 9.3.2 Workobject.

4.15 Robot Model Data Manager

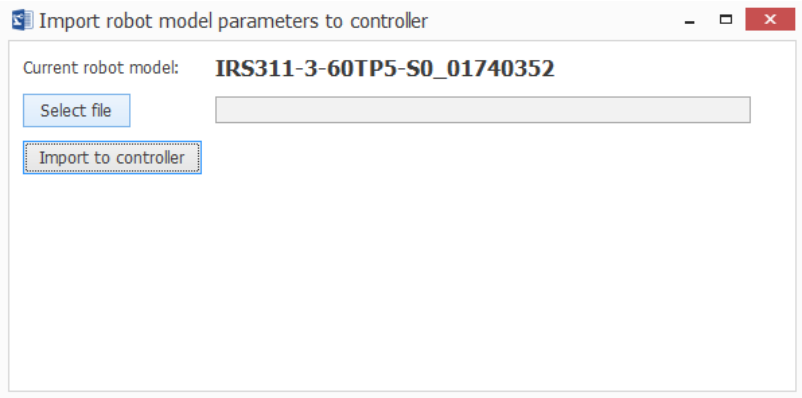
The function entry is shown in the figure below:



1. After clicking the "Robot model data management" button, you can enter the robot model data management interface, where you can use the "Import robot model parameters" function.

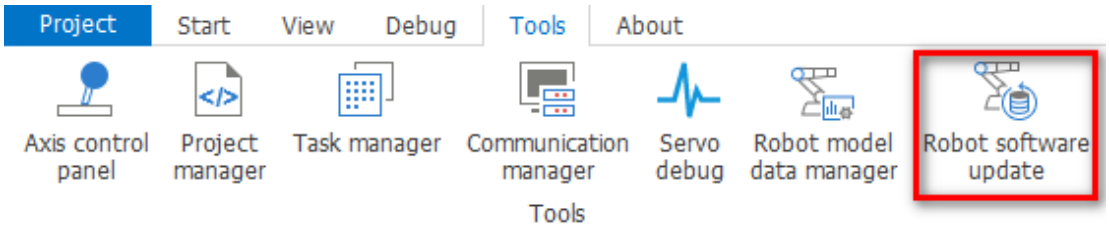


2. Select the robot model parameter package to import to the controller.

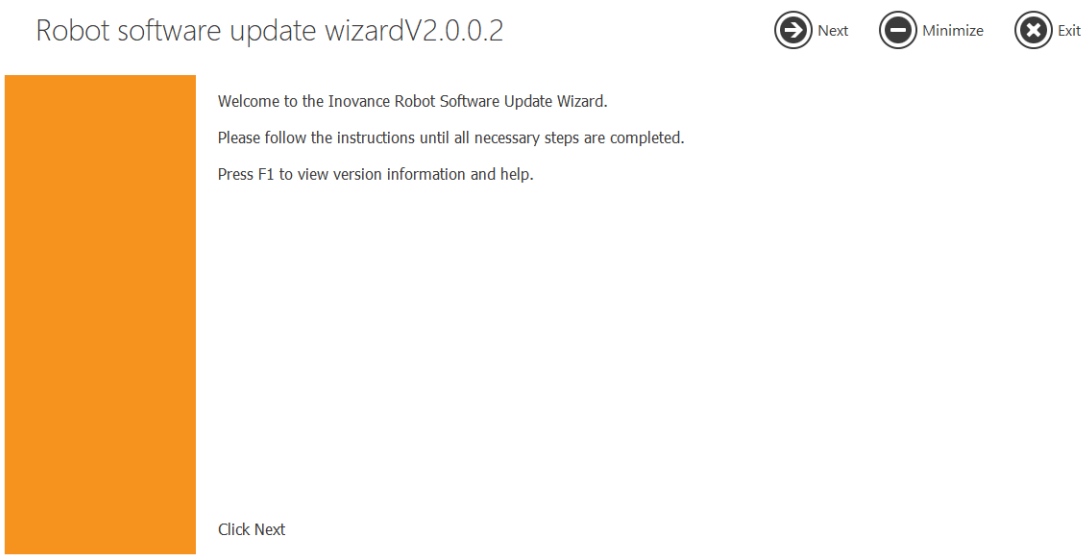


4.16 Robot Software Upgrade

The "Robot software update" function allows you to upgrade or flash the controller. You can access this function as shown in the figure below.



1. Click the "Robot Software Update" button to open the interface:



2. When performing software updates, you can press the F1 key to open the help guide and follow the instructions until all necessary steps are completed.

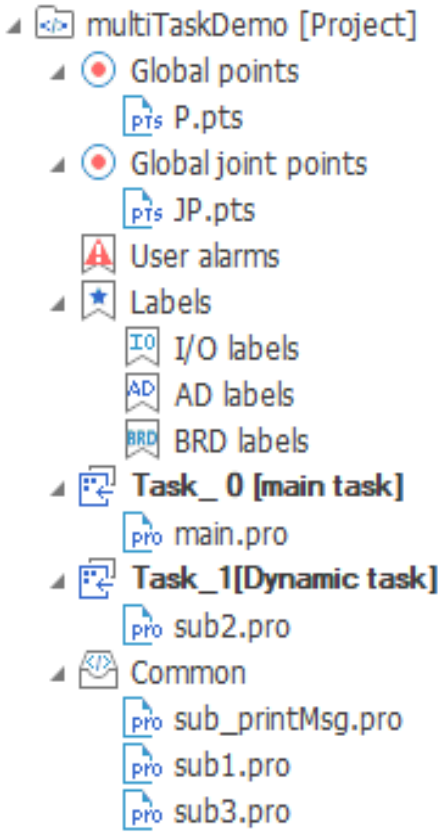
5 Project Management

5.1 Basic Concepts

5.1.1 Project

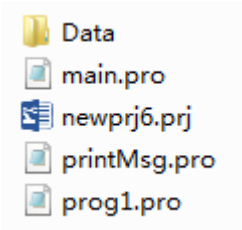
By default, the project consists of program files, global points, labels, user alarms, and task configuration. If you have used 5.2.3 Backing Up to Local Project or 5.2.4 Backing Up the Current Project function, the project will include tool parameters, user parameters, and load parameters. During program execution, use these elements according to the execution logic to enable the robot to perform the required work.

The project is presented in a tree structure as shown below.



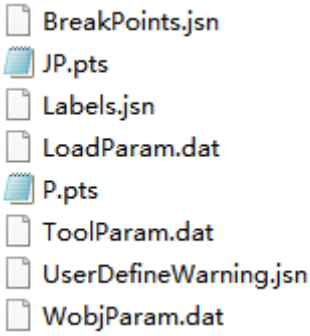
As shown in the above figure, a task node in bold font, such as "Task_0 [Main task]" indicates that the task is active; otherwise, the task is inactive. The project name is the *.prj file name without the extension.

The following figure shows the file level of the project on the local disk.



It contains the "Data" folder, program file (*.pro), and project information file (*.prj). The "main.pro" file is the project's main task entry, created by default and with a fixed name.

The "Data" folder contains the files shown in the following figure.



No.	File name	Description
1	*.prj	The project configuration file, such as newprj.prj, corresponding to a project named "newprj".
2	main.pro	The main task entry, a multitasking entry program file or public program file for other "*.pro" files.
3	Labels.jsn	Label information file, containing I/O labels, AD labels, and BRD variable labels. The label is an alias of a variable and helps improve the readability of the program. A proper label name can reduce the memory load and improve programming efficiency.
4	P.pts	Default global position point file (after you create a new position point file, there will be other point file with a ".pts" suffix in the "Data" folder.)
5	JP.pts	Default global joint point file
6	UserDefineWarning.jsn	User-defined alarm file. You can customize certain alarm contents and implement customized alarms through program call.
7	RobPointMappingFile.dat	Position point file configuration, detailed explanations are provided in the multi-point position configuration function.
8	BreakPoints.jsn	Breakpoint information file, may be absent.
9	LoadParam.dat	Load parameter file, may be absent.
10	ToolParam.dat	Tool parameter file, containing parameters of the tool frame, calibration points, load parameters, and may be absent.
11	WobjParam.dat	Workobject parameter file, may be absent.

NOTE

- The program file contains task entry functions, functions, variables, and instructions. For details, see the "Industrial Robot Instruction Guide".
- If tool parameters, workobject parameters, and load parameters need to be moved along with the project, you can use 5.2.3 Backing Up to Local Project or 5.2.4 Backing Up the Current Project function to place the tool parameters, workobject parameters, and load parameters into the project.
- Directly adding, deleting, or modifying files or content in the project files through external means may lead to unforeseen risks and issues. Therefore, such operations are generally not allowed.
- When you open the InoRobotLab for the first time to create a new project, the project is stored in this directory by default: C:\Users\Administrator\Documents\Inovance\Projects.

5.1.2 Task

A task in the controller refers to a collection of program modules (files) for a specific function. It contains an entry module and other cross-referenced modules, with a total of 16 tasks, 0 being the main task, 1-14 being dynamic or static tasks, and 15 being xqt task.

5.1.3 Module

A module is a program file, and a module consists of several functions. Each entry module corresponding to a task has an entry function (that is, the main function) Start…End. The entry function cannot be included in a module that is not a task entry.

The structure of the program file is as follows:

```
ProgramInfo
    Version = "S03.22"
    Time = "2022/12/28 14:31:18"
    RobotName = "IRS311-3-60TS5-02_01740271"
EndProgramInfo

LP[0] = 336.150462, 76.303993, 780.364317, -105.073400,-84.947312, -60.341487; 0, 0, -1,
-1;0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000;
LP[1] = 321.518901, 149.369996, 867.772839, -115.648598,-42.102643,-32.806266; 0, 0, 0,
0;0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000;

Include "Sub1.pro";     ### Module import and variable definitions
Struct strname          ### Structure variable name
......                  ### Structure member definitions
EndStruct;
    Int m_var1;             ### Module internal variable definition
    Int m_var2;             ### Module internal variable definition
Start;                      ### Module entry function()
......
End;
Func FunctionName1(ParameterList)   ### Function definition
......                              ### Statements
EndFunc;
Func FunctionName2(ParameterList)   ### Function definition
.....                               ### Statements
EndFunc;
```

Program file naming convention: The suffix is ".pro". A filename consists of letters, numbers and underlines with a length of up to 26 characters, starting with a letter. Letters are not case-sensitive.

Limit on the number of instructions in the program file: The instruction section of the program can contain up to 2,000 lines of instructions.

To call a function of module B in module A: Call include "ModelB.pro" in module A to declare that module B is included, and then call it in the form of "module name.function name", such as "ModelB.Func1()".

5.1.4 Function

Functions are divided into main function (entry function of tasks) and custom functions. Each module can consist of a number of functions, but there is only one entry module in each task and one entry function in that module; no other module can contain an entry function.

Format of the main function

Start;

…function body…

End;

Custom function format:

Func function name (parameter list)

…function body…

EndFunc

Format of custom function with function return value: (Compatible with above custom functions without function return value)

Func <ReturnKind> <FuncName>(<parameters>)

…function body…

Return variable name;

EndFunc;

Format of custom function with function return value:

- Function: Custom function with function return value and its call.
- Description: A custom function can be returned with a return value.
- Supported function return value types: Byte, Bool, Int, Float, Double, String.



- When the function return value type is Byte, a variable of type Byte, B, and LB, or a constant of type Byte can be returned.
- When the function return value type is Bool, a variable of type Bool, or a constant of type Bool (True, False, 0, 1 supported) can be returned.
- When the function return value type is Int, a variable of type Int, R, and LR, or a constant of type Int can be returned.
- When the function return value type is Float, a variable of type Float, or a constant of type Float can be returned.
- When the function return value type is Double, a variable of type Double, D, and LD, or a constant of type Double can be returned.
- When the function return value type is String, a variable of type String, Str, or a constant of type String can be returned.

When defining function parameters, only custom variables or arrays of type Bool, Byte, int, float, double, and String are allowed to be defined, and the definition of structure variables is not supported. However, the parameters passed into the function allow the use of variables or arrays of type Bool, Byte, int, float, and double in structure members.

Function Call

The function call may be performed in two forms in the program:

- Calling a function in the current module
  Format: Function name (parameter list)
- Calling a function in another module
  Format: Module name.Function name (parameter list)



For calls to functions in other modules, include the module first with the "include" instruction.

For example: include "ModelA.pro";

Call format of custom function with function return value:

- Calling a custom function in the main module

Func Int AddTeach1(Int aa, Int bb)

Int cc = aa + bb;

Return cc;

EndFunc;

Call mode 1:

Int dd = AddTeach1(3,5);

Call mode 2:

Int ee;

ee = AddTeach1(4,6);

- Calling, by the main module, a custom function in other modules

Define this function in the "test.pro" module:

Func Int AddTeach2(Int aa, Int bb)

Int cc = aa + bb;

Return cc;

EndFunc;

Call mode 1:

Int dd = test.AddTeach2(3,5);

Call mode 2:

Int ee;

ee = test.AddTeach2(4,6);

5.2 Project Operations

5.2.1 Creating a Project

You can create a new project. The project can be on any disk drive of the PC. By default, the created project is stored in folder in the C drive, or you can save it to a subfolder.

A new project contains a "main.pro" program file. The contents of other files in the project directory are empty or default.

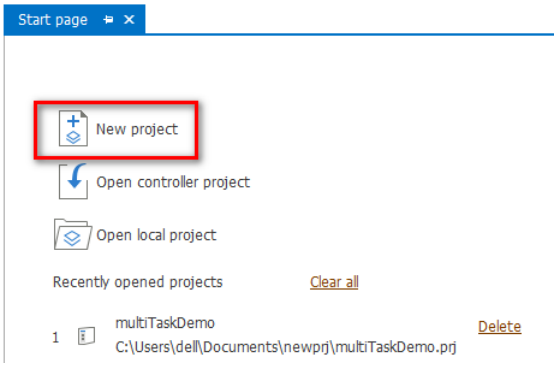


Ensure that InoRobotLab has control and is in manual mode.

You can create a new project in any of the following three methods.

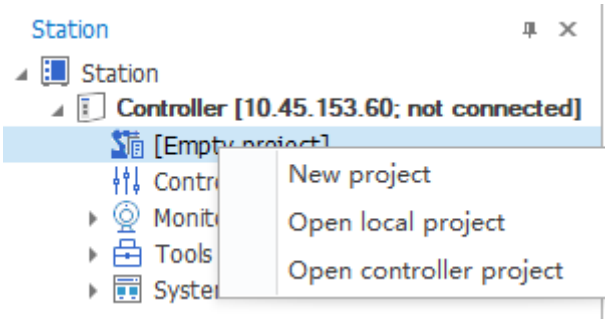
- Method 1: Creating a new project from the start page

Click the button in the upper left corner of the software to open the start page and then click the "New project" button.



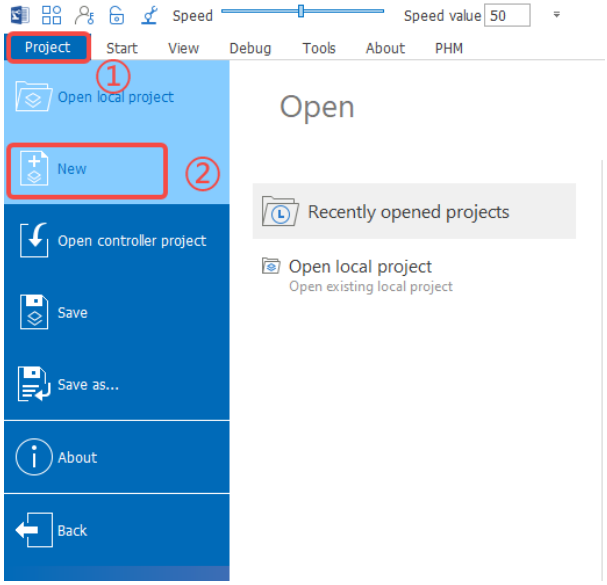
- Method 2: Creating a new project from the "Empty project" node

Right-click on the "Empty project" node and select "New project".

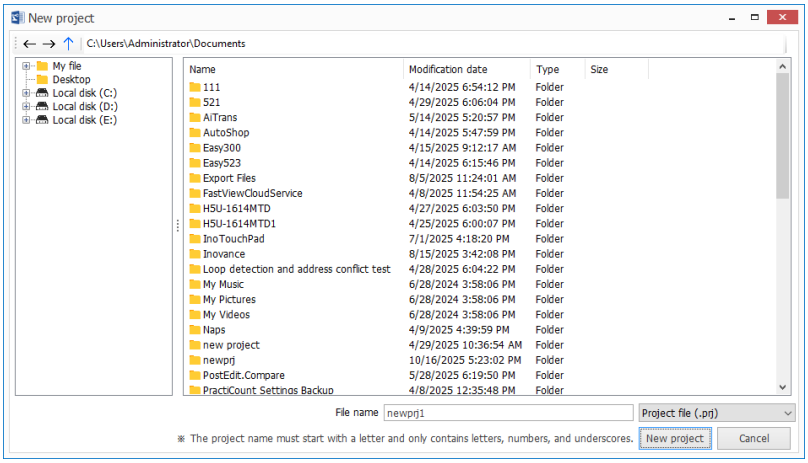


- Method 3: Creating a new project from "Project" > "New"

1. Click "Project" > "New" to enter the background page.



2. Click the "New project" button to create a new project, as shown in the figure below:



Item	Description
Disk selection	By default, the created project is saved in a folder in the C drive. Also, you can custom the save path as needed.
File name	• The project name must meet the naming convention, that is, it must start with a letter and contain only letters, numbers and underscores with a max length of 16 characters (excluding the extension ".prj"). • If you create a project with the same name in the same folder, the system will prompt that the project name already exists, and the new project creation will fail.
New project	Create a new project.
Cancel	Cancel creating a new project.

5.2.2 Opening a Local Project

You can open a local historical project.

CAUTION

Ensure that InoRobotLab has control and is in manual mode.

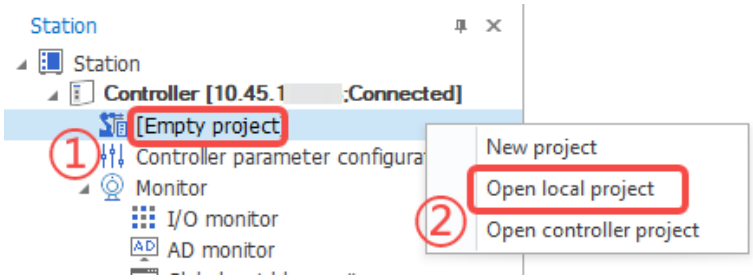
You can open a local project in any of the following three methods.

- Method 1: Opening a local project from the start page

1. Click the button in the upper left corner of the software to open the start page and then click "Open local project".
2. Select the project storage path, select the ".prj" file for the project, and click "Open local project".

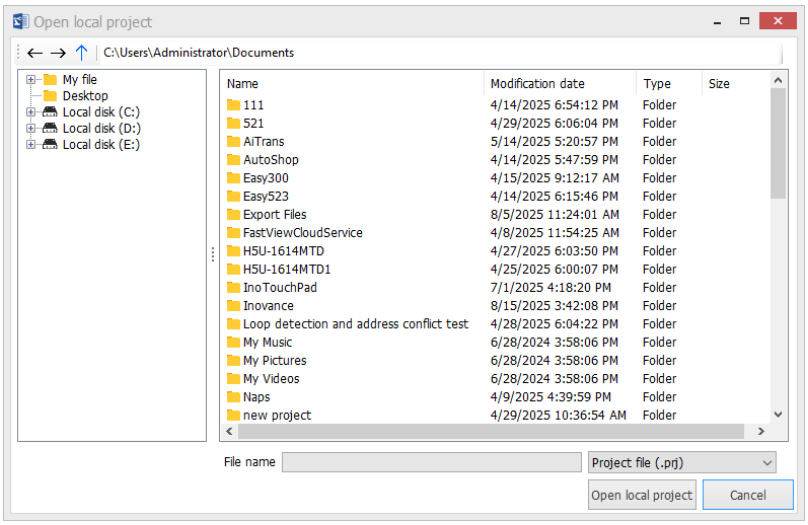
- Method 2: Opening a local project from the "Empty Project" node

1. Right-click the "Empty project" node and select "Open local project".
2. Select the project storage path, select the ".prj" file for the project, and click "Open local project".



- Method 3: Opening a local project from "Project" > "Open local Project"

1. Click "Project" > "Open local project", and select the project storage path.
2. Select the ".prj" file for the project, and click "Open local project".



Item	Description
Disk selection	Select the disk drive where the project to be opened is saved.
File name	To open a folder, double-click it.
Open local project	Open the selected project.
Cancel	Cancel opening project.

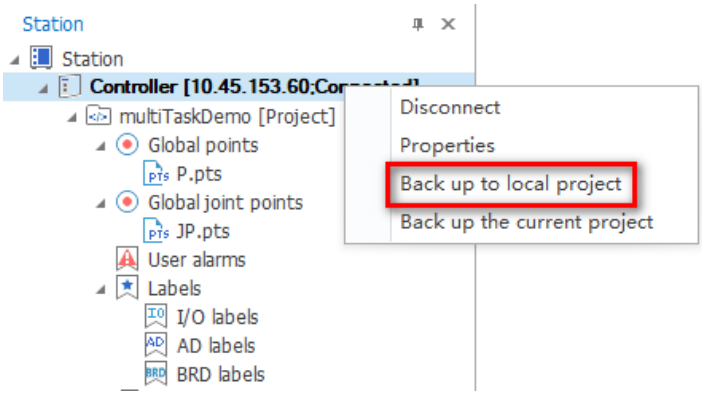
NOTE

When you open the project offline, the software switches to the robot model corresponding to "main.pro" file in the project.

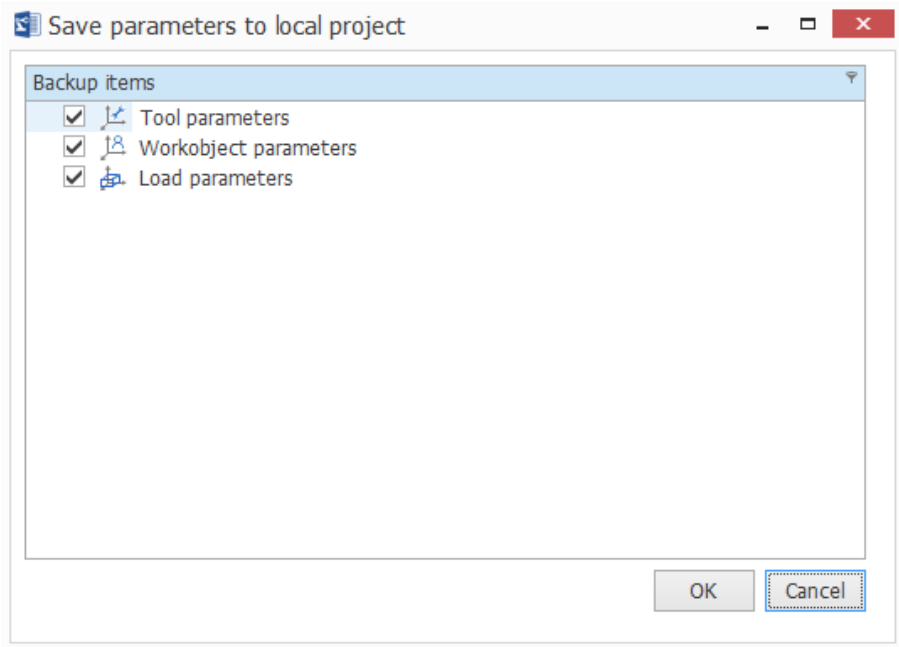
5.2.3 Backing Up to Local Project

You can back up the tool parameters, workobject parameters, and load parameters currently used by the controller to the local project currently opened.

1. Right-click on the controller node and select "Back up to local project".



2. In the "Save parameters to local project" dialog, check the items to be backed up, and then click "OK".

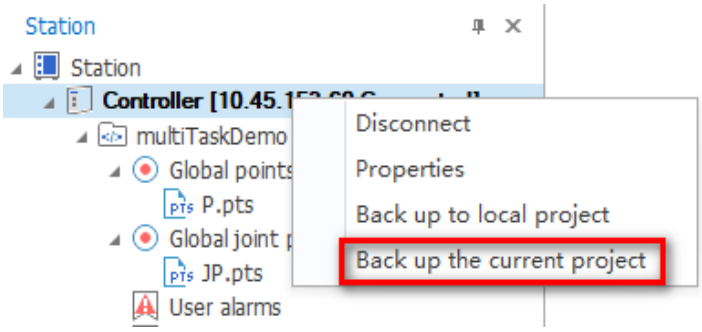


> **CAUTION**
>
> The backup operation will overwrite the parameter file in the local project that is currently open.

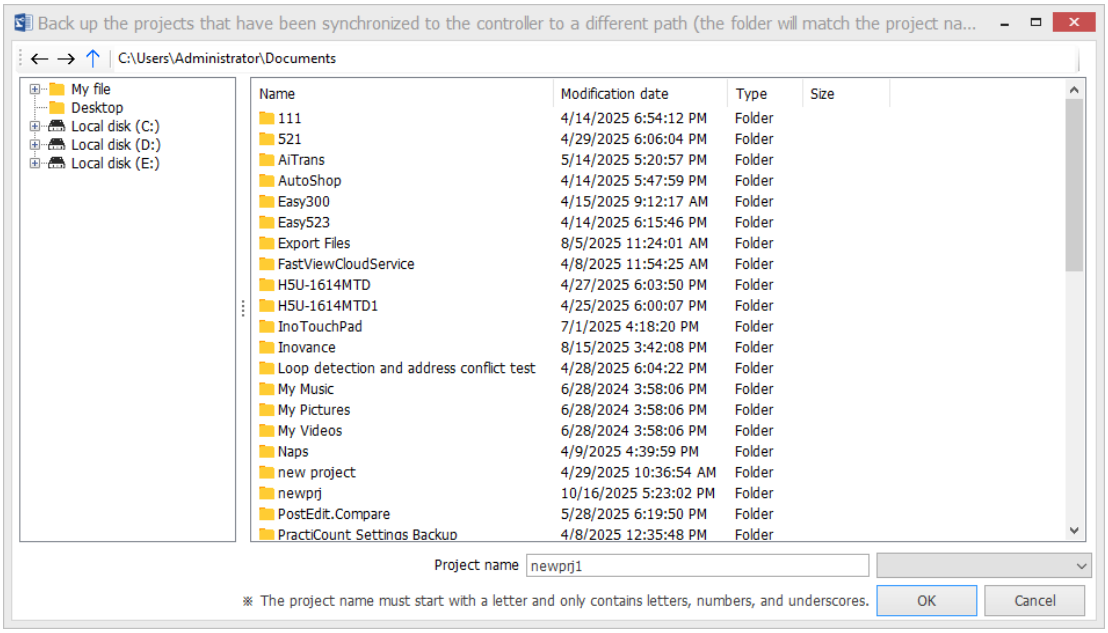
5.2.4 Backing Up the Current Project

You can back up the active project of the controller to a specified local directory. Different from the operation of 5.2.7 Saving the Project As, this backup operation also involves tool parameters, workobject parameters, and load parameters.

1. Right-click on the controller node and select "Back up the current project".



2. In the "Back up the current project" dialog, select the storage path, fill in the project name, and click "OK".



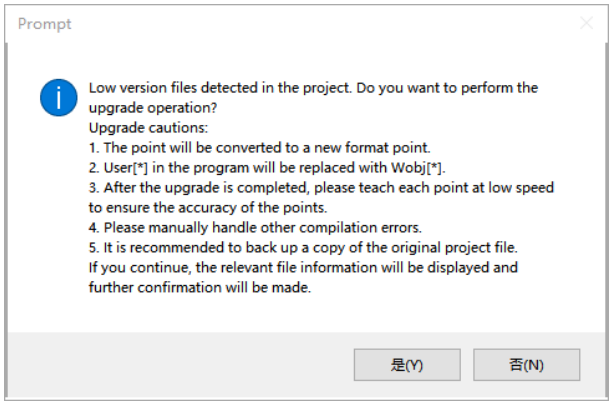
5.2.5 Upgrading the Project

If a file in the project does not match the current software version, a project upgrade prompt will be displayed.

The purpose of upgrading the project is to make the old version project comply with the current InoRobotLab software.

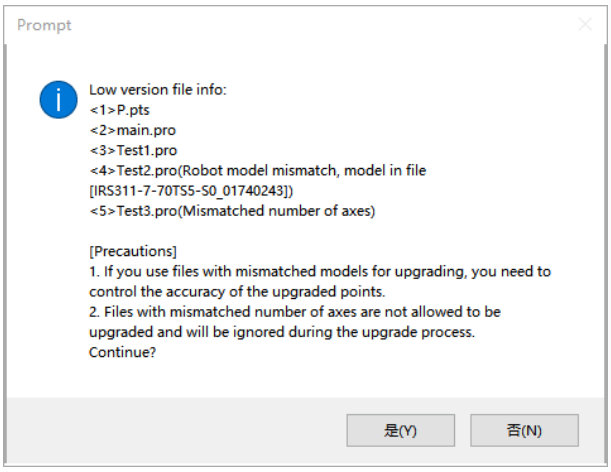
Online upgrade

1. when the controller is connected, if you open an old project or a project including an old version file, the following prompt appears. Click "Yes" to continue the upgrade operation, or "No" to exit the upgrade without making any changes to the project.

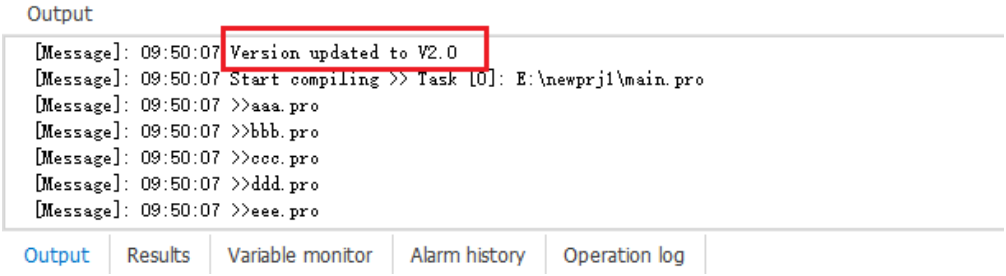


2. After you click "Yes", the following prompt appears. Then click "Yes" to continue the upgrade operation, or "No" to exit the upgrade without making any changes to the project.
   - If the robot model in the file is different from the robot model currently connected, a corresponding prompt will be displayed.
   - If the number of robot axes in the file is the same as that of the robot model currently connected, but the specific robot model is different, a prompt will be displayed. In this case, upgrade is allowed. However, the upgrade of points requires the use of the robot's structural parameters, etc. These parameters may vary from robot model to robot model, and the accuracy of points upgraded needs to be controlled by yourself.

- If the number of robot axes in the file differs from that of the robot model currently connected, upgrade is not allowed. In this case, the file will be ignored and not be operated. If you confirm that the file content is in line with the current robot, you can modify the robot model manually and load the file into the project.

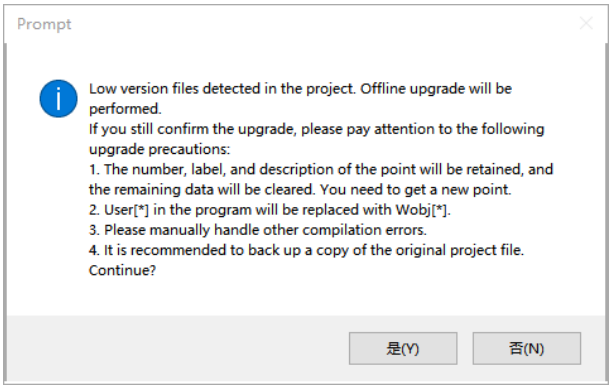


3. After the project upgrade is performed, the "Output" bar prints the version update success information, as shown below.



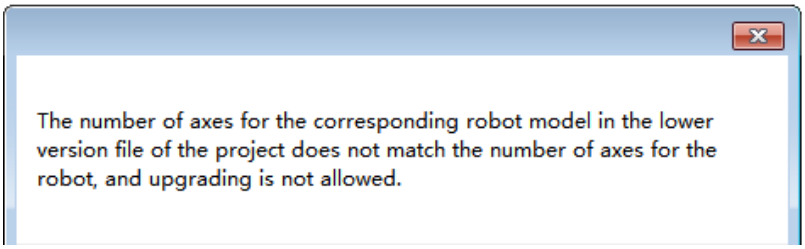
Offline upgrade

When the controller is not connected, if you open an old project or a project including an old version file, offline upgrade will be performed. Unlike the online upgrade, the position information of various points will be cleared, only labels and descriptive information are retained, as shown below.



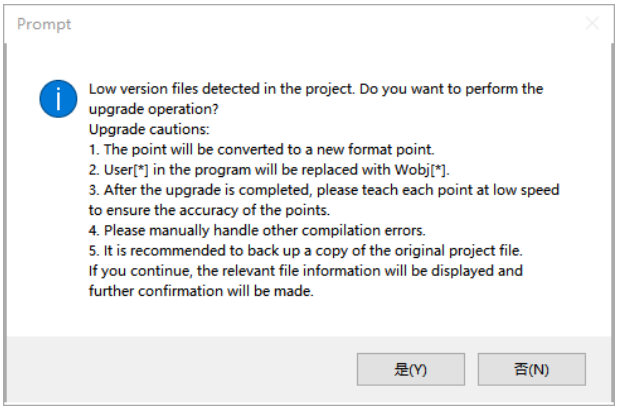
NOTE

- The project upgrade currently supports only upgrade of S03.20 and above projects and related files to the current version.
- The project upgrade objects include the "P.pts" global point files and ".pro" program files that have been added to the project in the "*.pri" project file.
- A compilation error may occur after the upgrade, and you need to modify it manually.
- When the robot model of each file in the opened project is different from the robot model currently connected in the number of robot axes, the upgrade operation is not allowed, as shown below.



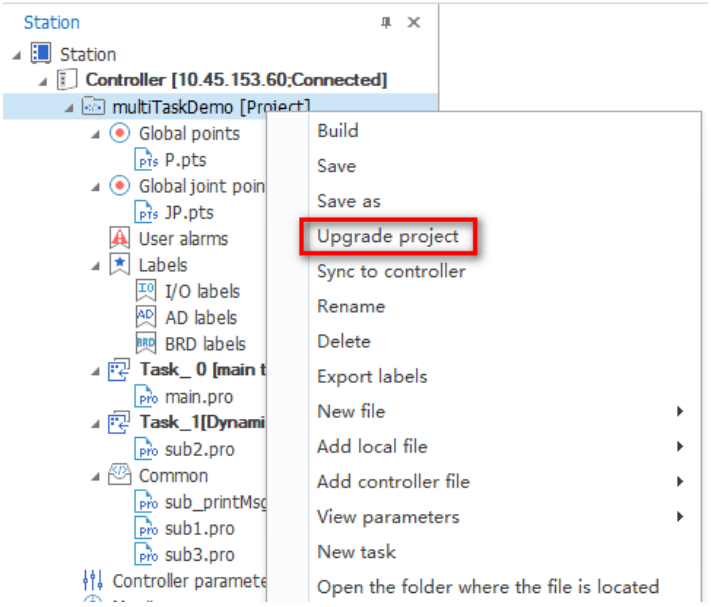
Manual upgrade

If you click "No" to exit the upgrade, you can manually trigger the project upgrade operation later.



Steps

1. Right-click on the project node and select "Upgrade project".

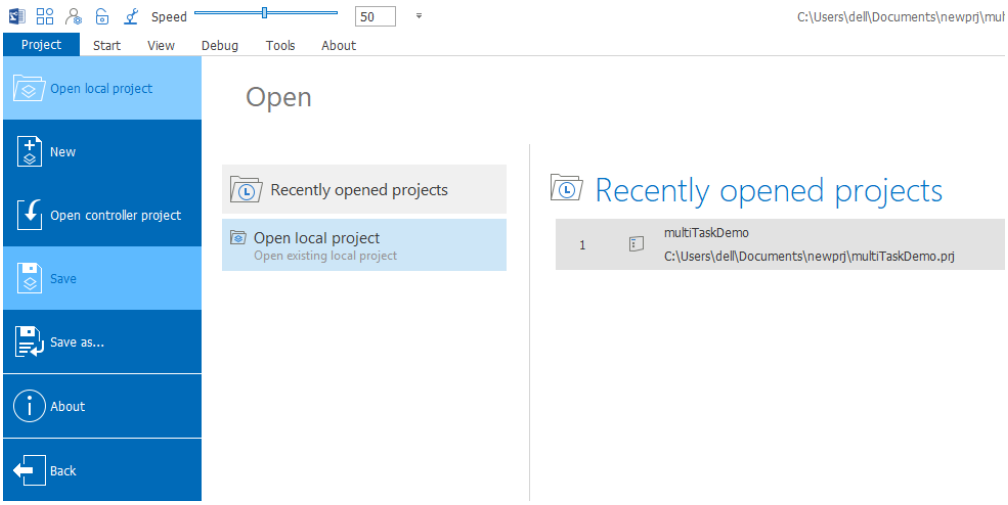


2. Complete the upgrade following the on-screen instructions.

5.2.6 Saving the Project

You can save the current project to the corresponding path. You can save the project in any of the following three methods.

- Method 1: Right-click on the project node and select "Save".
- Method 2: In the "Start" menu, click  to save the current project file, or  to save all project files.
- Method 3: In the "Project" menu, select "Save".



Results

When you open the program file or the point data page, you will see a "*" sign as the modification flag. The "*" sign will disappear after the saving is completed.

> **CAUTION**
>
> The above is the manually triggered project save. The project save can also be automatically triggered upon compiling the project, syncing the project to the controller, saving the project as, closing the project, and closing the software. For related content, see 5.2.9 Compiling the Project, 5.2.10 Synchronizing to the Controller, 5.2.7 Saving the Project As and 5.2.6 Saving the Project sections.

5.2.7 Saving the Project As

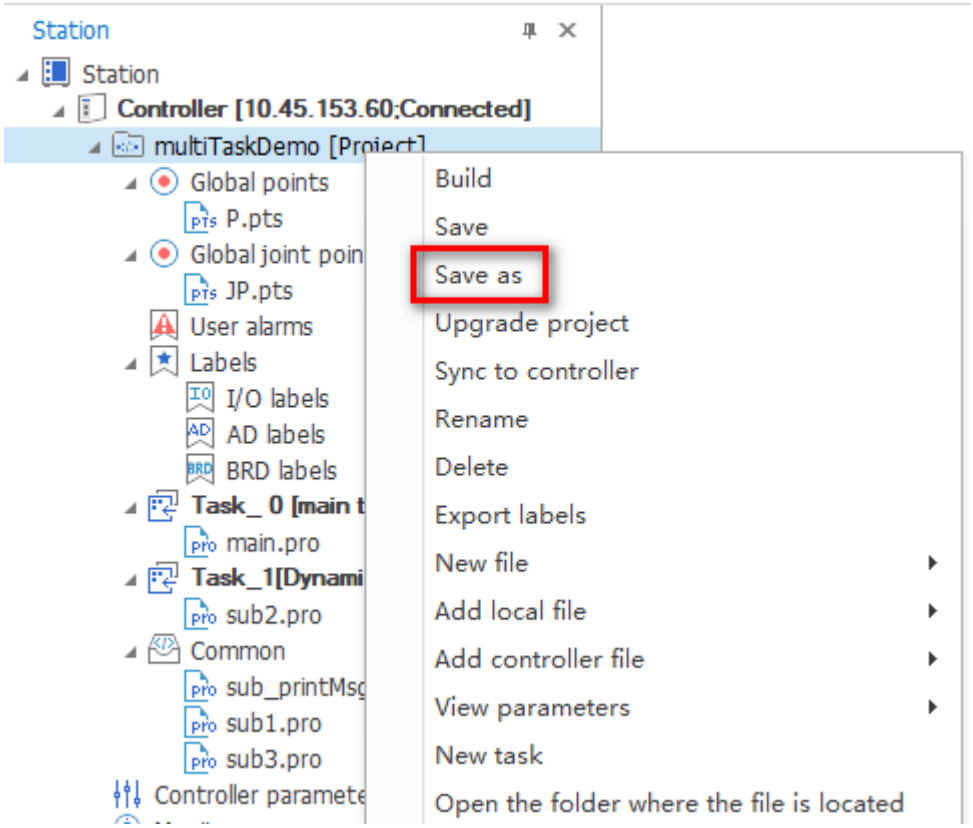
You can save a project to a different path. The project in the original path remains.

> **CAUTION**
>
> The project node must not be empty, that is, the target project has been opened by InoRobotLab software.

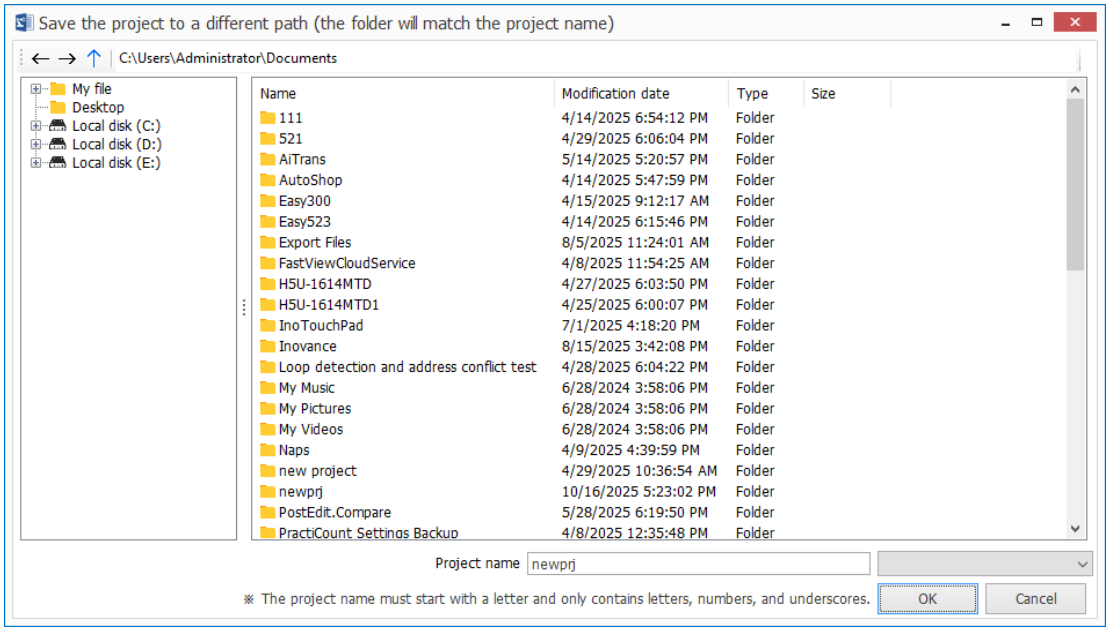
You can save the project as in either of the following two methods.

- Method 1: Saving the project from project node

1. Right-click on the project node and select "Save as".



2. In the pop-up dialog, select the storage path, fill in the project name, and click "OK".



- Method 2: Saving the project from "Project" menu

  Go to "Project" menu and select "Save as". Select the storage path, enter the file name and click "OK".

5.2.8 Opening the Controller Project

You can select a project from an existing project of the controller, save it to the local path where the InoRobotLab software is located, and open it automatically through the InoRobotLab software.

You can open the controller project in any of the following three methods.

- Method 1: Opening the project from the start page

  1. Open the start page and then click "Open controller project".
  2. In the "Select the controller project to open" dialog, select the project to open. If you want to change the default storage path, click "Browse". Then click "OK" to open the project.

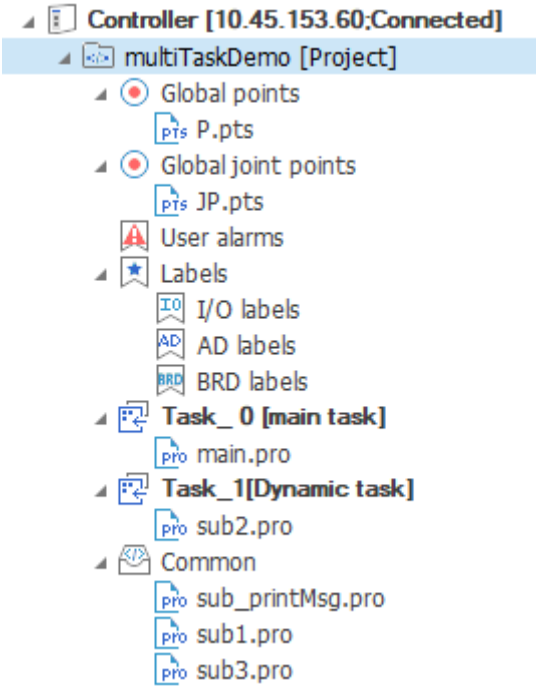
- Mode 2: Opening the project from the project node

  1. Right-click on the "Empty project" node and select "Open controller project".
  2. See Step 2 of Method 1.

- Mode 3: Opening the project from the project menu

  1. Go to "Project" > "Open controller project".
  2. See Step 2 of Method 1.

When the operation is completed, the opened project appears in the project node.



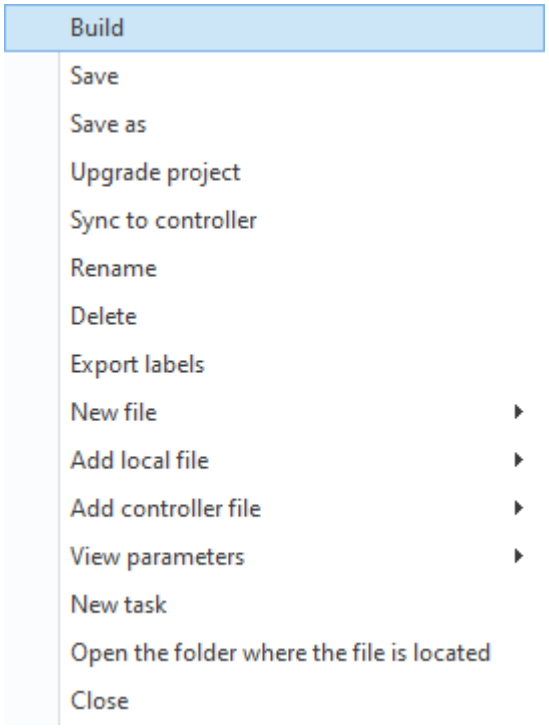
> **NOTE**
>
> If you need to transfer the project currently used by the controller to the PC and open it locally, it is recommended to use the "Sync from controller" option.

5.2.9 Compiling the Project

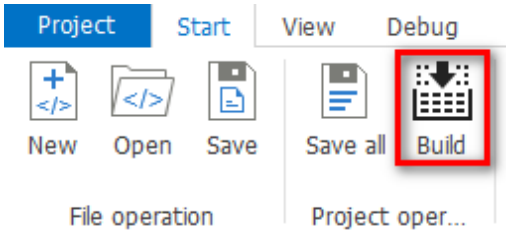
Project compilation can check syntax errors in program files, and update point variables and labels to the interpreter.

You can compile the project in either of the following three methods.

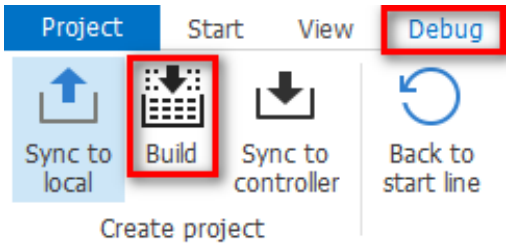
- Method 1: Right-click on the project node and select "Build".



- Method 2: Go to "Start" menu and select "Build".

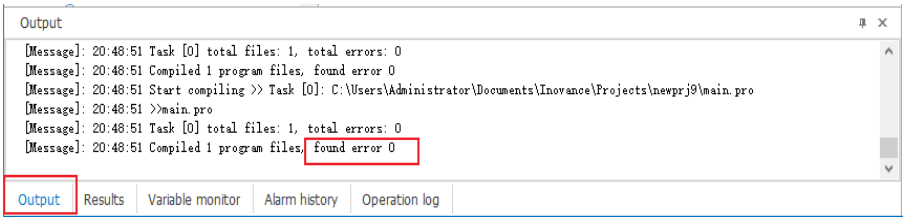


- Method 3: Go to "Debug" menu and select "Build".



Results

- There are no compilation errors.



- There are compilation errors.



NOTE

- Project-related files will be saved before project compiling.
- To compile a single program file of a large project, see 5.3.3 Compiling.
- If a robot model mismatch is reported during compiling, see the following two methods for checking:
  - Check if the program contents and point data are saved. If not, click "Save" to eliminate the error.
  - Check if the number of axes of the model matches the project file.
- If the project is confirmed to be error-free but compilation errors persist, it may be due to unresolvable characters introduced by external operations or changes to the encoding format of the project file. In this case, simply rename the project and save it again using the InoRobotLab software.

5.2.10 Synchronizing to the Controller

You can download the currently open project to the controller and set the project as the project currently used by the controller.

After synchronization to the controller is completed, the project opened by InoRobotLab software is consistent with the project currently active on the controller. In this case, you can start debugging.

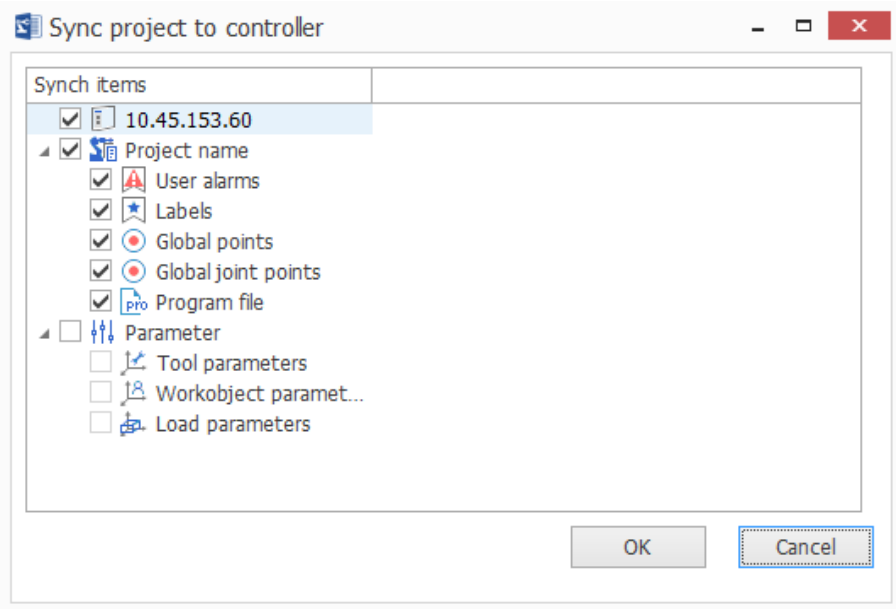
Prerequisite

- Ensure that the current state is not the debugging state, meaning the status bar in the lower-left corner does not display: "Debug: ***".
- Ensure that the project has been successfully compiled locally.

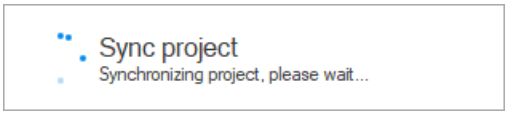
You can synchronize the currently open project to the controller in either of the following two methods.

- Method 1:

1. Right-click on the project node and select "Sync to controller".
2. In the "Sync to controller" dialog, check the items that need to be synchronized, and then click "OK".

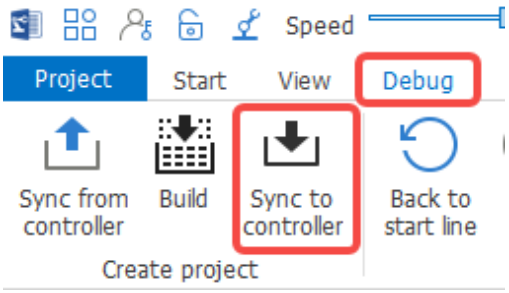


3. Wait until the synchronization is complete.



- Method 2:

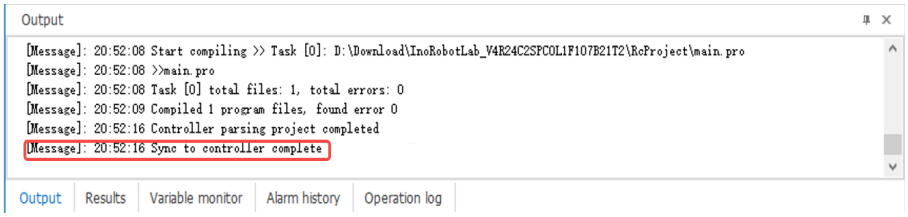
Go to "Debug" > "Sync to controller".



Repeat steps 2 and 3 in Method 1.

Results

When synchronization is complete, the information view displays: Controller parsing completed, as shown in the figure below. If a controller compilation error occurs, check the project for related problems.



NOTE

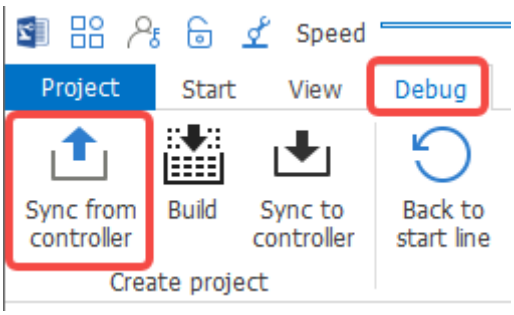
- When the same project exists in the controller, the original project will be overwritten.
- After synchronization is completed, all tasks are returned to the start line. The static task starts running from the start line.
- If you uncheck the default checked sync items, the synchronization is incomplete. When you perform some operations (for example, return to the start line, set the start line, start, and step-into debugging, etc.) later, you will be prompted that synchronization is required.
- If you only check one or more of tool parameters, user parameters, load parameters, and user alarms, synchronization to the controller will not trigger the decoding of the controller.
- Only when there are tool parameter, user parameter, and load parameter files in the project, the corresponding synchronization items will be selectable.
- When one or more of the sync items including labels, global points, global joint points, and program file are checked, synchronization to the controller triggers the decoding of the controller.
- If the current project name of the controller is inconsistent with the local project name, all items are checked by default and cannot be changed.
- If the robot model mismatch is reported during synchronization, check the program contents and point data.
- During synchronization, if a point file number configuration file (RobPointMappingFile.dat) exists in the project, the configuration will be checked. If the point file in the configuration does not exist in the project, you will be warned in the "Output" bar.

5.2.11 Synchronizing the Project to Local

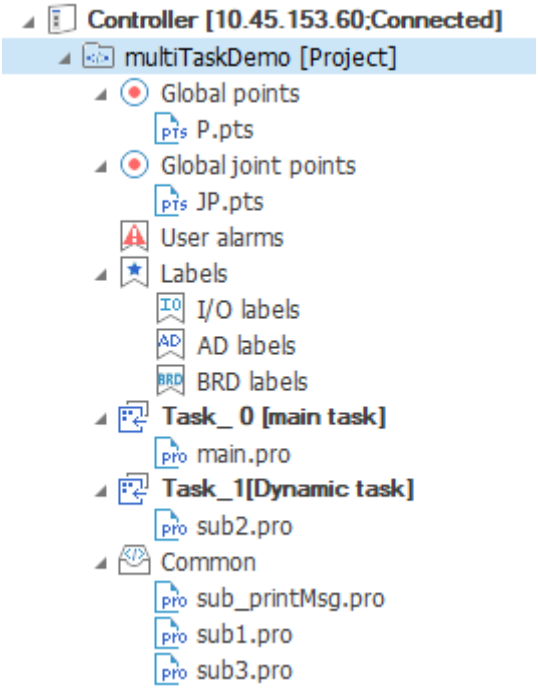
You can synchronize the current active project of the controller to the PC where the InoRobotLab software is installed.

For example, if you need to debug the current robot program or observe the running line information when the robot is running automatically, you need to synchronize the project to local.

Go to "Debug" > "Sync from controller".



When synchronization is complete, the active controller project appears in the project node.



NOTE

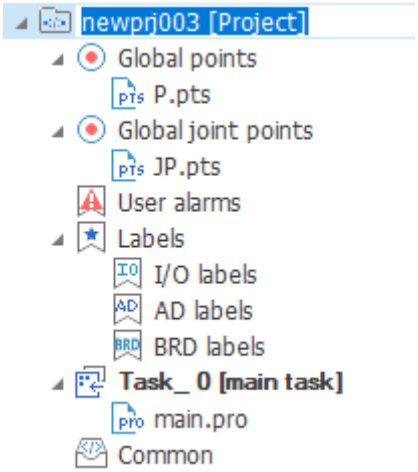
- If there is an open project before the project is synchronized locally, a pop-up box will prompt you to save it.
- The synchronized project will replace the original project in the default path.
- The key difference between the "Sync from controller" function and the "Open controller project" function is that the project taken from the controller is the current active project and does not require user selection.

5.2.12 Renaming the Project

You can rename the current project.

Steps

1. Right-click on the project node and select "Rename".
2. Enter the new project name, for example, "newprj003".



NOTE

- In the controller, the folder name of the saved project is the same as the project name. Therefore, if a project, which already exists in the controller before renaming, is renamed and then synchronized to the controller, one more renamed project will be added to the controller.
- The project name must comply with the project naming rules. (The project name starts with a letter and can contain only letters, numbers, and underscores, with a max length of 16 characters.)

5.2.13 Deleting the Project

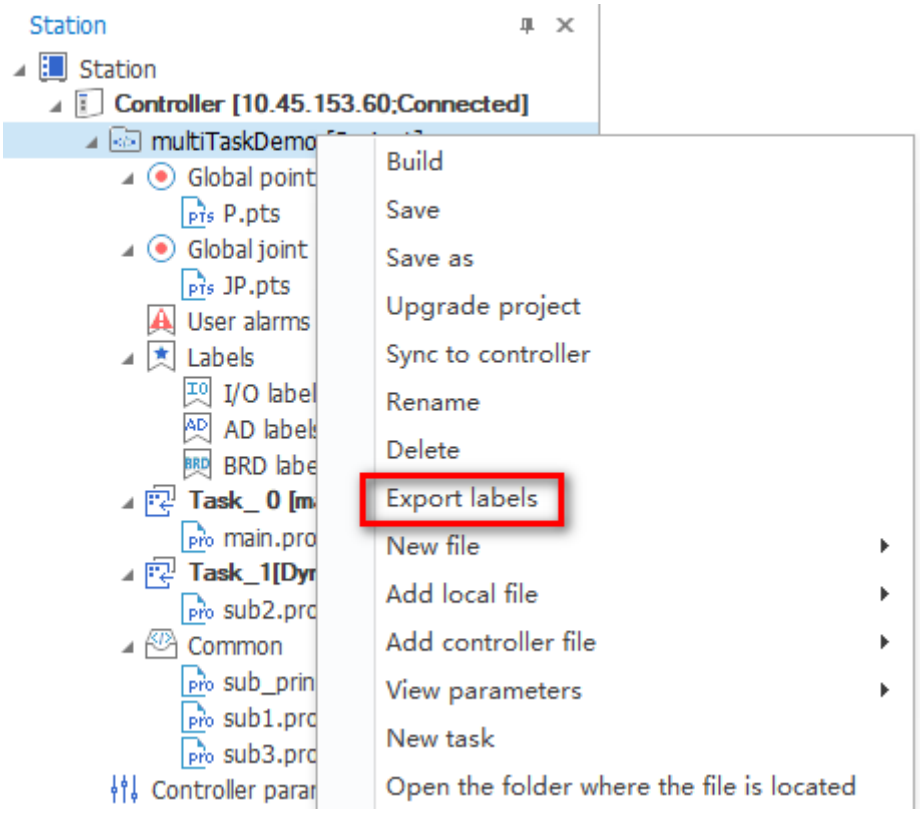
- You can delete the currently open project from the local disk. The project becomes empty.
- Right-click on the project node and select "Delete". Confirm delete in the pop-up dialog.

5.2.14 Exporting the Labels

You can export the global-type label information (global position labels, I/O labels, BRD labels, and AD labels) in the project into an excel file for easy view.

Steps

Right-click on the project node and select "Export labels".



Results

The path of the exported labels is the folder where the project is located. The file name is "Project name_Labels.xlsx".

5.2.15 Creating a File

5.2.15.1 Creating a Program File

A project can contain multiple program files. The default program file created by the project is "main.pro". You can create other program files or add created program files to the project.

Prerequisite

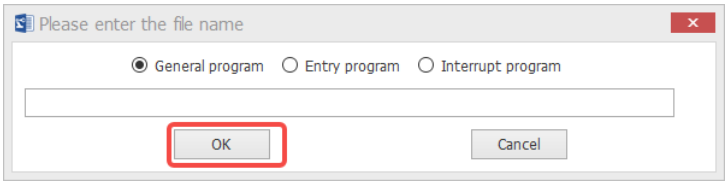
Make sure that the project is open and editable.

Steps

You can create a new program file in any of the following four methods.

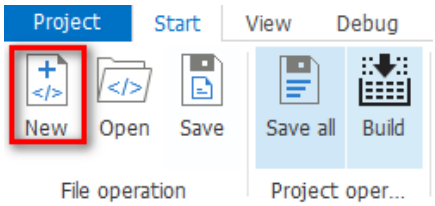
- Method 1:

1. Right-click on the project node and select "New file" > "New program file".
2. Enter the name of the program file in the dialog box, and then click "OK".



- Method 2:

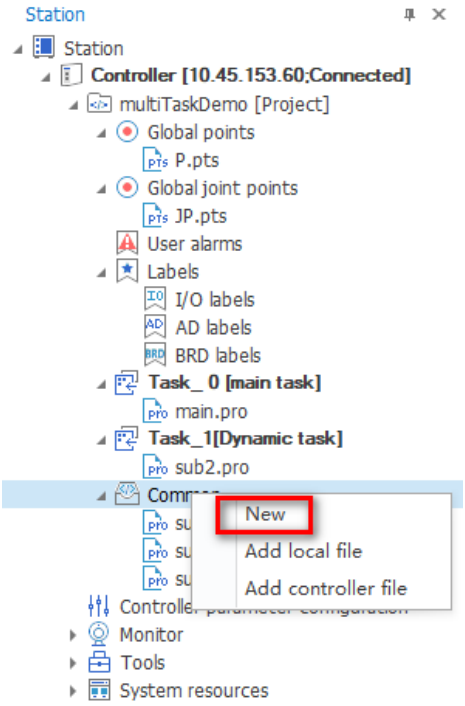
1. In the "Start" menu, click the "New" button.



2. Enter the name of the program file in the dialog box, and then click "OK".

- Method 3:

1. Right-click on the "Common" node of the project and select "New".



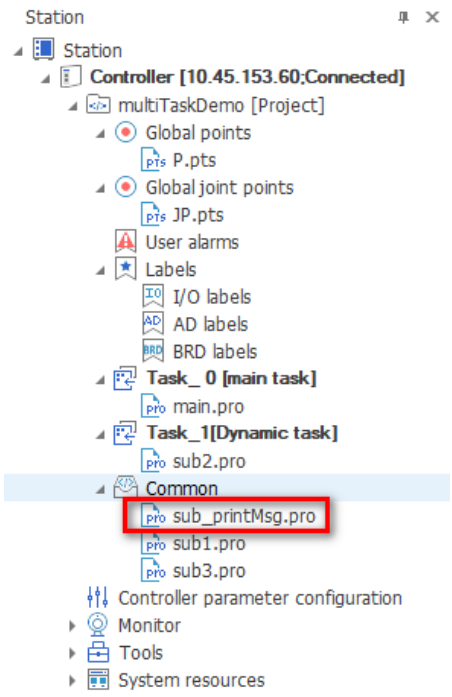
2. Enter the name of the program file in the dialog box, and then click "OK".

- Method 4:

1. Press CTRL+N combination at the same time.
2. Enter the name of the program file in the dialog box, and then click "OK".

Results

The newly created program file appears under the project node and in the project directory.



NOTE

- One project contains up to 24 program files. The number of program files can also be expanded to less than 256. It needs to be saved as an extended setting in "Controller parameter configuration" > "System settings" > "Program quantity settings". Additionally, when using the extended setting, add program files with caution. An excessive number of program files or an overly large project may cause issues during loading, compiling, or synchronization.
- The program file name starts with a letter and contains only letters, numbers and underscores, with a max length of 26 characters excluding the extension.
- The program file name cannot contain keyword, global label name, or a name defined by the operating system.
- The operation of creating a program file is to add a program file on the "Common" node without the "Start…End" entry function. If you configured the created program file as a task, manually add the "Start…End" section.
- Also, you can create a program file when creating a new task. For details, see the 5.2.16 Creating a Task section.

5.2.15.2 Creating a Point File

One project can include multiple point files. The default file created by the project is "P.pts". You can create other point files or add created point files to the project.

Prerequisite

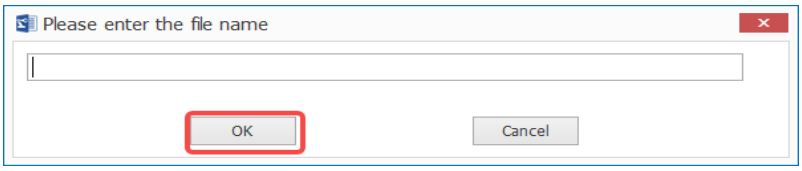
Make sure that the project is open and editable.

Steps

You can create a new point file in either of the following two methods.

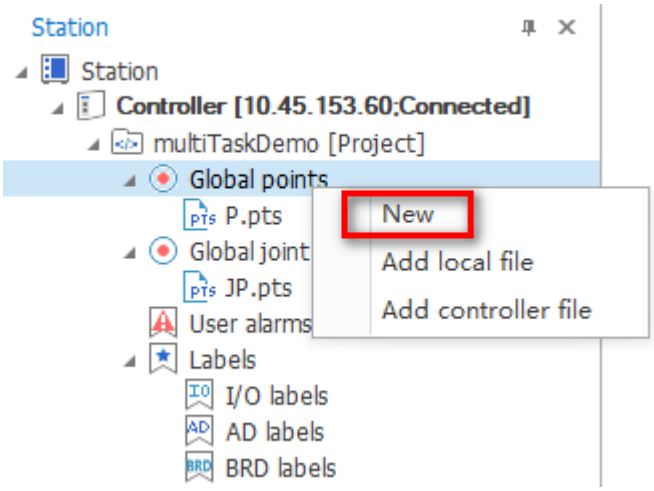
- Method 1:

1. Right-click on the project node and select "New file" > "New point file".
2. Enter the name of the point file in the dialog box, and then click "OK".



- Method 2:

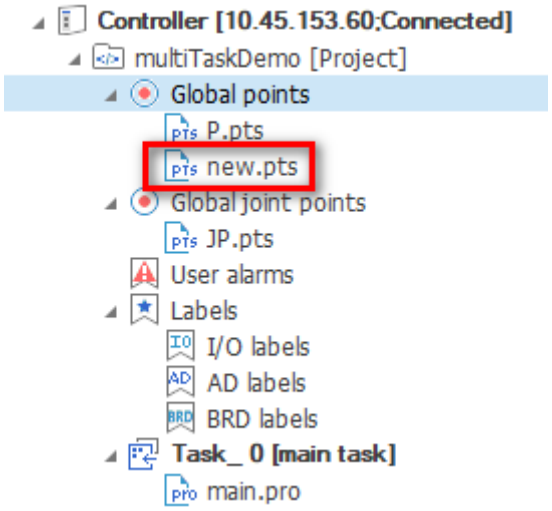
1. Right-click on the "Global points" node of the project and select "New".



2. Enter the name of the point file in the dialog box, and then click "OK".

Results

The newly created point file appears under the "Global points" node and in the "Data" folder of the project directory.



NOTE

- One project contains up to 128 point files.
- The program file name starts with a letter and contains only letters, numbers and underscores, with a max length of 26 characters excluding the extension.
- The program file name cannot contain keyword, global label name, or a name defined by the operating system.

5.2.16 Creating a Task

Purpose

You can create a new dynamic or static task. After a project is created, the main task is automatically created by default.

Steps

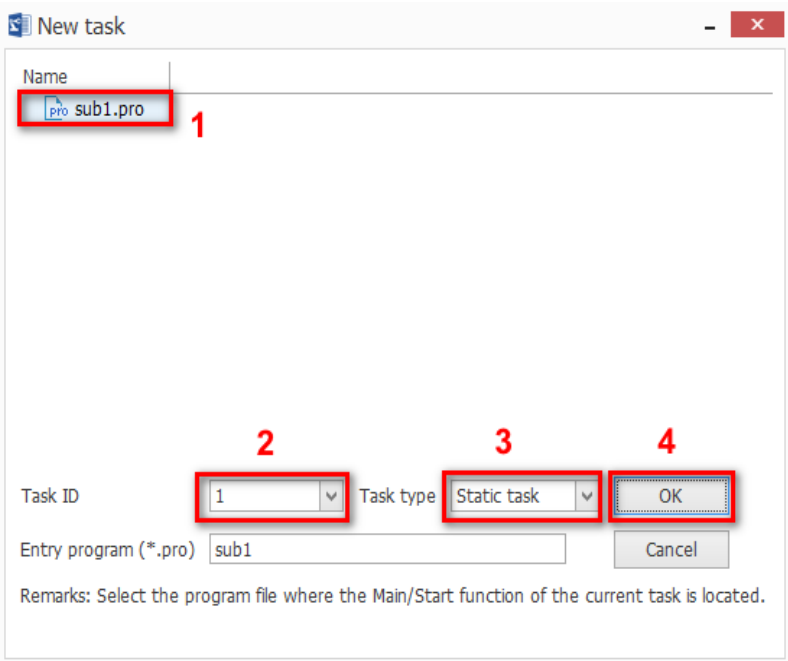
You can create a new task in either of the following two methods.

- Method 1:

You can create a new task using a program file under the "Common" node as the entry program. Right-click on the project node and select "New task".

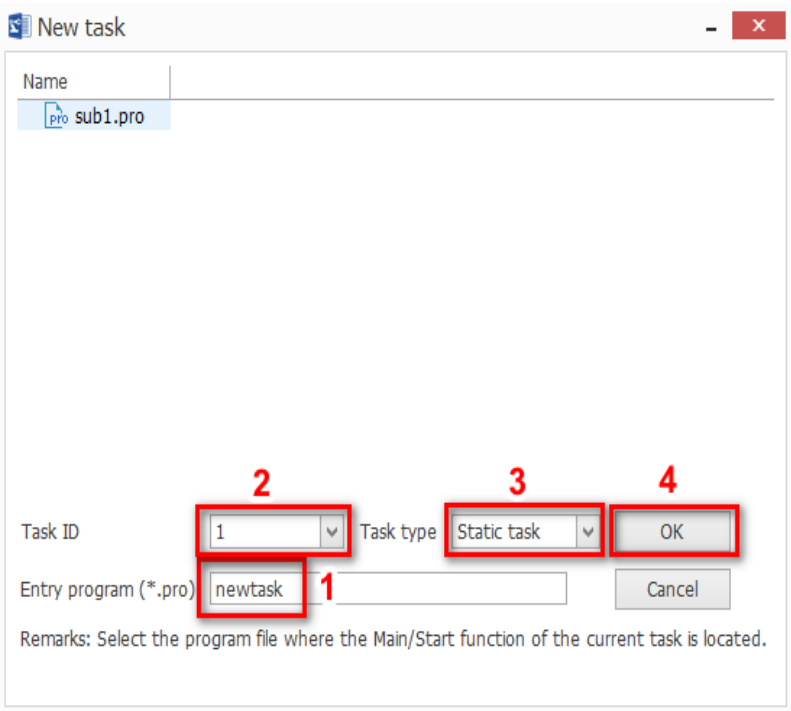
In the "New task" dialog, do as follows:

1. Select the entry program file.
2. Select the task ID.
3. Select the "task type" as static task or dynamic task.
4. Click "OK".



- Method 2:

You can create a new task by creating a new program file and using the created program file as the entry program for the task. Right-click on the project node and select "New task".

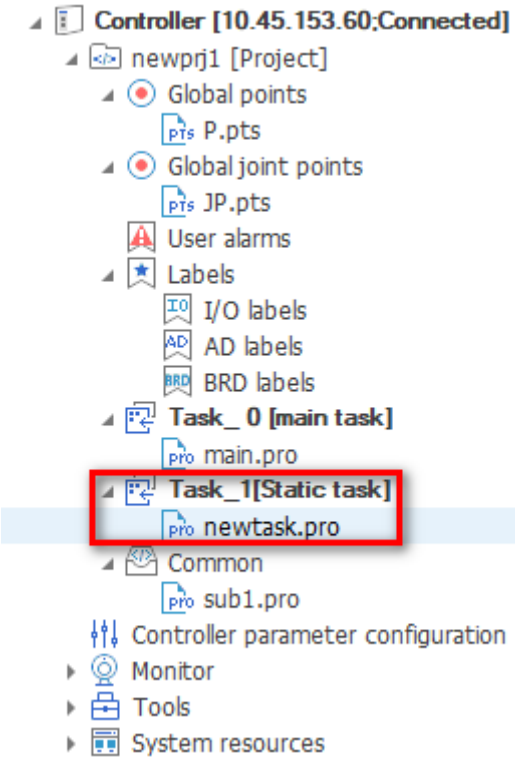


In the "New task" dialog, do as follows:

1. Enter the name of the program file to be created.
2. Select the task ID.
3. Select the "task type" as static task or dynamic task.
4. Click "OK".

Results

The created task appears as a task node, as shown below.



NOTE

- After you click "OK" when creating the task, if there is no entry function in the selected program file as shown below:
  Start;

  ……

  End;

  The compiling operation will result in an error. You need to enter the entry function manually.
- If there are already 14 tasks other than the main task, you cannot create a new task any more.
- The same operation result can also be achieved with the "Set as task" option. For details, see the 5.3.9 Setting the Program File as Task section.

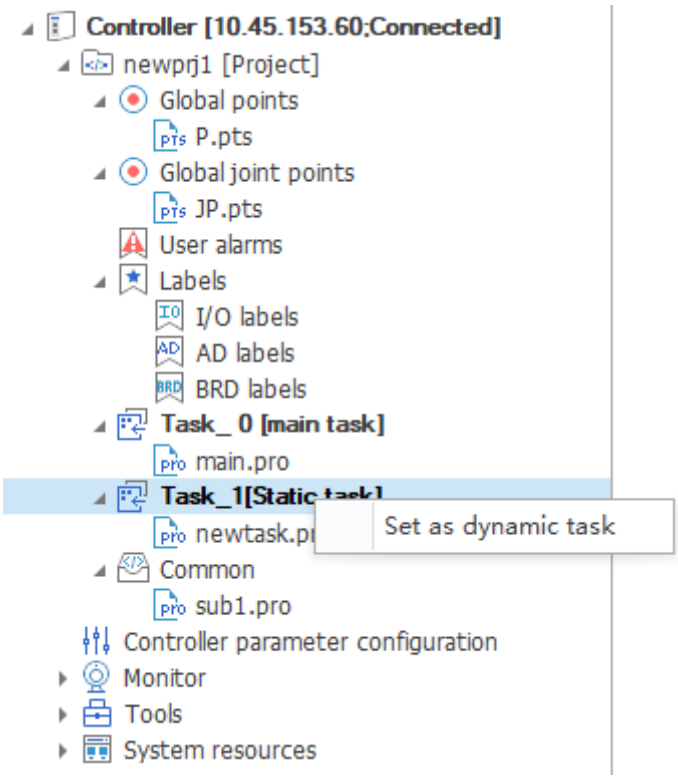
5.2.17 Setting the Task Type

Purpose

You can modify the type of multi-tasking. The types of multi-tasking include static and dynamic tasks.

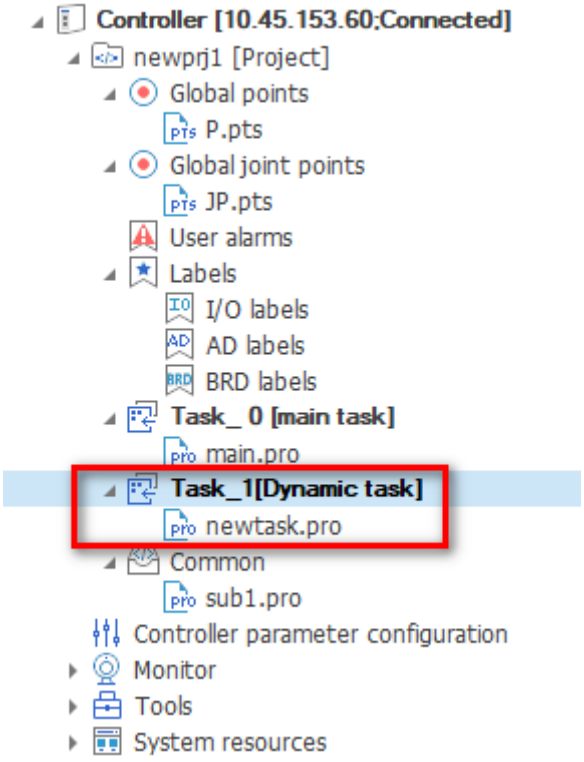
Steps

Right-click on the corresponding task node and modify the task type.



Results

The task type is modified.



NOTE

Dynamic tasks are started and stopped together with the main task. The static task is started automatically when the controller is powered on, or automatically after it is synchronized to the controller. For details, see the 7.3.1 Task Management section.

5.2.18 Adding a Local File to the Project

5.2.18.1 Adding a Local Program File to the Project

You can add the local program files to the project for reuse.

Prerequisite

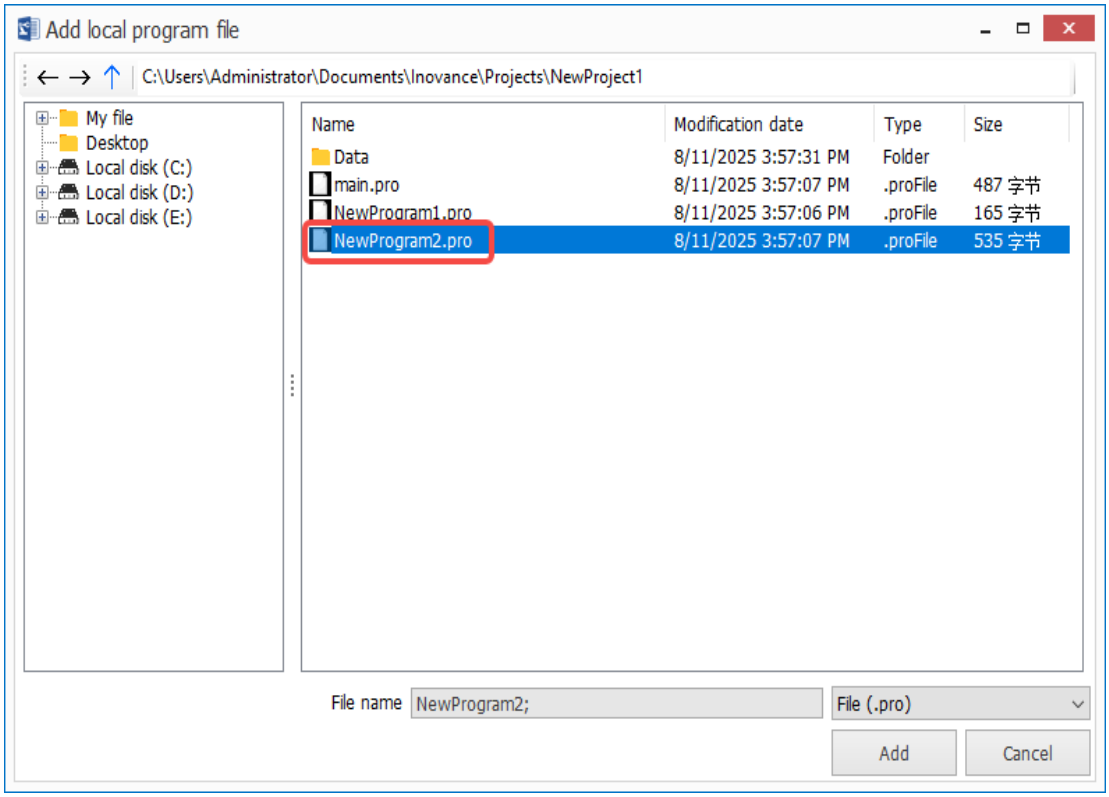
The project has been opened and the program file that needs to be added to the project exists.

Steps

You can add the local files to the project in either of the following two methods.

- Method 1:

1. Right-click on the project node and select "Add local file" > "Add local program file".
2. In the "Add local program file" dialog, select (Ctrl+left mouse button for multiple selection) the local program file(s) and click "Add".

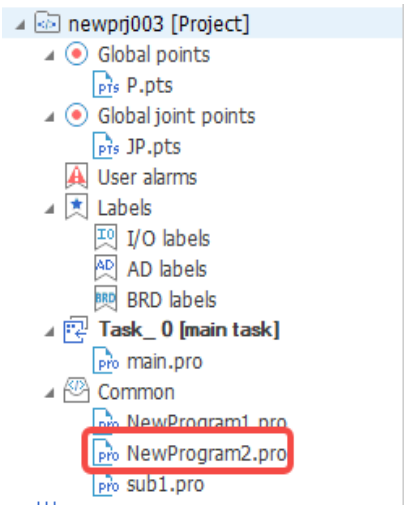


- Method 2:

1. Right-click on "Common" node and select "Add local file".
2. See Step 2 of Method 1.

Results

After the operation is completed, the selected program file is added to the project node and to the project directory.



NOTE

- The added program file name starts with a letter and contains only letters, numbers and underscores, with a max length of 26 characters excluding the extension.
- A similar operation to this one is adding a controller file, the latter being selecting a program file from the controller and adding it to the project. For details, see the 5.2.19 Adding a Controller File to the Project section.

5.2.18.2 Adding a Local Point File to the Project

You can add the local point files to the project for reuse.

Prerequisite

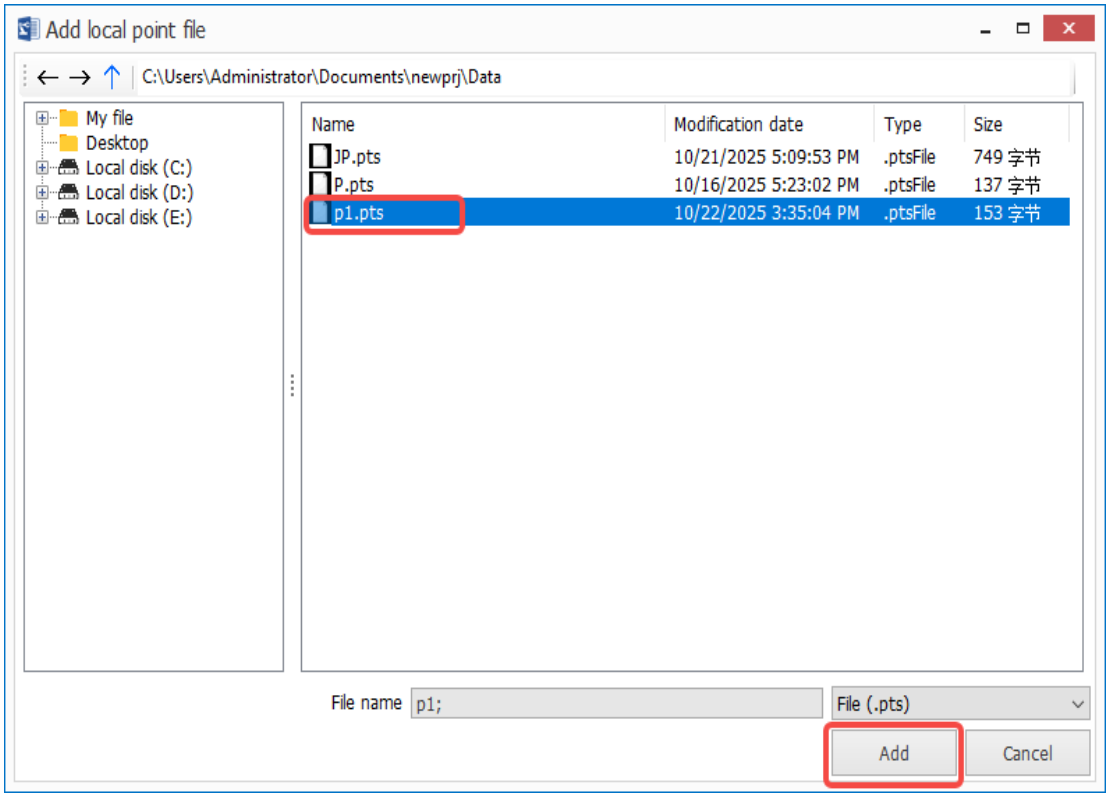
The project has been opened and the point file that needs to be added to the project exists.

Steps

You can add the local files to the project in either of the following two methods.

- Method 1:

1. Right-click on the project node and select "Add local file" > "Add local point file".
2. In the "Add local point file" dialog, select (multi-select) the local point file(s) and click "Add".

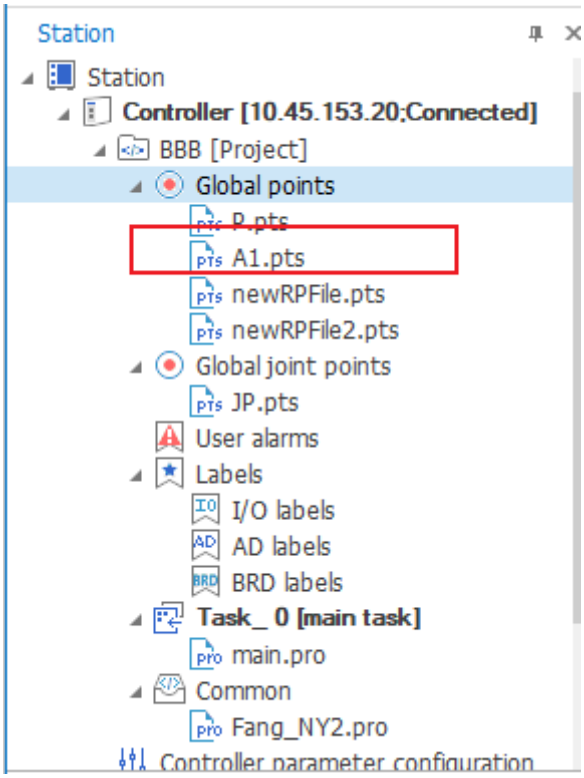


- Method 2:

1. Right-click on the "Global points" node and select "Add local file".
2. See Step 2 of Method 1.

Results

After the operation is completed, the selected point file is added to the project node and to the "Data" folder of the project directory.



NOTE

- The added point file name starts with a letter and contains only letters, numbers and underscores, with a max length of 26 characters excluding the extension.
- A similar operation to this one is adding a controller file, the latter being selecting a point file from the controller and adding it to the project. For details, see the 5.2.19 Adding a Controller File to the Project section.

5.2.19 Adding a Controller File to the Project

5.2.19.1 Adding a Controller Program File to the Project

You can select a program file from the controller and add it to the current project to facilitate reuse of the program file.

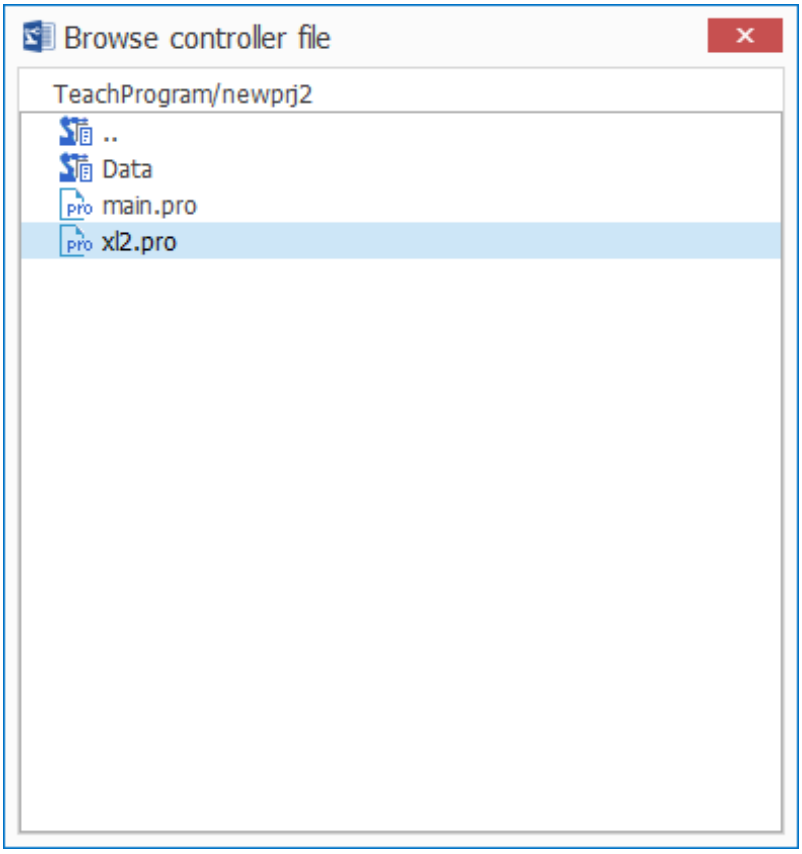
Steps

You can add the controller program file to the project in either of the following two methods.

- Method 1:

1. Right-click on the project node and select "Add controller file" > "Add controller program file".

2. In the "Browse controller file" dialog, double-click on the file you want to add.

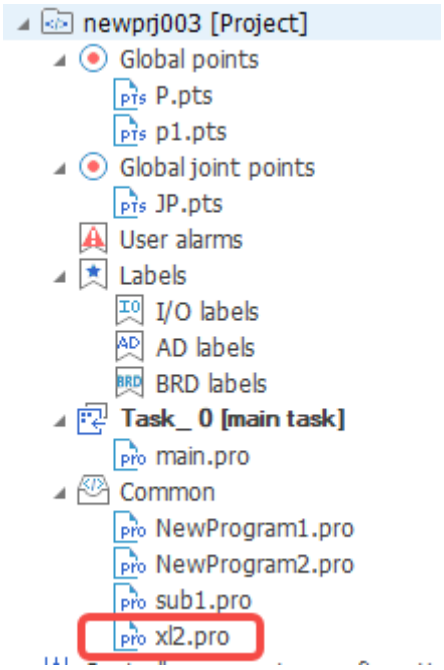


- Method 2:

1. Right-click on "Common" node and select "Add controller file".
2. See Step 2 of Method 1.

Results

When the operation is completed, the corresponding program file in the controller is copied to the project directory and a corresponding node is added to the project.



NOTE

- The added program file name starts with a letter and contains only letters, numbers and underscores, with a max length of 26 characters excluding the extension.
- A similar operation to this one is adding a local file, the latter being selecting a local program file and adding it to the project. For details, see the 5.2.18 Adding a Local File to the Project section.

5.2.19.2 Adding a Controller Point File to the Project

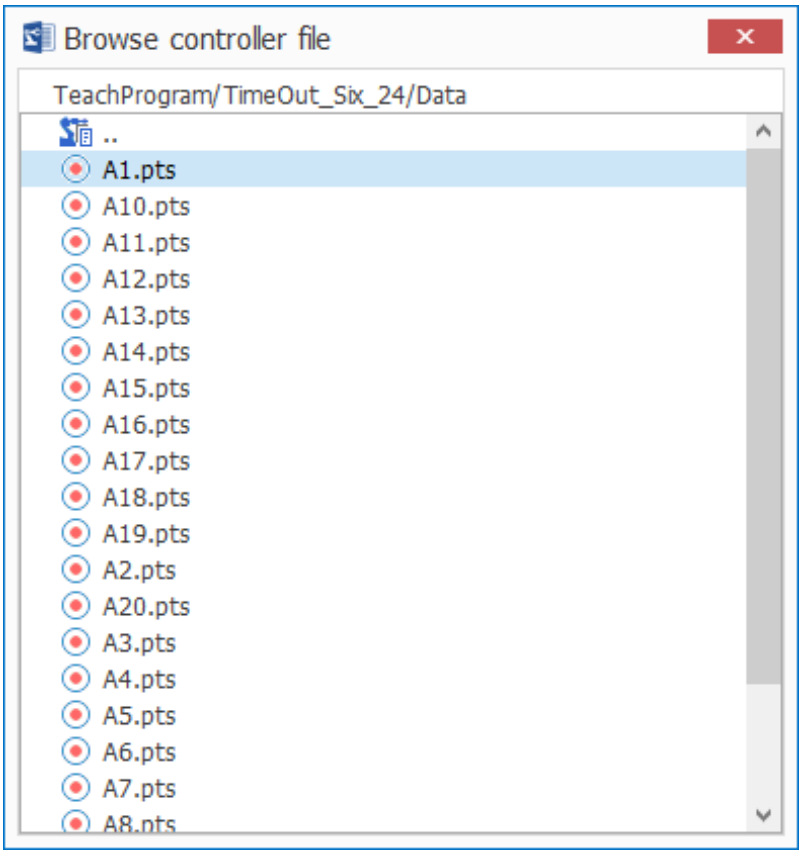
You can select a point file from the controller and add it to the current project to facilitate reuse of the point file.

Steps

You can add the controller point file to the project in either of the following two methods.

- Method 1:

1. Right-click on the project node and select "Add controller file" > "Add controller point file".
2. In the "Browse controller file" dialog, double-click on the file you want to add.

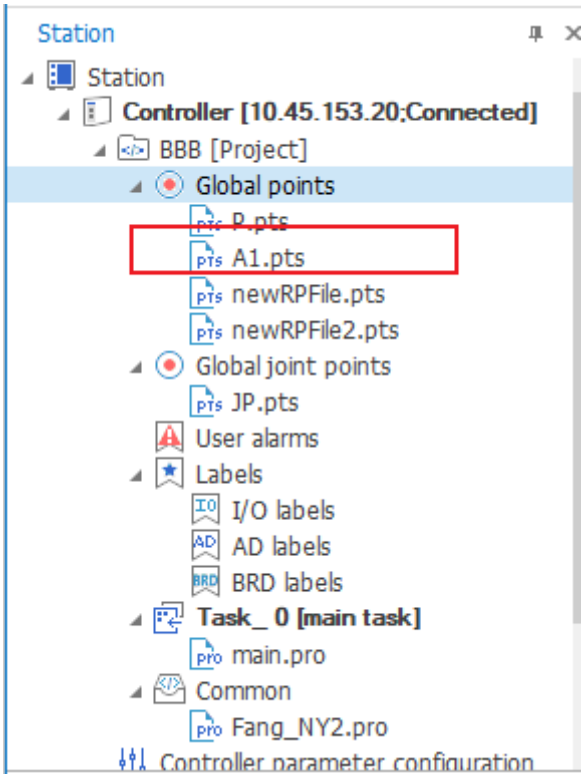


- Method 2:

1. Right-click on "Common" node and select "Add controller file".
2. See Step 2 of Method 1.

Results

When the operation is completed, the corresponding point file in the controller is copied to the "Data" folder of the project directory and a corresponding node is added to the project.



NOTE

- The added point file name starts with a letter and contains only letters, numbers and underscores, with a max length of 26 characters excluding the extension.
- A similar operation to this one is adding a local file, the latter being selecting a local point file and adding it to the project. For details, see the 5.2.18 Adding a Local File to the Project section.

5.2.20 Upgrading a Single File

When you use the "Add local file" function, if you add an old version of the file, the project upgrade process will also be triggered.

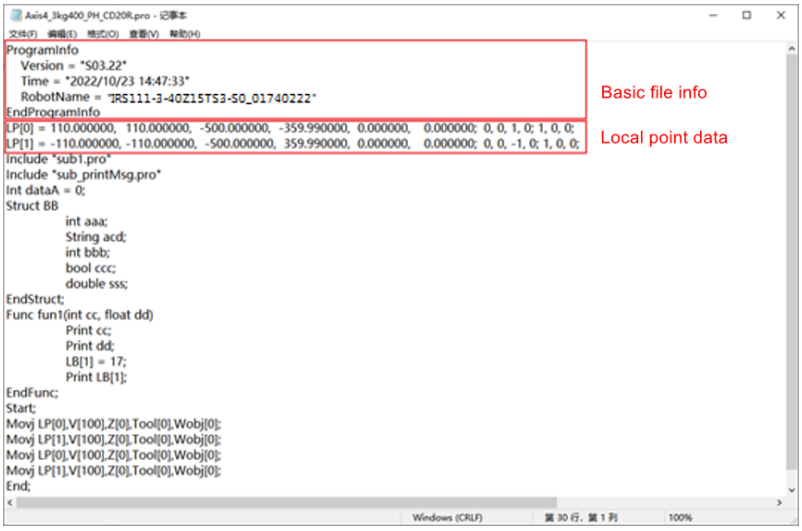
For details, see the 5.2.15.1 Creating a Program File section and the 5.2.5 Upgrading the Project section.

5.3 Program File Operations

5.3.1 Overview

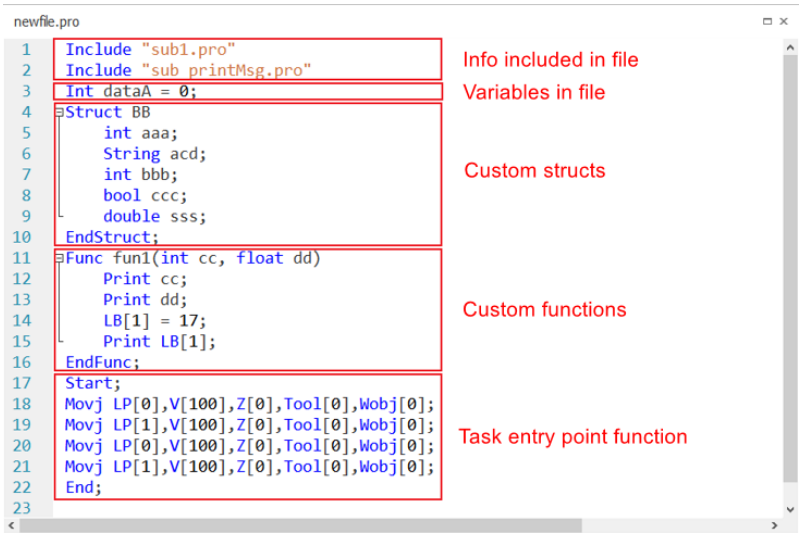
The contents of a program file include basic file information, point data, file information, custom functions, custom structures, in-file variables, and task entry functions. The basic file information and point data are not displayed in code editing.

The basic file information is not displayed in the code editor, but can be opened, but it can be viewed by opening it in Notepad.




```
ProgramInfo
  Version = "S03.22"
  Time = "2022/10/23 14:47:33"
  RobotName = "IRS111-3-40Z15T53-S0_01740222"
EndProgramInfo
LP[0] = 110.000000, 110.000000, -500.000000, -359.990000, 0.000000, 0.000000; 0, 0, 1, 0; 1, 0, 0;
LP[1] = -110.000000, -110.000000, -500.000000, 359.990000, 0.000000, 0.000000; 0, 0, -1, 0; 1, 0, 0;
Include "sub1.pro"
Include "sub_printMsg.pro"
Int dataA = 0;
Struct BB
        int aaa;
        String acd;
        int bbb;
        bool ccc;
        double sss;
EndStruct;
Func fun1(int cc, float dd)
        Print cc;
        Print dd;
        LB[1] = 17;
        Print LB[1];
EndFunc;
Start;
Movj LP[0],V[100],Z[0],Tool[0],Wobj[0];
Movj LP[1],V[100],Z[0],Tool[0],Wobj[0];
Movj LP[0],V[100],Z[0],Tool[0],Wobj[0];
Movj LP[1],V[100],Z[0],Tool[0],Wobj[0];
End;
```

The program file can be opened in the code editor, as shown below.




```
1  Include "sub1.pro"
2  Include "sub_printMsg.pro"
3  Int dataA = 0;
4  Struct BB
5      int aaa;
6      String acd;
7      int bbb;
8      bool ccc;
9      double sss;
10 EndStruct;
11 Func fun1(int cc, float dd)
12     Print cc;
13     Print dd;
14     LB[1] = 17;
15     Print LB[1];
16 EndFunc;
17 Start;
18 Movj LP[0],V[100],Z[0],Tool[0],Wobj[0];
19 Movj LP[1],V[100],Z[0],Tool[0],Wobj[0];
20 Movj LP[0],V[100],Z[0],Tool[0],Wobj[0];
21 Movj LP[1],V[100],Z[0],Tool[0],Wobj[0];
22 End;
23
```

5.3.2 Viewing the Point Data

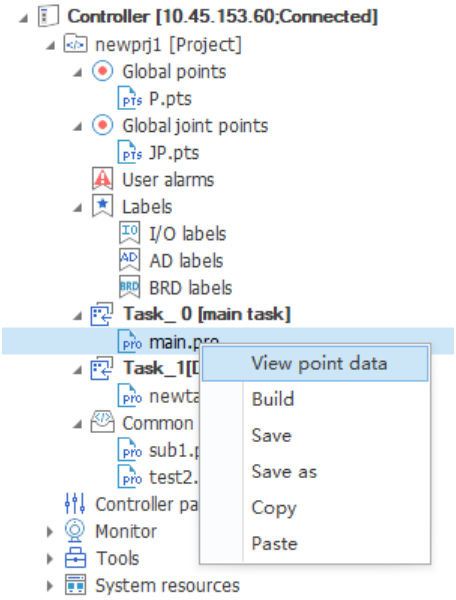
Each program file can contain its own point data, which we call local point variables.

Steps

There are three methods to view the point data:

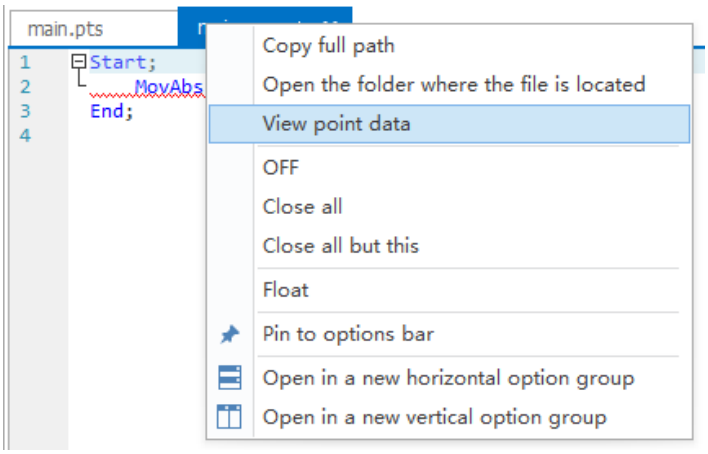
- Method 1:

Right-click on the program file node and select "View point data".



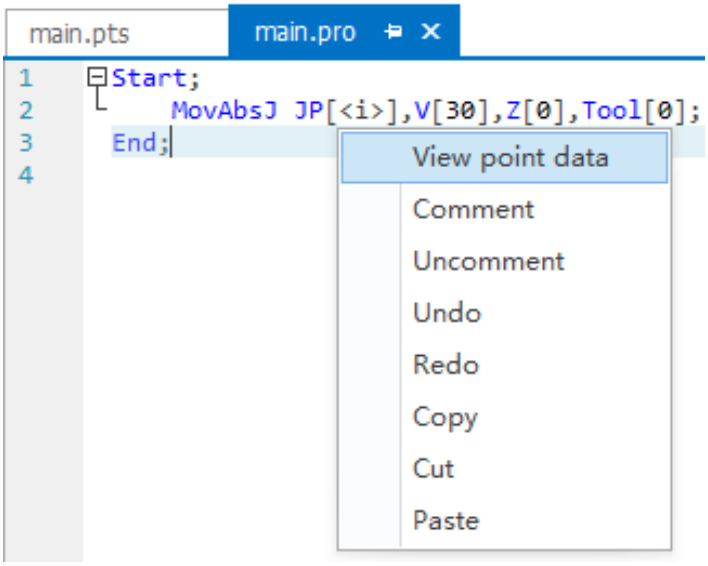
- Method 2:

Double-click on the program file name, right click on the program file title bar, and select "View point data".



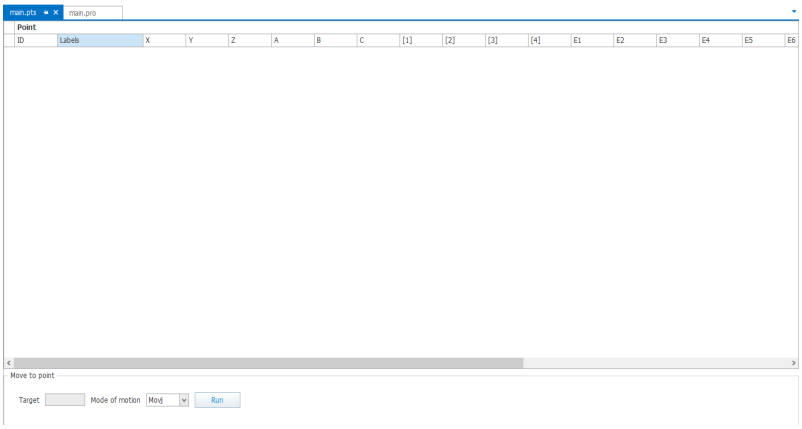
- Method 3:

In the code editor, right-click and select "View point data".

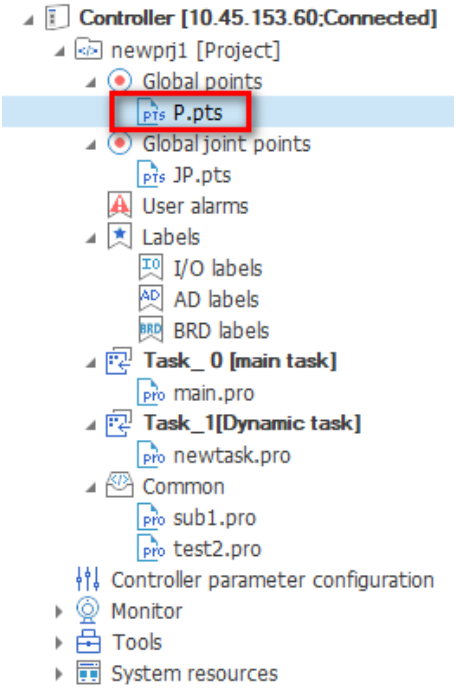


Results

After the operation is completed, you can see that a ".pts" file of a corresponding program file is opened. For example, if you view the point data of "main.pro", the "main.pts" page opens.



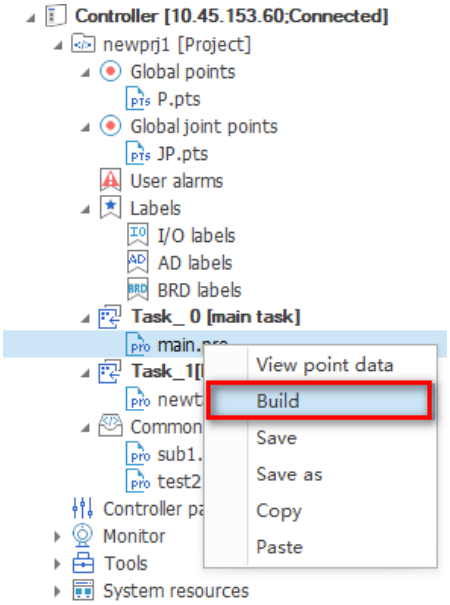
- For operation of editing the local point data file ".pts" in the program file, see the 6.2 Point Variables Editing section.
- If you want to view the global points, double-click "P.pts" file under the "Global points" node. For details, see the 6.2 Point Variables Editing section.



5.3.3 Compiling

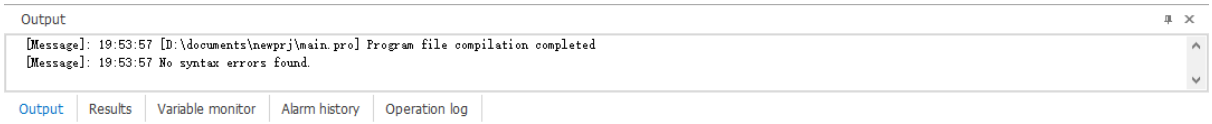
Compiling an individual program file will check the contents of the program file for syntax errors and update information about labels, local variables, custom functions, and custom structures within the program file to the code editor's syntax hints. While compiling the entire project also checks for syntax errors, it is not as efficient as compiling individual program files.

To compile a program file, such as "main.pro", right-click on it and select "Build".

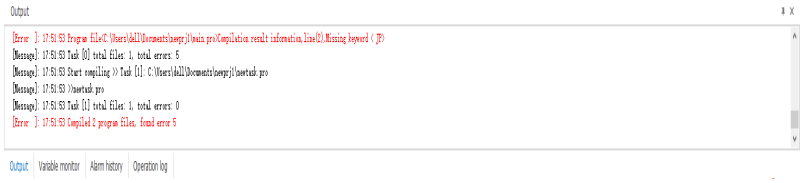


Results

- The compilation is successful:



- The compilation failed.

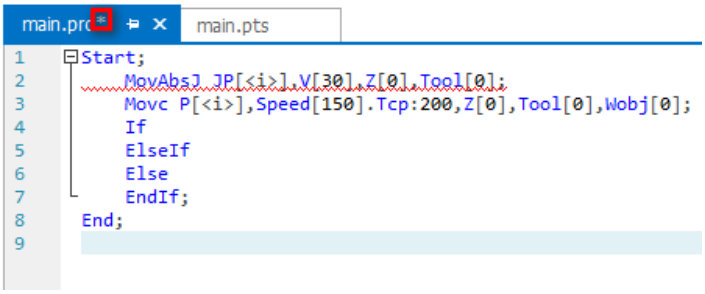


NOTE

- For compilation of the entire project, see the 5.2.9 Compiling the Project section.
- If a robot model mismatch is reported during compiling, refer to the following two methods for checking:
  - Check if the program contents and point data are saved. If not, click "Save" to eliminate the error.
  - Check if the number of axes of the model matches the project file.

5.3.4 Saving Changes to the Program File

You can save changes to the individual program files. If the point data are changed, such changes are also saved. When a program file is modified, the "*" sign appears in the title bar of the program file.



Steps

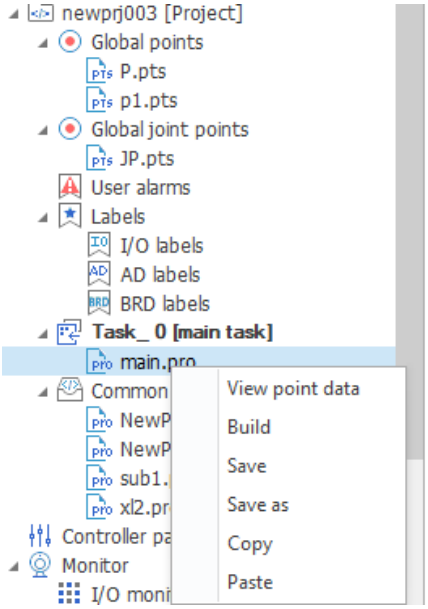
You can save a single program file in any of the following four methods.

- Method 1:

During program editing, press the shortcut key Ctrl+S.

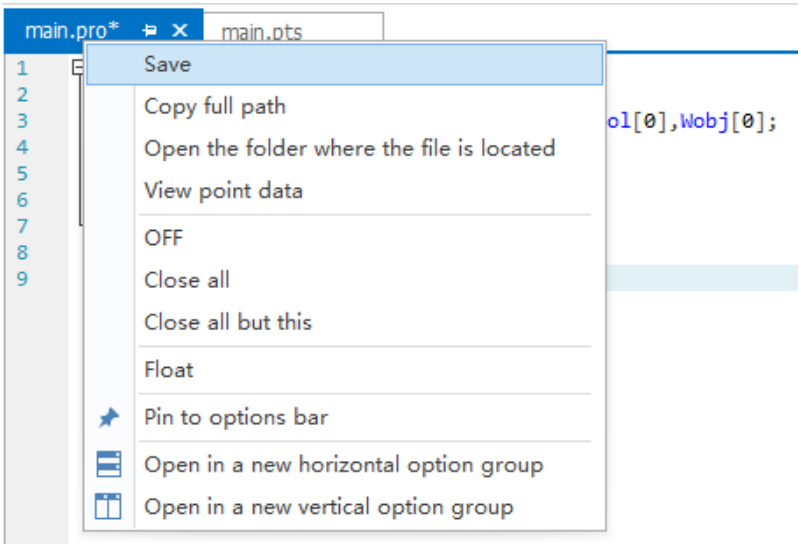
- Method 2:

Right-click on the program file node and select "Save".



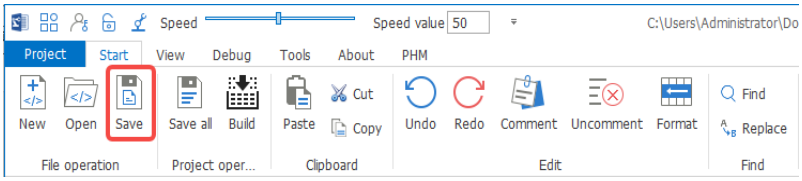
- Method 3:

In the program file title bar, right-click and select "Save".



- Method 4:

In the "Start" menu, click "Save".



Results

The contents of the file are saved to the file corresponding to the local project, and the "*" sign disappears.

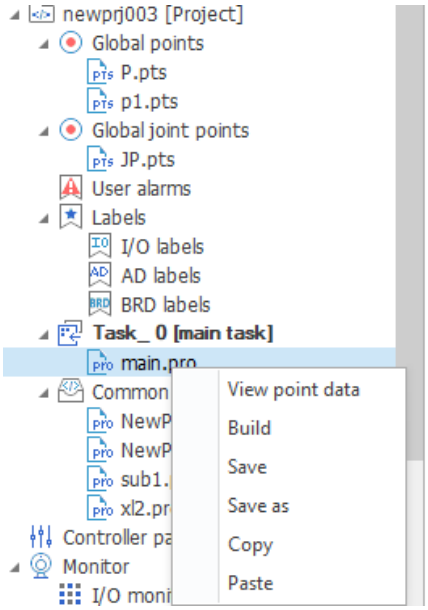
> **NOTE**
>
> After the program file is saved, the basic file information is updated. For example, the modification time is updated to the current system time and the robot model is updated to the current robot model.

5.3.5 Saving the Program File As

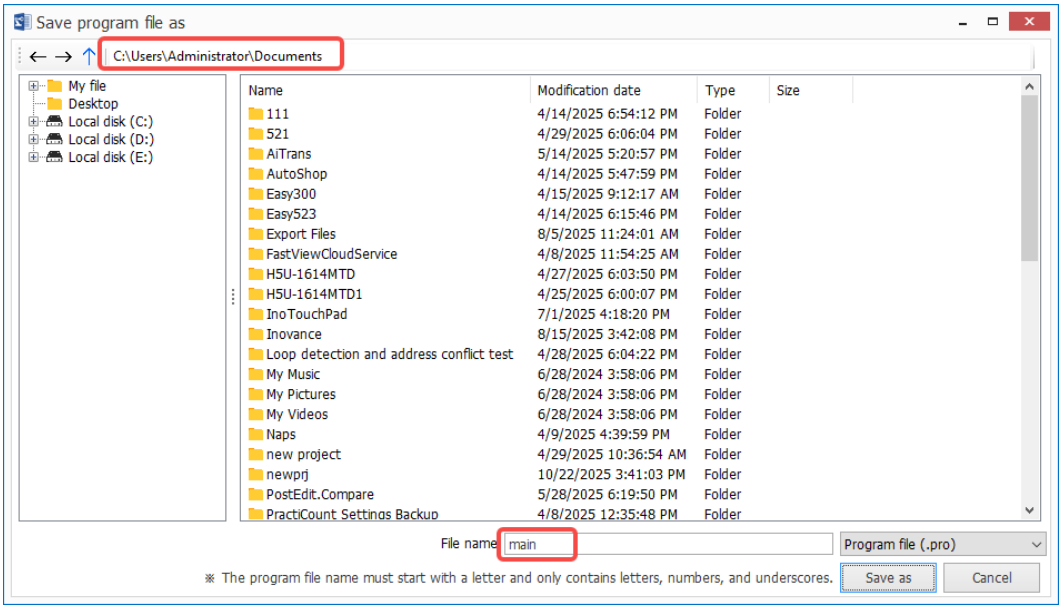
You can save a single program file to a different path for the purpose of exporting the file.

Steps

1. Right-click on the corresponding program file node and select "Save as".



2. In the pop-up dialog, select a local storage path, and enter the file name and click "Save as".



Results

The file is saved in the destination path.

5.3.6 Opening the Program file

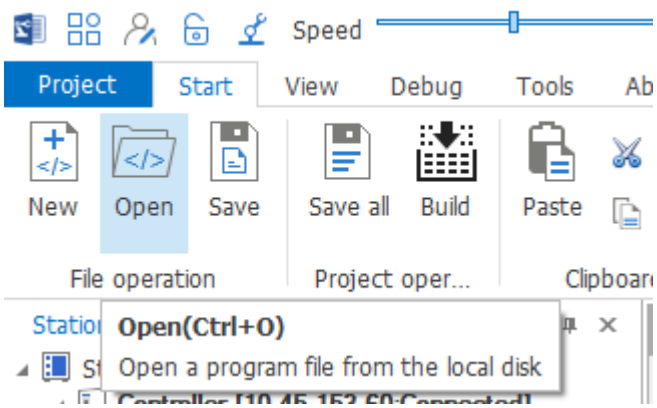
You can open the local program files. This function can be used when you want to open program files inside or outside the project.

File opening constraints:

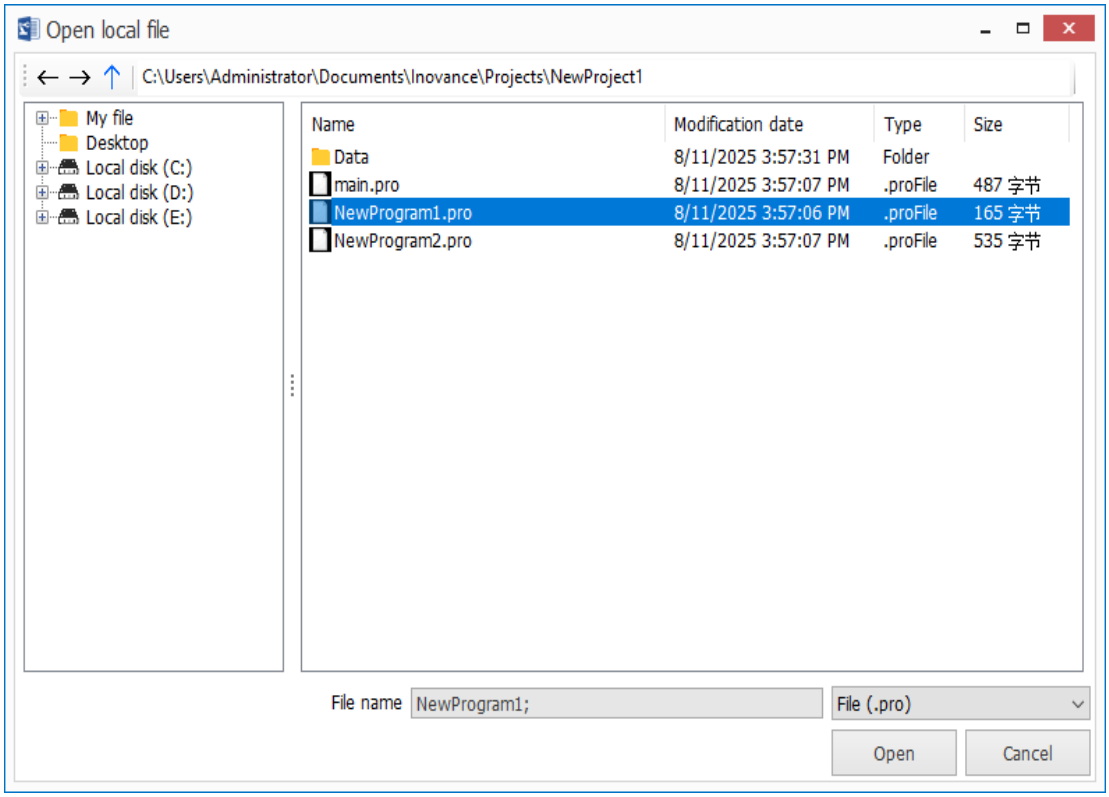
- When the project is empty, it is not allowed to open a program file.
- When you open a program file that is already in the project, open the program file directly.
- When you open a program file outside the project and there is no program file with the same name in the project, add the program file to the project and open it.
- When you open a program file outside the project and there is a program file with the same name in the project, it is not allowed to open the program file.

Steps

1. Go to "Start" > "Open".

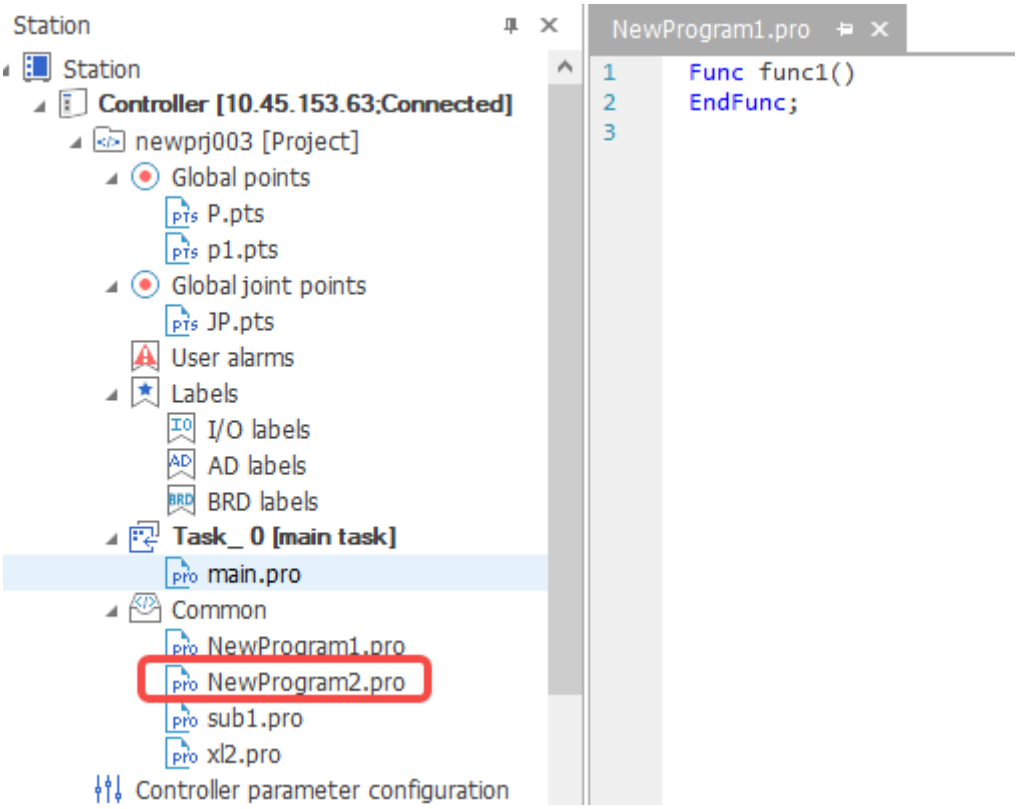


2. Browse the local disk for the program file you want to open and click "Open".



Results

After the operation is completed, if the selected program file is an in-project file or an out-of-project file with a unique name, the selected program file will appear in the project node. If the selected program file is an out-of-project file and there is a file with the same name in the project, it is not allowed to open selected program file.

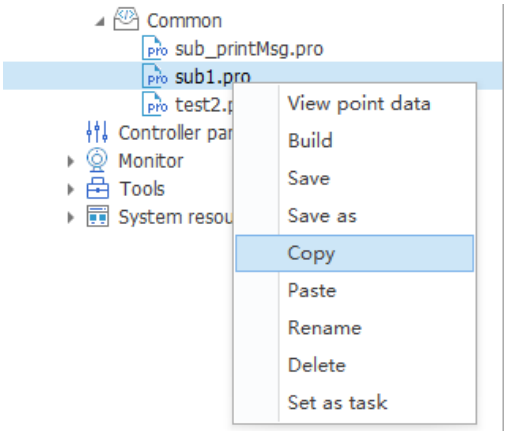


5.3.7 Copying and Pasting the Program File

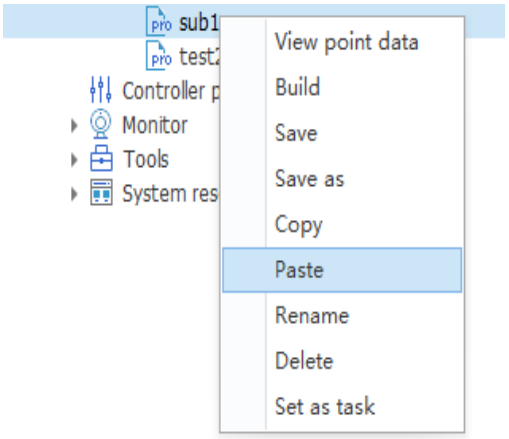
You can copy and paste program files inside the project. This function can be used when you want to modify a program without changing the original file.

Steps

1. Right-click on the corresponding program file node and select "Copy".

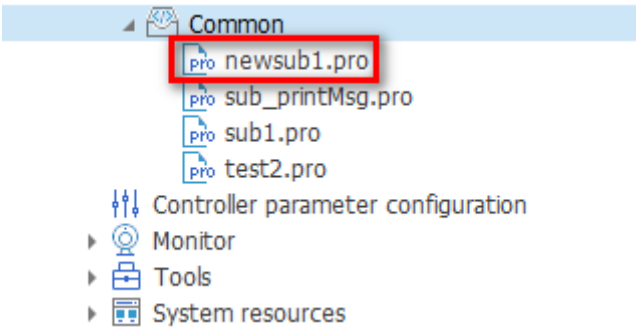


2. Right-click on the corresponding program file node and select "Paste".



Results

After the operation is completed, a new program file will appear in the project node. The contents of this program file are consistent with the copied program file.



NOTE

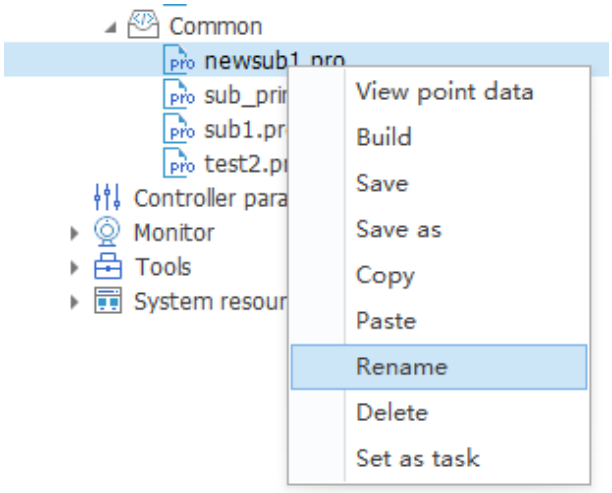
Each paste generates a new program file including "new" in its name.

5.3.8 Renaming the Program File

You can rename the project file.

Steps

1. Right-click on the corresponding program file node and select "Rename".



2. Enter a new program file name.

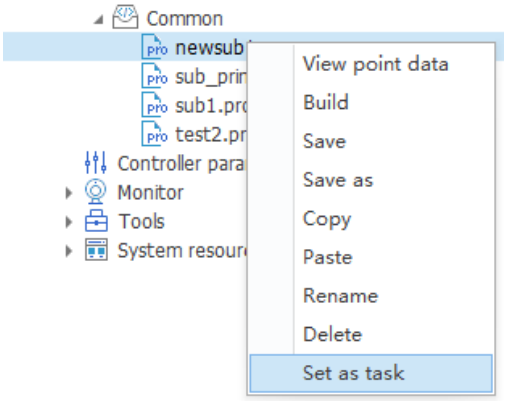
> **NOTE**
>
> - The main program file cannot be renamed.
> - For the naming rules for program files, see "Note" part in the 5.2.15.1 Creating a Program File section.

5.3.9 Setting the Program File as Task

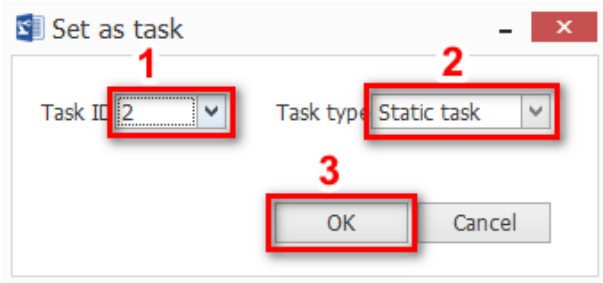
You can set the program file under the "Common" node as the entry program file for the task. It is suitable for projects requiring multi-tasking.

Steps

1. Right-click on the program file node and select "Set as task".

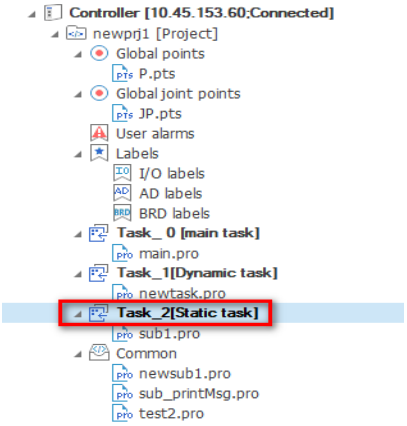


2. Set the task ID, task type, and then click "OK".



Results

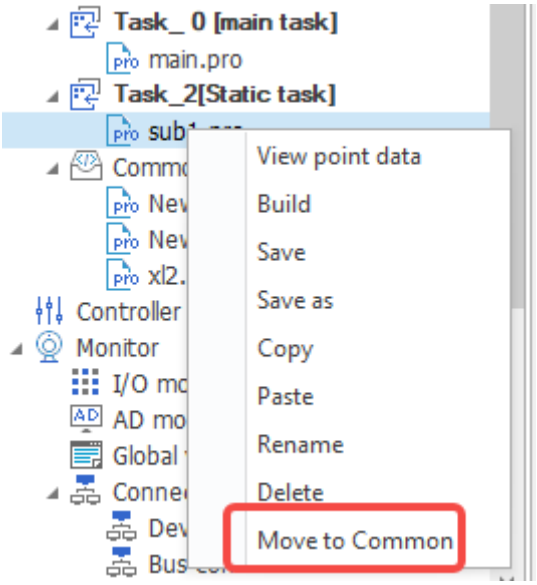
The task configuration, including the task ID, task type, and entry program file, appears under the project node.



NOTE

- You can override an existing task. If the main task is overridden, the original "main.pro" will be automatically renamed and moved to "Common" node, and the selected program file will be named to "main.pro". If the task overridden is not the main task, the original program will be directly moved to "Common".
- The operation with similar effect is "New task". For details, see the 5.2.16 Creating a Task section.

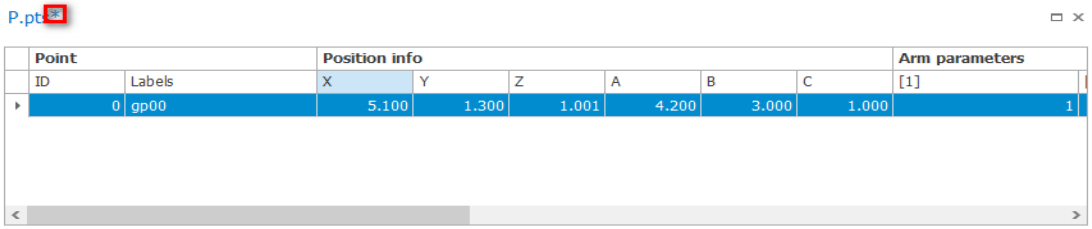
The reverse operation is to right-click on the task entry program and select "Move to Common".



5.4 Point File Operations

5.4.1 Saving Changes to the Point File

You can save the changes to the individual position point files. When a point file is modified, the "*" sign appears in the title bar of the point file.



Steps

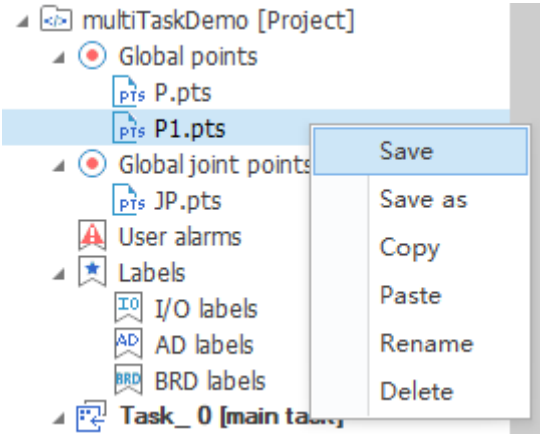
You can save a single point file in any of the following three methods.

- Method 1:

During the point editing, press the shortcut key Ctrl+S.

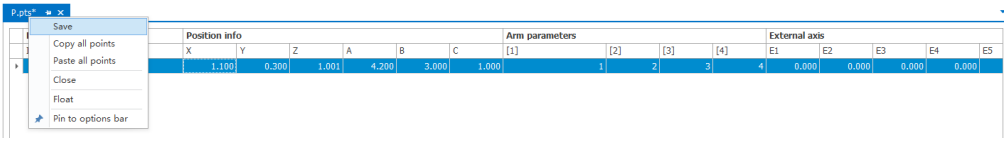
- Method 2:

Right-click on the point file node and select "Save".



- Method 3:

In the point file title bar, right-click and select "Save".



Results

The contents of the file are saved to the file corresponding to the local project, and the "*" sign disappears.

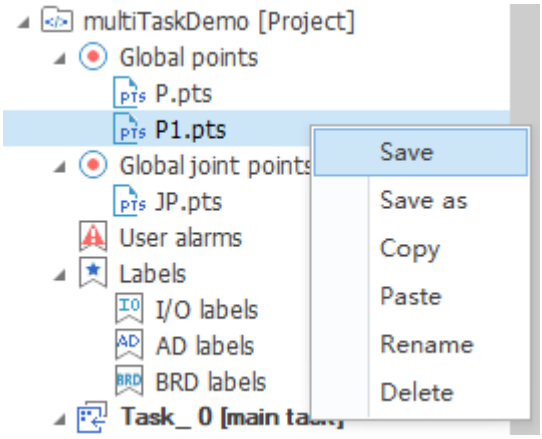
> **NOTE**
>
> After the point file is saved, the basic file information is updated. For example, the modification time is updated to the current system time and the robot model is updated to the current robot model.

5.4.2 Saving the Point File As

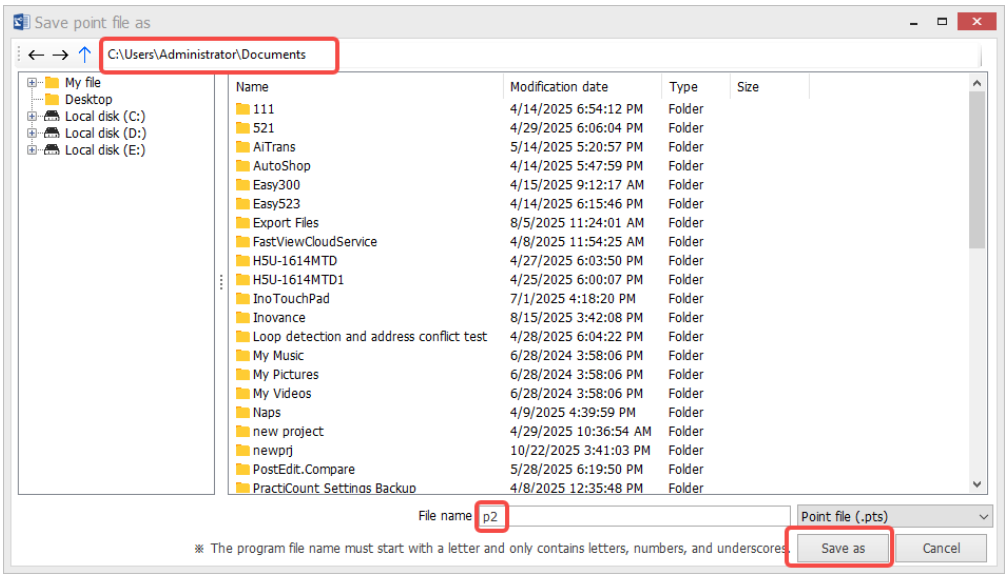
You can save a single point file to a different path for the purpose of exporting the file.

Steps

1. Right-click on the corresponding point file node and select "Save as".



2. In the pop-up dialog, select a local storage path, and enter the file name and click "Save as".



Results

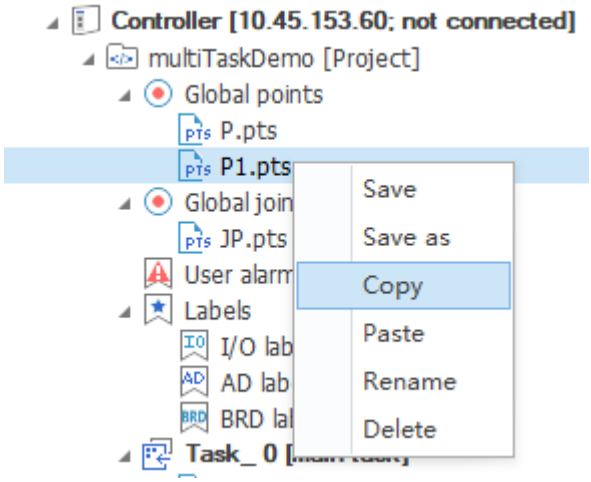
The file is saved in the destination path.

5.4.3 Copying and Pasting the Point File

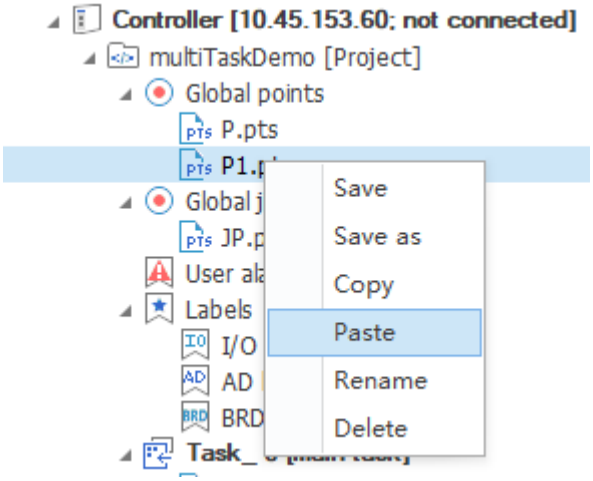
You can copy and paste point files inside the project. When you need to modify a program, but you want to keep the original file unchanged, the copy and paste function meets your needs.

Steps

1. Right-click on the corresponding point file node and select "Copy".

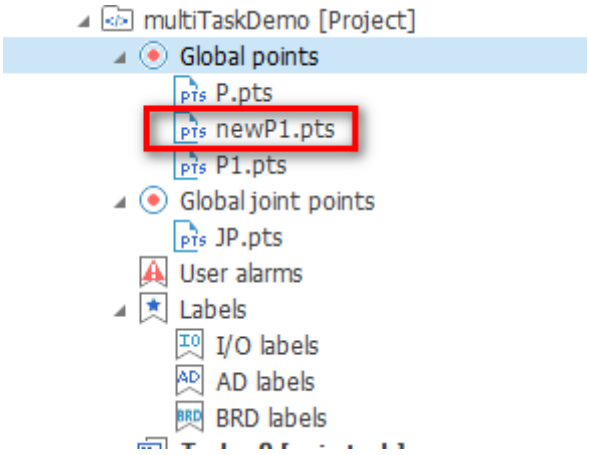


2. Right-click on the corresponding point file node and select "Paste".



Results

After the operation is completed, a new point file will appear in the project node. The contents of this point file are consistent with the copied point file.



NOTE

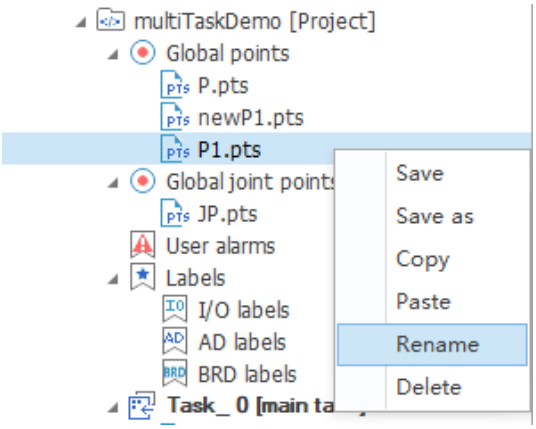
Each paste generates a new point file including "new" in its name.

5.4.4 Renaming the Point File

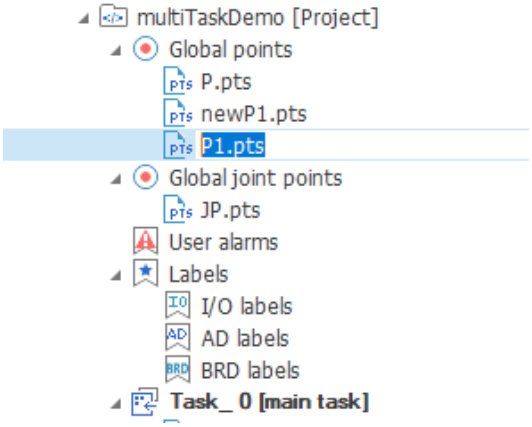
You can rename the point file.

Steps

1. Right-click on the point file node and select "Rename".



2. Enter a new point file name.



NOTE

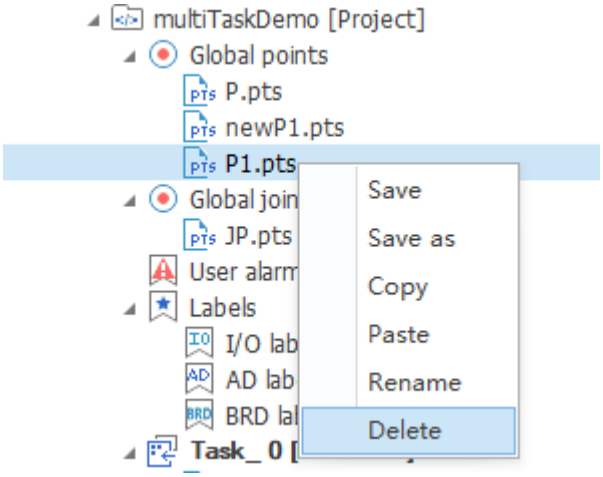
- The "P.pts" file cannot be renamed.
- For the naming rules for point files, see "Note" part in the 5.2.15.2 Creating a Point File section.

5.4.5 Deleting the Point File

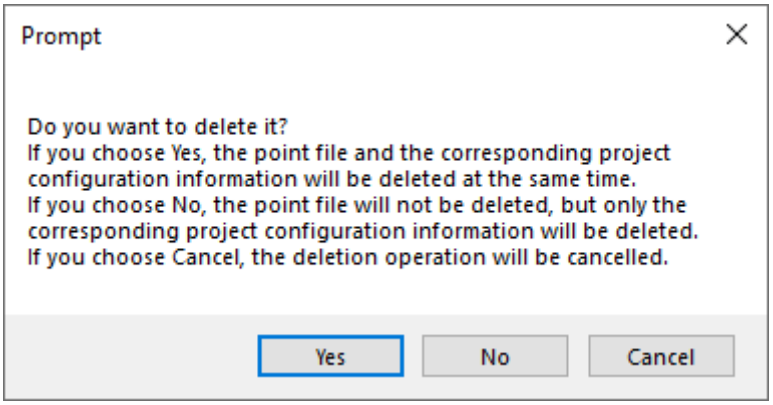
You can delete the point files from the project.

Steps

1. Right-click on the point file node and select "Delete".



2. After you select "Delete", a selection pop-up dialog appears, allowing you to select different delete modes, as shown below.



NOTE

The "P.pts" file cannot be deleted.

6 Programming

6.1 Program Editing

6.1.1 Overview

The program file is part of a project. Generally, a project consists of a "main.pro" file and several other program files. The extension of program file is ".pro".

A program file is a module; for details, see the 5.1.3 Module section.

You can open the program file editing page in either of the following two methods.

- Method 1: Double-click the program file node.
- Method 2: Go to "Start" > "Open" and select the program file.

Results

The program file editing page opens, as shown below.

main.pro

```
1   Include "sub1.pro";                          # Include program file: sub1.pro
2   Start;                                       # Task entry command
3       Print "MAIN Start";                      # Print "MAIN Start"
4       Int intA = 0;                            # Define local variable intA
5       While intA < 3                           # Execute loop if intA < 3
6           intA = intA+1;                       # Perform local variable calculation
7           B[0] = intA;                         # Variable assignment
8           sub1.funcPrint(intA);                # Call function from other program files
9           Movj pStart,V[30],Z[0],Tool[0];      # Movj move to pstart
10          Movj p001,V[30],Z[0],Tool[0];        # Movj move to p001
11          Movj pEnd,V[30],Z[0],Tool[0];        # Movj move to pEnd
12      EndWhile;
13      Print "Main End";                        # Print "MAIN End"
14  End;
```

6.1.2 Programming Rules

A program statement consists of characters. You can type the statement directly using the keyboard. During typing, the software will provide syntax hints based on the current input. For example, when you enter one or more characters of a keyword, the software prompts you to select a complete keyword that matches the entered characters. When you enter a space or "(" after entering a keyword, the software will prompt for the complete program statement. Use information is also displayed when different keywords are selected.

Keywords are highlighted in blue by the software. The highlighting is based on full keyword match, all upper case, and all lower case. The keywords are case-insensitive, while the variables are case-insensitive.

Indent statements in each loop. Press the "Enter" key at the end of a line within a cycle to automatically jump to the next line, with a certain indentation applied. Examples include Struct…EndStruct, Func…EndFunc, IF..Else…EndIf, Switch…Case…Default…EndSwitch, While…EndWhile, For…EndFor, etc.

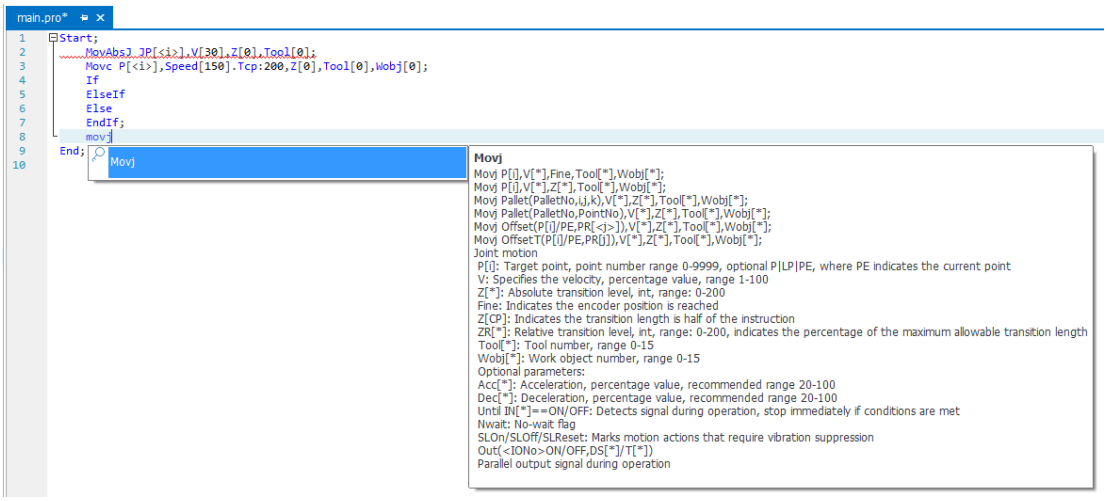
The indentation effect during editing is shown below.

```
newp.pro
1   Func func1()
2       For B[0]=0,B[0]<5,Step[2]
3           Movj P[0],V[30],Z[0],Tool[0];
4           Movj P[1],V[30],Z[0],Tool[0];
5       EndFor;
6   EndFunc;
7
```

6.1.3 Syntax Help

Keyword Hints

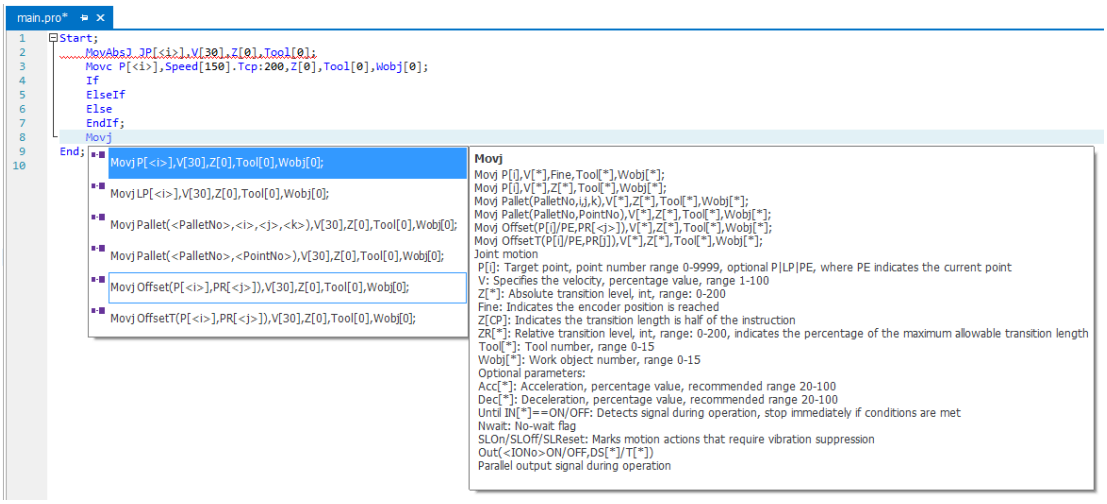
When one or two characters of an instruction are entered, a syntax help window appears, indicating the matching keyword, with a drop-down selection menu displaying the syntax of the statement or function of the corresponding instruction. Select an item using the up and down arrow keys or directly click the item, and then press "Tab" or "Enter" to confirm the selection.



Statement Completion

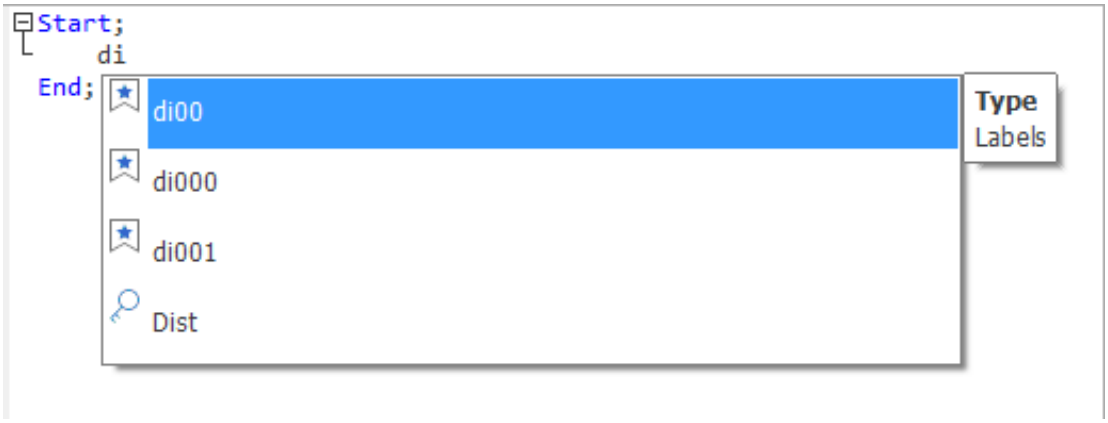
After entering a keyword or function name, enter a space to automatically display the menu of matching items of the instruction statement.

Select an item and press "Tab" or "Enter" to confirm the selection.



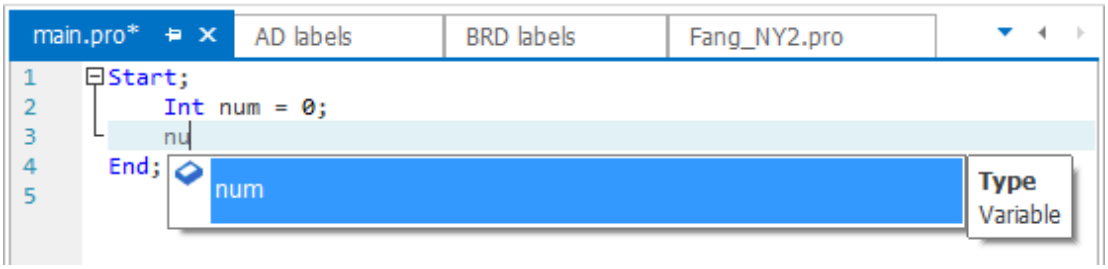
Label Hints

The software provides hints for global point variable labels, local point variable labels, DI/DO labels, and AD/DA labels (project compilation is required, otherwise the hints may not be complete.)



Variable Hints

The software provides hints for custom variables within function, custom variables within module, and module name.



Mechanical Unit Name Hints

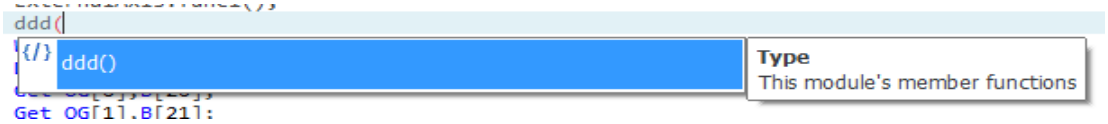
The software provides hints for the mechanical unit name.



Secondary Hints

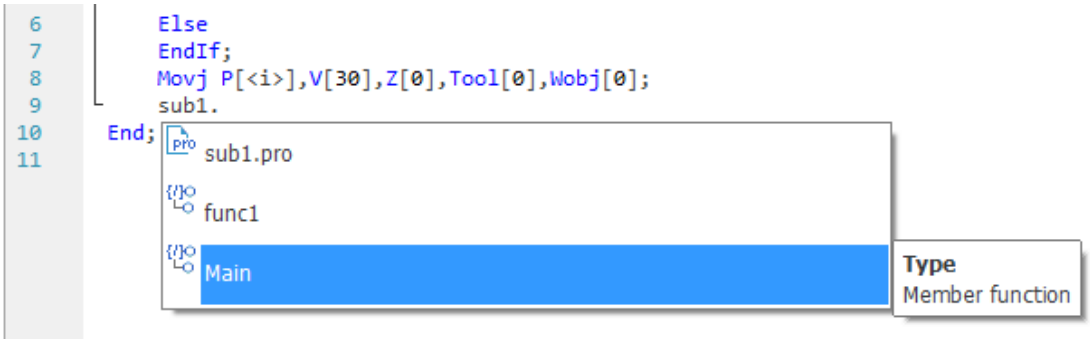
Auto-completion of functions in the module: After the function name is entered, enter "(" to get hints for completion of the full function including parameters. Press CTRL + Left/Right arrow key to move the cursor quickly to the position of the contents inside [] <>. When the cursor moves to <>, it will select <> as well as the content within it.

Auto-completion of function calls:

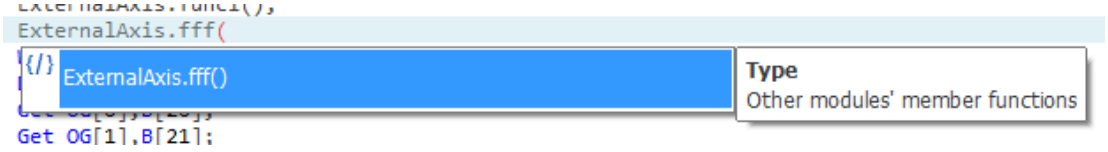


Hints for member function names: After entering the names of other modules, press "." to prompt the members of other modules for you to choose to complete the input of the function names of other modules.

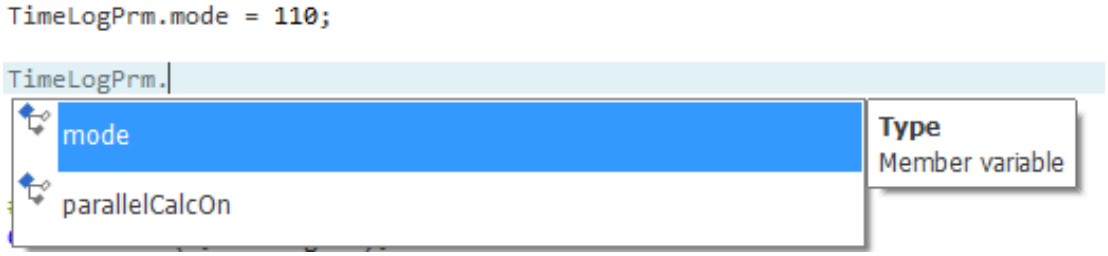
> **CAUTION**
>
> You need to have inserted include "***.pro" at the beginning of the program file, and have compiled the project.



Auto-completion of member functions: After entering the function names of other modules, enter "(" to prompt the completion of member functions.



Hints for structure members: The software provides input hints for the members of the structure.



6.1.4 Syntax Check

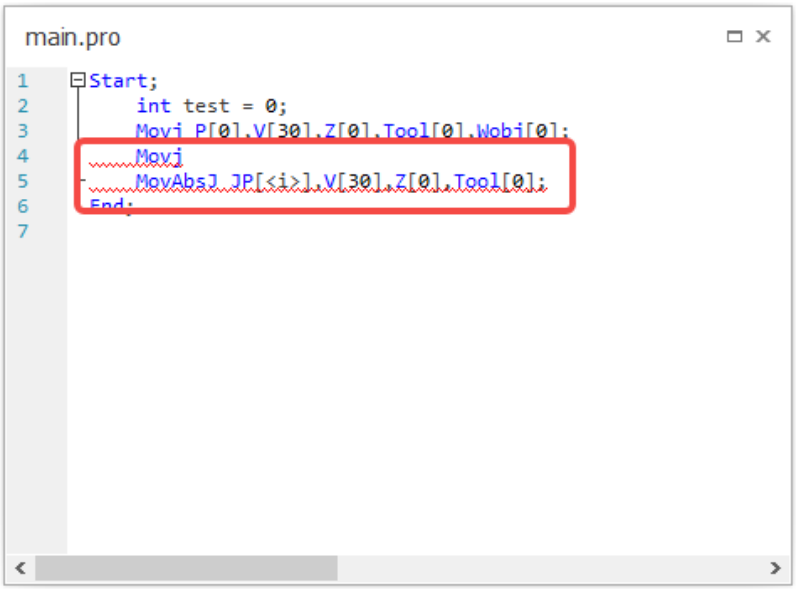
The syntax check is divided into automatic syntax check during editing, single program compilation check, and project compilation check.

Automatic syntax check

During program editing, the current program file is automatically checked. For example, after characters are entered, the red wave line is automatically added to the error line.

Single program compilation check

Like automatic syntax check which adds red wave line, the single program compilation check also adds a red wave line to the error line. For compilation of a single project, see the 5.3.3 Compiling section.



Project compilation check

Project compilation is to compile the whole project. If there is an error, the error information will be displayed in the Output pane. You can double-click on the error message line to automatically jump to the error line of the program file. Project compilation does not mark red wave lines. For project compilation, see the 5.2.9 Compiling the Project section.

6.1.5 Shortcuts

Ctrl+A: Select all.

Ctrl+C: Copy. Single-line copy is supported. In other words, pressing Ctrl+C copies the line where the cursor is located when no string is selected.

Ctrl+X: Cut. Single-line cut is supported. In other words, pressing Ctrl+X cuts the line where the cursor is located when no string is selected.

Ctrl+V: Paste.

Ctrl+Z: Undo.

Ctrl+Y: Redo.

Ctrl+S: Save.

Ctrl+D: Copy the current line and paste it to the next line.

Ctrl+T: Swap the current line from the previous line.

Ctrl+F: Find.

Ctrl+R: Replace.

Alt+A: Find all.

Alt+F: Find next.

Alt+P: Find previous.

Ctrl+K: Quick text alignment (adjust code indent).

Ctrl+i: Insert a tab character.

Ctrl+[: Jump to the beginning of the previous edit segment (the edit segments are separated by blank lines).

Ctrl+]: Jump to the beginning of the next edit segment (the edit segments are separated by blank lines).

Ctrl+|: Jump to the next word (The word here is delimited in terms of case, number, special symbols such as semi-colon, parentheses, comma.)

Ctrl+/: Jump to the previous word (The word here is delimited in terms of case, number, special symbols such as semi-colon, parentheses, comma.)

Ctrl+J: Wrap the line at the current cursor position, with the cursor at the front of the line after wrapping.

Ctrl+M: Wrap at the front of the current line, with the editing cursor at the front of the line after wrapping.

Ctrl+L: Delete the current line.

Ctrl+Scroll wheel: Text scaling.

Home: Jump to the beginning of the line.

End: Jump to the end of the line.

PageUp: Go to the previous page.

PageDown: Go to the next page.

Insert: Switch to the insert mode and the cursor changes from a blinking vertical line to a blinking horizontal line.

Delete: Delete a character after the cursor.

Backspace: Delete a character before the cursor.

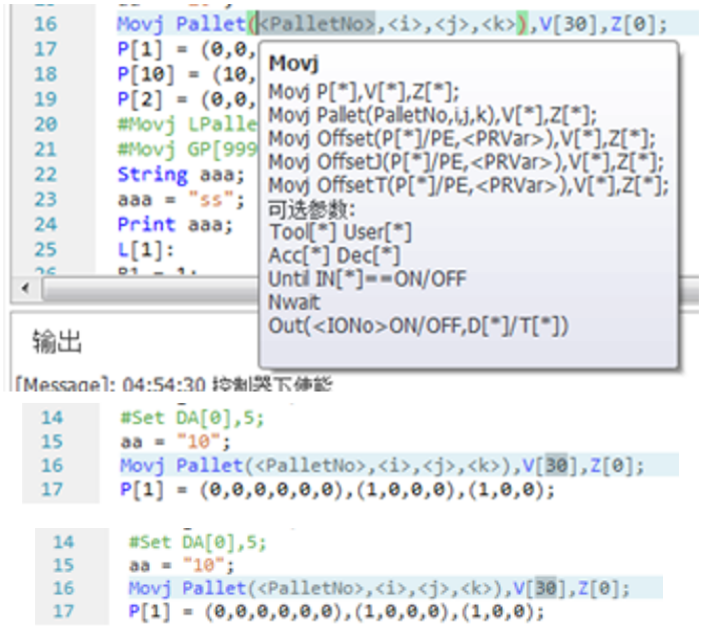
Ctrl+Backspace: Delete a word before the cursor (The word here is delimited in terms of case, number, special symbols such as semi-colon, parentheses, comma.)

Ctrl+Delete: Delete a word after the cursor (The word here is delimited in terms of case, number, special symbols such as semi-colon, parentheses, comma.) Taking "Movj P[3],V[20],Z[0],Tool[0];" as an example, when the cursor is after the semicolon, pressing Ctrl+Backspace will change it to "Movj P[3],V[20],Z[0],Tool[0".

Arrow keys: Move the cursor.

Ctrl+Alt+Mouse click-drag: Select the rectangle to delete the selection. You cannot edit the selection simultaneously.

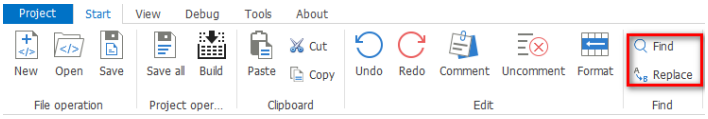
Ctrl+Left: Select the contents of the previous brackets on the current line (The types of brackets include square brackets and angle brackets. Note that the contents within the square brackets will be selected, excluding the square brackets, while both the contents and angle brackets will be selected at the same time.) to facilitate users to switch between editable areas. See the following figure. See the following figure.



Ctrl+Right: Select the contents of the next brackets on the current line.

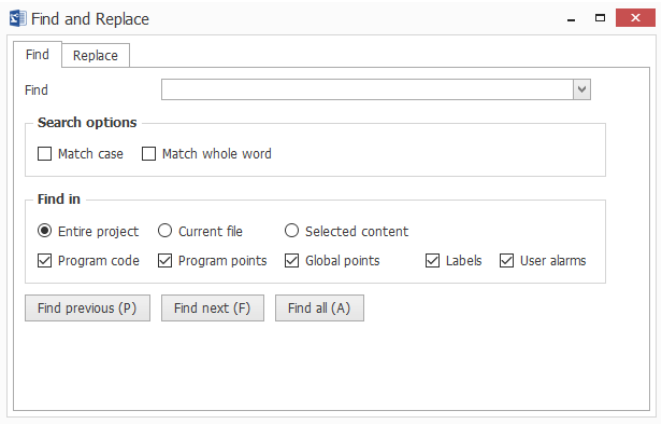
6.1.6 Find and Replace

Click "Find" or "Replace" button in the menu bar, or press Ctrl+F to open the "Find" dialog box, and Ctrl+R to open the "Replace" dialog box.

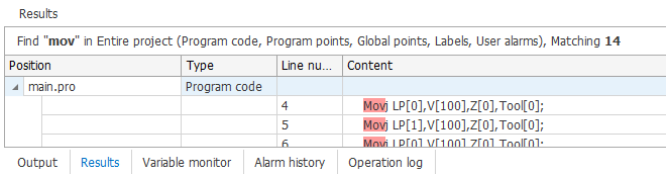


Find

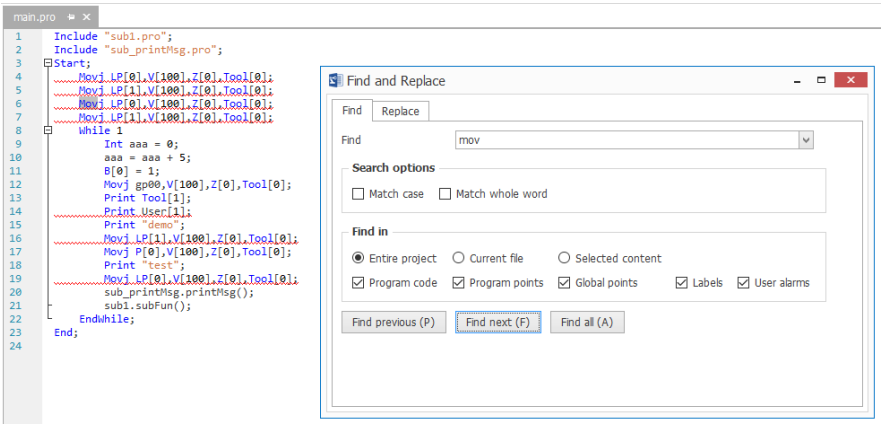
- The software supports condition-based search for program code, local points, global points, labels, and user alarms throughout the project, as shown below.



- When you click the "Find All" button, the search results are displayed in the bottom of the software window. Double-click the result to go to the specific position of the result and highlight it.

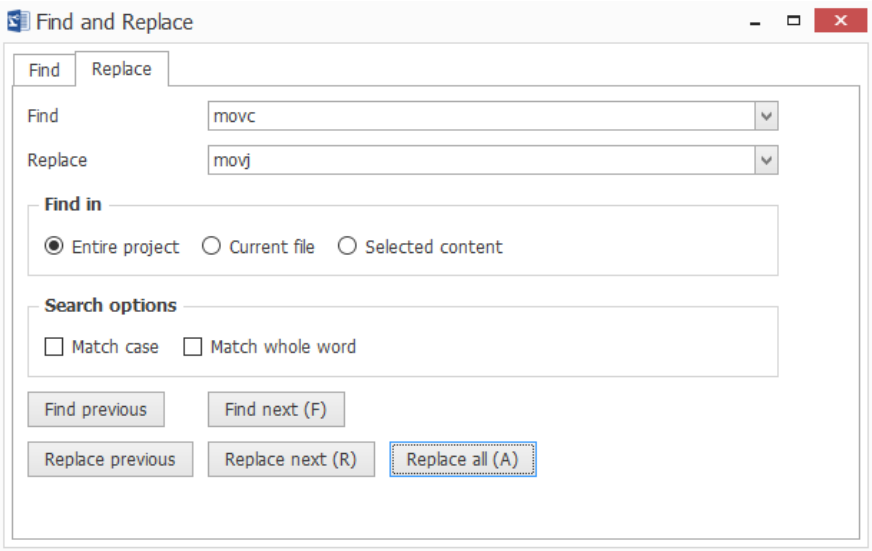


- When you click the "Find previous" or "Find next" button, the position of the matching item is automatically opened and highlighted.

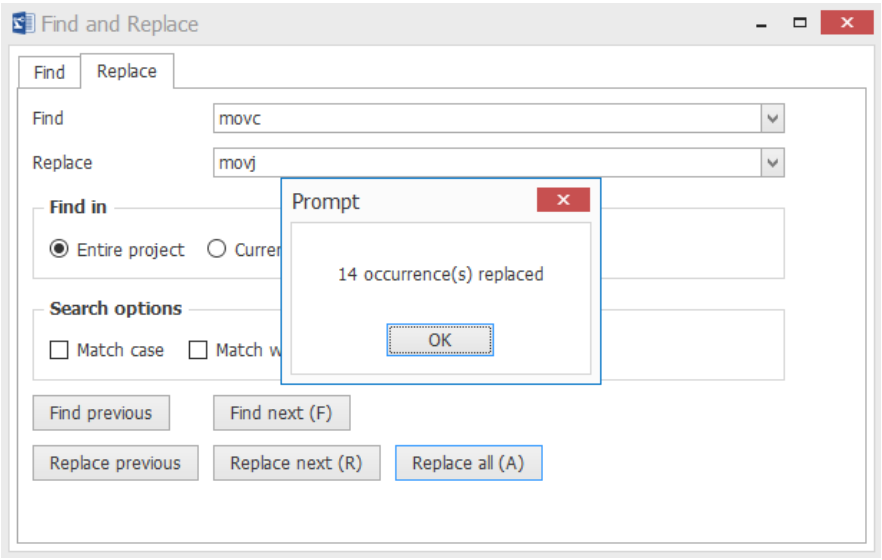


Replace

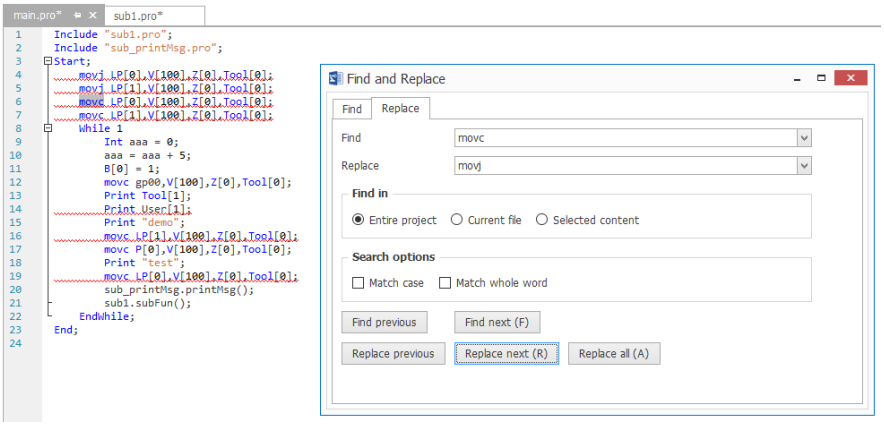
- The software supports replacing the contents of the program files in the entire project, as shown below.



- When you click the "Replace all" button, all matches are automatically replaced.



- When you click the "Replace previous" or "Replace next" button, the matching item is automatically selected. Click the button again to complete the replacement.



6.2 Point Variables Editing

6.2.1 Overview

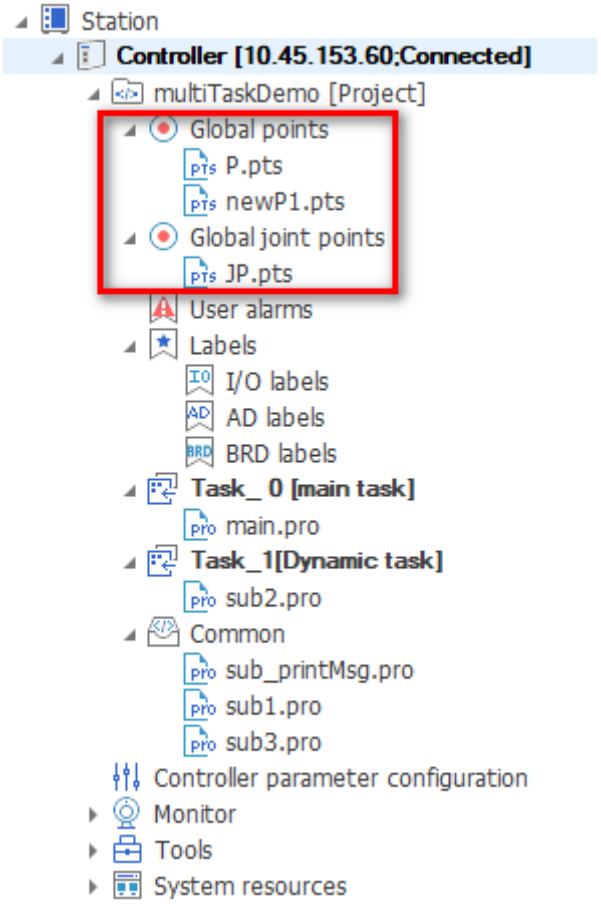
Position variables are categorized into local points and global points.

Local points

The local points exist in the program file and are only valid in the current program file. For editing of the local points, see the 5.3.2 Viewing the Point Data section.

Global points

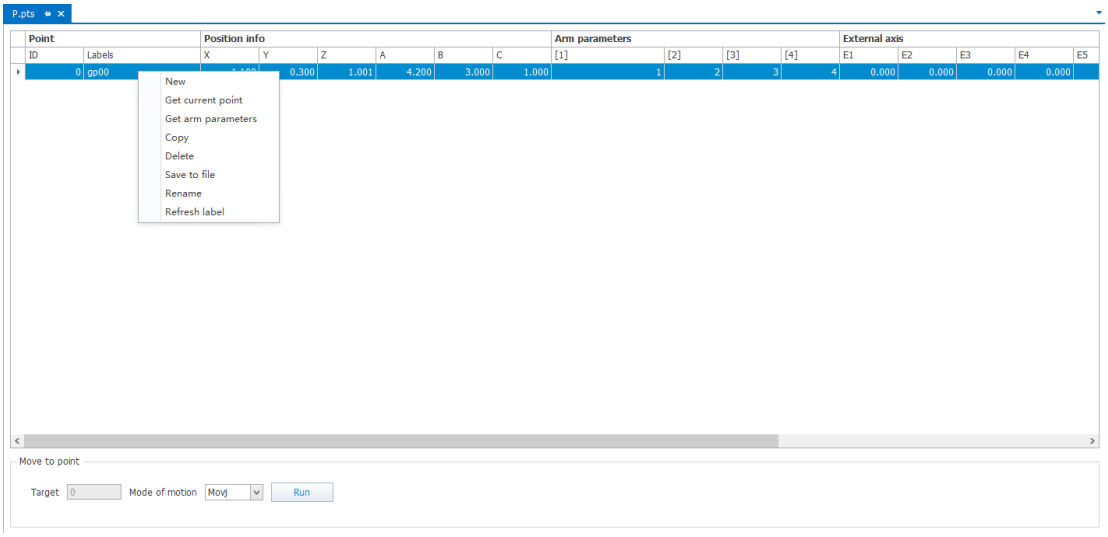
The global points are saved separately in the file under the project directory. The file name "P.pts" indicates global position points and J"P.pts" indicates global joint points. The global points are valid for the entire project. You can double-click the point file to edit the global points.



6.2.2 Editing the Point Variables

Editing the Global Points

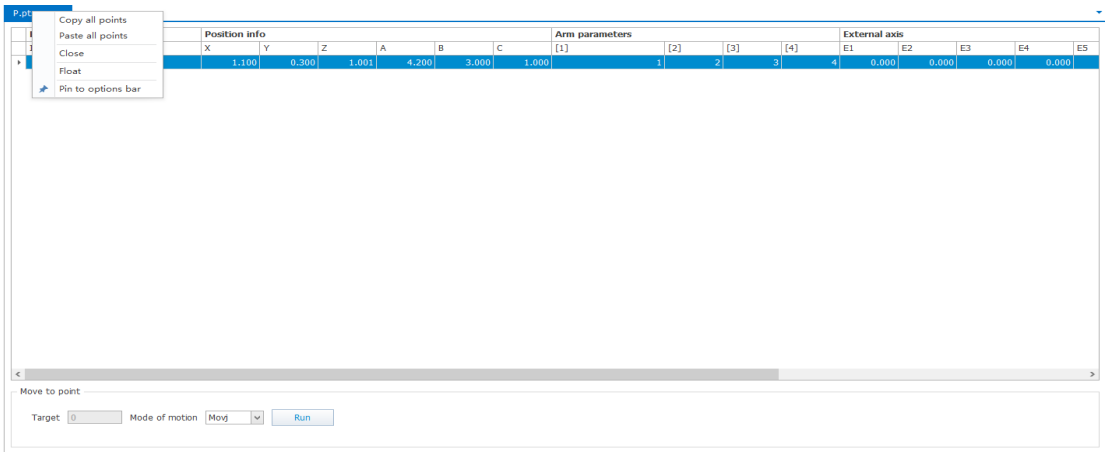
- Method 1: Right-click on the points page.



The related parameters are as shown below:

Function	Description
New	Add a new point variable to the list. By default, the number of the points is incremented by 1 from the existing largest point number.
Get current point	Assign the current position of the robot to the selected point.
Copy	Copy the selected point data to the clipboard. You can copy one or more points.
Paste and add	Paste the point data from the clipboard to add a new point. The index of each added point is incremented by 1 from the existing largest point index.
Paste and replace	Replace the selected point data with the point data from the clipboard. The index of the replaced point remains unchanged.
Delete	Delete the selected one or more points.
Save to file	Save changes to the "P.pts" file.
Rename	Modify the ID of a single point variable (or double-click the point ID).
Refresh label	Synchronize the labels of this file to other point files for a multi-point recipe.

- Method 2: Right-click on the title bar of the page.



The related parameters are as shown below:

Function	Description
Copy all points	Copy all points in the list on the current page to the clipboard.
Paste all points	Paste all points into the list on the current page in manner of paste and add.

- Method 3: Press the shortcut keys.

Delete: Delete the selected lines.

Ctrl+C: Copy the selected lines to the clipboard.

Ctrl+V: Paste the point data from the clipboard to the current list in manner of paste and add.

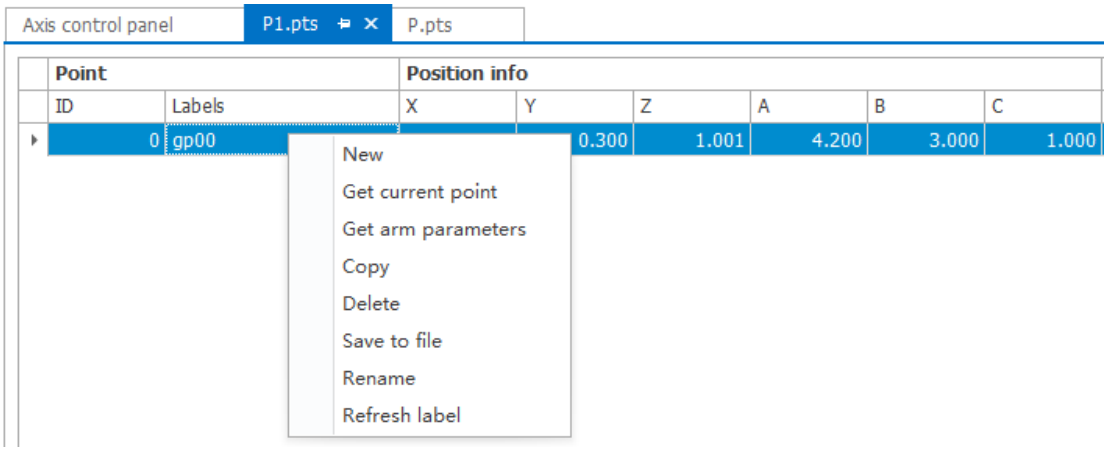
Ctrl+Shift+V: Paste the point data from the clipboard to the current list in manner of paste and replace.

Ctrl+X: Cut the currently selected point data to the clipboard.

Ctrl+S: Save changes to the "P.pts" file.

Editing the Local Points

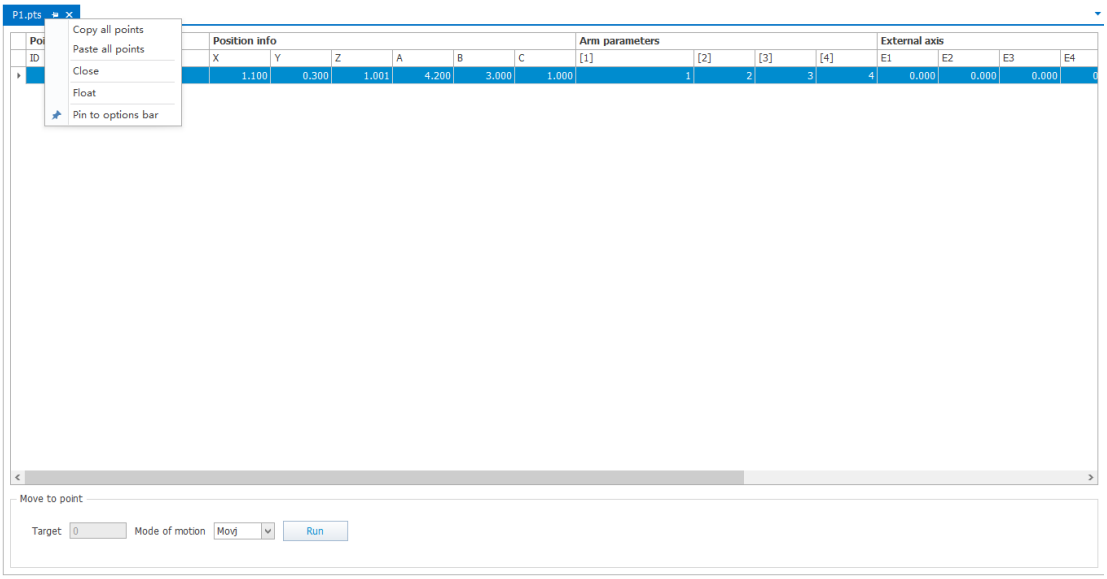
- Method 1: Right-click on the points page.



The related parameters are as shown below:

Function	Description
New	Add a new point variable to the list. By default, the number of the points is incremented by 1 from the existing largest point number.
Get current point	Assign the current position of the robot to the selected point.
Copy	Copy the selected point data to the clipboard. You can copy one or more points.
Paste and add	Paste the point data from the clipboard to add a new point. The index of each added point is incremented by 1 from the existing largest point index.
Paste and replace	Replace the selected point data with the point data from the clipboard. The index of the replaced point remains unchanged.
Delete	Delete the selected one or more points.
Save to file	Save changes to the program file.
Rename	Modify the ID of a single point variable (or double-click the point ID).

- Method 2: Right-click on the title bar of the page.



The related parameters are as shown below:

Function	Description
Save	Save point data to a file.

Function	Description
Copy all points	Copy all points in the list on the current page to the clipboard.
Paste all points	Paste all points into the list on the current page in manner of paste and add.

- Method 3: Press the shortcut keys.

Delete: Delete the selected lines.

Ctrl+C: Copy the selected lines to the clipboard.

Ctrl+V: Paste the point data from the clipboard to the current list in manner of paste and add.

Ctrl+Shift+V: Paste the point data from the clipboard to the current list in manner of paste and replace.

Ctrl+X: Cut the currently selected point data to the clipboard.

Ctrl+S: Save changes to the program file.

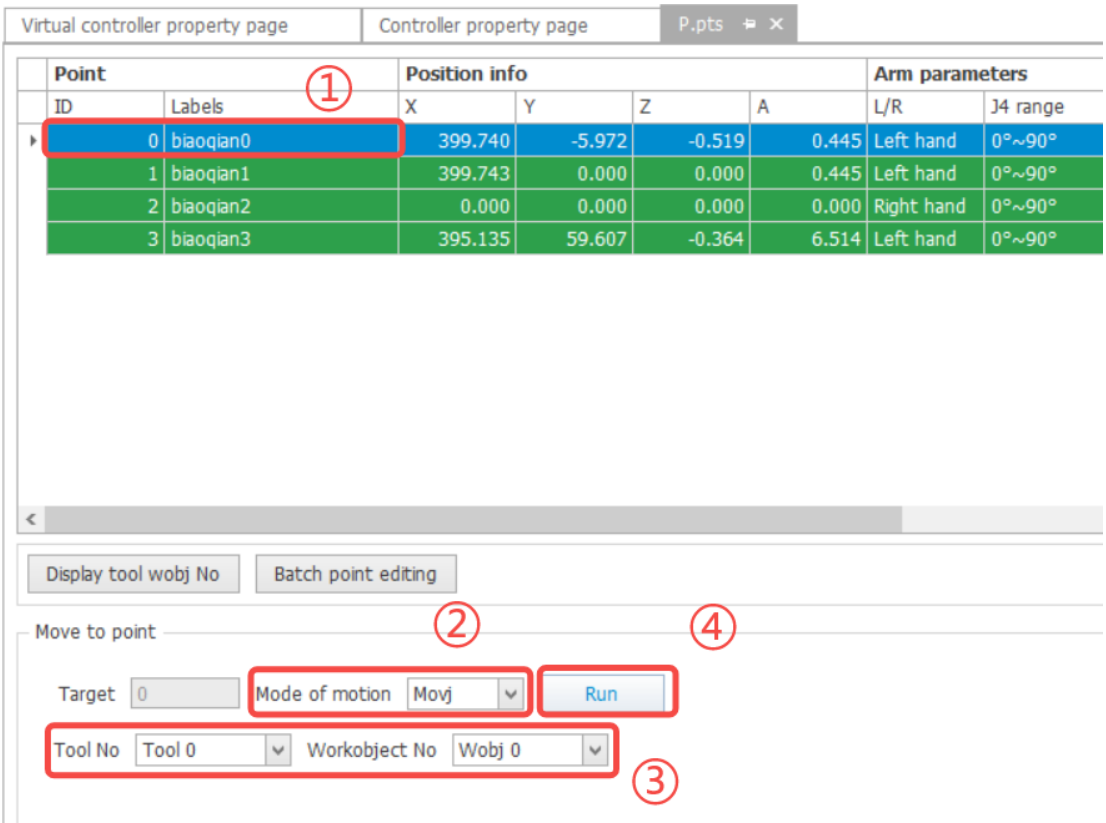
6.2.3 Moving to the Selected Point

You can verify whether the points in the point data list match the points that the actual robot needs to move to. If significant mismatch occurs, you need to correct the points.

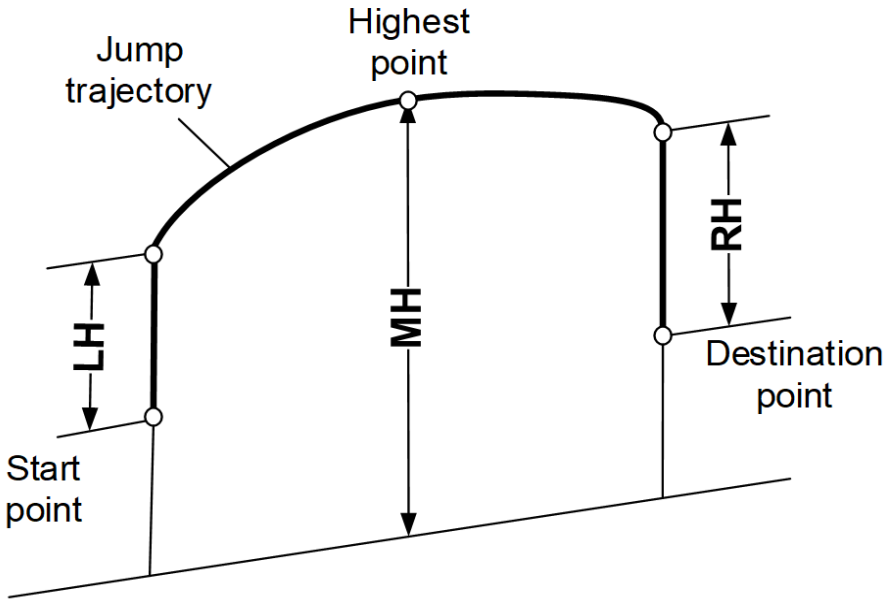
Prerequisite

Ensure that the robot servo is enabled.

Operate in the "Move to point" pane:

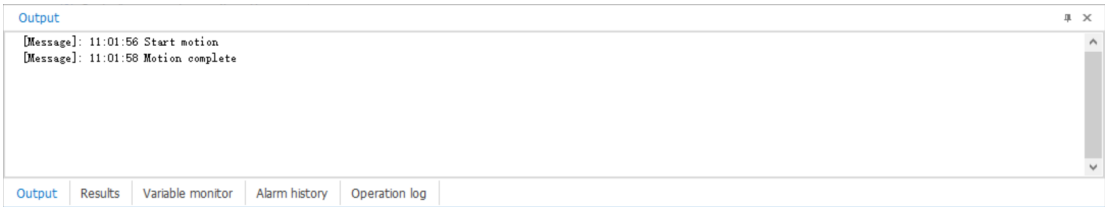


1. Select the target point to be moved to.
2. Select the motion mode, including Movj, Movl, Jump. JumpL. When you select Jump and JumpL, you need to fill in LH, MH, and RH values.



3. Select the tool/workobject number for motion. If the tool/workobject number is displayed in the point list, the default value is the tool/workobject number identified by the point list; otherwise, it is the currently activated tool/workobject number in the system. If the tool/workobject number used does not match the system tool/workobject number, a red message will appear below: "The tool/workobject number used does not match the system tool/workobject number!"
4. Click and hold the "Run" button and observe the movement to the target point. Release the button to stop the motion.

When the motion is complete, the following information is displayed. Then, you can release the "Run" button.



NOTE

- When you select Movl or Jump motion, if the starting/end point is in a singular position, or if there is a singular position in the process points, it will cause motion failure.
- The Jump and JumpL instructions are supported only by SCARA robots.
- If the tool/workobject number used does not match the system tool/workobject number, a red message will appear.

6.2.4 Using the Point Variables

Global points

The global point variable is named as "P[*]", for example, "P[1]".

- Method 1: Direct use

1. Open the point editing page and edit the point data, as shown below.

P.pts

	Point		Position info						Arm parameters
	ID	Labels	X	Y	Z	A	B	C	[1]
	0	pStart	417.200	-4.574	737.969	-170.866	-56.482	-10.925	
	1	p001	177.342	41.138	873.792	-161.339	-7.690	-5.416	
	2	pEnd	119.209	26.428	886.241	-137.516	-7.951	-8.758	
	3		127.285	22.353	880.275	177.366	-15.055	20.542	

2. Use the point data in the program file, as shown below.

main.pro

```
1   Include "sub1.pro";                    # Include program file: sub1.pro
2   Start;                                 # Task entry command
3       Print "MAIN Start";                # Print "MAIN Start"
4       Int intA = 0;                      # Define local variable intA
5       While intA < 3                     # Execute loop if intA < 3
6           intA = intA+1;                 # Perform local variable calculation
7           B[0] = intA;                   # Variable assignment
8           sub1.funcPrint(intA);          # Call function from other program files
9           Movj pStart,V[30],Z[0],Tool[0];  # Movj move to pstart
10          Movj p001,V[30],Z[0],Tool[0];    # Movj move to p001
11          Movj pEnd,V[30],Z[0],Tool[0];    # Movj move to pEnd
12      EndWhile;
13      Print "Main End";                  # Print "MAIN End"
14  End;
```

- Method 2: Use the labels

1. Edit the point labels on the point data editing page.

P.pts

	Point		Position info						Arm parameters
	ID	Labels	X	Y	Z	A	B	C	[1]
	0	pStart	417.200	-4.574	737.969	-170.866	-56.482	-10.925	
	1	p001	177.342	41.138	873.792	-161.339	-7.690	-5.416	
	2	pEnd	119.209	26.428	886.241	-137.516	-7.951	-8.758	
	3		127.285	22.353	880.275	177.366	-15.055	20.542	

2. Use the labels in the program file, as shown below.

main.pro

```
1   Include "sub1.pro";                    # Include program file: sub1.pro
2   Start;                                 # Task entry command
3       Print "MAIN Start";                # Print "MAIN Start"
4       Int intA = 0;                      # Define local variable intA
5       While intA < 3                     # Execute loop if intA < 3
6           intA = intA+1;                 # Perform local variable calculation
7           B[0] = intA;                   # Variable assignment
8           sub1.funcPrint(intA);          # Call function from other program files
9           Movj pStart,V[30],Z[0],Tool[0];  # Movj move to pstart
10          Movj p001,V[30],Z[0],Tool[0];    # Movj move to p001
11          Movj pEnd,V[30],Z[0],Tool[0];    # Movj move to pEnd
12      EndWhile;
13      Print "Main End";                  # Print "MAIN End"
14  End;
```

NOTE

- The label must start with a letter and can contain only letters, numbers, and underscores, with a max length of 20 characters.
- The label name must be unique.

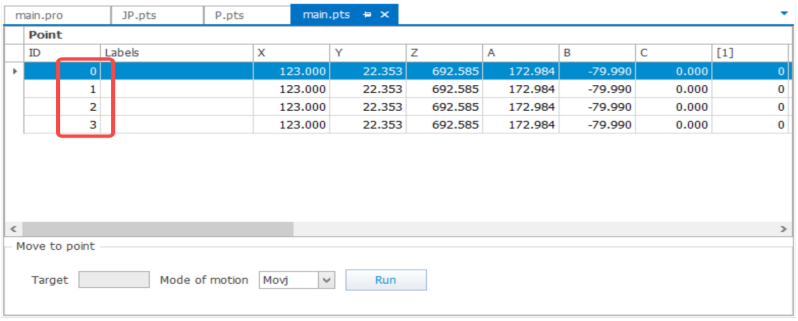
Local points

The local point variable is named as "LP[*]", for example, "LP[1]".

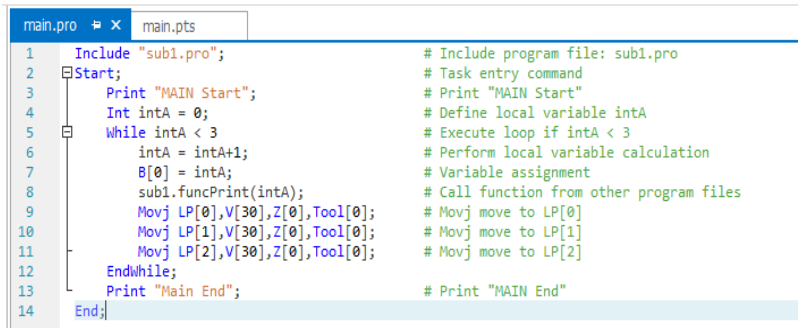
The following describes the operation of local point variables in the program file in two methods.

- Method 1: Direct use

1. Open the point editing page and edit the point data, as shown below.

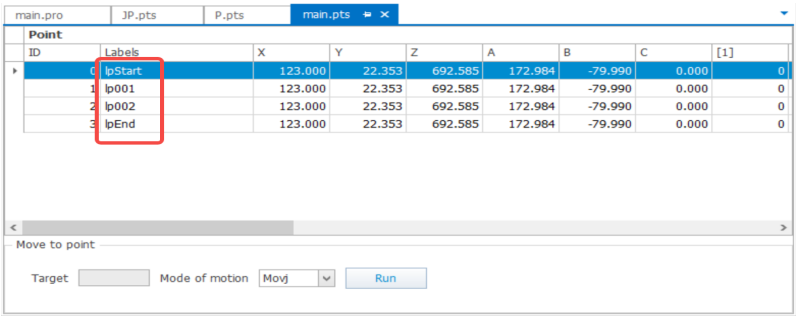


2. Use the point data in the program file, as shown below.

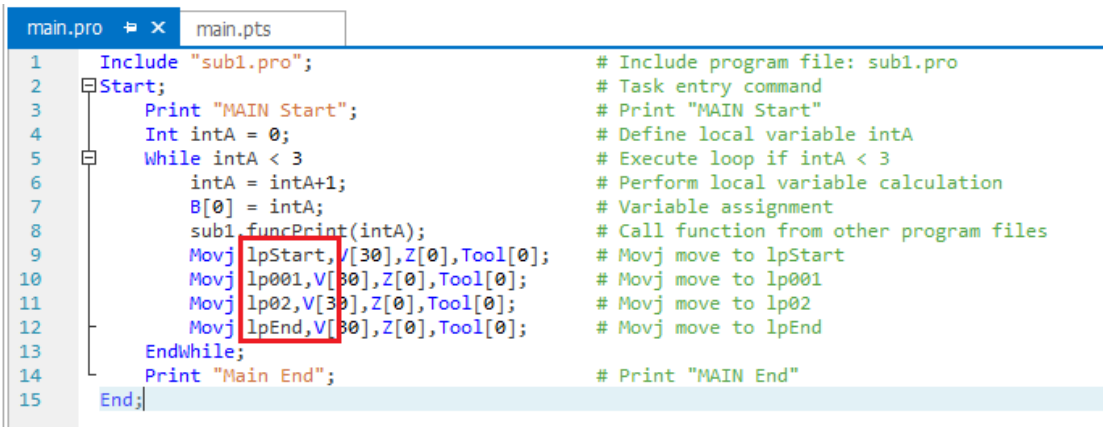


- Method 2: Use the labels

1. Edit the point labels on the point data editing page.



2. Use the labels in the program file, as shown below.



NOTE

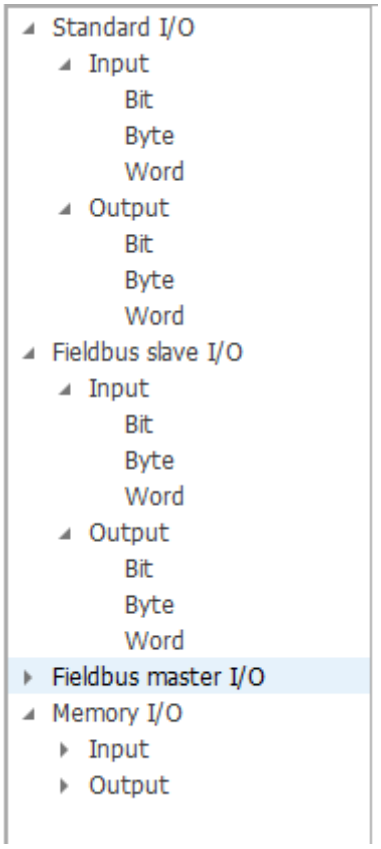
- The label must start with a letter and can contain only letters, numbers, and underscores, with a max length of 20 characters.
- The label name must be unique.

6.3 Labels Editing

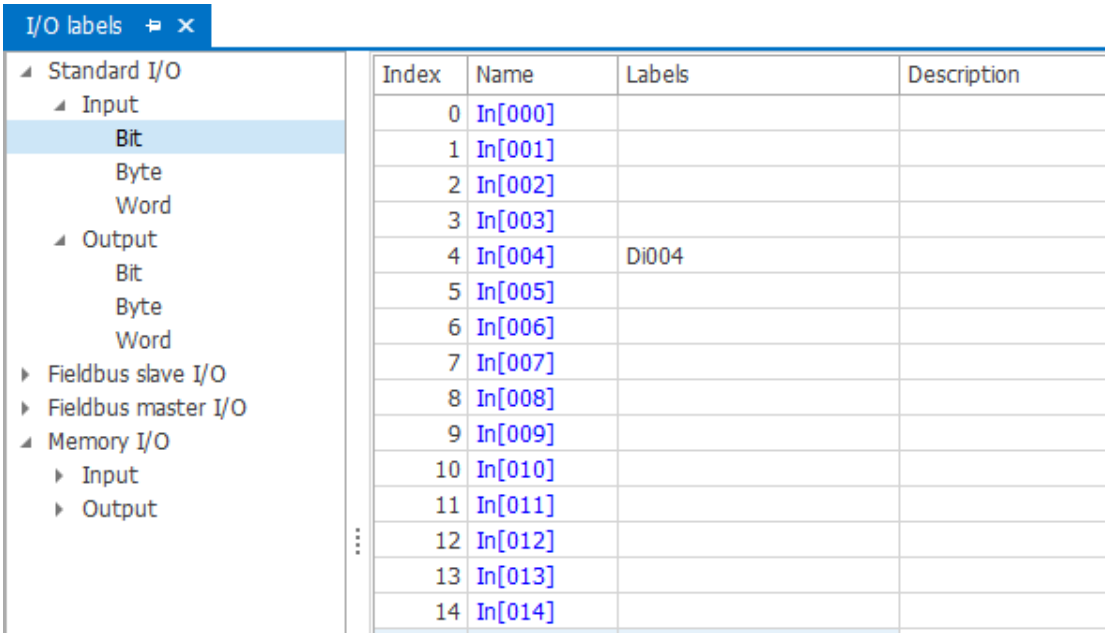
The label is an alias of a variable. It improves the readability of the program and reduces the memory load. Currently, the labels include I/O labels, AD labels, BRD labels, global position variable labels, and local position variable labels. Their settings and usage are basically similar.

6.3.1 I/O Labels

The I/O labels include standard I/O, fieldbus I/O, and memory I/O, which are further divided into input and output. The input and output are further divided into bit, byte and word, as shown in the figure below.



1. Double-click the I/O label to open the I/O editing page.
2. Select the input or output to be labeled and enter a label name. The label name starts with a letter and can contain only letters, numbers, and underscores, with a max length of 20 characters.



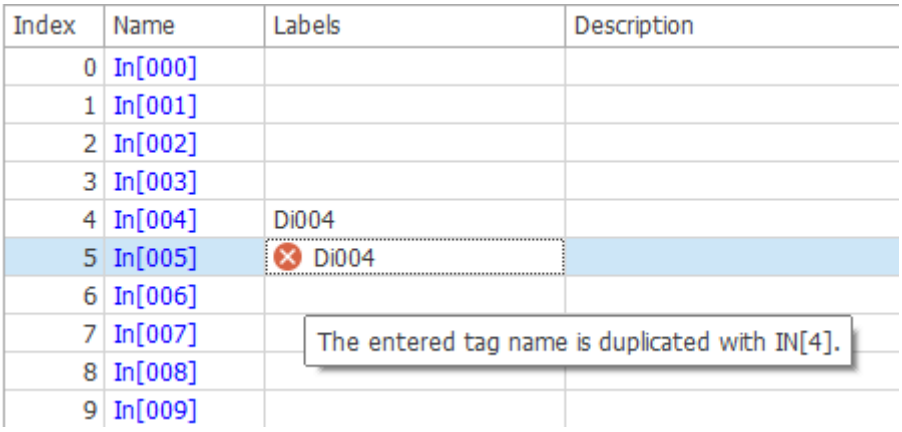
The label can be used in the program file, for example:

Wait di004 == ON, T[3], GOTO L[1];

Using "di004" in a statement has the same effect as using "IN[4]" directly.

NOTE

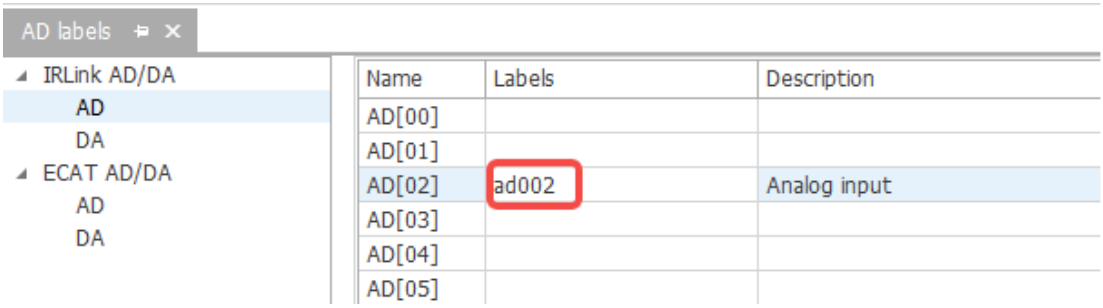
The label name must be unique, otherwise the following prompt will appear:



6.3.2 AD Labels

The AD labels includes analog input (AD) and analog output (DA).

1. Double-click the AD label to open the AD editing page.
2. Select the input or output to be labeled and enter a label name. The label name starts with a letter and can contain only letters, numbers, and underscores, with a max length of 20 characters.



3. The labels can be used in the program file, for example:

Get ad002,D1; ## Assign the value of AD[2] to D1.

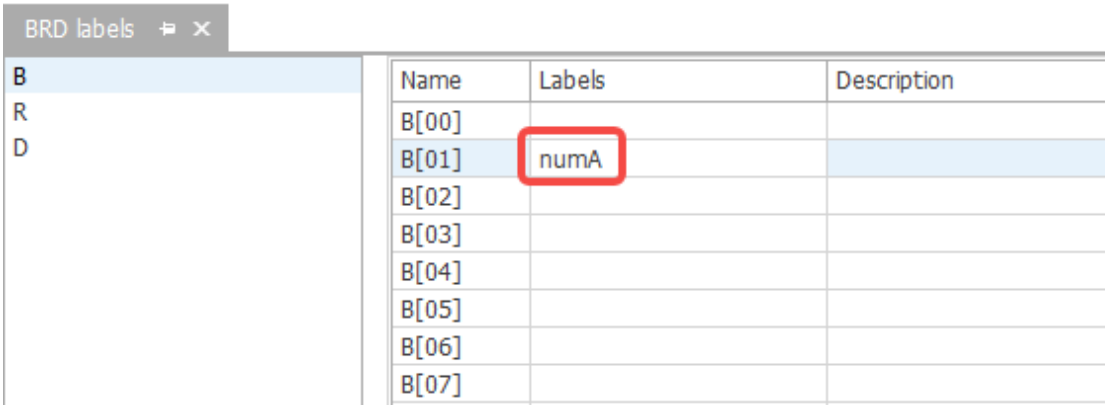
Using "ad002" in a statement has the same effect as using "AD[2]" directly.

> NOTE
>
> The label name must be unique.

6.3.3 BRD Labels

The BRD labels include B, R, and D variables.

1. Double-click the BRD label to open the BRD editing page.
2. Select the input or output to be labeled and enter a label name. The label name starts with a letter and can contain only letters, numbers, and underscores, with a max length of 20 characters.



The labels can be used in the program file, for example:

numA = 9; ## Assign 9 to B[1].

Using "numA" in a statement has the same effect as using "B[1]" directly.

> NOTE
>
> The label name must be unique.

7 Debugging

7.1 Debugging Mode

The debugging mode is mainly used for debugging the robot program. During debugging, you can monitor local variable values. Breakpoints and statement-by-statement debugging are supported.

Before entering the debugging mode, make sure you have synchronized the project with the controller.

You can enter the debugging mode in any of the following ways.

- Return to the start line
- Set the start line
- Start
- Step-into

After entering the debugging mode, the corresponding information is displayed in the status bar. For details, see the 1.4.6 Status Bar section.

You can exit the debugging mode in any of the following ways.

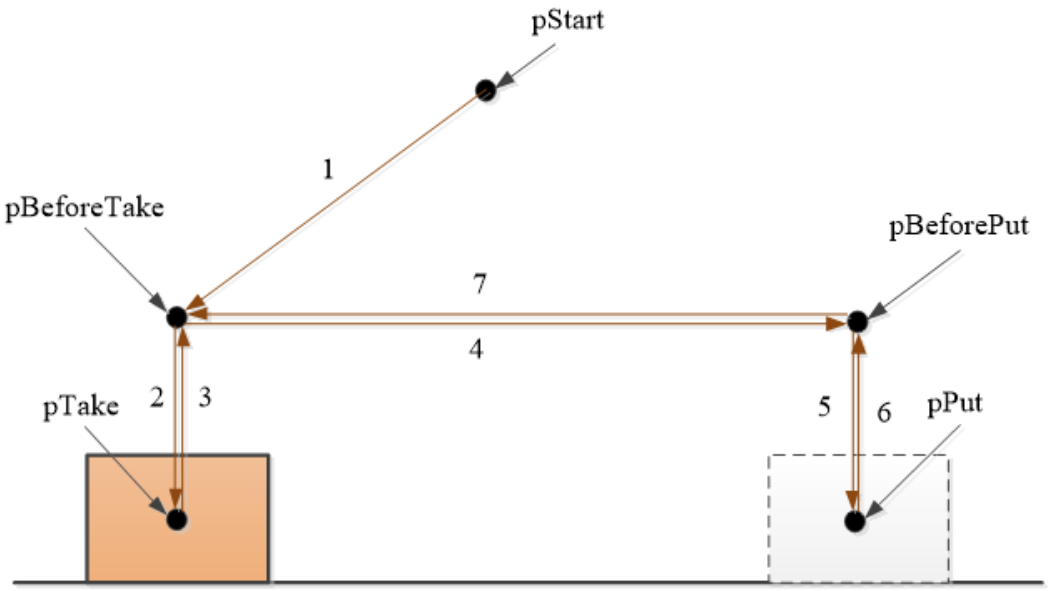
- Go to "Debug" > "Exit".
- Switch to the auto mode.
- Disconnect the controller.

7.2 Basic Debugging

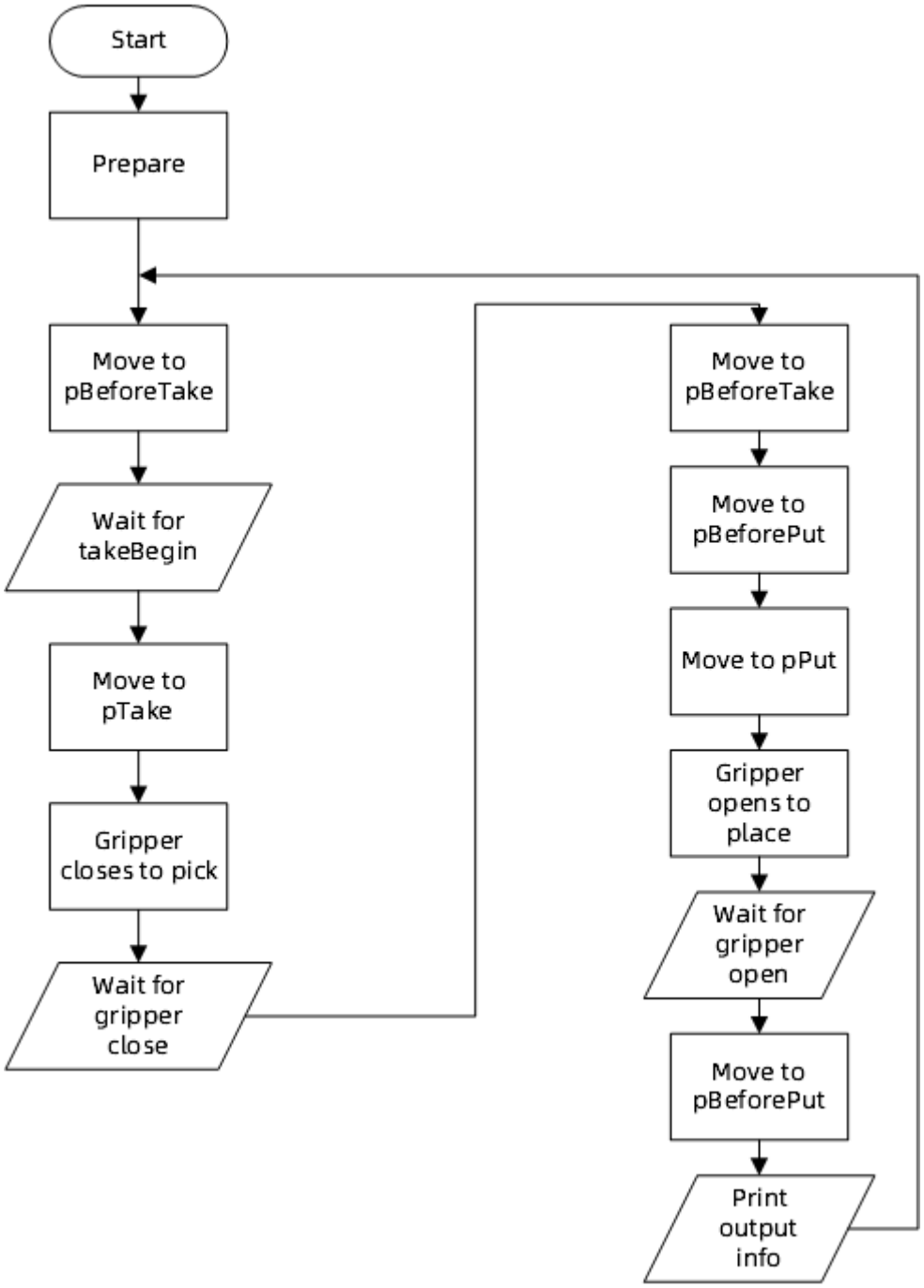
7.2.1 Project Example

Here we take a typical pick and place application an example.

Illustration



Flowchart



The following shows the contents of "main.pro" program file.

```
Start;
   # Preparation
   Movj pStart,V[30],Z[0],Tool[0];                    # Move to the Start point
   Int count = 0;                                      # Number of places
   Set GasClaw,OFF;                                    # Gripper open
   Wait GasClaw Open == ON, T[0];                      # Wait for the gripper to open
   # Start the work cycle
L[1]:                                                  # Start of the place cycle
Movl pBeforeTake,V[30],Z[0],Tool[0];                   # Move to the BeforeTake point
WaitInPos;
   Wait takeBegin == ON, T[0];                         # Wait until the pick start signal
Movl pTake,V[30],Z[0],Tool[0];                         # Move to the Take point
WaitInPos;
   Set GasClaw,ON;                                     # Gripper picks material and closes
   Wait GasClawClose == ON, T[0];                      # Wait for the gripper to close
   Movl pBeforePut,V[30],Z[0],Tool[0];                 # Move to the BeforePut point
Movl pPut,V[30],Z[0],Tool[0];                          # Move to the Put point
WaitInPos;
   Set GasClaw,OFF;                                    # Gripper opens to place material
   Wait GasClawOpen == ON, T[0];                       # Wait for the gripper to open
   Print "Number of places";                           # Print the number of places
   count = count + 1;
   Print count;
   Movl pBeforePut,V[30],Z[0],Tool[0];                 # Return to the BeforePut point
   Goto L[1];                                          # Return to the start of the next cycle
End;
```

Label settings are shown in the table below:

Variable name or signal name	Label name	Description
P[0]	pStart	Start point
P[1]	pBeforeTake	Pick ready point
P[2]	pTake	Pick point
P[3]	pBeforePut	Place ready point
P[4]	pPut	Place point

Variable name or signal name	Label name	Description
IN[5]	takeBegin	Pick start signal
IN[6]	GasClawClose	Gripper close signal
IN[7]	GasClawOpen	Gripper open signal
OUT[5]	GasClaw	Gripper control information

7.2.2 Starting and Stopping Debugging

After the debugging process is started, the program runs continuously in the debugging mode. You can operate the robot to move or stop according to the program running conditions.

Prerequisite

1. Editing the program file and synchronize the project to the controller.
2. Debug the program currently running in the robot controller and synchronize the project to local.
3. Check that the robot controller does not have alarms or warnings, and enable the robot servo.
4. After the project synchronization is completed, you can return all the project tasks to the start line before the robot starts motion for the first time to prevent the robot from unknown operating conditions.

Start

Click the "Start" button in the "Debug" menu to start debugging.

Stop

Click the "Stop" button in the "Debug" menu to stop debugging.

1. When the program is running in debugging mode, the blue icon (i.e., running line flag), yellow icon (i.e., start line flag), and editing cursor are continuously refreshed. You can check the running logic of the program based on these two flags, and determine whether the running logic is correct in combination with the robot actions. When debugging is stopped, these two flags stay on a line in the program file.

```
Start;
    # Preparation
    Movj pStart,V[30],Z[0],Tool[0];          # Move to the Start point
    Int count = 0;                           # Number of places
    Set GasClaw,OFF;                         # Gripper open
    Wait GasClaw Open == ON, T[0];           # Wait for the gripper to open
    # Start the work cycle
    L[1]:
    Movl pBeforeTake,V[30],Z[0],Tool[0];     # Move to the BeforeTake point # Start of the place cycle
    WaitInPos;
    Wait takeBegin == ON, T[0];              # Wait until the pick start signal
    Movl pTake,V[30],Z[0],Tool[0];           # Move to the Take point
    WaitInPos;
    Set GasClaw,ON;                          # Gripper picks material and closes
    Wait GasClawClose ==ON, T[0]             # Wait for the gripper to close
    Movl pBeforePut,V[30],Z[0],Tool[0];      # Move to the BeforePut point
    Movl pPut,V[30],Z[0],Tool[0];            # Move to the Put point
    WaitInPos;
    Set GasClaw,OFF;                         # Gripper opens to place material
    Wait GasClawOpen == ON, T[0];            # Wait for the gripper to open
    Print "Number of places"                 # Print the number of places
    count = count + 1;
    Print count;
    Movl pBeforePut,V[30],Z[0],Tool[0];      # Return to the BeforePut point
    Goto L[1];                               # Return to the start of the next cycle
End;
```

2. When the program is running in debugging mode, you can see the state information "[Debug: Run]" in the lower left corner.
3. After you click "Stop", the program debugging is stopped. You can see the state information "[Debug: Stop]" in the lower left corner.
4. When the program runs to the end, the state information "[Debug: End]" is displayed.

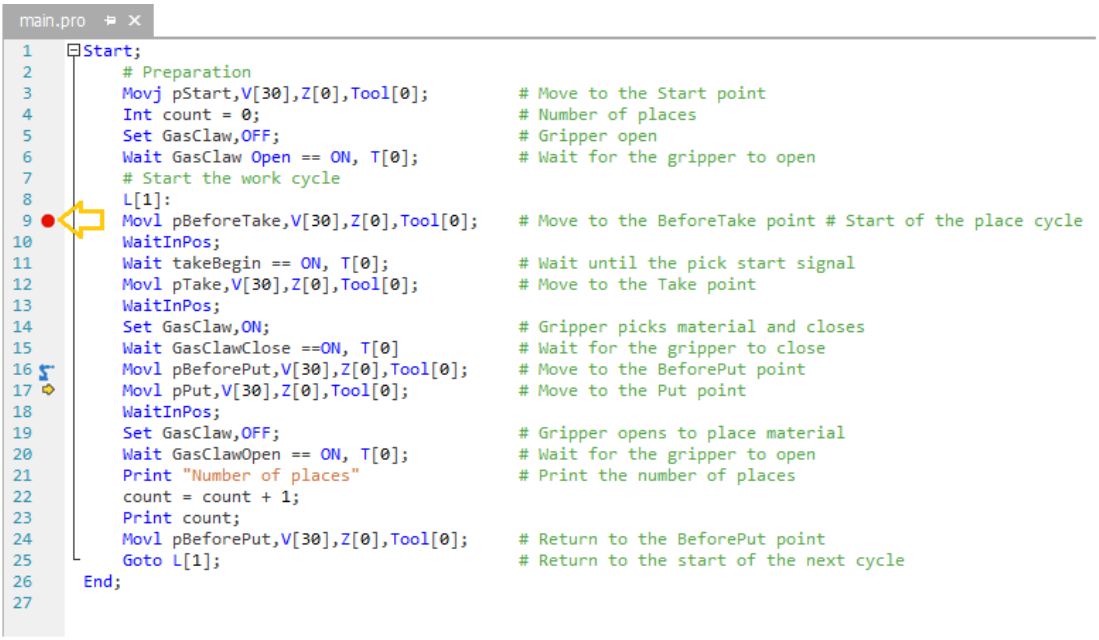
7.2.3 Adding and Deleting Breakpoints

Set the selected line as a breakpoint or delete the breakpoint to restore the normal state.

If a line has a breakpoint, the breakpoint icon is displayed on the left edge of the program window.

Adding a breakpoint (F9)

Click on right side of the program line number to add a breakpoint, as shown below, or press F9 to add or delete the breakpoint after selecting the program file line with the mouse.



Deleting a breakpoint (F9)

Click the added breakpoint again to delete the breakpoint.

> **NOTE**
>
> - The breakpoint cannot be operated when the program is in the debugging or auto mode.
> - The breakpoint is valid only in the main task and dynamic task during debugging.
> - Breakpoint cannot be added to a line outside the function body.
> - The max number of breakpoints is 100.
> - For the looping program section, breakpoints added after the yellow cursor will take effect in the next loop.

Breakpoints cannot be set to some instructions, as shown in the following table.

No.	Instructions	No.	Instructions
1	ENDIF	6	Curve
2	EDNFUNC	7	ENDFOR
3	ENDSWITCH	8	FUNC
4	ENDWHILE	9	EndCurve
5	START	10	END

CAUTION

Breakpoints cannot be set for comments.

7.2.4 Enabling, Disabling, and Deleting All Breakpoints

- Enabling all breakpoints means that the breakpoints are activated, allowing the program to enter the breakpoints during debugging.
- Disabling all breakpoints means that the program does not enter the breakpoints during debugging without deleting all breakpoints.
- Deleting all breakpoints means deleting all breakpoints in the current program file.

Enabling and disabling all breakpoints (Alt+F9)

- When the breakpoints are disabled , click the "Enable all" button in the "Debug" menu. The following figure shows the enabled breakpoints.

main.pro

```
1   Start;
2       # Preparation
3       Movj pStart,V[30],Z[0],Tool[0];          # Move to the Start point
4       Int count = 0;                           # Number of places
5       Set GasClaw,OFF;                         # Gripper open
6       Wait GasClaw Open == ON, T[0];           # Wait for the gripper to open
7       # Start the work cycle
8       L[1]:
9       Movl pBeforeTake,V[30],Z[0],Tool[0];     # Move to the BeforeTake point # Start of the place cycle
10      WaitInPos;
11      Wait takeBegin == ON, T[0];              # Wait until the pick start signal
12      Movl pTake,V[30],Z[0],Tool[0];           # Move to the Take point
13      WaitInPos;
14      Set GasClaw,ON;                          # Gripper picks material and closes
15      Wait GasClawClose ==ON, T[0]             # Wait for the gripper to close
16      Movl pBeforePut,V[30],Z[0],Tool[0];      # Move to the BeforePut point
17      Movl pPut,V[30],Z[0],Tool[0];            # Move to the Put point
18      WaitInPos;
19      Set GasClaw,OFF;                         # Gripper opens to place material
20      Wait GasClawOpen == ON, T[0];            # Wait for the gripper to open
21      Print "Number of places"                 # Print the number of places
22      count = count + 1;
23      Print count;
24      Movl pBeforePut,V[30],Z[0],Tool[0];      # Return to the BeforePut point
25      Goto L[1];                               # Return to the start of the next cycle
26  End;
27
```

- When the breakpoints are enabled , click the "Disable all" button in the "Debug" menu. The following figure shows the disabled breakpoints.

main.pro

```
Start;
    # Preparation
    Movj pStart,V[30],Z[0],Tool[0];          # Move to the Start point
    Int count = 0;                           # Number of places
    Set GasClaw,OFF;                         # Gripper open
    Wait GasClaw Open == ON, T[0];           # Wait for the gripper to open
    # Start the work cycle
    L[1]:
    Movl pBeforeTake,V[30],Z[0],Tool[0];     # Move to the BeforeTake point # Start of the place cycle
    WaitInPos;
    Wait takeBegin == ON, T[0];              # Wait until the pick start signal
    Movl pTake,V[30],Z[0],Tool[0];           # Move to the Take point
    WaitInPos;
    Set GasClaw,ON;                          # Gripper picks material and closes
    Wait GasClawClose ==ON, T[0]             # Wait for the gripper to close
    Movl pBeforePut,V[30],Z[0],Tool[0];      # Move to the BeforePut point
    Movl pPut,V[30],Z[0],Tool[0];            # Move to the Put point
    WaitInPos;
    Set GasClaw,OFF;                         # Gripper opens to place material
    Wait GasClawOpen == ON, T[0];            # Wait for the gripper to open
    Print "Number of places"                 # Print the number of places
    count = count + 1;
    Print count;
    Movl pBeforePut,V[30],Z[0],Tool[0];      # Return to the BeforePut point
    Goto L[1];                               # Return to the start of the next cycle
End;
```

Deleting all breakpoints (Ctrl+Shift+F9)

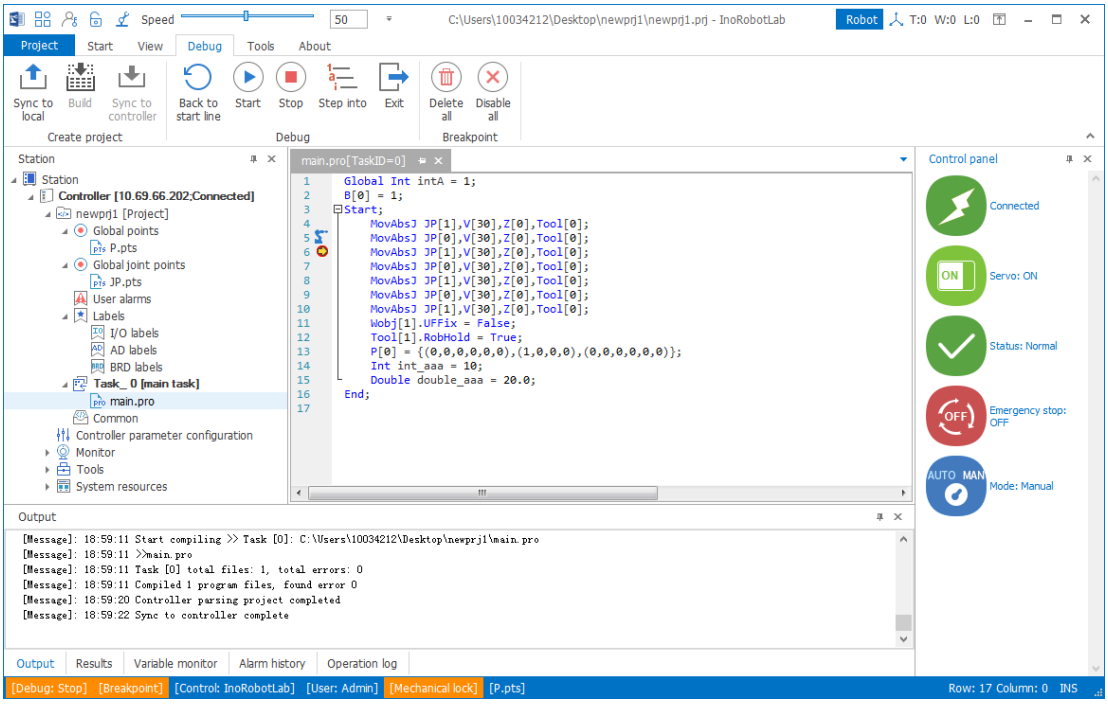
Click the "Delete all" button in the "Debug" menu to delete all breakpoints in the project file. Deleting breakpoints is independent of whether the breakpoints are enabled or not.

CAUTION

- The action of enabling, disabling and deleting all breakpoints takes effect on the entire project file.
- It is not allowed to add, delete, enable or disable breakpoints during debugging or auto mode.

7.2.5 Entering a Breakpoint

In the debugging mode, the breakpoint is enabled in the main task and dynamic task. The corresponding program file is opened when the program runs to the breakpoint line. The status of the current breakpoint is displayed in the status bar.



7.2.6 Running the Program Statement by Statement

It allows you to run the program statement by statement so that you can monitor the value of variables and observe the program running state. For variable monitor, see the 7.8.3 Viewing the Monitoring Variables section.

CAUTION

- You can stop the debugging by clicking the stop button, entering the breakpoint, returning to the start line, or setting the start line. When the debugging is stopped, the state information "[Debug: End]" is displayed in the lower left status bar.
- Click the "Step into" button in the "Debug" menu, the program runs by one line of statement.
- After the single-step operation is completed, you can click the button again to run the next line of statement.
- Shortcut: F11.

7.2.7 Adding a Monitoring Variable

To view the value of the current variable, you can add the variable to the monitoring list. When in the debugging state and the debugging state is "[Debug: Stop]" or "[Debug: End]", the values in the monitoring list are automatically refreshed.

Prerequisite

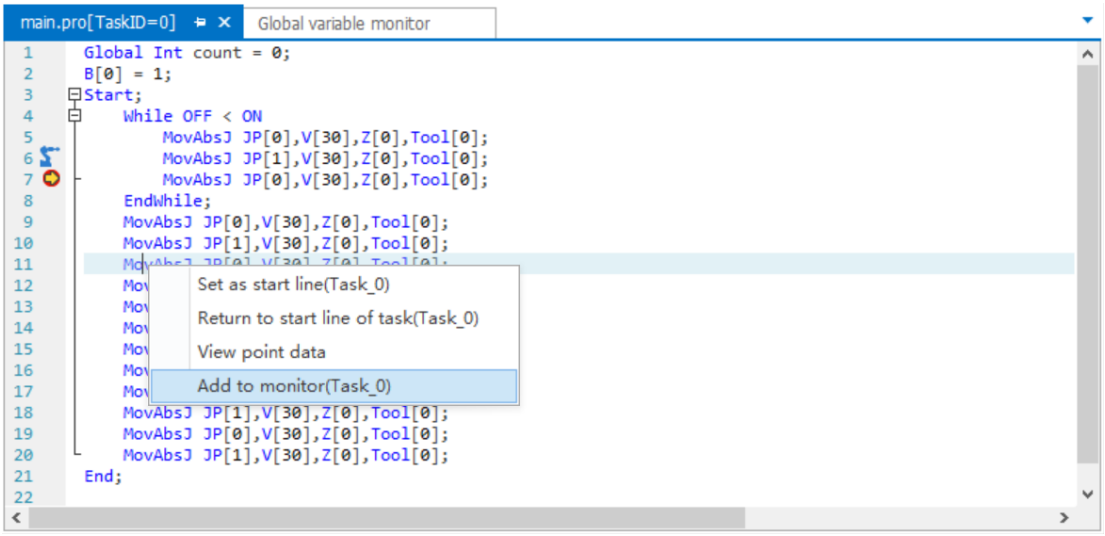
Ensure that the project is synchronized and that no errors are returned in compiling and parsing.

Steps

You can add monitoring variables in either of the following two methods.

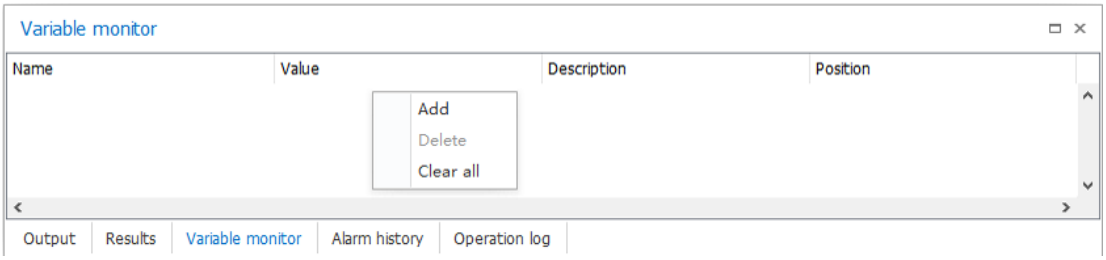
- Method 1: Adding a variable in the open program file

In the program file, right-click on the variable and select "Add to monitor(Task_0)".

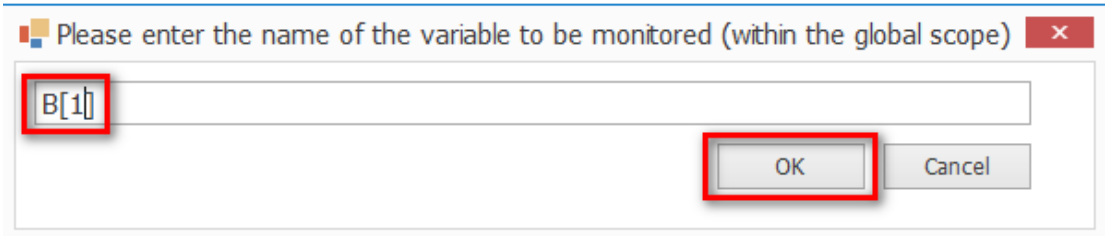


- Method 2: Adding a variable in the variable monitoring list

1. In the "Variable monitor" pane, right-click and select "Add".

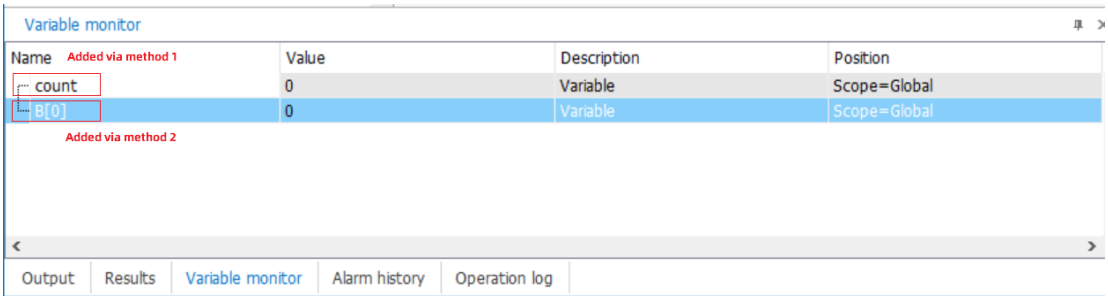


2. Enter the variable to be monitored in the input box and click "OK".



Results

The added variables appear in the variable monitoring list, as shown below.



NOTE

- When Method 2 is adopted, only variables within the global scope can be added, such as B, R, and R. D, P, PR, IN, OUT, IG, OG, TOOL, DA, AD, and user-defined global variables (with the keyword "Global").
- Variables within the global scope can also be added and viewed in "Monitor" > "Global variable monitor".
- When Method 2 is adopted, for example, when you choose to add "B[0]" variable with mouse click, if you select the left square bracket "[", an array will be added, such as "B[0:49]". In this case, the B variable is continuously monitored from B[0] to B[49], which is the monitoring of an array variable.

7.2.8 Modifying the Monitoring Variables

Sometimes, the added variables do not need to be monitored. This function allows you to modify the subscript of a single variable in an array variable and the subscript range of the array variable.

Prerequisite

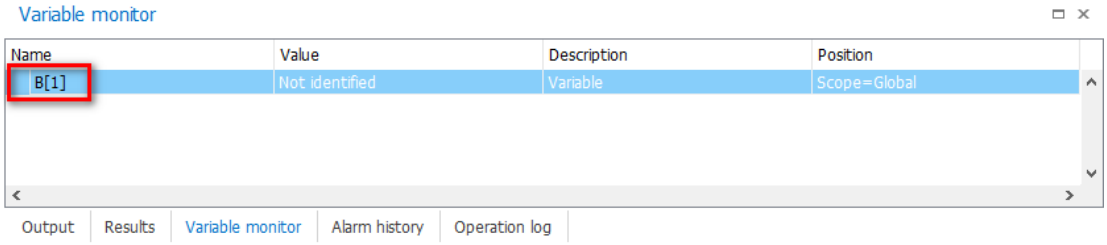
An individual variable in an array or an array variable with a range exists in the monitoring list. For specific addition methods, see the 7.7.2 Adding a Monitoring Variable section.

Steps

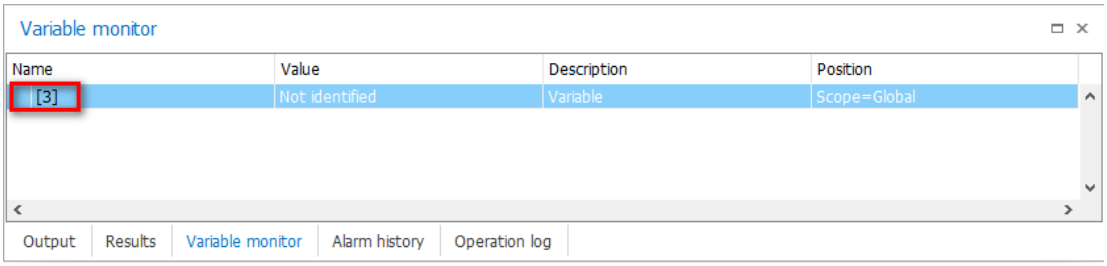
You can modify the variable in either of the following two methods.

- Method 1: Modifying the subscript of a single variable in an array

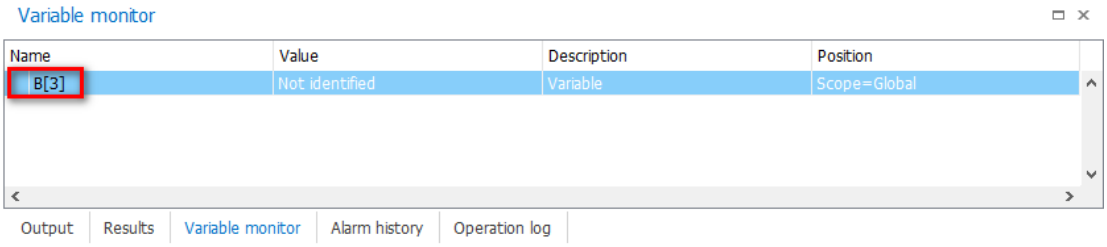
1. Click the name of a single variable to enter the editing state of the cell.



2. Enter the subscript value. For example, see "[3]" in the following figure.

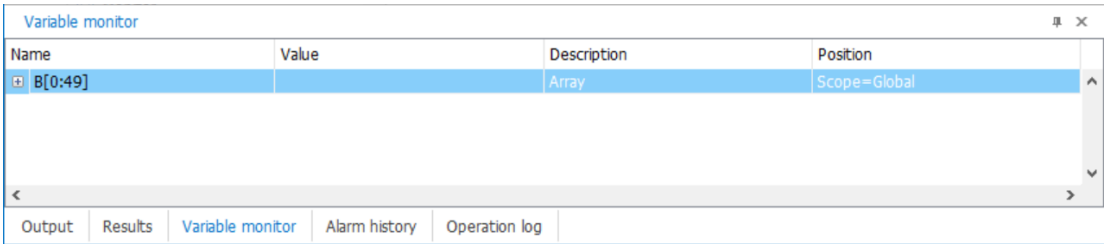


3. After the editing, the subscript of the variable is modified.

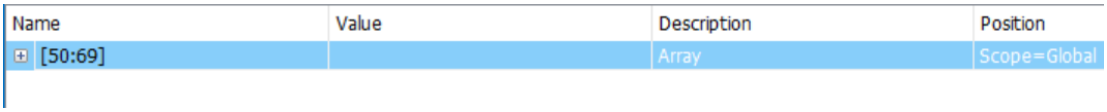


- Method 2: Modifying the subscript range of an array variable

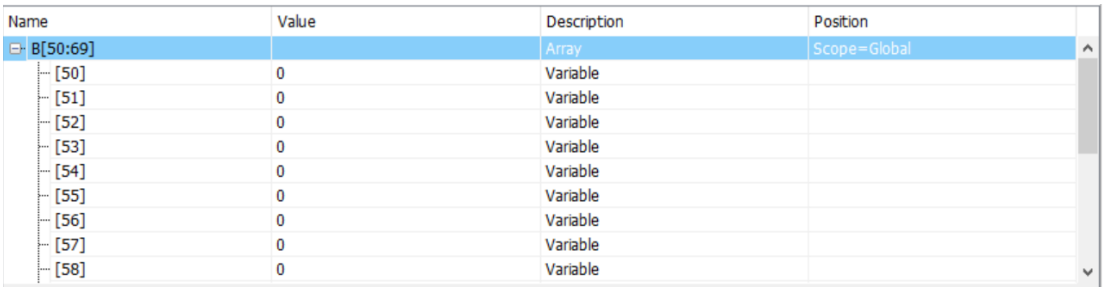
1. In the variable monitoring list, click the name of an array variable to make it editable.



2. Enter the subscript range of the array variable.



3. After editing, click "+" to the left of the variable to expand the node. You can see that the range of monitoring of the array variable has been modified.



Results

The subscript or subscript range is modified.

> **NOTE**
>
> You can monitor up to 50 variables for an array variable.

7.2.9 Viewing the Monitoring Variables

You can view the monitored variables from the monitoring list to better analyze the running logic and execution structure of the program. The monitoring list adopts a tree structure. Click "+" on the left side of the monitoring list to expand the node to show the members or items of the next level.

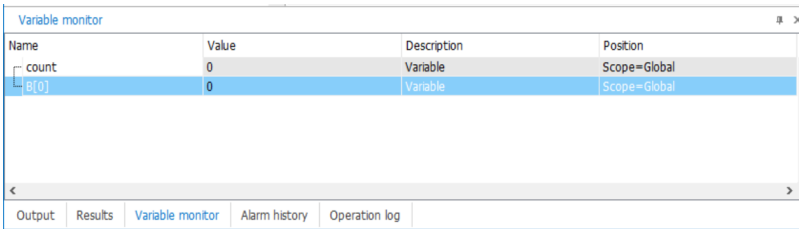
Prerequisite

1. The value of a variable in the monitoring list must be viewed when the value is effectively refreshed.
2. The variable value is refreshed only in debugging mode and in "[Debug: Stop]" state.

Steps

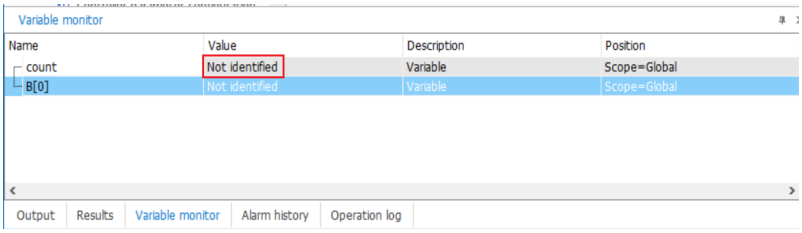
- Scenario 1: When the variable value is identified and not refreshed

When the "Value", "Description", and "Location" columns are gray in the monitoring list, the monitoring list is not refreshed. When not in debugging mode, or when the debug is being performed, these columns are gray.



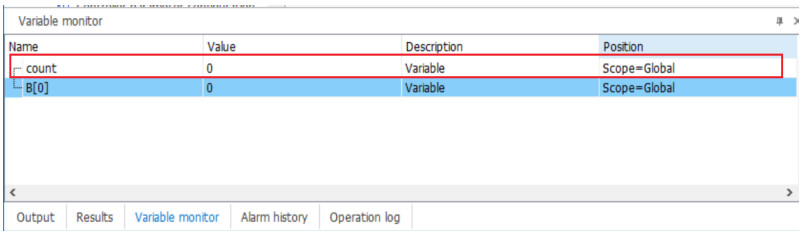
- Scenario 2: When the variable is not identified

When the variable value is displayed as "Not identified" and the Scenario 1 applies, the current program running line is out of the scope. Therefore, the value cannot be retrieved from the controller.



- Scenario 3: When the variable is identified and the value is refreshed

When not in Scenario 1 or Scenario 2, the variable is refreshed and the value viewed is valid.



NOTE

Local variables can be identified only when debugging is within the corresponding scope.

7.2.10 Returning to the Start Line

It allows you to return a task to the start line of its entry program file. You can return a single task to the start line or all tasks to the start line. This operation is valid only for the main task and dynamic task.

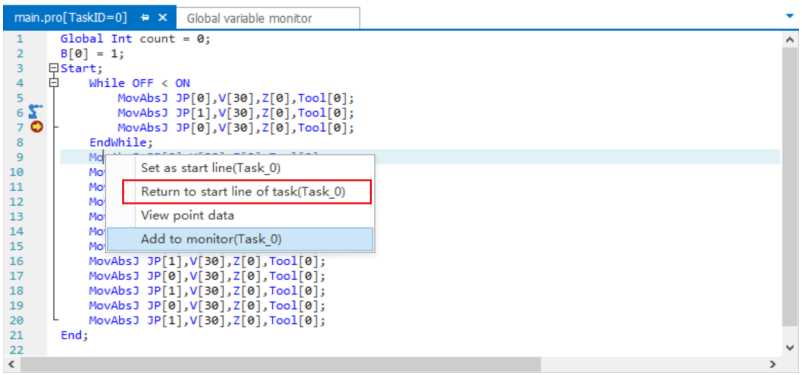
Prerequisite

Make sure that the project is synchronized, is currently in the editing state, or the debugging state is "[Debug: Stop]" or "[Debug: End]".

Steps

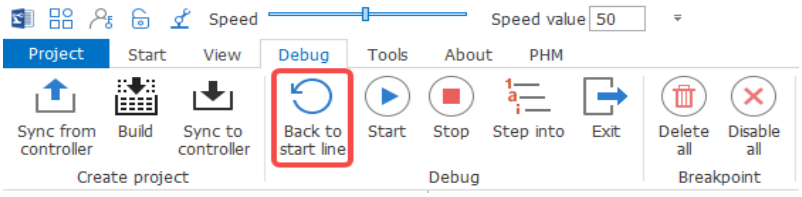
- Method 1: Returning a single task to the start line

Right-click on the program file and select "Return to start line of task(Task_0)".



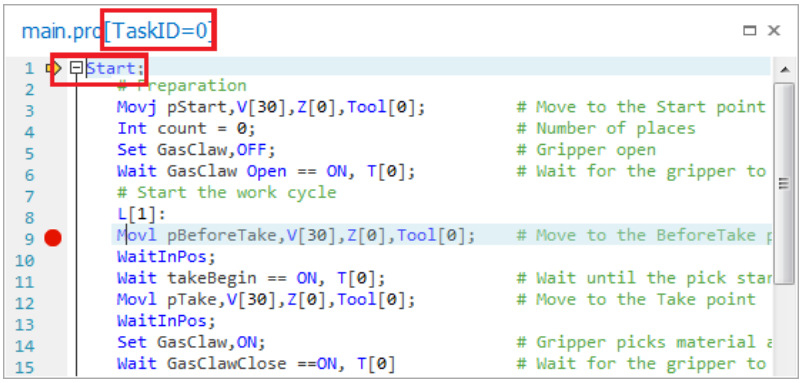
- Mode 2: Returning the main task and all dynamic tasks to the start line

Click "Back to start line" in the "Debug" menu.



- Results

The start line of the corresponding task returns to the Start of the entry program file of the task.



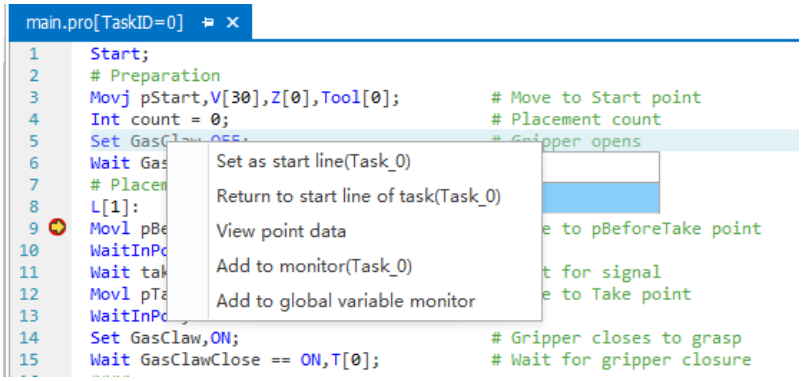
> **NOTE**
>
> In auto mode, if the main task and dynamic task are in the stop or end state, you can also return the task to the start line.

7.2.11 Setting the Start Line

You can set a line in the program file as the start line. When you click "Start", the task starts from this line.

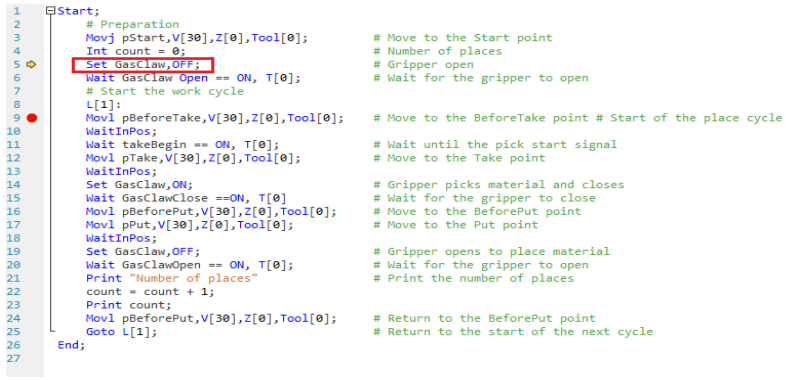
Prerequisite

1. Make sure that the project is synchronized and the debugging state is "[Debug: Stop]" or "[Debug: End]".
2. Right-click a line in the program file and select "Set as start line".



Results

The start line is switched to the set line.



> **NOTE**
>
> If you set the start line of a program in a function and then start debugging, the program file executes the logic according to the command in the function. In this case, the program file does not execute from the entry function of the task (Start…End function of the entry program file).

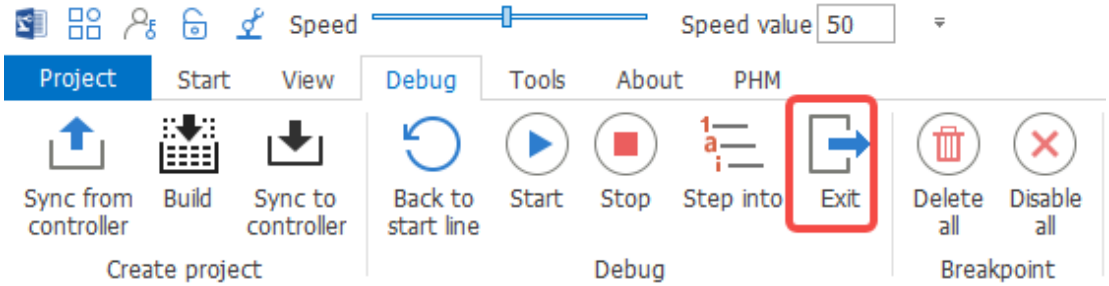
7.2.12 Exiting Debugging

After manually exiting the debugging mode, you can edit the project again.

Prerequisite

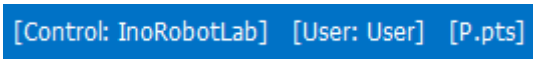
The program is currently in debugging state.

Click the "Exit" button in the "Debug" menu.



Results

The debug state disappears from the status bar in the lower left corner.



NOTE

After exiting the debugging mode, you can edit the project.

7.3 Multitasking Debugging

The types of robot task include main task, static task, dynamic task, and xqt task.

Type	Main task	Static task	Dynamic task	xqt task
Task type	Main.pro, which is always active and is used to perform robot motion. It is controlled by Start and Stop instructions.	When the controller is powered on and waits for system initialization and project loading to complete, activated static tasks in the project are started and run continuously until the stop conditions are met.	The dynamic tasks are not started immediately upon power-on of the robot and are controlled by Start and Stop instructions like the main task.	The xqt task is controlled separately by the instructions Xqt, Halt, and Resume.
Number of tasks	1	14, where Tasks 1 to 14 can be configured as static tasks	14, where Tasks 1 to 14 can be configured as dynamic tasks	1

<table>
<tr><th>Type</th><th>Main task</th><th>Static task</th><th>Dynamic task</th><th>xqt task</th></tr>
<tr><td rowspan="2">Activation of task</td><td>1. Automatically activated after the project is synchronized to the controller.
2. Activated and deactivated through the task management interface.</td><td>1. Automatically activated after the project is synchronized to the controller.
2. Activated and deactivated through the task management interface.</td><td>1. Automatically activated after the project is synchronized to the controller.
2. Activated and deactivated through the task management interface.</td><td>1. Activated by the xqt instruction and deactivated by the quit instruction.
2. The activation of xqt task is canceled when:
① The project is resynchronized to the controller.
② When any task returns to the start line.
③ When switching to other control authority.</td></tr>
<tr><td colspan="4">NOTE
There is no coupling between the activation and inactivation of each task.</td></tr>
</table>

<table>
<tr><th>Type</th><th>Main task</th><th>Static task</th><th>Dynamic task</th><th>xqt task</th></tr>
<tr><td rowspan="2">Start/Stop</td><td>1. Started via the "Start" button on the main interface.
2. Started via the task management interface.</td><td>1. Started automatically after activation.
2. If a static task is configured before restart, it runs automatically after restart.
3. Started via the task management interface.</td><td>When the main task is activated:
Started when the main task starts.
When the main task is not activated:
1. Started via the "Start" button on the main interface.
2. Started via the task management interface.</td><td>1. Activated and started by the xqt instruction.
2. Stopped by the halt instruction and resumed by the resume instruction.
3. The stop is triggered by the halt instruction or "Stop" button after the xqt task is activated. The xqt task starts when the main task starts again.</td></tr>
<tr><td>1. Stopped via the "Stop" button on the main interface.
2. Stopped upon alarm.</td><td>1. Stopped via the "Stop" button on the task management interface.
2. Stopped when an error occurs on the corresponding static task channel.
3. Stopped automatically when the main task is not running and the static task is deactivated.


The static task starts again when re-activated.</td><td>When the main task is activated:
Stopped when the main task stops.
When the main task is not activated:
Stopped via the "Stop" button on the task management interface when the dynamic task is started.</td><td>1. Stopped when the main task stops.
2. Stopped by the halt instruction.</td></tr>
</table>

<table>
<tr><th>Type</th><th>Main task</th><th>Static task</th><th>Dynamic task</th><th>xqt task</th></tr>
<tr><td></td><td colspan="4">NOTE
• The task must be active to be subject to start/stop control.
• Static tasks are not affected by the state of main task.</td></tr>
<tr><td>Conditions for resuming operation after an alarm</td><td>Clear the alarm and restart.</td><td>Clear the alarm and re-activate the static task or synchronize the project.</td><td>Clear the alarm and restart.</td><td>Clear the alarm and restart.</td></tr>
<tr><td rowspan="2">Statement-by-statement debugging</td><td>Supported</td><td>Not supported</td><td>Supported</td><td>Not supported</td></tr>
<tr><td colspan="4">NOTE
• When you click the "Step into" button, the main task and dynamic task each run forward by one line of statement.
• If you click the "Step into" button when the current line of the main task is not completed, the main task continues to run the current line, and the dynamic task runs forward by one line of statement.</td></tr>
<tr><td rowspan="2">Return to the start line</td><td>Supported. Supports returning a single task or all tasks to the start line.</td><td>Supported. Supports returning a single task to the start line.</td><td>Supported. Supports returning a single task or all tasks to the start line.</td><td>Not supported</td></tr>
<tr><td colspan="4">NOTE
Returning to the start line is allowed only when the task is in the stop or ready state.</td></tr>
</table>

7.3.1 Task Management

- In addition to controlling the robot motion, the controller also needs to interact with external devices, or control peripheral devices. This can be achieved through multi-task processing.
- Most of the operations of multi-task debugging are the same as those of main-task debugging. You can view the refresh of different task program lines by switching between the programs.
- The breakpoint debugging is valid only in the main task and dynamic task. The breakpoints are invalid in static tasks and xqt task.

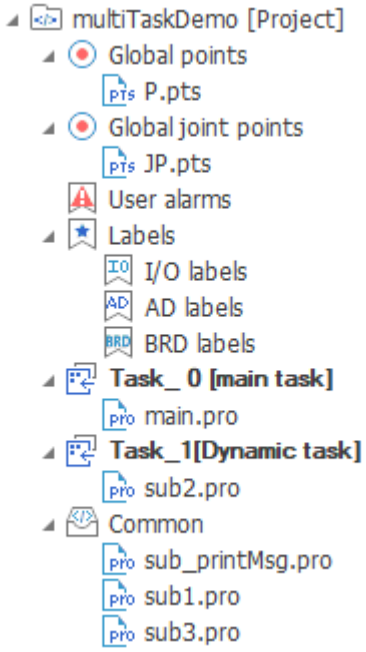
CAUTION

- Functions in a common program file can be called by multiple tasks at the same time.
- The breakpoint debugging for multi-task debugging is valid only in the main task and dynamic task. The breakpoints are invalid in static tasks and xqt task.

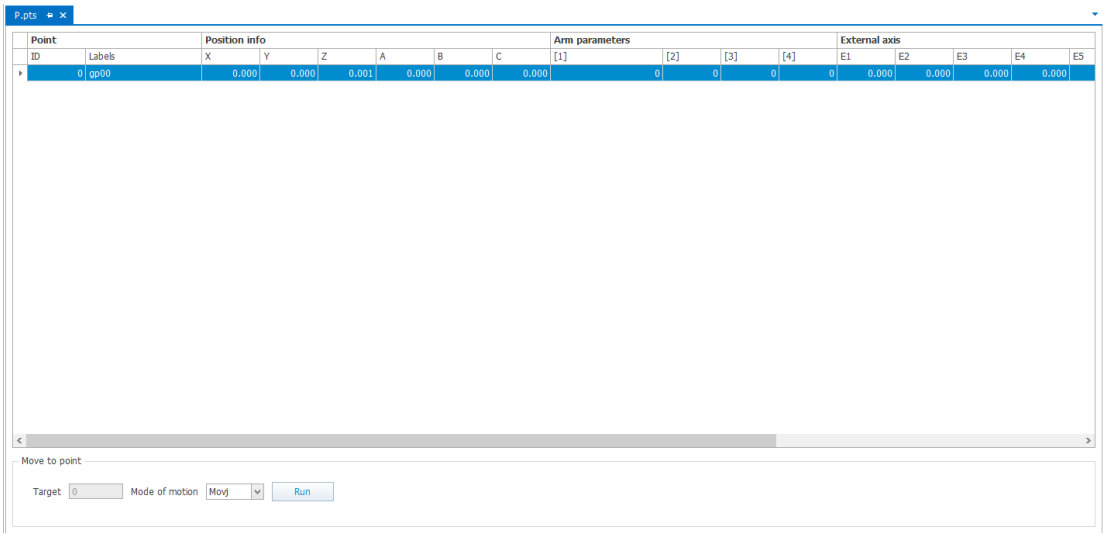
7.3.2 Project Example

The following project is an example of how to implement joint debugging of multiple tasks.

1. The project tree is shown below.



2. Global points



3. "main.pro" program file

```
Include "sub1.pro";
Include "sub_printMsg.pro";
Start;
   Movj LP[0],V[100],Z[0],Tool[0];
   Movj LP[1],V[100],Z[0],Tool[0];
   Movj LP[0],V[100],Z[0],Tool[0];
   Movj LP[1],V[100],Z[0],Tool[0];
   While 1
      Int aaa = 0;
      aaa = aaa + 5;
      B[0] = 1;
      Movj gp00,V[100],Z[0],Tool[0];
      PrintTool[1];
      Print User[1];
      Print "demo";
      Movj LP[1],V[100],Z[0],Tool[0];
      Movj P[0],V[100],Z[0],Tool[0];
      Print "test";
      Movj LP[0],V[100],Z[0],Tool[0];
      sub_printMsg.printMsg();
      sub1.subFun();
   EndWhile;
End;
```

4. sub2.pro program file:

```
Include "sub_printMsg.pro";
Start;
   L[1]:
   Print "sub2 working1";
   Delay T[2];
   Print "sub2 working2";
   Delay T[2];
   Print "sub2 working3";
   Delay T[2];
   Print "sub2 working4";
   Delay T[2];
   sub_printMsg.printMsg();
   Goto L[1];
End;
```

5. sub3.pro program file:

```
Start;
   L[1]:
   Print "sub3 working1";
   Delay T[2];
   Print "sub3 working2";
   Delay T[2];
   Print "sub3 working3";
   Delay T[2];
   Print "sub3 working4";
   Delay T[2];
   Goto L[1];
End;
```

6. sub_printMsg.pro program file:

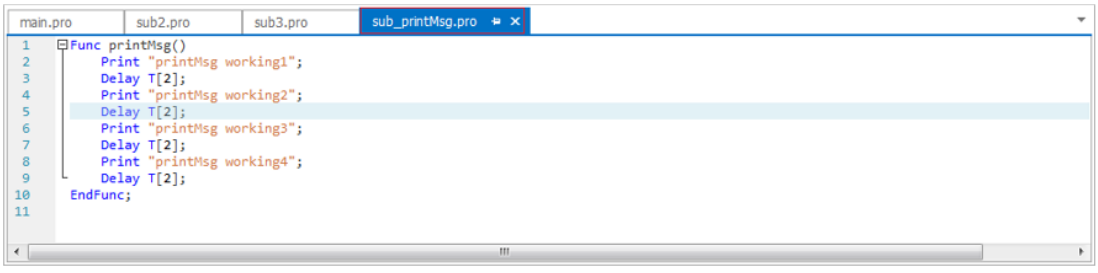
```
Func printMsg()
   Print "sub3 working1";
   Delay T[2];
   Print "sub3 working2";
   Delay T[2];
   Print "sub3 working3";
   Delay T[2];
   Print "sub3 working4";
   Delay T[2];
EndFunc;
```

7. sub1.pro program file:

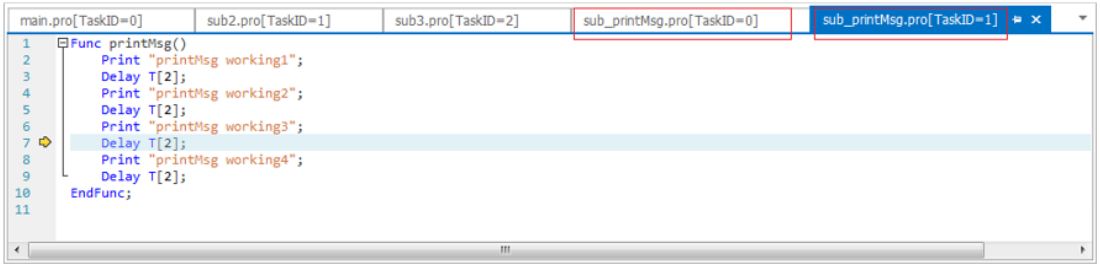
```
Func subFun()
   Movj LP[0],V[100],Z[0],Tool[0];
   Movj LP[1],V[100],Z[0],Tool[0];
   Movj LP[0],V[100],Z[0],Tool[0];
   Movj LP[1],V[100],Z[0],Tool[0];
   Movj LP[0],V[100],Z[0],Tool[0];
   Movj LP[1],V[100],Z[0],Tool[0];
EndFunc;
```

7.3.3 Displaying Multiple Tasks Calling the Same Function

For example, the "printMsg()" function in "sub_printMsg.pro" in the example project is called in Task 0 and Task 1. Before debugging, you only need to open "sub_printMsg.pro", as shown in the figure below.



As long as a task enters this function, there will be a program file corresponding to it. Therefore, during debugging, the program file will be opened automatically.



7.3.4 Viewing Task Switches During Debugging

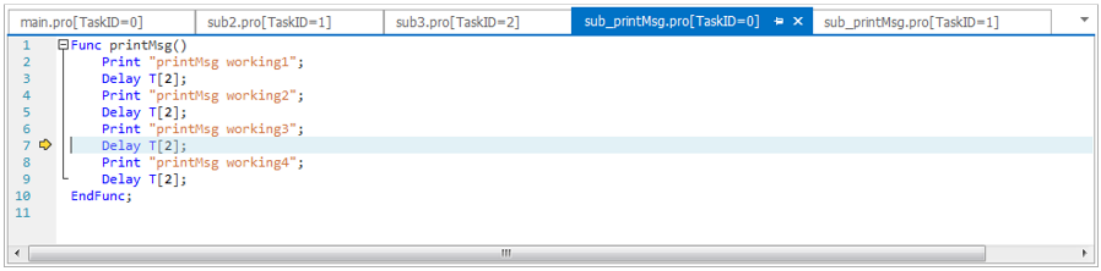
During multi-task debugging, you need to view the operation status of multiple tasks. Only one page is activated at a time. Therefore, it is necessary to switch the tasks viewed during debugging.

Prerequisite

Synchronize a multi-task project and then start debugging of the program.

Steps

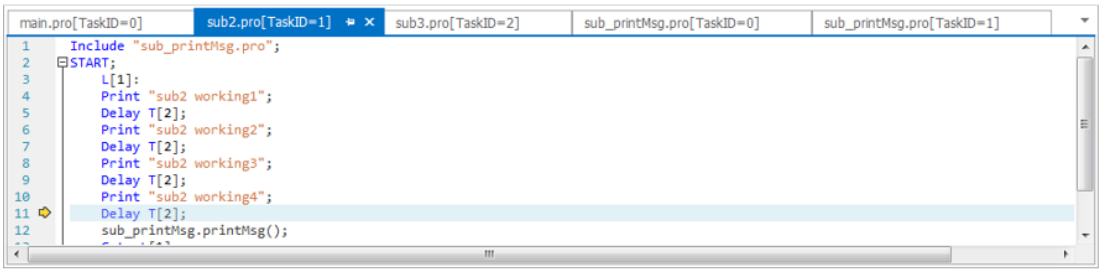
1. As shown below, the program files "main.pro[TaskID=0]" and "sub_printMsg.pro[TaskID=0]" are automatically switched for view. These two program files have the attribute "[TaskID=0]", indicating that they both belong to task 0.



2. When you need to switch between tasks, click the title bar of the program file for the corresponding task, for example, "sub2.pro[TaskID=1]" and "sub_printMsg.pro[TaskID=1]".

Results

The program files are switched between "sub2.pro[TaskID=1]" and "sub_printMsg.pro[TaskID=1]".

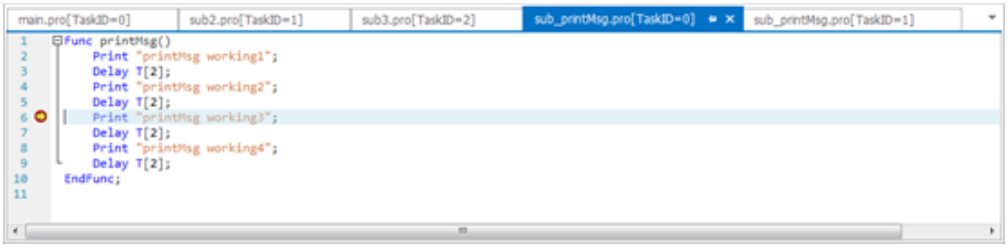


> **NOTE**
>
> When a breakpoint is entered, the program file that enters the breakpoint is automatically activated. In this case, the program file may not belong to the previous task. For details, see the 7.3.5 Entering a Breakpoint section.

7.3.5 Entering a Breakpoint

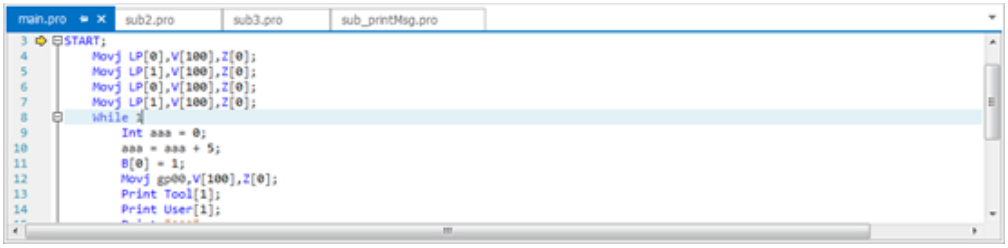
Scenario 1:

View the corresponding program file document that triggers the current breakpoint, as shown in the figure below:

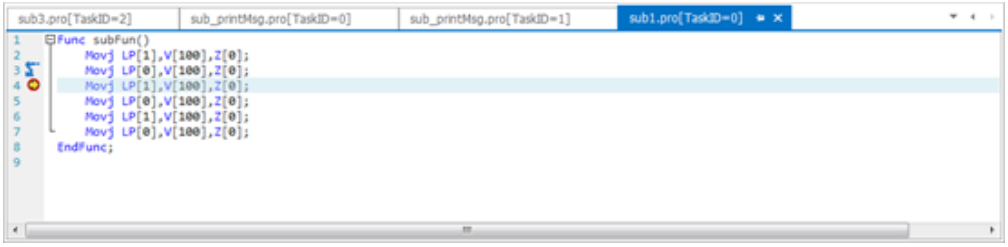


Scenario 2:

The program file, for example "sub1.pro", includes a breakpoint but the file is not opened before debugging.



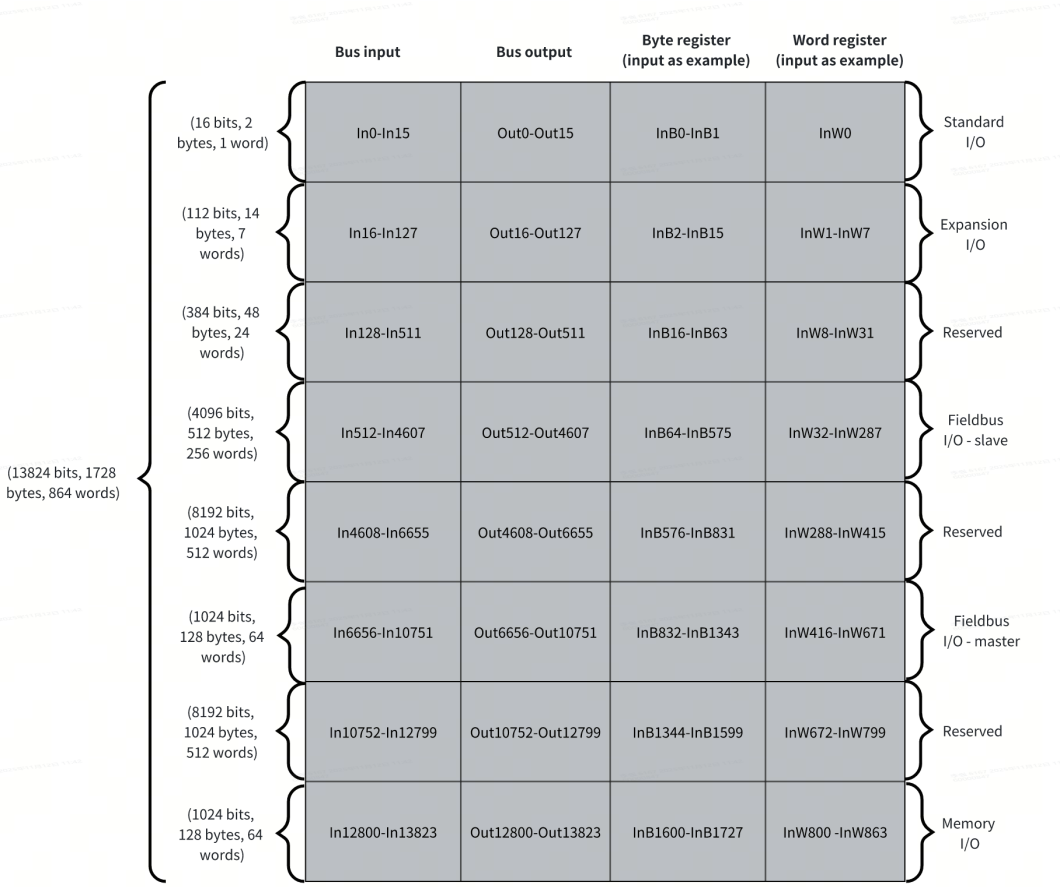
Then the program file "sub1.pro[TaskID=0]" is opened automatically.



7.4 I/O Monitor

7.4.1 Introduction of Robot Bus Addresses

The following introduces the concept of the robot bus addresses. The bus addresses of the robot include standard I/Os, fieldbus I/Os, and memory I/Os, as shown in the figure below.



- Standard I/O: Equivalent to digital I/O, i.e., I/O where the switching state is directly associated with the physical level high or low.
- Fieldbus I/O: Currently, the fieldbus I/O of the robot refers to the I/O used by the robot to interact with the master when the robot is acting as a fieldbus slave.
- Memory I/O: Equivalent to the robot's internal Bool-type variables, used for internal arithmetic.

7.4.2 Inputs

7.4.2.1 Overview

You can view the input status of standard I/Os, fieldbus I/Os, memory I/Os, and system I/Os. Also, you can force the input signal of standard I/Os, and observe the corresponding input status in the "Status" and "Value" columns.

The inputs can be displayed in Bit, Byte, and Word based on the size of the variable, corresponding to In, InB and InW of the system I/O variable respectively. The following describes the three monitoring modes: Bit, Byte, and Word.

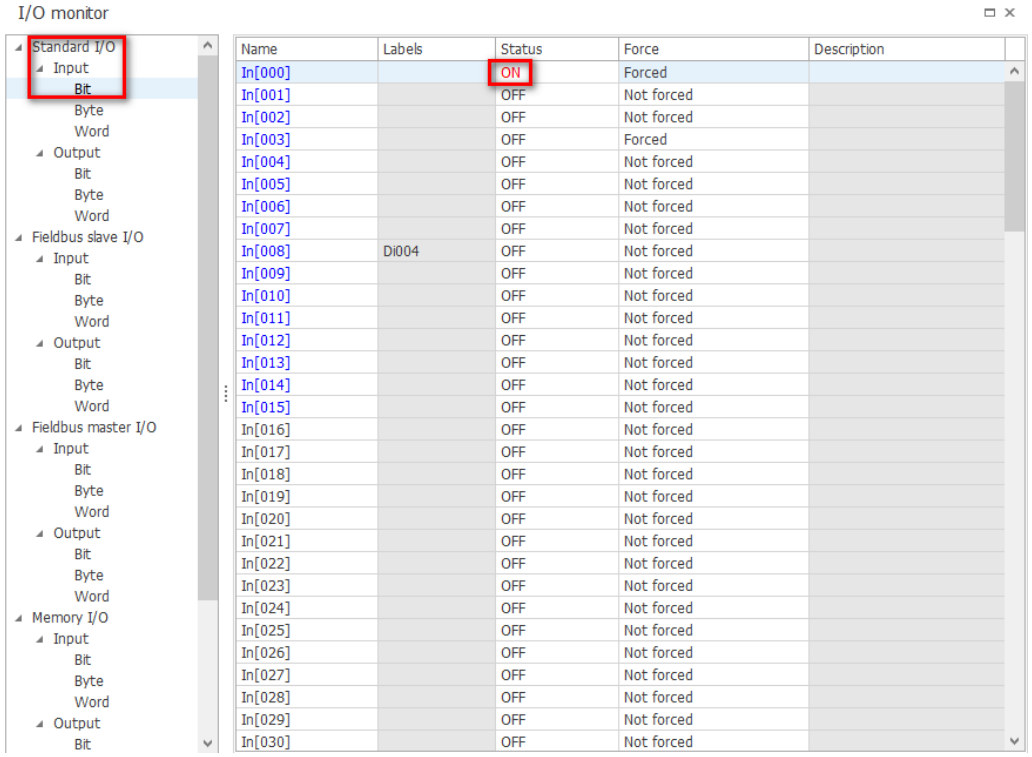
Prerequisite

- Ensure that the controller is connected, that the input and output modules are connected correctly, and that the IR-Link settings are made correctly.
- To monitor fieldbus I/Os, ensure that the address is assigned correctly and that the fieldbus master is connected.

7.4.2.2 Bit Monitoring

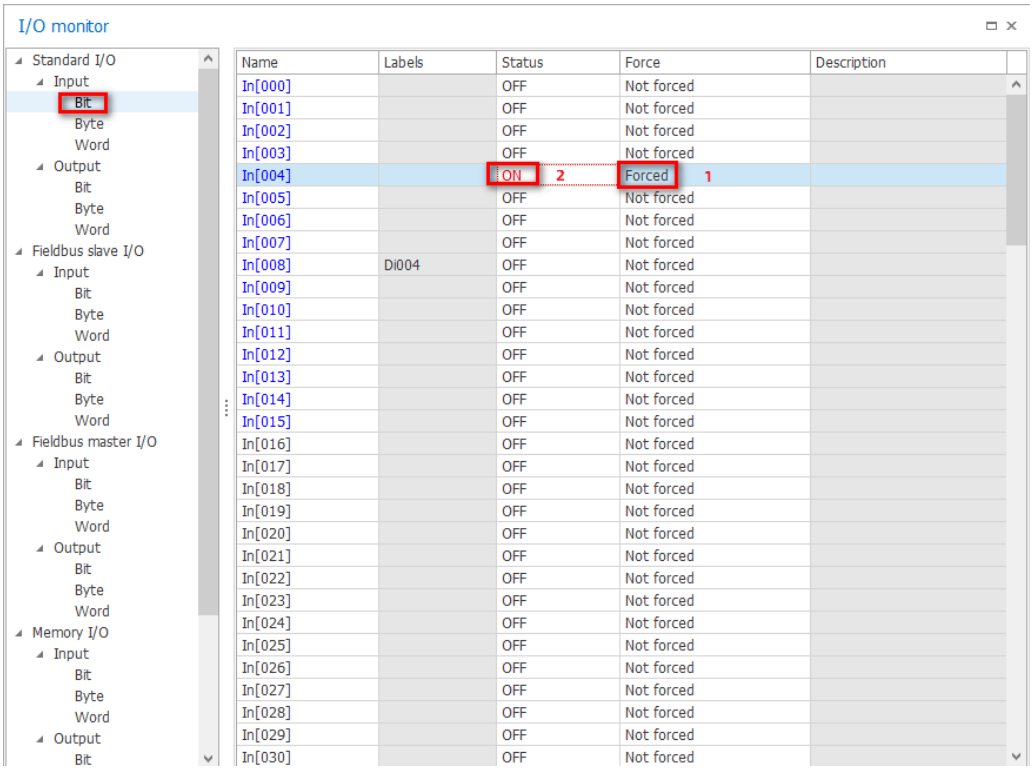
Viewing the Input Status

On the I/O monitor page, select the bus type to be monitored and go to "Standard I/O" > "Input" > "Bit". You can view the input status, as shown below.



Forcing the Input Status

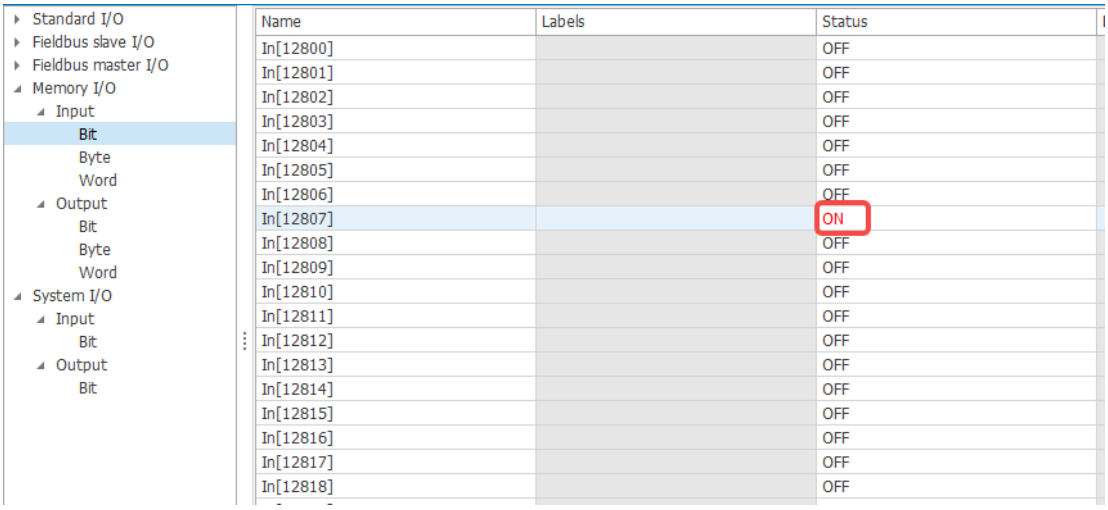
Select "Standard I/O" > "Input" > "Bit", click the force switch of an input, for example, In[005], to switch it to "Forced", and then click the input's status to change its status from "OFF" to "ON".



After the forcing takes effect, the input is judged according to the forced value in the logical judgment of the program.

Changing the Input Status of Memory I/Os

Select "Memory I/O" > "Input" > "Bit", click the status of an input, for example In[12806]. The status of the input is changed, for example, from "OFF" to "ON".

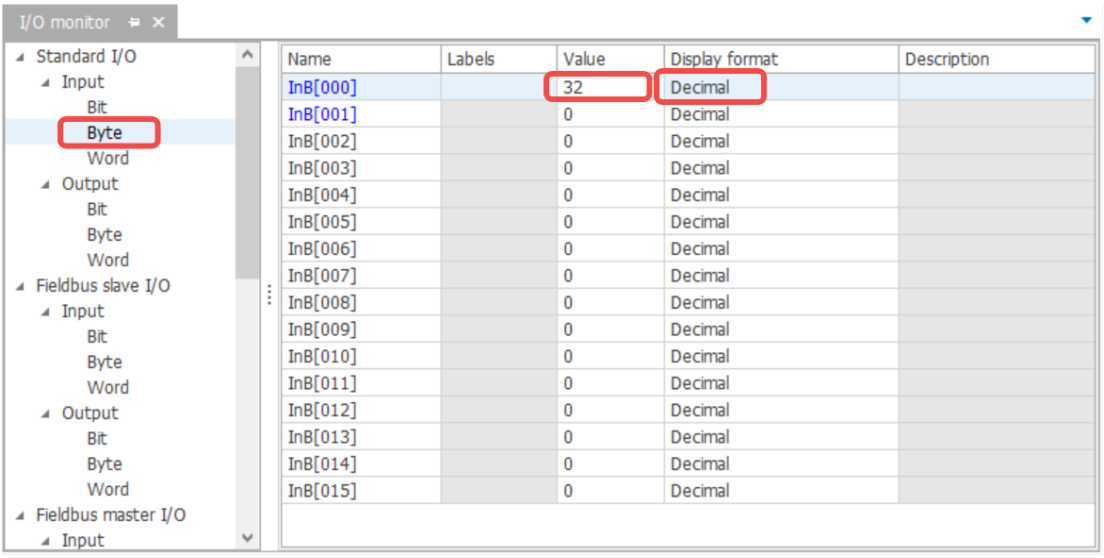


After the change takes effect, the input is judged according to the value displayed in the monitoring list in the logical judgment of the program.

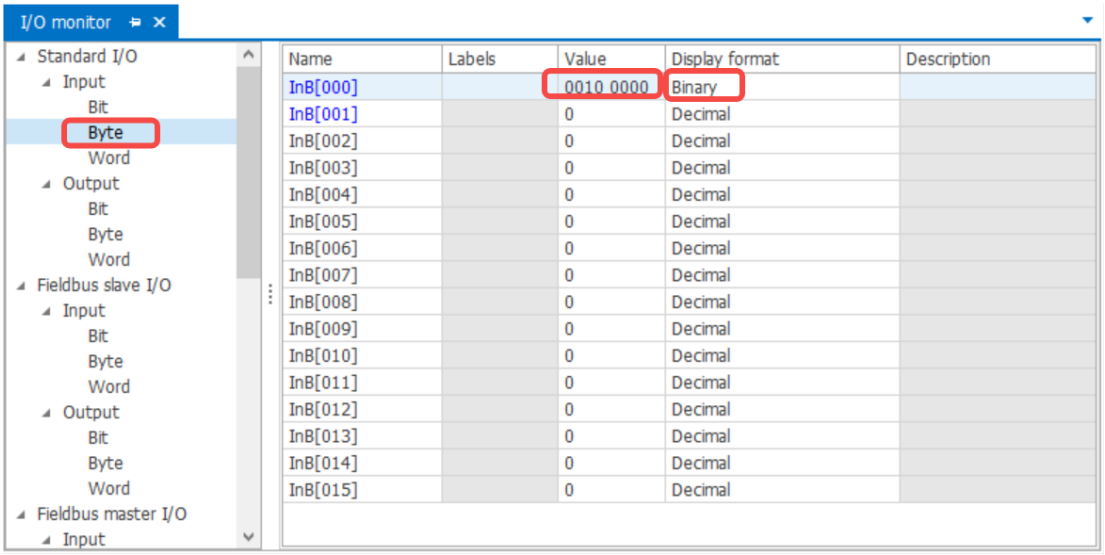
7.4.2.3 Byte Monitoring

Viewing the Input Status

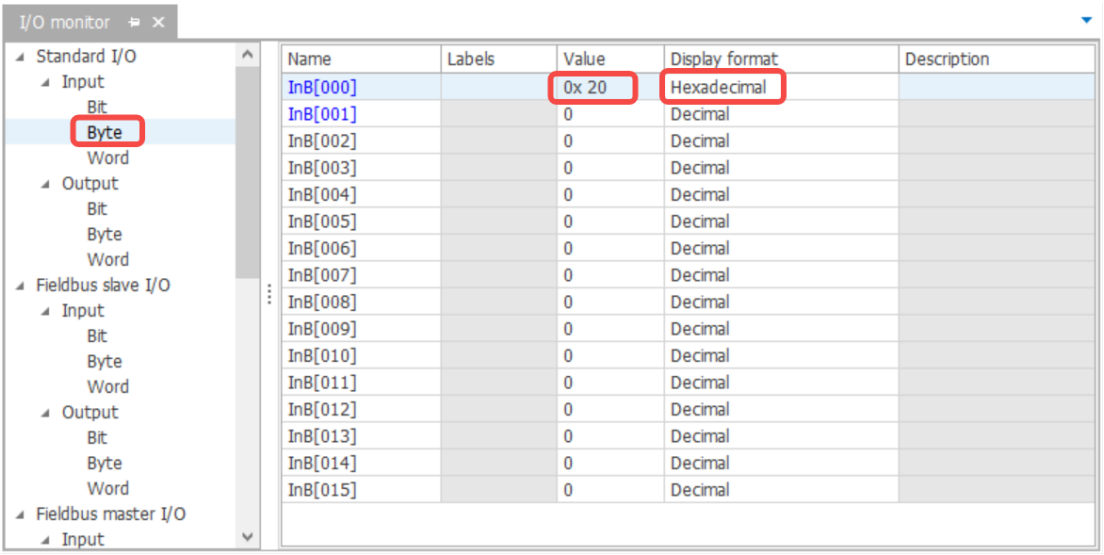
On the I/O monitor page, select the bus type to be monitored and go to "Standard I/O" > "Input" > "Byte". You can click the "Display format" cell to change the display format including decimal, binary, hexadecimal and 32-bit floating-point. For example, InB[000] is displayed in decimal with a value of 32.



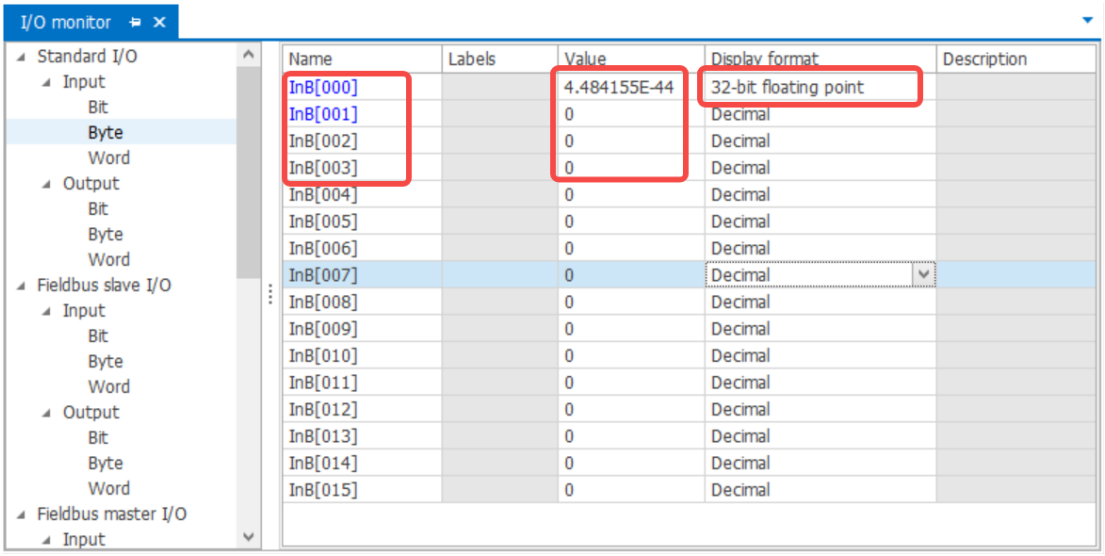
When displayed in binary, the value of InB[000] is 00100000.



When displayed in hexadecimal, the value of InB[000] is 0x20.



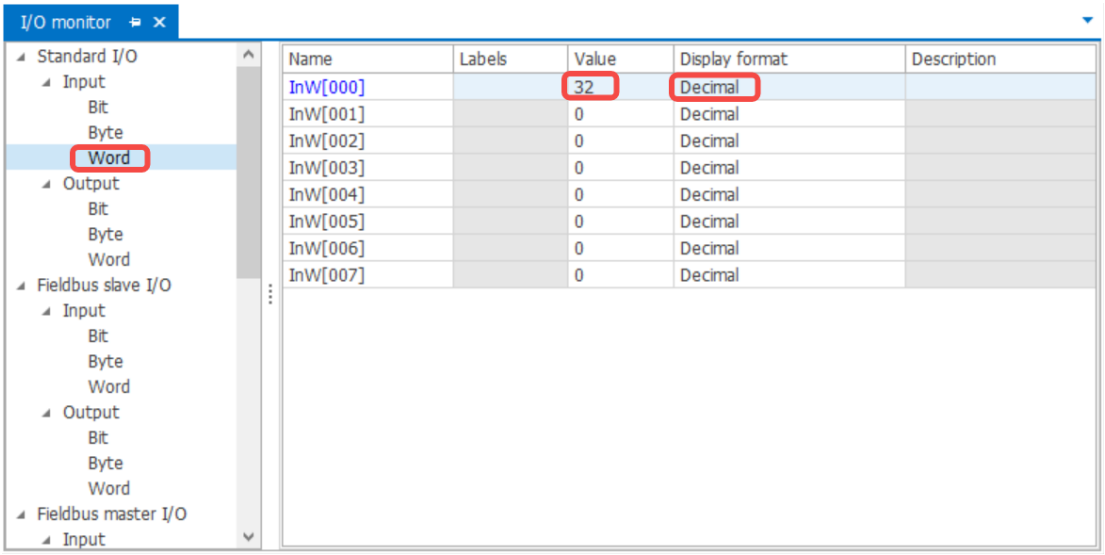
When the 32-bit floating-point format is selected, the value displayed is actually a 32-bit floating-point number consisting of four bytes, as there are only 8 bits in a byte. For example, InB[000] is displayed as a 32-bit floating-point number as shown below. The value 4.484155E-44 indicates the value of a 32-bit floating-point number consisting of four bytes InB[000], InB[001], InB[002], and InB[003].



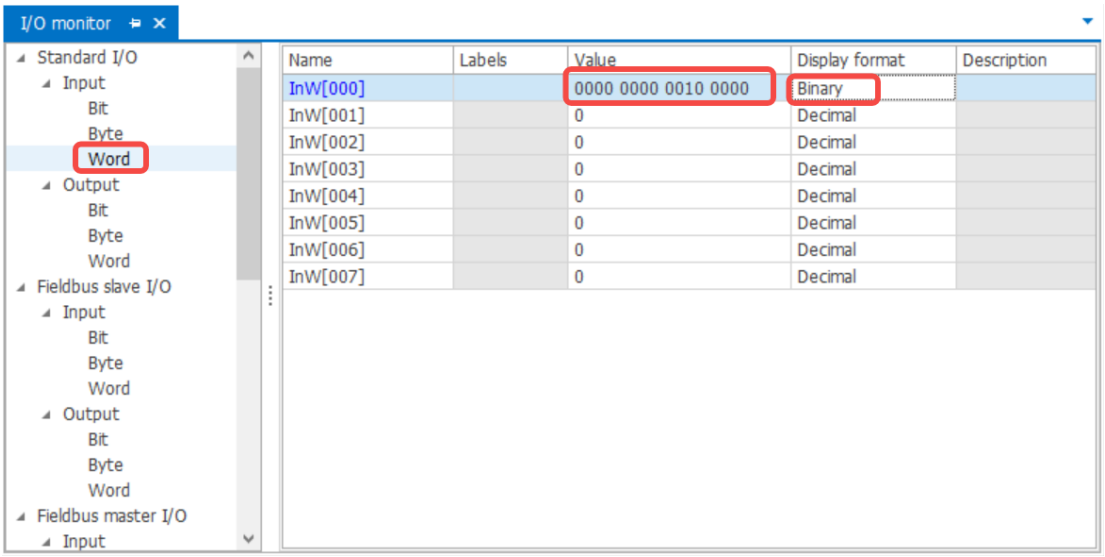
7.4.2.4 Word Monitoring

Viewing the Input Status

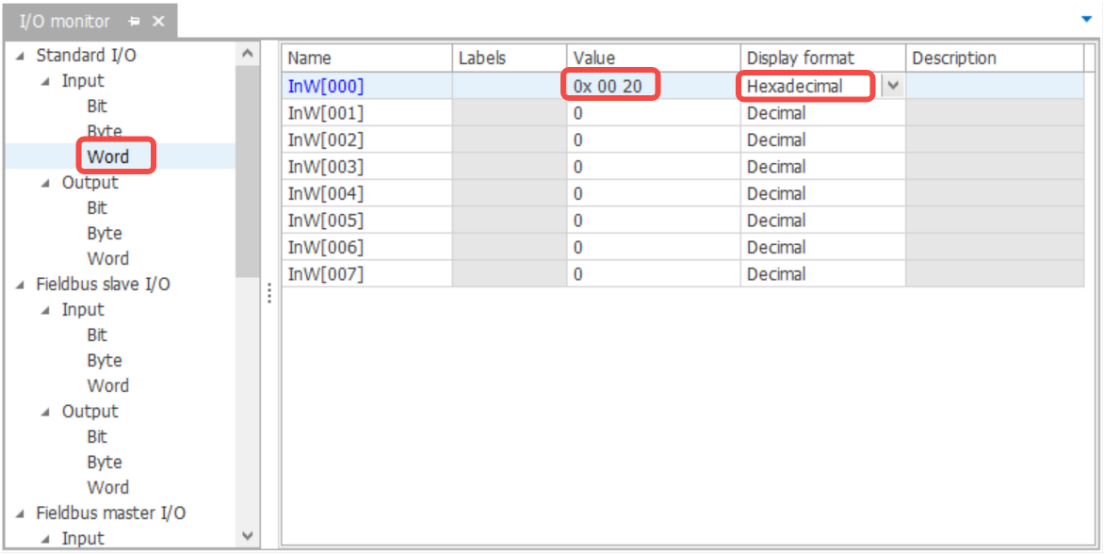
On the I/O monitor page, select the bus type to be monitored and go to "Standard I/O" > "Input" > "Word". You can click the "Display format" column to change the display format including decimal, binary, hexadecimal and 32-bit floating-point. For example, InW[000] is displayed in decimal with a value of 32.



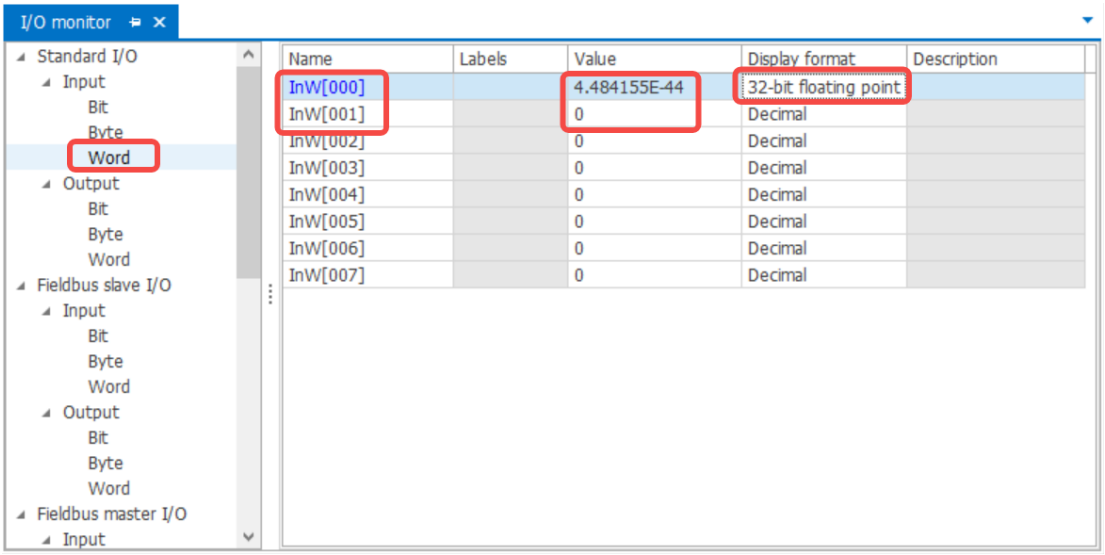
When displayed in binary, the value of InW[000] is 0000 0000 0010 0000.



When displayed in hexadecimal, the value of InW[000] is 0x 0020.



When the 32-bit floating-point format is selected, the value displayed is actually a 32-bit floating-point number consisting of two words, as there are only 16 bits in a word. For example, InW[000] is displayed as a 32-bit floating-point number as shown below. The value 4.484155E-44 indicates the value of a 32-bit floating-point number consisting of two bytes InW[000] and InW[001].



7.4.3 Outputs

7.4.3.1 Overview

You can view the output status of standard I/Os, fieldbus I/Os, memory I/Os, and system I/Os. Also, you can modify the output status. You can observe the output status in the "Status" and "Value" columns.

The outputs can be displayed in Bit, Byte, and Word based on the size of the variable, corresponding to Out, OutB and OutW of the system I/O variable respectively. The following describes the three monitoring modes: Bit, Byte, and Word.

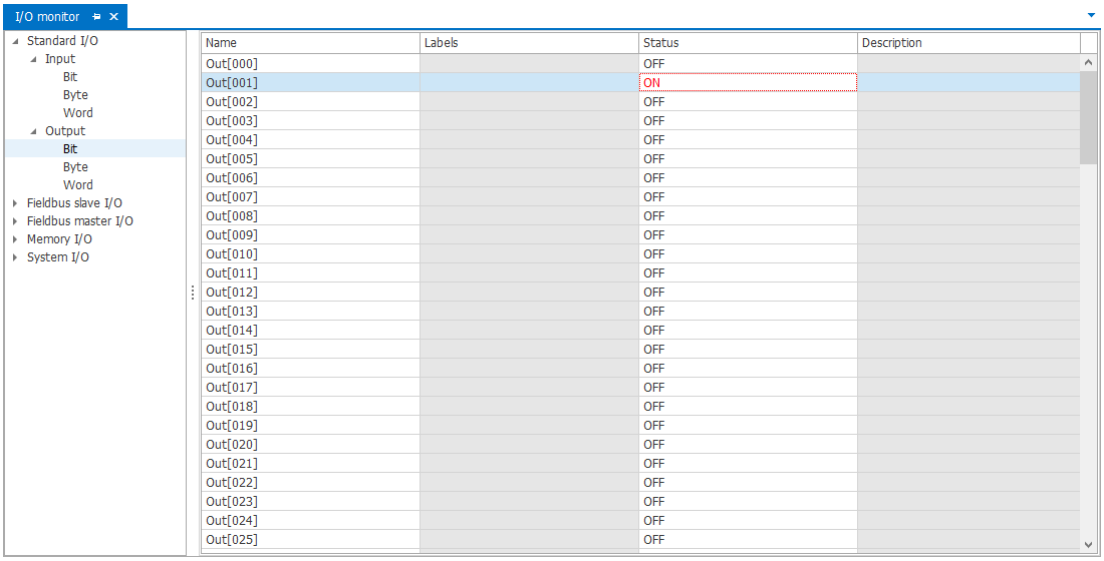
Prerequisite

1. Ensure that the controller is connected, that the input and output modules are connected correctly, and that the IR-Link settings are made correctly.
2. To monitor fieldbus I/Os, ensure that the address is assigned correctly and that the fieldbus master is connected.

7.4.3.2 Bit Monitoring

Viewing the Output Status

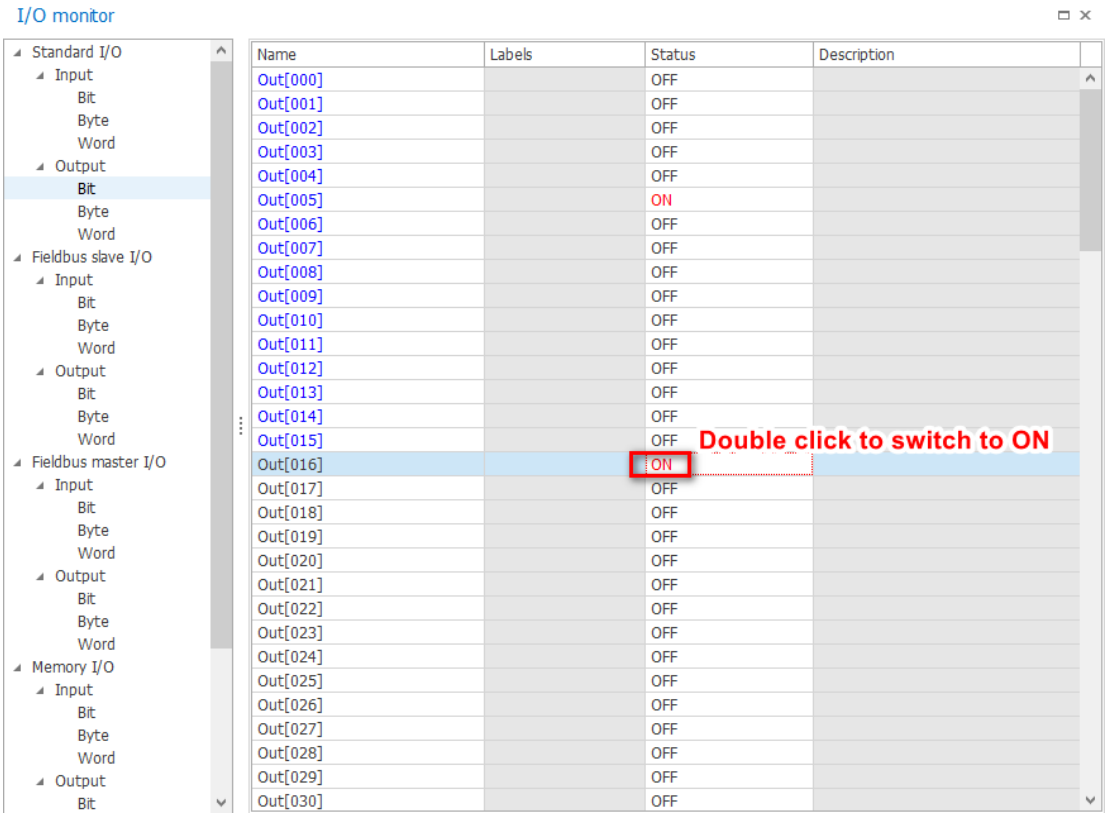
On the I/O monitor page, select the bus type to be monitored and go to "Standard I/O" > "Output" > "Bit". You can view the output status, as shown below.



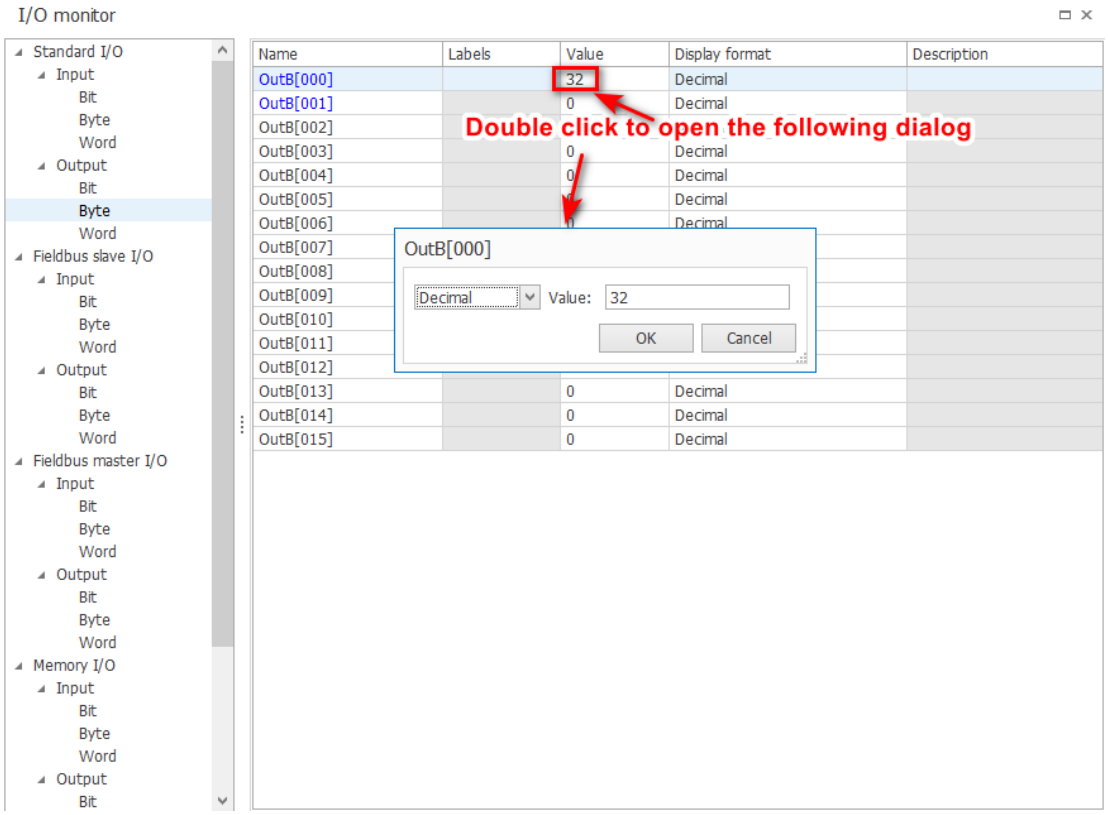
Switching the Output Status

Double-click the "Status" or "Value" cell of an output to change its status value.

- Modification of Bit



- Modification of Byte or Word



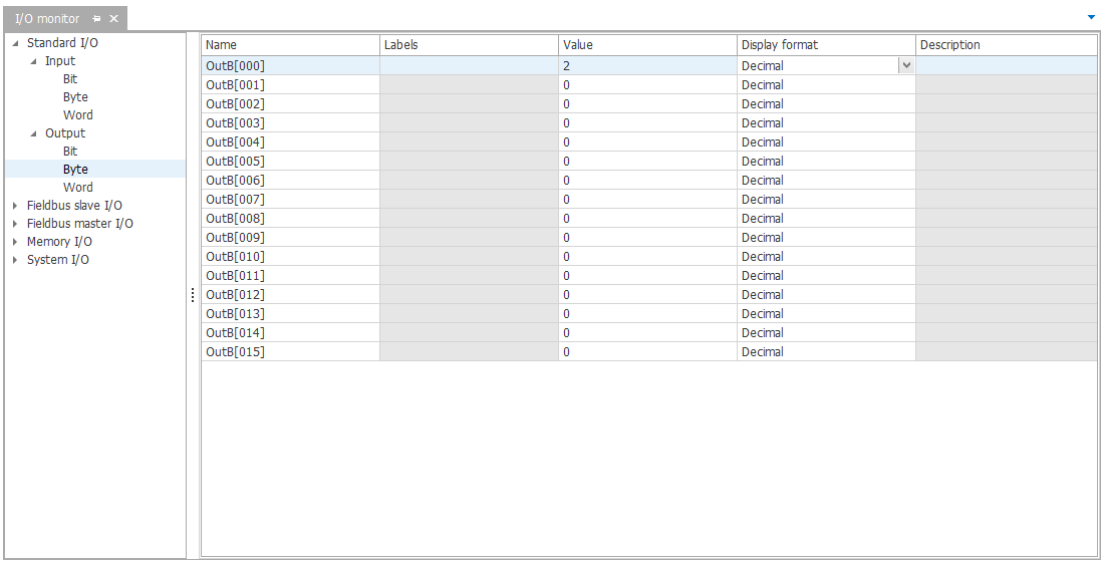
NOTE

When the output is used in the I/O configuration, the output state cannot be changed.

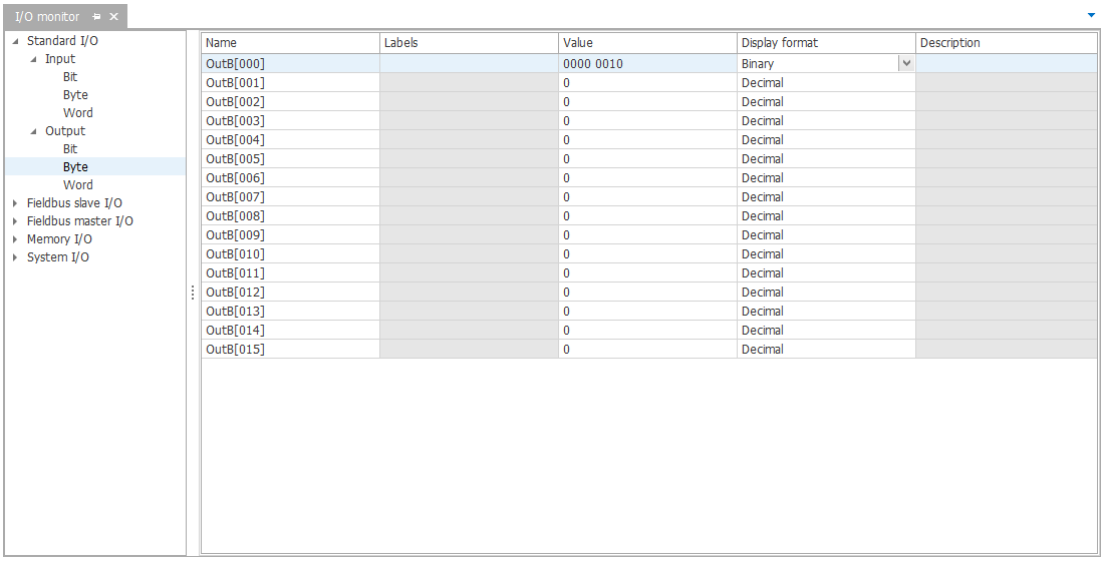
7.4.3.3 Byte Monitoring

Viewing the Output Status

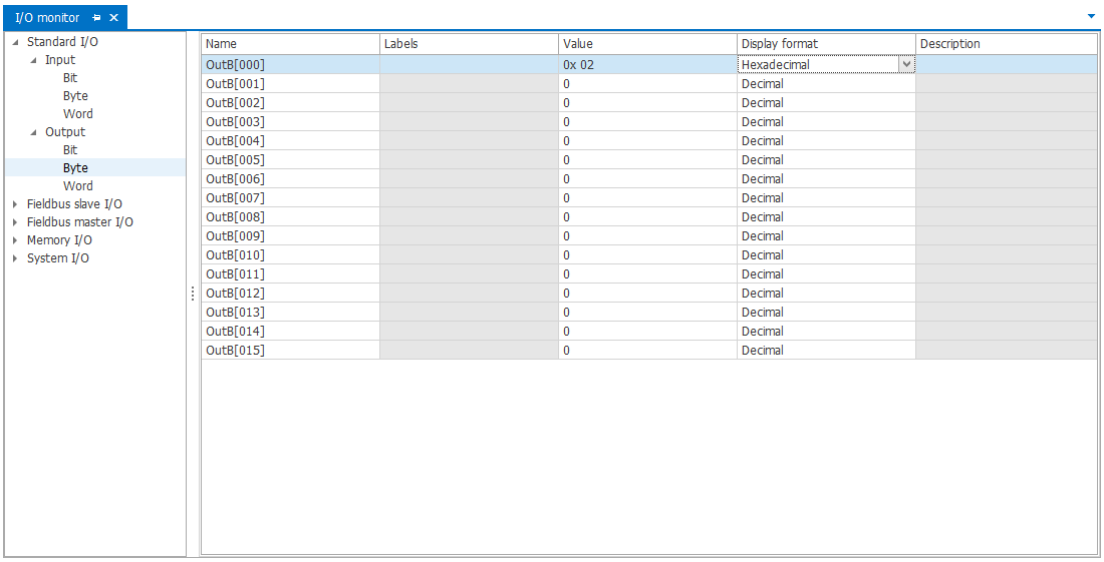
On the I/O monitor page, select the bus type to be monitored and go to "Standard I/O" > "Output" > "Byte". You can click the "Display format" cell to change the display format including decimal, binary, hexadecimal and 32-bit floating-point. For example, OutB[000] is displayed in decimal with a value of 2.



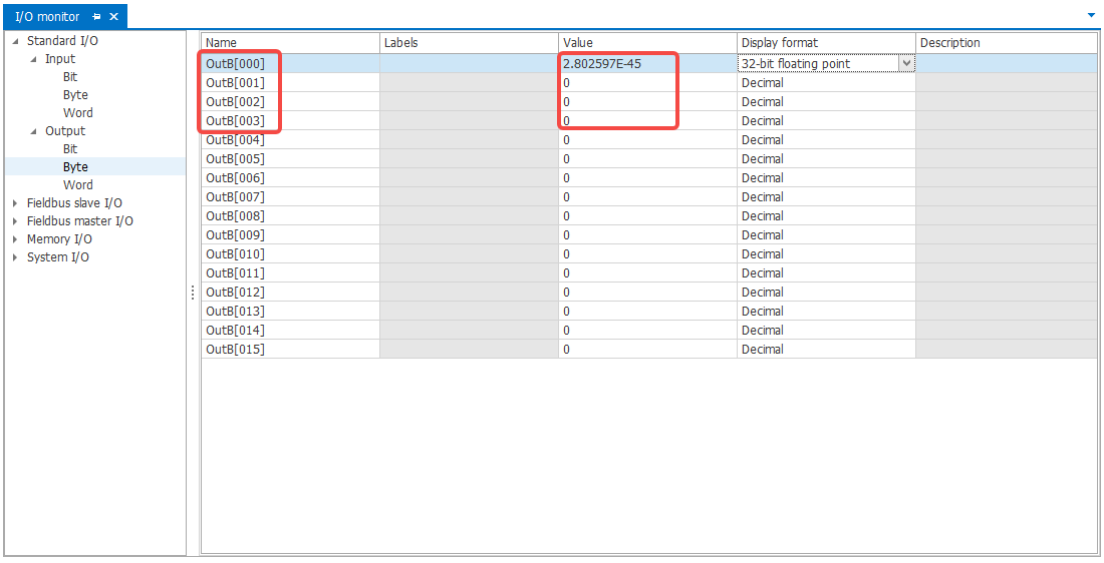
When displayed in binary, the value of OutB[000] is 0000 0010.



When displayed in hexadecimal, the value of OutB[000] is 0x02.



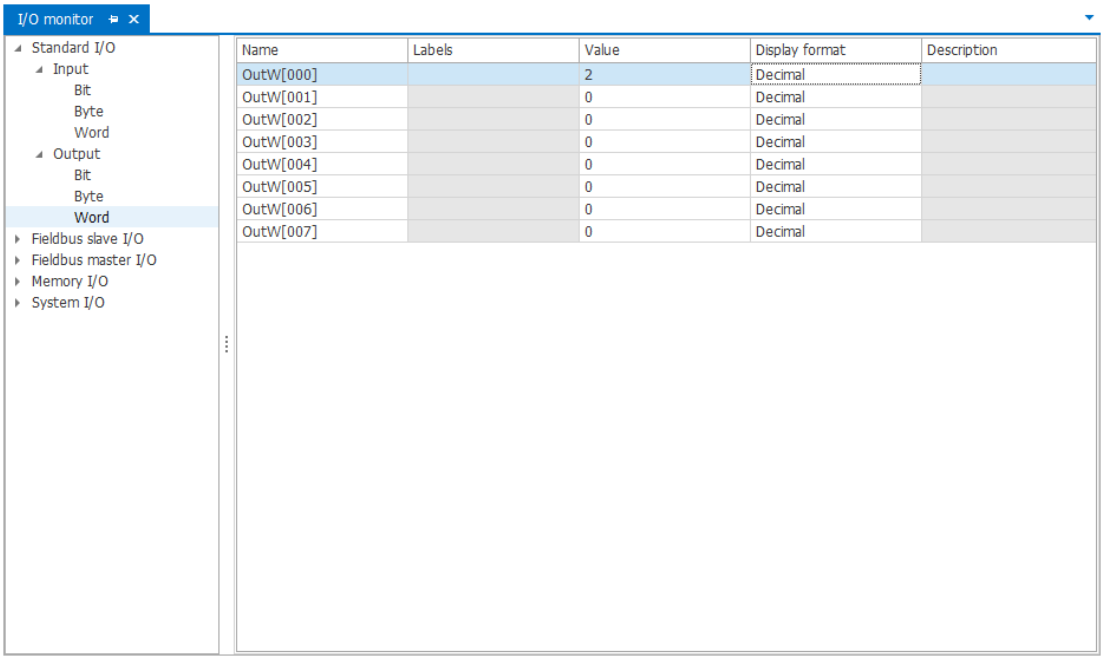
When the 32-bit floating-point format is selected, the value displayed is actually a 32-bit floating-point number consisting of four bytes, as there are only 8 bits in a byte. For example, OutB[000] is displayed as a 32-bit floating-point number as shown below. The value 2.802597E-45 indicates the value of a 32-bit floating-point number consisting of four bytes OutB[000], OutB[001], OutB[002], and OutB[003].



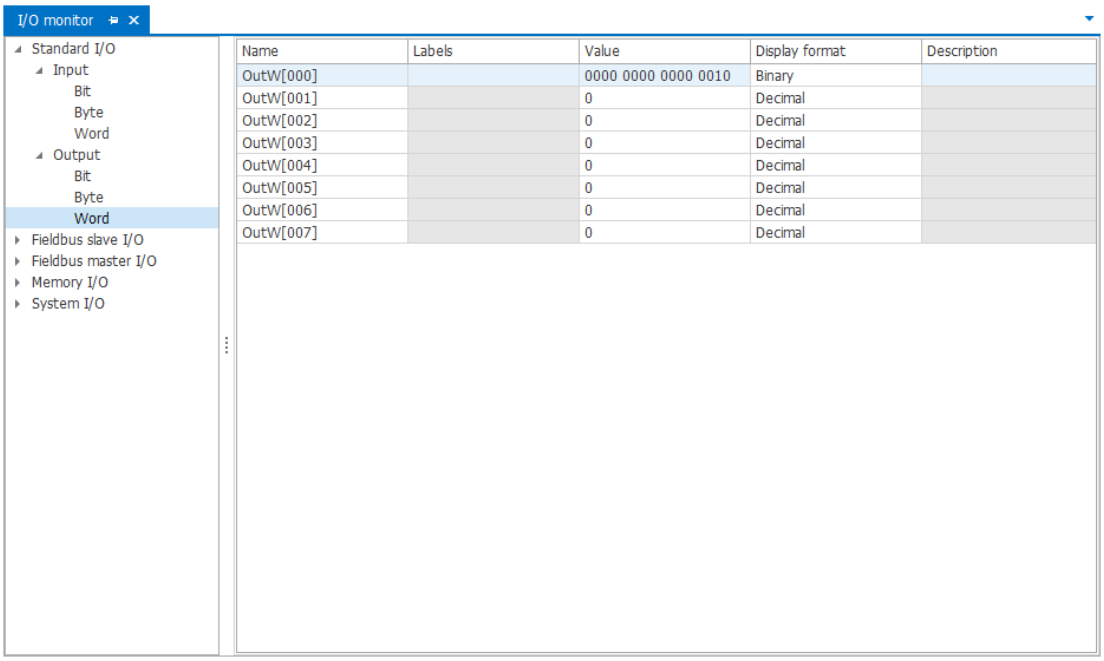
7.4.3.4 Word Monitoring

Viewing the Output Status

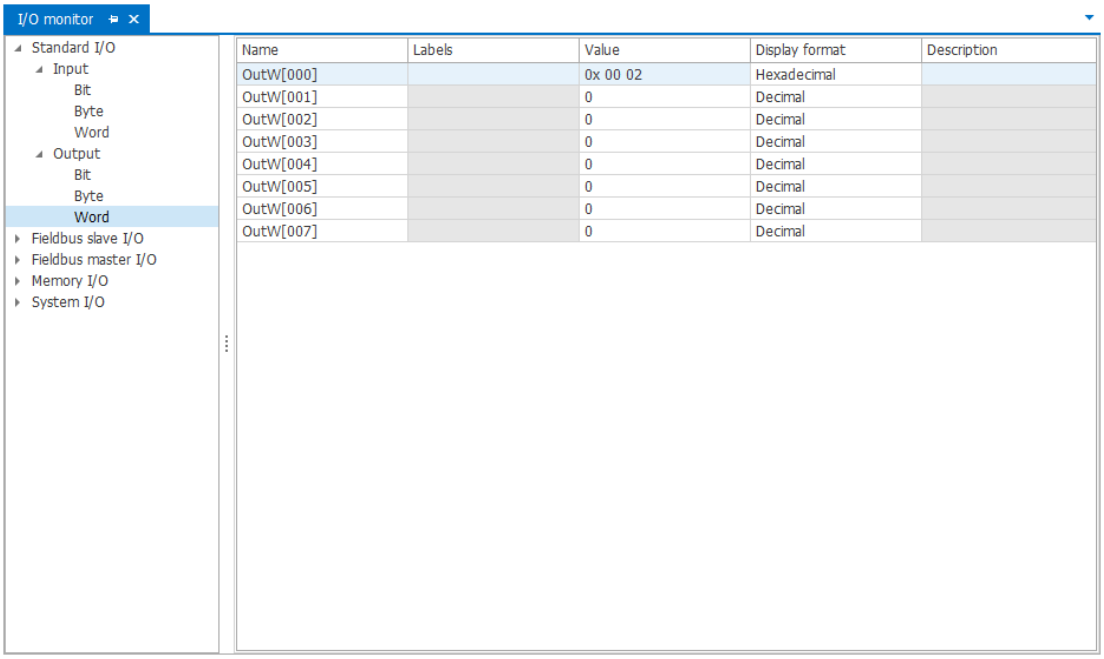
On the I/O monitor page, select the bus type to be monitored and go to "Standard I/O" > "Output" > "Word". You can click the "Display format" cell to change the display format including decimal, binary, hexadecimal and 32-bit floating-point. For example, OutW[000] is displayed in decimal with a value of 2.



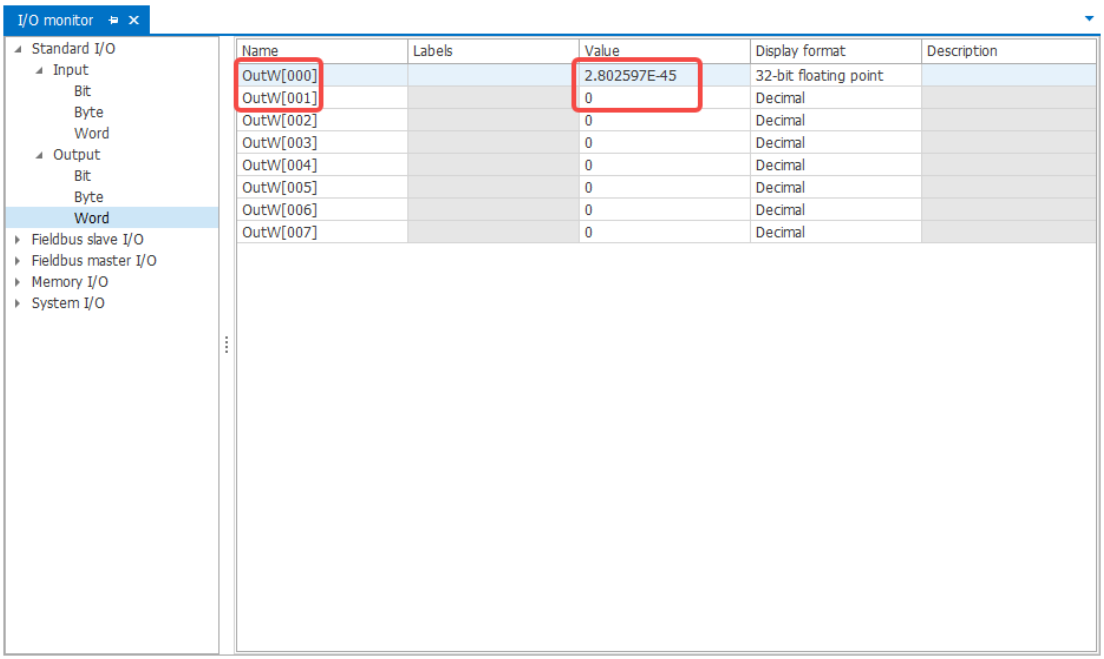
When displayed in binary, the value of OutW[000] is 0000 0000 0000 0010.



When displayed in hexadecimal, the value of OutW[000] is 0x0002.

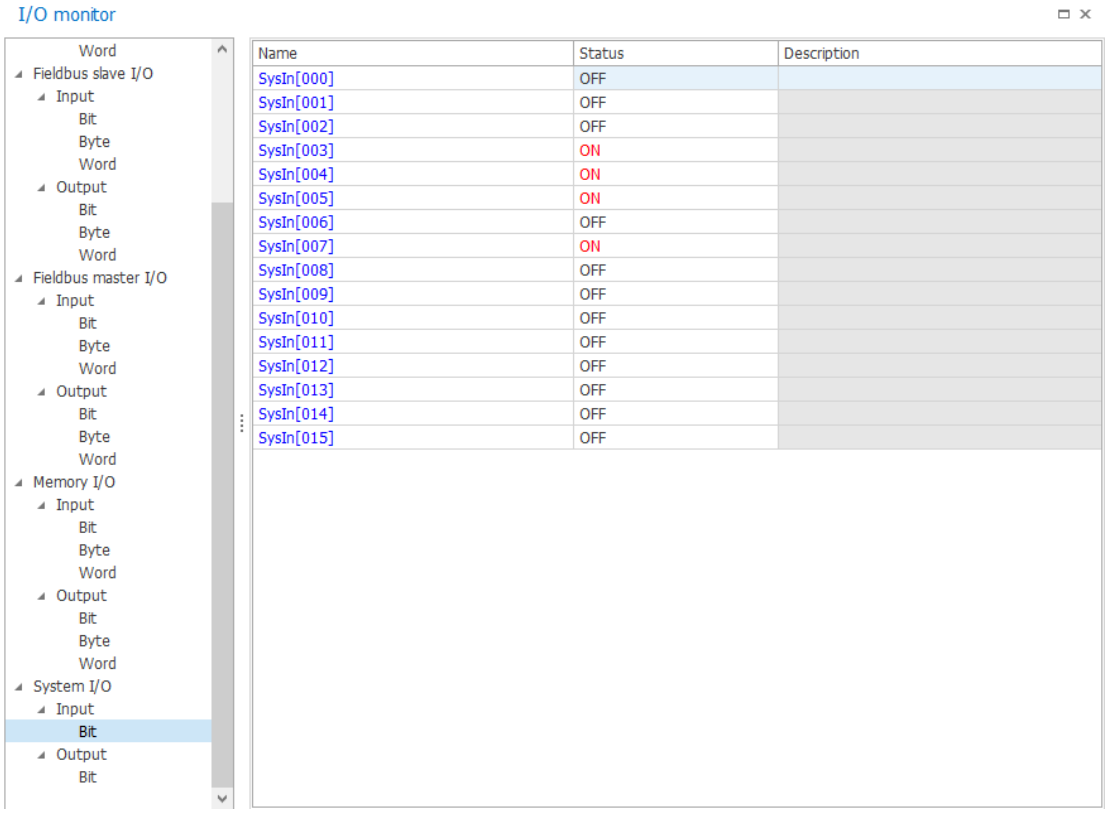


When the 32-bit floating-point format is selected, the value displayed is actually a 32-bit floating-point number consisting of two words, as there are only 16 bits in a word. For example, OutW[000] is displayed as a 32-bit floating-point number as shown below. The value 2.802597E-45 indicates the value of a 32-bit floating-point number consisting of two bytes OutW[000] and OutW[001].



7.4.4 System I/Os

The system I/Os include 16 inputs and 16 outputs, which can be displayed by bit only.



7.4.5 Meaning of Colors in I/O Monitor

In the I/O monitor, the variable names are differentiated by colors: blue, orange and black, respectively.

Color	Description
Blue	Indicates standard I/Os connected to the actual device or bus I/Os assigned with an address.
Orange	Indicates outputs which have been occupied and for which the control permissions is not RC_ ACTIVE. The status of such outputs cannot be changed through monitoring interface or instructions.
Black	• Indicates standard I/Os not connected to the actual device or bus I/Os not assigned with an address. • Indicates outputs which have not been occupied and for which the control permissions is RC_ ACTIVE.

NOTE

The orange display has a higher priority than blue and black. When an I/O is displayed in orange, if you want to know whether the I/O has blue or black attributes, you can infer based on the color of the I/Os before and after it.

Control permission of output signal is shown below:

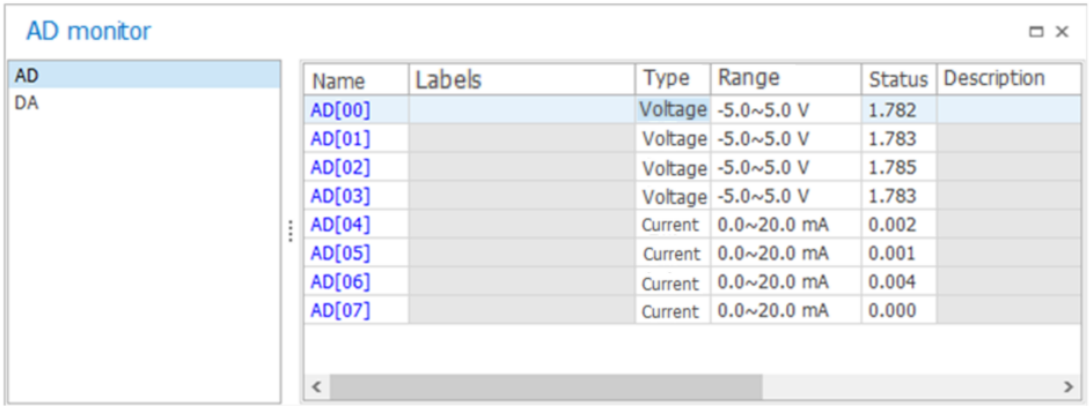
Control Permission	Description
RC_STATIC	Indicates that it is occupied by the RC system. That is, Out is bound to a system function. In this case, the output signal is related to the function only and the signal status cannot be manually changed.
RC_ACTIVE	Indicates that it is under normal RC control. In this case, the signal is normally controlled. For example, you can change the signal in "I/O Monitor" or using the Set instruction.
PLC_ACTIVE	Indicates PLC control. In this case, the signal is only controlled by other PLC software.

7.5 AD Monitor

You can view the status of the analogue inputs.

Steps

The page is displayed as shown in the figure below:



NOTE

If AD is not configured, the AD monitoring list is empty.

7.6 DA Monitor

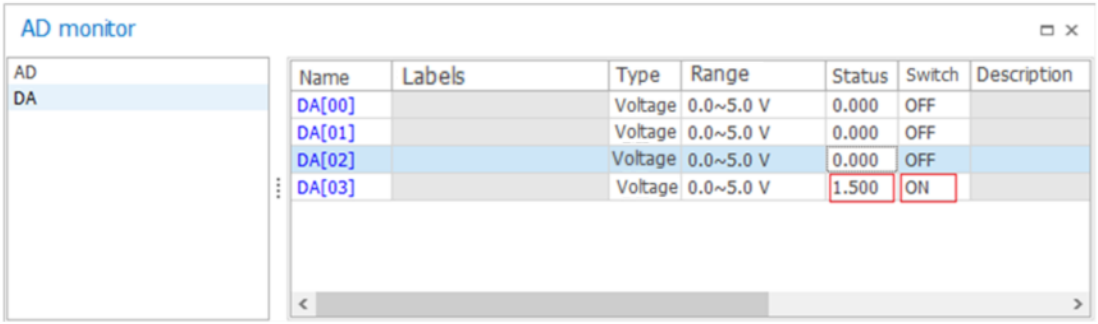
You can view the status of current analog outputs and specify the output values.

Prerequisite

Ensure that the controller is connected, that the DA module is connected correctly, and that the IR-Link settings are made correctly.

Analog outputs

Modify the value of a DA in the "Status" cell and change the switch to ON to enable the channel to output analog signals.



> **NOTE**
>
> If DA is not configured, the DA monitoring list is empty.

7.7 Global Variable Monitor

7.7.1 Overview

In the global variable monitor, you can customize the monitoring list by adding variables to the monitoring list. In addition, you can assign values to some global variables to change the current values.

Variables in the global scope can be added to the global variable monitoring list, including P, JP, B, and B. R, D, STR, PR, In, InB, InW, Out, OutB, OutW, IG, OG, AD, DA, TOOL, WOBJ, LOAD, and custom global variables (with keyword "Global"). In addition, converted bus variables such as InB[*].Float, InB[*].Double and InB[*].Int, including InB, InW, OutB, and OutW, are supported. It can be a single variable or an array of these variables.

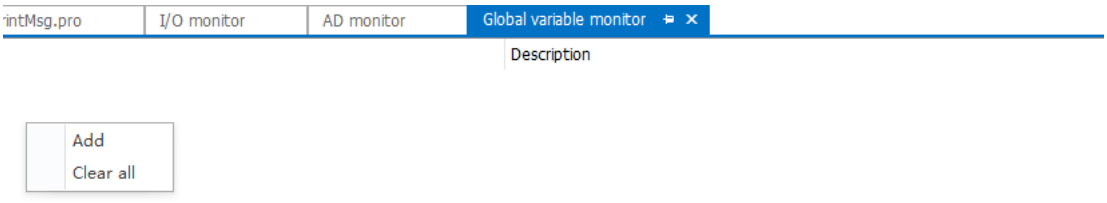
Opening the global variable monitor

Double-click the "Global variable monitoring" node on the "Station" tree to open the monitoring page.

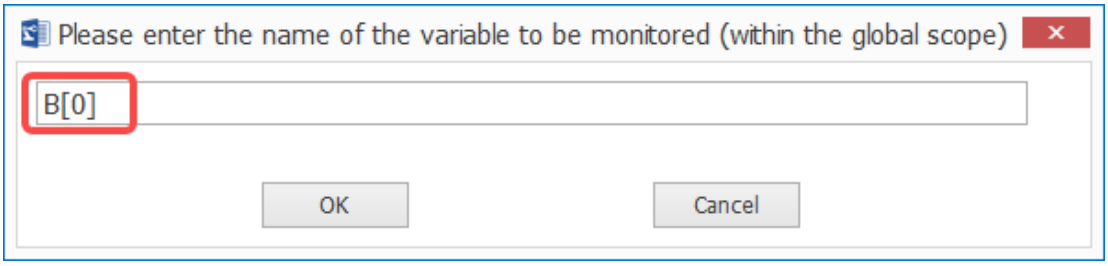
7.7.2 Adding a Monitoring Variable

- Method 1: Adding a variable in the monitoring list

1. Right-click in the monitoring list and select "Add".

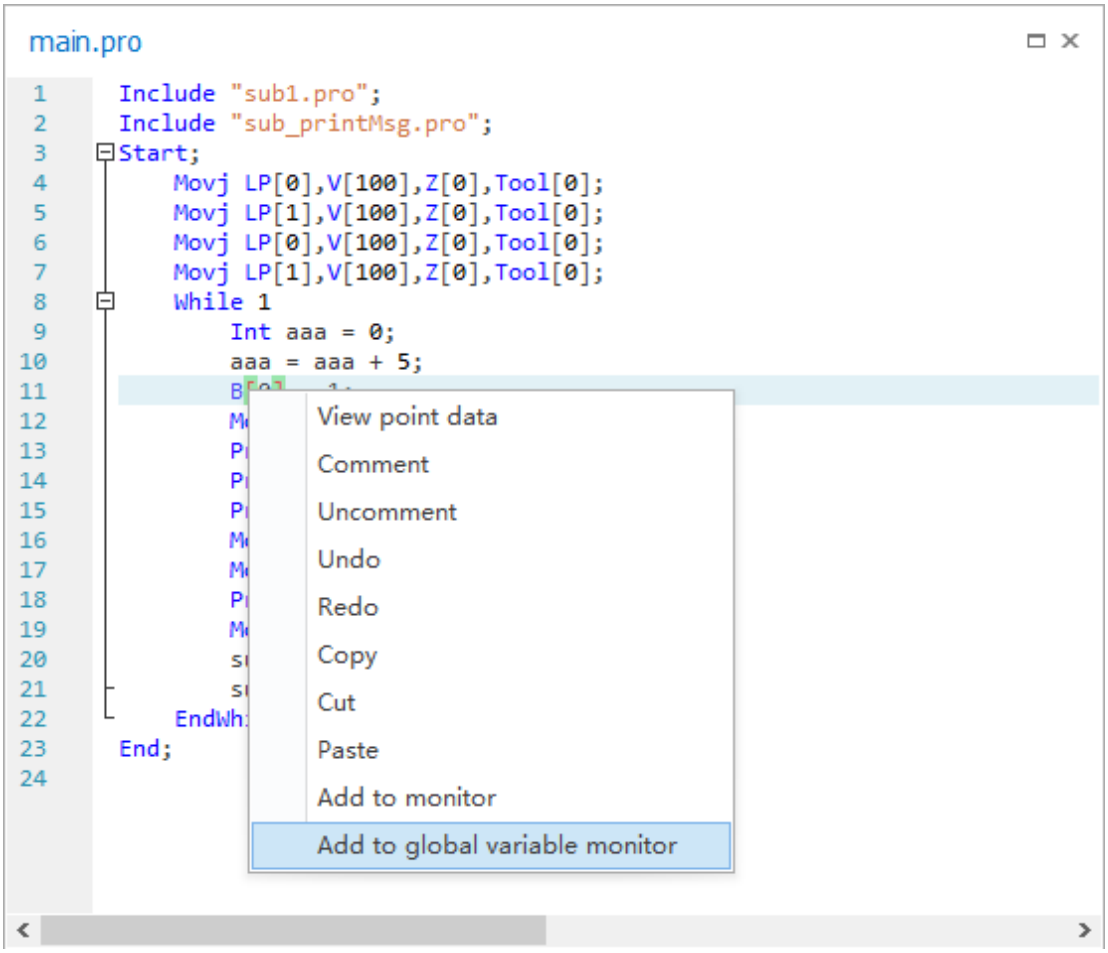


2. Enter the variable to be monitored in the input box and click "OK".



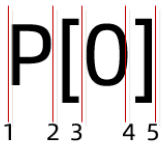
- Method 2: Adding a variable from the program file

After the project is synchronized, right-click the variable and select "Add to global variable monitor".



Additional description

- Rules of adding variables

1. Adding non-nested variables
   Here we take P[0] as an example. Adding at different cursor positions as shown below produces different effects.
2. 

   - Adding at cursor positions 1 and 2 adds P[0:49], an array of elements with a subscript range of 0 to 49.
   - Adding at cursor positions 3, 4, and 5 adds P[0], a global position point with a subscript 0.
   - Adding at the cursor position before "[" will add an array to the monitor; adding at the cursor position after "[" will add a single variable with the corresponding subscript in the array.

3. Adding nested variables

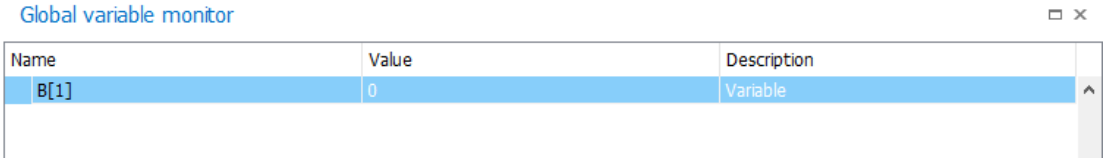
   For example, the following figure shows a variable string.



   - Adding at cursor positions 1 and 2 adds P[0:49].
   - Adding at cursor positions 3 and 4 adds B[0:49].
   - Adding at positions 5 and 6 adds B[0]. Adding at cursor positions 7 and 8 adds no variable to the monitor.
   - Whichever variable you need to monitor, move the cursor on that variable name or subscript.

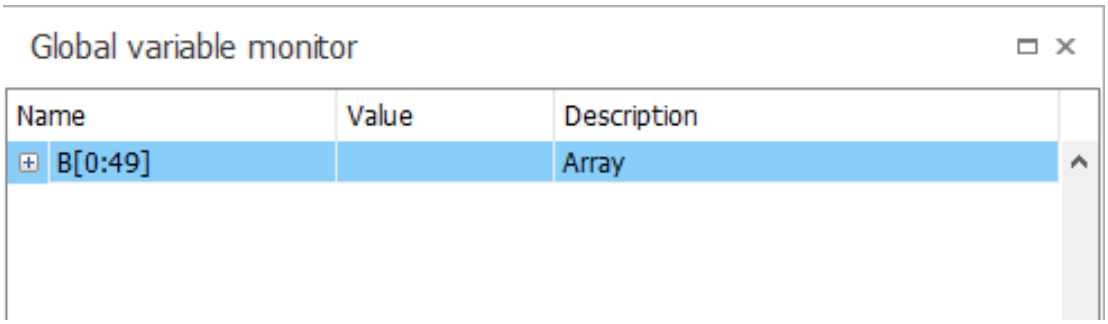
- Modification of subscripts

1. Modify the subscript of a single variable
   When you directly click B[1], you get the editing focus and can modify the subscript value.



2. Modify the subscript range of the array

When you directly click B[0:49], you get the editing focus and can modify the subscript range.



After the addition is completed, the corresponding variable is displayed in the monitoring list and the corresponding value is displayed.



NOTE

- The bottom "Variable monitor" requires the program to be in debugging mode so as to refresh the monitored variables in real time. Different from the bottom "Variable monitor", the monitored variables on the "Global variable monitor" page are refreshed in real time as long as the controller is connected.
- If you add a global variable from the program file to the monitor, you need to synchronize the current project.
- You can add a global variable to the monitor by selecting its label.
- For an array variable, up to 50 elements can be monitored.
- You can manually enter converted bus variables such as InW[0].Double, InW[0].Float and InW[0].Int. Such bus variables include InB[*], InW[*], OutB[*], and OutW[*].

CAUTION

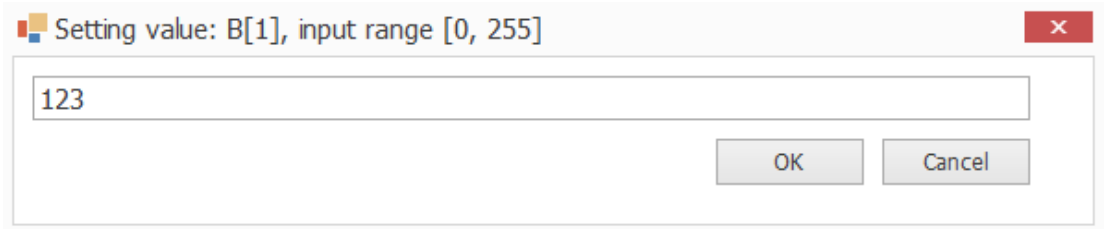
Ensure that the controller is connected. If the project is not synchronized, some variable values will not be refreshed.

7.7.3 Setting the Variable Value

1. Right-click the corresponding variable and select "Setting value".



2. Enter a value in the input box, and then click "OK".



3. You can see that the value of the corresponding global variable is changed to the set value.



CAUTION

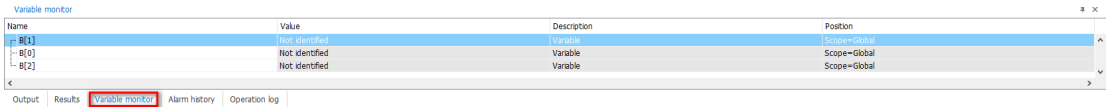
- It is not allowed to set the value of IN, IG, OG, DA, and AD.
- Ensure that the InoRobotLab software platform is connected to the controller.

7.8 Local Monitor

7.8.1 Overview

- Local monitoring is the monitoring of variables during debugging. When the debugging is stopped or finished, the values of monitored variables are refreshed.
- Variables within the local scope can be added to the local monitor. These variables include LP, LB, LR, LD, LPR, and variables customized within the module or function (without keyword "Global"). Local variables can be a single variable or an array.
- Variables within the global scope can also be added to the local monitor.

By default, the variable monitor window is at the bottom of the software, as shown below.



CAUTION

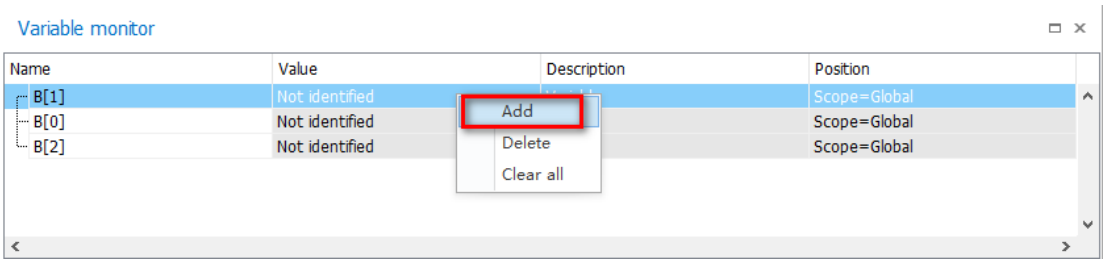
The monitored variable values are refreshed in real time only when the InoRobotLab software is connected to the controller, has the project synchronized, and is in a debugging but not running state. When the project running is stopped, the monitored variable values are refreshed once to ensure that the displayed values correspond to the actual values.

7.8.2 Adding a Monitoring Variable

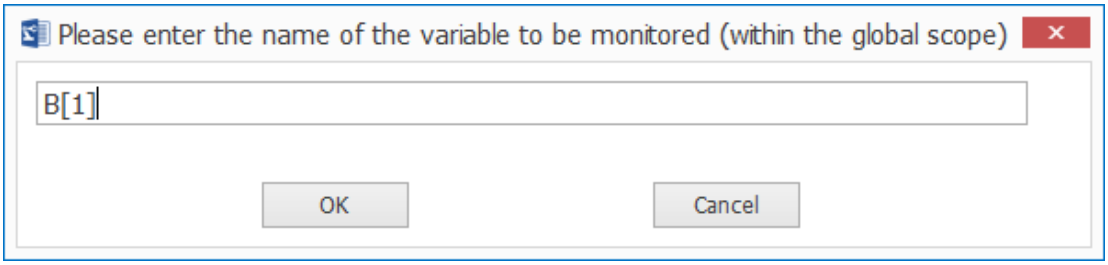
You can add the monitoring variables in any of the following ways.

- Method 1: Adding a variable in the monitoring list

1. Right-click in the monitoring list and select "Add".

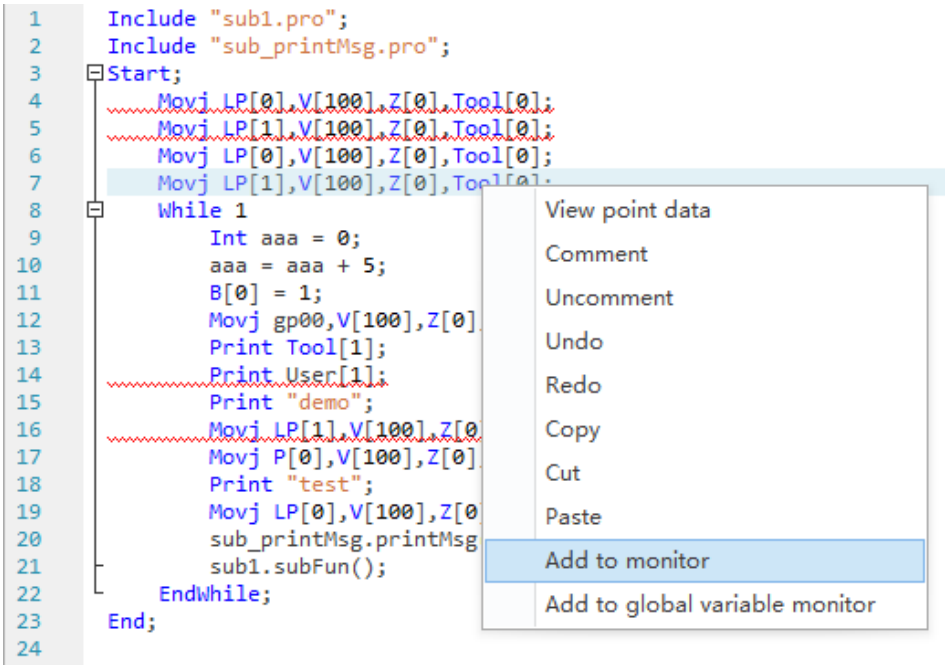


2. In the input box, enter the variable name and click "OK". (Only variables within the global scope can be added in this way. The local variables can only be added through the right-click menu since their scope is uncertain.)



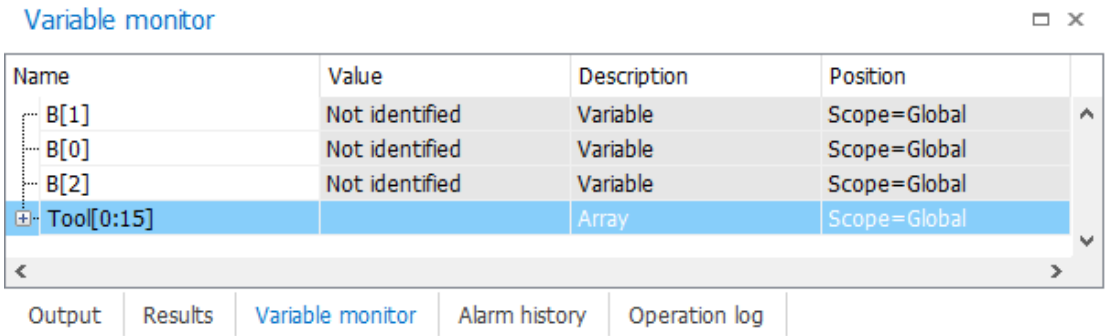
- Method 2: Adding a variable from the program file

After the project is synchronized, right-click the variable and select "Add to monitor".



For rules of adding variables from a file, see the Additional description of "Global Variable Monitor" > "Adding a Monitoring Variable".

After the addition is completed, the added variable appears in the monitoring list.

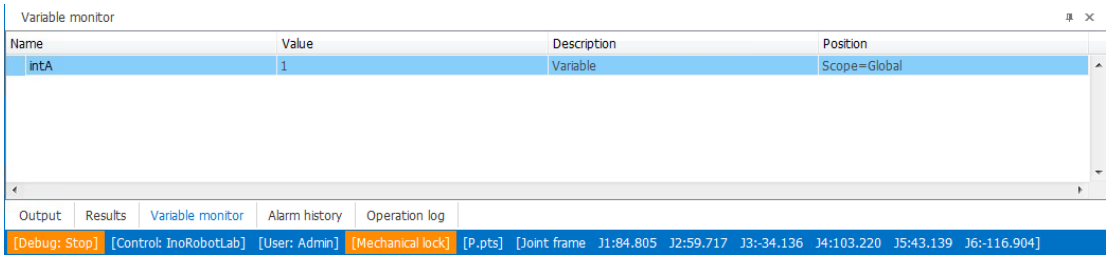


NOTE

- If you add a variable from the program file to the local monitor, you need to synchronize the current project.
- You can add a variable to the local monitor by selecting its label.
- For an array variable, up to 50 elements can be monitored.

7.8.3 Viewing the Monitoring Variables

You can view the variable values only when the program is in debugging and non-running state.



CAUTION

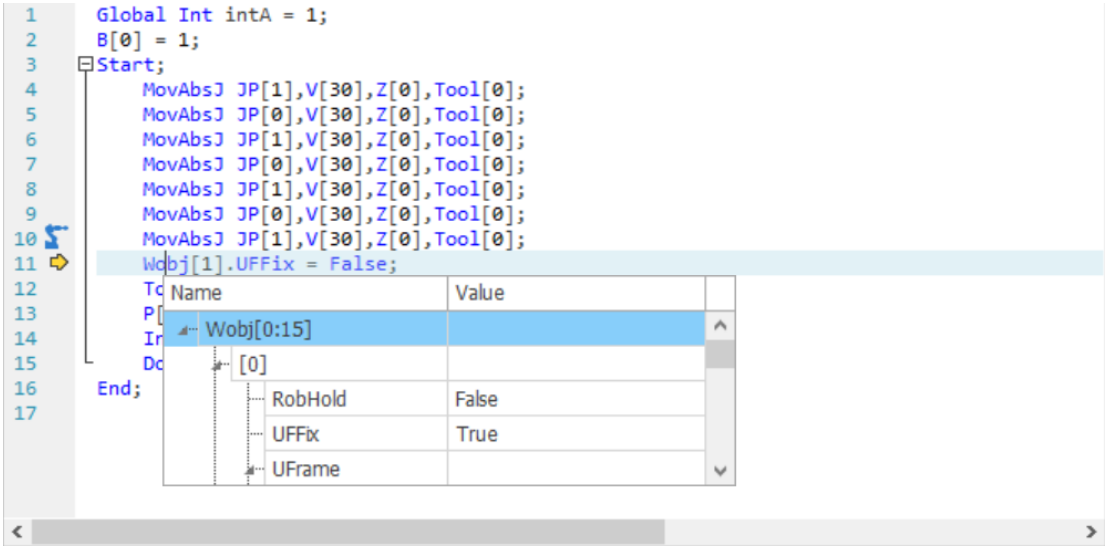
The scope of the local variables includes the task to which the variable belongs, the program file to which the variable belongs, and the function to which the variable belongs. When the program file or function where the yellow cursor (namely, pre-compiling line) is located is out of the scope, the variable value will not be refreshed.

7.9 Monitoring the Variable Using the Mouse

It allows you to view the variable value when debugging the program.

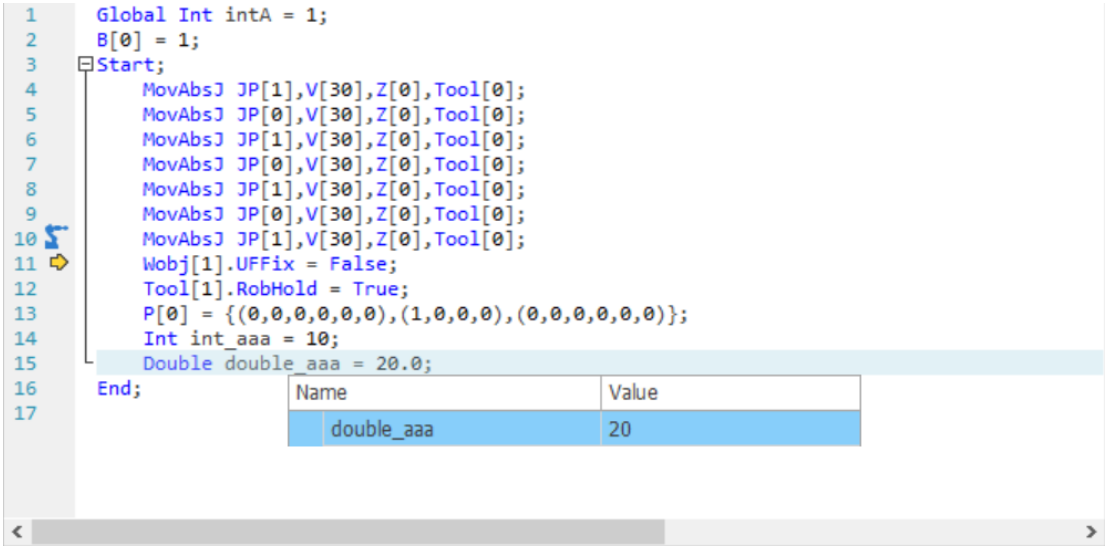
Prerequisite

- Ensure that the controller is connected, the program is in debugging/automatic mode, and the program is stopped or paused.
- Open the program editing page. The yellow cursor pauses/ends in the function. After the mouse is hovered on a variable for more than 0.5s, a pop-up box displays, showing the variable data. When the mouse is moved, the box is closed automatically.
- Variables that can be monitored by the mouse include system variables (Tool, Wobj, P, LP, JP, B, LB, R, LR, D, LD, PR, LPR, etc.) and custom variables (int, double, array, structure, nested structure, etc.)



Results

The pop-up box displays the name of the variable, as well as the value of the variable at the time the cursor is placed on the variable.



CAUTION

- Ensure that the InoRobotLab software platform is connected to the controller, and is in the debugging or auto mode.
- When the program is in a stopped/paused state, you can only monitor the value of the variable on which the mouse cursor is placed.

7.10 Connection Status Monitor

7.10.1 Device Connection Status

You can monitor the status of the system hardware, as shown below.

Name	Status type	Name	Status type
EtherNet1	Network cable not connected	EtherNet2	Network cable not connected
	Dynamic IP: XX.XX.XX.XX or Static IP: XX.XX.XX.XX		Static IP: 192.168.23.25
	Disabled		Disabled
	Undefined		Undefined
	Information acquisition failed		Information acquisition failed
Controller USB	Device not connected	SD card	Device not connected
	Connected and successfully mounted		Connected and successfully mounted
	Connected but failed to mount		Connected but failed to mount
	Undefined		Undefined
	Information acquisition failed		Information acquisition failed
EtherCAT1	Communication normal	IR-Link1	Communication normal
	Slave disconnected		Slave disconnected
	Network cable not connected		Network cable not connected
	Non-EtherCAT device connected		Non-IR-Link device connected
	Disabled		Disabled
	Undefined		Undefined
	Information acquisition failed		Information acquisition failed

NOTE

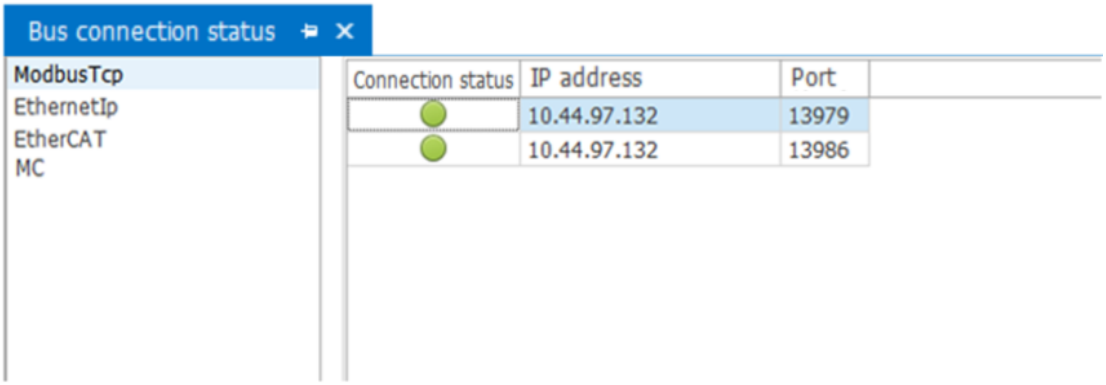
The controller USB drive must be in FAT32, EXT2, or EXT3 format to be successfully loaded.

7.10.2 Bus Connection Status

You can view the connection status of five fieldbuses including Modbus TCP, EtherNet/IP, EtherCAT, MC and PROFINET.

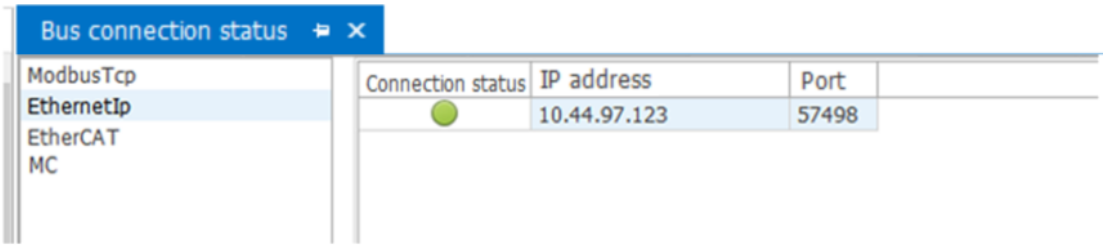
Modbus TCP connection status

- You can monitor the connection status of Modbus TCP, and the IP address and port of the master.
- When there is no master connection, the monitoring list is empty. When there is a master connection, a green dot is displayed.
- Up to 16 Modbus TCP masters can be connected simultaneously.



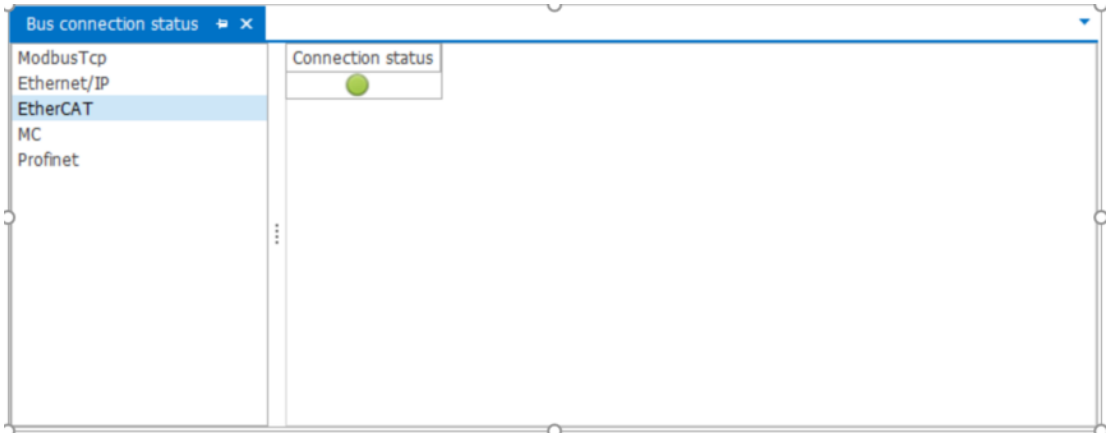
EtherNet/IP connection status

- You can monitor the connection status of EtherNet/IP, and the IP address and port of the master.
- When there is no master connection, the monitoring list is empty. When there is a master connection, a green dot is displayed.
- EtherNet/IP supports only one master connection.



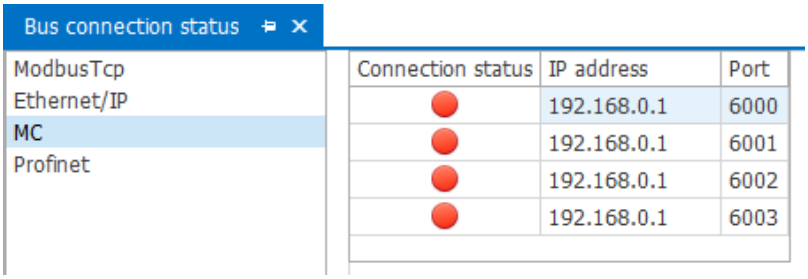
EtherCAT connection status

- You can monitor the connection status of the robot acting as an EtherCAT slave.
- When there is no master connection, the monitoring list is empty. When there is a master connection, a green dot is displayed.
- The EtherCAT function supports only one master connection.



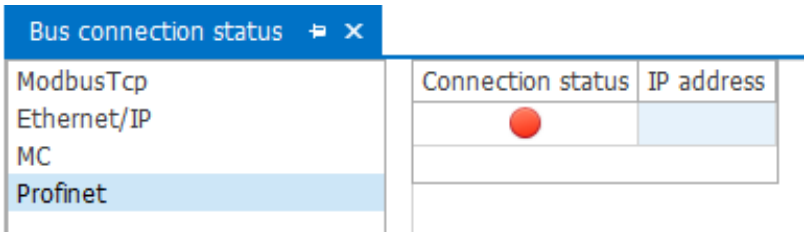
MC connection status

- You can monitor the connection status of the robot acting as an MC slave.
- When the master is not connected, four default master configurations are displayed by default. When the master is connected, a green dot is displayed.
- The MC function supports four master connections.



PROFINET connection status

- You can monitor the connection status of the robot acting as a PROFINET slave.
- When no master is connected, a red dot is displayed and the IP address is empty.
- When there is a master connection, a green dot is displayed, and the IP address bar of the active expansion card is displayed.
- The PROFINET function supports only one master connection.



7.11 Work Condition Monitor

Current protection

By default, the motion parameters of each robot model are factory default. When the motion efficiency on the application site cannot meet the requirements, you can adjust the motion parameters of the robot. For example, you can appropriately increase the running speed and acceleration in the "Controller parameter configuration" page.

When debugging the motion parameters, pay attention to the percentage of the average load rate and of the current.

- When the rate is too high, the robot generates warnings and alarms to protect the motor and reducer.
- When you have adjusted the motion parameters, you can appropriately increase the "global limit coefficient" and "local limit coefficient" to reduce the percentage and avoid warnings or alarms.

> **CAUTION**
>
> Increase the coefficient moderately since excessive increase will reduce the performance and life of the motor and reducer.

Description of parameters of the current protection function:

- $$\frac{\text{Real time value of average load rate}}{\text{Average load rate protection threshold}*\text{Global average load rate limit coefficient}*\text{Local average load rate limit coefficient}} = \text{Average load rate}$$

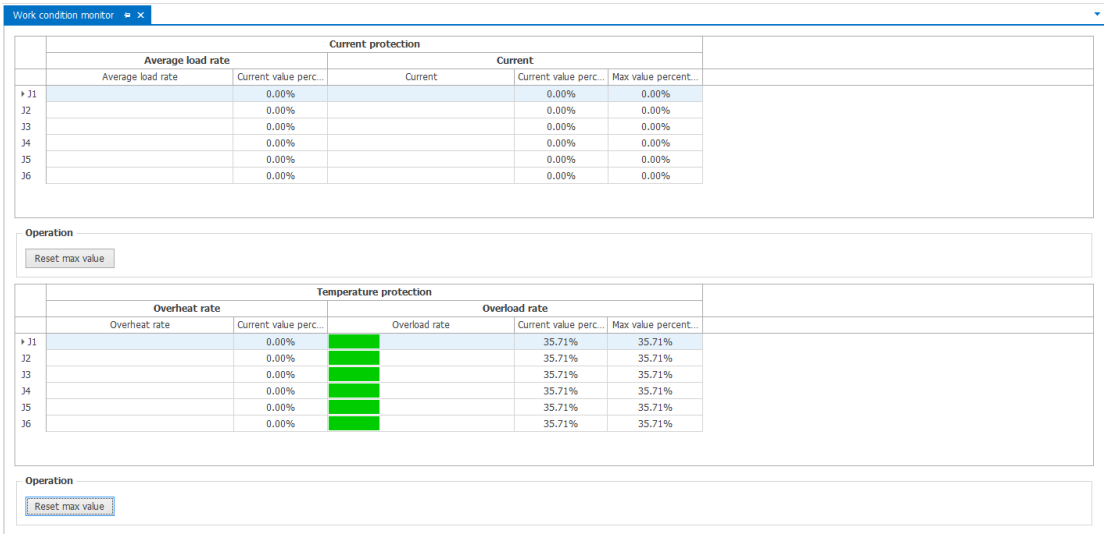
  To set the global average load rate limit coefficient, go to "Controller parameter configuration" > "Motion parameters" > "Axis parameter settings" > "Average load rate limit".
- To set the local average load rate limit coefficient, use the AvgCurLmt instruction.
- To view the real time percentage of the average load rate, go to "Work condition monitor" > "Current protection".
- $$\frac{\text{Real-time current value}}{\text{Current protection threshold}*\text{Global current limit coefficient}*\text{Local current limit coefficient}} = \text{Current rate}$$

  $$\frac{\text{Max current value}}{\text{Current protection threshold}*\text{Global current limit coefficient}*\text{Local current limit coefficient}} = \text{Max current rate}$$

  To set the global current limit coefficient, go to "Controller parameter configuration" > "Motion parameters" > "Axis parameter settings" > "Current limit".
- To set the local current limit coefficient, use the MaxTrqLmt instruction.
- To view the real time percentage of the current, go to "Work condition monitor" > "Current protection".
- To view the real time percentage of the maximum current, go to "Work condition monitor" > "Current protection". During robot motion, the current changes quickly and is not easy to view. Therefore, the maximum percentage is provided to allow users to view the maximum history value.

Method of monitoring and debugging

You can monitor the percentage of the average load rate and of the current in the "Work condition monitor" page.



1. For the current, in addition to the current value percentage, the maximum value percentage is also displayed to record the historical maximum current. You can reset the value by clicking the "Reset" button. The percentage is divided into the following three levels.

Level	Safe	Warning	Danger
Average load rate/Set threshold	0 to 100%	100% to 110%	Above 110%
Current/Set threshold	0 to 10%	110% to 130%	Above 130%

   The above thresholds are applicable after the robot is warmed up.

2. The current detection needs to last for 30 ms. A warning or alarm can be triggered only when the threshold is exceeded for 30 ms consecutively. An alarm will also be triggered when the instantaneous current is too high, at which time the maximum current value may not have exceeded 130%.
3. The average load rate detection needs to last for 30 ms. A warning or alarm can be triggered only when the threshold is exceeded for 30 ms consecutively.
4. Debugging principles:
   - Ensure that the percentage of the average load rate and of the current is in the "Safe" state.
   - When in the "Warning" state, a system warning is triggered, but the robot does not stop. In this case, the current parameters can still be used.
   - When in the "Danger" state, a system alarm is triggered and the robot is disabled and stops. In this case, adjust the motion parameters to return them to the safe range.
   - If you need to adjust motion parameters in applications where efficiency is required, without causing warnings or alarms due to too high average load rate or current, you can appropriately increase the "global limit coefficient" and "local limit coefficient" to reduce the percentage. It should be noted that in this case the performance and life of the motor and reducer are compromised. It is recommended that debugging be carried out under the guidance of the manufacturer.

5. If an alarm such as the average load rate alarm is triggered by a low temperature, warm up the robot first.

   - Setting the limit coefficient

     You can set the global limit coefficient of all axes through the software interface. Also, you can set the local limit coefficient of a certain axis in the program through instruction.

Switch: Only when both the master switch and the sub-switch for a particular axis are switched on, continuous detection of an axis is performed. If the master switch is turned on while the sub-switch for a particular axis is turned off, the detection of the axis is not possible.

Value: The global limit coefficient and local limit coefficient are multiplied together to jointly act on the threshold.

To configure the global limit coefficient through the software interface, go to "Controller parameter configuration" > "Motion parameters" > "Axis parameter settings" > "Current limit/Average load rate limit".

You can configure the local limit coefficient using AvgCurLmt/MaxTrqLmt instructions.

- Configuration through software interface

  In the Manufacturer mode, you can configure the current protection-related parameters through the setting page for "Current limit" and "Average load rate limit".

  Detection switch: Once turned on, all motion detection takes effect; and once turned off, no detection is performed. By default, it is turned on.

  Configuration value: Set the global limit coefficient. The value is in percentage, which is 100% by default.

- Configuration through instructions

  You can dynamically configure a single axis through instructions.

  You can flexibly and conveniently turn off/on single-axis detection and adjust the detection threshold in a certain section of the robot program.

Over-temperature protection

When the robot works in harsh conditions (short cycle time/heavy load/large motion duty cycle) or when the ambient temperature is high, it can easily cause overheating of core components such as the motor and reducer, shortening the service life of the robot. This function provides temperature protection for the robot by predicting the steady state temperature and real-time temperature.

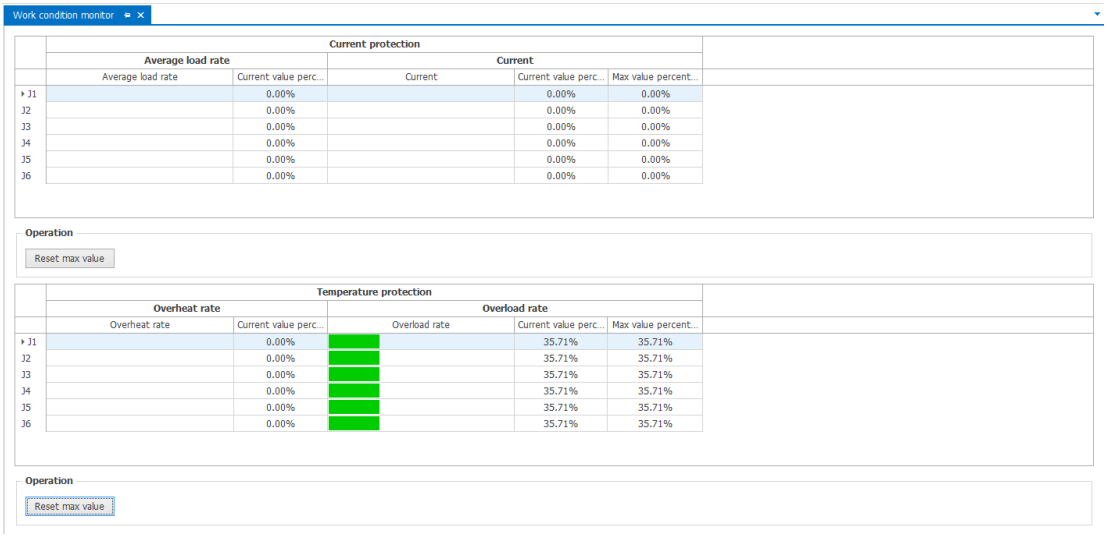
This function is temporarily not supported for the following models in V4R24C3:

IRGF-R10-200S10, IR-R10-200S5, IR-R12-201S7, IR-R35-185S7, HRTA100-10-50Z20TS3, IR-LS20*, IRS13-8-62Z25TS3, IRS111*, IRS112*, IR-TS4* and IR-TS5*.

For models supported by the current version, if the function is blocked after loading a backup file from a previous version, contact the manufacturer for assistance.

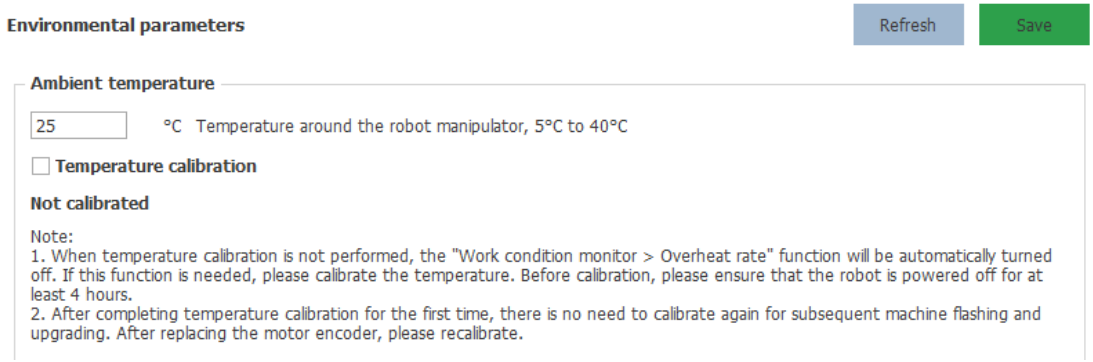
Software Interface and Function Description

- Overheat rate/overload rate



Function	Software interface	Description	Description
Overheat/overload rate monitor	Monitor/Work condition monitor	This function is used to dynamically indicate the overheat rate and overload rate of each axis in a visual bar graph.	1. Parameter definition: $\text{Overload rate} = \frac{\text{Steady-state temperature prediction}}{\text{Temperature alarm threshold}}$ $\text{Overheat rate} = \frac{\text{Real-time temperature prediction}}{\text{Temperature alarm threshold}}$ 2. Alarm (warning) condition: Warnings/alarms are generated when running program instructions. Motor/Reducer overload warning: The overload rate exceeds 90% for 30s continuously. Motor/Reducer overheat warning: The overheat rate exceeds 90% for 10s continuously. Motor/Reducer overheat alarm: The overheat rate exceeds 100% for 10s continuously. 3. In the following situations, the joint axes do not support the overheat rate (overheat rate is 0): 1) The corresponding axis's servo parameter H0004 is 2312.2. 2) The corresponding axis's servo parameter H0B66 is 0. 3) The mechanical lock is enabled. 4) If neither 1) nor 2) are met, investigate software issues, such as whether configuration files are abnormal.

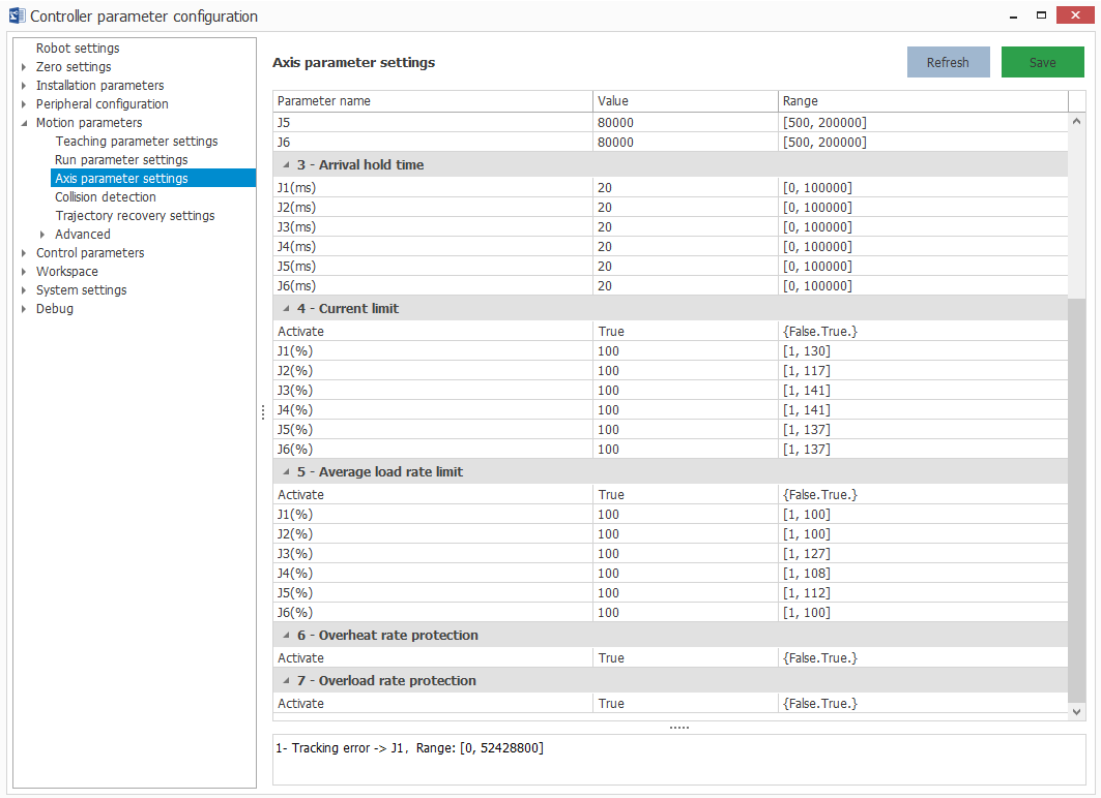
- Ambient parameter settings


Environmental parameters
Refresh
Save
Ambient temperature
25
°C Temperature around the robot manipulator, 5°C to 40°C
Temperature calibration
Not calibrated
Note:
1. When temperature calibration is not performed, the "Work condition monitor > Overheat rate" function will be automatically turned off. If this function is needed, please calibrate the temperature. Before calibration, please ensure that the robot is powered off for at least 4 hours.
2. After completing temperature calibration for the first time, there is no need to calibrate again for subsequent machine flashing and upgrading. After replacing the motor encoder, please recalibrate.

Set the ambient temperature/perform temperature calibration	Go to "Installation parameters" > "Environmental parameters".	Set the ambient temperature/perform temperature calibration	1. The ambient temperature ranges from 5°C to 40°C, and the maximum allowable input range is -40°C to +85°C. 2. The real-time calculation of the overload rate is affected by the set ambient temperature and robot operating conditions. Therefore, setting the ambient temperature correctly facilitates accurate calculation of the overload rate. 3. The real-time calculation of the overheat rate is affected by whether temperature calibration is performed. • If the temperature is not calibrated, the overheat rate will be set to 0 automatically and no overheat warning or alarm will occur. If necessary, ensure that the robot is powered off at least 4 hours before calibration. After the ambient temperature is set correctly, check "Temperature calibration" and click "Save". • After the temperature calibration is done for the first time, there is no need to recalibrate during subsequent robot flashing and upgrading. After the motor encoder is replaced,

Function	Software interface	Description	Description
			recalibration is required.

1. Overheat/Overload rate protection switch



Function	Software interface	Description
Overheat/Overload rate protection switch	Go to "Controller parameter configuration" > "Motion parameters" > "Axis parameter settings".	This function is used to enable and disable the overheat rate/ overload rate warnings/alarms.

2. GetEncoderTemper instruction

Function	Description	Format	Parameter	Return value
Get the current temperature information of the encoder	This instruction is used to obtain the current encoder temperature. 1. In the following situations, the encoder temperature is valid: The servo parameter H0B66 (encoder temperature) is not 0 and H0004 (encoder version) is not 2312.2. 2. This instruction can only be used in robot models that support the temperature rise protection function. Otherwise, the error code -505 will be returned. If the input axis number is incorrect (not between 1 and 8), the error code -404 will be returned. 3. About encoder temperature and related alarms: Encoder overheat alarm: The real-time encoder temperature exceeds the encoder temperature warning threshold for 10s continuously.	GetEncoderTemper (index)	index: Specifies an axis of the robot.	Double type. An alarm occurs when the encoder temperature of the ith axis is wrong.

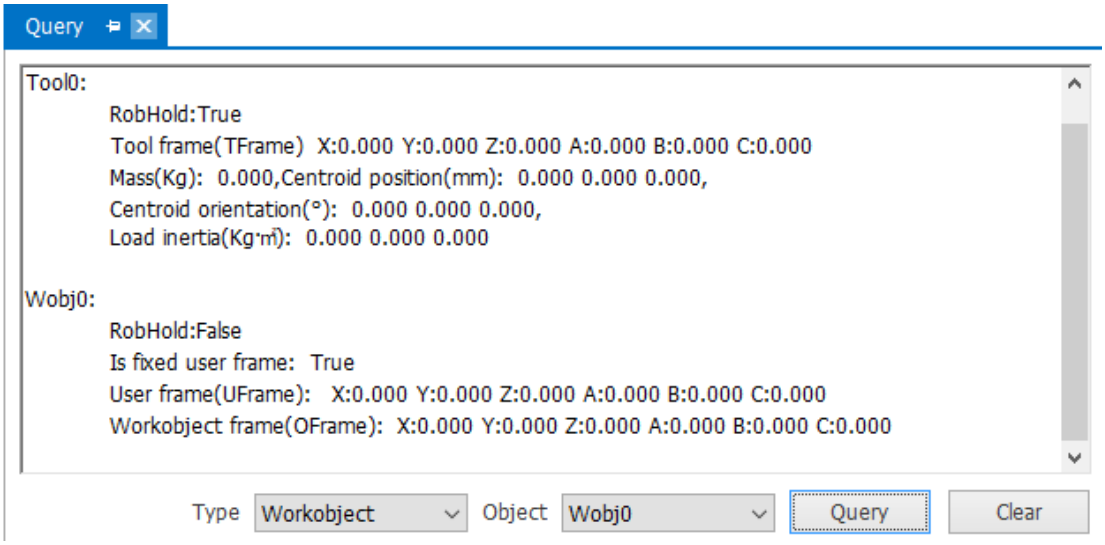
7.12 Monitoring the Load Setting Value

You can query the parameters of various tools, work objects, and loads in the controller.

Prerequisite

Ensure that the controller is connected.

Go to "Monitor" > "Query", select the query type and object, and then click the "Query" button.



Results

The results of the query appear in the dialog box.

8 Monitoring Permissions

8.1 Overview

The control permission is the prerequisite for a certain device to control the controller, thus the robot. The device can control the robot only after obtaining the control authority. Otherwise, the device can only read and monitor the status of the robot. Therefore, configuring the control permission is to configure the operating permission of the controller.

When the control permission is not taken by InoRobotLab, the following two situations may occur.

- Situation 1:

The control permission is taken by remote Ethernet client.

The control permission is taken by remote I/O unit (effective in both auto and manual modes)

The control permission is taken by remote I/O unit (only effective in auto mode), and the mode is Auto.

- Situation 2:

The control permission is taken by remote I/O unit (only effective in auto mode), and the mode is Manual.

The following describes the operations that can be performed through InoRobotLab when the control permission is not owned by InoRobotLab.

Operations		Situation 1	Situation 2
Controller	Connect, disconnect, and IP address modification	✓	✓
-	Back up to local project, back up the current project	✓	✓
Project menu	Open project, recently opened projects	×	×
-	New	×	×
-	Recently opened projects	×	×
-	Save	×	×
-	Save as	×	×
-	About	✓	✓
-	Back	✓	✓
Right-click menu of the project	New project	×	×
-	Open local project	×	×
-	Open controller project	×	×
-	Build	×	×

-	Save	×	×
-	Upgrade project	×	×
-	Sync to controller	×	×
-	Rename	×	×
-	Delete	×	×
-	Export labels	×	×
-	New program file	×	×
-	Add local file	×	×
-	Add controller file	×	×
-	View parameters	×	×
-	New task	×	×
-	Open the folder where the file is located	✓	✓
-	Close	✓	✓
Start menu	New, Open, Save	×	×
-	Save all, Build	×	×
-	Copy, Paste, Cut	×	×
-	Undo, Redo	×	×
-	Comment, Uncomment	×	×
-	Format	×	×
-	Find, Replace	×	×
Right-click menu of the program	View point data	✓	✓
-	Build	×	×
-	Save	×	×
-	Save as	×	×
-	Copy	×	×
-	Paste	×	×
-	Rename	×	×

-	Delete		×	×
-	Move to Common		×	×
-	Set as task		×	×
Global points	Add, Edit, Delete, Get current point		×	×
	Move to point		×	✓
Labels	I/O, AD, BRD label editing		×	×
User alarms	Edit user alarms		×	×
Controller parameter configuration	Robot settings	Structural parameters	×	×
-	-	Reduction ratio	×	×
-	-	Coupling parameters	×	×
-	-	Internal compensation parameters	×	×
-	Motion parameters	Teaching parameter settings	×	×
-	-	Run parameter settings	×	×
-	-	Axis parameter settings	×	×
-	-	Collision detection	×	×
-	-	Trajectory recovery settings	×	×
-	-	Advanced	×	×
-	Zero settings	Absolute zero	×	×
-	-	Work origin	×	×
-	Installation parameters	Tools	×	×
-	-	Workobject	×	×
-	-	Load	×	×

-	-	Arm load	×	×
-	-	Installation form	×	×
-	-	Environmental parameters	×	×
-	Workspace	Axis limit settings	×	×
-	-	Interference zone settings	×	×
-	System settings	System backup and load	×	×
-	-	Safety door-triggered stop mode	×	×
-	-	Robot maintenance	×	×
-	-	Flying trigger I/O	×	×
-	-	Network configuration	×	×
-	-	Time Settings	×	×
-	-	Emergency stop trigger method	×	×
-	-	Emergency stop mode	×	×
-	-	Start button work mode	×	×
-	-	Miscellaneous	×	×
-	-	Control permission	✓	✓
-	Debug	Network debug	×	×
-	-	COM switch	×	×
-	-	System diagnostics	✓	✓
Monitor	I/O monitor	Force switch and state setting	×	✓
	Global variable monitor	Add variable and modify variable	×	✓
Debug	Sync from controller		✓	✓

-	Build	×	×
-	Sync to controller	×	×
-	Back to start line, Start, Stop, Step-into, Exit	×	×
-	Delete all, Enable/Disable all	×	×
Tools	Axis control panel	×	×
-	Project manager	×	×
-	Task manager	×	×
-	Communication manager	×	×
-	Servo debug	×	×
-	Robot model data manager	×	×
-	Robot software upgrade	✓	✓
Control panel	Servo: ON/OFF	×	✓
	Clear alarm	×	✓
-	Switch mode	×	✓
-	Emergency stop	✓	✓

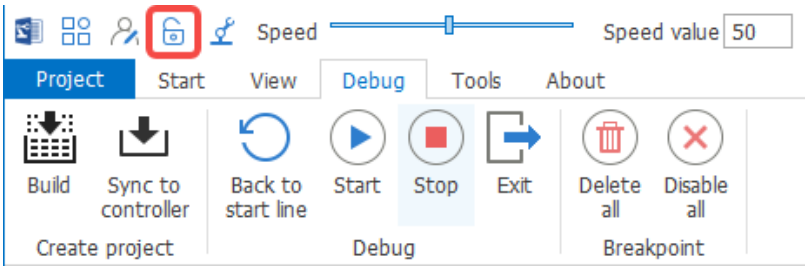
8.2 Setting the Control Permission

Steps

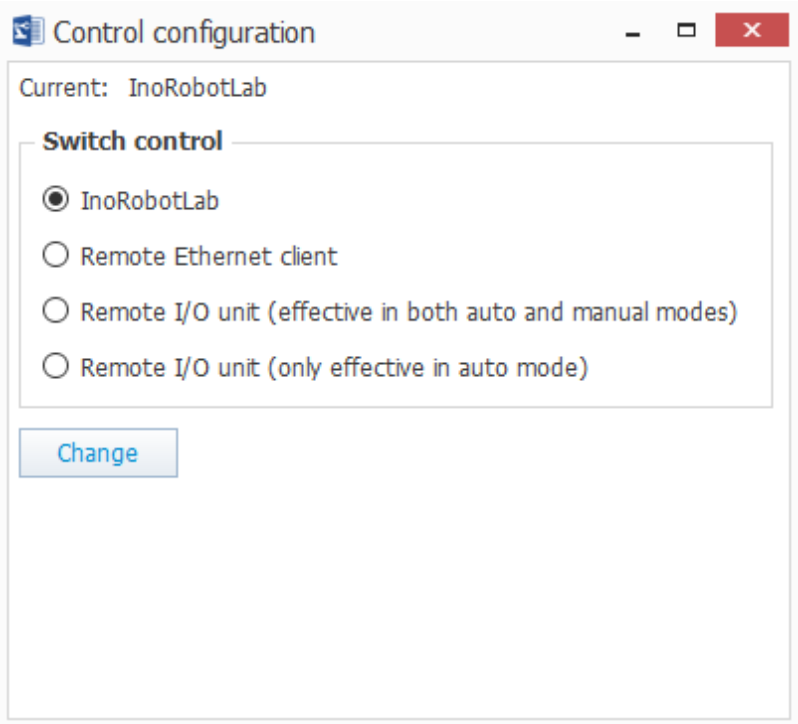
In the InoRobotLab software, you can configure the control permission in either of the following two methods.

- Method 1

On the Quick Access toolbar on the home page, there is a blue lock icon.



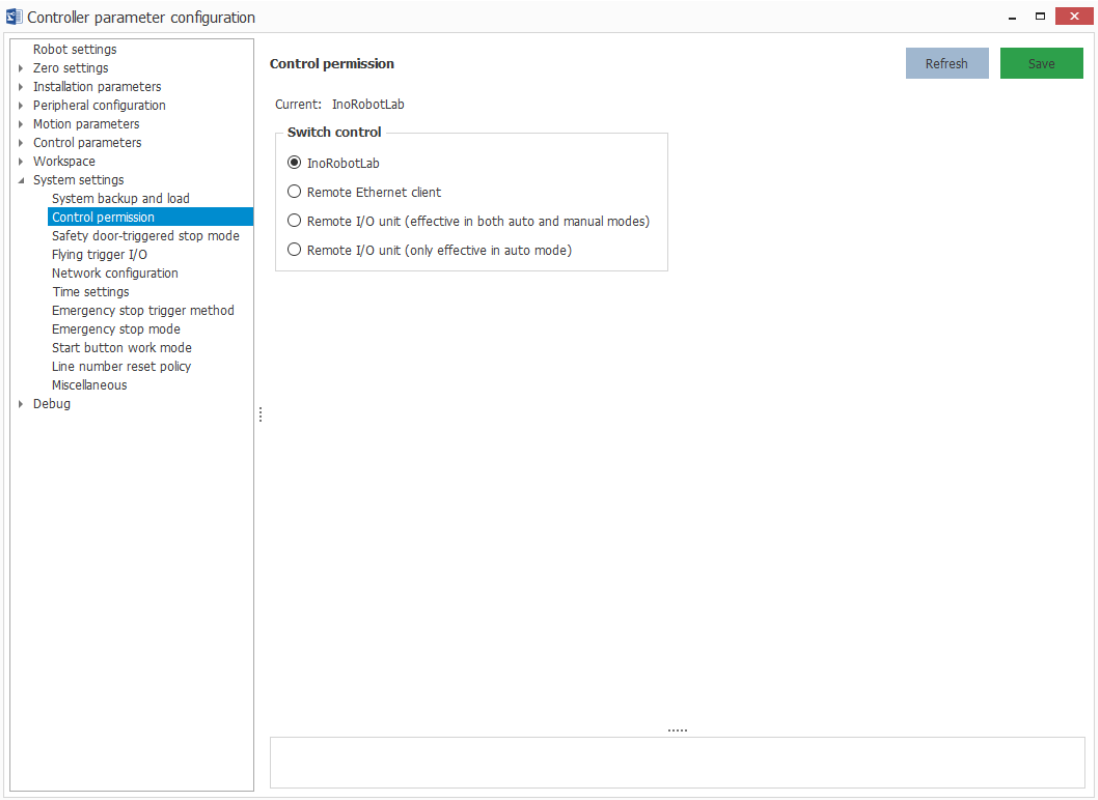
Click this icon to open the control permission setting window.



Select the target control device and click the "Change" button.

- Method 2

Go to "Controller parameter configuration" > "System settings" > "Control permission".



Select the target control device and click the "Save" button.

Refresh: Gets the device which currently takes the control permission.

Save: Saves the control permission settings to the controller.

Results

- InoRobotLab takes the control permission.

When the system control is granted to InoRobotLab, InoRobotLab can operate the controller. Other devices can only read or monitor parameters, and cannot modify parameters or operate the controller.

- The remote Ethernet device takes the control permission.

When the system control is granted to the remote Ethernet device, the remote Ethernet device can operate the controller. Other devices can only read or monitor parameters, and cannot modify parameters or operate the controller.

Up to 4 remote Ethernet devices can be connected to the robot simultaneously. When the system control is switched from another device to an Ethernet device, the four Ethernet devices do not have actual control immediately and need to request control through instructions.

After an Ethernet device has taken the system control, it can operate the controller. The other three devices have no control, and can only read parameters and monitor status. However, they can all forcibly take control away from the current control owner through instructions. When the control granted to the Ethernet device, the default control is on the first Ethernet device each time the devices are powered on.

- The remote I/O device (effective in both auto and manual modes) takes the control permission.

When the system control is granted to the remote I/O device, the remote I/O device can operate the controller regardless of whether it is in manual or auto mode. Other devices can only read or monitor parameters, and cannot modify parameters or operate the controller.

- The remote I/O device (only effective in auto mode) takes the control permission.

When the system control is granted to the remote I/O device, the remote I/O device can operate the controller in auto mode. Other devices can only read or monitor parameters, and cannot modify parameters or operate the controller. When in manual mode, InoRobotLab obtains some control, as shown in the following table.

Function	Description	Note	Authority
Enable/Disable the robot	Click the control panel to enable or disable the robot	Line number is not changed.	User, Administrator, and Manufacturer
Modify the global variables	Go to "Station" > "Monitor" > "Global variable monitor".	The line number is not changed after the global variables are changed.	User, Administrator, and Manufacturer
Modify the global points	Go to "Station" > "Monitor" > "Global variable monitor".	The line number is not changed after the global points are changed (only the memory points are modified here, and the modified points disappear after restart).	User, Administrator, and Manufacturer
Modify the I/Os	Go to "Station" > "Monitor" > "I/O monitor".	Modifying the I/Os while the program is running does not change the line number.	User, Administrator, and Manufacturer
Axis control panel	Function	The following functions do not reset the line number.	User, Administrator, and Manufacturer
	Speed settings	-	

Function	Description	Note	Authority
	Mechanical unit settings	-	
	Axis motion control	-	
	Mechanical lock	The robot needs to be disabled.	
	Dynamic brake/Singularity crossing	The robot needs to be disabled.	
	Homing	Homing without trajectory recovery	
	Independent axis reset	-	
	Orientation calibration	-	
	Change motion mode	-	
	Coordination function	-	
	Move to point (Movl, Movj, MovAbsJ)	Without trajectory recovery	
	Display the current position	-	
Axis control panel > Coordination function	Turn on/off coordination function	-	User, Administrator, and Manufacturer
Global speed modification	Top of the panel	The line number is not affected.	User, Administrator, and Manufacturer
Control permission switch (Remote I/O unit (only effective in auto mode) <-> Remote I/O unit (effective in both manual and auto modes))	Top of the panel	The control permission can be switched between the two devices without the need to switch back to teach pendant or InoRobotLab.	User, Administrator, and Manufacturer

However, the remote I/O unit that takes the control can only read or monitor parameters, and cannot modify parameters or operate the controller. After switching to the auto mode, InoRobotLab loses the functions described in the above table. The remote I/O unit is given the permission to modify parameters or operate the controller again. For details, see the "Differences between Two Remote I/O Unit Control Modes" section in Remote I/O Control Function User Guide.

NOTE

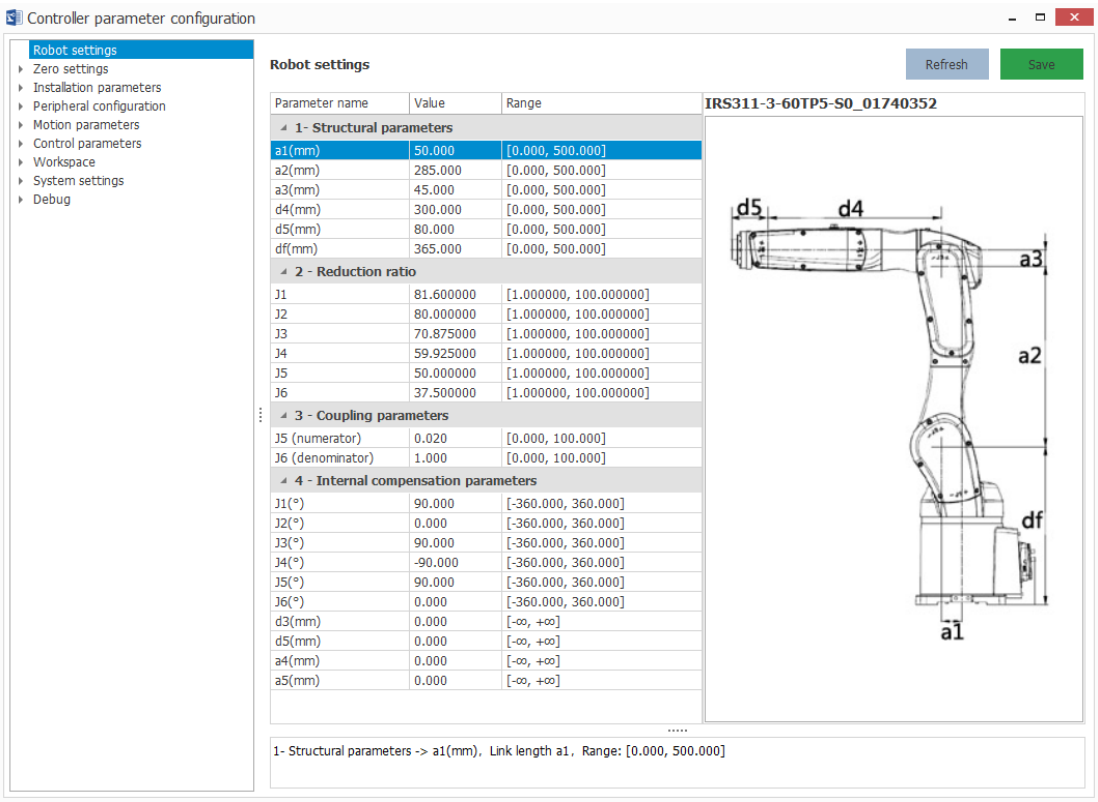
- Modifying the control permission requires the user authority and above, and that the robot must be in a non-moving state.
- On the PC platform, the control permission is granted to InoRobotLab by default.
- The factory default control permission of the system is granted to the teach pendant. When other devices (remote Ethernet device, I/O device) need to control the robot, the control authority must be granted to the corresponding device via the teach pendant.
- The robot controller does not need to be restarted after the control permission is switched. The hardware emergency stop switch is always active regardless of the type of control device.
- After S03.21R, the remote Modbus unit control has been merged into the remote I/O unit control.
- After V4R24C2, the remote I/O unit control is divided into remote I/O (effective in both manual and auto modes) and remote I/O unit control (only effective in auto mode). The remote I/O unit control (effective in both manual and auto modes) is consistent with that in software versions prior to V4R24C2. For the new remote I/O unit control (only effective in auto mode), the control permission of the remote I/O unit is deprived when in manual mode, retaining only the monitoring function, and InoRobotLab is provided with some control functions. In the auto mode, the remote I/O unit has full control while InoRobotLab has only some functions.
- When switching remote I/O control authority, a start speed can be set. This speed will take effect when switching remote I/O and when restarting under remote I/O control.

9 Controller Parameter Configuration

9.1 Robot Settings

The robot settings include structural parameters, deceleration ratio, coupling parameters, and internal compensation parameters. These parameters are related to the mechanical and kinematic models of the robot manipulator and have a decisive influence on the end effector accuracy of the robot. If you need to modify these parameters, please contact the manufacturer for help.

Double-click the "Controller parameter configuration" in the device tree. By default, the robot settings interface is displayed, as shown in the figure below:

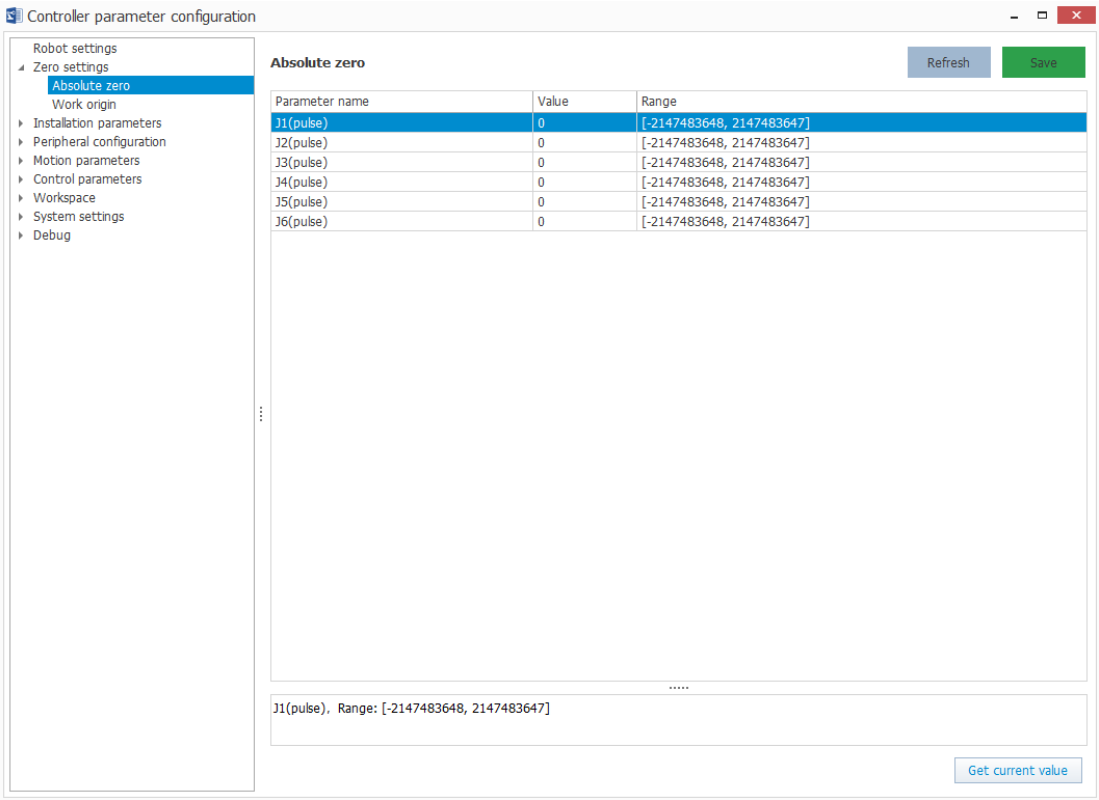


9.2 Zero Settings

9.2.1 Absolute Zero

The robot manipulator is mechanically calibrated to obtain an initial state at the factory, and the encoder position feedback value at the initial state is set to absolute zero. The servo system of the robot uses an absolute encoder. The position feedback value of the encoder is not affected by power-on or power-off of the robot controller. Therefore, it is possible to set mechanical limits in hardware and define the range of motion in software with absolute zero as a reference. Therefore, under normal conditions, the absolute zero point should not be changed after delivery. If you need to modify the absolute zero, please contact the manufacturer.

Double-click "Controller parameter configuration" in the device tree, then select "Zero settings" > "Absolute zero" in the opened interface for related settings, as shown in the figure below:



Enter the encoder position of the respective axes. Save the settings and you can verify the absolute zero by programming.

NOTE

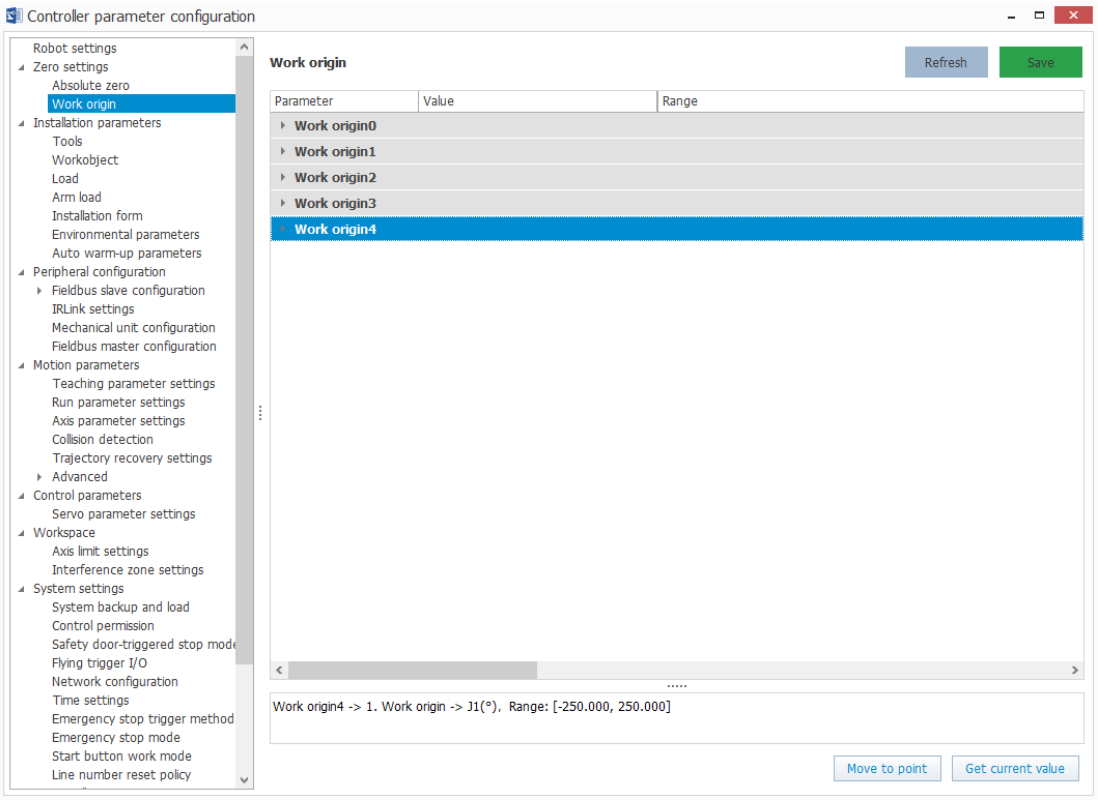
Only administrator and manufacturer can configure the absolute zero. The settings can be saved only when the robot is in the emergency stop state.

9.2.2 Work Origin

9.2.2.1 Setting the Work Origin

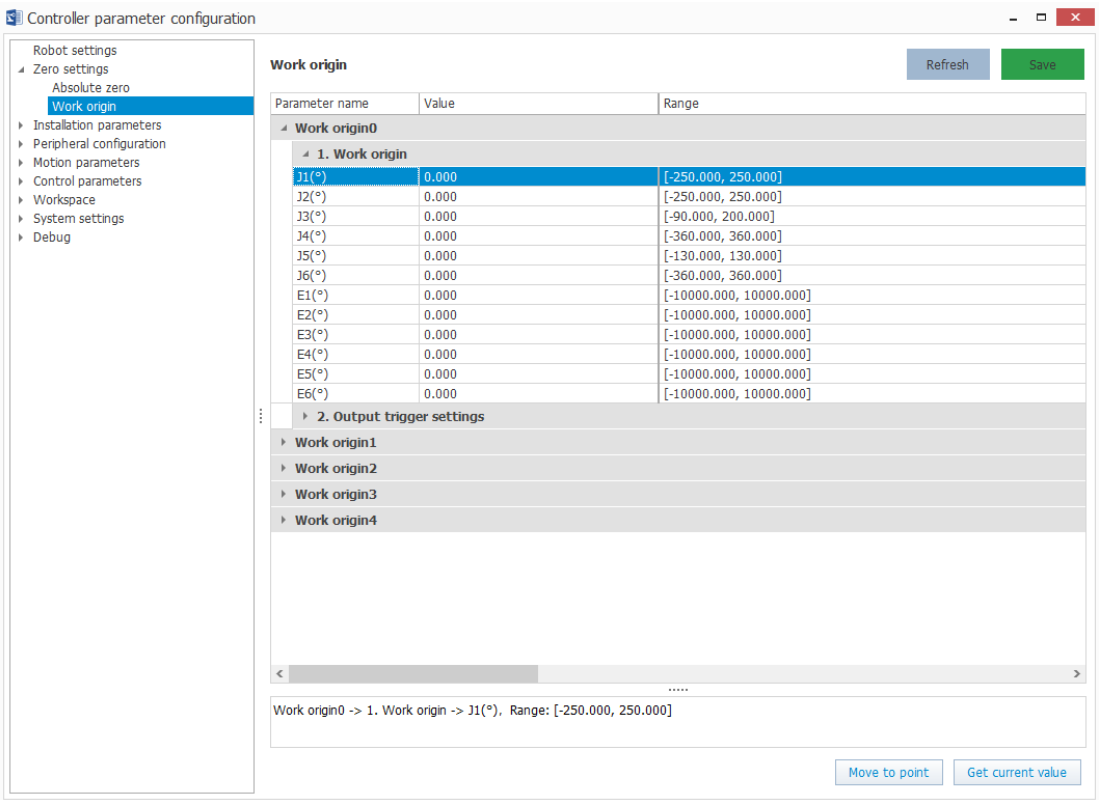
The work home is a user-defined position reference point. The robot supports five work origins.

1. Double-click "Controller parameter configuration" in the device tree, then select "Zero settings" > "Work origin" > "Work origin (0-4)" in the opened interface, as shown in the figure below:



Work origin number (0-4)

2. Click the first second-level tab "Work origin" under "Work origin (0-4)" for related settings. Enter the target angle in the work origin that needs to be set, or set it by getting the current value, as shown in the figure below:



Work origin

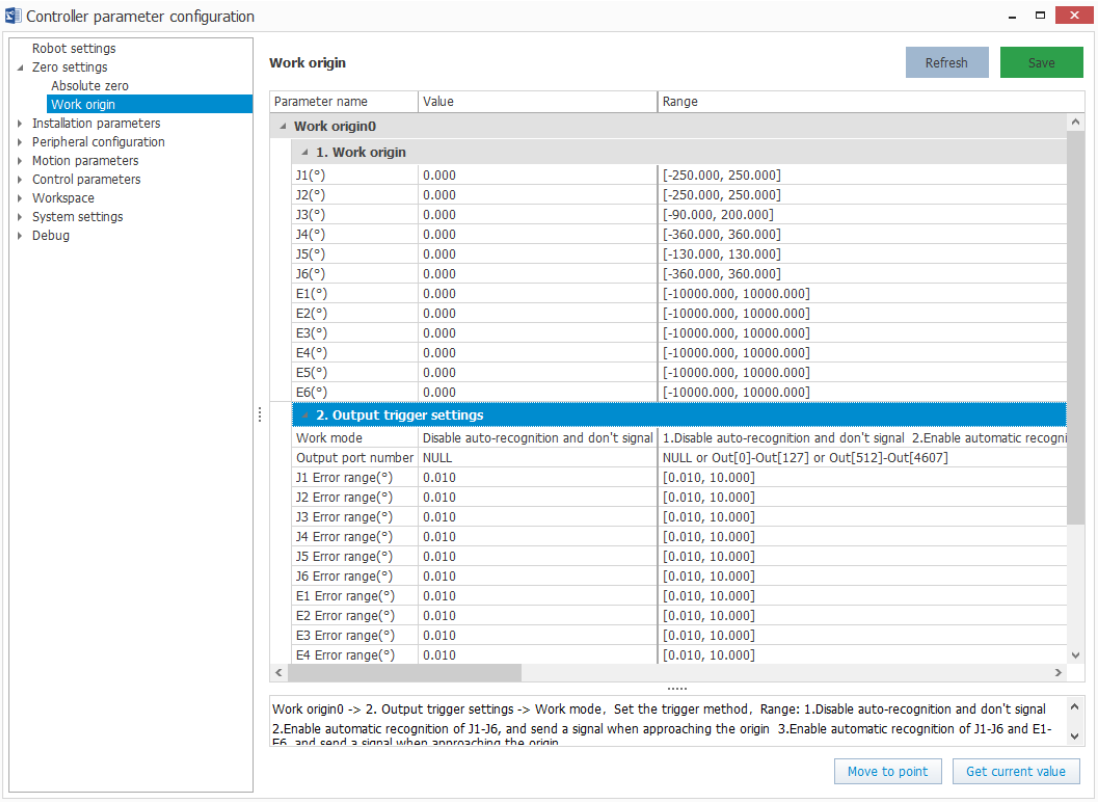
9.2.2.2 Triggering Signal Output when Approaching Work Origin

Function description

When the robot approaches a certain work origin, a signal is outputted.

Steps

1. Double-click "Controller parameter configuration" in the device tree, then select "Zero settings" > "Work origin" > "Work origin (0-4)" in the opened interface.



2. Perform related settings in the second-level tab "Output trigger settings" under "Work origin (0-4)".

Meaning of output trigger parameters:

- Work mode: There are three work modes, including:
  1. Disable auto-recognition and don't signal: This function is disabled.
  2. Enable automatic recognition of J1-J6, and send a signal when approaching the origin: It only detects whether the manipulator axes fall within the error range of the origin. If yes, the corresponding Out signal is set to ON.
  3. Enable automatic recognition of J1-J6 and E1-E6, and send a signal when approaching the origin: When both the manipulator axes and external axes fall within the error range of the origin, the corresponding Out signal is set to ON.
- Output port number: Specifies the number of the Out signal. The range is: NULL or Out[0]-Out[127] or Out[512]-Out[4607].
- Error range of the manipulator's axes and external axes: The error range of each axis is [0.01, 10].

CAUTION

- All types of InoRobotLab users have access to this function.
- This function does not support control via bus and API.
- When an axis is set as an independent axis, the motion of the axis is controlled only by independent axis instructions (IndCMove, IndReset, IndSpeed). Motion instructions using the Home point function will not take effect on the axis.
- If an Out port number is set in this function, and the work mode is set to mode 1 or mode 3, an alarm will be triggered when another function attempts to set this port number. Clear the alarm and set a new Out port number.
- If the set Out port number is used by other functions, the alarm will also be triggered. Clear the alarm and set an unused Out port number.
- If the Home instruction is used in the project file and the parameters related to the Home instruction are modified after running is suspended, the modification will not trigger a recompilation. If the modified line is before the running line or after the start line, the start line number and running line number are not affected. If the modified line is between the running line and the start line, the start line number and running line number are not affected, but the robot still moves according to the original planned trajectory during the actual execution of the program. This means that the modified Home instruction-related parameters will not take effect until the next execution of the instruction.

9.2.3 Homing Calibration (Zero Point Calibration)

This function is available only on some SCARA robot models (the SCARA 20 kg model does not support zero point calibration, and the ceiling-mounted models only support zero point calibration in the test mode), and on some 6-axis robot models (R7H model).

The zero point calibration function is developed to solve the problem of zero point loss.

CAUTION

1. SCARA robot

All axes use automated collision stops for zero point calibration. The stops for the J1 and J2 axes are hardware limit stops, while the J3 and J4 axes require additional calibration jig. In addition, due to the structure of the robot, the home operation for the J3 and J4 axes must be done at the same time.

2. 6-Axis robot

On the supported R7H models, zero point calibration for J1, J2, and J3 axes is performed by colliding a positioning pin against a torque sensor. For the J4 and J5 axes, the method involves colliding one positioning pin against another. For the J6 axis, zero point calibration is achieved by colliding a dedicated calibration fixture against a positioning pin.

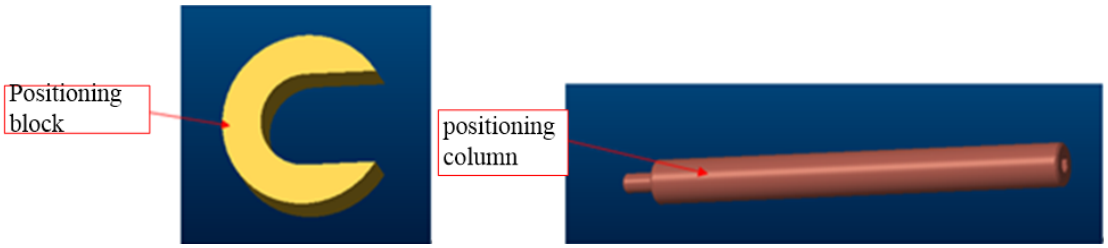
9.2.3.1 Zero Point Calibration for SCARA Robot

9.2.3.1.1 Preparation

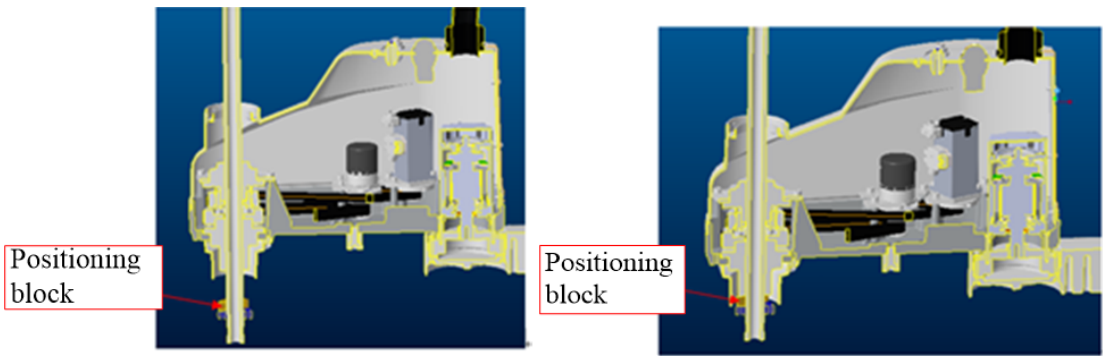
1. Reference position: The zero point calibration of all axes depends on the reference position. The reference positions of J1 and J2 axes are the positive and negative hardware limit stops, and the reference positions of J3 and J4 axes are limit rings. Before the calibration, ensure that the positive and negative hardware limit stops on J1 and J2, and the limit ring on J3 have not been changed from the factory settings.
2. Before calibrating the J3 and J4 axes, install the calibration fixture first.
3. Remove the fixture to avoid entanglement of the air tubes etc., which otherwise may lead to mistaken assumption that the reference positions are reached.
4. Remove any objects between the current position and the positive limit to ensure that there is no interference.

Installation of the zero point calibration fixture for J3 and J4:

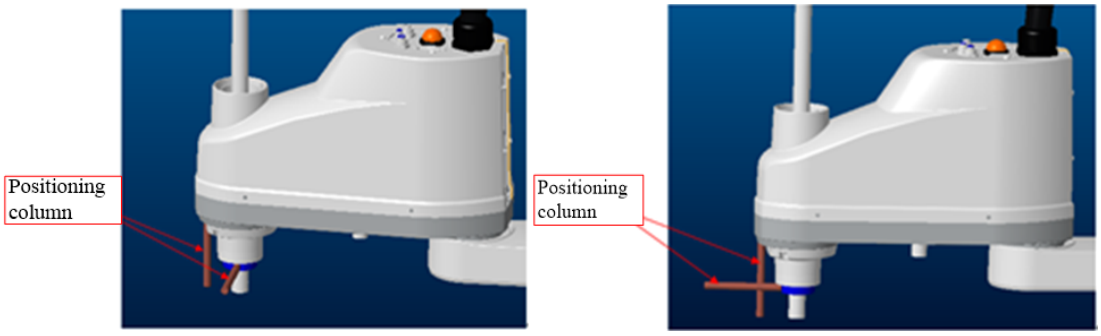
Two types of fixtures, positioning block and positioning column are required, as shown below.



Release the brake of the J3 axis, run the lead screw to a proper position where the positioning block can be easily stuck on the lead screw, as shown below. Then move the J3 axis slowly in the forward direction by hand. When the positioning block hits the spline nut or the spline nut housing, the zero point of the J3 axis is reached, as shown on the right in the following figure.

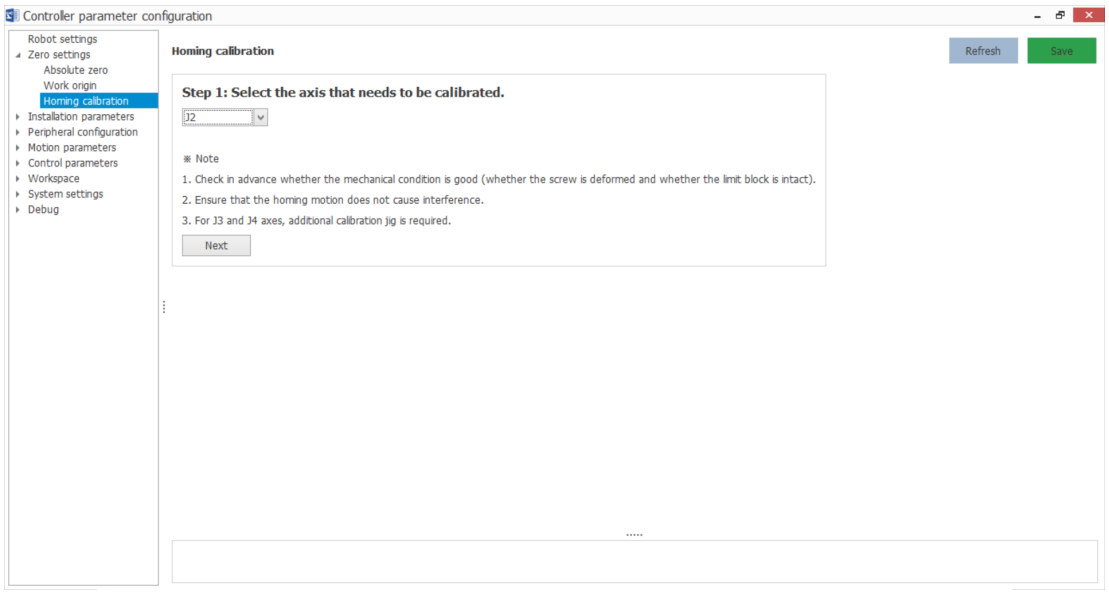


Fix the positioning columns on the forearm and the limit rings, as shown on the left in the following figure. (Ensure that the two columns do not collide with each other to leave space for zero point calibration of J3 and J4.)



9.2.3.1.2 Calibration Steps

1. Log in to InoRobotLab as administrator or manufacturer.
2. Open the homing page.
   Controller parameter configuration > Zero settings > Homing calibration
3. Select an axis.
   Select an axis for calibration from the drop-down menu, and click "Next".

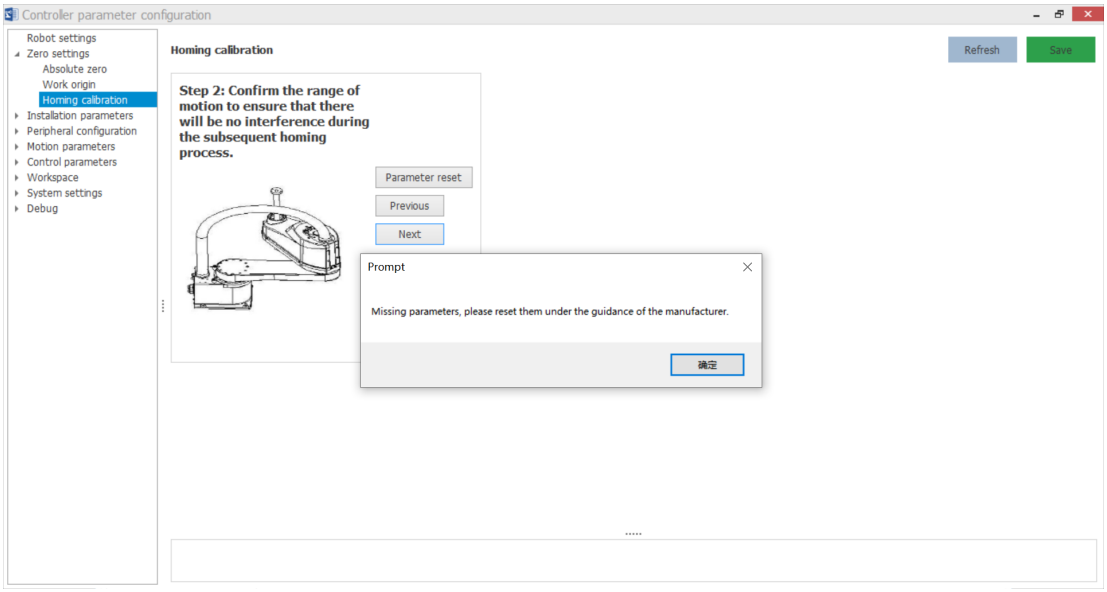


> **CAUTION**
>
> The calibration motion direction of all axes is positive.

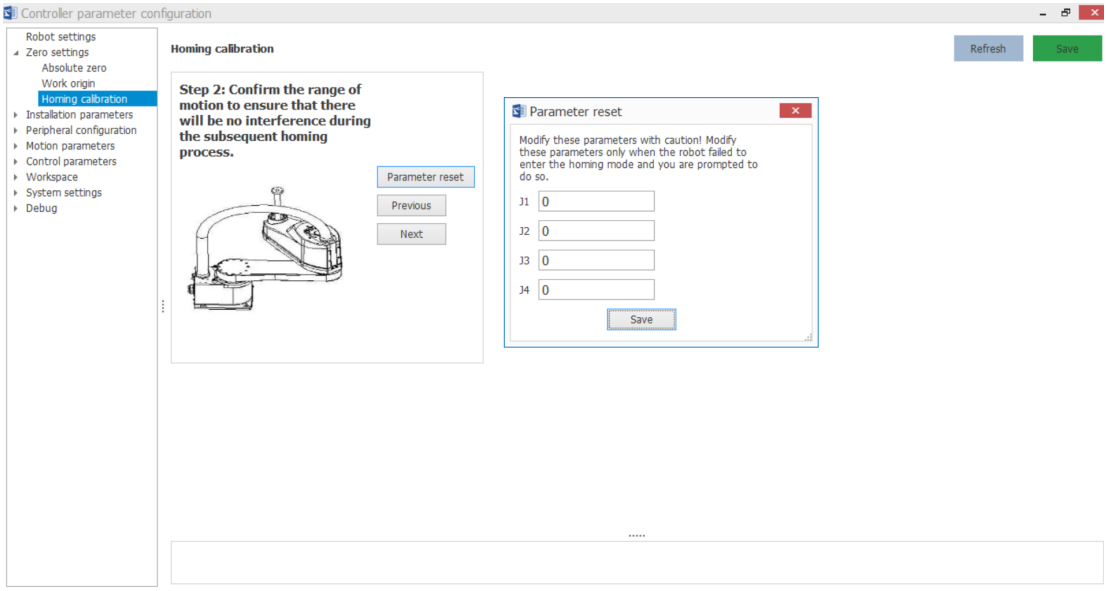
4. Check the zero point calibration environment.
   Ensure that no motion interference occurs during the process of zero point calibration. Then, click "Next".

   In the following two situations, it is impossible to enter the zero point calibration mode, and a message will be displayed: "Missing parameters, please reset parameters under the manufacturer's guidance."

   Situation 1: For robot systems that have been manufactured before S03.21R
   Situation 2: For robot systems that have been manufactured after S03.21R and that have lost zero point calibration parameters due to flashing



- The zero point calibration function is not available for the robot systems that have been manufactured before S03.21R. (The manufacturer only keeps the parameters of the new robot systems.)
- If the robot systems have been manufactured after S03.21R, contact the manufacturer to reset the parameters. (Ask for the parameters from the manufacturer, enter the parameters for the corresponding axis in the "Parameter reset" dialog and click "OK" and save the configuration. Then you can continue with the zero point calibration function.)

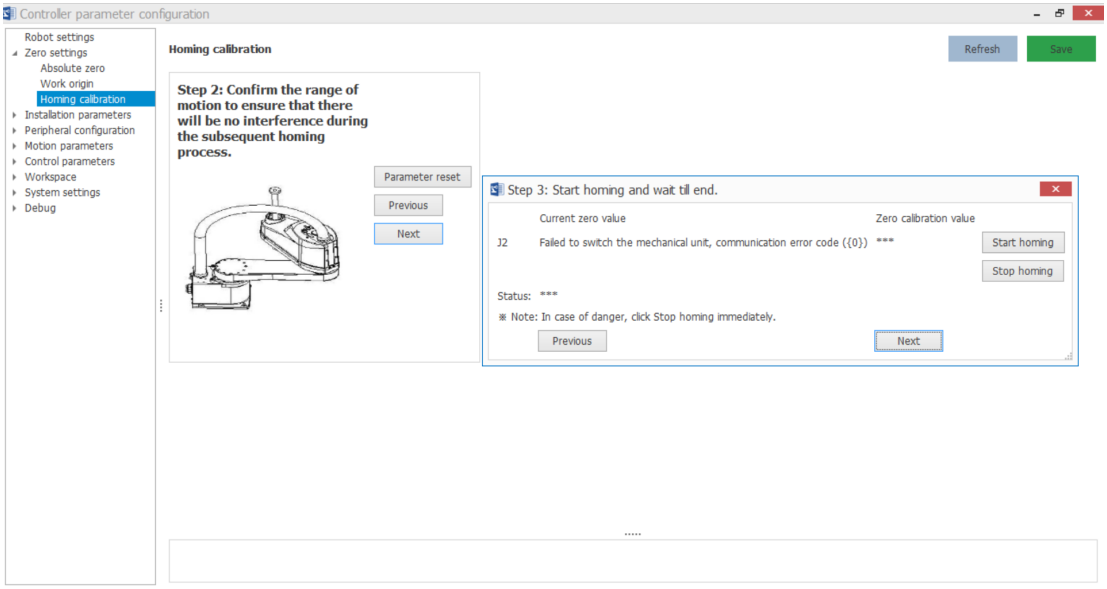


- If you cannot determine whether the robot system is manufactured before or after S03.21R, contact the manufacturer to check if the parameters are kept.

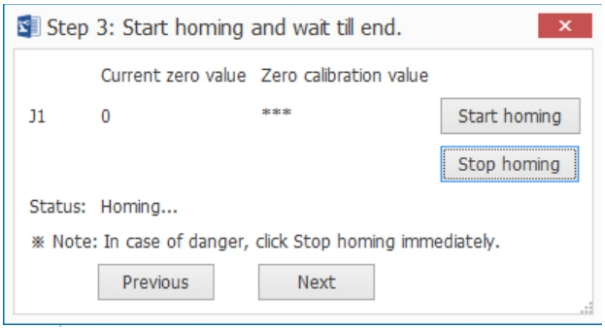
5. Perform the homing calibration.
   For example, select J2+ axis and start homing. The robot is enabled and starts moving. Beware of safety.

CAUTION

- To ensure accuracy, the homing calibration is slow and may take several minutes. Please wait patiently for the status feedback.
- (If you want to save time, you can manually teach the robot to be around 10° away from the hardware limit before starting zero point calibration.)

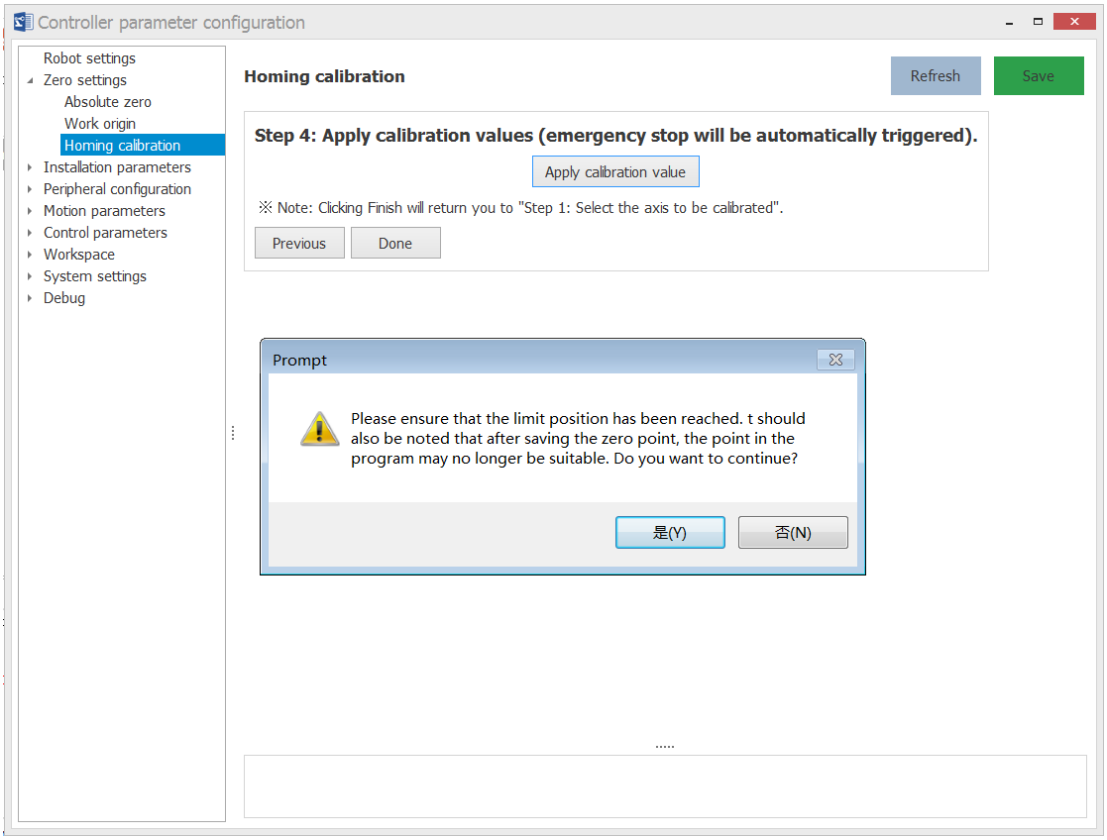


When the calibration is complete, a message "Homing successful" is displayed and the calibration value is updated. See the figure below:

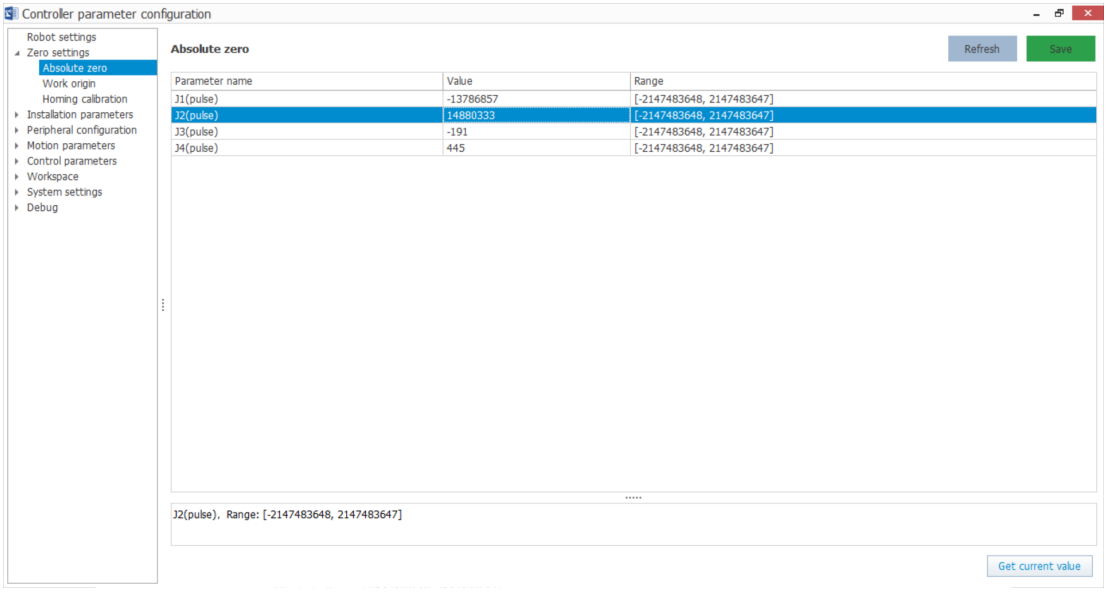


6. Update the zero point.

Click "Apply calibration value". The system triggers emergency stop automatically and displays a pop-up message. See the figure below:



Click "Yes". The current zero point is replaced by the zero calibration value. (Click "Refresh" to update the display.)



7. Check the zero point.
   After updating the system zero point, check the result.

   Recommended method: Teach a point, change the left and right hand parameters, and then observe the deviation of the center of the lead screw moving to that point.

9.2.3.1.3 Notes

- To perform the zero point calibration for J3 and J4 axes, install the calibration fixture first.
- Before the calibration, ensure that the positive and negative hardware limit stops on J1 and J2, and the limit ring on J3 have not been changed from the factory settings. The calibration is accurate only if they are the same with factory defaults.
- Remove the fixture to avoid entanglement of the air tubes etc., which otherwise may lead to mistaken assumption that the reference positions are reached.
- The calibration motion direction of all axes is positive. Remove any objects between the zero point and the positive limit to ensure that there is no interference.
- Zero point calibration is for a single axis. To calibrate multiple axes, perform the calibration sequentially.
- If the software prompts "Missing parameters", reset parameters under the manufacturer's guidance.
- If the multi-turn value of the encoder exceeds 2000, a warning pops up and emergency stop is triggered. In this case, you need to restart the controller and perform the zero point calibration operation again.
- After updating the system zero point, check the result.

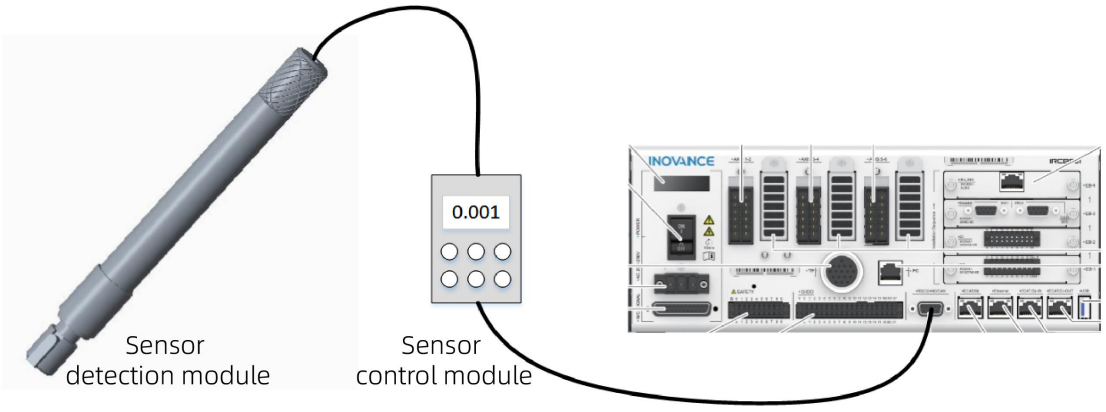
9.2.3.2 Zero Point Calibration for 6-Axis Robot

9.2.3.2.1 Preparation

Fixture and jig for zero point calibration:

			
Key pin	Round pin	J6 axis calibration fixture	Sensor detection module

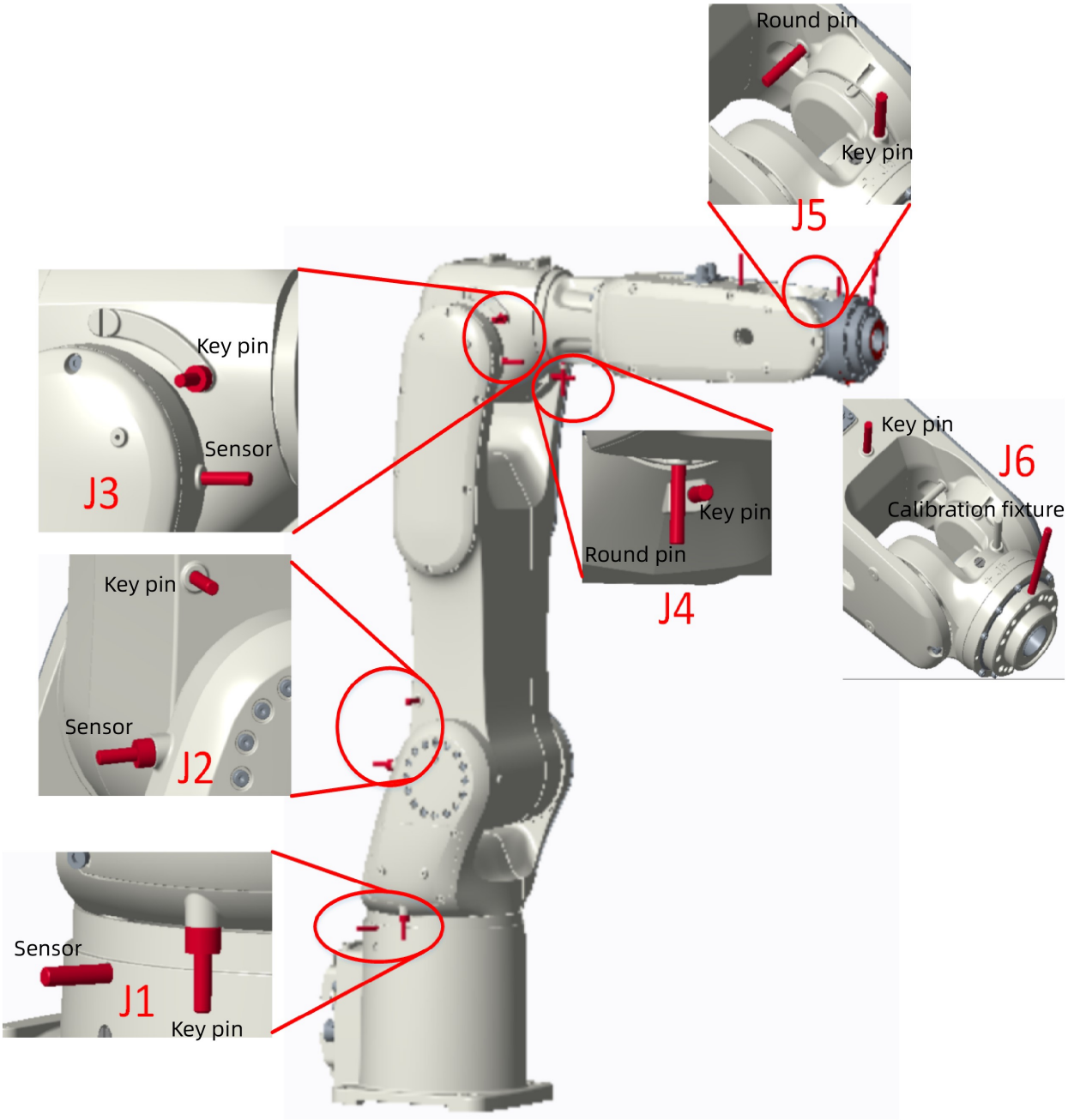
Sensor connection: As shown in the figure below, complete the connection of the sensor detection module to the control module, and complete the connection of the sensor control module to the RS232/485 interface of the robot controller. After connection is completed, start the sensor control module, gently bend the head of the detection end of the sensor, and check if the reading on the sensor control module changes. If it changes, the sensor is normal; otherwise, it indicates sensor damage, requiring replacement with a normal sensor.



Installation of sensor, positioning pin, and calibration fixture: When the zero point calibration process reaches the interface demonstrating the collision mode and prompts you to complete the collision tool installation, install the corresponding sensor, positioning pin, or calibration fixture for the target axis to be calibrated, following the installation method illustrated in the figure below. Installation methods vary by each robot model. Refer to the zero point calibration section in the corresponding "Robot user guide - manipulator" for specific installation instructions.

CAUTION

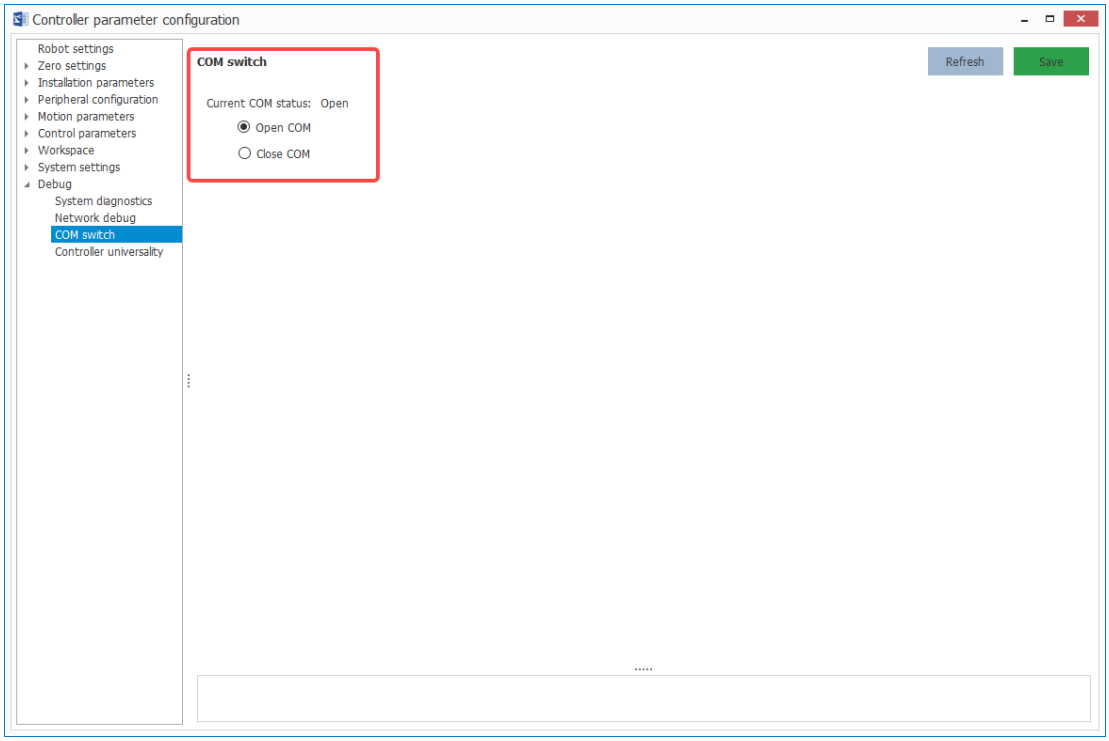
Ensure installation position accuracy and ensure that the sensor/positioning pin/calibration fixture is fully inserted. If the installation position is incorrect, e.g., using the key pin position for inserting a sensor, it will affect calibration accuracy.



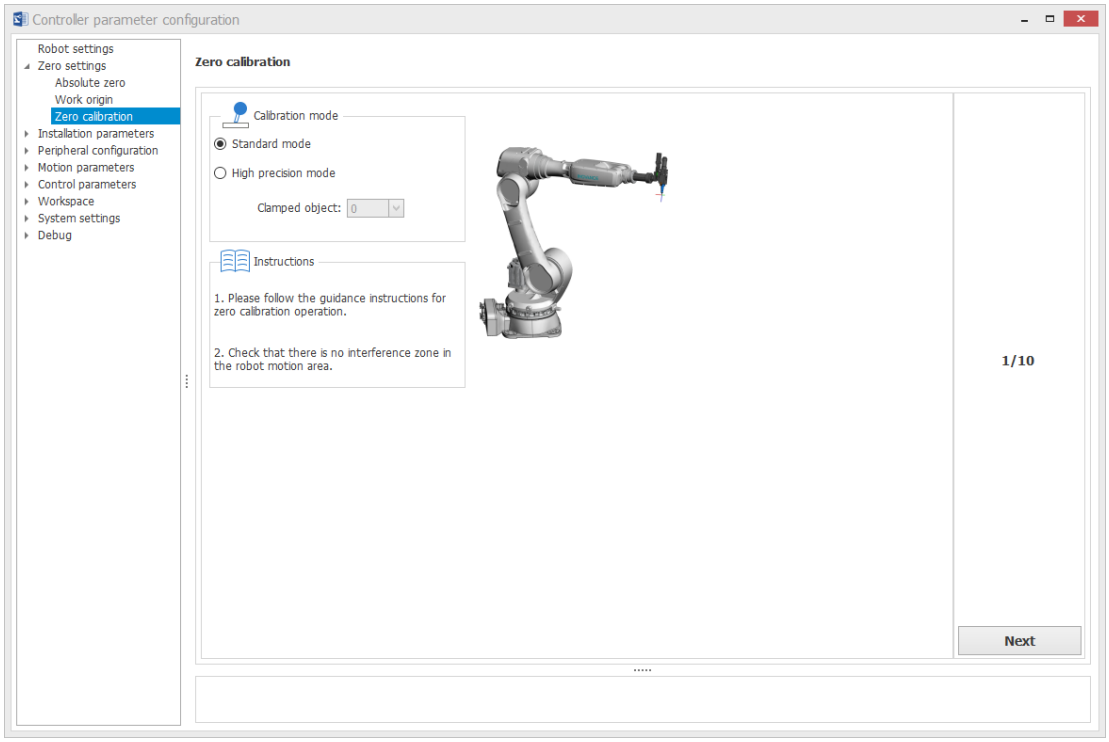
9.2.3.2.2 Calibration Steps

1. Open the serial port: Double-click "Controller parameter configuration" in the device tree, then select "Debug" > "COM switch" to open the serial port. Then, click "Save" and restart the controller.

Opening the serial port communication is for sensor connection. If the positioning pin collision mode is used for the calibration of the target axis, this step can be skipped.



2. **Enter the zero point calibration Interface:** On the "Controller parameter configuration" interface, select "Zero settings" > "Zero point calibration" to enter the calibration interface. For models that do not support zero point calibration function, the zero point calibration interface will not be displayed.



3. **Select the calibration mode:** The zero point calibration interface includes "Standard mode" and "High precision mode". Check the differences between them according to the following mode introduction and choose the mode based on the actual needs. After selection, click "Next" to proceed to the next step.

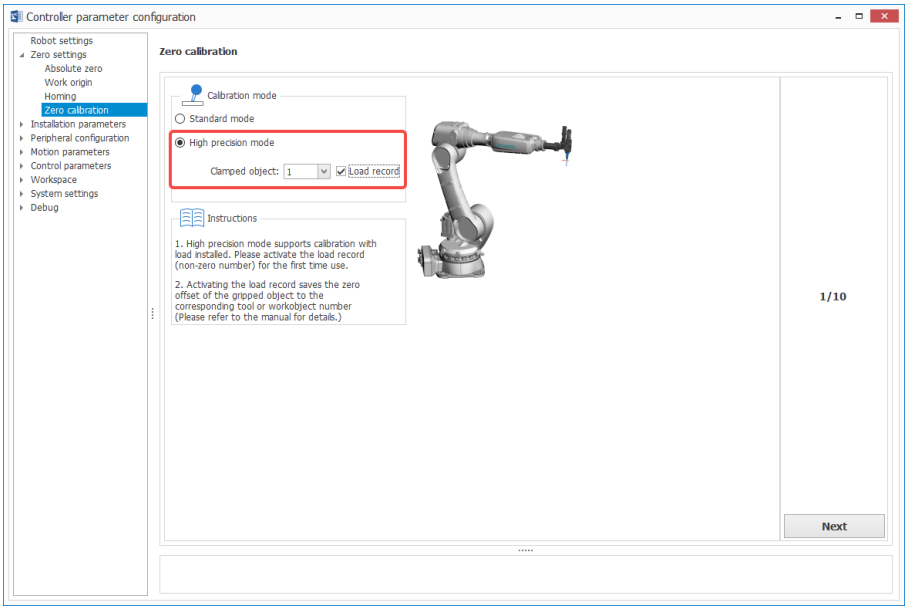
Standard mode: After performing zero point calibration in this mode, the robot's zero point will be restored to factory zero. If the zero point calibration in this mode is performed with the robot under load, the calibration accuracy will be less than that under no-load condition.

High precision mode: This mode requires load recording first when zero point loss has not occurred. Subsequently, if zero point loss occurs, after the zero point calibration in this mode is performed, the robot's zero point will be restored to the zero point at the time of load recording. If the zero point calibration in this

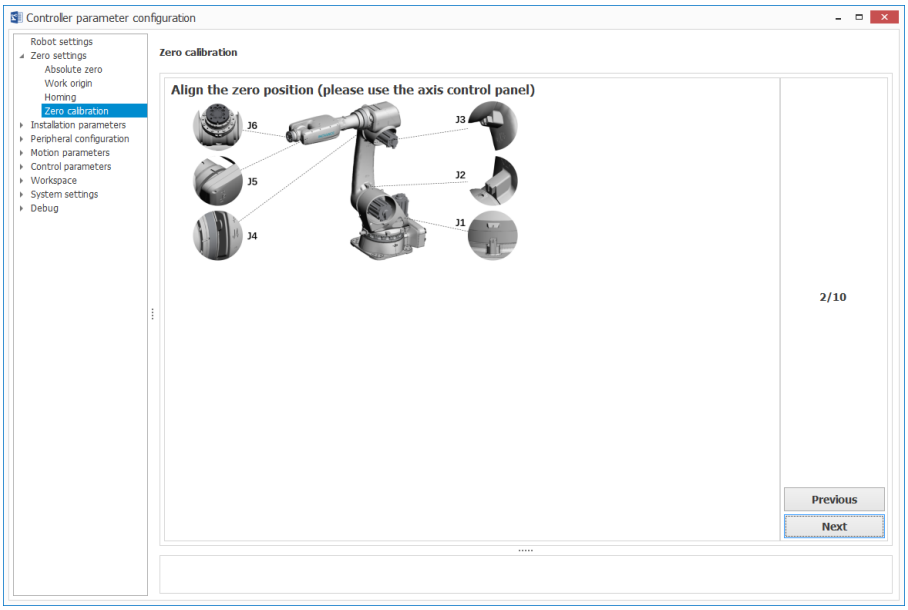
mode is performed with the robot under load, the calibration accuracy will be better than that in standard mode.

If high precision mode is selected, you need to select the number of the object currently gripped by the robot, with the object number corresponding to the tool/workobject/load actually gripped by the robot at present. If not corresponding, it will affect the calibration accuracy.

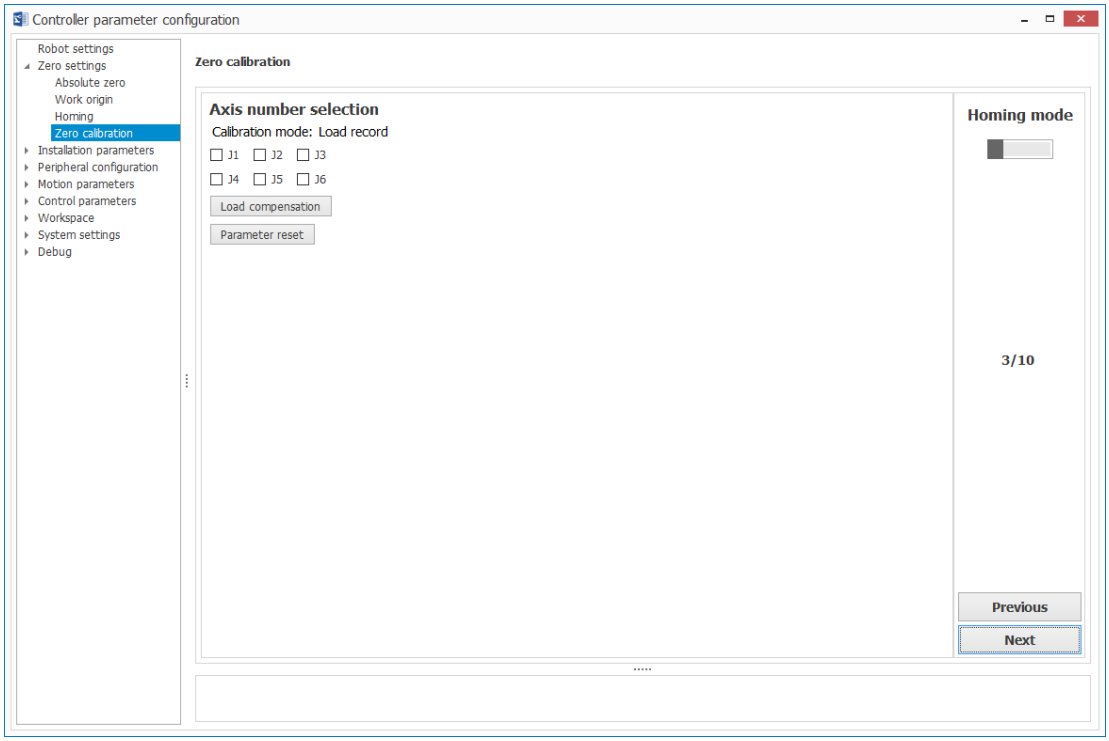
In high precision mode, the gripped object number 0 corresponds to no-load, which does not require load recording. If the robot is gripping a load, select the corresponding non-zero gripped object number, and check the "Load record" option before zero point loss occurs. Then perform the first zero point calibration under load recording, to record the zero point that is not lost in this load gripping state. Subsequently, if zero point loss occurs and you need to restore the zero point in this load gripping state, instead of checking the "Load record" option, directly select the corresponding gripped object number and proceed with the subsequent zero point calibration steps.



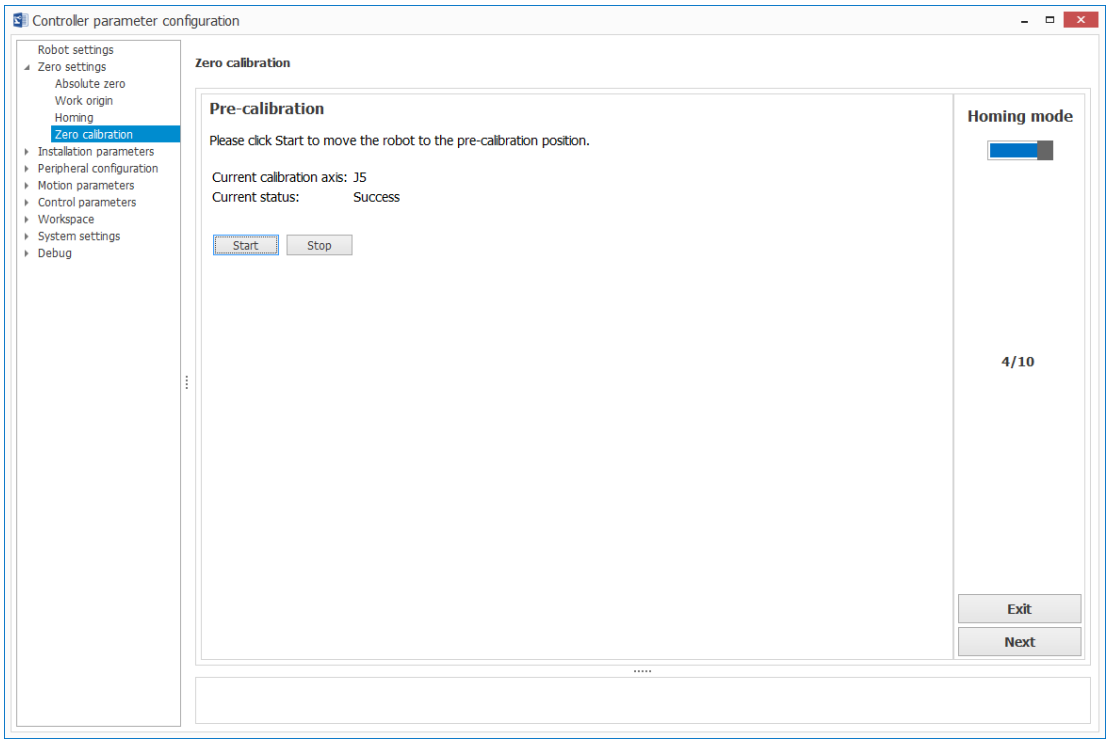
4. Align zero-point slot position: Use the axis control panel to teach the robot, aligning the zero-point slots for all axes. The zero-point slot positions can be referred to in the interface's slot position display, as shown in the figure below. Then, click "Next" to proceed to the next step.



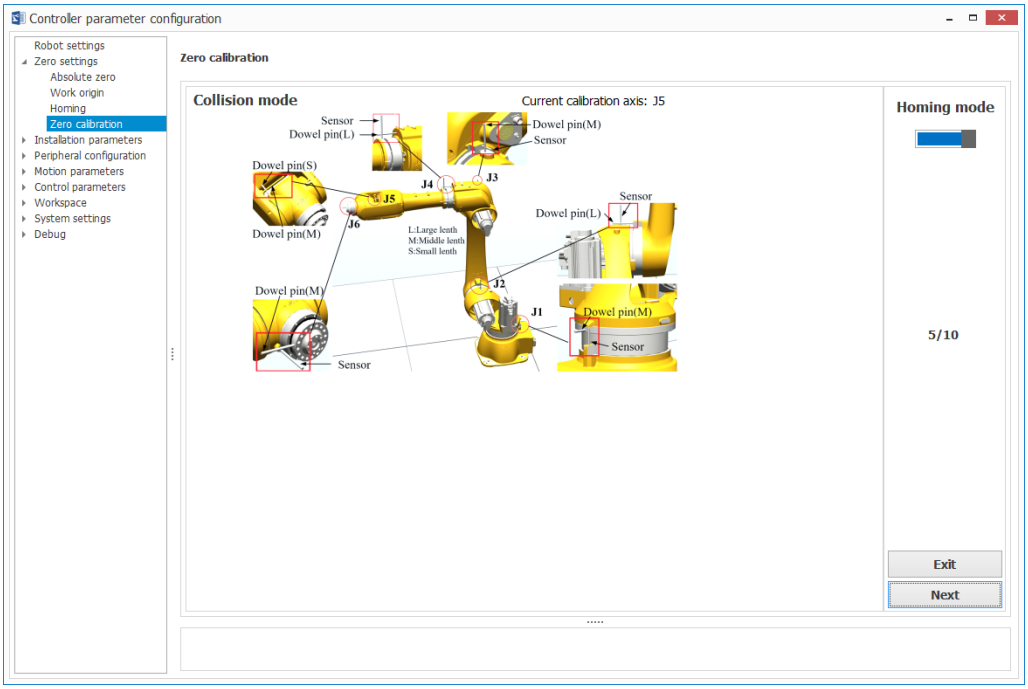
5. Select target axis number for calibration: According to calibration requirements, select the axis number and enable the zero point mode switch in the right function list. Multiple axis numbers can be selected simultaneously here. After selection, click "Next".



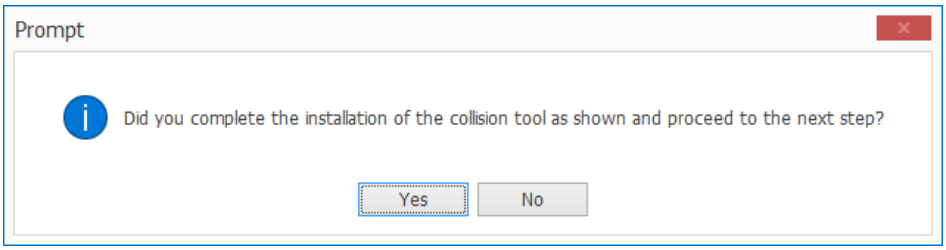
6. Move to pre-calibration position: Switch to the control panel (do not close the zero point calibration interface during switching), enable the robot, then return to the zero point calibration interface and click the "Start" button. The robot will slowly move to the pre-calibration position. After the current status indicates "Success", click "Next". If the robot axis to be calibrated does not have a pre-calibration position set, skip this step.



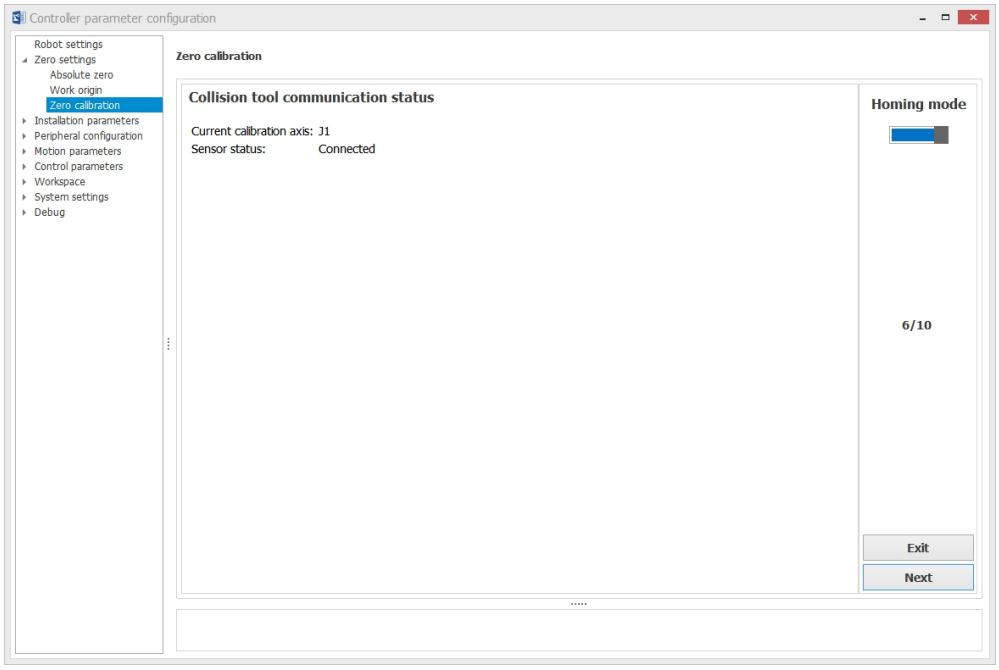
7. Install the calibration tool: Refer to the installation method description in pre-calibration preparation and install the calibration tool at the designated position of the current calibration axis. Do not install the calibration tool on other axes to be calibrated for the moment. Note that the installation methods vary by each robot model. For the specific installation method, refer to the description in the corresponding "Robot user guide - manipulator" and the installation method description in pre-calibration preparation. The installation method shown in the zero point calibration interface below is for reference only.



After completing calibration tool installation, click "Next" and a prompt box will pop up. Re-check if all zero-point slots for all axes are aligned. After confirmation, click "Yes" to proceed to the next step.

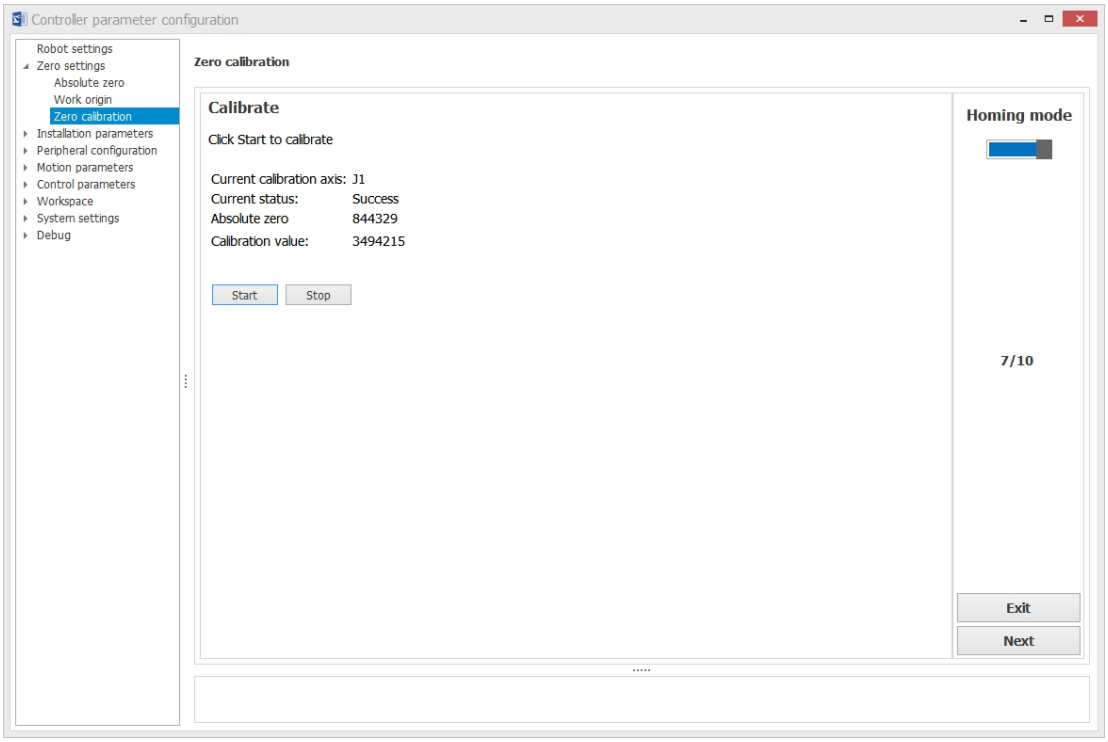


8. Check the sensor connection: Check the communication status of the sensor. If the sensor is connected as described in pre-calibration preparation and the serial port is successfully opened, the sensor status will display "Connected". Otherwise, troubleshoot for sensor damage, incorrect connection method, or failure to open the serial port. Click "Next" after communication is normal. Skip this step if the calibration mode for the target axis to be calibrated uses positioning pin collision mode.

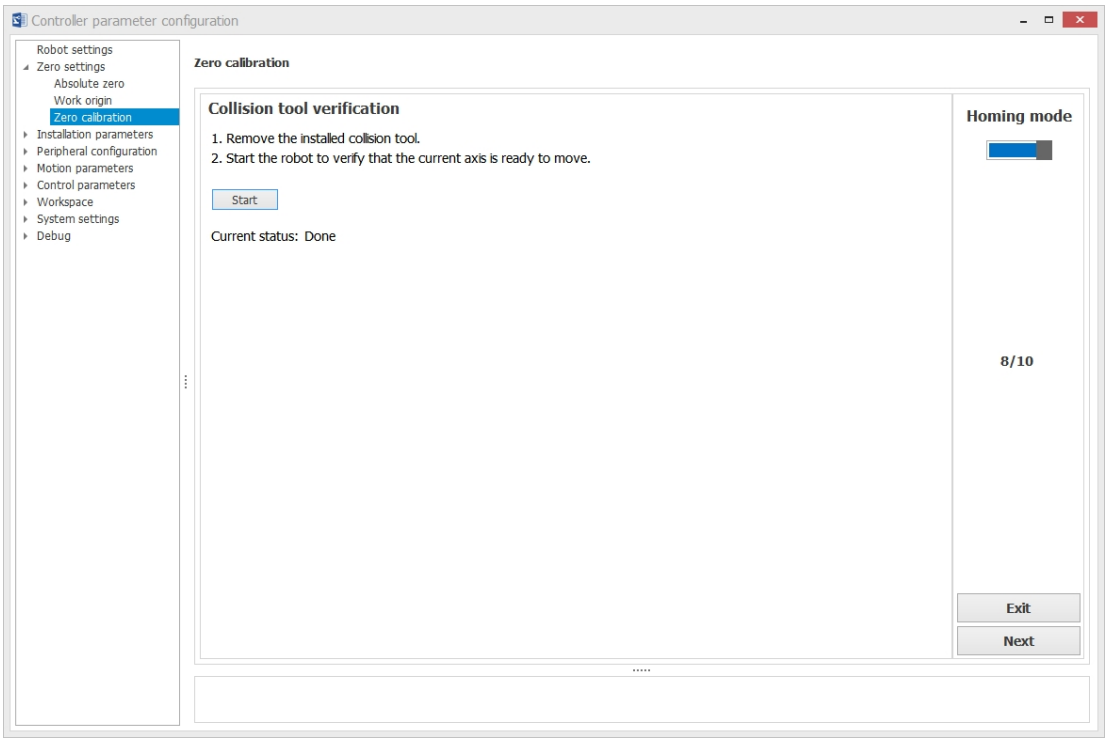


9. Perform calibration: Switch to the control panel on the main page (do not close the zero point calibration interface during switching), and enable the robot. Return to the zero point calibration interface and click the

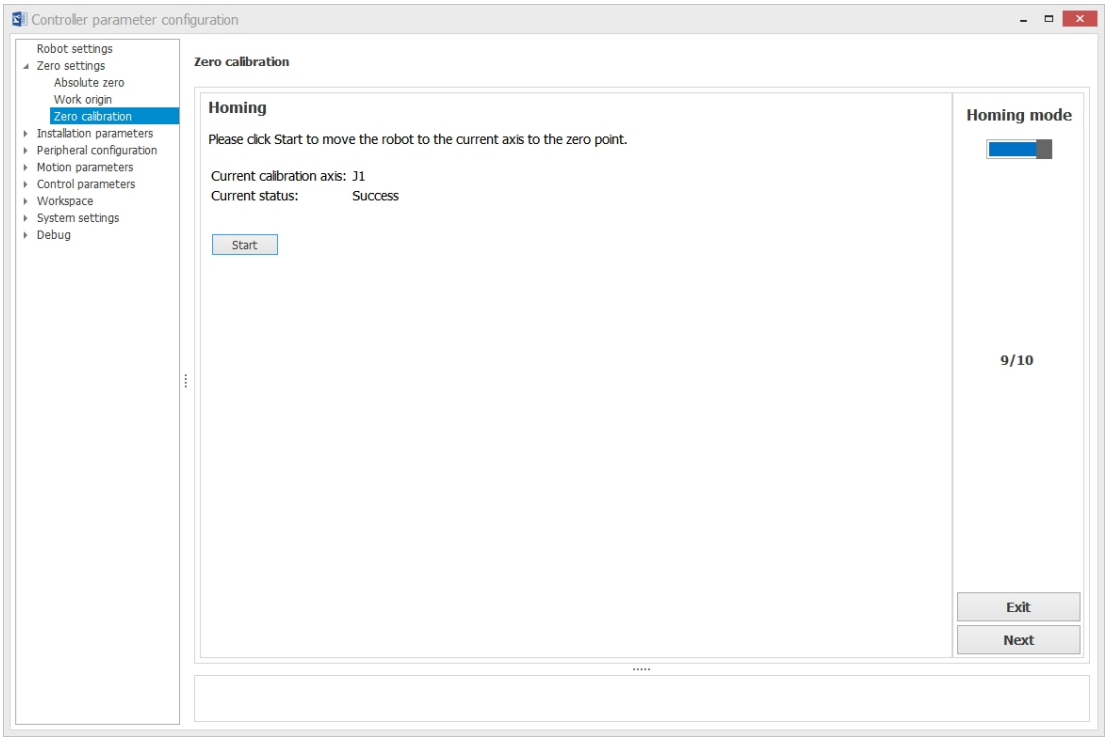
"Start" button. The robot will move slowly until the positioning pin collides with the sensor/positioning pin on the other side. After three collisions, the robot will automatically return to the pre-calibration position. At this time, the zero point calibration interface displays the current status "Success" and provides the calibration value. Then, click "Next" to proceed to the next step.



10. Remove the calibration tool and verify if removal is successful: Remove the sensor, positioning pin, or calibration fixture for the current axis. After removal is complete, switch to the control panel on the main page (do not close the zero point calibration interface during switching), and enable the robot. Click the "Start" button, and the robot will move to the collision point position to perform a test movement for verifying whether the calibration tool has been completely removed. After the current status displays "Done", click "Next".



11. Perform homing: Click "Start", the robot automatically moves to the zero-point slot alignment position of the current axis.

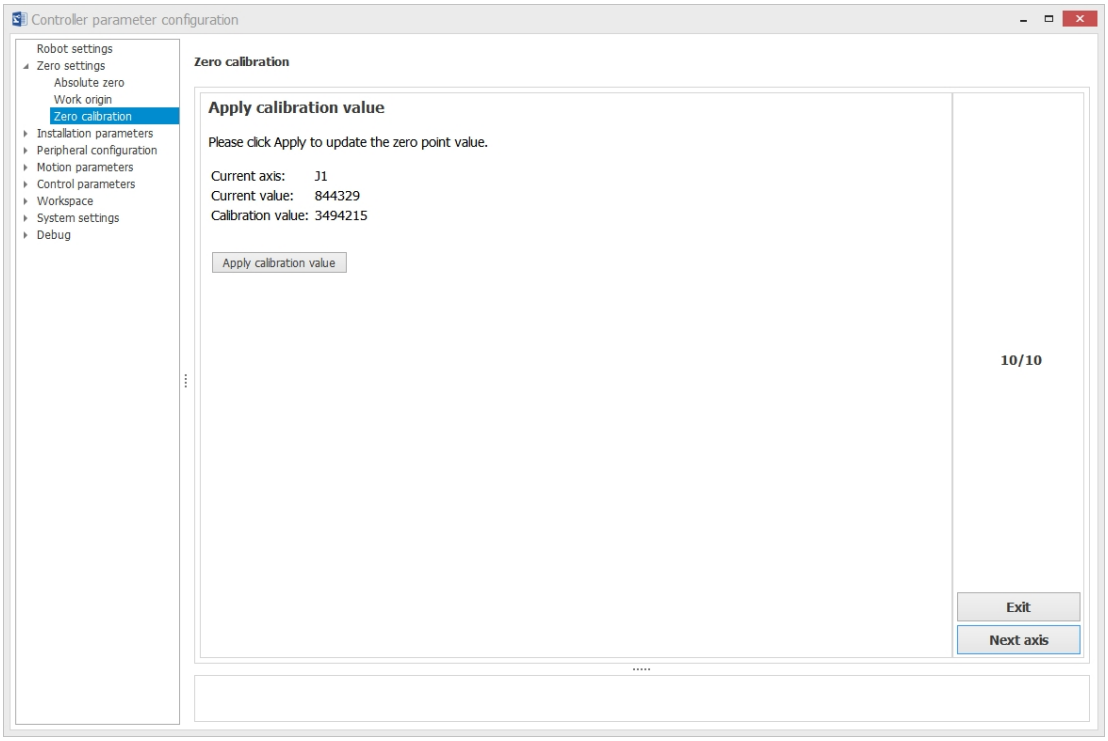


12. **Apply the calibration value:** Based on the current zero point calibration value, determine whether to apply the calibrated zero point. To apply the calibration value, first switch to the control panel and press the "Emergency stop" button, then click "Apply calibration value". After successful application, the absolute zero point value of the axis to be calibrated will be updated.

If you are performing load recording in high precision mode, this step is mainly used to record the zero point offset value in the load gripping state, so there is no need to apply the calibration value.



If the current value differs from the calibration value by no more than 3,000 pulses, the zero point deviation is not significant; applying the calibration value is not recommended. If the current value differs from the calibration value by more than 3,000 pulses, applying the calibration value to update the zero point is recommended.



9.2.3.2.3 Notes

- This function requires high precision of the calibration tool. Before performing calibration, wipe the calibration tool clean with a lint-free cloth, then check if the calibration tool calibration has residual dust, stains, or debris. If residue is present, it is recommended to clean thoroughly or replace the calibration tool.
- For 6-axis zero point calibration, the zero calibration tools (sensor, positioning pin, calibration fixture) must be inspected for surface damage or deformation before use. If obvious damage or deformation is found, replace with a normal zero calibration tool.
- If the zero point calibration interface is closed or exited during calibration due to some reasons, it indicates a calibration failure. In this case, promptly remove the calibration tool and ensure it is completely removed. Then, re-execute the zero point calibration function.
- The calibration process requires a certain amount of time; patiently follow the standard operating steps. If you insert the calibration tool at a non-prompted stage (e.g., at the start stage or other stages of the calibration), it may damage the calibration tool.
- Ensure sufficient movement space for the robot and check the surrounding environment for potential interference. If interference is possible, clean the surrounding environment.
- During calibration, if interference or other risks are detected, immediately press the "Stop" button on the calibration interface or press the "Emergency Stop" button. After risks are eliminated, re-execute the zero point calibration function.
- If inserting or removing the calibration tool is difficult, do not force it in with tools such as a hammer or force it out with pliers. Apply lubricant to the insertion end surface of the tool to assist insertion, or use a dedicated removal tool to assist removal of the calibration tool, avoiding damage to the calibration tool.
- When using a sensor for zero point calibration, avoid entangling the sensor's wiring with the robot or other surrounding moving equipment.
- After the zero point calibration is completed, immediately reinstall the protective cover onto the fixed calibration hole of each axis. If the protective cover is lost or damaged, replace it with a new spare part.

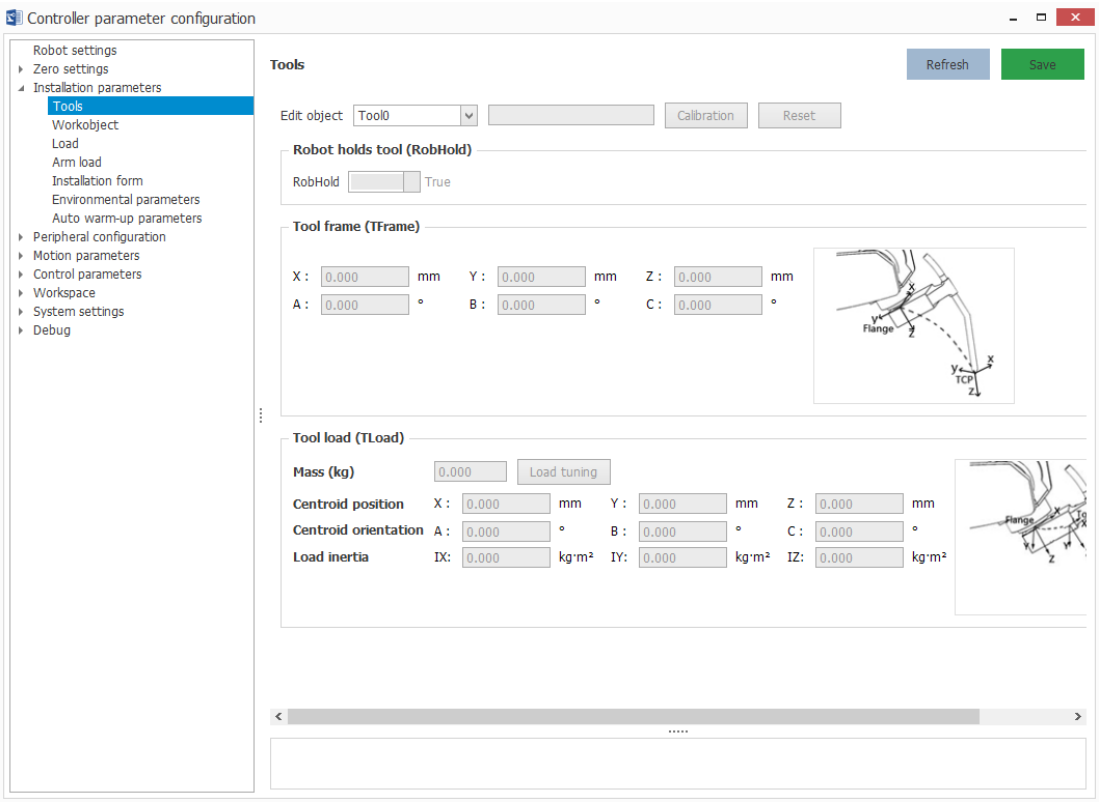
- After the calibration is completed, immediately place the calibration tool back into the storage box.

9.3 Installation Parameters

9.3.1 Tools

Steps

Double-click "Controller parameter configuration" in the device tree, then select "Installation parameters" > "Tools", as shown below. You can set the tool parameters, including tool frame and tool load parameters. Also, you can calibrate the tool frame with the "Calibration" button.



The tool parameters include grip properties, tool frame, and tool load parameters.

Robot holds tool	Tool frame	Tool load
Whether the robot grips the tool.	Includes the value of position parameters X, Y, and Z in mm, and the value of orientation parameters A, B, and C in degrees (°).	• Mass: The mass is the weight of the load in kg. • Centroid position: The X, Y, and Z coordinates of the mass center relative to the frame of the load to be tuned, in mm. • Centroid orientation: The orientation is the A, B, C coordinates of the mass center relative to the object frame of the load to be tuned, in degrees (°). • Load inertia: The moment of inertia of the load around the X, Y, and Z axes of its centroid frame, in $kg \cdot m^2$. CAUTION Enter the tool load parameters correctly as they affect the accuracy and stability of the robot motion.

The number range of the tool frame is 0 to 15. Tool0 is used when no tool is mounted (end flange) and is the default tool number (default parameters cannot be changed). Al other 15 frames except Tool0 frame can be calibrated based on the actual tool mounted.

The tool frame can be divided into TCP (Tool Center Point) and ECP (External Control Point). The following describes the tool calibration method according to the different tool types.

Calibration of the frame of grip tool in TCP mode

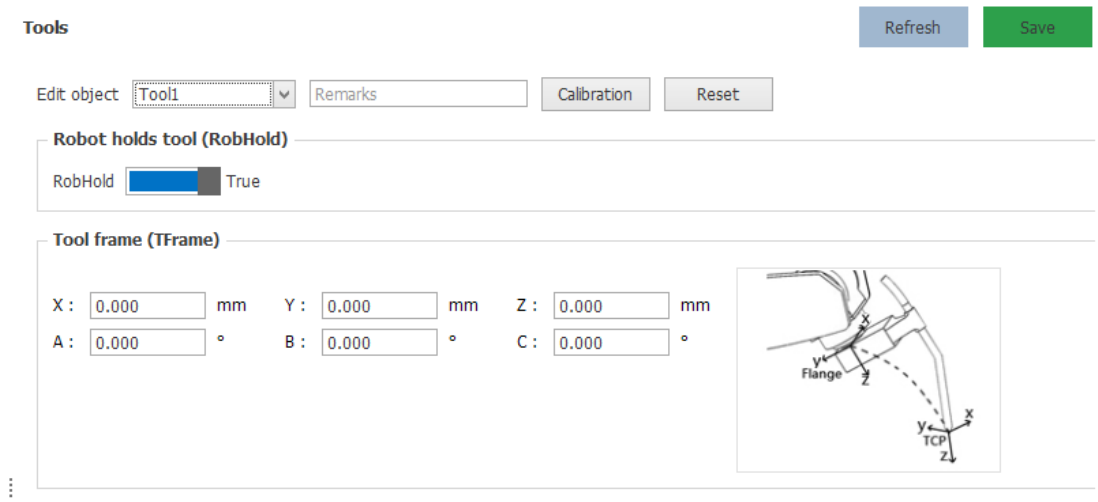
After a tool such as a gripper, a nozzle, or a welding gun is installed at the end of the robot, it is necessary to calibrate the tool frame before using the tool. The purpose is to determine the relationship between the working position and orientation of the tool and the frame of the end flange of the robot. In this way, you can accurately control the tool for operation.

The calibration methods for the grip tool include direct method, three-point TCP method, and three-point TCP+ZX method.

Method	Description	Applicable scenario
Direct method	Directly enter the frame parameters.	Frame parameters are known.
3-point TCP method	Obtain the tool position by three points.	Position values of the tool frame need to be manually calibrated.
3-point TCP+ZX method	Obtain the tool position by three points and then the tool orientation by three more points.	Position and orientation of the tool frame need to be manually calibrated.

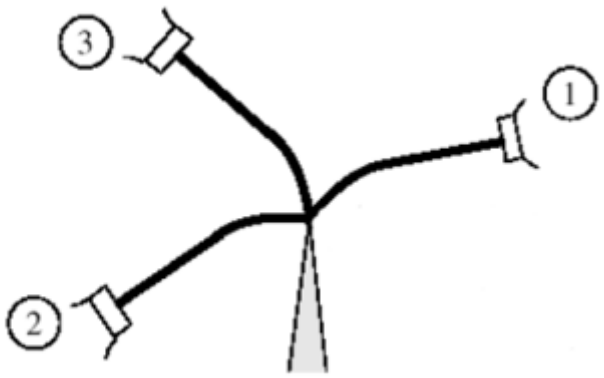
Direct method

If you know the values of the tool frame, you can select direct method, enter the values and save them.



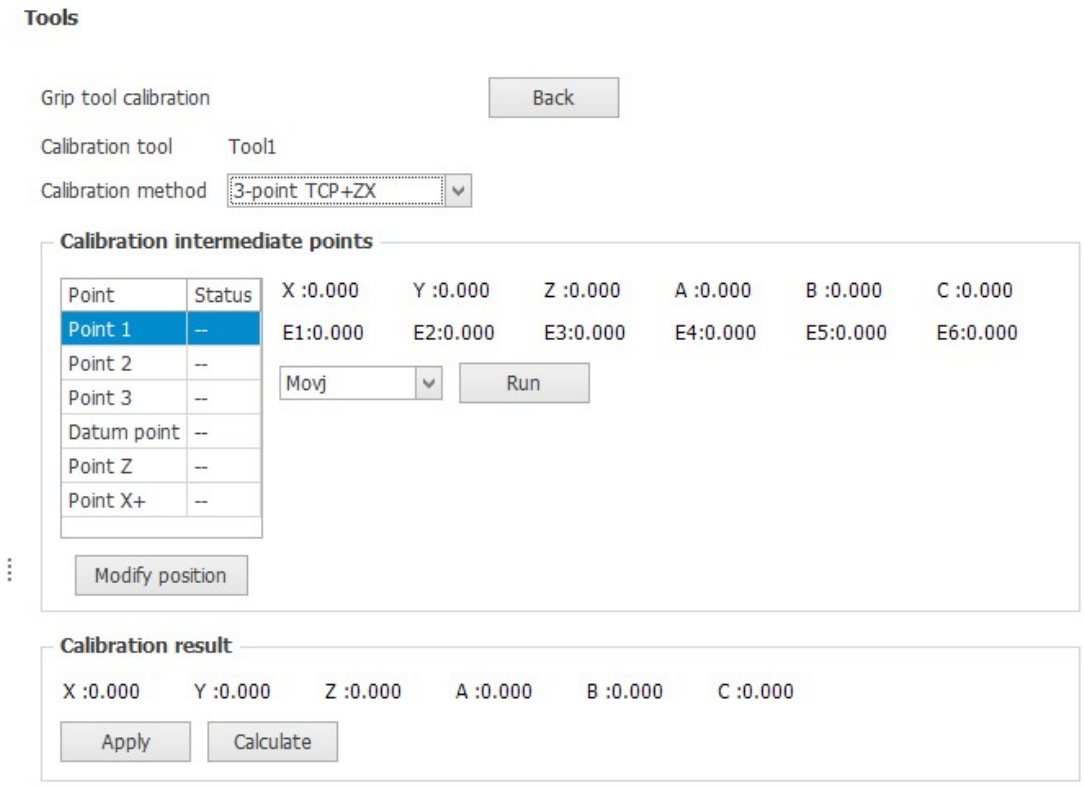
3-point method

This method is applicable to scenarios where it is only needed to get the TCP position, without the tool orientation.



After installing the tool at the end of the robot, perform the following steps.

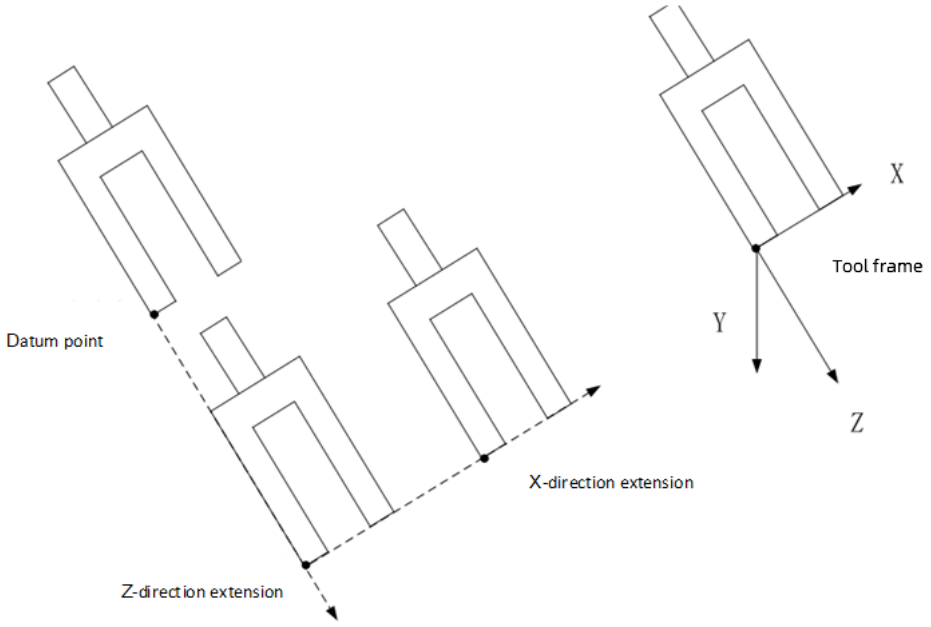
1. Select the tool to be calibrated, and click the "Calibration" button to enter the calibration page.
2. Control robot motion so that the TCP is aligned with the reference point in direction ①, select point 1 and then click "Modify position".
3. Control robot motion so that the TCP is aligned with the reference point in direction ②, select point 2 and then click "Modify position".
4. Control robot motion so that the TCP is aligned with the reference point in direction ③, select point 3 and then click "Modify position".
5. Click "Calculate" to obtain the calibration result.
6. Click "Apply" to record the calibration result and complete the calibration.



3-point TCP+ZX method

This method is applicable to scenarios where both the TCP position and the TCP orientation need to be obtained.

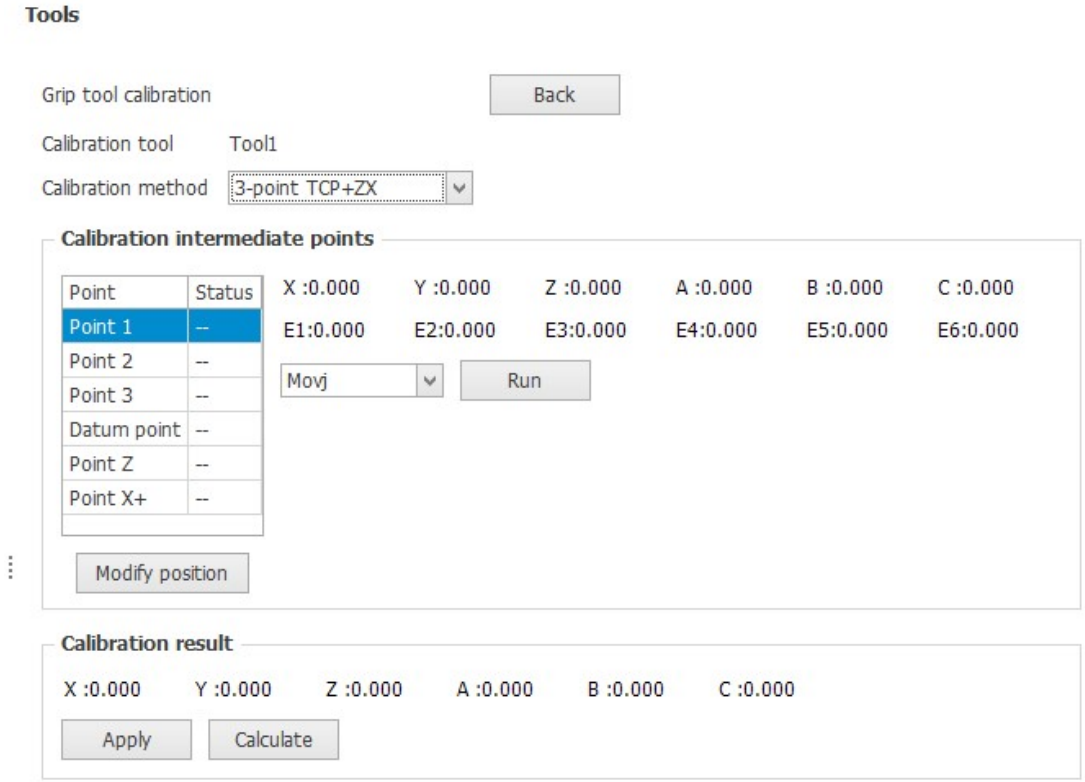
Three more points are taken to calibrate the orientation of the tool frame on the basis of 3-point TCP method. The Z extension point and the reference point compose the Z direction of the tool frame. The X and Z extension points compose the X direction of the tool frame. The Y direction is obtained from the Z and X directions by the right-hand rule. See the figure below.



The steps are as follows.

1. Select the tool to be calibrated, and click the "Calibration" button to enter the calibration page.
2. Control robot motion so that the TCP is aligned with the reference point in direction ①, select point 1 and then click "Modify position".

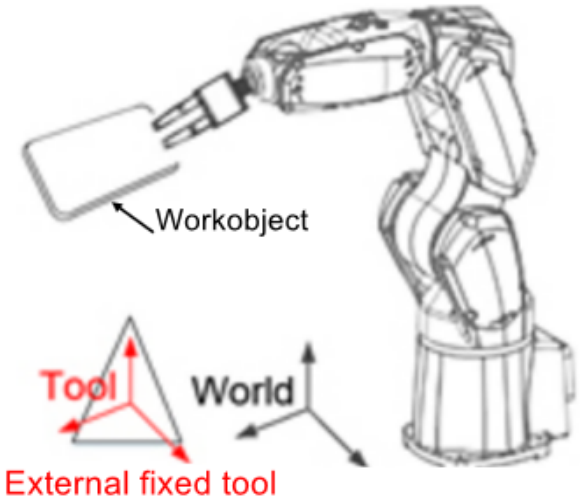
3. Control robot motion so that the TCP is aligned with the reference point in direction ②, select point 2 and then click "Modify position".
4. Control robot motion so that the TCP is aligned with the reference point in direction ③, select point 3 and then click "Modify position".
5. Control robot motion, select the TCP at any moment as the datum point and click "Modify position".
6. Control robot motion so that the tool extends by a distance in Z direction, select the Z extension point and click "Modify position".
7. Control robot motion so that the tool extends by a distance in X direction, select the X extension point and click "Modify position".
8. Click "Calculate" to obtain the calibration result.
9. Click "Apply" to record the calibration result and complete the calibration.



Precautions for calibration of grip tool

- TCP: Tool center point, which is used as a reference point for measuring the arrival position of the robot. The tool endpoint is generally taken as the TCP and its direction can be freely defined.
- The orientations of the three points taken shall vary as much as possible. An orientation difference of at least 20 degrees is recommended.
- After the calibration is complete, the error parameter will be displayed. It is generally accepted that the maximum error should be less than the absolute accuracy of the manipulator by + 0.1 mm. If the error is too large, it is recommended to calibrate again using new points.
- When the tool end of SCARA robot is coaxial with the robot's end axis (i.e., only Z direction values are available), it is not possible to find the Z-direction position parameter of this tool frame by the three-point TCP method.
- In the three-point TCP+ZX method, the datum point, Z+ and X+ together determine the relative pose of the tool and the end flange of the robot.

Calibration of the external fixed tool frame in ECP mode



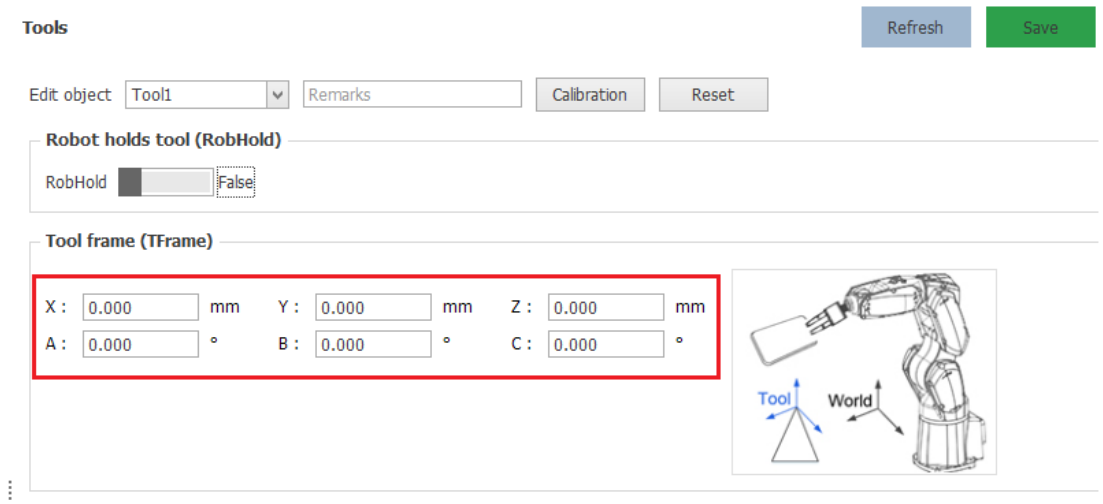
The tool is fixed in the world frame. When the robot grips a work object for operation, it is necessary to calibrate the frame of an external tool in order to determine the working position and orientation of that tool in the world frame. In this way, you can precisely control the work object to reach the end of the tool.

For calibration of the external fixed tool, set the grip property of the tool (RobHold) to "False". The calibration method for the external fixed tool is similar to that for the grip tool, including direct method, known tool 1-point TCP, known tool 1-point TCP+ZX, unknown tool 3-point TCP, and unknown tool 3-point TCP+ZX. The reference frame of the external fixed tool is the world frame.

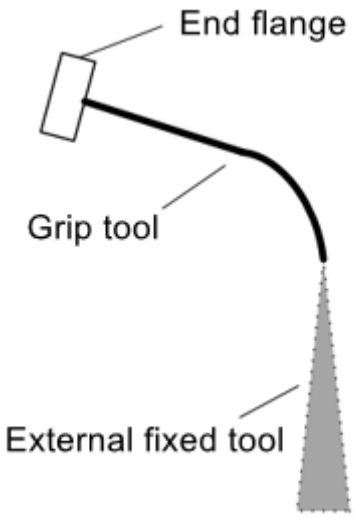
Method	Description	Applicable scenario
Direct method	Directly enter the frame parameters.	Frame parameters are known.
Known tool 1-point TCP	The currently active grip tool has been calibrated. In this case, calibrate the position of the external fixed tool by one point.	Position values of the tool frame need to be manually calibrated.
Known tool 1-point TCP+ZX	The currently active grip tool has been calibrated. In this case, calibrate the position of the external fixed tool by one point, and the orientation by two more points.	Position and orientation values of the tool frame need to be manually calibrated.
Unknown tool 3-point TCP	The currently active grip tool is unknown. In this case, calibrate the position of the external fixed tool by three points.	Position values of the tool frame need to be manually calibrated.
Unknown tool 3-point TCP+ZX	The currently active grip tool is unknown. In this case, calibrate the position of the external fixed tool by three points, and the orientation by the datum point and two more points.	Position and orientation values of the tool frame need to be manually calibrated.

Direct method

If you know the values of the external tool frame, you can select direct method, enter the values and save them.



Known tool 1-point TCP



This method is applicable to scenarios where it is only needed to obtain the position of the fixed tool, without the orientation, when the frame of the grip tool is known.

Fix the external tool relative to the world frame, install the grip tool at the end of the robot and calibrate the tool, and then perform the following steps.

1. Select the tool to be calibrated (must not be the same with the grip tool number), and click the "Calibration" button to enter the calibration page.
2. Control the robot motion so that the TCP of the grip tool is aligned with the ECP (External Control Point) of the external fixed tool. Select point 1 and click "Modify position", as shown in the preceding figure.
3. Click "Calculate" to obtain the calibration result.
4. Click "Apply" to record the tool calibration result and complete the calibration.

Known tool 1-point TCP+ZX

This method is applicable to scenarios where it is needed to obtain both the position and orientation of the fixed tool, when the frame of the grip tool is known.

Fix the external tool relative to the world frame, install the grip tool at the end of the robot and calibrate the tool, and then perform the following steps.

1. Select the tool to be calibrated (must not be the same with the grip tool number), and click the "Calibration" button to enter the calibration page.

2. Control the robot motion to align the TCP of the grip tool with the ECP (External Control Point) of the external fixed tool. Select point 1 and click "Modify position".
3. Select the calibration point in the Z+ direction of the external fixed tool, control the robot to move for a distance, select point Z+, and click "Modify position".
4. Select the calibration point in the X+ direction of the external fixed tool, control the robot to move for a distance, select point X+, and click "Modify position".
5. Click "Calculate" to obtain the calibration result.
6. Click "Apply" to record the calibration result and complete the calibration.

Unknown tool 3-point TCP

This method is applicable to scenarios where it is only needed to obtain the position of the fixed tool, without the orientation, when the frame of the grip tool is unknown.

The external tool is fixed relative to the world frame, and the parameters of the grip tool installed at the end flange of the robot are unknown. The operation process of this calibration method is basically the same as the "3-point" calibration of the grip tool. In the "unknown tool 3-point TCP" calibration, the ECP of the external fixed tool is used as the reference point and aligned with the TCP of the grip tool.

Unknown tool 3-point TCP+ZX

This method is applicable to scenarios where it is needed to obtain both the position and orientation of the fixed tool, when the frame of the grip tool is unknown.

The external tool is fixed relative to the world frame, and the parameters of the grip tool installed at the end flange of the robot are unknown. Perform the following steps.

1. Select the tool to be calibrated (must not be the same with the grip tool number), and click the "Calibration" button to enter the calibration page.
2. Modify the point 1, point 2, and point 3 in the way as described in relation to the "unknown tool 3-point TCP method".
3. Control the robot motion so that the TCP of the grip tool is aligned with the ECP of the external fixed tool. Select datum point and click "Modify position".
4. Select the calibration point in the Z+ direction of the external fixed tool, control the robot to move for a distance, select point Z+, and click "Modify position".
5. Select the calibration point in the X+ direction of the external fixed tool, control the robot to move for a distance, select point X+, and click "Modify position".
6. Click "Calculate" to obtain the calibration result.
7. Click "Apply" to record the calibration result and complete the calibration.

Precautions for calibration of external fixed tool:

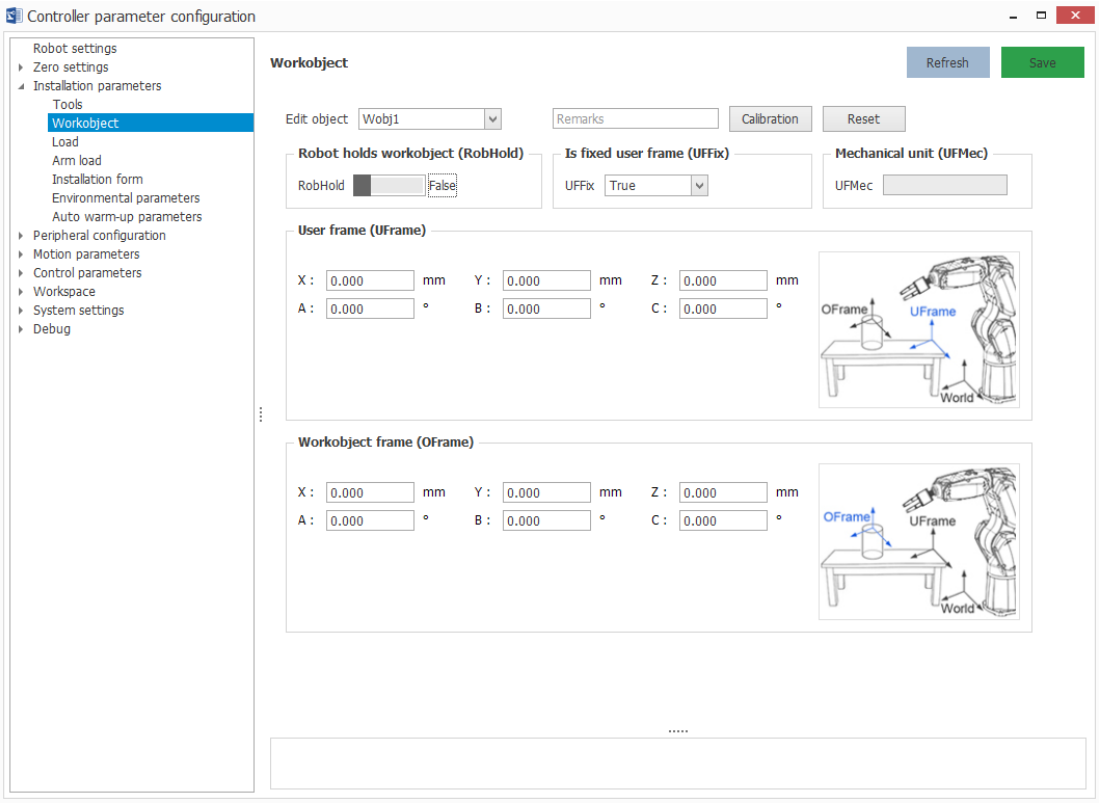
- ECP is the center point of the external fixed tool, and the reference frame of the external fixed tool is the world frame.
- The orientations of the three points taken shall vary as much as possible. An orientation difference of at least 20 degrees is recommended.

- After the calibration is complete, the error parameter will be displayed. It is generally accepted that the maximum error should be less than the absolute accuracy of the manipulator by +0.1 mm. If the error is too large, it is recommended to calibrate again using new points.
- Except for the direct method, other calibration methods need to calibrate the external fixed tool using the grip tool.
- The "Tool load" parameter set for the external fixed tool is also the load parameter of the active grip workobject.

9.3.2 Workobject

Steps

Double-click "Controller parameter configuration" in the device tree, then select "Installation parameters" > "Workobject", as shown below. You can calibrate the workobject frame.



The workobject parameters are described below.

Parameter	Description
Robot holds workobject (RobHold)	Whether the workobject is gripped by the robot.
Is fixed user frame (UFFix)	The UFFix parameter sets whether the user frame is fixed relative to the world frame (when not fixed, it can move with certain unit).
Mechanical unit (UFMec)	The UFMec parameter sets the mechanical unit associated with the frame.
User frame (UFrame)	The UFrame parameters include the value of position parameters X, Y, and Z in mm, and the value of orientation parameters A, B, and C in degrees (°).

Parameter	Description
Workobject frame (OFrame)	The OFrame parameters include the value of position parameters X, Y, and Z in mm, and the value of orientation parameters A, B, and C in degrees (°).

The number range of the workobject frame is 0 to 15. Wobj0 is the default value (default parameters cannot be changed) and coincides with the world frame. All other 15 frames except Wobj0 frame can be calibrated according to requirements.

Calibration of the non-grip workobject frame in TCP mode

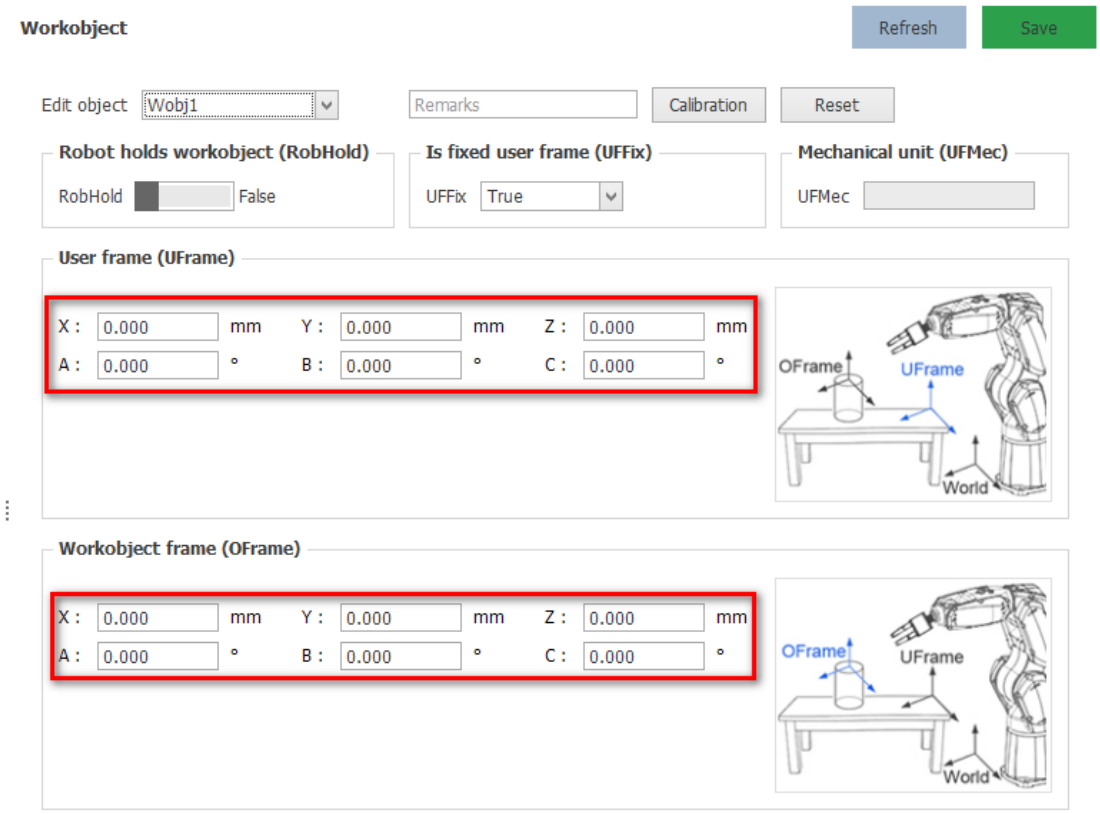
The workobject frame is a user-defined frame. The workobject frame can be set arbitrarily and is generally set on specific objects, such as a workbench or a conveyor. Before using the workobject frame, calibration is required. When the non-grip workobject is calibrated, the active tool should be grip tool that has been calibrated. You also need to change the grip attribute of the workobject (RobHold) to "False" and save the change.

Each frame includes the user frame (UFrame) and the workobject frame (OFrame). The user frame can be calibrated by direct method, 3-point method, and rotation method. The workobject frame can be calibrated by direct method and 3-point method.

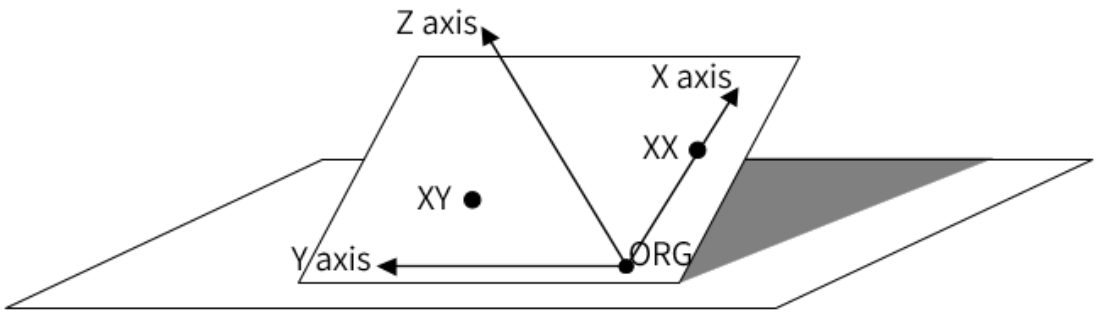
Method	Description	Applicable scenario
Direct method	Directly enter the frame parameters.	Frame parameters are known.
UFrame 3-point method	Calibrate the user frame using three points.	Values of the user frame need to be manually calibrated.
UFrame rotation method	Make marks on the turnplate. Rotate the turnplate and get points for teaching.	A turnplate is provided and the user frame is located at the center of the turnplate.
OFrame 3-point method	Calibrate the workobject frame using three points.	Values of the workobject frame need to be manually calibrated.

Direct method

If you know the values of the user frame or workobject frame, you can select direct method, enter the values and save them.



UFrame/OFrame 3-point method

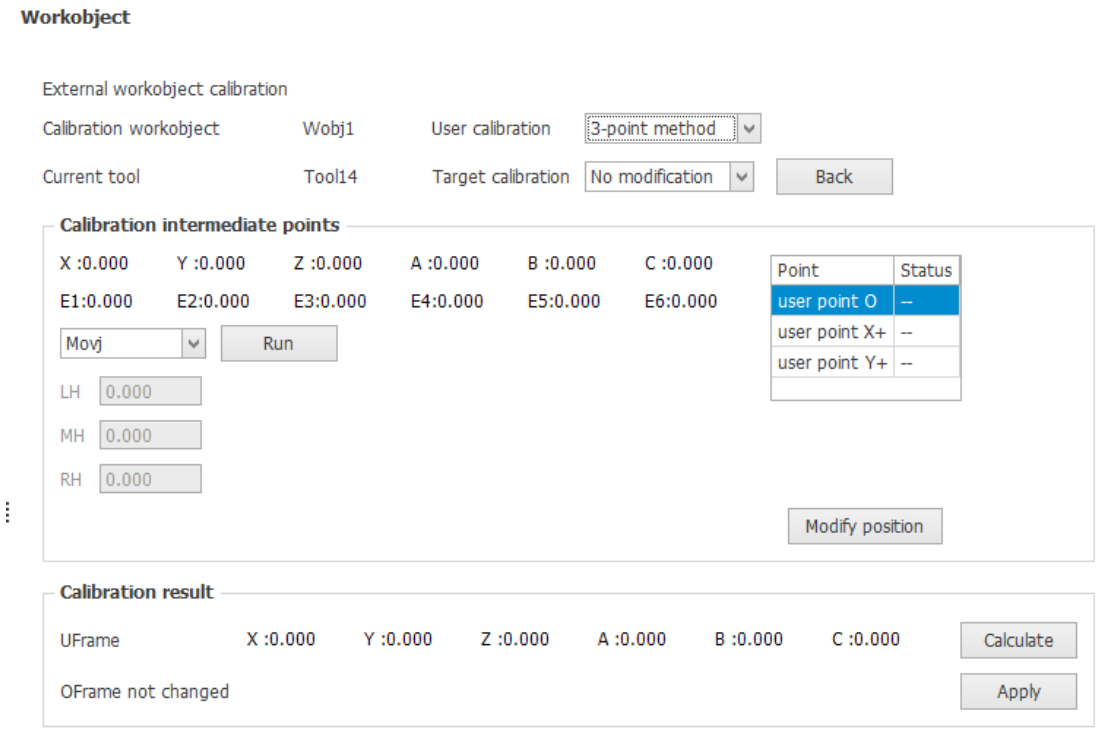


User coordinate defined points
ORG: Origin
XX: Point on X-axis
XY: Point on X-Y plane

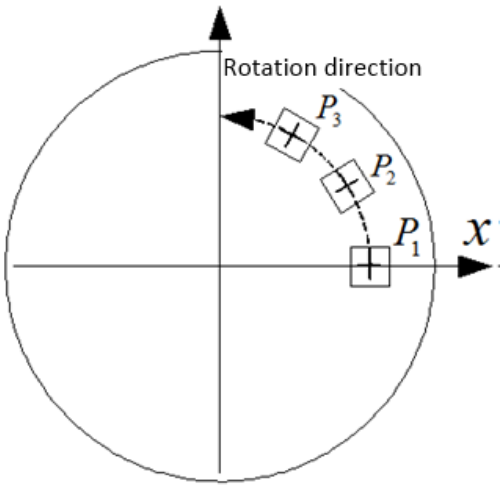
Follow the steps below.

1. Set RobHold to "False", click the "Calibration" button to enter the calibration page, and select "3-point method".
2. Get the origin (ORG) of the frame, select the user point O/workobject point O, and click "Modify position".
3. Take a point XX in the positive direction of the X-axis of the user frame, select user point X+/workobject point X+, and click "Modify position".
4. Take a point XY in the Y+ direction of the XY plane of the frame, select user point Y+/workobject point Y+, and click "Modify position".

5. Click "Calculate" to obtain the calibration result.
6. Click "Apply" to record the calibration result and complete the calibration.



UFrame rotation method



Follow the steps below.

1. Set RobHold to "False", click the "Calibration" button to enter the calibration page, and select "Rotation method".
2. Mark a fixed point on the turnplate, align the TCP with this mark, select user point 1 and click "Modify position".
3. Rotate the turnplate by an angle, align the TCP with this mark again, select user point 2 and click "Modify position".
4. Rotate the turnplate by a random angle again, align the TCP with this mark again, select user point 3 and click "Modify position".
5. Click "Calculate" to obtain the calibration result.
6. Click "Apply" to record the calibration result and complete the calibration.

Precautions for calibration:

- In the 3-point method, the point 1 is not directly on the Y axis, but is a point in the XY plane that moves a distance from the start point 0 in the positive direction of the Y axis. In this way, the point 1 indirectly defines the Y-axis. Finally, the Z axis is obtained through the right-hand rule.
- For SCARA robots, if the component of the Z direction of the user frame in the Z direction of the base frame is negative, the software will automatically reverse the Z and Y directions of the user frame, leaving the X direction unchanged.
- Each grip workobject frame includes a user frame (UFrame) and a workobject/target frame (OFrame). The reference frame of the UFrame is the world frame of the robot. The reference frame of the OFrame is the user frame.

Calibration of the grip workobject frame in ECP mode

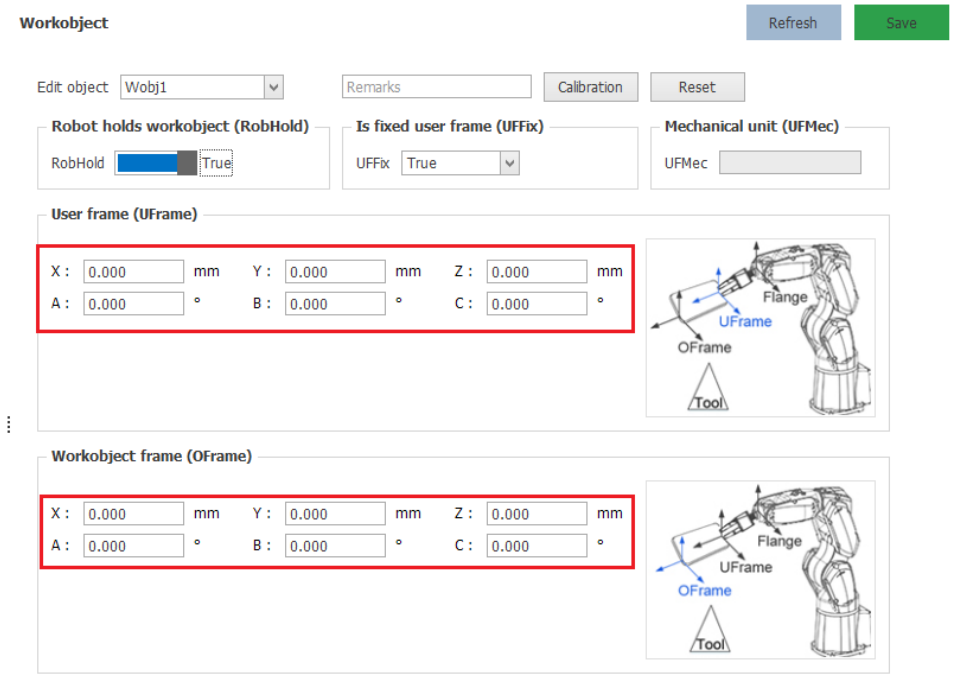
When the non-grip workobject is calibrated, the active tool should be an external fixed tool that has been calibrated. You also need to change the grip attribute of the workobject (RobHold) to "True" and save the change.

Both the user frame and the workobject frame can be calibrated by direct method and 3-point method.

Method	Description	Applicable scenario
Direct method	Directly enter the frame parameters.	Frame parameters are known.
3-point method	Calibrate the user/workobject frame using three points.	Values of the user/workobject frame need to be manually calibrated.

Direct method

If you know the values of the user frame or workobject/target frame, you can select direct method, enter the values and save them.



UFrame/OFrame 3-point method

The 3-point calibration process for the grip workobject is basically the same as that for the non-grip workobject described above. However, the calibration for the grip workobject requires the use of external fixed tool for point acquisition.

Precautions for calibration of grip workobject:

- Select the external fixed tool as the currently active tool and the grip workobject to be calibrated as the currently active workobject. It is recommended to clear the parameters of the workobject and change the grip attribute to "True" before entering the calibration page.
- Each grip workobject frame includes a user frame (UFrame) and a workobject/target frame (OFrame). The reference frame of the user UFrame is the flange of the robot. The reference frame of the OFrame is the user frame.

Non-fixed user frame

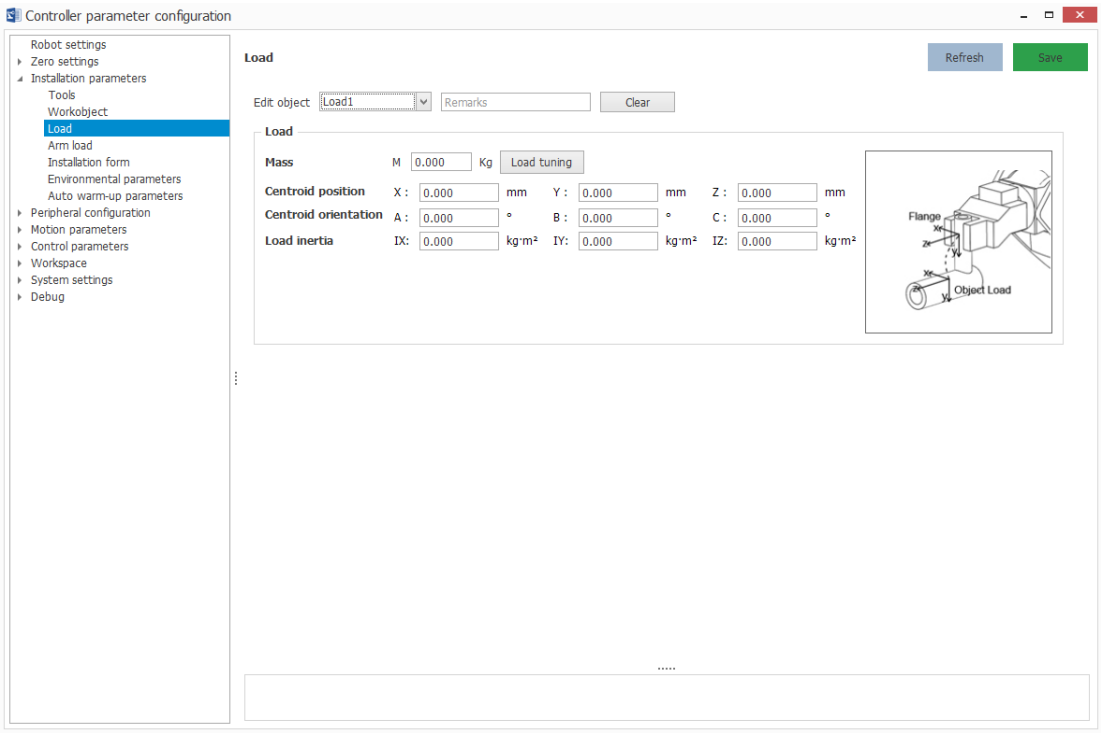
For the non-fixed user frame, see 12.2.2.3 Setting Workobject for the Coordinated Mechanical Unit.

9.3.3 Load

You can set load parameters to improve the accuracy and stability of robot motion.

Steps

Double-click "Controller parameter configuration" in the device tree, then select "Installation parameters" > "Load", as shown below.



Load parameters include mass, centroid position, centroid orientation, and load inertia.

Mass	Centroid position	Centroid orientation and load inertia
The mass is the weight of the load in kg.	Centroid position: The X, Y, and Z coordinates of the mass center relative to the frame of the load to be tuned, in mm.	Centroid orientation: The orientation is the A, B, C coordinates of the mass center relative to the object frame of the load to be tuned, in degrees (°). Load inertia: The moment of inertia of the load around the X, Y, and Z axes of its centroid frame, in $kg \cdot m^2$. Pay attention to the unit of the load parameter and ensure that the input parameters are correct. Otherwise, the accuracy and stability of the robot will be degraded.

The load number range is 0 to 15. Load0 is the default value (default parameters cannot be changed). Load0 indicates no load. All other 15 loads except load0 can be configured as required. If the load mass corresponding to the current load number is 0, the current load number cannot be switched.

- **Results**

  After setting and saving, you can check whether the accuracy and stability of the robot are improved by capturing the trajectory of the robot end.

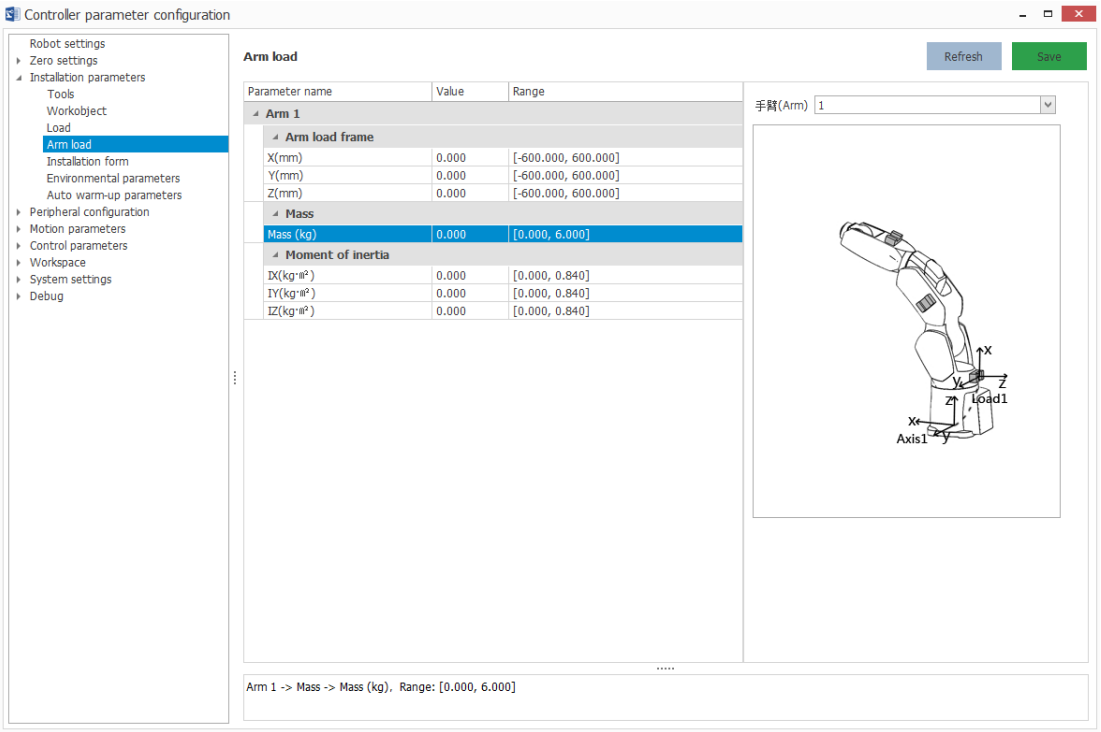
9.3.4 Arm Load

When a load exists between the joints of the robot arms, the robot dynamics will be affected. In severe cases, it may cause the robot to vibrate and affect the end accuracy. Setting the load on the arm helps to take the load characteristics into account in the dynamic calculation and avoid negative effects on robot control.

For a SCARA robot, you can set the joints with load on arm to J1 and J2, and for a 6-axis robot, you can set the joints with load on arm to J1, J2, and J4.

Steps

Double-click "Controller parameter configuration" in the device tree, then select "Installation parameters" > "Arm load" in the opened interface for related settings, as shown in the figure below:



Arm load

For each load, you need to configure the following parameters.

Arm load frame	Mass	Moment of inertia
The load frame includes the X, Y, and Z coordinates of the load's centroid relative to the frame of the joint to which it belongs, in mm.	The mass is the weight of the load in kg.	The moment of inertia of the load around the X, Y, and Z axes of its centroid frame, in kg*m^2. Pay attention to the unit of the load parameter and ensure that the input parameters are correct. Failure to do so may result in false collision detection alarms, cycle times that are too long or too short, or current anomalies.

Results

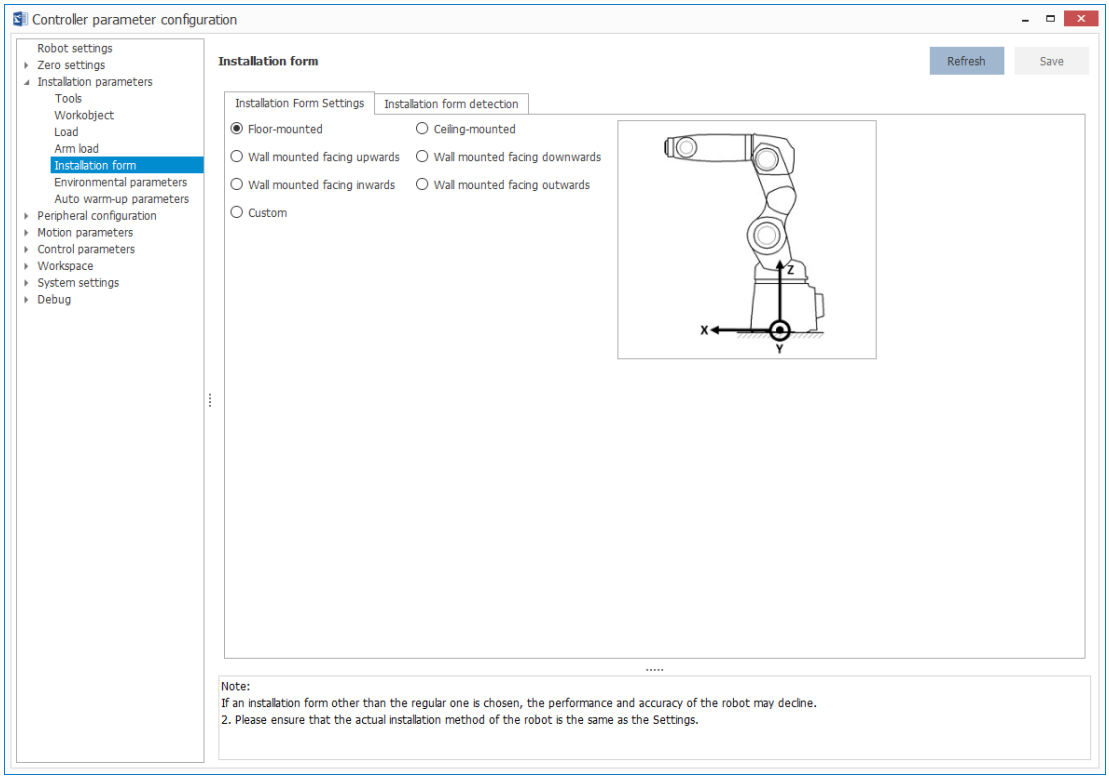
After setting and saving, you can check whether the vibration problem is improved by capturing the trajectory of the robot end.

9.3.5 Installation Form

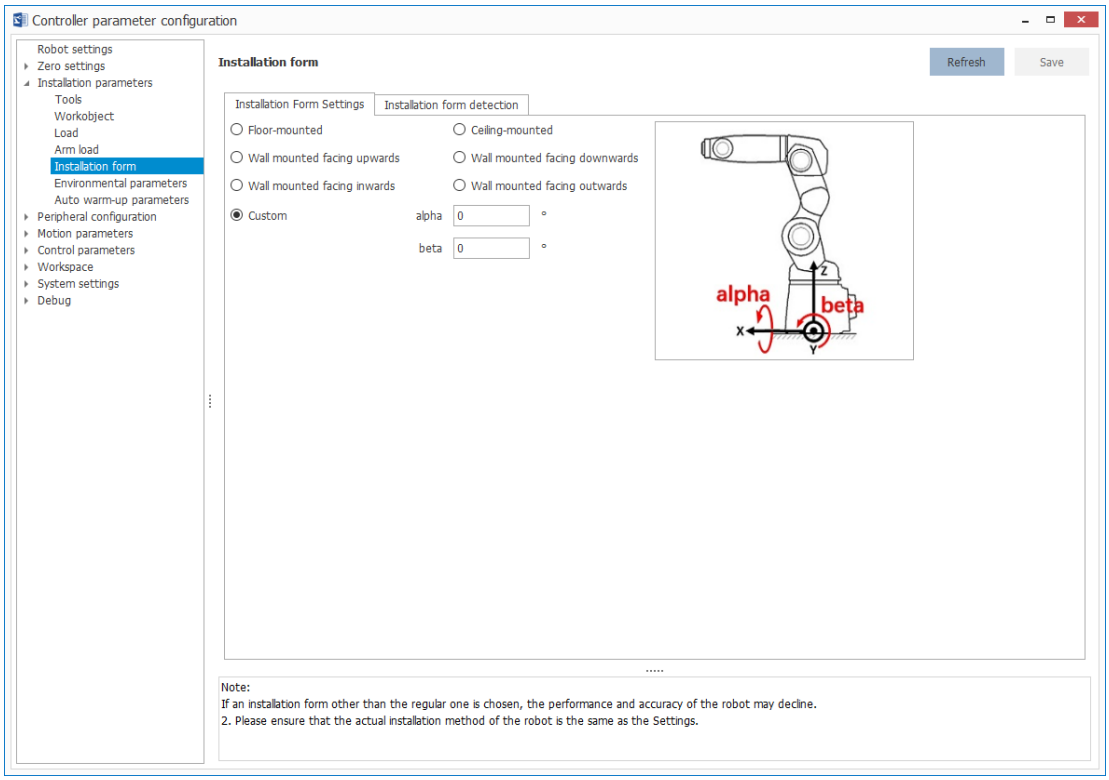
9.3.5.1 Installation Form Settings

Select the installation form according to the actual conditions. Save the selection and restart the controller to make the selection take effect.

Double-click "Controller parameter configuration" in the device tree, then select "Installation parameters" > "Installation form" in the opened interface for related settings, as shown in the figure below:



When selecting the "Custom" installation form, correctly input the alpha and beta installation angles (unit: °). Alpha is the rotation angle around the X-axis of the fixed frame, and beta is the rotation angle around the Y-axis of the fixed frame. The settable range for alpha and beta is -180° to +180°, as shown in the figure below:



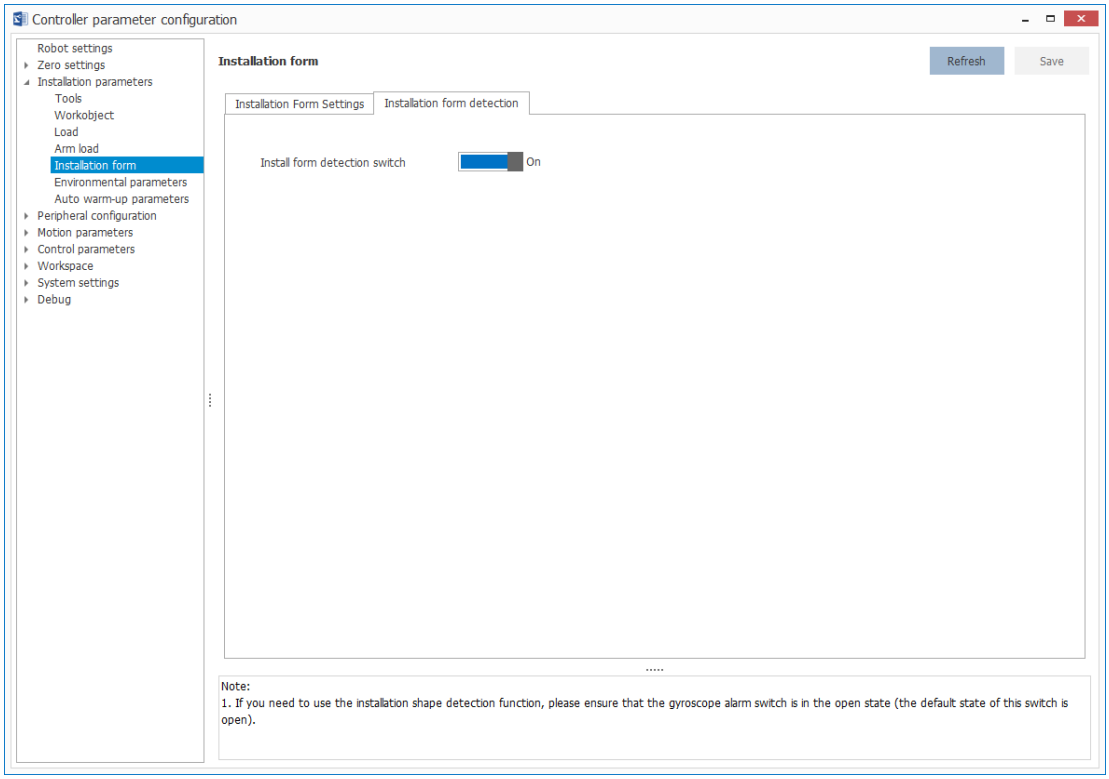
CAUTION

- The installation form function applies only to the 6-axis robots.
- Selecting an installation form other than "Floor-mounted" may reduce the performance and accuracy of the robot.
- This setting affects the robot motion. Ensure that the setting is consistent with the actual condition.

9.3.5.2 Installation Form Detection

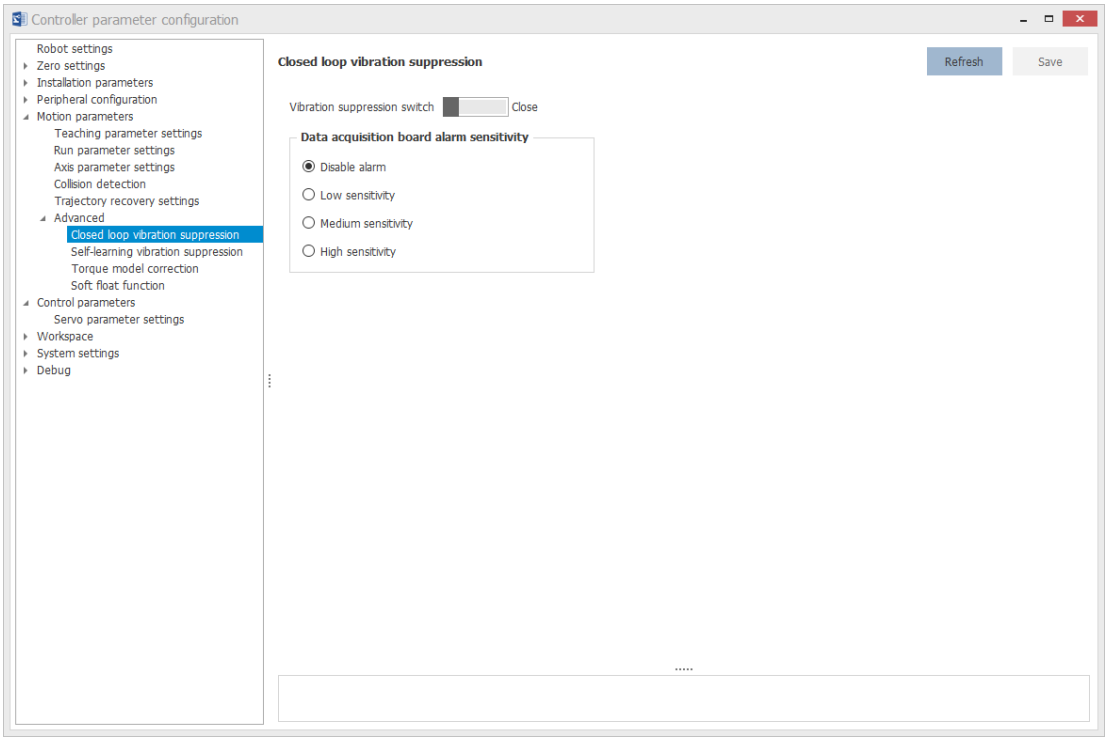
The 6-axis robot can be set up in various installation forms such as ceiling mounting, and also supports setting any installation angle. To ensure normal robot functions and performance, the set installation form must correspond to the actual installation form; if there is a mismatch, alarm 0x20AF (installation form error) will be triggered. This alarm cannot be cleared. Once this alarm occurs, after addressing the relevant error, restart the robot controller to eliminate the alarm.

In addition to incorrect installation form, if the zero point deviation of axis 2 exceeds 10°, this alarm may also be triggered; for models with gyroscopes installed at the elbow, a large zero point deviation of axis 3 may also trigger this alarm. Since the zero point is lost after each firmware update, it is necessary to recalibrate the zero point and restart the system to clear the x20AF alarm.



The installation form detection can be turned on or off on the interface. If 0x20AF alarm has occurred, the installation form detection alarm switch must be turned off and the controller must be restarted to clear the alarm.

The installation form detection function relies on gyroscope signals. If the gyroscope alarm is disabled, the installation form detection function will also be invalid.

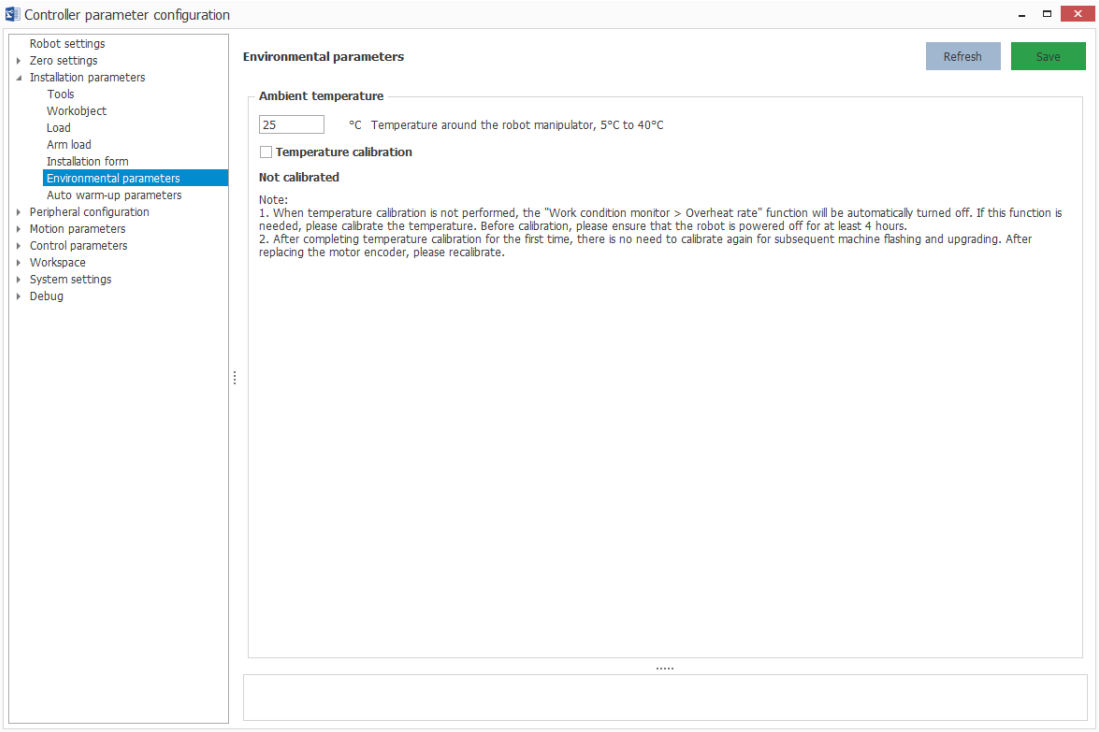


9.3.6 Environmental Parameters

Environmental parameters are used for temperature rise protection function. all models support environmental parameter settings except for the models shown in the table below:

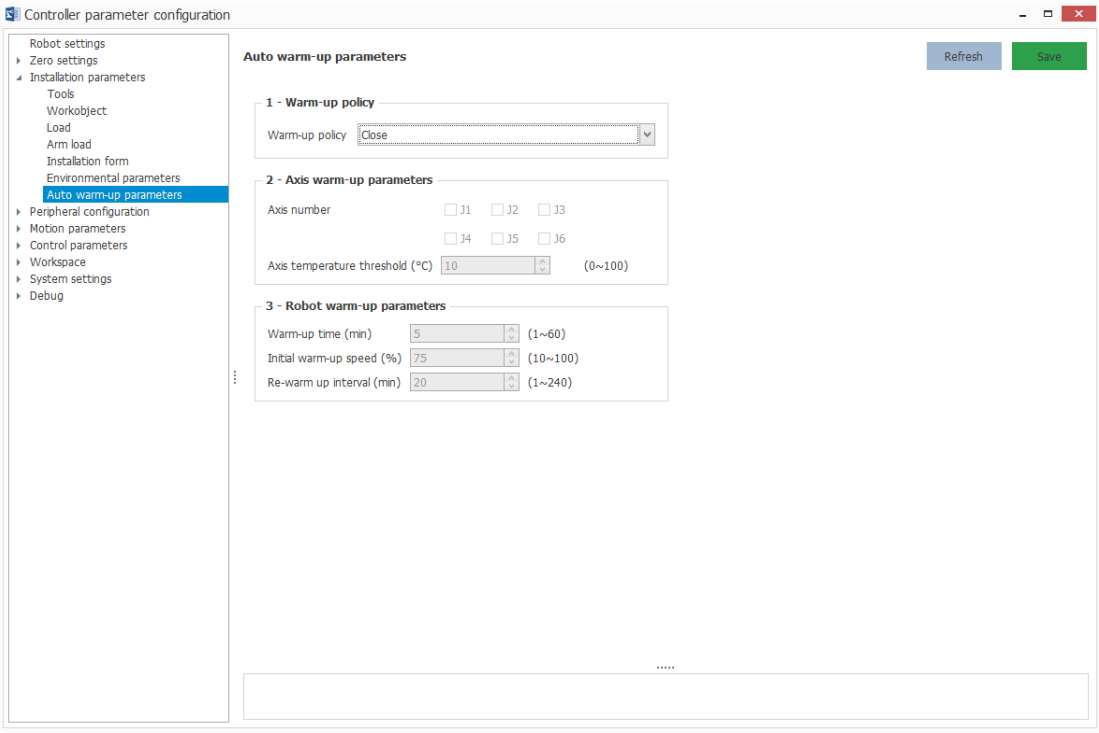
No.	Unsupported robot models	No.	Unsupported robot models
1	IRS111 series	6	IR-R10H-145S5-D0A_01741213
2	IR-TS series	7	IR-C8-62Z20S3-A1THC 01740430
3	HRTA100-10-50Z20IS3_01740422M00038	8	IR-R12-201S7-B1CD_01740678
4	IR-R10-200S5-B1NH_01740237	9	IR-R30H-183S7-D0A_01741212
5	IR-R10-200S5-D1NH-INT_01741038	10	IRGF-R10-200S10-B1_01740563

Enter the real temperature of the site where the robot is installed and save the settings. Click "Save" in the upper right corner to make settings take effect.

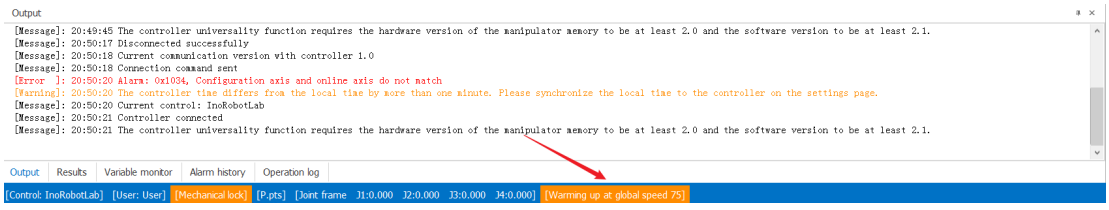


9.3.7 Auto Warm-up

The automatic warm-up function with multiple policies is implemented to solve the torque overlimit/ excessive tracking error caused by high friction when the robot is cold.



Auto warm-up parameter settings



Warm-up speed status and speed display

You can set the warm-up policy, axis warm-up parameters (axis number, axis temperature threshold), and robot warm-up parameters (warm-up time, initial warm-up speed, and re-warm up interval).

Parameter	Description
Warm-up policy	• Close: Disables the function. • Only consider warm-up time: The automatic warm-up is performed only according to the robot warm-up parameters (warm-up time, initial warm-up speed, and re-warm up interval). • Consider axis temperature and warm-up time: The automatic warm-up is performed only according to the axis warm-up parameters and robot warm-up parameters.  When the robot does not support temperature rise protection function, this option is not displayed.
Axis number	When the temperature threshold of the selected axis is reached, the automatic warm-up stops.  For axes that do not support temperature functions, have an overheating rate of 0, or are mechanically locked, the option is disabled and grayed out by default.
Axis temperature threshold	When the predicted temperature of the selected axis reaches the set threshold, the current speed switches to the global speed. If the predicted temperature of the robot axis is greater than or equal to the set threshold when the robot starts, the robot starts running at the global speed.
Warm-up time	When the actual robot running time reaches the set time, the robot switches to run at the global speed.
Initial warm-up speed	The initial speed when the robot performs automatic warm-up. If the global speed set is smaller than the initial warm-up speed, the robot runs at the global speed; otherwise, it runs at the initial warm-up speed.

Parameter	Description
Re-warm up interval	• If the robot is stopped during automatic warm-up and started again within this interval, the robot continues to warm up at the speed it was stopped. • If the robot is started again when this interval is exceeded, the robot starts running again at the initial warm up speed. • This interval record includes the power failure time. In case of power failure, the configuration file is loaded and the robot is restarted. If the stop time is within the re-warm up interval, the actual initial warm-up speed will be the same with the actual global speed before restart.

9.3.7.1 Steps

Example of parameter settings

Parameter	Value
1 - Warm-up policy	
Warm-up policy	Consider axis temperature and warm-up time
2 - Axis warm-up parameters	
Axis number	J1: R, J2: *, J3: *, J4: *, J5: *, J6: *
Axis temperature threshold (°C)	10
3 - Robot warm-up parameters	
Warm-up time (min)	10
Initial warm-up speed (%)	75
Re-warm up interval (min)	20

The temperature of J1 is detected. If the temperature is greater than or equal to 10℃, the automatic warm-up is not performed. The robot motion speed is the same as when this function is not enabled. Otherwise, proceed to the next step.

The robot starts running at a 75% initial warm-up speed. The speed will be increased evenly to the global speed set within 10 minutes. If the temperature of J1 reaches 10℃ within 10 minutes, the robot switches to run at the global speed set.

Run the robot for four minutes and then stop it. If the robot is started again within 20 minutes, the robot continues to warm up for six minutes starting from the speed it was stopped. If the robot is started again after 20 minutes, it will warm up for ten minutes again starting from the 75% initial warm-up speed.

9.3.7.2 Notes

- Automatic warm-up takes effect only when in auto mode and during motion instruction execution.
- When automatic warm-up is performed, a warm-up status prompt will be displayed.

- During automatic warm-up, the robot increases the speed evenly every 30 seconds and reaches the global speed set within the configured warm-up time.
- When the "Consider axis temperature and warm-up time" option is selected, ensure that the robot supports the temperature protection function and has finished temperature calibration without mechanical lock enabled.
- After you save the warm-up parameter settings, the robot starts warm-up with the parameters set.
- User, administrator and manufacture all have access to set the warm-up parameters.
- The external axis unit does not support the auto warm-up function.
- This function does not support control via bus and API.

9.4 Peripheral Configuration

9.4.1 Bus Configuration

9.4.1.1 Switch Settings

You can switch on or off five types of buses including Modbus, EtherNet/IP, EtherCAT, MC, and PROFINET, and configure the bus parameters.



Only one fieldbus can be activated at a time.

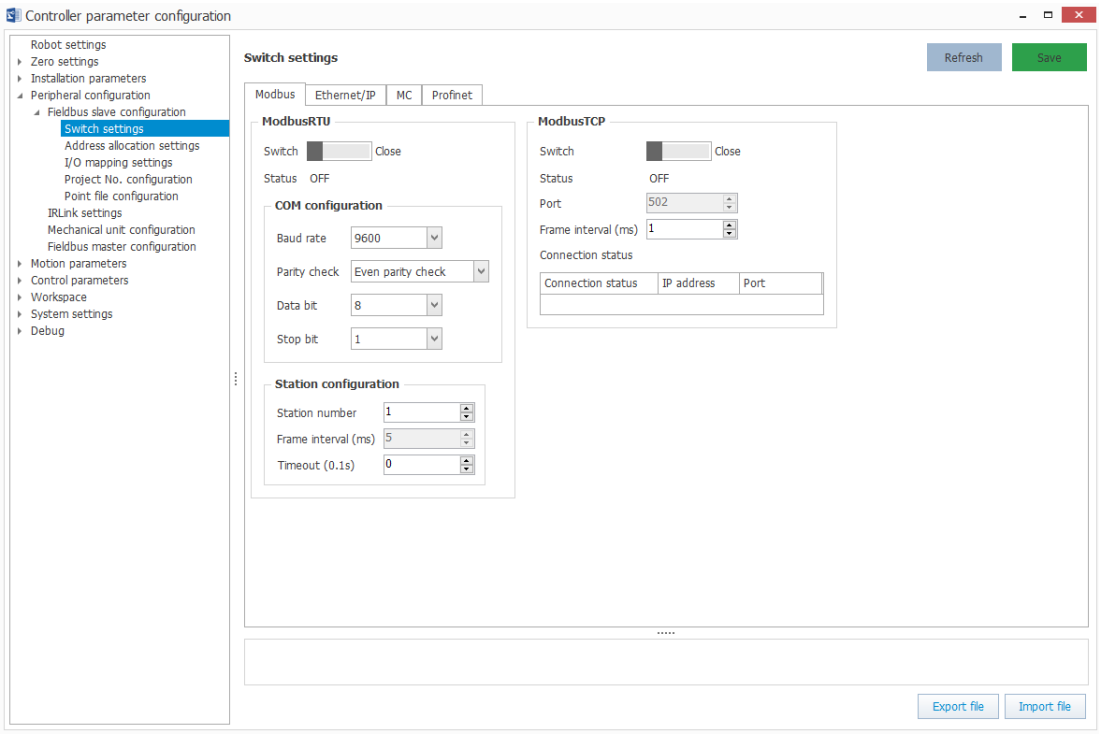
Modbus settings

Modbus-RTU and Modbus TCP functions of the robot controller can be enabled simultaneously without conflict.

Regardless of Modbus-RTU or Modbus TCP, the robot controller can only be configured as Modbus slave, not as Modbus master on InoRobotLab.

Steps

1. Double-click "Controller parameter configuration" in the device tree, then select "Peripheral configuration" > "Fieldbus slave configuration" > "Switch settings", as shown in the figure below.



2. Toggle the switch and click "Save" to turn on or off the Modbus-RTU and Modbus TCP functions.
3. In the Modbus-RTU function, the frame interval depends on the baud rate. The larger the baud rate, the smaller the frame interval. If the baud rate is changed, the frame interval will be automatically changed accordingly.
4. In the Modbus TCP function, the port number is fixed to 502 and the frame interval can be set as needed.

The order of toggle switch operation and parameter setting is not limited. The settings are saved to the controller only when you click the "Save" button.

For the Modbus TCP function, you can monitor the connection status on the configuration interface. For details, see the 7.10.2 Bus Connection Status section.

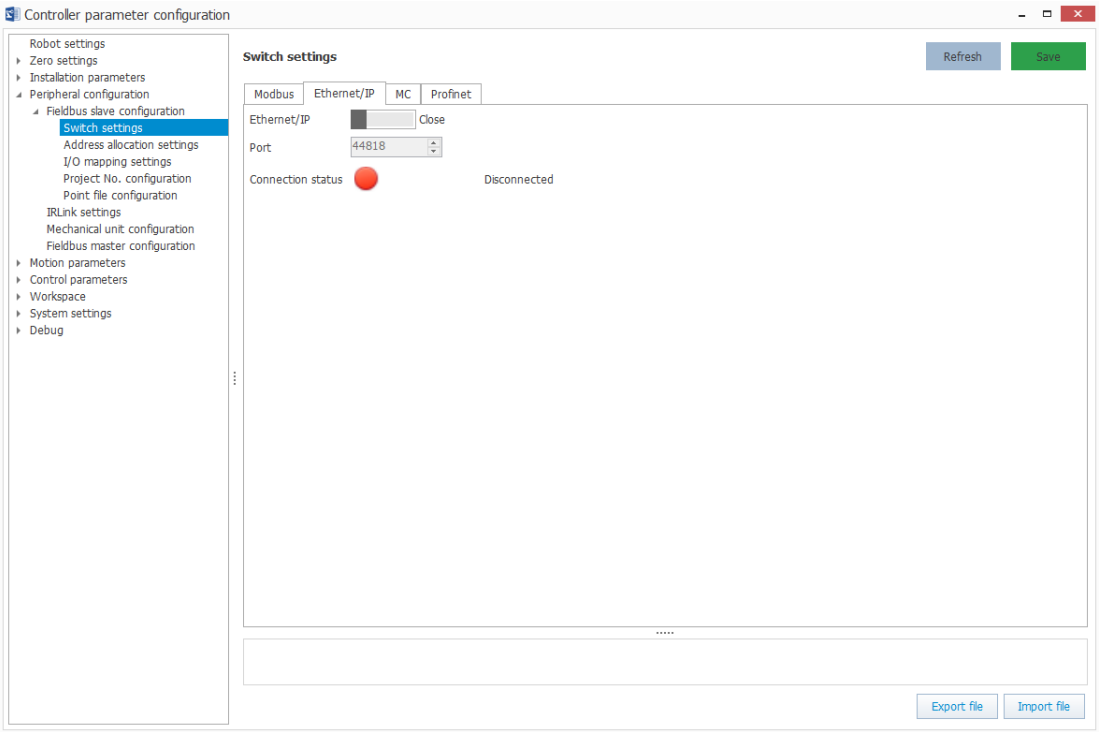
Results

After the configuration is complete, you can verify the configuration through communication with the Modbus client.

> NOTE
>
> For the usage of Modbus functions, see the "Modbus Bus Communication" section in the "Industrial Robot Communication Guide".

EtherNet/IP settings

You can enable or disable the EtherNet/IP slave function by toggling the switch and clicking the "Save" button in the upper right corner. The interface also displays the port number and connection status of the slave. When no connection exists, the indicator is red. When a master is connected, the indicator is green. The IP address and port number of the master are displayed.



For the usage of EtherNet/IP functions, see the "EtherNet/IP Bus Communication" section in the "Industrial Robot Communication Guide".

EtherCAT settings

The EtherCAT slave function can only be activated through encryption authorization on the controller that supports the EtherCAT slave function (see section 12.13 Function Activation for details).

You can enable or disable the EtherCAT slave function and configure the related parameters.

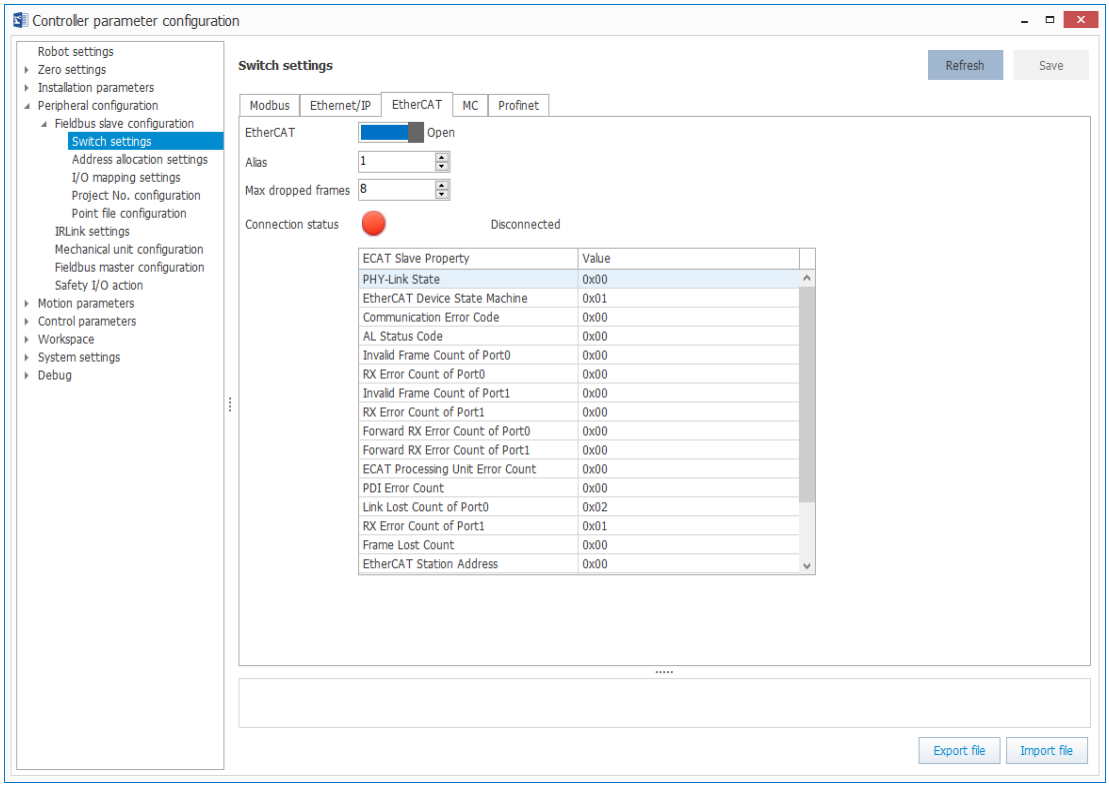
You can enable or disable the EtherCAT slave function by toggling the switch and clicking the "Save" button. You can also configure the alias of the slave and the maximum number of dropped frames.

The interface also displays the connection status of the EtherCAT slave. When no connection exists, the indicator is red. When a master is connected, the indicator is green.

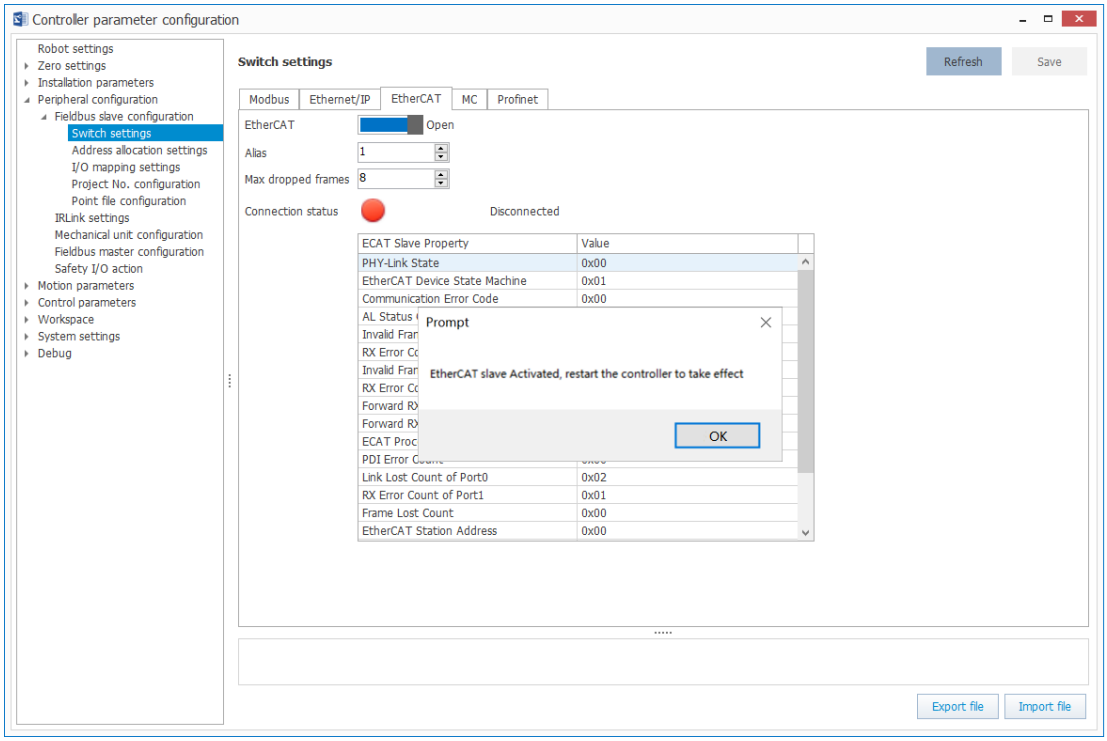
EtherCAT slave property	Description
PHY-Link State	The Link status of the port, where a value of 0 for the corresponding bit indicates LinkDown, and 1 indicates LinkUp. Bit0: status of port 0 Bit1: status of port 1
EtherCAT Device State Machine	Actual slave status 1: Init (initialization status) 2: PreOP (pre-operational status) 4: SafeOP (safe operational status) 8: OP (operational status)

EtherCAT slave property	Description
Communication error code	Communication error code, uploaded by MCU. The controller triggers the corresponding alarm if communication error code exists.
AL Status Code	Application layer status code, uploaded by MCU
Invalid Frame Count Of Port0	Invalid frame count received on port 0, uploaded by MCU
RX Error Count of Port0	Error frame count received on port 0, uploaded by MCU
Invalid Frame Count Of Port1	Invalid frame count received on port 1, uploaded by MCU
RX Error Count of Port1	Error frame count received on port 1, uploaded by MCU
Forward RX Error Count of Port0	Forwarded error frame count on port 0, uploaded by MCU
Forward RX Error Count of Port1	Forwarded error frame count on port 1, uploaded by MCU
ECAT Processing Unit Error Count	Data frame processing unit error count, uploaded by MCU
PDI Error Count	PDI error count, uploaded by MCU
Link Loss Count of Port0	Link loss count on port 0, uploaded by MCU
Link Loss Count of Port1	Link loss count on port 1, uploaded by MCU
Frame Lost Count	Accumulated frame loss count on slave, uploaded by MCU
EtherCAT Station Address	MCU uploads the site name configured by the master.
ESC Version	ESC version information, uploaded by MCU
XML Version	XML version information, uploaded by MCU
Software Version	Software version number, uploaded by MCU

The additional functions "Enhanced link function switch" and "XML reset switch" are available for the controllers with safety function.



After the alias is modified, the robot needs to be powered on again to make the modification take effect.



For the usage of EtherCAT functions, see the "EtherCAT Bus Communication" section in the "Industrial Robot Communication Guide".

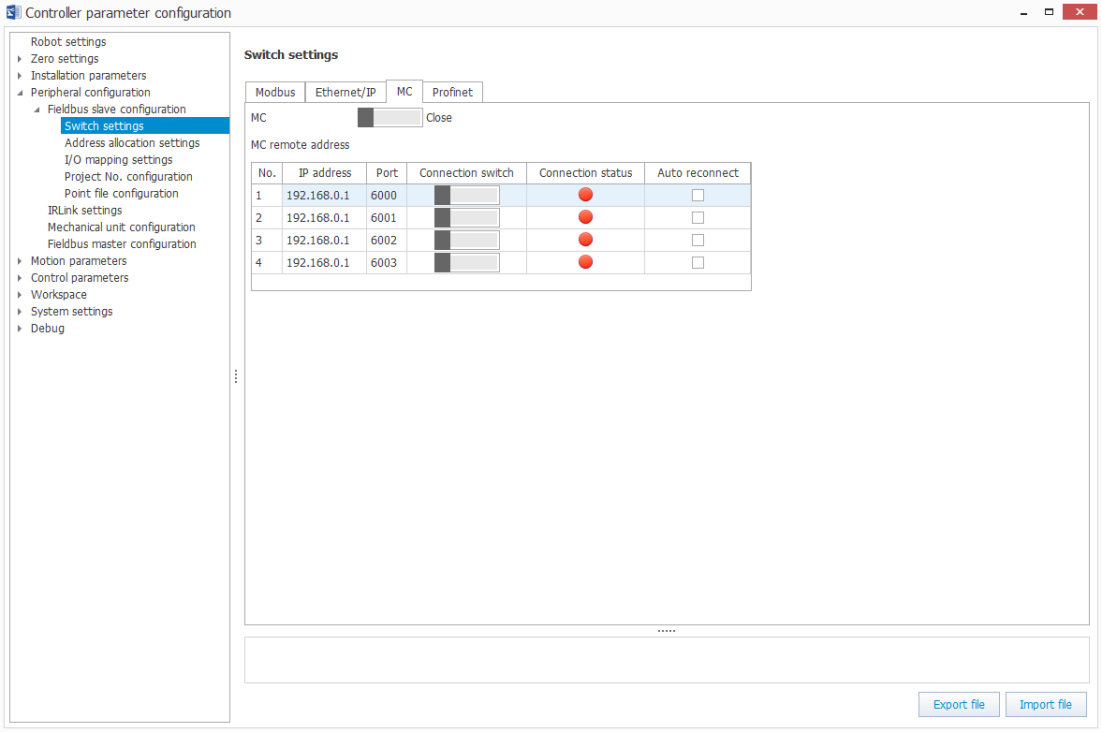
MC settings

You can enable the MC function of the robot controller. The robot acts as a slave.

Steps

Double-click "Controller parameter configuration" in the device tree, then select "Peripheral configuration" > "Fieldbus slave configuration" > "Switch settings" > "MC", as shown in the figure below.

In contrast to the previous three protocols, the MC function can be enabled or disabled directly through the toggle switch. The interface displays the IP address, port number, connection switch, connection status, and automatic re-connection switch of the four masters. You can establish a connection to the master by entering the IP address and port number of the master and then clicking the connection switch. The indicator is red when there is no master connection, and green when there is a master connection.



CAUTION

- After the master connection is successful, the IP and port numbers of the master cannot be modified. The auto re-connect option can be checked only when the master connection is successful.
- When the auto re-connect option is checked, the IP address, port number, and connection switch cannot be modified.

Results

After the configuration is complete, you can verify the configuration through communication with the MC client.

For the usage of MC functions, see the "MC Bus Communication" section in the "Industrial Robot Communication Guide".

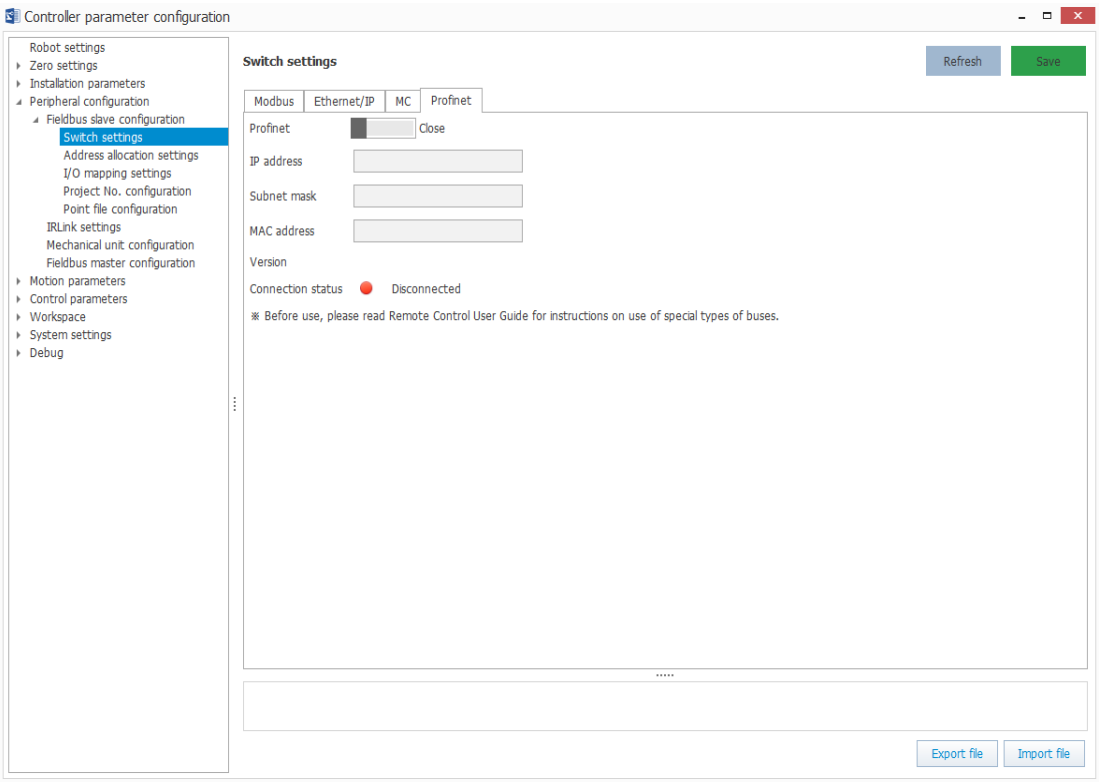
PROFINET settings

It allows you to configure the PROFINET slave function of the robot.

You can enable or disable the PROFINET slave function by toggling the switch and clicking the "Save" button.

The interface also displays the connection status of the PROFINET slave, the IP address, subnet mask, MAC address, and software version of the slave expansion card. The connection status indicator is red when the PROFINET slave function is not enabled or communication with the master fails. When communication with the master is normal, the connection status indicator is green.

> **CAUTION**
>
> To use the PROFINET slave function, the controller must be installed with an external PROFINET slave expansion card.



For the usage of PROFINET functions, see the "PROFINET Communication with a Siemens PLC"" section in the "Industrial Robot Communication Guide".

9.4.1.2 Address Allocation Settings

Before using the address allocation function, you need to understand the concept of the robot bus addresses first. See 7.4.1 Introduction of Robot Bus Addresses.

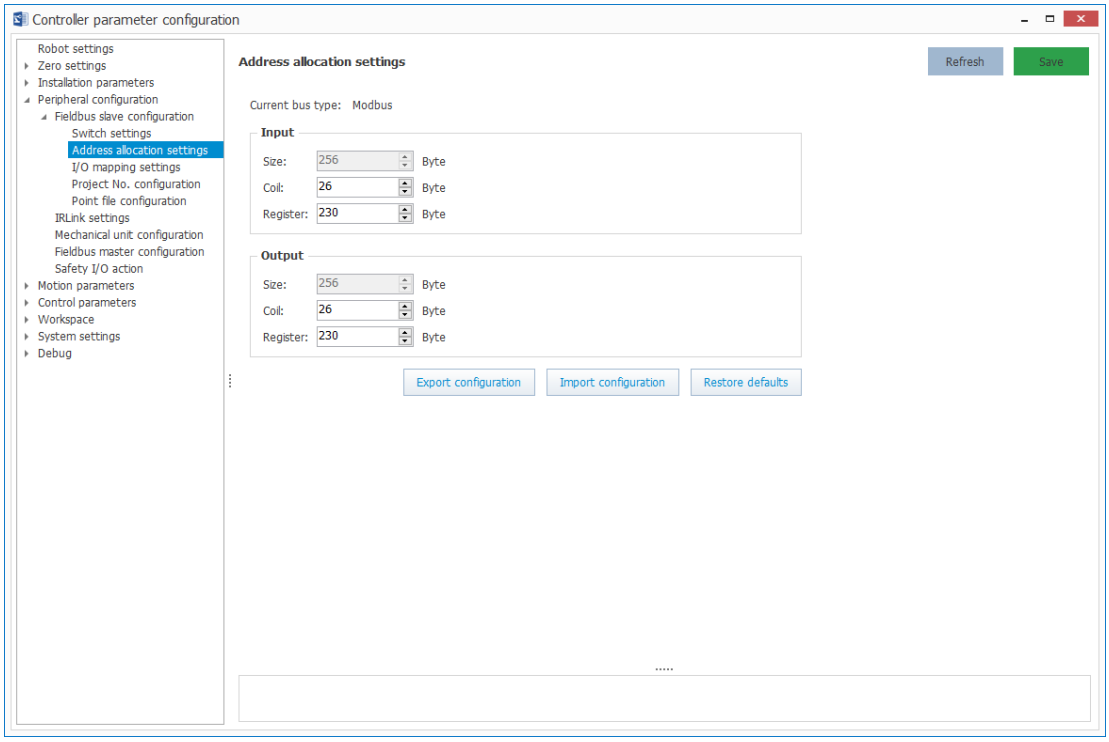
The so-called address allocation is to specify the range of fieldbus I/Os for communication with the master. Only I/O with an assigned address can interact with the master (PLC or HMI in most cases) to input and output data. The following takes Modbus, EtherNet/IP, EtherCAT, and MC functions as examples to describe the address allocation.

Modbus Address Allocation

When the Modbus-RTU or Modbus TCP function is activated, the current bus type is displayed as Modbus.

You can change the number of bytes of the allocated coil and register. The bytes are automatically added to the total size of the input address. By default, the size of coil is 26 bytes and the size of the register is 230 bytes. The relationship between the Modbus address table and the robot bus address is as follows when the default configuration is used.

Modbus address	Robot address
Coil 4096-4303 (208 coils, 26 bytes)	In512-In719 (208 bits, 26 bytes)
Register 34800-34914 (115 registers, 230 bytes)	In720-In2559 (1840 bits, 230 bytes)
Coil 0-207 (208 coils, 26 bytes)	Out512-Out719 (208 bits, 26 bytes)
Register 2048-2162 (115 registers, 230 bytes)	Out720-Out2559 (1840 bits, 230 bytes)



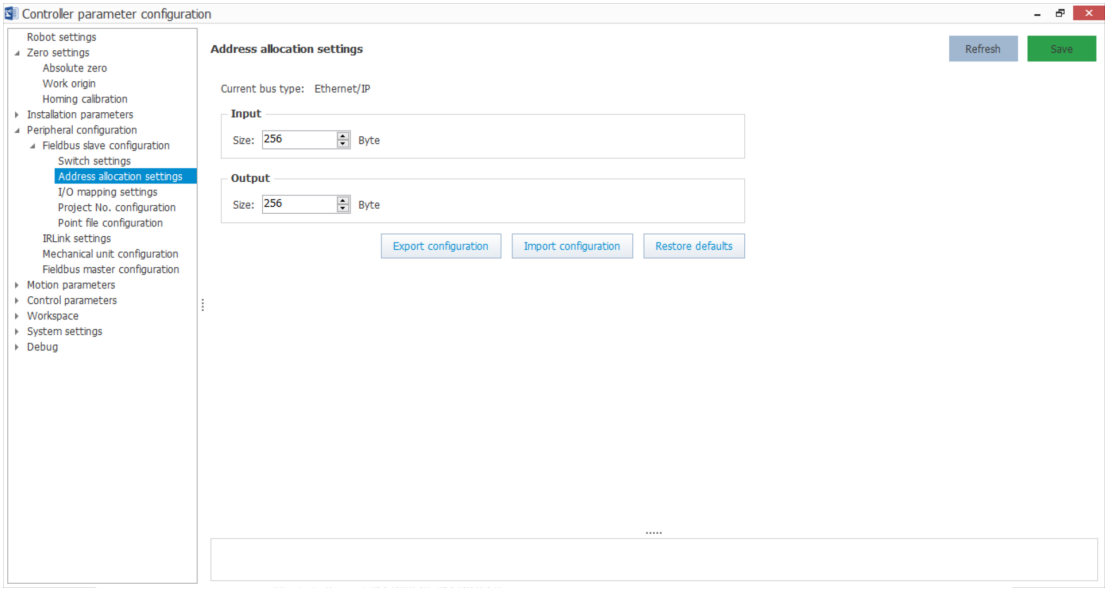
EtherNet/IP Address Allocation

When the EtherNet/IP function is activated, the current bus type is displayed as EtherNet/IP.

You can change the number of bytes of the input and output addresses. By default, the size of input/output address is 256 bytes. The relationship between the EtherNet/IP address table and the robot bus address is as follows when the default configuration is used.

(m is the start word address of the data transmitted to the robot by the EtherNet/IP master, and n is the start word address of the data transmitted to the EtherNet/IP master by the robot.)

EtherNet/IP address	Robot address
m to m+127 (128 words, 256 bytes)	In512-In2559 (2048 bits, 256 bytes)
n to n+127 (128 words, 256 bytes)	Out512-Out2559 (2048 bits, 256 bytes)



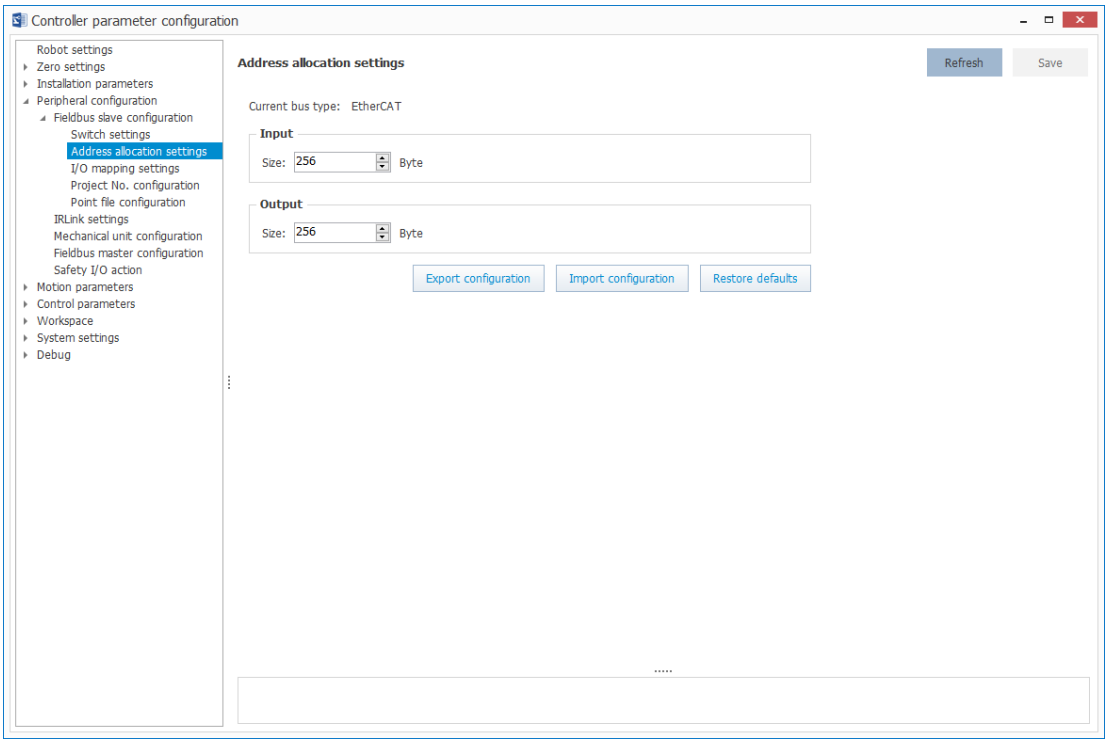
EtherCAT Address Allocation

When the EtherCAT function is activated, the current bus type is displayed as EtherCAT.

You can change the number of bytes of the input and output addresses. By default, the size of input/output address is 256 bytes. The relationship between the EtherCAT address table and the robot bus address is as follows when the default configuration is used.

(p is the first object of RxPDO in the EtherCAT protocol, and q is the first object of TxPDO in the EtherCAT protocol.)

EtherCAT address	Robot address
p to p+127 (128 RxPDOs, 256 bytes)	In512-In2559 (2048 bits, 256 bytes)
q to q+127 (128 TxPDOs, 256 bytes)	Out512-Out2559 (2048 bits, 256 bytes)



MC Address Allocation

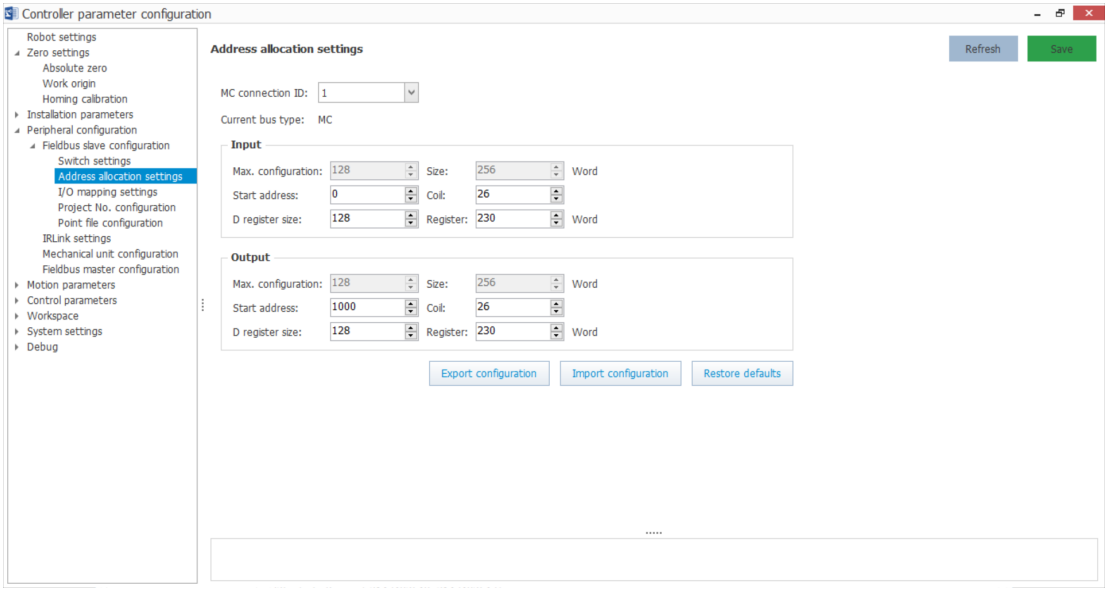
When the MC function is activated, the current bus type is displayed as MC. In addition, an MC connection ID is displayed, indicating the number of the current master.

You can change the number of bytes of the start address and D register for the input and output. By default, the size of input/output address is 256 bytes. The size of the input/output D register is between 2 and 256 bytes. The size must be an even number. The sum of the number of bytes of the start address and the D register of the input must be smaller than the start address of the output. The relationship between the MC address table and the robot bus address is as follows when the default configuration is used.

MC address	Robot address
D register D0-D255 (128 registers, 256 bytes)	In512-In2559 (2048 bits, 256 bytes)
D register D1000-D1255 (128 registers, 256 bytes)	Out512-Out2559 (2048 bits, 256 bytes)



According to technical data of Mitsubishi PLC, the maximum number of soft components is 40,000. Therefore, the sum of the number of bytes of the start address and the output D register should be less than or equal to 40,000. The configuration page only limits that the input/output start address is less than or equal to 40,000, and the user need to ensure on their own that the address size is correct. Input start address + Byte count of input D register ≤ Output start address < Output start address + Byte count of output D register ≤ 40000.



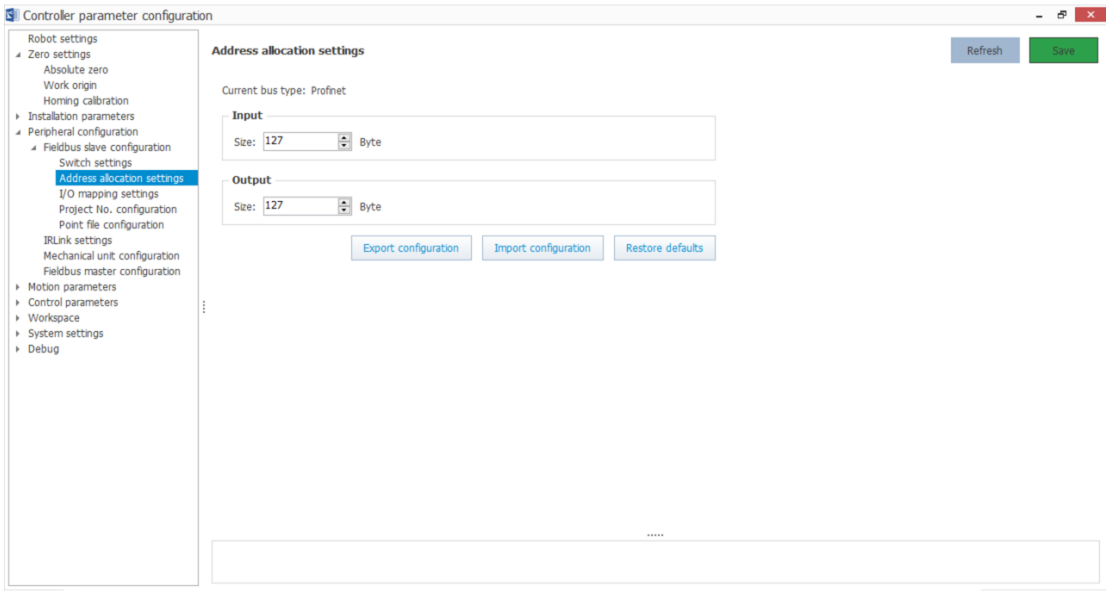
PROFINET Address Allocation

When the PROFINET function is activated, the current bus type is displayed as PROFINET.

You can change the number of bytes of the input and output addresses. By default, the size of input/output address is 127 bytes. The relationship between the PROFINET address table and the robot bus address is as follows when the default configuration is used.

(In Siemens S300, S1200, and S1500 series PLCs, Q is the first object of the PROFINET output register, and I is the first object of the PROFINET input register.)

EtherCAT address	Robot address
Q to Q+126 (127 registers, 254 bytes)	In512-In2543 (2032 bits, 254 bytes)
I to I+126 (127 registers, 254 bytes)	Out512-Out2543 (2032 bits, 254 bytes)

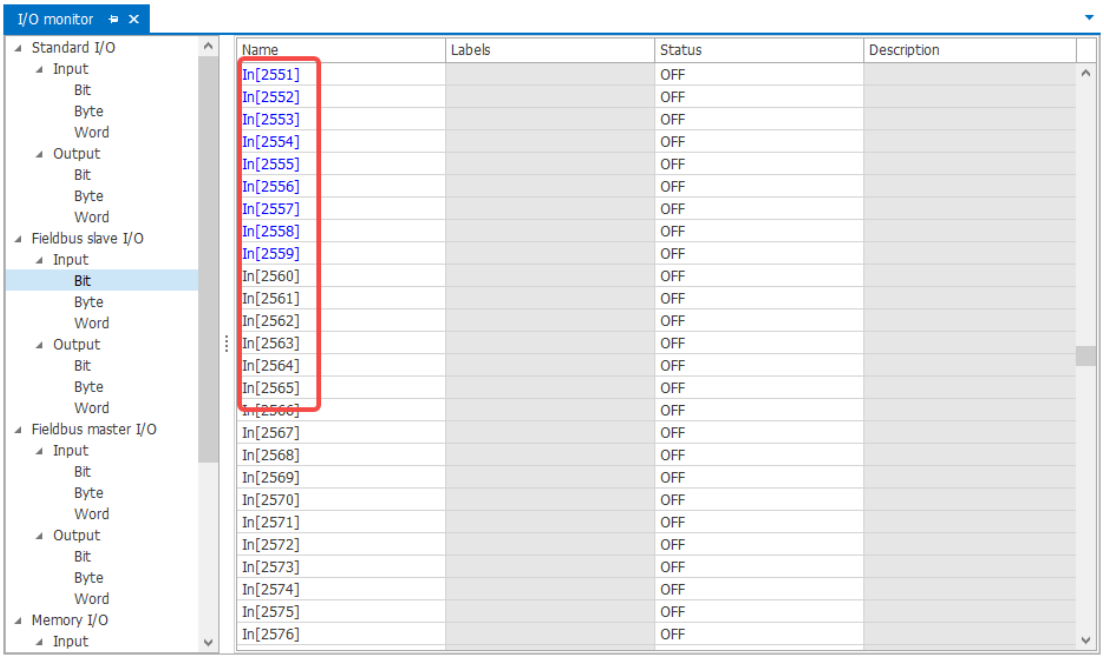


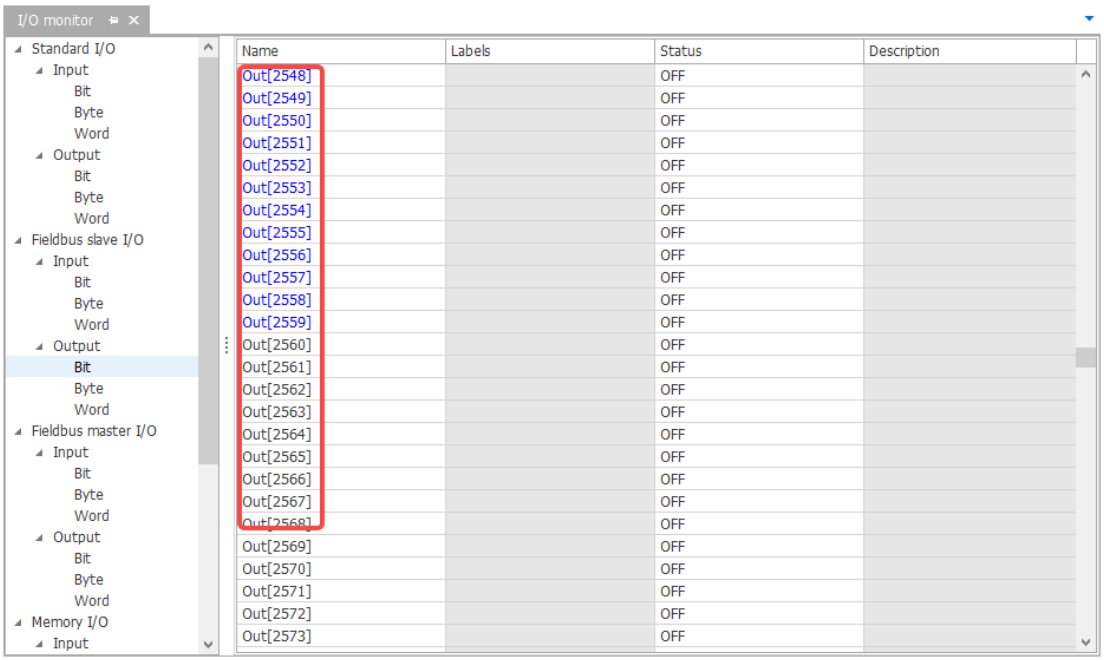
CAUTION

To ensure the integrity of the PROFINET communication data, ensure that the address length of the master is the same with that of the robot. When an address length difference is detected, H3049 alarm (which cannot be cleared) will be displayed on the robot and a prompt message is printed as follows on the InoRobotLab interface. To clear the alarm, eliminate the address length difference.

```
[Message]: 12:30:01 Prompt: PROFINET bus data address length [input: 64, output: 64] does not match PLC configuration length [input: 127, output: 127].
[Message]: 12:30:01 Prompt: Saving configuration file...
[Message]: 12:30:02 Prompt: System configuration file saved successfully.
[Message]: 12:30:02 Command to save the bus address allocation configuration to the controller was sent successfully.
[Warning]: 12:30:04 Warning: 0x3049, The master and slave data of the Profinet expansion card are inconsistent in length.
```

For the convenience of identification, the fieldbus I/O assigned with addresses are displayed in blue, while those not assigned with addresses are displayed in black. For details, see 7.4.4 System I/Os.





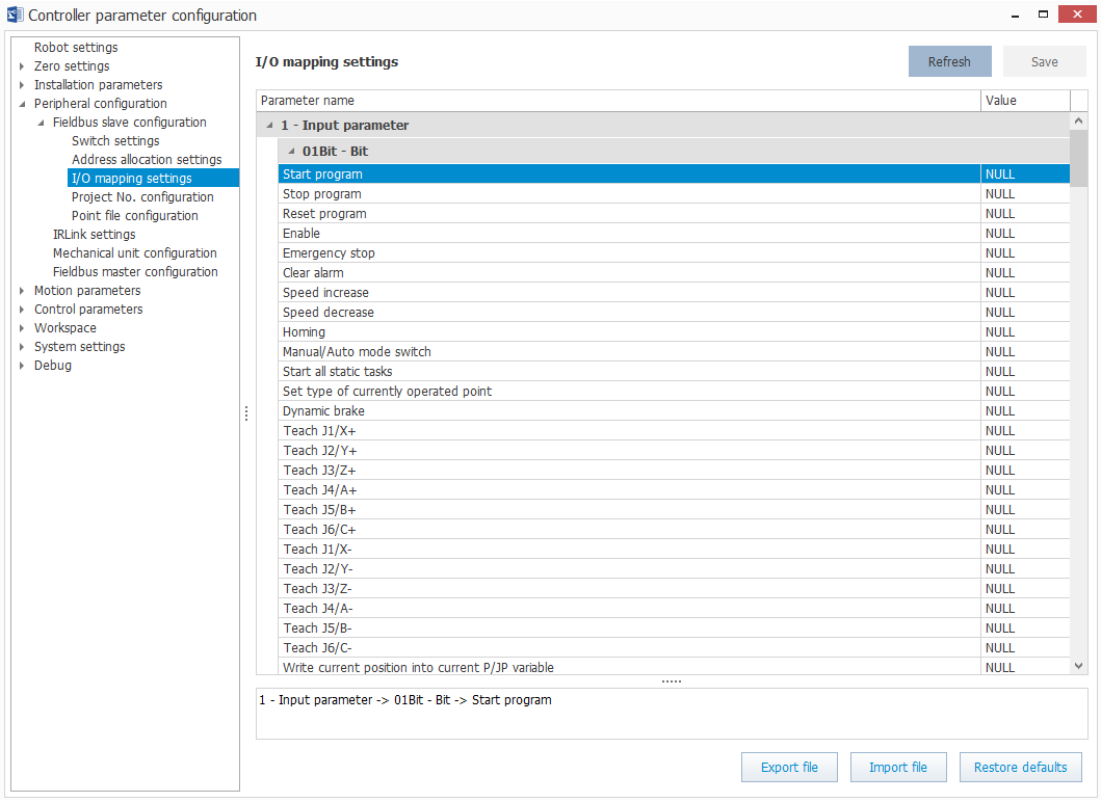
9.4.1.3 I/O Mapping Settings

Remote I/O control allows you to control program start/stop, reset, and emergency stop of the robot via I/Os (including digital I/Os, fieldbus I/Os, and memory I/Os). You can also monitor the program running and fault state of the robot.

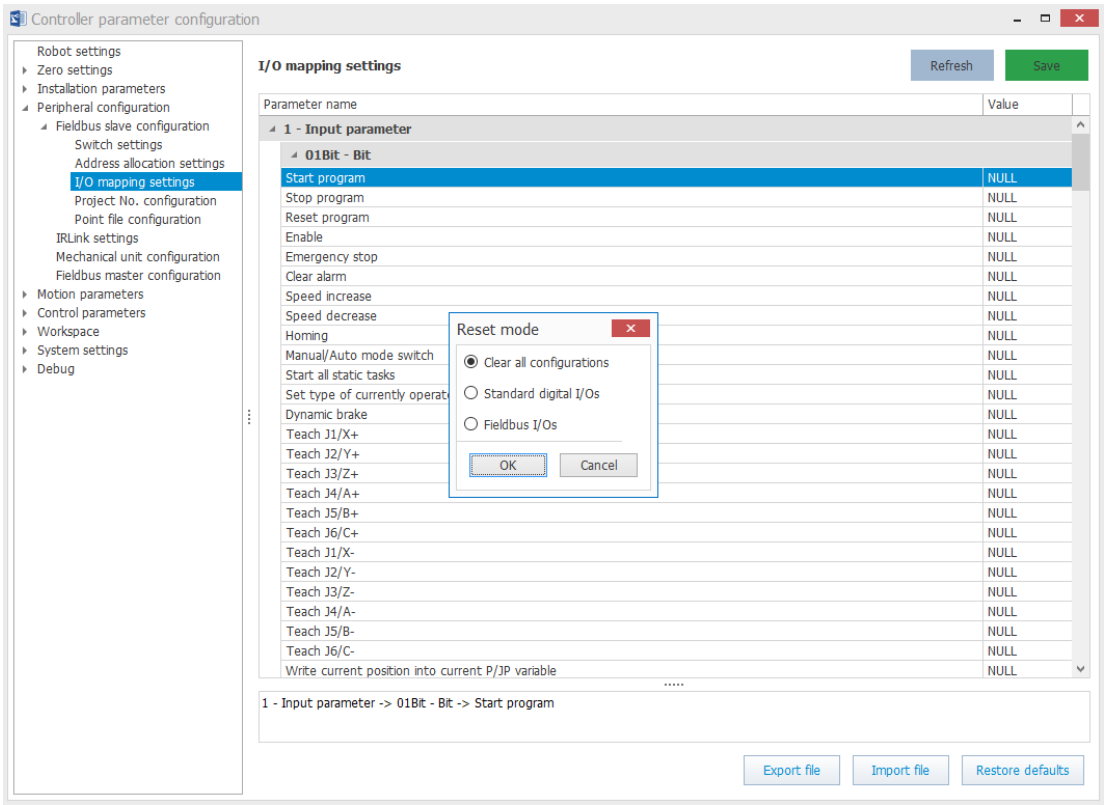
The I/O mapping is to bind some commands of the remote I/O control function to the input signals (In) of the robot, and the status of the robot to the output signals (Out). In this way, you can control the robot using the input signals and monitor the robot using the output signals.

Steps

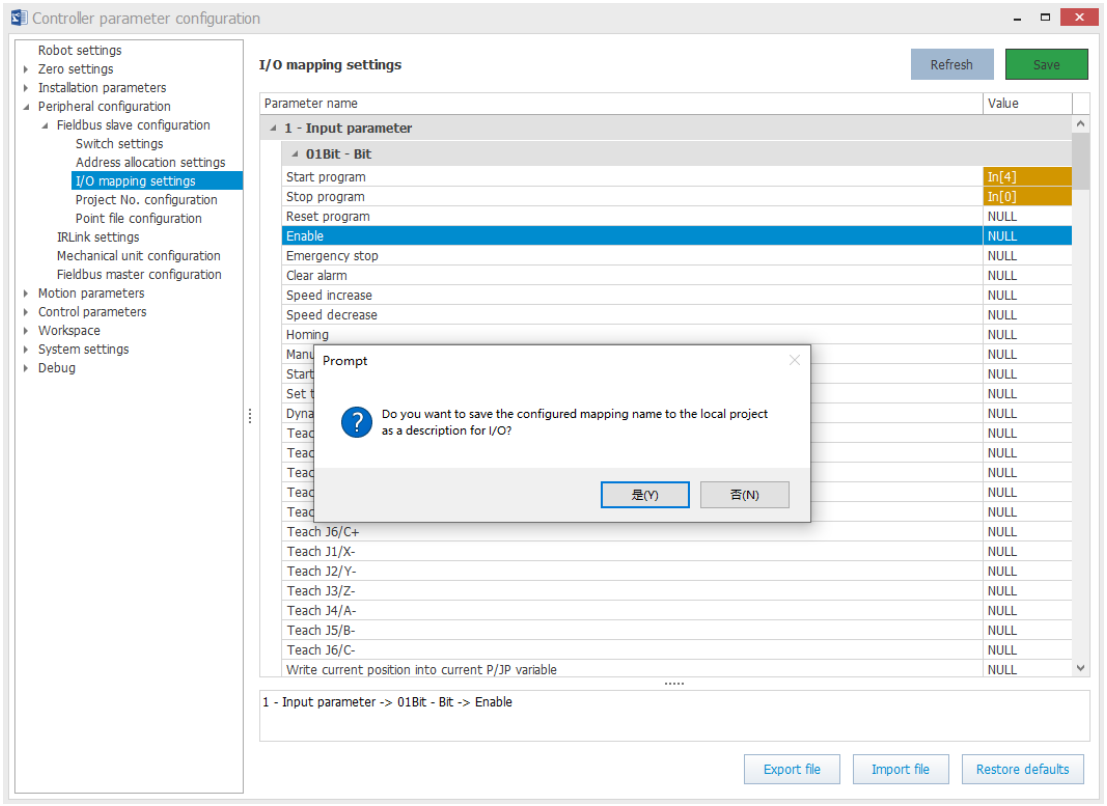
1. Double-click "Controller parameter configuration" in the device tree, then select "Peripheral configuration" > "Fieldbus slave configuration" > "I/O mapping settings", as shown in the figure below.



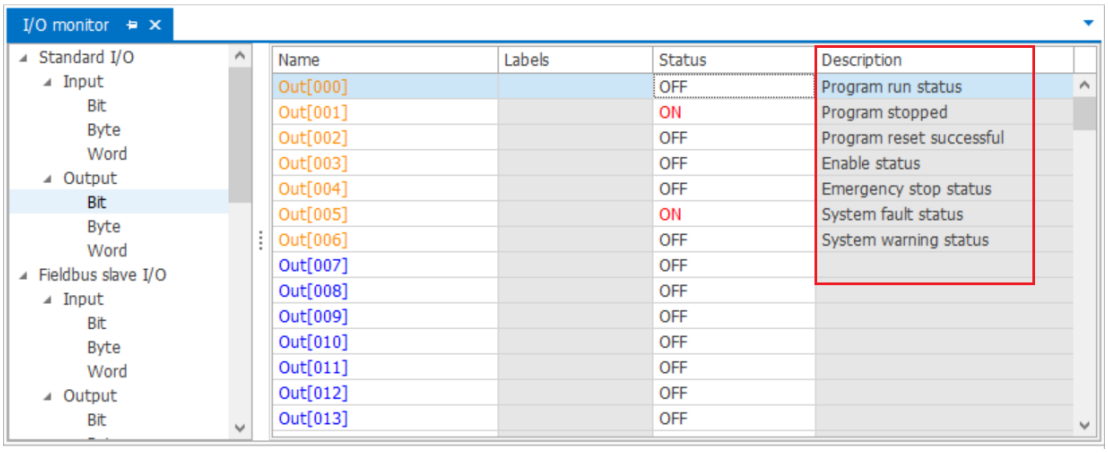
2. The default I/O mapping is empty. To facilitate user operation, two sets of default I/O configurations are provided. One group is standard digital I/Os and the other group is fieldbus I/Os. You can select the default I/O configurations by clicking the "Restore defaults" button in the lower right corner. You can also choose "Clear all configurations" to clear the existing setting values.



3. If you select "Standard digital IO" or "Fieldbus IO", the default I/O mapping is generated. You can adjust the default mapping as needed and click the "Save" button. When a project is currently open in InoRobotLab, a pop-up prompt displays after you click the "Save" button.



4. Click "Yes" to save the settings. Then you can see the auto-generated I/O description on the monitoring interface. You can also view the auto-generated I/O description in the I/O labels of the project file.



For the configuration of standard digital I/O mapping and fieldbus I/O mapping, see the "I/O mapping settings" section in the "Industrial Robot Remote I/O Control Function User Guide".

9.4.1.4 Project Number Configuration

See the "Program Number Configuration" section in the "Industrial Robot Remote I/O Control Function User Guide".

9.4.1.5 Point File Configuration

It allows you to bind a serial number to a point file to create a mapping between the two. At most 128 mappings are supported. In this way, you can switch the point files, or read/write point data of a specified point file via the fieldbus.

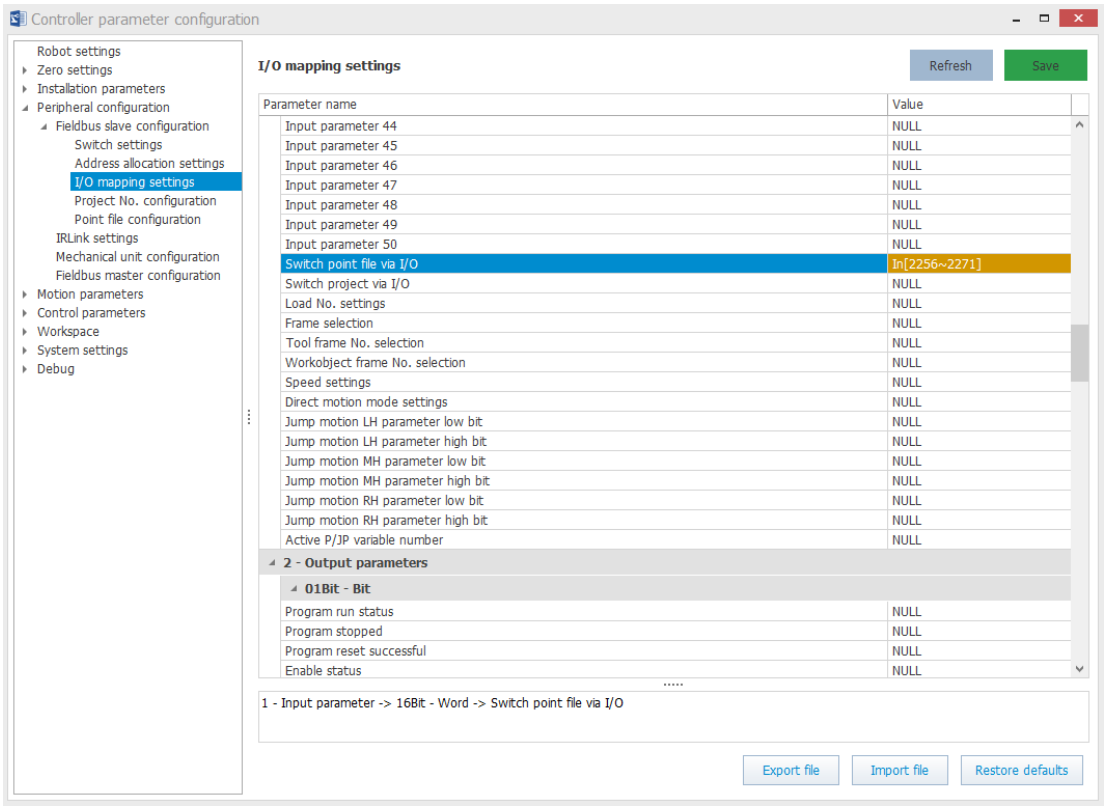
Steps

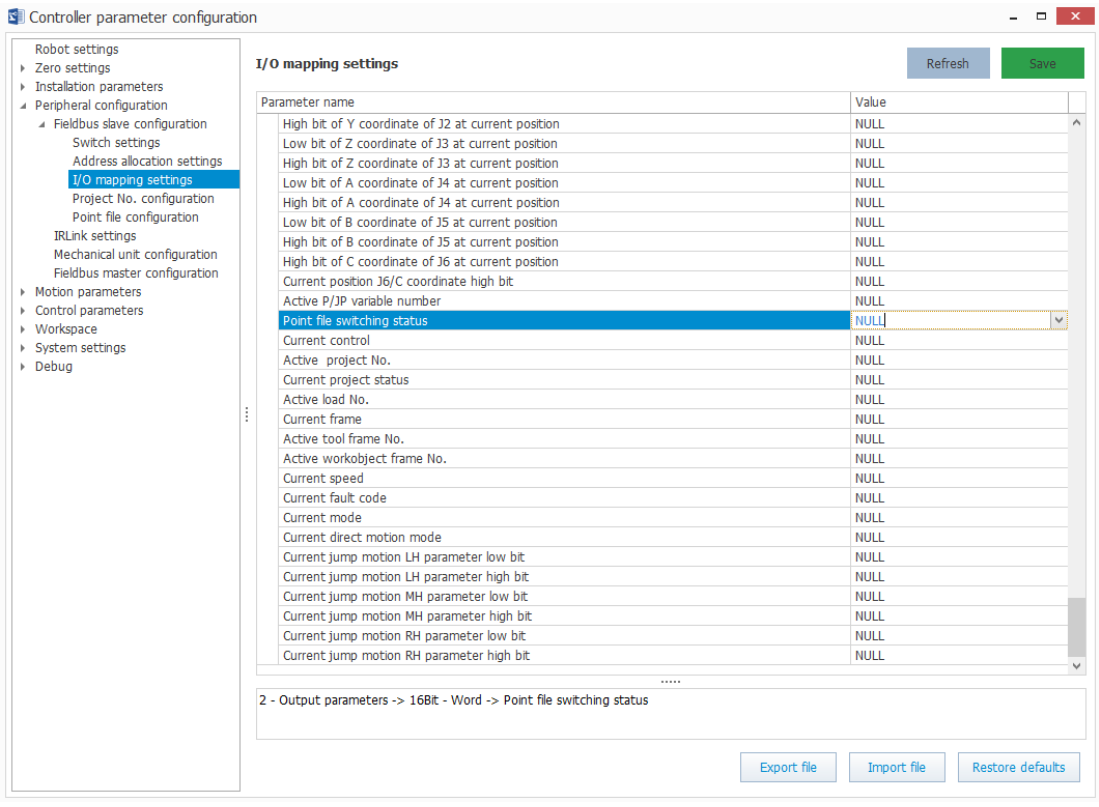
1. I/O Mapping Settings

(1) Switching the point file via the fieldbus

The "Switch point file via I/O" and "Point file switching status" parameters are configured as shown in the figure below.

1000-1127: Position point file with corresponding serial number. By default, 1000 is P.pts (thousand's place: Position point file type, unit's place to hundred's place: Point file number). The file number can be configured in the interface, range 0-127.



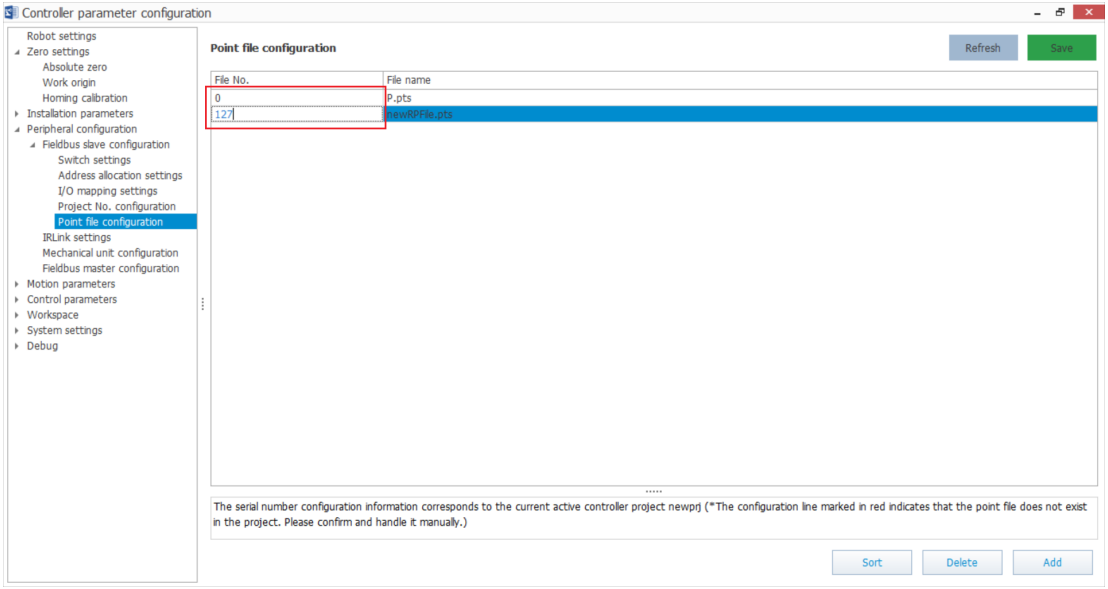


(2) Reading/Writing point data of a specified point file via the fieldbus

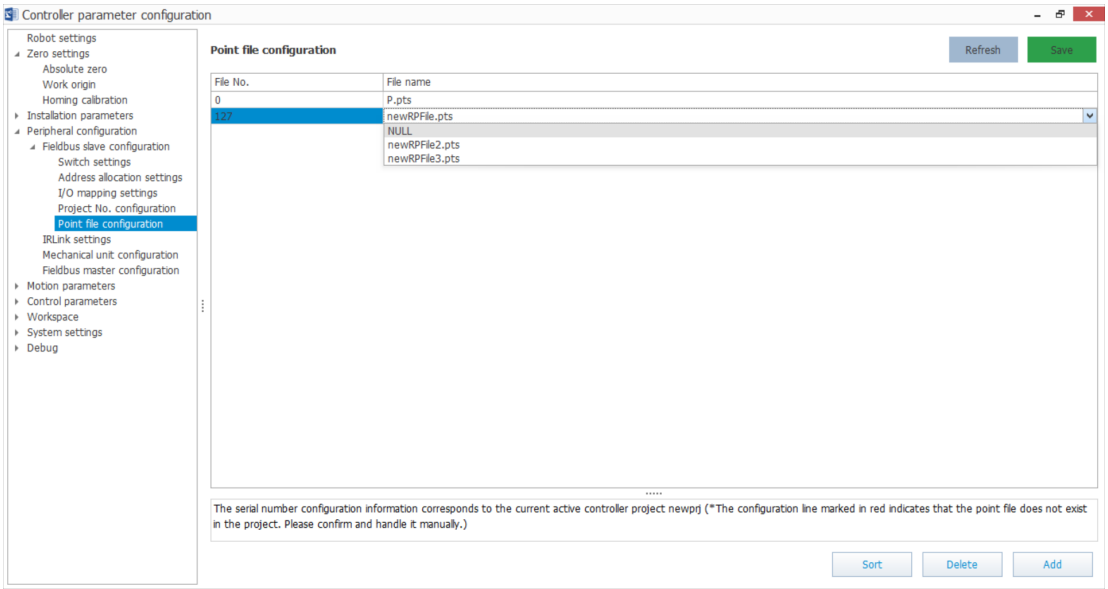
See the "Appendix 2: Extended Remote Control Command List" section in the "Industrial Robot Remote I/O Control Function User Guide".

2. Configure the point file serial number

- Go to Controller parameter configuration > Peripheral configuration > Point file configuration. The configuration page automatically loads the list of point files contained in the current active project of the controller.
- Refresh: Refreshes the list of point files contained in the current active project of the controller.
- Add: Click "Add" to add a point file number mapping. The number is automatically sorted from small to large. The point file name is "NULL" by default.
- File No.: Range 0-127. Click to edit the file number. The number must be unique. The serial number of "P.pts" file is 0 by default. This mapping cannot be modified.



- Select the point file: Select the point file from the drop-down list. Point files that have already been configured with a serial number are not displayed.



- Sort: Sorts the serial number from the smallest to the largest.
- Delete: Deletes the selected one or more lines of the list.

- Multi-selection: Same as file selection operation in Windows system. Select All: Ctrl+A; Continuous multi-selection: Shift+Click; Cumulative multi-selection: Ctrl+Click.
- Save: Saves the mapping between the serial number and the point file to the controller. It is recommended to click "Sort" before saving.

CAUTION

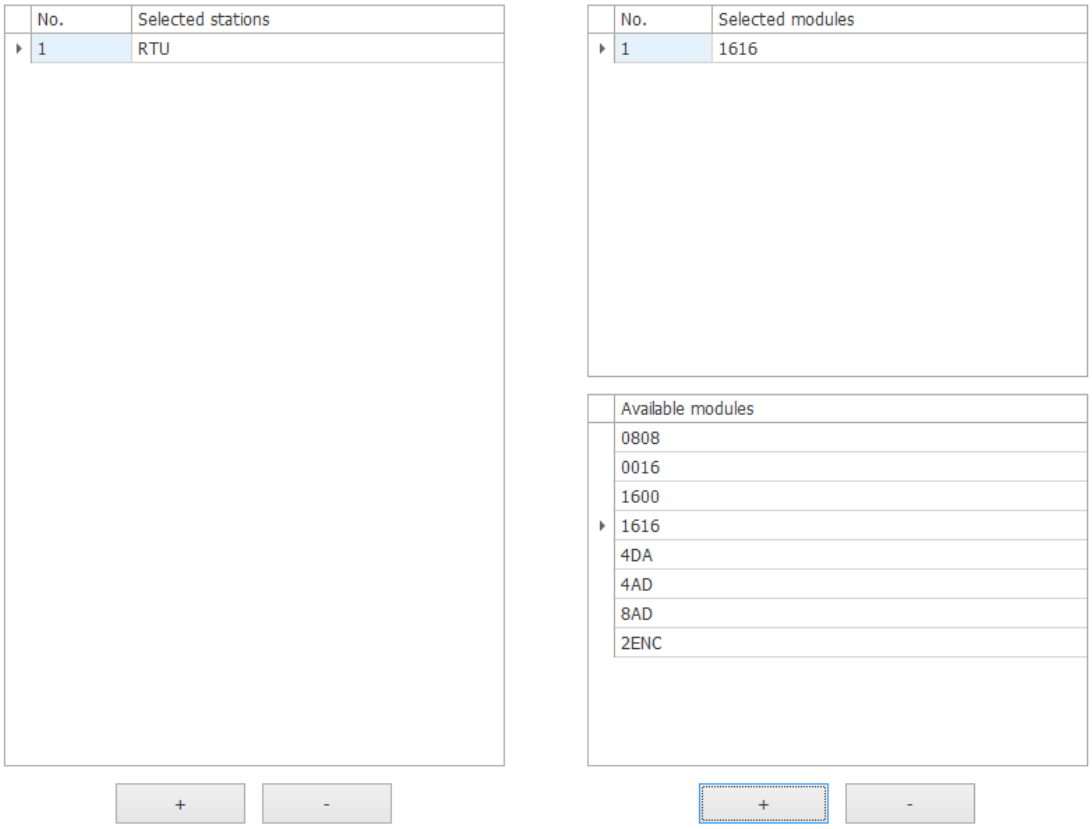
- If you do not click "Save", when you close the controller parameter setting interface and reopen it, the interface will be restored to the data before modification.
- If the local project name matches the current active project name of the controller, the serial number configuration will be saved to the local project. If not, a message will be prompted in the output bar that the serial number configuration is saved only to the controller.

9.4.2 IR-Link Settings

IR-Link is a customized communication protocol used by the IMC100 series expansion modules. This protocol is used to manage the networking pattern and parameter settings of expansion modules such as I/O, AD, DA, and encoder. Configuring IR-Link in InoRobotLab requires IR-Link configuration permission.

Steps

1. Double-click "Controller parameter configuration" in the device tree, then select "Peripheral configuration" > "IRLink settings", as shown in the figure below.

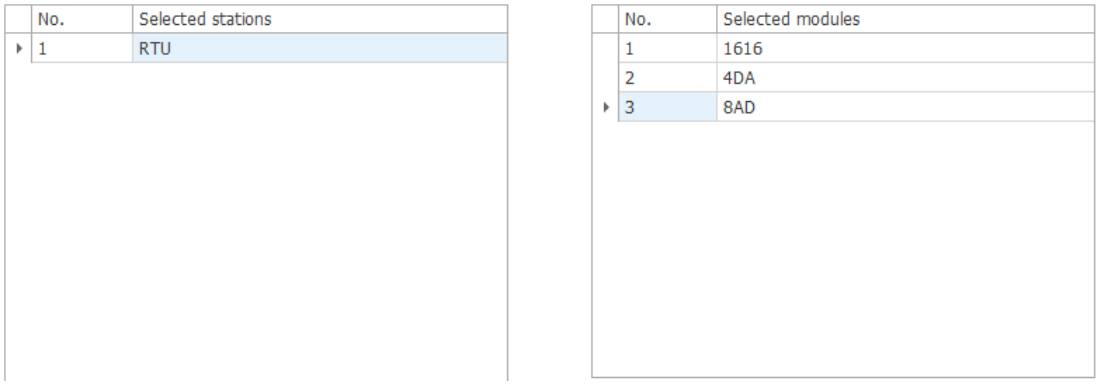


2. The remote terminal unit (RTU) is a station on the IR-Link link. Under the RTU, I/O modules (0808 type module with 8 inputs and 8 outputs, 0016 type module with 16 outputs, 1600 type module with 16 inputs, 1616 type module with 16 inputs and 16 outputs), AD modules (4AD type module with 4 analog inputs,

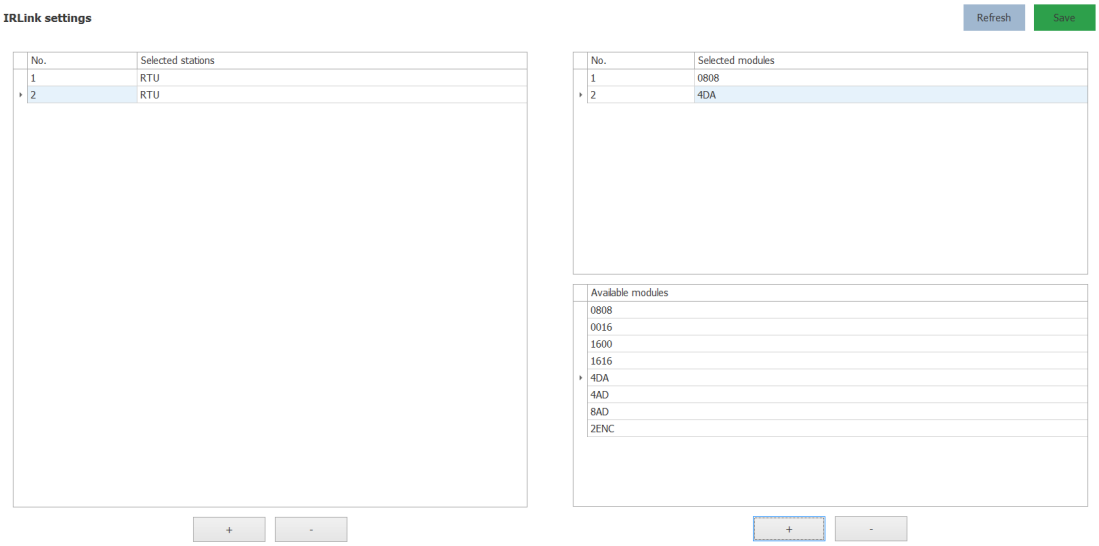
8AD type module with 8 analog inputs), DA modules (4DA type module with 4 analog outputs), and ENC modules (2ENC type module with 2 encoder inputs) can be mounted.

3. Click the "+" and "-" buttons on the left side of the interface to add or delete RTU stations. Click the "+" button on the right side to add the selected module to the selected RTU, and the "-" button to delete the selected module.

4. As shown below, when you select an RTU, the modules configured for that RTU are displayed.

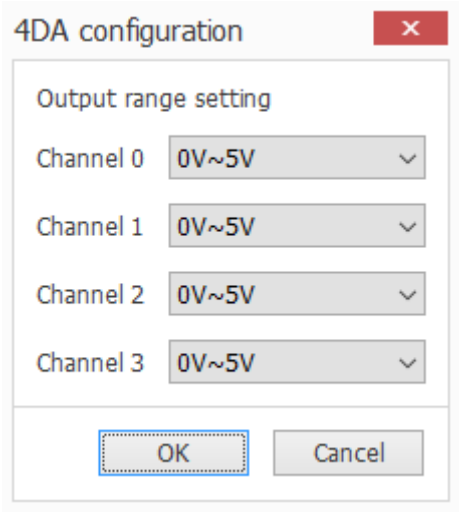


Module configuration displayed when RTU_1 is selected

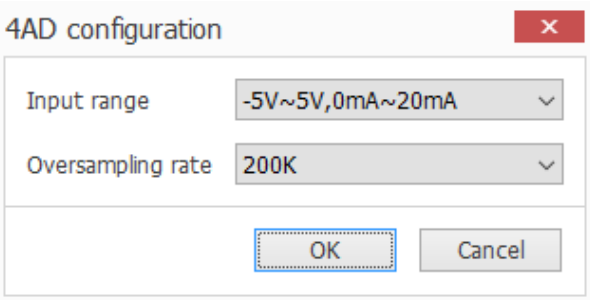


Module configuration displayed when RTU_1 is selected

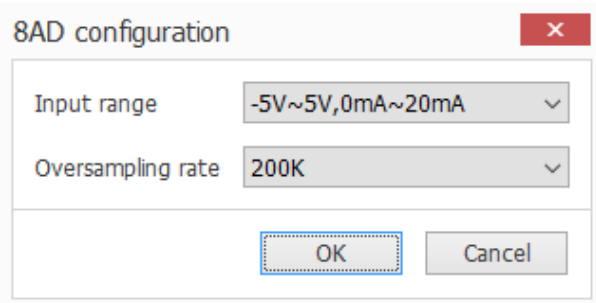
5. As shown below, when you double-click the selected 4DA, 4AD, 8AD, or 2ENC modules, the corresponding configuration interface appears. For the 0808, 0016, 1600, and 1616 modules, no additional configuration is required.



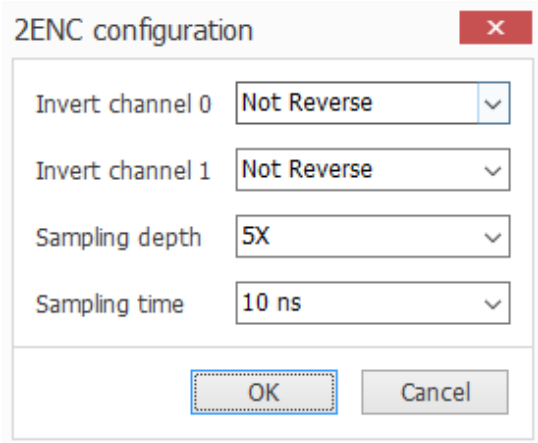
4DA module configuration interface



4AD module configuration interface



8AD module configuration interface



2ENC module configuration interface

For the 4DA module, you can set the analog output range of the four channels.

For 4AD and 8AD modules, there are two input ranges: default input current range 0 mA to 20 mA for input voltage range -5 V to +5 V; and default input current range 0 mA to 40 mA for input voltage range -10 V to +10 V. The oversampling rate affects the sampling accuracy of the input value. The lower the oversampling rate, the more accurate the value is. The oversampling rate of 3.125K is recommended.

For the 2ENC module, you can set whether to invert the channel and the signal filter time. The filter time is the product of the sampling depth and sampling time. The longer the filter time, the lower the rated signal input frequency.

Results

After the configuration is complete, save the configuration and restart the controller to initialize the communication. You can check whether the number of resources displayed on the I/O monitor and AD monitor pages is the same as the IR-Link configuration made and verify if the configuration takes effect.

- Power consumption specifications

For IRCB10 series controller, IR-Link resources can only be added through the RTU expansion module. Each RTU has a power limit, and the controller has a total IR-Link resource limit.

For IRCB300, IRCB500 and IRCB501 series controllers, IR-Link resources can be added through "expansion card + RTU". For each controller, four expansion cards can be added without power limit. However, they are subject to the total IR-Link resource limit of the controller.

Each RTU provides 15 W power output. Make sure the sum of the power consumption of the modules mounted to each RTU do not exceed the maximum power output of the RTU. The power consumption specifications of the modules are described as follows.

Type	Power consumption (W)
IMC100-0808-ETND	1.44
IMC100-1600-END	1.25
IMC100-0016-ETPD	1.25
IMC100-0016-ETND	1.25
IMC100-4DA	1.44
IMC100-8AD	2.88
IMC100-2ENID	2.88

- Resource specifications

The total number of IR-Link resources supported by the robot controller is limited and must not be exceeded. The number of resources represented by each module is as follows.

Module	Number of digital inputs	Number of digital outputs	Number of analog inputs	Number of analog outputs	Number of encoder channels
Max controller IR-Link capacity	64	64	16	16	8
Channels occupied per 1616 module	16	16	-	-	-
Channels occupied per 0808 module	8	8	-	-	-
Channels occupied per 1600 module	16	-	-	-	-

Module	Number of digital inputs	Number of digital outputs	Number of analog inputs	Number of analog outputs	Number of encoder channels
Channels occupied per 0016 module	-	16	-	-	-
Channels occupied per 8AD module	-	-	8	-	-
Channels occupied per 4AD module	-	-	4	-	-
Channels occupied per 4DA module	-	-	-	4	-
Channels occupied per 2ENC module	-	-	-	-	2

- For IRCB300 series controller, a 1616 module is included in the controller.

For IRCB100-6AT series controller, one 1616 module, one 0016 module, and one 1600 module are included in the controller.

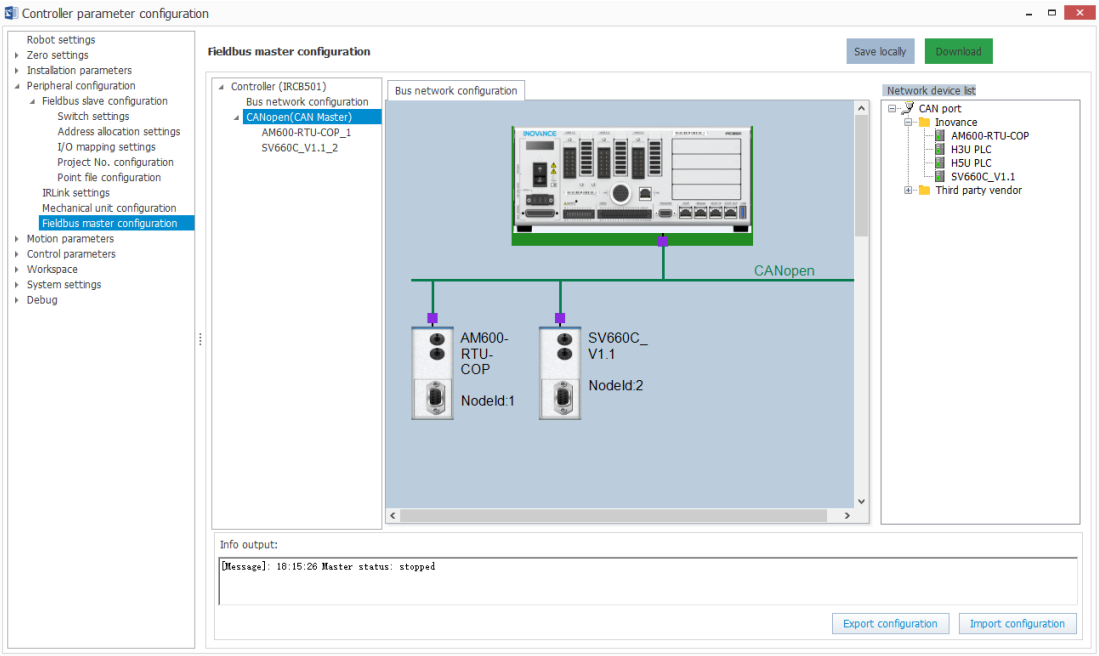
9.4.3 Mechanical Unit Configuration

See the 12.2 External Axis section.

9.4.4 Fieldbus Master Configuration

9.4.4.1 CANopen Master Configuration

It allows you to complete bus network configuration in a graphical way, connecting fieldbus devices to the controller.



Bus configuration box

By default, only a controller is displayed in the bus network configuration interface. When you click the controller icon, a bus selection box is displayed on the right side of the controller. You can check the bus type in the bus selection box.

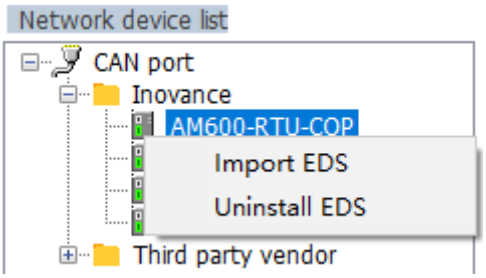
CANopen master: The robot controller acts the master and external devices act as slaves. The master and slaves communicate using CANopen protocol through the CAN interface. For example, the typical applications include the controller controlling servo and welding machine. When "CANopen master" is checked, the CANopen bus is marked with a green line.

Configuring devices on the bus

On the "Network device list" pane, double-click (or click and drag) a device to add it to the bus. To delete a device on the bus, select the device and press "Del" (or right-click and select "Delete").

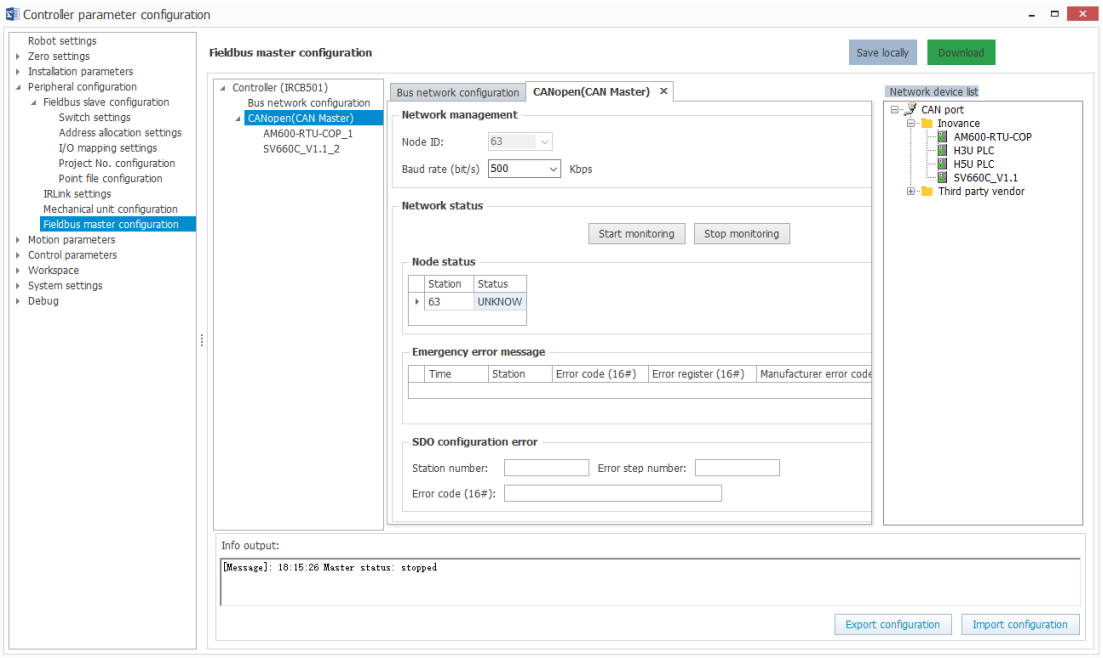
Importing the device

If a device you want to use is not included on the network device list, right-click the list and select "Import EDS" to import the EDS file. The EDS file can be obtained from the manufacturer. Also, you can uninstall the EDS file.



9.4.4.1.1 Master Configuration

Double-click the CANopen master node to open the master configuration page.



Network management

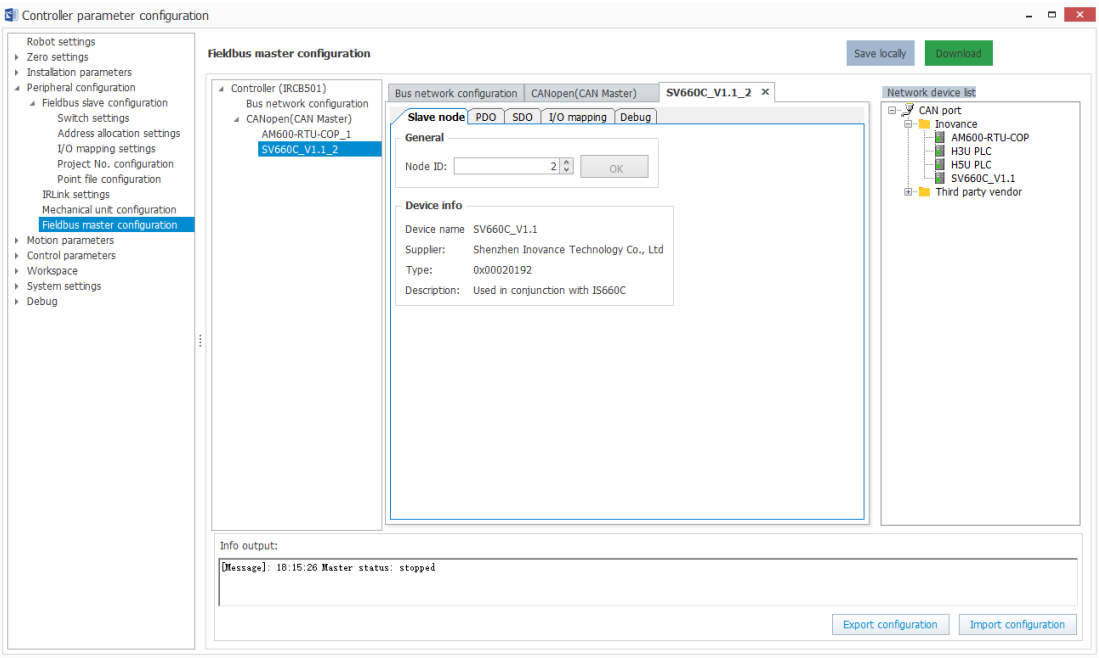
- Node ID: The unique identifier of the master in the CANopen network, ranging from 1 to 63, fixed to 63.
- Baud rate The baud rate used for transmission on the bus. In Kbit/s, the following baud rates can be used: 20, 50, 100 125, 250, 500, 800, and 1000, 500 by default.

Network status

- Start monitoring: After you click this button, the software starts monitoring the information of the nodes on the bus, including node status, emergency error, and SDO configuration error. When an error is acquired, it is displayed on the page in real time. After the error is cleared, the error displayed on the page is cleared. Node status and error messages are cleared when the controller is disconnected. Monitoring stops when the master configuration page is closed.
- Node status: Displays the current operational status of network nodes, which is meaningful only for the master. The status value comes from the node status monitoring register.
- Emergency error message: Displays emergency error messages in the current network, which is meaningful only for the master.
- SDO configuration error: When an SDO configuration error is detected, the station number, error step number (that is, the SDO error number) and error code (that is, SDO error code: standard CANopen error code) are displayed

9.4.4.1.2 Slave Configuration

The following takes SV660C slave as an example to describe how to configure the CANopen slave. Double-click the slave in the network to open the slave configuration page. The configuration page includes five tabs: Slave node, PDO, SDO, I/O mapping, and Debug.



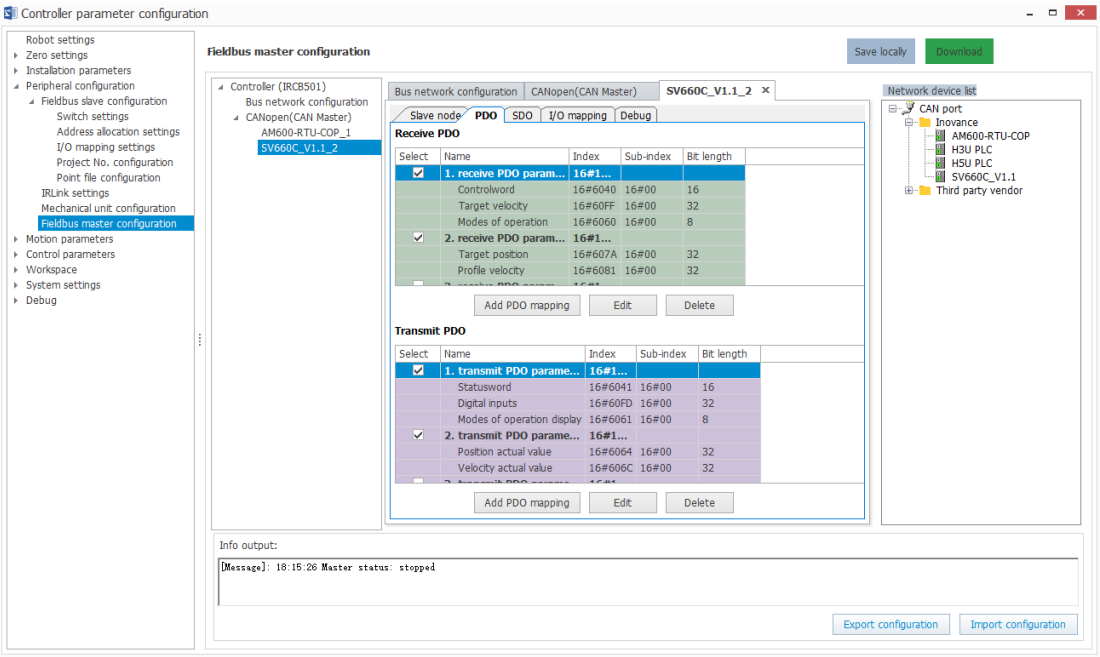
Slave node

- Node ID: Each slave is assigned with a unique node ID in the CANopen network. The range is 1 to 62. The node ID must be consistent with the identity of the slave itself (such as set by a DIP switch).
- Device info: The device information of the slave is obtained from the EDS file.

PDO

The process data object (PDO) is used for real-time data transfer between master and slave. The PDO includes communication parameters and mapping parameters. The communication parameters include COB-ID, transmission type, and transmission control, etc. The PDO mapping parameter indicates that the PDO transmission data comes from the index and sub-index of the object dictionary.

PDOs are divided into two types: receive PDO and transmit PDO.



- Receive PDO: Indicates the real time data sent by the master to the slave. The receive PDO comes from the objects in the object dictionary starting from index 1400 and ending at index 1600. The default communication parameters for each PDO come from its corresponding subindexes. The objects that can

be mapped by the receive PDO come from objects in the object dictionary that allow write access, such as RW, RWW, and WO.

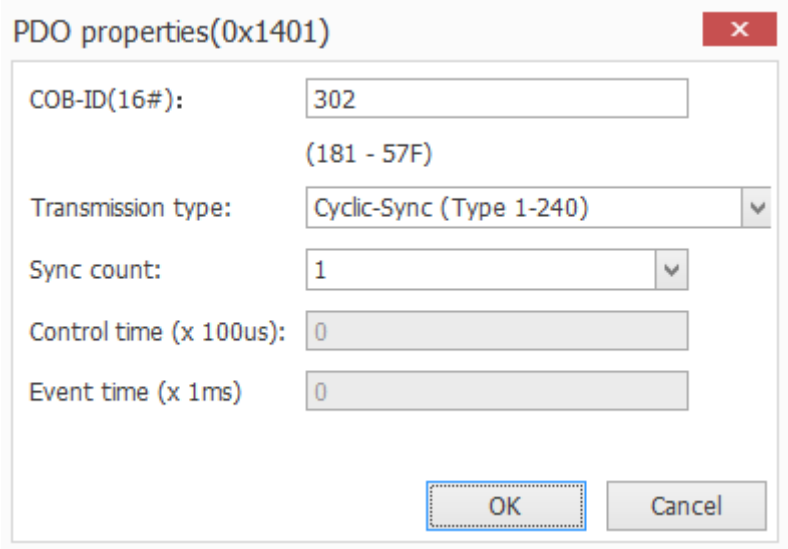
- Transmit PDO: Indicates the real time data sent by the slave to the master. The transmit PDO comes from the objects in the object dictionary starting from index 1800 and ending at index 1A00. The default communication parameters for each PDO come from its corresponding subindexes. The objects that can be mapped by the transmit PDO come from objects in the object dictionary that allow write access, such as RW, RWR, RO, and CONST.

Description

- You can check the checkbox in front of a PDO to activate it. The default active PDOs in the EDS file of the slave are checked by default.
- Add PDO mapping: Adds a PDO mapping to the currently selected PDO, which is added after the current PDO. The maximum length of PDO mapping is 64 bits. The PDO mapping comes from the data dictionary. The objects that can be mapped by the receive PDO come from objects in the object dictionary that allow write access, such as RW, RWW, and WO.
- Edit: Edits the currently selected PDO communication parameter or PDO mapping parameter. If the PDO is selected, edit the PDO communication parameters (that is, the PDO properties). If the PDO mapping is selected, edit the PDO mapping.
- Delete: Deletes the currently selected PDO or PDO mapping. If the PDO is selected, the PDO is deleted. If the PDO mapping is selected, the PDO mapping is deleted.

PDO properties

The PDO properties are used to configure the PDO communication parameters, including COB-ID (communication object identifier), transmission type, control time, and event time. The following dialog box pops up when you edit the PDO or double-click the PDO.



- COB-ID: Indicates the ID for sending a PDO parameter. Based on the CANopen DS301 protocol, default COB-IDs are available for the first four PDO parameters. For other PDO parameters, you need to configure their COB-IDs (if supported by the slave). The COB-IDs must be unique within the entire network, ranging from 0x180 to 0x57F.
- Transmission type: PDO communication transmission mode. PDO can be transmitted in various modes, as shown in the following table.

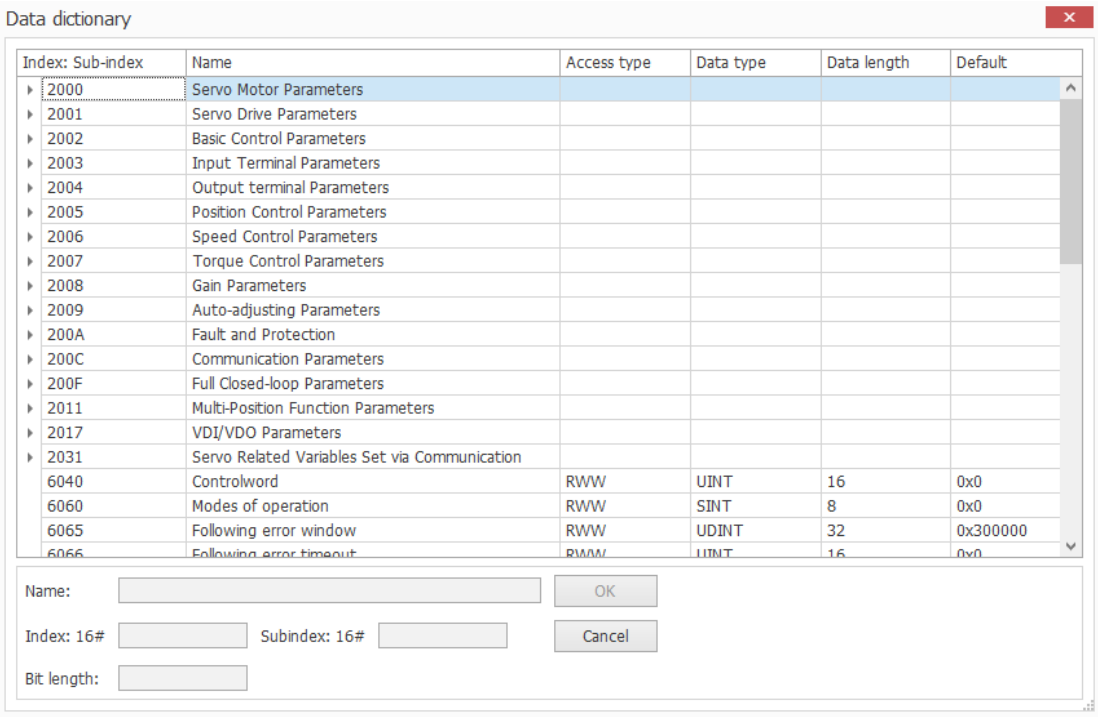
Type	Data transmission conditions	Data validity condition
Cyclic-Sync (Type 0)	Data changes and a synchronization frame is received.	Data does not take effect immediately upon reception; it takes effect only after receiving a synchronization frame.
Cyclic-Sync (Type 1-240)	Data is sent after receiving the corresponding "sync count" of frame synchronization.	Data does not take effect immediately upon reception; it takes effect only after receiving a synchronization frame.
Asynchronous - Only RTR (Type 253)	Not supported	Not supported
Asynchronous - Only RTR (Type 253)	Not supported	Not supported
Manufacturer specified (Type 254)	Defined by individual manufacturers.	Defined by individual manufacturers.
Device profile specified (Type 255)	Data changes or meets event time, and the change frequency is less than the depression time.	It takes effect immediately.

The default transmission type of each PDO comes from the object corresponding to PDO subindex 02 in the object dictionary. If the write permission of the corresponding object dictionary is not granted, the transmission type cannot be changed.

- Sync count: Indicates the number of sync frames the slave receives before it can start processing PDO data transmission. It depends on the transmission type and can be modified only when the transmission type is Cyclic-Sync (Type 1-240).
- Control time: Indicates the product of the minimum interval between the transmission of two messages by the same PDO and 100 μs. This parameter prevents PDOs from being sent too frequently when a value is changed. The range is 0 to 65535, with default 0. This parameter can be configured only when the transmission type is 252, 253, 254, and 255. The default control time of each PDO comes from the object corresponding to PDO subindex 03 in the object dictionary. If the write permission of the corresponding object dictionary is not granted, the control time cannot be changed.
- Event time Indicates the product of the minimum interval between the transmission of two messages by the same PDO, in ms. The range is 0 to 65535, with default 0. This parameter can be configured only when the transmission type is 252, 253, 254, and 255. The default event time of each PDO comes from the object corresponding to PDO subindex 05 in the object dictionary. If the write permission of the corresponding object dictionary is not granted, the event time cannot be changed.

Data dictionary

The data dictionary dialog box is used to add or edit PDO mappings or SDOs.



This list comes from the objects in the EDS file.

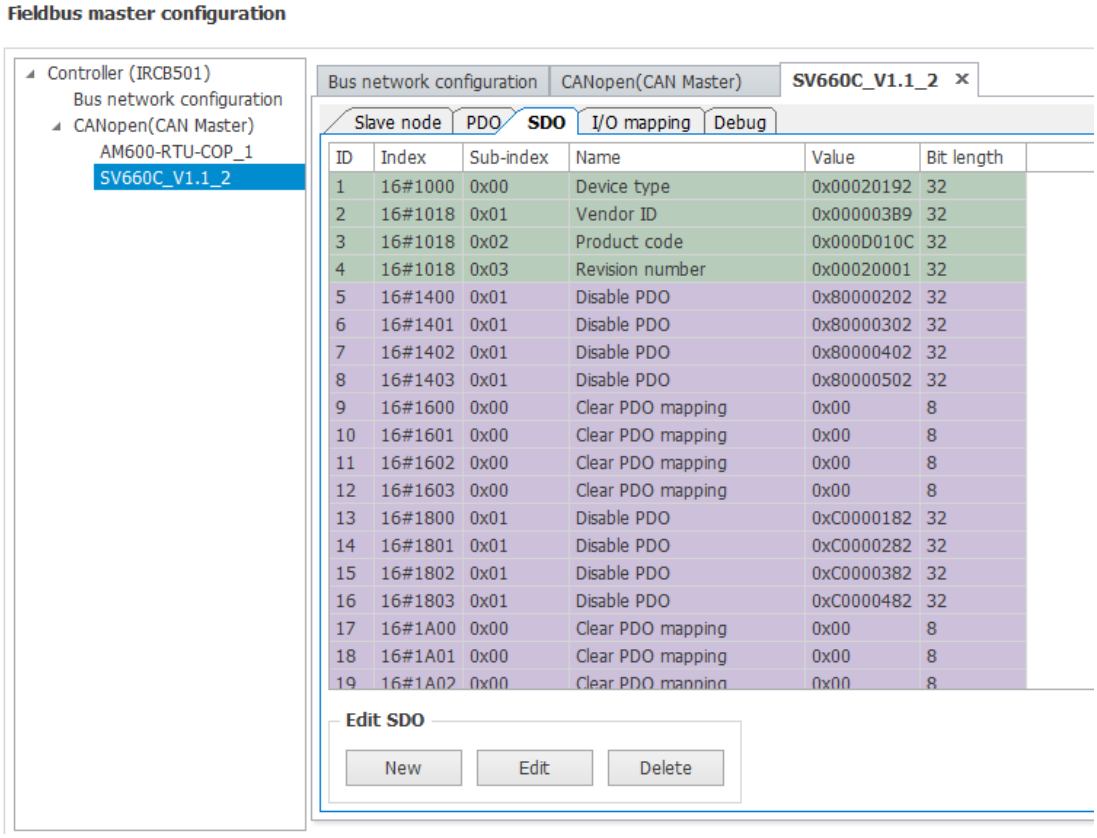
During editing of the receive PDO mapping, only objects with access right of RW, RWW, or WO and an index larger than 16#2000 can be displayed.

During editing of the transmit PDO mapping, only objects with access right of RW, RWR, RO or CONST and an index larger than 16#2000 can be displayed.

During editing of the SDO, only objects with access right of RW, RWW, RWR, or WO and an index larger than 16#2000 can be displayed.

SDO

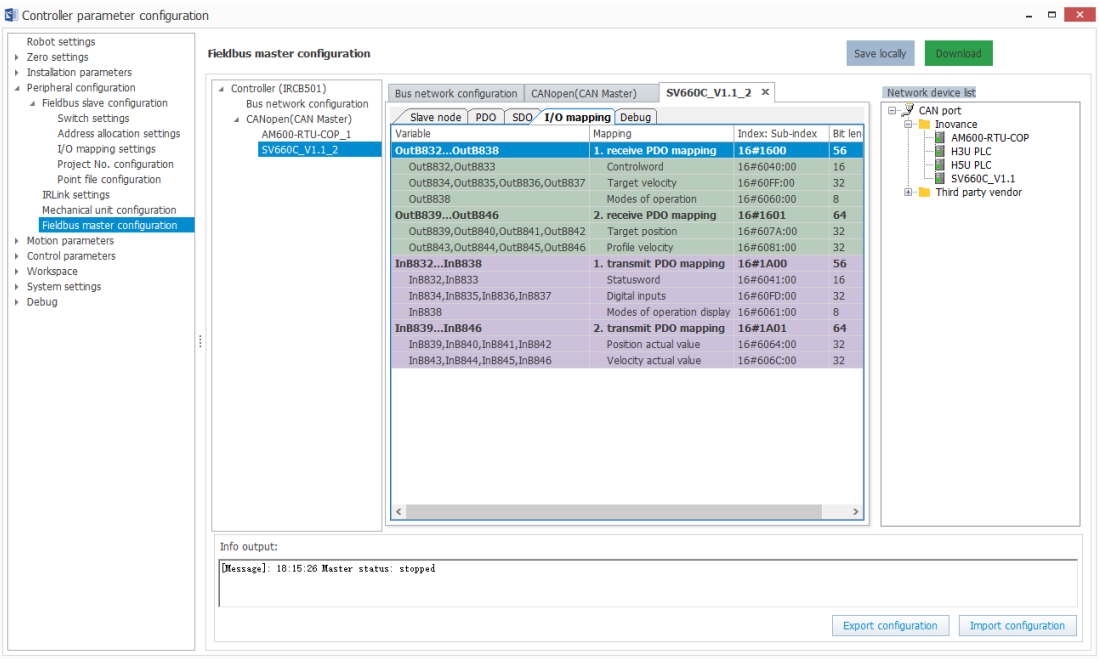
The service data object (SDO) can be used for data transfer during slave initialization or during slave operation. All configurations on this tab are written to the slave during initialization of the slave.



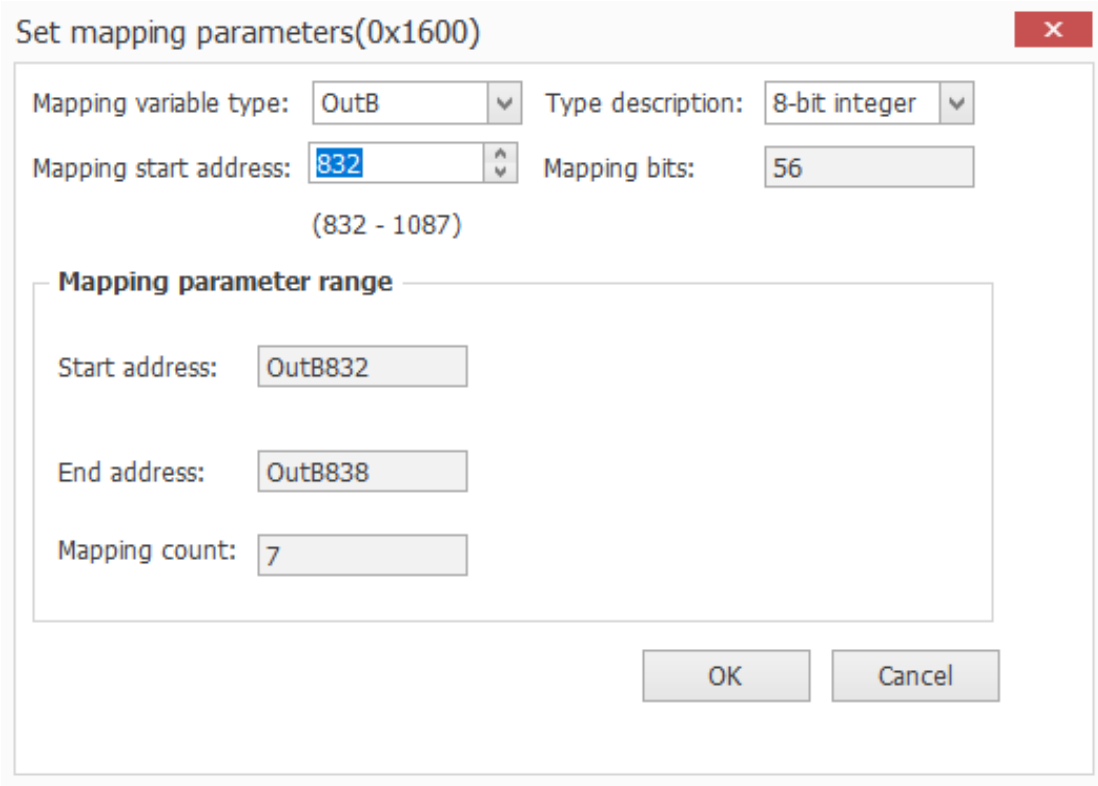
The SDO list displays the SDOs written during initialization of the slave. The top green SDOs are related to the device and version information, and do not need to be written. The rest of the SDOs are generated automatically based on the EDS file and PDOs and PDO mapping. You can also add, modify, and delete SDOs by using the buttons below.

I/O mapping settings

This tab is used to set the communication relationship between master and slave PDOs, as shown in the figure below.

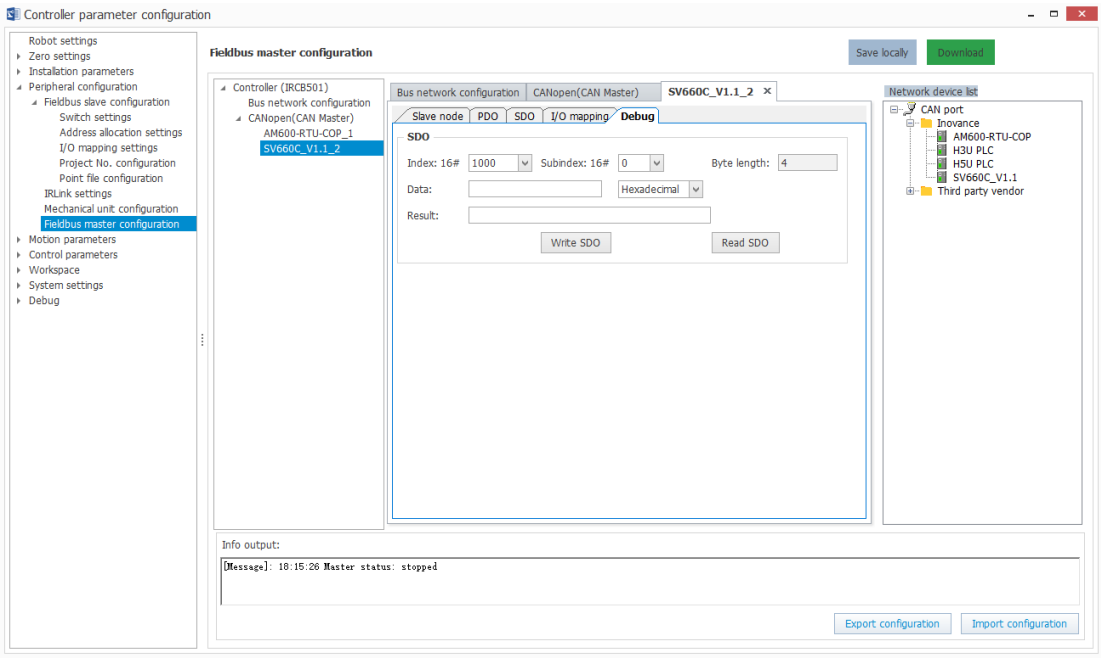


You can set the register start address of each PDO by double-clicking the PDO mapping parameter, as shown in the figure below.



Debug

It allows you to read and write SDOs, as shown in the figure below.



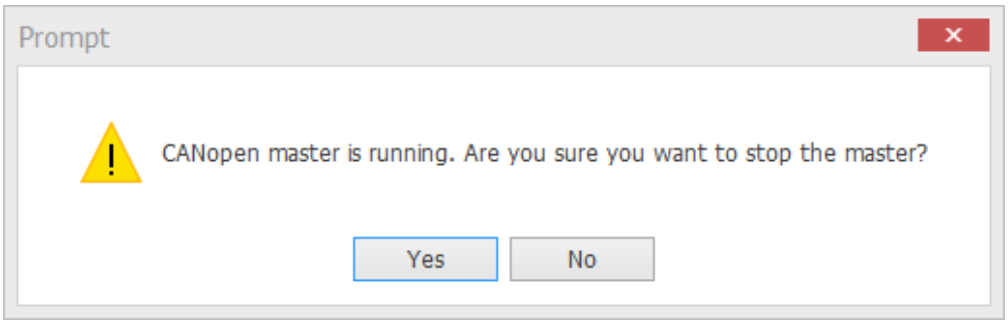
You can read or write SDOs during running of the slave. To read or write an SDO, you need to specify the index, subindex, and byte length of the SDO. For the write operation, you also need to specify the data to write.

- You can select "Index" and "Subindex" from the drop-down list.
- The "byte length" is automatically generated based on the object dictionary in the EDS. It cannot be modified.
- The "data" refers to data to send or return.
- The "result" refers to the result of SDO read/write operation. If the read/write is abnormal, an error message is displayed.

- The "Read SDO" and "Write SDO" buttons allow you to read or write the object dictionary.

Configuration file

- Download: Click the "Download" button to save the configurations as a configuration file (canopen.json) and send it to the controller. Make sure the controller is connected. If the master is in the stop state, the configuration file is sent directly to the controller. If the master station is in the running state, a prompt will pop up. The configuration file can be sent to the controller only when the master is stopped.



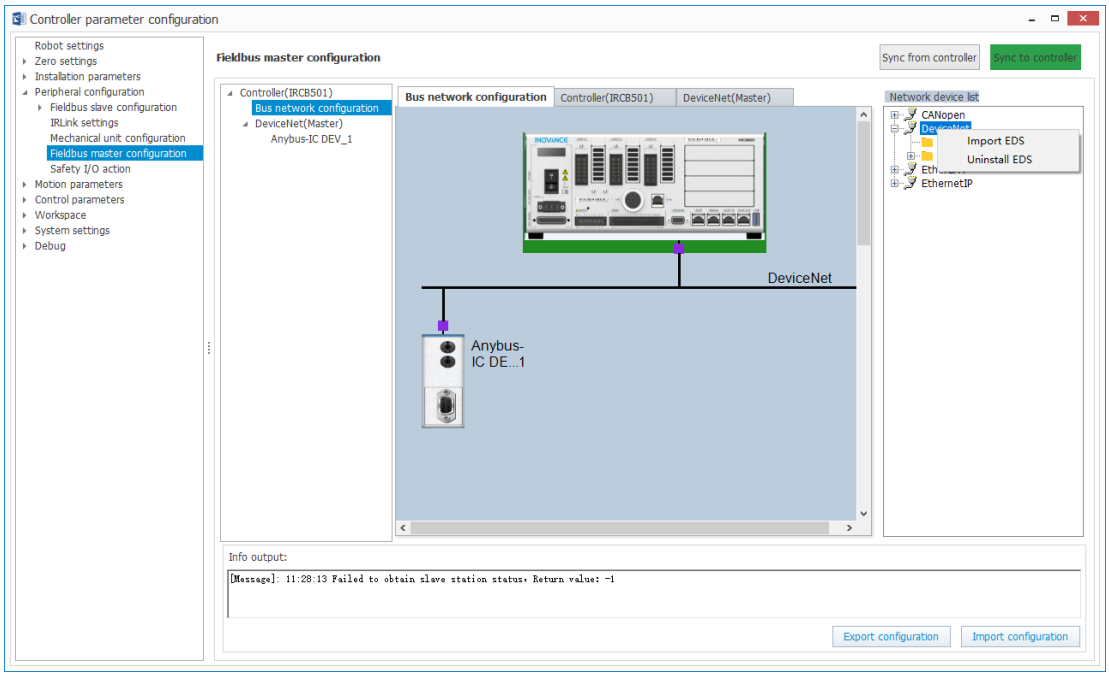
When "OK" is selected, the configuration file is sent to the controller.

- Sync from controller: When the controller is connected, the configuration file is automatically uploaded and parsed when the master bus configuration is opened, and the configuration is displayed in the interface.

9.4.4.2 DeviceNet Master Configuration

9.4.4.2.1 Device Import

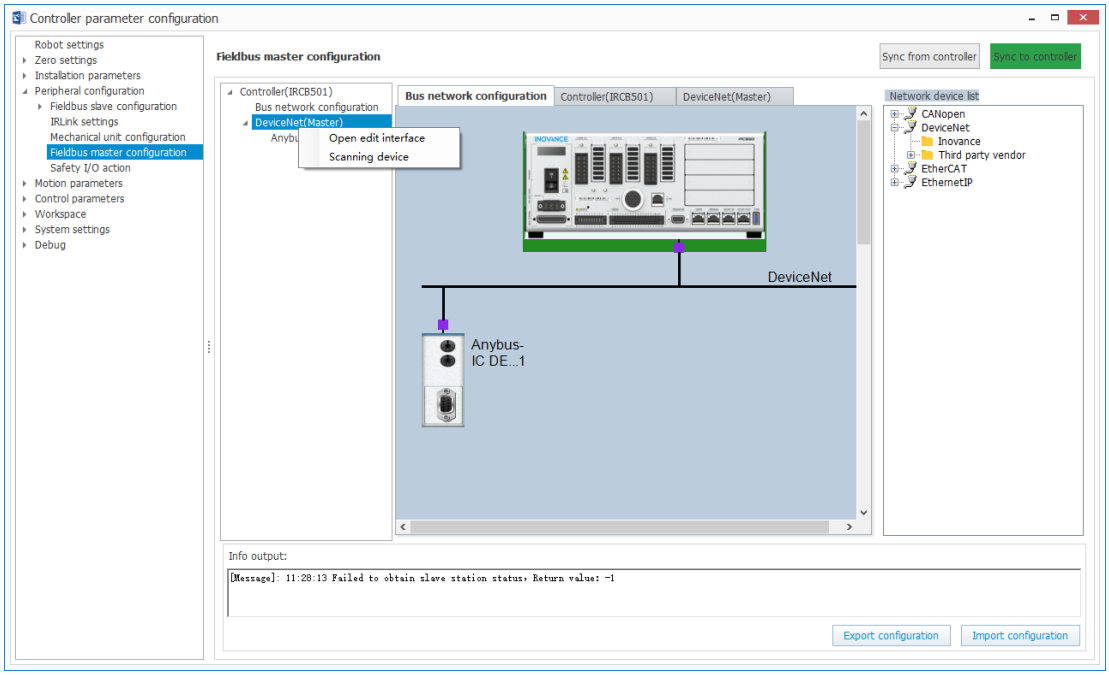
The robot DeviceNet bus identifies slave device information based on the EDS file, as shown in the figure below:



- Import EDS: Select "Network Device List" > "DeviceNet", right-click and select "Import EDS", and then select the corresponding EDS file for the slave device to complete the import.
- Uninstall EDS: Select the device to be deleted from the network device list, right-click and select "Uninstall EDS" to remove the device from the list.

9.4.4.2.2 Configuration Settings

The robot DeviceNet bus supports two configuration methods: manual configuration and automatic scanning, as shown in the figure below.



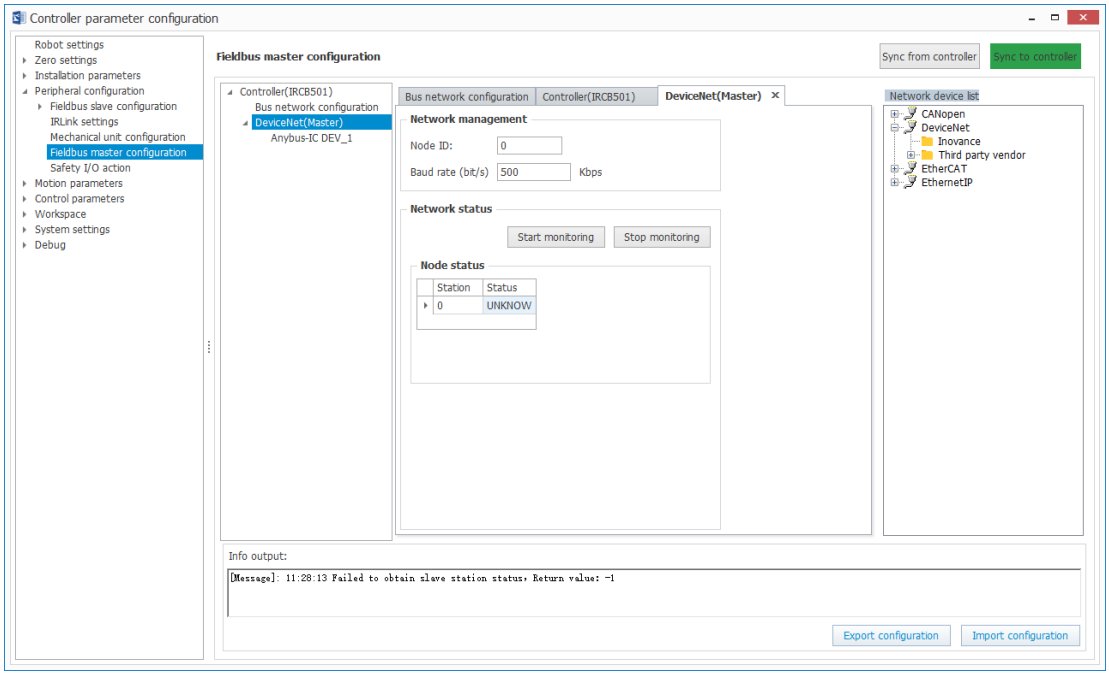
- Manual configuration: Select the target device in the network device list, then double-click or drag it to the central area of the interface to complete the bus device addition.
- Automatic scanning: Select Controller > DeviceNet(Master), right-click and select "Scanning device". The robot can automatically scan bus devices, and the results are displayed in the central area.



If the scanned device does not exist in the installed device list, the device will not be displayed.

9.4.4.2.3 Master Configuration

Double-click the DeviceNet master node to open the master configuration page.



Network management

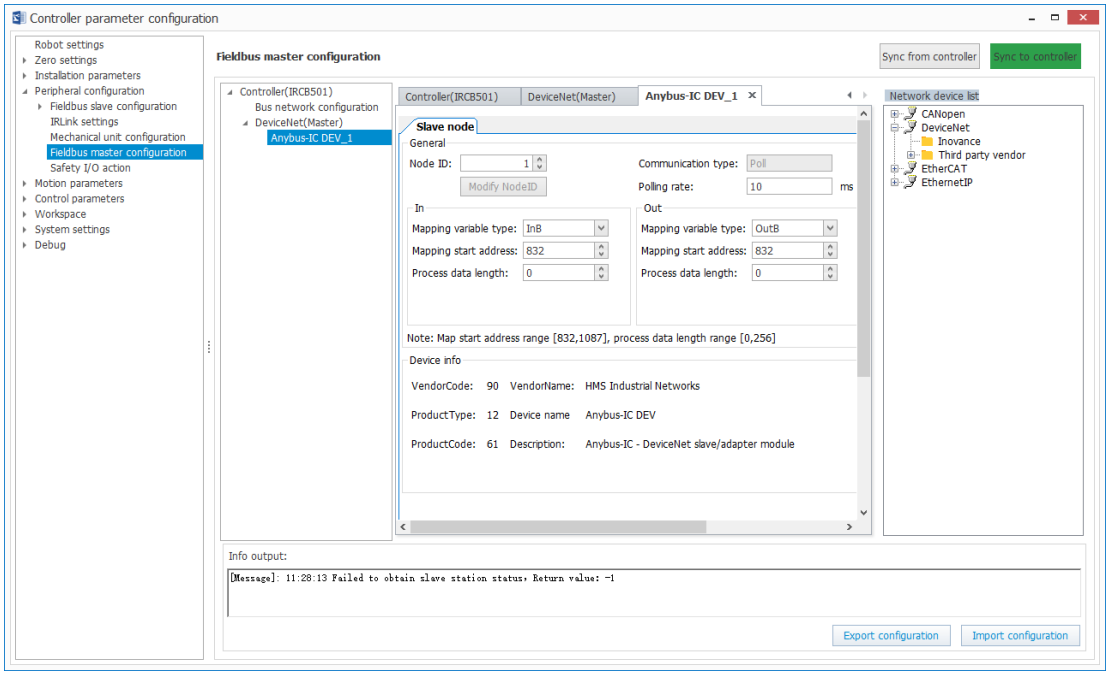
- Node ID: The unique identifier of the master in the DeviceNet network, ranging from 0 to 63, with default 0.
- Baud rate The baud rate used for transmission on the bus. In Kbit/s, the following baud rates can be used: 12, 250, and 500, 500 by default.

Network status

- Start monitoring: Click this button to start monitoring the status of bus nodes. Monitoring automatically stops after the master configuration page is closed.
- Stop monitoring: Click this button to stop monitoring the status of bus nodes.
- Node status: Displays current status information of bus nodes, including: connection status and operation status of master and slave nodes.

9.4.4.2.4 Slave Configuration

Using the DECOWELL slave as an example, this section introduces the configuration process and relevant parameters for DeviceNet slaves. Double-click the slave in the network to open the slave configuration page, which includes general parameter configuration and device information display.



- Node ID: Each slave is assigned with a unique node ID in the DeviceNet network. The range is 0 to 63. The node ID must be consistent with the identity of the slave itself (such as set by a DIP switch).
- Communication type: The communication type between the master and slave devices in the DeviceNet network. Only Poll communication mode is supported.
- Poll rate: The communication cycle between the master and slave devices in the DeviceNet network. It supports only even-numbered cycles, with a minimum poll rate of 2 ms. The poll rate for each slave device is individually configurable.
- Mapping variable type: Only byte-type variables are supported.
- Mapping start address: Maps the process data of the current DeviceNet slave to the I/O data area of the robot master bus. You can access the bus process data directly using I/O monitoring or I/O instructions. The configurable mapping range is 832 to 1343.
- Process data length: The process data length of the current DeviceNet slave. The configurable range is 0 to 256. Note: Ensure that the mapping start address + process data length < 1343 to guarantee complete data transmission and reception.

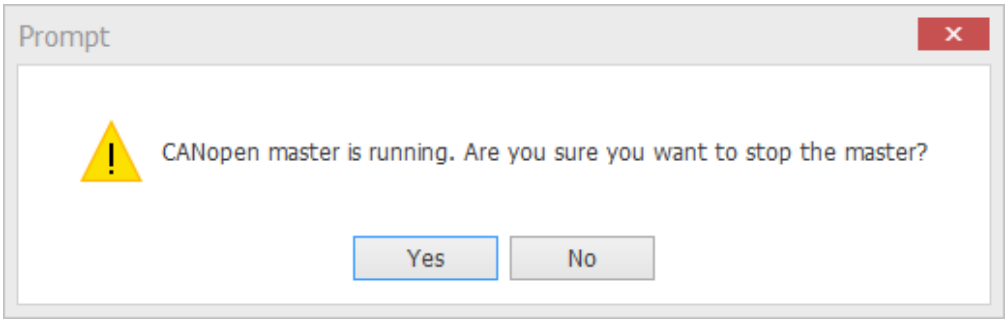


- The In/Out mapping address ranges of all DeviceNet slaves must not overlap.
- The total In/Out process data length of all DeviceNet slaves must not exceed 256 bytes.

Device info: The device information of the slave is obtained from the EDS file.

9.4.4.2.5 Sync to/from Controller

- Sync to controller: Click the "Sync to controller" button to save the configurations as a configuration file and send it to the controller. Make sure the controller is connected and in the emergency stop state. If the master is in the stop state, the configuration file is sent directly to the controller. If the master station is in the running state, a prompt will pop up. The configuration file can be sent to the controller only when the master is stopped.



When "OK" is selected, the configuration file is sent to the controller.

- Sync from controller: When the controller is connected, the configuration file is automatically uploaded and parsed when the master bus configuration is opened, and the configuration is displayed in the interface.

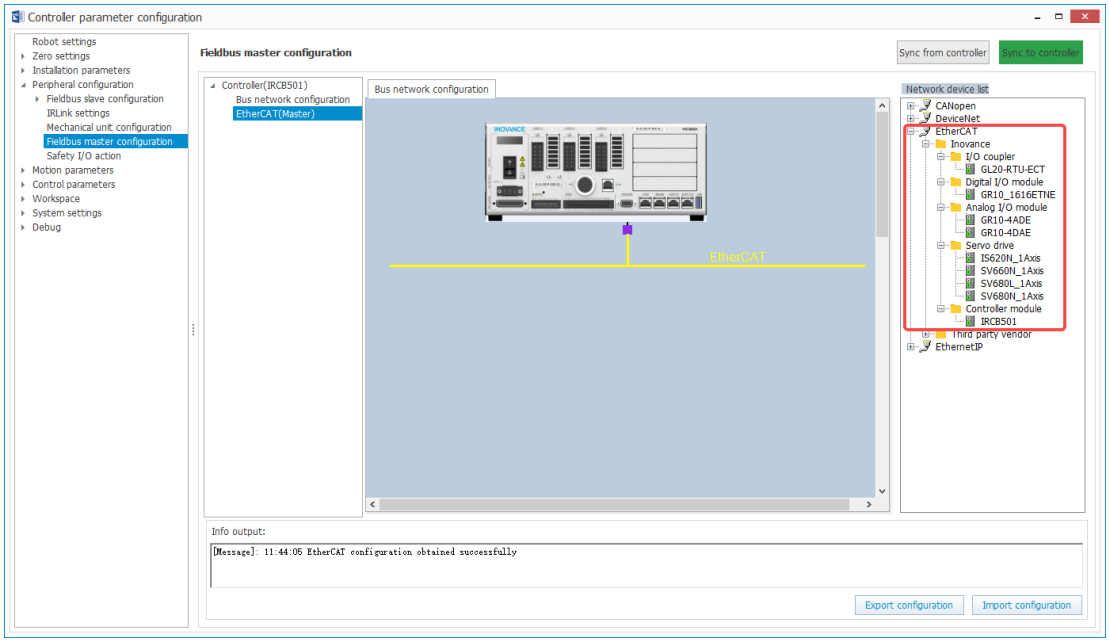
9.4.4.2.6 Mutual Exclusion Between Bus Function and Arc Welding Plug-in

- CANOpen and DeviceNet functions of the master bus are mutually exclusive with the arc welding plug-in. After installing the arc welding plug-in, the use of the aforementioned bus functions is not allowed.
- Installing the arc welding plug-in will clear the bus configuration.

9.4.4.3 EtherCAT Master Configuration

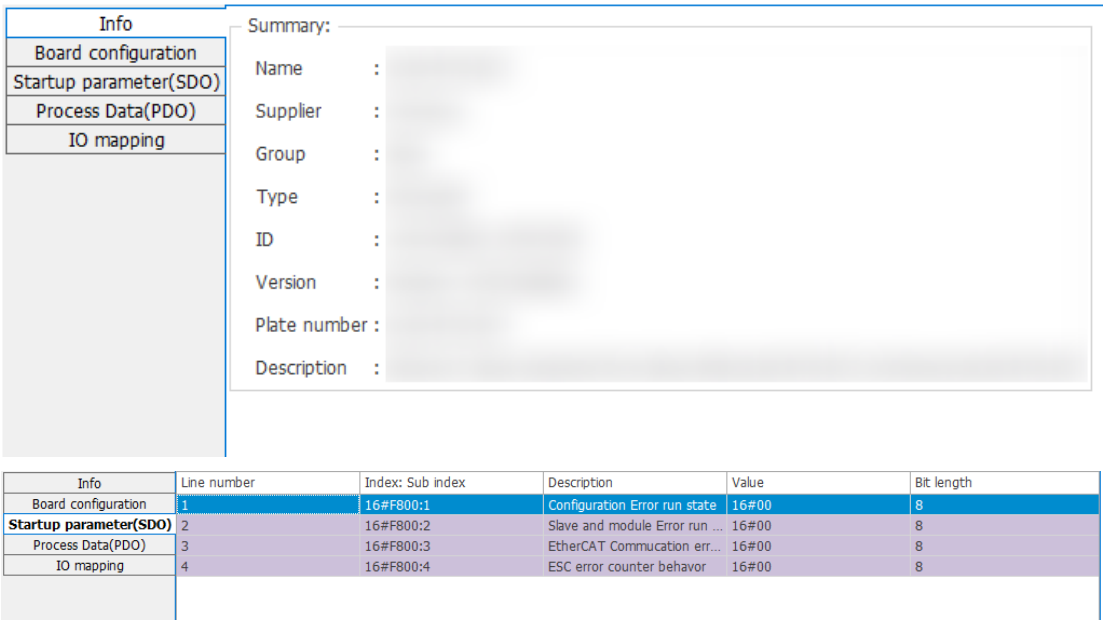
9.4.4.3.1 Device Import

The robot EtherCAT bus identifies slave device information based on the ENI file, as shown in the figure below:



- Import ENI: Select "Network Device List" > "EtherCAT", right-click and select "Import ENI", and then select the corresponding ENI file for the slave device to complete the import. Up to 50 EtherCAT slave devices are supported. ESI files are also supported, which will be automatically converted to the required ENI files.

- Uninstall ENI: Select the device to be deleted from the network device list, right-click and select "Uninstall ENI" to remove the device from the list. For a newly imported device, information, SDO, PDO, and I/O mapping configuration interfaces are available.



Info	Line number	Index: Sub index	Description	Value	Bit length
Board configuration	1	16#F800:1	Configuration Error run state	16#00	8
Startup parameter(SDO)	2	16#F800:2	Slave and module Error run ...	16#00	8
Process Data(PDO)	3	16#F800:3	EtherCAT Commucation err...	16#00	8
IO mapping	4	16#F800:4	ESC error counter behavor	16#00	8

If the number of SDOs in the configuration file is 0, this page will be hidden.

Info	IO	Description	Index: Sub index	Bit length	Mark	SM
Board configuration	**IN**	**Device TPDO Mappin...**	**16#1BA0**	**32**	**F**	**3**
Startup parameter(SDO)	input	LBus status	16#F100:16#1	16		
Process Data(PDO)	input	Fault ID	16#F100:16#2	16		
IO mapping	**OUT**	**Device RPDO Mappi...**	**16#17A0**	**16**	**F**	**2**
	output	Device control	16#FB00:16#0	16		



- Newly imported devices support double-click modification of SDO values. Invalid parameters may occur after modification, causing issues with third-party slaves.
- Modify the SDO values according to the guide of the slave device. Only hexadecimal values are supported for display and input.
- Do not import coupler devices, as this may cause error exceptions in the master.
- Exercise caution when deleting any EtherCAT slave devices that are pre-configured with the software, as this may cause unexpected anomalies and issues with re-importing these devices, which could render them unusable.
- ENI device imports are not supported for devices with PDO data of 1 bit in version V4R24C5.
- For the imported slave configuration, ensure that the XML file burned into the slave EEPROM is the same as the XML file imported into the software.

9.4.4.3.2 Master Configuration

The robot EtherCAT master is always in an enabled state. The initial several slaves of the robot EtherCAT master are fixed as internal slaves, while the remaining external slaves can be configured and modified by the user. The currently supported slave types include I/O modules, analog modules, servo drives, controller modules, third-party imported slaves, etc.

The currently supported slave modules include GL20-RTU-ECT, GL20-4AD, GL20-4DA, GL20-0016ETN, GL20-1600END, GR10-4ADE, GR10-4DAE, IS620N_1Axis, SV660N_1Axis, SV680L_1Axis, SV680N_1Axis, and IRC501.

The current EtherCAT slave usage has the following limitations:

- The maximum configurable number of slaves is 16.
- The maximum number of AD/DA channels is 16, including series 10 and 20.
- The maximum number of digital I/O channels is 128, including series 10 and 20.
- The maximum number of servo slaves is 6.
- The maximum number of controller slaves is 4.
- The maximum number of GL20 couplers is 2.
- A single GL20 coupler supports up to 16 expansion slot modules.



9.4.4.3.3 Slave Status Management Function

This function enables different management methods for modules after configuring the slaves on the EtherCAT master bus, facilitating on-site device debugging. The management methods include two types: enable and disable.

- Enable: The EtherCAT master normally monitors and exchanges data; if a slave status is incorrect or offline, the alarm will be triggered.
- Disable: Shield slave status error/offline alarm, stop data exchange

Right-click "EtherCAT (Master)" to enable the slave status management function. When enabling this function, all slave modules must be online. After successful enabling, both the controller and all slave modules need to be restarted.

Each slave can display the following information:

- Slave module name: Indicates the name of the slave module.
- Slave node: Slave ID.
- Enable switchover function: Indicates whether the current slave supports status management (Open/Close).

- Slave enable status: Indicates the current management status of the module (Activate/Disable).
- Slave communication status: Indicates the communication status of the slave, (Disconnected/Init/Pre-OP/Safe-OP/OP/Communication separated).

Steps

1. Right-click "EtherCAT (Master)" to enable the slave enable management function. When enabling this function, all slave modules must be online.
2. After enabling the enable management function, you need to restart the controller and all slave modules.
3. Right-click the module to enable its enablement switchover function.
4. Right-click the module or instruction SetIOBusState to set the management status for enabling/disabling.



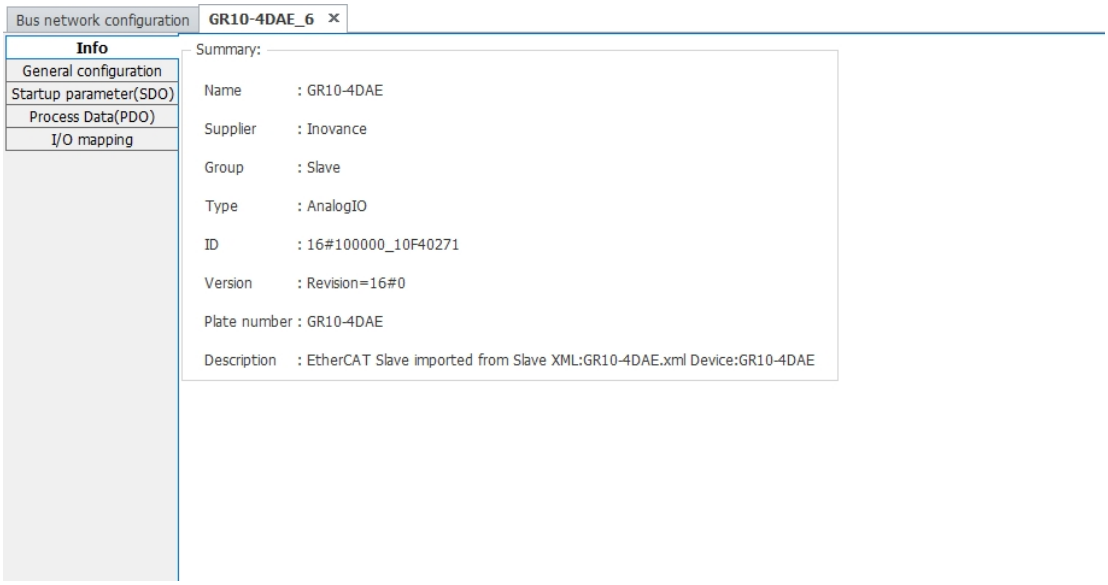
- To configure status management settings for a module, you need to enable the enablement switchover function.
- The module is set by default to disable the enablement switchover function, and the management method is enable.
- If the module is set to disable, after the enablement switchover function is disabled, the management method will be forcibly switched to enable.
- When the management method is disable, if you want to switch to enable, the module must be online and the communication status must be switchable to OP.
- If the slave configuration or I/O mapping is modified through the interface, the status management of all slaves will be reset to default.
- If the slave configuration is modified by loading a configuration file while the enablement switchover function is enabled, after the controller is restarted, disable the enablement switchover function and restart the controller and peripherals again; otherwise, abnormalities may occur.

9.4.4.3.4 Slave Configuration

The following takes GR10_4DAE slave as an example to describe how to configure the EtherCAT slave. Double-click the slave in the network to open the slave configuration page. The configuration page includes five tabs: Info, General configuration, Startup parameter (SDO), Process data (PDO), I/O mapping.

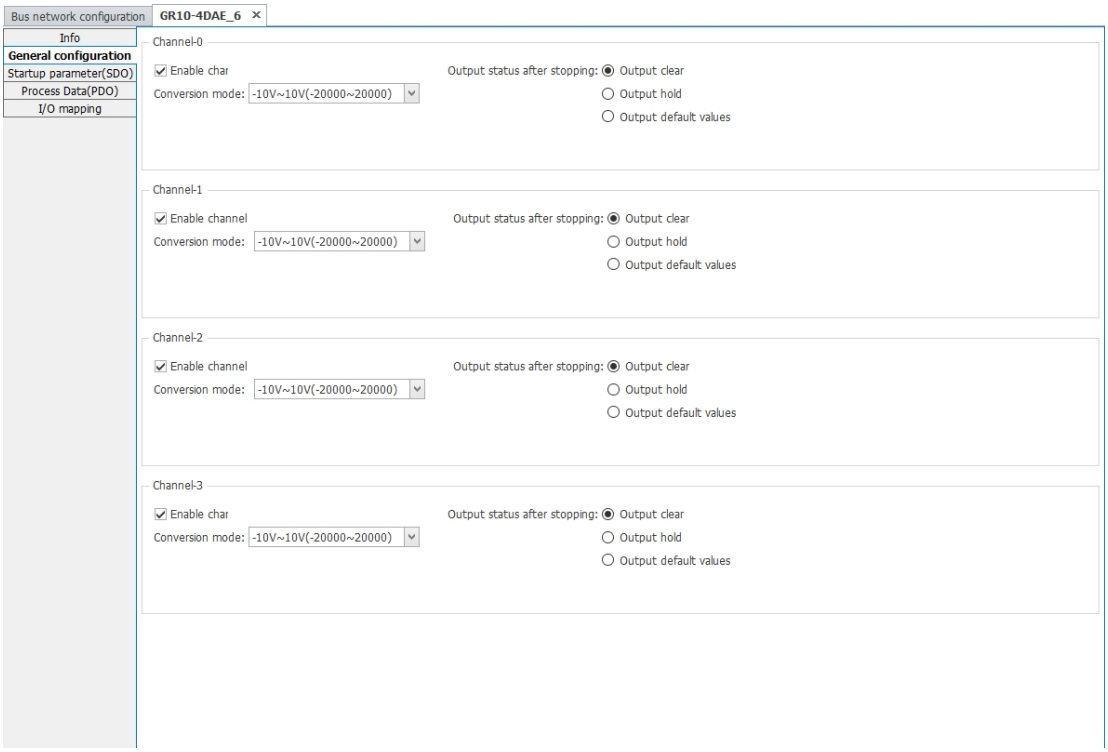
Info

Device info: The device information of the slave is obtained from the ENI file.



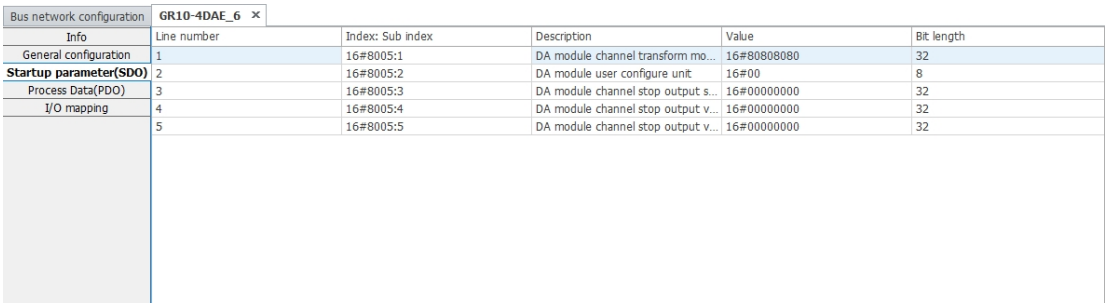
General configuration

The "General configuration" interface primarily provides the startup parameter (SDO) settings, mainly including some configuration parameters for the slave during startup. For example, channel enable, digital-to-analog conversion mode, output status after stop, etc. for GR10_4DAE slave. When the corresponding configuration is checked on this page, the corresponding parameters in the startup parameter (SDO) will change accordingly.



Startup parameter (SDO)

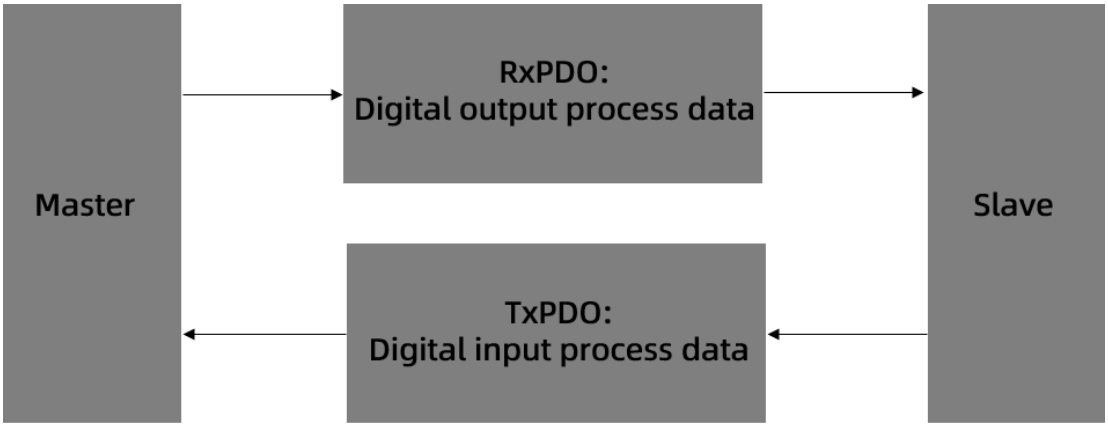
The SDO is used to transfer non-cyclic data, such as communication parameter configuration, and drive running parameter configuration. The SDO data is used for parameter configuration of slave devices. Note that SDO parameters cannot be modified arbitrarily by users; they can only be modified by adjusting general configuration settings.



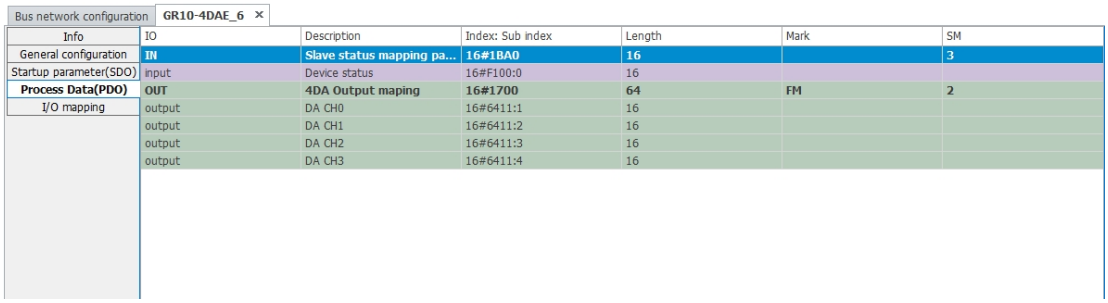
SDO list: Displays SDOs written during slave initialization. This SDO data is retrieved from the corresponding slave ENI file, which includes index, sub-index, description, value, and data size.

Process data (PDO)

PDO data is transmitted in the producer-consumer model. It is distinguished in transmit PDO (TPDO) and receive PDO (RPDO). The slave receives instructions from the master through the RPDO and sends its status to the master through the TPDO.



PDO mapping is used to establish the mapping relation between the object dictionary and the PDO. 1600h to 17FFh are RPDOs, and 1A00h to 1BFFh are TPDOs.



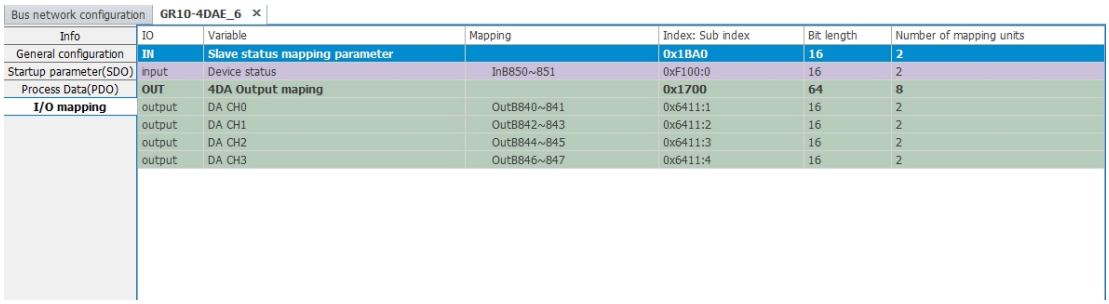
PDO data includes: input/output type, description, index, sub-index, data length, and data type, etc.

I/O mapping settings

Each PDO can be configured for I/O mapping. The mapped I/O range corresponds to the master I/O address range, which is IN/OUT (Byte): 832-1343.

The length of the I/O mapping must match the data length of the PDO. After mapping is completed, for output-type PDOs, you can control the PDO output by controlling the corresponding output I/O; for input-type PDOs, you can obtain the slave status by reading the corresponding input I/O.

For each PDO, double-click to open the I/O mapping window and enter the mapping address. The address cannot be set to one that is already occupied (the address may be occupied by other EtherCAT PDOs or by the DeviceNet master).



CAUTION

- For AD/DA slaves, I/O mapping can also be performed. However, AD/DA has a conversion range. If the DA value is negative, the I/O mapping data is an unsigned number and cannot be used directly. It must be converted to the corresponding binary input or output.
- If conflicts occur in I/O mapping, all I/O mappings will fail.
- Since DA control can be implemented through various methods (Set DA/OutW/API/Modbus), ensure correct timing during control operations.
- In general configurations for GL20-type boards, when stop output is configured for GL20, if module issues occur, the subsequent operational status of the module is considered abnormal and requires module restart to reset the status.
- For the slave configuration, ensure that the XML file burned into the slave EEPROM is the same one.

9.4.4.3.5 Servo/Controller Compatibility Configuration

- When configuring the EtherCAT master, you can select servo slaves and controller slaves for configuration. However, note that the servo slaves in the current robot's EtherCAT master are only used for external axis functions. Therefore, when external axis functions are required, you need to configure both the servo slaves and the mechanical unit.
- The current controller slaves are only used for the one-drive-many vision function of conveyor tracking. Therefore, to use the function, you not only need to configure the slaves but also perform conveyor tracking-related configurations.
- The SDO and PDO of the servo slaves and controller slaves only support viewing and do not allow modification. They also do not have I/O mapping function.

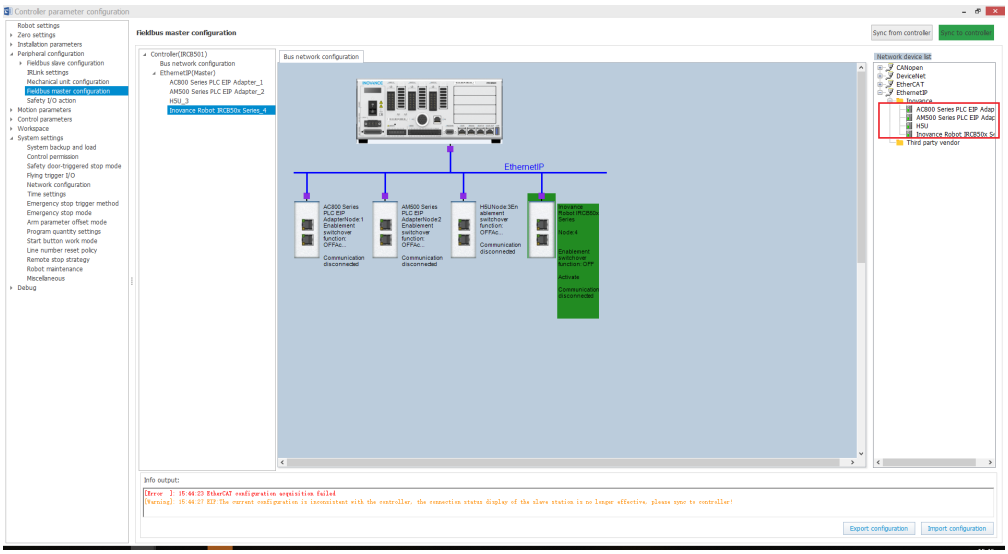
CAUTION

The servo type of each mechanical unit must correspond to the sequence of servo types defined in the configuration.

9.4.4.4 EtherNet/IP Master Configuration

9.4.4.4.1 Device Import

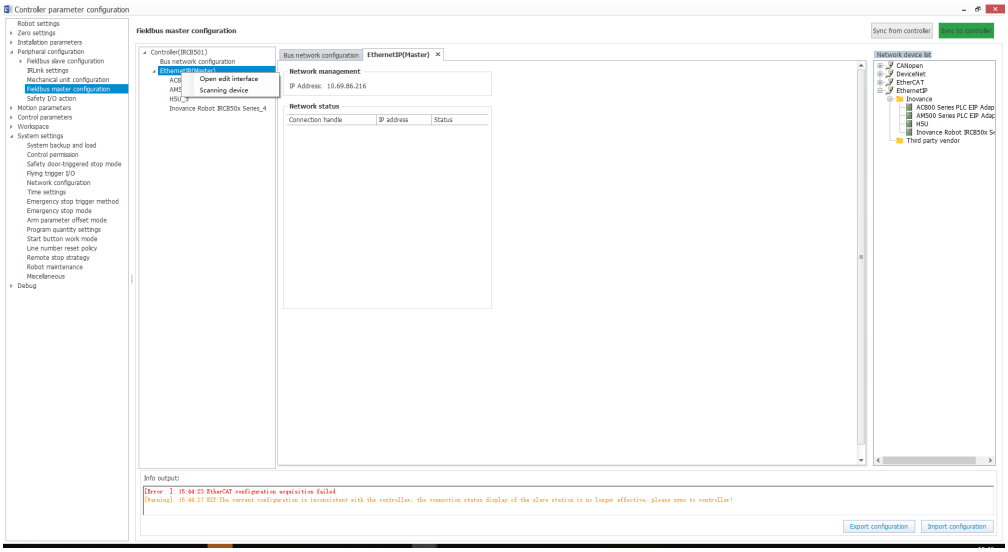
The robot EtherNet/IP bus identifies slave device information based on the EDS file, as shown in the figure below:



- Import EDS: Select "Network Device List" > "EtherNet/IP", right-click and select "Import EDS", and then select the corresponding EDS file for the slave device to complete the import.
- Uninstall EDS: Select the device to be deleted from the network device list, right-click and select "Uninstall EDS" to remove the device from the list.

9.4.4.4.2 Configuration Settings

The robot EtherNet/IP bus supports two configuration methods: manual configuration and automatic scanning, as shown in the figure below.



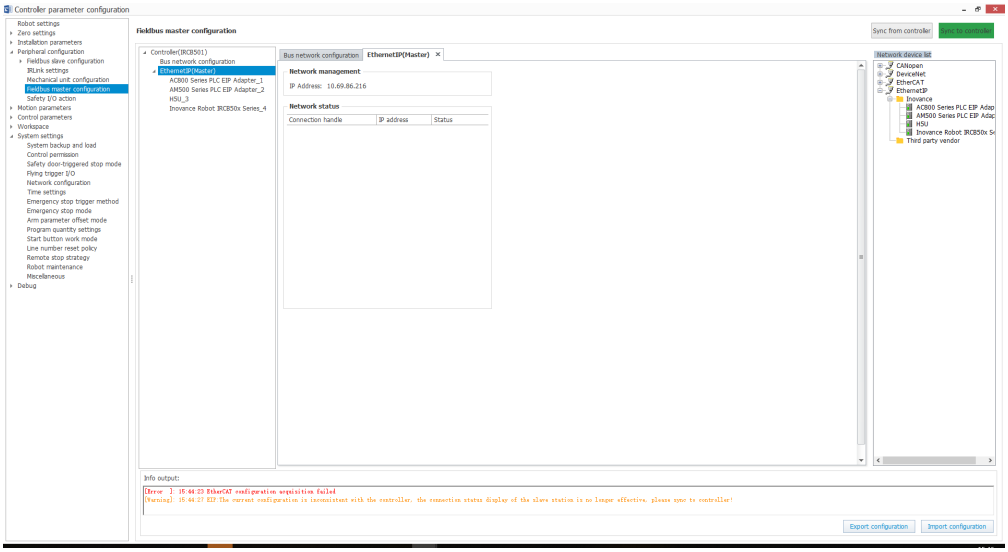
- Manual configuration: Select the target device in the network device list, then double-click or drag it to the central area of the interface to complete the bus device addition.
- Automatic scanning: Select Controller > DeviceNet(Master), right-click and select "Scanning device". The robot can automatically scan bus devices, and the results are displayed in the pop-up window. Select the

slave device in the pop-up window and click "OK" to add it to the configuration. If you click "Cancel", the scan results will be discarded and the original configuration will be retained.

> CAUTION
>
> If the scanned device does not exist in the installed device list, the device will not be displayed.

9.4.4.4.3 Master Configuration

Double-click the EtherNet/IP master node to open the master configuration page.



Network management

IP Address: The unique identifier for the master in the configured network.

Network status

Node status: Displays current status information of bus nodes, including: connection handle, IP address and connection status of master and slave nodes.

> CAUTION
>
> EtherNet/IP master supports up to 8 slaves, with each slave supporting up to 4 connections.

9.4.4.4.4 Slave Status Management Function

This function enables different management methods for modules after configuring the slaves on the EtherNet/IP master bus, facilitating on-site device debugging. The management methods include two types: enable and disable.

- Enable: The EtherNet/IP master normally monitors and exchanges data; if a slave status is incorrect or offline, the alarm will be triggered.

- Disable: Shield slave status error/offline alarm, stop data exchange

Each slave can display the following information:

- Slave module name: Indicates the name of the slave module.
- Slave node: Slave ID.
- Enable switchover function: Indicates whether the current slave supports status management (Open/Close).
- Slave enable status: Indicates the current management status of the module (Activate/Disable).
- Slave communication status: Indicates the communication status of the slave, (Disconnected/OP/Communication separated).

Steps

1. Right-click the module to enable its enablement switchover function.
2. Right-click the module or instruction SetIOBusState to set the management status for enabling/disabling.



- The module is set by default to disable the enablement switchover function, and the management method is enable.
- When the management method is disable, if you want to switch to enable, the module must be online and the communication status must be switchable to OP.

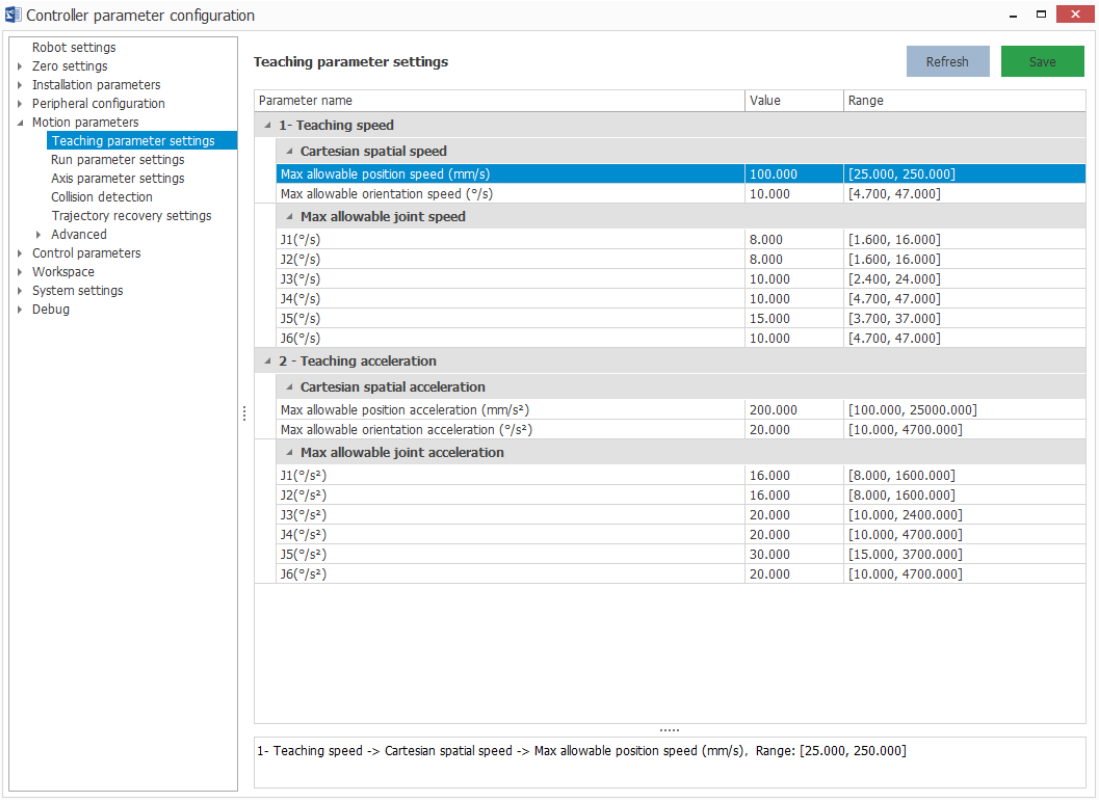
9.5 Motion Parameters

- The motion parameters include teaching parameters, run parameters, axis parameters, collision detection, trajectory recovery settings, and advanced functions.
- The teaching parameters will affect the speed and acceleration of jog motion in manual mode.
- The run parameters will affect the maximum speed, acceleration/deceleration, and transition characteristics of the respective motion instructions in the automatic mode.
- The axis parameters will affect the tracking error monitoring, arrival determination, and current limit, etc.

9.5.1 Teaching Parameter Settings

Steps

1. Double-click "Controller parameter configuration" in the device tree, then select "Motion parameters" > "Teaching parameter settings", as shown in the figure below.



2. You can set the teaching speed and teaching acceleration as needed.
   - The teaching speed refers to the maximum speed allowed during teaching. It includes the maximum speed in the joint frame and the maximum speed in the Cartesian frame. The speed during teaching is the product of the maximum speed set here and the selected speed percentage.
   - The teaching acceleration refers to the maximum acceleration allowed during teaching. It includes the maximum acceleration in the joint frame and the maximum acceleration in the Cartesian frame.



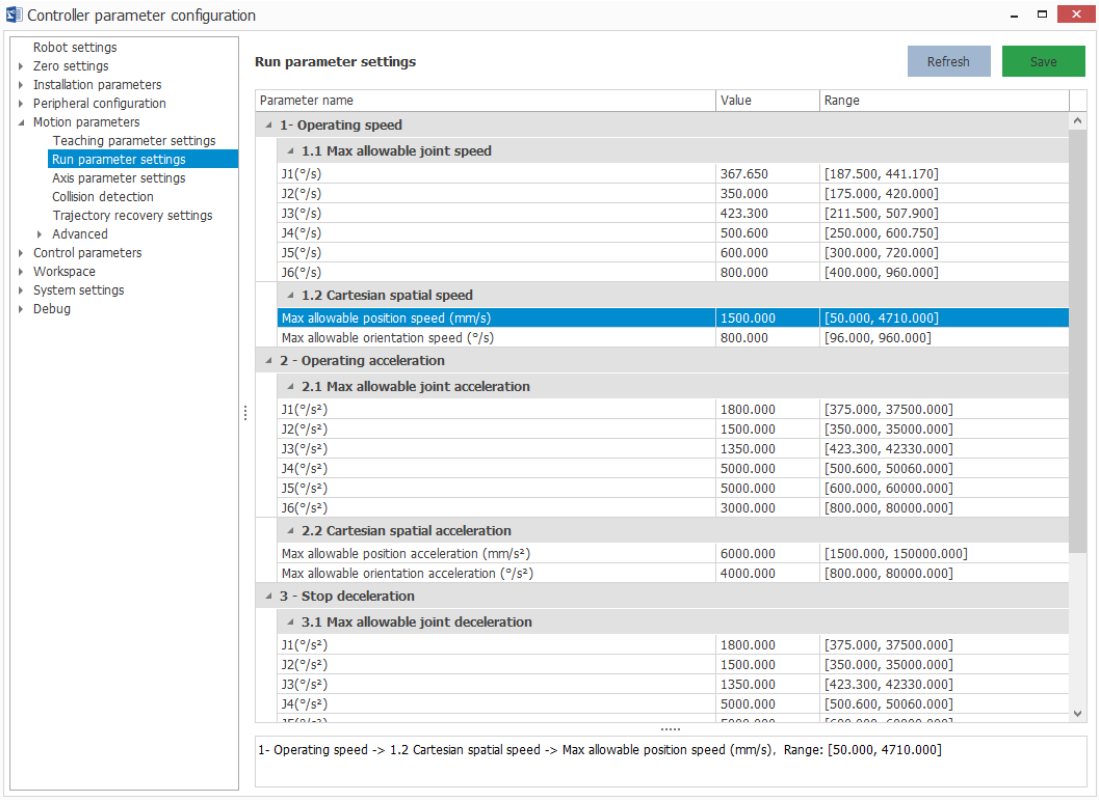
The settings here is to set the speed and acceleration of the robot motion in various frames. In the manual mode of the program, the stepwise or continuous operation of the robot uses 50% of the operating speed and acceleration, rather than the parameters set here.

Results

The settings take effect immediately after being saved. After saving the settings, you can verify the settings on the axis control panel.

9.5.2 Run Parameter Settings

Double-click "Controller parameter configuration" in the device tree, then select "Motion parameters" > "Run parameter settings", as shown in the figure below.



Run parameter settings

Name	Meaning
Operating speed	Maximum speed during operation.
	Operating speed = max. speed * global speed percentage * percentage set by instruction V in motion instruction Affected by Velset value Remove effect by Velset OFF Speed percentage set in toolbar Affected by Velset Rate[value] Maximum speed set here
Operating acceleration	Maximum acceleration during operation.
	Operating acceleration = max. acceleration * global speed percentage * percentage set by instruction Acc in motion instruction Speed percentage set in toolbar Affected by Velset Rate[value] Maximum acceleration set here
Stop deceleration	Deceleration upon pause or stop.

CAUTION

The operating speed, acceleration, and deceleration are limited by the performance of the servo system.

Transition settings: The transition precision set in the transition feature represents the transition unit length between two adjacent motion instructions.

The transition levels are described as below.

Type	Description
Z[0]	No transition. The robot cannot be ensured to reach the target point precisely.
Z[1]	Transition at 1x transition unit length.
Z[2]	Transition at 2x transition unit length.
Z[3]	Transition at 3x transition unit length.
Z[4]	Transition at 4x transition unit length.
Z[5]	Transition at 5x transition unit length.

NOTE

In the manual mode of the program, the stepwise or continuous operation of the robot uses 50% of the operating speed and acceleration. The stop speed parameters used are the same as those used in the auto mode.

9.5.3 Axis Parameter Settings

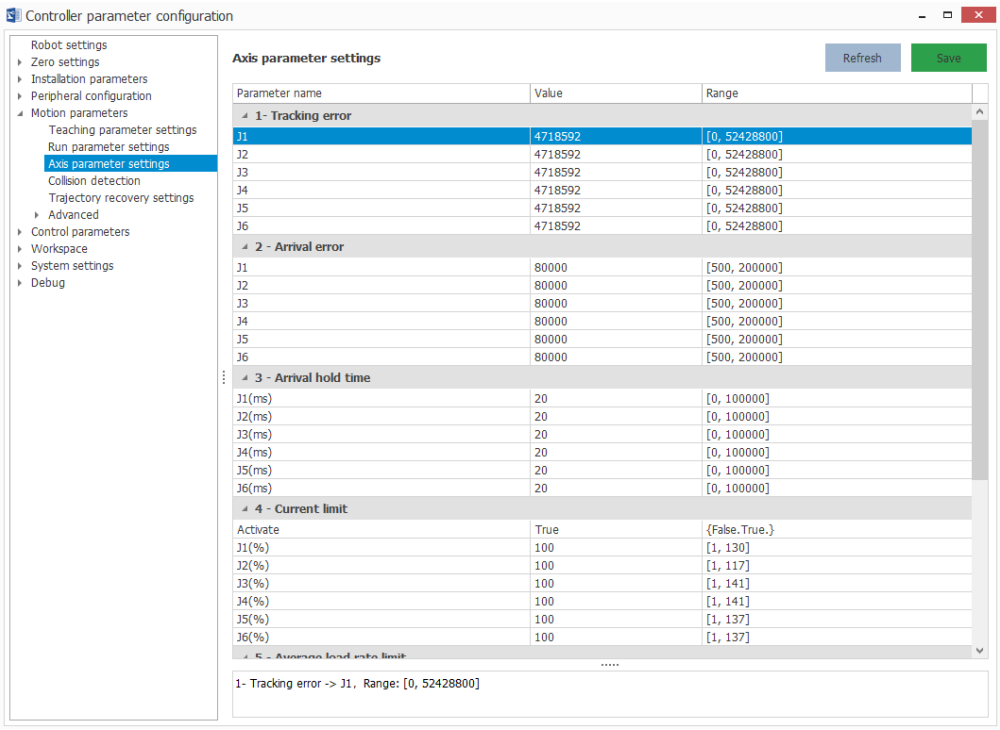
The axis parameters include tracking error, arrival error, arrival hold time, current limit, average load rate limit, overheat rate protection and overload rate protection.

- The tracking error is the threshold used to monitor whether the deviation between the planned position of the axis and the feedback position exceeds the threshold during robot operation. If the limit is exceeded, an alarm occurs. In this case, properly decrease the acceleration or increase the tracking error on the premise that the motor can respond to motion instructions.
- The arrival error is the threshold for determining whether the motor reaches the planned position of the robot. If the arrival error is exceeded, it is considered that the motor fails to reach the planned position.
- The arrival hold time is the time for determining whether the motor has continuously reached the planned position. The robot does not immediately determine that the axis arrives at the planned position the moment the difference between the actual position of the motor and the planned position becomes smaller than the arrival error. Instead, it determines that the axis arrives at the planned position when the difference is smaller than the arrival error for a period of time, so that the mechanical and control system can stabilize. In this way, accurate point control is achieved.
- The current limit and average load rate limit protect the motor and reducer from damage caused by overload. They are strongly related to the current protection function.
- The overheat rate protection and overload rate protection protect the motor and the reducer from damage caused by overtemperature. They are strongly related to the temperature protection function. When the overheat rate protection is set to "False", the motor/reducer overheat warning/alarm will not

occur. When the overload rate protection is set to "False", the motor/reducer overload warning/alarm will not occur.

Steps

Double-click "Controller parameter configuration" in the device tree, then select "Motion parameters" > "Axis parameter settings", as shown in the figure below.



Axis parameter settings

Results

The settings take effect immediately after being saved. After saving the settings, you can verify the settings on the axis control panel or through programming.



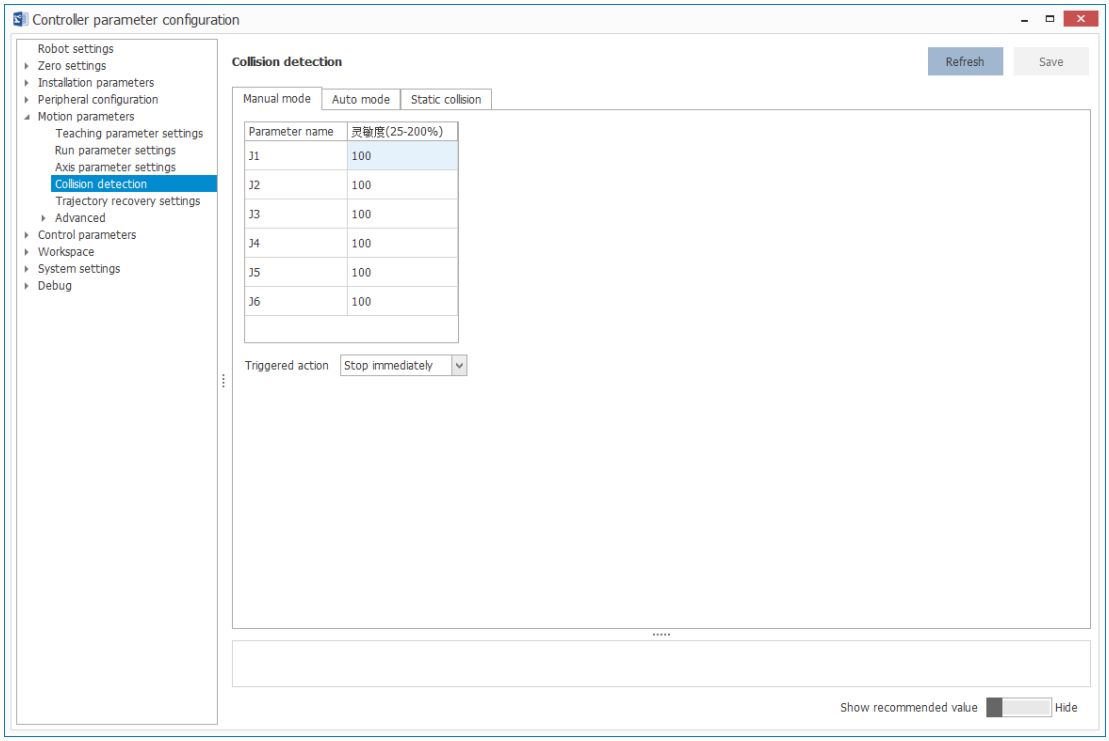
- What is set here is the speed and acceleration limits of the robot's motion in various frames. In the manual mode of the program, the stepwise or continuous operation of the robot uses 50% of the operating speed and acceleration, rather than the parameters set here.
- The actual operating speed, acceleration, and deceleration are limited by the performance of the servo system.

9.5.4 Collision Detection

You can configure the actions taken by the robot after a collision occurs.

Steps

Double-click "Controller parameter configuration" in the device tree, then select "Motion parameters" > "Collision detection", as shown in the figure below.



Collision detection

Results

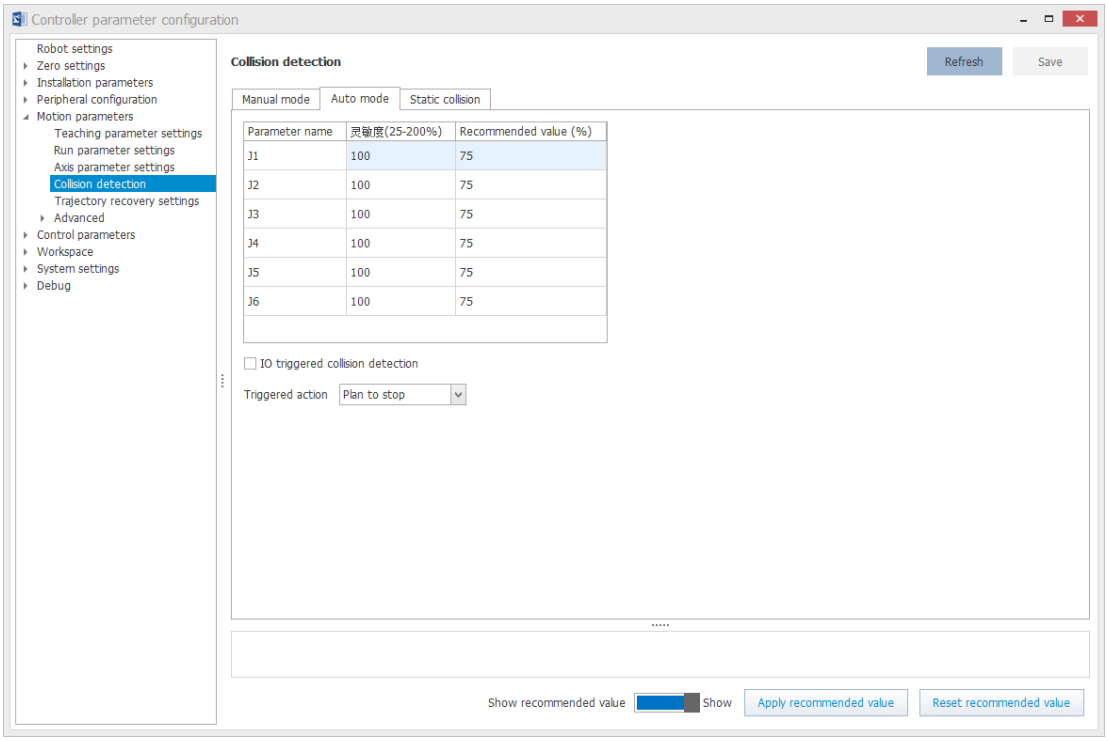
The collision detection parameters are divided into collision detection parameters in manual mode and collision detection parameters in auto mode. The collision detection parameters in auto mode are valid only in the auto mode.

The collision detection parameters are divided into 2 types:

- Parameters for sensitivity, with a range limited to 25-200 and a default value of 100.
- Parameters for triggered action after a collision, currently supporting three actions: Plan to stop, Stop immediately, and Immediate rollback.



- "Alarm and move back" action requires special use scenarios. Before use, see the 12.6 Collision Response section.
- Since there is no collision detection switch in the interface, you can temporarily disable the collision detection function by calling the SetAxisCollMode instruction, and then re-enable it by calling the instruction again after the operation is complete.



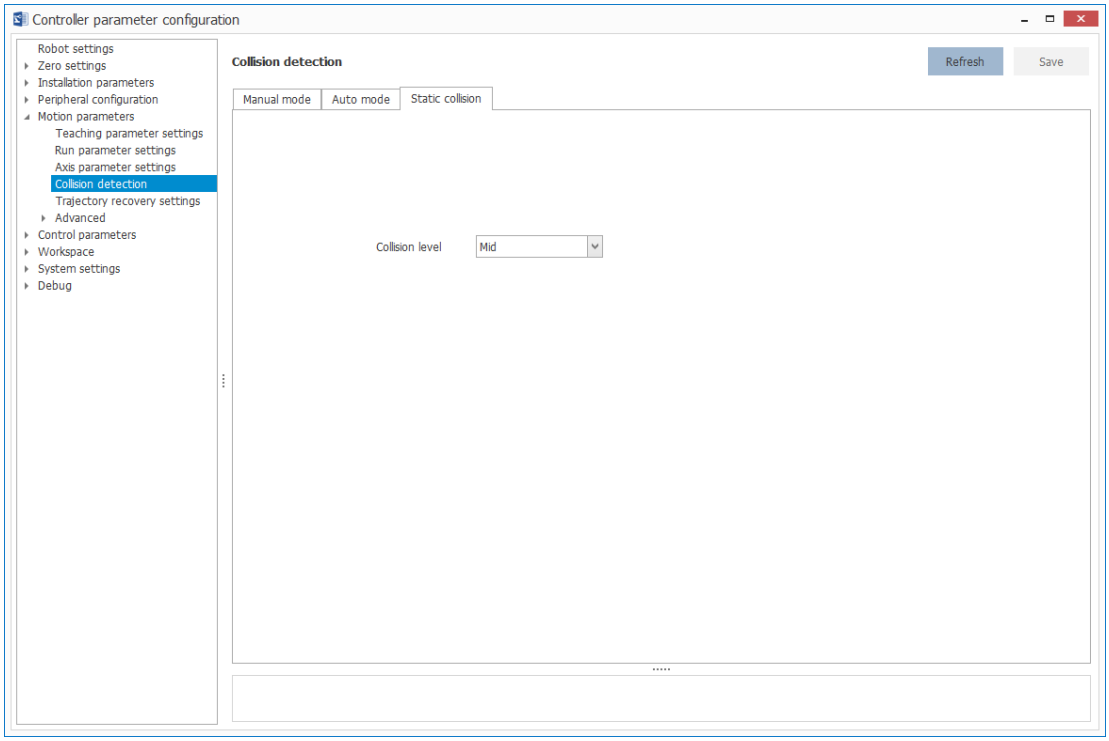
- In addition to the parameters mentioned above, The "Show recommended value" slide button is used to obtain the recommended sensitivity of the system.
- When you click the "Show recommended value" slide button, the "Apply recommended value" and "Reset recommended value" buttons are displayed on the right. The recommended value of each axis is displayed in the table in real time.
- The recommended sensitivity is automatically reset to 100 after each power-on.
- Each time the robot moves, the system automatically calculates the recommended sensitivity. When the recommended sensitivity calculated by the system is greater than the sensitivity value displayed on the interface, the sensitivity value displayed is updated.
- The "Reset recommended value" allows you to reset the displayed recommended sensitivity to 100. When the recommended sensitivity has increased unexpectedly due to minor contact of the robot, you can click this button to calculate the recommended sensitivity again.

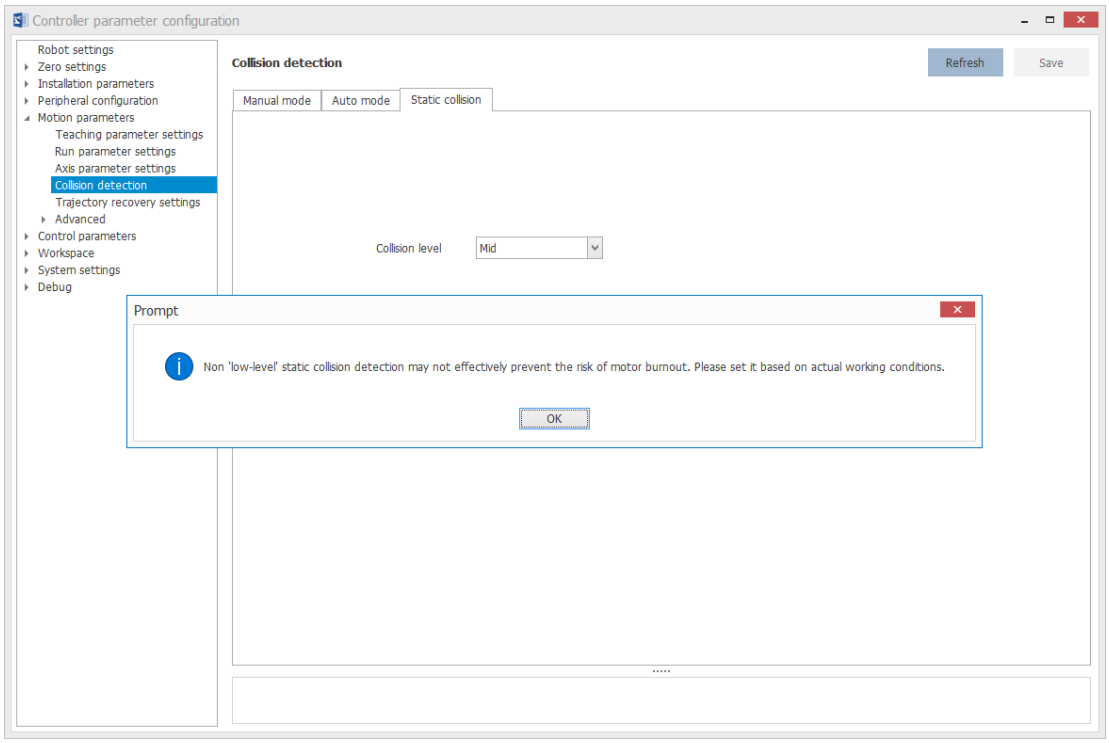
NOTE

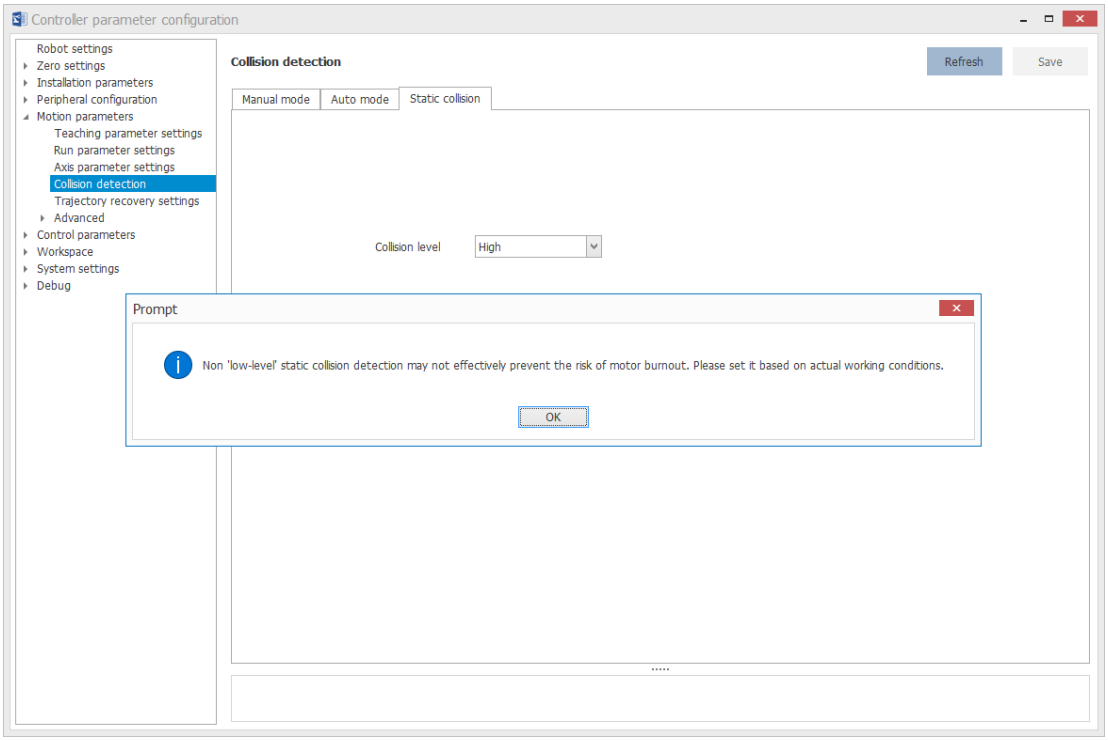
- To avoid false alarms and ensure sufficient sensitivity when the operating conditions change significantly (such as a significantly larger or smaller load, a significantly faster or slower speed, or an abnormal increase in the recommended value due to a minor collision or contact), it is recommended to reset the recommended value and apply the latest recommended value after the robot runs for a period of time (greater than 3 minutes).
- After the robot runs for a long time, the auto-recommended value may change even if the robot still runs under the same operating conditions due to joint friction. When the change is within 10%, it is not necessary to frequently adjust the collision detection sensitivity (unless a false alarm has occurred). If the change exceeds 10%, it is recommended to apply the latest recommended value.
- To reduce false alarms, if the calculated recommended value is less than 100%, it is not recommended to change it to a value less than 100%. Otherwise, false alarms may occur when the operating conditions change. It is appropriate to set the sensitivity between 75% and 100% if and only if you have a high requirement for collision detection sensitivity.
- If you need to switch the speed manually during automatic motion, if the speed changes greatly (for example, directly from 25% to 100%), a false alarm may occur even if the recommended sensitivity at 100% speed is used. This is normal. If you need to switch the speed manually during automatic motion, it is recommended to switch the speed step by step. For example, switch from 25% to 50% first, then to 80% and finally to 100%.

Static collision detection function

Double-click "Controller Parameter Configuration" in the device tree, then select "Motion parameters" > "Collision detection" in the opened interface to configure relevant settings. The static collision detection level is divided into Low, Medium, and High. When switching to Medium or High level, a prompt will pop up and require manual closure by the user, with the message "Non 'low-level' static collision detection may not effectively prevent the risk of motor burnout. Please set it based on actual working conditions." See the figure below:


Controller parameter configuration
Robot settings
Zero settings
Installation parameters
Peripheral configuration
Motion parameters
Teaching parameter settings
Run parameter settings
Axis parameter settings
Collision detection
Trajectory recovery settings
Advanced
Control parameters
Workspace
System settings
Debug
Collision detection
Refresh
Save
Manual mode
Auto mode
Static collision
Collision level
Mid


Controller parameter configuration
Robot settings
Zero settings
Installation parameters
Peripheral configuration
Motion parameters
Teaching parameter settings
Run parameter settings
Axis parameter settings
Collision detection
Trajectory recovery settings
Advanced
Control parameters
Workspace
System settings
Debug
Collision detection
Refresh
Save
Manual mode
Auto mode
Static collision
Collision level
Mid
Prompt
Non 'low-level' static collision detection may not effectively prevent the risk of motor burnout. Please set it based on actual working conditions.
OK



NOTE

This function is to add collision detection for the robot when the enable signal is active and not in motion. You can set the collision level to low, intermediate, or high. The low collision level is the default and also the safer setting. If you have special operating condition requirements, the collision level can be set to intermediate or high, but the protection effectiveness of these two levels will be reduced. You can set it as appropriate.

Debugging steps for collision detection

1. Check the load parameters.
   The load parameters include mass, eccentricity, inertia, etc. and can be calculated from the 3D model of the load. For loads with a simple shape, you can calculate the load parameters by referring to the calculation method in the Appendix.

   Enter the load parameters to the selected tool load or grip load.

2. Activate the tool.
   There are two methods to activate the tool load:
   - Add the corresponding tool number to the motion instruction. This method takes effect in the automatic, stepwise and continuous running states.
   - Switch the corresponding tool number in the tool frame interface. This method takes effect in the jog state.

3. Activate the corresponding grip load.
   There are two methods to activate the grip load:
   - Activate the specified grip in the program through the GripLoad instruction. This method takes effect in automatic, stepwise and continuous running states.

- Switch the corresponding grip load number in the monitoring interface. This method takes effect in the jog state (if the grip load number is not changed in the program by using the GripLoad instruction, the grip load number set in the monitoring interface is always active).

4. Set the collision detection parameters for the teaching state.
   Adjust the collision detection sensitivity for manual mode to 200%, and click the "Save" button. Then click the "Show recommended value" button to display the "Recommended value" column. After a period of teaching operations, return to the manual collision detection parameter interface. Click "Apply recommended value" to write the recommended value to the "Sensitivity" column on the left and click the "Save" button.

   Adjust the collision detection sensitivity for manual mode to 200%, and click the "Save" button. Then click the "Show recommended value" button to display the "Recommended value" column. After a period of teaching operations, return to the manual collision detection parameter interface. Click "Apply recommended value" to write the recommended value to the "Sensitivity" column on the left and click the "Save" button.

   CAUTION

   Normally, the factory-set sensitivity can prevent false alarms. The auto-recommended sensitivity is only updated during teaching motion, and it must be ensured that the teaching motion time is sufficient (approximately 30s for an update, but it is recommended to perform teaching motion for more than 2 minutes); if the recommended value exceeds 150, check whether the tool load during teaching is set correctly.

5. Set the collision detection parameters for auto mode.
   Similar to the settings for the manual mode, adjust the collision detection sensitivity for auto mode to 200% and click the "Save" button. Then click the "Show recommended value" button to display the "Recommended value" column. After a period of automatic operation (5 minutes and more recommended), return to the auto collision detection parameter interface. Click "Apply recommended value" to write the recommended value to the "Sensitivity" column on the left. Then, turn on the collision detection switch and click "Save".

   If a collision alarm is detected, do as follows.

   1) Check whether parameters are set properly.

   a) Check whether the collision detection sensitivity is too low. When the sensitivity is less than 100, false alarms are more likely to occur. Ensure that all sensitivity levels are greater than or equal to 100.

   b) Check that the load matches the set tool load. Check whether the correct tool number is activated. If false alarm occurs in automatic, stepwise or continuous running, check whether the correct tool number is included in the instruction; check whether the corresponding tool load parameters are correct. Check whether the eccentricity is set correctly, especially when the eccentricity is large.

   c) Check whether the load matches the set grip load. Check whether the correct grip load number is activated, whether the correct grip load number is activated using the GripLoad instruction, or whether the grip load number is correct on the monitoring interface; check whether the corresponding grip load parameters are correct. Check whether the eccentricity is set correctly, especially when the eccentricity is large.

   2) If no fault is found in the preceding parameter check, check the operating conditions of the robot.

a) Check whether the operating conditions at the time of alarm are greatly different from those at the time of sensitivity setting. The required collision detection sensitivity may vary under different operating conditions. In addition, if the ambient temperature at the time of the alarm differs greatly from that at the time of parameter setting, collision detection may be affected. In this case, apply the latest recommended value.

b) Check whether the robot comes into contact with the outside world during motion. In case of contact, the torque fed back to the robot motor is unpredictable because the contact force is unpredictable. Therefore, the set collision detection sensitivity cannot be used for a long time. In this case, it is recommended to temporarily increase the detection sensitivity by using SetAxisCollLevel instruction near instructions in the program that may have contact with the outside world, or temporarily disable the collision detection function using the SetAxisCollMode instruction. This method is available only in automatic mode.

c) Check whether there is strong vibration during robot motion. The strong vibration greatly affects the collision detection and easily causes false alarms. In this case, it is recommended to increase the set sensitivity by 10 to 30 on the basis of the recommended value.

3) Check the robot manipulator.

a) Check whether the robot base or robot end gripper is properly fastened.

b) Check whether the brake of the alarming axis is opened. Turn off collision detection (with sensitivity of 200%), run the axis with false alarms (low-speed teaching) independently. Open the current protection interface, observe whether the average load rate and maximum current of the axis are too large, and whether an axis drive alarm occurs. If yes, check the servo parameters or cables related to the brake.

c) When the robot servo is enabled, rotate the robot axis (the axis with alarm) firmly and observe whether the robot has a large clearance. If the clearance is large, the reducer or timing conveyor is abnormal and needs to be dismantle for inspection.

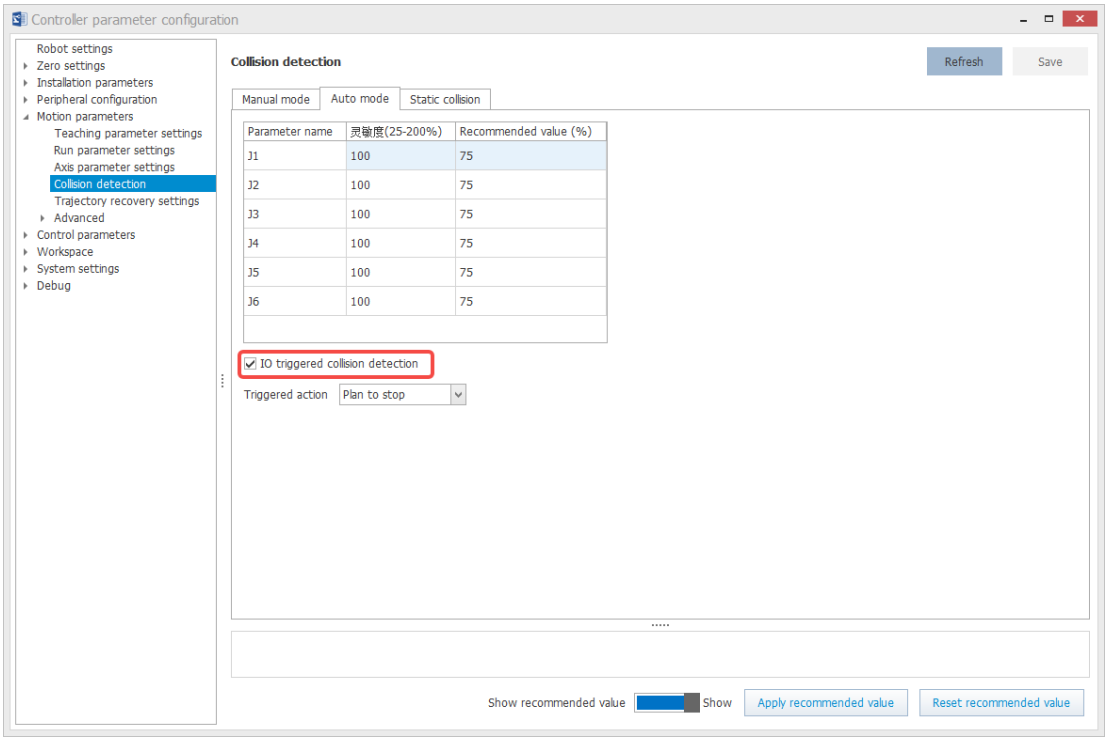
d) If no problem is found in above checks, it is possible that the reducer is damaged. Export the fault information promptly and contact the manufacturer for analysis.

4) If a collision alarm occurs with the collision detection sensitivity set to 200%, and all previous troubleshooting steps and attempts fail to resolve the issue, you can adjust the static collision level to medium or high, and observe whether the false alarm disappears. If adjusting to the high level still results in a collision alarm, it is recommended to export the system diagnosis and check if the robot is continuously in contact with external objects.

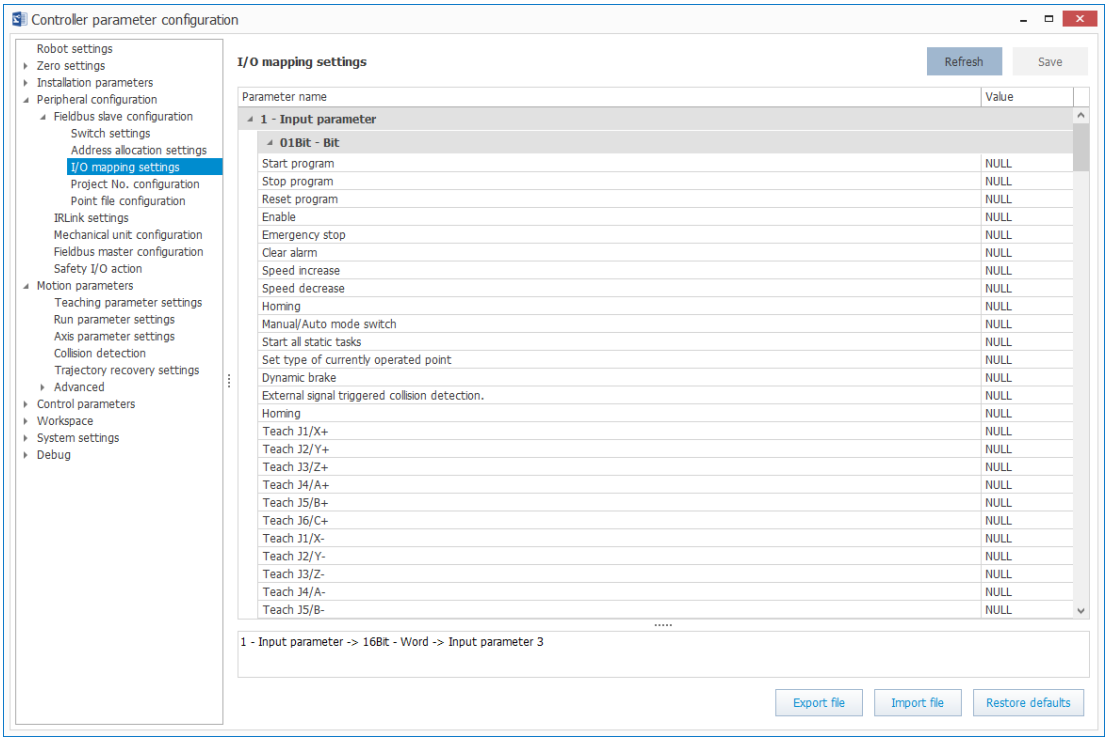
External signal triggered collision detection:

In auto mode under remote I/O mode, collision signals can be triggered by external inputs. The specific procedure is as follows:

Enable the I/O triggered collision detection function:



Set the I/O mapping:



After the above configuration is completed, when the robot is in remote I/O mode and automatically running the program, upon receiving a rising edge signal (such as ON for In[4 as shown above] in the figure above) from an external input, the robot will perform actions according to the user-set collision handling actions.

CAUTION

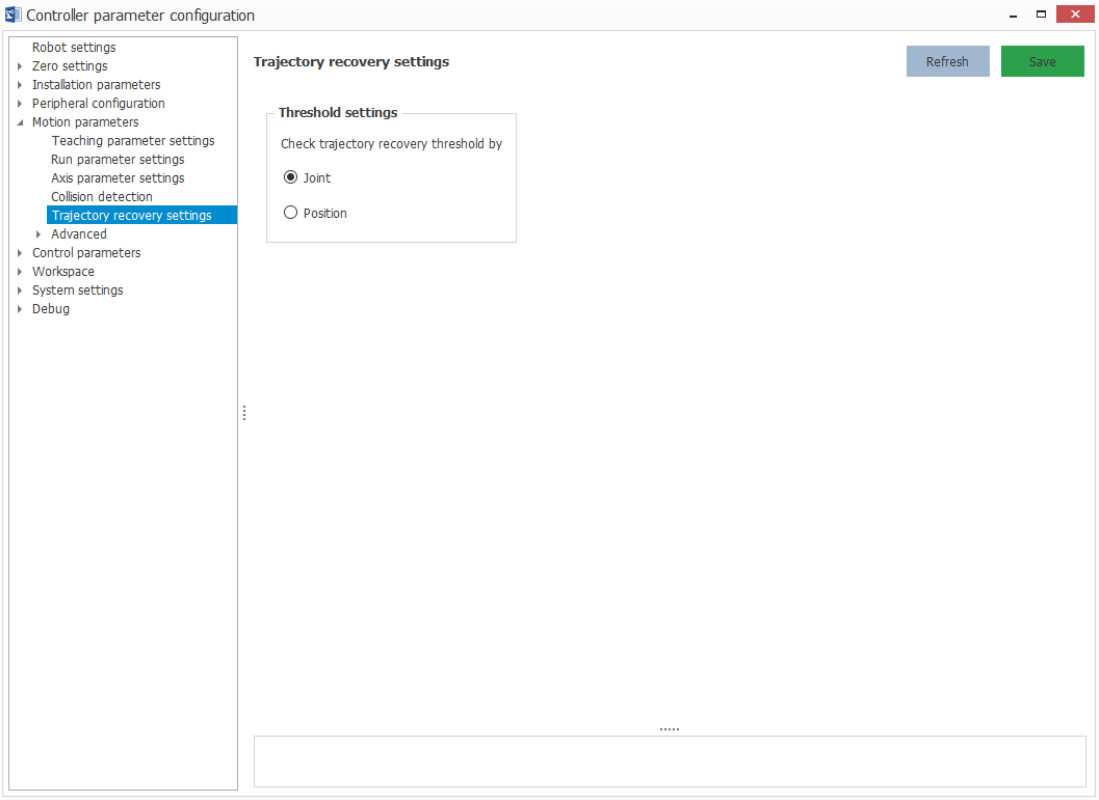
- The collision detection function is only effective in the auto mode under remote I/O mode.
- The collision detection function is triggered by a rising edge.

9.5.5 Trajectory Recovery Settings

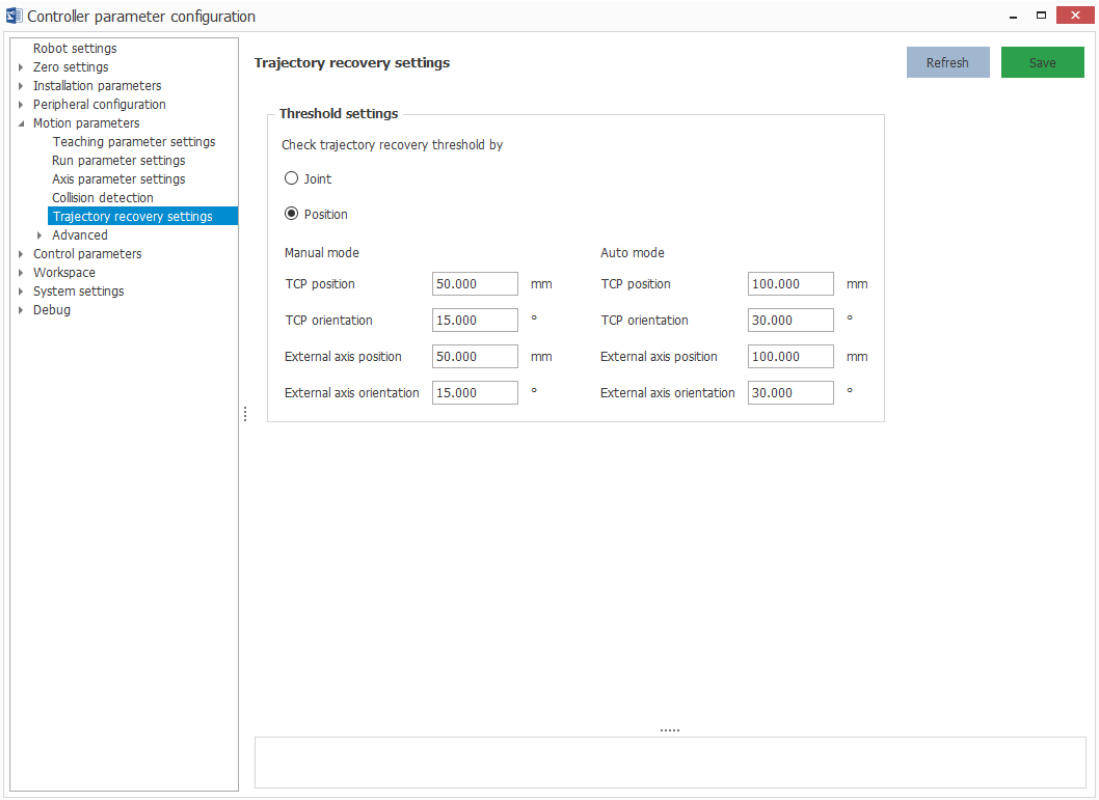
- You can choose to check the threshold for trajectory recovery of the robot by joint or by position. If you choose joint, the robot determines whether to pop up a prompt for trajectory recovery by comparing the changes in joints.
- If you choose position, the robot determines whether to pop up a prompt for trajectory recovery by comparing the change in position of the TCP and the external axis. When you choose position, you can set TCP position, TCP orientation, external axis position, and external axis orientation for the manual and automatic modes.

Steps

Double-click "Controller parameter configuration" in the device tree, then select "Motion parameters" > "Trajectory recovery settings", as shown in the figure below.



Joint mode



Position mode

Results

When you choose the joint mode, the settings take effect after being saved. When you choose the position mode, enter the setting data and save the settings to make them take effect.

CAUTION

- This function is only available to the manufacturer. The relevant parameters set will only take effect under the disabled state.
- It is recommended that the threshold in position mode be set to a value above 0.1 mm or 0.1°. When it is set to 0, the prompt keeps popping up.

The following table lists the parameter description.

Parameter	Description	Default for manual mode	Default for auto mode	Unit	Range
TCP position	The maximum allowable TCP distance between the robot arm deviation point and the point to be reached.	50	100	Millimeter (mm)	[0, 20000]

Parameter	Description	Default for manual mode	Default for auto mode	Unit	Range
TCP orientation	The maximum allowable TCP orientation deviation between the robot arm deviation point and the point to be reached.	15	30	Degree (°)	[0, 360]
External axis position	The maximum external axis distance between the external axis deviation point and the point to be reached.	50	100	Millimeter (mm)	[0, 20000]
External axis orientation	The deviation of the rotation angle of the external axis between the external axis deviation point and the point to be reached.	15	30	Degree (°)	[0, 360]

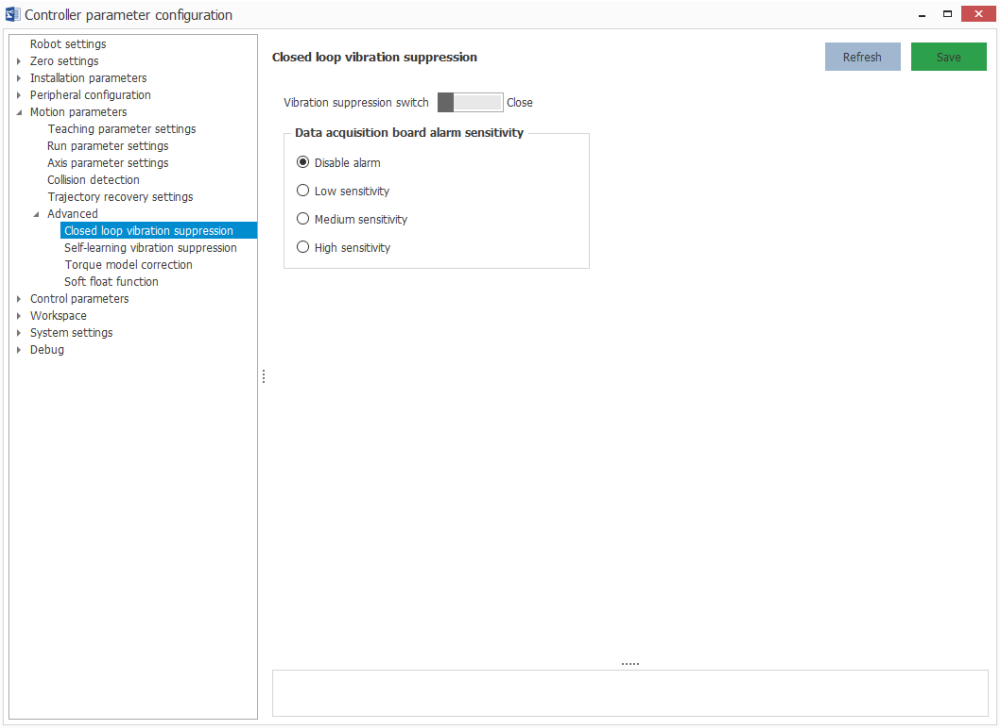
9.5.6 Advanced

9.5.6.1 Closed-loop Vibration Suppression

This function is only available for IRCB501, IRCB500 and IRCB100 series controller which comes with a data acquisition board.

Steps

Double-click "Controller parameter configuration" in the device tree, then select "Motion parameters" > "Advanced" > "Closed loop vibration suppression", as shown in the figure below.



Closed loop vibration suppression

Results

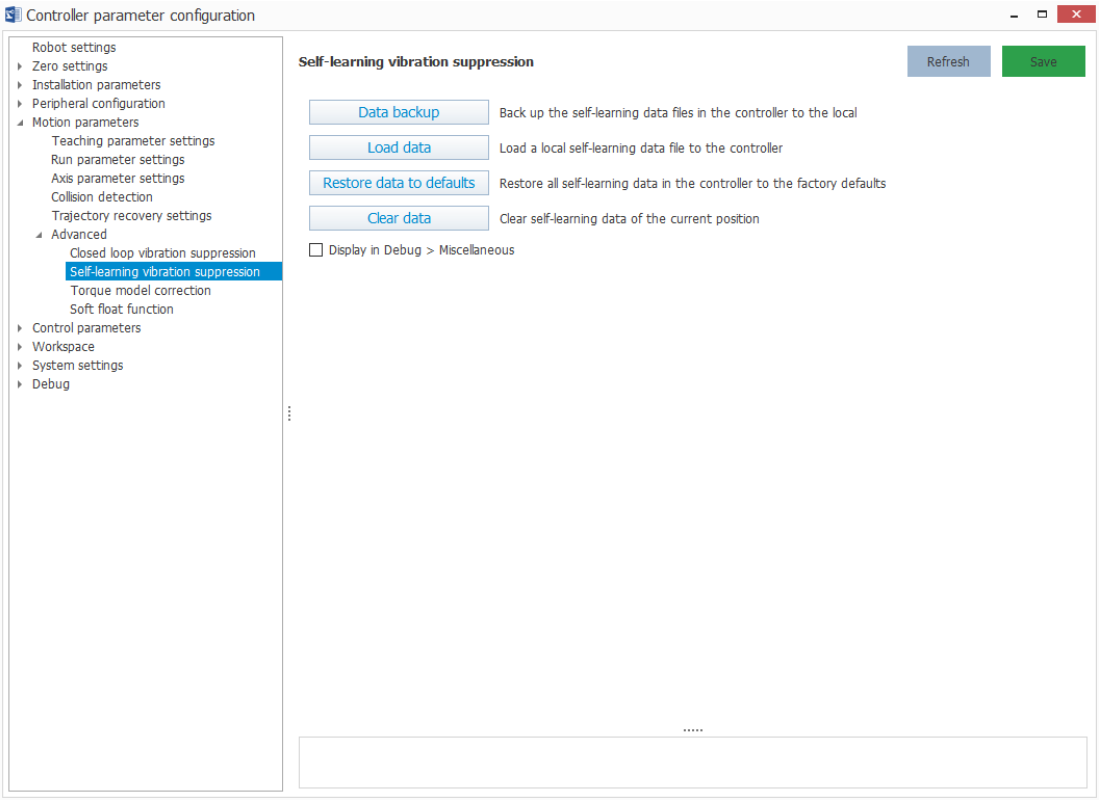
Set and save settings to take effect.

> NOTE
>
> When the data collection board is missing or damaged, set the data collection board alarm sensitivity to "Disable alarm" to avoid continuous alarms.

9.5.6.2 Self-learning Vibration Suppression

You can back up the learning data file of a robot that has already undergone self-learning through InoRobotLab and teach pendant. Then you can import the learning data file into a robot that has not undergone self-learning and works under the same working conditions. This can save self-learning time for robots that have not undergone self-learning.

Go to "Controller parameter configuration" > "Motion parameters" > "Advanced" > "Self-learning vibration suppression". Four options are included on the interface, including "Data backup", "Load data", "Restore data to defaults", and "Clear data".



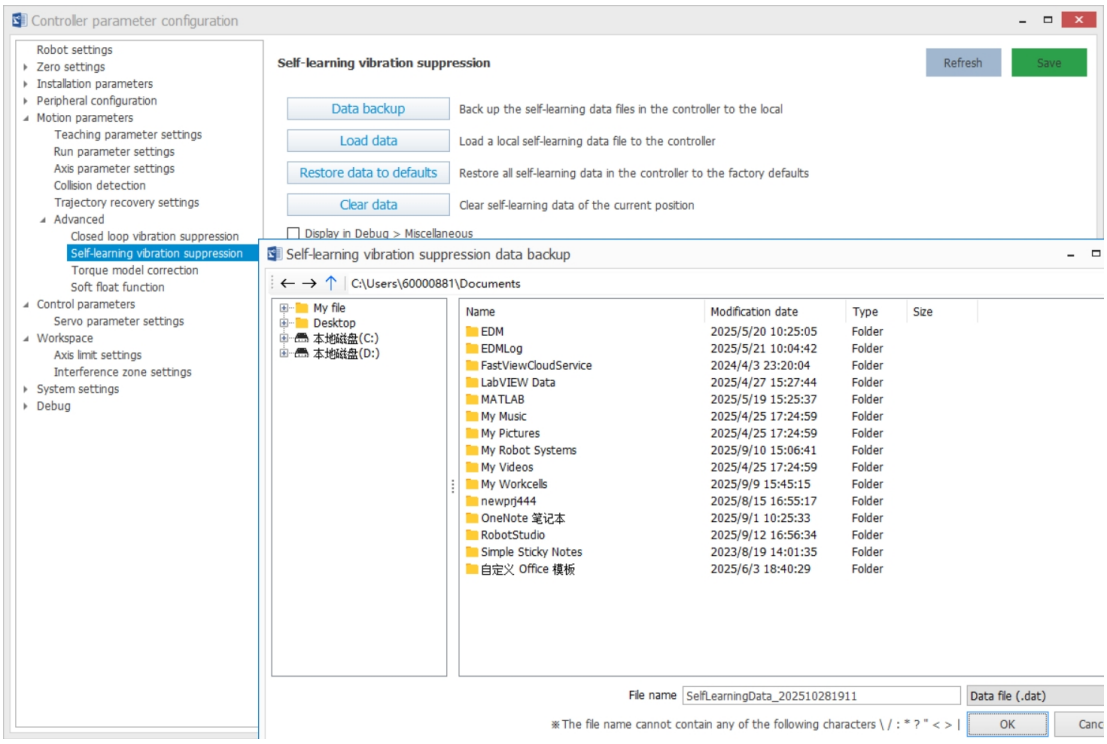
Data backup

- Description:

It allows you to save the auto-learning data in the robot controller to the learning data file and store the file to the local disk of the computer.

- Steps:

1. Click the "Data backup" button.
2. Select the save path and enter the file name in the pop-up box, as shown below.



3. Click "OK". After the backup is complete, you can find the backup file in the specified path.

> **NOTE**
>
> The default save path is the last used save path. The default file name automatically includes the project name and time information for ease of file management.

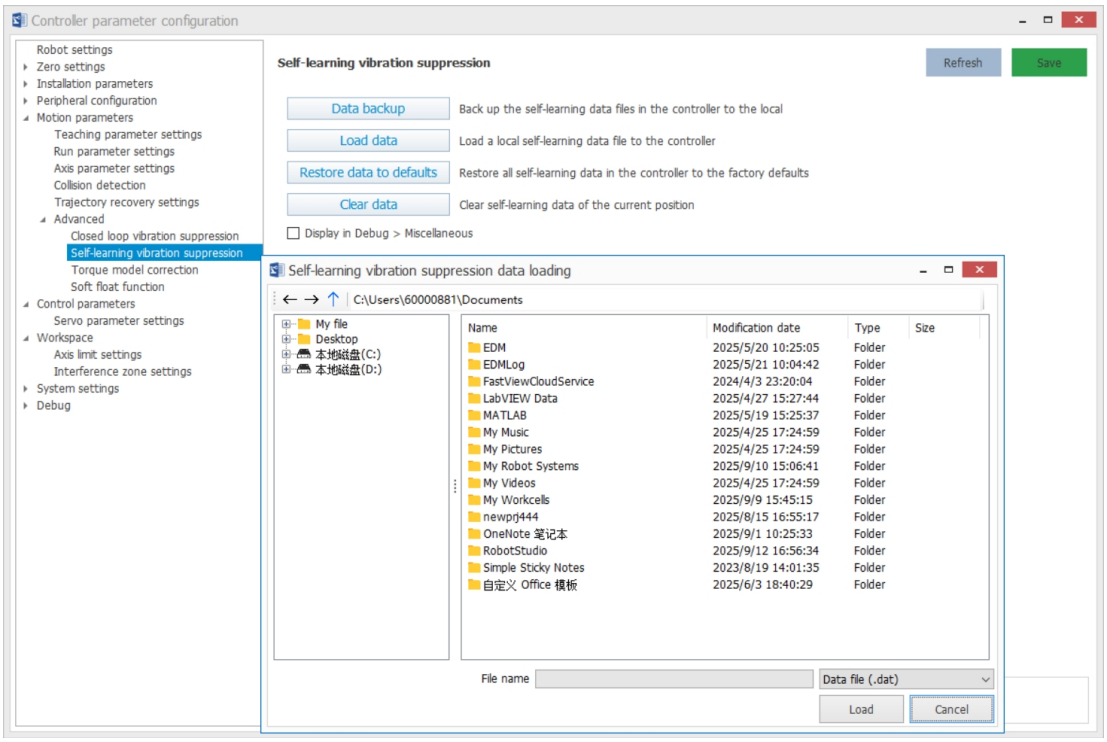
Load data

- Description:

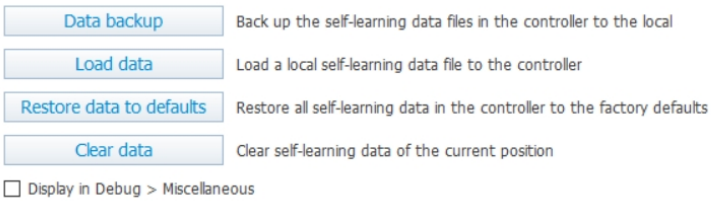
It allows you to load the locally stored learning data file into the robot controller.

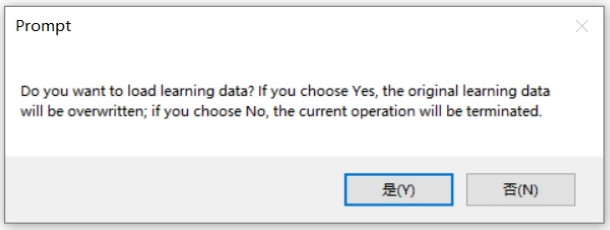
- Steps:

1. Click the "Load data" button.
2. Select the file to be loaded in the pop-up box, as shown below.



3. After clicking the "Load" button, you will be prompted whether to load the learning data. Select "Yes" to load the learning data, or "No" to cancel the current operation. After the data is loaded successfully, the original learning data will be overwritten and the message "Auto-learning vibration suppression data file imported successfully" will be displayed in the message output box.





NOTE

- When the learning data is loaded, the file type, robot model, software version, and check code information are verified. If the robot model, software version, and controller model of the loaded learning data file do not match, you will be notified in the message output box that the load fails.
- If the content of the learning data file is changed, you will also be notified in the message output box that the load fails.
- If you import the learning data from versions before V4R24C2 into V4R24C2 and later versions, the robot may stop to learn the same project again. When the old version of learning data is imported, it is recommended that the robot run the program once for program debugging, and then put the robot into formal operation after confirming that the learning is complete.

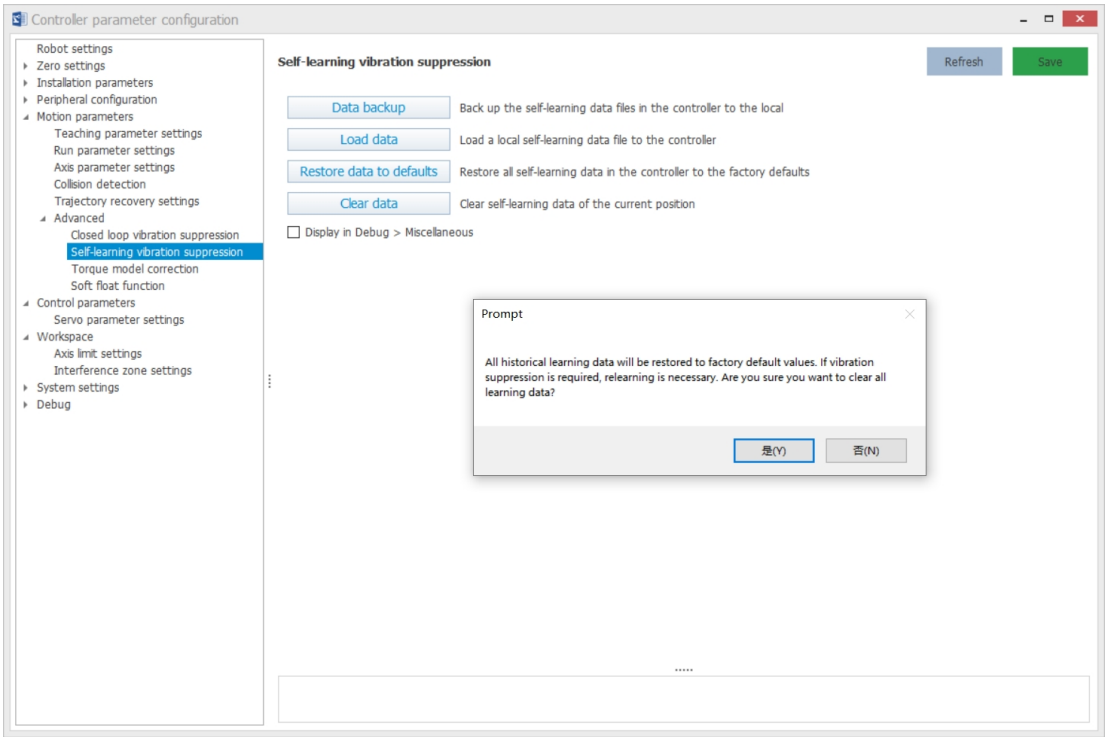
Restore data to defaults

- Description:

It allows you to restore all auto-learning data in the controller to default values.

- Steps:

1. Click the "Restore data to defaults" button.
2. After clicking the button, you will be prompted whether to restore the learning data to defaults. Select "Yes" to restore the learning data to defaults, or "No" to cancel the current operation.



NOTE

- After the learning data are restored to default values, the auto-learning vibration suppression function is not effective because there is no valid learning data. You need to perform auto-learning again to make the function take effect.
- When the system is re-programmed, the learning data are restored to the default values. During system upgrade, the data file will not be automatically upgraded. If the data file is not compatible after the upgrade, you will be prompted to upgrade the data file.

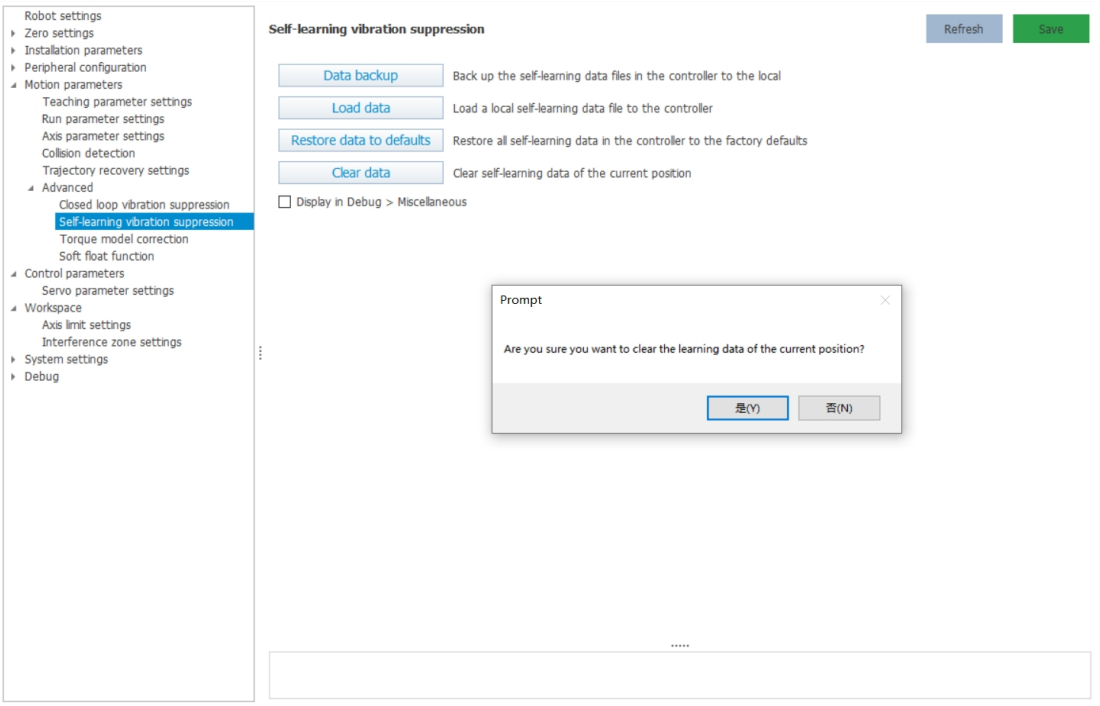
Clear data

- Description:

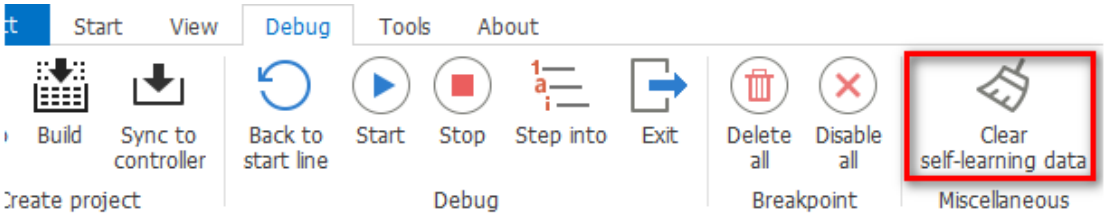
It allows you to clear the auto-learning data of the robot's current position in the controller.

- Steps:

1. Click the "Clear data" button.
2. After clicking the button, you will be prompted to confirm whether to clear the learning data of the current position. Select "Yes" to clear the learning data, or "No" to cancel the current operation.



3. Also, you can choose to display the "Clear data" button in the "Debug" menu for quick access to this function.



General specifications of data file

- Description:

If the robot model in the data file is different from the robot model in the controller, the following conditions must be met before the data file can be loaded into the controller for use.

- The number of axes of the robot model in the data file must be the same as that of the robot model in the controller.
- For the SCARA robots, the robot model in the data file must have the same arm length, load and lead screw stroke as the robot model in the controller.
- For the 6-axis robots, the robot model in the data file must have the same arm length and load as the robot model in the controller.

CAUTION

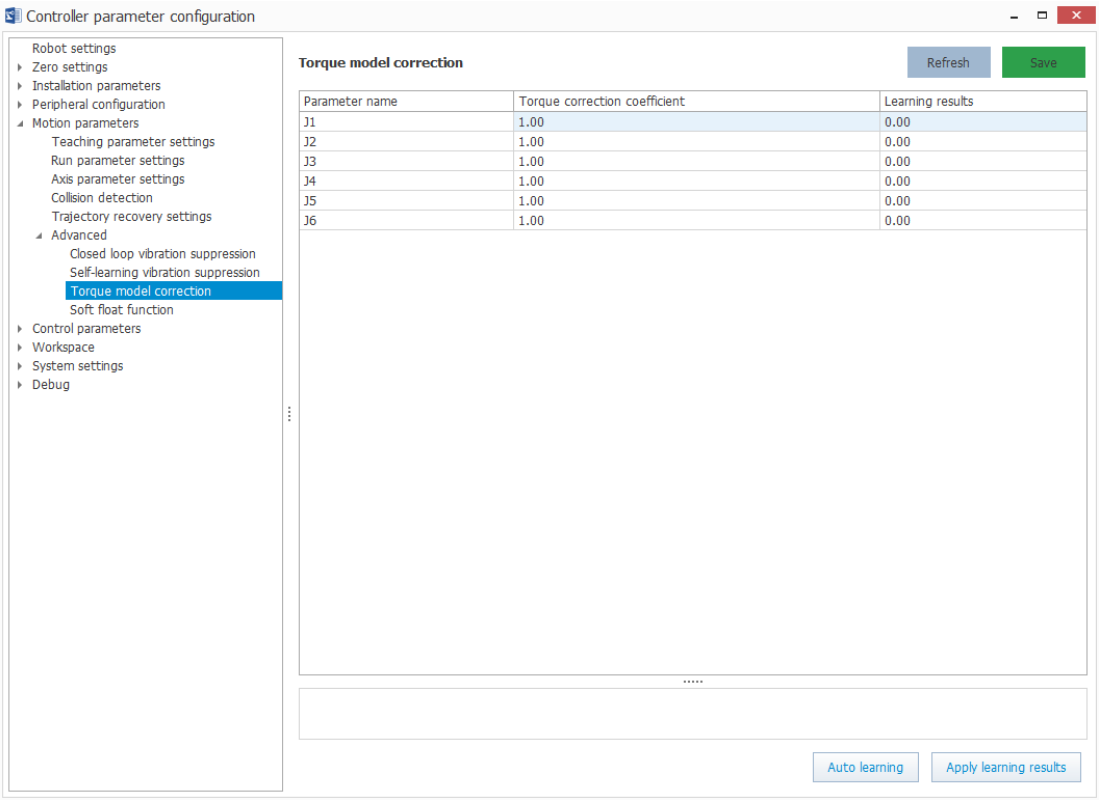
- When the robot model in the data file is different from the robot model in the controller, the data file meeting above specifications can be loaded into the controller for use. However, the actual vibration suppression effect needs to be further verified by the user.
- If you need to manually modify the auto-learning vibration suppression data file, back up the data file before modifying it. Note whether the text encoding format is tampered with after modification. If it is found that the text encoding format has been tampered with after being modified and saved in the local text tool, resulting in CRC verification failure, please modify the backup file with another third-party text tool.

9.5.6.3 Torque Model Correction

When the RapidMove function is enabled, the controller automatically generates motion trajectories based on the model. If the model is inaccurate, the current will be too large, which will cause an alarm. In this case, you need to correct the model to avoid the alarm. The torque model correction function is used to set and learn these model errors.

Manual correction

If the overcurrent alarm persists when the load is set correctly, set the torque correction coefficient of the corresponding axis to a value less than 1. The smaller the parameter value, the smaller the current, and the slower the motion, as shown in the figure below.



CAUTION

This coefficient changes the joint output. If this coefficient is reduced, all motion instructions with RapidMove enabled will become slower. Therefore, if high current is present only at one point, try to reduce the local acceleration coefficient of the instruction corresponding to that point first.

Automatic correction

The automatic correction function automatically identifies the deviation between the theoretical current and the actual current based on the data recorded before the alarm, and automatically calculates the model correction coefficient. Follow the steps described below.

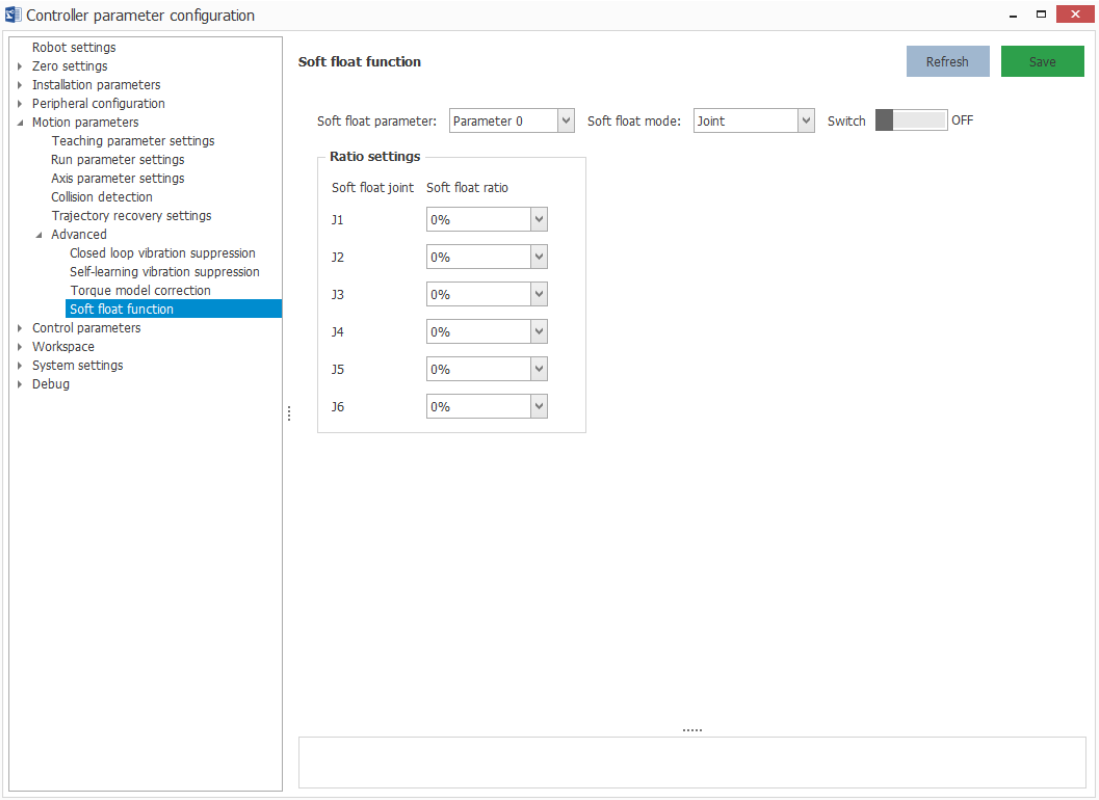
1. Wait for the overcurrent alarm to occur and robot is stop.
2. Click "Auto learning" and wait until a pop-up box appears, indicating that the learning is completed.
3. Click "Apply learning results". The learning results will be copied to the "Torque correction coefficient" section.
4. Click "Save" to save the parameters to the project file and then execute the program for verification.

CAUTION

The auto learning is based on data from a period of time before the alarm. The result of the learning only ensures that no more alarms will occur for this period of motion. Alarms may still occur when a new motion is executed, requiring learning again.

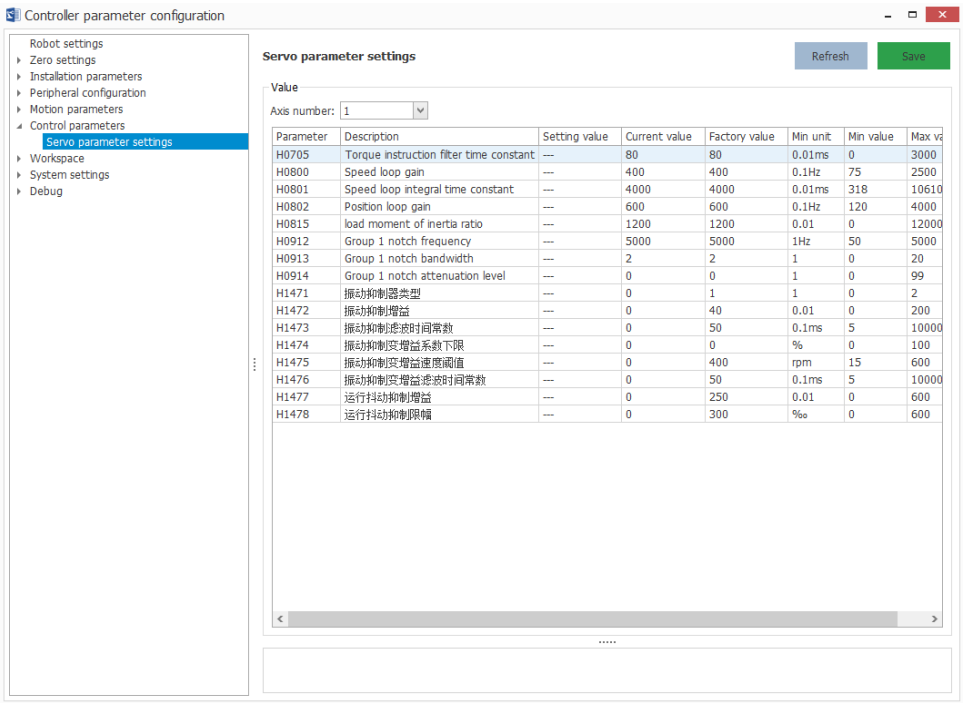
9.5.6.4 Soft Float Function

For related introduction, see the 12.9 Soft Float Function section. The parameters of the soft float function interface are shown in the figure below:



9.6 Control Parameters

It allows you to configure the servo parameters of the robot axes, as shown in the figure below.



- Refresh: Click the "Refresh" button to obtain all the servo parameters and update the monitored values on the page.
- Save: Click the "Save" button to send all setting values to the controller. After saving the settings, you can click the "Refresh" button to see whether the monitored values and setting values are the same.

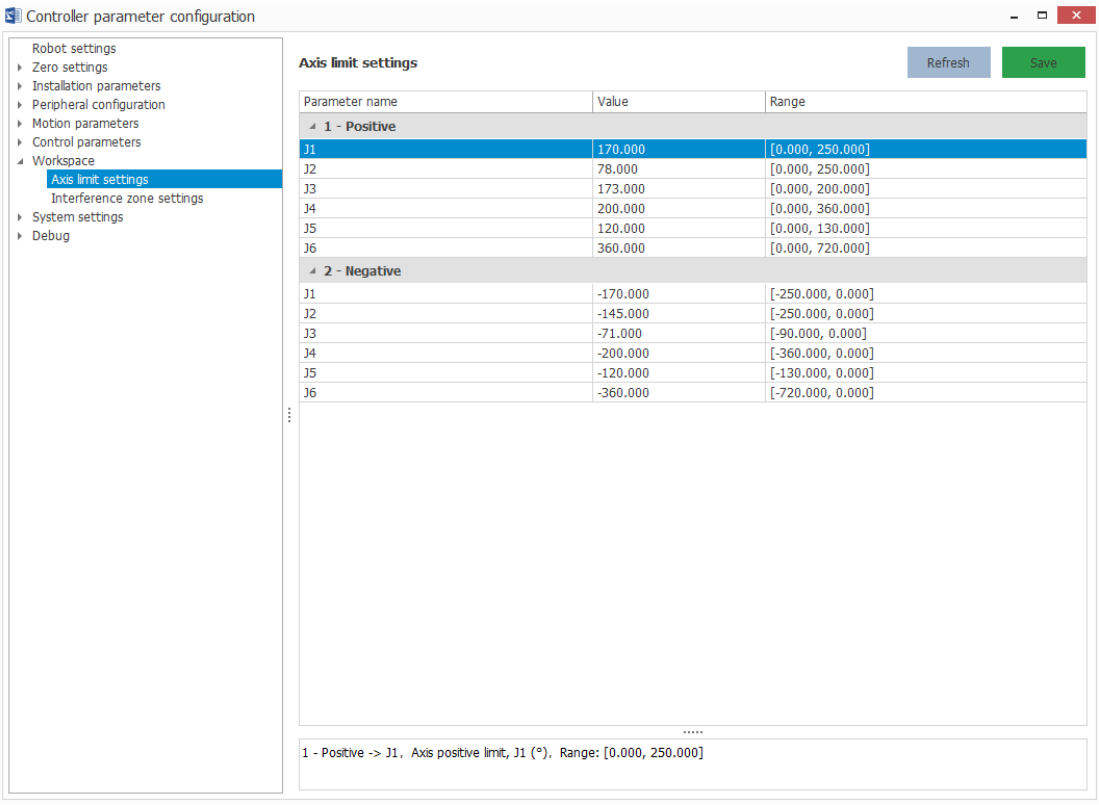
9.7 Workspace

The workspace settings include axis limit settings and interference zone settings.

- The axis limit settings define the working range of the robot in the joint space. The axis limit range is a subset of the mechanical limit range.
- The interference zone settings define the area where the robot end monitoring object cannot reach, which can be considered as the space occupied by the obstacles.

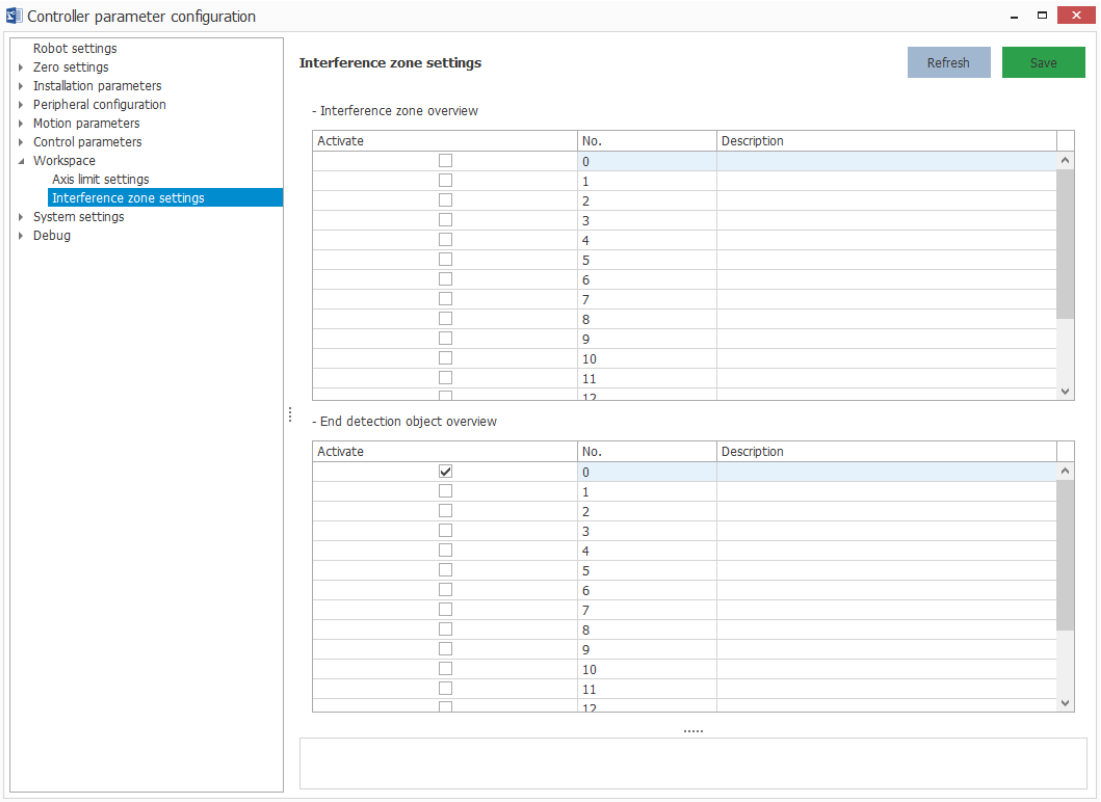
9.7.1 Axis Limit Settings

Go to Controller parameter configuration > Workspace > Axis limit settings. For specific numerical values, refer to the mechanical limit blocks and actual requirements for configuration, as shown in the figure below.



9.7.2 Interference Zone Settings

Double-click "Controller parameter configuration" in the device tree, then select "Workspace" > "Interference zone settings" in the opened interface for related settings, as shown in the figure below:



Interference zone settings

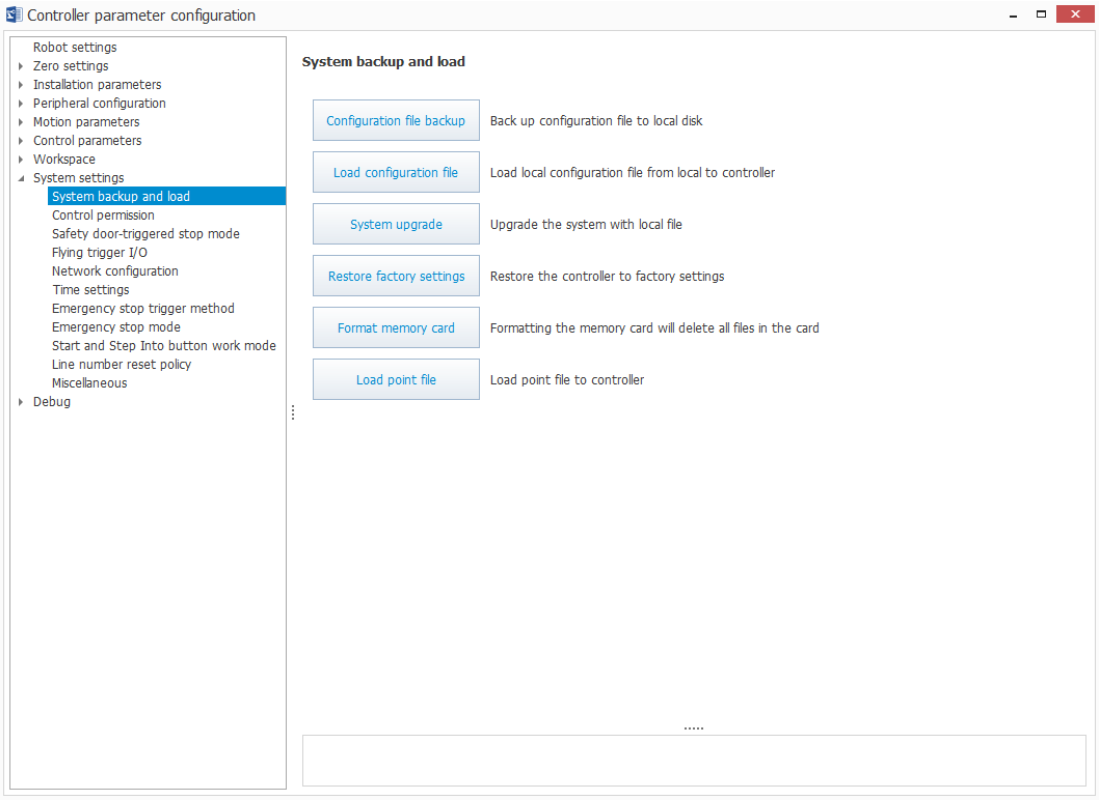
The interference zone is the area where the robot end monitoring object cannot reach. A total of 16 interference zones and 16 monitoring objects can be set.

For details, see the 12.11 Interference Zone section.

9.8 System Settings

9.8.1 System Backup and Load

The configuration file backup function is generally divided into six sub-functions: configuration file backup, configuration file loading, system upgrade, factory reset, memory card formatting, and point file loading. Double-click "Controller parameter configuration" in the device tree, then select "System settings" > "System backup and load" in the opened interface for related settings, as shown in the figure below:



Configuration file backup

- Function:

It allows you to save all robot-related parameters into a file and store the file in local disk.

- Steps:

1. Click the "Configuration file backup" button, select the save path and enter the file name.
2. After the backup is complete, you can find the backup file in the specified path.



The backup file is named "robotcfg.cfg.bk" by default and covers the following:

- Network configuration, operation logs [systemConf].
- Robot model and controller model information [initConf].
- Robot settings, peripheral configuration, motion parameters, zero point settings, etc. in controller parameter configuration [RobotParams, RobotParamsDefault].
- Process data files (vision calibration and conveyor tracking process).
- Mechanical unit configuration parameters.
- Tool, work object, and load parameters.
- BRD variables and PR variables.
- Bus files (EtherCAT I/O mapping files, master and slave management files; DeviceNet configuration files; CANopen configuration files; master bus switch configuration files).

- Results:

After the backup is completed, you can see the exported configuration file in the corresponding directory.

Load configuration file

- Function:

It allows you to import the robot-related setting parameters into the controller.

- Steps:

1. Click the "Load configuration file" button, select the configuration file and click "OK".
2. Wait until a dialog pops up to prompt that the configuration file is loaded successfully. The loaded configuration file takes effect after the controller is restarted.



The loaded configuration file must meet the following criteria:

- It is best to use the configuration file that was backed up previously. It is not recommended to use a customized configuration file or a locally modified configuration file.
- If you use a custom configuration file, the file name must be "robot.cfg.bk" and the file must be a compressed package.
- If you use a customized configuration file, ensure that all parameters mentioned above are covered in the configuration file and that the robot model and controller model in the file do not conflict with the current controller.
- After the upgrade, restart the controller before loading the configuration file; otherwise, loading may fail.
- Ensure that the mechanical unit configuration and master bus configuration information match when loading the configuration file. If servo parameter read/write errors occur after loading, check whether the mechanical unit configuration, master bus configuration, and actual peripheral devices match. If they do not match, modify them to the correct configuration information and restart for normal use.

- Results:

After loading the configuration file and restarting the controller, you can check whether the configuration in the configuration file takes effect. For example, you can check whether the corresponding structural parameters are consistent with the parameters in the configuration file.

System upgrade

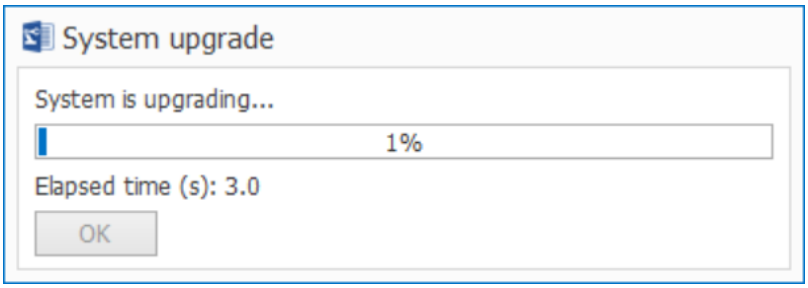
- Function:

It allows you to upgrade the robot controller with a software package from the local disk.

- Steps:

1. Check the items to select which versions to upgrade.
2. Click the browse button "..." to select the package on the local disk. Select the package that starts with "P-", which contains the process plug-ins.

3. After the selection is complete, click "Start" to proceed with the upgrade. If the upgrade includes plug-in packages, a plug-in selection dialog box will pop up. Select the plug-ins that need to be upgraded and continue with the upgrade.
4. After the upgrade is complete, restart the controller.



CAUTION

- The system upgrade is performed by using the controller package. The robot model and controller model remain unchanged. For package larger than 65 MB, you need to generate a dedicated package using the InoRobotUpdater before you can upgrade the system.
- Do not power off or disconnect the controller during system upgrade.
- After the system upgrade is complete, the text above the progress bar indicates whether the upgrade is successful.
- After restarting the system, check whether the controller version is consistent with the target version of the upgrade by going to About > Version.

- Results:

After the system is upgraded and restarted, you can check whether the controller version is the target version. For example, you can go to About > Version.

Restore factory settings

- Function:

It allows you to initialize all robot parameters to the factory defaults.

- Steps:

1. Click the "Restore factory settings" button and confirm it.
2. Wait until the progress bar is complete, that is, the default settings are restored.

- Results:

After you restore the default settings and restart the controller, verify whether the previous settings in the controller are cleared.

Format memory card

- Function:

It allows you to format the memory card on the controller into a program storage card suitable for use by the robot.

- Steps:

1. Check the status of the memory card. The memory card must be connected and successfully mounted.
2. Click the "Format memory card" button to start formatting.
3. Wait until the formatting is complete.



- Formatting the memory card clears all files in the memory card, resets the slave configuration, and deletes the mechanical unit configuration. However, the EtherCAT master configuration will not be affected.
- Some functions may affect the normal operation of the formatting function, such as digitization log, plug-in, system diagnostics, mechanical unit, EtherCAT slave enable management, etc. If the device is busy during formatting, check whether any of the above functions are in use.
- The memory card cannot be formatted during task operation, including static tasks. Stop the task before formatting.

- Results:

After the card is formatted, verify whether the files in the card are restored to the default state.

Load point file

- Function:

It allows you to load the point file into the robot controller.

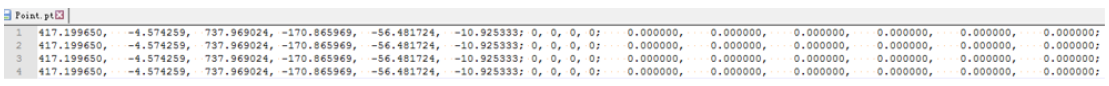
- Steps:

Click the "Load point file" button, browse the point file on the local disk, and then click "Load point file".

About point file:

- The point file uses ".pt" as a name suffix.
- The file contents are data information of position variables. One line indicates one piece of position variable information.
- For the format of every line, refer to the definition of position variable. Each line is divided into three subparagraphs. The first six parameters, which are robot coordinates, compose the first subparagraph. The middle four parameters, which are arm parameters, compose the second subparagraph. The last six parameters, which are external axis data, compose the third subparagraph.
- The subparagraphs are separated with ";", and numbers in a subparagraph are separated with ",".

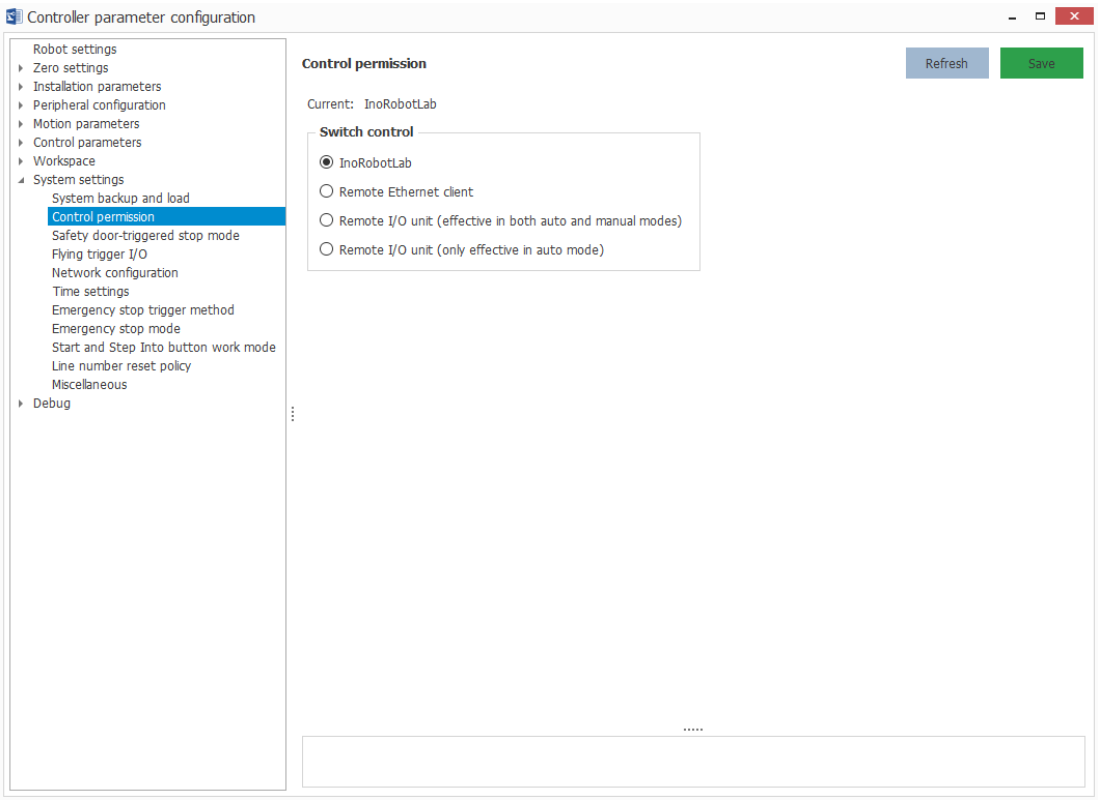
An example is shown in the figure below:


Point.pt

```
1  417.199650,   -4.574259,  737.969024, -170.865969,  -56.481724,  -10.925333; 0, 0, 0, 0;    0.000000,    0.000000,    0.000000,    0.000000,    0.000000,    0.000000;
2  417.199650,   -4.574259,  737.969024, -170.865969,  -56.481724,  -10.925333; 0, 0, 0, 0;    0.000000,    0.000000,    0.000000,    0.000000,    0.000000,    0.000000;
3  417.199650,   -4.574259,  737.969024, -170.865969,  -56.481724,  -10.925333; 0, 0, 0, 0;    0.000000,    0.000000,    0.000000,    0.000000,    0.000000,    0.000000;
4  417.199650,   -4.574259,  737.969024, -170.865969,  -56.481724,  -10.925333; 0, 0, 0, 0;    0.000000,    0.000000,    0.000000,    0.000000,    0.000000,    0.000000;
```

9.8.2 Control Permission

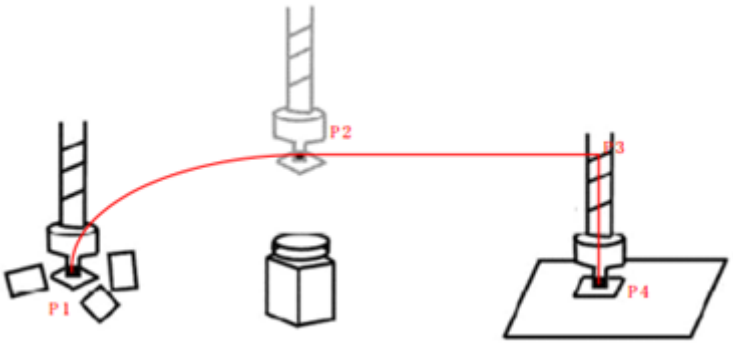
For details, see the 8.2 Setting the Control Permission section.



9.8.3 Flying Trigger I/O

The flying trigger function allows the robot to take photos and adjust the pose of the part accordingly without stop during movement from the part pickup point to the part placement point. This function reduces the cycle time.

As shown below, P1 is the part pickup point, P2 is the photo taking point, P3 is the transition point, and P4 is the part placement point. Normally, the robot needs to pause at P1, P2, P3 and P4 respectively. With the flying trigger function enabled, the robot does not stop at P2. Instead, when the robot is predicted to reach P2 during the movement from P1 to P3, the controller controls the digital output (DO) to trigger the camera to take photos and the servo to latch the position. Then the controller determines the position and orientation at which the part will be placed at P4 based on the vision processing results and the latched servo position.



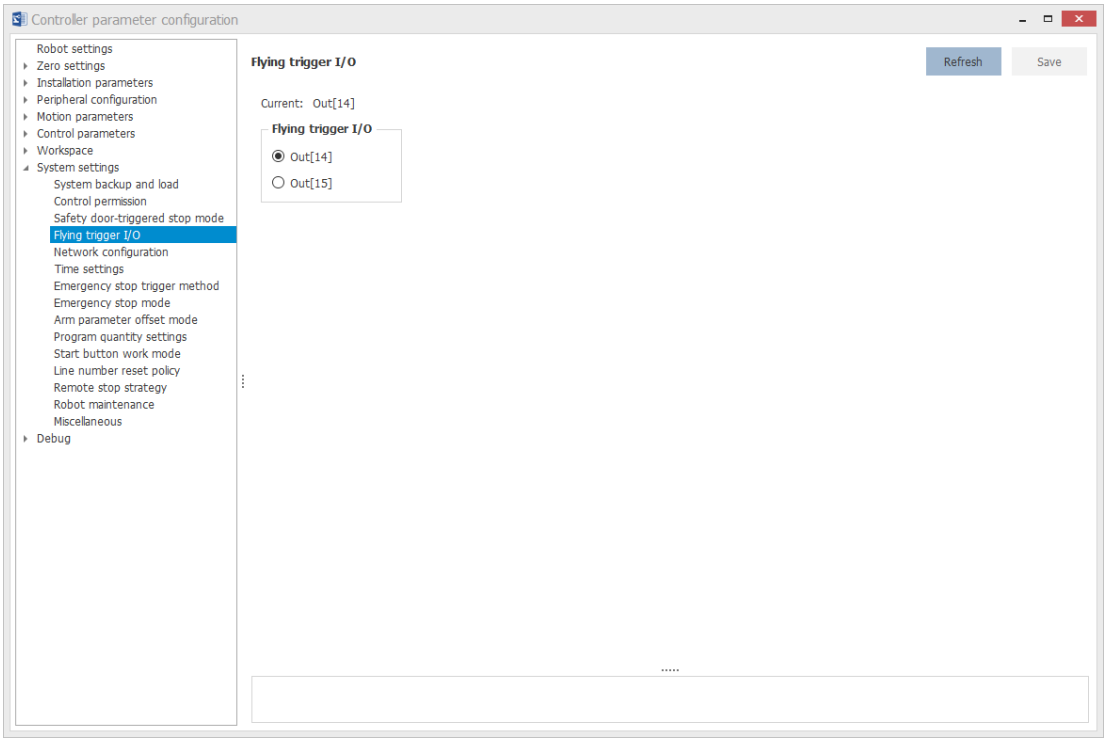
Flying trigger

The flying trigger is not a stand-alone function. It is implemented by the combination of three functions including position latching, I/O in motion, and motion without wait.

For programming examples of the flying trigger function, see the Programming Guide. The trigger signals that can be used include Out[14] and Out[15].

Steps

1. Double-click "Controller parameter configuration" in the device tree, then select "System settings" > "Flying trigger I/O" in the opened interface for related settings, as shown in the figure below:



2. The settings take effect immediately after being saved.

Results

You can check whether the settings taken effect by performing the flying trigger function.

NOTE

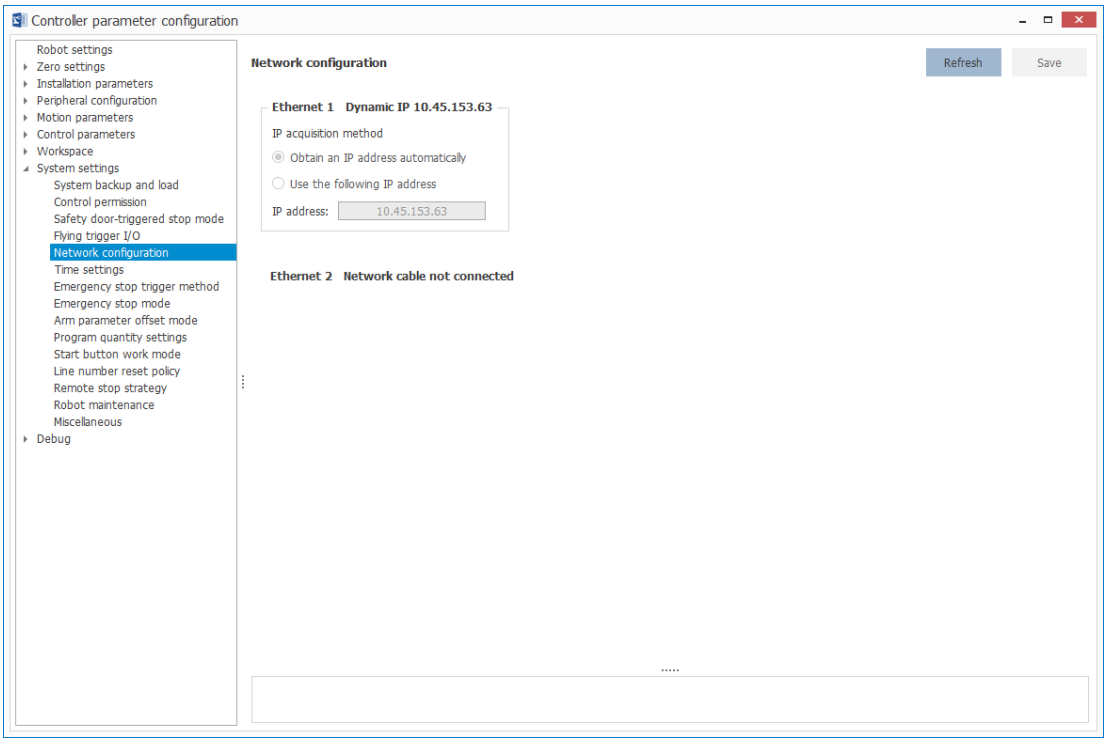
- The IRCB501, RCB500 and IRCB300 series controllers support the position latching function. You only need to connect the corresponding I/O to the camera as a trigger for taking photos.
- The IRCB10 series controller does not support the flying trigger function.
- The position latching of IRCB300 and IRCB101 series controllers is triggered by one signal (fixed to "Out[15]" for IRCB300 and "Out[14]" for IRCB101). To prevent the position latching function from being unavailable due to damage, the IRCB500 and IRCB501 series controllers support position latching triggered by 2 signals: Out[14] or [Out15], and the default is Out[14].

9.8.4 Network Configuration

It allows you to configure the Ethernet ports. The Ethernet port 1 can obtain an IP address automatically or use a specified IP address. The Ethernet port 2 can only use a fixed IP address 192.168.23.25.

Steps

1. Double-click "Controller parameter configuration" in the device tree, then select "System settings" > "Network configuration" in the opened interface for related settings, as shown in the figure below:



2. Configure the Ethernet port 1 and save the settings.

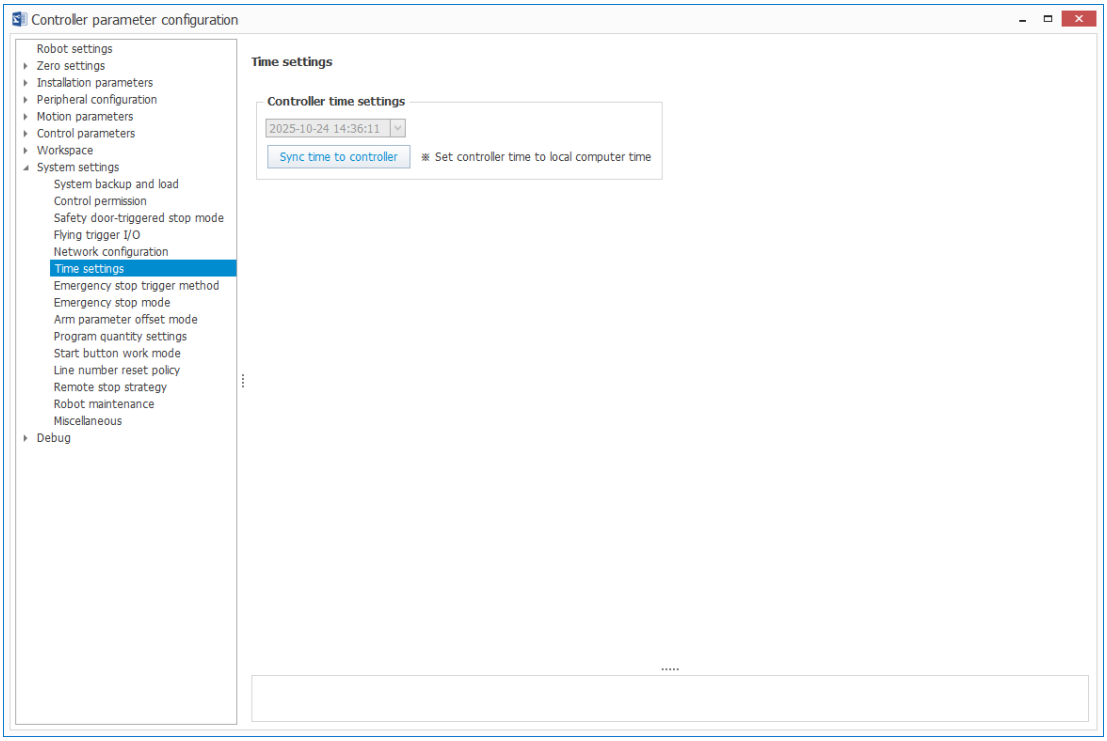
Results:

You can check whether the IP setting of Ethernet port 1 is valid by pinging this IP address from another device.

9.8.5 Time Settings

Steps

1. Double-click "Controller parameter configuration" in the device tree, then select "System settings" > "Time settings" in the opened interface for related settings, as shown in the figure below:



2. Click the "Sync time to controller" button.

Results

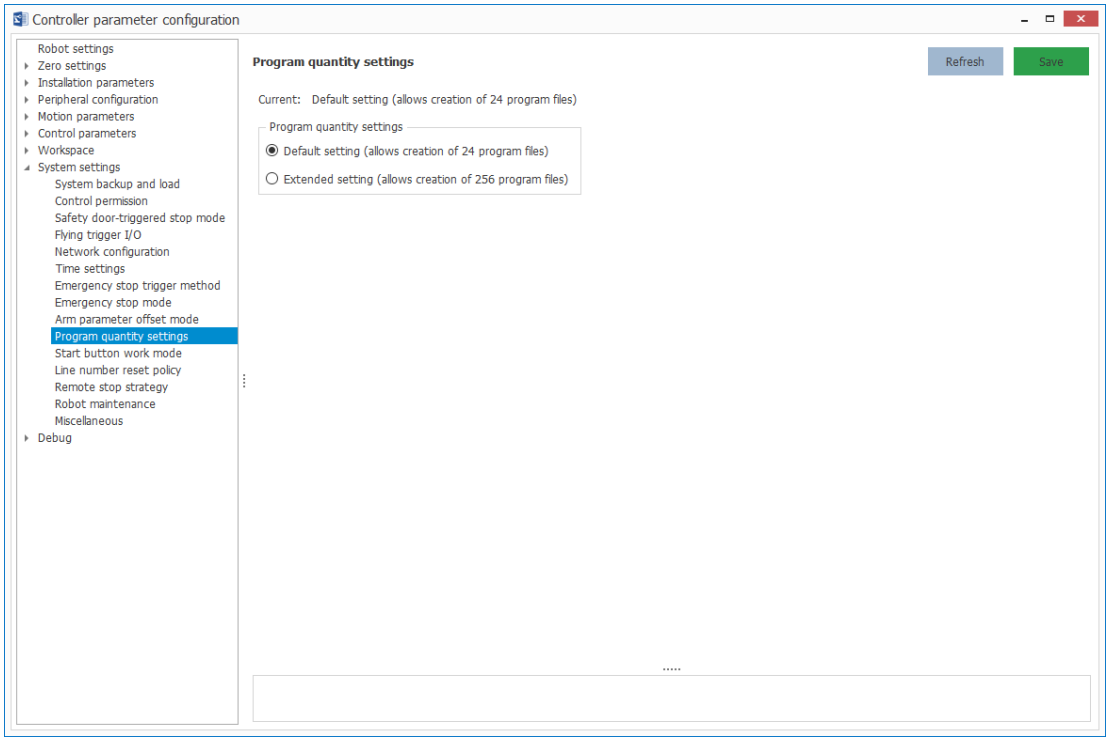
Check whether the time displayed on the interface is the same with the system time of PC where InoRobotLab is installed.



When the control permission is not granted to InoRobotLab, the "Sync time to controller" button is not operable.

9.8.6 Program Quantity Settings

This function extends the number of program files available for use.



Currently, only two settings are supported:

- Default setting (allows creation of 24 program files)
- Extended setting (allows creation of 256 program files)

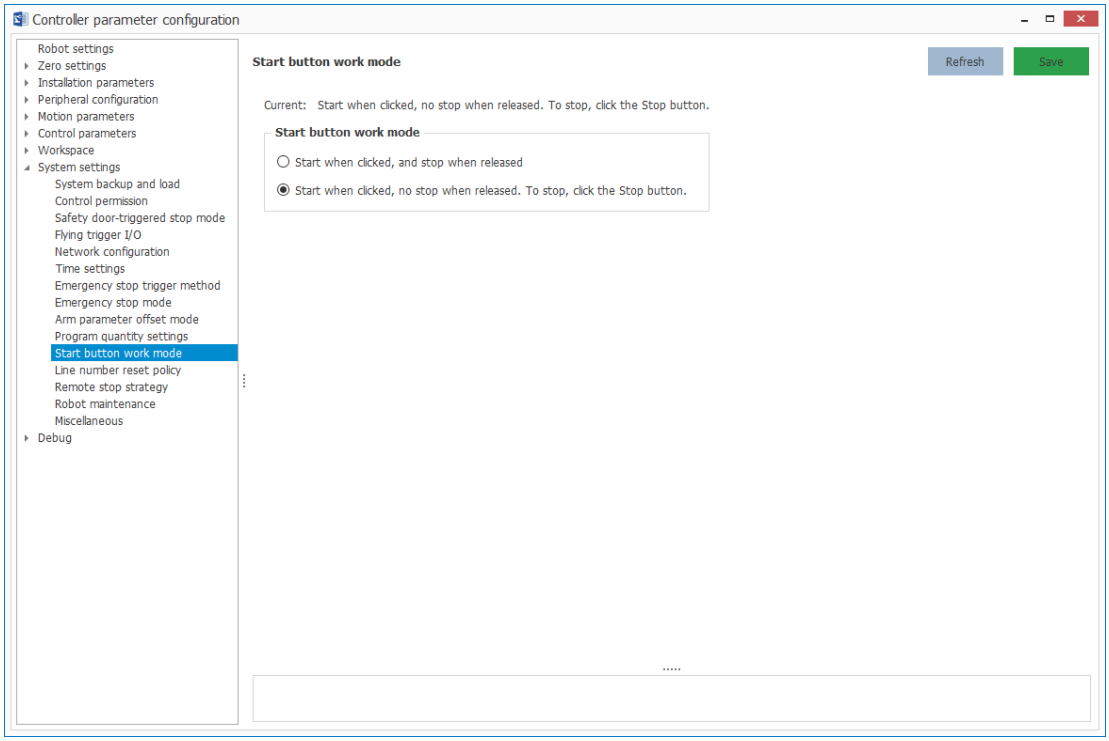


When using the extended setting, add program files with caution. An excessive number of program files or an overly large project may cause issues during loading, compiling, or synchronization.

9.8.7 Start Button Work Mode

It allows you to configure the work mode of the "Start" button and "Step into" button in the debug mode.

Go to Controller parameter configuration > System settings > Start button work mode.



Two modes are available:

- Start when clicked, stop when released
- Start when clicked, no stop when released. To stop, click the Stop button.

> **CAUTION**
>
> By default, mode 2 is selected.

Mode description

- Mode 1 (Start when pressed and stop when released):


Start : Ensure that the program is edited and switched to the debugging mode. The program starts running when this button is clicked, and stops running when this button is released. If you want to keep the program running to the end, keep the "Start" button held.


Step into : Ensure that the program is edited and switched to the debugging mode. The program starts running from the cursor line when this button is clicked, and stops running when this button is released. If you want to keep the current line running to the end, keep the "Step into" button held.

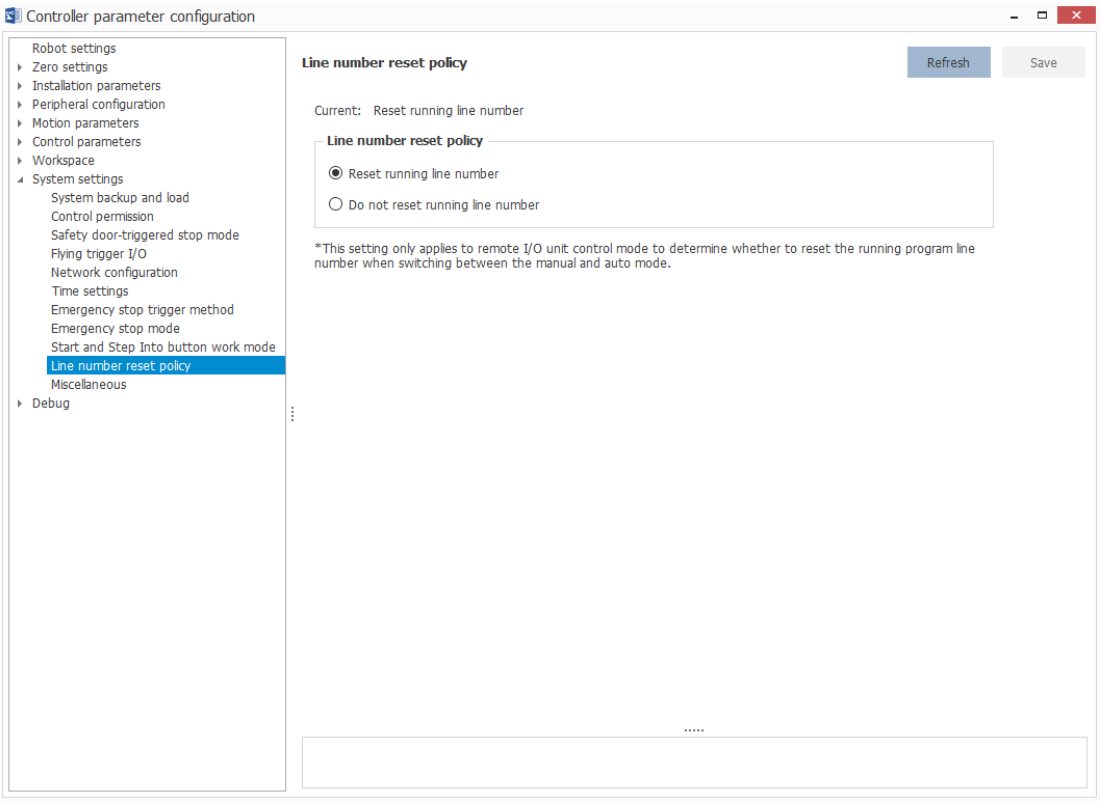
- Mode 2: (Start when clicked, no stop when released. To stop, click the Stop button.)


Start : Ensure that the program is edited and switched to the debugging mode. The program starts running when this button is clicked and keeps running to the end of the program unless the "Stop" button is clicked.


Step into : Ensure that the program is edited and switched to the debugging mode. The program starts running from the cursor line when this button is clicked, and keeps running to the end of the line unless the "Stop" button is clicked.

9.8.8 Line Number Reset Policy

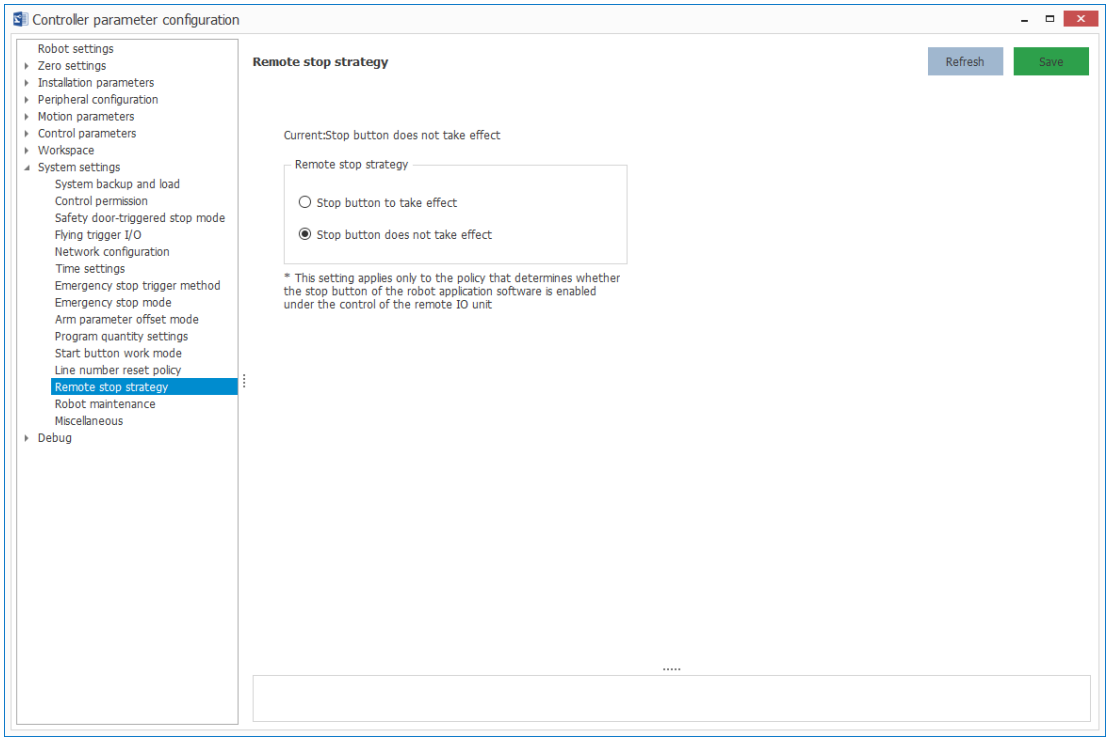
It allows you to configure whether to reset the running program line number when a switchover between manual and auto modes occurs under the control of the remote I/O unit.



- Reset running line number: When a switchover between manual and auto modes occurs under the control of the remote I/O unit, the program running line number will be reset.
- Do not reset running line number: When a switchover between manual and auto modes occurs under the control of the remote I/O unit, the program running line number will not be reset.

9.8.9 Remote Stop Strategy

It allows you to configure whether the stop button of the robot application software is enabled when a switchover between manual and auto modes occurs, under the control of the remote I/O unit (when effective in both auto and manual modes, or only effective in auto mode).



- Stop button to take effect: When the stop button of the robot application software is pressed under the control of the remote I/O unit, the running will be stopped.
- Stop button does not take effect: When the stop button robot application software is grayed out under the control of the remote I/O unit, the running can only be stopped through the remote control end.

9.8.10 Robot Maintenance

This function is used to display the lifespan of partially worn components of the robot and related maintenance information, reminding you to perform maintenance operations on the robot. This function monitors the usage of each component. When the component lifespan reaches the set lifespan threshold, a warning will be triggered to remind you to perform related maintenance operations. This warning can be cleared. After completing the maintenance settings operation, no warning will be generated again within the lifespan threshold.

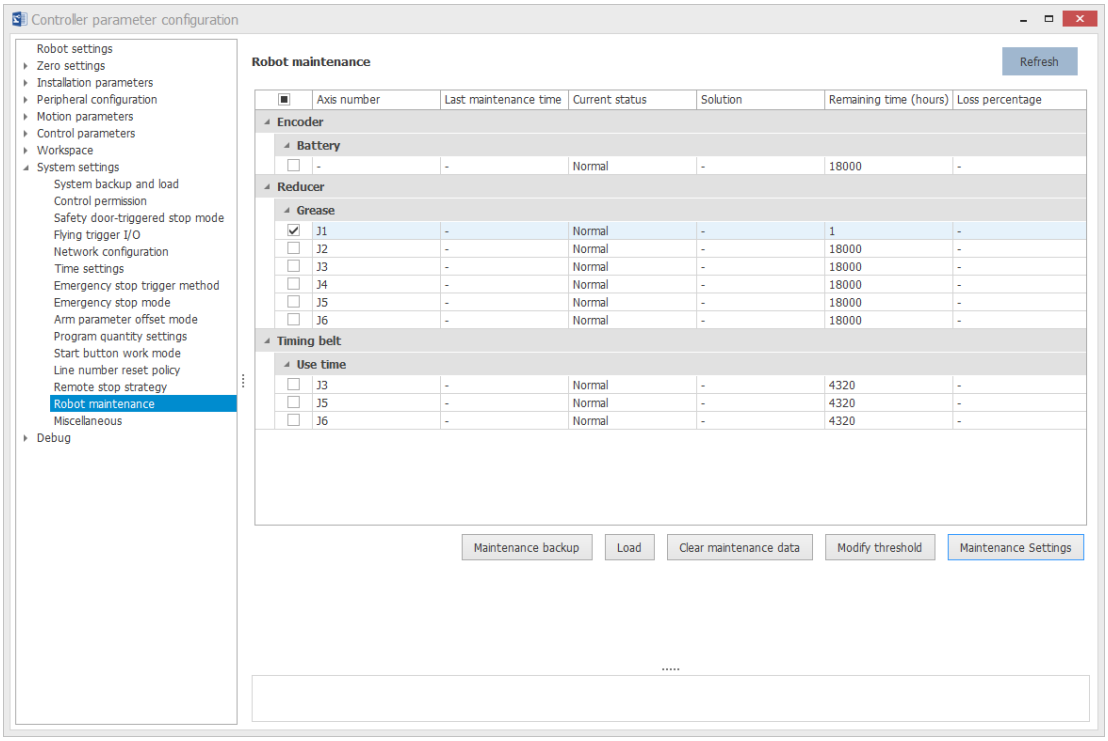
You can access this function by navigating to "Controller parameter configuration" > "System settings" > "Robot maintenance". The function main interface displays all currently maintainable component types, with each row in the table corresponding to one maintenance component. The component types are displayed according to the models. The table includes: Last maintenance time, current status, solution, remaining time (hours), and loss percentage.

- If no maintenance has been performed, the value of "Last maintenance time" is displayed as "1".
- If the "Current status" is "Normal", the value of "Solution" is displayed as "1". If the "Current status" is "Pending maintenance", the value of "Solution" is displayed as the corresponding method.

The maintenance data for each component is described as follows:

Maintenance component	Description
Encoder battery usage time	The power-off time of the controller is the usage time of the encoder battery.
Reducer grease usage time	The power-on time of the controller
Conveyor usage time	The power-on time of the controller

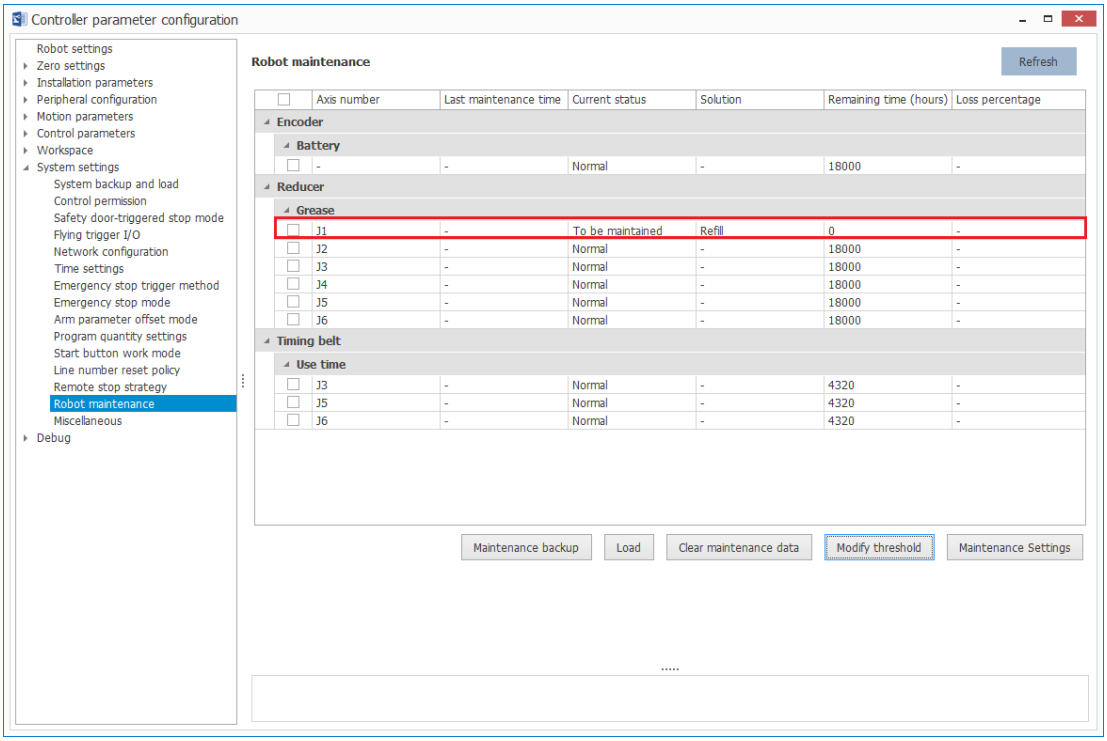
Maintenance component	Description
Ball screw grease usage time	The power-on time of the 4-axis robot controller
Ball screw wear ratio	The ratio of the ball screw motion distance to the maximum stroke
Balance cylinder	The ratio of the usage count to the maximum allowable count for balance cylinder spring

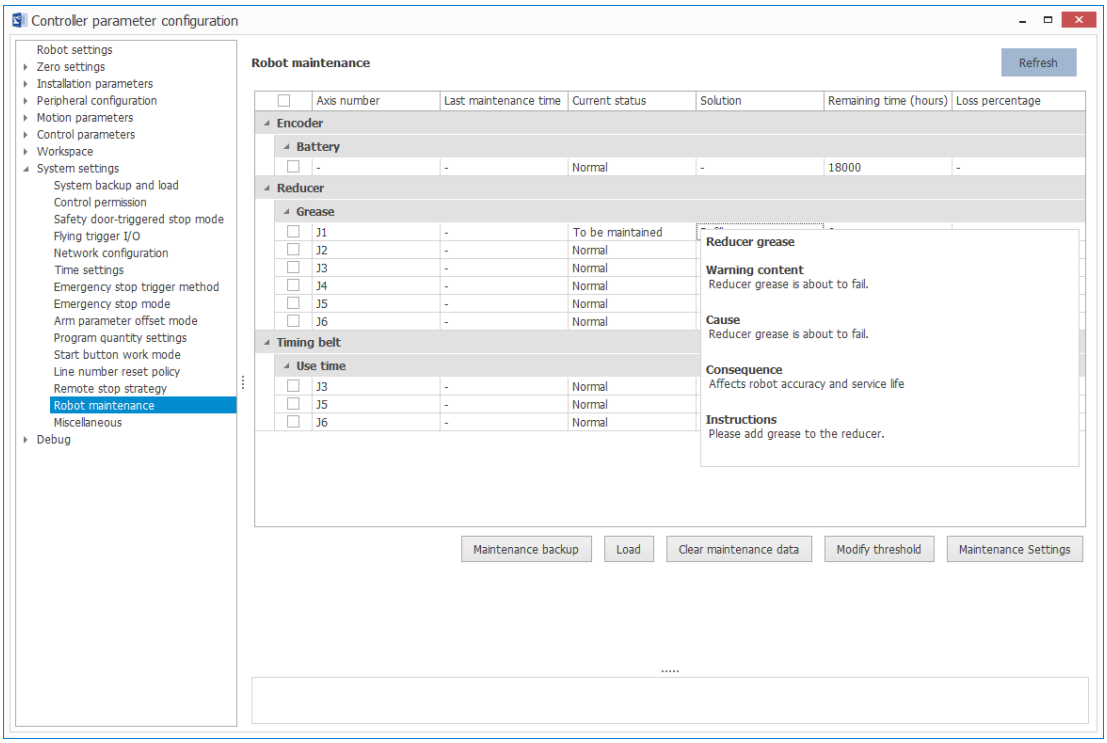


Steps

1. View maintenance handling method

   Double-click a certain component in the table. If the current status of the component is "To be maintained", a small pop-up window will appear (as shown in the figure below), displaying detailed maintenance method information of the component. Same components on different axes can be selected simultaneously for maintenance, but different components cannot be selected simultaneously.





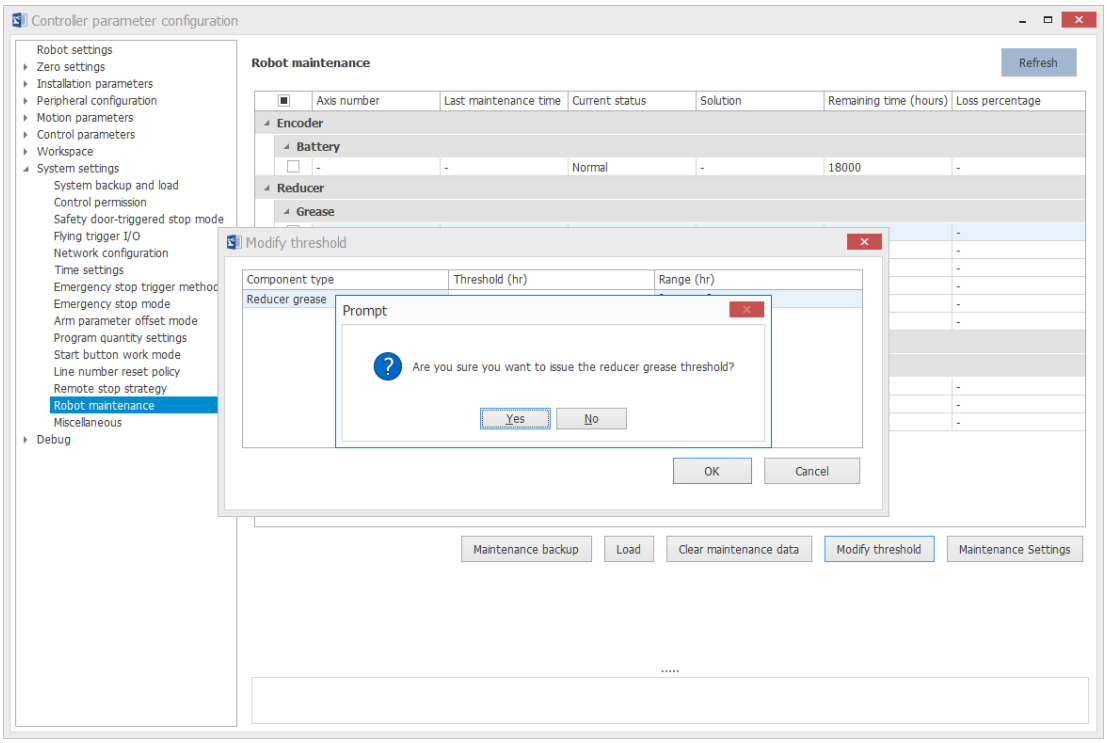
> **NOTE**
>
> If maintenance operations on the relevant components are not performed after a warning is generated, the warning will be triggered again after one hour.

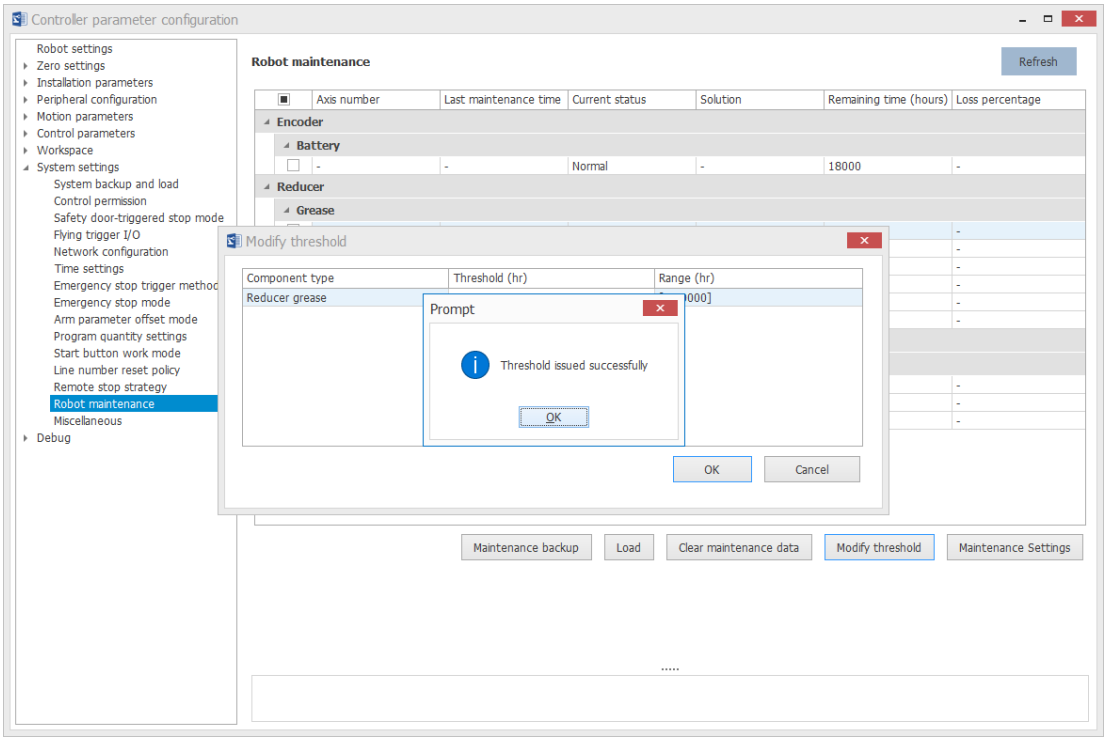
2. Modify the lifespan threshold.

   After the administrator performs maintenance checks and finds that the equipment usage does not yet require maintenance, they can use this function to set a new lifespan threshold. The "Modify threshold" button is not displayed under User mode.

   Be sure to operate in Admin or Manufacturer mode, and in the non-motion state under manual mode: Select the maintenance component, click the "Modify threshold" button to open the threshold

modification interface. The button is only displayed under Admin and Manufacturer modes. The interface includes the threshold modification related settings. Enter the parameter, and click "OK" to send the data. There will be a second confirmation prompt, and a success dialog will pop up after successful modification.



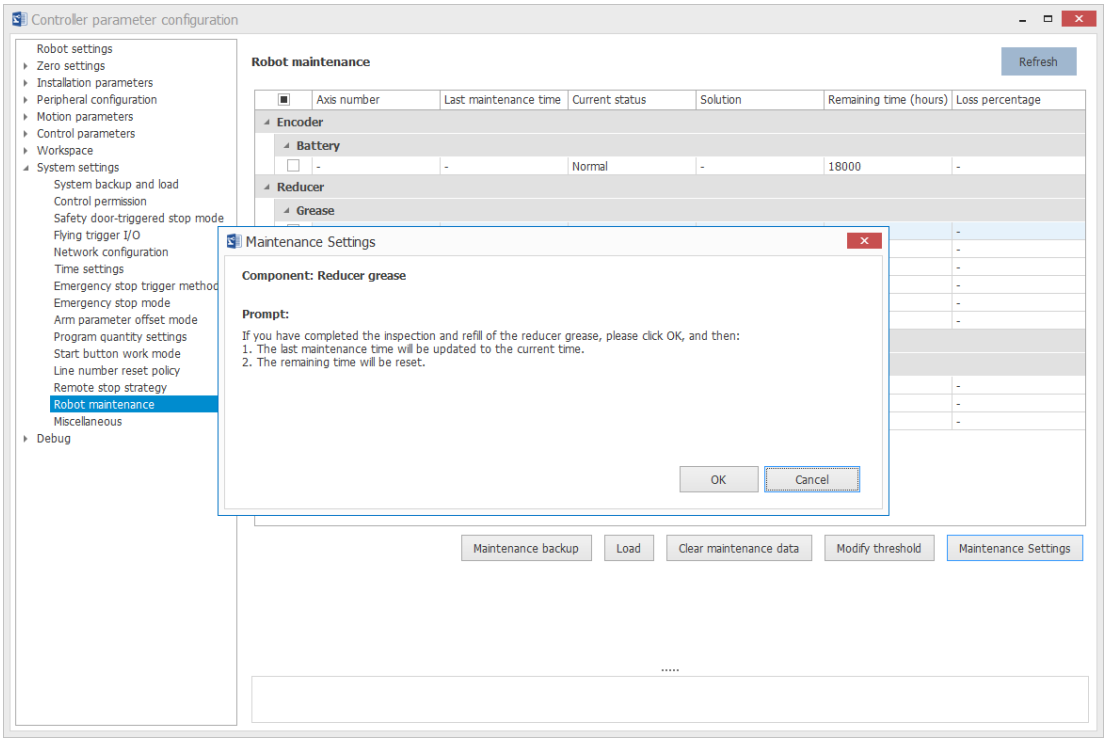


3. Perform the maintenance settings.

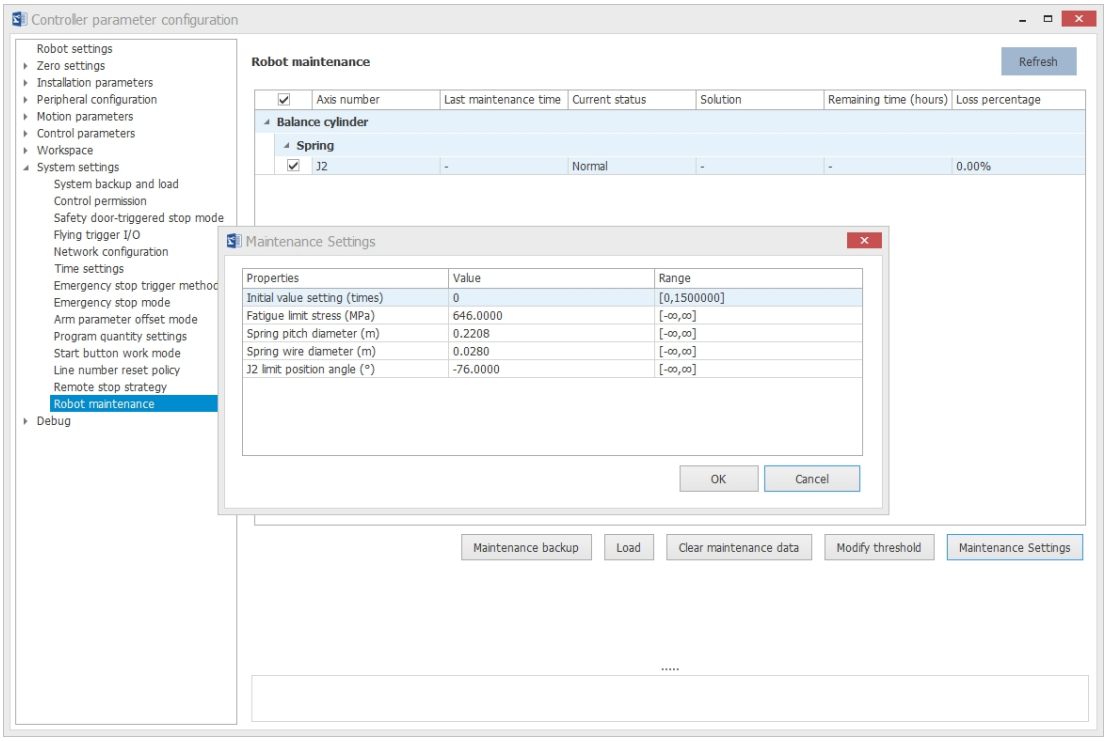
After completing the maintenance of the robot components, you need to perform maintenance settings to reset the maintenance function of the robot components. In the User mode, the "Maintenance Settings" button is not displayed.

Be sure to operate in Admin or Manufacturer mode, and in the emergency stop mode and the non-motion state under manual mode: Select the maintenance component, click the "Maintenance Settings" button. The button is only displayed under Admin and Manufacturer modes. Maintenance settings related operation information will be displayed. Click "OK", and an operation confirmation prompt will be popped

up. After confirmation, the component maintenance data will be reset to the threshold set (or to the default value if not set), and the last maintenance time will be recorded as the current time.



For the balance cylinder component, if a spring with parameters different from the original factory parameters needs to be replaced, click the "Modify initial properties" button in Manufacturer mode to display the interface for setting the initial parameters of the spring and other parameters, and enter the correct parameters. Click "OK" or "X" to return to the "Maintenance Settings" interface. If you click "OK", a secondary confirmation prompt will be displayed. After confirmation, the current usage count will be reset to the "Initial value" parameter in the "Modify initial properties" interface, and the last maintenance time will be recorded as the current time.

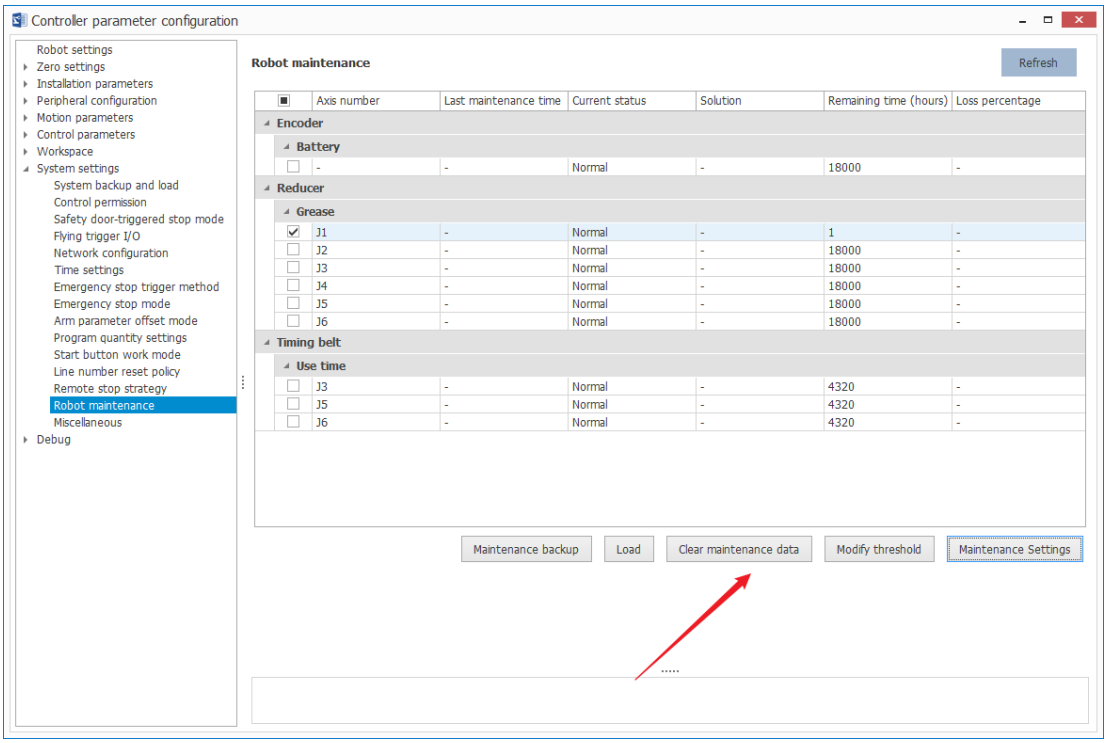


4. Clear the maintenance data.

   To reset the pre-maintenance function, you need to clear the maintenance data and settings. In the User mode, the "Clear the maintenance data" button is not displayed.

Be sure to operate in Admin or Manufacturer mode, and in the emergency stop mode and the non-motion state under manual mode: Select the maintenance component, click the "Clear maintenance data" button. The button is only displayed under Admin and Manufacturer modes. Then click "OK" to reset all maintenance data to default values.

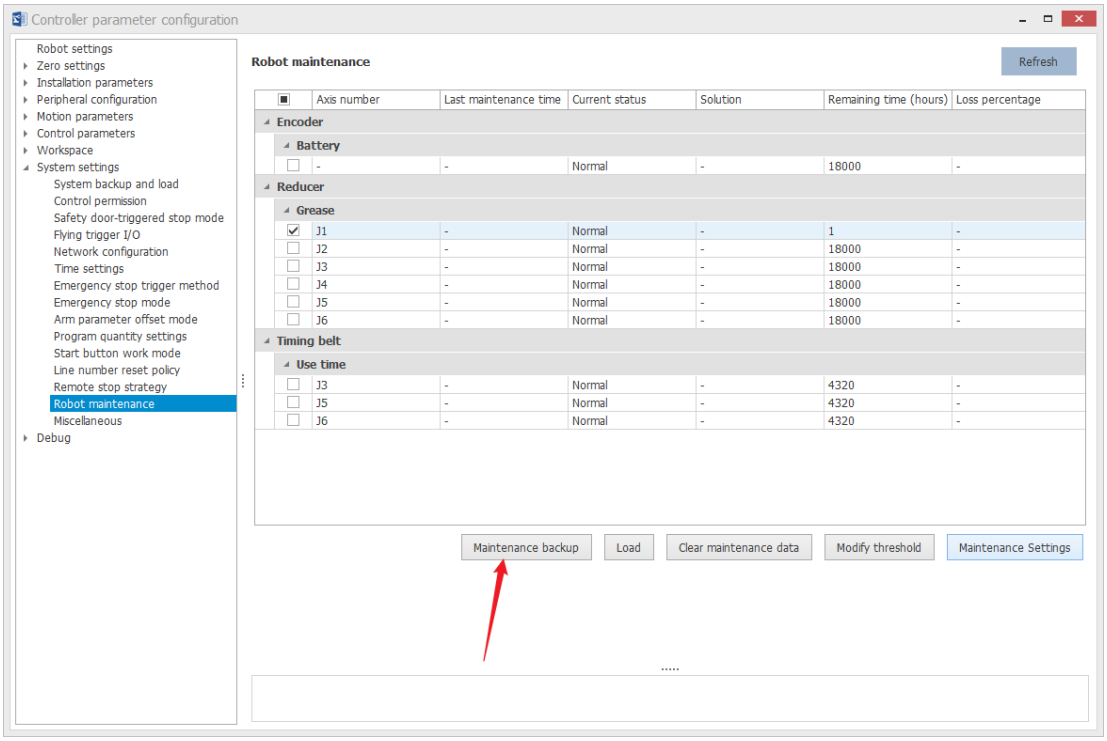
> **CAUTION**
>
> If the controller is flashed to a different model, alarm 0x1058 will occur. In the Admin or Manufacturer mode, click "Clear the maintenance data" to prevent alarm 0x1058 from recurring after each reboot.



5. Back up the maintenance data.

   To back up maintenance data, perform a "Maintenance backup". In the User mode, the "Maintenance backup" button is not displayed.

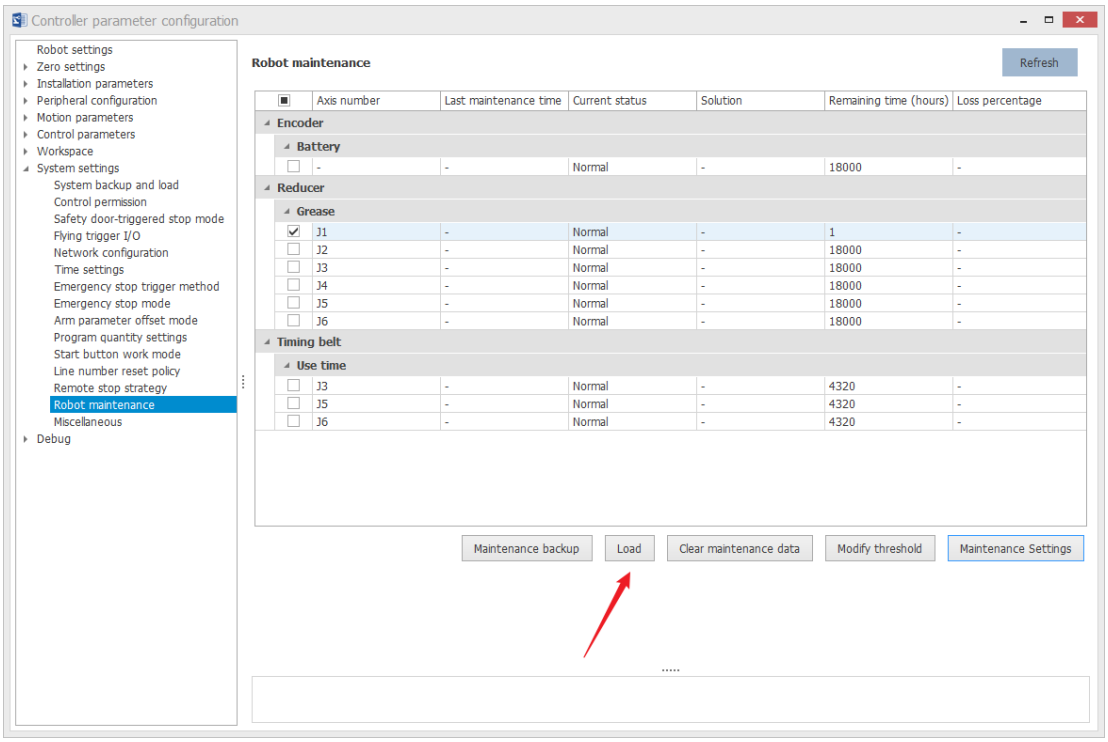
   Be sure to operate in Admin or Manufacturer mode, and in the non-motion state under manual mode: Click the "Maintenance backup" button. Then click "Yes" to proceed.



6. Load the maintenance data.

   To load the maintenance data, click the "Load" button. In the User mode, the "Load" button is not displayed.

   Be sure to operate in Admin or Manufacturer mode, and in the non-motion state under manual mode: Click the "Load" button. Then click "Yes" to proceed.



9.8.11 Miscellaneous

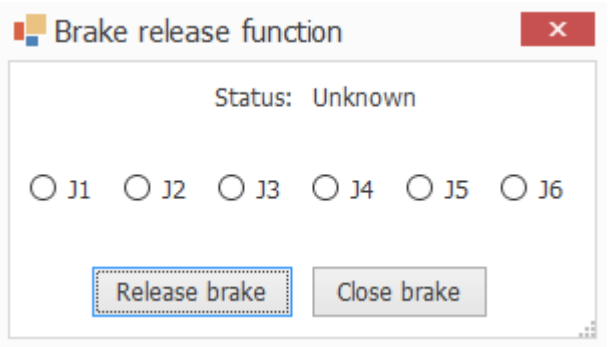
Brake release function

The brake release function can be used to engage or release the brake for one of the J1-J6 axes. This function is available when the following conditions are met:

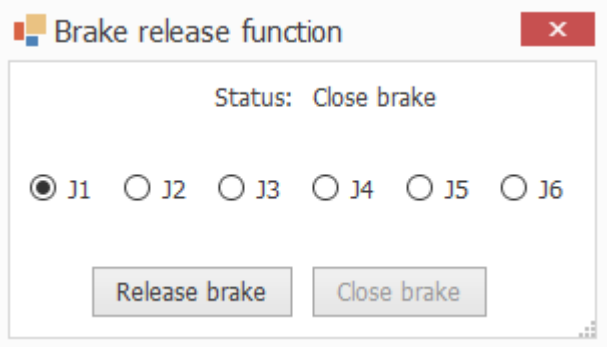
- The control permission is granted to the InoRobotLab software.
- The mode is Admin or Manufacturer.
- The robot is in emergency stop state.
- The control model is a 6-axis robot.

Steps

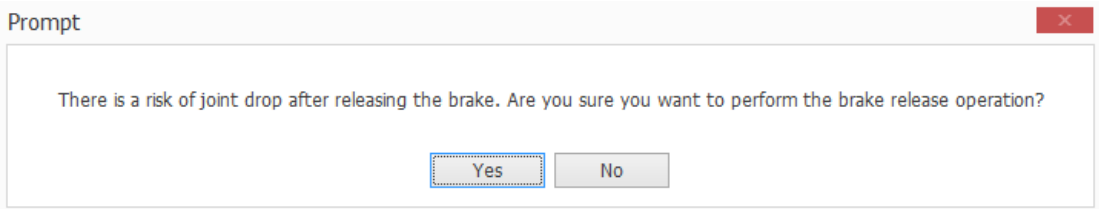
1. Click the "Brake release function" button. The following dialog appears.



2. Select the axis.
   Select one of the J1~J6 axes. For example, when you select J1, you can see the current state of J1, as shown below.



3. Release the brake.
   After you click the "Release brake" button, a prompt pops up. Click "Yes" to confirm. The brake of the selected axis is released.



4. Close the function.
   When you click the "X" button to close the brake release function, the brake of the selected brake is engaged.

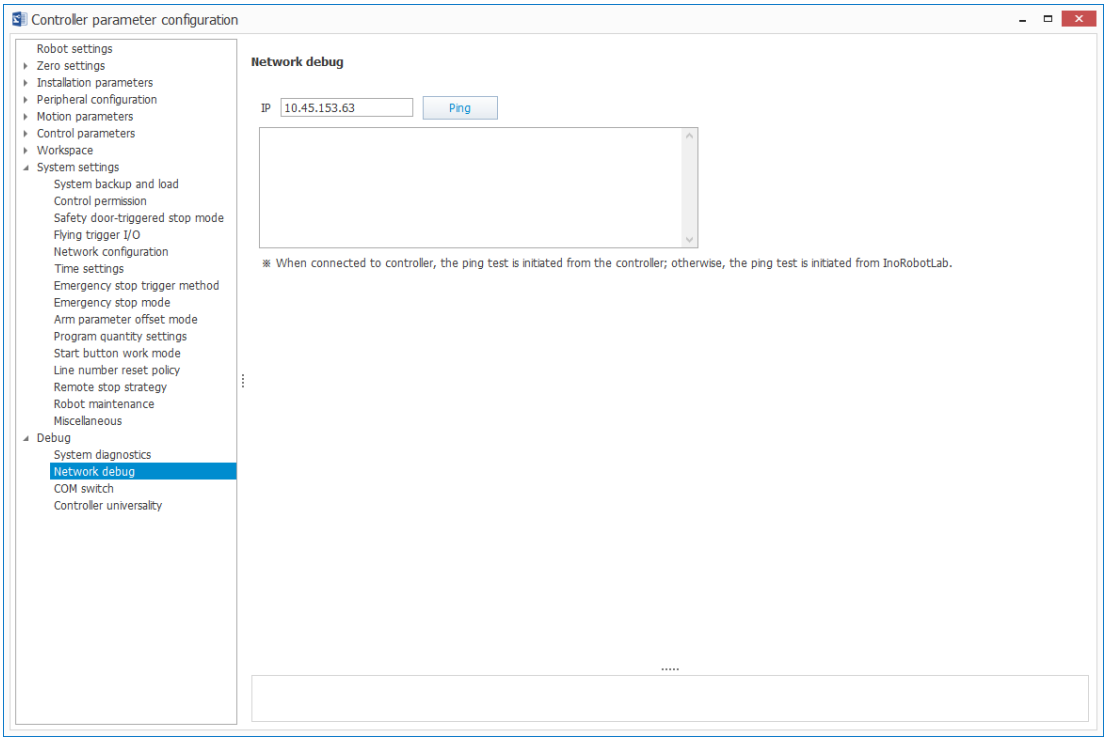
9.9 Debug

9.9.1 Network Debug

It allows you to debug communication of InoRobotLab or controller with other devices.

Steps

1. Double-click "Controller parameter configuration" in the device tree, then select "Debug" > "Network debug" in the opened interface for related settings, as shown in the figure below:



2. Click the "Ping" button.
   - When InoRobotLab is connected to the controller, the controller acts as a client to ping the target IP to detect the communication status of the controller with other devices.
   - When InoRobotLab is not connected to the controller, InoRobotLab acts as a client to ping the target IP to detect the communication status of InoRobotLab with other devices.

Results

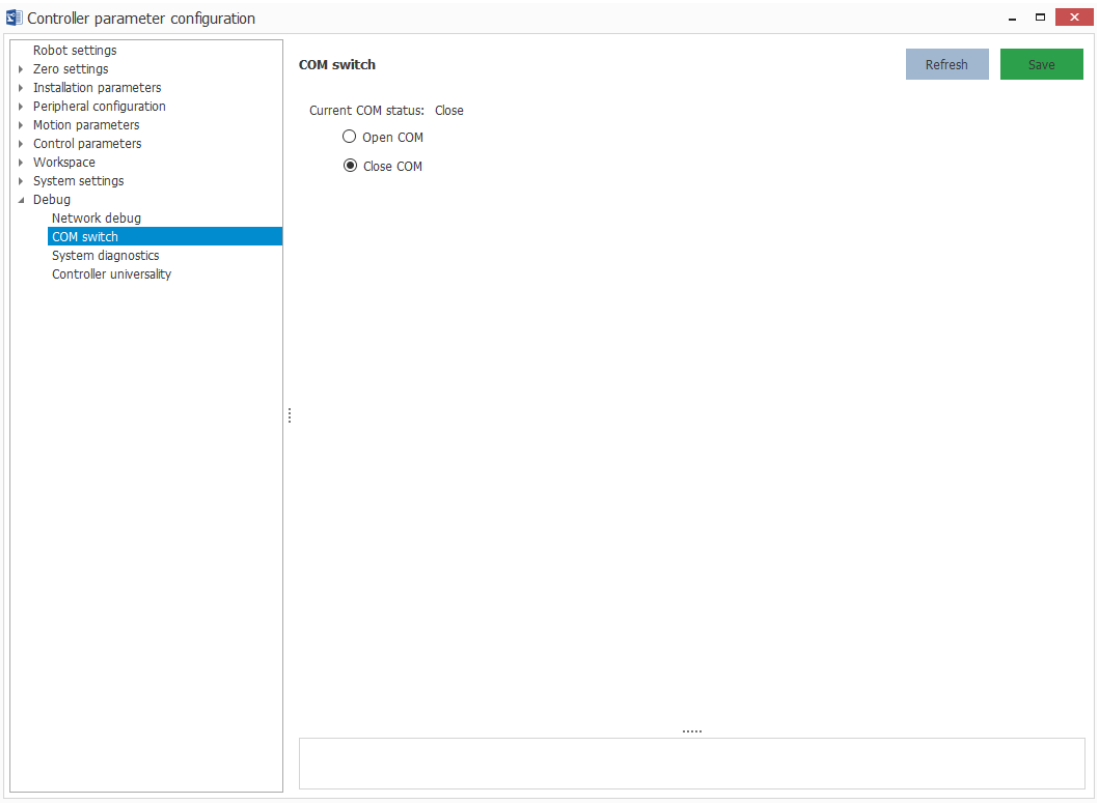
The statistics including the number of sent and accepted packets, packet loss rate, and packet round-trip time statistics of the ping are displayed.

9.9.2 COM Switch

You need to open the serial port before using the serial port-related instructions.

Steps

1. Double-click "Controller parameter configuration" in the device tree, then select "Debug" > "COM switch" in the opened interface for related settings, as shown in the figure below:



2. Open or close the serial port as needed. Save the settings and restart the controller to make the settings take effect.

Results

After the system is restarted, connect to the controller through the serial port.

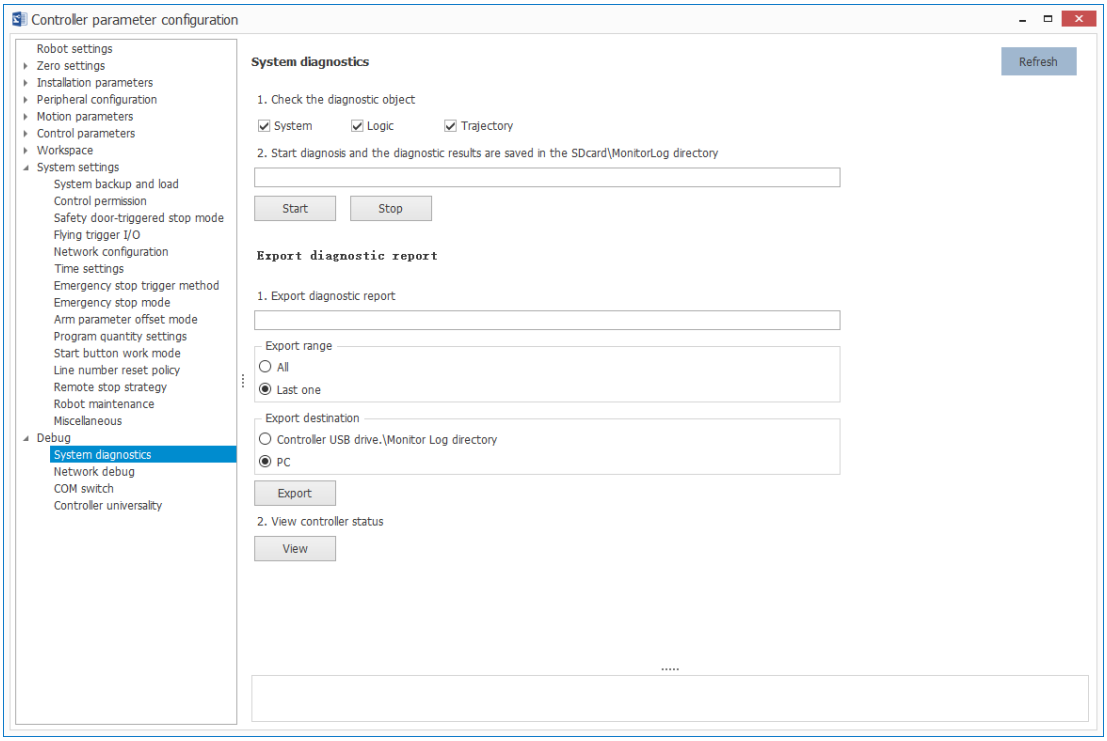
9.9.3 System Diagnostics

The controller diagnoses the current system operating state. The diagnosis objects include the system configuration parameters, state of logic variables, and robot motion trajectory.

9.9.3.1 Starting System Diagnostics and Exporting Diagnostic Results

Steps

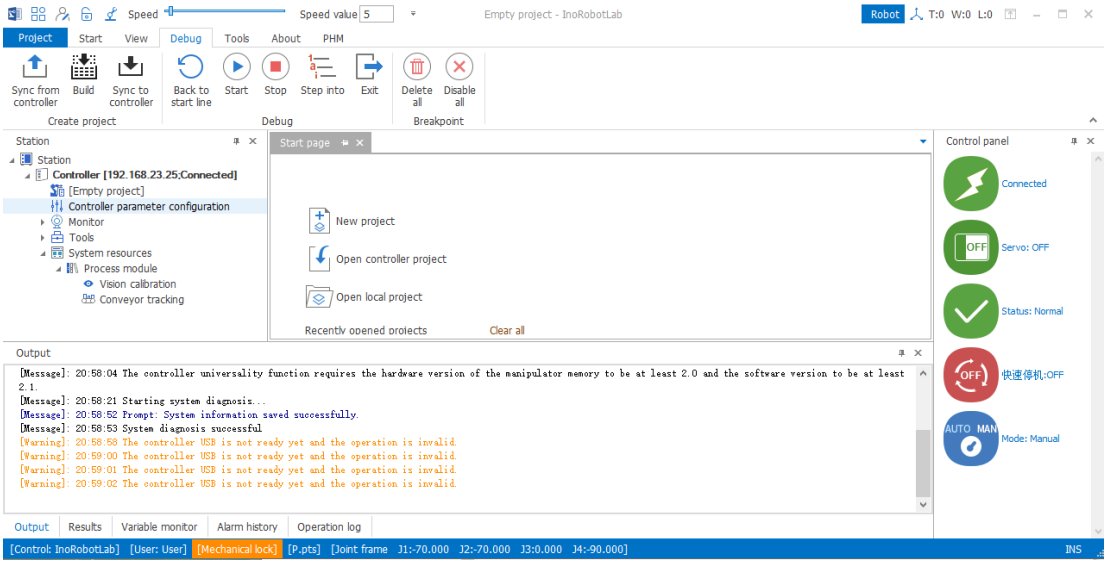
1. Double-click "Controller parameter configuration" in the device tree, then select "Debug" > "System diagnostics" in the opened interface for related settings, as shown in the figure below:



2. Check the object to be diagnosed, and then click "Start". The diagnostic results are saved in the controller file.
3. After the diagnosis is complete, select the export range and click the "Export" button to export the diagnostic file to the PC or USB drive on the controller.

Results

- After the diagnosis is complete, the diagnostic file is generated inside the controller. If you click "Stop" during the diagnosis, the diagnostic file inside the controller is still the last saved one.
- After the export is complete, you can view the export results in the save path on the PC or in the "MonitorLog" folder on the USB drive.
- If you set the export range to USB drive of the controller which does not exist, an error occurs.



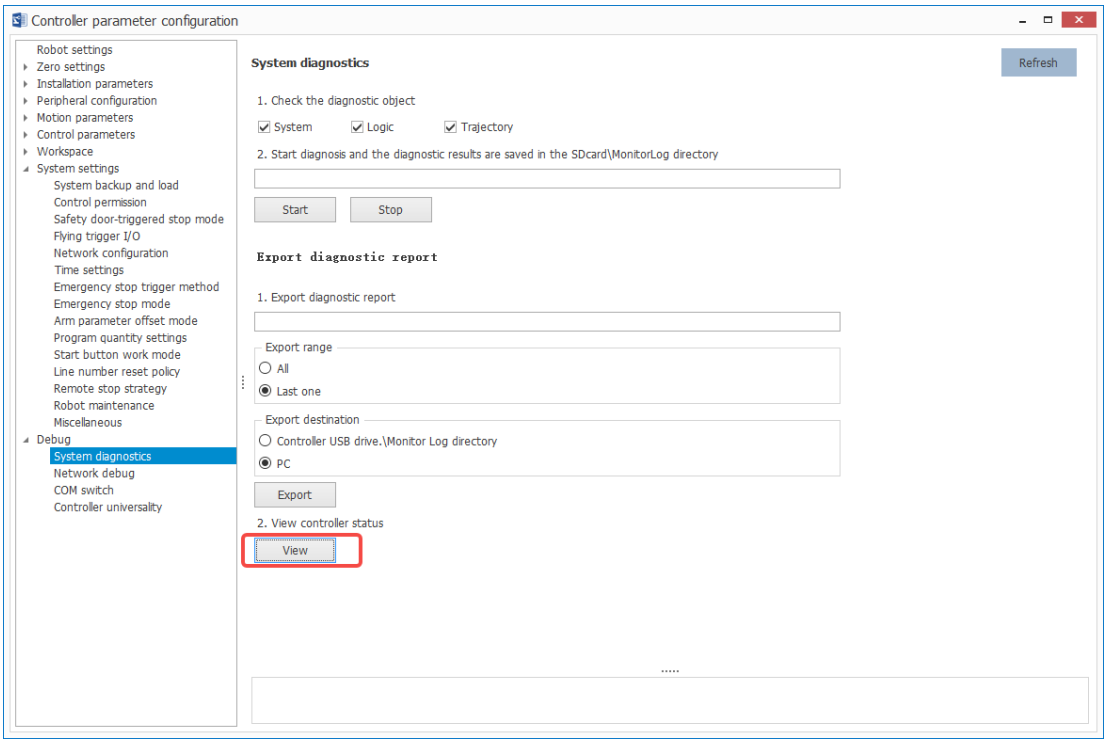
NOTE

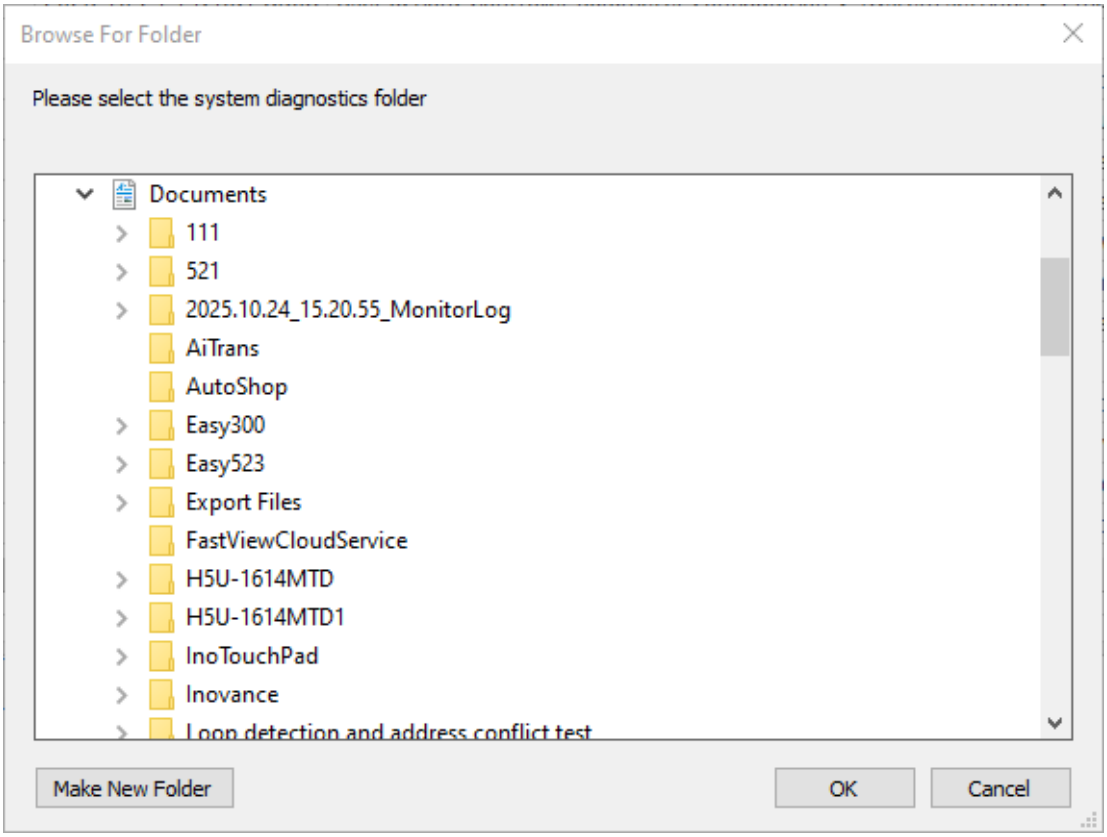
- If an error occurs during diagnosis, you can click "Clear alarm" to attempt to clear the alarm and continue.
- Multiple methods are supported to trigger the system diagnostic save operation, including instructions, fieldbus, and API.
- The "Refresh" button updates the parameter selection status for the current save operation.

9.9.3.2 Viewing Controller Status

Steps

1. Click the "View" button. A dialog pops up where you can select the diagnostic file to be opened. If you have exported the diagnostic file before, the software directly opens the exported diagnostic file previously exported.

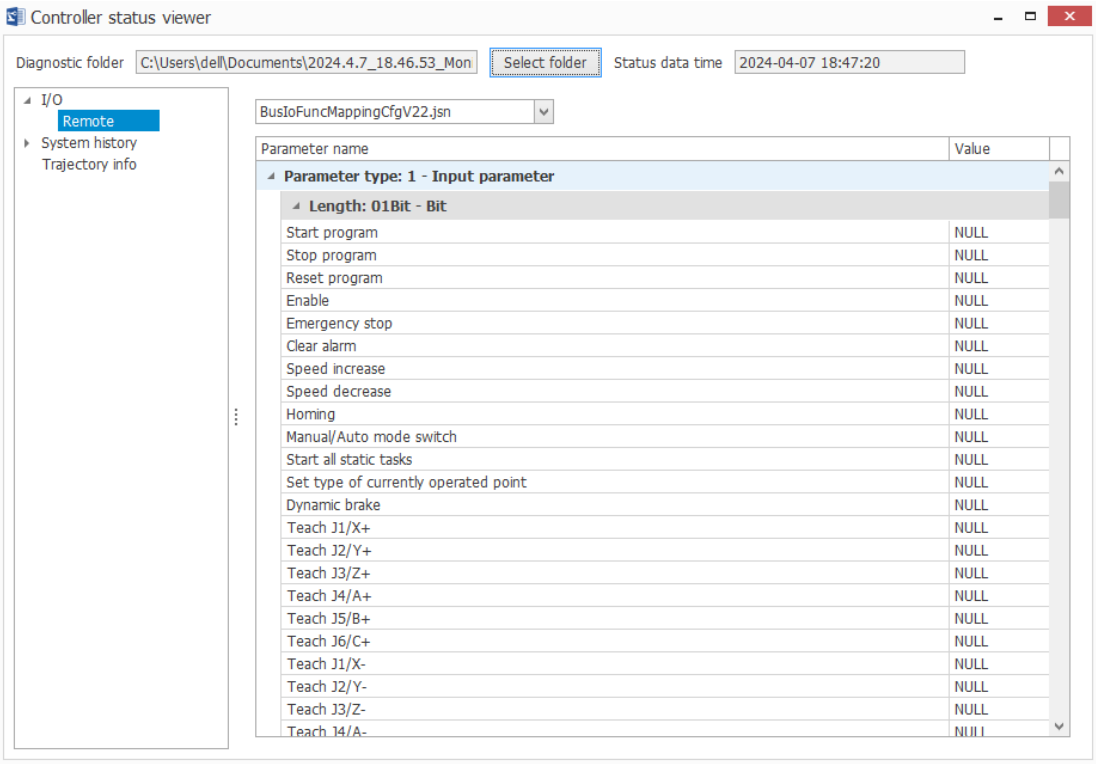


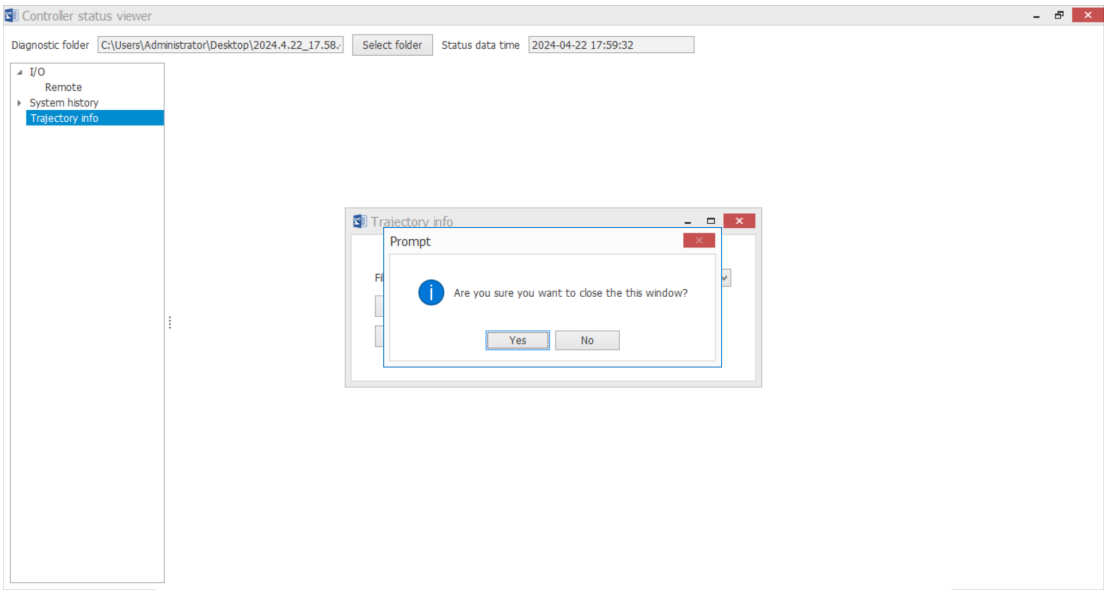


2. After the diagnostic file is opened, you can view the parsed data from the diagnostic file. At the top of the pop-up window, the current diagnostic folder path and the data of modification of the diagnostic folder are displayed. You can change the folder by clicking the "Select folder" button. After the folder is changed, the interface refreshes automatically.

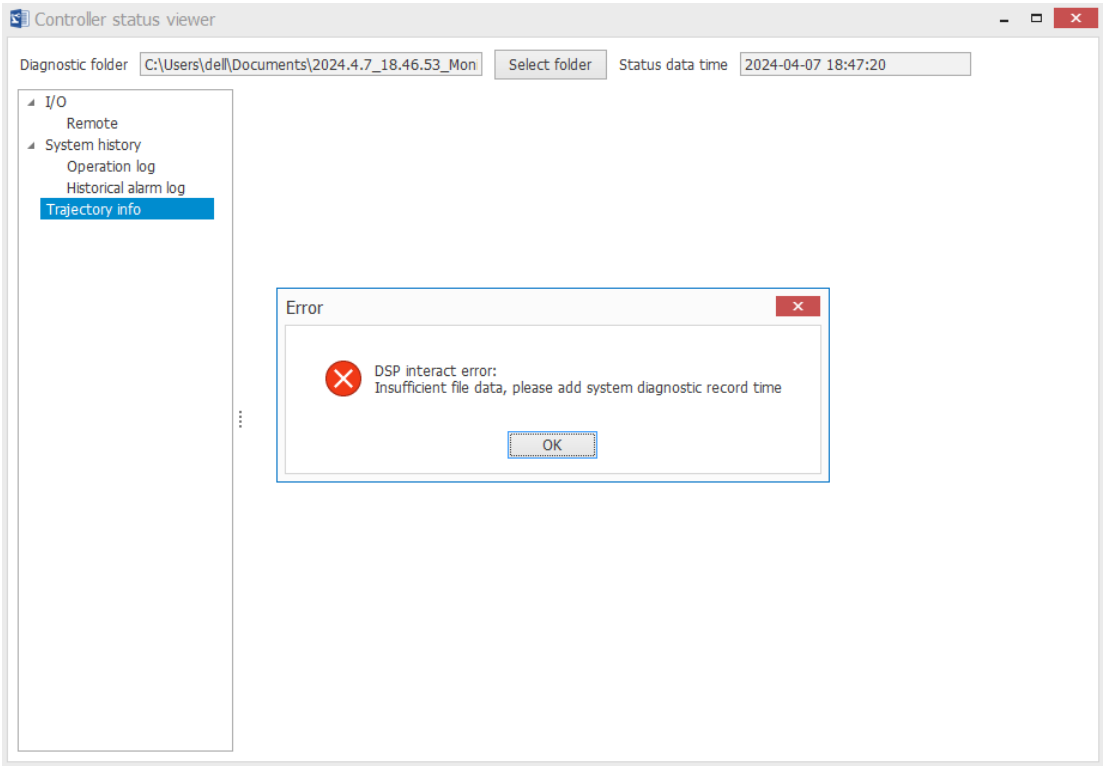
NOTE

If the "Trajectory info" window is opened when the folder is changed, you will be prompted to close the window before proceeding to the next step.



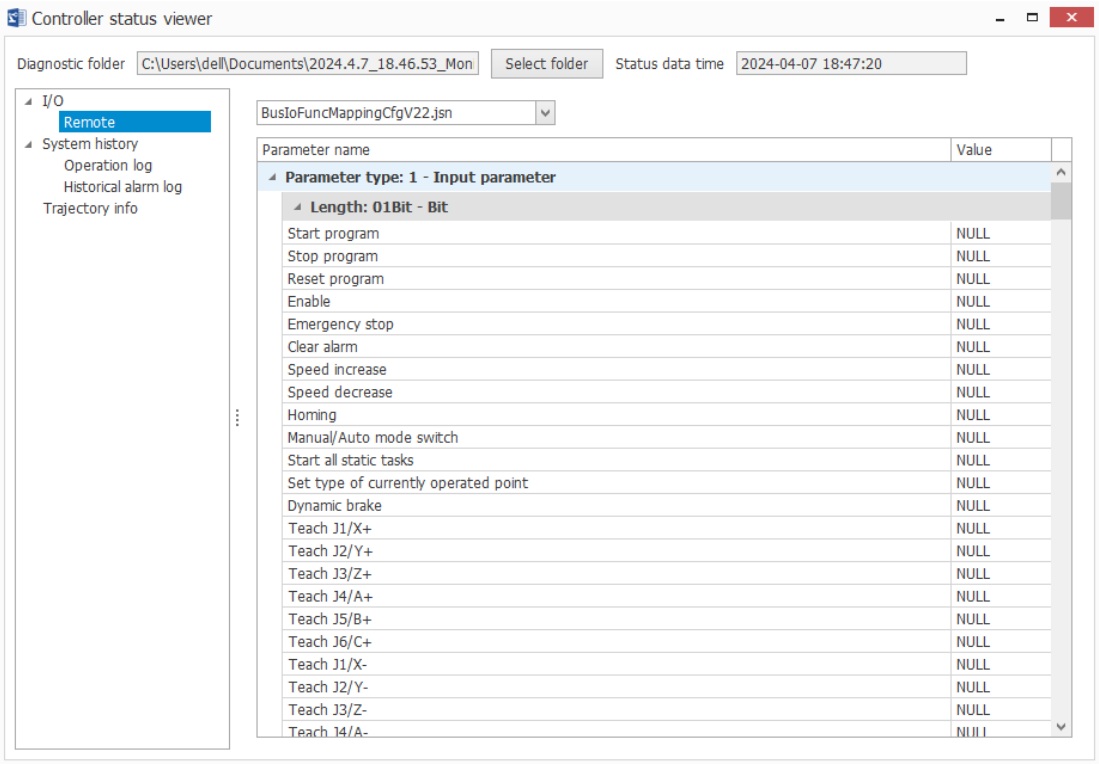


3. You can view different diagnostic information by switching different nodes. If there is no node information in the diagnostic folder, a pop-up window will be displayed.



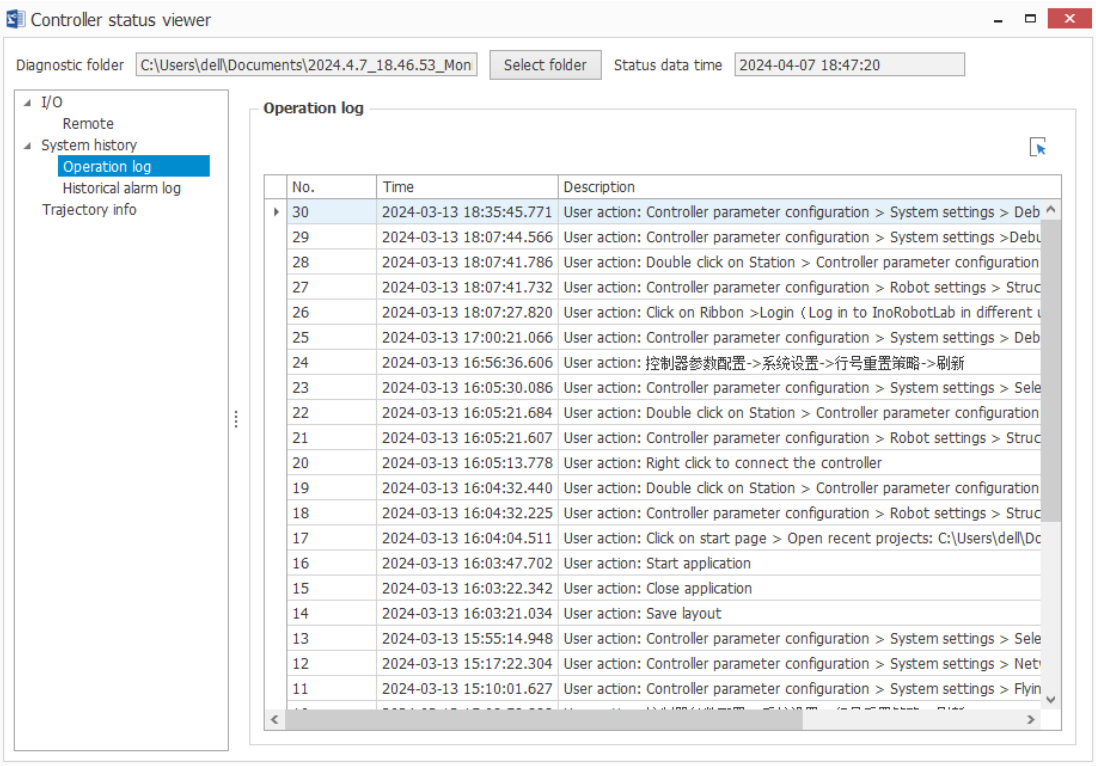
9.9.3.2.1 Remote

In the controller status viewer, click "I/O" > "Remote" to view the remote I/O configuration information. If there are multiple copies of remote configuration information in the folder, the first one is parsed by default. You change switch between different copies by using the drop-down box.

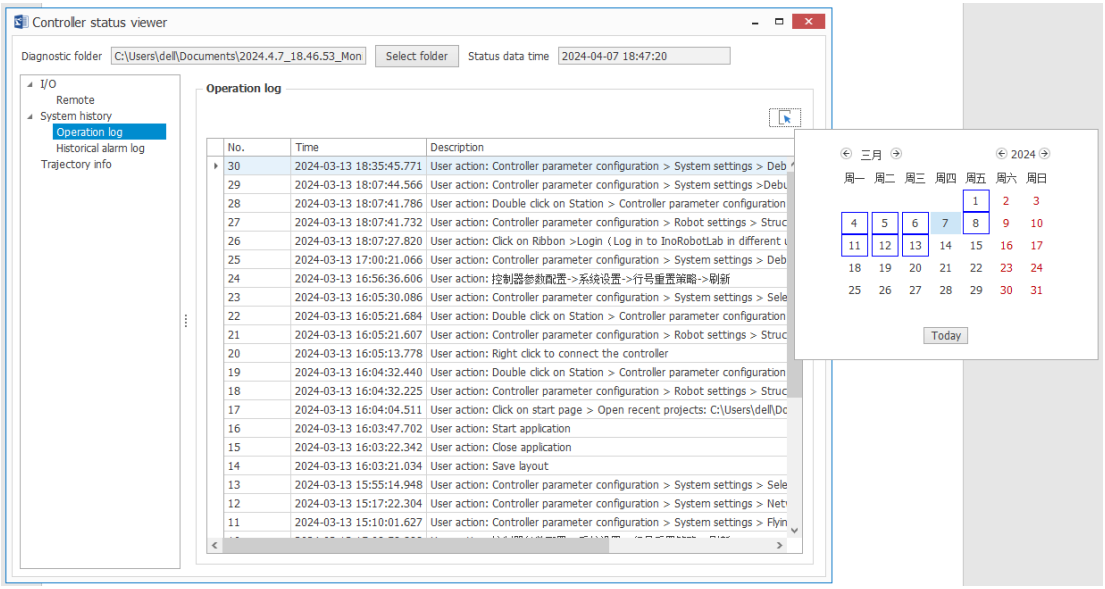


9.9.3.2.2 Operation Log

1. In the controller status viewer, click "System history" > "Operation log" to view the operation log information.

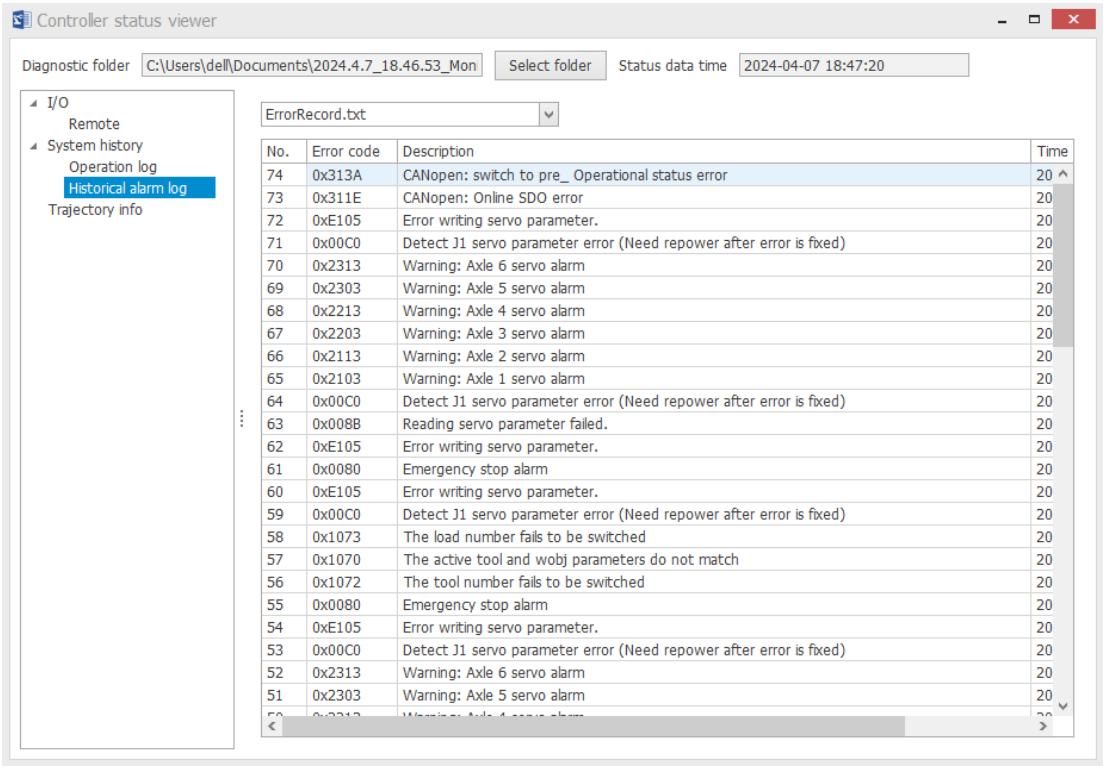


2. On the operation log page, click the blue arrow button in the upper right corner to display the calendar. The date with operation log in the calendar will be marked by a blue box.

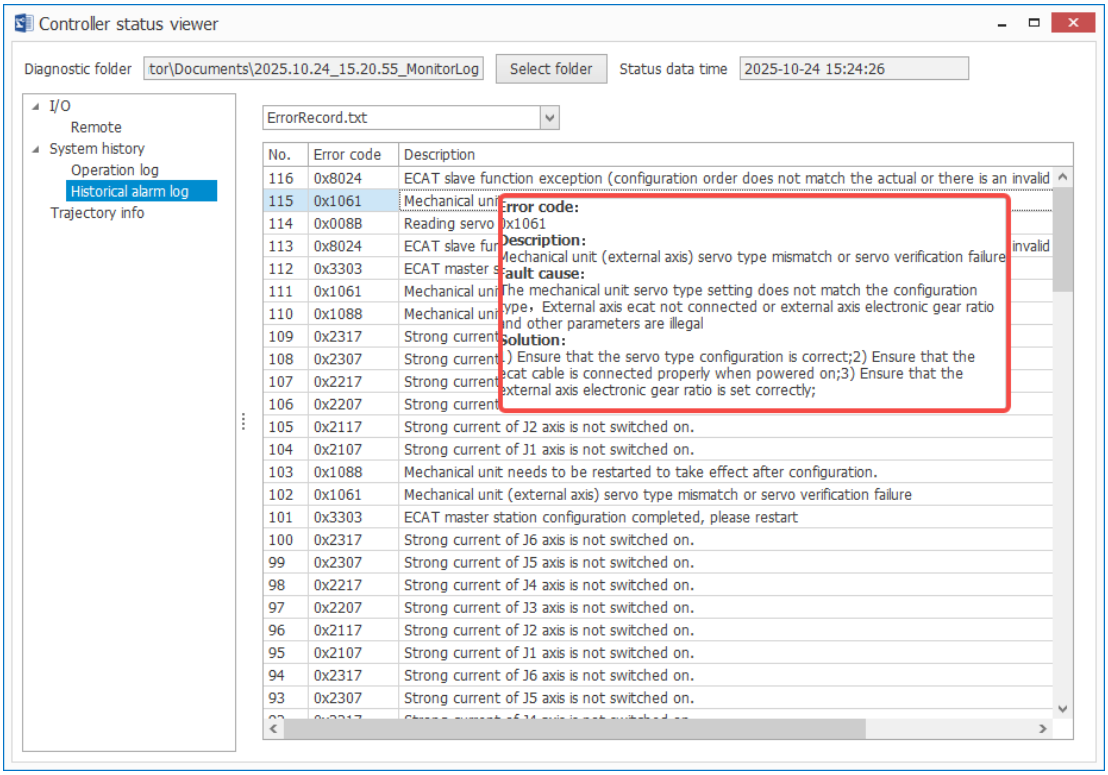


9.9.3.2.3 Historical Alarm Log

1. In the controller status viewer, click "System history" > "Historical alarm log" to view the historical alarm information. You can also use the drop-down box to switch the parsed historical alarm files.



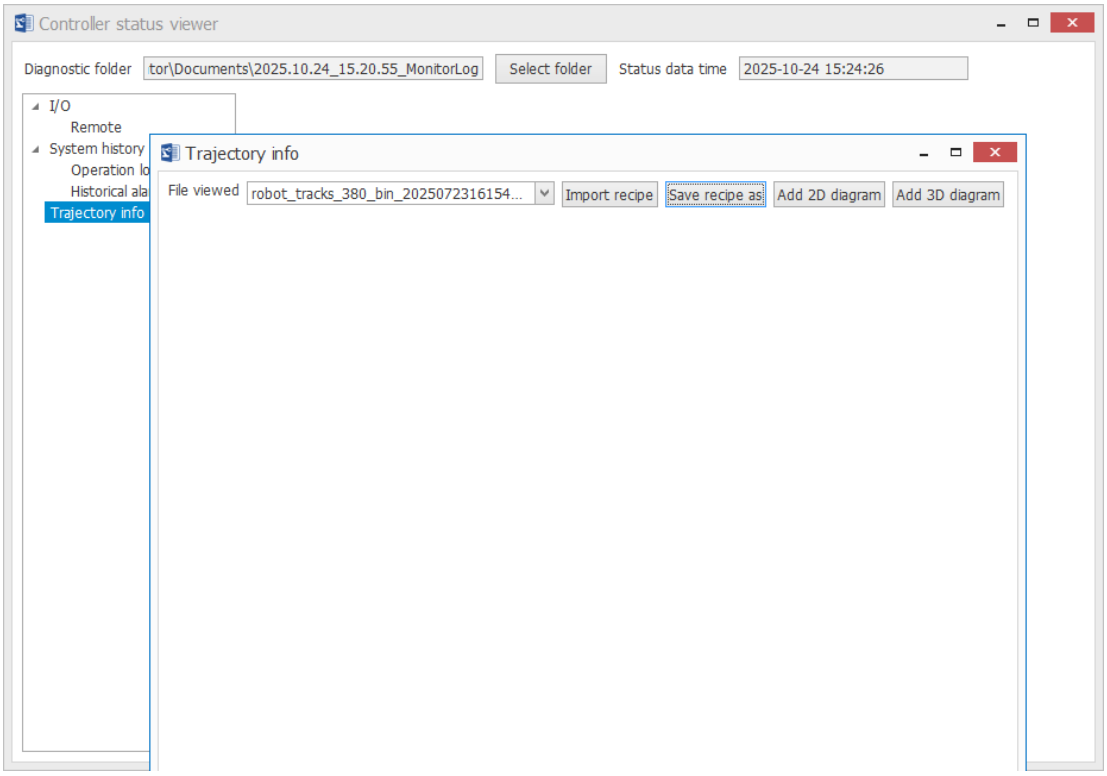
2. Double-click any alarm record. A floating box appears, displaying the error code, description, fault cause, and solution.



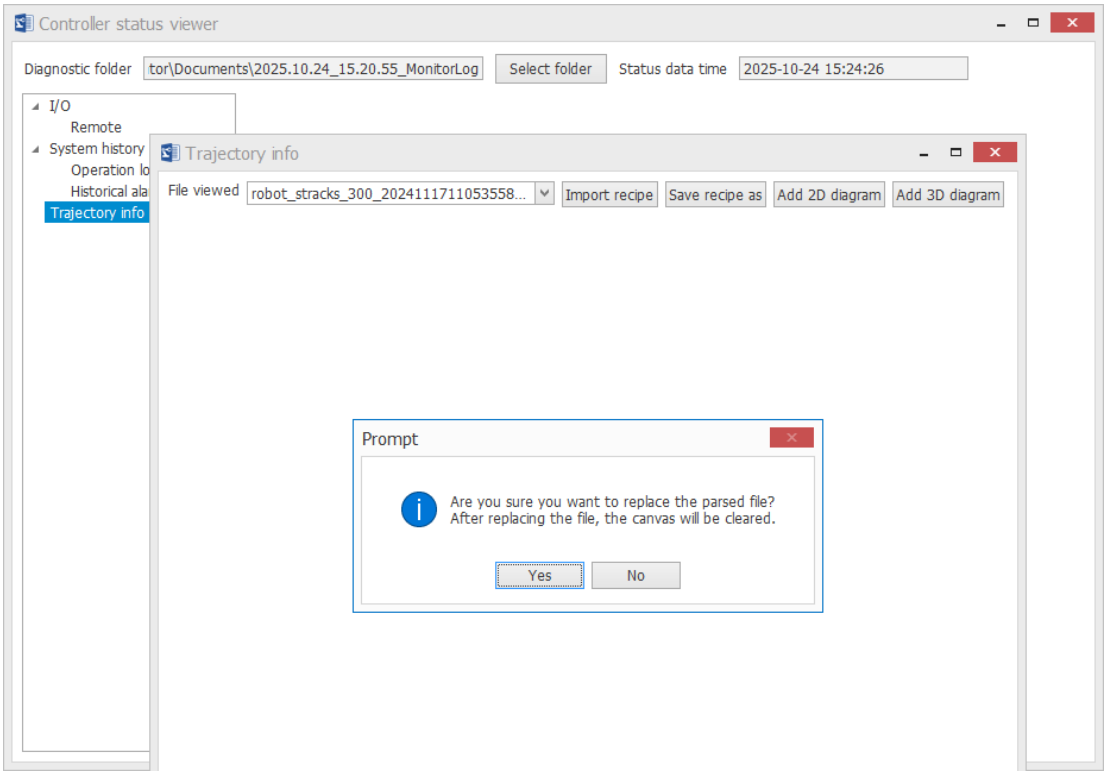
9.9.3.2.4 Trajectory Information

Trajectory Information

1. In the controller status viewer, click "Trajectory Info" to view the trajectory information. This window is a floating window. You can close this window and open it again by double-clicking "Trajectory Info". The first trajectory file is parsed by default.

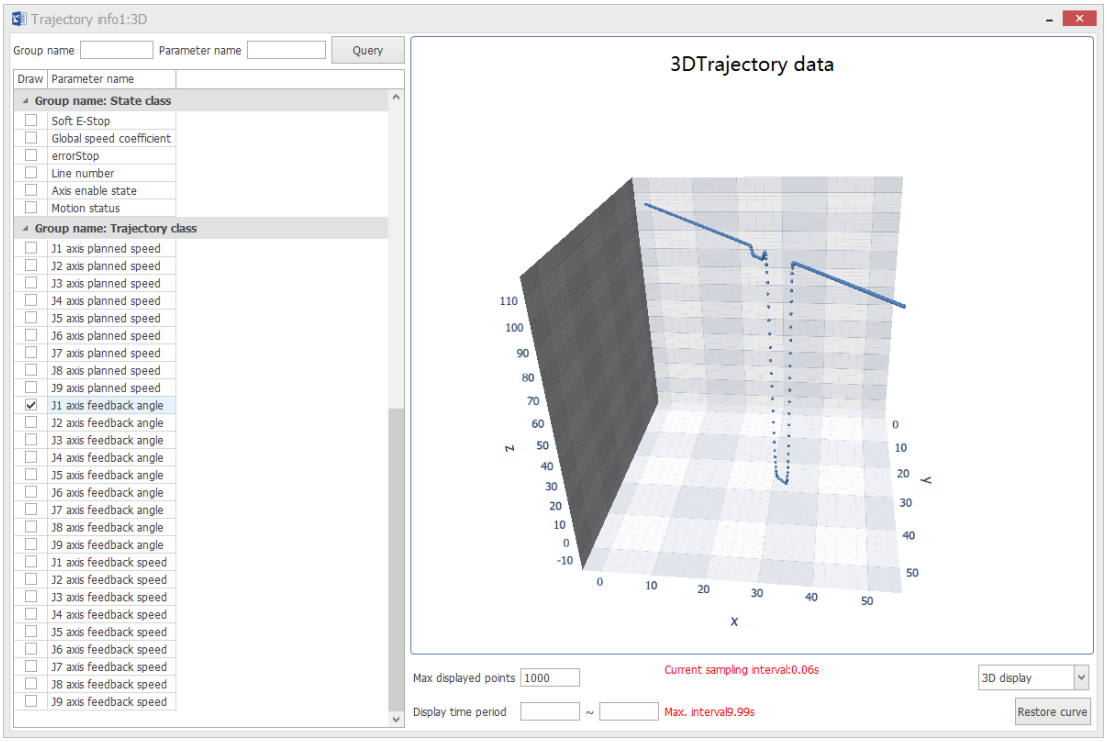


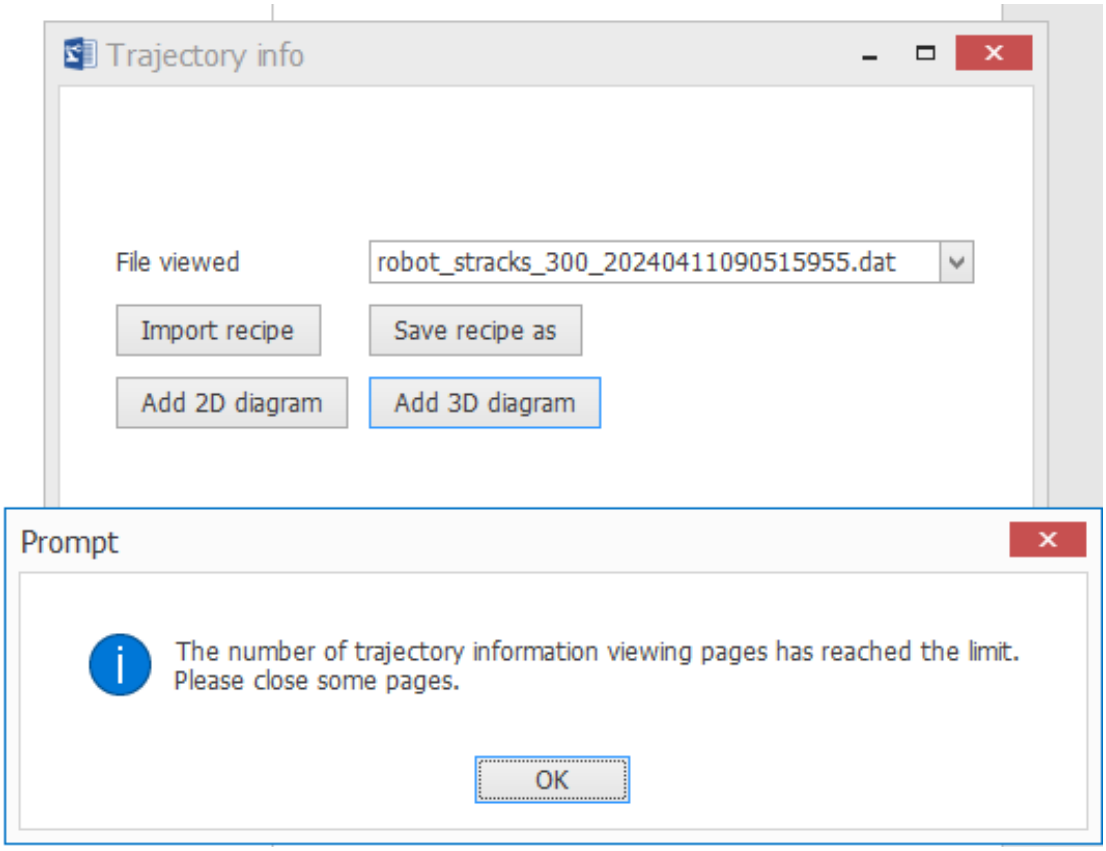
2. Use the drop-down box to switch the parsed trajectory file. When the file is switched, all 2D/3D interfaces will be closed.



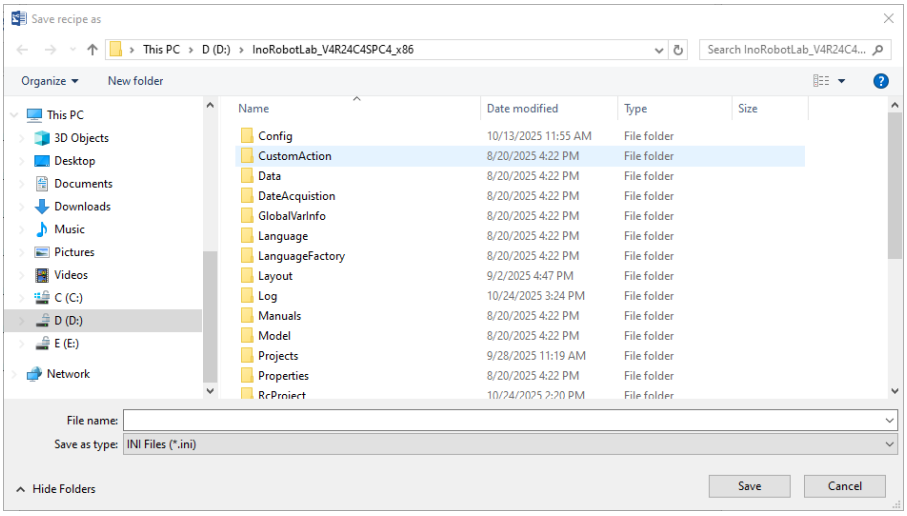
3. Click the "Add 2D diagram" or "Add 3D diagram" button to add 2D or 3D diagrams. At most eight 2D and 3D diagrams can be opened at the same time.


Trajectory info1:2D
Group name
Parameter name
Query
Draw
Parameter name
Color
Curve point
Multiple
Group name: State class
Soft E-Stop
Global speed coefficient
errorStop
Line number
Axis enable state
Motion status
Group name: Trajectory class
J1 axis planned speed
J2 axis planned speed
J3 axis planned speed
J4 axis planned speed
J5 axis planned speed
J6 axis planned speed
J7 axis planned speed
J8 axis planned speed
J9 axis planned speed
J1 axis feedback angle
J2 axis feedback angle
J3 axis feedback angle
J4 axis feedback angle
J5 axis feedback angle
J6 axis feedback angle
J7 axis feedback angle
J8 axis feedback angle
J9 axis feedback angle
J1 axis feedback speed
J2 axis feedback speed
J3 axis feedback speed
J4 axis feedback speed
J5 axis feedback speed
J6 axis feedback speed
J7 axis feedback speed
J8 axis feedback speed
J9 axis feedback speed
Black
2DTrajectory data
J1 axis feedback angle
J3 axis feedback angle
Value
Time
Restore curve


Trajectory info1:3D
Group name
Parameter name
Query
Draw
Parameter name
Group name: State class
Soft E-Stop
Global speed coefficient
errorStop
Line number
Axis enable state
Motion status
Group name: Trajectory class
J1 axis planned speed
J2 axis planned speed
J3 axis planned speed
J4 axis planned speed
J5 axis planned speed
J6 axis planned speed
J7 axis planned speed
J8 axis planned speed
J9 axis planned speed
J1 axis feedback angle
J2 axis feedback angle
J3 axis feedback angle
J4 axis feedback angle
J5 axis feedback angle
J6 axis feedback angle
J7 axis feedback angle
J8 axis feedback angle
J9 axis feedback angle
J1 axis feedback speed
J2 axis feedback speed
J3 axis feedback speed
J4 axis feedback speed
J5 axis feedback speed
J6 axis feedback speed
J7 axis feedback speed
J8 axis feedback speed
J9 axis feedback speed
3DTrajectory data
Z
X
Y
Max displayed points
1000
Current sampling interval:0.06s
3D display
Display time period
Max. interval9.99s
Restore curve



4. Click the "Save recipe as" button. In the pop-up dialog, select the save path and enter the file name. The parameters configured on the current interface will be saved as a recipe file. A pop-up message will be displayed when the save is successful or fails. Currently, only all open 2D interfaces can be saved as a recipe file.

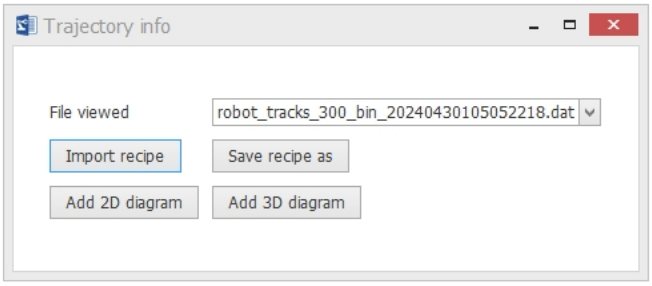


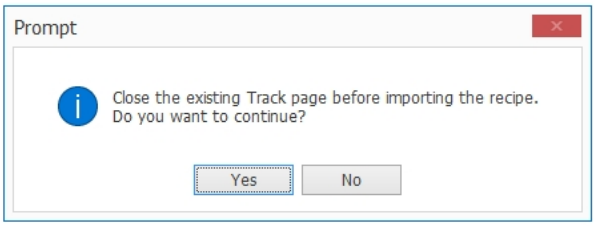
5. Click the "Import recipe" button. A pop-up dialog appears. After you select the recipe file, the software opens the last saved interface.

   a) When parsing the recipe file, if the data parsed does not contain a parameter in the original recipe file, this parameter will be skipped in the drawing process.

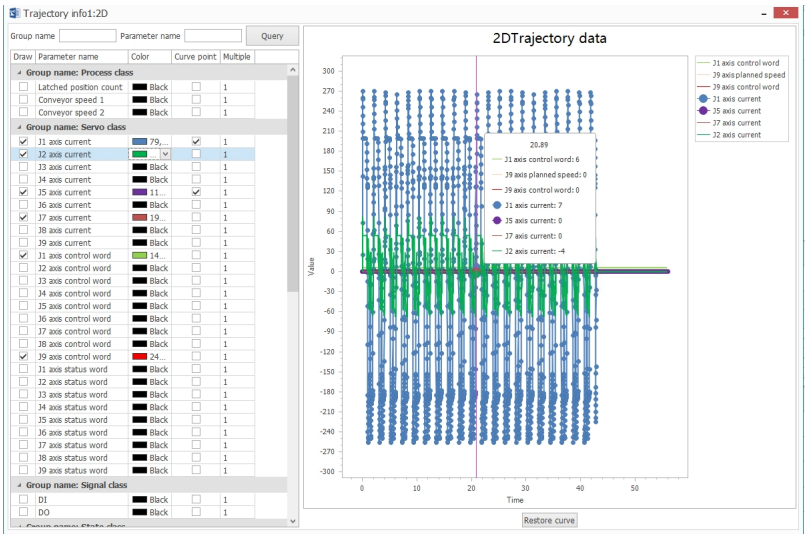
   b) If the format of the parsed recipe file is incorrect, a pop-up window appears indicating that the recipe file failed to be imported.

   c) Importing a recipe file closes all existing trajectory information interfaces.


Trajectory info
File viewed
robot_tracks_300_bin_20240430105052218.dat
Import recipe
Save recipe as
Add 2D diagram
Add 3D diagram

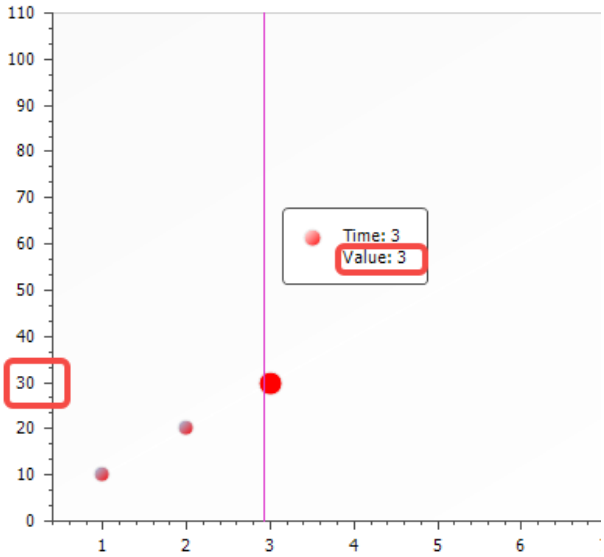

Prompt
Close the existing Track page before importing the recipe.
Do you want to continue?
Yes
No

- 2D Diagram:


Trajectory info1:2D
Group name
Parameter name
Query
Draw
Parameter name
Color
Curve point
Multiple
Group name: Process class
Latched position count
Conveyor speed 1
Conveyor speed 2
Group name: Servo class
J1 axis current
J2 axis current
J3 axis current
J4 axis current
J5 axis current
J6 axis current
J7 axis current
J8 axis current
J9 axis current
J1 axis control word
J2 axis control word
J3 axis control word
J4 axis control word
J5 axis control word
J6 axis control word
J7 axis control word
J8 axis control word
J9 axis control word
J1 axis status word
J2 axis status word
J3 axis status word
J4 axis status word
J5 axis status word
J6 axis status word
J7 axis status word
J8 axis status word
J9 axis status word
Group name: Signal class
DI
DO
2DTrajectory data
J1 axis control word
J9 axis planned speed
J9 axis control word
J1 axis current
J5 axis current
J7 axis current
J2 axis current
20.89
J1 axis control word: 6
J9 axis planned speed: 0
J9 axis control word: 0
J1 axis current: 7
J5 axis current: 0
J7 axis current: 0
J2 axis current: -4
Value
Time
Restore curve

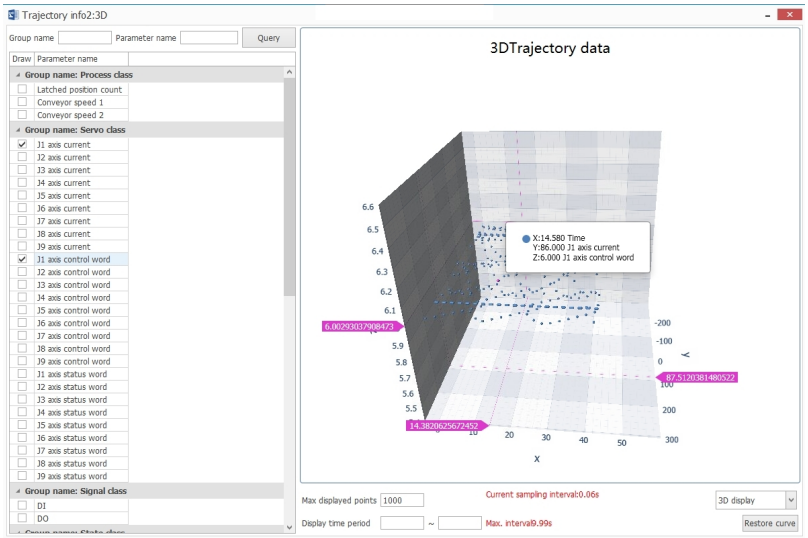
NOTE

- By default, the table displays information by group name.
- The table supports the filtering function. You can enter the group name and parameter name to search for a specific record.
- Description of table columns:
  - Draw: When checked, the parameter is drawn. A 2D diagram supports drawing up to eight curves.
  - Color: Click to display a color selection box to change the color of the curve.
  - Curve point: When checked, the data points are displayed. Otherwise, only a solid line is displayed.
  - Multiplier: How many times the Y-axis can be magnified/reduced. Example: The original value 3 is 30 after being multiplied by 10, but the indicated value remains 3.



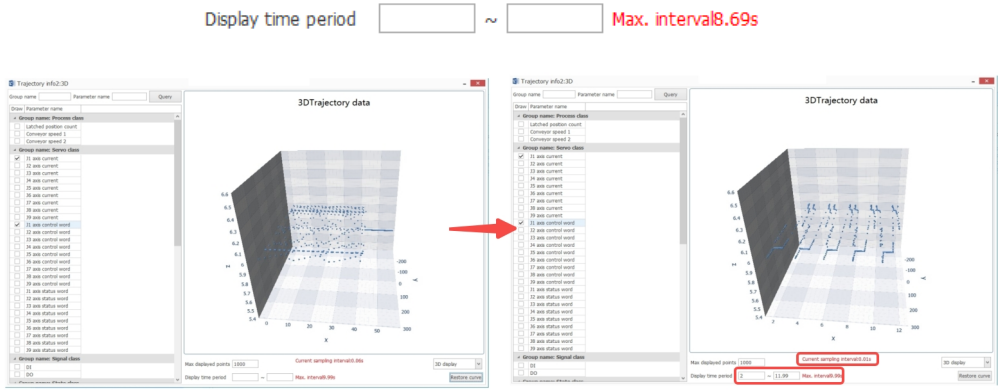
- Drawing area: The curve supports zoom-in/zoom-out by the mouse wheel and drag by the left mouse button. When the mouse cursor stops at a drawing point, the current data is displayed.
- Restore curve: After the curve is zoomed in/out or dragged, click this button to restore the curve to its original size.

- 3D diagram



NOTE

- Similar to the 2D diagrams, the table also supports the filtering function.
- Color, curve point, and multiplier are not supported. The curve points are displayed in blue. You can select up to three parameters to draw. When three parameters are selected, the parameters are selected as X, Y, and Z axes. When two parameters are selected, the parameters are selected as Y and Z axes respectively, and the X axis is a time axis. When only one parameter is selected, the parameter is selected as Z axis, and the X and Y axes are all time axes.
- Drawing area: The 3D curve supports zoom-in/zoom-out, drag, and rotation. When the mouse cursor stops at a drawing point, the current data is displayed.
  - Use the left mouse button to rotate, mouse wheel to zoom in or out, and Ctrl + left mouse button to move (press Ctrl first) the 3D curve.
  - Key inputs are supported. Press the arrow key to move the diagram in the direction specified by the keyboard arrow key. Press the "+" key on the numeric keypad to zoom in the diagram, or "-" to zoom out the diagram.
- The curve is drawn with a maximum of 1000 points by default. The maximum number of points is 3000. 3D drawings depend largely on computer performance. If the computer runs slowly, reduce the number of drawing points. (Recommended computer configuration: Operating system win7 and above, processor: 12th Gen Intel(R) Core(TM) i5-1235U, 1.30 GHz, memory: 16.0GB)
- Current sampling interval: Displays the interval time between points in the current drawing process.
- The curve points displayed by default may be missing. You can enter the start and end time to view all the data during this period of time.

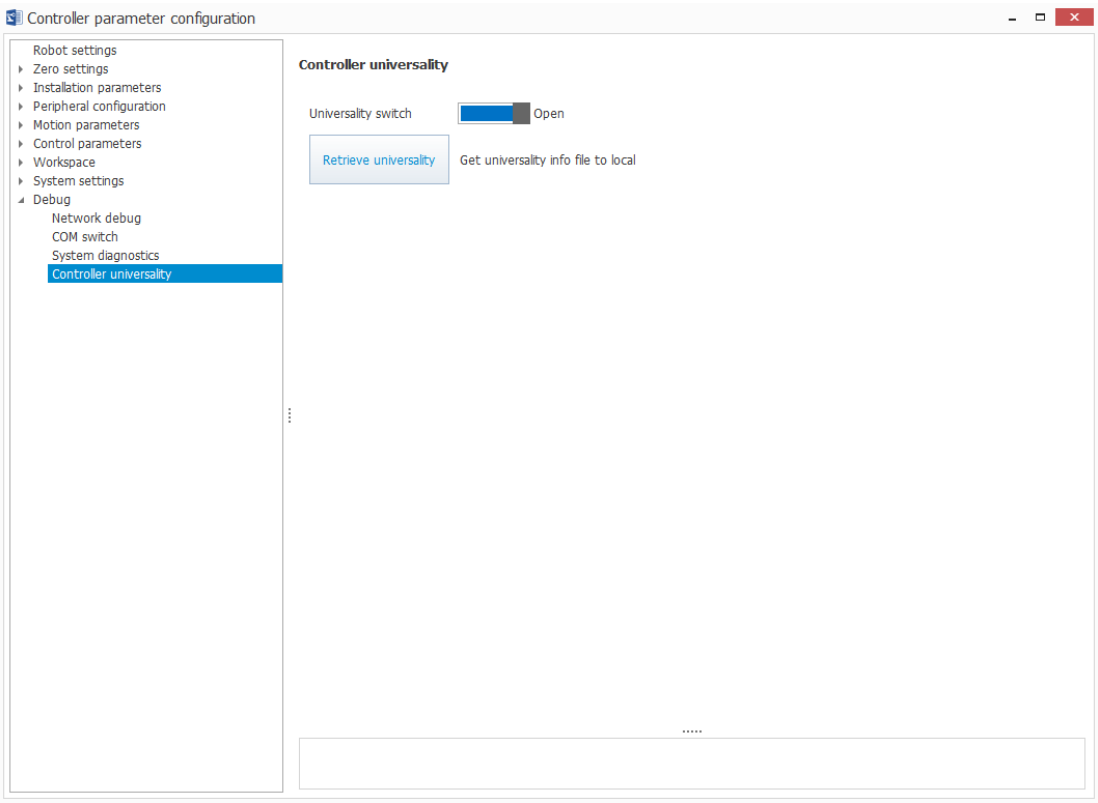


- Projection selection: You can switch to "XY projection", "YZ projection", "XZ projection" or "3D display" to change the viewing angle. Note that the curve does not support rotation in the projection mode.
- Restore curve: After the curve is dragged, zoomed in or out, rotated, or filtered, you can click this button to restore the curve to 3D display mode. The curve is restored to its original size, angle, and position, and the time range entered is cleared. The trend for the entire time period is displayed again.

9.9.4 Controller Universality

The controller universality function is generally divided into two sub-functions: Universal switch (12.8.1 Functions and Operating Steps > Universal Switch) and retrieve universality (12.8.1 Functions and Operating Steps> Retrieve Universality). Double-click "Controller parameter configuration" in the device tree, then

select "Debug" > "Controller universality" in the opened interface for related settings. The specific operations are shown in the figure below:



10 Communication Manager

10.1 Overview

It allows you to monitor the connection between the controller and remote devices, view and modify the server/client type, IP address, port, and connection status.

NOTE

Description of client and server:The robot can be selected as a client or server , or as both a client and a server during communication with an external device. This function is mostly used for the vision function. You can designate the role of the robot in the vision function as a client or server. Two hardware network interfaces EtherNet 1 and EtherNet 2 are provided. Therefore, the robot supports communication with third-party devices via the two network interfaces regardless of whether it acts as a client or server.

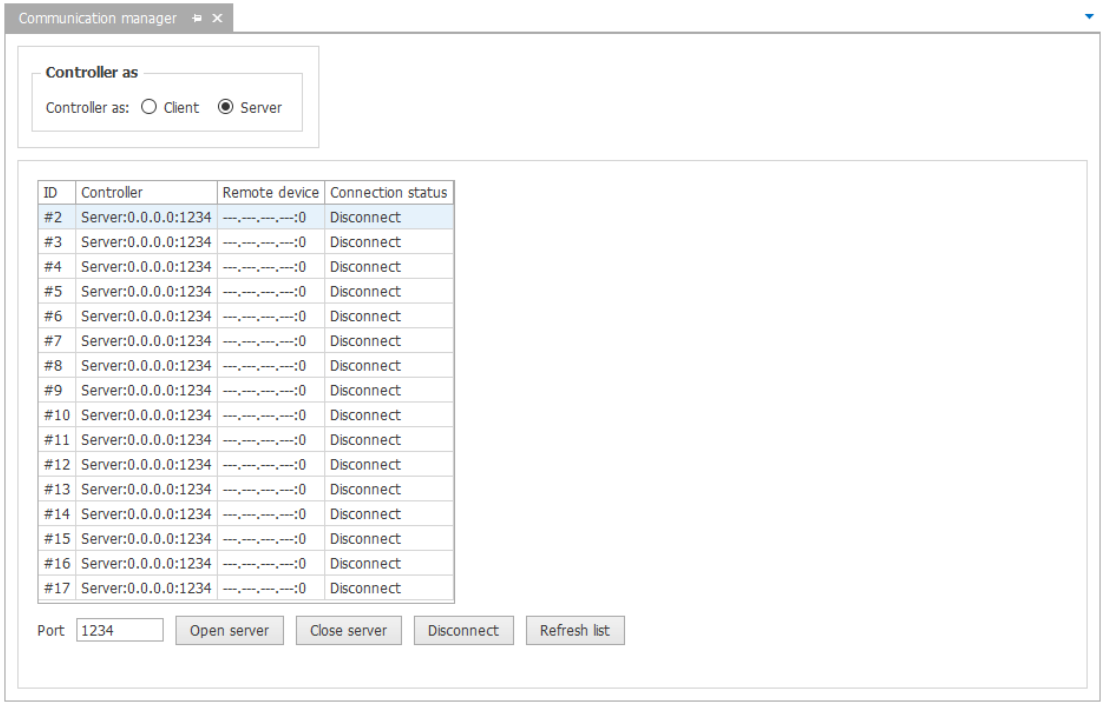
10.2 Server Mode

10.2.1 Opening the Server

Set the controller as a server so that other client devices (such as vision software) can directly connect to the controller for communication.

Steps

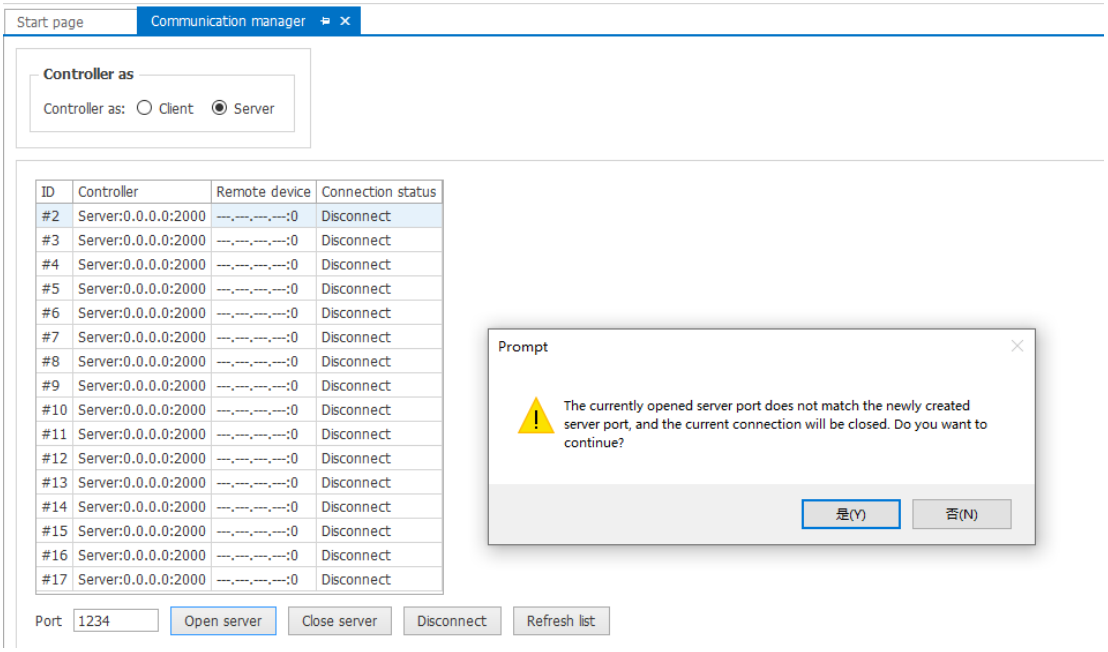
1. Set the controller as server, enter the port number, and click "Open server". The controller will then bind to the local IP address and the entered port number to listen for connections from remote clients.



2. The settings take effect immediately after being saved.

Results

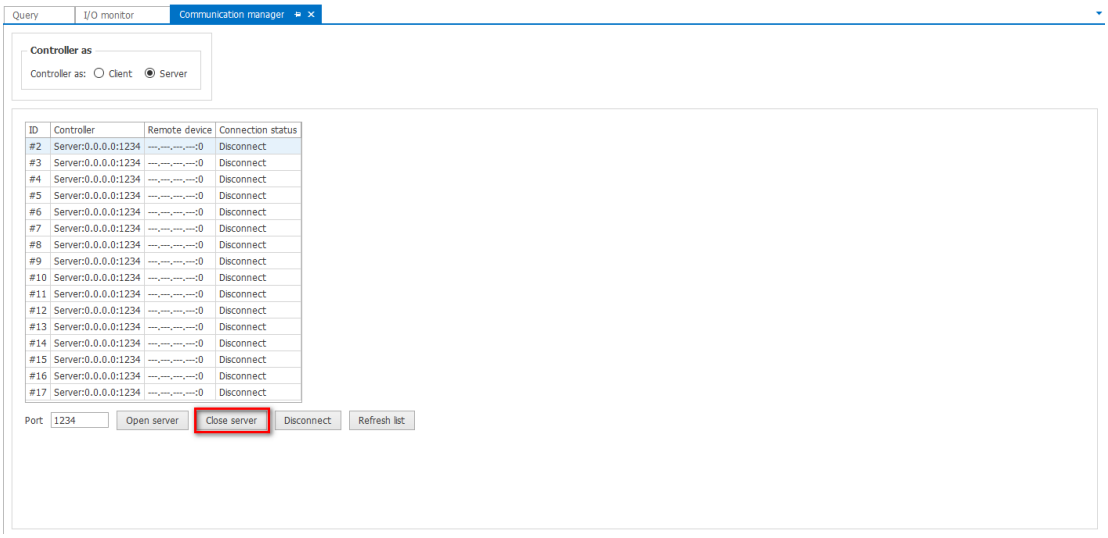
- If there is no client connection, the corresponding connection status displays "Disconnect"; If there is a client connection, the corresponding connection status displays "Connect", and the output interface will display: "Prompt: Remote client (10.45.151.163:1) successfully connected to server (2000)"; If the client actively closes at this time, the output interface will display: "Prompt: Client [10.45.151.163:1] closed_3".
- Enter a proper port number before you open the server. The port number must be an integer between 1024 and 65535. Special port numbers cannot be used, including 1111, 2222, 3333, 4444, 5555, 6666, 7777, 8888, 9999, 1217, 8080, 11226, 11740, 4003, and 44818.
- When the server is already open, if the port number is changed to be different from the previous one, a prompt pops up requiring for confirmation.
  - If you select "Yes", all the current connection will be closed to open new server port for listening.
  - If you select "No", the original connection remains and the interface does not change.



- In this mode, the controller acts as a server. Up to 16 clients connections are supported.

10.2.2 Closing the Server

When the server is in the "Closed" status, perform the "Close server" operation on the controller. At this point, the controller becomes "Closed" as a server, preventing remote clients from connecting and communicating with the controller.



Steps

1. Select "Server" mode. The "Port" input field will display the currently open server port. Then, click the "Close server" button.
2. The settings take effect immediately after being saved.

Results

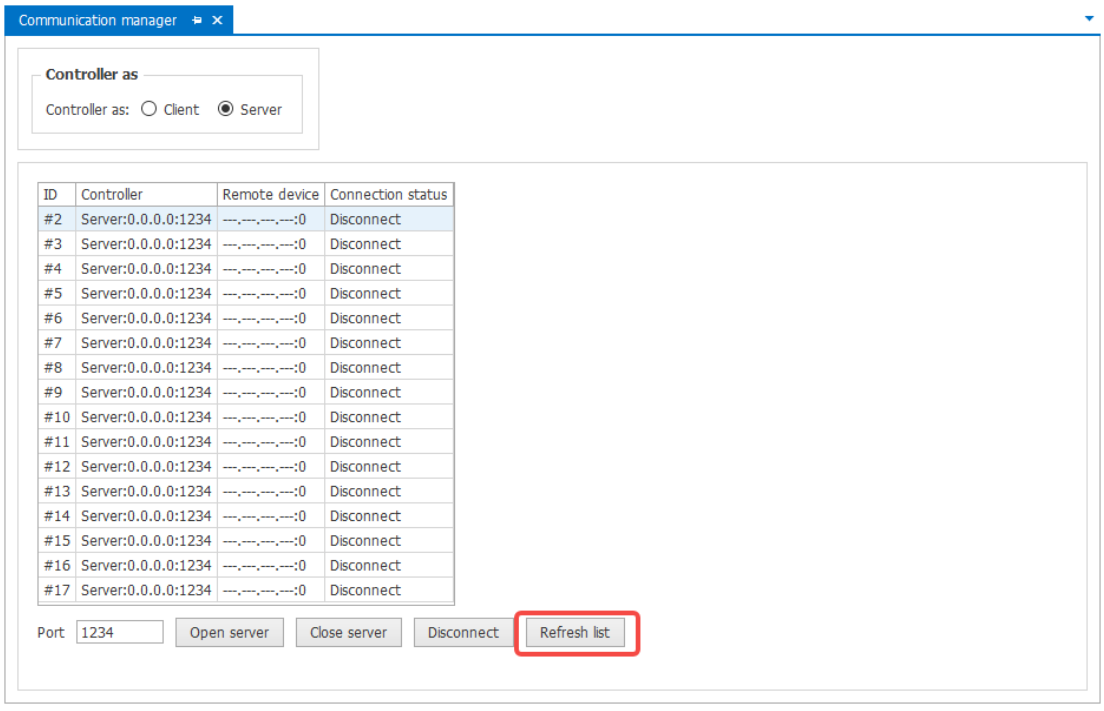
All connections from remote clients are disconnected, and the controller as a server switches to the "Closed" status.

10.2.3 Disabling the Connection

When the controller acts as a server, you can disconnect a specified remote client from the server.

Steps

1. Select "Server" mode for the controller, highlight the specified connection, and click the "Disconnect" button.



2. The settings take effect immediately after being saved.

Results

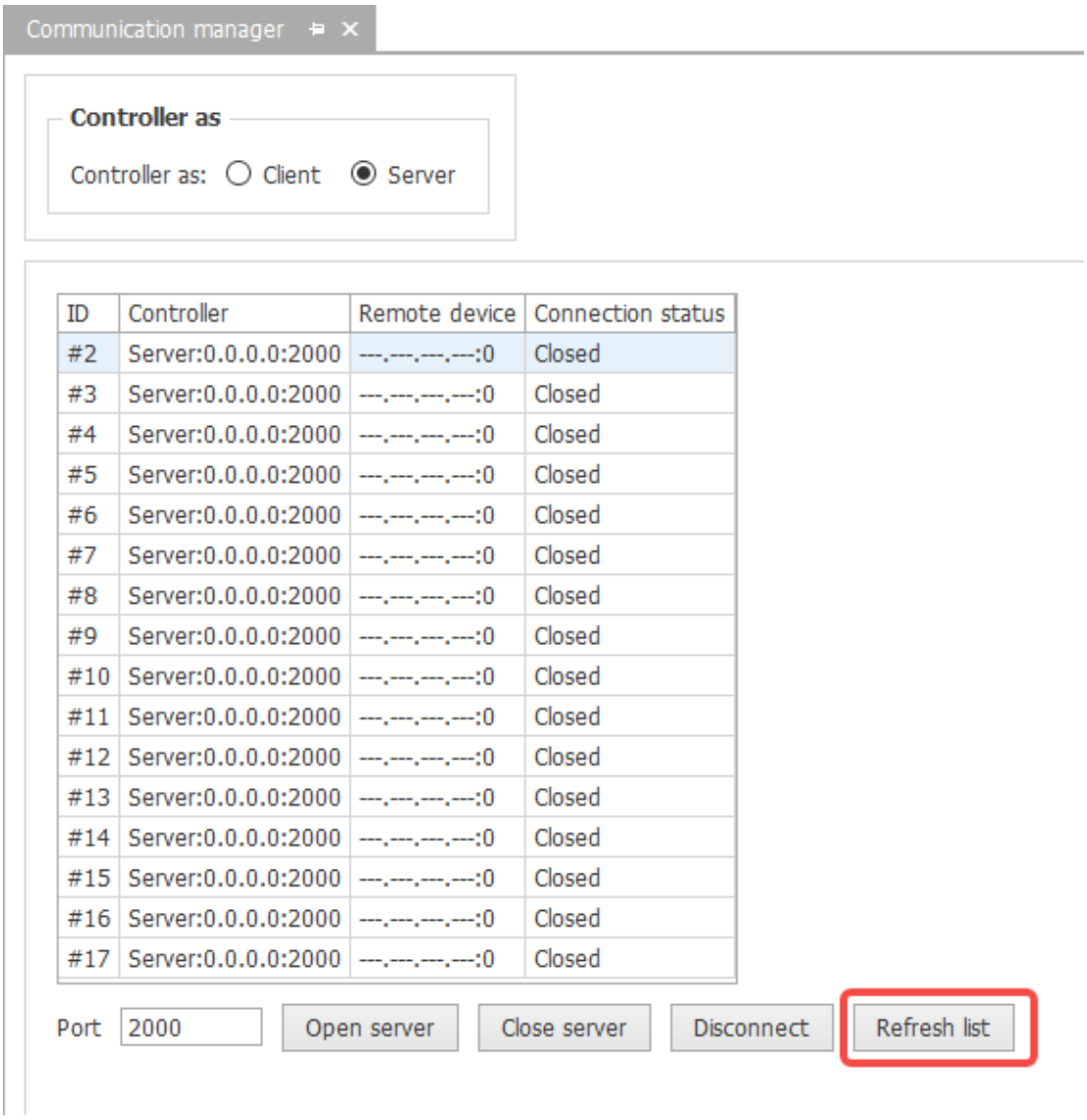
The currently selected connection on the controller is shown as "Disconnect". The output interface will display: "Prompt: Client [10.45.151.163:1] closed_3".

10.2.4 Refreshing the List

When the controller acts as a server, InoRobotLab does not refresh the client connection status in real time. Therefore, when a client is connected to the controller, the connection list does not refresh automatically. You can click "Refresh list" to obtain the latest connection status of the controller.

Steps

1. Click the "Refresh list" button.



2. The settings take effect immediately after being saved.

Results

The latest connection information and status is updated in the list.

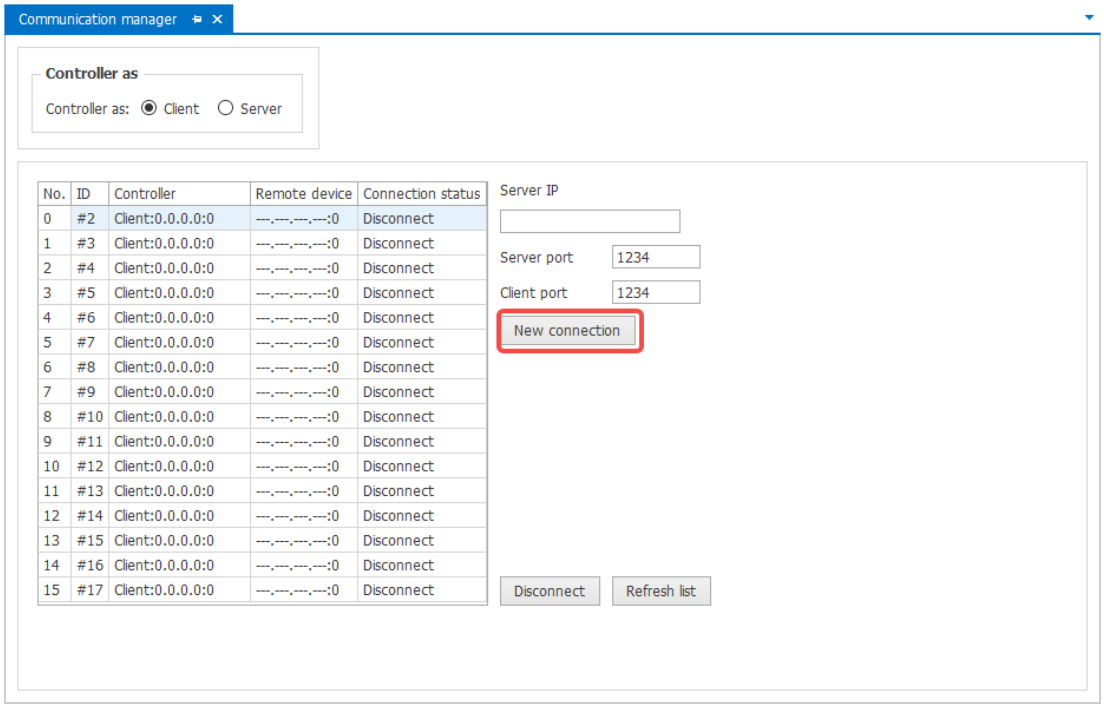
10.3 Client Mode

10.3.1 Creating a Connection

When the controller acts as a client, you can click the "New connection" button to connect the controller to other server device for TCP communication.

Steps

1. Select "Client" mode, enter the remote "Server IP" and "Server port", as well as the local "Client port" in the input fields, then click the "New Connection" button.



2. The settings take effect immediately after being saved.

Results

The controller is connected to other devices (vision devices) as a client. When the connection is successful, the output interface displays: "Prompt: TCP client (1234) successfully connected to (10.45.151.163:1234)".

NOTE

Description text style

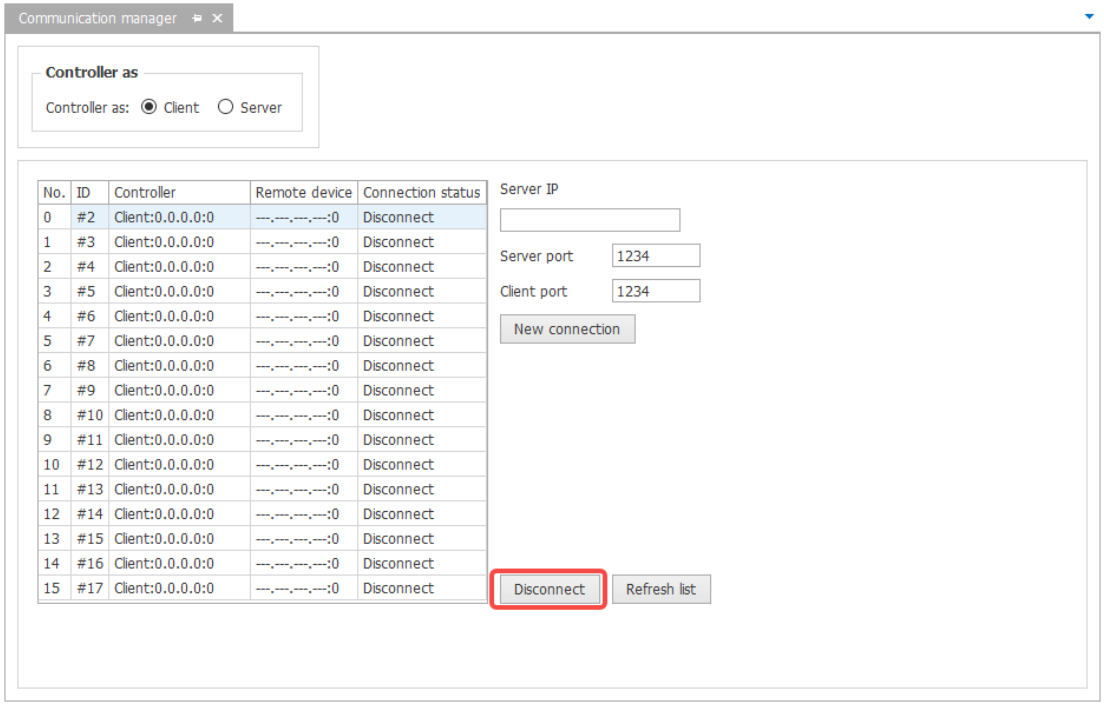
- The controller can establish up to 16 connections as a client.
- When the controller switches to client mode, both the server port and client port default to "1234".
- When the controller acts as a client, each client port number can only connect to one server. If you connect a client port number, which already connects to a server, to another server again, you will be prompted to change another port.
- The client port number ranges from 2 to 17, and 1024 to 65535. Special port numbers cannot be used, including 1111, 2222, 3333, 4444, 5555, 6666, 7777, 8888, 9999, 1217, 8080, 11740, 4003, and 44818. Ports 2 to 17 are port aliases representing the 1st to 16th Socket connections. When using 2 to 17 as client port numbers, the system automatically assigns a real port number.

10.3.2 Disabling the Connection

When the controller acts as a client, you can disconnect it from a specified server.

Steps

1. Select a connection in the list and click "Disconnect".



2. The settings take effect immediately after being saved.

Results:

- The "Remote device" column of the selected connection displays "--.--.--.--:0", and the connection status displays "Disconnect". The output interface will display: "Prompt: Client [1234] Closed_5".
- The server is not able to communicate with the controller.

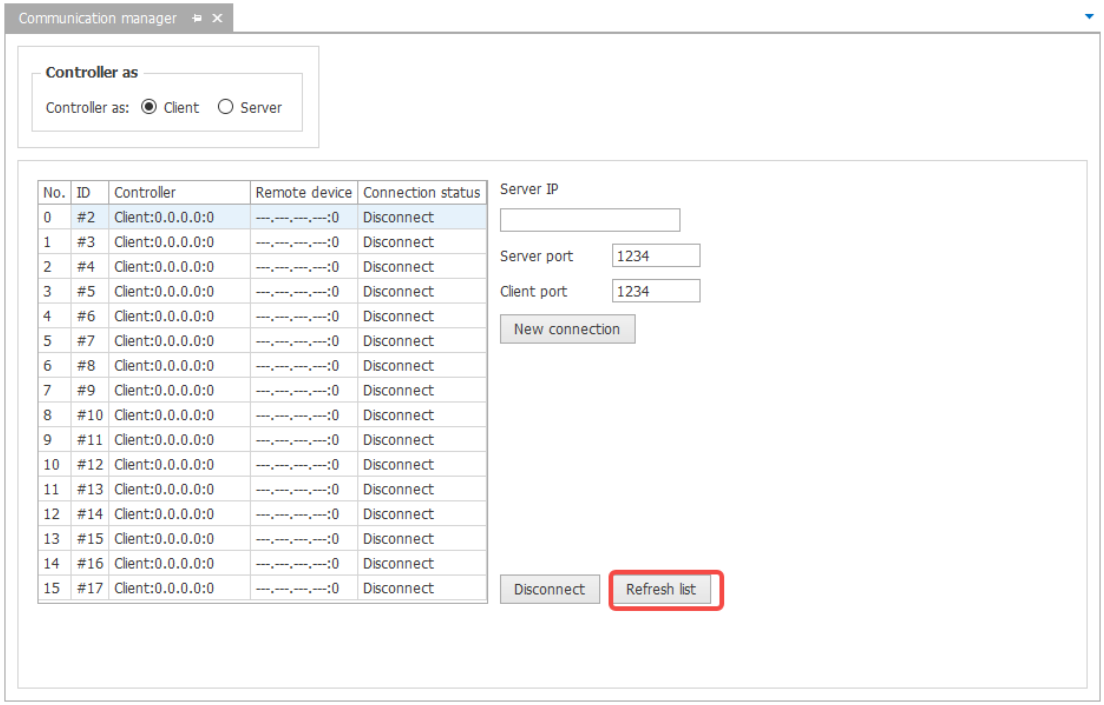
> **NOTE**
>
> When the connection is disabled, the connection list refreshes. If the operation is successful, the first line will be selected by default.

10.3.3 Refreshing the List

When the controller acts as a client, InoRobotLab does not refresh the client connection status in real time. Therefore, when the client connects to or disconnects from the server, the connection list does not refresh automatically. You can click "Refresh list" to obtain the latest connection status of the controller.

Steps

1. Click the "Refresh list" button.



2. The settings take effect immediately after being saved.

Results

The latest connection information and status is updated in the list.

11 System Resources

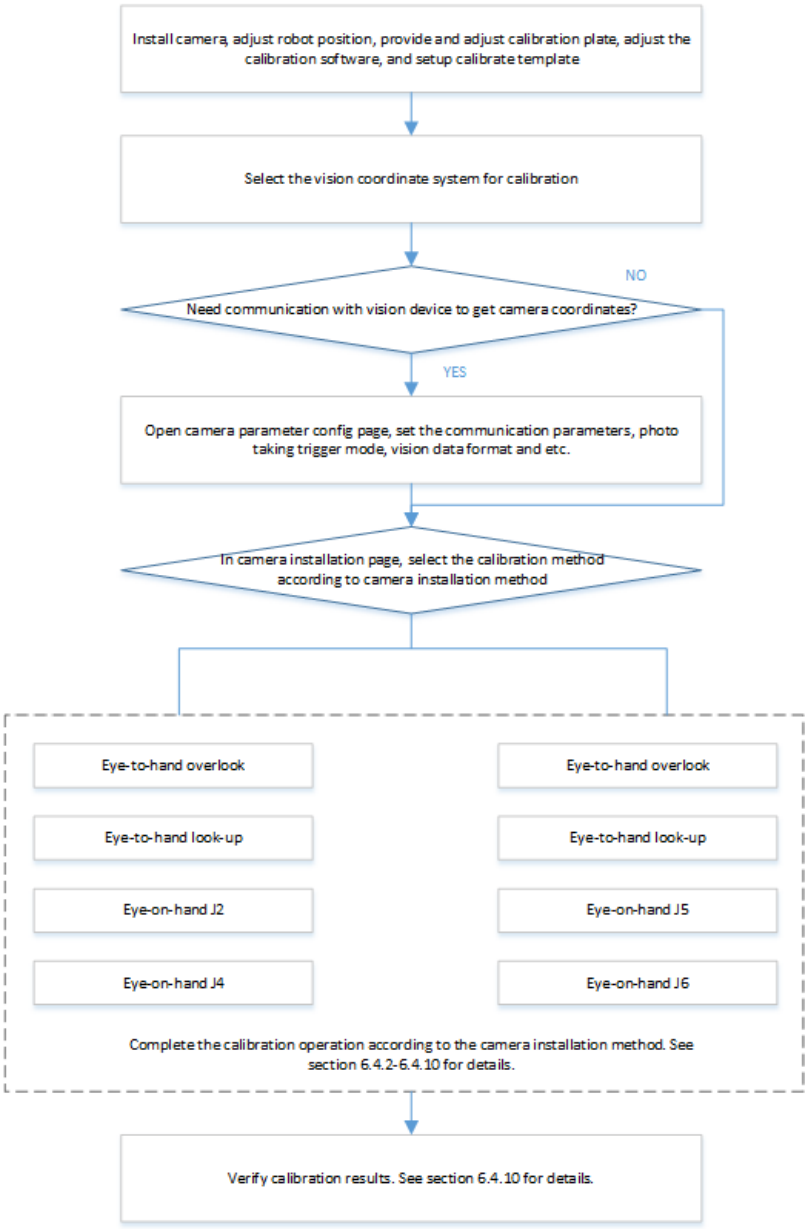
11.1 Vision Calibration Process

11.1.1 Overview

Vision calibration is the prerequisite to use the vision function of the robot system. The purpose of calibration is to obtain the relative position relationship between the camera and the robot, so that the pixel coordinates can be converted to the robot coordinates in subsequent operations. Inovance provides SCARA robots and 6-axis robots. These two types of robots have the following differences in the vision calibration operations.

Calibration steps	Operations for SCARA robots	Operations for 6-Axis robots
Calibrate the end fixture	-	Calibrate the end fixture
Calibrate the workobject frame	-	Calibrate the workobject frame (using calibrated end fixture)
Set the calibration parameters	Select parameters such as the calibration mode and camera installation mode	Select parameters such as the calibration mode and camera installation mode
Select datum points	Select two datum points	Select three datum points
9-point teaching	Perform 9-point teaching	Perform 9-point teaching
Finish	Finish	Finish

The following figure shows the vision calibration flowchart.

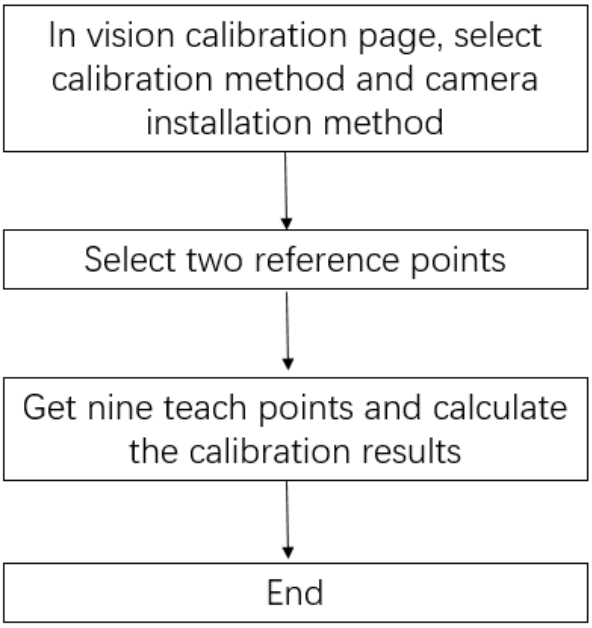


11.1.2 Vision Calibration of SCARA Robots

The vision calibration of SCARA robots includes Eye-to-Hand overlook calibration, Eye-to-Hand look up calibration, Eye-in-Hand J2 calibration, and Eye-in-Hand J4 calibration. All controller versions support vision calibration of SCARA robots.

Due to its structural limitations, SCARA robots generally work on user plane parallel to the base plane. The calibration process of SCARA robots does not require an associated workobject frame, and only two datum

points are need for the datum point teaching. The vision calibration process of SCARA robots is shown below.

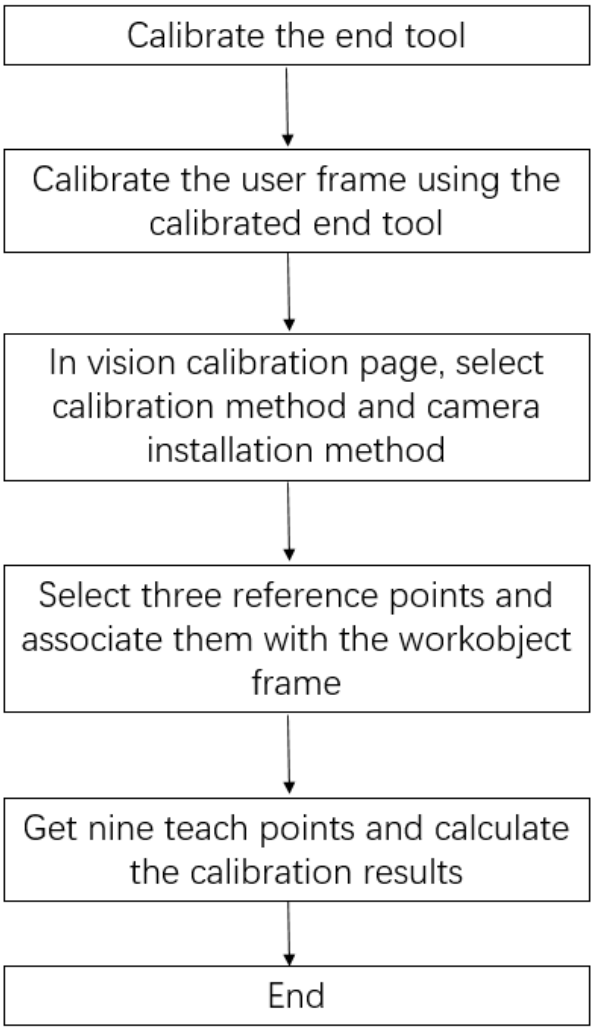


Vision calibration process of SCARA robots

11.1.3 Vision Calibration of 6-Axis Robots

The vision calibration of 6-axis robots includes Eye-to-Hand overlook calibration, Eye-to-Hand look up calibration, Eye-in-Hand J5 calibration, and Eye-in-Hand J6 calibration.

The vision calibration of 6-axis robots is only supported by controller version S01.17 and later.
The 6-axis robots can work on the user plane parallel to the base plane, as well as on the user plane not parallel to the base plane (but the Z direction must be upwards). During calibration, you need to select the associated workobject frame. When you select the associated workobject frame, ensure that the tool frame and workobject frame have been calibrated. In the datum point teaching, you need to select three datum points to determine the calibration tool. The vision calibration process of 6-axis robots is shown below.



Vision calibration process of 6-axis robots

CAUTION

- When performing datum point teaching, keep the camera and calibration board in relative parallel relationship. The orientation change between each two datum points must be larger than 30°.
- During calibration, pay attention to the introduction of vision calibration errors. Factors that affect the calibration accuracy include manual operation accuracy, tooling precision, vision detection accuracy, absolute accuracy of the manipulator, and calibration algorithm accuracy.

11.1.3.1 Steps

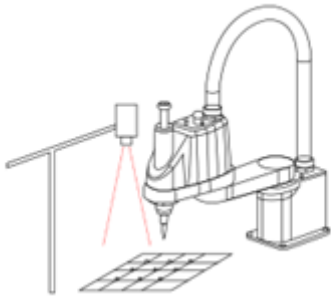
Depending on the installation mode of the camera, different calibration methods can be used. For detailed operations, see the 11.1.4 Eye-to-Hand Overlook Calibration section, 11.1.5 Eye-to-Hand Look Up Calibration section, 11.1.6 Eye-in-Hand J2 Calibration section, 11.1.7 Eye-in-Hand J4 Calibration section, 11.1.8 Eye-in-Hand J5 Calibration section, and 11.1.9 Eye-in-Hand J6 Calibration section.

11.1.3.2 Result Verification

The calibration results obtained by various calibration methods can be verified through programming. For details, see 11.1.10 Vision Process Calibration Examples and Result Verification.

11.1.4 Eye-to-Hand Overlook Calibration

The camera is installed above the calibration board, as shown below.



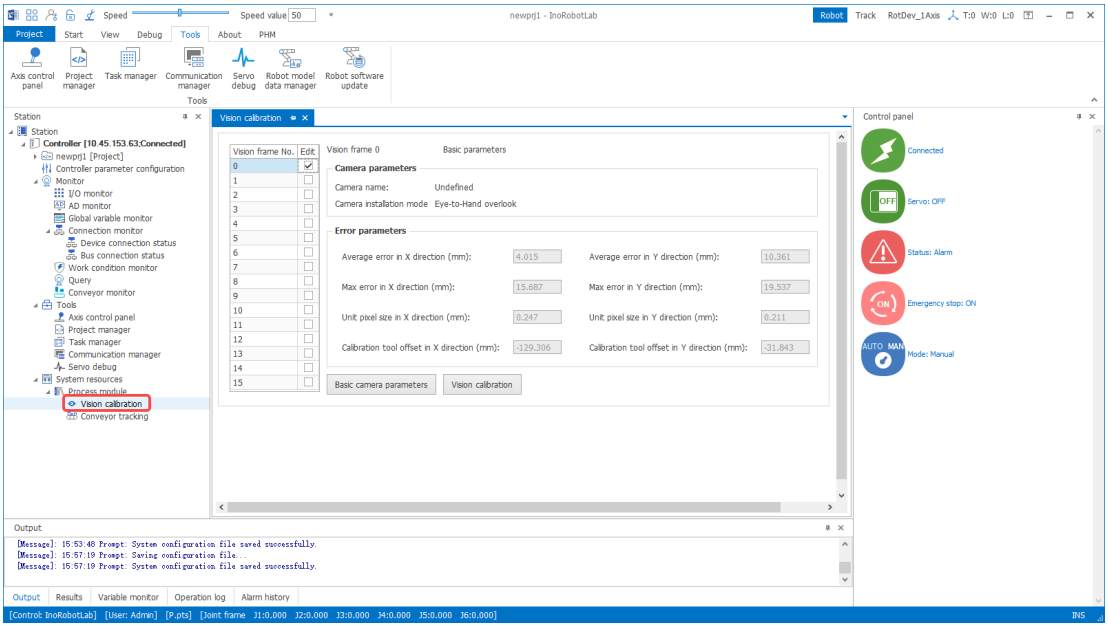
11.1.4.1 Installation Condition

- In the figure, a nine-square calibration board is placed directly below the camera. Keep the camera, robot, and calibration board on the same level. The calibration board contains nine marker points that need to be calibrated by the robot.
- Ensure that the nine marker points on the calibration board can be clearly imaged in the camera lens.
- Install a fixture at the end of the robot, which can be a tip fixture or a visually recognizable fixture.
- In case of the tip fixture, an additional 9-point calibration board is required; in case of the visually recognizable fixture, no additional device is required.

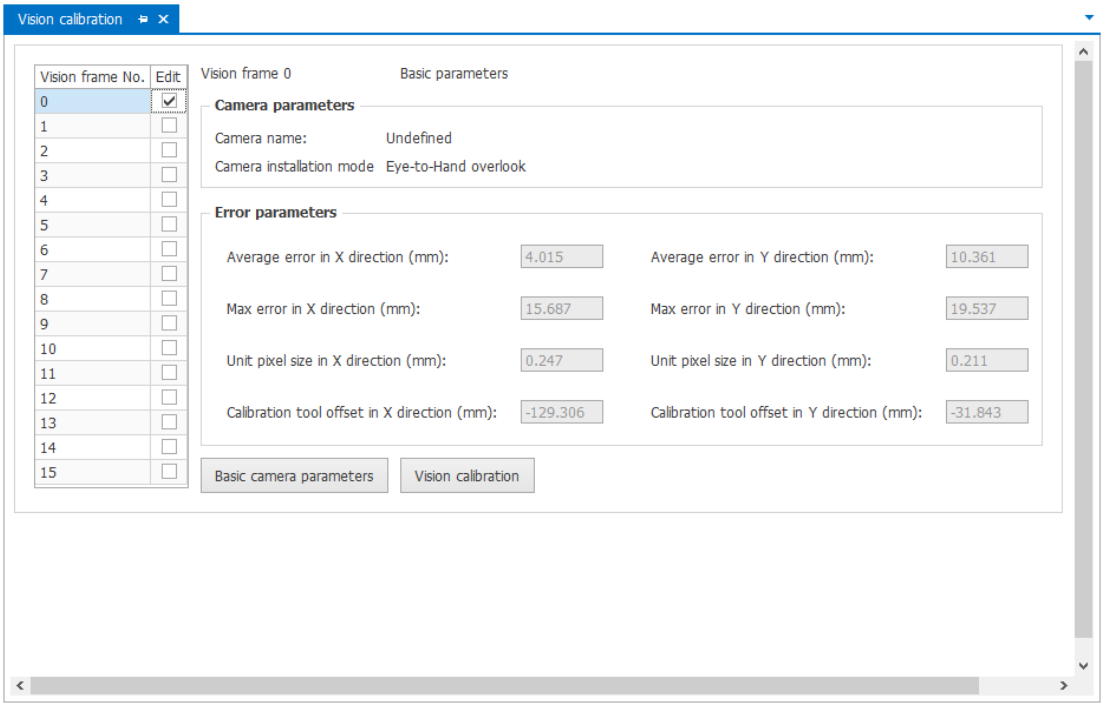
11.1.4.2 Calibration Steps

Step 1: Install the camera, adjust the robot position, place and adjust the calibration board, adjust the vision software, and create calibration templates, etc.

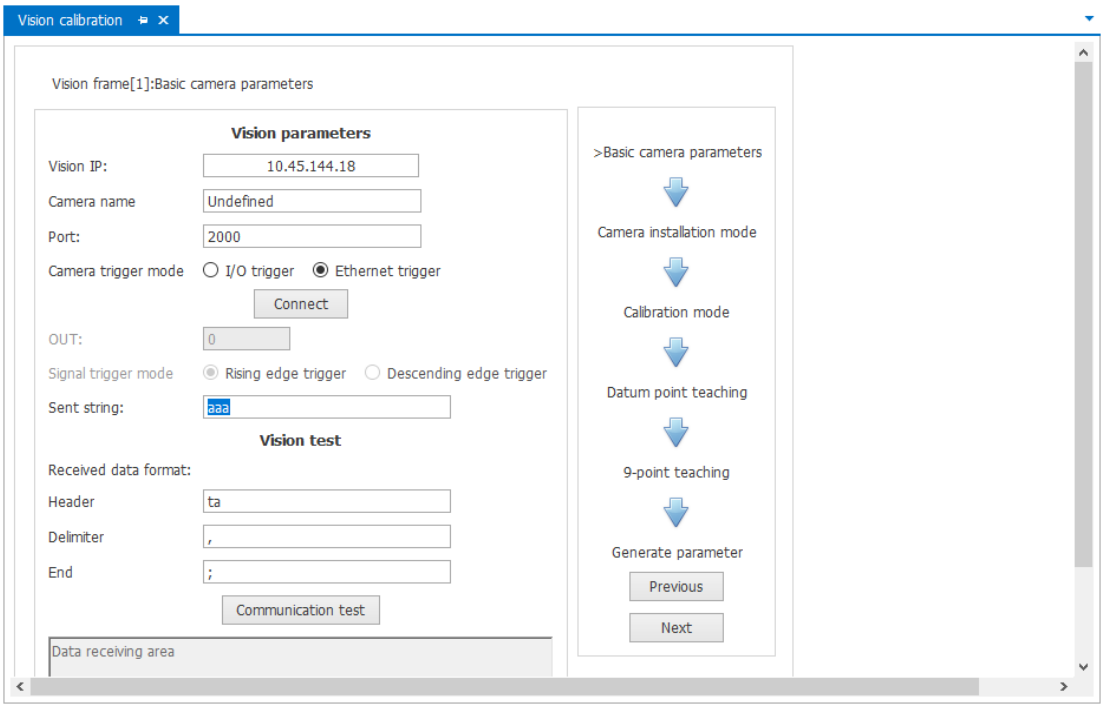
Step 2: Switch to the "Vision calibration" interface.



Select the frame to be calibrated, as shown below.



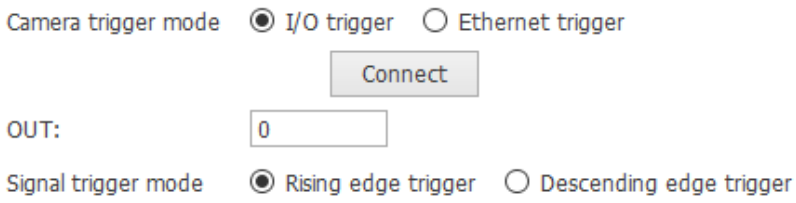
Step 3: Click the "Vision calibration" button to access the basic camera parameters setting page.



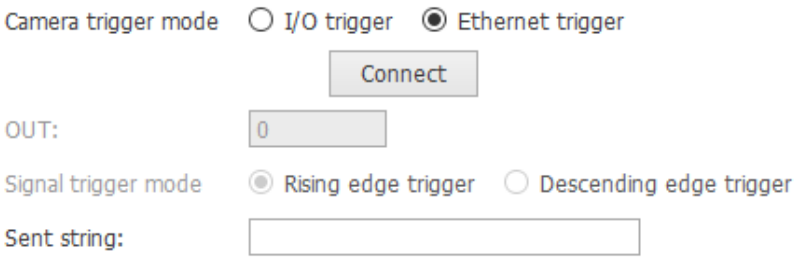
Set the vision (camera) IP address, port, and camera name (optional). After setting is complete, click "Connect" to establish communication.

Camera trigger mode includes "I/O trigger" and "Ethernet trigger". If you choose "I/O trigger", the camera is to be triggered to take photo by a level signal sent from a DO of the robot. In this case, the camera's trigger method should be set to hardware trigger. If you choose "Ethernet trigger", the camera (namely, server) is triggered to take photo by a string sent from the robot (namely, client) through the Ethernet network. In this case, the camera trigger method should be set to software trigger.

When you select I/O trigger, you need to configure the following parameters:



When you select Ethernet trigger, you need to set the string that triggers camera to take photos:



The format of the data received by the vision device includes header, delimiter and end mark.

If you do not need to communicate with vision device to obtain the camera coordinates, click "Next".

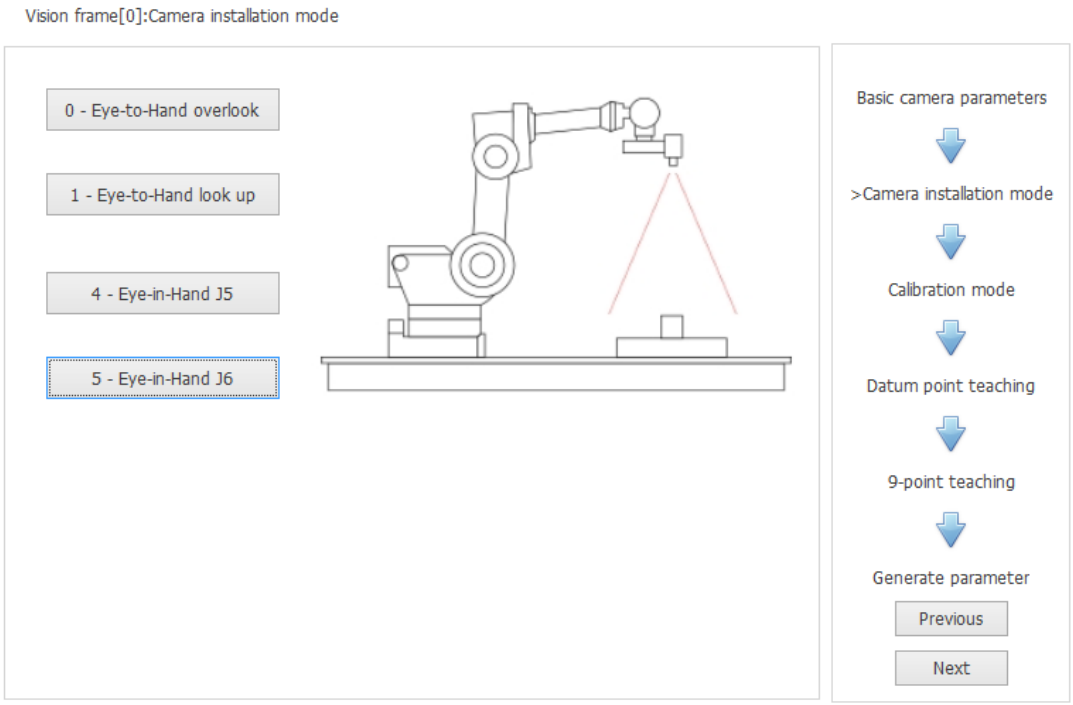
CAUTION

When the camera trigger mode is set to "Ethernet trigger", click "Communication test" to test the communication. If data sent by the peer has not been received in the receiving area, check the network cable and the connection between the peer and the robot controller.

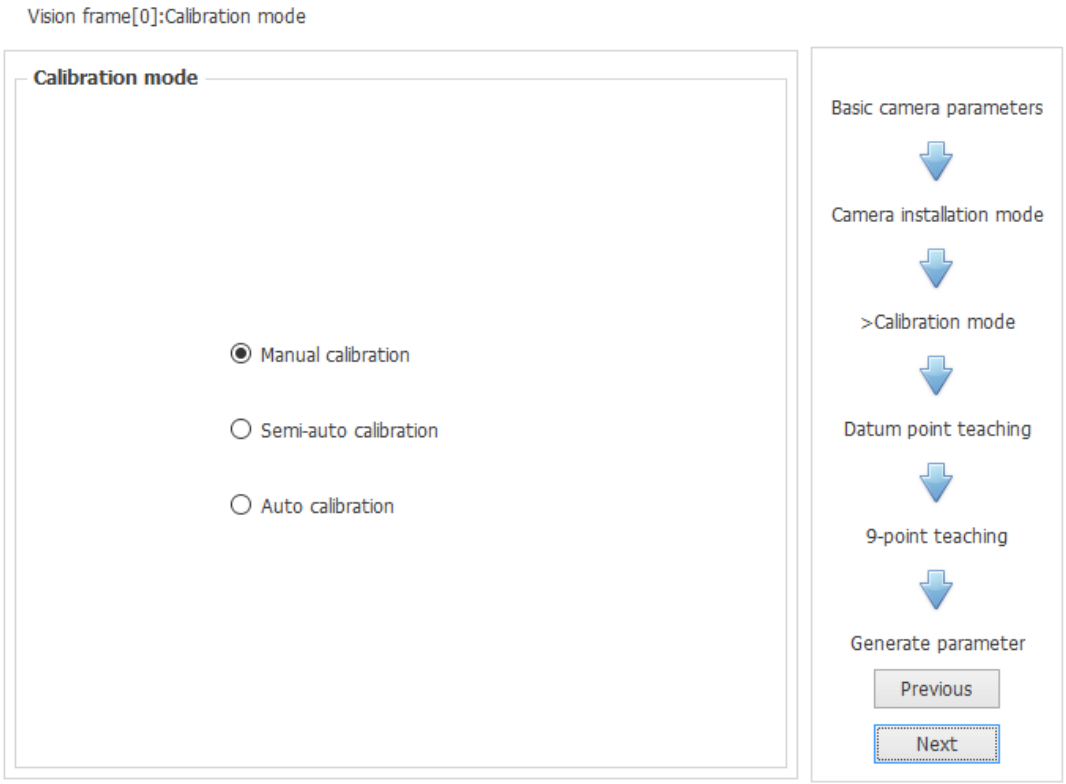
11.1.4.3 Camera Communication Test

It allows you to test communication between the robot controller and the vision software. The robot controller triggers a signal (I/O signal or a string) to the vision device. The string sent by the vision device is displayed in the data receiving area, allowing you to check whether the format of data sent is consistent with the specified format.

Step 1: Go to the camera installation mode selection page, select "0 - Eye-to-Hand overlook", and click "Next".



Step 2: Select the calibration mode and click "Next".



Three calibration modes are provided.

- Manual calibration: Manually teach the robot to calibrate the nine marker points in the calibration board and obtain the corresponding camera coordinates.
- Semi-auto calibration: Set three marker points on the pallet, from which nine marker points are generated. The robot automatically runs to the nine marker points and obtains the corresponding camera coordinates.

- Auto calibration: Generate nine marker points by teaching the robot with the center of the camera's field of view and setting the distance between nine points. The robot automatically runs to the nine marker points and obtains the corresponding camera coordinates.

Step 3: Go to the datum point acquisition page. Figure 1 shows the datum point teaching of SCARA robots, and Figure 2 shows the datum point teaching of 6-axis robots. After getting the points, click "Next".

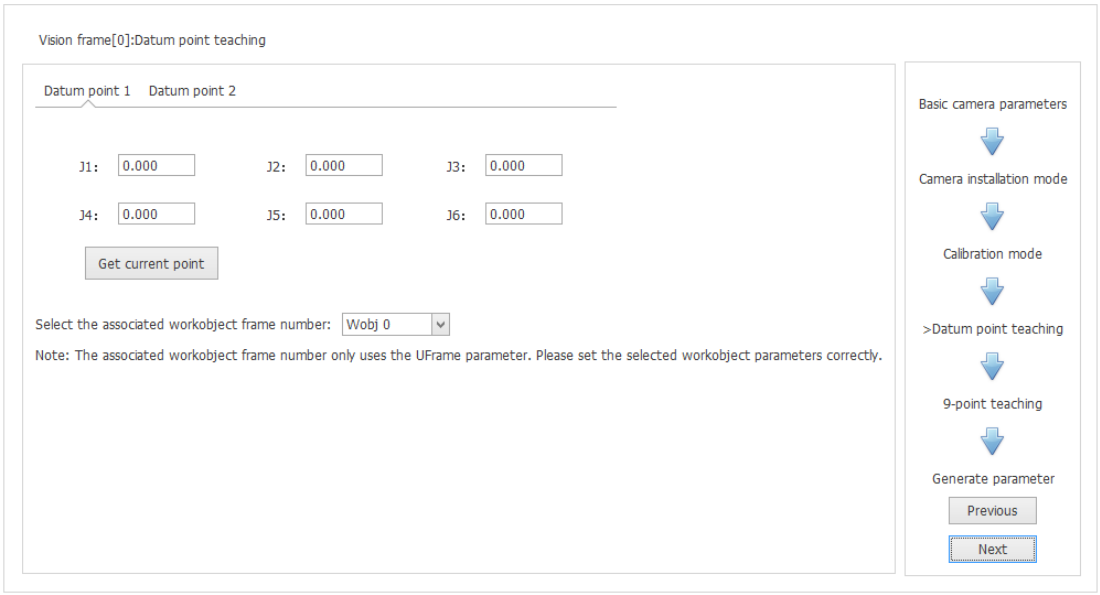


Figure 1

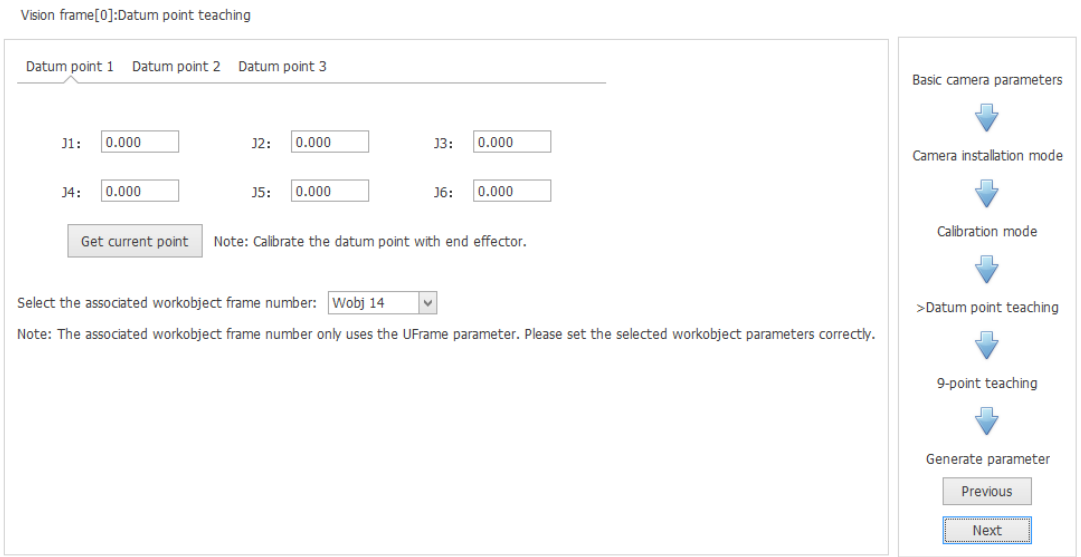


Figure 2

CAUTION

Two datum points are required for calibration of SCARA robots, and three for 6-axis robots. Set the datum point according to the calibration mode selected.

If the calibration fixture at the end of the robot is a tip fixture, teach the robot to align the tip of the fixture with the marker point at the center of the calibration board:

(1) The calibration of the SCARA robots requires two datum points. First, teach the robot to align the tip of the fixture with the marker point on the calibration board, and click "Get current point" to obtain the position

of the robot as datum point 1. Then rotate the J4 axis of the robot to a certain angle and re-align the tip of fixture with the marker point, and click "Get current point" to obtain the position of the robot as datum point 2.

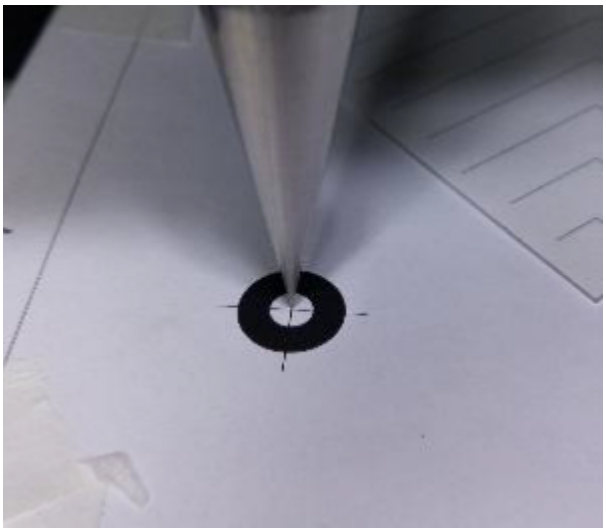
CAUTION

Do not select the two datum points by left and right hand rules.

(2) The calibration of 6-axis robots requires three datum points. For the calibration of tip tool of 6-axis robots, see 9.3.1 Tools. Align the tip of the fixture with the marker point at the center of the calibration board in three different orientations.

CAUTION

- If the user plane is not parallel to the robot's base plane, before performing vision calibration, first calibrate the tool frame of the fixture's tip at the end of the robot, and then use the tool frame to calibrate the user plane, that is, the user frame (UFrame) in the workobject frame. Use "Select associated workobject frame number" to select the number of the workobject frame calibrated by the user plane.
- Place the calibration board on the user plane. Keep the orientation difference between every two datum points as large as possible. For details, see the calibration method of the tool frame.



If the calibration fixture at the end of the robot is a visually recognizable template tool:

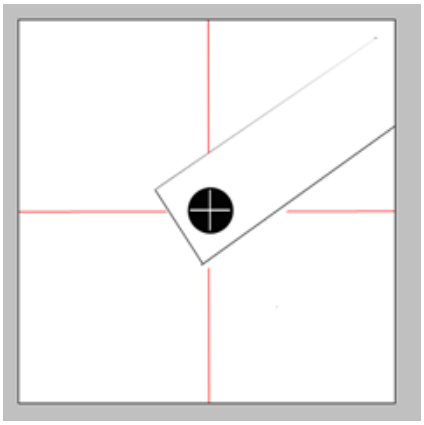
For a SCARA robot, teach it with two datum points in either of the following two ways:

- Align the center of the camera's field of view with a visually recognizable template tool, and click "Get current point" to obtain the position of the robot as datum point 1. Then rotate the J4 axis of the robot to a certain angle and re-align the template tool with the center of the camera's field of view, and click "Get current point" to obtain the position of the robot as datum point 2.
- These two points are obtained according to the tip calibration method of the SCARA robot. For the tip calibration method, see 9.3.1 Tools. Align the template tool with a certain pointed device in three different orientations. For details, see the calibration method of the tool frame of the SCARA robot.

CAUTION

Do not select the two datum points by left and right hand rules.

(3) The calibration of 6-axis robots requires three datum points. For the calibration of tip tool of 6-axis robots, see 9.3.1 Tools. Align the template tool with a certain pointed device in three different orientations. Keep the orientation difference between every two datum points as large as possible. For details, see the calibration method of the tool frame of the 6-axis robot.



Step 4: In the 9-point teaching page, get the nine points and click "Next".

(1) Manual calibration:

- If the calibration fixture at the end of the robot is a tip fixture:

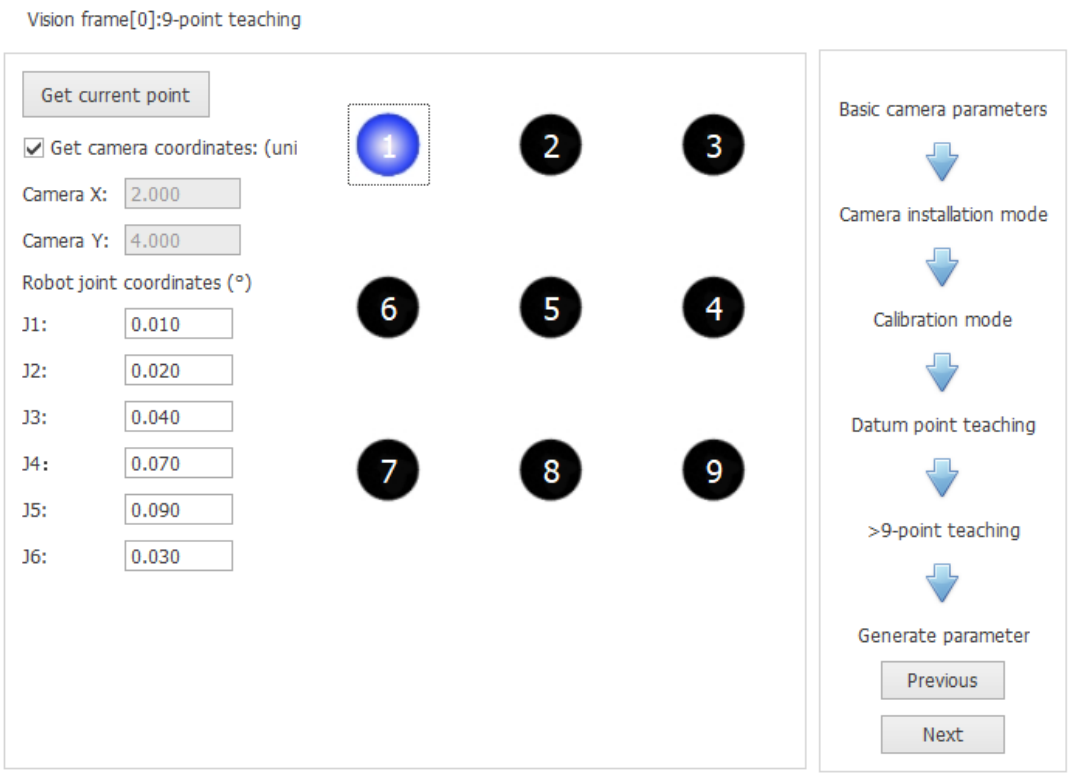
Place the calibration board on the user plane, trigger the camera to take a photo to obtain the pixel coordinates of the nine points on the calibration board at once. Teach the robot to align the tip of the fixture with one of the points on the calibration board. Turn off "Get camera coordinates" and manually enter the pixel coordinates of the point, and then click "Get current point" to get the current joint coordinates of the robot. In the similar way, enter the pixel coordinates of all nine points in sequence to get the joint coordinates of the robot for the nine points on the calibration board.

- If the calibration fixture at the end of the robot is a visually recognizable template tool:

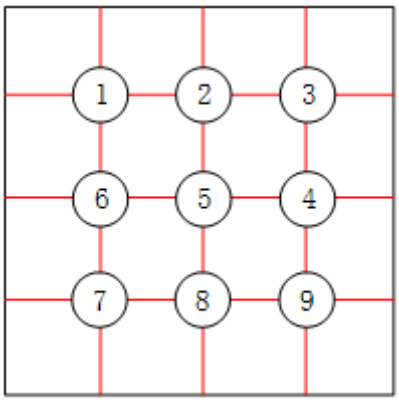
Teach the robot to move the template tool into the camera's field of view. The camera is triggered to take a photo to get the pixel coordinates of a point on the template tool. You can get the coordinates via communication, or manually enter the coordinates. Then click "Get current point" to get the current joint coordinates of the robot. In the similar way, get pixel coordinates of nine different points on the template tool in the camera's field of view, and the corresponding robot joint values.

CAUTION

The visually recognizable template tool should be kept relatively horizontal with the camera as much as possible, while the vertical distance between the two should remain unchanged.



When the visually recognizable template tool is used as the end calibration fixture of the robot, the points on the calibration fixture should cover 2/3 of the camera's field of view as much as possible to ensure visual calibration accuracy.



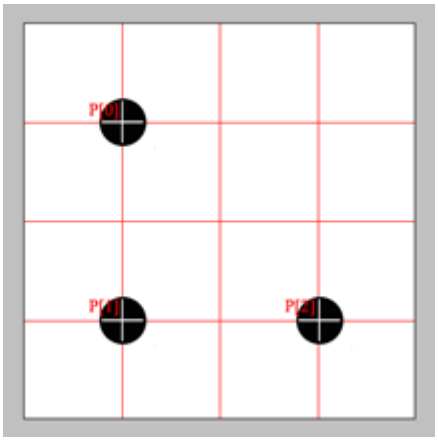
(2) Semi-auto calibration:

- If the calibration fixture at the end of the robot is a tip fixture:

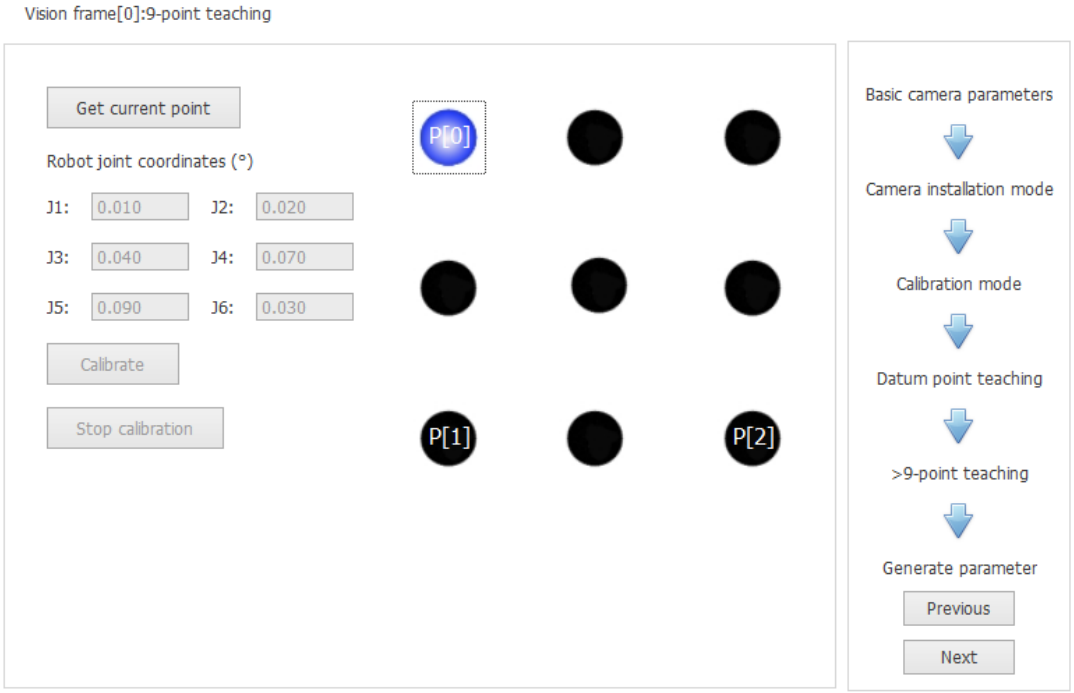
The calibration function is not available.

- If the calibration fixture at the end of the robot is a visually recognizable template tool:

After establishing communication between the robot and the camera, select three points in the camera's field of view and click "Get current point" to obtain P[0], P[1], and P[2], as shown below. Align the intersection points of the grid lines in the camera's field of view with the marker points on the calibration board in sequence, and complete the three-point pallet teaching.



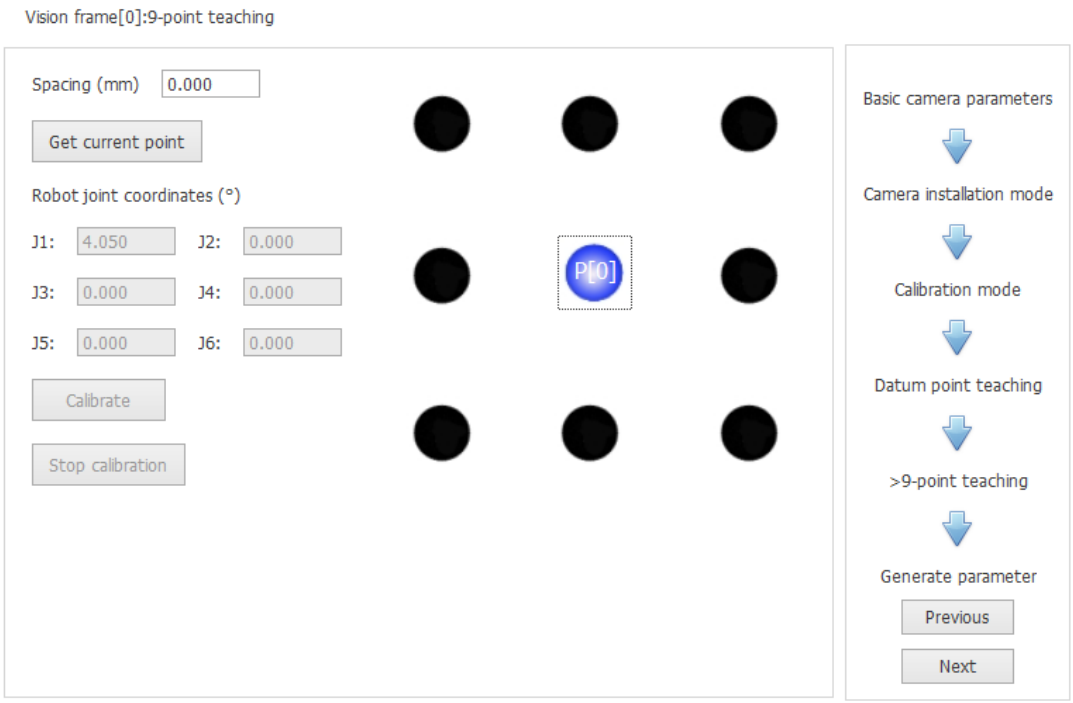
Click "Calibrate". The robot calculates the corresponding nine points using P[0], P[1], and P[2] as boundary points. At the same time, the robot automatically moves to the nine points in sequence, saving the robot coordinates as well as the corresponding pixel coordinates until the movement is completed.



(3) Auto calibration:

- If the calibration fixture at the end of the robot is a tip fixture, this function is not available.
- If the calibration fixture at the end of the robot is a visually recognizable template tool:

After establishing communication between the robot and the camera, teach the robot to move the template tool to the center of the camera's field of view and click "Get current point" to get the robot position P[0]. Set the "Spacing" (in mm) between points appropriately so that the nine points are all within the camera's field of view, then click "Calibrate". The robot automatically moves to the nine points in sequence, saving the robot coordinates as well as the corresponding pixel coordinates until the movement is completed.

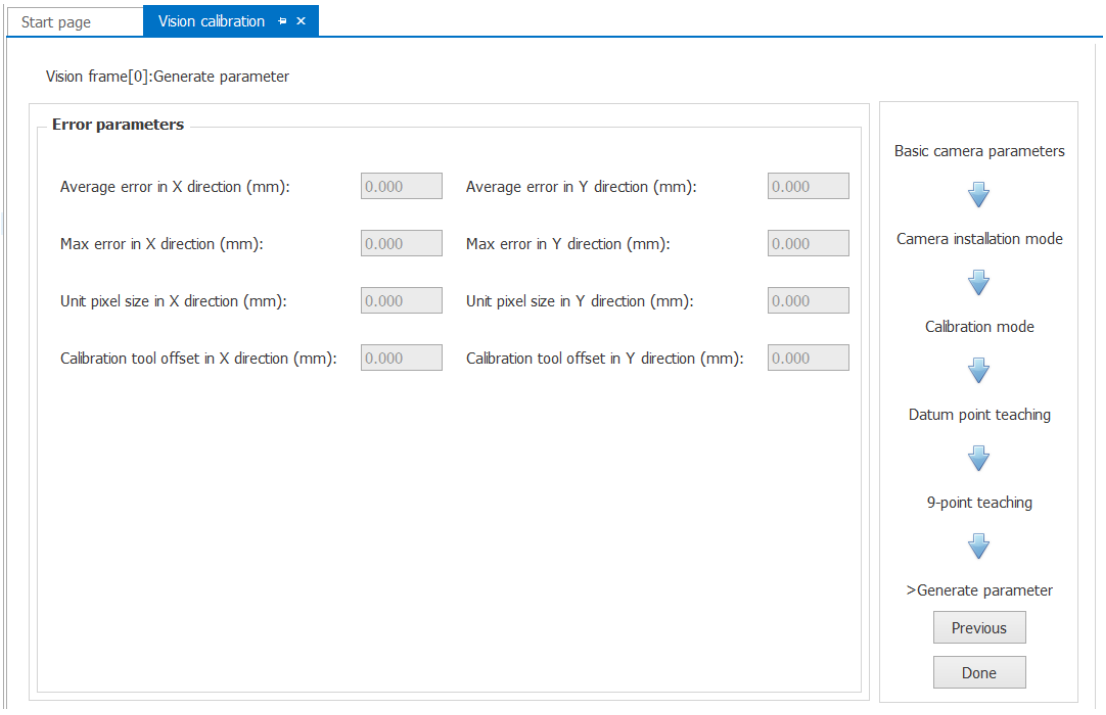


CAUTION

The setting value of "Spacing" should be reasonably selected to ensure that the robot's motion range within the camera's field of view can cover about 2/3 of the camera's center field of view, while also ensuring that the robot's motion range does not exceed the camera's field of view.

Step 5: On the teaching point list, check whether each of the position coordinates and pixel coordinates is abnormal. You can also double-click the point in the list to modify the point data. Click "Next" when you are done.

Step 6: The system calculates and generates the calibration results, as shown below.



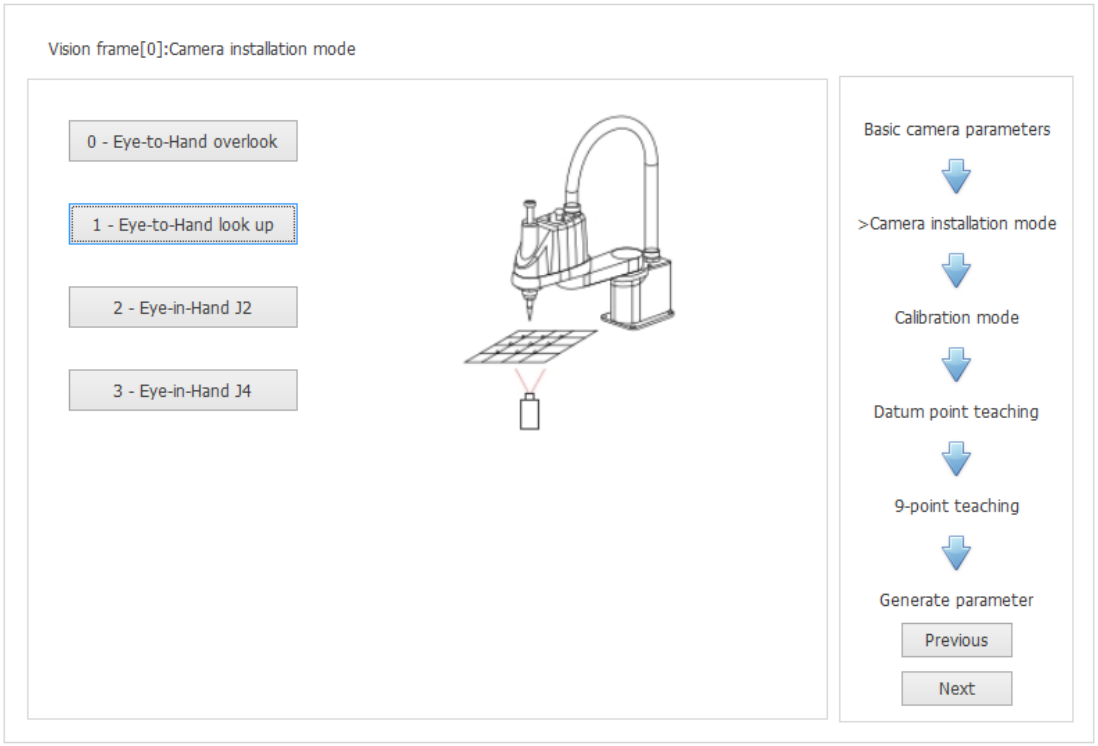
The results include average error in the X direction, average error in Y direction, maximum error in X direction, maximum error in Y direction, unit pixel size in X direction, unit pixel size in Y direction, calibration tool offset in X direction, and calibration tool offset in Y direction.

The calibration tool offset in X direction and calibration tool offset in Y direction indicate the tool parameters without rotation direction for the calibration tool.

Step 7: Click "Done" to complete the vision calibration.

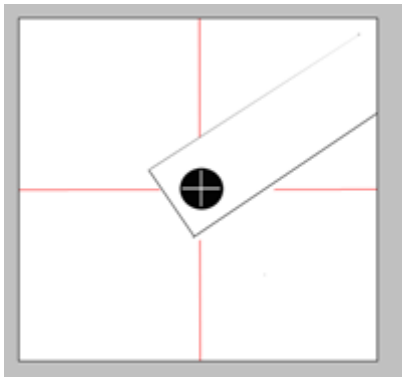
11.1.5 Eye-to-Hand Look Up Calibration

The camera is installed below the calibration board, as shown below.



11.1.5.1 Installation Condition

The camera is mounted directly below the robot, facing upwards. Keep the camera and robot on the same level. Install a fixture at the end of the robot. The fixture is generally a visually recognizable template tool. Adjust the camera focus and robot height to ensure that the marker points on the fixture can be clearly identified in the camera's field of view. If conditions permit, the process of eye-to-hand overlook calibration is the same with that of eye-to-hand look-up calibration.



11.1.5.2 Calibration Steps

Steps 1 to 5: Same as Steps 1 to 3 in 11.1.4.2 Calibration Steps and Steps 1 to 2 in 11.1.4.3 Camera Communication Test of "Eye-to-Hand Overlook Calibration".

Step 6: For SCARA robots, move the marker point on the fixture to the center of the camera's field of view to obtain datum point 1. Adjust the orientation of the robot and rotate direction of the fixture, and then move the marker point to the center of the camera's field of view to obtain datum point 2. Note that the calibration of 6-axis robots requires three datum points and the selection of associated workobject frame.

CAUTION

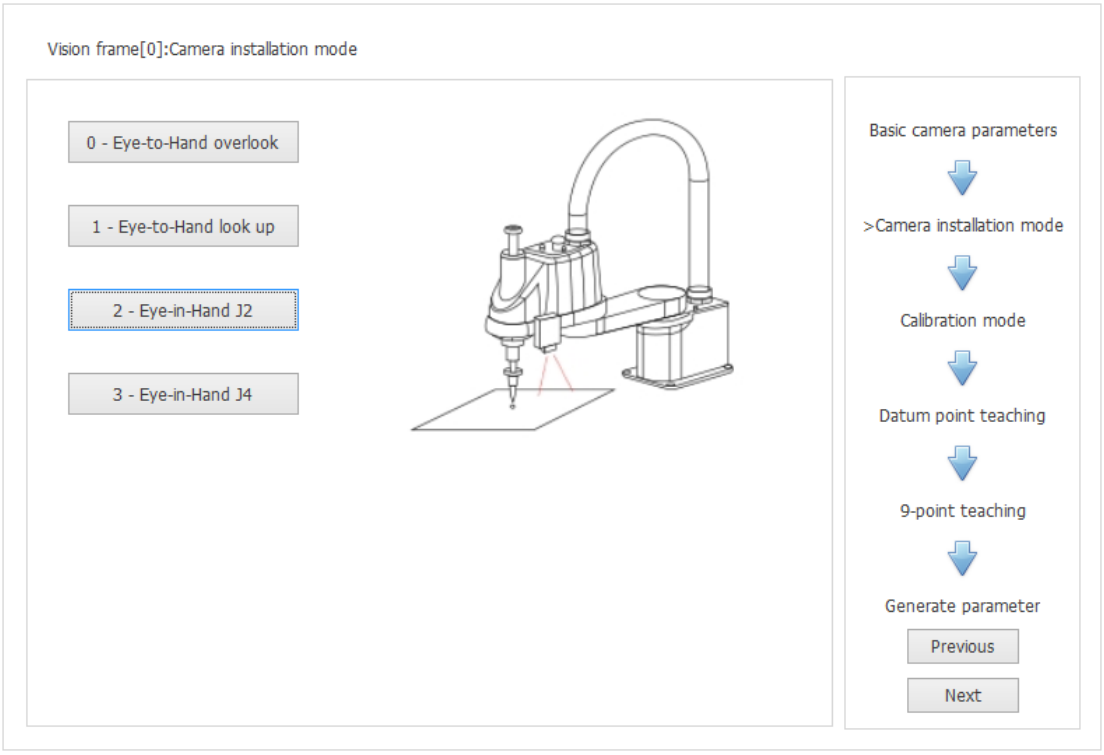
- When performing datum point teaching, keep the camera and calibration board in relative parallel relationship. The orientation change between each two datum points must be larger than 30°.
- Ensure that the plane of the selected datum points is consistent with the workobject frame associated with calibration.

Step 7: Same as step 4 in 11.1.4.3 Camera Communication Test of "Eye-to-Hand Overlook Calibration". The process of manual point acquisition is same as step 3. Operate the robot and move the fixture to obtain nine points distributed in the camera's field of view evenly, and obtain corresponding vision coordinates at the same time. You can obtain the vision coordinates automatically or by manual input.

Steps 8 to 10: Same as steps 5 to 7 in 11.1.4.3 Camera Communication Test of "Eye-to-Hand Overlook Calibration".

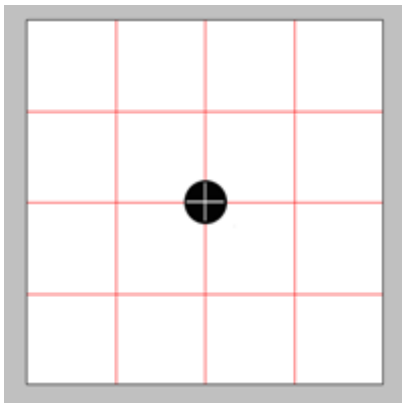
11.1.6 Eye-in-Hand J2 Calibration

The camera is mounted looking downwards on the robot J2 axis, as shown below. This calibration mode is applicable to SCARA robots only.



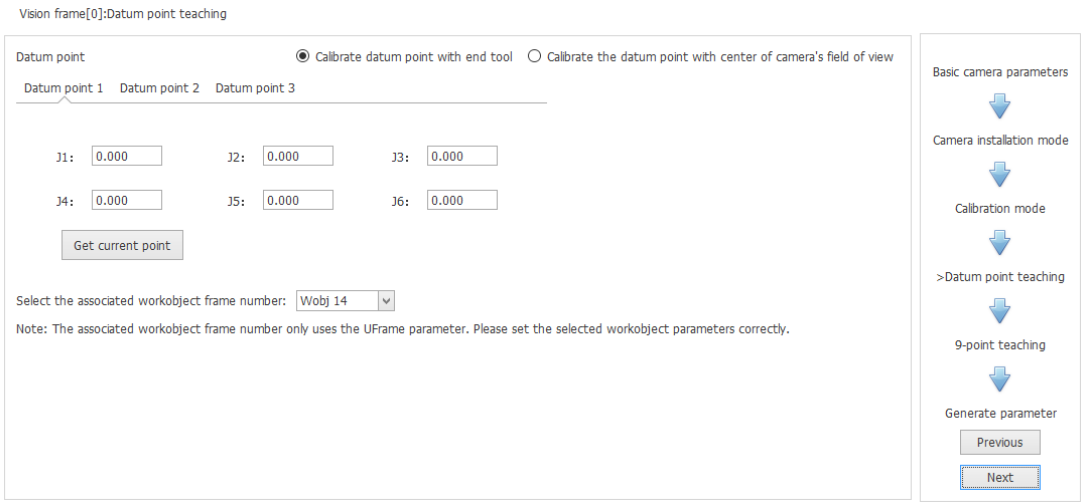
11.1.6.1 Installation Condition

The camera is mounted looking downwards on the robot J2 axis. Keep the camera plane parallel to the horizontal plane. As shown below, the black marker point represents the marker point on the calibration board, the red lines represent the grid lines in the camera's field of view, and the intersection of the grid lines is the required marker point. The calibration board used can contain only one visually recognizable marker point. The 9-point calibration board must not be used.



11.1.6.2 Calibration Steps

Steps 1 to 5: Same as Steps 1 to 5 in 11.1.5.2 Calibration Steps of "Eye-to-Hand Look Up Calibration".



If you choose to calibrate the datum point by "Calibrate datum point with end tool", you need to install a tip fixture at the end of the robot. It is needed to teach the robot with two datum points. Teach the robot to align the tip of the fixture with the marker point on the calibration board, and click "Get current point" to obtain the position of the robot as datum point 1.

If the tip fixture is an eccentric tool, rotate the J4 axis of the robot to a certain angle and re-align the tip of fixture with the marker point. If the tip fixture is a non-eccentric tool, adjust the robot's arm type and re-align the tip of fixture with the marker point, and click "Get current point" to obtain the position of the robot as datum point 2.

If you choose to calibrate the datum point by "Calibrate the datum point with center of camera's field of view", there is no need to install a tip fixture at the end of the robot. It is needed to teach the robot with two datum points. Teach the robot to align the tip of the fixture with the center of the camera's field of view installed on the J2 axis with the marker point on the calibration board, and click "Get current point" to obtain the position of the robot as datum point 1.

Adjust the robot's arm type and re-align the center of the camera's field of view with the marker point, and click "Get current point" to obtain the position of the robot as datum point 2.

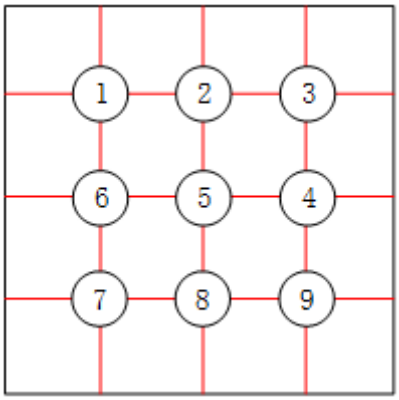
CAUTION

- If you use the center of the camera's field of view to calibrate the datum point, select the option "Calibrate the datum point with center of camera's field of view". The tool parameters displayed after calibration represent the deviation of the center of the camera's field of view from the end of the J2 axis.
- If you use the end tool of the robot to calibrate the datum point, select the option "Calibrate datum point with end tool". The tool parameters displayed after calibration are those of the tool installed at the end of the robot.
- If you choose to use the camera as a tool for manual calibration, then do not use the end tool.

Step 6. Click "Next" to enter the 9-point calibration page.

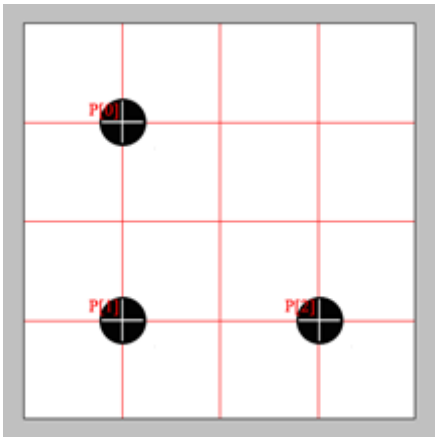
- Manual calibration

Move the robot so that the marker point on the calibration board appear sequentially at different positions within the camera's field of view, as shown below. Make sure the motion range of the maker point covers 2/3 of the camera's field of view as much as possible to ensure visual calibration accuracy. Obtain the robot coordinates and pixel coordinates when the robot moves to each point. The camera coordinates can be automatically obtained or manually entered.



- Semi-auto calibration

After establishing communication between the robot and the camera, select three points in the camera's field of view and click "Get current point" to obtain P[0], P[1], and P[2], as shown below. Align the intersection points of the grid lines in the camera's field of view with the marker points on the calibration board in sequence, and complete the three-point pallet teaching.



Click "Calibrate". The robot calculates the corresponding nine points using P[0], P[1], and P[2] as boundary points. At the same time, the robot automatically moves to the nine points in sequence, saving the robot coordinates as well as the corresponding pixel coordinates until the movement is completed.

CAUTION

- The robot calculates the 9 points on the pallet based on the end of the robot, while the camera is installed on the J2 axis of the robot.
- If three datum points are calibrated according to the grid in the camera's field of view, the actual operation of the robot may not be based on the 9 points in the camera's field of view, but may be a parallelogram which may exceed the camera's field of view. Therefore, when calibrating the three datum points, calibrate them in the middle of the field of view.

- Auto calibration

After establishing communication between the robot and the camera, teach the robot to align the center of the camera's field of view with the marker point on the calibration board and click "Get current point" to get the robot position P[0]. Set the "Spacing" (in mm) between points appropriately so that the nine points are all within the camera's field of view, then click "Calibrate". The robot automatically moves to the nine points in sequence, saving the robot coordinates as well as the corresponding pixel coordinates until the movement is completed.

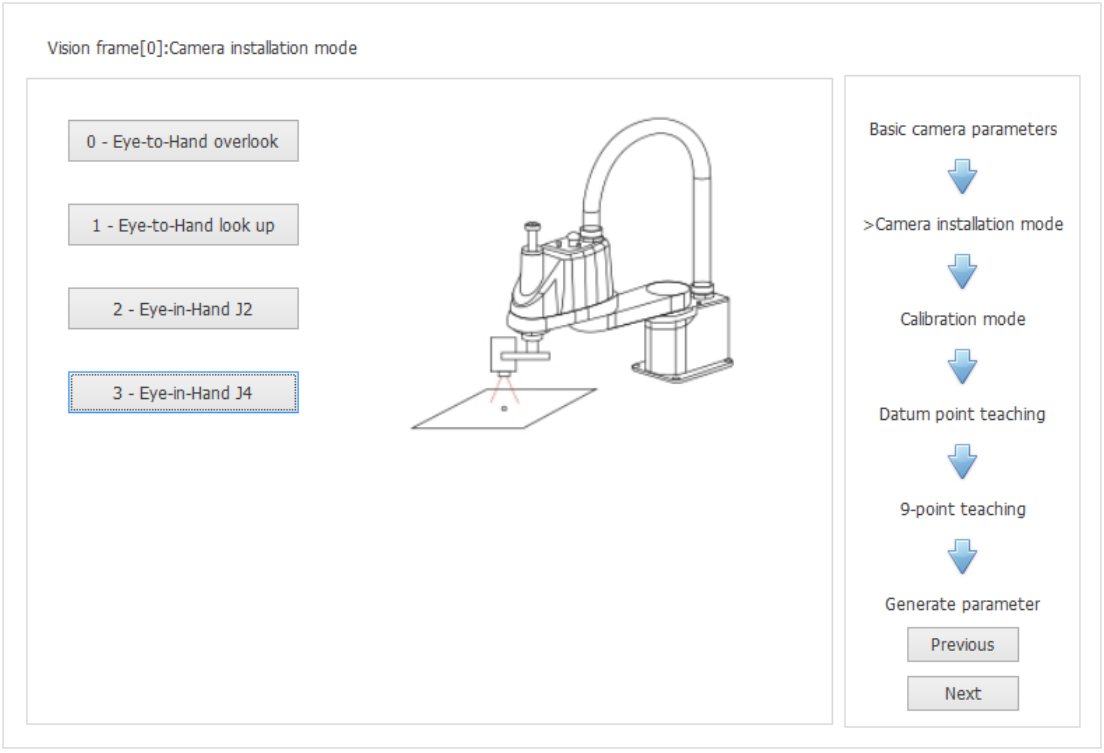
CAUTION

The setting value of "Spacing" should be reasonably selected to ensure that the robot's motion range within the camera's field of view can cover about 2/3 of the camera's center field of view, while also ensuring that the robot's motion range does not exceed the camera's field of view.

Steps 7 to 9: Same as steps 5 to 7 in 11.1.4.3 Camera Communication Test of "Eye-to-Hand Overlook Calibration".

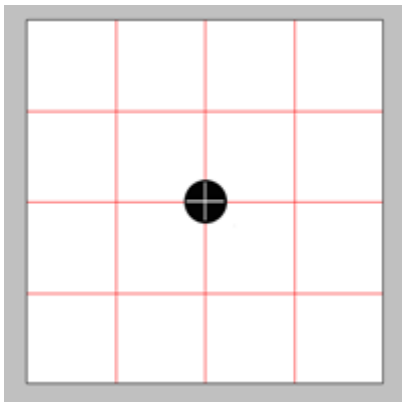
11.1.7 Eye-in-Hand J4 Calibration

The camera is mounted looking downwards on the robot J4 axis, as shown below.



11.1.7.1 Installation Condition

The camera is mounted looking downwards on the robot J4 axis. Keep the camera plane parallel to the horizontal plane. As shown below, the black marker point represents the marker point on the calibration board, the red lines represent the grid lines in the camera's field of view, and the intersection of the grid lines is the required marker point. The calibration board used can contain only one visually recognizable marker point. The 9-point calibration board must not be used.



11.1.7.2 Calibration Steps

Same as those of 11.1.6.2 Calibration Steps of "Eye-in-Hand J2 Calibration".

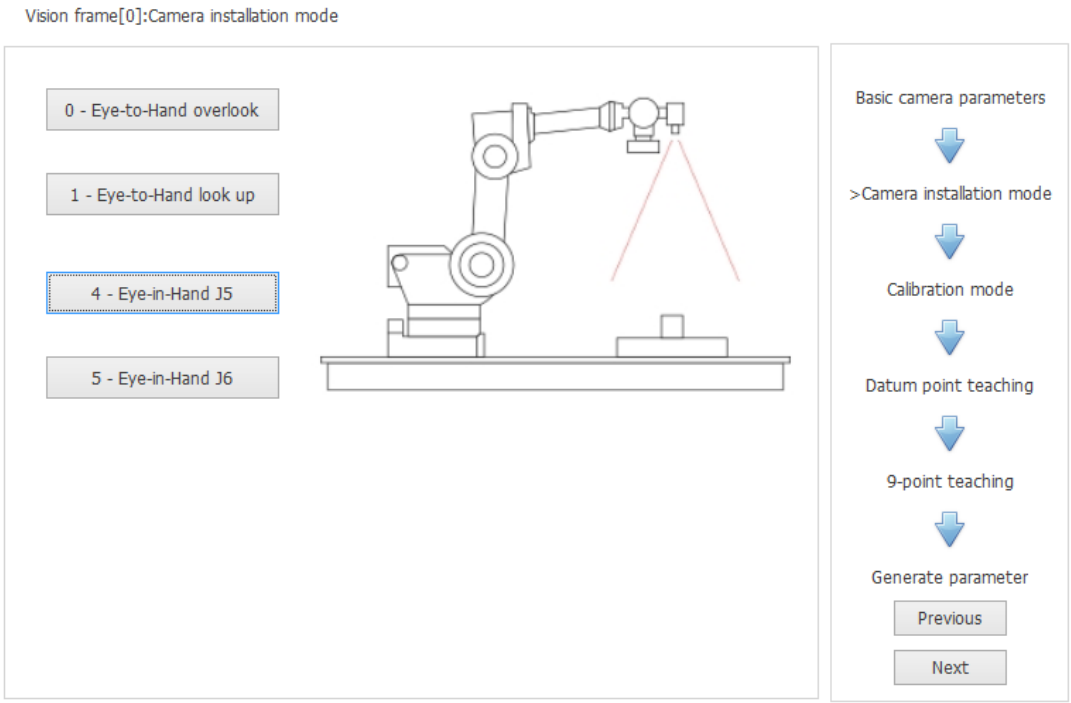
CAUTION

When calibrating the datum points, there is no need to choose the calibration method on the interface. You can calibrate the datum points in either of the following two ways:

- Calibrate the datum points using the center of the camera's field of view.
- Install a tip fixture at the end of the robot and calibrate the datum points using such fixture.
  - If you use the center of the camera's field of view to calibrate the datum point, the tool parameters displayed after calibration represent the deviation of the center of the camera's field of view from the end of the J4 axis.
  - If you use the end tool of the robot to calibrate the datum point, the tool parameters displayed after calibration are those of the tool installed at the end of the robot.

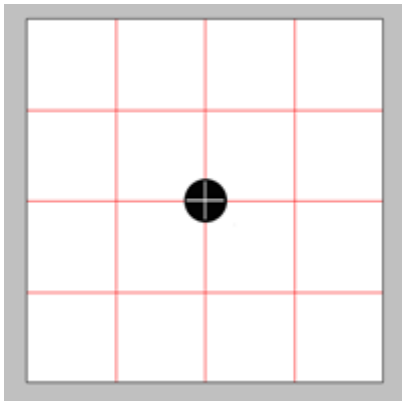
11.1.8 Eye-in-Hand J5 Calibration

The camera is mounted looking downwards on the robot J5 axis, as shown below.



11.1.8.1 Installation Condition

The camera is mounted looking downwards on the robot J5 axis. Keep the camera plane parallel to the horizontal plane. No calibration fixture is provided. As shown below, the black marker point represents the marker point on the calibration board, the red lines represent the grid lines in the camera's field of view, and the intersection of the grid lines is the required marker point. The calibration board used can contain only one visually recognizable marker point. The 9-point calibration board must not be used.

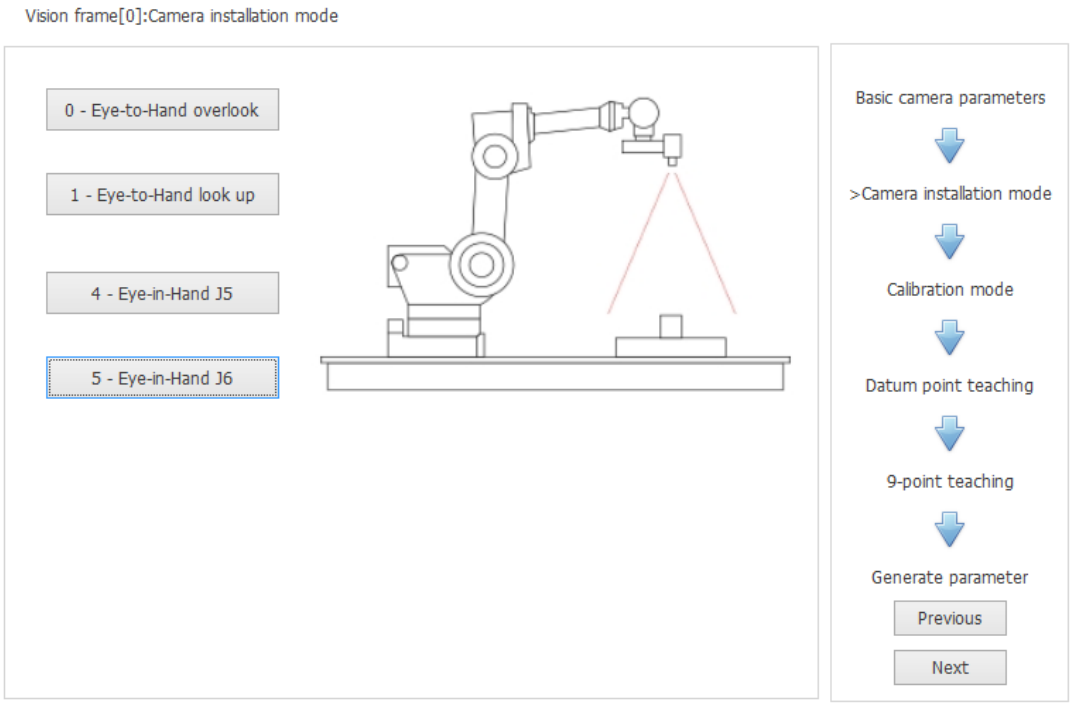


11.1.8.2 Calibration Steps

Same as those of 11.1.9.2 Calibration Steps of "Eye-in-Hand J6 Calibration".

11.1.9 Eye-in-Hand J6 Calibration

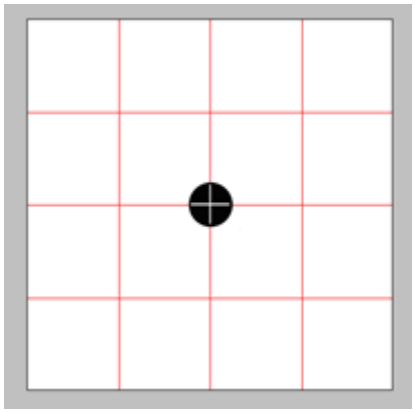
The camera is mounted looking downwards on the robot J6 axis, as shown below.



11.1.9.1 Installation Condition

The camera is mounted looking downwards on the robot J6 axis. Keep the camera plane parallel to the horizontal plane. As shown below, the black marker point represents the marker point on the calibration board, the red lines represent the grid lines in the camera's field of view, and the intersection of the grid lines is the required marker point. The calibration board used can contain only one visually recognizable marker point. The 9-point calibration board must not be used.

A tip fixture must be installed at the end of the robot.



11.1.9.2 Calibration Steps

Step 1: Install a tip tool at the end of the robot, and place a calibration board that contains only one visually recognizable marker point on the user plane.

Step 2:If the user plane is not relatively parallel to the robot's base frame, it is necessary to calibrate the user plane using the calibrated tip tool. That is, calibration of both tip tool and user plane is required.

If the user plane is relatively parallel to the robot's base frame, calibration of tip tool and user plane is not required.

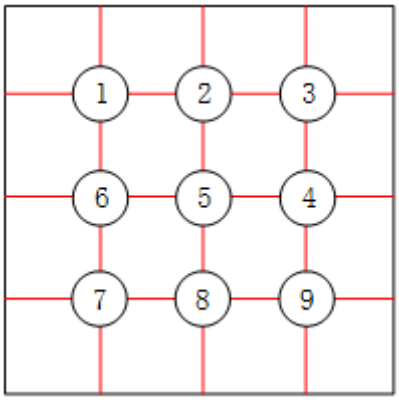
Steps 3 to 7: Same as Steps 1 to 5 in 11.1.5.2 Calibration Steps of "Eye-to-Hand Look Up Calibration".

Step 8: Calibrate the datum point by "Calibrate datum point with end tool" method. The calibration requires three datum points. For the calibration of tip tool of 6-axis robots, see 9.3.1 Tools. Align the tip of the fixture with the marker point at the center of the calibration board in three different orientations.

Step 9. Click "Next" to enter the 9-point calibration page.

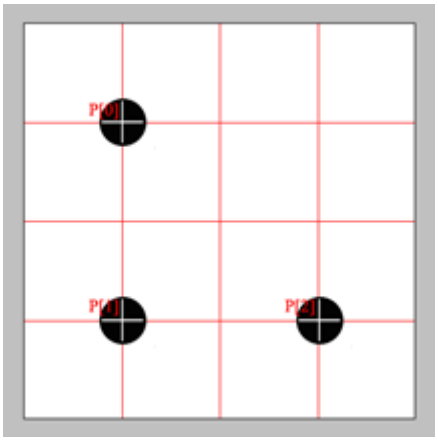
- Manual calibration

Move the robot so that the marker point on the calibration board appear sequentially at different positions within the camera's field of view, as shown below. Make sure the motion range of the maker point covers 2/3 of the camera's field of view as much as possible to ensure visual calibration accuracy. Obtain the robot coordinates and pixel coordinates when the robot moves to each point. The camera coordinates can be automatically obtained or manually entered.



- Semi-auto calibration

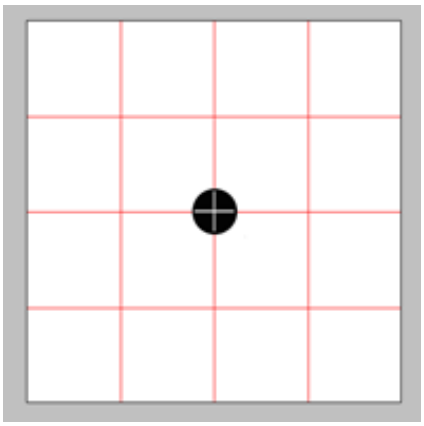
After establishing communication between the robot and the camera, select three points in the camera's field of view and click "Get current point" to obtain P[0], P[1], and P[2], as shown below. Align the intersection points of the grid lines in the camera's field of view with the marker points on the calibration board in sequence, and complete the three-point pallet teaching.



Click "Calibrate". The robot calculates the corresponding nine points using P[0], P[1], and P[2] as boundary points. At the same time, the robot automatically moves to the nine points in sequence, saving the robot coordinates as well as the corresponding pixel coordinates until the movement is completed.

- Auto calibration

After establishing communication between the robot and the camera, teach the robot to align the center of the camera's field of view with the marker point on the calibration board and click "Get current point" to get the robot position P[0]. Set the "Spacing" (in mm) between points appropriately so that the nine points are all within the camera's field of view, then click "Calibrate". The robot automatically moves to the nine points in sequence, saving the robot coordinates as well as the corresponding pixel coordinates until the movement is completed.



CAUTION

The setting value of "Spacing" should be reasonably selected to ensure that the robot's motion range within the camera's field of view can cover about 2/3 of the camera's center field of view, while also ensuring that the robot's motion range does not exceed the camera's field of view.

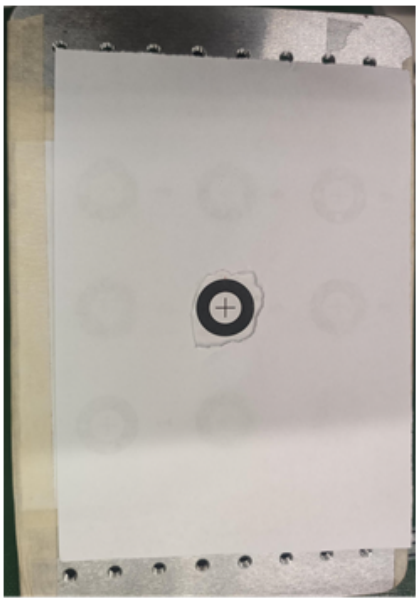
Steps 10 to 12: Same as steps 5 to 7 in 11.1.4.3 Camera Communication Test of "Eye-to-Hand Overlook Calibration".

11.1.10 Vision Process Calibration Examples and Result Verification

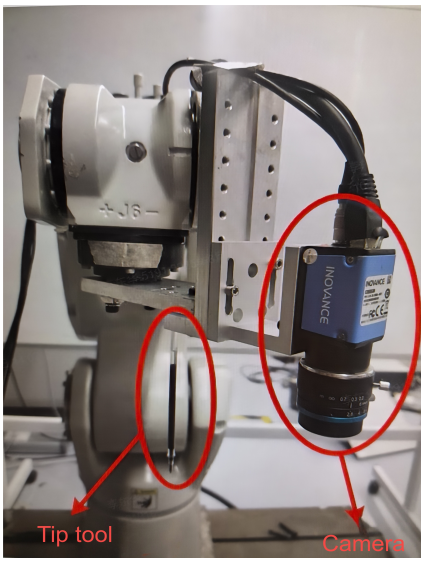
The following is an example of the calibration process with the dynamic camera of a 6-axis robot in the condition that the user plane is not level.

- Model: IRS311-3-60TS5-01_01740272 6-axis robot.

- Camera mounting method: On J6 axis of the robot.
- Calibration method: Semi-auto calibration.
- Tools: Camera, tip tool, calibration board that contains only one visually recognizable marker point.
- User plane: The tilted calibration board is used as the user plane.



One visually recognizable marker point

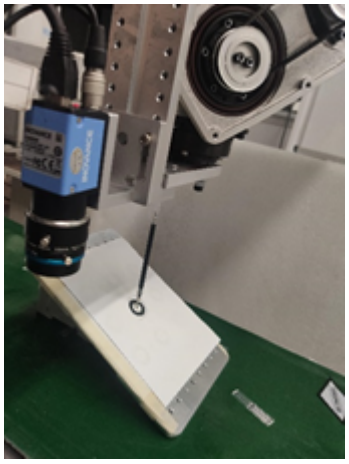


Device diagram

Step 1: Install the camera, adjust the robot position, place and adjust the calibration board, adjust the vision software, and create calibration templates, etc.

Step 2: Calibrate the tool frame of the tip tool.

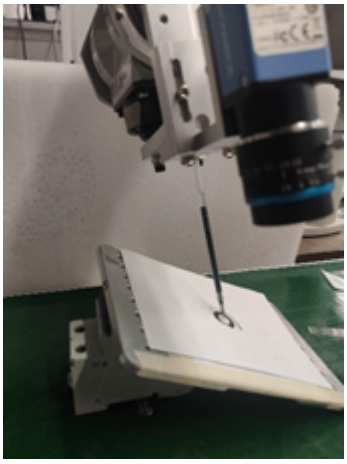
For the calibration of tip tool, see 9.3.1 Tools. Align the tip of the fixture with a fixed point in three different orientations. It is recommended to place the calibration board on the user plane that needs to be calibrated, and then use the marker point on the calibration board as a fixed point to calibrate the tip tool. Record the three positions of the robot for use in datum point calibration in the subsequent vision calibration process. The tool number of the tip tool used is Tool15.



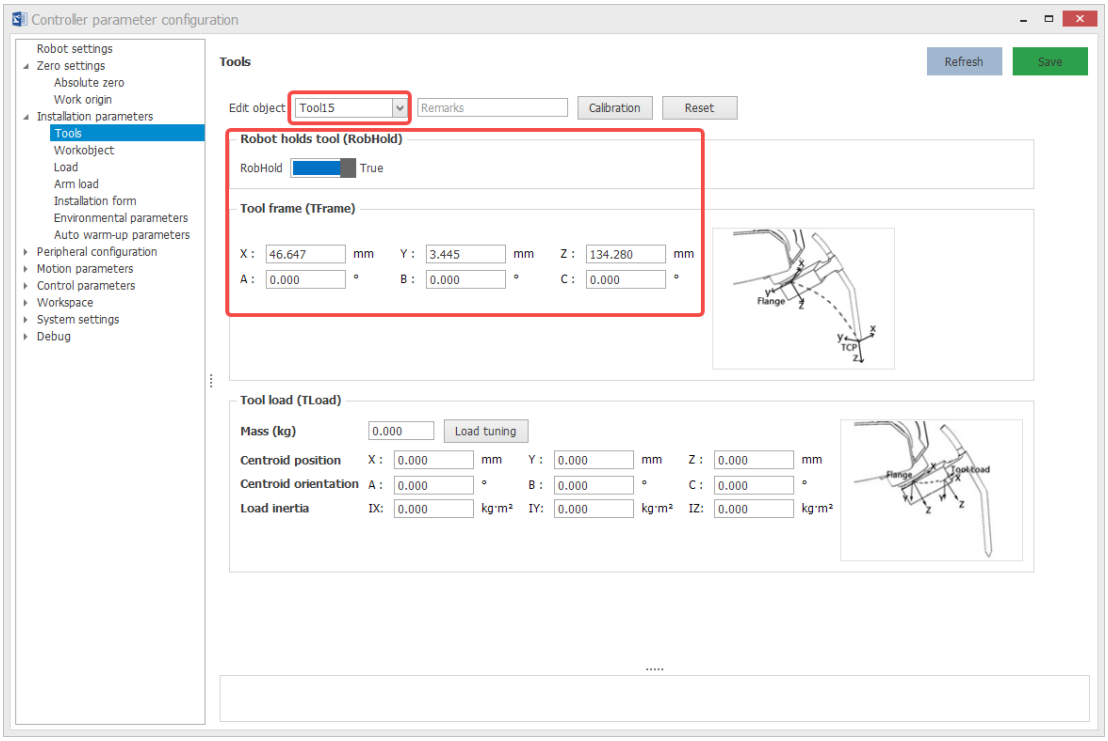
Teaching point 1



Teaching point 2



Teaching point 3



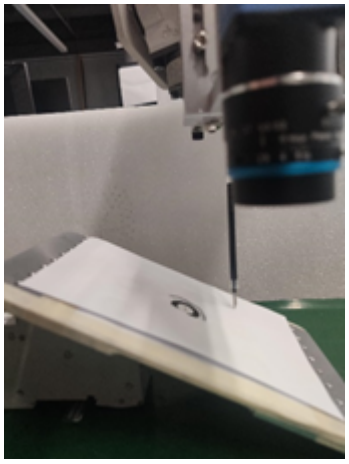
Tool frame

CAUTION

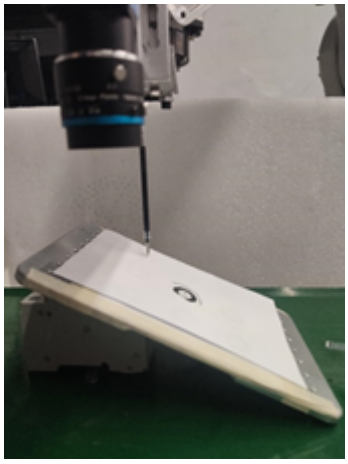
When calibrating the tip tool, you can also use a point other than the marker point on the calibration board as the fixed point. However, it is necessary to align the tip tool with the marker point on the calibration board as the fixed point in the subsequent datum point calibration process. Therefore, it is recommended to directly use the marker point on the calibration board as the fixed point for calibrating the pointed tool.

Step 3: Calibrate the workobject frame of the user plane.

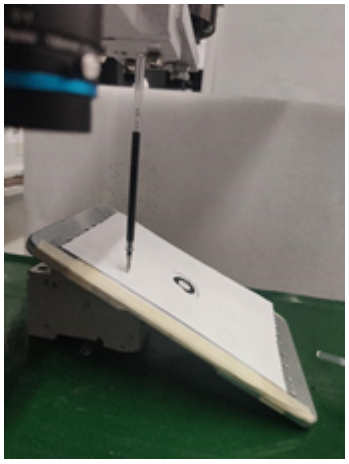
For the calibration method of the workobject frame, see section 9.3.2 Workobject. Use the calibrated tip tool to calibrate the workobject frame of the user plane. When calibrating the workobject frame, you only need to calibrate the user frame (Uframe). In the example, the number of the workobject frame of the user plane is Wobj15.



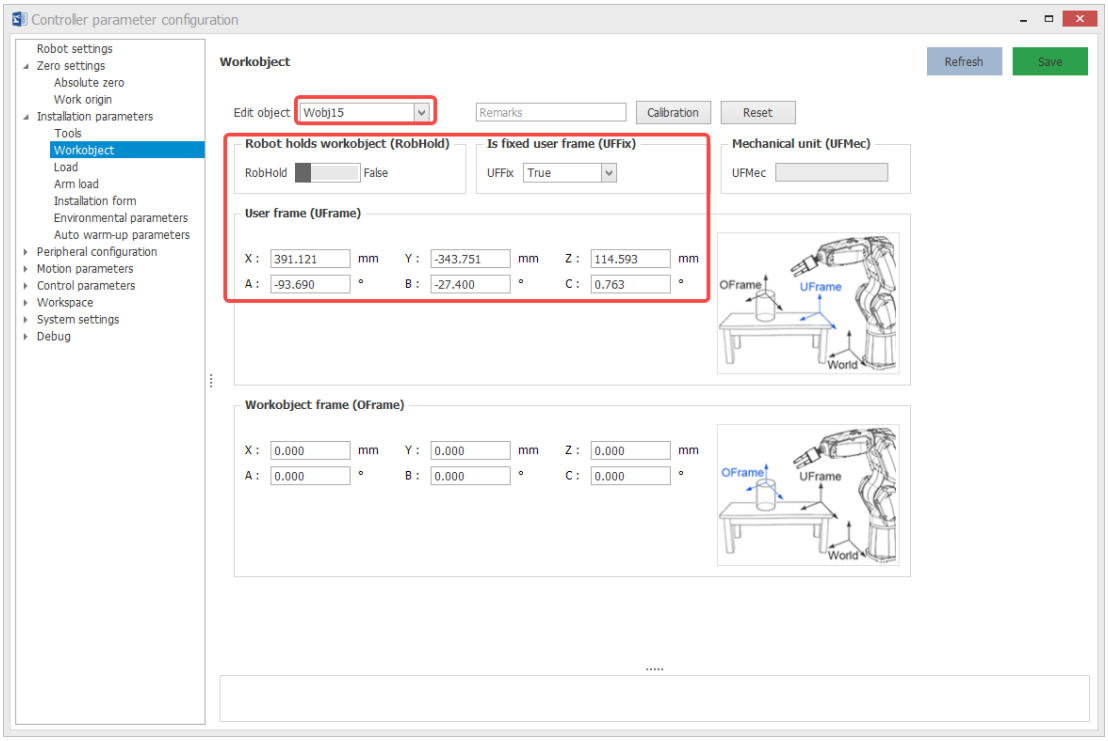
Teaching point 1



Teaching point 2



Teaching point 3



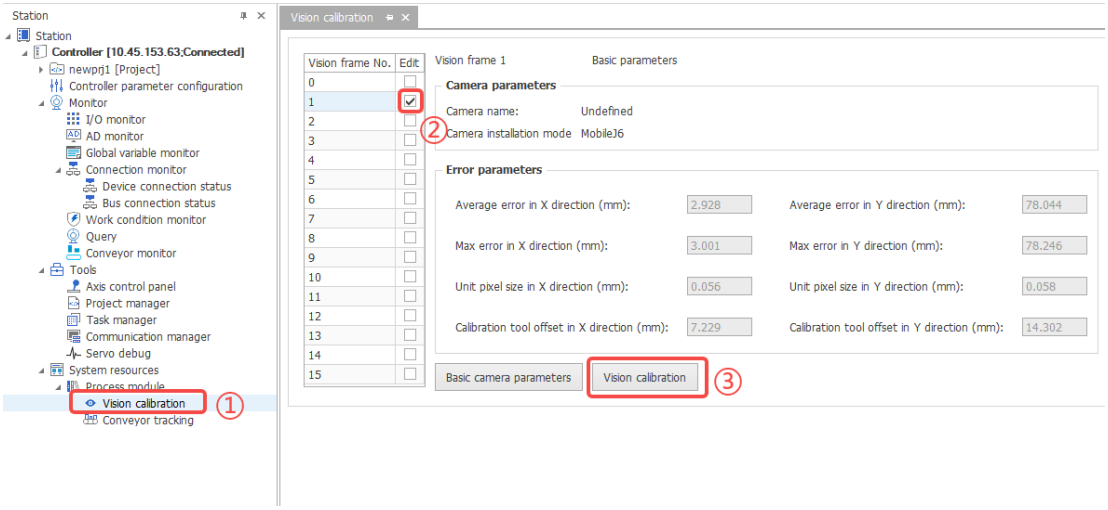
Workobject frame

CAUTION

When calibrating the workobject frame, switch the tool number to the number of the tip tool used. Otherwise, it will result in incorrect calibration results for the subsequent visual calibration process.

Step 4: Enter the vision calibration interface.

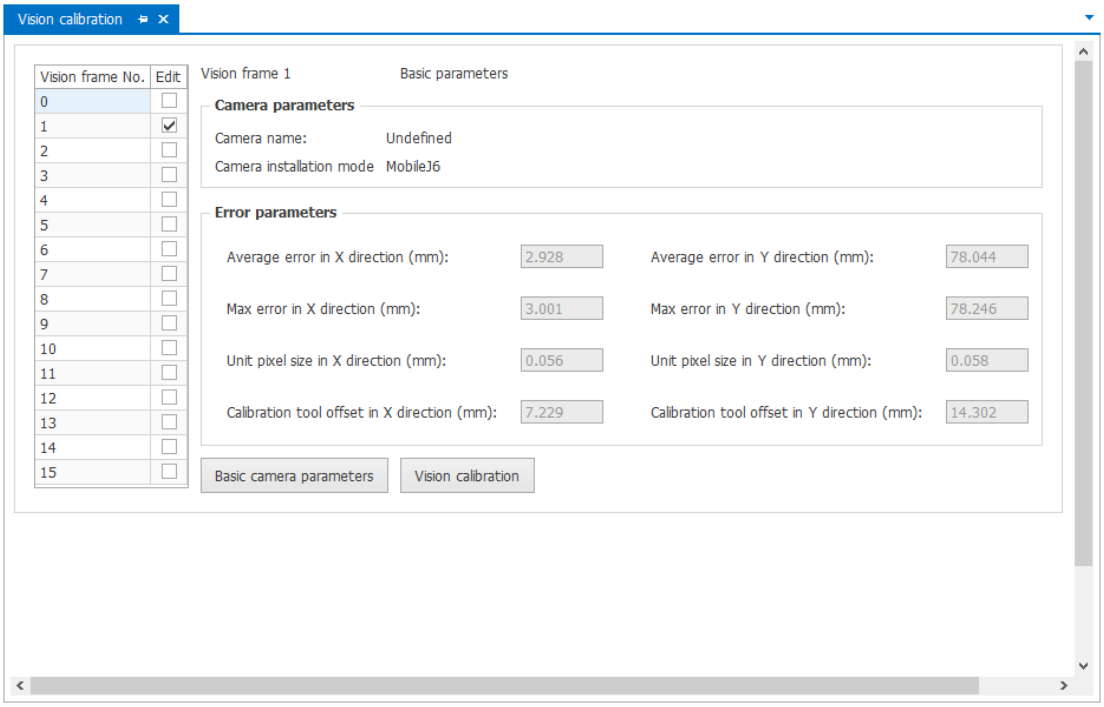
Select a vision frame, such as "Vision frame1", and click "vision calibration" to enter the visual calibration process.



Step 5: Set the basic camera parameters.

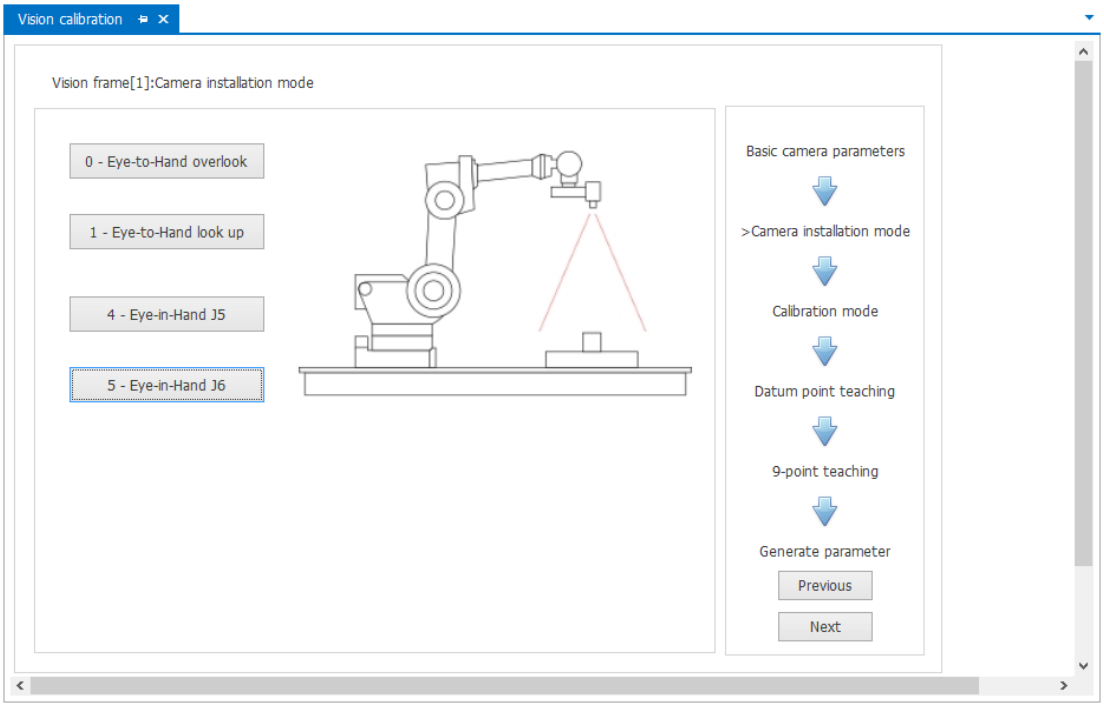
Set the basic camera parameters according to on-site conditions.

In this example, semi-auto calibration is adopted. Therefore, it is necessary to establish communication between the robot and the camera. Set the trigger mode to "Ethernet trigger".



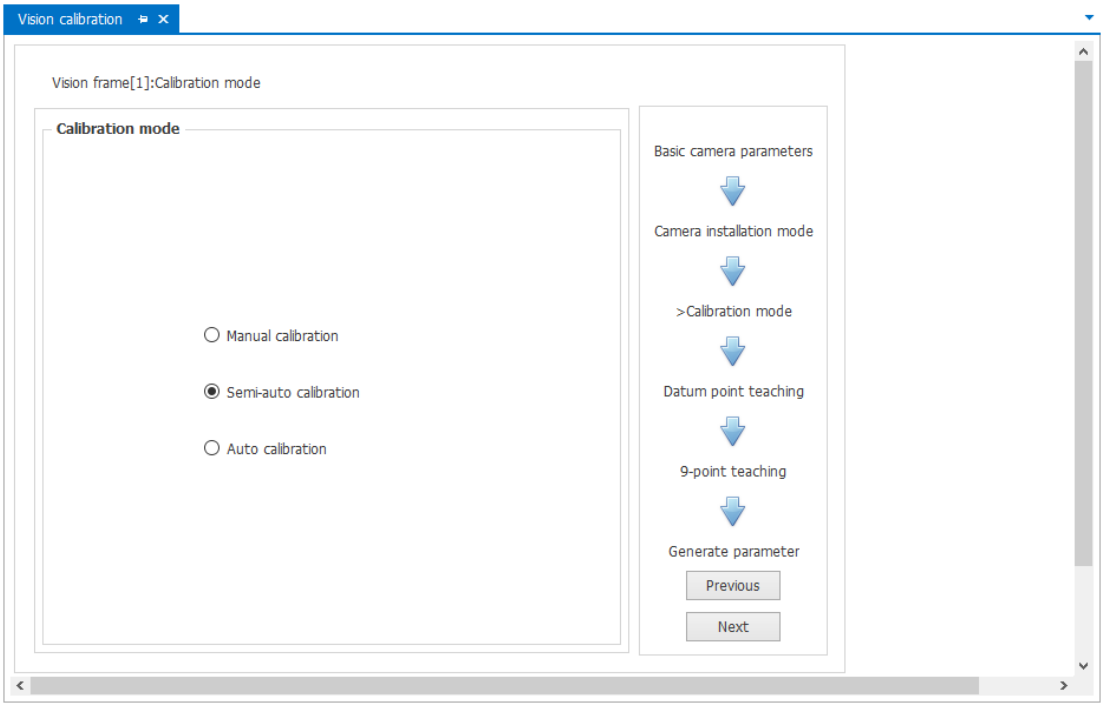
Step 6: Set the installation mode of the camera.

Select the installation mode according to on-site conditions. Here, as an example, select "5 - Eye-in-Hand J6".



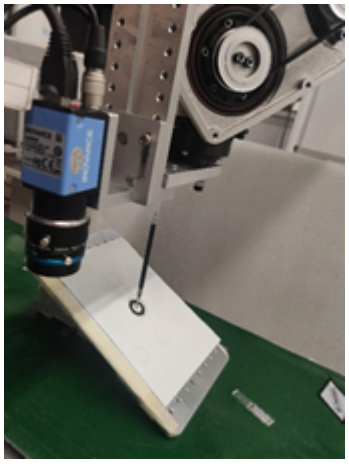
Step 7: Select the calibration mode.

Select the calibration mode according to on-site conditions. Select "Semi-auto calibration" in this example.



Step 8: Calibrate the datum points.

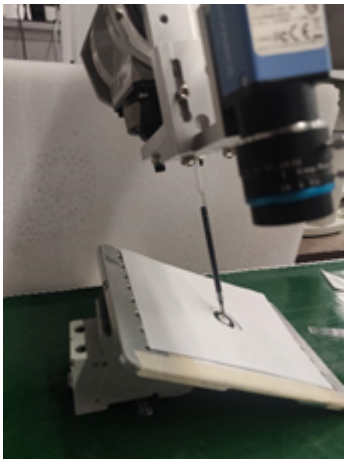
Calibrate the datum point by "Calibrate datum point with end tool" method. The calibration requires three datum points. Align the tip of the fixture with the marker point at the of the calibration board in three different orientations. Click "Get current point" to get the current joint position of the robot. Select the associated workobject frame number corresponding to the user plane (here, Wobj15) by the "Select associated workobject frame number" option.



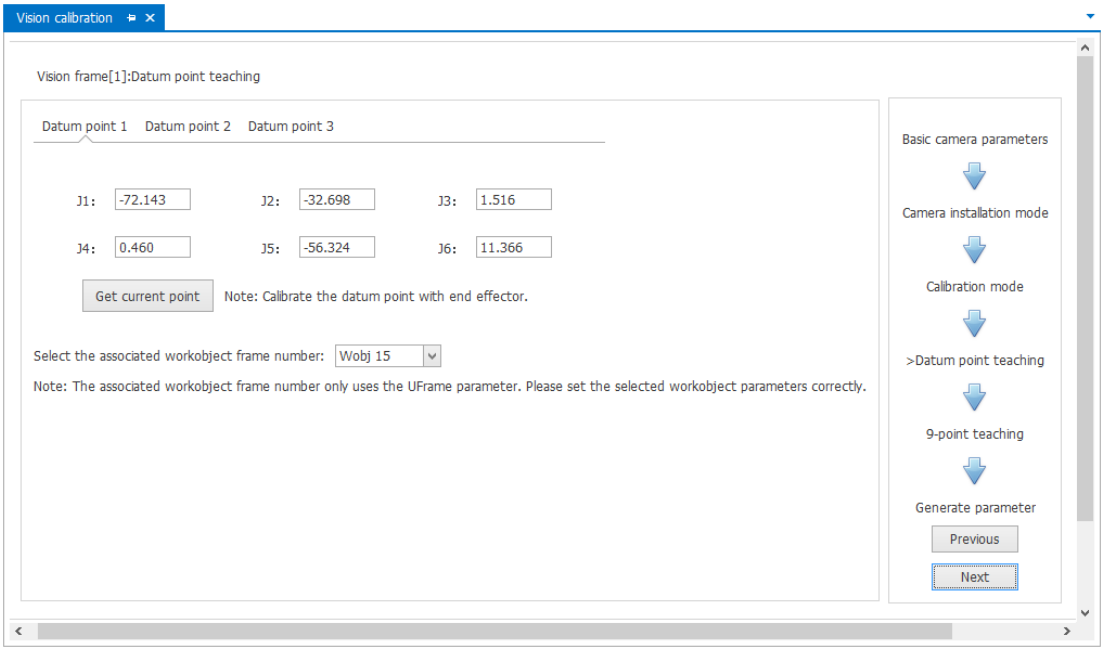
Teaching point 1



Teaching point 2

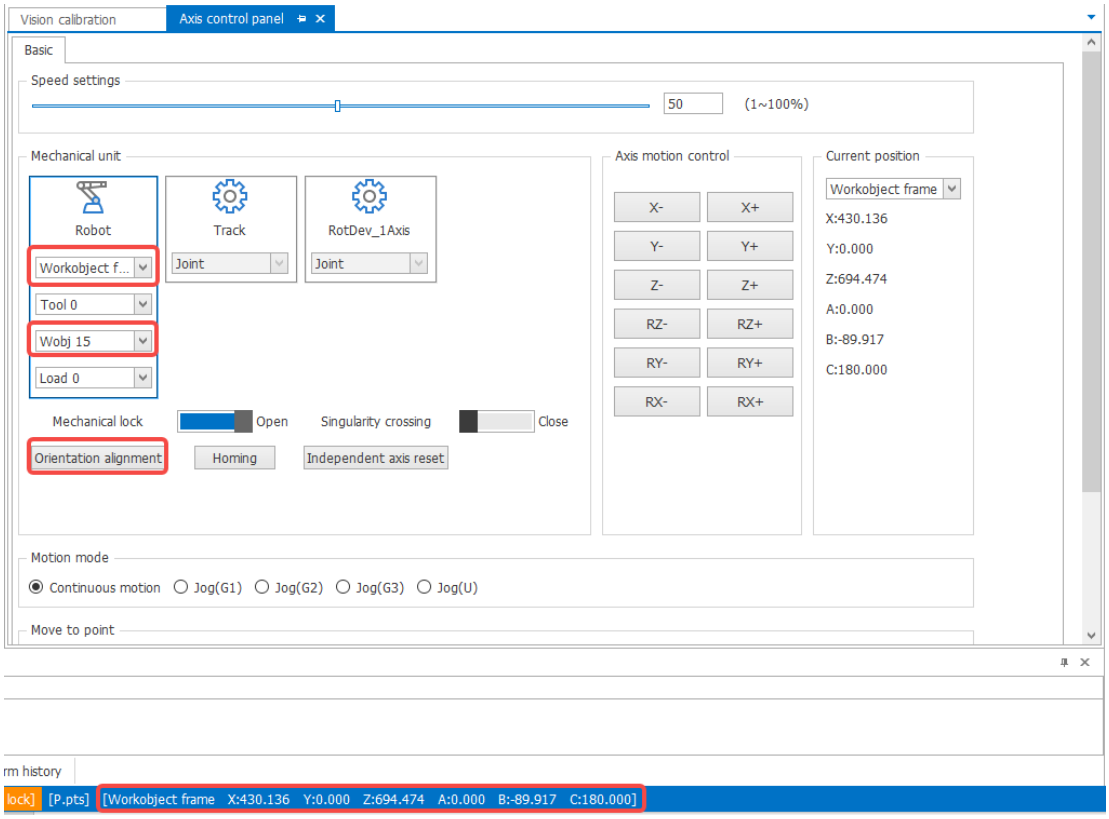


Teaching point 3



Step 9: Teach the robot with nine points on a pallet.

(1) Adjust the robot's pose so that the camera is relatively parallel to the user plane. Then, switch to the workobject frame and select the workobject frame number corresponding to the user plane. Use the "Orientation alignment" button to make the camera face the user plane, as shown in the following figure.

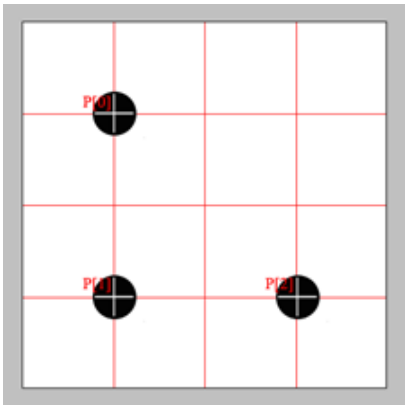


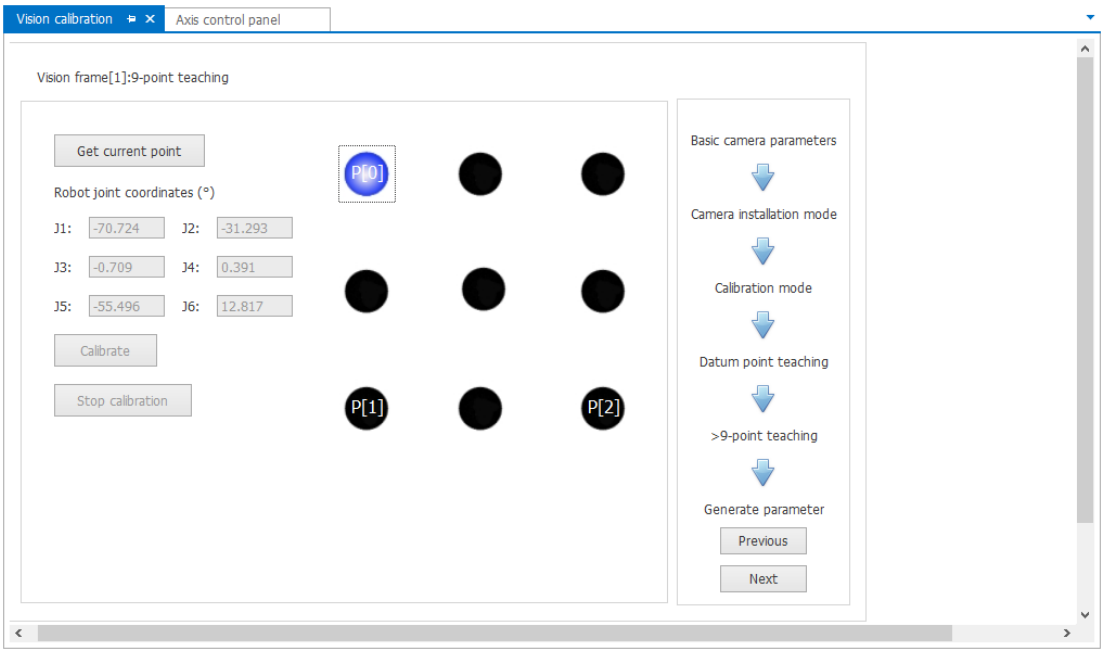


Relative position between the camera and the user plane

Adjust the relative height between the camera and the user plane in the Wobj15 frame and focus the camera.

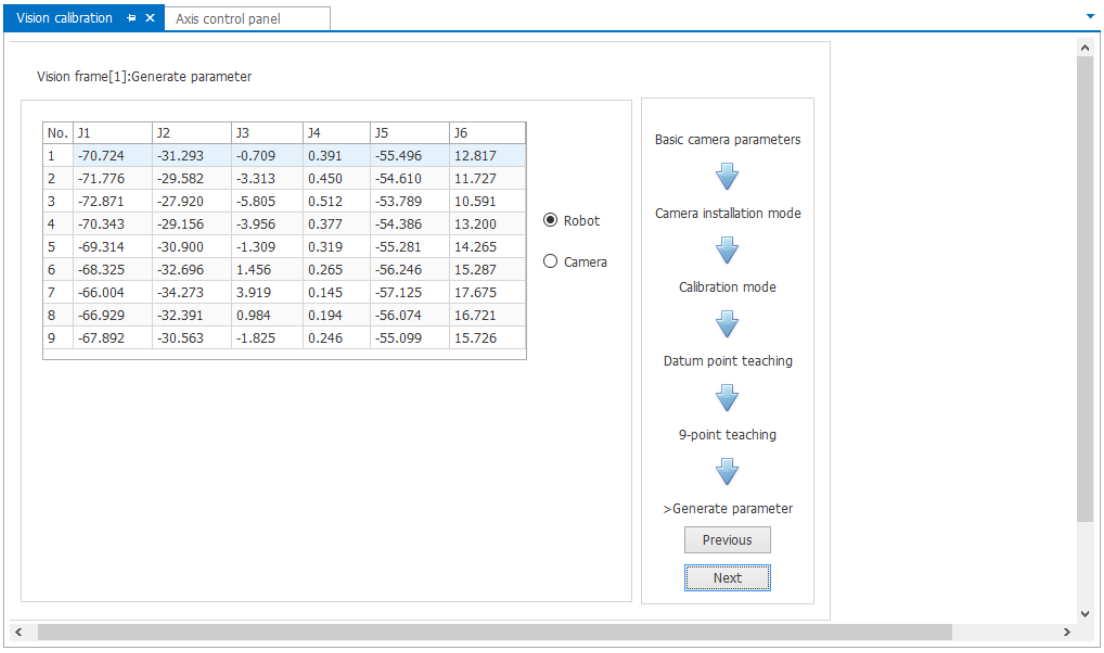
After establishing communication between the robot and the camera, select three points in the camera's field of view and click "Get current point" to obtain P[0], P[1], and P[2], as shown below. Align the intersection points of the grid lines in the camera's field of view with the marker points on the calibration board in sequence, and complete the nine-point pallet teaching. And then click "Calibrate" to allow the robot to automatically complete acquisition of the coordinates of the nine points.

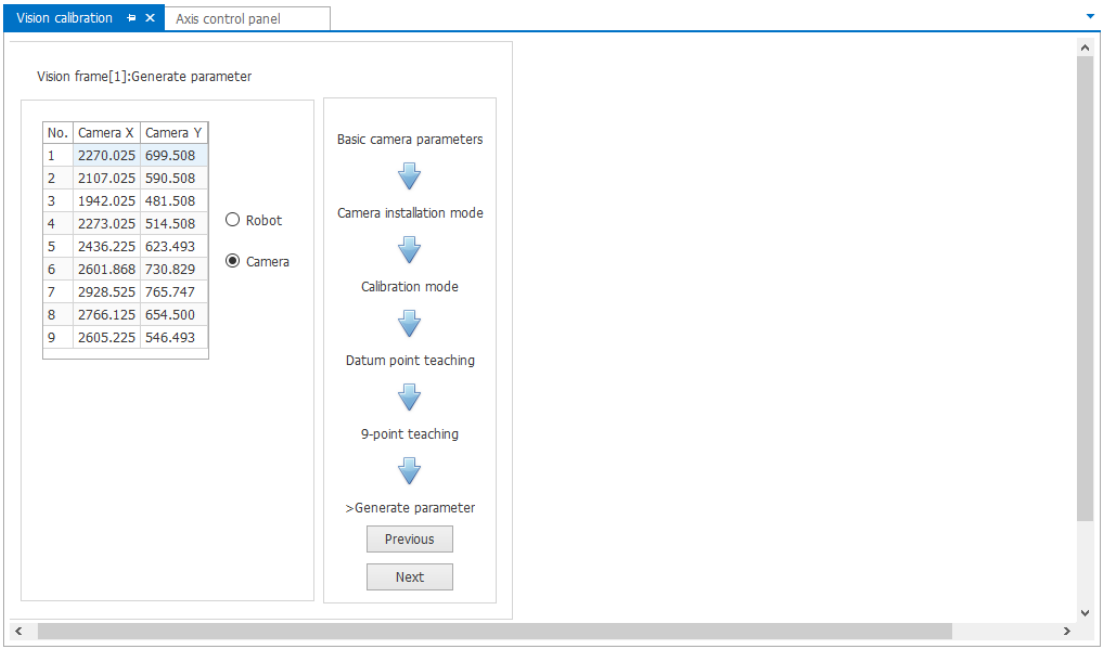




Step 10: Check the information about the nine points.

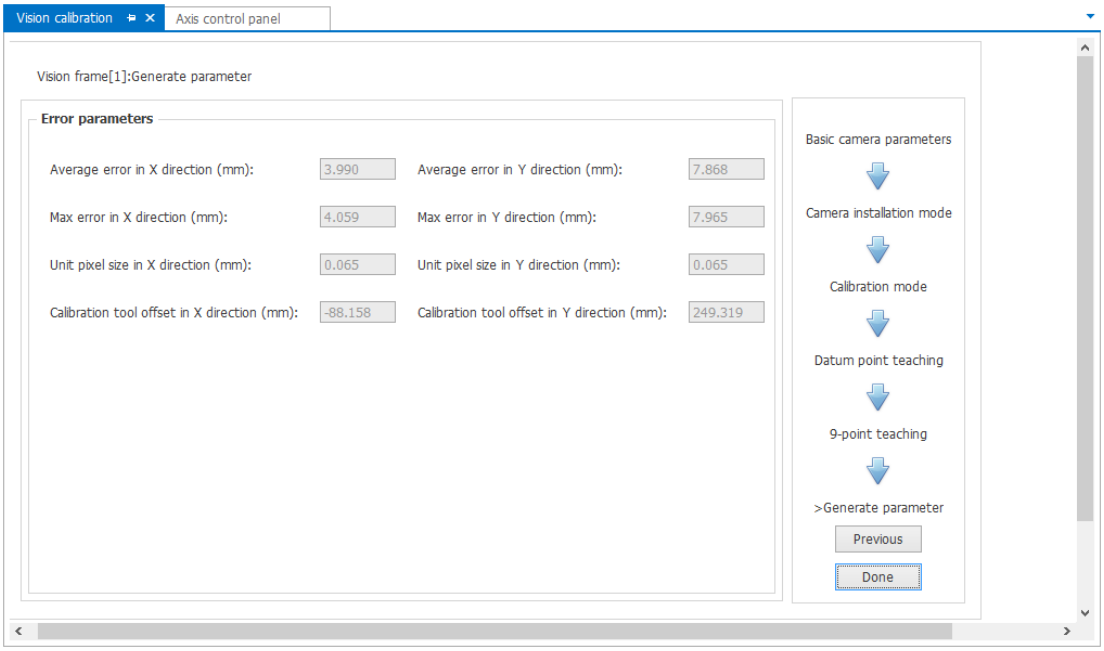
On the teaching point list, check whether each of the position coordinates and pixel coordinates is abnormal, for example: whether there are duplicate points, pixel coordinates error, etc. You can also double-click the point in the list to modify the point data. Click "Next" when you are done.





Step 11: Check the calibration results.

The system calculates and generates the vision frame calibration matrix and the calibration result parameters, as shown below. The calibration tool offset in X direction and calibration tool offset in Y direction indicate the tool parameters of the tip tool without rotation direction.



CAUTION

Under normal calibration conditions, the error is generally controlled below 2 mm.

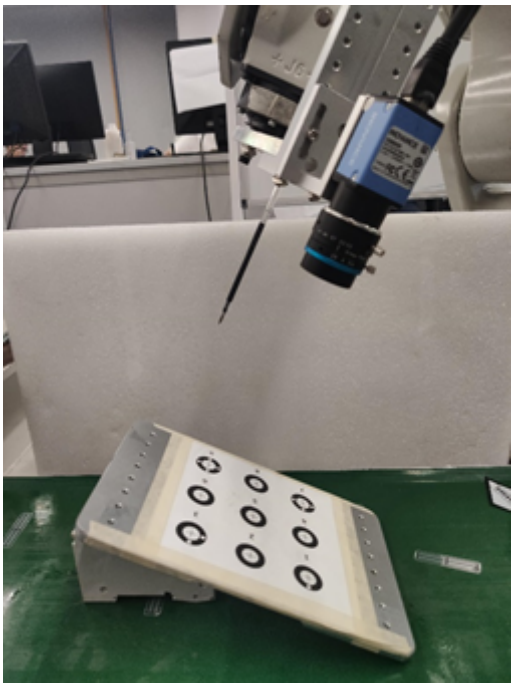
Step 11: Select "Done" to complete the vision calibration operation.

The calibration results can be verified through programming.

Step 12:

Move the camera relative to the user plane while keeping the camera parallel to the user plane and the distance between the camera and the user plane unchanged. As shown below, the camera takes a photo of

a random point on the user plane to get the pixel coordinates of the point. At the same time, record the current joint value of the robot at the time of photo taking.



Step 13: Write the calibration program to verify the calibration results.

The following shows an example of vision calibration program.

```
Start;
   P[100] = {(1303.8,1045.4,0,0,0,0),(1,0,0,0),(0,0,0,0,0,0)};
   P[101] = {(0,0,0,0,0,0),(1,0,0,0),(0,0,0,0,0,0)};
   P[101] = CVToRobPos(P[100],1,JP[0]);
   Print P[101];
End;
```

CVToRobPos instruction: Converts the pixel coordinates to the workobject coordinates.

(1) CVToRobPos (VisionP, VisionCrdNo) is the conversion format for camera installed in eye-to-hand mode.

VisionP is the vision pixel point, stored in the P or LP variable, and VisionCrdNo is the vision calibration frame number (0-15).

(2) CVToRobPos (VisionP, VisionCrdNo, TrigJP) is the conversion format for camera installed in eye-in-hand mode.

TrigJP is the robot joint value at the position where photo is taken, stored in the JP variable.

11.5.25 CVToRobPos

- **Function**: Converts vision coordinates into robot coordinates.
- **Description**: Converts the pixel coordinates sent from the vision equipment into the corresponding workobject coordinates.
- **Format**:

1. CVToRobPos(P[*], int VisionCrdNo);
   Meaning: Converts pixel coordinates into workobject coordinates when the camera is installed external to the robot.
2. CVToRobPos(P[*], int VisionCrdNo, JP[*]);

   Meaning: Converts the pixel coordinates into workobject coordinates when the camera is installed on the robot arm. JP is the joint coordinates of the photo-taking point.

   Parameter:

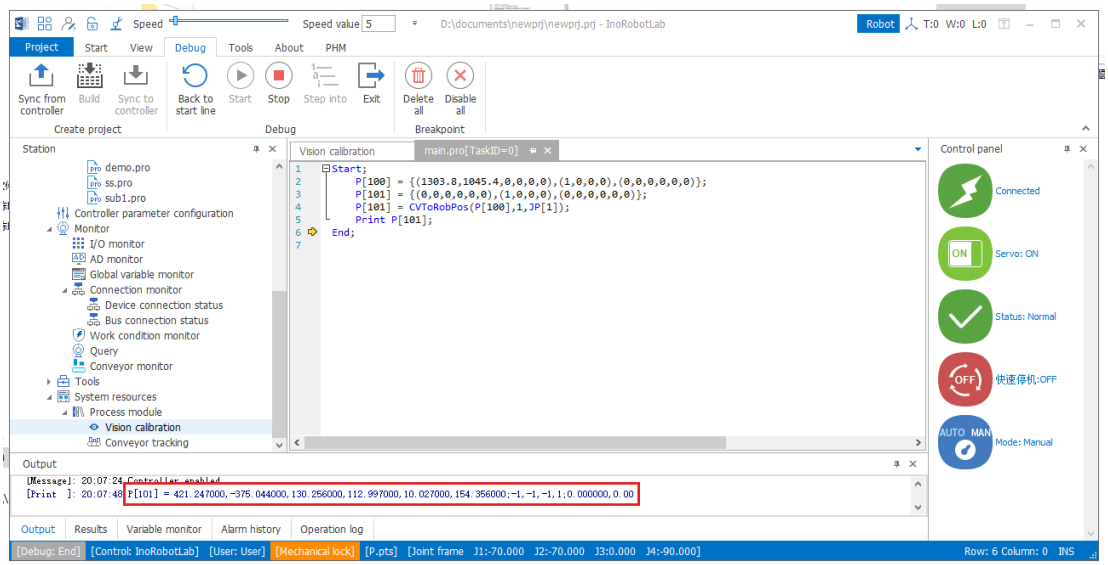
P[*]: Pixel coordinates sent from the vision equipment, in which the arm parameter and external axis data use the arm parameter and external axis data of the target point.

VisionCrdNo: Vision frame number.

JP[*]: Joint coordinates of the photo-taking point when the camera is mounted on the robot arm.

In this example, the camera is installed on the J6 axis. Therefore, the second format of CVToRobPos instruction is used, for example, P[101] = CVToRobPos(P[100],1,JP[0]). Here, the parameter "P[100]" indicates the pixel coordinates in the vision frame, wherein X, Y, and A indicate the visually recognized pixel coordinates and angle. The parameter "1" indicates the vision frame number used in the previous vision calibration. The parameter "JP[0]" indicates the joint value of the robot at the photo-taking point.

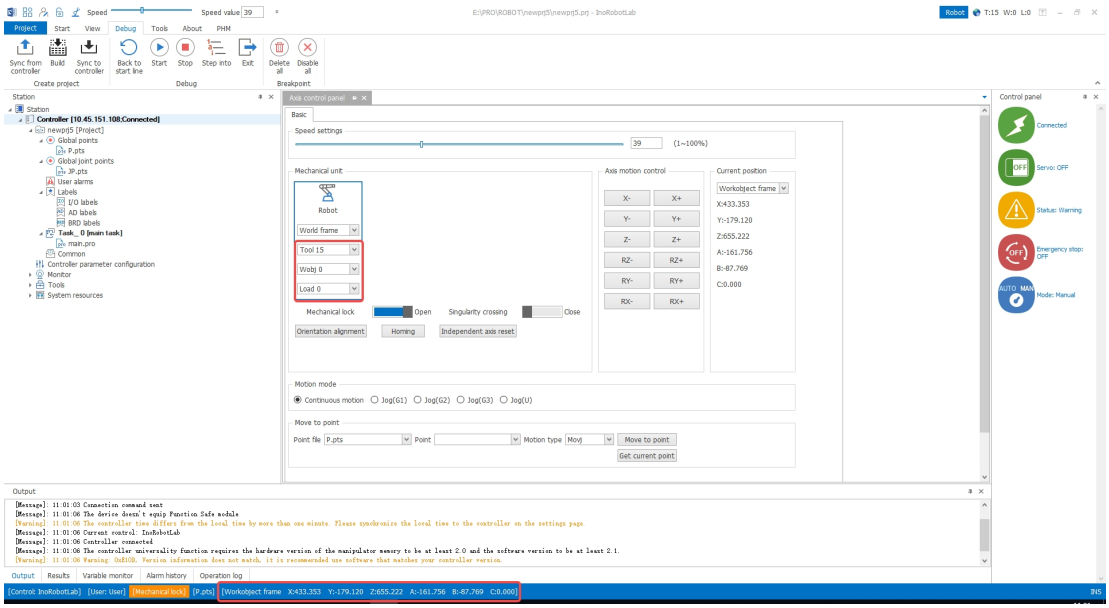
Run the program to calculate the converted workobject coordinates, as shown below. The meaningful coordinates (X,Y) = (421.246955,-375.044309) indicate the coordinates of the marker point in the base frame.



Teach the robot to align the tip tool with the marker point, as shown below.



Switch the frame to the base frame, and tool to the tip tool. Then view the robot coordinate, namely, the actual spatial position value of the recognized point, (423.198, -375.175). Compare this coordinate with the converted workobject coordinate (421.246955, -375.044309) to determine the calibration accuracy.



The value A -140.896 of the converted tool coordinates is inconsistent with value A 137.574 of the tool coordinates of the robot where the photo is taken, due to the inconsistency in direction of the vision frame, direction of the robot frame, and direction of the tool.

"A" indicates the rotation angle of the object in the vision frame, and is converted to the angle in the tool frame as A1.

A1 = A + A', where A' indicates the conversion angle difference.

A2 = Angular difference between the direction of the robot frame and the direction of the camera frame.

A3 = Angular difference between the direction of the robot frame and the gripper direction or load gripping direction.

A4 = Angle at the position where photo is taken in the tool frame.

A4 = A + A'+ A2 + A3 or A4 = A1 + A2 + A3

It is known that A4 = 137.574, A = 137.574, and A1 = -140.896, then:

A2 + A3 = A4-A1 = 137.574 - (-140.896) = 278.47 orA' + A2 + A3 = A4-A = 137.574 - 137.574 = 0

Therefore, when a pixel coordinate A is given, the coordinates of the robot at the position where the photo is taken can be calculated as: A4 = A + A' + A2 + A3 = A + 0.

Alternatively, you can use the CVToRobPos instruction to obtain the corresponding value A1 in the tool frame from the pixel coordinate A:A4= A1 + A2 +A3 = 278.47 + A1.

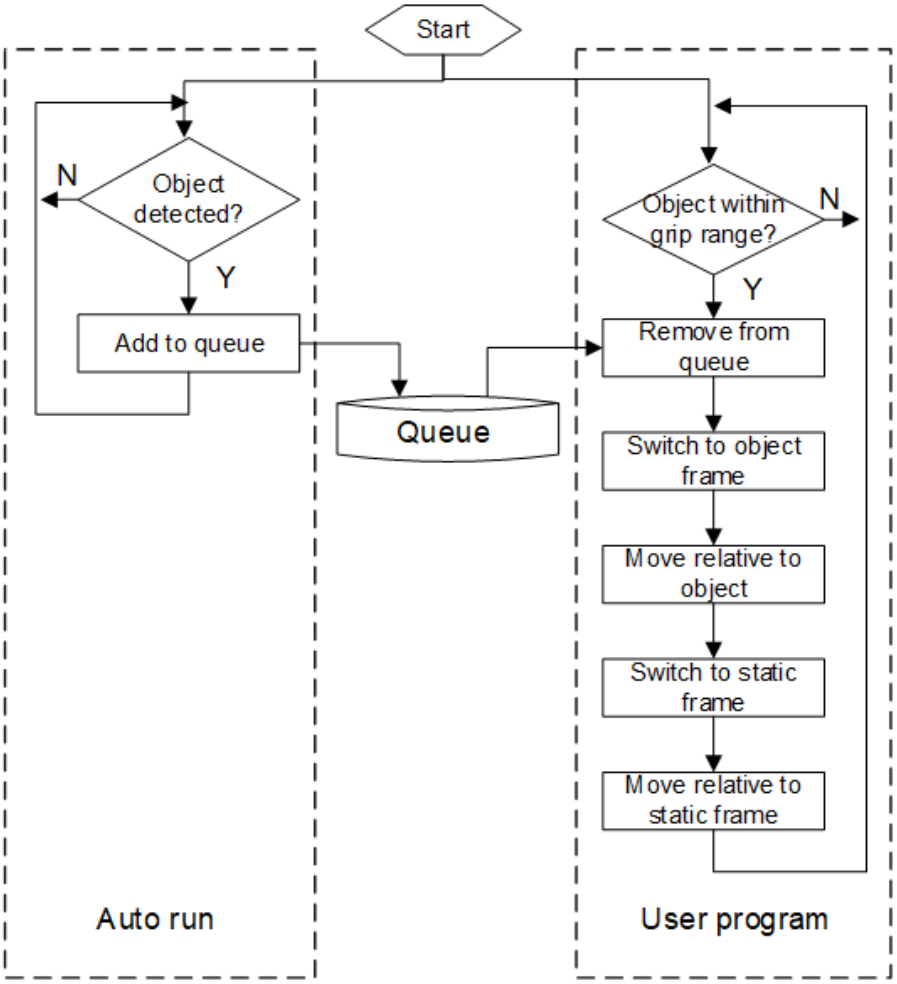
11.2 Conveyor Tracking

11.2.1 Overview

The tracking process allows you to track moving objects and plan robot movements with reference to the moving objects. Conveyor tracking is a typical application of the tracking process. It detects objects on the conveyor through vision device or photoelectric sensors to enable the robot to accurately grasp moving objects.

11.2.1.1 Workflow

1. The position of an object on a conveyor is acquired through vision device or photoelectric sensors. The acquired position information is placed into a conveyor object queue, and the changes in the position of tracked object is fed back through the encoder.
2. The robot moves with the moving object in the object queue as the reference frame. The position detection and robot movement are processed in parallel. The following figure shows the detailed flowchart.



Conveyor tracking process

The conveyor application is also known as dynamic grasping application. The position of the material changes with the movement of the conveyor. The vision device sends the position data of material to the robot at the time of taking a photo of the material. The material moves in real time during the photo taking period. Therefore, if the robot grasps the material according to the original position data, it will definitely be inaccurate. Therefore, the real-time offset value is calculated from the data of the encoder on the conveyor to achieve accurate dynamic grasping.

Dynamic grasping can be divided into two types: parallel grasping and predictive grasping.

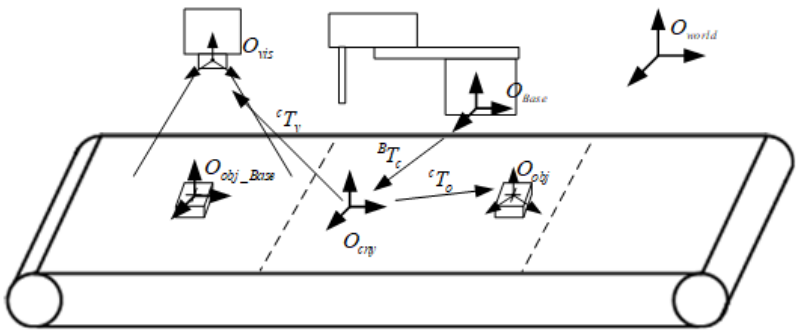
- In parallel grasping, the robot calculates the position of the material in real-time through the encoder data while maintaining a relatively static state with the conveyor, thereby achieving grasping.
- In predictive grasping, the robot calculates the position of the point where the robot meets the material in advance based on the movement time after obtaining the position of the material, thereby achieving grasping.

11.2.1.2 Position Description System

The tracking motion takes the object frame as the reference frame. The description of the trajectory and point position during motion is relative to the dynamic frame. When no motion instruction is applied, the robot and the object remain relatively static (i.e., the robot moves in synchronization with the object). When a motion instruction is applied, the motion trajectory is a trajectory under the object frame, and the target point is a coordinate point under the object frame.

- Conveyor frame (O_{cny}): Indicates the position and orientation of the conveyor under the base frame. It is used to describe the relative relationship between the conveyor and the robot. The conveyor frame is static.
- Object frame (O_{obj}): Indicates a frame that is fixedly attached to the object and moves with the object. It is a dynamic frame. The object frame describes the position and orientation of the object on the conveyor. It is the coordinate of the datum point of the object under the conveyor frame.
- Vision frame (O_{vis}): Describes the position conversion relationship between the camera and the conveyor. The instantaneous position of the object on the conveyor is obtained through the camera, and the amount of movement of the object on the conveyor is obtained in real time through the encoder, enabling the robot to track the position of the object in real time.

The relationship between frames in the tracking process is shown below.



Relationship between frames in the tracking process

11.2.1.3 Function Specifications

- Conveyor: Linear, circular

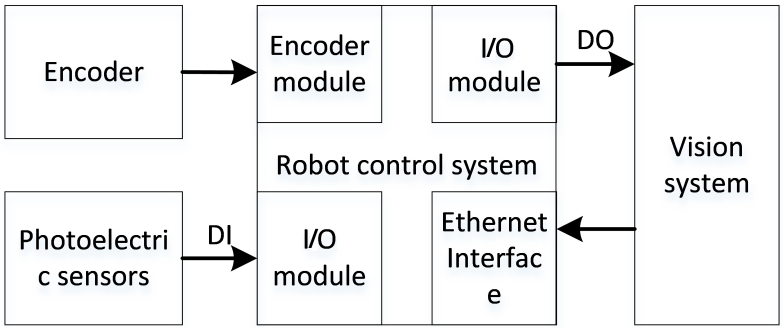
- Number of conveyors: 4 can be configured, and up to 2 can be used simultaneously.
- Conveyor speed: Linear: 0 to 1 m/s; Circular: 0 to 180°/s. (The conveyor speed that different robots can keep up with varies depending on the specific robot)
- Detection method: Vision device or photoelectric sensor, with a maximum of one vision device used.
- Tracking motion instructions: Movl, Movc, Movs, JumpL, CnvFastCatch.
- Number of types of workobjects: 16 different types of workobjects can be set on the same conveyor.
- Vision data type: Robot coordinates or pixel coordinates.
- Vision communication format: Fixed format (For details, see the CnvVison instruction in the "Industrial Robot Instruction Guide".)
- Number of objects in a single picture: 0-50.
- Object queue length: 1000.

11.2.1.4 Configuration Flowchart

The following are required to use the tracking process. For details, see the 11.2.2 Hardware Configuration section, 11.2.3 Frame Settings section, and 11.2.4 Parameter Settings section.

11.2.2 Hardware Configuration

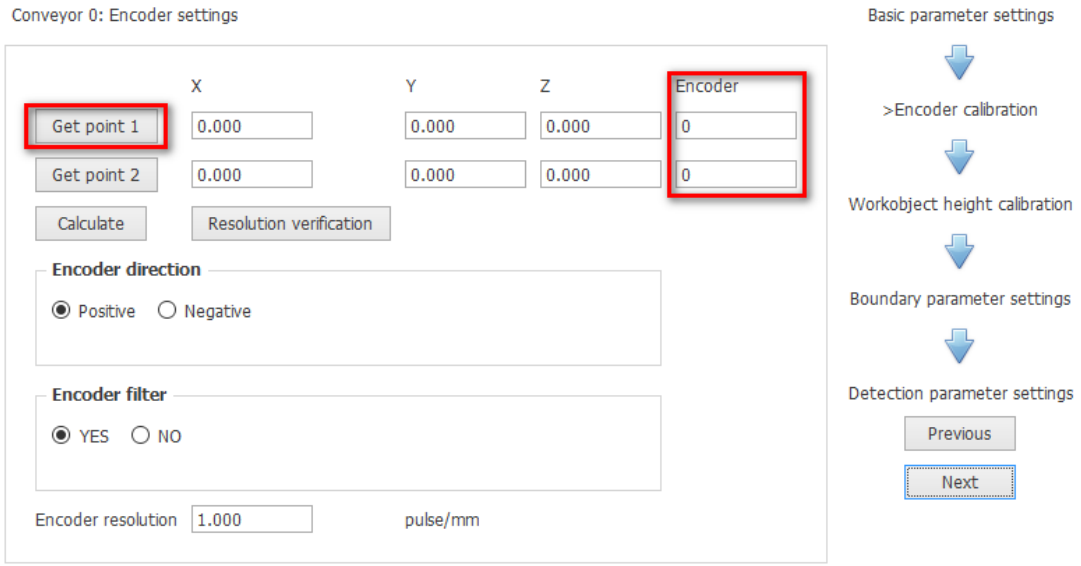
The following figure shows the configuration of the following process hardware. The photoelectric sensor and vision system can be configured according to the detection mode. If vision system is used, the camera must be triggered by hardware.



Hardware configuration for the tracking process

11.2.2.1 Connecting the Encoder

Connect the signal and power lines of the pulse encoder to the encoder interface of the robot controller. See the controller user guide for specific wiring logic. After the physical connection is completed, rotate the encoder and read the value of the encoder through the following interface. Observe the changes in the encoder value to determine whether the physical connection is correct.



Reading the encoder position

11.2.2.2 Connecting the Photoelectric Sensor

The photoelectric sensor outputs different level signals when triggered by an object. The robot controller detects the object through the signal edge. Connect the signal wire of the sensor to a general DI interface of the I/O module of the controller. For the wiring requirements, see the controller user guide. Once the wiring is complete, observe the change of corresponding DI value on the I/O monitoring interface to determine whether the sensor works normally.

11.2.2.3 Vision Device

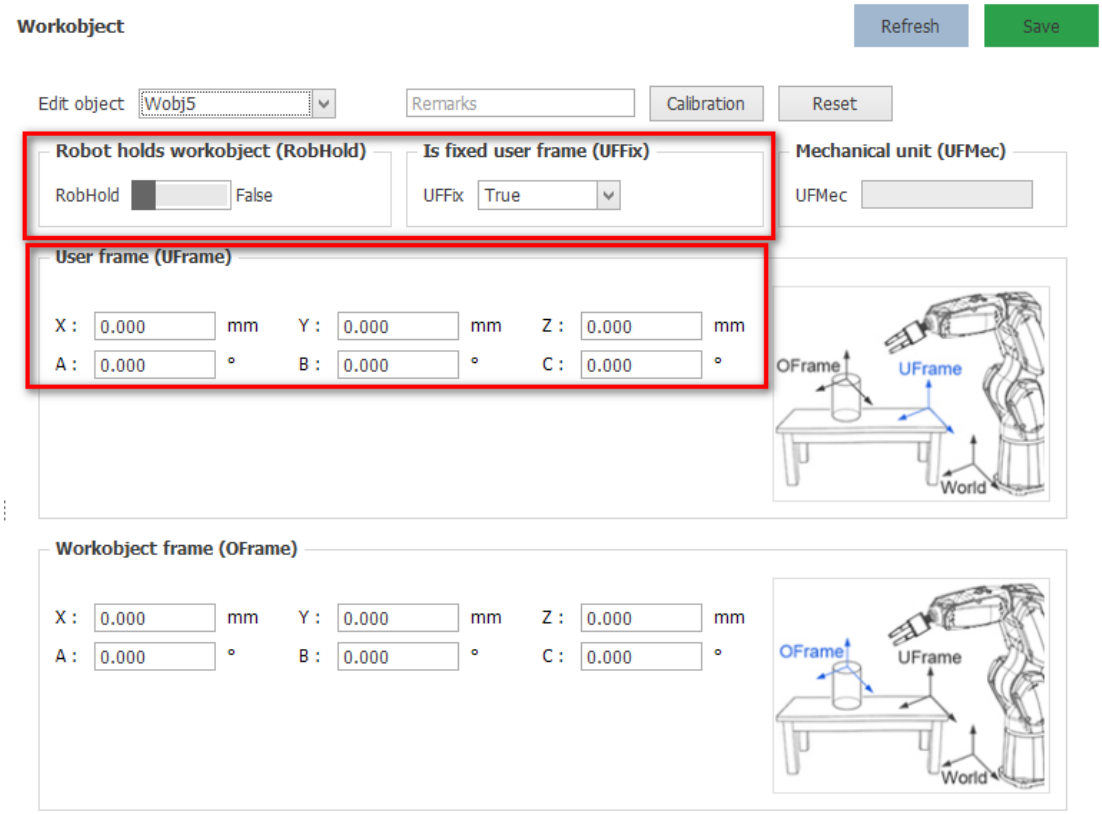
The interaction between vision device and the robot includes DO triggering and Ethernet communication. The DO triggering requires a signal line connecting the DO interface of the robot controller and the I/O trigger interface of the camera. After the physical connection is completed, manually operate the DO to see if the camera can be triggered properly.

11.2.3 Frame Settings

The conveyor frame is the workobject frame associated with the tracking process. You need to set or calibrate the frame parameters in "Controller parameter configuration" > "Installation parameters" > "Workobject". The conveyor frame describes the position and orientation of the conveyor. The positive X-axis direction coincides with the direction of conveyor movement. The setting method is described as follows.

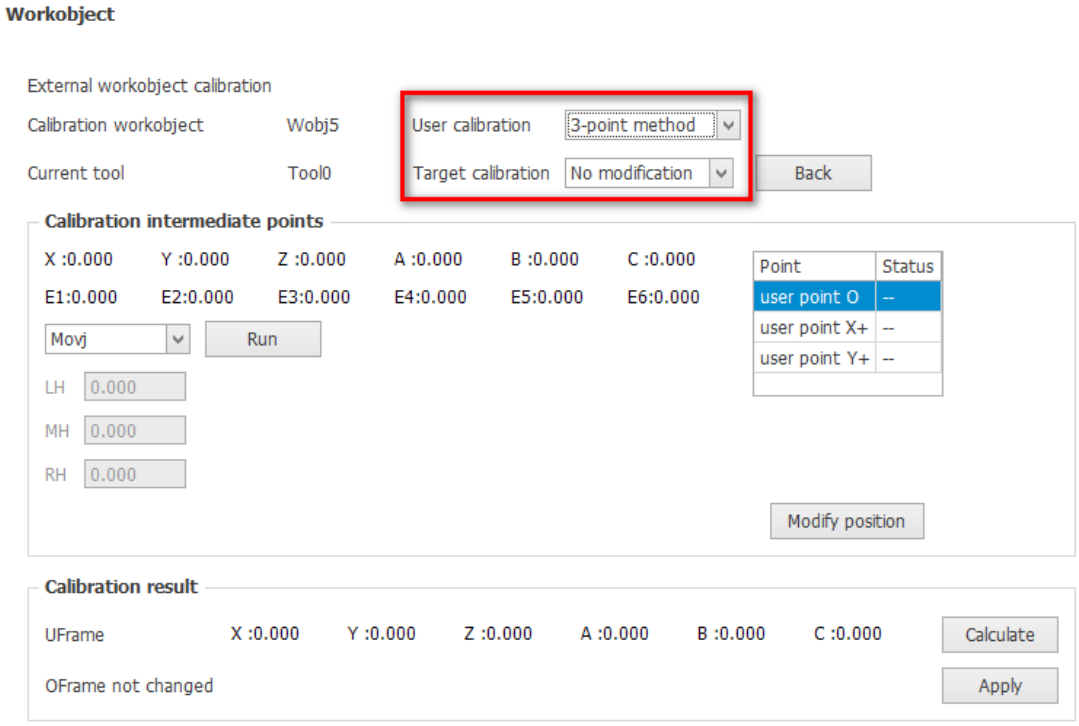
CAUTION

- When calibrating the workobject frame in the tracking process, you need to use the tool frame used in the actual tracking project.
- If the workobject frame is used as the conveyor frame, you only need to calibrate the user frame parameters (UFrame) without the need to calibrate the workobject frame parameters (Oframe). For the workobject frame, the RobHold parameter must be set to "False" and UFFix parameter to "True".



Setting the conveyor frame

- For a linear conveyor, use the 3-point method, as shown below.



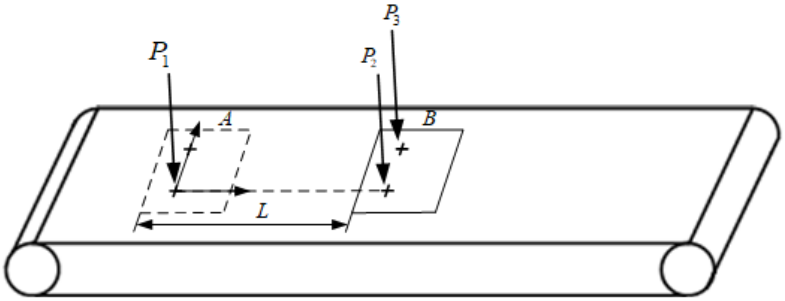
3-point method

Select three points as follows:

- Set the origin of the conveyor frame: Place the calibration board or another calibration reference on the conveyor. Align the end of the robot tool with the datum point to get point P1, which is the origin of the conveyor frame.
- Set a point in the X direction of the conveyor frame: Keep the position of the calibration board relative to the conveyor unchanged during movement. Turn on the conveyor to move from P1 to P2, and then stop

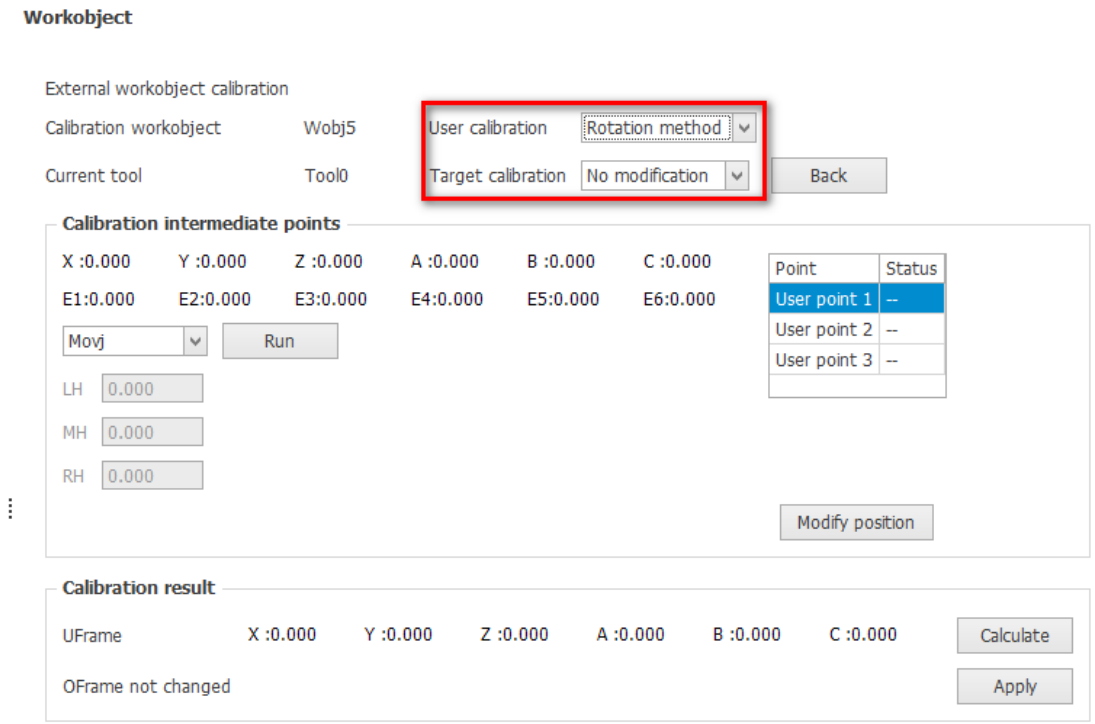
the conveyor. Align the end of the robot tool with the datum point to get point P2. The direction from P1 to P2 is the positive X direction.

- Set a point in the Y direction of the conveyor frame: Move the end of the robot tool more than 10 mm to the left in the direction of conveyor movement. Getting a point in the first quadrant XY as point P3 according to the right hand rule ensures that the Z axis is perpendicular to the conveyor direction.



Calibrating the linear conveyor using the 3-point method

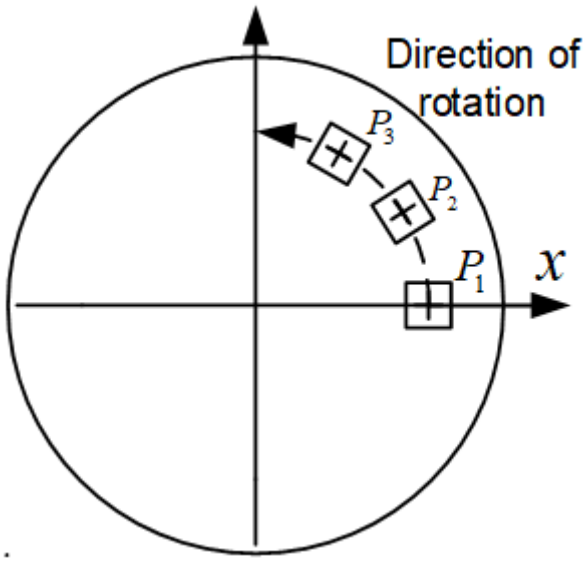
- For a circular conveyor, use the rotation method. The calibration interface is shown below.



Rotation method

The rotation method takes three points, as shown below.

- Fix a calibration datum point on the conveyor. After rotating the datum point to a proper position, align the end of the robot tool with the datum point to get point P1. The direction from the center of rotation to P1 is the positive X direction of the frame.
- Rotate the conveyor counterclockwise, align the end of the robot tool with the datum point to get points P2 and P3 in sequence. Three points determine an arc. The center of the arc is the origin of the frame. Since the conveyor is rotated counterclockwise, it is ensured that the Z axis is perpendicular to the conveyor direction. The Y axis direction can be determined according to the right hand rule.



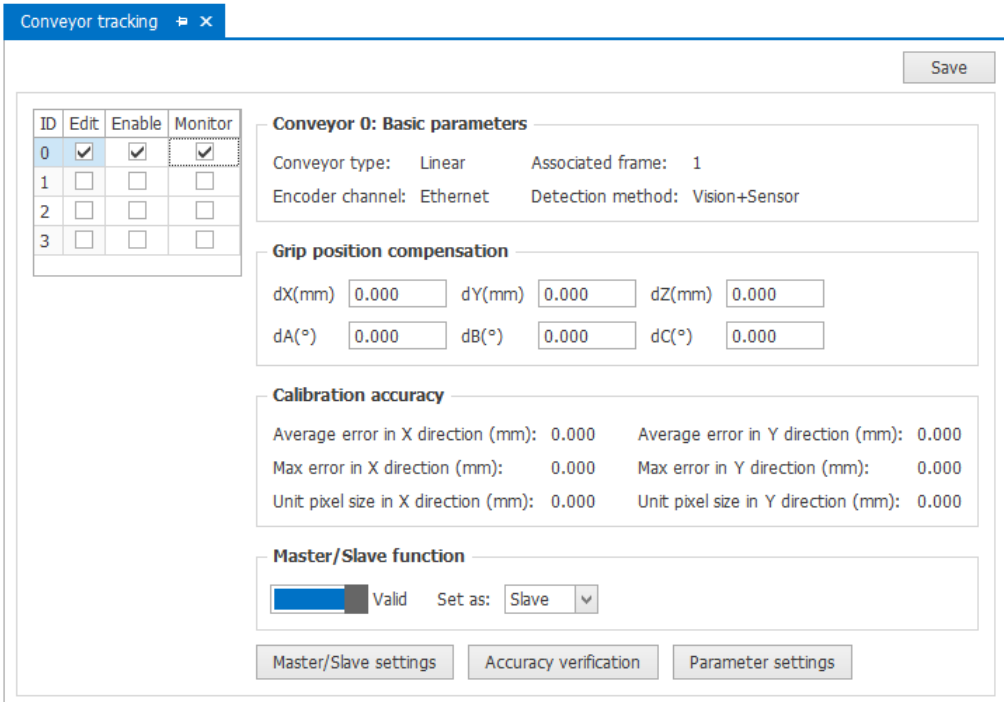
Calibrating the circular conveyor by rotation method

CAUTION

- To improve accuracy, it is recommended that the three points are spaced as far away as possible, with points separated by >30° from the origin.
- If a tool is installed at the end of the robot, it is required to first calibrate the tool frame and activate the corresponding tool frame number.

11.2.4 Parameter Settings

The conveyor parameters include basic parameters, encoder parameters, workobject height parameters, boundary parameters, and detection parameters. Depending on the detection mode, the detection parameters are divided into vision parameters or sensor parameters. You can set the conveyor parameters in "Station" > "Controller" > "System resources" > "Process module" > "Conveyor tracking". The main interface is shown below.



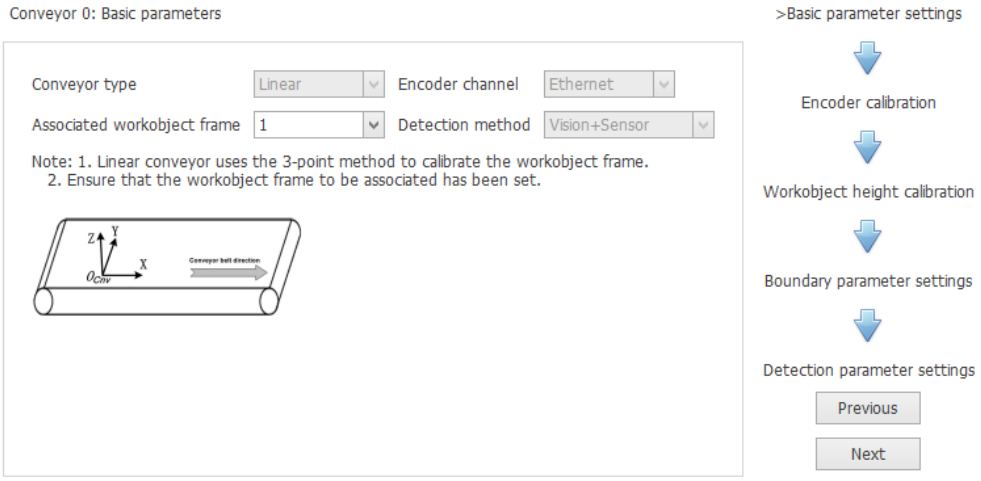
Conveyor tracking parameter settings

The following describes the options and buttons on the conveyor tracking interface.

Function	Description
Edit	When checked, you can view or edit the parameters for that conveyor.
Enable	When checked, the conveyor can be used in the user program.
Monitor	When checked, you can monitor the conveyor tracking information.
Basic parameters	Displays the basic parameters of the selected conveyor. The parameters are for view only and cannot be modified
Grip position compensation	You can fine-tune the grip position. This is for position compensation of the origin of the object frame in the conveyor frame.
Master/Slave function	For details, see the 11.2.7.2 One-Drive-Many Vision Function section.
Parameter settings	Click to enter the parameter setting page.
Accuracy verification	Click to enter the accuracy verification page.

11.2.4.1 Basic Parameter Settings

Click the "Parameter settings" button to access the parameter setting interface. The parameter settings are wizard-based. Click "Previous" or "Next" button in the lower right corner to switch the interface. After setting the parameters, click "Next", "Save" or "Done" button to save the parameters to the controller.



Basic parameter setting interface

The first step is to set the basic parameters, including:

- Conveyor type: Linear or circular.
- Encoder channel: The signal input channel of the conveyor encoder on the robot controller.
- Associated workobject frame: The workobject frame obtained in 11.2.3 Frame Settings and to be associated with the conveyor.
- Detection method: Vision, Sensor, Vision + Sensor, and Vision + Instruction.

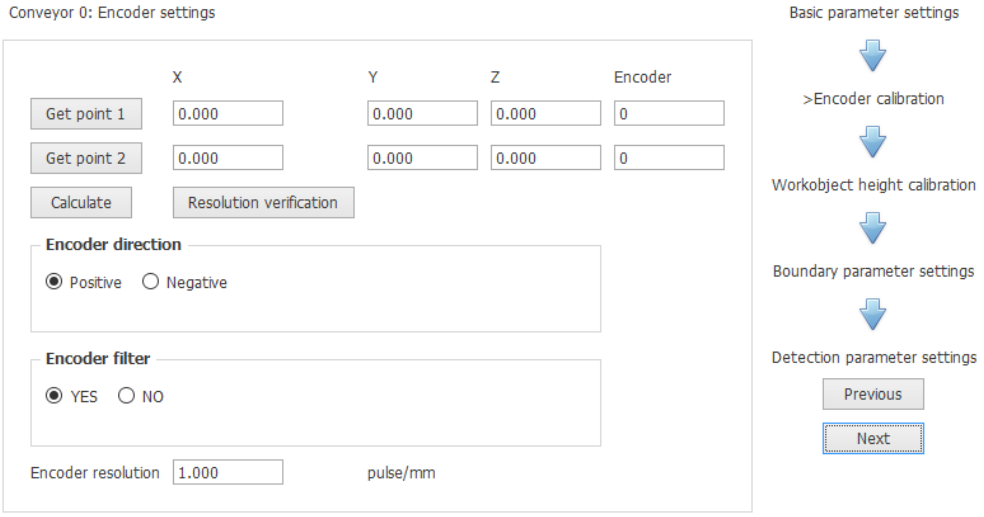
CAUTION

As the workobject frame 0 is fixed in the system and is coincident with the base frame, it cannot be modified. When calibrating the conveyor frame, use workobject frames 1-15.

11.2.4.2 Encoder Calibration

The second step is to calibrate the encoder. You can set the following parameters.

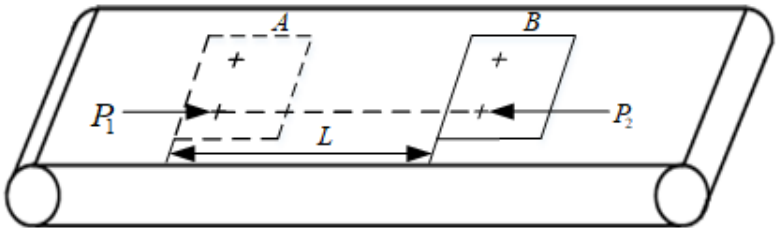
- Encoder resolution: Indicates the pulse increment of the encoder when conveyor moves by a unit length (mm or °).
- Encoder direction: Indicates whether the encoder pulse value increases or decreases when the conveyor moves forward.
- Encoder filter: By default, the encoder filter is enabled. Disable the encoder filter only when the conveyor speed is stable and external interference is low, to improve the grip accuracy.
- Resolution verification: Allows you to re-calibrate the encoder resolution to calculate the encoder resolution error and conveyor frame calibration error. The resolution verification is required only for checking for errors in encoder resolution or conveyor frame calibration when gripping accuracy is low.



Encoder resolution settings

The calibration process is described as follows.

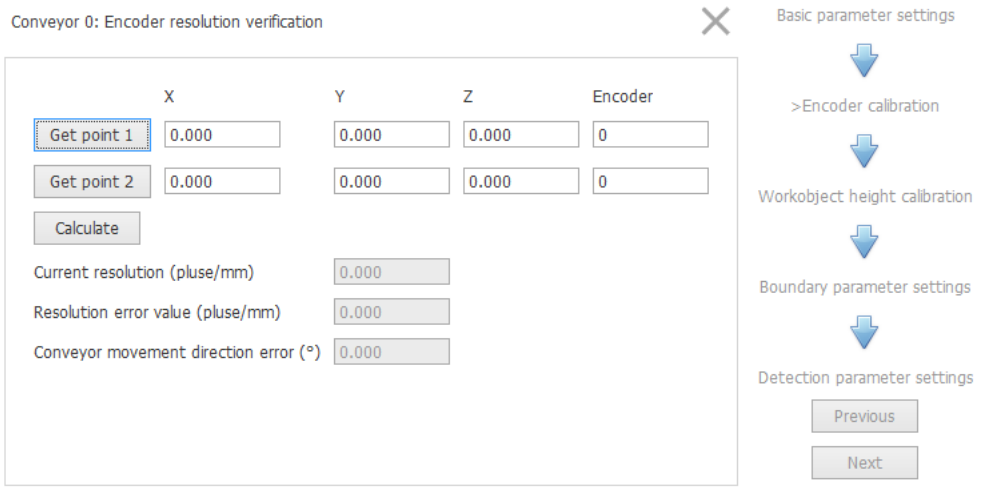
1. Place a marker point P1 on the conveyor. Align the end of the robot tool with P1. Click "Get point 1".
2. Keep the position of the marker point relative to the conveyor unchanged. Move the conveyor to P2. Align the end of the robot tool with P2, and click "Get point 2".
3. Click "Calculate" to complete automatic calculation of the resolution and direction of the encoder.



Encoder calibration

The encoder resolution verification is described as follows.

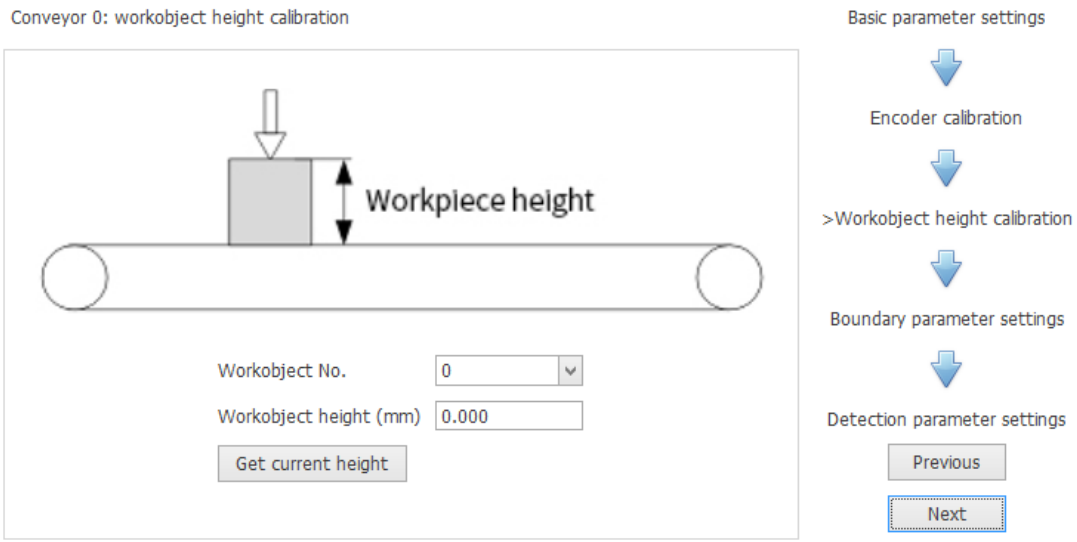
Click "Resolution verification" to enter the encoder resolution verification interface. Repeat the preceding encoder resolution calibration process once. The current resolution, resolution error value, and conveyor movement direction error are calculated automatically.



Encoder resolution verification

11.2.4.3 Workobject Height Calibration

The third step is to calibrate the height of the workobject, as shown below. The height of the workobject cannot be detected by the vision device or sensor. Therefore, you need to specify the height of each workobject type through parameter setting. The workobject height value is the Z coordinate value of the workobject datum point (i.e., origin of the object frame) under the conveyor frame.



Workobject height calibration

Follow the steps below.

1. Select the workobject number.
2. Move the end of the robot tool to the workobject datum point and click "Get current height" to read the height automatically.

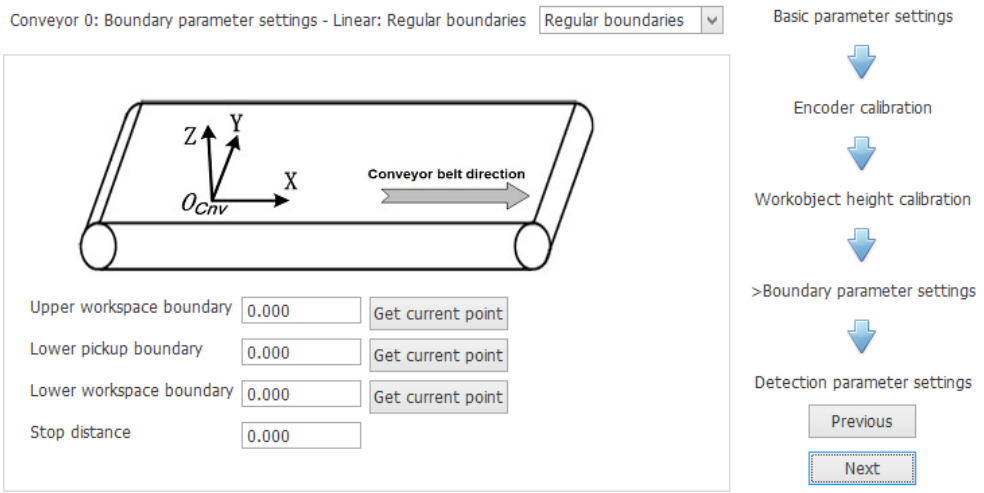
> **NOTE**
>
> If a tool is installed at the end of the robot, it is required to first activate the corresponding tool frame number.

11.2.4.4 Boundary Parameter Settings

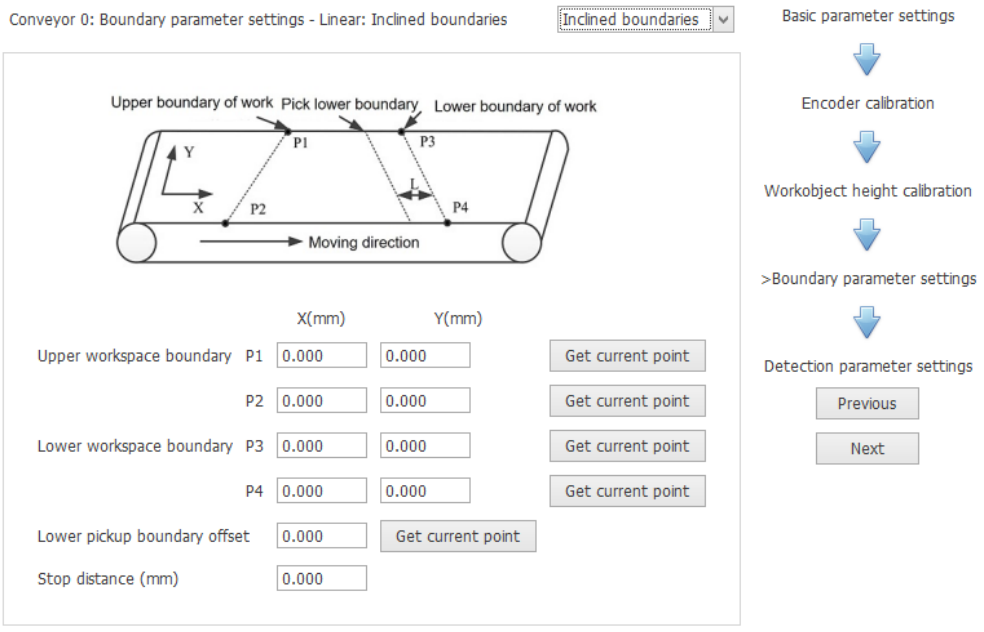
The fourth step is to set the boundary parameters. The interfaces for setting linear and circular conveyor are shown in the following figures (a and b). The linear conveyor includes regular boundaries and inclined boundaries. You can set the following parameters on the boundary parameter setting interface.

- Upper workspace boundary: Indicates the highest boundary the robot can reach, as well as the upper boundary of object search area. When the object exceeds this boundary, it can be queried by the query instruction GetCnvObject. The upper workspace boundary must be within the range of the robot's right-angle motion and away from the singular position.
- Lower workspace boundary: Indicates the lowest boundary the robot can reach. When the distance between the robot distance and this boundary is smaller than the stop distance, an alarm occurs. The lower workspace boundary must be within the range of the robot's right-angle motion and away from the singular position.

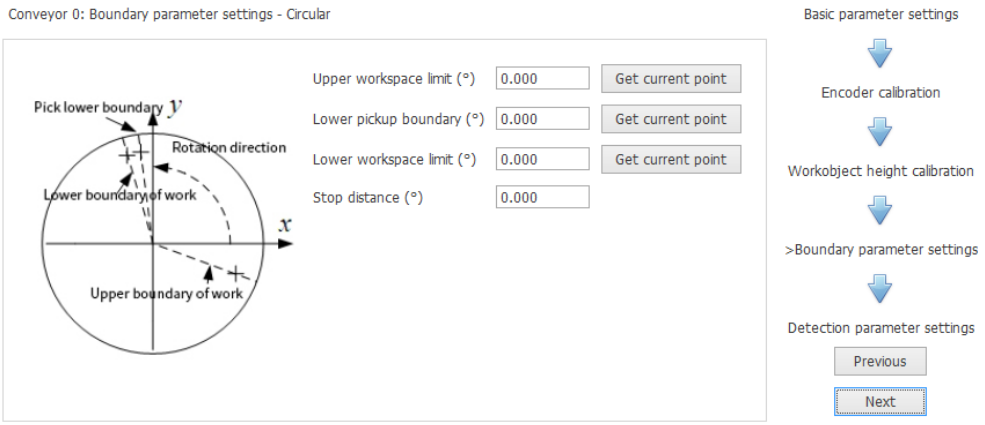
- Lower pickup boundary: Indicates the lower boundary of the object search area. When the object exceeds this limit and is not picked up, the object will be removed from the queue without being picked up. Dynamic grasping takes a certain time. If an object has exceeded the lower workspace boundary before the grasping is complete, an alarm occurs. You can set the lower workspace boundary properly to prevent this situation. When setting the lower workspace boundary, ensure that the time for the object to move from the lower pickup boundary to the lower workspace boundary is greater than the time for completing grasping.
- Stop distance: Indicates the smooth stop distance when the object approaches the lower workspace boundary. The smooth stop starts when the distance to the lower workspace boundary is smaller than this parameter.



a1. Boundary parameters for linear conveyor: Regular boundaries



a2. Boundary parameters for linear conveyor: Inclined boundaries



b. Boundary parameters for circular conveyor

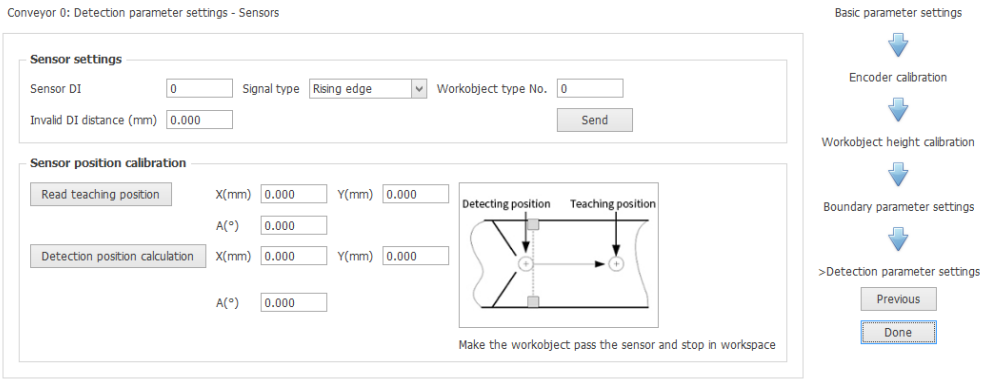
Description of boundary parameters:

- When the boundary type of the linear conveyor is set to regular boundaries, the upper workspace boundary, the lower workspace boundary and the lower pickup boundary are three lines perpendicular to the X-axis direction of the conveyor frame, represented by the x coordinate value under the conveyor frame.
- When the boundary type of the linear conveyor is set to inclined boundaries, the upper workspace boundary is a straight line defined by two points P1 and P2, and the lower workspace boundary is a straight line defined by two points P3 and P4, and the lower pickup boundary is a straight line obtained by moving the lower workspace boundary by a lower pickup boundary offset value in parallel to the opposite direction of the conveyor.
- The boundary type of the circular conveyor can only be set to regular boundaries. The upper workspace boundary, the lower workspace boundary and the lower pickup boundary are three straight lines corresponding to different angles in the conveyor frame, with the angle being positive in a counterclockwise direction.
- The above boundary parameters can be entered directly. Alternatively, you can obtain these parameters automatically by moving the robot to the target position and click the "Get current point" button. The stop distance is generally set to 5 mm or 5°.
    - After the object is visually recognized or sensed by the sensor, the object will be placed in a queue. When the object exceeds the lower pickup boundary, it will be moved out of the queue.
    - The robot starts the pick-up operation only when the workobject exceeds the upper workspace boundary. You can set the waiting point close to the upper workspace boundary to reduce the robot grasping time.
    - During the tracking process, the robot moves to the upstream offset point relative to the object frame. There are also cases where the offset point exceeds the working range.
    - Set the lower workspace boundary properly. Too large lower workspace boundary may cause the robot to straighten and damage the reducer.

11.2.4.5 Detection Parameter Settings

The fifth step is to set the detection parameters. The detection method includes sensor, vision, vision + sensor, and instructions + vision.

1. If you set the detection method to "Sensor", you need to configure the parameters as follows.



Detection parameter settings - Sensor

Sensor detection parameter settings include sensor settings and sensor position calibration.

a) The sensor settings include the following four parameters.

(1) Sensor DI: The input port number of the photoelectric sensor on the IR-Link module.

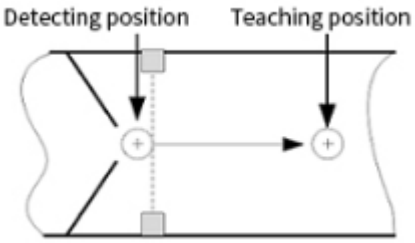
(2) Signal type: Indicates the signal edge type to output when the sensor is triggered by the workobject.

(3) Workobject type No.: One conveyor supports up to 16 types of workobjects. The workobject type cannot be identified when the sensor is used. You need to specify the workobject type.

(4) Invalid DI distance: After a valid signal is detected, if a signal change is detected again within a certain distance, it is considered that the signal is triggered by the same workobject and automatically removed as an invalid signal. This distance is called the invalid DI distance.

After setting the above parameters, click "Send" before performing sensor position calibration.

b) The purpose of sensor position calibration is to obtain the pose of the datum point of the object in the conveyor frame at the moment the sensor is triggered. That is, the detecting position on the left in the following figure. The pose of the object when the sensor is triggered is considered to be the value of "Detect position calculation".



Sensor position calibration

The detailed operations are as follows:

1) Adjust the conveyor to the normal operating speed. Place an object upstream of the sensor and move the object together with the conveyor. The sensor is triggered when the object passes the sensor. Stop the conveyor after the object moves to the robot motion range, ensuring that the sensor is triggered only once during this process. If the sensor is not triggered or is triggered more than once during this process, a warning will be displayed when you click "Detect position calculation".

2) Align the end of the tool with the datum point of the object and click "Read teaching position". In this case, the position and orientation of the robot in the conveyor frame is the pose of the object frame established on the object relative to the conveyor frame when the object passes through the sensor.

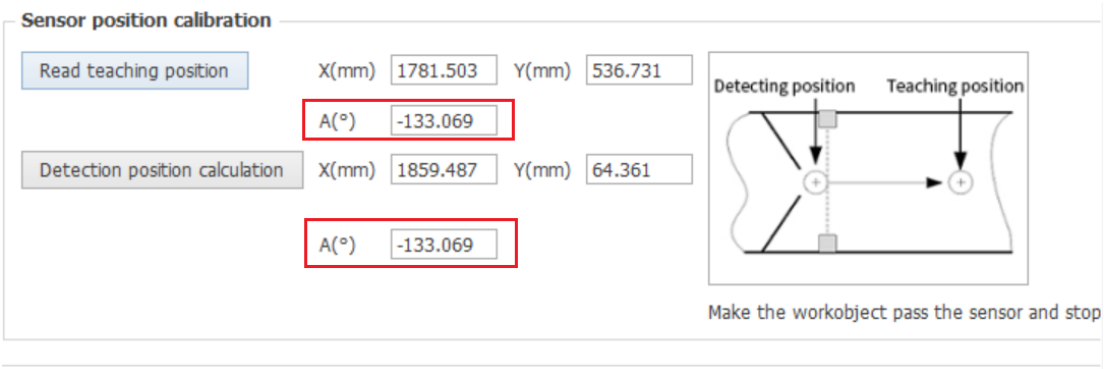
3) Click "Detect position calculation". The controller automatically calculates the pose of the object at the moment the sensor is triggered by the object.

4) After the setting is done, click "Done".

> **NOTE**
>
> - Ensure that the speed at which the workobject passes the sensor during calibration is the same as during normal operation.
> - Do not use hands or other objects to trigger the sensor during calibration.
> - If the robot is equipped with a tool at the end, select the corresponding tool number when reading the teaching position.
> - When sensor is used for detection, the orientation of the fed object is not limited. Therefore, in general, you can directly set the value A for "Detection position calculation" to 0°, and then click "Save".

In case of "Sensor" detection method for circular conveyor, due to the two possible rotation directions (clockwise and counterclockwise), the system cannot automatically calculate the object orientation angle A correctly through "Detect position calculation". If grasping orientation is required, perform manual calculation as follows:

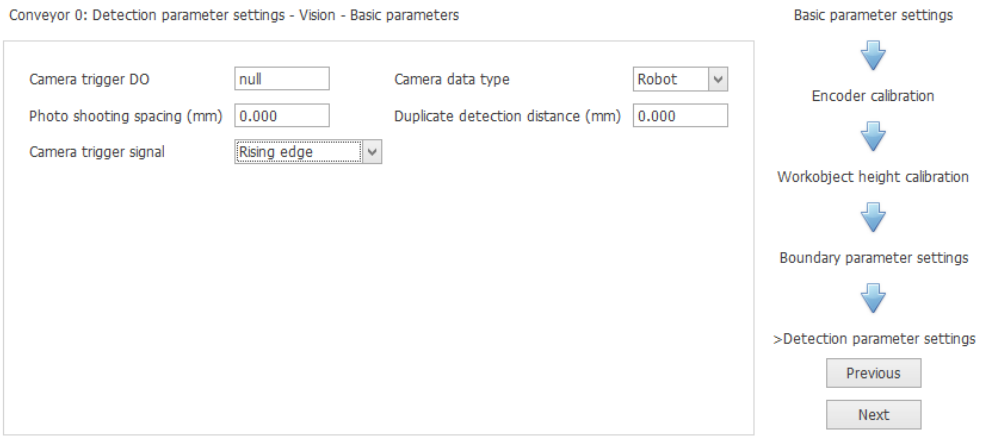
(1) Assume the "Sensor position calibration" is as shown in the figure below:



(2) Calculate the offset angle: The difference between the teaching position angle and the detection position angle = |(atan2(536.731,1781.503) - atan2(64.361,1859.487))/ pai*180 |= 14.784 ° (Calculated absolute value).

(3) Detection position A value = Teaching position A value ± offset angle. If the circular conveyor rotates counterclockwise (plus), then the value is -133.069 + 14.784 = -118.285 °; if the circular conveyor rotates clockwise (minus), the value is -133.069 - 14.784 = -147.853 °; input the final calculated A value into the A value of "Detection position".

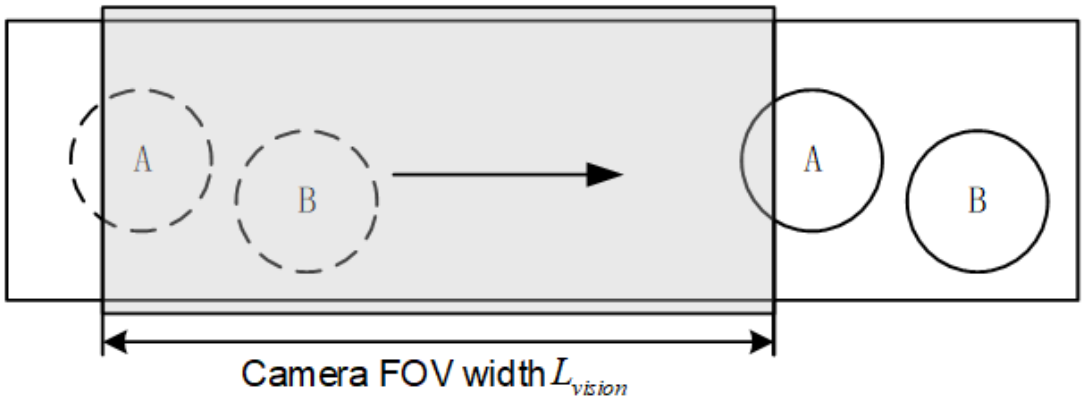
2. If you set the detection method to "Vision", you need to set the vision-related parameters. The vision settings include basic parameter settings and vision calibration. The basic parameter setting interface is as follows.



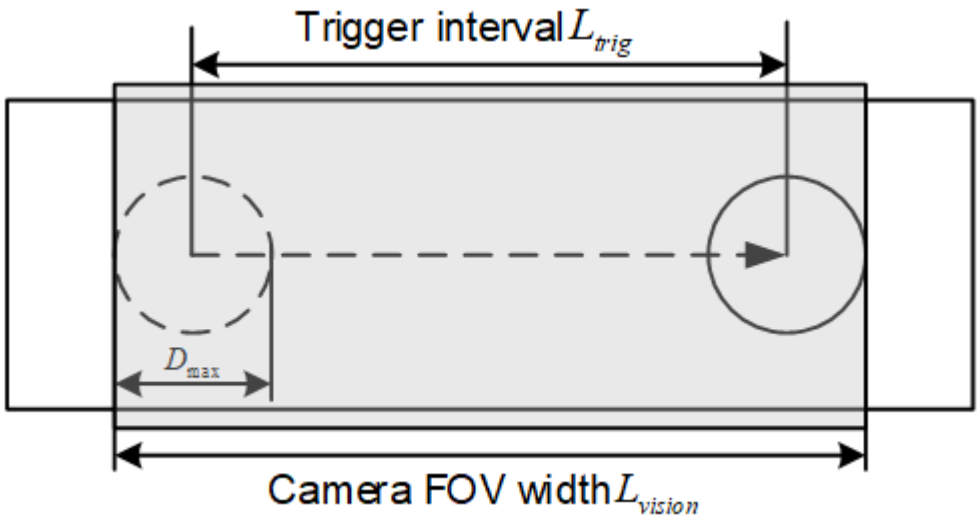
Basic camera parameters

- Camera trigger DO: The output port number of the I/O signal triggering the camera hardware to take photo on the IR-Link module.
- Camera trigger signal: Indicates the signal edge type that triggers the camera to take photos via the I/O module.
- Shooting spacing: Indicates the distance (mm) or degree (°) the conveyor moves forward to trigger the camera to take photos. Each time the conveyor moves a "Shooting spacing", the controller sets the level of the DO port corresponding to the "Camera trigger DO" to the signal type selected by the "Camera trigger signal", triggering the visual camera to take photos.

  The principle for setting the photo shooting spacing is to ensure that the identification of each object on the conveyor can be captured without missing any shots. If the set photo shooting spacing is equal to the width of camera's field of view, the camera's previous and subsequent shots are seamless, so that each object on the conveyor can be captured. If an object happens to be located at the junction of the two shots, then half of the object will be captured in each shot, and the object cannot be identified. As shown below, the workobject was located at the dashed line position during the first photo taking, and the workobject moved to the solid line position on the right during the second photo taking. Both previous and subsequent photos did not recognize the workobject A.



To avoid this situation, the previous and subsequent shots should overlap slightly, with an overlap width greater than the maximum width of the object identification, achieving at least one complete photo, as shown below.



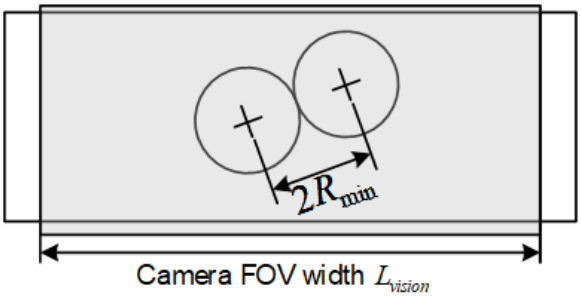
The vision system must complete image recognition within a photo shooting cycle, which is equal to photo shooting spacing/conveyor speed. Setting the photo shooting spacing too small can lead to a shorter photo shooting cycle, requiring faster vision processing speed. Therefore, the photo shooting spacing should be as large as possible without missing shots. In general, the photo cycle must be at least > 0.6s. Based on the requirements of various aspects, the photo shooting spacing must be appropriate. You can set the photo shooting spacing according to following formula. L_{trig} is the photo shooting spacing, L_{vision} is the field of view width, D_{max} is the maximum width of object identification, and δ_{vision} is the margin setting, which is generally 5 to 10 mm or °.

$$\begin{cases} L_{trig} = L_{vision} - D_{\max} - \delta_{vision} \\ \delta_{vision} < 10mm \end{cases}$$

- Duplicate detection distance: Based on the above-mentioned photo shooting spacing setting principle, there is overlap between the previous and subsequent photos, which may cause the same object to be repeatedly recognized in the two photos. The distance between the work objects at the time of the two photographs is called duplicate detection distance.

  The controller compares the two positions of the object. If the distance between the two positions is less than the distance determined by the duplicate detection distance, it is considered a duplicate and the object obtained from the subsequent shot is removed. The principle for setting the duplicate detection distance is to correctly identify one same object that is overlapped and two objects that are normally close. As shown below, two objects can be considered as two different objects when the distance is greater than $2R_{min}$, otherwise as the same object.

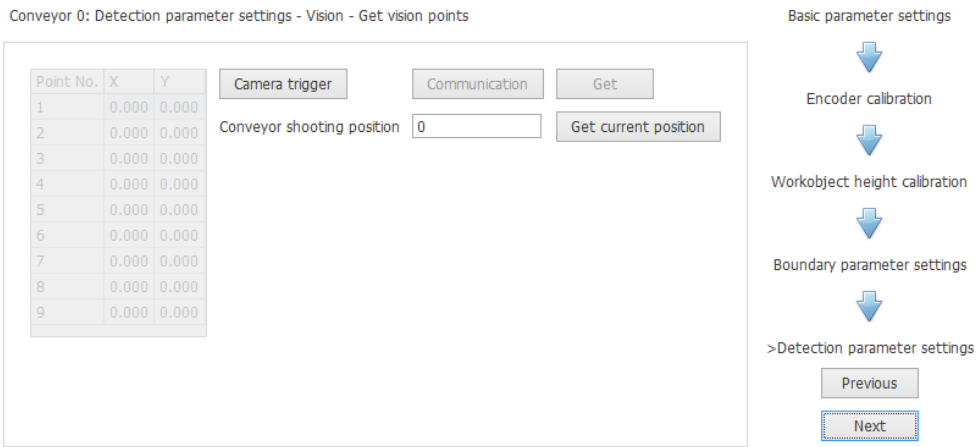
$$L_{remove} < 2R_{\min} = D_{\min}$$



- Camera data type: Pixel or Robot. Select "Pixel" when the data sent by the vision system to the robot is pixels. In this case, vision calibration is performed at the robot side. Otherwise, select "Robot". (When camera data type is set to "robot", coordinate conversion is performed in the vision system. In this case, the robot controller does not need to perform vision calibration. When vision calibration is performed in the vision software, you can still complete the acquisition of nine points using the conveyor vision calibration interface.)

Visual calibration:

If the position information sent by the vision system to the robot is pixel coordinates, it is needed to perform visual calibration in the robot controller. Click "Next" to enter the vision calibration interface after setting the basic parameters.

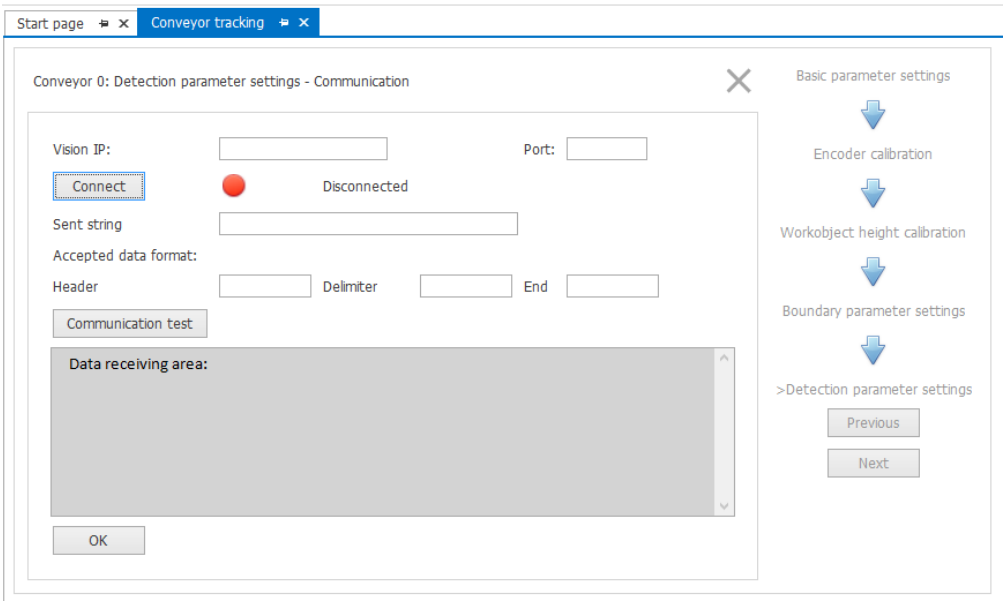


Vision calibration interface

Place the calibration board on the conveyor below the camera. Take a photo to obtain the pixel coordinates of the nine points on the calibration board. Fill in the corresponding coordinates in points on the "Get vision point" interface in sequence. Click "Get current position" to read the position information of the conveyor at the time of photo taking.

The "Communication" button allows the vision system to send the pixel coordinates of the nine points at one time to the robot. The "Get" button allows the received data to be parsed and automatically filled in the points on the "Get vision points" interface in sequence.

Click "Communication" to open the vision communication configuration page.



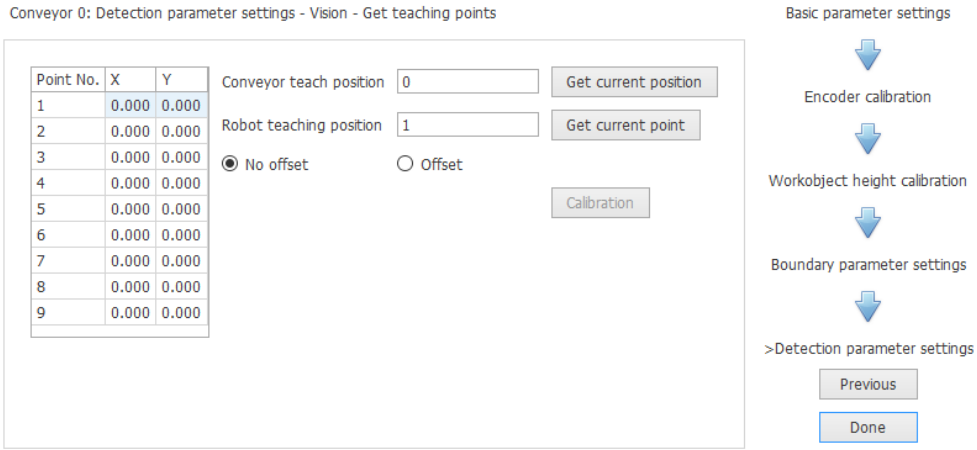
Vision communication configuration

- Enter the vision IP address and port, and then click "Connect" to establish communication.
- Sent string: When you select Ethernet trigger method, you need to set the string that triggers camera to take photos.
- The format of the data received by the vision device includes header, delimiter and end mark.

After you have completed the above settings, click "Communication test" to verify whether the robot and vision device can send and receive data properly. If data sent by the vision device has not been received in the receiving area, check the network cable and the connection between the vision device and the robot controller. The format of data sent by the vision device includes header "TA", delimiter ",", and end mark ";". The coordinates of each point are separated by ";". For example, the data sent is "TA, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200, 1300, 1400, 1500, 1600, 1700, 1800;".

After confirming that the communication test is successful, that is, the data receiving area displays the pixel coordinates of the nine points sent by the vision device, click "OK" to close the vision communication configuration page. Finally, click "Get" to complete the automatic filling of pixel coordinates in the points on the "Get vision points" interface.

After entering the pixel coordinates and reading the photo taking position, click "Next" to proceed to get the teaching points.



Visual calibration - Get teaching points

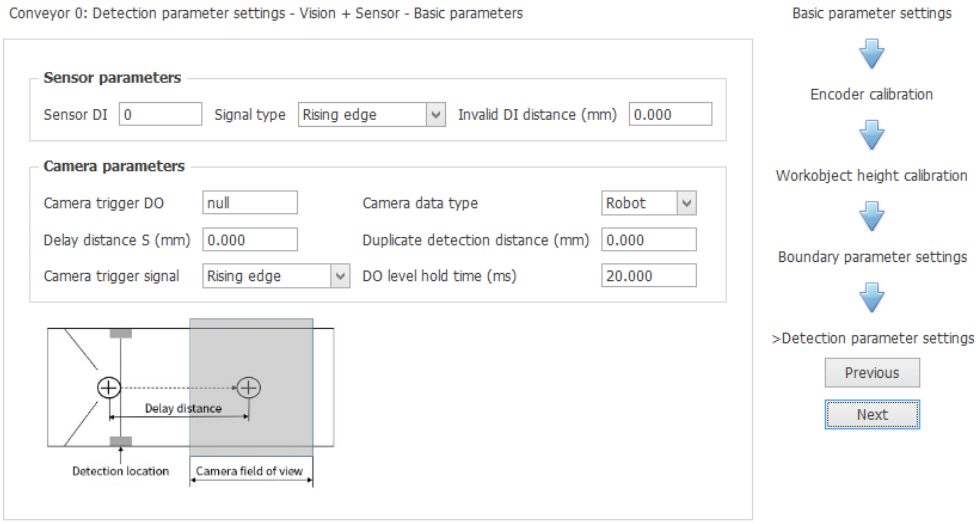
Keep the relative position between the calibration board and the conveyor unchanged. Move the conveyor and stop it when the calibration board enters the robot's motion range. Align the robot tool end to the nine points in sequence. Click "Get current point" to obtain the coordinates of the nine points. Click "Get current position" to read the position information of the conveyor at the time of teaching. After reading the current position of the conveyor and the coordinates of the nine points, click "Offset" to calculate the coordinates of the nine points in the photo-taking position. Click "Calibration" to calculate the calibration results.

After completion, click "Save" to end all settings and save the results.

NOTE

- During vision calibration, you need to activate the corresponding tool. Ensure that the activated tool is correct.
- If the camera data type is set to "Robot", you need to fill in the offset coordinate values obtained in the "Get teaching point" interface for vision calibration.

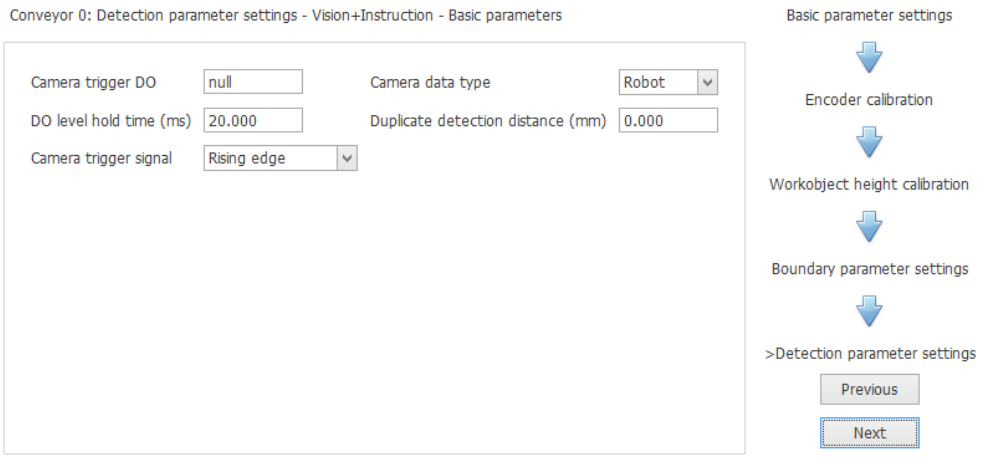
3. If you set the detection method to "Vision + Sensor", you need to set the basic parameters and perform vision calibration. The vision calibration process for the "Vision + Sensor" detection method is the same as that for the "Vision" detection method and will not be described again. The following describes the configuration of basic parameters.



Basic Vision + Sensor parameters

- Sensor DI: The input port number of the photoelectric sensor on the IR-Link module.
- Signal type: Indicates the signal edge type to output when the sensor is triggered by the workobject.
- Invalid DI distance: After a valid signal is detected, if a signal change is detected again within a certain distance, it is considered that the signal is triggered by the same workobject and automatically removed as an invalid signal. This distance is called the invalid DI distance.
- Camera trigger DO: The output port number of the I/O signal triggering the camera hardware to take photo on the IR-Link module.
- Camera data type: Pixel or Robot. For details, see the definition of "Camera data type" described for the "Vision" detection method.
- Delay distance S: When the sensor detects an object, it triggers photo taking after the conveyor moves a certain distance. This distance is called the delay distance. When the delay distance S = 0, the sensor trigger photo taking immediately when detecting the object.
- Duplicate detection distance: For details, see the definition of "Duplicate detection distance" described for the "Vision" detection method.
- Camera trigger signal: Indicates the signal edge type that triggers the camera to take photos via the I/O module.
- DO level hold time: When the level of the DO corresponding to "Camera trigger DO" is set to the signal type selected by "Camera trigger signal", the DO for a certain period of time before the level is reversed. This period of time is called the DO level holding time. The minimum value is 20 ms.

4. If you set the detection method to "Instructions + Vision", you need to set the basic vision parameters and perform vision calibration. The vision calibration process for the "Instructions + Vision" detection method is the same as that for the "Vision" detection method and will not be described again. The following describes the configuration of basic vision parameters.



Basic Vision + Instruction parameters

- Camera trigger DO: The output port number of the I/O signal triggering the camera hardware to take photo on the IR-Link module.
- Camera data type: Pixel or Robot. For details, see the definition of "Camera data type" described for the "Vision" detection method.
- DO level hold time: When the level of the DO corresponding to "Camera trigger DO" is set to the signal type selected by "Camera trigger signal", the DO for a certain period of time before the level is reversed. This period of time is called the DO level holding time. The minimum value is 20 ms.
- Duplicate detection distance: For details, see the definition of "Duplicate detection distance" described for the "Vision" detection method.
- Camera trigger signal: Indicates the signal edge type that triggers the camera to take photos via the I/O module.

11.2.5 Notes

- After the RefSys Conveyor instruction is executed, the robot keeps moving with the conveyor until the RefSys Base instruction is executed. If the RefSys Base instruction fails to be executed in time, the robot will follow the conveyor to move out of boundary. Therefore, it is necessary to ensure that RefSys Conveyor and RefSys Base instructions exist in pairs logically. If you use Goto instruction or run subprograms, return to the program in time to ensure that the RefSys Base instruction can be executed. The time of the program segment between the two instructions cannot be greater than the time the conveyor takes to move to the boundary. (The same applies to the CnvFastCatch instruction.)
- The instructions for switching frames should be used in pairs. At the end of a dynamic tracking, it is needed to switch to the base frame first. It is not allowed to directly switch between different dynamic reference frames.
  For example, RefSys Conveyor and CnvFastCatch instructions cannot be used continuously and must include a RefSys Base instruction between them. That is, the RefSys Conveyor instruction is used in pairs with the RefSys Base instruction, and the CnvFastCatch instruction is used in pairs with the RefSys Base instruction.

- The robot motion in the tracking mode follows the Cartesian trajectory. Joint motion is prohibited. After entering the synchronization mode with the conveyor, the robot remains in motion and the instruction to stop the robot will be invalid.
  If such instruction is used, running error occurs.
  - Only Movl/Movc/Movs/JumpL instructions are allowed, and other motion instructions are prohibited.
  - Do not use Until and PE parameters in motion instructions.
  - Do not use non-motion instructions WaitInPos(Fine/Corner), but WaitInPos(Value) instruction is supported.
- During the conveyor tracking process, if both the robot and conveyor run at a high speed, it is normal that a robot joint speed planning error occurs. In this case, reduce the robot speed or conveyor speed.
- If parameter Offset is used in the CP motion instruction in the tracking process, it indicates offset of the TCP under the frame of the object on the conveyor. In this case, only the offset in the X/Y/Z directions can be performed. The offset in the A/B/C directions is not superimposed.
  P[30]=(0,0,10,0,0,0),(0,0,0,0),(0,0,0,0,0,0); ## Define P[30] as a point in the object frame, with coordinates (0,0,10,0,0,0)

  PR[0]=(10,20,30,10,20,30); ## Define the offset variable as (10,20,30,10,20,30)

  RefSys Conveyor(1,Tool[2]);

  Movl Offset(P[30],PR[0]),V[100],Z[1],Tool[2],Cnv; ## Use tool 2 to move to the offset point under the object frame

  RefSys Base; ## Switch the motion reference system to the robot

  The above point Offset(P[30],PR[0]) is actually equal to (10,20,40,0,0,0). P[30] superimposes only the XYZ of PR[0] and not the ABC of PR[0]. If you want to superposition the ABC of PR[0] at the same time, you can perform the offset first, and then execute the motion instruction, for example:

  P[30]=(0,0,10,0,0,0),(0,0,0,0),(0,0,0,0,0,0); ## Define P[30] as a point in the object frame, with coordinates (0,0,10,0,0,0)

  PR[0]=(10,20,30,10,20,30); ## Define the offset variable as (10,20,30,10,20,30)

  RefSys Conveyor(1,Tool[2]);

  P[31]= Offset(P[30],PR[0]); ## The point obtained after the offset is (10,20,40,10,20,30)

  Movl P[31],V[100],Z[1],Tool[2],Cnv; ## Use tool 2 to move to the P[31] point under the object frame

  RefSys Base; ## Switch the motion reference system to the robot

  The value of P[31] is equal to (10,20,40,10,20,30).
- In the version later than S01.18, the initial orientation of the Z axis of the dynamic point coordinate is consistent with the Z orientation of the object frame. It is necessary to assign values to the orientation according to actual situation. Otherwise, a limit alarm will be triggered.
  For example, when a 6-axis robot is used to grasp materials, the flange face at the end of the robot is normally required to be vertically downwards and perpendicular to the horizontal plane of the conveyor. However, the Z-axis for establishing the conveyor frame is vertical upwards, resulting in the Z-axis of the object frame also being vertical upwards. Therefore, when defining the C value of point in the object

frame, the C value needs to be ±180° to ensure that the robot's orientation is vertical downwards when grasping materials, that is,

P[30]=(0,0,10,0,0,-180),(0,0,0,0),(0,0,0,0,0,0); ## Define P[30] as a point in the object frame, with coordinates (0,0,10,0,0,-180)

RefSys Conveyor(1,Tool[2]);

Movl P[30],V[100],Z[1],Tool[2],Cnv; ## Use tool 2 to move to the P[30] point under the object frame

RefSys Base; ## Switch the motion reference system to the robot

- The RefSys Base instruction cannot be directly followed by a motion instruction with PE parameter. If necessary, use motion instructions with a fixed position before using motion instructions with PE parameter. For example,
  RefSys Base;

  Movj P[0],V[100],Z[0];

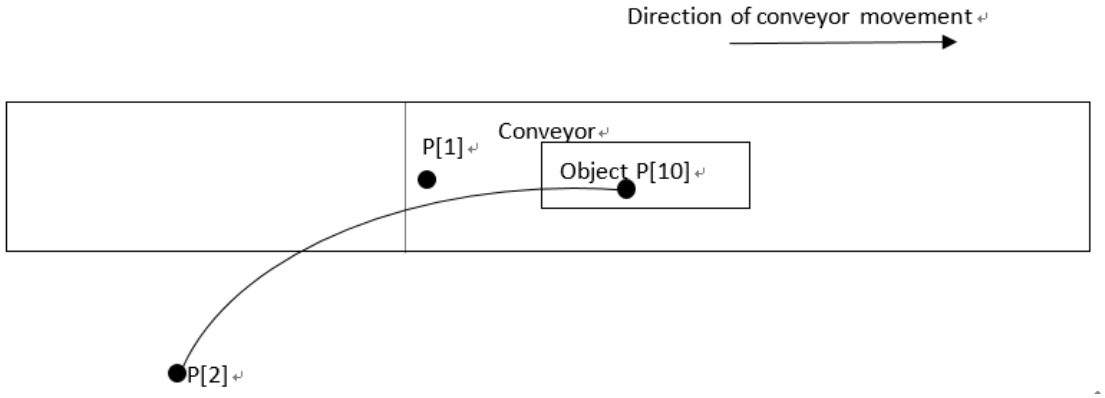
  Movl Offset(PE,PR0),V[100],Z[0];
- When the robot is not in the tracking state, it is allowed to stop the robot. In this case, data in the object queue will not be cleared and the queue will still be updated in real time (the object will be added to the queue after entering the upper workspace boundary, and removed from the queue when exceeding the lower pickup boundary.)
- CP motion instructions (Movl, Movc, and JumpL) should use the parameter Cnv in the tracking state, and the parameter Wobj in the non-tracking state. Incorrect use of the parameters will trigger alarms.
- When the timing instruction (Timeout) is used in the tracking state, the output time is incorrect. Therefore, do not use the timing instruction in the tracking state. However, it can be used in the non-tracking state.
- The parameter file of the tracking process is "FollowCraftCaliCfg.dat". If the parameter file of robot B is replaced by the parameter file of robot B, you need to restart the controller.
- The "RobHold" property of the tool used in the tracking process must be set to "True", the "RobHold" property of workobject frame must be set to "False" and "UFFix" property set to "True". Therefore, the tracking process does not support ECP mode.
- The tracking process supports no-model external axis only.
- Independent axis instructions are not supported when the robot is tracking the conveyor, and can only be used in non-tracking state.
- The configuration process is affected by EtherCAT configuration. For detailed notes, see section 9.4.4.3.5 Servo/Controller Compatibility Configuration.
- When the one-drive-many vision function is enabled, it is not allowed to use CnvVision(*, ON, *) instruction to bind port number.

11.2.6 Application Cases

11.2.6.1 Dynamic Grasping

Point Diagram

As shown in the following figure, P[10] is a point on an object. A dynamic object frame is used. P[1] is a waiting point and P[2] is a placement point. The robot waits from point P[1]. When the object passes the upper workspace boundary, it is quired through the GetCnvObject instruction and the motion reference system switches to the object frame. Then the robot moves to point P[10] under the object frame to complete the grasping of the object. After grasping the object, the robot lifts the object 10 mm off the conveyor surface, switches to the static frame, and moves to the placement point P[2] where the object is placed down.



Program

```
Start;                                    ## Program start flag
                                          ## Open the port. The external device acts as the server, with an IP 192.168.24.55 and a port 1025.
                                          ## The robot controller acts as a client and the port number is 1026.
R[0]=1026;
B[0]=0;                                   ## B[0] variable initialization
P[10]=(0,0,0,0,0,0),(0,0,0,0),(0,0,0,0,0,0);      ## Define a point under the object frame: Object Origin
P[11]=(0,0,10,0,0,0),(0,0,0,0),(0,0,0,0,0,0);     ## Define a point under the object frame: 10 mm directly above the object
While B[0]<>1
Open Socket("192.168.24.55",1025,R[0],Single,B[0]);      ## Establish a connection between the robot side and the vision device
EndWhile;
CnvVision(Conveyor[0],OFF,R[0]);          ## Turn off vision of the conveyor
CnvVision(Conveyor[0],ON,R[0]);           ## Turn on vision of the conveyor
L[0]:                                     ##Flag L[0]
```

Jump P[1],V[100],Z[0],Tool[0],Wobj[0],LH[10],MH[-50],RH[10]; ## Robot wait point, which is normally set near the upper workspace boundary

L[1]: ##Program jump flag L[1]

GetCnvObject(0,1),Goto L[1]; ## Query for the object. If the object is queried, continue to execute the following instructions; otherwise, jump to L[1]

RefSys Conveyor(0,Tool[1]); ## Switch motion reference system to the object frame on the conveyor

JumpL P[10],V[100],Z[0],Tool[1],Cnv,LH[10],MH[-50],RH[10]; ## Move to P[10] under the object frame

Delay T[1]; ## Keep the robot moving with the conveyor for 1s

Set Out[8],ON; ## Turn on the switch to suck the object

Movl P[11],V[100],Z[0],Tool[1],Cnv; ## Lift the object 10 mm off the conveyor surface

RefSys Base; ## Make the robot out of synchronization with the conveyor and switches the motion reference system to the robot

Jump P[2],V[100],Z[0],Tool[0],Wobj[0],LH[10],MH[-50],RH[10]; ## Run the robot to the placement point P[2]

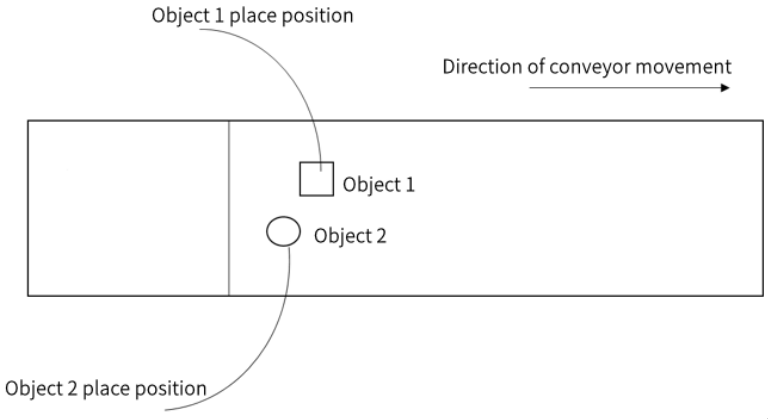
Set Out[8],OFF; ## Turn off the switch and places the object

Goto L[0]; ## Program loop

End; ## Program end flag

11.2.6.2 Dynamic Sorting

Point Diagram



When there are two or more objects on the conveyor, the vision system recognizes the attributes of the objects and sends them to the robot. The robot can sort the different objects according to the object attributes in the sent data.

Program

Start;

R[0]=1026;

B[0]=0;

P[10]=(0,0,0,0,0,0),(0,0,0,0),(0,0,0,0,0,0);

P[11]=(0,0,10,0,0,0),(0,0,0,0),(0,0,0,0,0,0);

```
While B[0]<>1
Open Socket("192.168.24.55",1025,R[0],Single,B[0]);
EndWhile;
CnvVision(Conveyor[0],OFF,R[0]);
CnvVision(Conveyor[0],ON,R[0]);
L[0]:
Jump P[1],V[100],Z[0],Tool[0],Wobj[0],LH[10],MH[-50],RH[10];    ## Robot wait point
L[1]:
GetCnvObject(0,1),Goto L[1];                                    ## Query for object #1
RefSys Conveyor(0,Tool[1]);
JumpL P[10],V[100],Z[0],Tool[1],Cnv,LH[10],MH[-50],RH[10];      ## Grasp object #1
RefSys Base;
Jump P[2],V[100],Z[0],Tool[0],Wobj[0],LH[10],MH[-50],RH[10];    ## Place the object in discharge zone #1
L[2]:
GetCnvObject(0,2),Goto L[2];                                    ## Query for object #2
RefSys Conveyor(0,Tool[1]);
JumpL P[10],V[100],Z[0],Tool[1],Cnv,LH[10],MH[-50],RH[10];      ## Grasp object #2
RefSys Base;
Jump P[3],V[100],Z[0],Tool[0],Wobj[0],LH[10],MH[-50],RH[10];    ## Place the object in discharge zone #2
Goto L[0];
End;
```

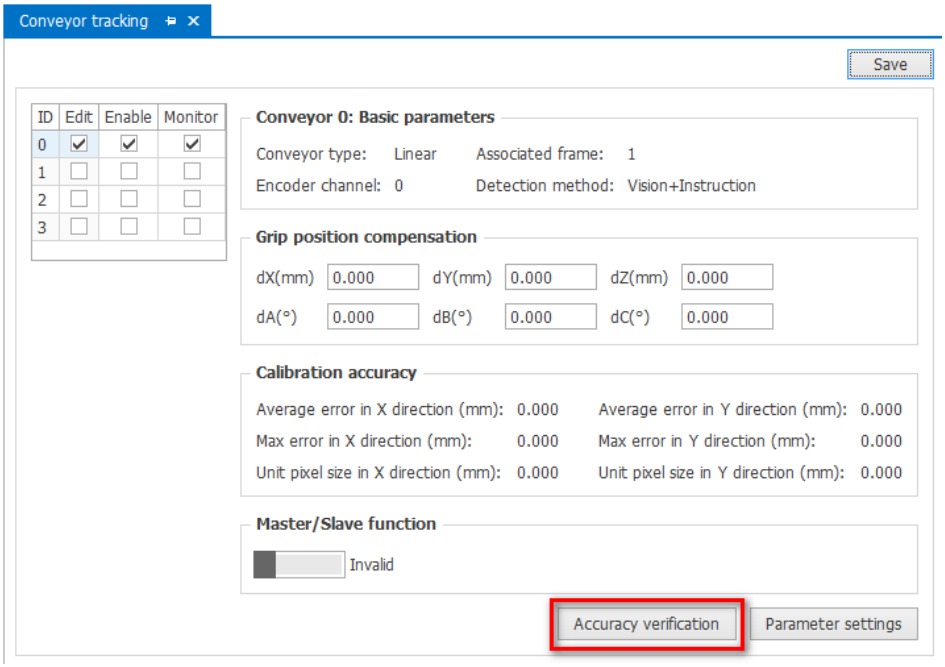


- Dynamic sorting supports 16 different types of objects, and the data of up to 50 objects can be sent during photo taking.
- The objects for sorting need to be of similar size.

11.2.7 Auxiliary Functions

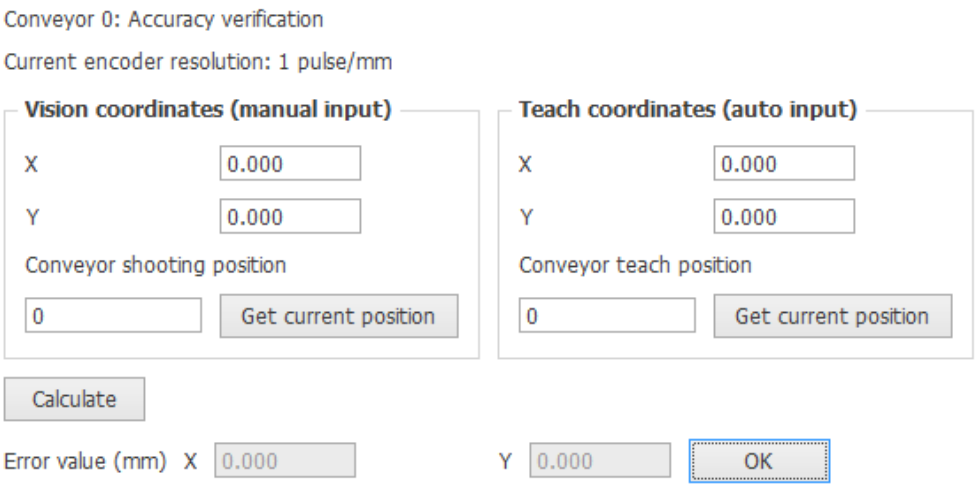
11.2.7.1 Accuracy Verification

After setting all parameters of the tracking process, you can perform error calculation on the calibration accuracy of the whole tracking process through the accuracy verification function. The error calculation covers the encoder resolution calibration, conveyor frame calibration, and vision calibration. You can access the "Accuracy verification" function in "Station" > "Controller" > "System resources" > "Process module" > "Conveyor tracking". The main interface is shown below.



Conveyor tracking parameter settings

Click the "Accuracy verification" button to access the accuracy verification function.



Accuracy verification interface

The verification process is as follows.

1. Stop the conveyor. Place a marker point that can be recognized by the vision device on the conveyor below the camera. Take a photo to obtain the pixel coordinates (X,Y, A) of the marker point and fill in the "Vision coordinates (manual input)" box, and click "Get current position" under "Conveyor shooting position" to record the encoder value.
2. Move the conveyor with the position of the marker point relative to the conveyor unchanged. After the marker point enters the robot's motion range, stop the conveyor. Align the robot tool end to with the marker point and click "Get current position" under "Conveyor teach position". The coordinates of this marker point under the conveyor frame and the encoder value are obtained automatically.
3. Click "Calculate" to automatically calculate the vision coordinate conversion values, moving distance, and overall accuracy error. (Error Value = Teaching Coordinate - Offset Vision Coordinate).

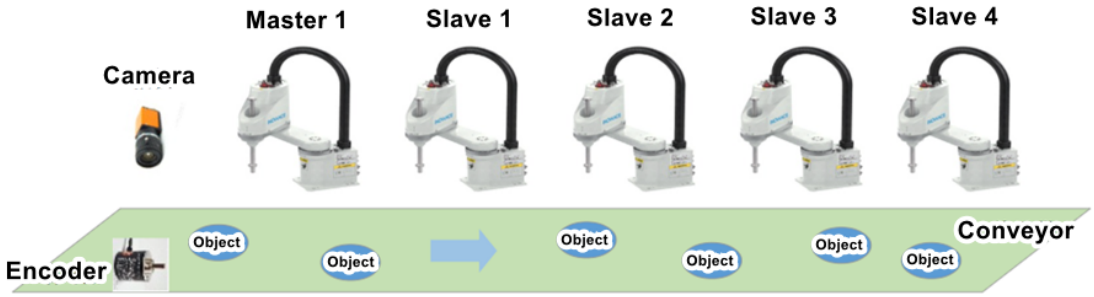
> **CAUTION**
>
> If a tool is installed at the end of the robot, you need to activate the corresponding tool frame number.

11.2.7.2 One-Drive-Many Vision Function

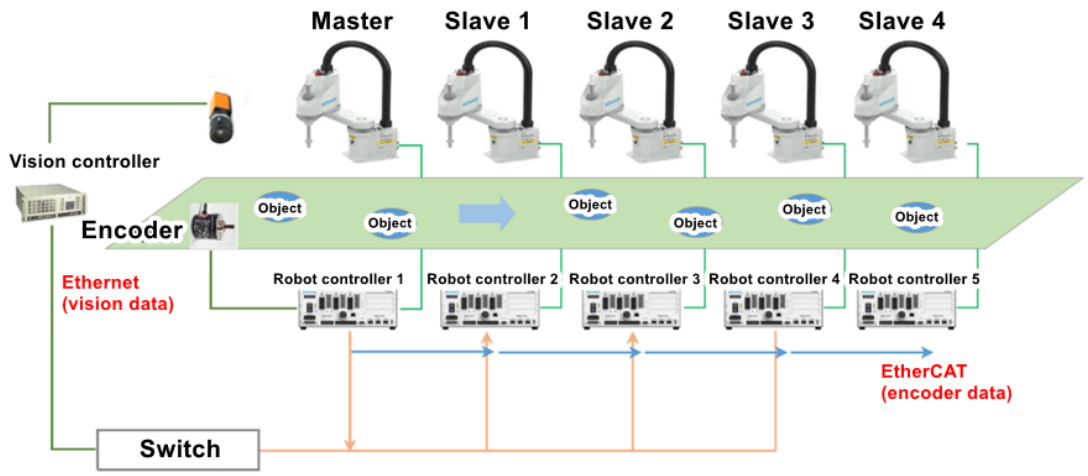
Overview

In the normal conveyor tracking process, one robot requires one sensor plus one encoder to grasp the objects on the conveyor.

With the one-drive-many vision function, at most five robots only require one camera plus one encoder to grasp the objects on the conveyor. The object missed by one robot will be grasped by the next robot.



Hardware Configuration



- Connect the encoder:

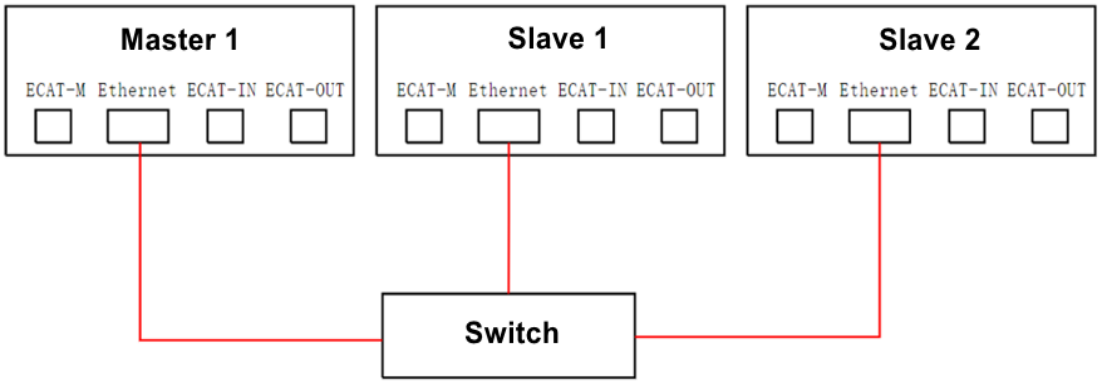
Connect the encoder to the master robot controller. For details, see section 11.2.2.1 Connecting the Encoder.

- Install the camera and I/Os:

Connect the I/O trigger cable of the camera to the I/O terminal of the master robot controller. For details, see the 11.2.2.3 Vision Device section.

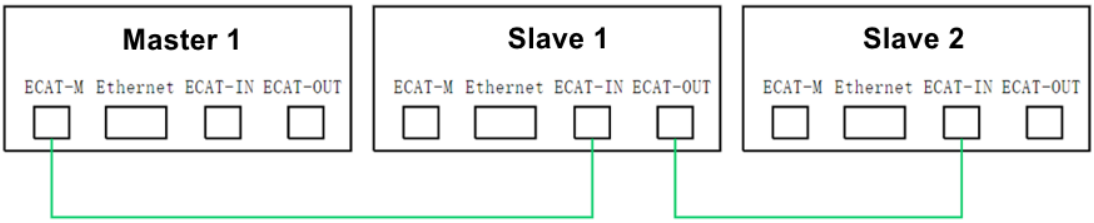
- Configure the TCP protocol:

The sharing of camera data is based on the TCP/IP protocol. To ensure normal communication between the master and slave robot controllers, it is needed to configure the IP addresses of all controllers to be on the same network segment.



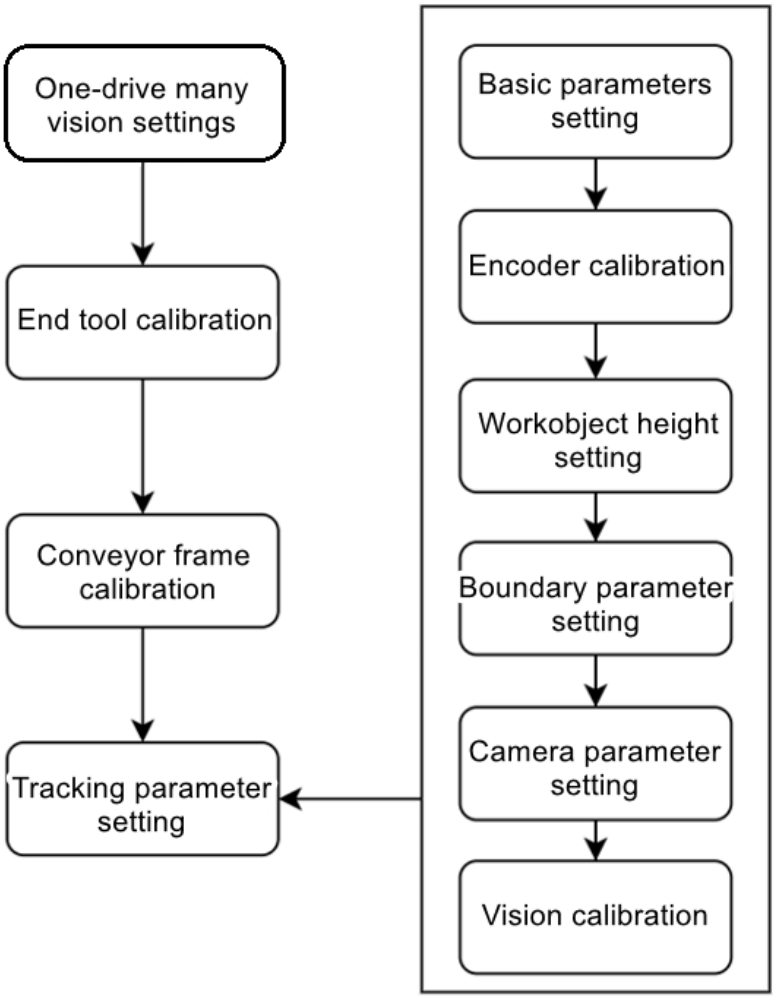
- EtherCAT:

The sharing of encoder data is based on the EtherCAT protocol. To ensure sharing of encoder data, make sure all devices are connected according to the figure shown below.



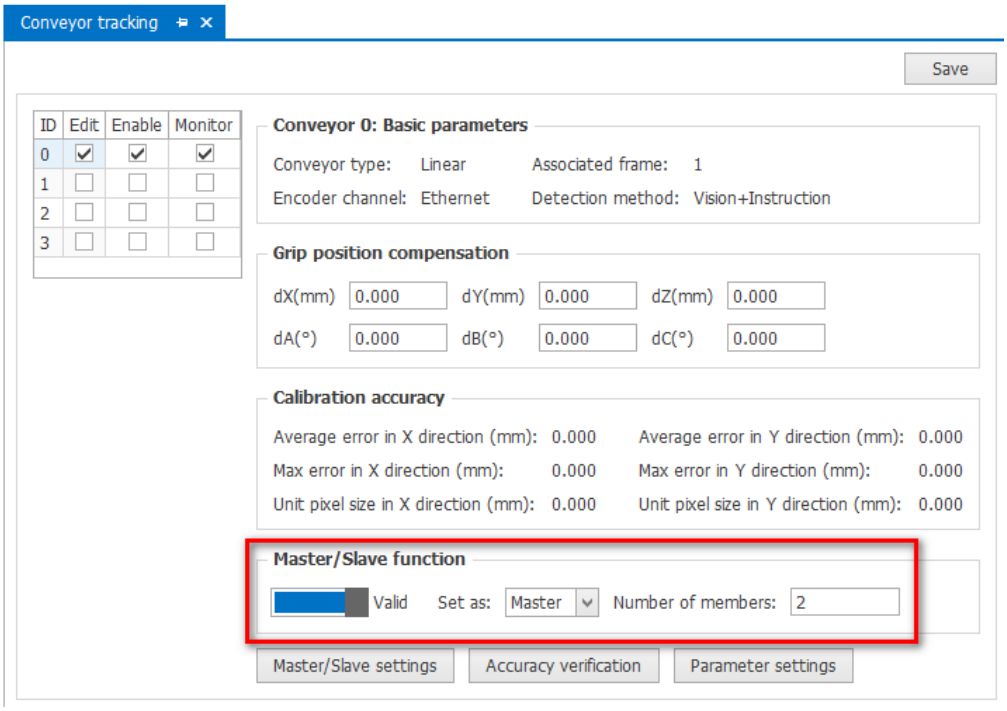
Master robot configuration:

The end tool calibration, conveyor coordinate calibration, and the tracking process parameters for the one-drive-many vision function are the same as those for the regular tracking process.

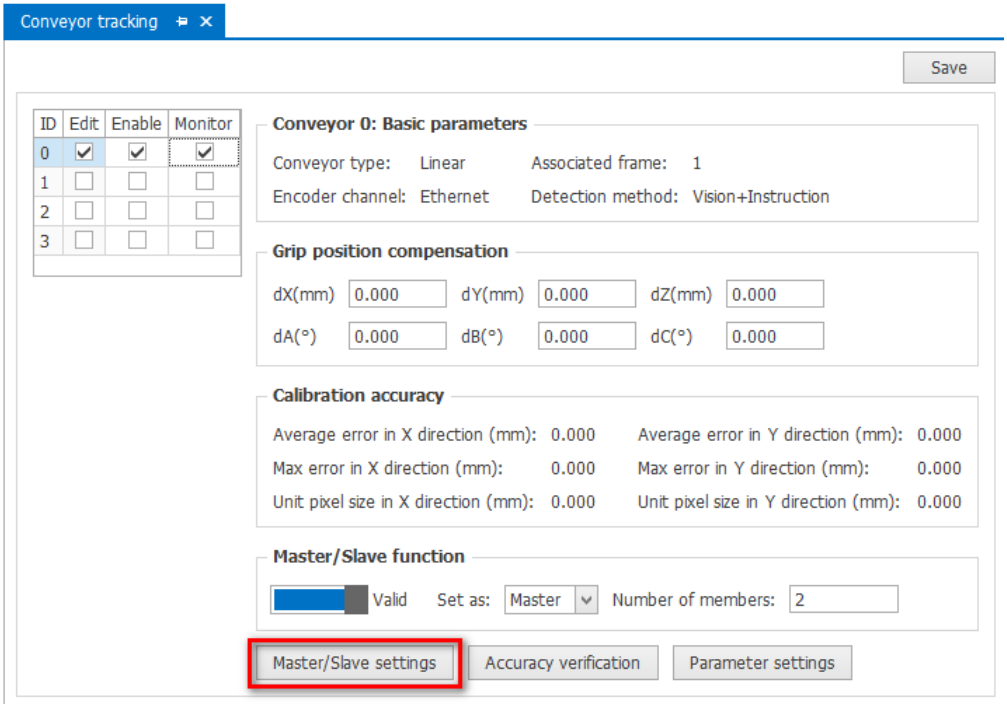


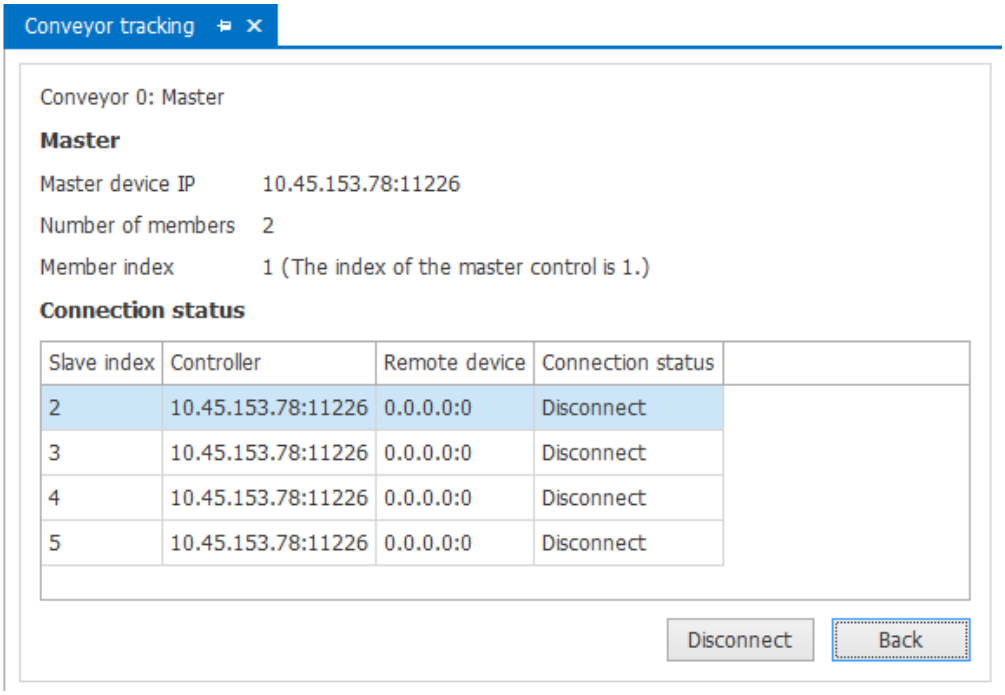
- One-drive-many vision function configuration:

Turn on "Master/Slave function", set the robot as "Master", set "Number of members", and click "Save". Then restart the controller.



You can click "Master/Slave settings" to view the TCP connection status of the slaves.





CAUTION

- The number of members refers to the sum of master and slave robots.
- After setting a robot as the master, you must restart the controller of that robot.
- Currently, you can enable the master/slave function for only one conveyor.
- It is not allowed to modify the IP address of the controller after the one-drive-many vision function is enabled. It is recommended to use a fixed IP address.

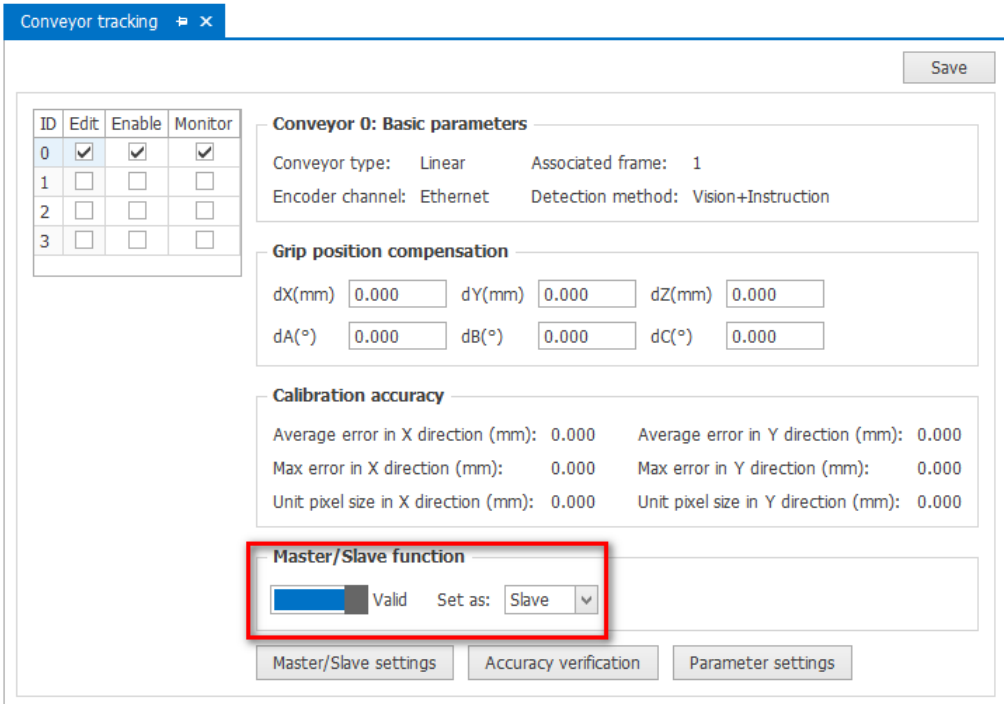
- Tracking process configuration:

The end tool calibration, conveyor coordinate calibration, and the tracking process parameters for the one-drive-many vision function are the same as those for the regular tracking process.

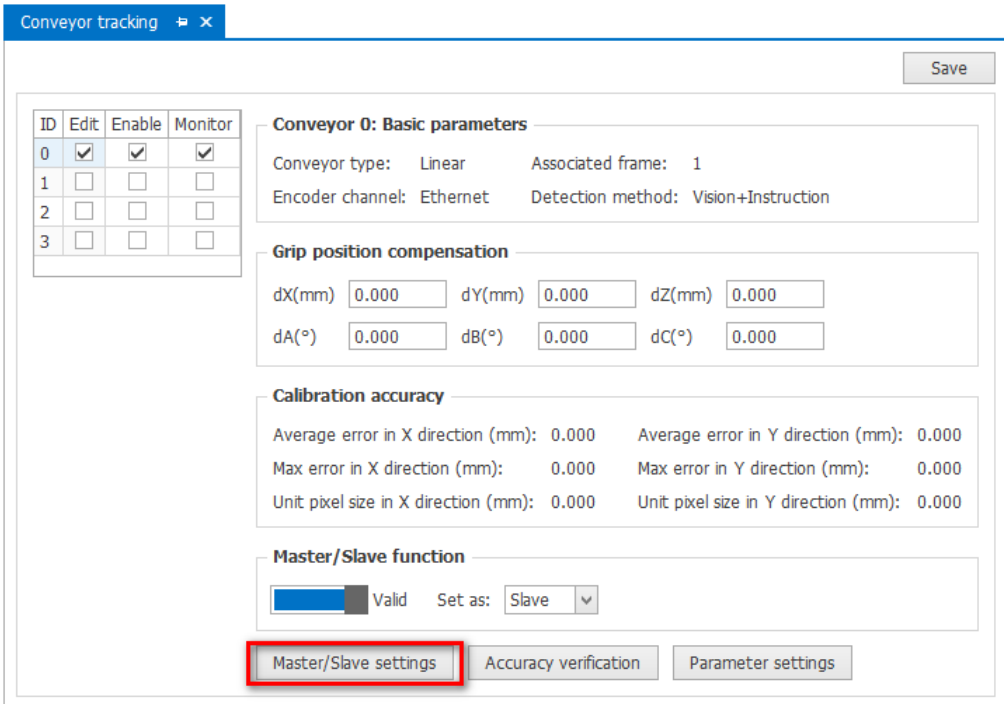
Slave robot configuration

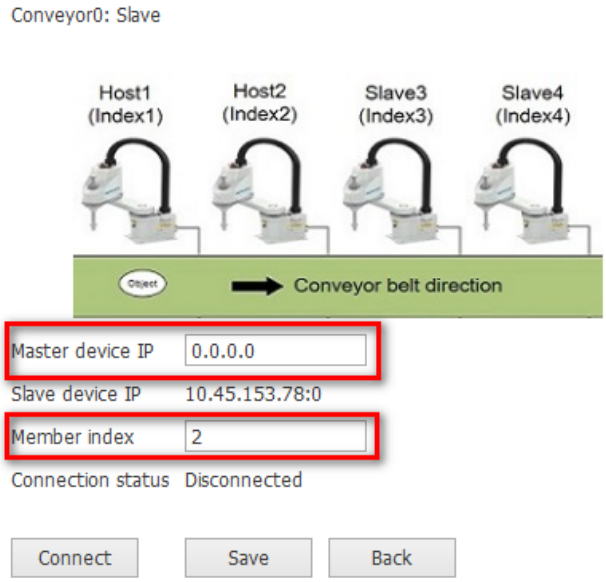
- One-drive-many vision function configuration:

Turn on "Master/Slave function", set the robot as "Slave", set "Number of members", and click "Save". You do not need to restart the controller.



Click "Master/Slave settings", set the "master IP" and "Member index", and click "Connect".





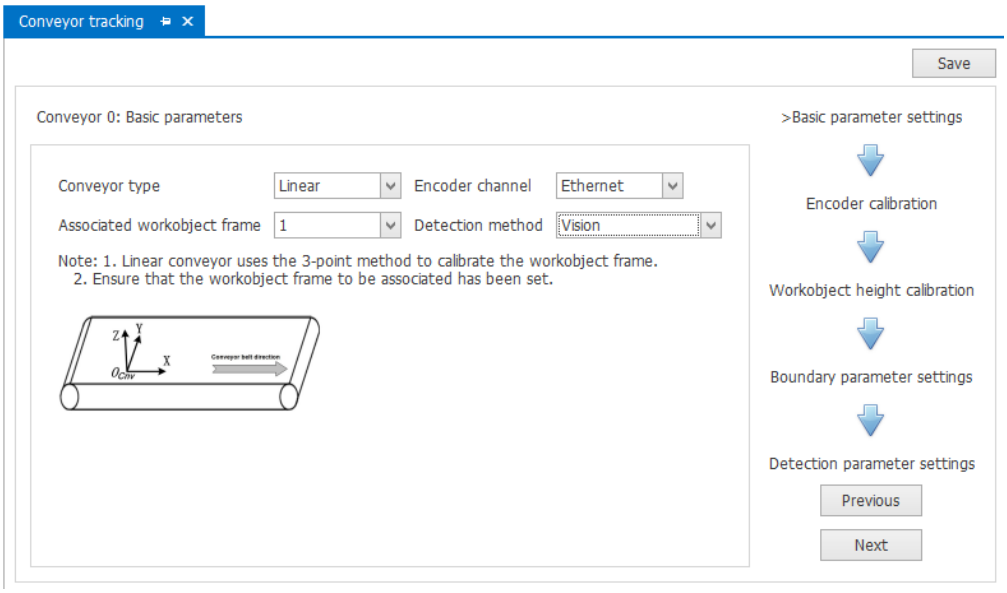
CAUTION

- After setting a robot as the slave, you do not need to restart the controller of that robot.
- Currently, you can enable the master/slave function for only one conveyor.
- It is not allowed to modify the IP address of the controller after the one-drive-many vision function is enabled. It is recommended to use a fixed IP address.
- The member index must be set from small to large according to the conveyor direction. It is not allowed to set different robots to the same index.

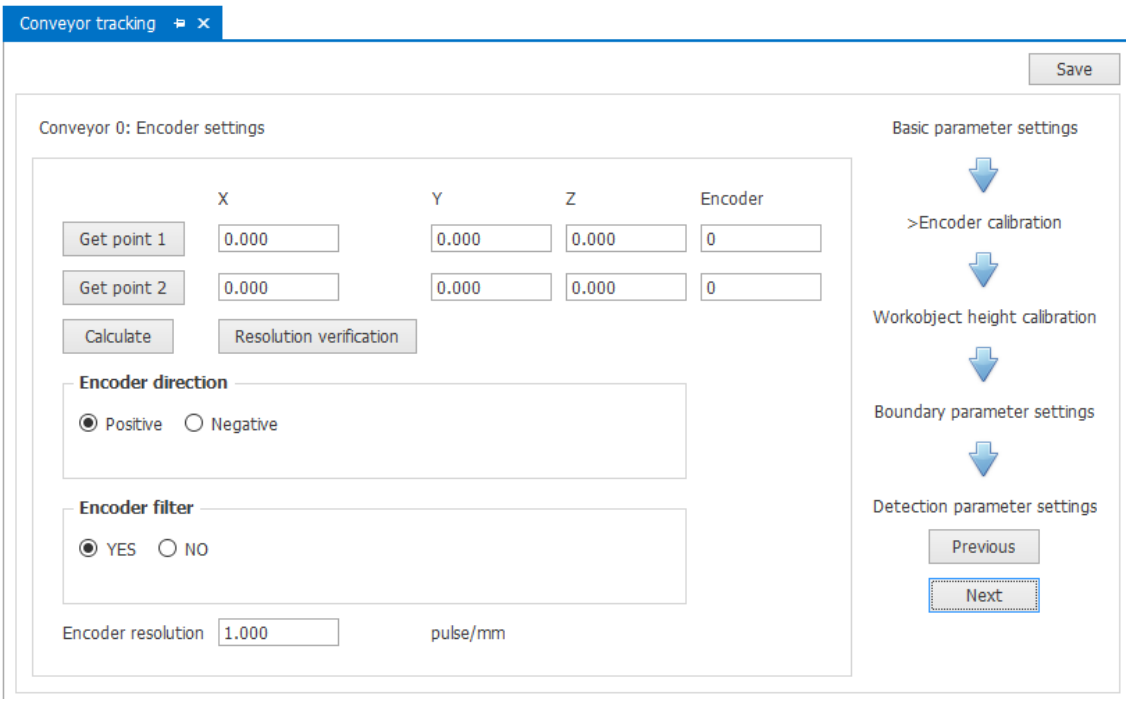
- Tracking process configuration:

The end tool calibration, conveyor coordinate calibration, and the tracking process parameters for the one-drive-many vision function are basically the same as those for the regular tracking process, with certain slight differences.

When a robot is set as a slave, the "Conveyor type" is fixed to "Linear", "Encoder channel" parameter is fixed to "Ethernet" and the "Detection method" is fixed to "Vision".

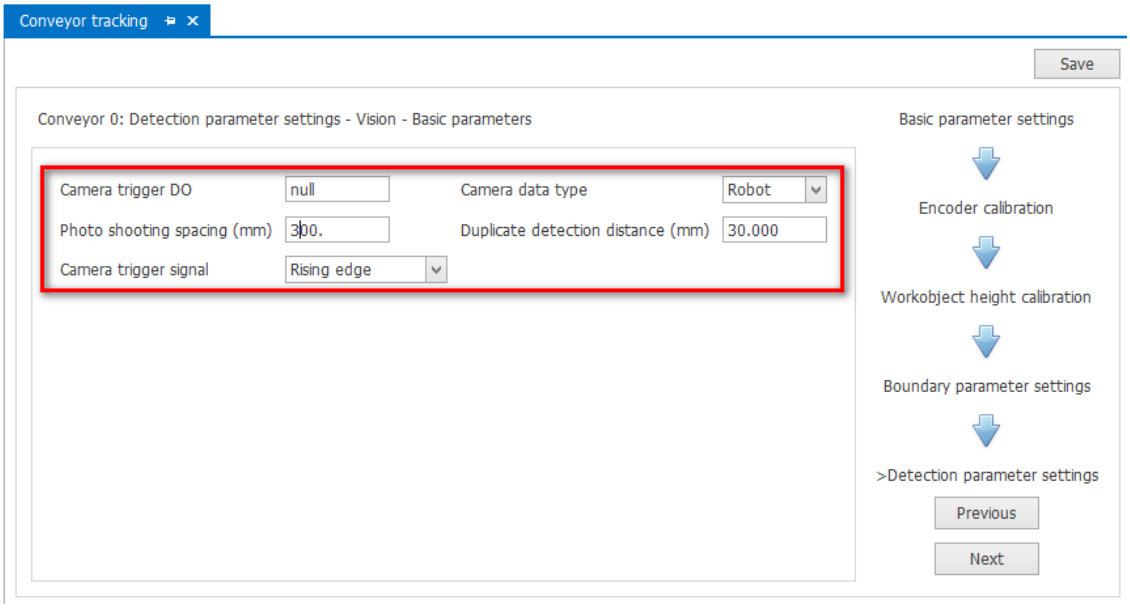


The tracking process needs to use the encoder value. Before configuring the one-drive-many vision function of the slave robots, it is recommended that you connect the master and slave robots via EtherCAT and configure the one-drive-many vision function of the master robot, so that all slave robots can acquire the conveyor encoder value.



When set as the master, "Camera data type" in the "Detection parameter settings" only supports selection of "Pixel".

When set as the slave, parameters in the "Detection parameter settings" cannot be modified.



Vision software configuration

See section 11.2.7.2 One-Drive-Many Vision Function for setting the vision software. Only the master robot receives data sent from the vision controller. Connect the trigger I/O of the vision controller to I/O terminal of the master robot controller. The vision controller acts as a server and the master robot acts as a client.

Example program:

The robot starts from wait point P[0], picks up an object from the conveyor (P[30] fixed to the object), and then moves to P[1] to complete the task. The task flow is repeated. The object missed by one robot will be picked up by the next robot.

- Master robot program:

```
Start;
## Open the port. The external device acts as the server, with an IP 10.45.151.59 and a port 2000.
## The local controller acts as a client and the port number is 1234.
B[0] =0;
While B[0] <> 1
Open Socket("10.45.151.59",2000,1234,Single,B[0]);
EndWhile;
CnvVision(Conveyor[1],OFF,1234);
CnvVision(Conveyor[1],ON,1234);          ## Open the vision port of conveyor 1, client port number 1234
P[30]= (0,0,10,0,0,0),(0,0,0,0),(7,0,0);          ## Define P[30] as a point in the object frame, with coordinates (0,0,10,0,0,0)
L[0]:
Movj P[0],V[100],Z[0],Tool[0];          ## Move to the wait position
L[1]:
GetCnvObject(1,0), Goto L[1];          ## Query the queue for object of type 0 on conveyor #1
RefSys Conveyor(1,Tool[2]);          ## Switch the motion reference frame to the object that has just been found
Movl P[30],V[100],Z[1],Tool[2];          ## Use tool 2 to move to the P[30] point under the object frame
Delay T[1];          ## Keep the robot moving with the conveyor for 1s
Set Out[1],ON;          ## Turn on the switch to suck the object
RefSys Base;          ## Switch the motion reference system to the robot
Jump P[1],V[100],Z[0],Tool[0],LH[10],MH[-750],RH[10];   ## Move the object to P[1]
Set Out[1],OFF;          ## Place the object
Goto L[0];
End;
```

- Slave robot program

```
Start;
## Open the port. The external device acts as the server, with an IP 10.45.151.59 and a port 2000.
## The local controller acts as a client and the port number is 1234.
B[0] =0;
CnvVision(Conveyor[1],OFF,1234);     ## Turn off vision of the conveyor #1
```

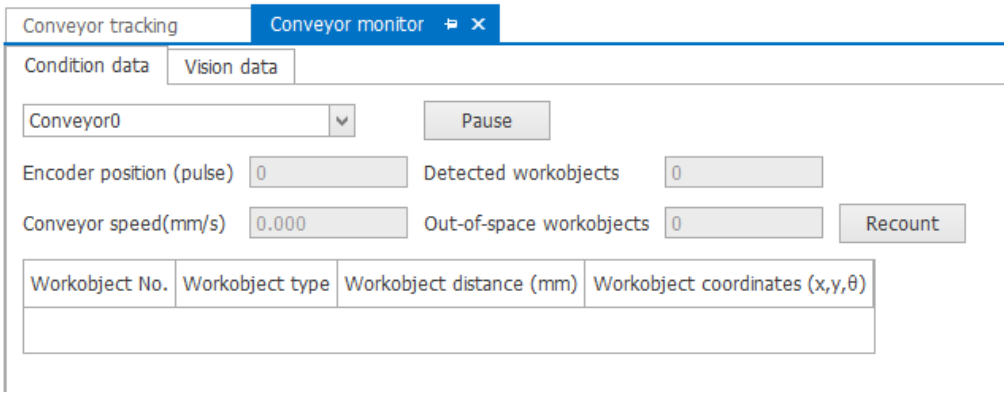
```
P[30]= (0,0,10,0,0,0),(0,0,0,0),(7,0,0);      ## Define P[30] as a point in the object frame, with coordinates (0,0,10,0,0,0)
L[0]:
Movj P[0],V[100],Z[0],Tool[0];                ## Move to the wait position
L[1]:
GetCnvObject(1,0), Goto L[1];                 ## Query the queue for object of type 0 on conveyor #1
RefSys Conveyor(1,Tool[2]);                   ## Switch the motion reference frame to the object that has just been found
Movl P[30],V[100],Z[1],Tool[2];               ## Use tool 2 to move to the P[30] point under the object frame
Delay T[1];                                   ## Keep the robot moving with the conveyor for 1s
Set Out[1],ON;                                ## Turn on the switch to suck the object
RefSys Base;                                  ## Switch the motion reference system to the robot
Jump P[1],V[100],Z[0],Tool[0],LH[10],MH[-750],RH[10];   ## Move the object to P[1]
Set Out[1],OFF;                               ## Place the object
Goto L[0];
End;
```



- Avoid interference and collision risks between robots:
  - When conditions permit, ensure that the motion ranges of different robots do not physically interfere with each other.
  - If the above conditions cannot be met, it is necessary to reasonably set the tracking process-boundary parameters for each robot (upper workspace boundary, lower workspace boundary, stop distance) so that there is an appropriate spacing between the working spaces of each robot on the conveyor for grasping tasks.
- Before running the one-drive-many vision program on the master robot, check that the TCP connections of all the slave robots are normal. If a slave robot is not connected, an error occurs when running the one-drive-many vision program on the master robot.

11.2.7.3 Monitoring Interface

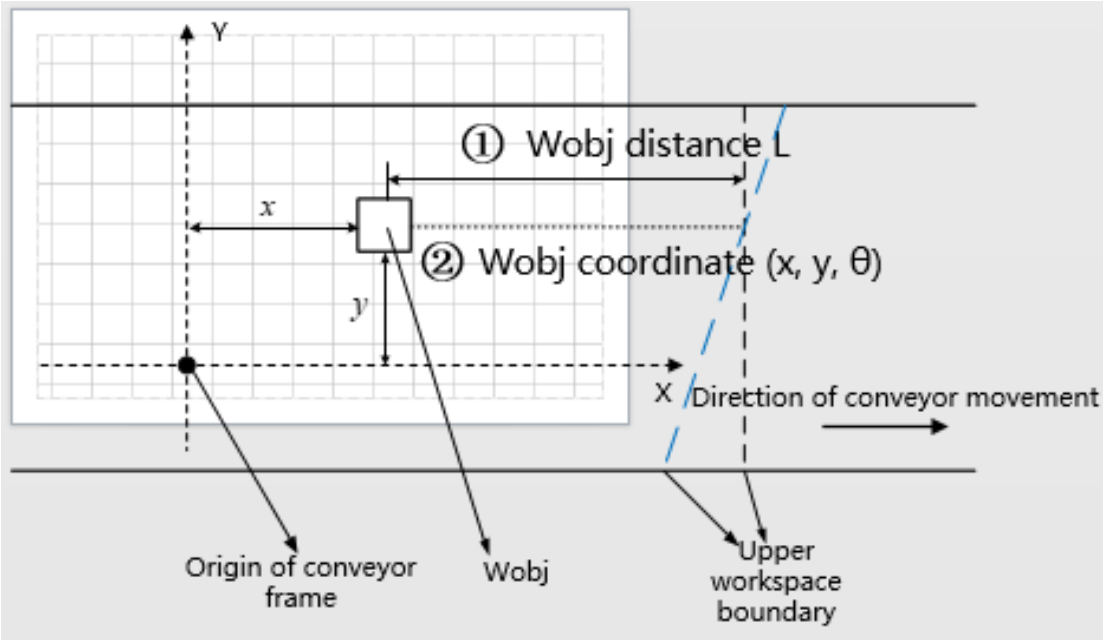
If you click the "Monitor" button for a conveyor on the "Conveyor tracking" setting page, you can view the real time monitoring information of the selected conveyor in "Monitor" > "Conveyor monitor" > "Condition data". The monitor displays various monitoring information such as conveyor speed, the number of detected workobjects etc. You can switch monitoring of different conveyors.



Monitoring interface for the conveyor tracking process

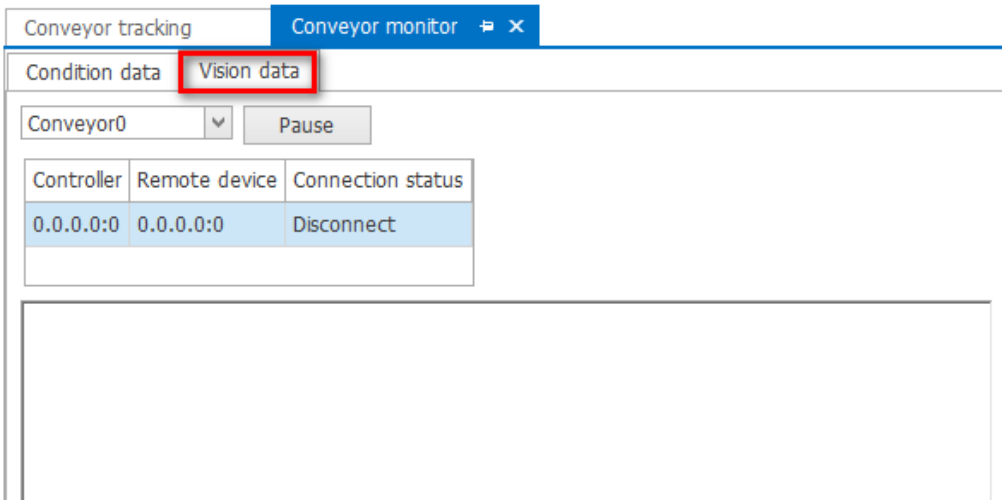
The following information is displayed on the monitoring interface of the conveyor tracking process.

- Encoder position: The encoder value associated with the conveyor, in pulse.
- Conveyor speed: The moving speed of the conveyor, in mm/s or °/s.
- Detected workobjects: The number of workobjects in the workobject queue, that is, the total number of workobjects between the sensor or camera's field of view and the lower pickup boundary.
- Out-of space workobjects: The total number of workobjects that have been detected by the sensor or camera, but failed to be picked up and thus exceeding the lower pickup boundary, from the start of the program to the current time.
- You can click the "Recount" button on the right to clear the current count and start counting again.
- Queue of workobjects to be picked up (Up to the first 10 workobjects can be displayed)
  - Workobject No.: The workobjects are ordered in the queue according to their positions on the X axis of the conveyor frame. The workobject furthest downstream of the conveyor is displayed first. Workobject #1 represents the next workobject to be picked up by the robot. It is the workobject furtherest downstream between the sensor or camera's field of view and the lower pickup boundary. When a workobject is retrieved by GetCnvObject or CopyCnvObject instruction or it exceeds the lower pickup boundary, the workobject is automatically removed from the workobject queue. The value ranges from 0 to 10.
  - Workobject type: The type of workobject. The value ranges from 0 to 15.
  - Workobject distance: The distance L (① as shown below) between the workobject on the conveyor and the upper workspace boundary. This distance is negative when the workobject is located upstream of the upper workspace boundary, and positive when it exceeds the upper workspace boundary. When the upper workspace boundary is diagonal, the workobject distance is the distance between the current position of the workobject and the intersection point of the horizontal line in the X direction of the conveyor frame and the upper workspace boundary. Unit: mm or °.
  - Workobject coordinates (x, y, θ): The coordinate values of the workobject in the conveyor frame (② as shown below).



- Pause/Refresh button: By default, the button displays "Pause". After you click the "Pause" button, the data on the monitoring interface remains unchanged. The button name changes from "Pause" to "Refresh". When you click the "Refresh" button, the monitoring data is refreshed at a rate of 1s.

Click "Vision data" tab to switch to the vision data monitoring interface:



The vision data monitoring interface displays the following information.

- Controller: The IP address of the current controller.
- Remote device: The IP address of the remote device.
- Connection status: The status of connection with remote device.
- Conveyor coordinates: The spatial coordinate values of the workobject in the conveyor frame.
- Pixel coordinates: The pixel coordinate values of the workobject in the camera frame.
- Pause/Refresh button: By default, the button displays "Pause". After you click the "Pause" button, the data on the monitoring interface remains unchanged. The button name changes from "Pause" to "Refresh". When you click the "Refresh" button, the monitoring data is refreshed.

12 Function Applications

12.1 ECP Function

12.1.1 Overview

The robot generally operates in TCP mode. In the TCP mode, the robot holds a tool and moves in a frame of a workobject that is not held by the robot. However, when the size or mass of the machining tool/equipment is large, the robot cannot hold the tool for operation due to limits of its motion space and rated load. In this case, you can choose to fix the large volume and high weight machining tool relative to the ground, and configure the robot to hold the workobject to complete the machining. In this situation, the robot works in ECP mode.

12.1.2 Notes

- In this section, when the robot works in TCP mode, set the "RobHold" parameter of the tool to "True" and the "RobHold" parameter of the workobject to "False". That is, the tool is held by the robot and fixed relative to the flange, and the workobject is not held by the robot and fixed relative to the world frame.
- When the robot works in ECP mode, set the "RobHold" parameter of the tool to "False" and the "RobHold" parameter of the workobject to "True". That is, the tool is an external tool and not held by the robot and fixed relative to the world frame, and the workobject is held by the robot and fixed relative to the flange.
- In both TCP and ECP modes, make sure the "RobHold" parameter of the workobject and tool are correctly configured. Otherwise, an alarm occurs that the activated tool and workobject parameters do not match.
- The ECP mode supports no-model external axis only.
- In ECP mode, the load properties of the activated external fixed tool are those of the activated workobject whose "RobHold" parameter is set to "True".
- If there is a transition between TCP and ECP movements, the transition does not take effect.
- ECP does not support the Jump/JumpL instruction or dynamic tracking function, which when used, causes an alarm.

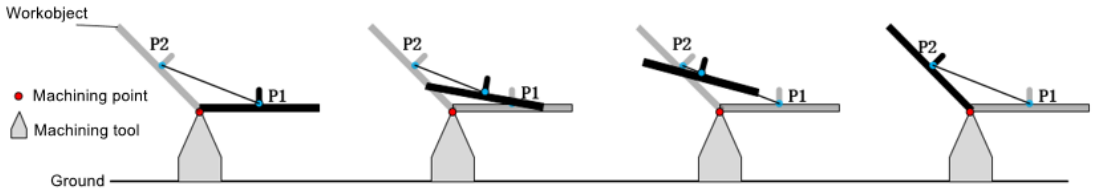
12.1.3 Calibration of the ECP Frame

See the 9.3.1 Tools section and the 9.3.2 Workobject section.

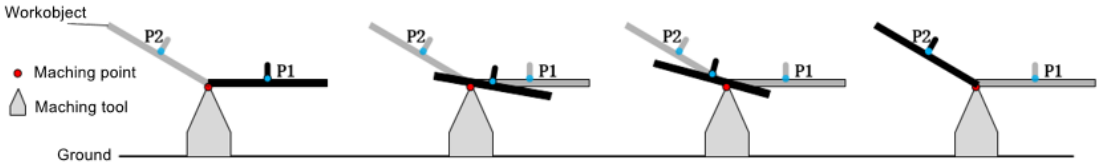
12.1.4 Motion in ECP Mode

In contrast to the TCP mode, the ECP mode simplifies the teaching process as the machining tool is fixed relative to the world frame (that is, the tool is an external fixed tool). The robot holds the workobject and moves around the external fixed tool, performing linear, circular, and other movements on the surface of the workobject for machining.

Here we take machining a straight line on the workobject surface as an example to illustrate the differences between the motion in ECP mode and the motion in TCP mode.



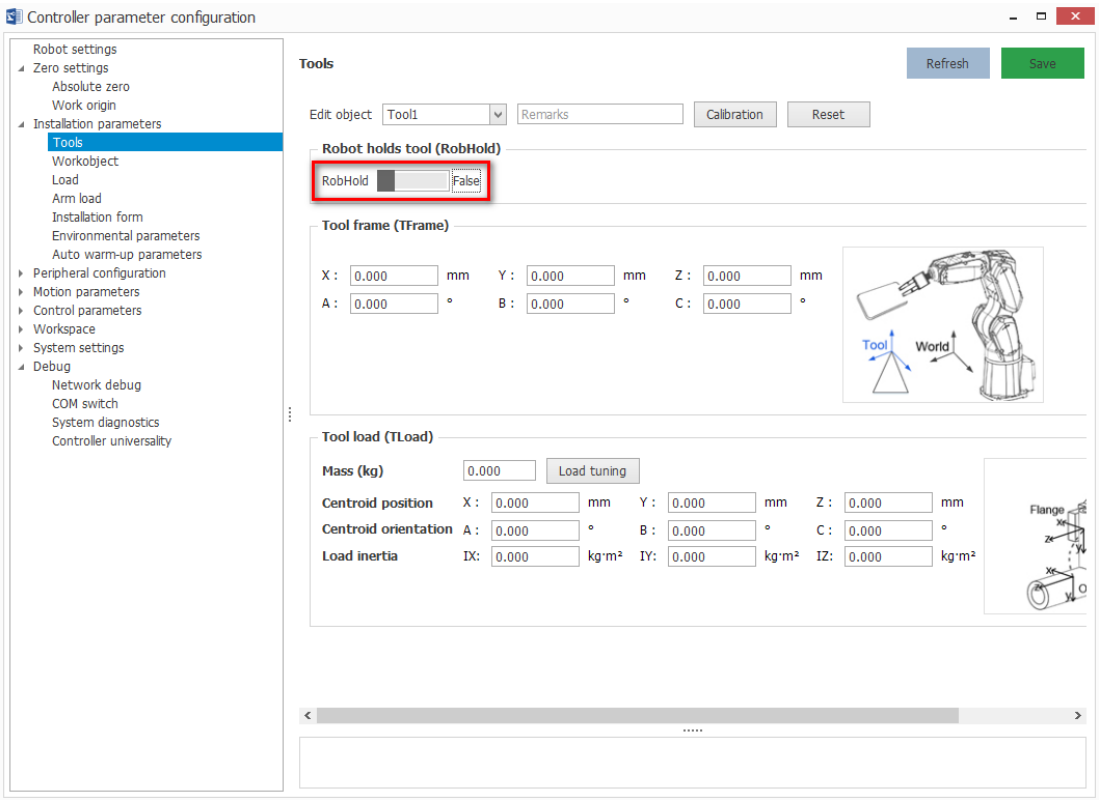
The above figures show the linear motion in TCP mode. P1 and P2 are center points of the workobject at the start and end of the straight line machining. Due to factors such as the size or load of the tool, the machining tool must be fixed relative to the ground and the robot holds the workobject for processing. As shown in the above figures, the solid line between P1 and P2 represents the trajectory of the robot performing linear motion in TCP mode. When the robot is only taught with two points P1 and P2, it is unable to machine a straight line on the surface of the workobject. An additional taught point is required.



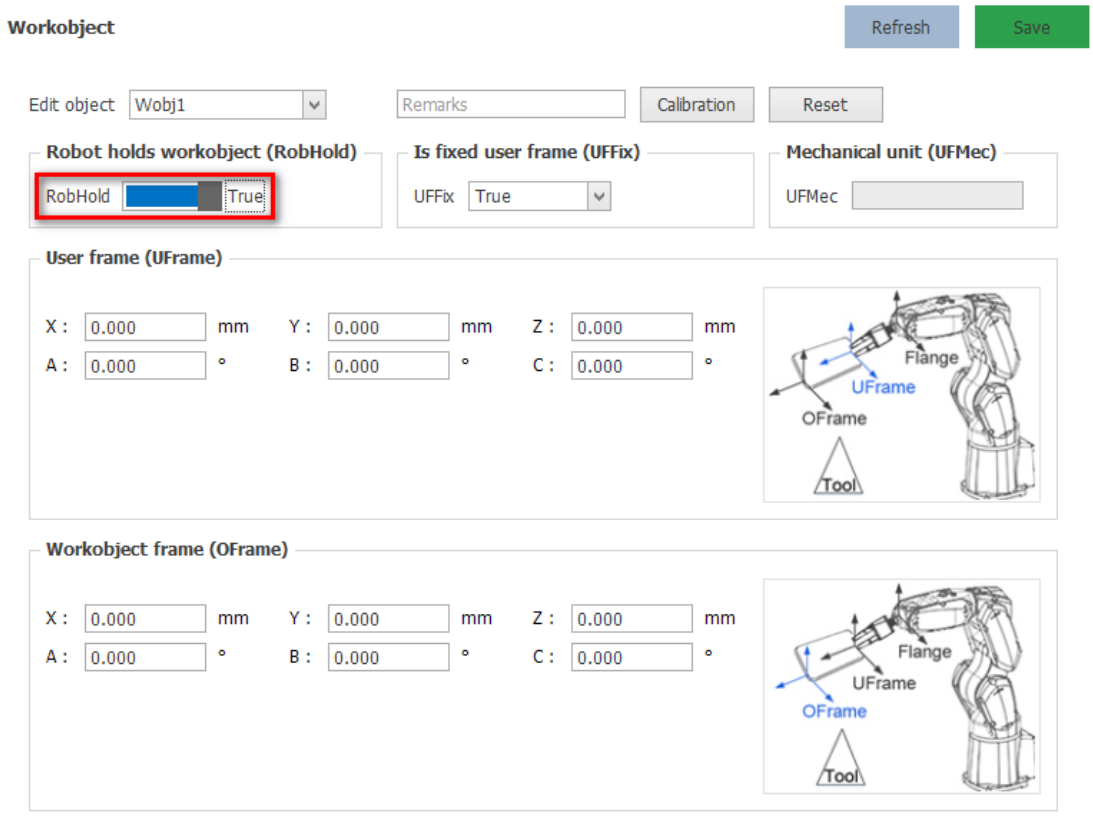
The above figures show the linear motion in ECP mode. P1 and P2 are center points of the workobject at the start and end of the straight line machining. Due to factors such as the size or load of the tool, the machining tool must be fixed relative to the ground and the robot holds the workobject for processing. In ECP mode, the robot can machine a straight line using only two taught points.

Use cases of motion instructions in ECP mode:

External fixed Tool1 is used, as shown below.



Wobj1 is set to be held by the robot, as shown below.



Example of program instructions in ECP frames Tool1 and Wobj1:

MovAbsJ JP[0],V[100],Z[0],Tool[1],Wobj[1]; # MovAbsJ instruction in ECP frames Tool1 and Wobj1

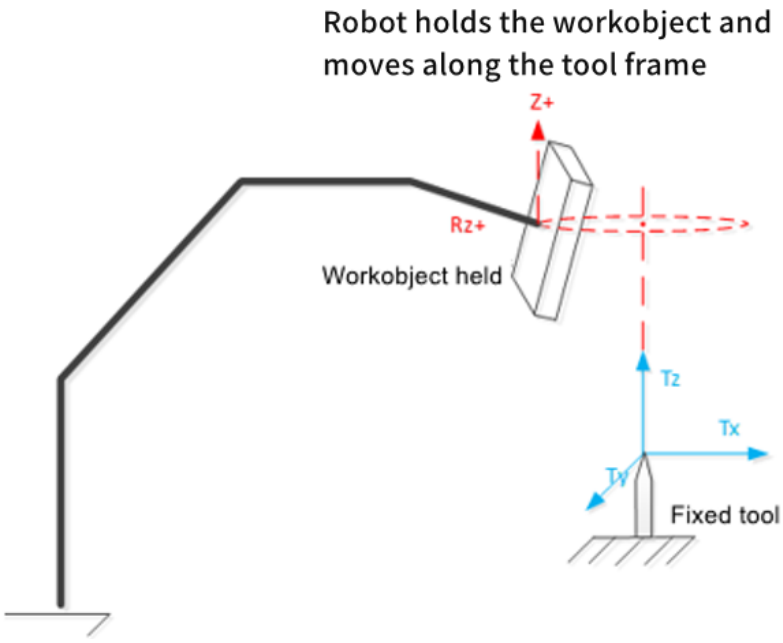
Movj P[0],V[100],Z[0],Tool[1],Wobj[1]; # Movj instruction in ECP frames Tool1 and Wobj1

Movl P[1],V[30],Z[0],Tool[1],Wobj[1]; # Movl instruction in ECP frames Tool1 and Wobj1

12.1.5 Jog Motion in ECP Mode

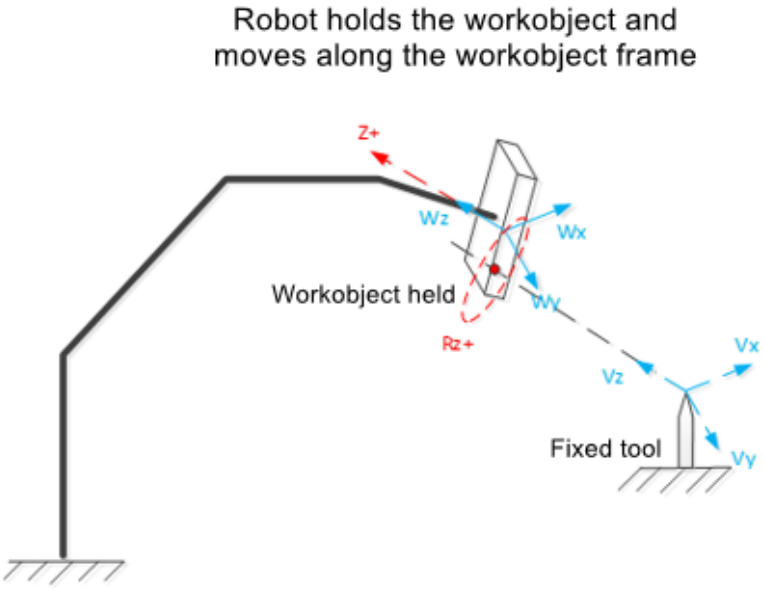
- Motion along tool frame

The robot gripping workobject moves along the frame of the external fixed tool, as shown below.



- Motion along workobject frame

The workobject moves around a virtual frame established on the external fixed tool and parallel to the workobject frame. For the offset movement, it can be considered that the workobject moves in the direction of the workobject frame itself. For the RZ, RY, and RX movement, the position and orientation of the robot change at the same time, as shown below.



12.2 External Axis

12.2.1 External Axis Linked Motion

The external axis is linked to the robot, starting and stopping together with the robot axes simultaneously. However, the external axis does not satisfy the relative spatial positional relationship with the robot axes. It can be understood that the robot and the external axes start and stop simultaneously, but move independently. Currently, the external axis function is only supported by Inovance 680LN, 680N, 660N, and 620N series servos, and the 680LN only supports linear motors with incremental encoders.

12.2.1.1 Prerequisite

- Software requirements: InoRobotLab software and servo configuration tool.
- Hardware requirements: Network cables for communication between the servos and between the servos and the robot; servo motor encoder battery; and 24V power supply and relay if motor brake is used.
- Servo debugging cable and RS232/485-to-USB serial port for 660N and 620N series; Type-C servo debugging cable for 680LN and 680N series.
- Servo debugging knowledge.

12.2.1.2 Wiring

- A 680LN/680N/660N/620N servo and motor is used. Connect EtherCAT port of the robot controller to the IN port of the servo controller using a network cable. If two servos are used, connect the OUT port of the servo controller #1 to the IN port of the servo controller #2 using a network cable. If three servos are used,

connect the OUT port of the servo controller #2 to the IN port of the servo controller #3 using a network cable. The current version supports the connection of up to three servos.

- For example, configure two external axes of 660N. Connect EtherCAT port of the robot controller to the IN port of the servo controller #1 using a network cable, and connect the OUT port of the servo controller #1 to the IN port of the servo controller #2 using a network cable.
- For the wiring of servo motor brake, see the user guide of the corresponding servo.

12.2.1.3 Configuring the Servo

Prerequisite

Connect the CN5 port of the servo drive to the computer via a servo debugging cable and a RS232/485-to-USB converter (only required for 660N and 620N series).

Servo parameter settings

Open the servo configuration tool and set the servo parameters.

Parameter	Description	Setting value
H02-01	Absolute system selection	1-absolute position linear mode, select 0-incremental mode for the incremental encoder linear motor supported by 680LN
H02-09	Delay from brake output ON to command received	Set the brake delay of the motor according to the motor user guide. Fill in 50 for the motor without brake.
H05-07	Numerator of electronic gear ratio	Encoder PPR/1048576
H05-09	Denominator of electronic gear ratio	1
H08-00	Speed loop gain	Debug this parameter using the servo configuration tool
H08-01	Speed loop integral time constant	Debug this parameter using the servo configuration tool
H08-02	Position loop gain	Debug this parameter using the servo configuration tool
H08-15	Load moment of inertia ratio	Debug this parameter using the servo configuration tool

If $\frac{\text{Numerator of electronic gear ratio}}{\text{Denominator of electronic gear ratio}} \neq \frac{\text{Encoder PPR}}{1048576}$, an external axis servo check error will be reported. To clear the alarm, reset the correct numerator and denominator of electronic gear ratio and restart the controller.

Check H00-31 for the encoder PPR.

Other servo parameter settings

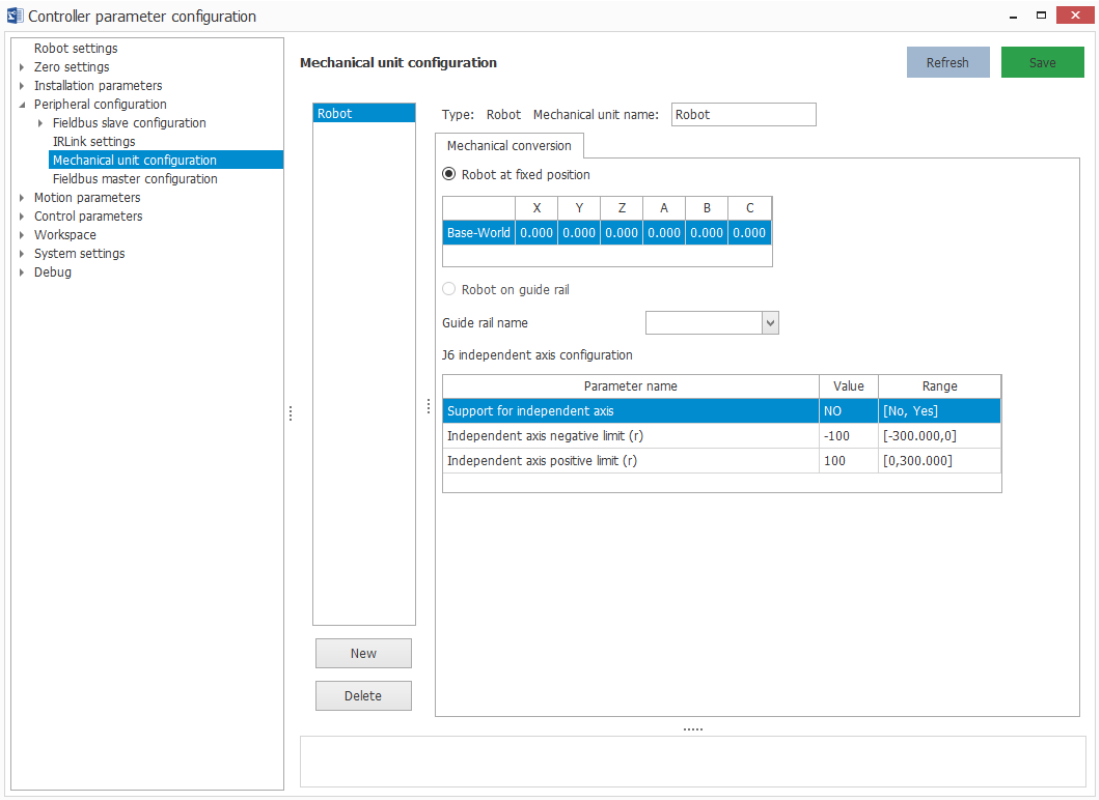
You can set other servo parameters as needed. For example, safety protection parameters, brake-related parameters, notch parameters, feedforward parameters, and so on.

12.2.1.4 Properties

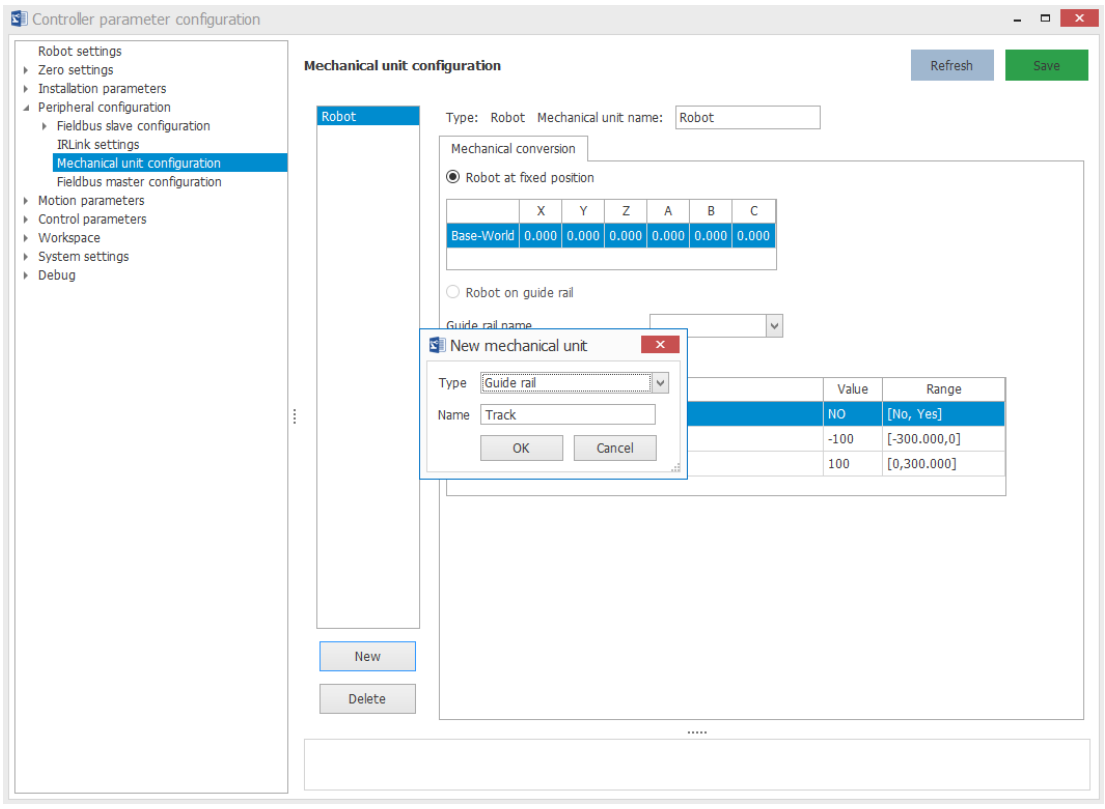
No.	Parameter	Setting value
1	Drive type	620N/660N/680N/680LN
2	Servo control mode	Position
3	Motor shaft property	Rotation axis, linear axis
4	Transmission mode	Motor motion (rotational/linear) - End actuator motion mode (rotational/linear)
5	Physical transmission ratio	Movement distance (°/mm) of the end of the external axis/Movement distance of the motor (°/mm) For example, if the external axis moves by 1° when the motor moves by 100°, then the physical transmission ratio is 100.
6	Zero point	After the external axis is configured, set the zero point of the external axis by clicking "Get current point as zero" in the lower right corner.
7	Positive limit	Set as needed
8	Negative limit	Set as needed
9	Teaching speed	Set as needed
10	Teaching acceleration	Set as needed
11	Max speed	Max speed x Physical transmission ratio ≤ Rated motor speed (the unit must be the same)
12	Max acceleration	Max acceleration ≤ Stop deceleration
13	Stop deceleration	Stop deceleration ≥ Max acceleration
14	Tracking error	Set as needed
15	Arrival error	Set as needed
16	Arrival time	Set as needed
17	Current limit	Set as needed
18	Average load rate limit	Set as needed

12.2.1.5 Setting the Controller Parameters

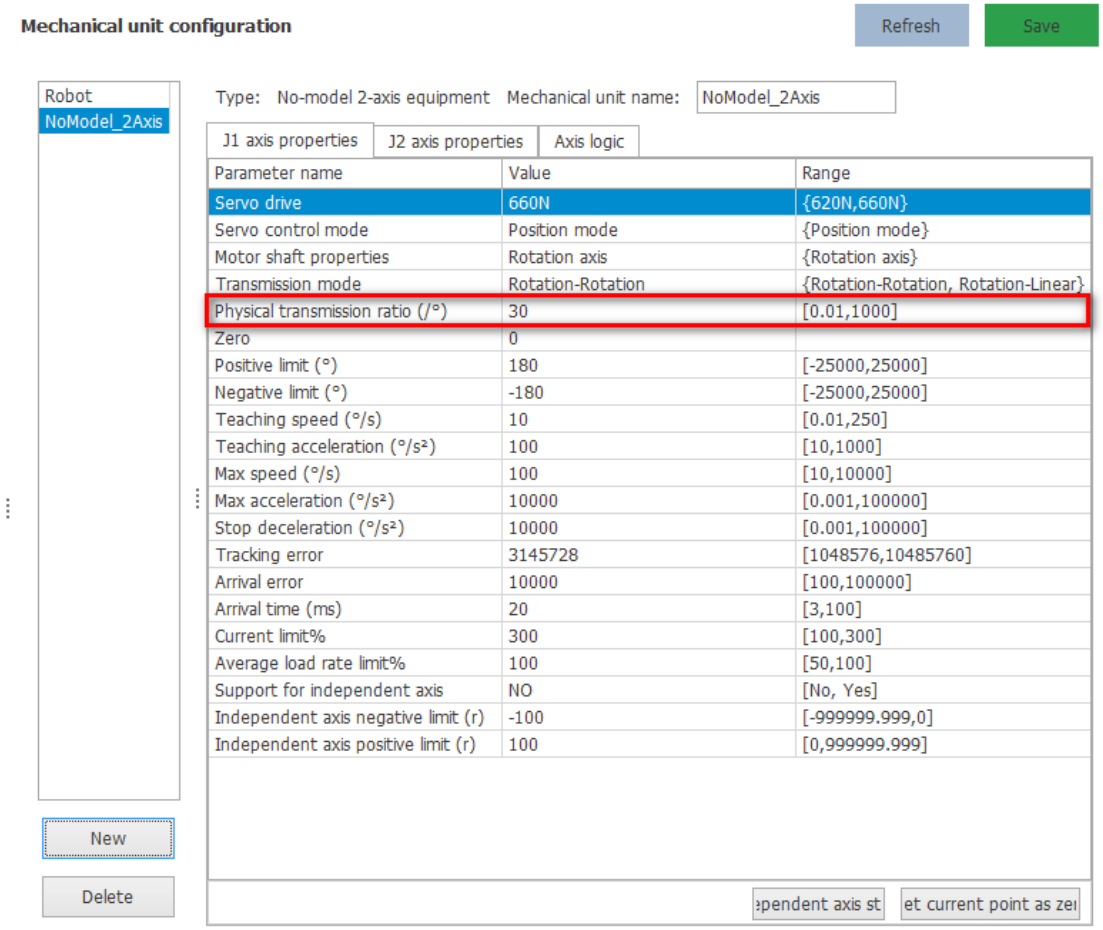
1. Open InoRobotLab software and go to "Controller parameter configuration" > "Peripheral configuration"> "Mechanical unit configuration".
   Currently, at most four mechanical units (including the robot, which is also a mechanical unit) are supported.



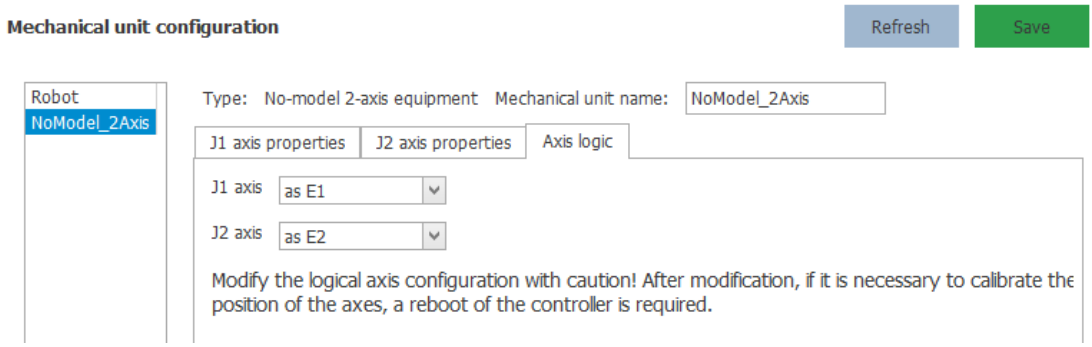
2. Click "New". A dialog pops up, where you can select the type of mechanical unit you want to add.



3. Take the no-model 2-axis equipment as an example. Select the correct servo type (680LN/680N/660N/620N) and the physical transmission ratio (If no other drive mechanism is connected other than the motor, it is recommended to change the physical transmission ratio to 20.



4. When configuring the axis logic, set the motor driven by the servo controller #1 to "E1", the motor driven by the servo controller #2 to "E2", and the motor driven by the servo controller #3 to "E3". Otherwise, mismatch between the servo and input parameters occurs.



5. After configuring the mechanical unit, add the servo type configurations corresponding to ①②③ servo controllers in sequence on the "Fieldbus master configuration" page and download them.
6. After the configuration, click "Save" (This operation must be done by manufacturer in emergency stop mode). When no error occurs, restart the robot controller to complete the configuration of the external axis. If the robot controller is not restarted, it keeps an alarming state. After the robot controller is restarted, the alarm is cleared automatically.
7. After the mechanical unit is configured, the mechanical unit selected in the axis control panel is reset to "Robot".

CAUTION

- The external axis can be used with the collision detection function, but if only the external axis is subjected to force and not the robot itself, the collision detection will not be triggered. Ensure the robot does not interfere with its surroundings. When the J6 axis is configured as an independent axis and activated, collision detection will not take effect even if an external axis is configured.
- The external axis cannot be used with the high trajectory accuracy function. Therefore, the high trajectory accuracy function will be disabled automatically after the external axis is configured. The medium trajectory accuracy function can be used when accuracy is required.
- When a guide rail is configured, the vibration suppression function will be disabled automatically. If the guide rail is configured as a no-model axis equipment and the robot is actually placed on the guide rail, the vibration suppression function may cause false alarms.
- Since the master bus function of the virtual controller is not yet available, configuring external axis for the virtual controller does not require adding the servo configuration of the corresponding servo controller in the "Fieldbus master configuration" interface.
- For external axis configuration of the virtual controller, only mechanical unit configuration is required, and no configuration setup is needed for normal operation.

12.2.1.6 Notes

- The external axis can be used with the collision detection function, but if only the external axis is subjected to force and not the robot itself, the collision detection will not be triggered. Ensure the robot does not interfere with its surroundings. Note that when the J6 axis is configured as an independent axis and activated, collision detection will not take effect even if an external axis is configured.
- The external axis cannot be used with the high trajectory accuracy function. Therefore, the high trajectory accuracy function will be disabled automatically after the external axis is configured. The medium trajectory accuracy function can be used when accuracy is required.
- When a guide rail is configured, the vibration suppression function will be disabled automatically. If the guide rail is configured as a no-model axis equipment and the robot is actually placed on the guide rail, the vibration suppression function may cause false alarms.
- When an external axis is used, teach the external axis within its travel to ensure that the encoder does not overflow.
- For the scenario where the robot moves to a fixed position using a guide rail and then operates independently, you can configure a no-model axis equipment as the guide rail.
- If the joint value of the external axis changes dramatically, check whether the value of the servo parameter H02-01 of the external axis is 1. If it is 1, the encoder overflow occurs.

Cause	Troubleshooting Method	Solution
The number of forward revolutions exceeds 32767 or the number of reverse revolutions exceeds 32768.	Check whether the value of H0B 70 is 32767 or 32768 when the servo drive works in the absolute linear mode (H02 01=1).	Set H0D 20=2 to power on again. Perform zero point calibration if necessary.

- The external axis function can be adapted to different 660N servo motors. The acceleration range allowed for the external axis is large to meet most operating conditions. You can set it according to actual needs.
- If the tracking error of the external axis is reported to be too large, check whether the servo gain is correct. Check whether the maximum speed (actuator) in the properties of the external axis exceeds the maximum speed of the motor.
- During Movc motion, the external axis does not pass through the points specified in the first Movc instruction.
- If the network cable of the external axis is disconnected during operation, a slave offline error occurs. To clear this error, reconnect the network cable.
- If $\frac{\text{Numerator of electronic gear ratio}}{\text{Denominator of electronic gear ratio}} \neq \frac{\text{Encoder PPR}}{1048576}$, an external axis servo check error will be reported. To clear the alarm, reset the correct numerator and denominator of electronic gear ratio and restart the controller.
- After the external axis is configured or modified, a restart of the controller is required to make the configuration or modification take effect.
- The servo gain of the external axis must be set properly. Otherwise, the position feedback fluctuates greatly when the external axis is enabled, causing an error.
- Currently, the stop mode of the external axis can only be a planned stop. If there are special requirements on the stop distance, use the stop function provided by the general-purpose servo.
- For the external axis function of 620N/660N/680N, only motors with multi-turn absolute encoders are supported. For 680LN, only motors with incremental encoders are supported.
- The zero point of incremental encoders may be lost. Note the current configured zero point value and whether the limits are reasonable during use.
- If the axis pulse of the start, middle, and target points of the motion instruction exceeds 2147480000, the motion instruction cannot be executed. Please check with the manufacturer whether the limit of the external axis can be reached under the current transmission ratio. Or adjust the zero point of the external axis so that the pulse of the start, middle, and target points of the motion instruction is within the range of -2147480000 to +2147480000.
- The configuration process is affected by EtherCAT configuration. For detailed notes, see section 9.4.4.3.5 Servo/Controller Compatibility Configuration.

12.2.2 External Axis Coordination

- The external axis coordination function is an advanced function based on the external axis linked motion. The external axis starts and stops together with the robot axes. Also, the trajectory of the external axis is coupled to that of the robot axes. It can be understood that the external axis moves in coordination with the robot axes, while the robot axes keep the original trajectory in the frame of the external axis.

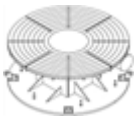
- The external axis coordination function currently supports coordination of the guide rail, coordination of the single-axis rotary positioner, coordination of the dual-axis rotary positioner, and coordination of the guide rail and single/dual-axis rotary positioner.

12.2.2.1 Prerequisite

See the settings in the 12.2.2 External Axis Coordination section. Ensure that the physical transmission ratio is set correctly.

12.2.2.2 Coordinated Mechanical Units

The types of supported mechanical units are as follows.

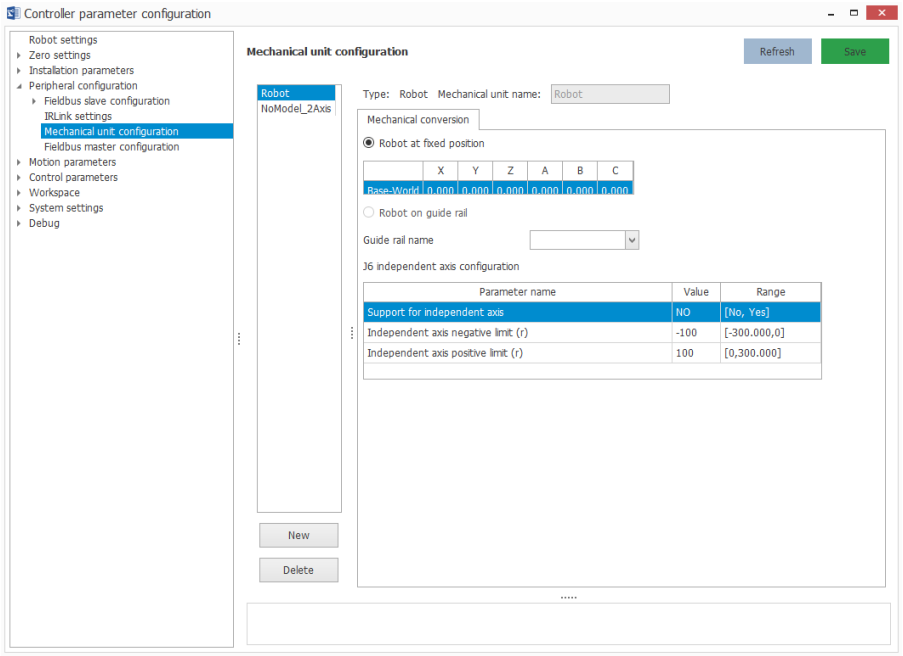
Type	Note
Robot base guide rail	 The guide rail acts as the guide rail only when a robot is located on it. Otherwise, it acts as a no-model axis equipment.
XYZ three-axis base guide rail	 The guide rail acts as the guide rail only when a robot is located on it. Otherwise, it acts as a no-model axis equipment.
Dual-axis base guide rail (XZ/YZ)	 The guide rail acts as the guide rail only when a robot is located on it. Otherwise, it acts as a no-model axis equipment.
Single-axis rotary positioner	
Dual-axis rotary positioner	



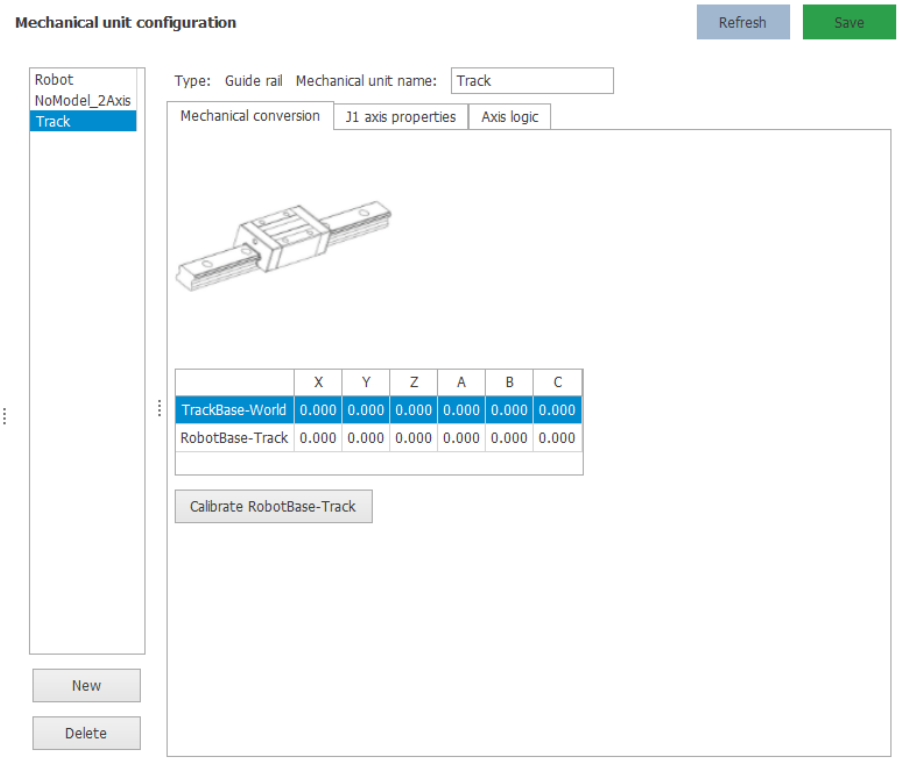
- Up to four mechanical units (including robots) are currently supported.
- Up to six external axes are currently supported.
- Up to one guide rail mechanical unit (single-axis/dual-axis/three-axis guide rail) are currently supported.
- The planes of each axis guide rail should be parallel or perpendicular to each other, and the axial motion direction of each guide rail axis and the frame definition must satisfy the right-hand rule.

Mechanical conversion: Describes the position of the mechanical unit and the position of its internal components.

Robot



The fixed position of the robot cannot be modified temporarily. When the robot is in the fixed position, the base frame of the robot is the world frame.



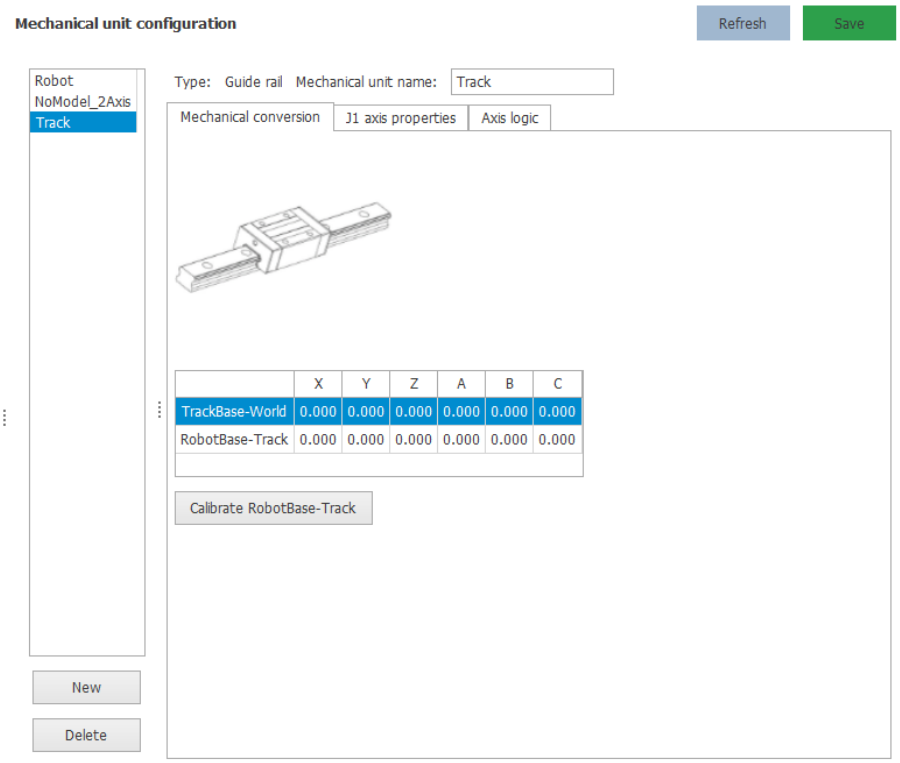
As shown in the figure, "Track" is a configured guide rail. After the guide rail is configured, you can choose to place the robot on the guide rail. You can choose the name of the guide rail from the drop-down list (Track, TrackXZ, TrackYZ and TrackXYZ are currently available). In this case, the base frame of the guide rail is the world frame.

CAUTION

If you choose "Robot on guide rail", after the robot controller is restarted, the coordination is enabled for Movl and Movc instructions by default.

X-axis robot guide rail

I. Introduction to the X-axis robot guide rail



CAUTION

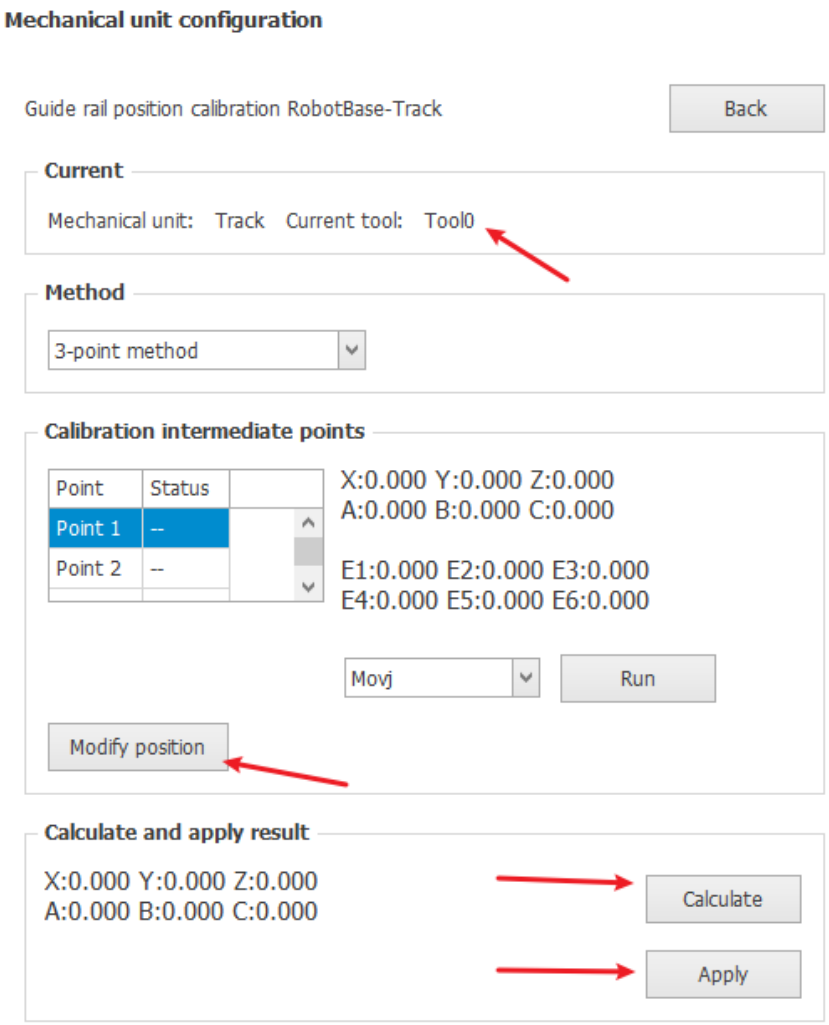
- Before calibrating the external axis, ensure that the axis properties and axis logic are set correctly and restart the robot controller. Otherwise, the calibration result will not match the actual value.
- Before calibrating the guide rail, ensure that the physical transmission ratio is set correctly.

- The forward motion direction of the guide rail is the X-axis direction of the guide rail base frame. The outward direction perpendicular to the guide rail is the Z-axis direction of the guide rail base frame. The Y-axis of the guide rail base frame is determined by the right hand rule. The origin of the guide rail base frame is also the origin of the robot base frame at the origin of the guide rail.
- TrackBase-World is the positional relationship between the guide rail base frame and the world frame. When the robot is placed on the guide rail, the guide rail frame is the world frame. Therefore, the parameters of TrackBase-World are 0 and cannot be modified.
- RobotBase-Track is the positional relationship between the robot base frame and the guide rail base frame. You can perform calibration by manual input or clicking the "Calibrate RobotBase-Track" button.

II. Calibration of the positional relationship between the robot and the guide rail

CAUTION

- Before calibrating the external axis, ensure that the axis properties and axis logic are set correctly and restart the robot controller. Otherwise, the calibration result will not match the actual value.
- Before calibrating the guide rail, ensure that the physical transmission ratio is set correctly.



Calibration method:

1. Set a fixed point as the reference point in the world frame.
2. Install a known tool at the end of the flange and set the tool as the active tool.

CAUTION

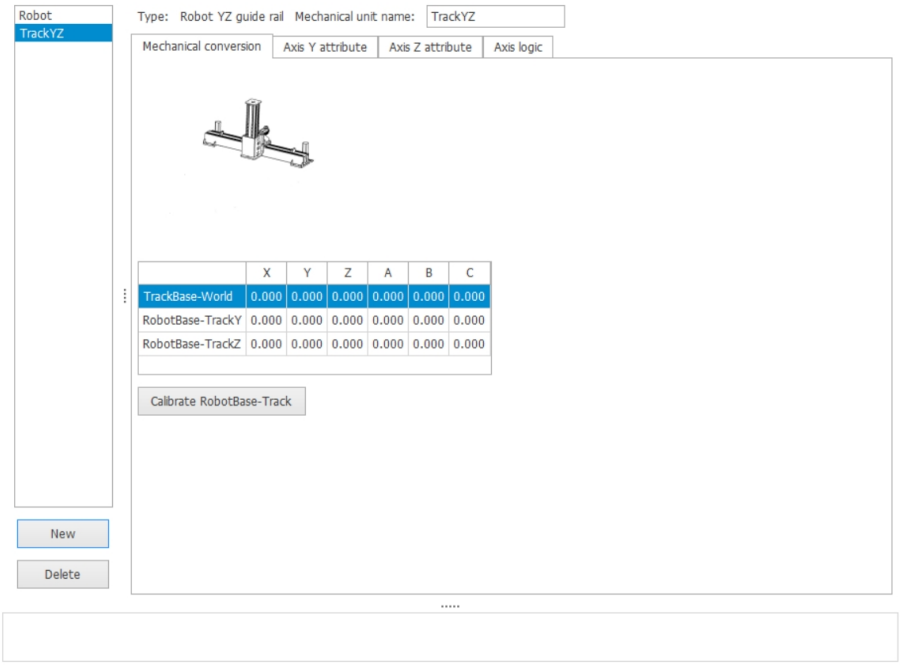
Ensure that the tool is calibrated correctly.

3. Select "Point 1", move the guide rail forward and then the manipulator arm to align the TCP with the reference point, and then click "Modify position".
4. Select "Point 2", move the guide rail forward and then the manipulator arm to align the TCP with the reference point. and click "Modify position".

Object	Content of mechanical conversion
	5. Select "Point 3", move the guide rail forward and then the manipulator arm to align the TCP with the reference point. and click "Modify position". 6. Click "Calculate" to automatically generate the result parameters. The error value is displayed in the pop-up box. 7. Click "Apply". ◦ The calibration result indicates the positional relationship between the robot base frame and the guide rail frame. ◦ The coordinates of the intermediate points indicate the position of the tool in relation to the robot base frame. ◦ Note: When calibrating the guide rail, move the robot as far as possible for higher calibration accuracy. ◦ Note: After the guide rail calibration is complete, select "Robot on guide rail" option in the mechanical unit configuration interface.

YZ base guide rail

I. Introduction



CAUTION

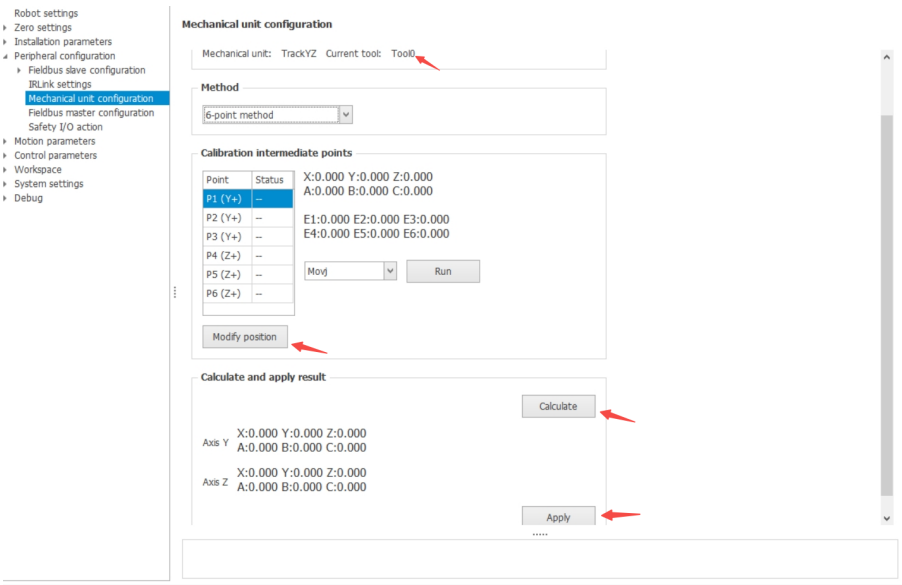
- Before calibrating the external axis, ensure that the axis properties and axis logic are set correctly and restart the robot controller. Otherwise, the calibration result will not match the actual value.
- Before calibrating the guide rail, ensure that the physical transmission ratio is set correctly.
- If the perpendicularity of the dual-axis guide rail is sufficiently high and the guide base frame is perfectly aligned with the robot base frame during installation, calibration can be omitted.

- Definition of the forward motion direction for each axis of the YZ dual-axis guide rail: The forward direction of the Y-axis guide rail is the Y-axis direction of the guide rail base frame, and the forward direction of the Z-axis guide rail is the Z-axis direction of the guide rail base frame. The forward direction of the X-axis is determined by the right-hand rule.
- Definition of the guide rail base frame: The origin of the guide rail base frame is located at the zero position of the guide rail, and the axis directions of the guide rail base frame are consistent with the motion directions of the respective guide rail axes, conforming to the right-hand rule.
- TrackBase-World is the positional relationship between the guide rail base frame and the world frame. When the robot is placed on the guide rail, the guide rail frame is the world frame. Therefore, the parameters of TrackBase-World are 0 and cannot be modified.
- RobotBase-TrackY and RobBase-TrackZ are respectively the positional relationship between the robot base frame and the Y-axis guide rail frame, and that between the robot base frame and the Z-axis guide rail frame. You can perform calibration by manual input or clicking the "Calibrate RobotBase-Track" button.

II. Calibration of the positional relationship between the robot and the guide rail

CAUTION

- Before calibrating the external axis, ensure that the axis properties and axis logic are set correctly and restart the robot controller. Otherwise, the calibration result will not match the actual value.
- Before calibrating the guide rail, ensure that the physical transmission ratio is set correctly.
- When installing the robot on the guide rail, ensure that the guide rails of each axis are perpendicular to each other and that the axis directions of the guide rail align with those of the robot base frame.



Calibration method:

1. Set a fixed point as the reference point in the world frame.
2. Install a known tool at the end of the flange and set the tool as the active tool.

CAUTION

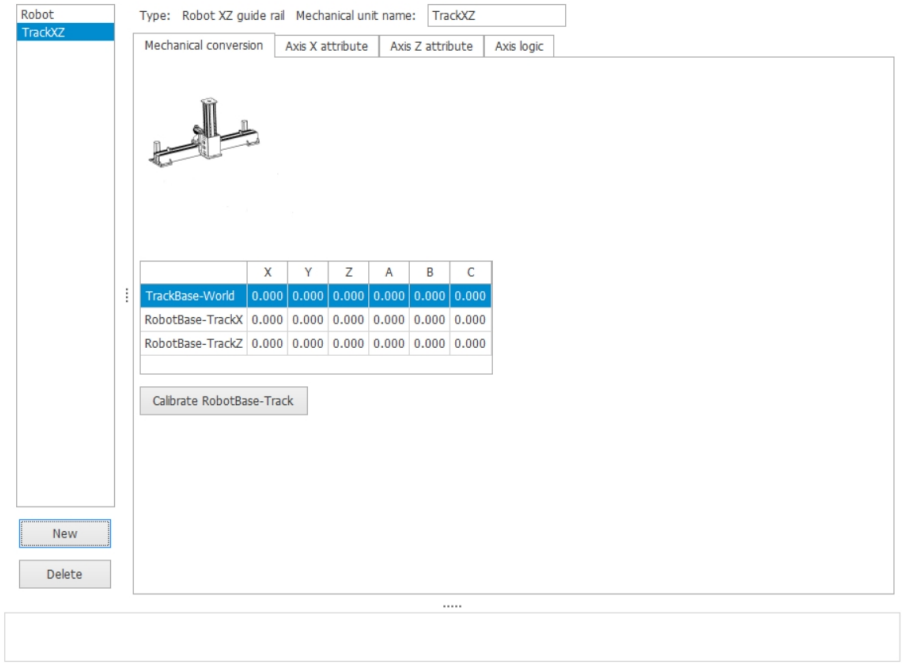
Ensure that the tool is calibrated correctly.

3. Select "Point 1 (Y+)", move the Y-axis guide rail forward and then the manipulator arm to align the TCP with the reference point, and then click "Modify position".
4. Select "Point 2 (Y+)", move the Y-axis guide rail forward and then the manipulator arm to align the TCP with the reference point, and then click "Modify position".
5. Select "Point 3 (Y+)", move the Y-axis guide rail forward and then the manipulator arm to align the TCP with the reference point, and then click "Modify position".
6. Select "Point 4 (Z+)", move the Z-axis guide rail forward and then the manipulator arm to align the TCP with the reference point, and then click "Modify position".

Object	Content of mechanical conversion
	7. Select "Point 5 (Z+)", move the Z-axis guide rail forward and then the manipulator arm to align the TCP with the reference point, and then click "Modify position". 8. Select "Point 6 (Z+)", move the Z-axis guide rail forward and then the manipulator arm to align the TCP with the reference point, and then click "Modify position". 9. Click "Calculate" to automatically generate the result parameters. The error value is displayed in the pop-up box. 10. Click "Apply". ◦ The calibration result indicates the positional relationship between the robot base frame and the guide rail base frame. ◦ The coordinates of the intermediate points indicate the position of the tool in relation to the robot base frame. NOTE • When calibrating the guide rail, move the robot as far as possible for higher calibration accuracy. • After the guide rail calibration is complete, select "Robot on guide rail" option in the mechanical unit configuration interface.

XZ base guide rail

I. Introduction



CAUTION

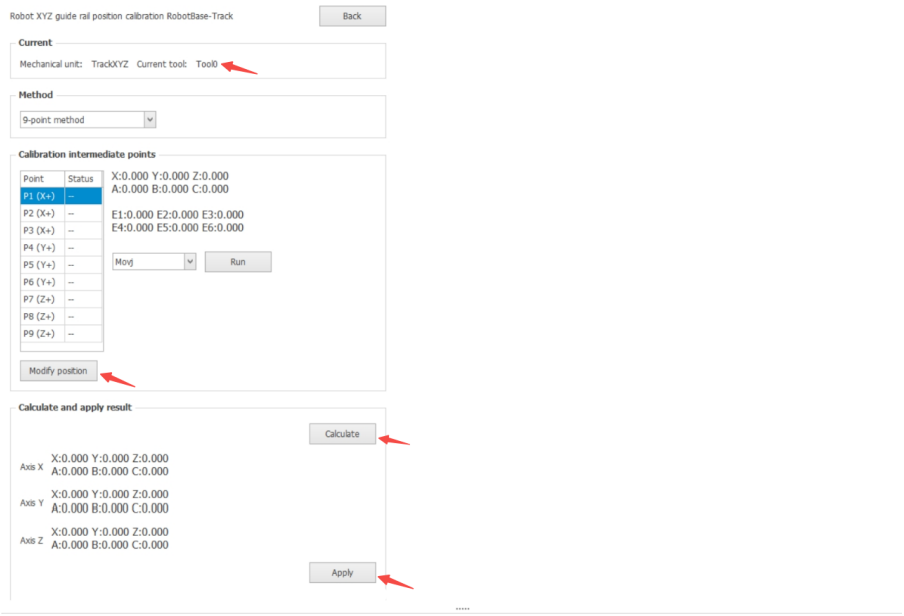
- Before calibrating the external axis, ensure that the axis properties and axis logic are set correctly and restart the robot controller. Otherwise, the calibration result will not match the actual value.
- Before calibrating the guide rail, ensure that the physical transmission ratio is set correctly.
- If the perpendicularity of the dual-axis guide rail is sufficiently high and the guide base frame is perfectly aligned with the robot base frame during installation, calibration can be omitted.

- Definition of the forward motion direction for each axis of the XZ dual-axis guide rail: The forward direction of the X-axis guide rail is the X-axis direction of the guide rail base frame, and the forward direction of the Z-axis guide rail is the Z-axis direction of the guide rail base frame. The forward direction of the Y-axis is determined by the right-hand rule.
- Definition of the guide rail base frame: The origin of the guide rail base frame is located at the zero position of the guide rail, and the axis directions of the guide rail base frame are consistent with the motion directions of the respective guide rail axes, conforming to the right-hand rule.
- TrackBase-World is the positional relationship between the guide rail base frame and the world frame. When the robot is placed on the guide rail, the guide rail frame is the world frame. Therefore, the parameters of TrackBase-World are 0 and cannot be modified.
- RobotBase-TrackX and RobBase-TrackZ are respectively the positional relationship between the robot base frame and the X-axis guide rail frame, and that between the robot base frame and the Z-axis guide rail frame. You can perform calibration by manual input or clicking the "Calibrate RobotBase-Track" button.

II. Calibration of the positional relationship between the robot and the guide rail

CAUTION

- Before calibrating the external axis, ensure that the axis properties and axis logic are set correctly and restart the robot controller. Otherwise, the calibration result will not match the actual value.
- Before calibrating the guide rail, ensure that the physical transmission ratio is set correctly.
- When installing the robot on the guide rail, ensure that the guide rails of each axis are perpendicular to each other and that the axis directions of the guide rail align with those of the robot base frame.



Calibration method:

1. Set a fixed point as the reference point in the world frame.
2. Install a known tool at the end of the flange and set the tool as the active tool.

CAUTION

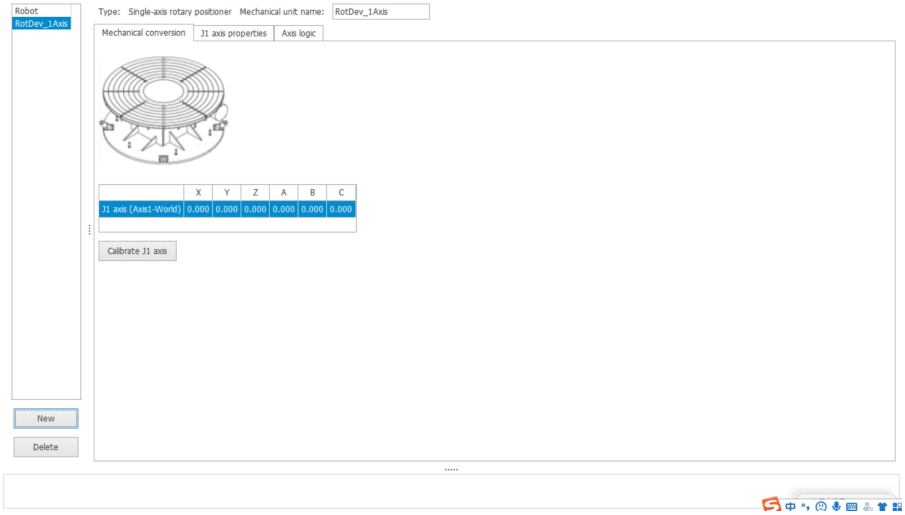
Ensure that the tool is calibrated correctly.

3. Select "Point 1 (X+)", move the X-axis guide rail forward and then the manipulator arm to align the TCP with the reference point, and then click "Modify position".
4. Select "Point 2 (X+)", move the X-axis guide rail forward and then the manipulator arm to align the TCP with the reference point, and then click "Modify position".
5. Select "Point 3 (X+)", move the X-axis guide rail forward and then the manipulator arm to align the TCP with the reference point, and then click "Modify position".
6. Select "Point 4 (Z+)", move the Z-axis guide rail forward and then the manipulator arm to align the TCP with the reference point, and then click "Modify position".

Object	Content of mechanical conversion
	7. Select "Point 5 (Z+)", move the Z-axis guide rail forward and then the manipulator arm to align the TCP with the reference point, and then click "Modify position". 8. Select "Point 6 (Z+)", move the Z-axis guide rail forward and then the manipulator arm to align the TCP with the reference point, and then click "Modify position". 9. Click "Calculate" to automatically generate the result parameters. The error value is displayed in the pop-up box. 10. Click "Apply". ◦ The calibration result indicates the positional relationship between the robot base frame and the guide rail base frame. ◦ The coordinates of the intermediate points indicate the position of the tool in relation to the robot base frame. NOTE • Note: When calibrating the guide rail, move the robot as far as possible for higher calibration accuracy. • Note: After the guide rail calibration is complete, select "Robot on guide rail" option in the mechanical unit configuration interface.

XYZ base guide rail

I. Introduction



CAUTION

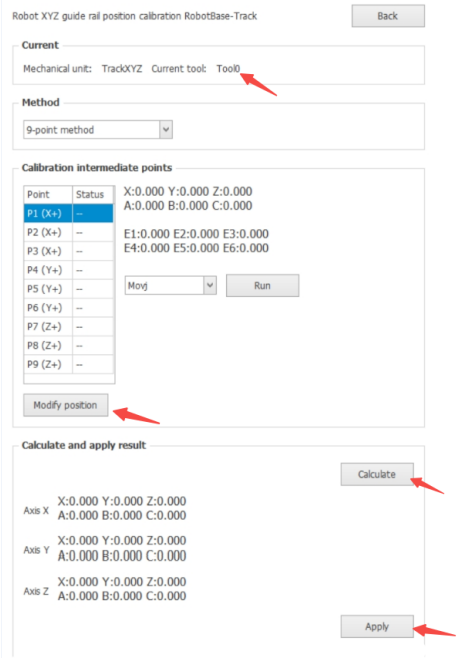
- Before calibrating the external axis, ensure that the axis properties and axis logic are set correctly and restart the robot controller. Otherwise, the calibration result will not match the actual value.
- Before calibrating the guide rail, ensure that the physical transmission ratio is set correctly.
- If the perpendicularity of the three-axis guide rail is sufficiently high and the guide base frame is perfectly aligned with the robot base frame during installation, calibration can be omitted.

- Definition of the forward motion direction for each axis of the three-axis guide rail: The forward directions of the X, Y and Z-axis guide rail are respectively the X, Y and Z-axis directions of the guide rail base frame.
- Definition of the guide rail base frame: The origin of the guide rail base frame is located at the zero position of the guide rail, and the axis directions of the guide rail base frame are consistent with the motion directions of the respective guide rail axes, conforming to the right-hand rule.
- TrackBase-World is the positional relationship between the guide rail base frame and the world frame. When the robot is placed on the guide rail, the guide rail frame is the world frame. Therefore, the parameters of TrackBase-World are 0 and cannot be modified.
- RobotBase-TrackX, RobotBase-TrackY and RobotBase-TrackZ are respectively the positional relationship between the robot base frame and the X-axis guide rail frame, the Y-axis guide rail frame, and the Z-axis guide rail frame. You can perform calibration by manual input or clicking the "Calibrate RobotBase-Track" button.

II. Calibration of the positional relationship between the robot and the guide rail

CAUTION

- Before calibrating the external axis, ensure that the axis properties and axis logic are set correctly and restart the robot controller. Otherwise, the calibration result will not match the actual value.
- Before calibrating the guide rail, ensure that the physical transmission ratio is set correctly.
- When installing the robot on the guide rail, ensure that the guide rails of each axis are perpendicular to each other and that the axis directions of the guide rail align with those of the robot base frame.



Calibration method:

1. Set a fixed point as the reference point in the world frame.
2. Install a known tool at the end of the flange and set the tool as the active tool.

CAUTION

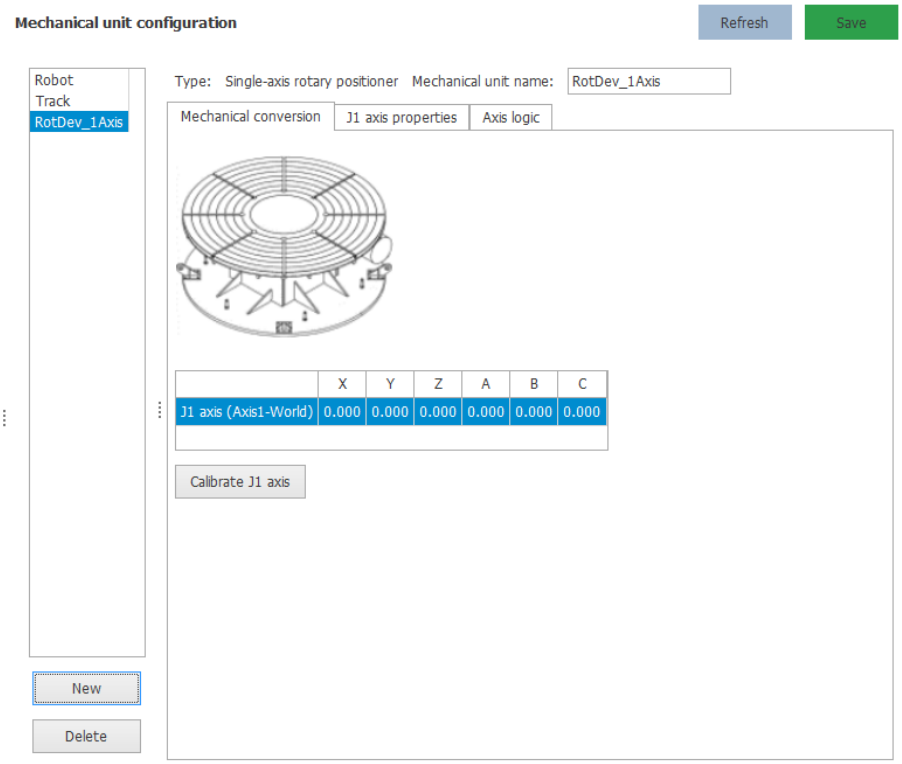
Ensure that the tool is calibrated correctly.

3. Select "Point 1 (X+)", move the X-axis guide rail forward and then the manipulator arm to align the TCP with the reference point, and then click "Modify position".
4. Select "Point 2 (X+)", move the X-axis guide rail forward and then the manipulator arm to align the TCP with the reference point, and then click "Modify position".
5. Select "Point 3 (X+)", move the X-axis guide rail forward and then the manipulator arm to align the TCP with the reference point, and then click "Modify position".
6. Select "Point 4 (Y+)", move the Y-axis guide rail forward and then the manipulator arm to align the TCP with the reference point, and then click "Modify position".

Object	Content of mechanical conversion
	7. Select "Point 5 (Y+)", move the Y-axis guide rail forward and then the manipulator arm to align the TCP with the reference point, and then click "Modify position". 8. Select "Point 6 (Y+)", move the Y-axis guide rail forward and then the manipulator arm to align the TCP with the reference point, and then click "Modify position". 9. Select "Point 7 (Z+)", move the Z-axis guide rail forward and then the manipulator arm to align the TCP with the reference point, and then click "Modify position". 10. Select "Point 8 (Z+)", move the Z-axis guide rail forward and then the manipulator arm to align the TCP with the reference point, and then click "Modify position". 11. Select "Point 9 (Z+)", move the Z-axis guide rail forward and then the manipulator arm to align the TCP with the reference point, and then click "Modify position". 12. Click "Calculate" to automatically generate the result parameters. The error value is displayed in the pop-up box. 13. Click "Apply". ◦ The calibration result indicates the positional relationship between the robot base frame and the guide rail base frame. ◦ The coordinates of the intermediate points indicate the position of the tool in relation to the robot base frame. **NOTE** • When calibrating the guide rail, move the robot as far as possible for higher calibration accuracy. • After the guide rail calibration is complete, select "Robot on guide rail" option in the mechanical unit configuration interface.

Single-axis rotary positioner

I. Introduction to the single-axis rotary positioner



CAUTION

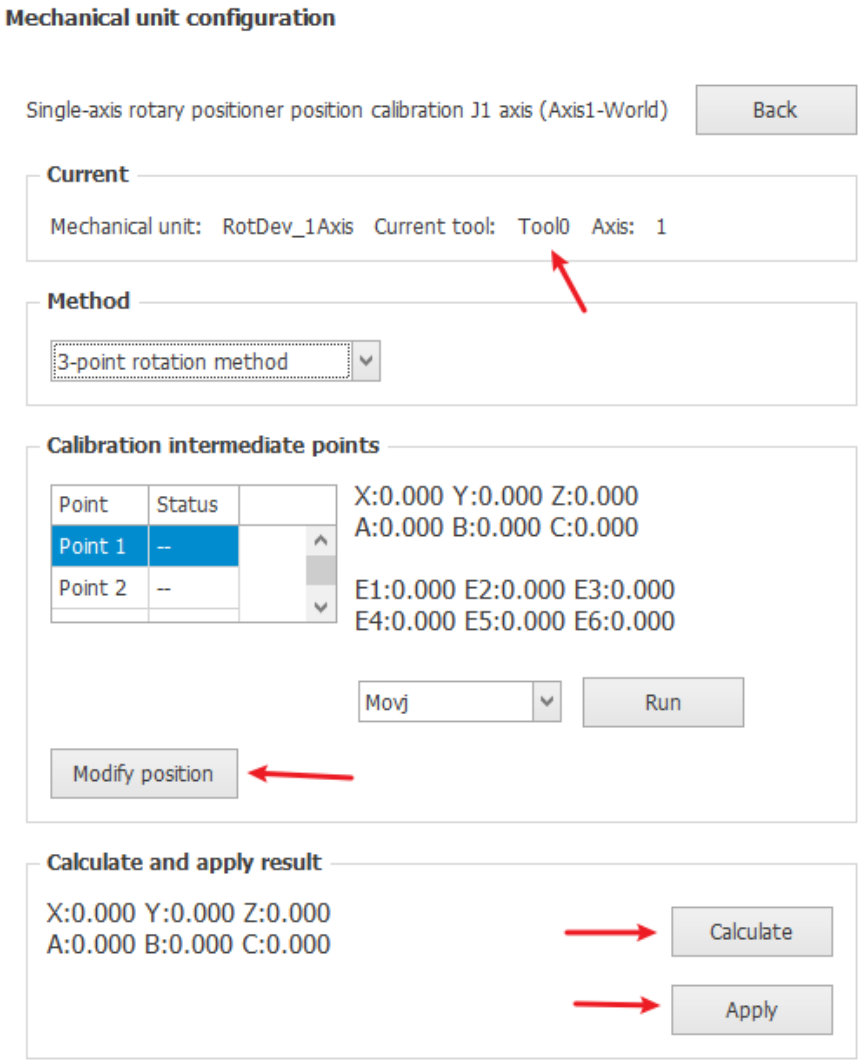
- Before calibrating the external axis, ensure that the axis properties and axis logic are set correctly and restart the robot controller. Otherwise, the calibration result will not match the actual value.
- Before calibrating the single-axis rotary positioner, ensure that the physical transmission ratio is set correctly.
- For applications requiring coordination of the robot, guide rail, and rotation positioner simultaneously, calibrate the guide rail first and then the rotation positioner.

- J1 axis (Axis1-World): The positional relationship of the frame of the single-axis rotary positioner relative to the world frame when the positioner is at the origin. Calibration can be performed by manual input or by clicking the "Calibrate J1 axis" button.
- The origin of the frame used by the single-axis rotary positioner is located in the center of the rotary table. The Z direction coincides with the rotation axis. The X axis points to the zero-degree direction of the positioner when the first point is calibrated.

II. Calibration of the position relation of the frame of the single-axis rotary positioner relative to the world frame

CAUTION

- Before calibrating the external axis, ensure that the axis properties and axis logic are set correctly and restart the robot controller. Otherwise, the calibration result will not match the actual value.
- Before calibrating the single-axis rotary positioner, ensure that the physical transmission ratio is set correctly.
- For applications requiring coordination of the robot, guide rail, and rotation positioner simultaneously, calibrate the guide rail first and then the rotation positioner.



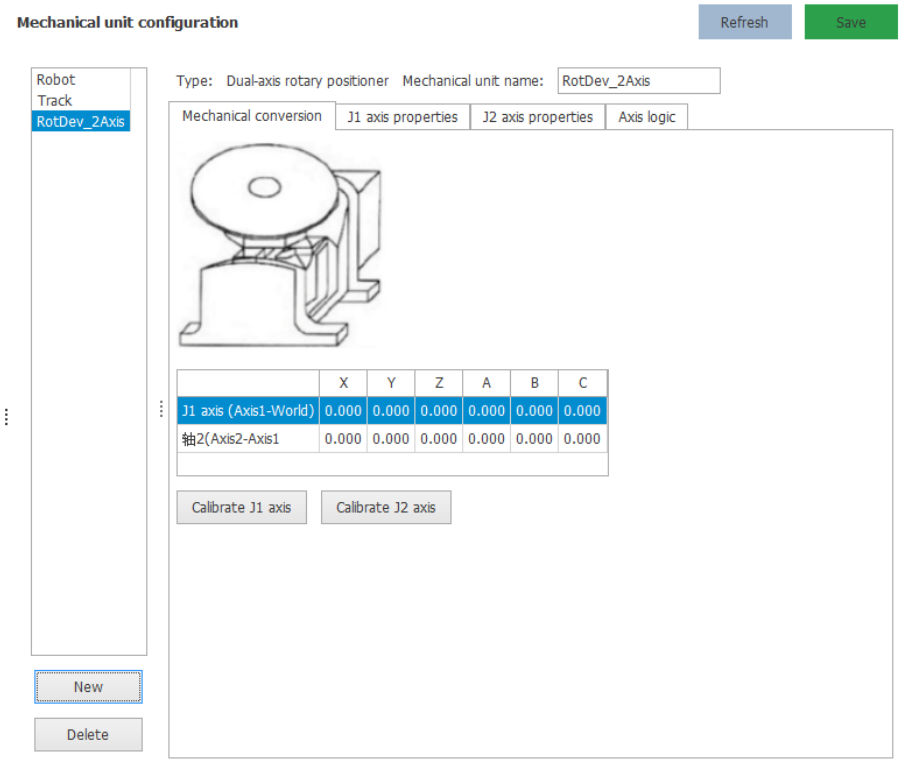
Calibration method:

1. Install a known tool at the end of the flange and set the tool as the active tool.
2. Mark a reference point on the turn table of the rotary positioner.
3. Select "Point 1", move the manipulator to align the TCP with the reference point, and click "Modify position".
4. Select "Point 2", rotate the J1 axis of the rotary positioner by a certain angle in the positive direction, then move the manipulator arm to align the TCP with the reference point, and click "Modify position".

Object	Content of mechanical conversion
	5. Select "Point 3", rotate the J1 axis of the rotary positioner by a certain angle in the positive direction, then move the manipulator arm to align the TCP with the reference point, and click "Modify position". 6. Click "Calculate" to automatically generate the result parameters. 7. Click "Apply". NOTE When calibrating the rotary positioner, the rotation angle is preferably larger than 30° for higher calibration accuracy.

Dual-axis rotary positioner

I. Introduction to the dual-axis rotary positioner



CAUTION

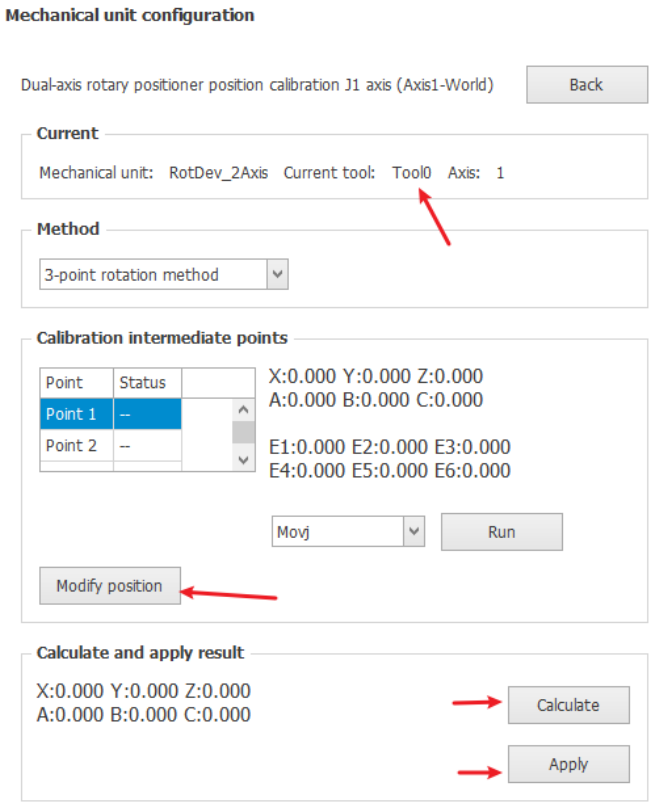
- Before calibrating the external axis, ensure that the axis properties and axis logic are set correctly and restart the robot controller. Otherwise, the calibration result will not match the actual value.
- Before calibrating the dual-axis rotary positioner, ensure that the physical transmission ratio is set correctly.
- For applications requiring coordination of the robot, guide rail, and rotary positioner simultaneously, calibrate the guide rail first, then restart and calibrate the rotary positioner.

1. J1 axis (Axis1-World): The positional relationship of the J1 axis frame of the dual-axis rotary positioner and the world frame when the J1 axis is at the origin. Calibration can be performed by manual input or by clicking the "Calibrate J1 axis" button.
2. J2 axis (Axis2-Axis1): The positional relationship of the J2 axis frame of the dual-axis rotary positioner relative to the J1 axis frame when the J2 axis is at the origin. Calibration can be performed by manual input or by clicking the "Calibrate J2 axis" button.
3. The origin of the frame used by the dual-axis rotary positioner is located in the center of the rotary table of the J2 axis. The Z direction coincides with the rotation axis. The X axis points to the zero-degree direction of the positioner when the first point is calibrated.

II. Calibration of the positional relationship of the frame of the dual-axis rotary positioner relative to the world frame

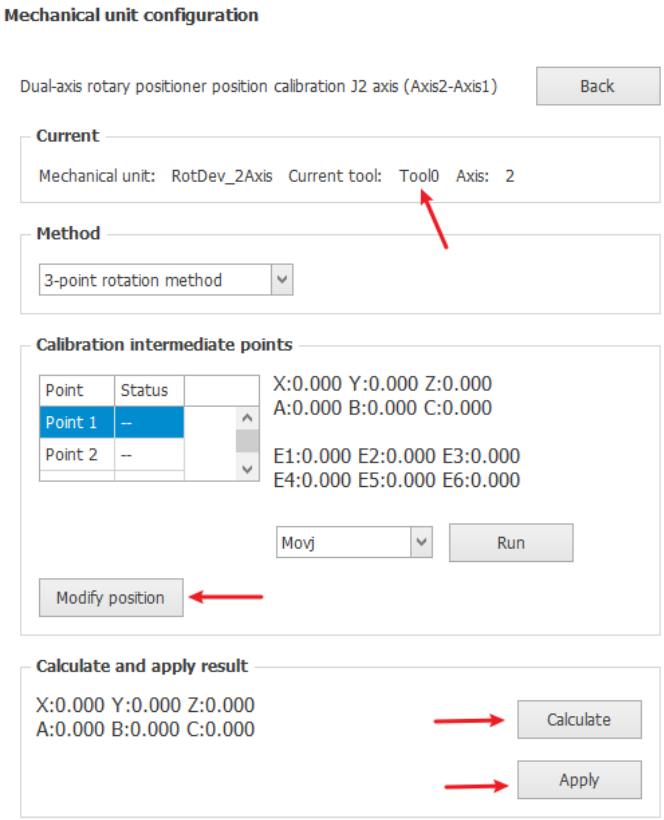
CAUTION

- Before calibrating the external axis, ensure that the axis properties and axis logic are set correctly and restart the robot controller. Otherwise, the calibration result will not match the actual value.
- Before calibrating the dual-axis rotary positioner, ensure that the physical transmission ratio is set correctly.
- For applications requiring coordination of the robot, guide rail, and rotation positioner simultaneously, calibrate the guide rail first and then the rotation positioner.



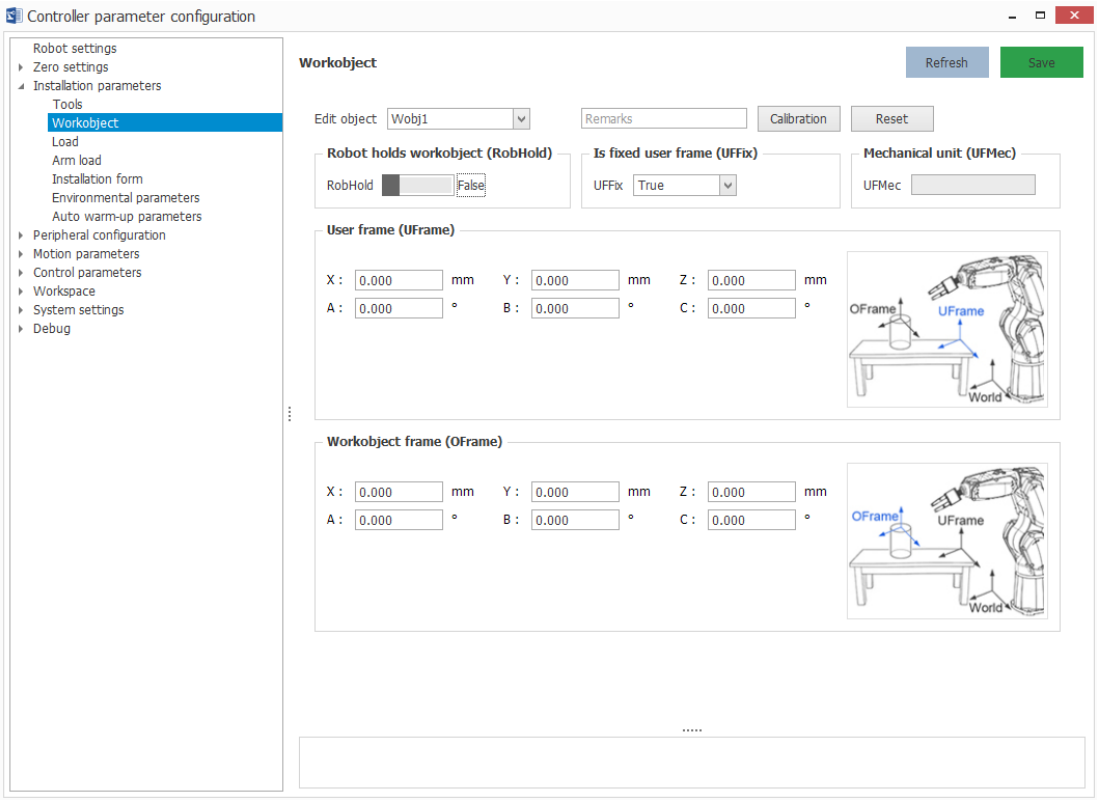
As an example, the following describes the calibration method of the J1 axis.

1. Install a known tool at the end of the flange and set the tool as the active tool.
2. Mark a reference point on the turn table of the rotary positioner.
3. Select "Point 1", move the manipulator to align the TCP with the reference point, and click "Modify position".
4. Select "Point 2", rotate the J1 axis of the rotary positioner by a certain angle in the positive direction, then move the manipulator arm to align the TCP with the reference point, and click "Modify position".
5. Select "Point 3", rotate the J1 axis of the rotary positioner by a certain angle in the positive direction, then move the manipulator arm to align the TCP with the reference point, and click "Modify position".
6. Click "Calculate" to automatically generate the result parameters.
7. Click "Apply".

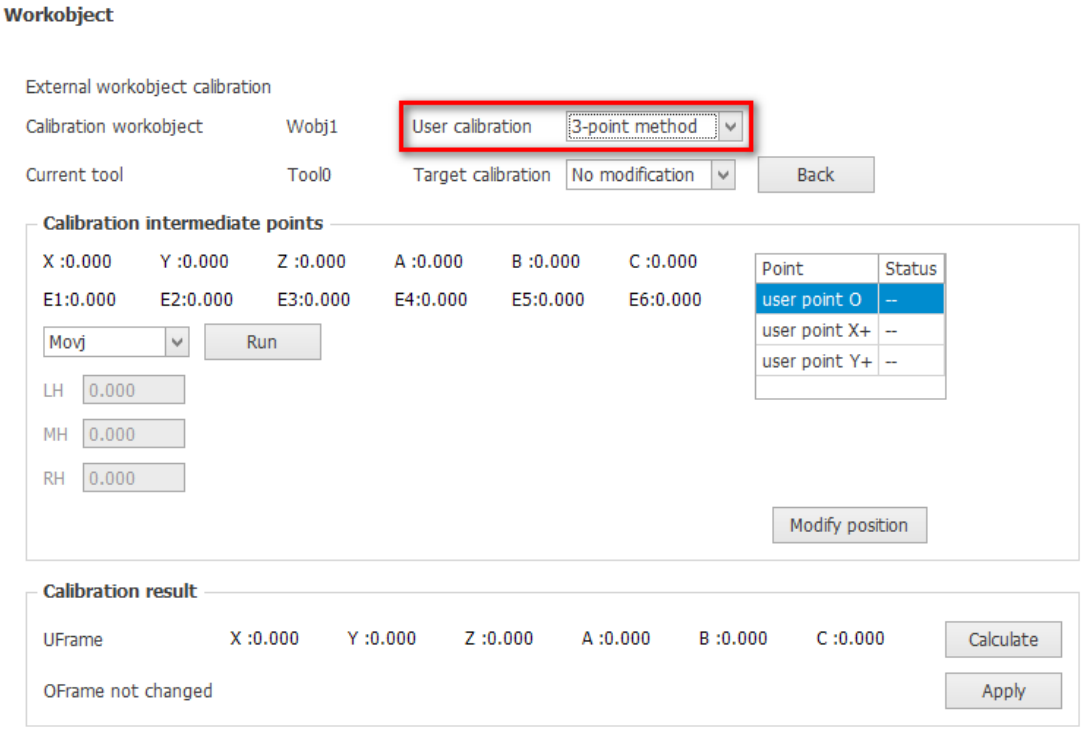
Object	Content of mechanical conversion
	NOTE When calibrating the rotary positioner, the rotation angle is preferably larger than 30° for higher calibration accuracy.  The calibration of J2 axis is the same with that of J1 axis. NOTE When calibrating the rotary positioner, the rotation angle is preferably larger than 30° for higher calibration accuracy.

12.2.2.3 Setting Workobject for the Coordinated Mechanical Unit

1. Select the workobject to be used, set "RobHold" to "False", "UFFix" to "False", and "UFMec" to the name of the single/dual-axis rotary positioner.



2. In coordination, the user frame changes with the rotation of the single/dual-axis rotary positioner.
3. You can set the workobject frame by clicking the "Calibration" button or directly entering the coordinate values of the workobject frame.



4. When calibrating a workobject frame for which the parameter "UFFix" is set to "False", you can use the 3-point calibration method. The calibration method is the same as the traditional workobject calibration method.

12.2.2.4 Coordinated Motion Instructions

- When a guide rail is configured and the robot is placed on the guide rail, the Movl and Movc instructions perform coordinated motion.

- When no guide rail is configured or a guide rail is configured but the robot is placed at a fixed position, if the workobject frame in the Movl and Movc instructions are associated with the name of single/dual-axis rotary positioner, then the Movl and Movc instructions are coordinated.
- When a guide rail is configured and the robot is placed on the guide rail, and the workobject frame in the Movl and Movc instructions is associated with the name of a single/dual-axis rotary positioner, then the Movl and Movc instructions enable the robot, the guide rail and the single/dual-axis rotary positioner to move in coordination.

Movl linear motion of robot and external axis

Movl P, V, Z, Tool, [Wobj], [Acc],[Dec], [NWait], [Until In == value], [Out(No,value,Type)…]



If the workobject is movable and associated with a mechanical unit, then the motion between the robot and the mechanical unit is coordinated.

Movc circular motion of robot and external axis

Movc P, V, Z, Tool, [Wobj], [Acc],[Dec], [NWait], [Until In == value], [Out(No,value,Type)…]



For the two Movc instructions, if they have the same robot axis points, they must also have the same external axis points.

12.2.2.5 Precision of Coordinated Motion

For applications requiring high precision of the external axis coordination, pay attention to the following.

- Check that the transmission ratio of the external axis is set correctly. Make sure the rotation angle (or distance) of the external axis set on the teach pendant is equal to the actual rotation angle (or distance) of the external axis.
- The orientation of the tool needs to be calibrated to such a precision that the end position of the tool is not changed upon rotation around any axis of the tool. After completing calibration of the tool, calibrate the guide rail and the single/dual-axis rotary positioner.
- When the workobject frame is associated with the positioner and not calibrated, check whether the origin of the workobject frame coincides with the rotation center of the positioner after calibrating the positioner.
- If the robot is found to have poor coordination performance during automatic running, you can use the instruction to enable medium trajectory accuracy to improve the coordination performance.
- If you choose the "Robot on guide rail" option, all Movl and Movc motions will be coordinated motions after you save the settings and restart the robot controller. If you just need to move the robot to a certain

position and then move the robot independently, you can choose not to place the robot on the guide rail. This will reduce the error caused by the guide rail calibration.

- For applications requiring simultaneous coordination of the guide rail and the positioner, configure the guide rail and the positioner first. After calibrating the guide rail, place the robot on the calibrated guide rail in the external axis configuration interface. Then restart the robot controller and calibrate the displacement positioner.

12.2.3 External Axis Supported Servo Gantry Synchronization Function

The external axis function can support the gantry synchronization function of Inovance 680N/680LN series servos. All 680 series servo types supported by the external axis can be configured for gantry synchronization. The slave axis not configured as an external axis can follow the master axis configured as the external axis for synchronized motion.

12.2.3.1 Prerequisite

- Encoder cables: Connect the encoder cables between two 680N/680LN servo drives.
- Software requirement: InoDriverShop (can be downloaded from the Inovance official website).
- Hardware requirement: Before using the gantry synchronization function, at least one external axis must be successfully configured (for master axis configuration).

12.2.3.2 Wiring

For example, configure two external axes. Configure external axis #1 and external axis #2, with external axis #1 configured as the master axis and connected to the slave axis externally.

- Wiring for external axis: Connect the IN port of external axis #1 to the controller, and the OUT port to the IN port of external axis #2.
- Wiring for gantry synchronization: Use encoder cables to connect the second encoder port of the servo drive of external axis #1 with the second encoder port of the servo drive of the slave axis.



The slave axis cannot be configured as an external axis, meaning there is no need to connect the servo drive of the slave axis to the controller/master axis servo drive through IN/OUT ports.

12.2.3.3 Configuring the Servo

- After completing the external axis configuration and the wiring between the external axis and the slave axis, use InoDriverShop to access to the servo background.
- Check if the gantry synchronization function is available in the servo background parameters. If not, contact Inovance technical support.
- Check the communication status of the two axes. If H14-02=1 for the two axes, they are in gantry synchronization status.

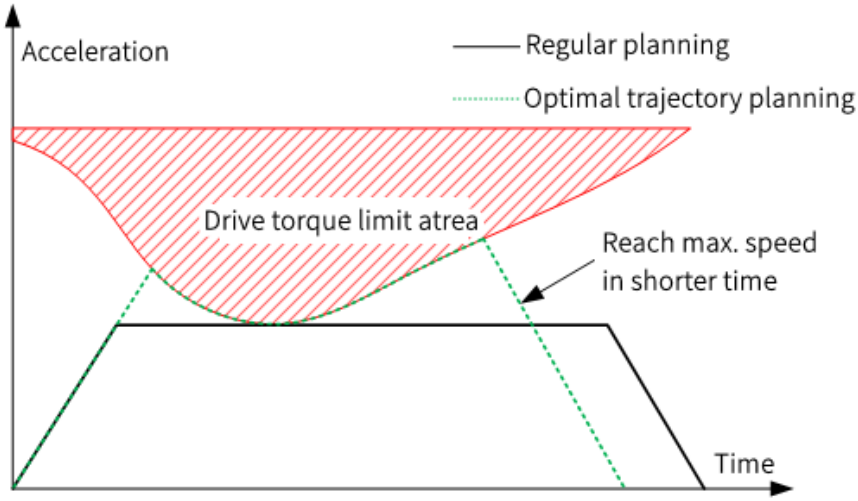
- Set the master and slave axes. Set the external axis as the master axis by configuring H14-03, and check if H14-00 of the slave axis is 1. If yes, proceed to the next step.
- Confirm the forward direction of both axes. Push both axes simultaneously (the axes are connected through the gantry, pushing the gantry to rotate the axes in the same physical direction), and observe whether the signs of H14-40 and H14-41 of the master axis are consistent.
- The basic servo configuration is now complete. For detailed usage and configuration methods of the gantry synchronization function, see the "SV680 Series Servo Guide".

> CAUTION
>
> If an axis drive alarm occurs during use, check whether the configuration of master and slave axes is correct, and connect to the servo background to view the alarm reason.

12.3 Optimal Trajectory

The optimal trajectory function automatically adjusts motion parameters based on the load, requiring no user intervention. It brings about two improvements.

- Improved ease of use. The robot automatically uses the suitable acceleration to realize the best cycle time, without exceeding the electrical current specifications.
- Improved efficiency. The acceleration allowed varies from position to position. The optimal trajectory can continuously change with the allowable value, keeping the joint output at maximum and achieving optimal acceleration and deceleration efficiency.



12.3.1 Debugging Steps

After the optimal trajectory function is enabled, the motion is not affected by the "Acceleration" parameter configured on the software interface. Instead, the acceleration parameter is determined automatically based on the model. Follow the steps below.

Step 1: Set the load parameters correctly, mainly including the tool load and workobject load.

CAUTION

The load must be set even if it is very small. If the load mass is zero, the system assumes that the user forgets to set the load and moves according to the maximum load for safety reasons, resulting in very slow motion.

Step 2: Use the instruction "RapidMove (PTP, ON)" to enable the RapidMove mode. You can enable the RapidMove mode at the beginning of the program, so that the RapidMove mode takes effect for the entire project. You also can enable or disable the RapidMove mode as needed during execution of the program. For details, see the "Industrial Robot Instruction Guide".

Step 3: Debug the electrical current and cycle time using instructions. The magnitude of the current is positively related to the acceleration coefficient in the instruction. You can modify the acceleration coefficient of a single instruction, or modify the acceleration coefficient of multiple instructions in batch using the SetAcc instruction. For details of the SetAcc instruction, see the "Industrial Robot Instruction Guide".

- If it is found during execution of the program that the maximum current is less than 100%, it indicates that the cycle time can be further improved. In this case, you can gradually increase the acceleration to 120%.
- If no vibration or current alarm occurs, you can increase the value of SetAccRamp parameter gradually from 50.
- Increasing Acc or AccRamp may cause current overlimit alarm or drive alarm. If the current overlimit alarm or drive alarm occurs, first check whether the load is set correctly. If the vibration is obvious, reduce the value of SetAccRamp parameter first, and then reduce the value of SetAcc parameter.

Step 4: If the requirement on the cycle time is high and cannot be met by the preceding steps, you can reduce the acceleration coefficient at the place where the current overlimit alarm occurs. The acceleration coefficient at other places can still use 100% or 120%.

CAUTION

This function is based on the dynamics model. If the model is inaccurate, it may cause current alarms or low efficiency. If the over-current alarm occurs only on certain axes, the model may be inaccurate. In this case, you can correct the model by using the model correction function. For details, see the 9.5.6.3 Torque Model Correction section.

12.3.2 Application Example

```
Start;
  GripLoad 1;       //Activates the load and sets the corresponding load parameters
  RapidMove( ALL,ON);  //Enables the optimal trajectory function
  SetAcc(100,100);    //Adjusts the magnitude of the current through the SetAcc instruction
  L[0]:
```

```
    Movj P[0],V[100],fine,Tool[0];
    Movj P[1],V[100],fine,Tool[0];
Goto L[0];
End;
```

12.4 Independent Axis

12.4.1 Overview

- The independent axis function allows an axis in the robot system to perform independent motion without being affected by other axes. When the independent axis function is enabled for a certain axis, the motion of this axis is affected only by the series of independent axis instructions (IndXXX).
- After the independent axis rotates for multiple turns, you can reset it to the normal axis by instruction IndReset. Also, you can control the multi-turn value of this independent axis.
- Target axis that can be set as an independent axis has no hard limit. For the 6-axis robots, only the J6 axis can be set as an independent axis. Only the external axis whose transmission mode is set to "rotation-rotation" can be set as an independent axis (For example, the conveyor cannot be set as an independent axis.)

12.4.2 Description of Independent Axis Function

For details of the independent axis instructions, see the "Industrial Robot Instruction Guide".

- Function 1: Switch the independent axis to speed mode and move it

IndCMove MechUnit, Axis, Speed, [Acc] [Dec]; This instruction commands an axis in the mechanical unit to move at constant speed.

- Function 2: Detect whether the independent axis reaches specified speed

Wait IndSpeed MecUnit Axis; This instruction checks whether an axis in the mechanical unit reaches the planned speed.

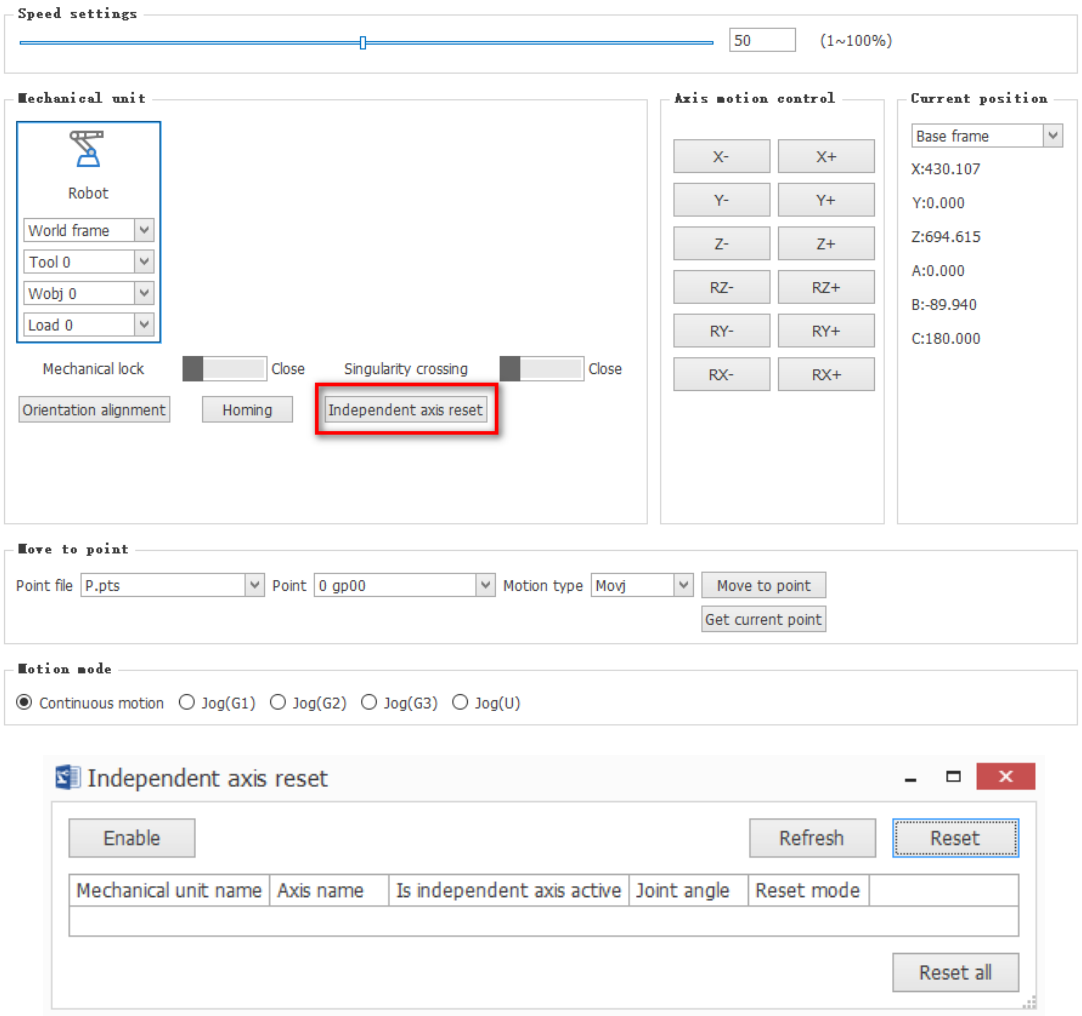
- Function 3: Switch the independent axis to normal mode

Function	Description
IndReset MecUnit, Axis, Old	This instruction resets an axis in the mechanical unit to the normal mode.
IndReset MecUnit, Axis, RefPos, Direction	This instruction switches the independent axis to the normal mode. The number of turns of the axis is adjusted according to RefPos and Direction.
IndReset MecUnit, Axis, RefNum, Direction	This instruction switches the independent axis to the normal mode. The number of turns of the axis is adjusted according to RefNum and Direction.

> **CAUTION**
>
> When the IndReset instruction is executed, the current independent axis and normal axis must not be in motion. After executing the IndCMove instruction at zero speed, a delay of 0.2 seconds is required to ensure that the correct state is achieved. For the external axis with poor performance, pause for a longer time to ensure that the axis stops completely.

For the IndCMove instruction, the speed must be greater than or less than 0 so that the independent axis can move. During motion, you can detect whether the independent axis reaches the specified speed by using the IndSpeed instruction. To end continuous rotation of independent axes, set the speed of the axis to 0 by using the IndCMove instruction. After the axis stops rotation, you can reset the independent axis using the IndReset instruction to switch it back to a normal axis.

- Function 4: Monitoring and quick reset of the independent axis



You can quickly view and reset the independent axis state on the axis control panel.

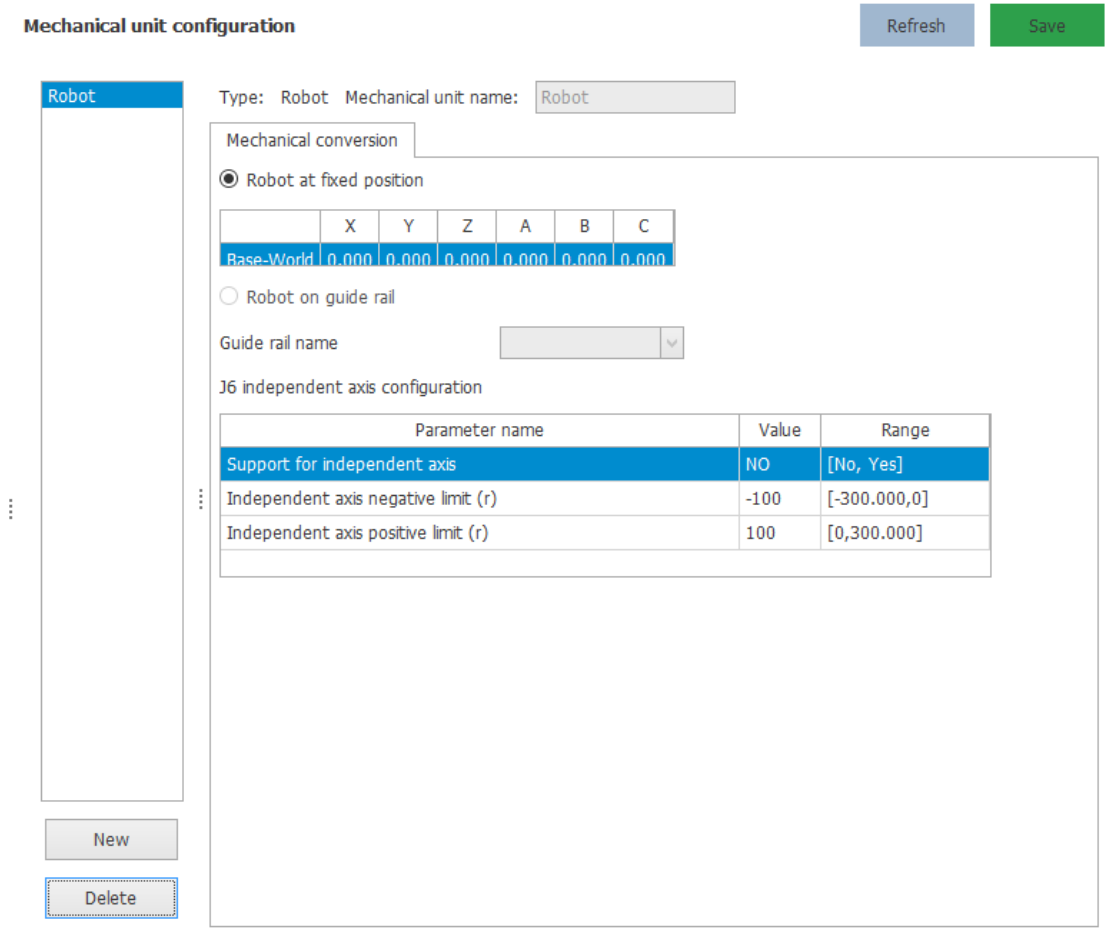
Steps

1. Configure the servo parameters (620N servo not supported).
   For the external axis, you need to configure the servo parameter according to the following table.

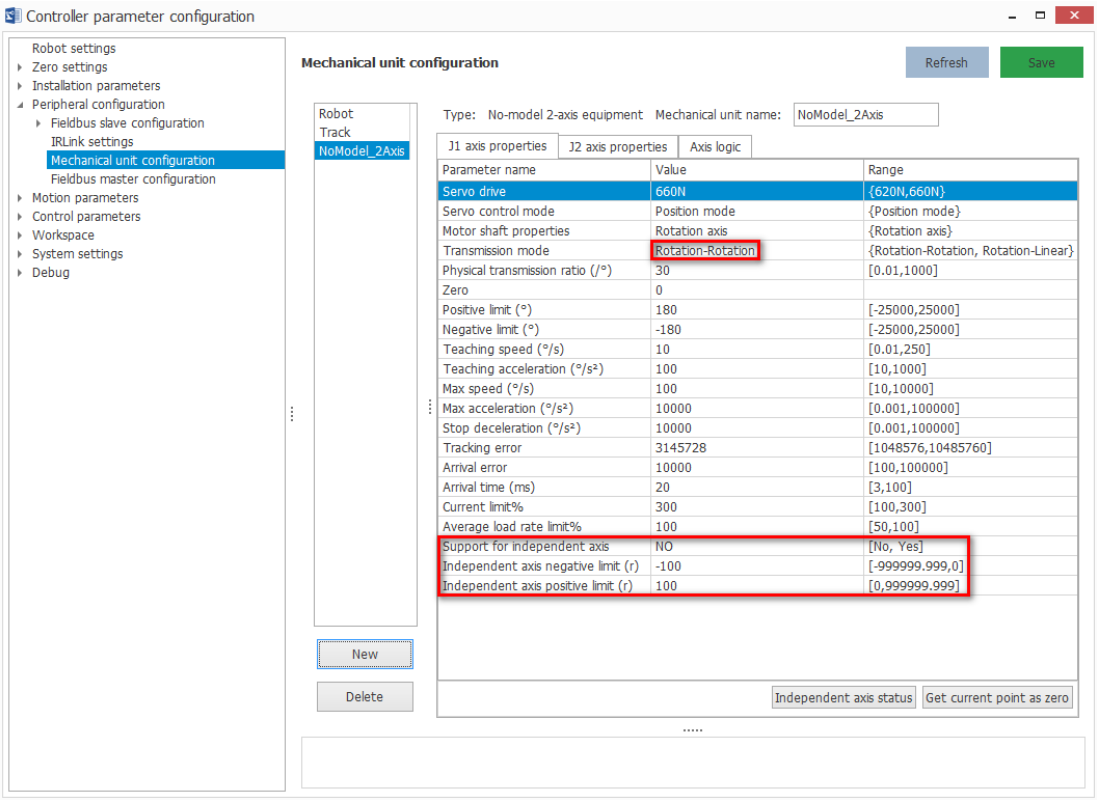
Parameter	Description	Setting value
H0A-36	Encoder multi-turn overflow fault selection	1: Enabled

2. Configure the robot controller.

   The interface for configuring the robot independent axis is as follows. Setting the last axis as independent axis is only available for 6-axis robot. Set the "Support for independent axis" parameter to "YES". Set the limit of the independent axis as needed. The unit is the number of turns. After the "Support for independent axis" parameter of J6 axis is set to "YES", the original joint limit is changed to the independent axis limit. To make the independent axis configuration take effect, save the configuration in emergency stop state and restart the robot controller.



   The following is the interface for configuring the independent axis function for external axes. Only the external axis whose transmission type is set to "rotation-rotation" can be set as an independent axis. Set the "Support for independent axis" parameter to "YES". Set the limit of the independent axis as needed. The unit is the number of turns. After the "Support for independent axis" parameter is set to "YES", the original joint limit is changed to the independent axis limit. To make the independent axis configuration take effect, save the configuration in emergency stop state and restart the robot controller.

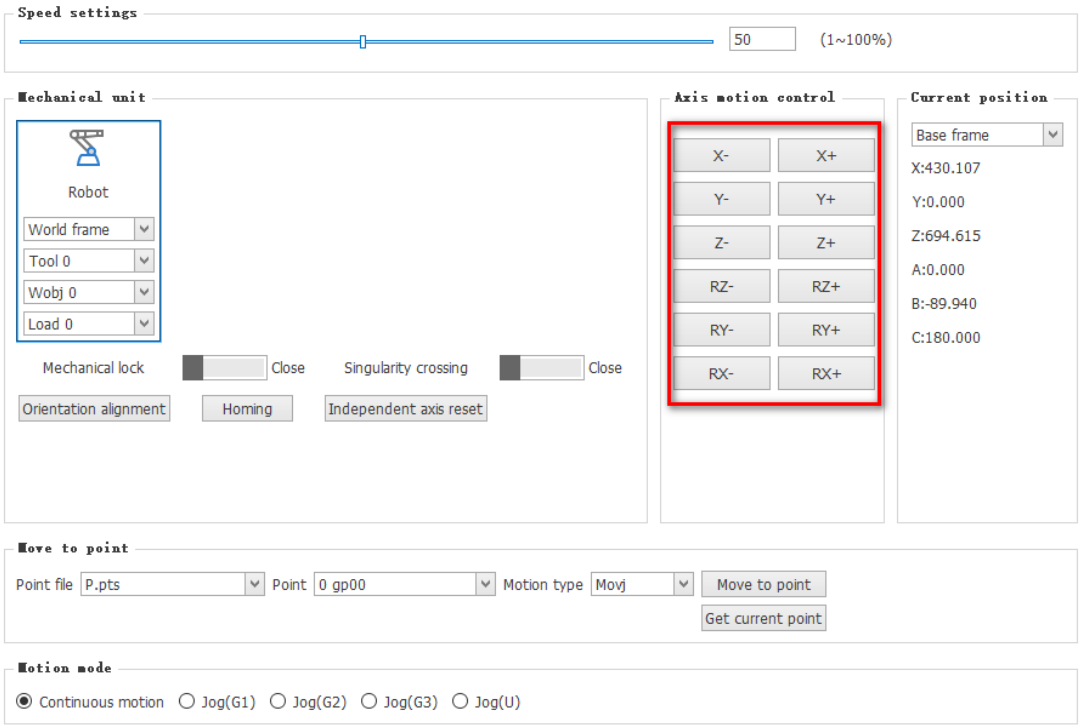


3. Instructions for use
   - Independent axis configuration takes effect after the robot controller is restarted.
   - After the independent axis function is enabled for a normal axis, you need to switch the normal axis to the independent axis and the position mode to the speed mode in the static state of the robot before continuous rotating motion. After that, the robot ignores the position of the independent axis in the point data when executing motion instructions of non-independent axes.
   - A new IndCMove instruction can be executed immediately before the speed specified in the first IndCMove instruction is reached. If a new IndCMove instruction is executed immediately before the execution of the previous instruction is completed, the new instruction immediately overwrites the previous instruction.
   - If the program is stopped or the robot is disabled during continuous rotational motion of an independent axis, the axis will stop. When the program is resumed, the independent axis is started automatically.
   - If the program returns to the start line by user manually or due to other conditions, the independent axis is automatically reset and returns to the normal mode. (The effect is equivalent to executing the instruction IndReset MecUnit, Axis,Old.)
   - If the power is lost during continuous rotating motion of the independent axis, the axis properties of the robot will remain unchanged after power-on again. The independent axis is automatically reset to the normal mode. If the independent axis has a large joint angle upon power loss, the joint angle will be reset upon power-on again.
   - Before executing the IndRest instruction, make sure axes other than the independent axis arrive at the specified position or speed.
   - In single-step motion, when the IndCMove instruction is executed, the independent axis starts and pauses as the program starts and pauses. When the program is in the running state, the independent axis also runs. When the program stops, the independent axis also stops.

- When the motion trajectory of the robot contains an independent axis, if the program is paused and the independent axis is reset to a normal axis, the trajectory recovery will not be performed.
- When the program is finished, the independent axis stops.

4. Jogging and teaching

When an active independent axis exists, the axis control panel is grayed and inoperable, and robot teaching cannot be performed. If you need to teach an axis, execute the IndReset instruction, or reset all active independent axes to normal axes in the axis control panel.



When a certain joint is configured as an independent axis, the position limit of the joint is modified to the position limit of the independent axis. When this axis exceeds the original joint limit, the inverse kinematics fails when the joint is involved. Before teaching, clear the multi-turn value of the axis in the axis control panel.

CAUTION

- The target axis to be set as an independent axis must not have a hard limit.
- If an external axis is configured as an independent axis, functions not supported by the external axis are not supported by the independent axis either.
- The trajectory recovery function is not available to joints set as independent axes.
- If a joint is switched to the independent axis state, the points of the joint can be acquired, but the acquired points cannot be reached.
- If a joint is switched to the independent axis state, the joint will not return to zero.
- The independent axis function cannot be used with the ECP function.
- When an active independent axis exists, the self-learning vibration suppression function is not supported. When the "SLVSMode level" parameter is used, all instructions are configured with a self-learning level. This is mutually exclusive with the independent axis unless this parameter is set to "OFF".
- When the J6 axis is configured as the active independent axis, high trajectory precision feature is automatically turned off.
- When the angle of a joint configured as an independent axis is greater than the original joint limit, the "Move to point" operation results in a prompt that the point exceeds the limit range. When the CP instruction (motion instructions in Cartesian space, such as: Linear, circular, jumpL, free curve, etc.) is executed, it prompts that inverse kinematics cannot be performed.
- The independent axis function does not support the use of eccentric tools.
- When the robot is in dynamic tracking state, independent axis instructions are not supported.
- When an active independent axis exists, the teaching function is not supported unless all the independent axes are reset to normal axes.
- If a coordination axis is configured as an independent axis and the coordination function is enabled, the coordination effect may not be satisfying. This is not recommended.
- The independent axis function is not supported for 620N servo drives.
- After the independent axis is reset, the trajectory is cleared. If the independent axis reset is triggered in the stop state, the program runs from the yellow cursor line when it is restarted and the preceding trajectory is cleared.
- For the IR-R60-210S5-E1N, IR-R60-210S5, IR-R60-210S5-E1ZY, IR-R80-210S10-E1N, and IR-R80-210S5-E1ZY series robots, their axes cannot be configured as independent axes.
- When the joint angle exceeds the joint threshold of the axis before it is configured as an independent axis, the joint angle is only used as a reference. When the joint angle is very large, it will be reset to a reasonable value after the controller is powered on again.
- Do not take points when the joint angle is very large. When the axis pulse of the start, intermediate, and target points of the motion instruction exceed 2147480000, the motion instruction cannot be executed.
- If the J6 axis of the robot is in the independent axis state and does not move, the position of the J6 axis changes slightly due to axis coupling when the J5 axis moves.
- The virtual controller currently does not support the independent axis function.
- When the J6 axis is configured as an independent axis and activated, collision detection will not take effect.

12.5 Load Auto-Tuning

12.5.1 Overview

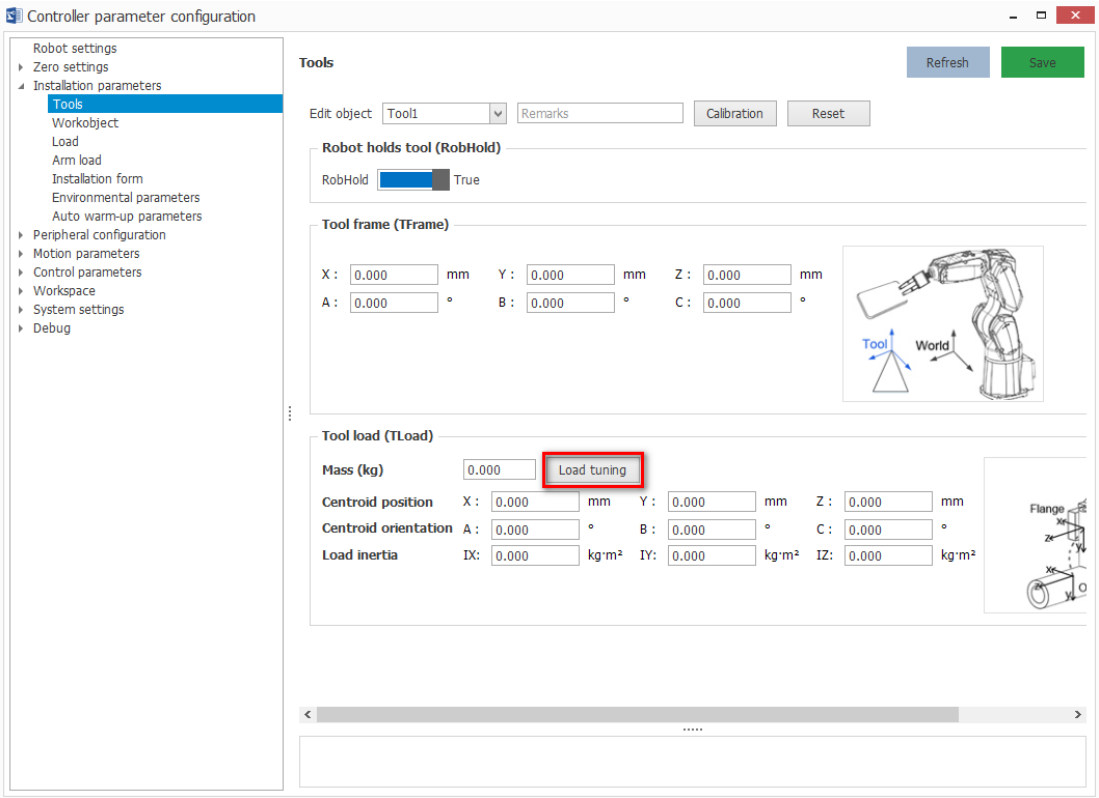
- The load auto-tuning function is used to obtain load parameters by running set robot motion trajectories. It simplifies the acquisition of load parameters.
- The load auto-tuning function supports the auto-tuning of tool load (TLoad) and grip load (Load).
- Load auto-tuning is recommended when the shape of load is complex and no 3D load model is available. If the 3D load model is available, it is recommended to use 3D software to export the load parameters. Generally, the accuracy of the exported load parameters is higher than that obtained by the load auto-tuning function.

12.5.2 6-Axis Robot Load Auto-Tuning

12.5.2.1 Opening Load Auto-Tuning

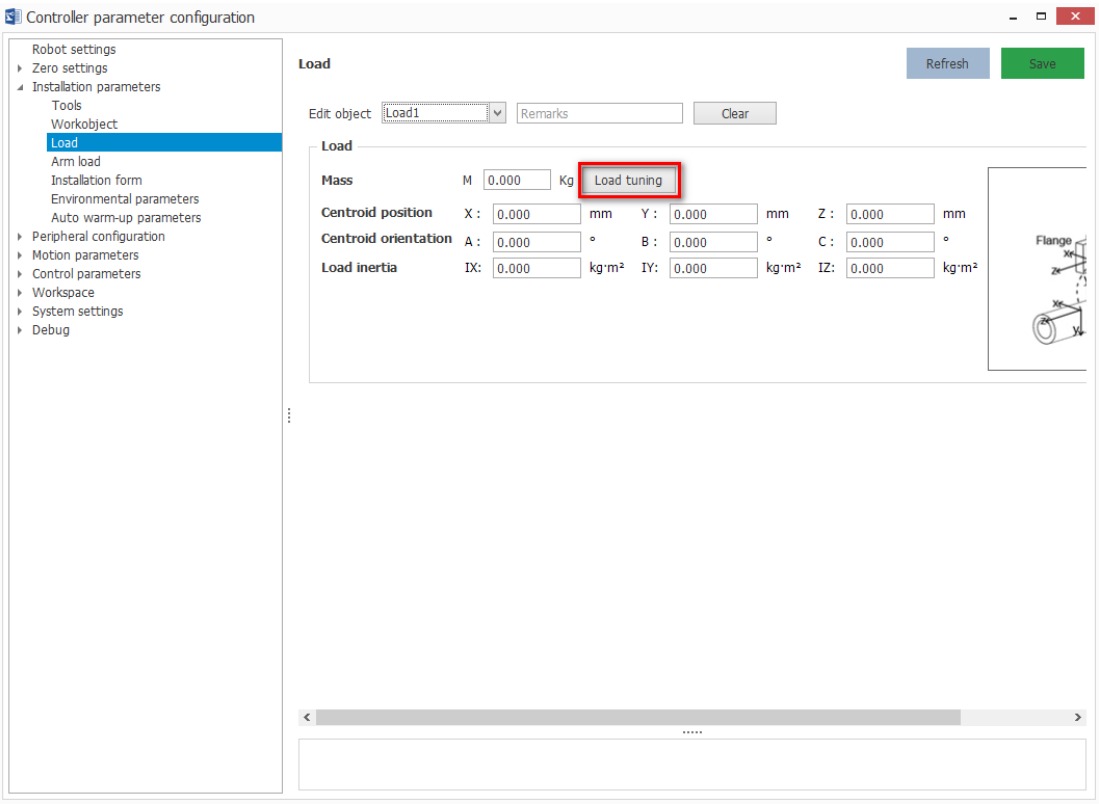
Tool load auto-tuning

Go to "Controller parameter configuration" > "Installation parameters" > "Tools", select a tool other than Tool0, and click "Load tuning" under "Mass" section to enter the load tuning interface.



Grip load auto-tuning

Go to "Controller parameter configuration" > "Installation parameters" > "Load", select a load other than Load0, and click "Load tuning".

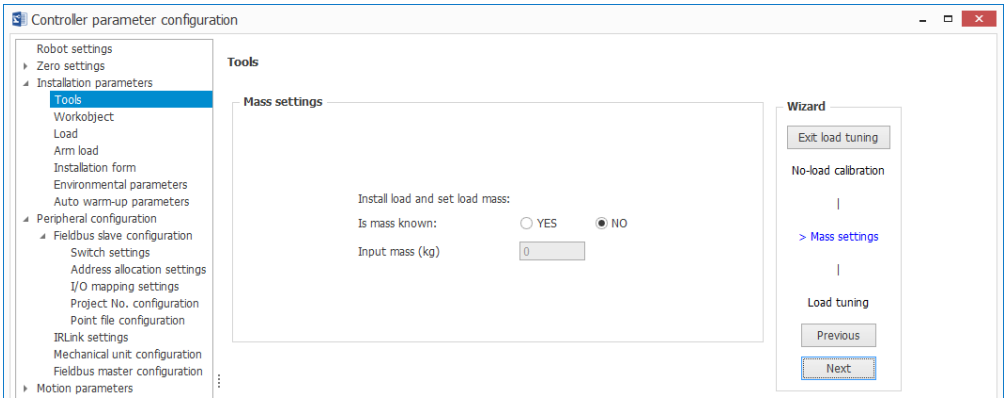


12.5.2.2 Tool Load Auto-Tuning

1. Set the mass.

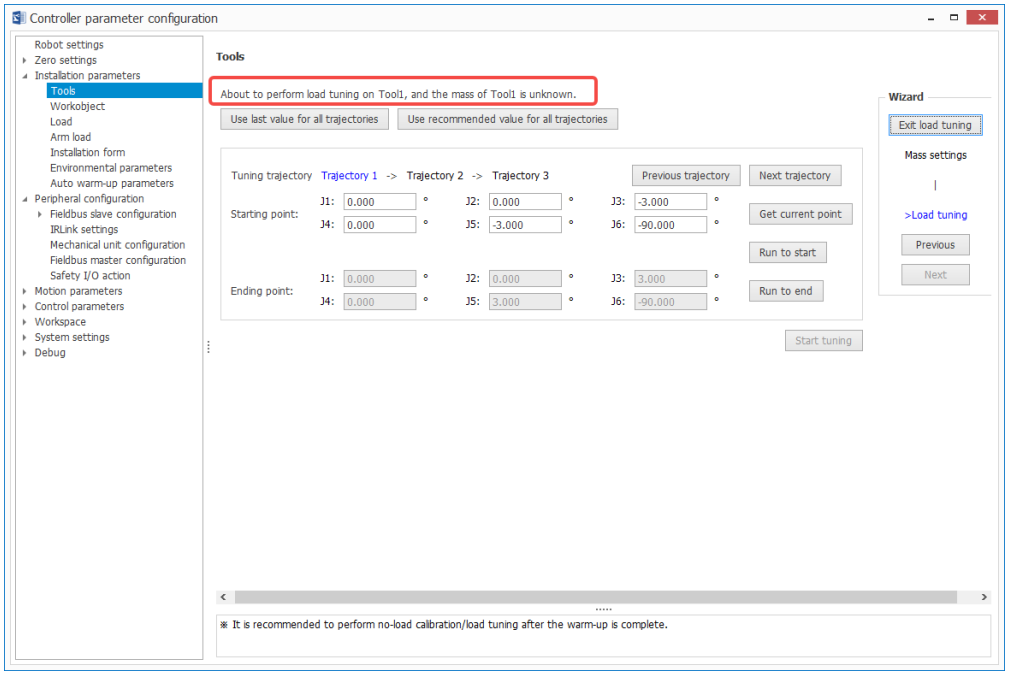
   1) If the load mass is known, select "YES" and enter the mass. If the load mass is unknown, select "NO". Select "YES" and correctly set the load weight for more accuracy result.

   2) Click."Next"to enter the trajectory setting interface.

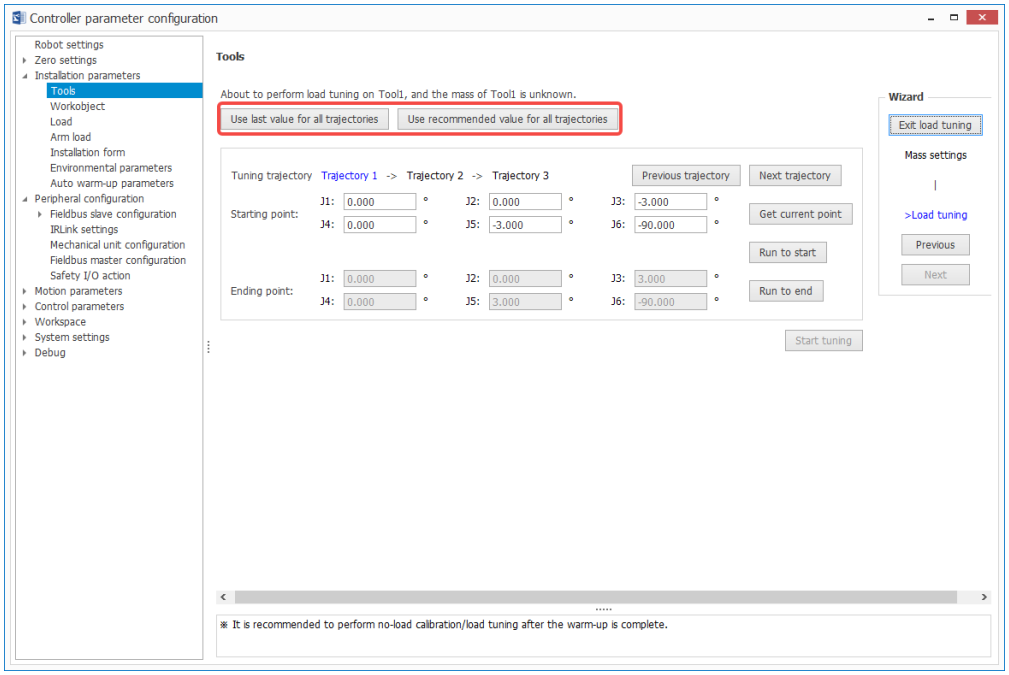


2. Set the auto-tuning trajectory.

   1) Check that the load number and mass setting value of the tool for auto-tuning are correct.



2) If it is the first use of the load auto-tuning function or the robot installation method has changed, it is recommended to click "Use recommended value for all trajectories" to recalculate the recommended trajectory; in other cases, it is recommended to click "Use last value for all trajectories", which will default to using the last successfully auto-tuned trajectory.



3) You need to set 3 auto-tuning trajectories: If they have been successfully auto-tuned before, the successful trajectories will be displayed by default; otherwise, the recommended values will be displayed.

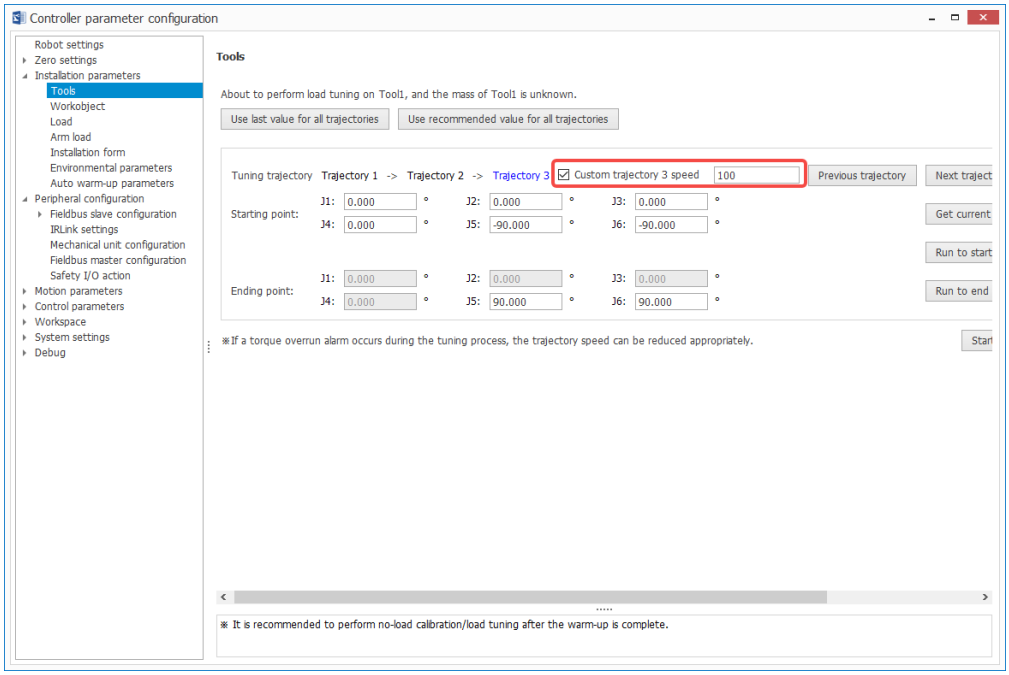
6-axis robot trajectory requirements:

For Trajectory 1 and Trajectory 2, ensure the forearm and the axes of J3/J5 are as parallel to the ground as possible, otherwise the auto-tuning accuracy may be affected. It is recommended to use the recommended values as much as possible. The joint angles in the edit box can only be manually modified in Admin/Manufacturer mode.

- Trajectory 1: For J1, J2, J4 and J6, the joint angle at the start point is identical to that at the end point. For J3 and J5, the displacement from the start point to the end point is 6°.

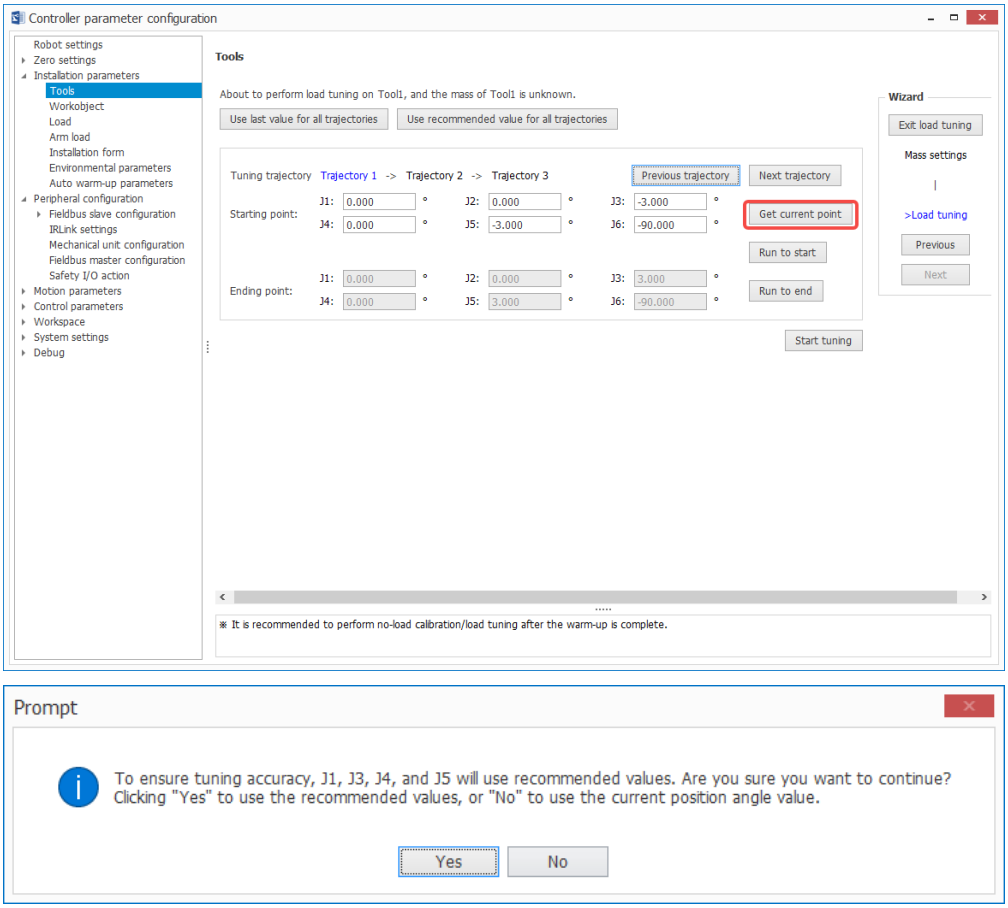
- Trajectory 2: For J1 to J5, the joint angle at the start point is identical to that at the end point. For J6, the difference between the joint angles at start point and the end point should be as large as possible.
- Trajectory 3: For J1 to J4, the joint angle at the start point is identical to that at the end point. For J5 and J6, the difference between the joint angles at start point and the end point should be as large as possible.

  If a "Torque overlimit" alarm occurs during the auto-tuning process, return to the Trajectory 3 settings interface, check "Custom trajectory 3 speed", and appropriately reduce the speed value. The adjustment principle is: Choose the highest possible operating speed without triggering a "Torque overlimit" alarm.



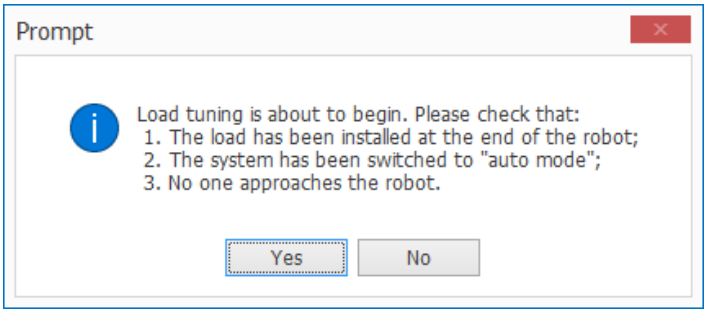
4) Check that the recommended trajectory is reachable. You can click "Run to start" and "Run to end" to see whether interference occurs on the current trajectory. If the trajectory is unreachable, adjust it manually according to the orientation requirements of the trajectory.

It is recommended to use "Get current point" for point calculation, and the system will automatically calculate the recommended point positions for Trajectory 1 and Trajectory 2 based on the robot installation angle and J2 joint angle. After using "Get current point", reconfirm whether the start/end points are reachable.

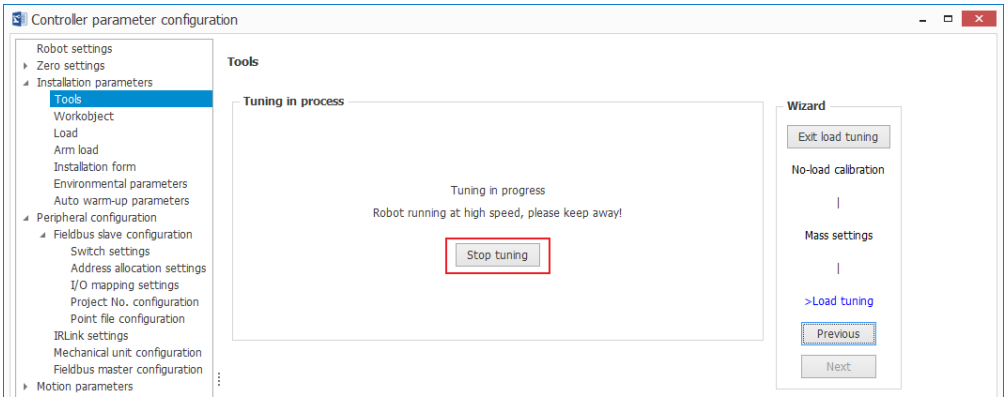


5) If you confirm that the trajectory is reachable, switch the robot operating mode to "Auto" and click "Start tuning".

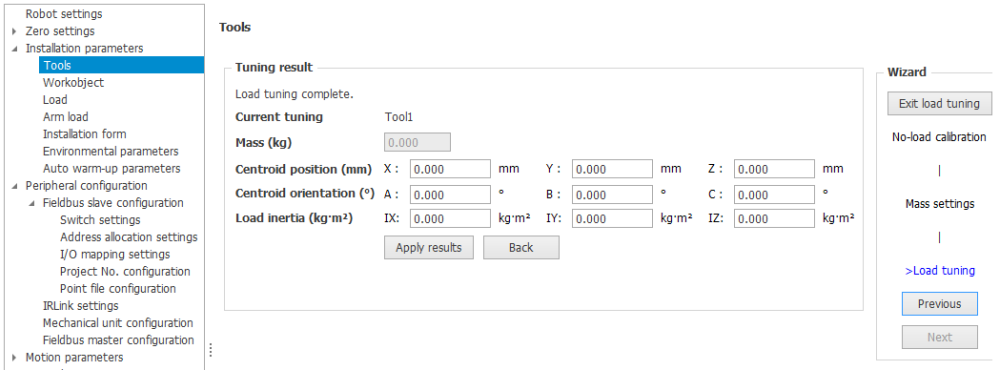
1. In the pop-up dialog, click "Yes" to start tuning if all conditions are met. Otherwise, click "No" to return to the setting page and click "Start tuning" again once all conditions are met.

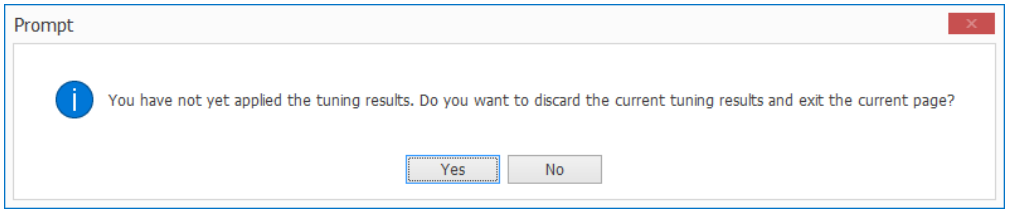


2. During auto-tuning, if you want to stop auto-tuning in advance, click "Stop tuning".



3. After auto-tuning is completed, the screen displays the auto-tuning results. Click "Apply results" to write the auto-tuning results to the corresponding tool. If you want to discard the auto-tuning results, click "Back" and then click "Yes" in the pop-up dialog.





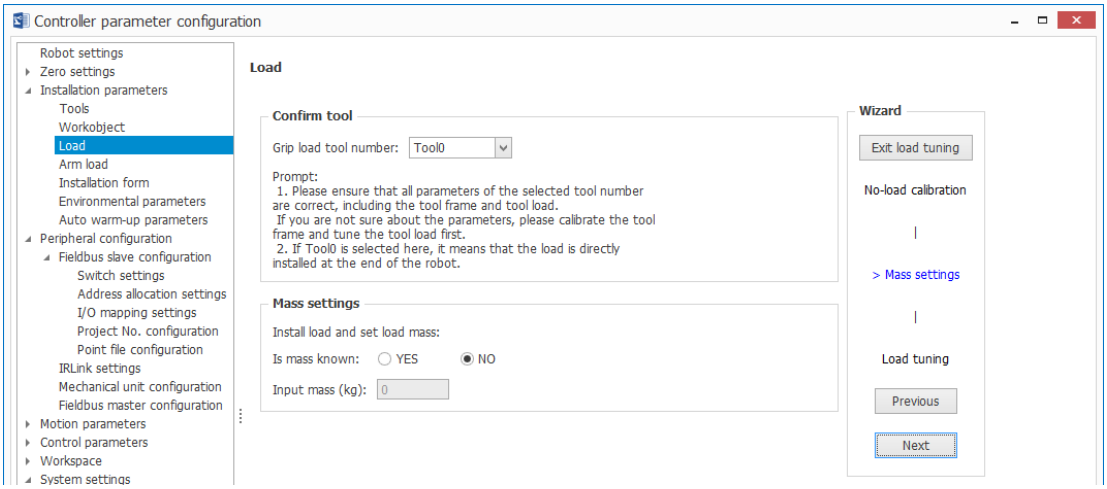
12.5.2.3 Grip Load Auto-Tuning

When performing grip load auto-tuning, you need to set accurate tool load frame and tool load parameters. The tuning result will only include grip load parameters.

1. Set the mass.

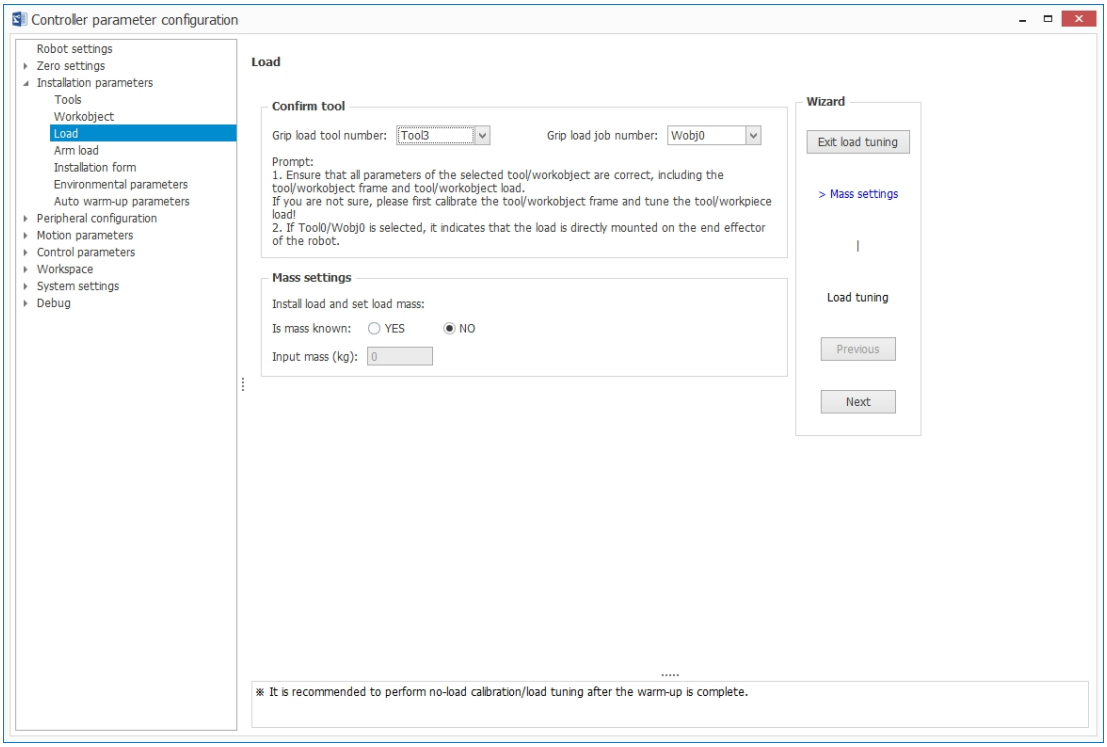
   TCP mode:

   In TCP mode (the selected tool number is in gripping state), workobject number is not required.



   ECP mode:

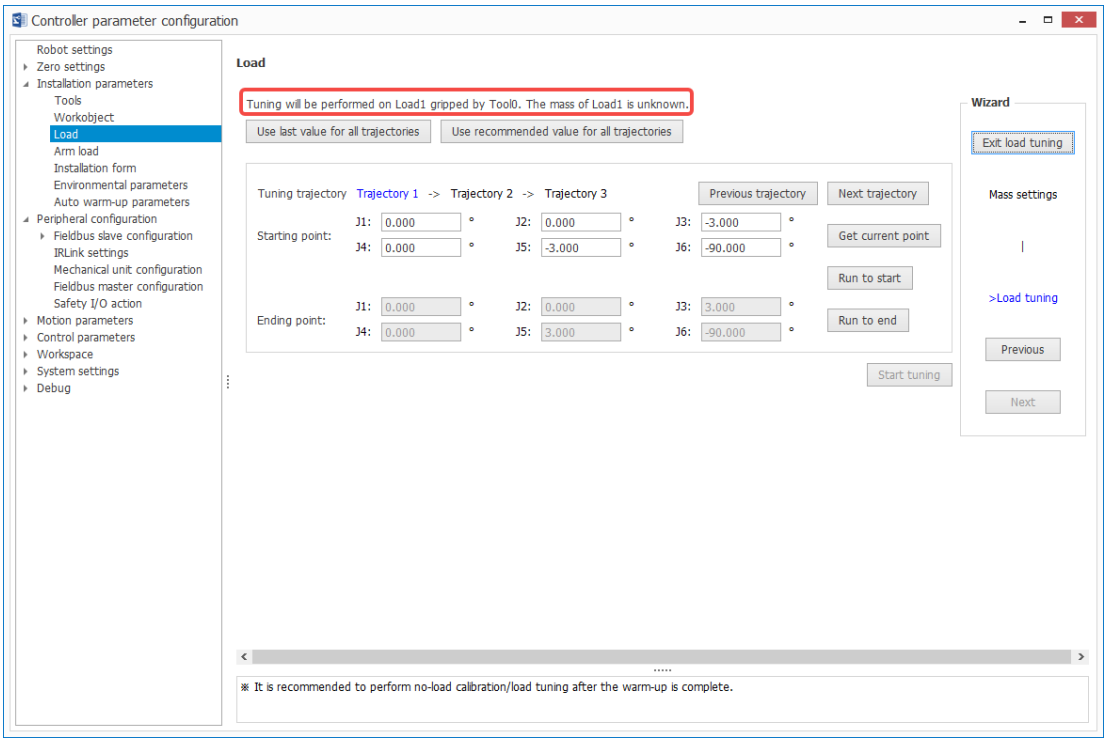
   In ECP mode (the selected tool number is in non-gripping state), the setting of workobject number is required. The tool number and workobject number must be used in pairs (to ensure mutual exclusivity between gripping and non-gripping states). If any settings are incorrect, you will not be able to click "Next" to proceed.



- Select the number of the tool that grips the load. Ensure that the "TFrame" and "TLoad" parameters of the selected tool are correctly set. Otherwise, the results will be incorrect. If you select "Tool0", it means that the load is directly installed on the end flange.
- If the load mass is known, select "YES" and enter the mass. If the load mass is unknown, select "NO". Select "YES" and correctly set the load weight for more accuracy result. Click."Next"to enter the trajectory setting interface.

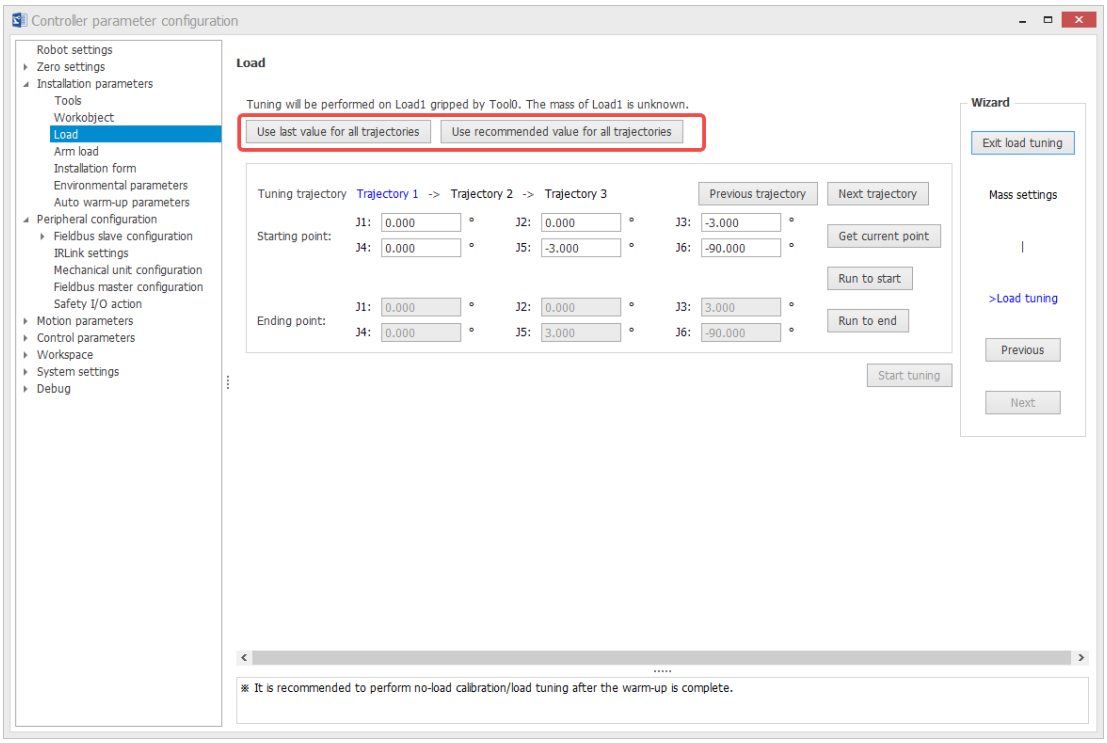
2. Set the auto-tuning trajectory.

1) Check that the load number and mass setting value for auto-tuning are correct.



2) If it is the first use of the load auto-tuning function or the robot installation method has changed, it is recommended to click "Use recommended value for all trajectories" to recalculate the recommended

trajectory; in other cases, it is recommended to click "Use last value for all trajectories", which will default to using the last successfully auto-tuned trajectory.



3) You need to set 3 auto-tuning trajectories: If they have been successfully auto-tuned before, the successful trajectories will be displayed by default; otherwise, the recommended values will be displayed.

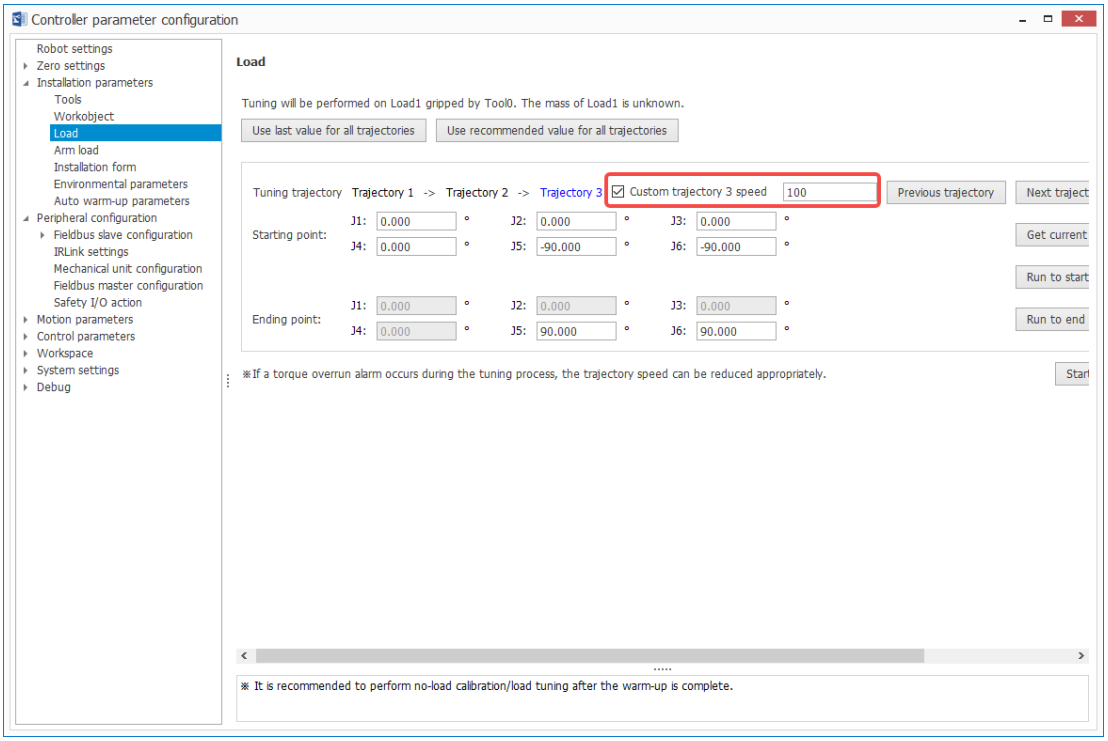
6-axis robot trajectory requirements:

For Trajectory 1 and Trajectory 2, ensure the forearm and the axes of J3/J5 are as parallel to the ground as possible, otherwise the auto-tuning accuracy may be affected. It is recommended to use the recommended values as much as possible. The joint angles in the edit box can only be manually modified in Admin/Manufacturer mode.

- Trajectory 1: For J1, J2, J4 and J6, the joint angle at the start point is identical to that at the end point. For J3 and J5, the displacement from the start point to the end point is 6°.
- Trajectory 2: For J1 to J5, the joint angle at the start point is identical to that at the end point. For J6, the difference between the joint angles at start point and the end point should be as large as possible.
- Trajectory 3: For J1 to J4, the joint angle at the start point is identical to that at the end point. For J5 and J6, the difference between the joint angles at start point and the end point should be as large as possible.

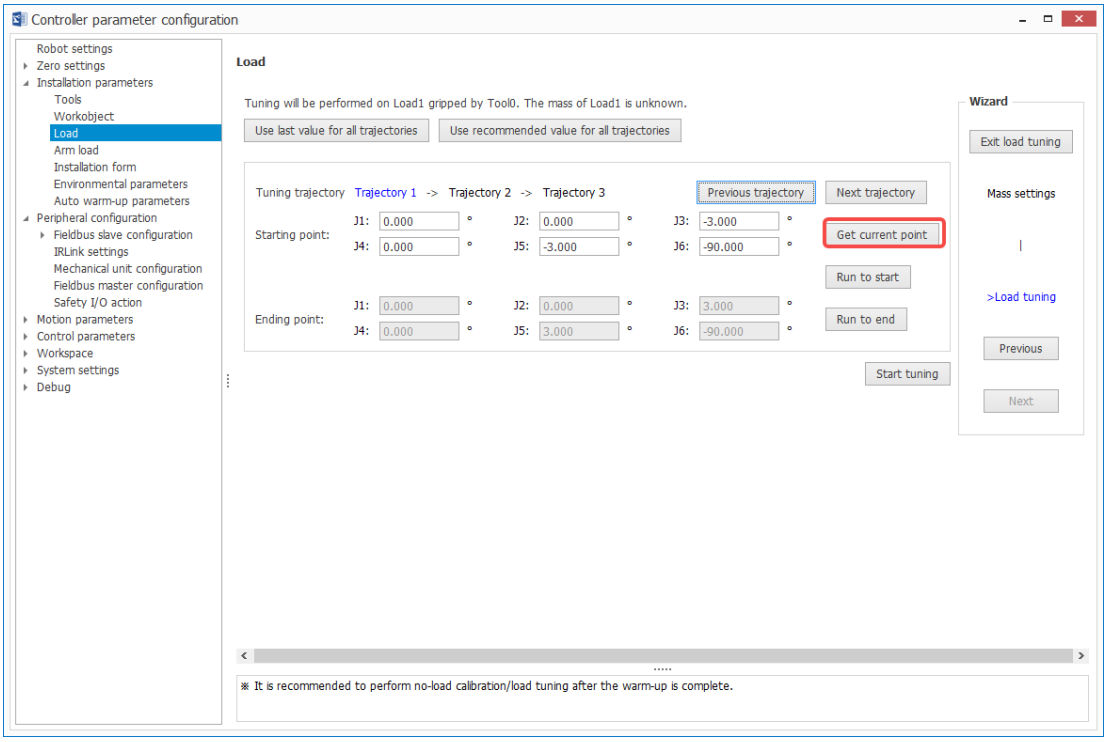


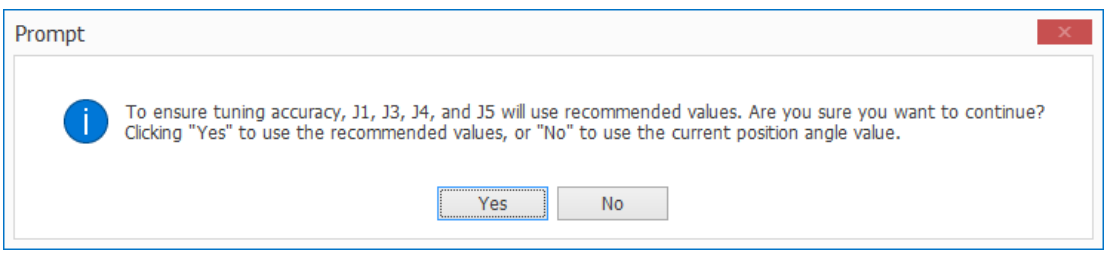
If a "Torque overlimit" alarm occurs during the auto-tuning process, return to the Trajectory 3 settings interface, check "Custom trajectory 3 speed", and appropriately reduce the speed value. The adjustment principle is: Choose the highest possible operating speed without triggering a "Torque overlimit" alarm.



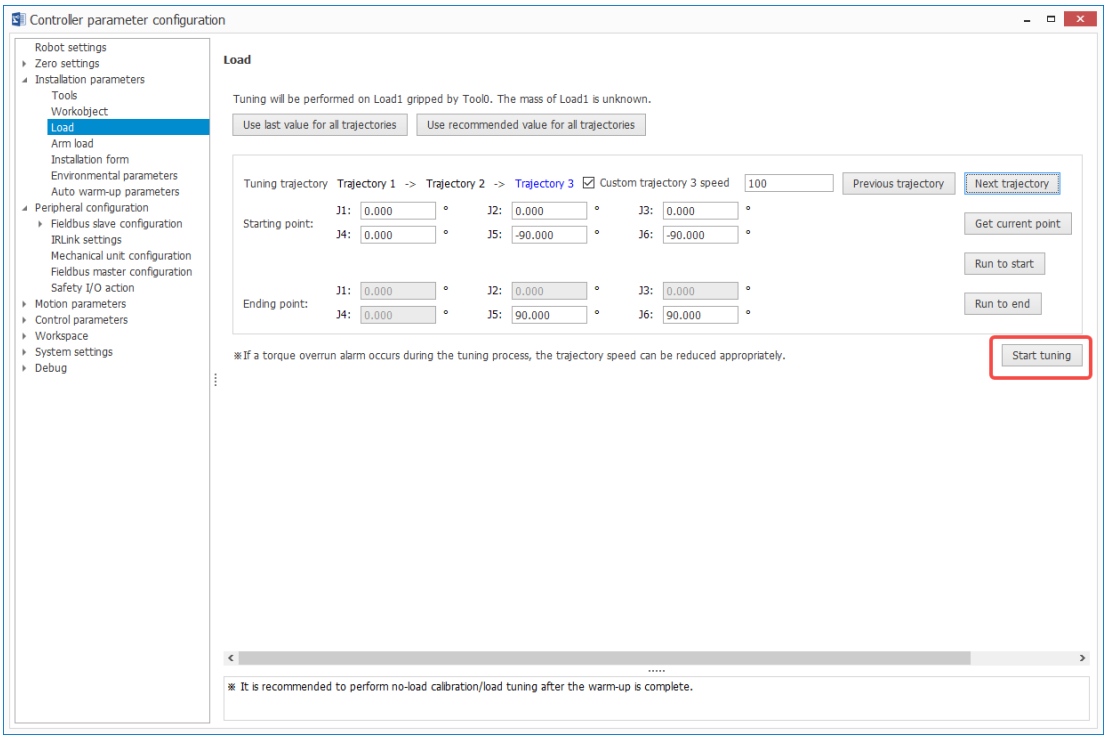
4) Check that the recommended trajectory is reachable. You can click "Run to start" and "Run to end" to see whether interference occurs on the current trajectory. If the trajectory is unreachable, adjust it manually according to the orientation requirements of the trajectory.

It is recommended to use "Get current point" for point calculation, and the system will automatically calculate the recommended point positions for Trajectory 1 and Trajectory 2 based on the robot installation angle and J2 joint angle. After using "Get current point", reconfirm whether the start/end points are reachable.

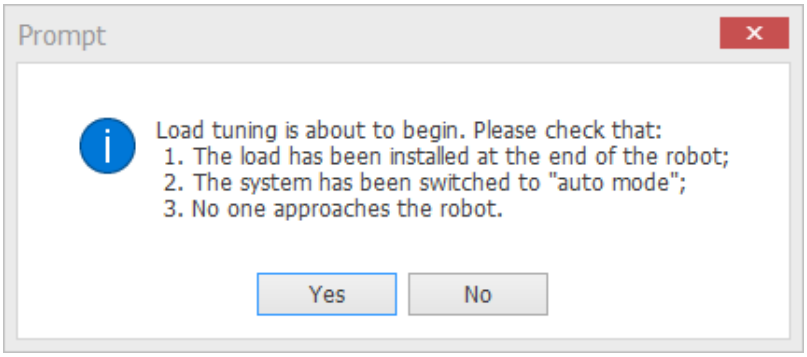




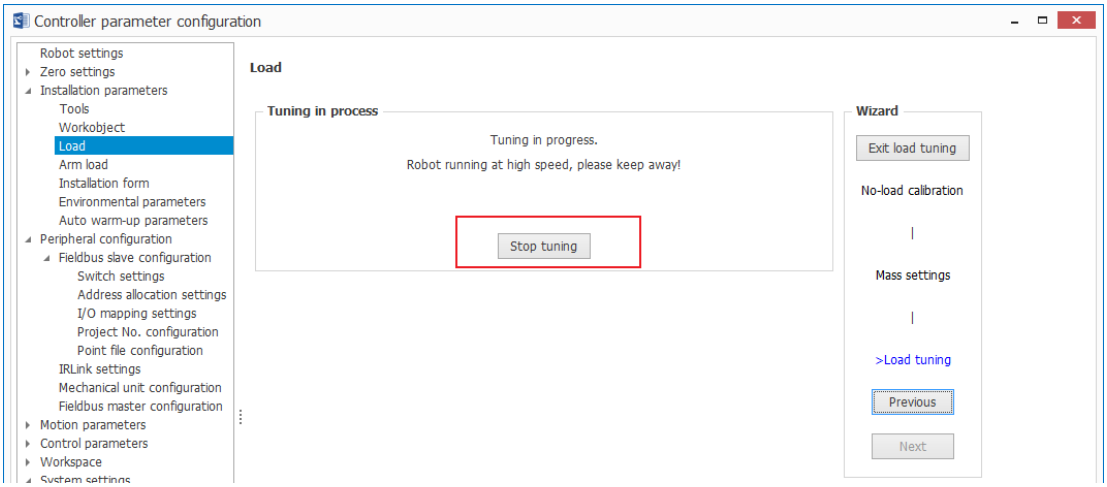
5) If you confirm that the trajectory is reachable, switch the robot operating mode to "Auto" and click "Start tuning".



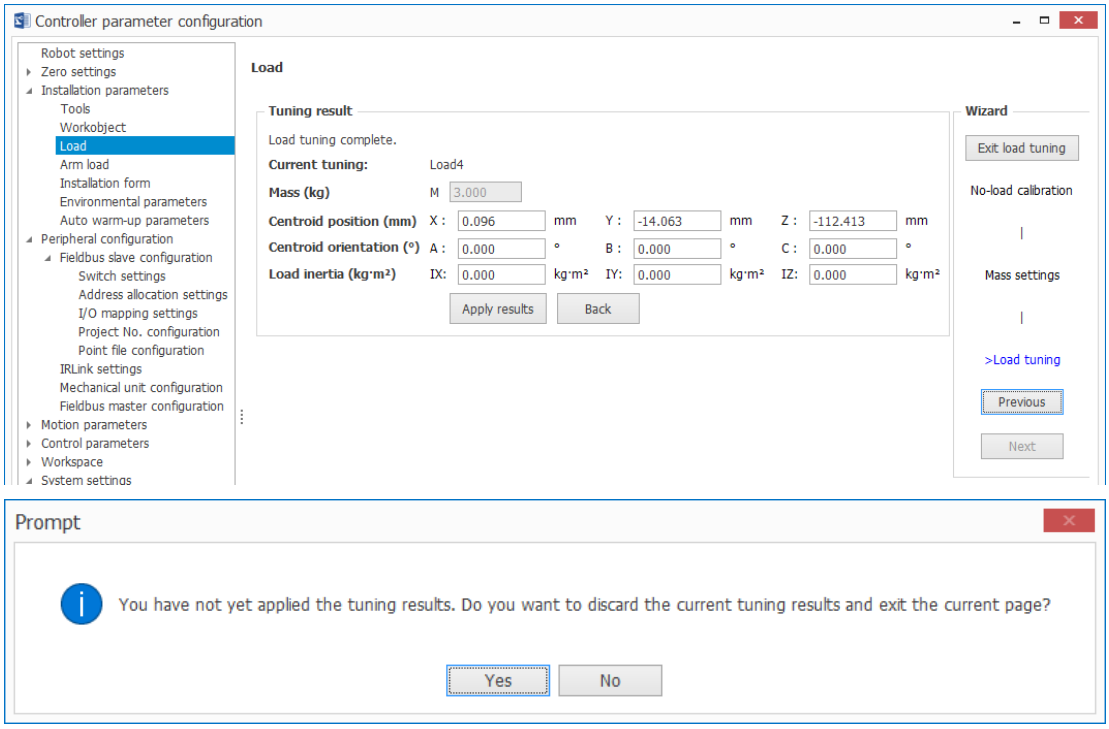
3. In the pop-up dialog, click "Yes" to start tuning if all conditions are met. Otherwise, click "No" to return to the setting page and click "Start tuning" again once all conditions are met.



4. During auto-tuning, if you want to stop auto-tuning in advance, click "Stop tuning".



5. After auto-tuning is completed, the screen displays the auto-tuning results. Click "Apply results" to write the auto-tuning results to the corresponding load. If you want to discard the auto-tuning results, click "Back" and then click "Yes" in the pop-up dialog.

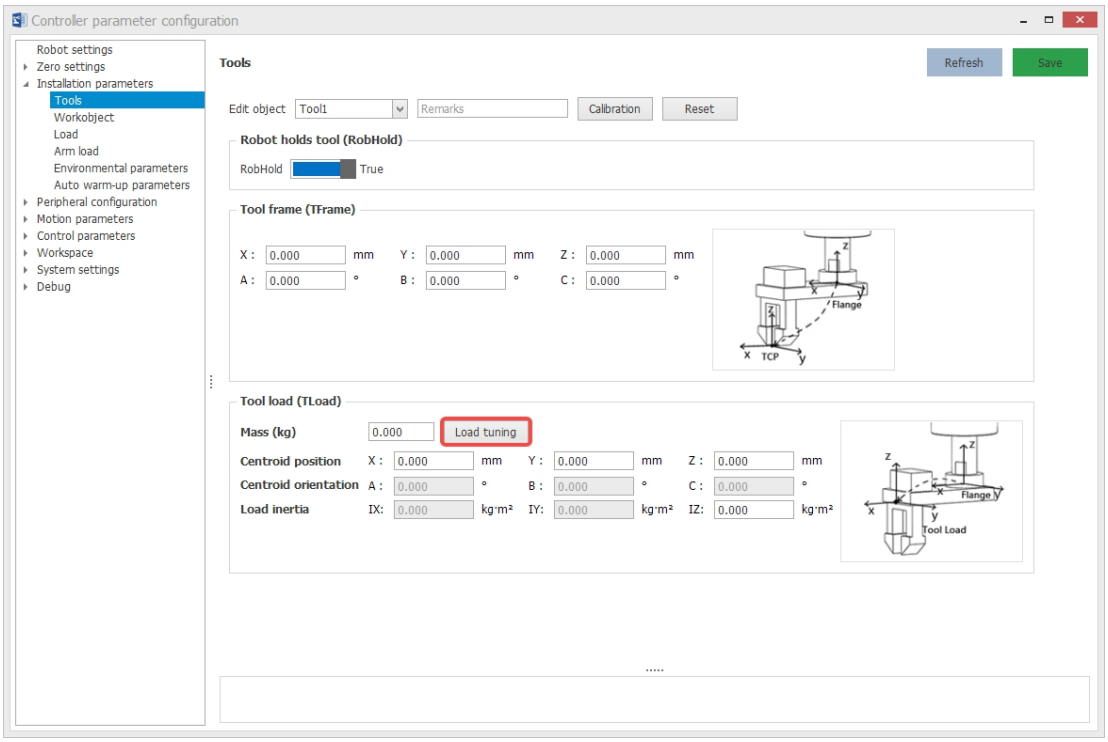


12.5.3 SCARA Robot Load Auto-Tuning

12.5.3.1 Opening Load Auto-Tuning

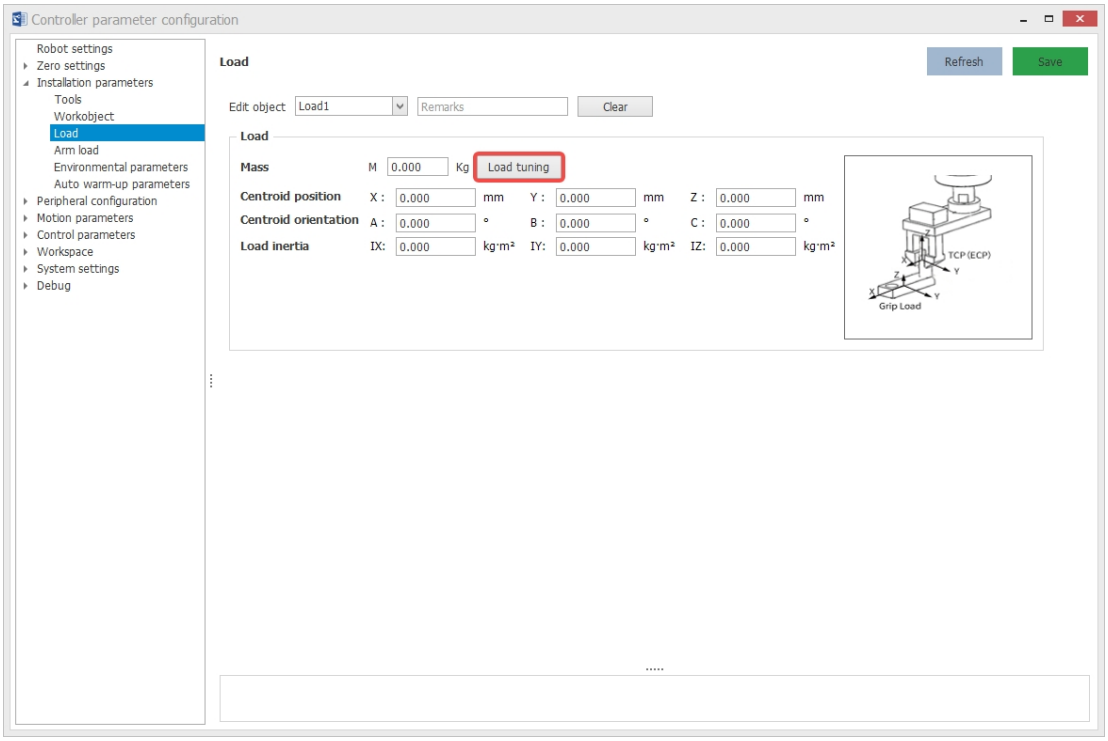
Tool Load Auto-Tuning

Go to "Controller parameter configuration" > "Installation parameters" > "Tools", select a tool other than Tool0, and click "Load tuning" under "Mass" section to enter the load tuning interface.



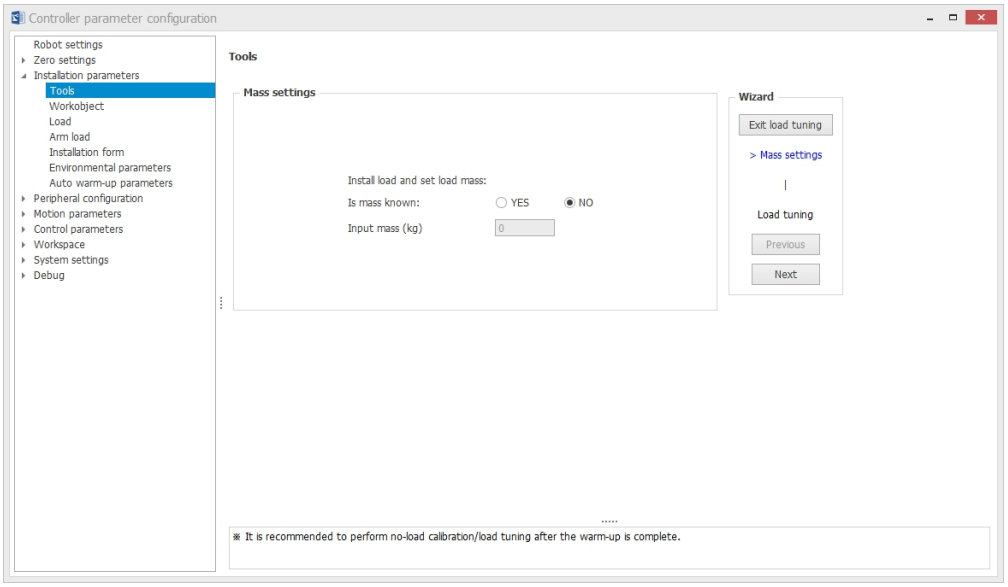
Grip load auto-tuning

Go to "Controller parameter configuration" > "Installation parameters" > "Load", select a load other than Load0, and click "Load tuning".

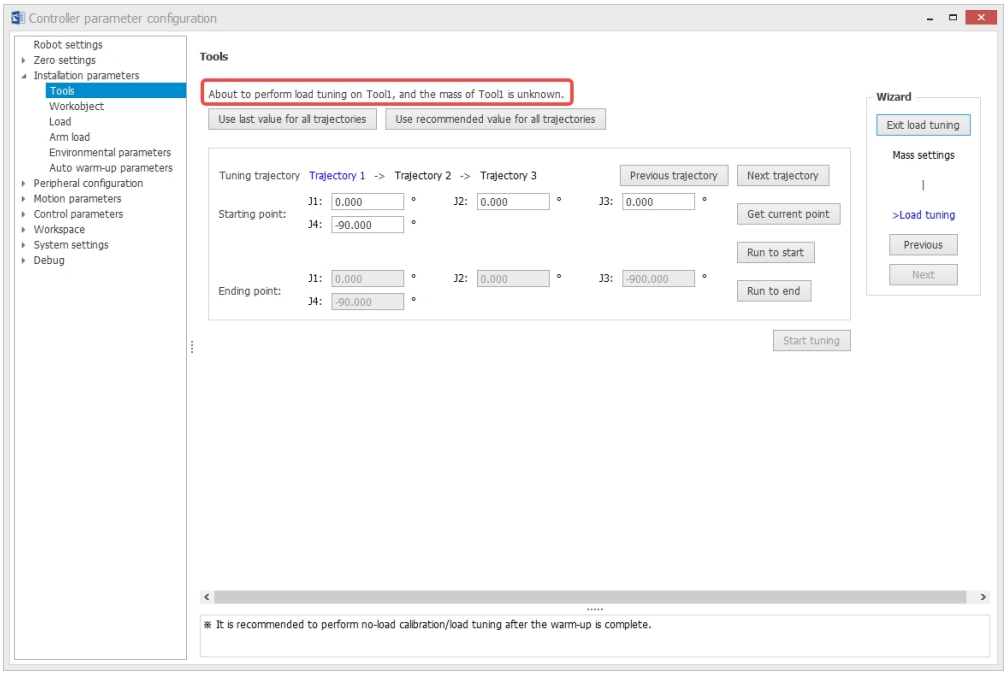


12.5.3.2 Tool Load Auto-Tuning

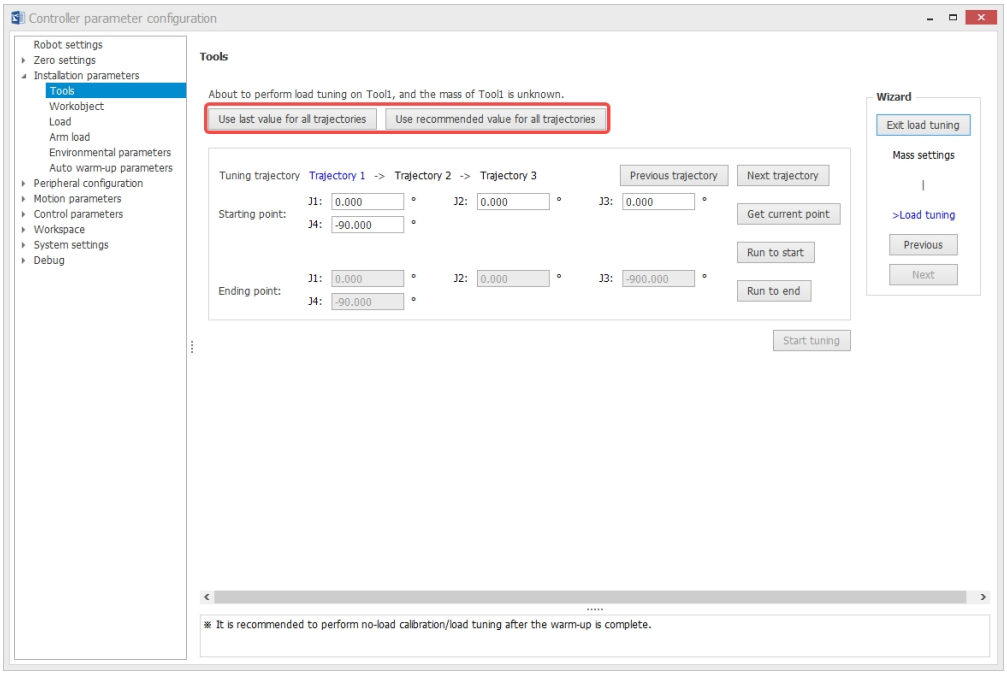
1. Set the mass. If the load mass is known, select "YES" and enter the mass. If the load mass is unknown, select "NO". Select "YES" and correctly set the load weight for more accuracy result.
   Click."Next"to enter the trajectory setting interface.



2. Set the auto-tuning trajectory.
   1) Check that the load number and mass setting value of the tool for auto-tuning are correct.



2) If it is the first use of the load auto-tuning function or the robot installation method has changed, it is recommended to click "Use recommended value for all trajectories" to recalculate the recommended trajectory; in other cases, it is recommended to click "Use last value for all trajectories", which will default to using the last successfully auto-tuned trajectory.



3) You need to set 3 auto-tuning trajectories: If they have been successfully auto-tuned before, the successful trajectories will be displayed by default; otherwise, the recommended values will be displayed.

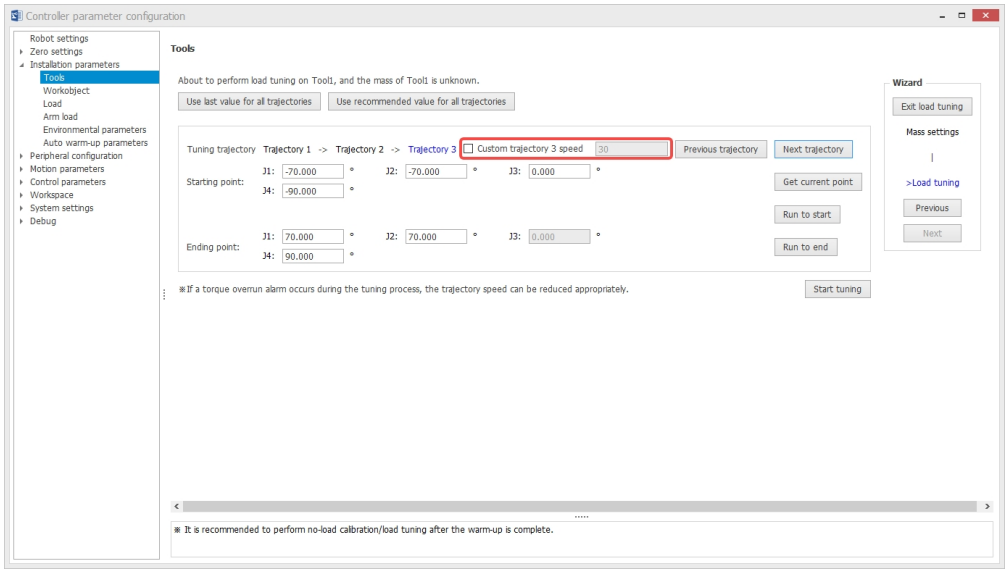
SCARA robot trajectory requirements:

It is recommended to use the recommended values as much as possible. The joint angles in the edit box can only be manually modified in Admin/Manufacturer mode.

- Trajectory 1: For J1, J2 and J4, the joint angle at the start point is identical to that at the end point. For J3, the displacement from the start point to the end point is 50 mm.
- Trajectory 2: For J1 to J3, the joint angle at the start point is identical to that at the end point. For J4, the difference between the joint angles at start point and the end point should be as large as possible.

- Trajectory 3: For J3, the joint angle at the start point is identical to that at the end point. For J1, J2 and J4, the difference between the joint angles at start point and the end point should be as large as possible.

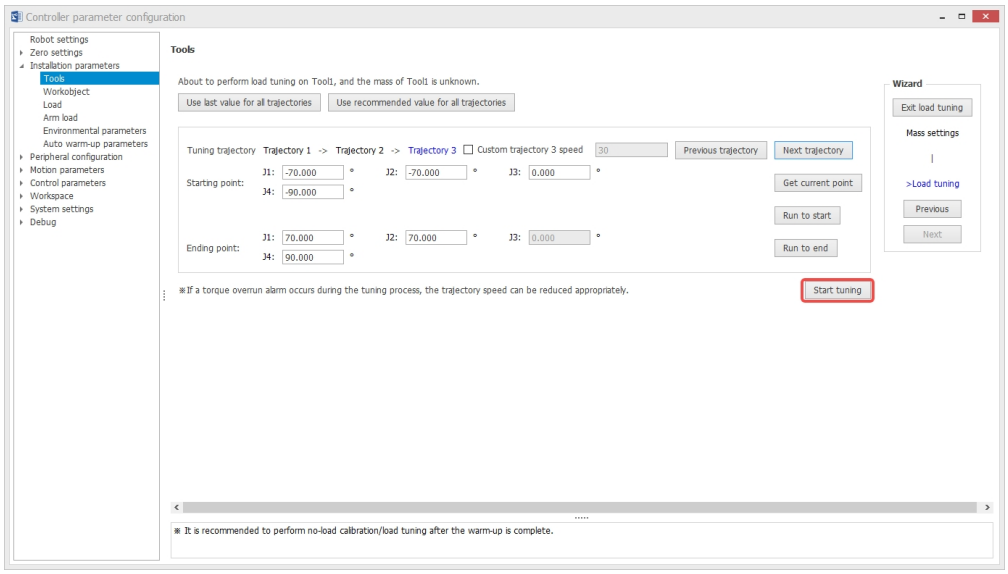
> **CAUTION**
>
> If a "Torque overlimit" alarm occurs during the auto-tuning process, return to the Trajectory 2 or Trajectory 3 settings interface, check "Custom trajectory speed", and appropriately reduce the speed value (30 by default). The adjustment principle is: Choose the highest possible operating speed without triggering a "Torque overlimit" alarm.



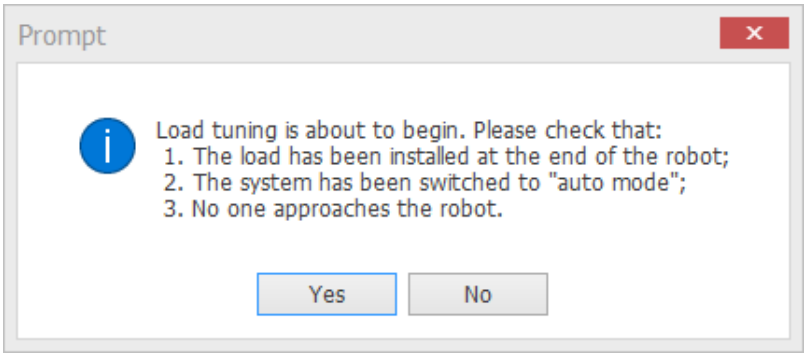
4) Check that the recommended trajectory is reachable. You can click "Run to start" and "Run to end" to see whether interference occurs on the current trajectory. If the trajectory is unreachable, adjust it manually according to the orientation requirements of the trajectory.

Click "Get current point", and the system will set the current joint angles as the start point of the current trajectory.

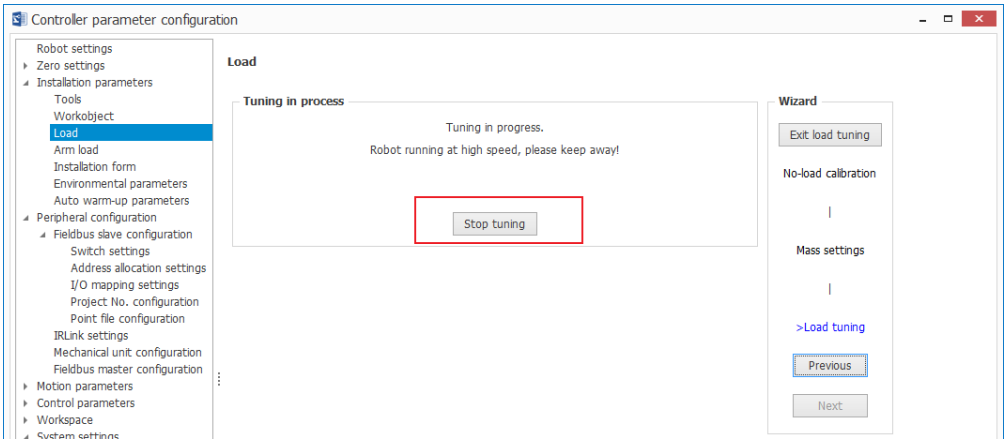
5) If you confirm that the trajectory is reachable, switch the robot operating mode to "Auto" and click "Start tuning".



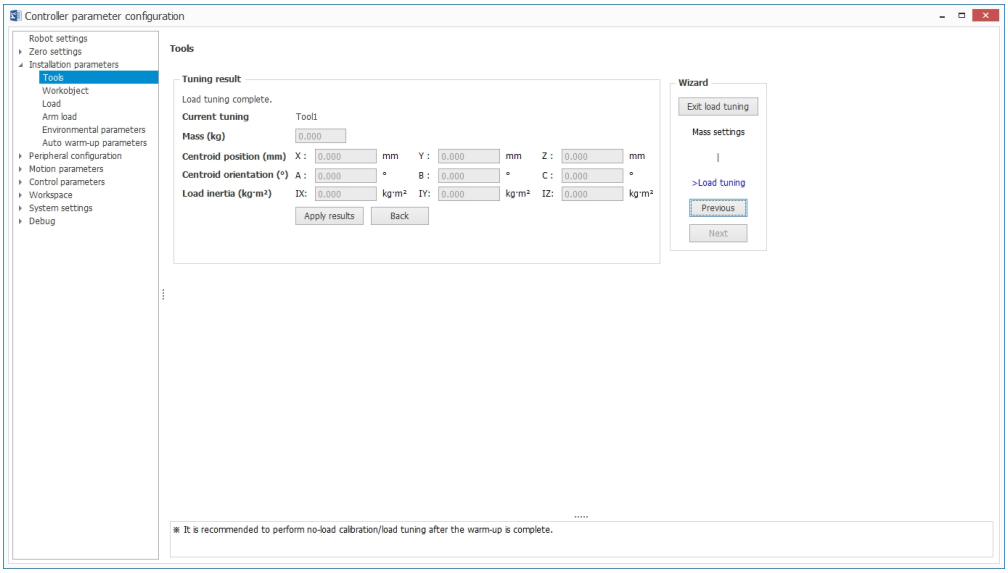
3. In the pop-up dialog, click "Yes" to start tuning if all conditions are met. Otherwise, click "No" to return to the setting page and click "Start tuning" again once all conditions are met.

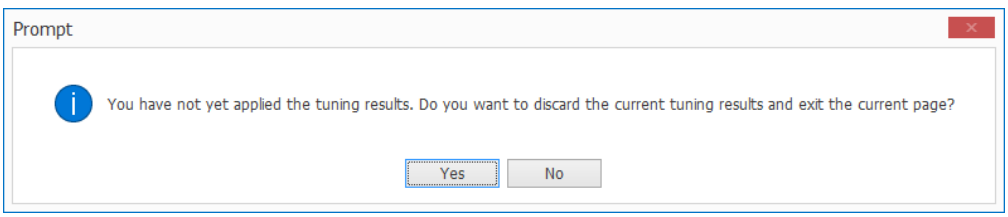


4. During auto-tuning, if you want to stop auto-tuning in advance, click "Stop tuning".



5. After auto-tuning is completed, the screen displays the auto-tuning results. Click "Apply results" to write the auto-tuning results to the corresponding tool. If you want to discard the auto-tuning results, click "Back" and then click "Yes" in the pop-up dialog.





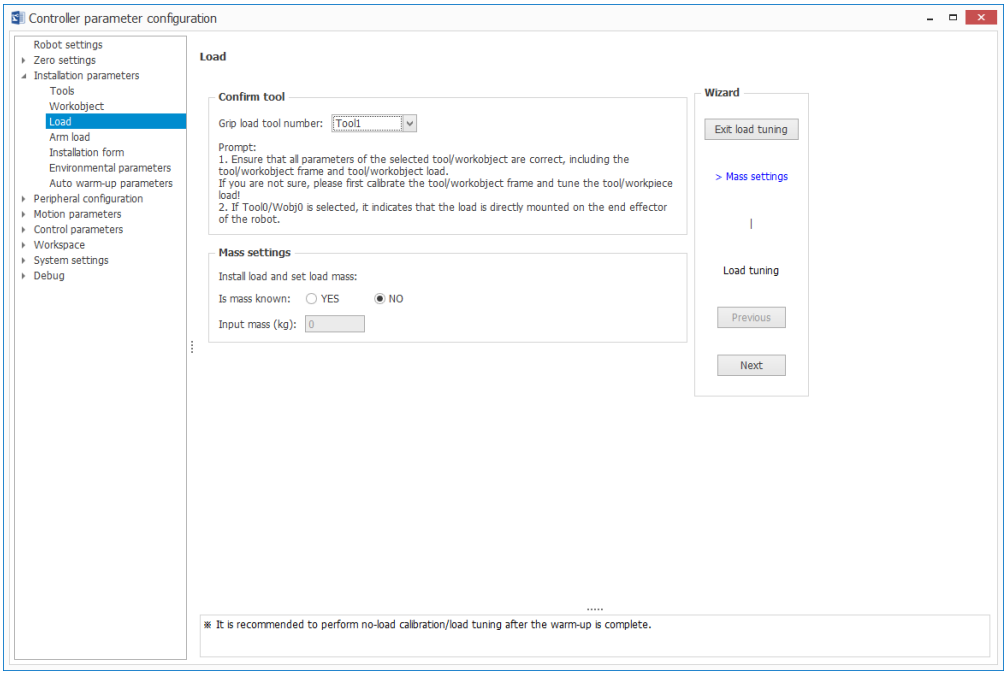
12.5.3.3 Grip Load Auto-Tuning

When performing grip load auto-tuning, you need to set accurate tool load frame and tool load parameters. The tuning result will only include grip load parameters.

1. Set the mass.

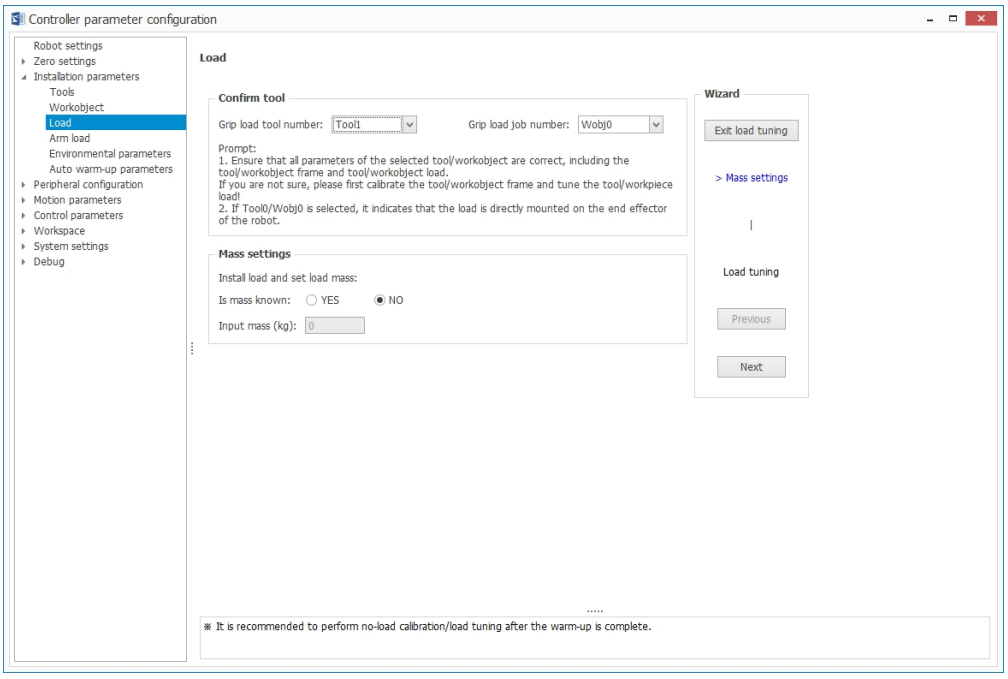
   TCP mode:

   In TCP mode (the selected tool number is in gripping state), workobject number is not required.



   ECP mode:

   In ECP mode (the selected tool number is in non-gripping state), the setting of workobject number is required. The tool number and workobject number must be used in pairs (to ensure mutual exclusivity between gripping and non-gripping states). If any settings are incorrect, you will not be able to click "Next" to proceed.

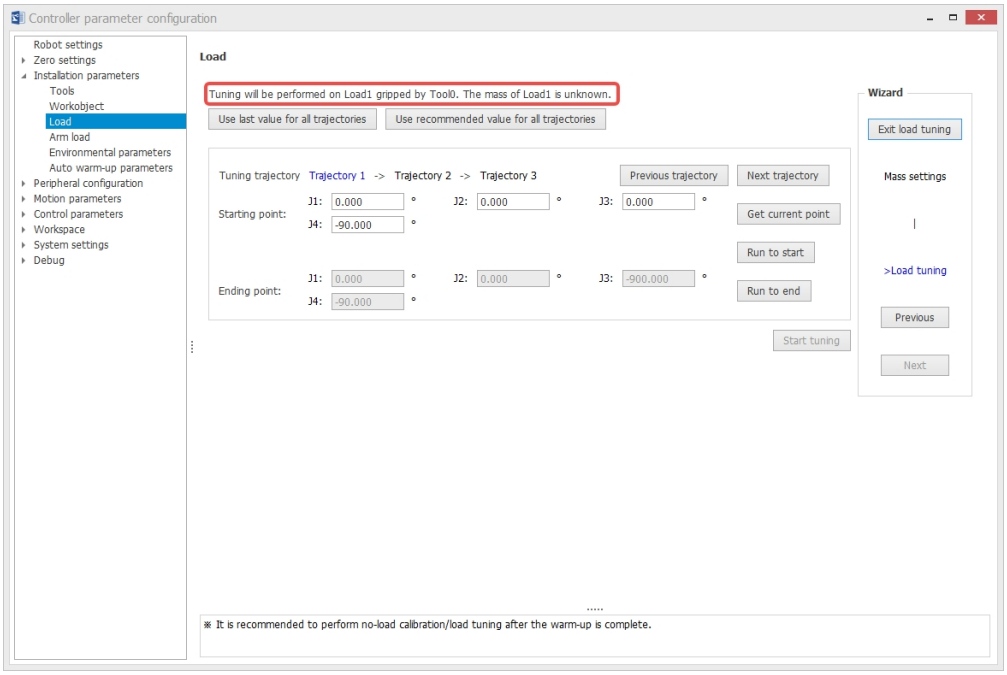


1) Select the number of the tool that grips the load. Ensure that the "TFrame" and "TLoad" parameters of the selected tool are correctly set. Otherwise, the results will be incorrect. If you select "Tool0", it means that the load is directly installed on the end flange.

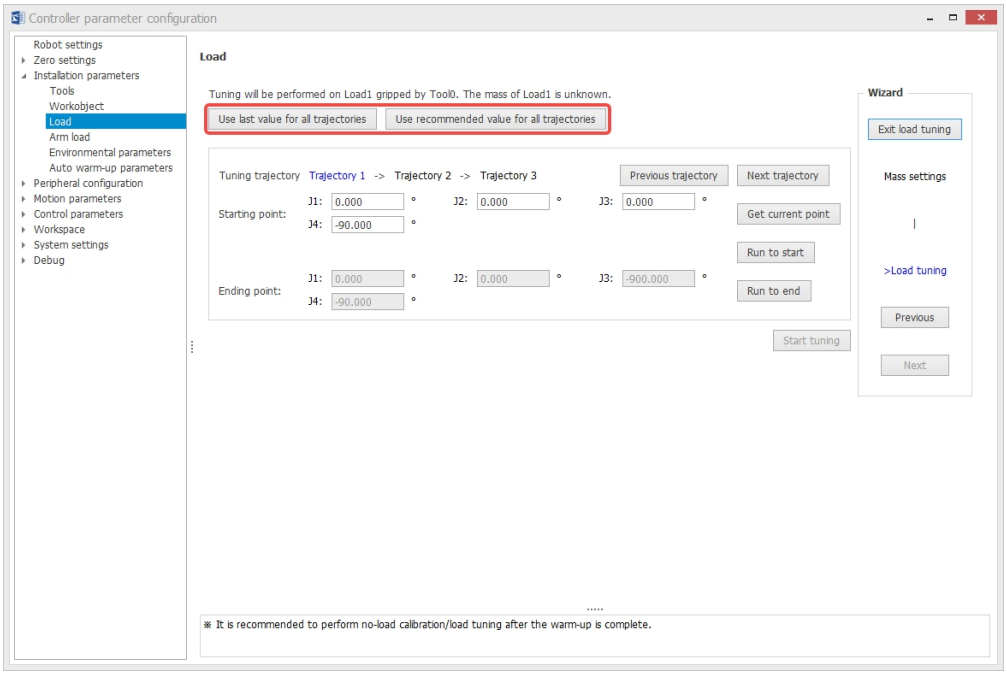
2) If the load mass is known, select "YES" and enter the mass. If the load mass is unknown, select "NO". Select "YES" and correctly set the load weight for more accuracy result. Click."Next"to enter the trajectory setting interface.

2. Set the auto-tuning trajectory.

1) Check that the load number and mass setting value for auto-tuning are correct.



2) If it is the first use of the load auto-tuning function or the robot installation method has changed, it is recommended to click "Use recommended value for all trajectories" to recalculate the recommended trajectory; in other cases, it is recommended to click "Use last value for all trajectories", which will default to using the last successfully auto-tuned trajectory.



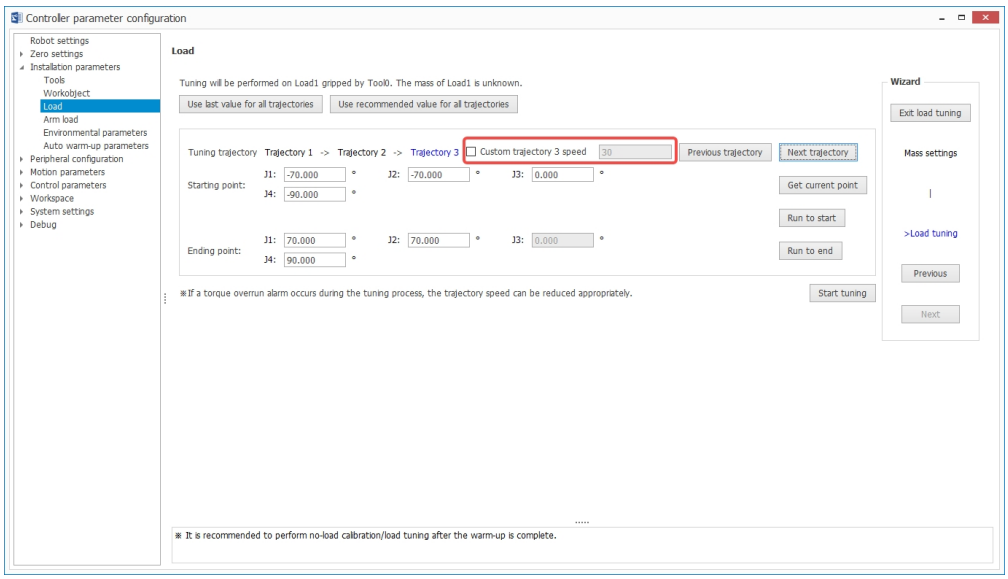
3) You need to set 3 auto-tuning trajectories: If they have been successfully auto-tuned before, the successful trajectories will be displayed by default; otherwise, the recommended values will be displayed.

SCARA robot trajectory requirements:

It is recommended to use the recommended values as much as possible. The joint angles in the edit box can only be manually modified in Admin/Manufacturer mode.

- Trajectory 1: For J1, J2 and J4, the joint angle at the start point is identical to that at the end point. For J3, the displacement from the start point to the end point is 50 mm.
- Trajectory 2: For J1 to J3, the joint angle at the start point is identical to that at the end point. For J4, the difference between the joint angles at start point and the end point should be as large as possible.
- Trajectory 3: For J3, the joint angle at the start point is identical to that at the end point. For J1, J2 and J4, the difference between the joint angles at start point and the end point should be as large as possible.

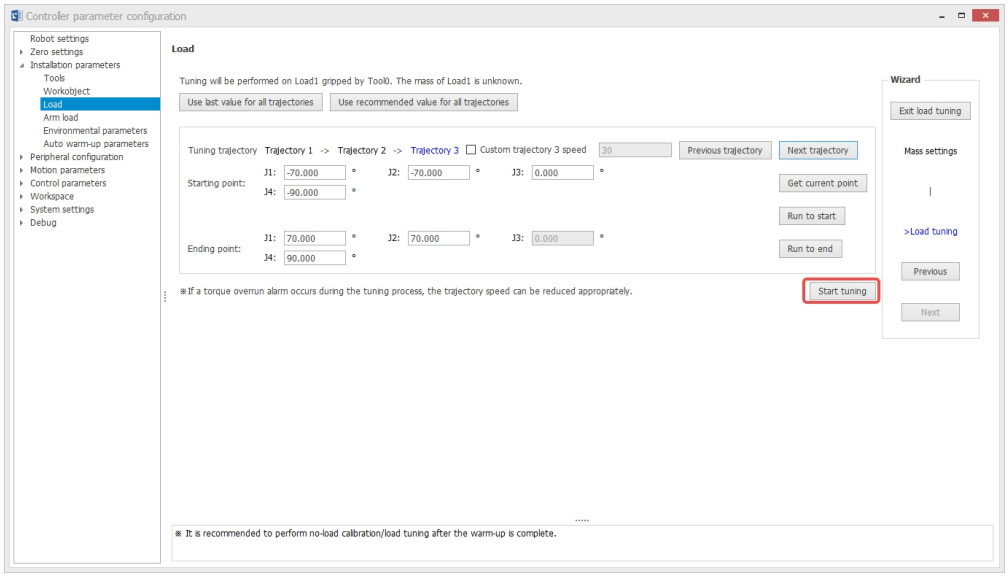
> **CAUTION**
>
> If a "Torque overlimit" alarm occurs during the auto-tuning process, return to the Trajectory 2 or Trajectory 3 settings interface, check "Custom trajectory speed", and appropriately reduce the speed value (30 by default). The adjustment principle is: Choose the highest possible operating speed without triggering a "Torque overlimit" alarm.



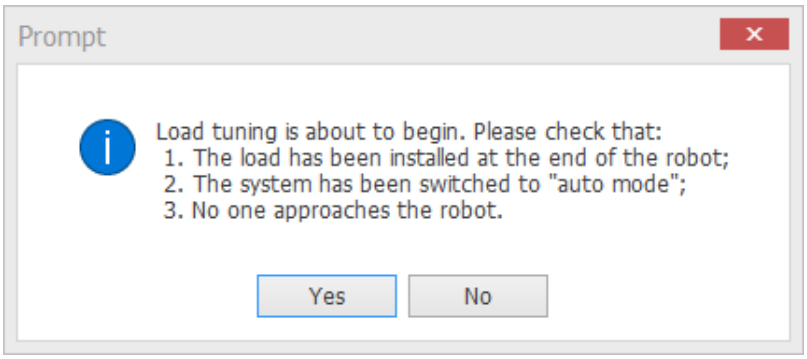
4) Check that the recommended trajectory is reachable. You can click "Run to start" and "Run to end" to see whether interference occurs on the current trajectory. If the trajectory is unreachable, adjust it manually according to the orientation requirements of the trajectory.

Click "Get current point", and the system will set the current joint angles as the start point of the current trajectory.

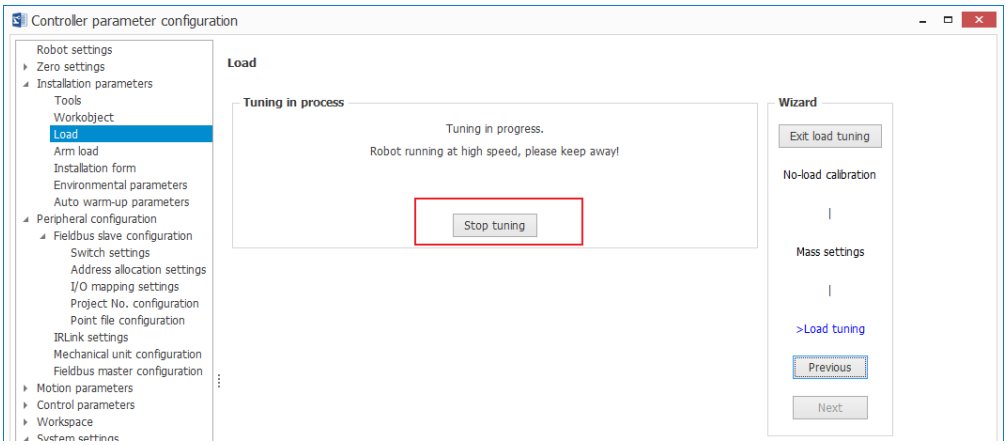
5) If you confirm that the trajectory is reachable, switch the robot operating mode to "Auto" and click "Start tuning".



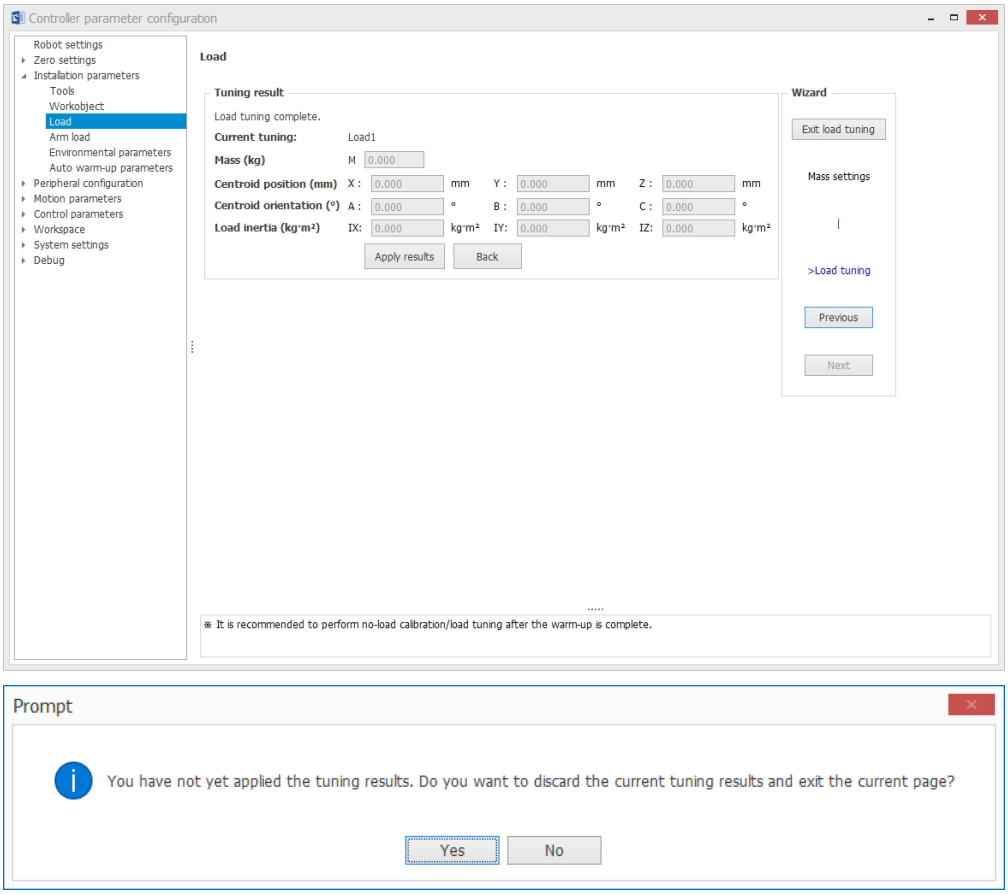
3. In the pop-up dialog, click "Yes" to start tuning if all conditions are met. Otherwise, click "No" to return to the setting page and click "Start tuning" again once all conditions are met.



4. During auto-tuning, if you want to stop auto-tuning in advance, click "Stop tuning".



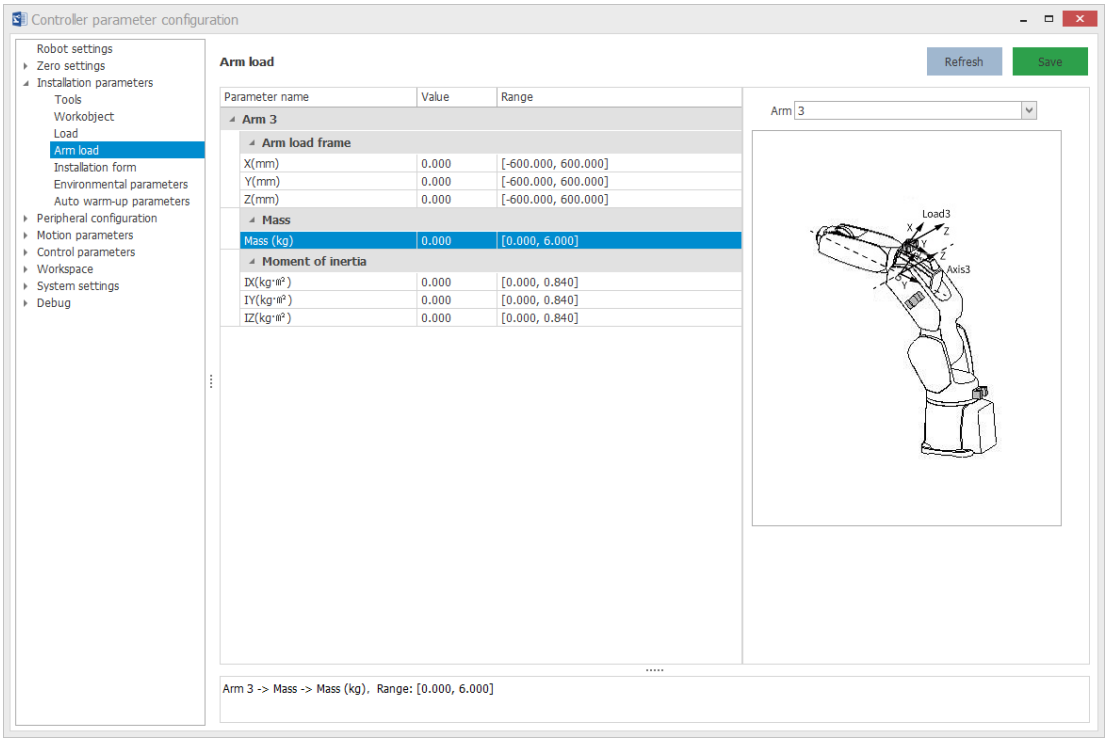
5. After auto-tuning is completed, the screen displays the auto-tuning results. Click "Apply results" to write the auto-tuning results to the corresponding load. If you want to discard the auto-tuning results, click "Back" and then click "Yes" in the pop-up dialog.



12.5.4 Notes

- Ensure that the zero point of the robot is set correctly before performing load auto-tuning.
- The accuracy of load auto-tuning is affected by the warm or cold state of the robot. It is recommended to perform load auto-tuning after the robot is warmed up and the robot temperature is stable. The auto-tuning in the cold state may result in poor auto-tuning accuracy.
- If the robot is a new robot, or is not warmed up, a collision detection alarm may occur. In this case, you can temporarily disable the collision detection function if the auto-tuning trajectory does not collide with the surrounding environment and people. After auto-tuning is completed, enable the collision detection function manually again.
- 6-axis robot trajectory requirements:
  - For Trajectory 1 and Trajectory 2, ensure the forearm and the axes of J3/J5 are as parallel to the ground as possible, otherwise the auto-tuning accuracy may be affected. It is recommended to use the recommended values first when conditions allow. If the recommended values are not reachable, it is recommended to change the angles of J1 and J2 at the current position of the robot, and use the "Get current point" function to calculate the recommended angles for Trajectory 1 and Trajectory 2. If none of the above conditions can be met, adjust according to the trajectory orientation requirements. The joint angles in the edit box can only be manually modified in Admin/Manufacturer mode.
  - When setting the auto-tuning trajectory, ensure that the angle difference between the start point and end point of Trajectory 2 for J6, and the angle difference between the start point and end point of Trajectory 3 for J5/J6 is large, preferably greater than 90°. If the angle difference is small, the auto-tuning error increases. It is recommended to use the default recommended values when conditions allow.

- If the robot arm Arm3 is equipped with a load on arm, correctly set the load on arm parameters or remove the load on arm from Arm3 before performing load auto-tuning.

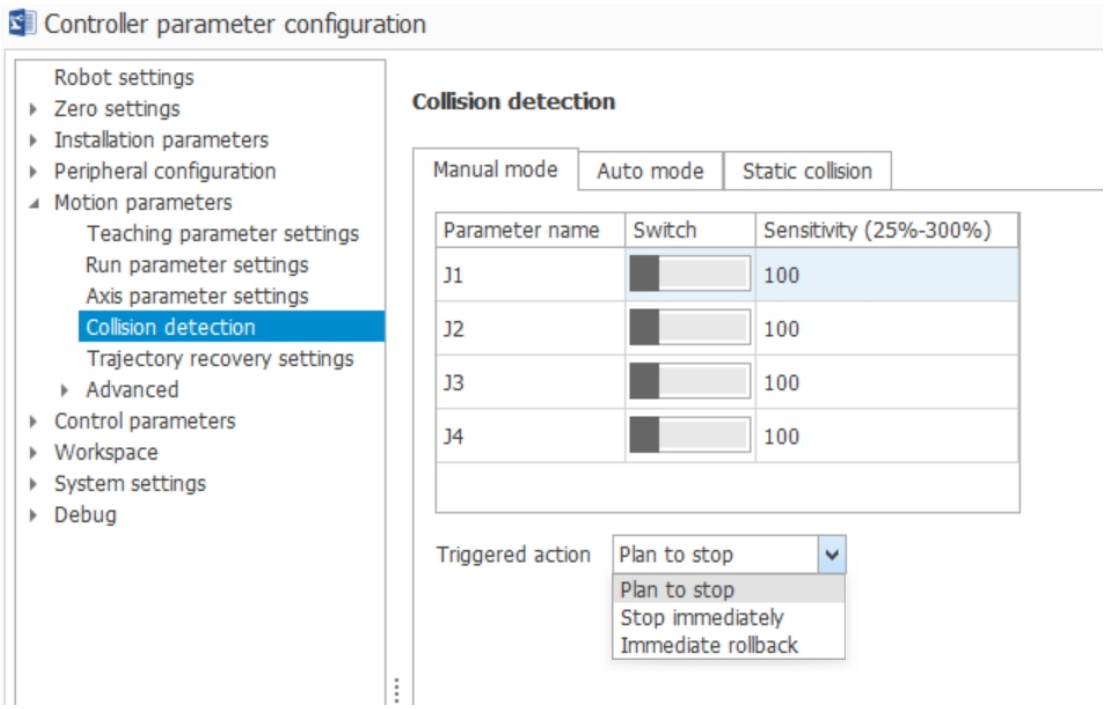


- SCARA robot trajectory requirements:
  - The joint angles in the edit box can only be manually modified in Admin/Manufacturer mode.
  - When setting the auto-tuning trajectory, ensure that the angle difference between the start point and end point of Trajectory 2 for J4, and the angle difference between the start point and end point of Trajectory 3 for J1/J2/J4 is large, preferably greater than 90°. If the angle difference is small, the auto-tuning error increases. It is recommended to use the default recommended values when conditions allow.
  - If significant shaking of the robot occurs during the auto-tuning process, first correctly identify the inertia ratio in the servo background.
  - The accuracy of the load auto-tuning results is related to the dynamics model. When the ratio of actual-to-rated load mass is greater than 20%, the load mass error is generally within 20%, and the center of mass error is generally within 25%.
- "Custom trajectory speed" should be used only when a "Torque overlimit" alarm occurs, and the value set should not be too small when in use.
- The center of mass of the load should be sufficiently far away from the flange. For loads with small eccentricity, the error in auto-tuning results is typically large.
- If the load mass is known, it is preferred to enter the mass.
- When performing grip load auto-tuning, you need to select the tool that grips the load. Ensure that the "TFrame" and "TLoad" parameters are correctly set. Otherwise, errors occur in the auto-tuning results.
- When load auto-tuning is performed, all tasks except static tasks must be stopped.
- If an emergency occurs during auto-tuning, press an emergency stop button to stop the motion.
- When auto-tuning of tools/workobjects are performed, all tasks will return to the start line except for clearing the motion trajectory.

12.6 Collision Response

12.6.1 Overview

After a collision detection alarm is triggered in the robot, you can select the collision response method, i.e., the action triggered by the collision detection alarm. The available trigger actions include: plan to stop, stop immediately, and immediate rollback. In addition, external signals are supported to trigger the collision detection alarm, which can effectively avoid collisions of the manipulator. For the settings of this function, see 9.5.4 Collision Detection.



12.6.2 Plan to Stop

12.6.2.1 Description

After selecting the trigger action as "Plan to stop", upon the occurrence of a collision detection alarm, the robot speed will be gradually planned to 0, until the robot stops.

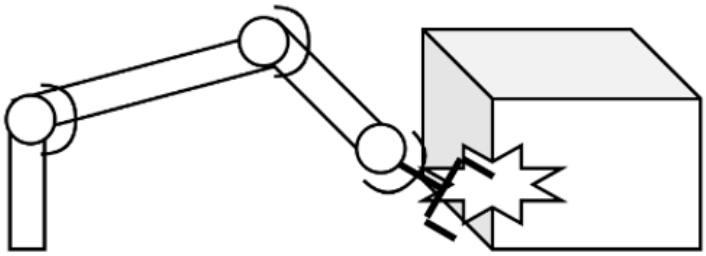
12.6.2.2 Steps

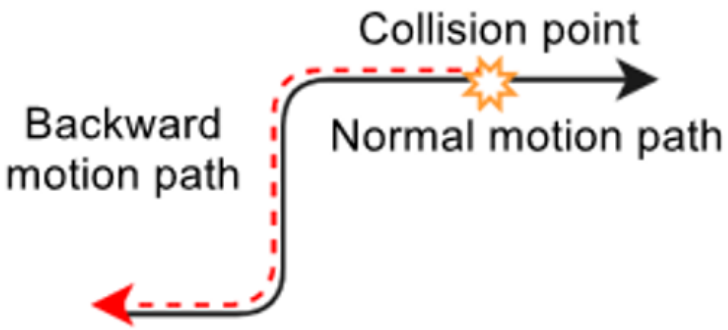
- In the manual mode, you can set the action triggered upon collision in the manual mode to "Plan to stop".
- In the auto mode, you can set the action triggered upon collision in the auto mode to "Plan to stop".
- In addition, in the auto mode, you can also enable the "Plan to stop" function by using instruction SetCollMode (ON,Coll_ PlanStop).

12.6.3 Immediate Rollback

12.6.3.1 Description

When the "Triggered action" is set to "Immediate rollback", upon the occurrence of a collision detection alarm, the robot will move back a certain distance along the planned path before the collision. Additionally, you can set the planning distance during moving back.





12.6.3.2 Steps

- In the manual mode, you can set the action triggered upon collision in the manual mode to "Immediate rollback", where you can also set the "Move back distance".
- In the auto mode, you can set the action triggered upon collision in the auto mode to "Immediate rollback", where you can also set the "Move back distance".
- In addition, in the automatic mode, you can also enable the immediate rollback function by using instruction SetCollMode (ON,Coll_Rebound).

12.6.3.3 Notes

- Before using the immediate rollback function, make sure that the collision detection function is enabled and the collision detection sensitivity is set correctly.
- The immediate rollback function can only ensure that the robot returns along the planned path before the collision. It cannot avoid secondary collision during moving back. Do not use the immediate rollback function in places with high risk of secondary collision, such as narrow space and complicated environment.
- The robot stops rollback immediately when any one of the following conditions is met: The rollback planning distance exceeds the planned value; The rollback planning distances exceeds the distance

planned at the time of 320 ms before alarm; There is a decreasing trend in the spatial distance between the planned rollback point and the collision point.

- Make sure the servo firmware version corresponds to the controller version.
- The move back function must be enabled manually.
- The move back function needs to be enabled respectively for the manual mode and auto mode.
- The instruction SetCollMode takes effect only in auto mode.
- For a robot that has performed the move back function, it is recommended to re-tune the collision detection sensitivity after using this function.
- The acceleration of move back is large and the impact on the robot joints is large. Avoid collision as much as possible. Use the recommended sensitivity to avoid false collision alarms.
- In the move back process after high-speed collision, the servo drive alarm is very likely to occur. This is normal as long as the servo drive alarm does not occur within the first 50 ms of the move back process.

12.6.4 Stop Immediately

12.6.4.1 Description

After selecting the trigger action as "Stop immediately", upon the occurrence of a collision detection alarm, the robot speed will be immediately decreased to 0, until the robot stops.

12.6.4.2 Steps

- In the manual mode, you can set the action triggered upon collision in the manual mode to "Stop immediately".
- In the auto mode, you can set the action triggered upon collision in the auto mode to "Stop immediately".
- In addition, in the auto mode, you can also enable the "Stop immediately" function by using instruction SetCollMode (ON,Coll_ RapidStop).

12.6.4.3 Notes

- Before using the immediate stop function, make sure that the collision detection function is enabled and the collision detection sensitivity is set correctly.
- Make sure the servo firmware version corresponds to the controller version.
- The move back function needs to be enabled respectively for the manual mode and auto mode.
- The instruction SetCollMode takes effect only in auto mode.
- For a robot that has performed the move back function, it is recommended to re-tune the collision detection sensitivity after using this function.

12.6.5 Priority of Actions Triggered by Instruction Settings and Interface Settings

- During running in manual mode, the triggered action after a collision detection alarm will be consistent with the "Triggered action" setting under "Manual mode" in the interface settings.

> **CAUTION**
>
> When running a program in manual mode, the trigger action set in the instruction settings will not take effect, and the trigger action will still follow the "Triggered action" setting under "Manual mode" in the interface settings. The same logic applies to "Move back distance".

- During running in auto mode, before the instruction settings take effect, the triggered action after a collision detection alarm will be consistent with the "Triggered action" setting under "Auto mode" in the interface settings. After the instruction settings take effect, the triggered action after a collision detection alarm will be consistent with the "Triggered action" in the instruction settings. After compiling the project or returning to the start line of the main task or all start lines, the "Triggered action" under auto mode will be consistent with the that in the interface settings. The same logic applies to "Move back distance".

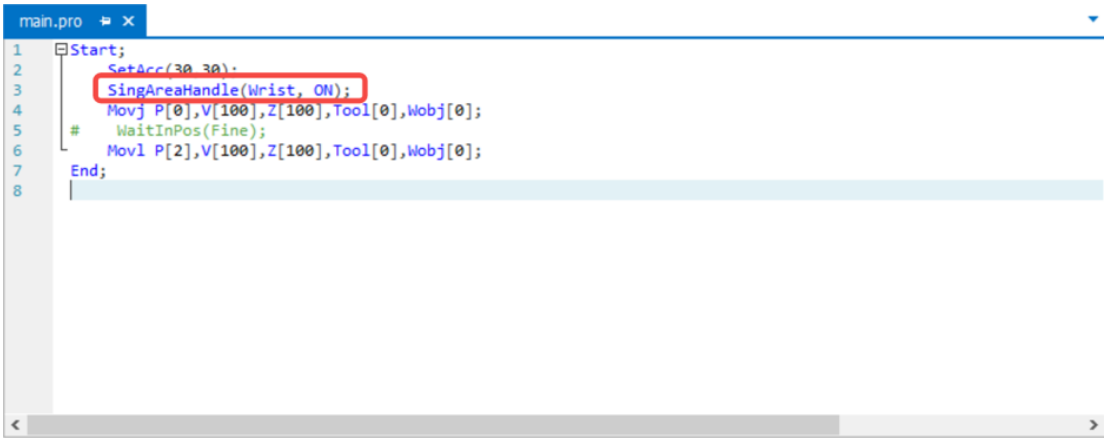
12.7 Singularity Crossing

A 6-axis industrial robot has singularities at its shoulder, elbow, and wrist respectively.

The shoulder singularity and elbow singularity can be avoided by changing the robot motion trajectory or changing the teaching points. However, it is difficult to avoid wrist singularity that may occur at any time with positive or negative changes in the J5 angle using the CP instruction (motion instructions in Cartesian space, such as: Linear, circular, JumpL, free curve, etc.). The singularity function solves such problems.This function only supports the 6-axis industrial robot to cross the wrist singular area.

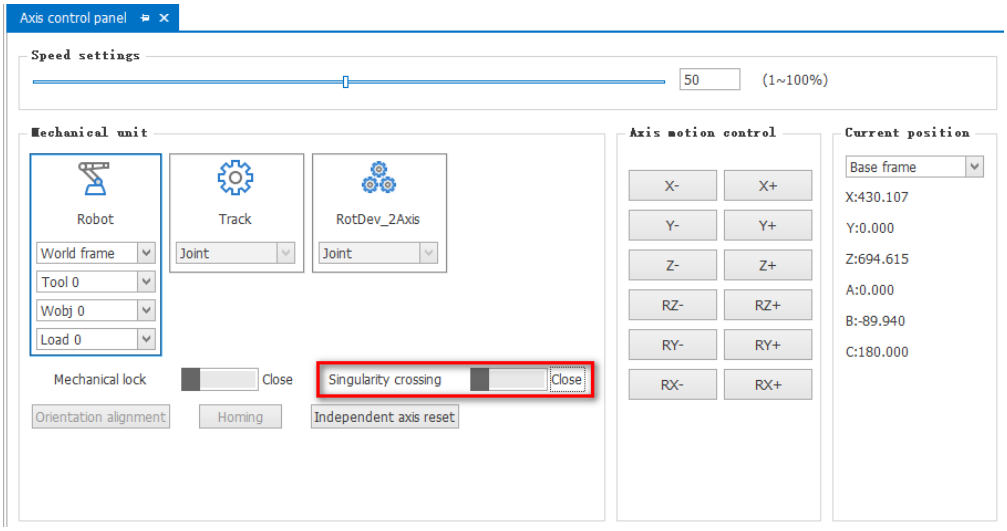
12.7.1 Steps

1. Enable/disable the singularity crossing function through instruction:



   In the above figure, "SingAreaHandle(Wrist, ON);" is the control instruction of the wrist singular traversing function. The instruction contains two parameters, "Wrist" is the default parameter and cannot be modified, the other parameter is "ON" or "OFF". "ON" indicates that the wrist singularity crossing function is enabled, and "OFF" indicates that the wrist singularity crossing function is disabled.

2. Enable/disable the singularity crossing function for jog motion:



In the above figure, the red box shows the singularity crossing function switch.

12.7.2 Notes

- After the singularity crossing function is enabled, it takes effect for the subsequent motion instructions, until it is disabled by a new instruction or the parameter is restored to default value (Wrist, OFF). The default value is restored in the following conditions:
  - The program has a file or point modification, which requires the controller to compile the project.
  - The program running pointer is return to the start line.
  - The program running pointer is set to another function.
  - The controller is powered on again.

When any of the above conditions occur, the parameter is restored to default value "Wrist, OFF"

- The transition instruction does not take effect between the motion instruction enabled with this function and the motion instruction that is not enabled with this function.
- After this function is enabled, a small portion of the trajectory cannot reach the orientation/joint angle of the target position set by the user.
- The trajectory of the wrist singularity crossing movement realized by instruction is different from that realized by jog motion.
- After this function is enabled, the orientation of TCP during operation is different from the orientation of TCP during user teaching.
- The wrist singularity crossing movement realized by instruction is independent of that realized by the jog motion.

The compatibility of functions is described in the following table:

Function	Function	Compatibility	Action if incompatible
Wrist singularity crossing in instruction	Motion I/O	Yes	-
	Pause/Start	Yes	-
	TCP	Yes	-

Function	Function	Compatibility	Action if incompatible
	ECP	No	Alarm occurs when the singular area is reached
	External axis	Only no-model external axes are supported	Alarm occurs when the singular area is reached
	Model-based external axes	Axis singularity crossing is not supported for model-enabled external axes.	Alarm occurs when the singular area is reached
	Collision detection	Yes	-
	Vibration suppression	No	Alarm occurs when the singular area is reached
	Trajectory recovery	Yes	-
	Move to point	No	Alarm occurs when the singular area is reached
	Point offset	Yes, supports solution selection based on the same arm parameters	-
	Dynamic tracking	No	Alarm occurs when the singular area is reached
Wrist singular crossing during teaching	Position teaching TCP	Yes	-
	Jog	No	Alarm occurs when the singular area is reached
	Coordinated external axis	No	Alarm occurs when the singular area is reached
	Position teaching ECP	No	Alarm occurs when the singular area is reached

12.8 Unordered Matching

Unordered matching (hereinafter referred to as "universality") means that the robot model parameters are stored at the robot controller and at the robot gyroscope 2.0 (hereinafter referred to as "manipulator"). When the controller is powered on, the robot model parameters are compared between the controller and the manipulator. When mismatch occurs, an alarm occurs (see section 16.2 Robot Alarms and Handling Method). You can synchronize the robot model parameters between the controller and the manipulator using the InoRobotLab software.

Use scenario: When the robot controller or manipulator is replaced, you can automatically import the robot model parameters using this function, without the need to calibrate and set the robot model parameters again.

12.8.1 Functions and Operating Steps

Function 1: Universality

This function checks the difference in the robot model parameters between the robot controller and the manipulator and achieves the mutual transmission of parameters between the controller and the manipulator. The following scenarios may occur.

- Scenario 1: The robot model stored on the controller is the same with that stored on the manipulator, but some parameters (for example, zero point and link length) are different.

After the InoRobotLab is connected, a pop-up box indicates "The robot model parameters are inconsistent between the manipulator and the controller. Do you want to sync robot model parameters from manipulator to controller?" and alarm 0xE10A occurs (see section 16.2 Robot Alarms and Handling Method), as shown in Figure 1. If you select "No", the window closes without synchronization of robot model parameters, and the alarm cannot be cleared. If you select "Yes", the robot model parameters will be synchronized between the controller and the manipulator, as shown in Figure 2. After the synchronization is complete, a prompt "Robot model parameter synchronization complete, please restart the controller!" appears, as shown in Figure 3.

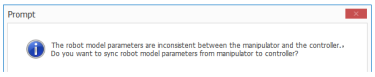


Figure 1

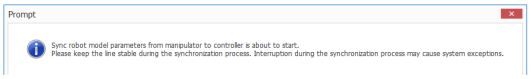


Figure 2

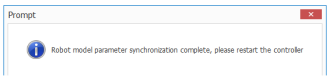


Figure 3

- Scenario 2: The robot model stored on the controller is different from that stored on the manipulator.

After the InoRobotLab is connected, a pop-up box indicates "The robot model parameters are inconsistent between the manipulator and the controller. Do you want to sync robot model parameters from manipulator to controller?" and alarm 0xE10A occurs (see section 16.2 Robot Alarms and Handling Method), as shown in Figure 4. If you select "No", the window closes without synchronization of robot model parameters, and the alarm cannot be cleared. If you select "Yes", the robot model parameters will be synchronized between the controller and the manipulator, as shown in Figure 5. After the synchronization is complete, a prompt "Robot model parameter synchronization complete, please restart the controller!" appears, as shown in Figure 6.

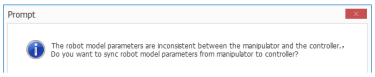


Figure 4



Figure 5

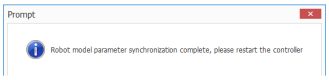


Figure 6

- Scenario 3: No robot model parameters are stored in the manipulator.

After the InoRobotLab is connected, a pop-up box indicates "The robot model parameters are inconsistent between the manipulator and the controller. Do you want to sync robot model parameters from manipulator to controller?" and alarm 0xE10A occurs (see section 16.2 Robot Alarms and Handling Method), as shown in Figure 7. If you select "No", the window closes without synchronization of robot model parameters, and the alarm can be cleared. If you select "Yes", the robot model parameters will be synchronized between the

controller and the manipulator, as shown in Figure 8. After the synchronization is complete, a prompt "Robot model parameter synchronization complete!" appears, as shown in Figure 9.

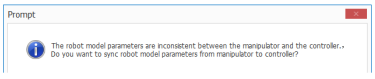


Figure 7

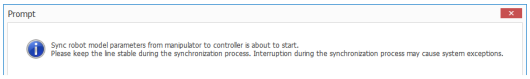


Figure 8

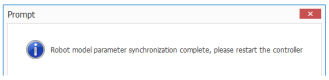


Figure 9

- Scenario 4: The power of the manipulator is inconsistent with the power of the controller. For example, the drive power is smaller than the servo power, the servo DB resistance is not between the upper and lower limits stored in the servo drive, and the brake configuration of the servo is inconsistent with the brake configuration stored in the manipulator.

After InoRobotLab is connected, the following information will be displayed in a pop-up box (as shown in Figures 10, 11 and 12), and alarm 0xE10B occurs (see section 16.2 Robot Alarms and Handling Method).

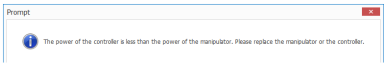


Figure 10 Inconsistent power



Figure 11 Inconsistent resistance

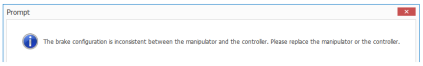


Figure 12 Inconsistent brake

- Scenario 5: The difference in the parameter version between the manipulator and the controller is too large (including parameter addition, parameter deletion, parameter name change, and parameter structure change).

After the InoRobotLab is connected, a pop-up dialog indicates "The controller software is not compatible with the manipulator parameters. Please disable the universality or upgrade the software.", as shown in Figure 13, and alarm 0xE109 occurs (see section 16.2 Robot Alarms and Handling Method).

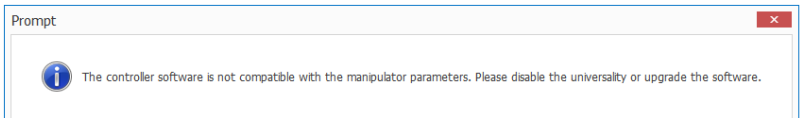


Figure 13

- Scenario 6: For some older models, the hardware conditions do not meet the universality requirements (e.g. the robot does not include a gyroscope, the gyroscope version is 1.0, the firmware version of the gyroscope 2.0 does not support universality, the controller FPGA brand is not ANLOGIC).

After the InoRobotLab is connected, a pop-up dialog indicates that "The controller universality function requires the hardware version of the manipulator memory to be at least 2.0 and the software version to be at least 2.1.", as shown in Figure 14. It only indicates that the hardware conditions do not meet the requirements for universality configuration. The robot can still be used normally.

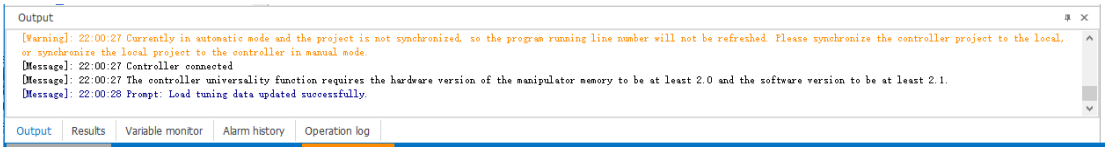


Figure 14

Function point 2: Universality switch

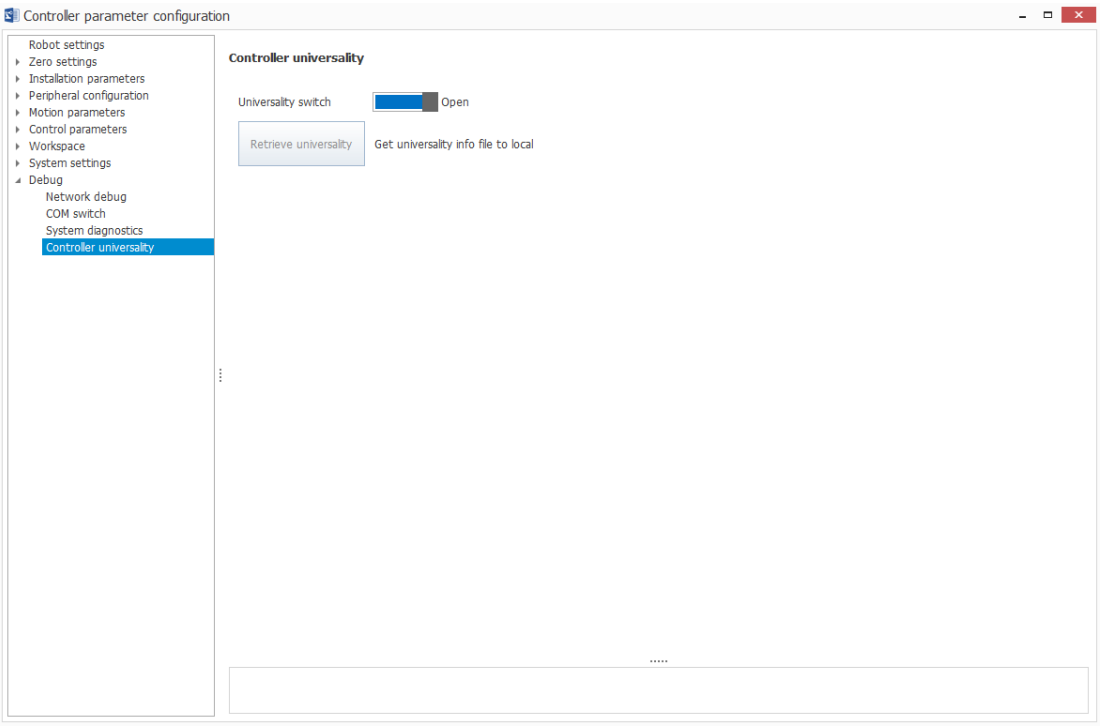


Figure 15

When the controller is inconsistent with the manipulator in parameters or in robot model, or the controller software is not compatible with the manipulator parameters, you can close the pop-up dialog, but the alarms cannot be cleared. In this case, you can log in to the InoRobotLab as the manufacturer and go to "Controller parameter configuration" > "Debug" > "Controller universality" > "Universality switch", as shown in Figure 15. After the universality switch is turned off, the parameters modified in the controller parameter configuration page are not written to the robot manipulator. The parameters in the robot manipulator will still be modified by the functions of flashing, upgrade, and importing robot model parameters. After the controller is restarted, the universality switch is turned on by default.

Function point 3: Retrieve universality

Log in to the InoRobotLab as the manufacturer and go to "Controller parameter configuration" > "Debug" > "Controller universality" > "Retrieve universality", as shown in Figure 15. You can export the universality information of the controller and the manipulator.

12.8.2 Notes

When performing the parameter synchronization, if the manipulator does not store any parameters, or the manipulator stores parameters but the parameters are not synchronized for the first time, you need to enter the password before you can continue. If the password is entered incorrectly more than 3 times, or the password dialog is closed abnormally more than 3 times, the pop-up dialog will not appear anymore and you will not be able to operate the robot and write parameters into the robot.



Manufacturer

When the controller and the manipulator are inconsistent in power or brake configuration, the universality switch cannot be turned off. You will be prompted that the operation is invalid and the alarm cannot be cleared The prompt includes "Not allowed to disable the universality as the controller power is smaller than the manipulator power.", or "Not allowed to disable the universality as the manipulator differs from the controller in brake configuration.", or "Not allowed to disable the universality as the manipulator differs from the controller in power (DB resistance)."

12.9 Soft Float Function

12.9.1 Overview

When the soft float function enabled, the robot changes from a rigid body to a compliant body. This enables the robot to change its position and orientation based on changes in external forces when the robot is subjected to external forces, preventing the robot from being damaged.

The soft float function includes Cartesian soft float function and joint soft float function.

- With the Cartesian soft float function, the robot becomes compliant relative to the Cartesian frame. It can be applied to processes such as die-casting parts pick-up.
- With the joint soft float function, the robot becomes compliant relative to the joint frame.

12.9.2 Steps

For a detailed description of the soft float function, see the "Soft Float-Related Instructions" in "Industrial Robot Instruction Guide".

12.9.2.1 Enabling the Soft Float Function

You can enable the soft float function using the SoftModeAct(SoftNum) instruction. The system includes 16 sets of soft float function settings, which can be called directly through the "SoftNum" parameter and the system variables. After this instruction is executed, the robot switches from position control mode to soft float control mode. This instruction is a blocking type instruction.

12.9.2.2 Disabling the Soft Float Function

You can disable the soft float function using the SoftModeDeact instruction. After this instruction is executed, the robot switches from soft float control mode to position control mode. When the SoftModeAct instruction

is enabled, this instruction is a blocking type instruction. This instruction can be used independently. If it is called when the SoftModeAct instruction is not active, the robot functions will not be changed.

12.9.2.3 Soft Follow Function

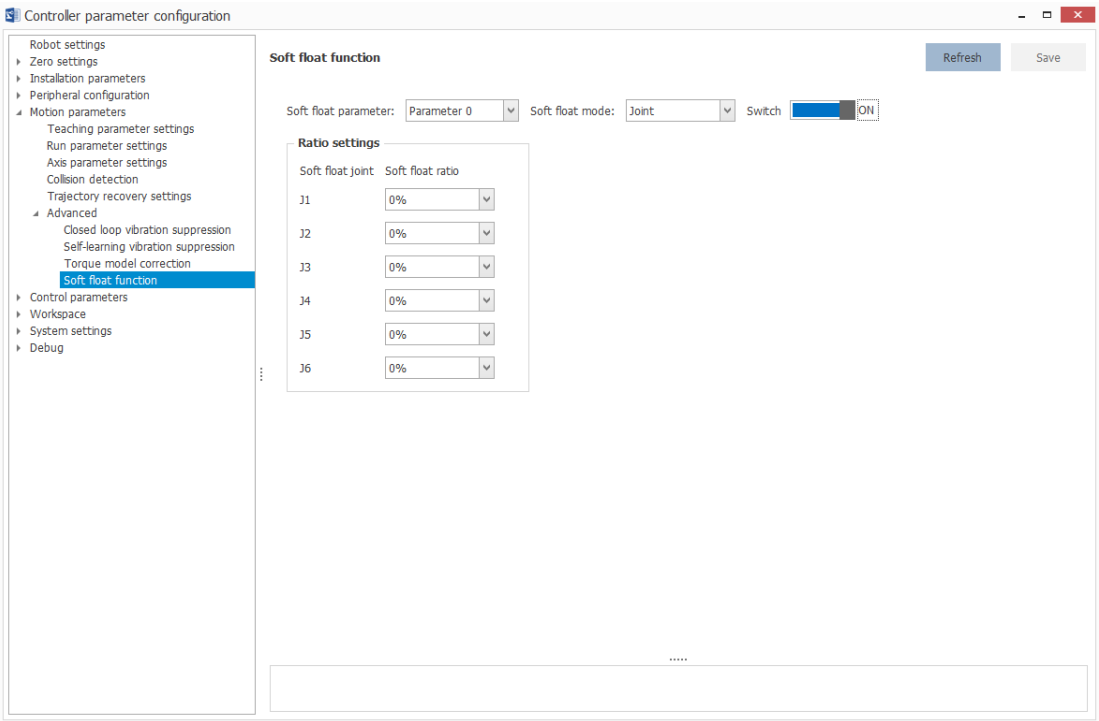
You can enable the soft follow function using the SoftFollow instruction. After the soft float function is enabled, if the robot is pushed and no external force is applied, it will return to the teaching point. However, when this instruction is executed, the current position is regarded as the teaching point, and the robot will not return to the original teaching point even after the external force is removed. When the SoftModeAct instruction is enabled, this instruction is a blocking type instruction. This instruction can be used independently. If it is called when the SoftModeAct instruction is not active, the robot functions will not be changed.

12.9.2.4 Soft Float Function Parameter Configuration

The configuration interface of the soft float function is as follows. The soft float function is only supported by 6-axis robots including the models of IR-R4-56S5-A1L_01740496, IR-R35-185S7-B1NH_01740676, IR-R35-185S7-B1ZD_01740679, IR-R60-210S5-E1N_01740415, IR-R80-210S5-E1ZY_01740464, IR-R80-210S10-E1N_01740465, IR-R170-270S5-E1N_01740669 and IR-R220-270S5-E1N_01740513. You can configure the soft float parameters on the soft float function interface, as shown below:

InoRobotLab interface

Go to "Controller parameter configuration" > "Motion parameters" > "Advanced" > "Soft float function".



Description

- On the configuration interface of the soft float condition parameters, you can configure 16 sets of condition parameters. The condition parameters include soft float parameter, soft float mode, soft float switch, reference frame, soft float direction, soft float ratio, direction of movement, and range value. The parameters can only be modified in the manual mode.
- Soft floating parameter: Indicates the condition parameter number, which ranges from 0 to 15.

- Soft float mode: Indicates the soft float mode, including joint mode and Cartesian mode.
- Switch: You can decide whether to activate each set of parameters. Only the activated set of parameters can be called in the instruction.
- Reference frame: This parameter is available only when the soft float mode is set to "Cartesian". The reference frame includes world frame, tool frame, and workobject frame. The default value is "World frame".
- Soft float direction:
  - For the soft float relative to the joint frame, you can custom the joints including J1 to J6.
  - For the soft float relative to the Cartesian frame, you can custom the Cartesian parameters including X direction, Y direction, Z direction, X rotation (rotation around X axis), Y rotation (rotation around Y axis), and Z rotation (rotation around Z direction).
- Soft float ratio: Indicates the degree of compliance in the respective directions or axes. For soft float relative to the Cartesian frame, the soft float ratio includes 0% and 100%. For the soft float relative to the joint frame, the soft float ratio includes 0%, 1%, 25%, 50%, 75%, and 100%, which can be selected by the drop-down box. 0% indicates that soft float function is disabled. The larger the value, the more compliant the robotic arm. The value of soft float ratio in the respective directions and axes can be adjusted independently. The default value is 0%.
- Direction of movement: Indicates the direction of the soft float motion relative to the Cartesian frame. It includes X, Y and Z directions.
- Range value: This parameter is available only when the soft float mode is set to "Cartesian". It indicates the maximum amount of movement the robot is allowed to perform. The range value varies depending on the robot model. For example:
  - For IR-R60-210S5-E1N_01740415 robot, the range is 0-500 mm, and the default value is 200 mm.
  - For IR-R4-56S5-A1L_01740496 robot, the range is 0-200 mm, and the default value is 100 mm.

12.9.3 Notes

When using the soft float function, pay attention to the following.

- The soft float function can be enabled only by the instruction.
- The soft float function will be automatically disabled by:
  - Executing the instruction of disabling the soft float function.
  - Starting to execute a program.
  - Disabling the robot (normal disable, disable upon alarm, disable upon emergency stop).
  - Running the program to the end.
  - Powering on/off the controller.
  - Switching the project, changing the program cursor line number, returning the program to the start line, or setting the start line.
  - Performing homing, orientation calibration, point to point motion, and jog motion.
- When enabling or disabling the soft float function, make sure that no external force is applied to the robot.
- The soft float function is incompatible with the following robot functions: Collision detection function, trajectory accuracy function, optimal trajectory function, trajectory recovery function, independent axis

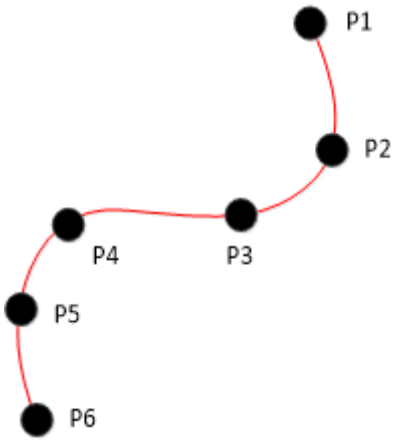
function, dynamic tracking function, self-learning vibration suppression function, and load auto-tuning function. The current soft float function is only supported by the floor-mounted robot.

- When the soft float function is enabled, after the robot is pushed beyond the set safety alarm threshold, the robot reports an alarm message and stops.
- Before enabling the soft float relative to the Cartesian frame, move the robot and make sure the target of movement is different from the current position of the robot.
- After the soft float function is enabled, if you click the Stop or Exit button on InoRobotLab or teach pendant, the robot stops and the soft float function remains active. If you click the Start button when the robot is still moving, the start action does not take effect.
- It is not recommended to use the soft float relative to the Cartesian frame near the singular position. Otherwise, the robot motion may become not compliant.
- When the robot runs in single steps, the soft float instruction cannot be executed.

12.10 Free Curve

12.10.1 Overview

The free curve function is applicable to motion with a large number of path points. Free curves are used to fit the sequence of spatial points input by the user, allowing the robot to perform curve motion based on the input sequence of spatial points. The free curve function supports two forms of instruction. You can use an independent Movs instruction in the program to implement a curve motion, or use the Curve instruction to implement a combination of single/multiple curve segments.



No.	Specifications
1	Number of input points: 4 to 500
2	Maximum spacing of input points: 100 mm
3	Minimum spacing of input points: 0.5 mm
4	Maximum orientation spacing of input points: 60°
5	Maximum curve length: 10,000 mm

12.10.2 Steps

For details of the Movs and Curve instructions, see the "Movs" and "Curve" sections in "Industrial Robot Instruction Guide".

12.10.3 Configuration of Curve Parameters

To adapt to the specific application conditions, you can configure "CurveParm" parameter in the curve instruction to adjust the trajectory properties of the curve. The "CurveParm" parameter includes the position stress coefficient, orientation stress coefficient, curve mode, basic error, corner determination threshold, smoothing error rate, and filter level.

CurveParm: Curve property tuning parameters

```
{
    int PosPress;              # Position stress coefficient, default value 10, range 1 to 100
   int OrientPress;            # Orientation stress coefficient, default value 10, range 1 to 100
   int CurveMode;      # Curve mode, 0 default mode, 1 smoothing mode, 2 precision mode
   double PosErr;      # Basic position error, default value 1 mm, range 0.1 mm to 100 mm
double OrientErr;   # Basic orientation error, default value 1°, range 0.1° to 10°
double CorneThre:      # Corner determination threshold, default value 90°, range 45° to 135°
double SmoErrRate;   # Corner smoothing error rate, default value 10, range 1 to 100
int TrajRecord;             # Trajectory record switch, default value 0 (OFF), range 0/1
int FilterLevel;            # Filter level, default value 0, range 0/1/2/3
}
```

PosPress: Position stress coefficient

- Data type: integer (supports variables)
- Default: 10
- Range: 1 to 100
- Parameter description: When the robot moves in a curve through a position corner, the running speed is automatically limited due to the position stress limit. Based on actual application requirements, you can adjust the speed limit of the robot when it passes the position corner by adjusting the position stress coefficient. When the position stress coefficient increases, the robot running speed near the position corner increases, but the motion shock or vibration becomes obvious. When the position stress coefficient decreases, the robot running speed decreases near the position corner, and the motion shock or vibration decreases. Avoid specifying an excessively large position stress coefficient to avoid severe shock and vibration or triggering an excessive acceleration alarm.

OrientPress: Orientation stress coefficient

- Data type: integer (supports variables)
- Default: 10
- Range: 1 to 100

- Parameter description: When the robot moves in a curve through an orientation corner, the running speed is automatically limited due to the orientation stress limit. Based on actual application requirements, you can adjust the speed limit of the robot when it passes the orientation position by adjusting the orientation stress coefficient. When the orientation stress coefficient increases, the robot running speed near the orientation corner increases, but the motion shock or vibration becomes obvious. When the orientation stress coefficient decreases, the robot running speed decreases near the orientation corner, and the motion shock or vibration decreases. Avoid specifying an excessively large orientation stress coefficient to avoid severe shock and vibration or triggering an excessive acceleration alarm.

CurveMode: Curve mode setting

- Data type: integer (supports variables)
- Default: 0
- Value range: 0 default mode, 1 smoothing mode, 2 precision mode
- Parameter description: You can select different modes based on the application scenario. The default mode is applicable to most operating conditions and can tune the curve according to default or user input curve parameters. The smoothing mode is applicable to operating conditions with dense input data, high acquisition noise, and low path accuracy requirements. In this mode, the curve does not pass strictly through the input points, but moves in the form of approaching the target points. The proximity of the motion is related to the density of the points, and the trajectory error does not exceed 50% of the spacing of neighboring points. This mode provides better smoothness. The precision mode is applicable to operating conditions with accurate input data, low noise, smooth original trajectory, and strict requirements for reaching the path points. The global trajectory error in this mode does not exceed the basic trajectory error, and the accuracy is not affected by the smoothing error rate.



PosErr: Basic position error

- Data type: Float (supports variables)
- Default: 1 (mm)
- Range: 0.1 to 100
- Parameter description: Because the position of the curve does not always pass the input point, there may be a theoretical position approximation error between the curve path and the input point. The allowable position error between the curve path and the target point is generally referred to as the basic position error. Generally, the default basic position error of the curve is 1 mm. This parameter is used as the reference value for curve fitting. In applications with high precision requirements, you can reduce the position error of the curve by adjusting this parameter. Note that when the maximum position error allowed for the curve decreases, the smoothness of the curve path generated by the robot decreases. This may cause the robot to stall or generate shock and vibration.

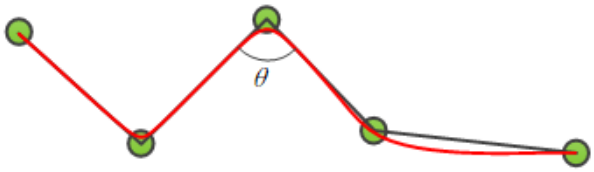
OrientErr: Basic orientation error

- Data type: Float (supports variables)
- Default: 1 (°)
- Range: 0.1 to 10

- Parameter description: Because the orientation of the curve does not always pass the input point, there may be a theoretical position approximation error between the curve path and the input point. The allowable orientation error between the curve path and the target point is generally referred to as the basic orientation error. Generally, the curve allows a 1° orientation error by default. This parameter is used as the reference value for curve data processing. In applications with high precision requirements, you can reduce the orientation error of the curve by adjusting this parameter. Note that when the basic orientation error of the curve decreases, the smoothness of the curve path generated by the robot decreases. This may cause the robot to stall or generate shock and vibration.

CorneThre: Corner determination threshold

- Data type: Float (supports variables)
- Default: 90 (°)
- Range: 45° to 135°
- Parameter description: When there are obvious corner features in the curve path, the smoothness of the path is greatly affected. This can easily lead to local sharpness or twisted curves that may cause interference with the environment, resulting in impacts or significant speed reduction. You can adjust the smoothing level of the curve by adjusting the corner determination threshold (CorneThre) and smoothing error rate (SmoErrRate), to smooth the corner features in the trajectory. The corner determination threshold is used to set the maximum path angle that allows for smoothing. Path points smaller than this angle will be determined as corners. In particular, points with a spacing less than 5 mm between them will not be included in the corner determination.

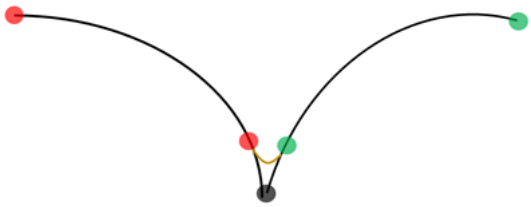


SmoErrRate: Smoothing error rate

- Data type: Float (supports variables)
- Default: 10
- Range: 1 to 100
- Parameter description: When the corner features of the trajectory are smoothed, the smoothness of the trajectory will be improved, but the accuracy of the trajectory at the corners may be affected. Therefore, it is necessary to limit the smoothing error. The corner determination threshold determines area on the curve that need to be smoothed, while the smoothing error rate determines the degree of smoothing. This parameter represents the rate of the maximum allowable error at the smoothed points relative to the set position and orientation error. Note that the lager the smoothing error rate the better the smoothing effect of the curve, but the arrival accuracy at the corners will be reduced. The default value of the rate is 10. You need to adjust the rate according to practical conditions.



Smoothing error = Basic error * Smoothing error rate

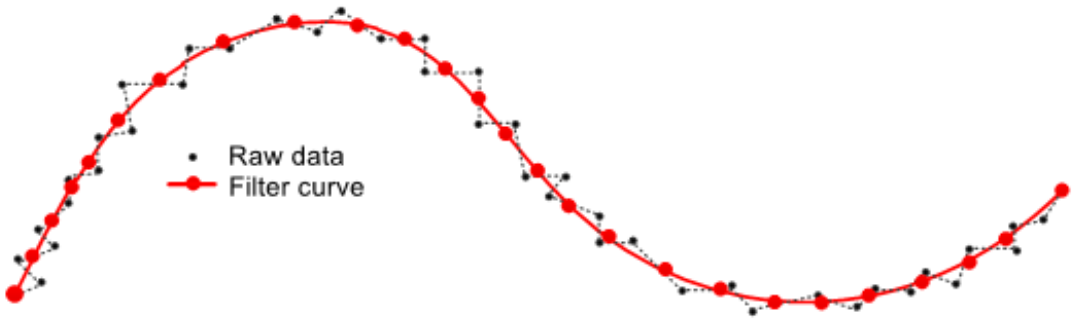


TrajRecord: TCP trajectory record switch (not valid in the current version)

- Data type: integer (supports variables)
- Default: 0
- Range: 0/1
- Parameter description: It is used to identify whether to record the current segment of the trajectory. 0 for no, and 1 for yes.

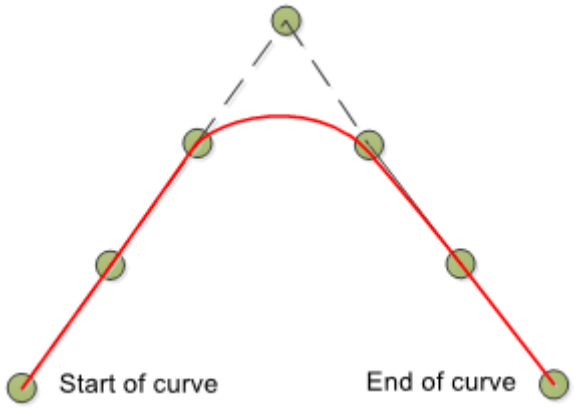
FilterLevel: Filter level

- Data type: integer (supports variables)
- Default: 0 (no filtering)
- Range: 0 to 3
- Parameter description: For applications where the accuracy of the path identification sensor is low, the curve motion may become stuck and have poor data collection accuracy and severe jitter. You can adjust the filter level to trim the raw data to filter out high-frequency noise spikes, and improve the smoothness of the curve. Note: The default value is 0, indicating that filtering is not performed. After the filter is turned on, the system filters the raw data to remove jitter noise. It is recommended to turn on the filter in situations with dense data and turn off the filter in situations with sparse data. In particular, filter processing will not be performed on points with a spacing greater than 2 mm or 2°.

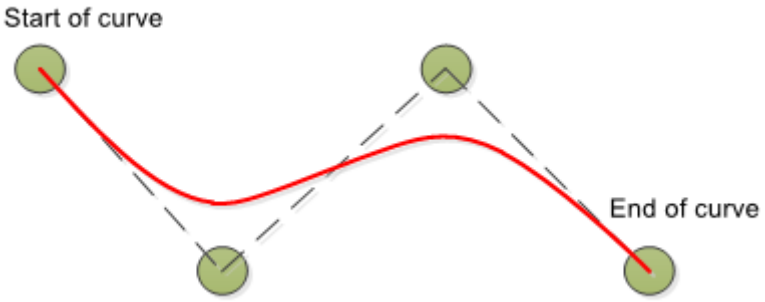


12.10.4 Notes

- It is not allowed to change the tool and workobject during the same curve motion.
- In the default mode, you can adjust the smoothing level of the trajectory by adjusting the corner determination threshold and smoothing error rate, to smooth the corner features in the trajectory to improve the passability. Note that the trajectory smoothness increases as the smoothing error rate increases, but the trajectory accuracy at the corners decreases accordingly. The curve will not pass through the original input points. Therefore, the smoothing coefficient needs to be adjusted considering both the precision and smoothness requirements.

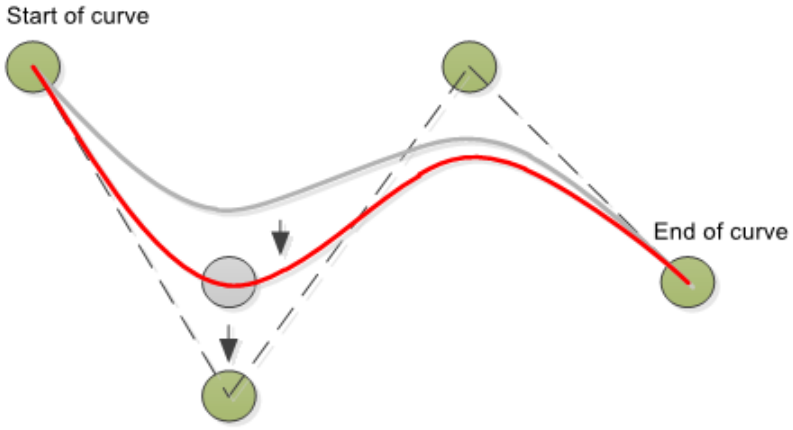


- In the default mode, after the smoothing error rate is configured, the robot approaches the corners but does not pass the points at corners when multiple teaching points form consecutive corners.

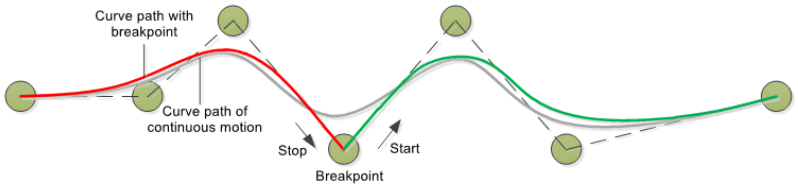


As the path errors at the corners cannot be guaranteed, avoid input points with excessive spacing. The recommended spacing of input points is 1 mm to 10 mm. In applications with high precision requirements or interference risk, reduce the spacing of input points and check whether the robot motion path interferes with surrounding objects at low speed.

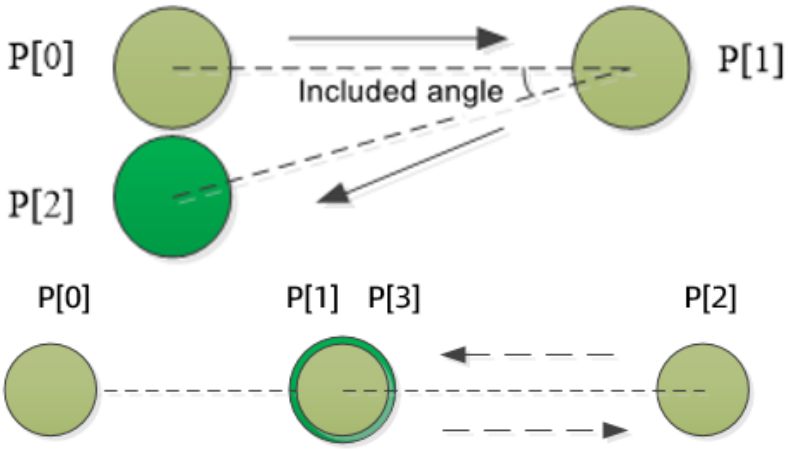
- When using the free curve function, it is generally required that all data points be described with the correct arm parameters. If you want the robot to automatically rotate in the shortest path, use the SlewMode instruction to switch the current rotation mode to continuous rotation mode before performing curve motion. For details, see the description of the SlewMode.
- If the spacing of the two adjacent input points is smaller than the minimum specification and smaller than the trajectory accuracy setting in the curve parameter, the previous input point bit will be ignored. In particular, if a previous point is bound to a motion I/O, the motion I/O will be transferred to a next point. If the previous point is the start point, the start point will be retained. The next point will be ignored and its I/O will be transferred to the start point. If the total number of I/Os of the next point exceeds the limits, an alarm occurs.
- The path of any point on the curve is affected by the pose and distribution of multiple input points before and after that point. When one of the input points is changed, the curve path of multiple points before and after that point changes.



- When a breakpoint debugging is performed on a curve group, the robot ends the motion before entering the breakpoint and stops at the target position. Note that the motion path of the robot near the stop point will be different from the curve path during continuous teaching or automatic running. In particular, if the MovS motion delimited by a single-step operation or a breakpoint contains only one point, the robot performs a MovL motion instead of the MovS motion.

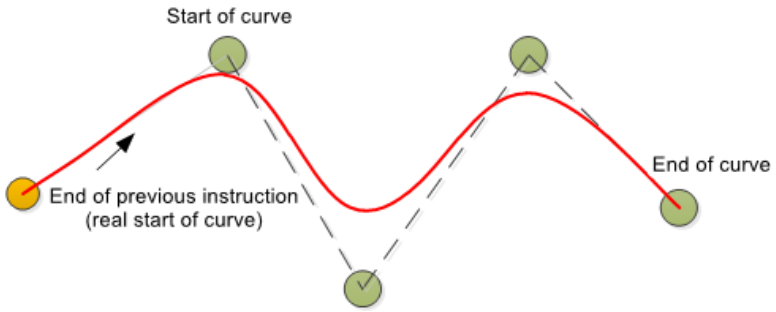


- When the input points are close to a robot workspace boundary or a singular area, the curve path constituted by the input points may trigger an alarm even if the input points are inside the robot workspace. Adjust the working area or the position of the input points properly to avoid going beyond the workspace or entering the singular area.
- The MovS instruction supports up to three motion I/Os associated with the input points (up to three motion I/Os per point). The I/O action is triggered automatically when the robot moves near the input point.
- As the reciprocating line cannot be approximated through the free curve, the robot alarm may be triggered when the angle of two motion paths formed by three consecutive input points is less than 0.5°.



- When you specify the motion speed of the curve through the MovS instruction, be aware that the robot is limited by motion performance and thus requires a certain distance for acceleration and deceleration when changing the speed. When the distance the robot moves is smaller than the distance required for the robot to change speed or multiple different desired speeds are specified within the distance for speed change, the robot may not reach the desired speeds.

- When a free curve is defined as a curve group, the robot calculates the transition length of the curve path based on the transition level of the last target teach point in the curve group. Transition level settings of input points other than the end point are ignored.
- During curve transition, the maximum allowable transition length of the curve path cannot exceed half the curve length and cannot exceed the total length of the curve segment consisting of the first or last four points.
- The free curve starts with the current position of the robot as the true start point of the curve, and always approaches the nominal start point in the instruction by a motion approaching a straight line. However, there may be a large gap or bend angle between the current point and the nominal start point of the curve segment to be executed. In particular, when the curve instruction is set as the start line, an error may occur because the motion planning conditions between the current point and the executed curve segment are not meet, for example, the arm parameters are different or a reciprocating motion is caused. In addition, the nominal start point is always treated as an inflection point. Therefore, the arrival accuracy near it is affected. It is recommended to move to the nominal start point of the curve through other motion instructions before executing the curve instruction.



The compatibility of functions is described in the following table:

No.	Function specifications
1	Manual (single-step debugging, breakpoint debugging, etc.) and automatic modes supported
2	Motion I/O supported
3	Dynamic tracking supported
4	ECP supported
5	Nwait supported
6	Until, Offset, and self-learning vibration suppression not supported
7	All external axes remain static during free curve motion
8	Free curves are not supported when the guide rail is used or when a curve workobject is bound to the positioner
9	J6 independent axis not supported

Exception descriptions are as follows:

No.	Description
1	Too big spacing of input points
2	Too small spacing of input points
3	Too big change in orientation of input points
4	Number of curves exceeds the limit, which should be within the range of 4 to 500 points
5	Too many input points
6	Too less input points
8	Incompatible instruction parameters or functions
9	Too large curve length
10	Too small angle between input points
11	Non-MovS instructions included in curve group
12	Target point associated with motion I/O is not on the current curve segment.
13	The true start point of the curve is incompatible.

12.10.5 Movs Instruction Usage Example

CurveParm c1 = {10,10,0,1,1,80,10,0,0}; # Set the motion parameters for Movs

Movs P[114:122],V[100],Z[0],Tool[0],Wobj[0]; # Run with the default parameters without setting Movs parameters

Movs P[114:122],V[100],Z[0],Tool[0],Wobj[0],CurveParm[c1]; # Set the parameters in the Movs instruction, and runs with the set parameters

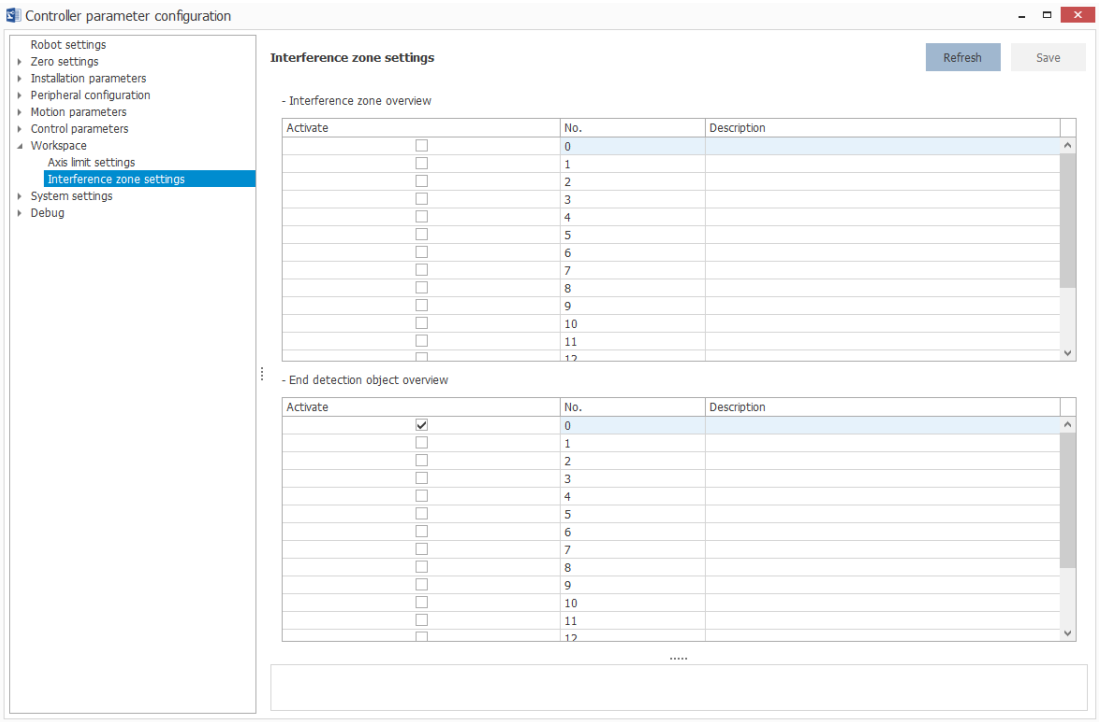
12.11 Interference Zone

12.11.1 Overview

The interference zone function prevents the monitoring objects at the end of the robot from entering the interference zone to avoid collision with obstacles or other devices. You can control whether to allow the robot to enter the interference zone through external input signals, and interact with the robot or other devices through output signals. You can configure a total of 16 interference zones and 16 monitoring objects.

12.11.2 Steps

- See the "Industrial Robot Instruction Guide" for the description of the interference zone instruction SetInterferZone and the monitoring object instruction SetInterferTool.
- Go to "Workspace" > "Interference zone settings".
- The interface of the interference zone function is shown below.

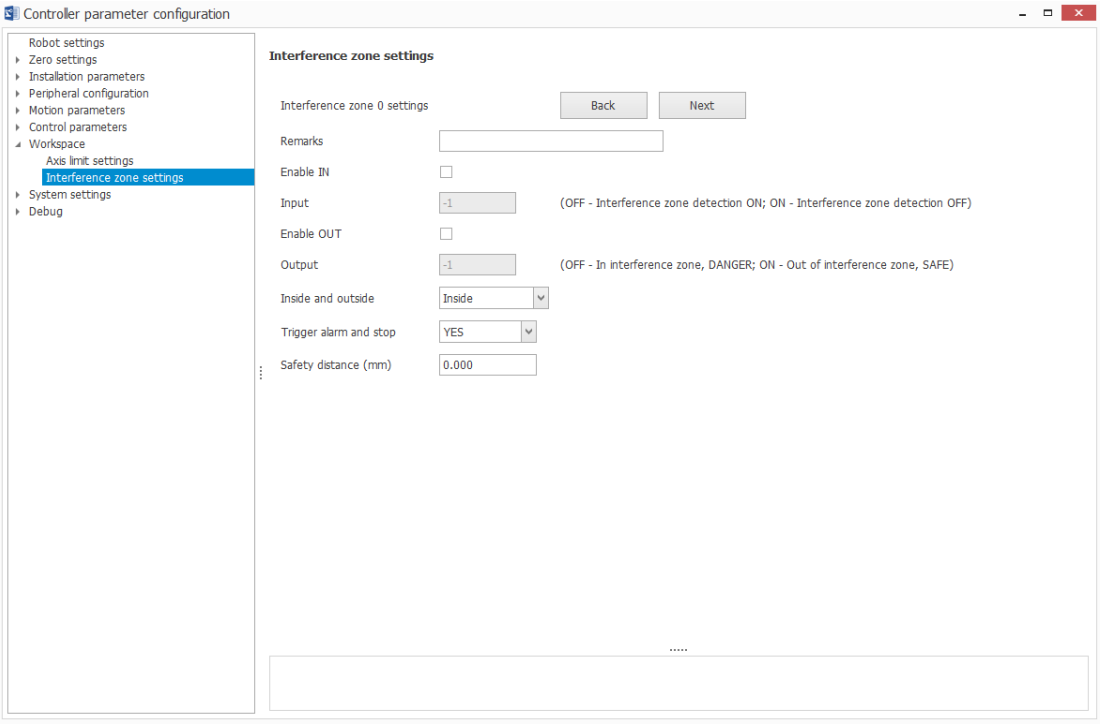


You can activate or deactivate the 16 interference zones and 16 monitoring objects, and view the description of the interference zones and monitoring objects.

CAUTION

Multiple interference zones can be activated simultaneously, but only one end monitoring object can be activated.

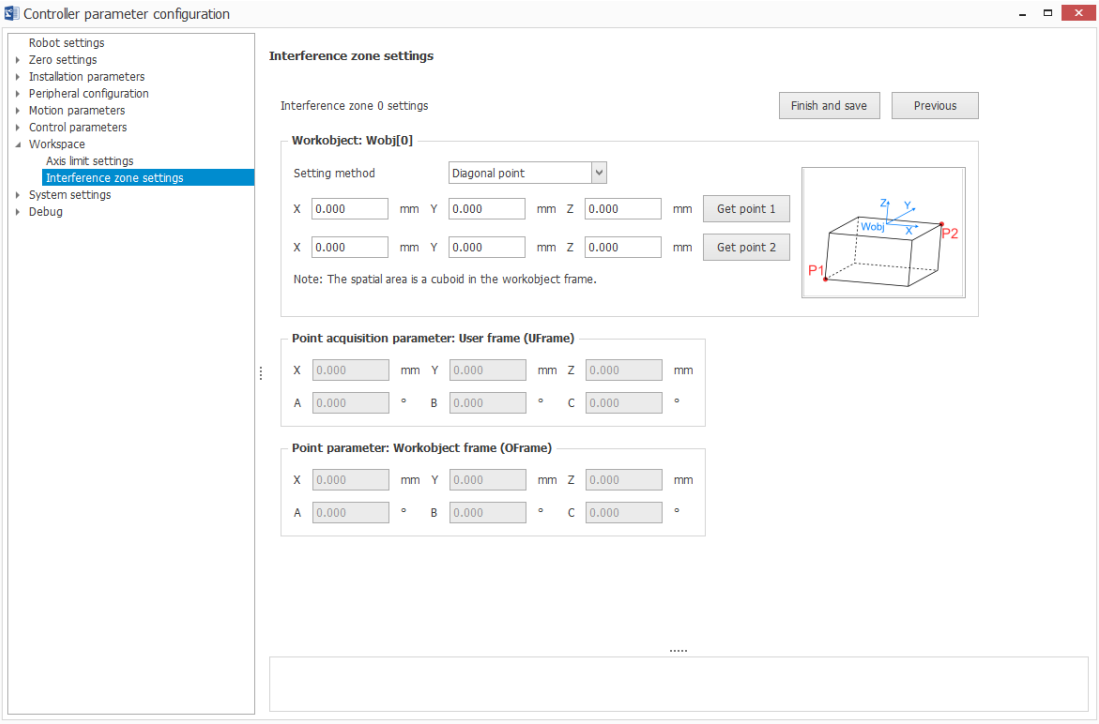
Select an interference zone and double-click it to open the parameter setting interface.



Parameter description:

- Remarks: Descriptive information of the current interference zone
- Enable IN: By default, this option is not checked. Check this option to set the input signal.
- Input: Sets the input signal of the interference zone. When the input signal is OFF, the interference zone detection is enabled. When the input signal is ON, the interference zone detection is disabled.
- Enable OUT: By default, this option is not checked. Check this option to set the output signal.
- Output signal: Sets the output signal of the interference zone. When the output signal is OFF, it indicates that the robot is in the interference zone and is in a dangerous state. When the output signal is ON, it indicates that the robot is out of the interference zone and is in a safe state.
- Inside and outside: Specifies the inside or outside of the interference zone as the interference zone.
- Trigger alarm and stop: Sets whether to generate an alarm when the robot enters the interference zone. The default is "YES". When it is set to "NO", the robot does not alarm and stop when it enters the interference zone.
- Safety distance: Increases or decreases the interference zone as a whole according to the "inside and outside" options. The range of safety distance is 0 to 10000, in mm.

After setting the relevant parameters, click "Next" in the upper right corner.



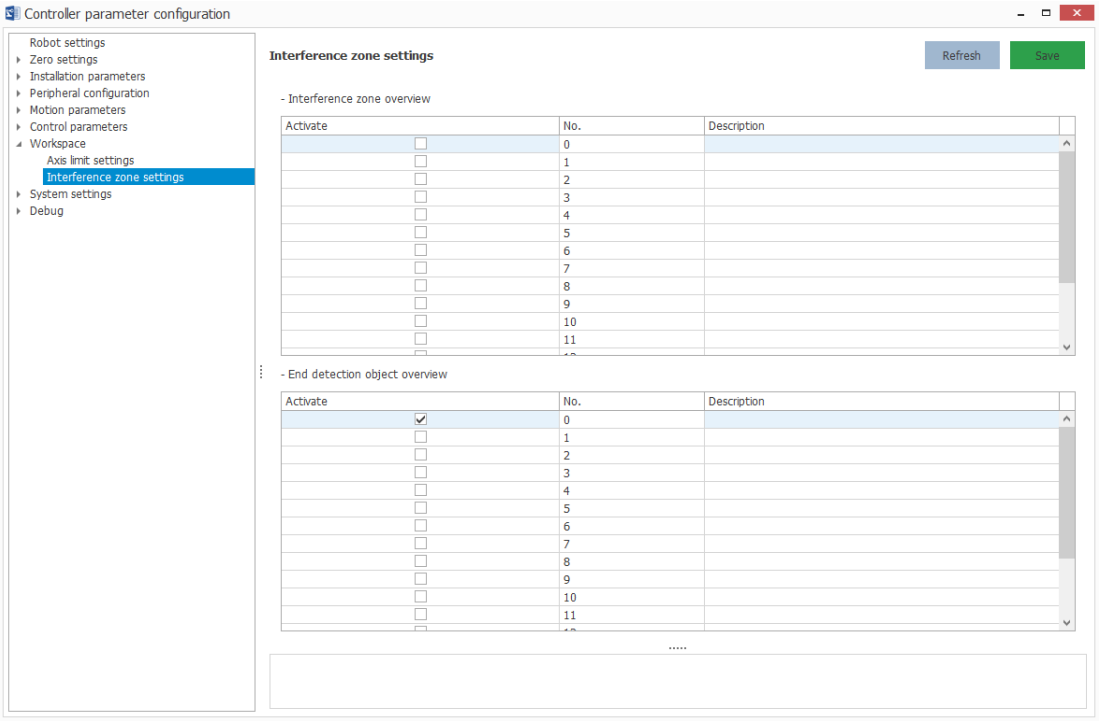
Parameter description:

- Setting method: Sets the interference zone by diagonal point, or by datum point + offset. The points are taken in the current workobject frame.
- You can enter the point manually or click the "Get point" button to acquire the current point automatically. The range of the point value is [-10000,10000], in mm. The range of the offset value is [-10000,0), (0,10000], in mm.
- After getting the points or configuring the points of the interference zone, you can view the frame parameters in the lower part of the interface.

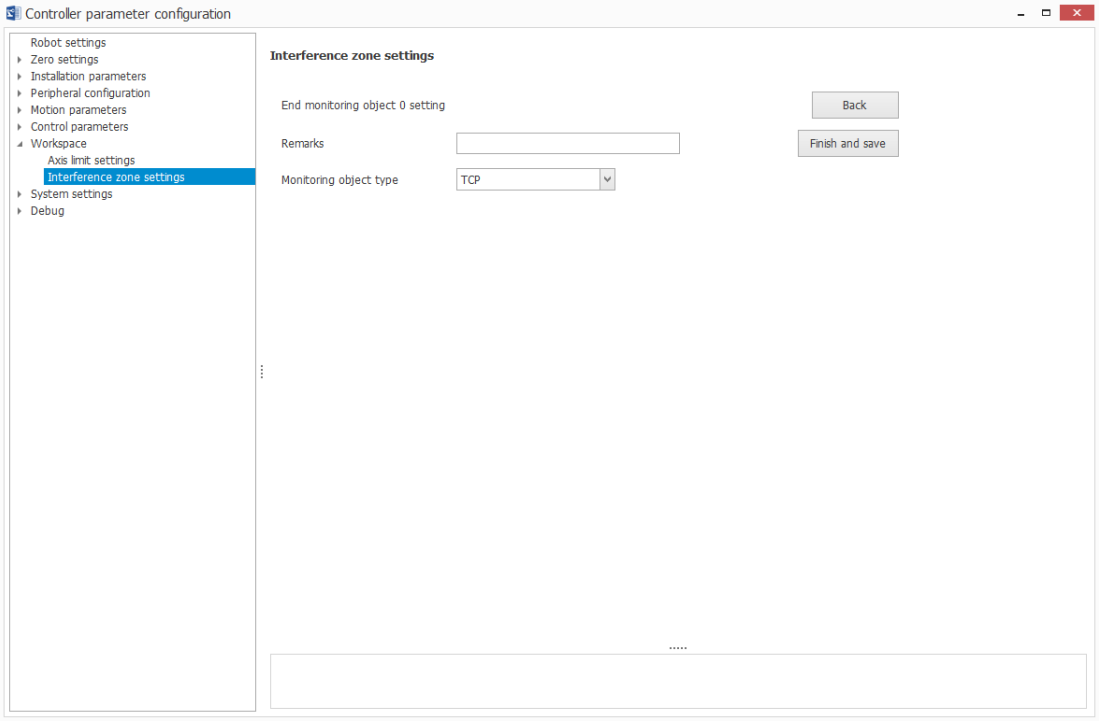
CAUTION

- The value of the safety distance cannot be less than 0.
- The position of the interference zone is relative to the active workobject frame.
- You cannot get the points of the interference zone in a workobject frame where the workobject is not held by the robot.
- You cannot get the points of the interference zone in a workobject frame where the user frame is not fixed.
- If the points cannot form a cuboid, the settings cannot be saved.
- The robot decelerates to stop from the time the end monitoring object enters the interference zone. Therefore, the actual stop position of the robot is inside the interference zone, or the robot passes through the interference zone when the interference zone is small. Note that the greater the robot speed, the greater the distance the robot enters the interference zone. Therefore, set the interference zone as large as possible.

Click the "Finish and save" button to save the interference zone parameters and return to the interference zone overview interface.



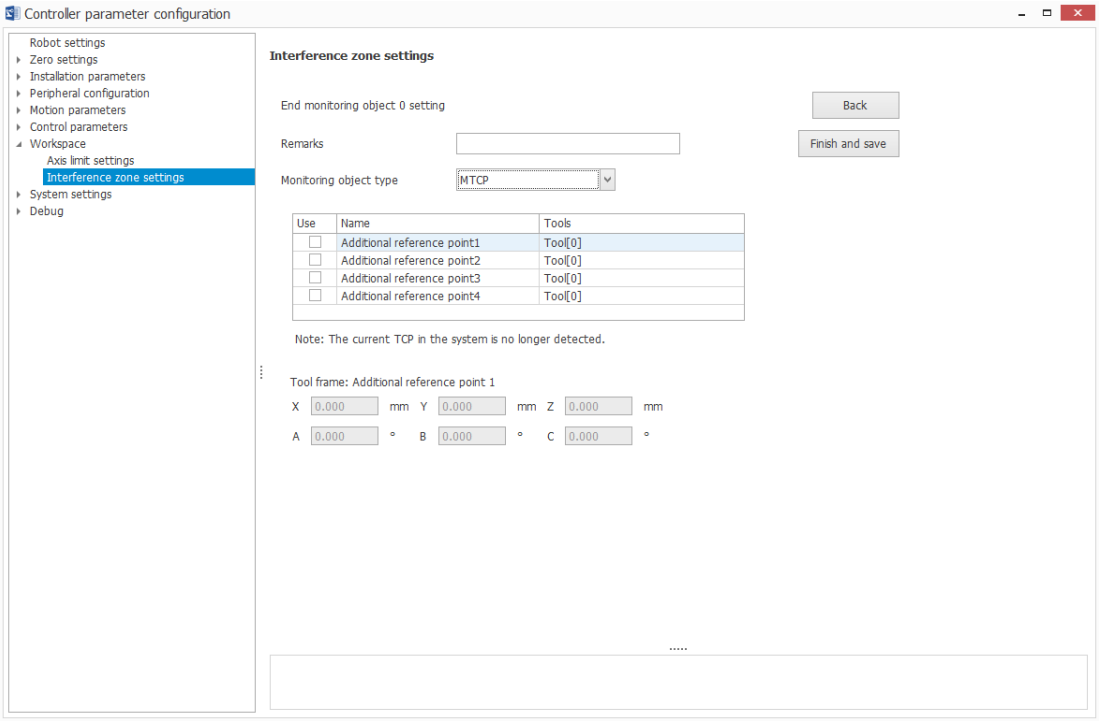
Select an end monitoring object and double-click it to open the parameter setting interface.



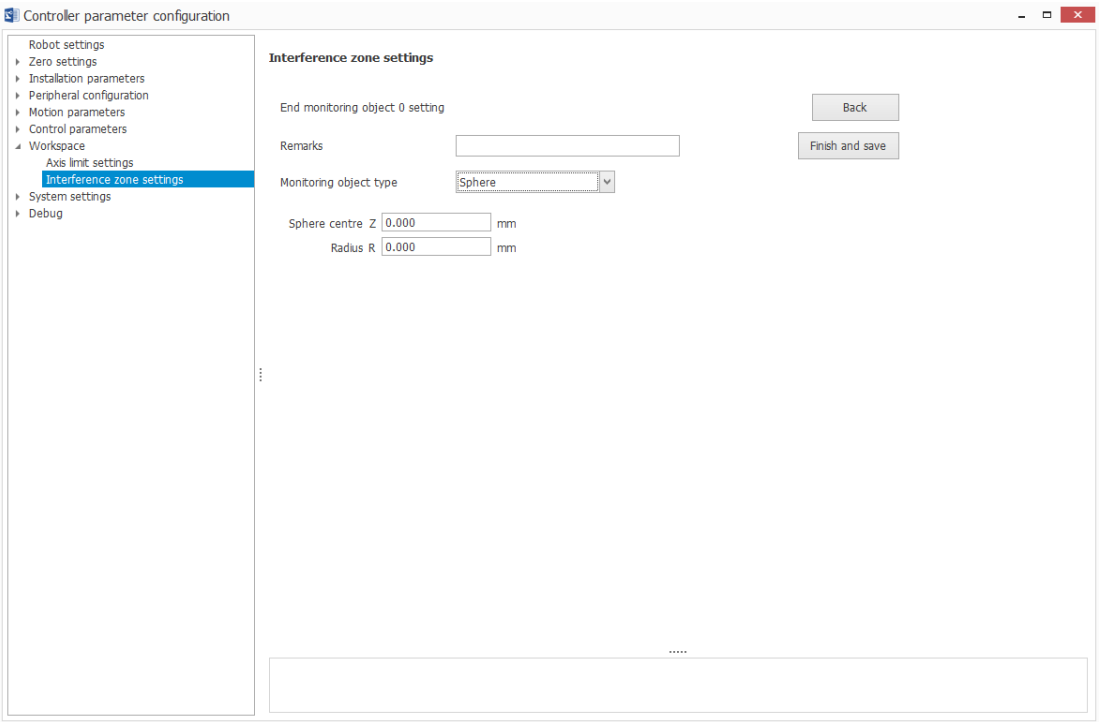
Parameter description:

- Remark: Descriptive information of the current monitoring object.
- Monitoring object type: Includes TCP, MTCP, sphere, and cuboid bounding box. TCP by default.
  - TCP: When the robot grips the tool, the current TCP is used for monitoring. When the robot grips the workobject, the flange center is used for monitoring.
  - MTCP: Supports up to four tools at the same time. You must select a reference point for the tool. The current TCP of the robot is not used for monitoring any more.

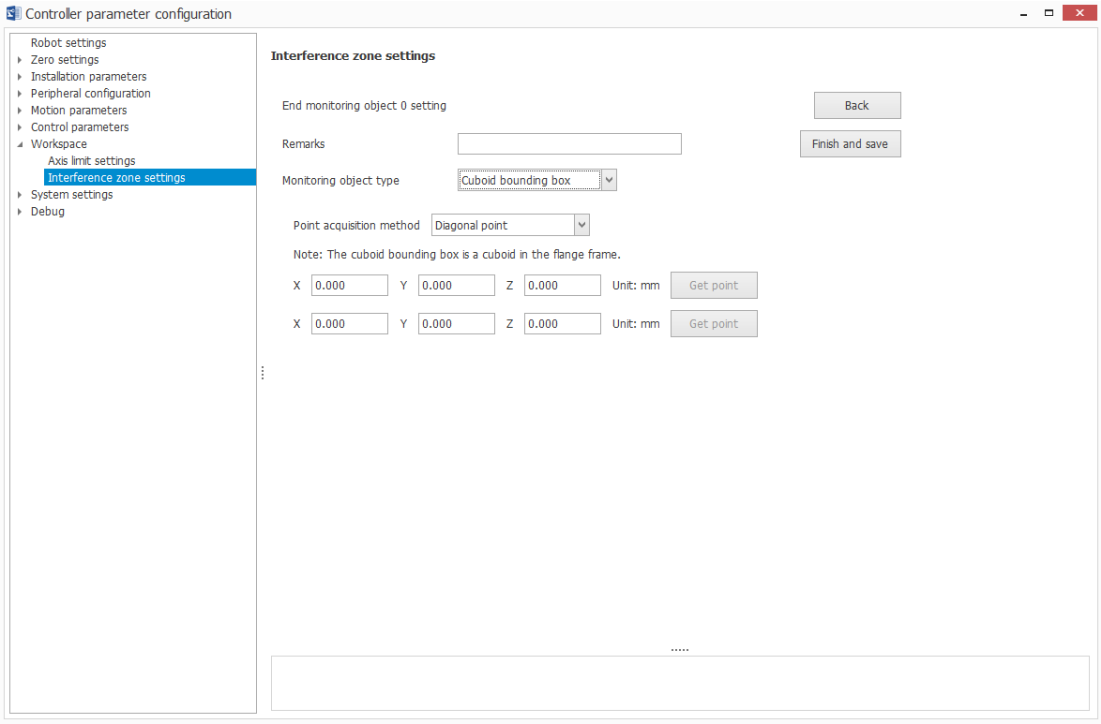
You can view the parameters of the tool frame in the lower part of the interface.



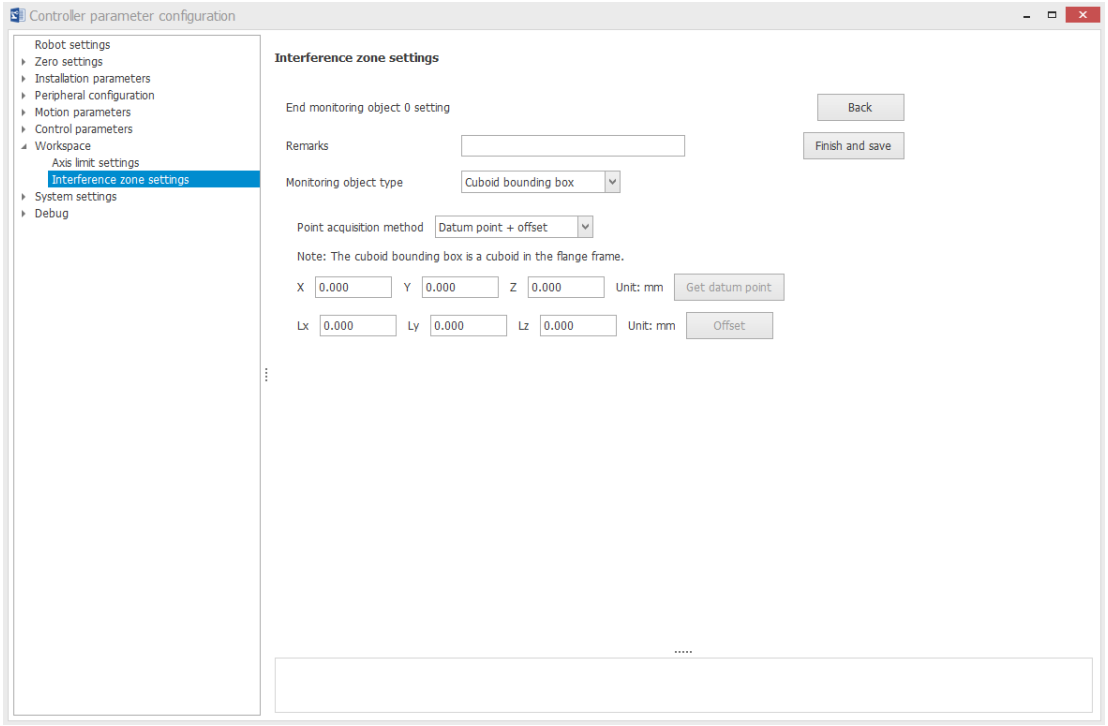
Sphere: The spherical center is offset from the Z direction of the flange frame, the spherical center range is [-10000,10000] and the radius range is (0,10000], in mm. The eccentric sphere is not supported.



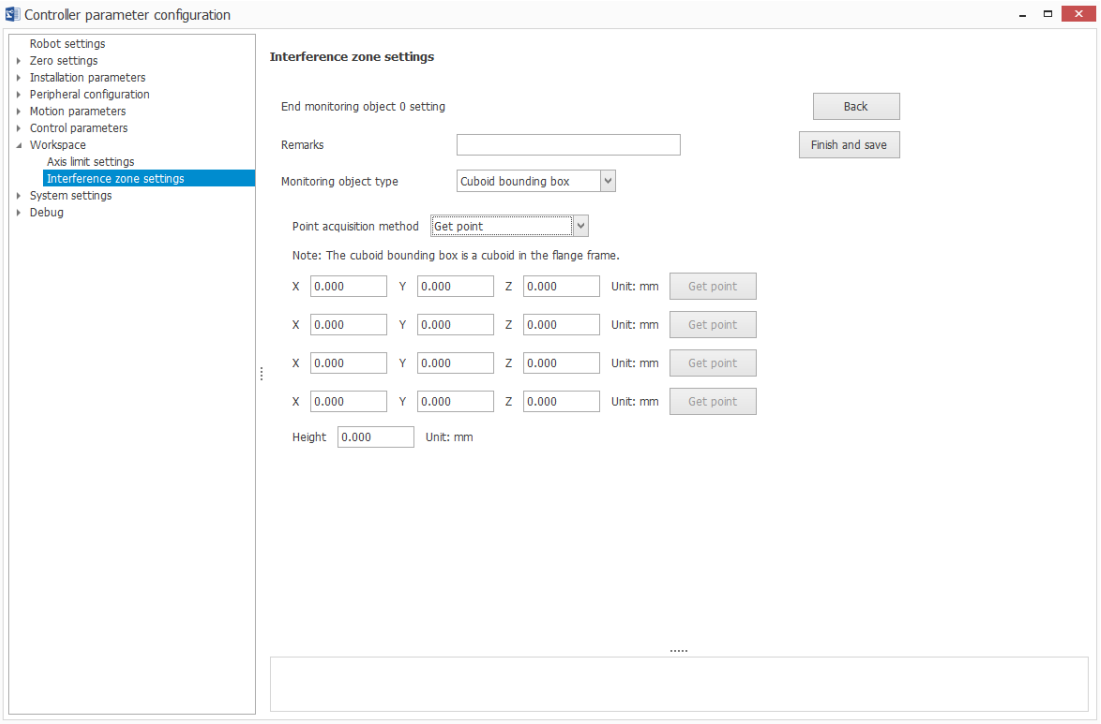
Cuboid bounding box: There are three methods to acquire the points, namely, diagonal point, datum point + offset, and get point.



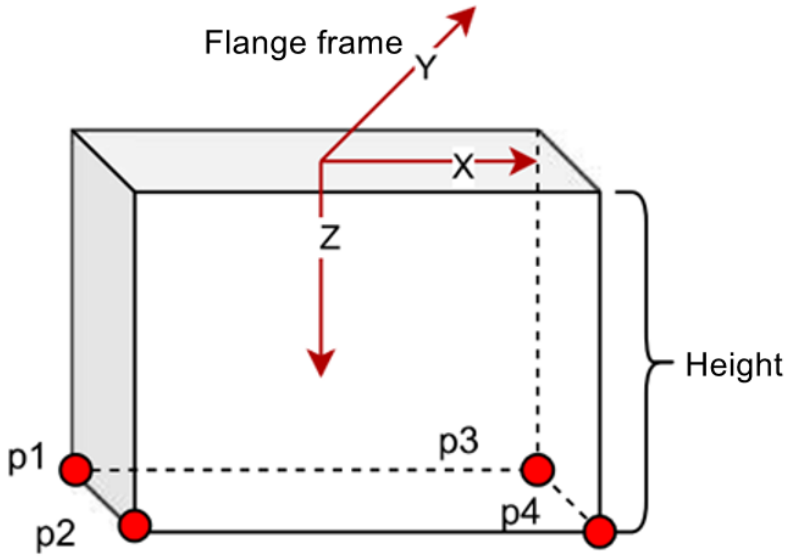
When the point acquisition method is set to "Diagonal point", you can enter the point manually or click the "Get current point" button to acquire the current point automatically. Note that the "Get point" button is active only when the robot is in ECP mode. The points are taken in the flange frame. The range of the point value is [-10000,10000], in mm.



When the point acquisition method is set to "Base point + offset", you can enter the datum point and the offset manually, or click the "Base point" button to get the point automatically and then manually enter the offset. Note that the "Base point" button takes effect only when the robot is in ECP mode. The point and offset are taken in the flange frame. The range of the point value is [-10000,10000], and the range of the offset is [-10000,0), (0,10000], in mm.



When the point acquisition method is set to "Get point", you can enter the point manually, or click the "Get current point" button to get the point automatically. Note that the "Get current point" button is active only when the robot is in ECP mode. The points are taken in the flange frame. Note that the points 1-4 are the four points on the bottom face of the cuboid. The height is the distance in the Z direction under the flange frame. The range of the point value is [-10000,10000] and the range of height is (0,10000], in mm, as shown in the figure below.

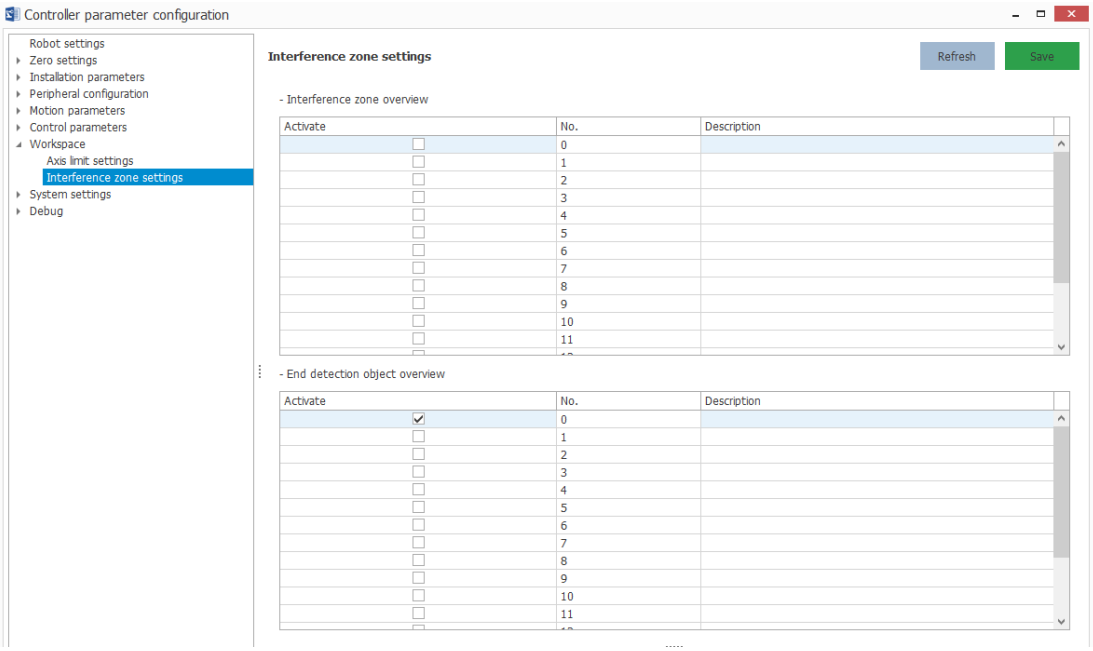


CAUTION

Notes of the cuboid bounding box:

- The orientation of the cuboid bounding box is the same as that of the flange frame. That is, a tilted cuboid bounding box cannot be defined under the flange frame.
- Before getting a point, you must calibrate an external tool and then move the robot to the point on the external tool.
- Regarding the setting of the four points + height, for the 6-axis robots, you need to rotate the J5 axis so that the Z axis of the flange frame points downwards before setting the points and height; for the SCARA robots, just set the points and height normally.

After configuring the monitoring object, click the "Finish and save" button.



Check the interference zone and monitoring object to be used. Click the "Save" button to complete the settings.

12.11.3 Notes

When using the interference zone function, pay special attention to the following notes:

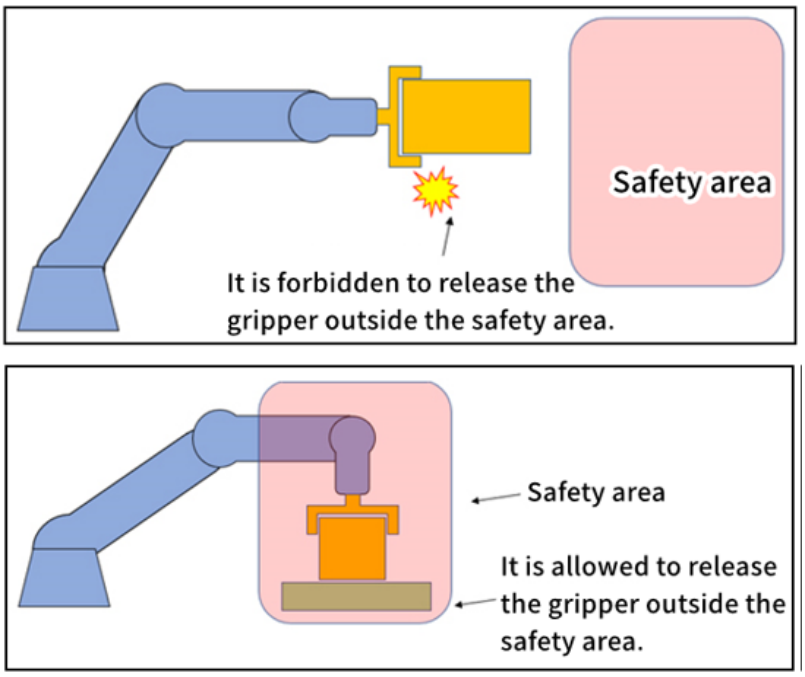
- Once determined, the position of the interference zone under the world frame will not change even if the corresponding tool/workobject parameters are modified. If you need to modify the position of the interference zone, you need to modify the points of the interference zone and save the changes.
- After the monitoring object at the end of robot enters the interference zone, you need to disable the interference zone function or switch the input signal to ON to clear the alarm and continue motion.
- If the "Trigger alarm and stop" parameter is set to "NO", when the robot enters the interference zone, it only triggers the output signal, but does not alert and stop. Be aware of safety!
- The larger the robot speed, the greater the distance the robot enters the interference zone. Therefore, you need to set the interference zone and safety distance as large as possible according to the actual working conditions.

- When the robot grips the workobject, if you set the monitoring object type to "TCP", the flange center is used for detection, which may result in collision between the workobject and external devices or obstacles. In this case, it is recommended to set the monitoring object type to sphere or cuboid bounding box.
- When the interference zone is activated, it is detected in any motion mode.
- When the cuboid bounding box is defined by the four points + height,for the 6-axis robots, you need to rotate the J5 axis so that the Z axis of the flange frame points downwards before setting the points and height. Otherwise, the cuboid cannot be formed properly.

12.12 Safety I/O Operation

12.12.1 Overview

This function is mainly for lithium battery customers. The opening and closing action of the gripper needs to be carried out within the designated area. Opening the gripper outside the safety area may cause the battery to accidentally fall, which may cause a safety accident, as shown below.

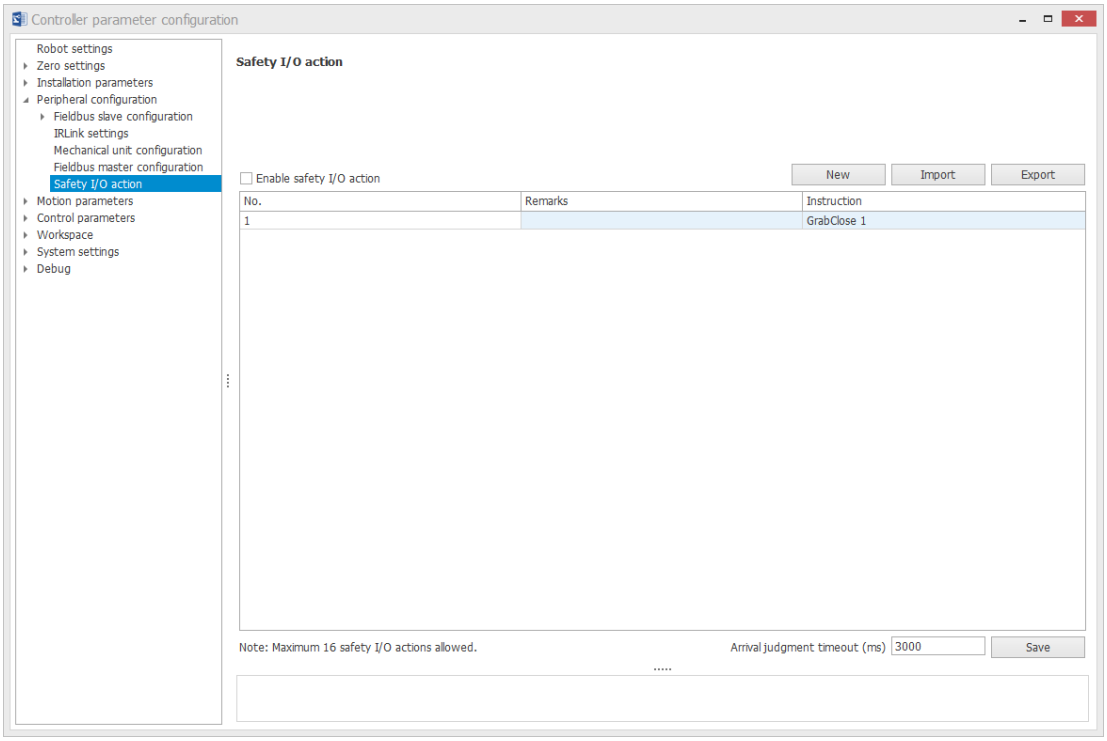


This problem can be solved by defining safety I/O actions and limiting the range of actions.

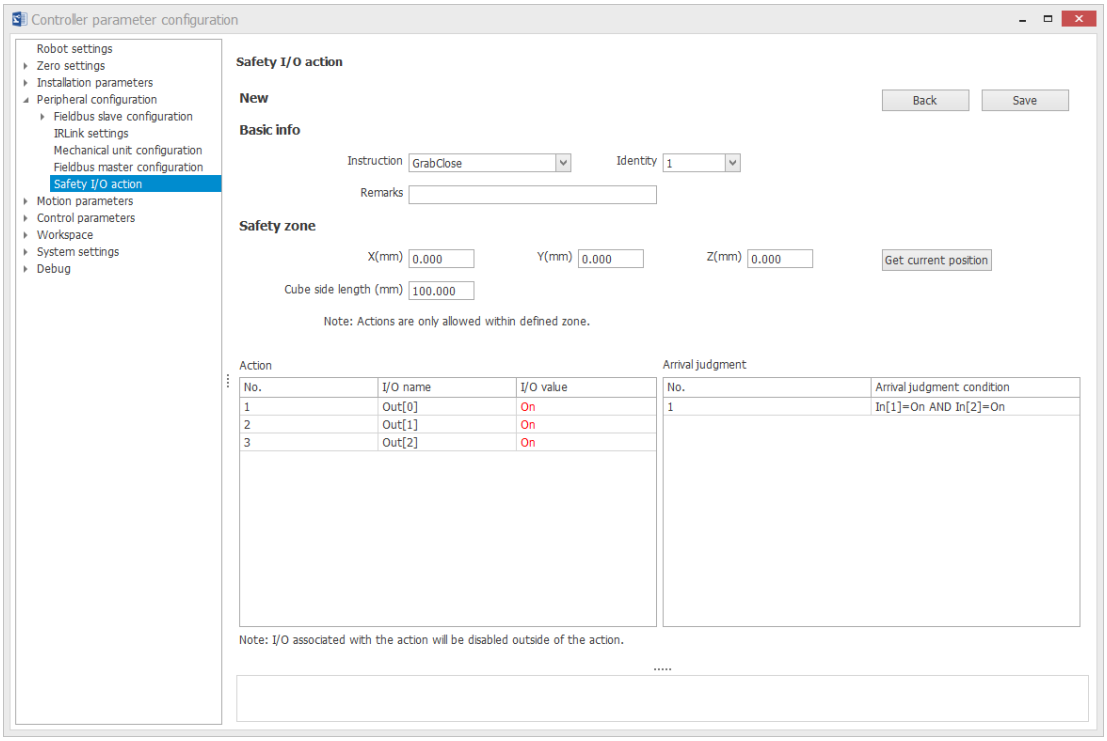
- The safety I/O action is the action of the gripper rather than the motion of the robot. The user can define the area for executing the action, the output signal of the action, and the completion determination conditions of the action.
- The safety I/O actions can be executed through user interface or through instructions in the program.

12.12.2 Steps

1. "Admin" mode is required to access this interface. Go to "Controller parameter settings" > "Peripheral configuration" > "Safety I/O action".



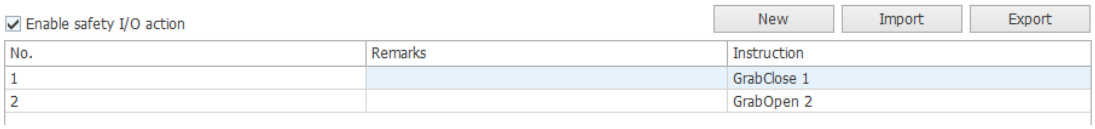
2. Click "New" and add the GrabClose 1 instruction as example.



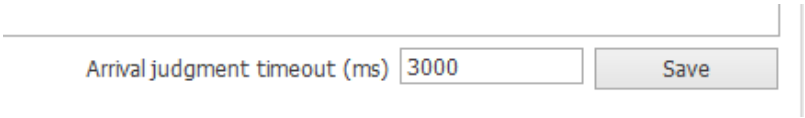
The relevant parameters are described as follows:

- Instruction: GrabClose and GrabOpen instructions are available.
- Identity: Select the unique identification number of the instruction.
- Remarks: Add remark for the instruction, with a character limit of 16 characters.
- Safety zone: Configure the center point of the safety zone and the side length of the safety zone (10 mm to 5,000 mm).

3. Right-click in the "Action" table to display the corresponding operation menu.
   The relevant parameters are described as follows:
   - New: Opens the "Add action" window for configuring the I/O Out value during instruction execution. The number of configurable actions is limited to 16, with a minimum of 1. The I/O number for Out is limited to [0, 127][512, 4607].
   - Switch I/O value: Switches the I/O value (On/Off) of the selected action.
   - Delete: Delete the selected action.
   - Drag with mouse: Drags the selected action to change its order in the execution list.
4. Right-click in the "Arrival judgment" table to display the corresponding operation menu.
   The relevant parameters are described as follows:
   - New: Adds a new arrival judgment instruction. The judgment condition can be left unconfigured or configured as an empty judgment condition. The I/O number for In is limited to [0, 127][512, 4607]. You can configure duplicate judgment conditions, but all duplicate judgment conditions must be identical. For example, multiple judgment conditions can be configured using In[3], but all In[3] must be set to Off.
   - Delete: Deletes the selected arrival judgment condition.
   - Drag with mouse: Drags the selected arrival judgment condition to change its order in the execution list.
5. Click the "Save" button to complete the configuration of the GrabClose 1 instruction. At this point, you can check the "Enable safety IO action" to enable the function on the main interface.



6. You can configure the "Arrival judgment timeout" time (ms) and click "Save" to save the configuration. All GrabClose/GrabOpen actions will use this time as the maximum waiting time for In signals.



7. Call GrabClose1 in the instruction interface.



CAUTION

If the I/O used in the safety I/O action conflicts with other functions, calling the GrabClose/GrabOpen instruction will result in an error.

12.13 Function Activation

12.13.1 Activating the Function

Some functions are paid functions. Before using the paid functions, you need to activate them first.

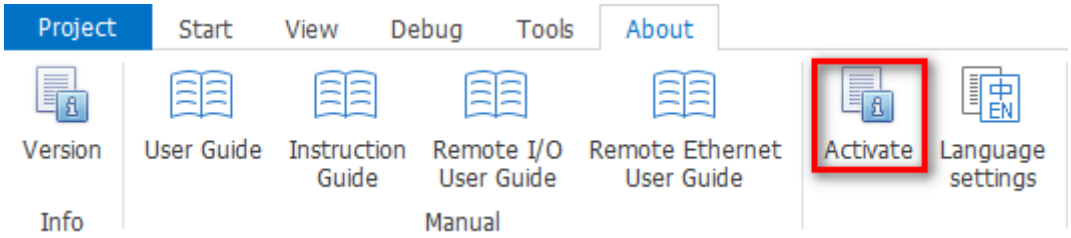


Figure 1 Function activation button

The function activation button is shown in Figure 1. It is available in the following version: S03.21R12 or later, or V4R24C1SPC0 or later. The interface is shown in Figure 2.

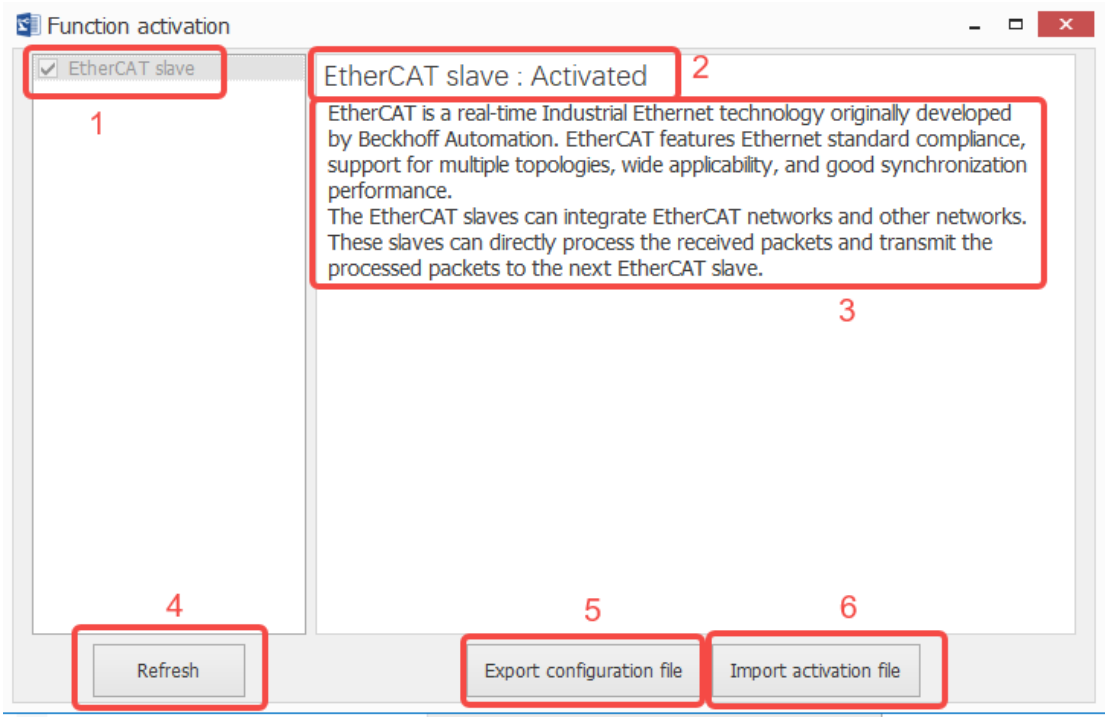


Figure 2 Function activation interface

No.	Function
①	Function module directory
②	Function module title
③	Function module overview
④	"Refresh" button
⑤	"Export configuration file" button
⑥	"Import activation file" button

Take EtherCAT slave function as an example. You can activate this function according to the following steps.

1. The EtherCAT slave function is not available. Go to "Station" > "Controller parameter configuration" > "Peripheral configuration" > "Fieldbus slave configuration" > "Switch settings", you can see that the EtherCAT slave configuration interface is not included.
2. Check "EtherCAT slave" and click the "Export configuration file" button. In the pop-up dialog, specify the export path and export file name, then click the "Export" button.
3. Log in to the authorization platform where you can use the exported configuration file to generate the function activation file (.ino). Download the activation file to the local disk.
4. Go back to the function activation interface, click the "Import activation file" button. In the pop-up dialog, select the downloaded activation file and then click "OK".
5. If the import is successful, you will be promoted that the controller has obtained the activation file successfully. Go to "Station" > "Controller parameter configuration" > "Peripheral configuration" > "Fieldbus slave configuration" > "Switch settings". Now the EtherCAT slave configuration interface is included where you can configure and use the EtherCAT slave function.

CAUTION

- Ensure that the InoRobotLab software version matches the controller software version.
- If the controller hardware is of an old or special version, the function activation interface cannot be opened after InoRobotLab is connected to the controller. You will be promoted that function activation is not required.

12.13.2 Generating the Activation File Using the Authorization Platform

Open https://inoencrypt.inovance.com:8081/ using the computer browser and log in using your account and password.

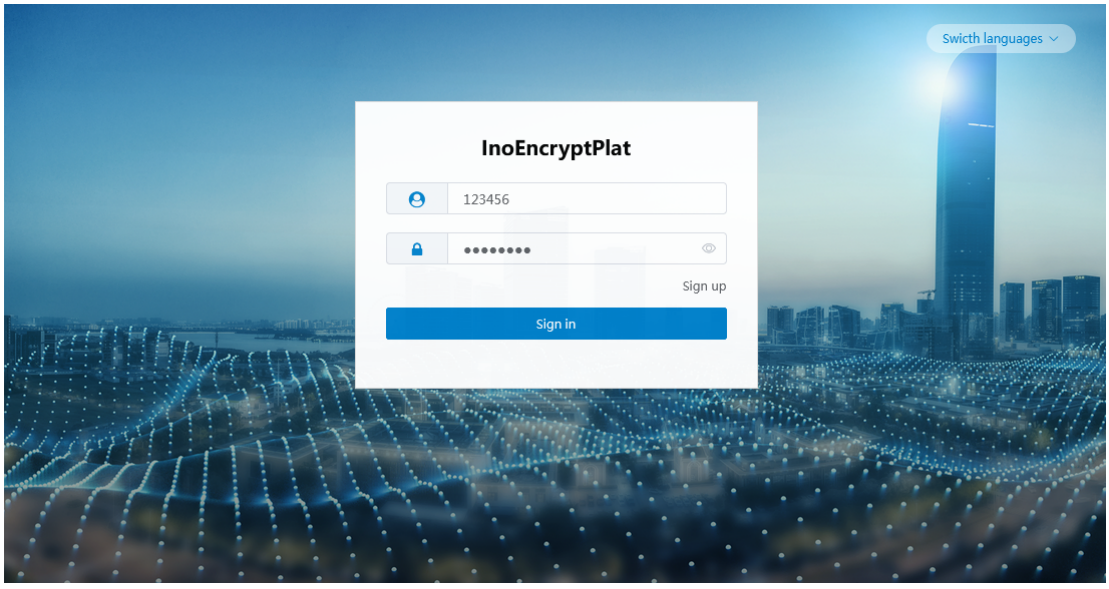


Figure 1 Logging into the authorization platform

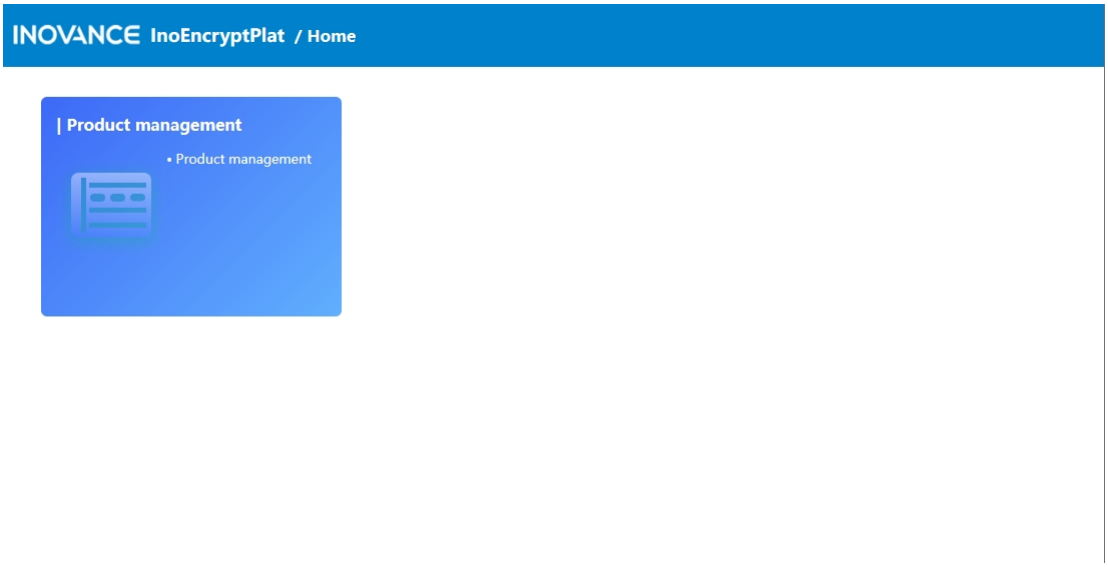


Figure 2 Home page of the authorization platform

If you do not have an account, click "Sign up" and fill in necessary information to register with the authorization platform.

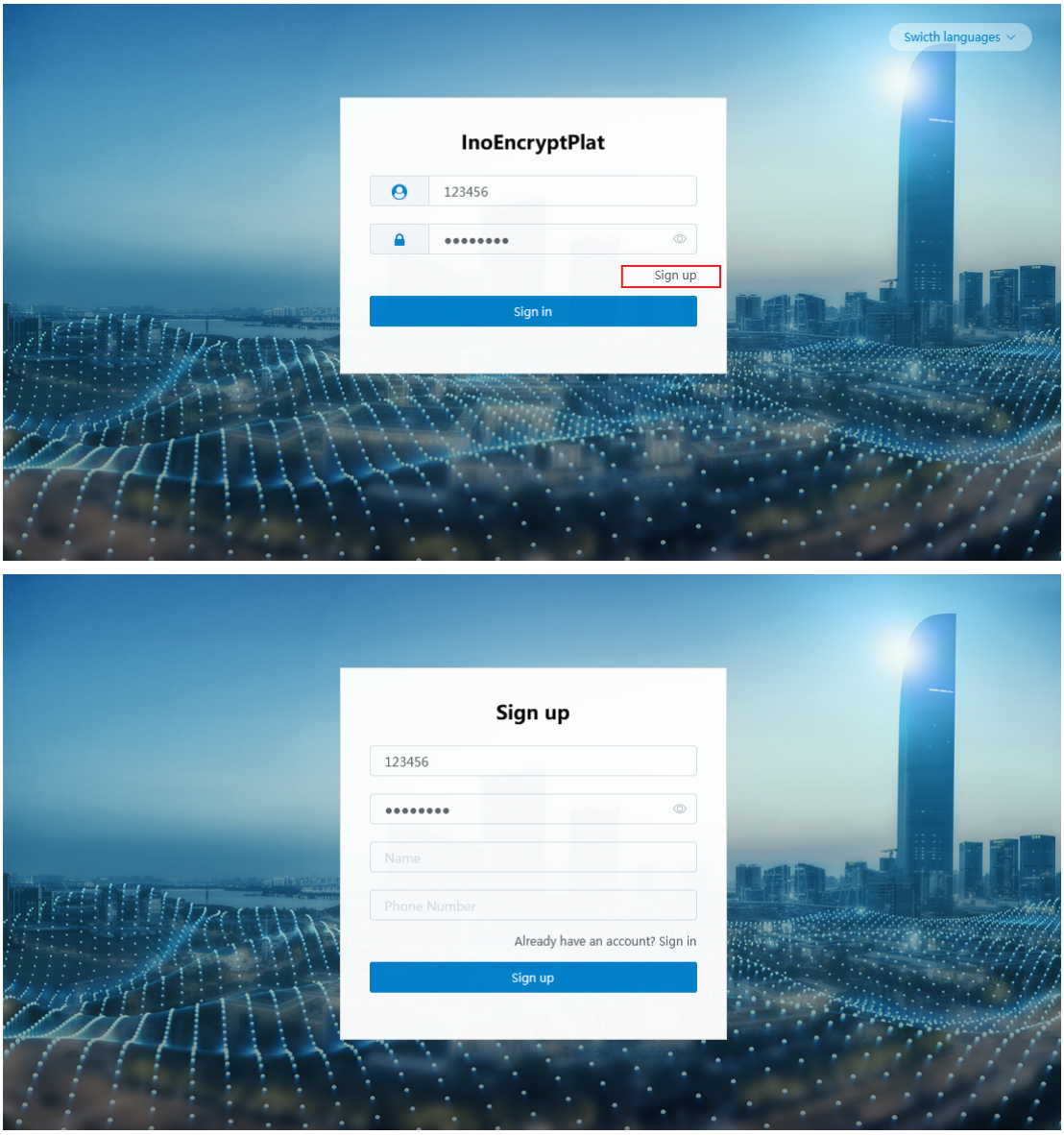


Figure 3 Registering with the authorization platform

After the account registration is complete, wait for the result. You can send an email to chenhaokun@inovance.com for accelerate the approval of your registration. After the approval, log in to the authorization platform with your account and password.

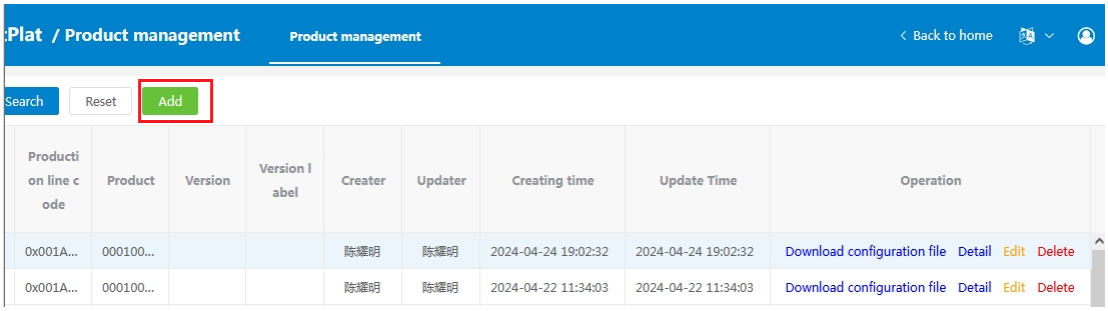


Figure 4

Click the "Add" button in the upper left corner and follow the steps below to generate the function activation file.

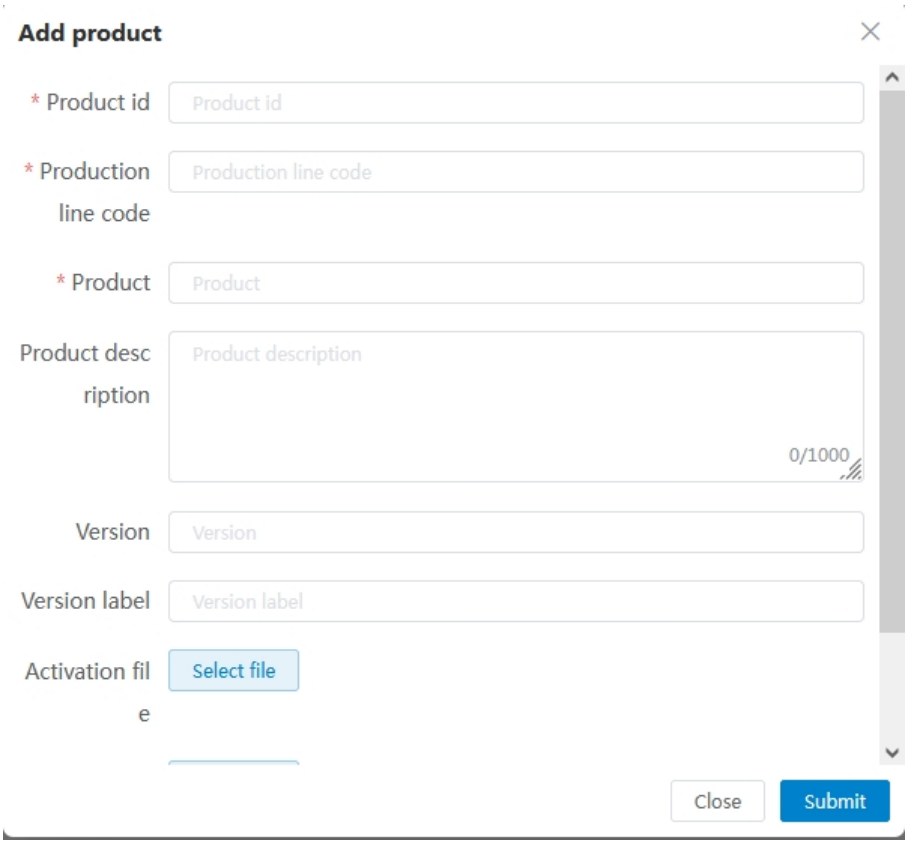


Figure 5

1. Click the "Select file" button, open the function configuration file (.ino), and click the "Submit" button.

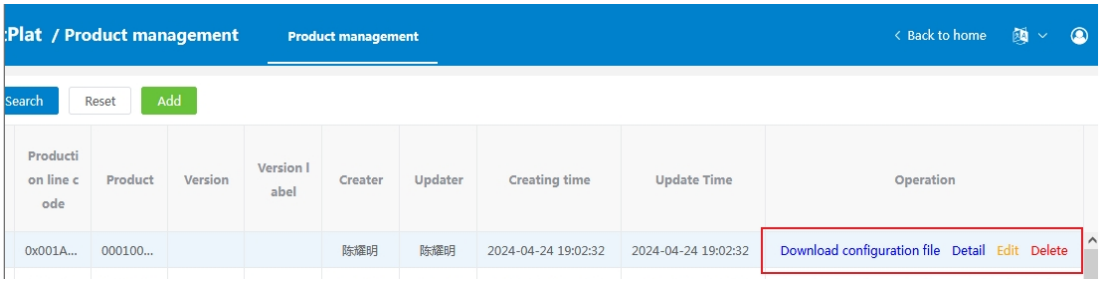


Figure 6

2. A function activation file (.ino) has been generated. Click the "Download configuration file" button to download the function activation file (.ino) to the specified path. If you want to upload a new function

configuration profile (.ino), click "Delete" to delete the generated function activation file (.ino) and then generate it again.

12.14 Interrupt

12.14.1 Overview

- The interrupt function refers to the capability to pause current operations in response to changes in external signals or internal conditions during robot task execution. This function is crucial for ensuring production efficiency, safety, and flexibility.
- Through effective multitasking management and interrupt handling mechanisms, Inovance robots can more flexibly respond to dynamic changes and unexpected situations in production. This not only improves the flexibility and efficiency of the production line, but also enhances the overall reliability and safety of the system. In practical applications, these functions are widely used in automated assembly lines, logistics sorting systems, precision manufacturing, polishing, and other fields.

12.14.2 Function Introduction

12.14.2.1 Registering an Interrupt

Inovance robot provides a series of interfaces for interrupt registration, deletion, individual interrupt signal activation and deactivation, and interrupt function enabling and disabling.

- Call the IConnect instruction to connect the interrupt callback function with the interrupt ID. After the connection is established, the interrupt signal is in the active state by default.
- Write the callback function Trap trapfunc()…EndTrap, and place the events that need to be handled after an interrupt occurs into the interrupt callback function.
- Configure the signal that triggers the interrupt. The currently supported signals for triggering interrupts include I/O, AD/DA, and timer.

```
Start;
  Int IrqId = 0;     ## Interrupt ID
  Int IONum = 50;    ## Interrupt signal I/O number
  IConnect IrqId, TrapFunc();   ## Bind callback function TrapFunc and Interrupt ID
  ISigIn (IrqId, IONum, ON);    ## Bind I/O signal and sets trigger condition to rising edge trigger
  ## User processing logic...
  While 1
    Delay T[1];
  EndWhile;
End;
## Callback function
Trap TrapFunc()
```

```
    Print "hello";
EndTrap;
```

12.14.2.2 Special Notes

12.14.2.2.1 Parameter Class Limitations and Rules

- Interrupts can take effect in multiple tasks, and are independent between tasks. Each task can register 128 interrupts, with a limited interrupt queue length of 64. When the number of generated interrupt tasks exceeds 64, an alarm is triggered and the interrupt task is discarded.
- The interrupt signal can only be triggered when the task is in the running state; otherwise, the interrupt signal will not be checked.
- The interrupt program is also a task, and its operation and stop are consistent with the original task.
- Interrupt triggering rule: When an interrupt is currently being executed and the interrupt condition is met again, the interrupt task will continue to be queued. If interrupt x already exists in the queue and its trigger condition is met again, this interrupt task will be generated once more and the generated interrupt task will still be queued.
- Interrupt queue: When the interrupt queue is full, it can only be cleared through project synchronization and compilation. The maximum length of the interrupt queue is 64. If the queue is already full, any subsequently generated interrupt tasks will not be queued and will be directly discarded, triggering alarm 0x009E.
- Trap functions do not support return values or input parameters. The current interrupt callback function can be defined in task files as well as in public files. When defined in other files, the header file of that file must be included. For example, IConnect 10, Module.trapfunc().
- In I/O class interrupt tasks, input and output signals are used only for signal checking, and not for actual I/O state control. Therefore, no I/O occupancy verification is performed.

12.14.2.2.2 Motion Instruction Processing Timing

Instructions supporting trajectory recovery: MovJ, Movl, Movc, Movabsj, Home, ArmChange.

Instructions not supporting trajectory recovery: Jump, JumpL.



The free curve function Movs and the interrupt function cannot be used simultaneously.

- Entering interrupt does not stop original motion (excluding StopMoving instruction)

If the interrupt task does not contain motion instructions, it executes normally. If it contains blocking motion completion instructions such as WaitInPos, it will wait for the original task to complete its motion before continuing execution.If the interrupt task contains motion instructions, an error 0xA030 will occur.

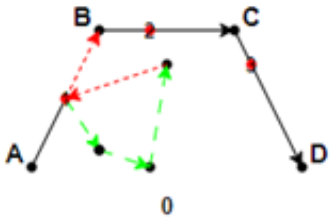
- Entering interrupt stops original motion (including StopMoving instruction)

Stop the original main task motion and retain the preprocessed trajectory information of the original motion, returning to the current stop point. When exiting the interrupt, you need to call the instruction to return to the stop point. If not returning to the stop point, an alarm 0x009F will be triggered. Manually return to the stop point at the time of the interrupt, click "Start", and when prompted that the trajectory has deviated, click "No" to continue execution.

When exiting the interrupt, a prompt will appear indicating the restoration of the original task trajectory data. If the trajectory data is not properly restored, alarm 0x009D will be triggered.

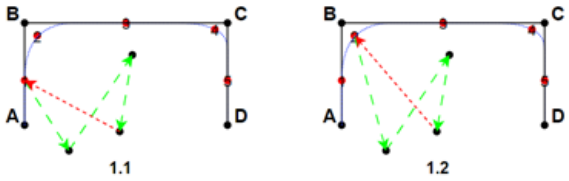
Exit the interrupt motion and clear the motion data to restore the original trajectory. If there is independent axis operation related data when entering the interrupt, the independent axis data must be restored.

Without transition



When an interrupt occurs when the robot moves to point 1 between A and B, the system enters the interrupt and executes the cyan trajectory. After completing the trajectory, you need to call the instruction to return the robot to point 1. Upon arrival, the robot will continue executing from point 1 to point B.

With transition



When an interrupt occurs when the robot moves to point 1 between A and B, the system enters the interrupt and executes the cyan trajectory. After completing the trajectory, you need to call the instruction to return the robot to point 1. Upon arrival, the robot will continue executing from point 1 to point B.

When an interrupt occurs when the robot moves to point 2 between A and B, the system enters the interrupt and executes the cyan trajectory. After completing the trajectory, you need to call the instruction to return the robot to point 2. Upon arrival, the robot will continue executing along the transition segment.

- Notes for Instructions not supporting trajectory recovery

When an interrupt occurs at an instruction that does not support trajectory recovery, after the interrupt operation ends, the system will skip the motion instruction line upon exiting the interrupt, and the motion I/O associated with that motion instruction will not take effect.

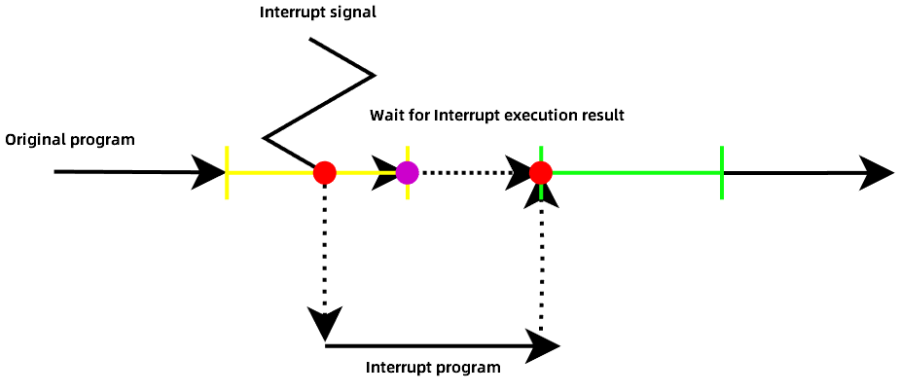
- Special instructions and parameters

Offset: When entering an interrupt, the target point of the first motion instruction will use the point where StopMoving() was called to stop the motion as the Offset parameter reference point.

ArmChange: When entering an interrupt, the target point of the first motion instruction will use the point where StopMoving() was called to stop the motion as the ArmChange parameter reference point.

12.14.2.2.3 Non-Motion Instruction Processing Timing

During the execution of the original program, if a non-motion instruction has not been completed and an interrupt occurs at this time, the interrupt program and the original task program will execute in parallel. The original task will complete the instruction and then wait for the interrupt program to complete before continuing with the subsequent instructions.



CAUTION

- When a stop operation is performed during the execution of an interrupt program and the original task, both the original task and the interrupt task will stop. When starting again, the original task will wait for the interrupt to complete before resuming execution of the instruction.
- If the original task is executing a blocking instruction, such as Wait, Delay, WaitInPos, etc., the original task program and the interrupt program are executed in parallel. If the interrupt program ends before the blocking instruction completes, the interrupt ends normally, and the blocking instruction continues execution. After the blocking instruction completes, the program proceeds to the next instruction.

12.14.3 Instructions

This section only provides a summary of all interrupt-related instructions. For detailed specifications, see the "Industrial Robot Instruction Guide".

- Configuration-related instructions: IConnect, IActive, IDeactive, IEnable, IDisable, IDelete.
- Signal-related instructions: ISigIn, ISigOut, ISigAD, ISigDA, ISigTimer.
- Motion-related instructions: StopMoving.
- Callback function: Trap trapfunc()…EndTrap.

Example:

#1. Without motion instruction

```
Start;
  Int IrqId = 0;     ## Interrupt ID
  Int IONum = 50;    ## Interrupt signal I/O number
  IDelete IrqId;   ## Delete Interrupt and clears corresponding configuration of Interrupt ID
```

```
    IConnect IrqId, TrapFunc();   ## Bind callback function TrapFunc and Interrupt ID
    ISigIn (IrqId, IONum, ON);    ## Bind I/O signal and sets trigger condition to rising edge trigger
    ## User processing logic...
    While 1
        Delay T[1];
    EndWhile;
End;
## Callback function
Trap TrapFunc()
    Print "hello";
EndTrap;
```

2. With motion instruction

```
Start;
    Int IrqId = 0;      ## Interrupt ID
    Int IONum = 50;     ## Interrupt signal I/O number
    IDelete IrqId;    ## Delete Interrupt and clears corresponding configuration of Interrupt ID
    IConnect IrqId, TrapFunc();   ## Bind callback function TrapFunc and Interrupt ID
    ISigIn(IrqId, IONum, ON);    ## Bind I/O signal and sets trigger condition to rising edge trigger
    While 1
        MovAbsJ JP[0],V[30],Z[0],Tool[0], Until IN[0]==ON;
        MovAbsJ JP[1],V[30],Z[0],Tool[0], Until IN[0]==ON;
    EndWhile;
End;
Trap TrapFunc()
JP[100] = StopMoving;     ## Stop the original motion and record the current stop position
    MovAbsJ JP[1],V[30],Z[0], Tool[0];  ## Execute new motion
    ## Other processing logic
    ## ...
    ## Other processing logic
    MovAbsJ JP[100],V[30],Z[0], Tool[0];     ## Return to stop position
EndTrap;
```

12.14.4 Function Coupling

12.14.4.1 Breakpoint

- Breakpoints cannot be set in interrupt programs.
- Breakpoints can be set outside interrupt programs.



In debug mode, if you click "Start", the instruction will run to the breakpoint and trigger an interrupt stop. If an interrupt occurs, it will also be stopped. If an interrupt occurs while the system is waiting for the original motion to stop at the breakpoint, the system will enter the interrupt program for execution. After execution is completed, it will exit the interrupt and continue executing the original motion (requires instructions that support trajectory recovery) before stopping.

12.14.4.2 Single Step

- Single step execution is not supported in interrupt programs.
- Single step execution is possible outside interrupt programs.

12.14.4.3 Set Start Line or Return to Start Line

Do not set the start line within an interrupt program.

12.14.4.4 Start/Stop Type

- During the execution of an interrupt program, start, stop, activate, emergency stop, disable, mode switching, exit, and other stop buttons can normally halt the interrupt motion in progress (including the start and stop buttons on the task management interface).
- When an interrupt program is being executed, you can exit the interrupt program by setting the start line to a non-interrupt program or returning to the start line, or through project synchronization.

12.14.4.5 Project Synchronization

During project synchronization, if the task triggers compilation, the interrupt registration configuration information and the interrupt tasks generated in the interrupt queue will be cleared, and any running interrupt programs (including motion) will be stopped.

12.14.4.6 Trajectory Recovery

- If the original task has no motion trajectory when an interrupt is triggered, and the interrupt task itself has no motion instructions, then clicking the pause button will make trajectory recovery impossible once any trajectory deviation occurs.

- If the original task has a motion trajectory when an interrupt is triggered, and StopMoving is called inside the interrupt without any new trajectory being issued, then upon resuming operation the system will not perform trajectory recovery but will instead continue from the current stop point. However, before exiting the interrupt, the robot must return to the original stop point to continue execution.
- If the interrupt task itself has a motion trajectory that deviates, normal trajectory recovery can be achieved by pausing and then resuming operation.

12.14.4.7 Function Type

- Load auto-tuning: If this function is activated while an interrupt has not been exited, the project will automatically return to the start line and the interrupt will be exited.
- Self-learning vibration suppression: When an interrupt is triggered, self-learning vibration suppression function will be disabled, and the self-learning file being learned will be cleared.
- Independent axis: When entering an interrupt, the independent axis stops and its current status is recorded. Upon exiting the interrupt, the independent axis is restored.
- Tracking process: Interrupt programs do not support tracking process related instructions (see the relevant descriptions in the "Industrial Robot Instruction Guide").
- Process plug-in: Interrupt function and process plug-in function cannot be used simultaneously.
- Soft float:
  - Entering an interrupt will disable the enabled soft float, and exiting the interrupt will not re-enabled it.
  - Interrupt programs do not allow instructions to enable the soft float.
- Free curve: Interrupt function and free curve function cannot be used simultaneously.
- Until parameter: When a motion instruction contains an Until parameter and the condition is not triggered, executing StopMoving will forcibly exit the Until state. Upon exiting the interrupt, the program will restore this point and recover the corresponding state.
- Multi-point recipe: If a new point file is loaded during an interrupt and the original point file is not restored upon exiting the interrupt, the file loaded within the interrupt will remain in use after the interrupt is exited.

12.14.5 Notes

- InqId is a unique value, and cannot be occupied by other callback functions after configuration. The same interrupt ID cannot be bound to multiple callback functions, but different interrupt IDs can be bound to the same callback function.
- Interrupts can take effect in multiple tasks, and are independent between tasks.
- Motion instructions can only be executed in the main interrupt task.
- When an interrupt is being executed and the same interrupt is triggered again, the interrupt event will still be queued; if there is already a triggered signal in the interrupt queue and it is triggered again, the interrupt event will still be queued.
- New interrupts can be defined in the interrupt program.
- During decoding, the interrupt queue and interrupt configuration information of the corresponding task will be cleared, the same applies to static tasks. If decoding is not triggered, the configuration information will not be cleared.

- When the original task runs to the End and stops, if an interrupt occurs at this exact moment, the interrupt task will also stop.

13 Multi-language (Language Settings)

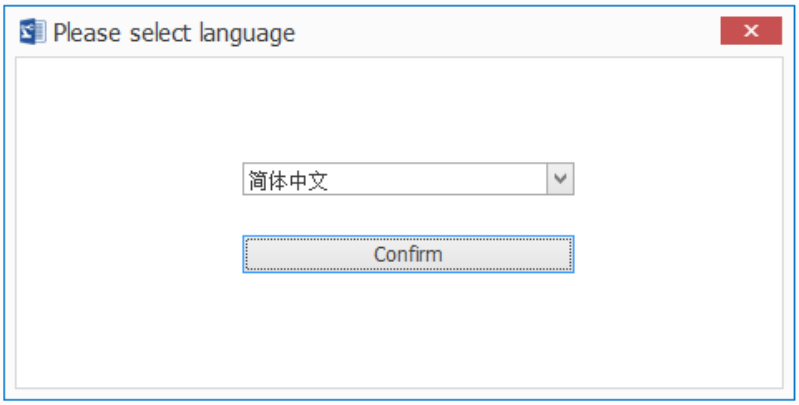
13.1 Overview

Multilingual support is designed to accommodate the language requirements of different countries and regions. Inovance robot interaction softwares support language switching, interface translation, localized images and icons, localized help and documentation, and multilingual keyboard support.

13.2 Multilingual Interface

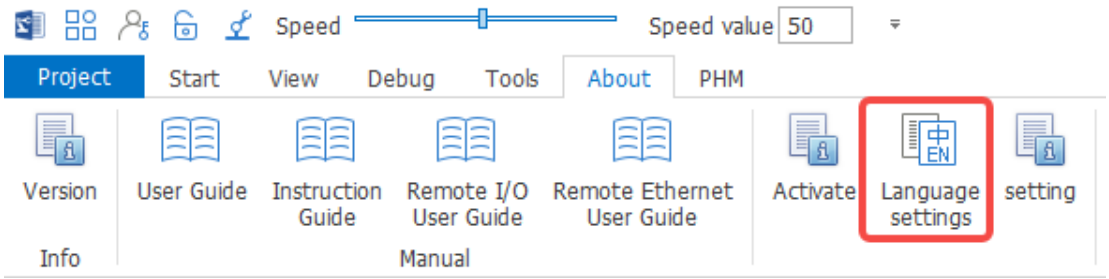
13.2.1 Initial Setup on First Launch

1. A language initialization dialog box pops up when the software is launched for the first time.
2. The language corresponding to your current operating system is pre-selected in the dialog box.
3. This dialog box will no longer appear after initialization is complete.
4. This dialog will pop up again after the language is restored to factory settings.



13.2.2 Accessing the Function

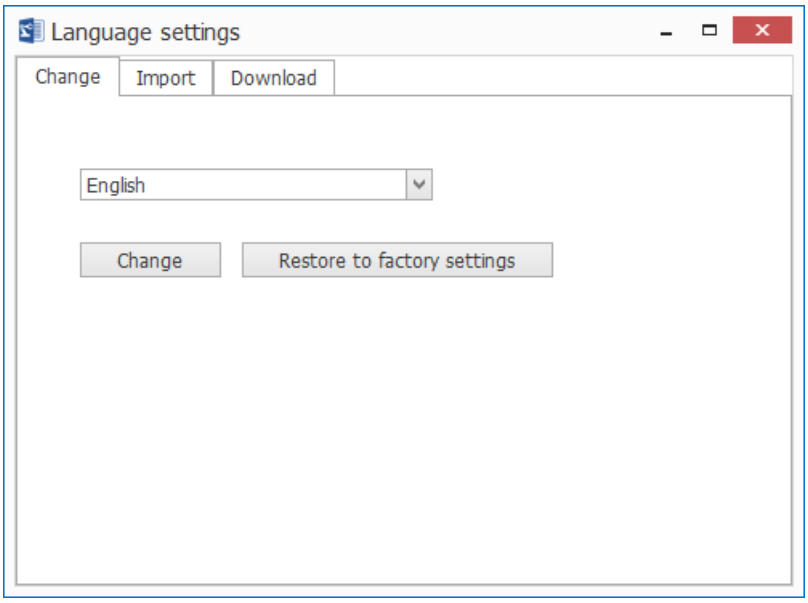
Go to "About" > "Language settings" under any user authority.



13.2.3 Switching the Language

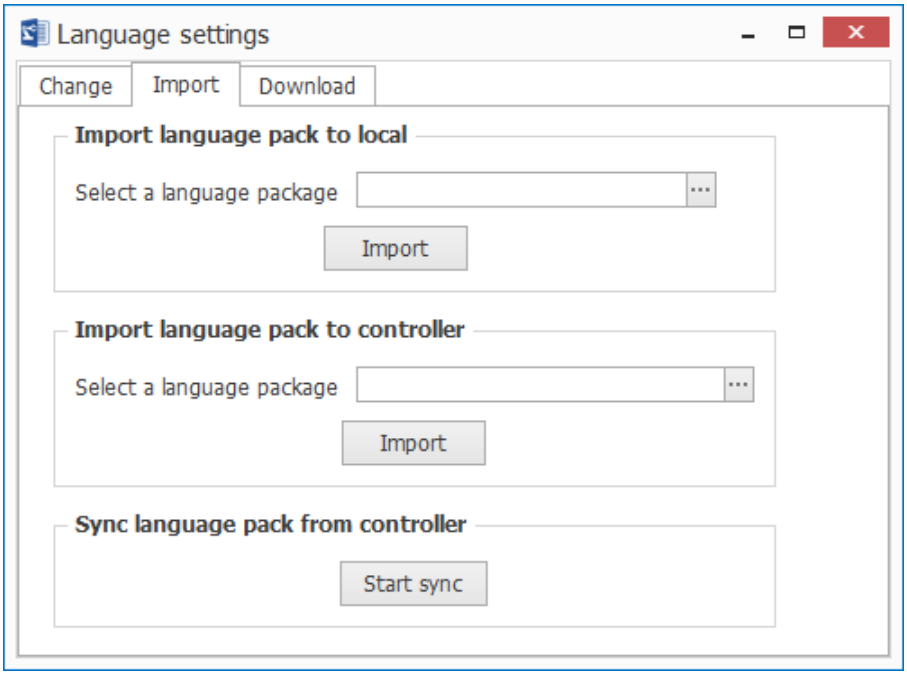
Select the desired language and click "change". A prompt will indicate that a software restart is required for changes to take effect. If the same language is selected, clicking "change" will prompt you to select a different language. If you click "Restore to factory settings", all language-related settings will be restored to

their factory state, and a prompt will indicate that a software restart is required for changes to take effect. The factory language pack is stored in the LanguageFactory folder under the program directory.



13.2.4 Importing the Language Pack

1. Import a language pack to local: Load a language pack to the local.

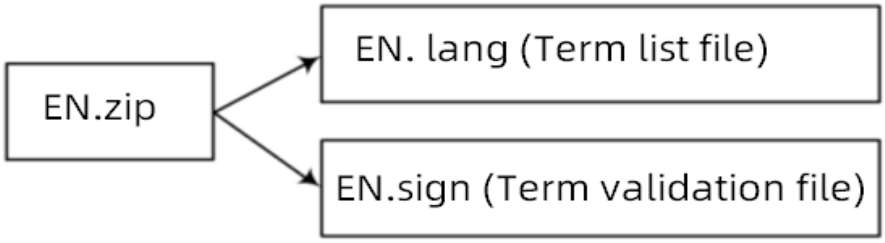


A failure will be prompted if the language pack format is incorrect.

A prompt will also appear if the language pack version is incompatible:

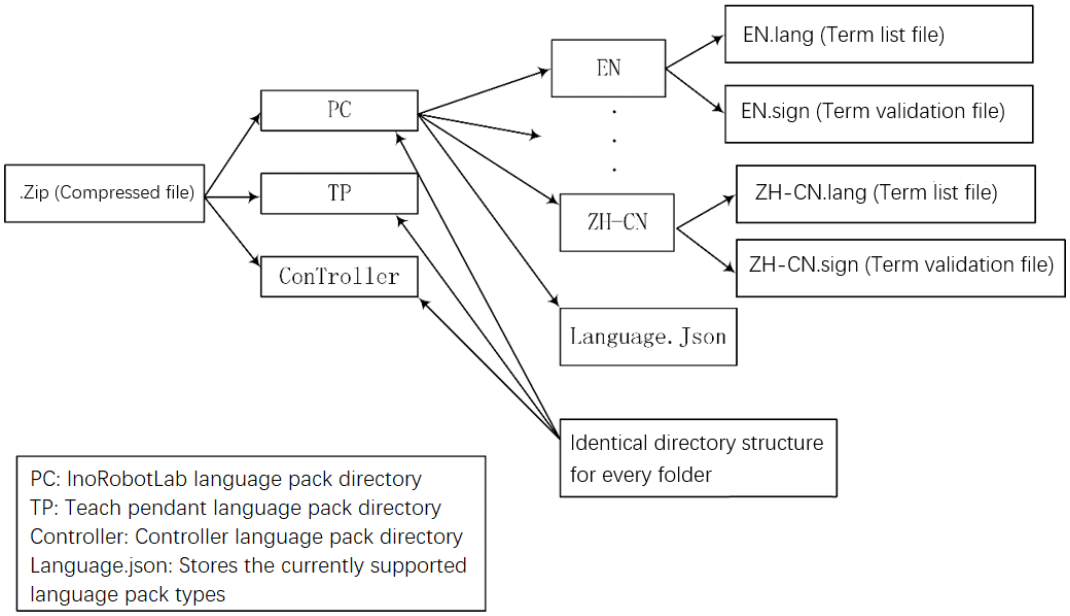
- If the version of the language package to be imported is higher than the current software version, it prompts whether to continue the import.
- If the version of the language pack to be imported is lower than the current software version, it warns that it may cause display anomalies and prompts whether to continue.

Language package structure:

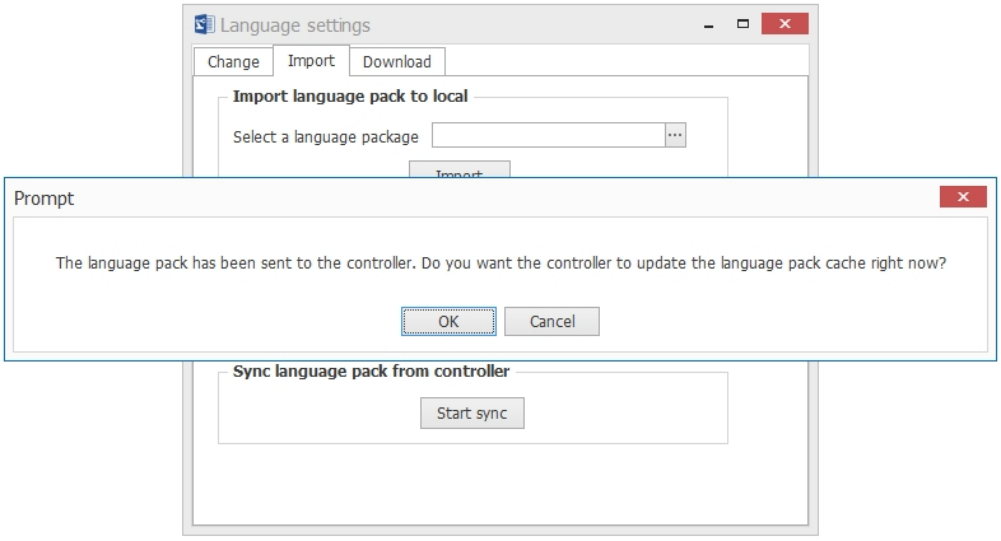


Import directory: "Language" folder under the program directory.

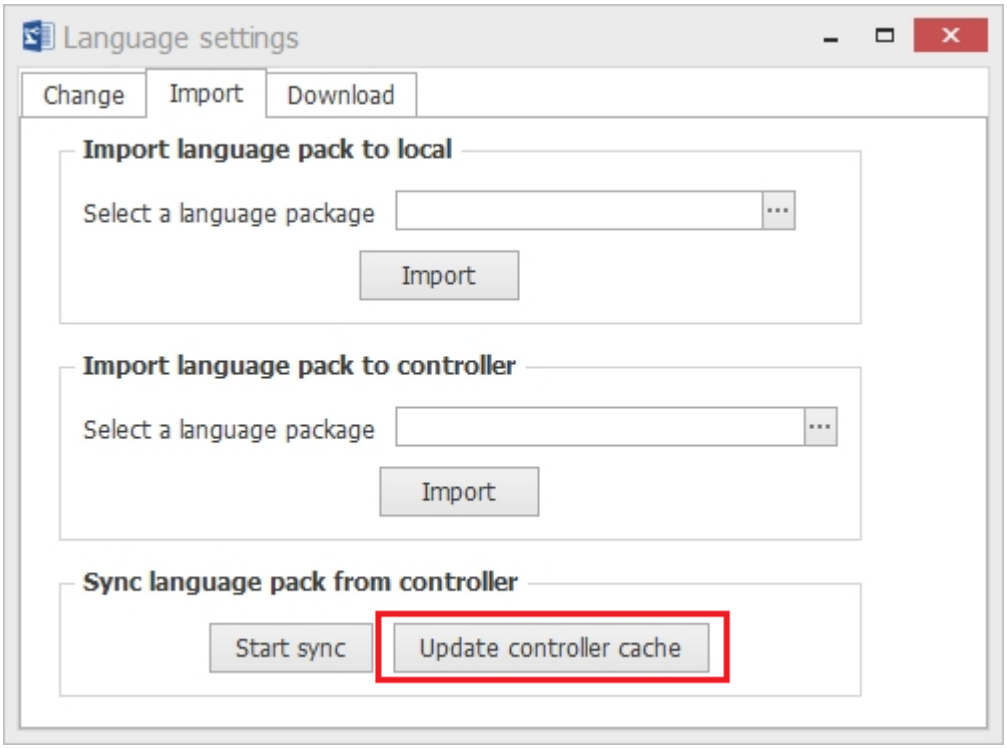
2. Import language pack to controller: Deploy the language pack to the "Language" folder under the SD card directory of the controller.
   1. Compressed file password: InoLanguageV4R24C2.
   2. Compressed file directory structure (as shown in the figure below).



   3. All existing language files in the controller will be cleared during the deploying.
   4. After successful deploying, it prompts whether to update the controller's language package cache (i.e., make the imported language package effective in the cache).

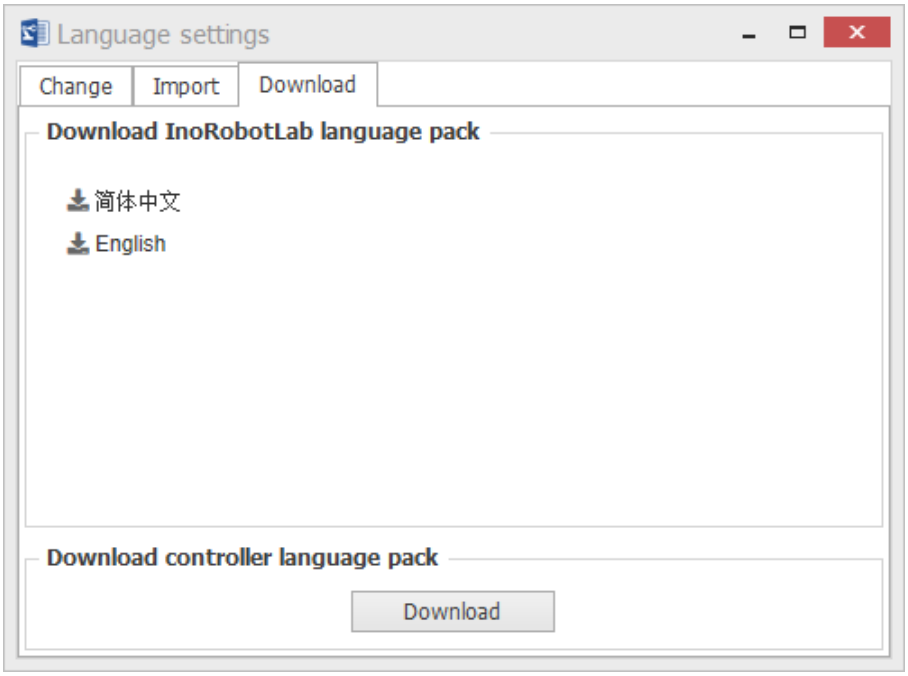


   5. Update controller cache: You can click the "Update controller cache" button to force the controller to re-read the latest language pack.

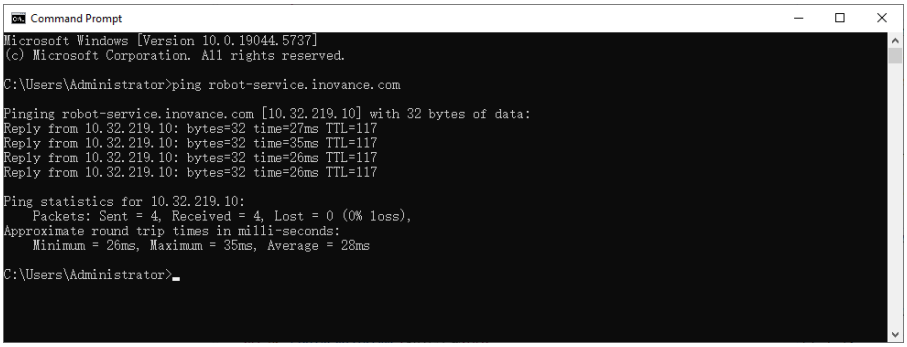


3. Synchronize the language pack from controller.
   - Synchronize the controller language pack to local system.
   - If a specific language pack is not available locally, read and write operations will proceed normally (e.g., if the Korean language pack exists in the controller but not in the InoRobotLab, the data will still be written locally).
   - If a specific language pack file already exists locally, it will be overwritten (e.g., if a Korean language pack exists in both the controller and the InoRobotLab, the local file will be overwritten).
   - If the language pack exists locally but not in the controller, no action is taken (e.g., if a Korean language pack exists in the InoRobotLab but not in the controller, it will be retained with no action taken).

13.2.5 Downloading the Language Pack



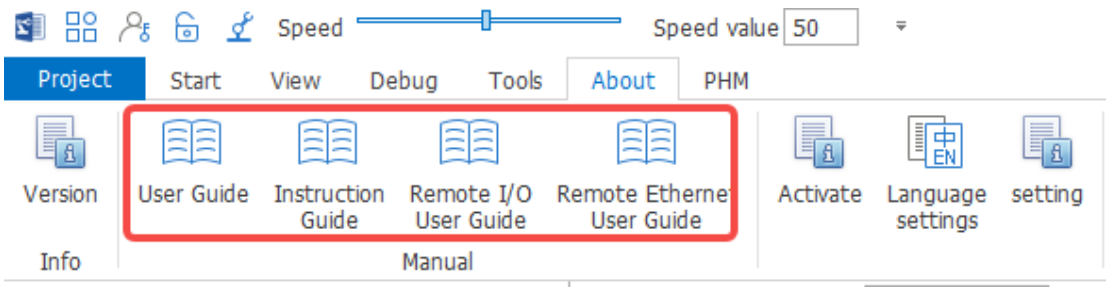
1. When robot-service.inovance.com is pingable, the download page is displayed; otherwise, the download page is automatically hidden:



2. Click the language pack link, and the browser will open automatically and start the download.
3. Download the InoRobotLab language pack separately based on language types.
4. The controller language pack does not distinguish between language types.

13.2.6 Multilingual Guide Translation

- The user guide, instruction guide, and remote control guide are available in both Chinese and English versions.
- When InoRobotLab is set to Chinese language, the corresponding Chinese guides are adopted.
- When InoRobotLab is set to the language other than Chinese, the English guides are adopted.



14 Virtual Controller

14.1 Virtual Controller Applications

14.1.1 Overview

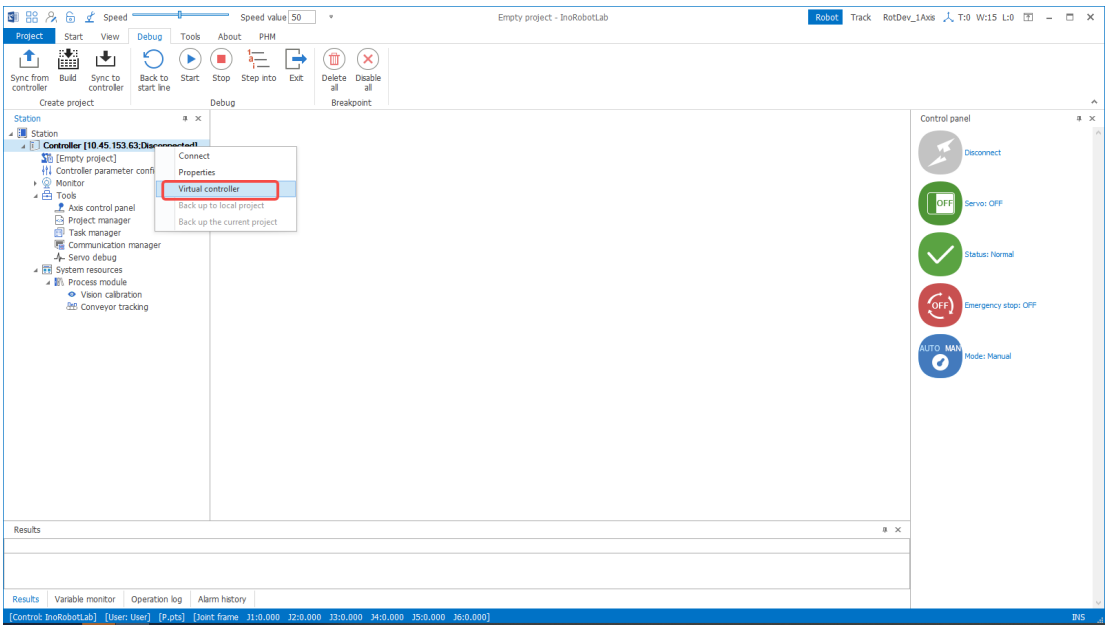
The robot virtual controller is a tool for simulating and managing robot operations, applicable for simulation and testing, development and debugging.

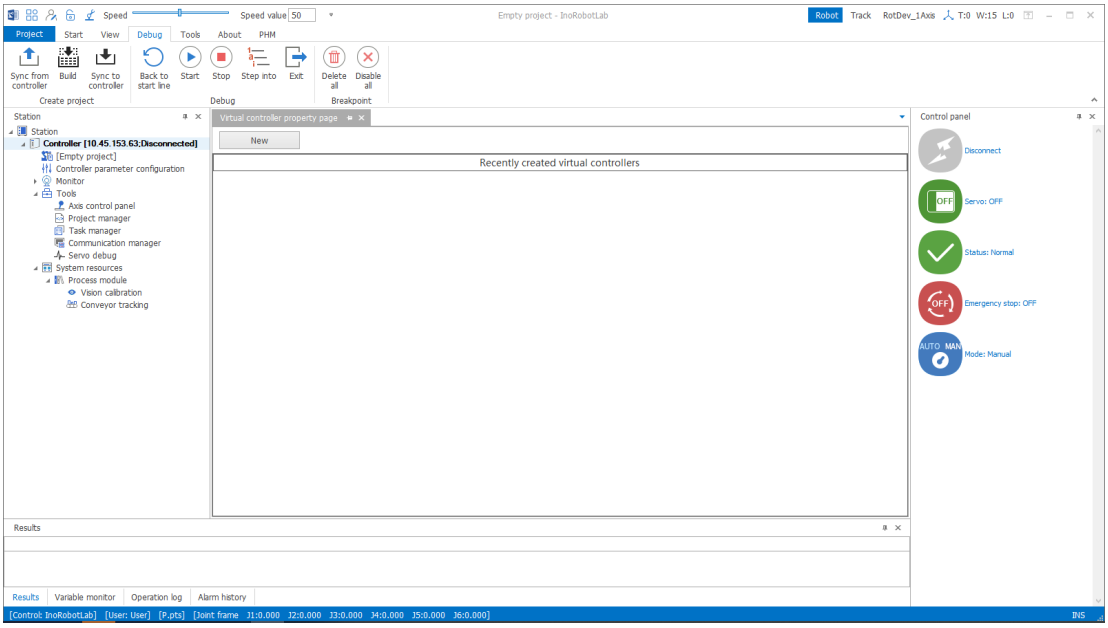
The virtual controller enables simulation and debugging independent of the actual controller environment, addressing the inability to pre-research production line conditions due to unavailable robot resources. Free from the hardware constraints of physical controllers, it allows for more flexible and convenient programming and debugging, improving project debugging and delivery efficiency.

14.2 Virtual Controller Operation

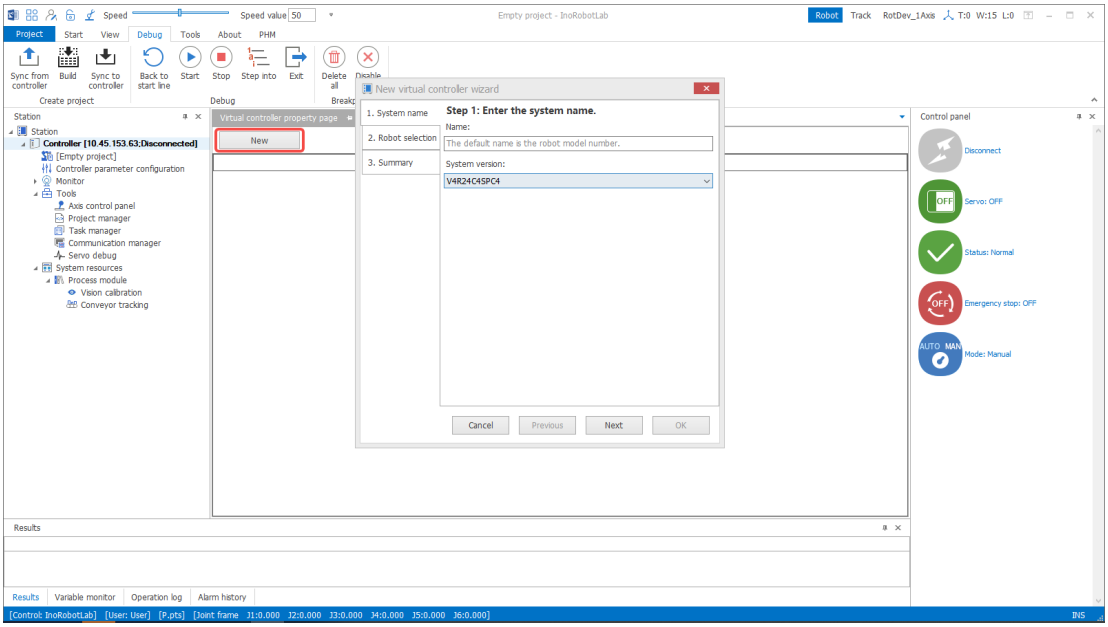
14.2.1 Accessing the Virtual Controller

After the software runs, select the controller tree node, right-click and select "Virtual controller" from the pop-up page to enter the virtual controller view. Note that the following example is creating a virtual controller for a 6-axis robot, demonstrating how to create a virtual controller system on the InoRobotLab software for use.



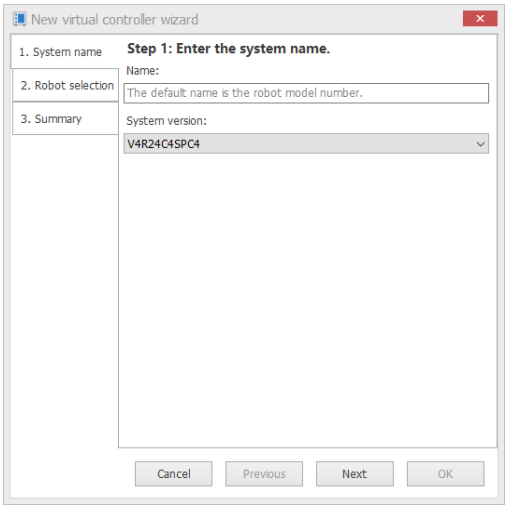


Click the "New" button to open the "New virtual controller wizard" dialog box.



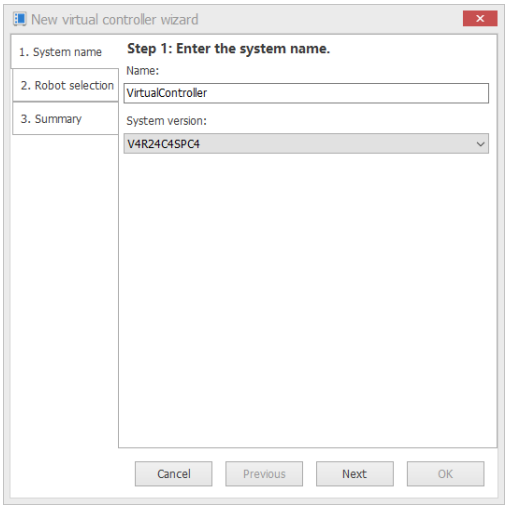
14.2.2 New Virtual Controller Wizard

The new virtual controller setup is divided into three steps: "System Name", "Robot Selection", and "Summary".



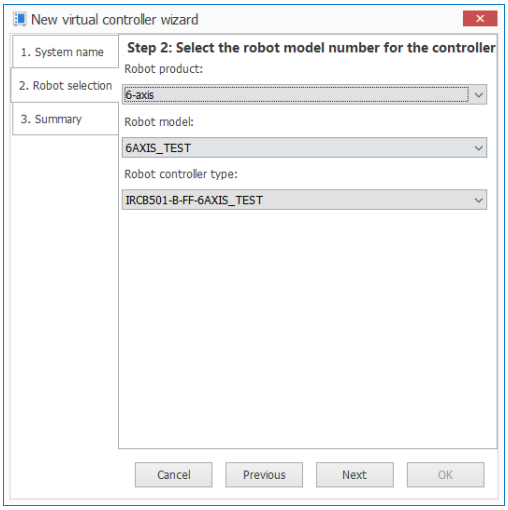
14.2.2.1 System Name

The system name is customizable, supporting only numbers and English characters, with a maximum of 36 characters. Note that if no custom system name is defined, it will default to the robot model name.



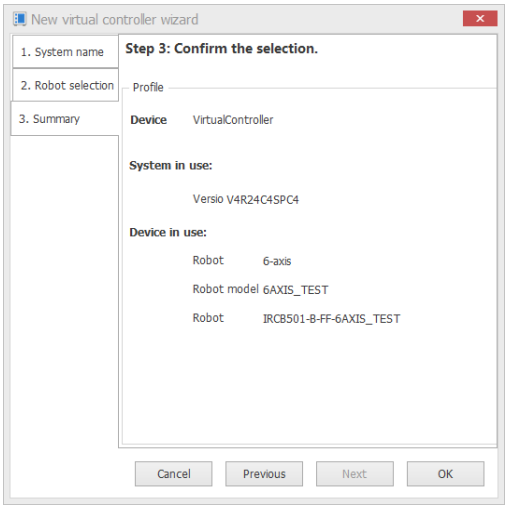
14.2.2.2 Robot Selection

You can select the robot product from 3 series (supporting 6-axis robot, floor-mounted and ceiling-mounted SCARA robots). After selection, all models of the current series product and the corresponding controller models are available for selection.



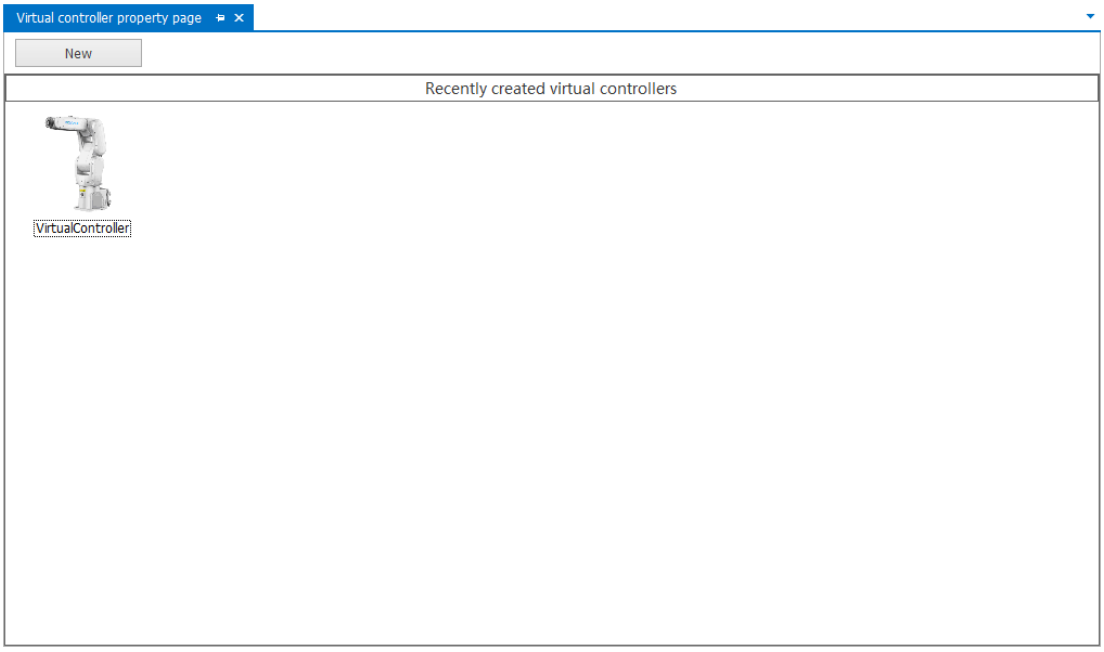
14.2.2.3 Summary

You can view the summary of the configured information. After confirming that everything is correct, click the "OK" button to create the virtual controller. Wait until the virtual controller device is successfully created.



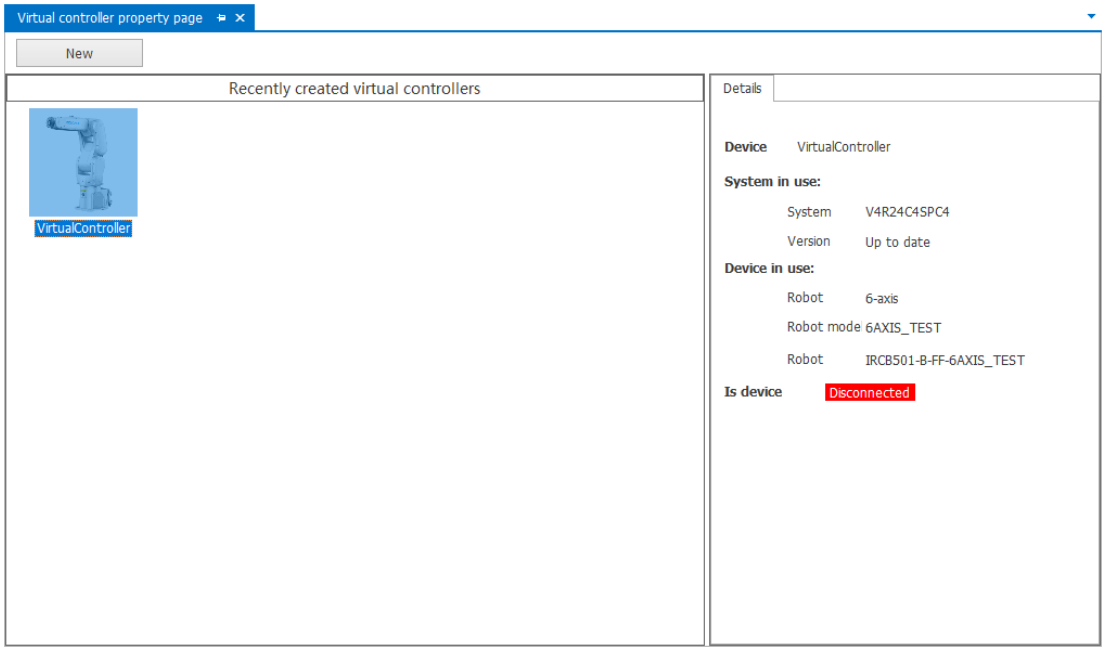
14.3 Recently Created Virtual Controller

You can view information about the recently created virtual controller.



14.3.1 Device Preview

You can view details of the selected device.

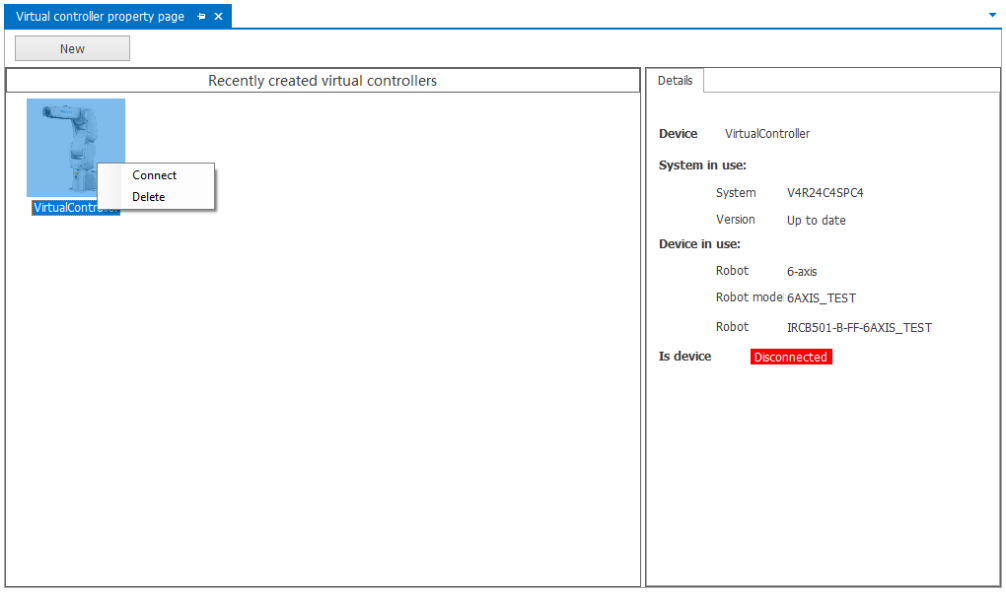


14.3.2 Device Connection and Disconnection

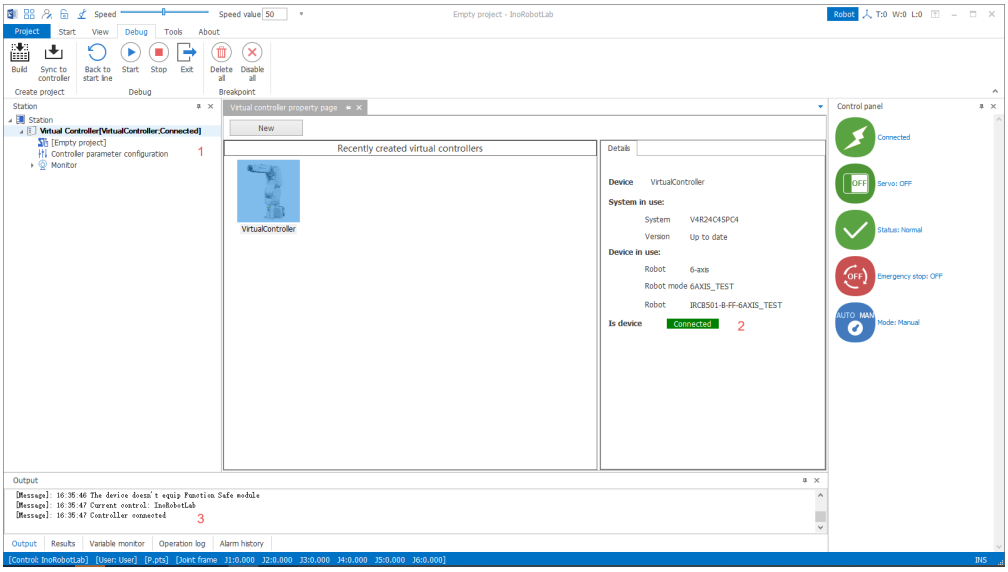
Select the device, right-click and select "Connect" to establish the virtual controller connection for the current device.

During the connection process, wait patiently for the virtual controller creation. It may take approximately 10s to 15s to complete the virtual controller system creation.

Currently, only one InoRobotLab software can connect to the virtual controller at a time. If another InoRobotLab attempts to connect while a virtual controller is already connected, a pop-up window will prompt whether to disconnect the already connected virtual controller.

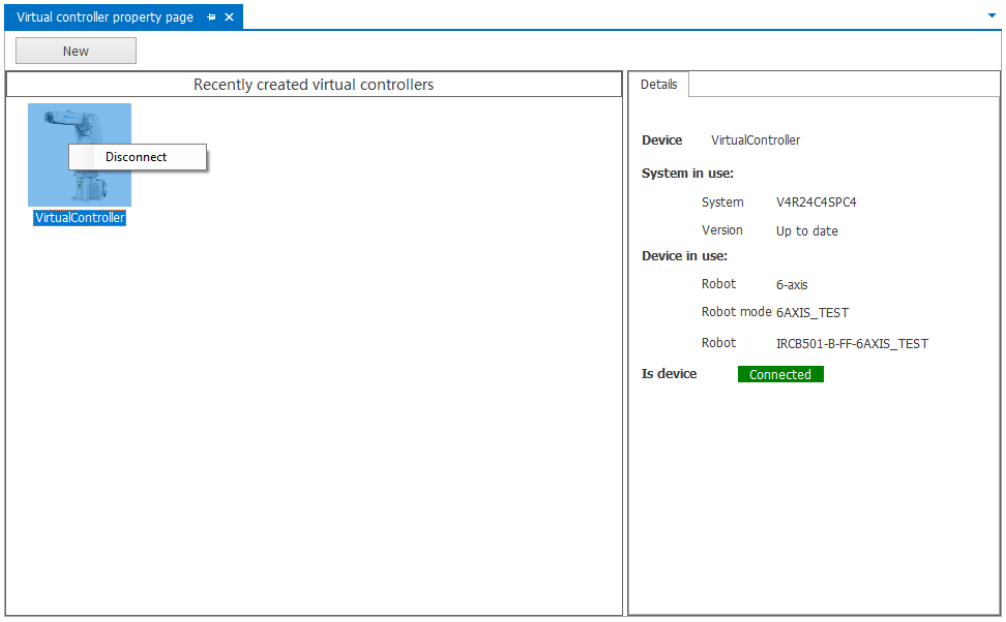


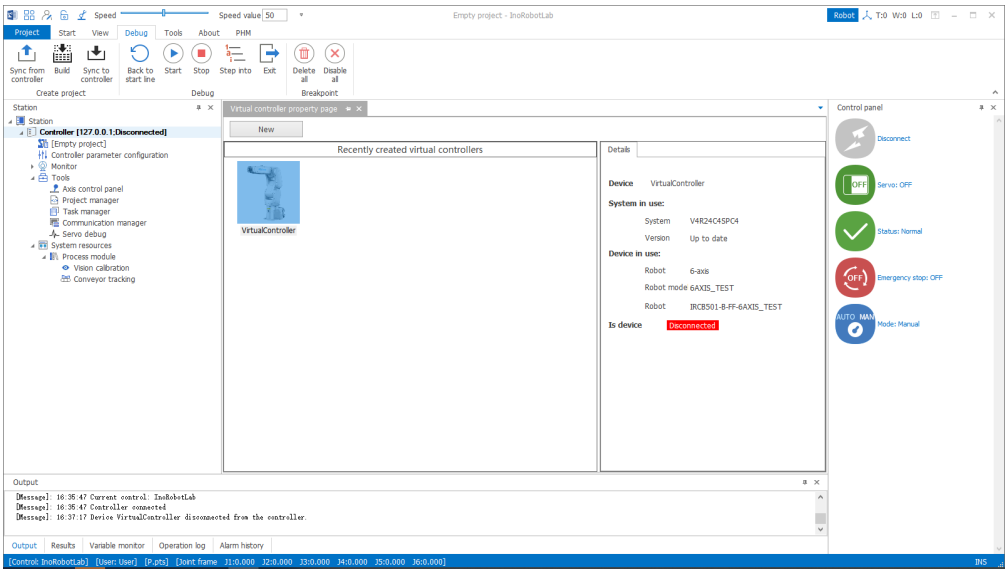
After successful connection, you can view "Virtual Controller [XXXX: Connected]" in the controller tree node. You can also check the device connection status by selecting the device on the virtual controller property page, indicating "Connected" highlighted in green or "Disconnected" in red. Additionally, you can view the output log information.



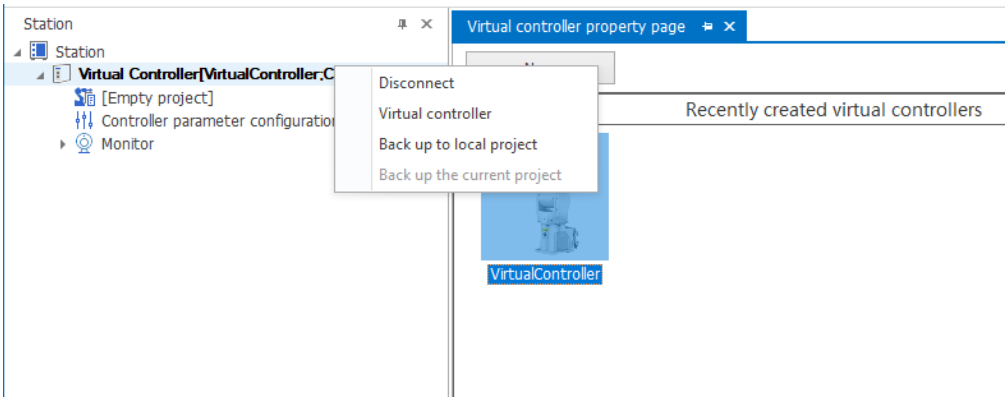
Note that once a connection operation is performed for one device, other virtual controller devices will be hidden to facilitate operation. Meanwhile, all functions under the current controller tree node belong to the currently active virtual controller environment. Therefore, you can operate the current device using the functions under the tree node as needed, such as program writing and debugging.

Click the "Disconnect" button to disconnect the current device and display other devices. For operations on other devices, repeat the above steps.





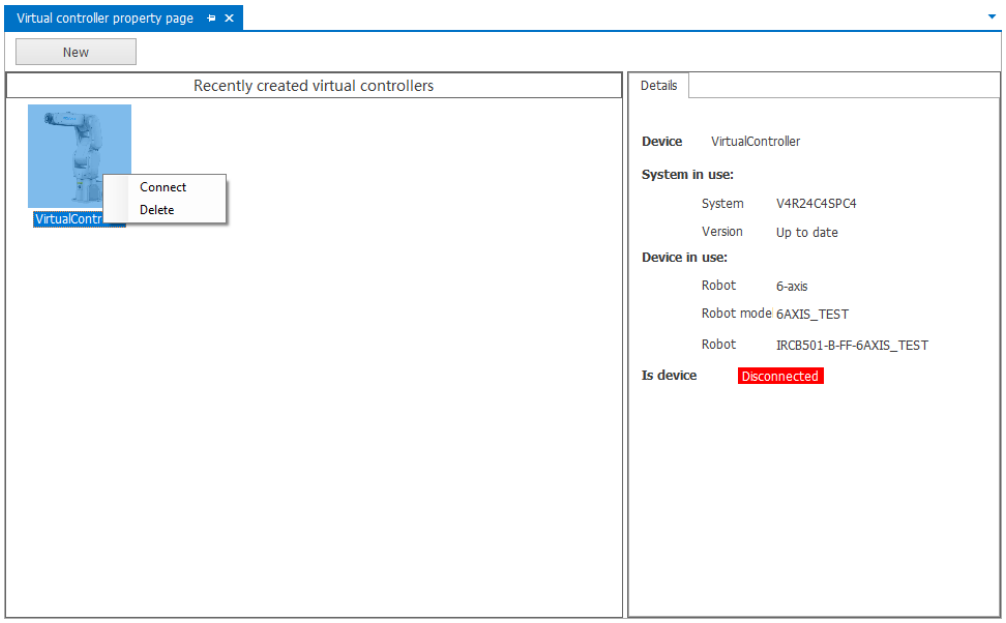
You can also disconnect the virtual controller under the Station.



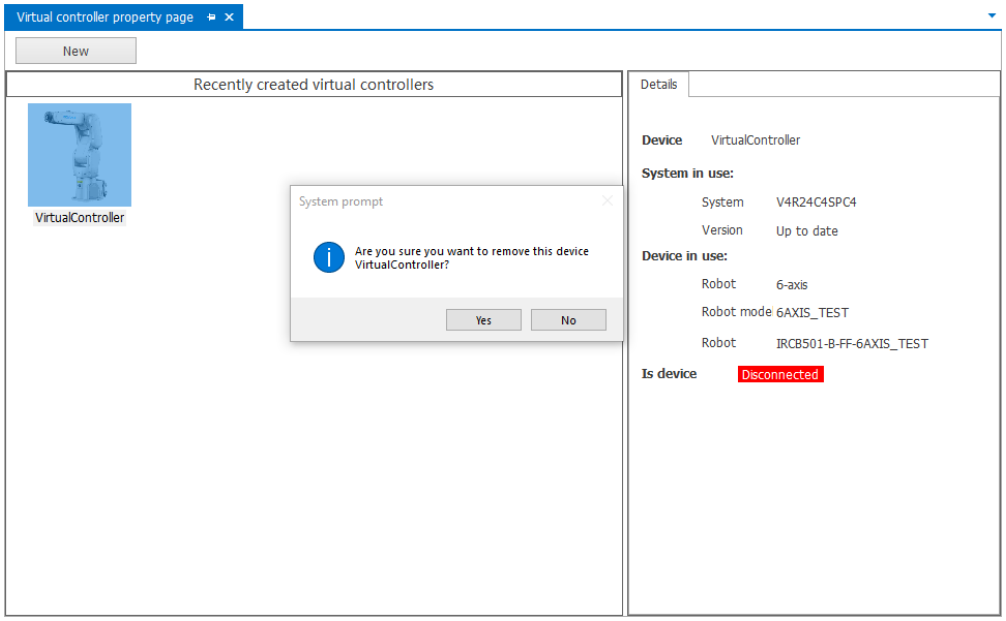
Note: If you close the "Virtual controller property page" and then click the virtual controller again to enter, it will display the state from when it was last closed.

14.3.3 Device Deletion

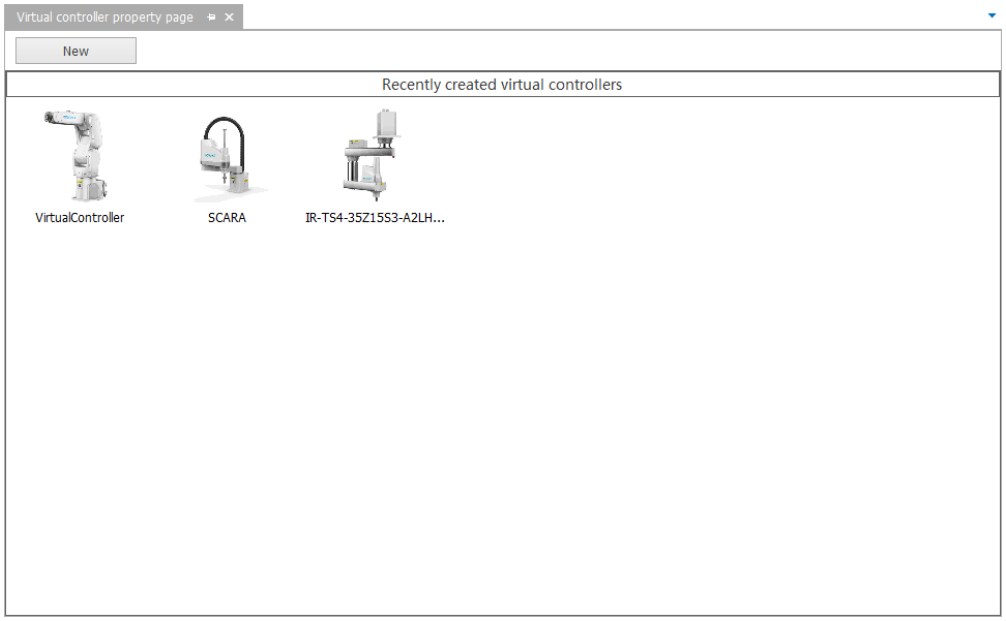
To delete a device, right-click on the selected device and select "Delete".



Note: Since deleting a device is a critical operation, a confirmation dialog "Are you sure you want to remove this device?" will appear. Click "Yes" to delete the device, otherwise no action will be taken.



This completes the entire process of creating a virtual controller on InoRobotLab. Note that if you need to create the virtual controller for other robots such as SCARA robot, repeat the above steps.



14.4 List of Supported Virtual Controller Functions and Instructions

14.4.1 List of Supported Functions

Function	Description	Sub-function	Remarks
Basic functions	Creation, connection, and deletion of virtual controllers	-	-
	Creation, deletion, closing, saving, and renaming of projects	-	-

Function	Description	Sub-function	Remarks
	Creation, saving, copying, and pasting of point files	-	-
	Permission switching	-	-
	Project management, task management, continuous execution in auto and manual modes (single step not supported)	-	-
	Information output: alarms, prompt messages	-	-
Axis control panel	Speed settings	-	-
	Mechanical unit switching	-	Added in V4R24C5
	Coordinated motion	-	Added in V4R24C5
	Motion mode switchover	-	-
	Move to point, Get current point	-	-
	Axis motion control	-	-
	Orientation calibration	-	Added in V4R24C5
	Homing	-	-
	Workobject, tool and load number switching	-	-
	Frame switching	-	-
	Display current position	-	-
Controller parameter configuration	Robot settings	Structural parameters	-
		Reduction ratio	-
		Coupling parameters	-
		Internal compensation parameters	-
	Zero settings	Absolute zero	-
		Work origin	-
	Installation parameters	Tool	Tool calibration (added in V4R24C5)

Function	Description	Sub-function	Remarks
		Workobject	Workobject calibration (added in V4R24C5)
		Load	_
		Arm parameter	_
		Installation form	Added in V4R24C5
	Peripheral configuration	Mechanical unit configuration (independent axis not supported)	Added in V4R24C5
	Motion parameters	Viewing angle parameter settings	_
		Run parameter settings	_
		Axis parameter settings	_
	Workspace	Axis limit settings	_
	Debug	System diagnostics	_
Monitor	I/O monitor	Standard I/O	_
		Fieldbus slave I/O	_
		Fieldbus master I/O	_
		Memory I/O	_
		System I/O	_
	Global variable monitor	_	_
	Work condition monitor	Current, Average load rate	_
	Query	_	_
Permission	Remote Ethernet	-	-
About (Interface)	Version	_	_
	User Guide	_	_
	Instruction Guide	_	_
	Remote I/O User Guide	_	_
	Remote Ethernet User Guide	_	_

Function	Description	Sub-function	Remarks
	Language settings	-	Added in V4R24C5

14.4.2 List of Supported Instructions

Instruction description	Instruction definition	Remarks
Variables	B/LB/R/LR/D/LD/BOOL/BYTE/INT/FLOAT/DOUBLE	-
	IN/OUT/INB/OUTB/INW/OUTW	-
	STR/String	-
	PR/LPR	-
	P/LP	-
	JP	-
	V/SPEED	-
	Tool	-
	Wobj	-
	Load	-
Arrays	-	-
Structs	-	-
Operators	-	-
Motion instruction	Movj	-
	Movl	-
	MovAbsJ	-
	Movc	-
	Jump	-
	JumpL	-
	Home	-
	ArmChange	-
Signal processing instruction	Set	-
	Get	-

Instruction description	Instruction definition	Remarks
	Wait	-
	Group	-
	Invert	-
System parameter instruction	SetAccRamp	-
	SlewMode	-
	Velset	-
	GetAlarmNo	-
	GetRunState	-
	SetSysToolNo	-
	SetSysWobjNo	-
	Clear	-
	GetTorque	-
	RapidMove	-
	SetAcc	-
	Speed	-
	SetCirOriMode	-
	SetAbortAccRamp	-
	SetPauseAccRamp	-
	PauseAbortMinAcc	-
Process control instruction	#Comment	-
	L-Goto	-
	Delay	-
	Include	-
	Func	-
	Ret	-
	If-Else-EndIf	-
	For-EndFor	-

Instruction description	Instruction definition	Remarks
	Switch-Case-Default-EndSwitch	-
	While-EndWhile	-
	Break	-
	Continue	-
	WaitInPos	-
Task control instruction	Xqt	-
	Halt	-
	Resume	-
	Quit	-
	Pause	-
File operation instruction	FileHandle	-
	Fopen	-
	FClose	-
	FRead	-
	FReadALine	-
	FWrite	-
	FWriteALine	-
	FRewind	-
	FTell	-
	FSeek	-
	FFlush	-
	Fcopy	-
	FRename	-
	FDelete	-
	FileExists	-
Folder operation instruction	FileDirHandle	-
	FOpenDir	-

Instruction description	Instruction definition	Remarks
	FCloseDir	-
	FReadDir	-
	FCreatDir	-
	FDeleteDir	-
Operation instructions > Numerical operations	Incr	-
	Decr	-
	Sin	-
	Cos	-
	Tan	-
	Asin	-
	Acos	-
	Atan	-
	Sqrt	-
	Pow	-
	Abs	-
	Max	-
	Min	-
	AngleToRad	-
	RadToAngle	-
Operation Instructions > String Operations	Left	-
	Right	-
	MId	-
	GetAt	-
	StrFindEnd	-
	StrRePlace	-
	StrReverse	-
	Strlen	-

Instruction description	Instruction definition	Remarks
	StrFind	-
	TrimLeft	-
	TrimRight	-
	Strcmp	-
	SPrintf	-
Operation Instructions > Value Conversion	Left	-
	Right	-
	MId	-
	GetAt	-
	StrFindEnd	-
	StrRePlace	-
	StrReverse	-
	Strlen	-
	StrFind	-
	TrimLeft	-
	TrimRight	-
	Strcmp	-
	SPrintf	-
Operation Instructions > Coordinate Operations	Dist	-
	GetCurPos	-
	GetCurJPos	-
	P[i]=	-
	OffsetT	-
	Offset	-
	Msft	-
	EOffsOn	-
	EOffsOff	-

Instruction description	Instruction definition	Remarks
	P=Offset	-
	PR Sum	-
	PR[i]=	-
	CalfixWobjPos	-
	CalRobholdWobjPos	-
	PToJ	-
	JToP	-
	CalculateTool	-
	CalculateWobj	-
Information interaction instruction	Alarm	-
	Print	-
	TimeStart	-
	TimeOut	-
Gravity load	GripLoad	-
Current protection	AvgCurLmt	-
	MaxTrqLmt	-

15 Predictive Maintenance

15.1 Overview

During the operation of robots at customer sites, sudden downtime may occur due to damage to components such as reducers and lead screws, affecting normal production.

To address the issue of sudden robot downtime at customer sites that impacts production capacity, this function enables data collection, status monitoring, and fault prediction for robots, providing alerts and warnings before actual failures occur.

15.2 Software and Hardware Conditions

Type	Description
Model	All models
Controller	IRCB501 series controller
Network structure	Master-slave network, consisting of a single InoRobotLab.exe and a single controller

15.3 Usage Workflow

1. Mark the motion instructions. Add marks to the motion instructions requiring data collection.
2. Configure the data collection. By configuring data collection, the data of marked motion instructions can be collected to the InoRobotLab software according to certain strategies.
3. Configure the fault warning. The algorithm will perform status monitoring and fault warning according to the configured fault warning method.
4. Monitor the faults. The current fault status will be displayed in graph form, and the related indicator values will be recorded.
5. Configure the continuous running. To run the fault monitoring function for an extended period, InoRobotLab.exe must remain running. To prevent InoRobotLab.exe from exiting due to misoperation, relevant settings can be configured.
6. Configure the automatic restart. If InoRobotLab is accidentally closed, causing monitoring interruption, the function can be quickly restored according to the original configuration when the InoRobotLab is restarted.

15.3.1 Motion Instruction Marking

Purpose:

To enable the collection of trajectory data corresponding to specific motion instructions by adding relevant markers to those instructions in the program file within the InoRobotLab software.

Requirement

- The modified motion instructions must be executed by the field program (or executed at a high frequency):
  A program file typically contains multiple cases, each with different motion instructions. The field program is usually controlled by a PLC and may enter different cases, thereby executing different motion instructions.

  Therefore, it is necessary to pre-identify the motion instructions that will definitely be executed (or frequently executed) during robot operation and add markers to these instructions.
- The actual motion trajectory corresponding to this instruction should have the largest possible operating range and operating speed.
- The modified motion instruction must follow a fixed motion trajectory: since motion instruction consists of points, speed, acceleration, and transition levels, ensure that each time the marked instruction is executed, the conditions such as start point, end point, speed, acceleration, transition level, and optimal trajectory on/off status remain consistent.
- When adding a datasample marker to a motion instruction with a transition (non-Z[0], non-Z[fine]), the following two conditions must be simultaneously satisfied: 1) The adjacent preceding line in the program must be a motion instruction; if it is not an adjacent line, the transition level of that motion instruction must be Z[0]/Z[fine]; 2) The adjacent following line in the program must be a motion instruction.

Method

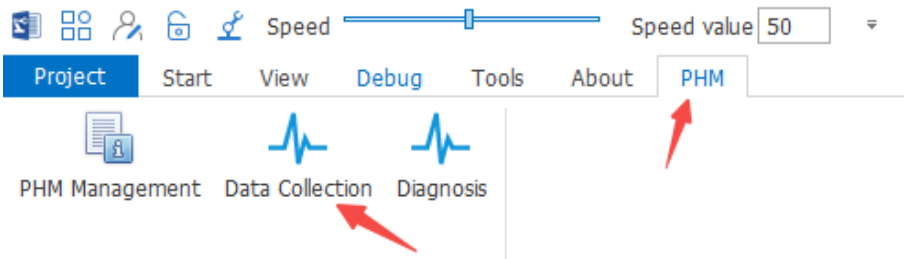
For the collected motion instructions, add DATASAMPLE[n] sequentially, where the allowable range for n is 1-64. Generally, marking 2 to 3 items is sufficient.

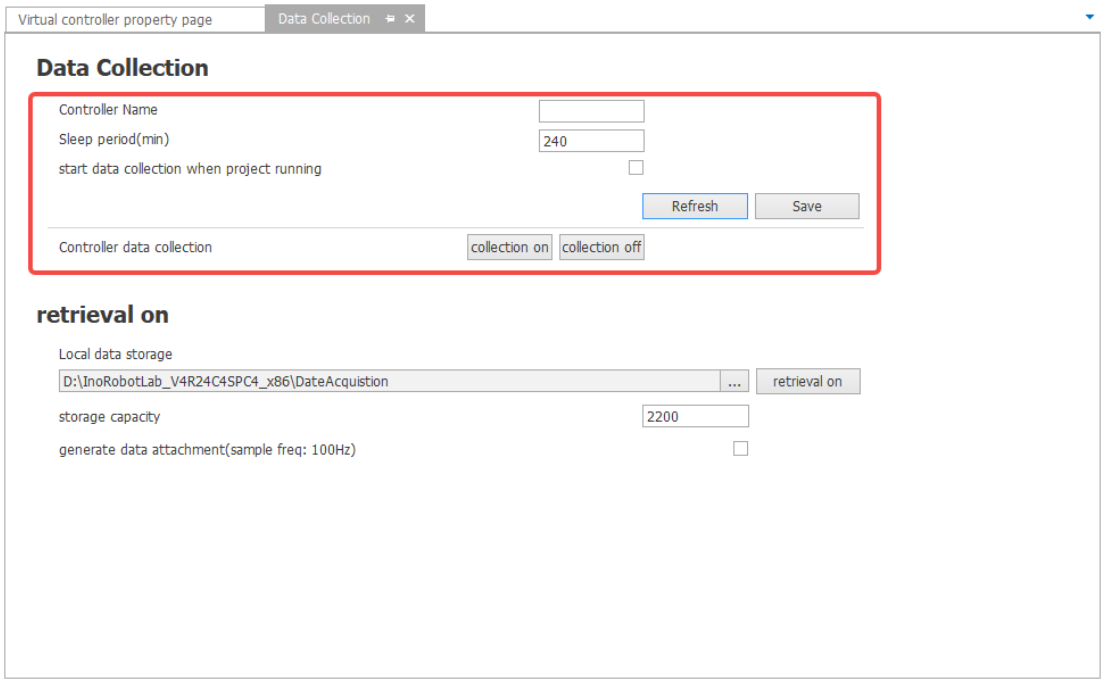
Example

In the following program, DATASAMPLE[1-3] are marked. When the instruction reaches DATASAMPLE[1], data collection will start. Subsequently, when the instruction reaches DATASAMPLE[2] and DATASAMPLE[3], data collection will continue respectively.

15.3.2 Data Collection Configuration

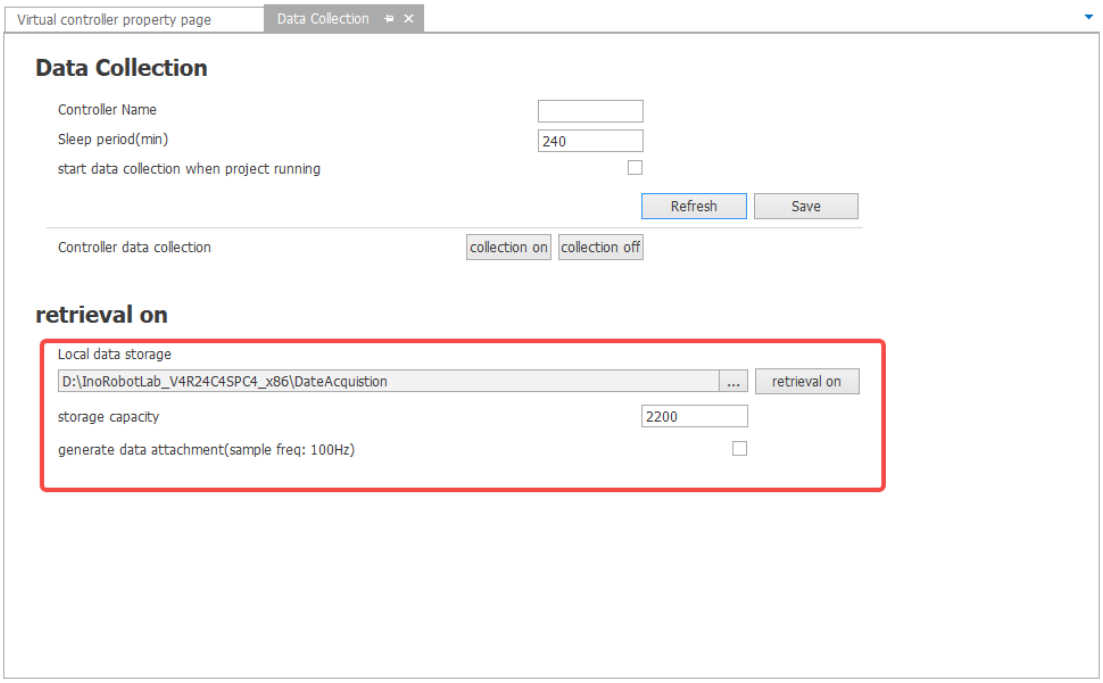
1. Before running the project, configure data collection first:





- Controller name: Set a name for this controller. It is recommended that the naming clearly characterizes the information of the controller, limited to 32 English characters.
- Sleep period (min): After every set period of time, data will be automatically collected and analyzed. It is recommended to set it to 20 min; or customize it based on the scenario (set a smaller value in the early stages to quickly accumulate data, and a larger valuer after stable algorithm tools are available).
- Start data collection when project running: Must be checked.
  Click the "Save" button to send this configuration to the controller.

2. Set the local data storage path and start data monitoring.



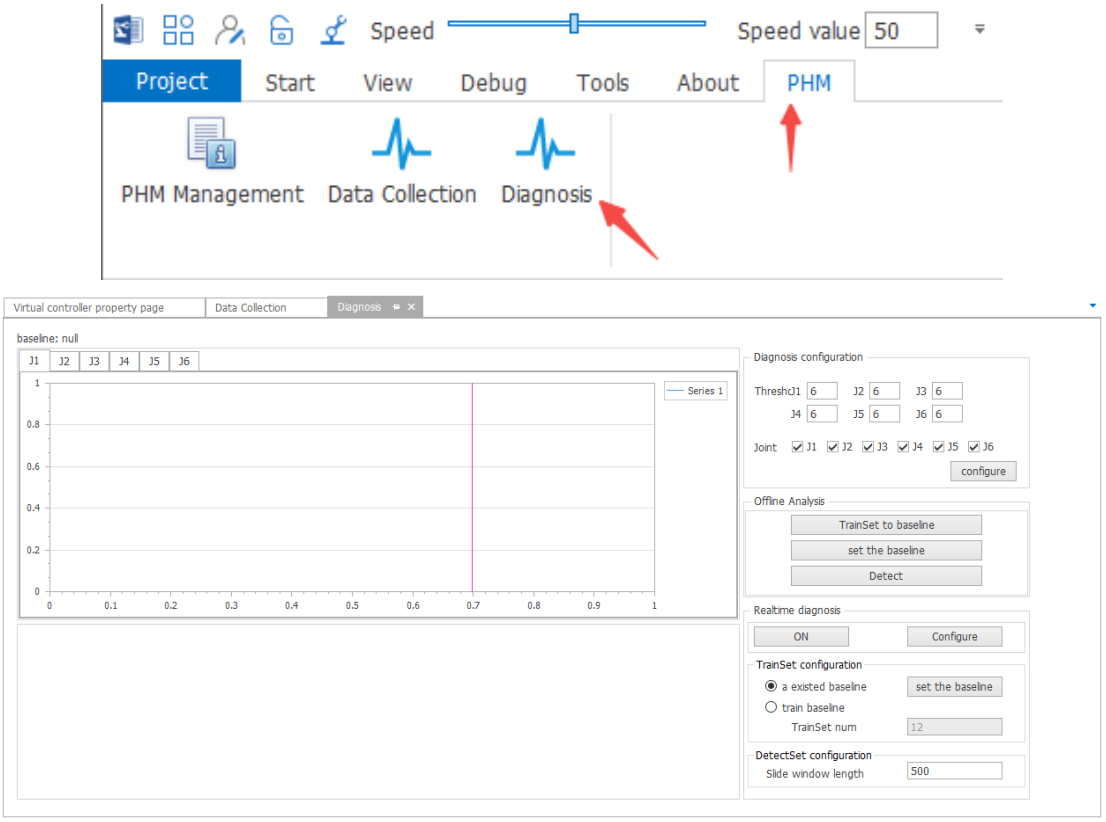
- Local data storage: The data files collected will be placed in the set path. The default path is not recommended. Check the storage status of each disk in the industrial personal computer, create a new folder dedicated to predictive maintenance operations, and configure the path to the "Local data storage".
- Storage capacity: Configure to a maximum of 2200.

- Generate data attachment (sample freq: 100 Hz): If checked, a sub-path/Attach will be added under the local data storage path to store plain text data. Check this option based on customer requirements.

  Click the "retrieval on" button to enable data monitoring; at this moment, wait for the onsite controller to start the project.

15.3.3 Fault Warning Configuration

Real-time monitoring configuration



The related parameters are described as follows:

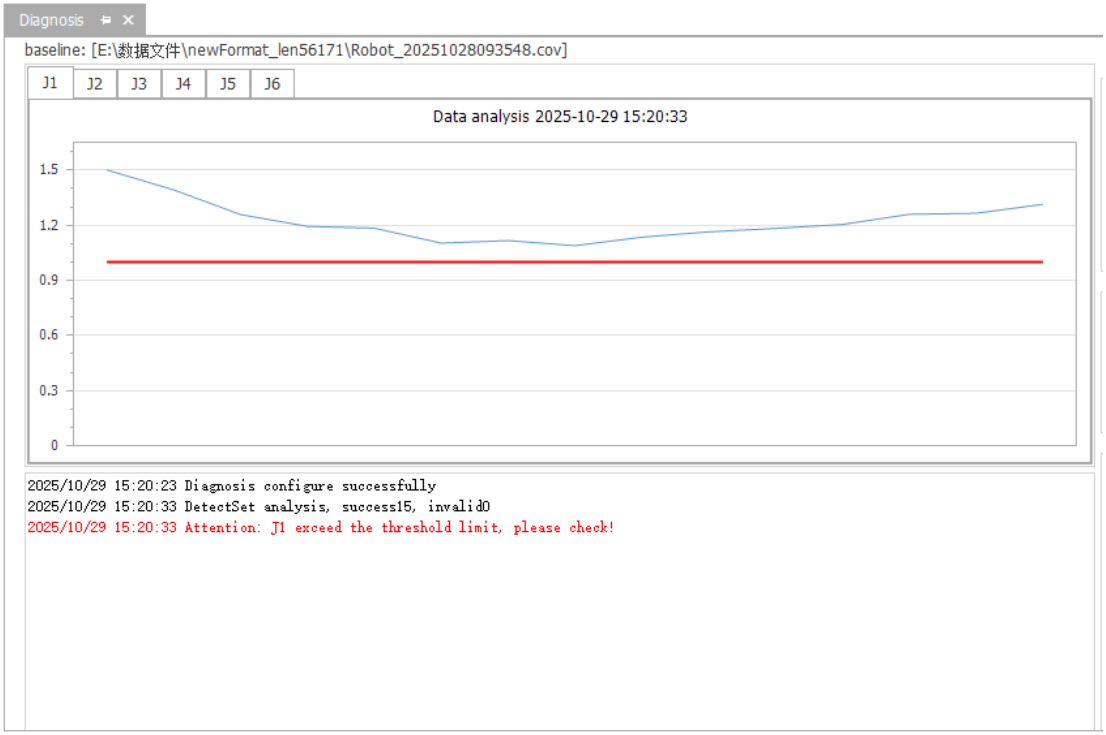
- Threshold: Use the default configuration.
- Joint: Select all axes according to the robot model.
- Click "Configure" to apply the data analysis configuration.
- Train baseline TrainSet number: 140 is recommended.
- Slide window length: 500 is recommended.

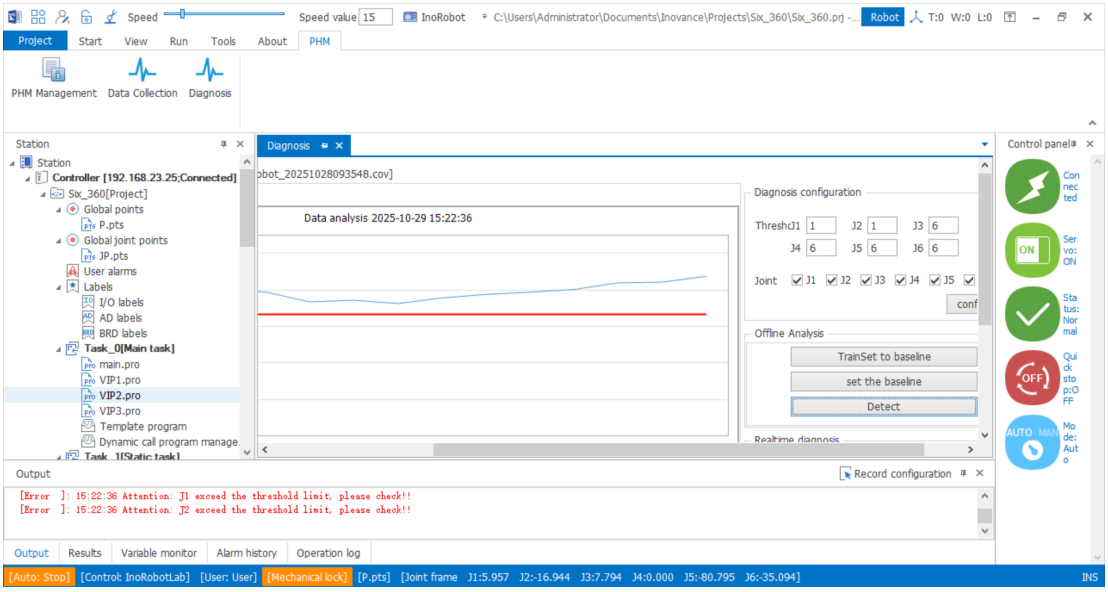
Click "Configure" to apply the real-time monitoring configuration.

Click "ON" to activate the real-time monitoring service. When the robot starts running, data will periodically be fed into the real-time monitoring service in conjunction with "Data Collection". The real-time monitoring service will provide algorithm analysis results periodically according to the above configuration.

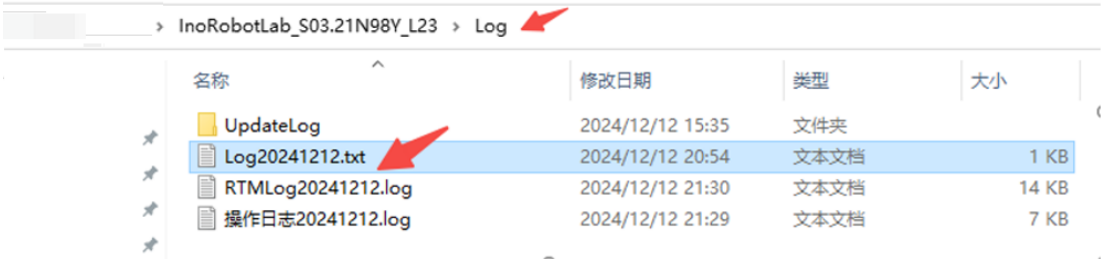
15.3.4 Fault Monitoring

If data analysis results exceed threshold limits, red warning alerts will appear in both the "Diagnosis" and "Output" interfaces. It is then recommended to check the operation status of the robot.

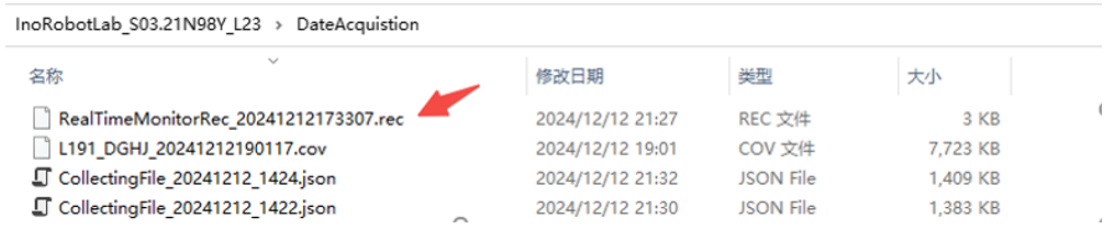


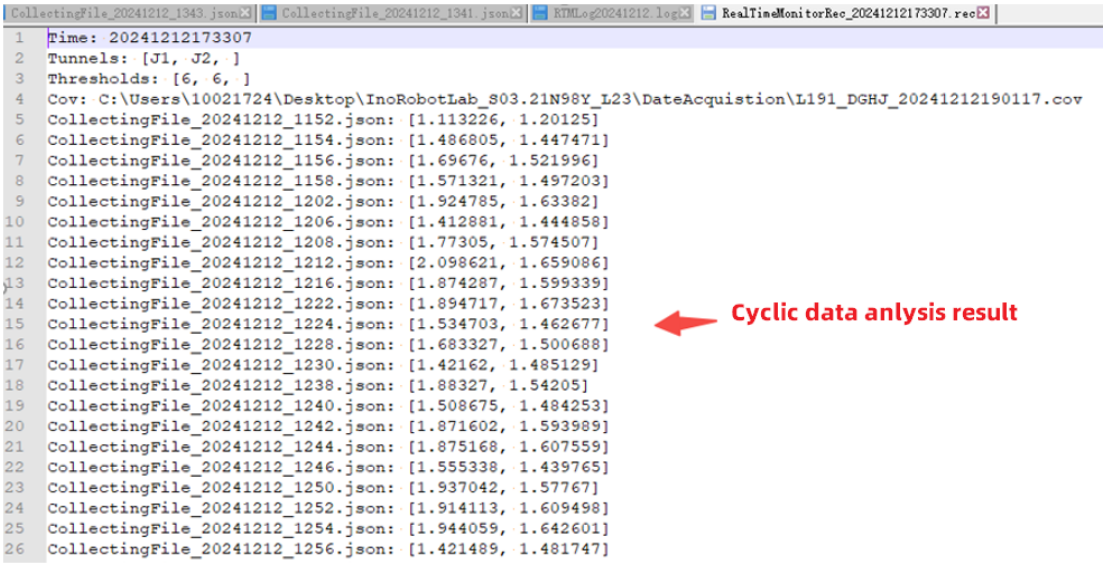


All historical announcements are recorded in InoRobotLab/Log/RTMLog.txt.



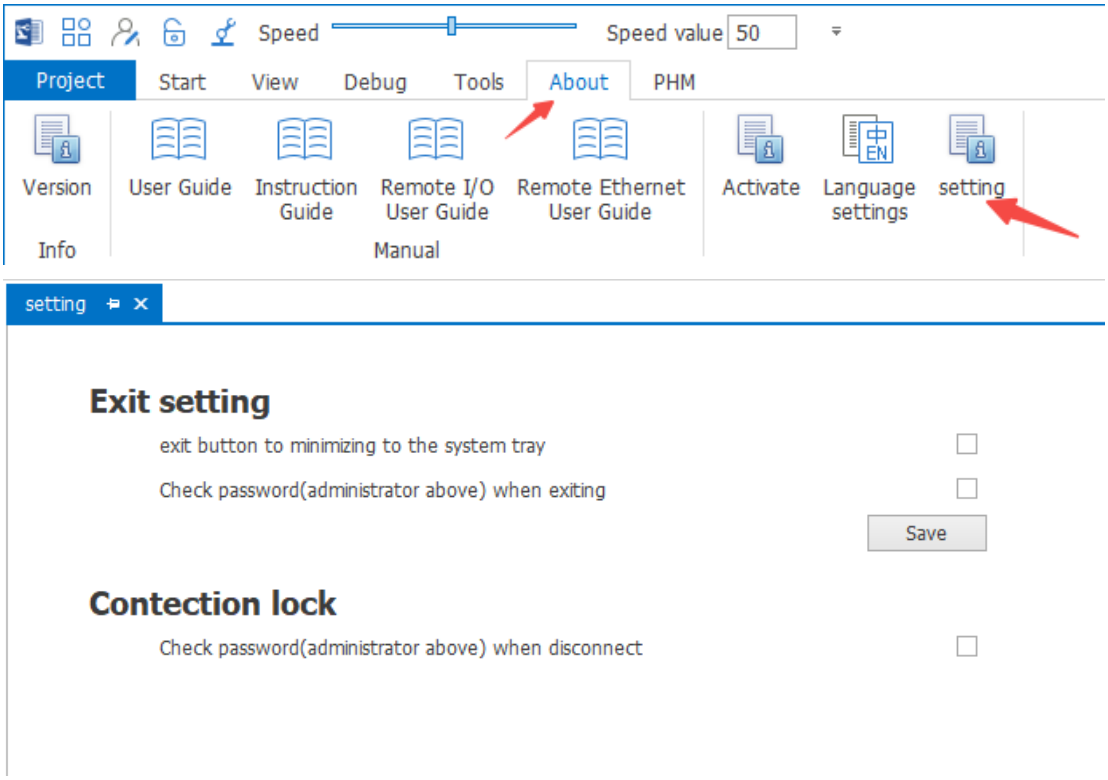
All real-time monitoring data analysis results are recorded in the .rec file in the data file path.



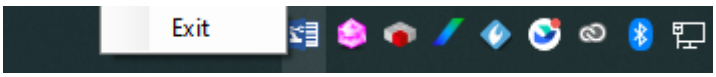


15.3.5 Configuration for Continuous Running of the Functions

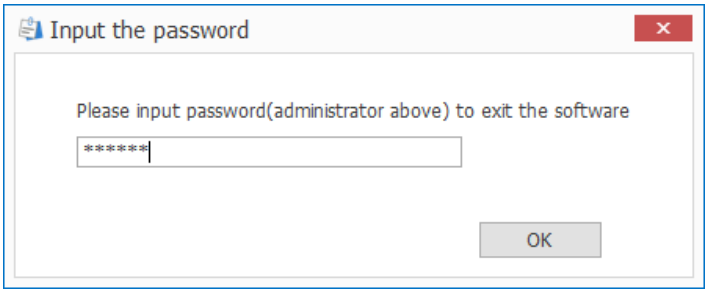
To run the real-time monitoring function for an extended period, InoRobotLab.exe must remain running. To prevent InoRobotLab.exe from exiting due to misoperation, relevant settings can be configured, as shown in the figure below.



- Exit button to minimizing to the system tray: When this option is checked, clicking "X" to close the software will not shut down InoRobotLab.exe but minimize it to the system tray. You can double-click the tray icon in Windows to restore the interface, or right-click to display the exit button and click "Exit" to exit the software.



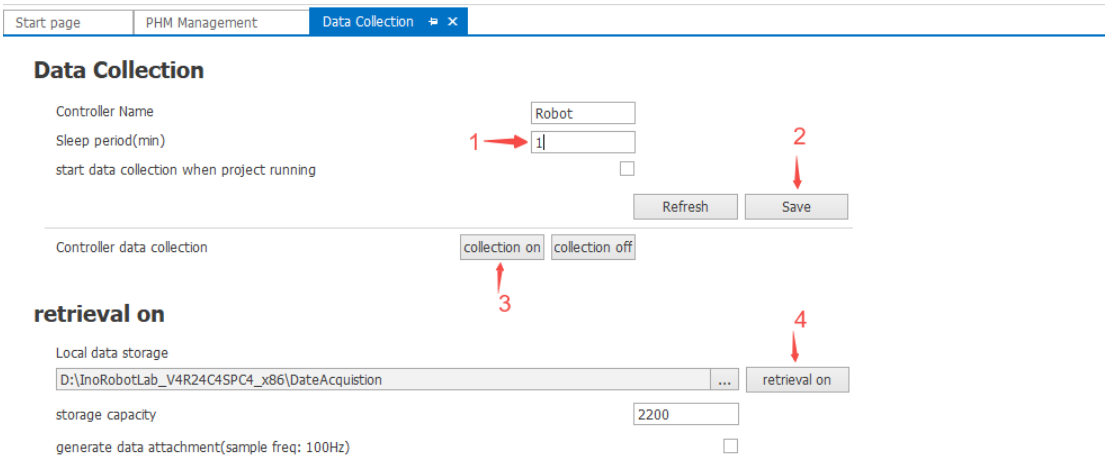
- Check password (administrator above) when exiting: When exiting the software with this option checked, you need to enter a password of admin-level or higher (or you are already at admin-level or higher permissions) to exit InoRobotLab.exe.

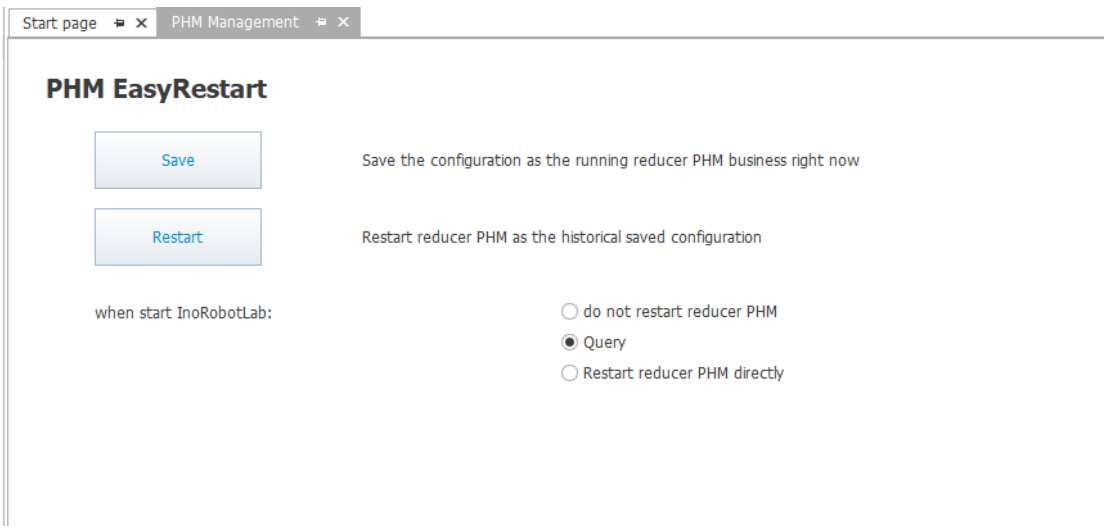


- Save: Click to write the above configuration to local. The configuration will be applied when InoRobotLab.exe is restarted.
- Check password (administrator above) when disconnect: When disconnecting from the controller with this option checked, you need to enter a password of admin-level or higher (or you are already at admin-level or higher permissions) to disconnect the controller. The configuration applies immediately after checking.

15.3.6 Configuration for Automatic Restart of the Functions

If the on-site industrial PC is shut down, the real-time monitoring service needs to be manually restarted. Theoretically, a "one-click" restart can be achieved:





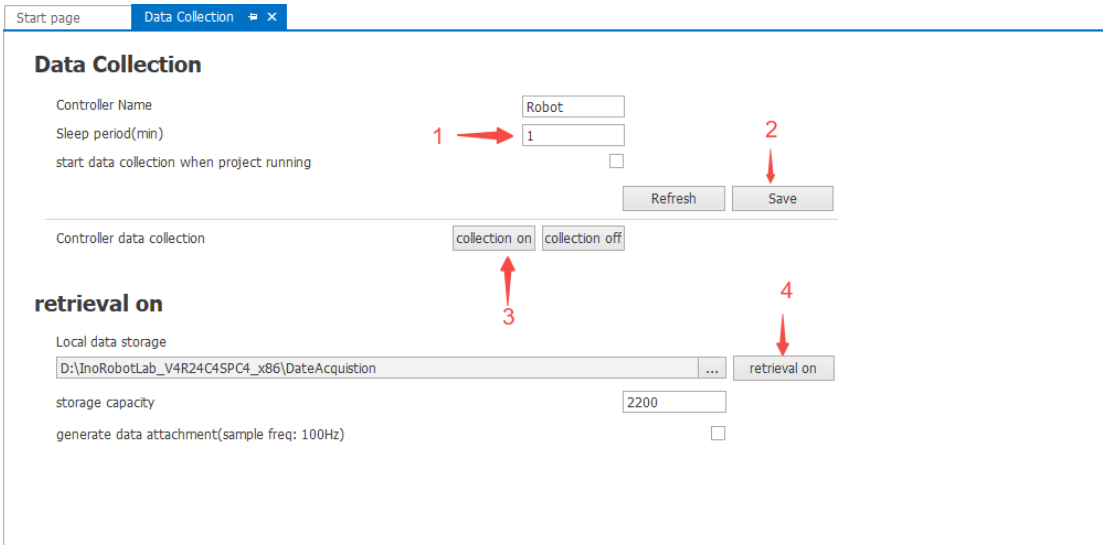
- Save: It is recommended to click this button after the deployment process is completed for the first time, to write several service fields of predictive maintenance into the local file ./EasyRestart/PDMRe.cfg.

- Restart: Click this button to start the predictive maintenance service according to the configuration in ./EasyRestart/PDMRe.cfg, including:
  - Connect to the target controller.
  - Determine if the project matches, and proceed if it does.
  - Determine if the data collection configuration matches, and proceed if it does.
  - Restart the data collection mode.
  - Start data collection.
  - Enable real-time monitoring.
- Prompt when restarting InoRobotLab: When this option is checked, a dialog box will appear upon opening InoRobotLab querying whether to restart the predictive maintenance service. If yes, the predictive maintenance service will be started according to the configuration in ./EasyRestart/PDMRe.cfg.

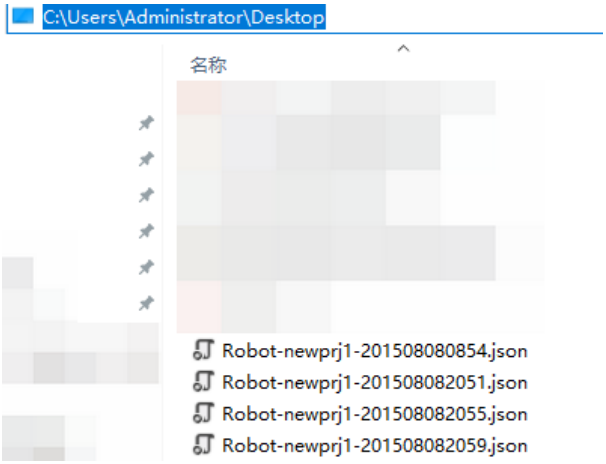
15.4 Quick Verification

You can quickly verify whether data can be normally collected from the controller by following the steps below:

1. Set the "Sleep period" of the controller to 1 min and click "Save". Then click the "collection on", and finally click "retrieval on".



2. Run the field project. Observe if there is relevant acquisition printout in the output bar and if new data files are obtained in the "Local data storage". If yes, it is considered that data can be normally collected from the controller.



3. If the verification is successful, restore the normal sleep period for data collection (a sleep period of 1 min would overload the controller). Click "retrieval off" and "collection off" to disable the data collection mode.
4. Modify the sleep period to an appropriate value and click "Save". Then click "collection on" to enable the data collection mode, and finally click "retrieval on".

16 Appendix

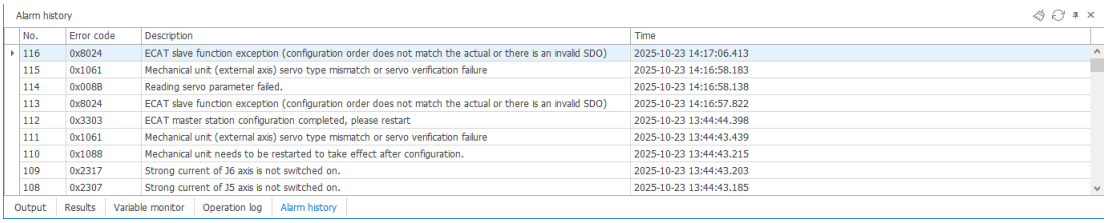
16.1 Historical Information Inquiry

16.1.1 Alarm History

Historical alarms are stored in the robot controller. You can clear and refresh the historical alarms.

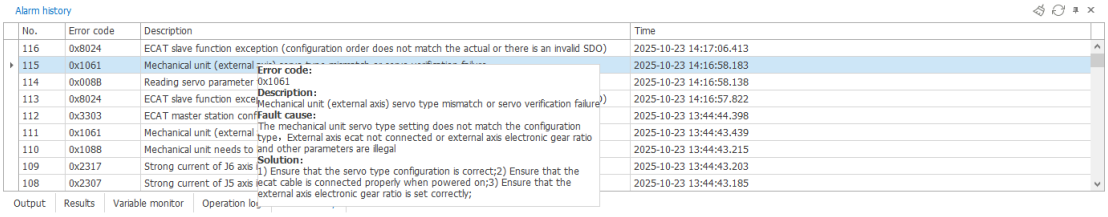
Steps

1. Find the alarm history interface at the bottom of the software, as shown below.



Alarm history

2. When InoRobotLab is connected to the controller, it reads the historical alarm file from the controller. You can clear the historical alarms displayed in the interface and stored in the controller by clicking the "Clear historical alarms" button in the upper right corner. When InoRobotLab is connected to the controller again, no history alarms will be displayed.
3. During use of the InoRobotLab software, the controller may generate some new alarms. These alarms are stored in the controller and are not displayed in the software interface in real time. To display these alarms in the software interface, click the "Refresh historical alarms" button in the upper right corner.
4. To view detailed information of a historical alarm such as cause and solution, double-click the alarm.



Results

- After clearing the historical alarms, close the software, open it and connect it to the controller again. If no historical alarms are displayed in the software interface, it means the clearing operation is successful.
- After an emergency stop alarm is triggered, the emergency stop alarm may not be displayed in the software interface. Click the "Refresh historical alarms" button and check whether the emergency stop alarm is displayed.

NOTE

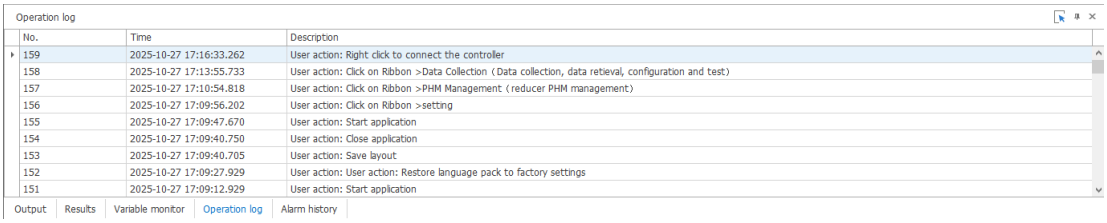
Only 1,000 historical alarm messages are retained.

16.1.2 Operation Log

The operation log keeps a record of the user's operations in InoRobotLab, which are saved in the local computer. You can select a date to view all the operations performed on that date.

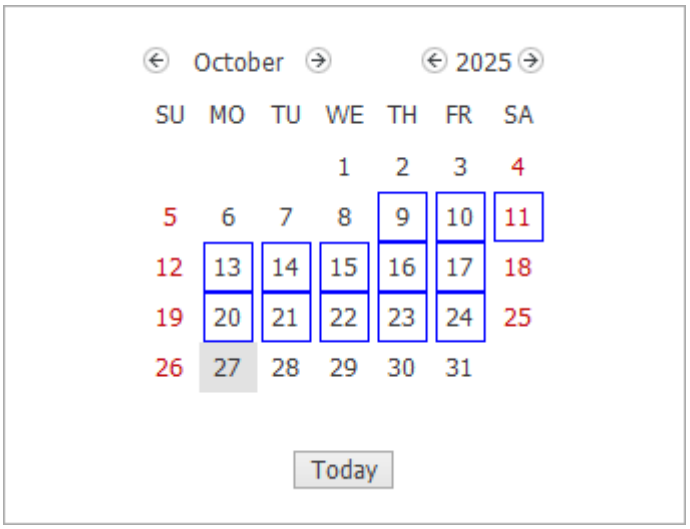
Steps

1. Find the operation log interface at the bottom of the software, as shown below.



Viewing the operation log of a certain day

2. Click the "Select history operation log" button in the upper right corner to open the calendar. The date with operation records is marked by a box.



Selecting date of operation log

3. Click any date (including the current date), all the operation information of the selected date will be refreshed in the interface.

Results

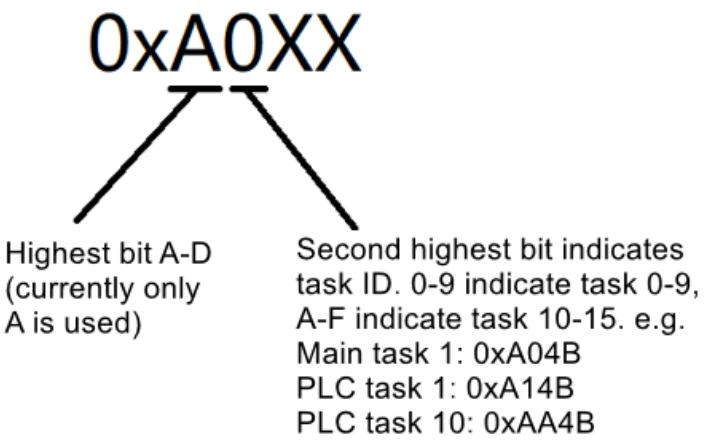
- Click a date marked with a box and operation records are displayed on the interface.
- Click a date not marked with a box and no operation records are displayed on the interface.
- Trigger an emergency stop. Operation information related to the emergency stop is added to the operation log.

> **NOTE**
>
> It is strongly recommended that you do not modify the operation log saved on the local computer.

16.2 Robot Alarms and Handling Method

Description of multitasking alarms

Multitasking alarms comply with the following rules.



Only the main task alarms are listed in the following alarm list.

Error code	Description	Cause	Solution
0x0001	Initialization failed (Reboot required after error is fixed)	1. PF file creation or opening failed. 2. PF file parsing failed.	Restart the controller and check if the error is fixed. If not, please contact the manufacturer.
0x0002	Teach pendant communication scheduling failed (Reboot required after error is fixed)	The teach pendant thread was not started properly.	Restart the controller and check if the error is fixed. If not, please contact the manufacturer.
0x0003	Vision communication scheduling failed (Reboot required after error is fixed)	The vision communication thread was not started properly.	Restart the controller and check if the error is fixed. If not, please contact the manufacturer.
0x0004	Internal communication scheduling failed (Reboot required after error is fixed)	The DSP communication thread was not started properly.	Restart the controller and check if the error is fixed. If not, please contact the manufacturer.

Error code	Description	Cause	Solution
0x0005	Automatic and manual function module scheduling failed (Reboot required after error is fixed)	The ARM scheduling thread was not started properly.	Restart the controller and check if the error is fixed. If not, please contact the manufacturer.
0x0006	Interpolation module scheduling failed (Reboot required after error is fixed)	The interpolation thread was not started properly.	Restart the controller and check if the error is fixed. If not, please contact the manufacturer.
0x0007	Opening EtherCAT communication failed (Reboot required after error is fixed)	1. Configuration file error. 2. EtherCAT slave does not match system configuration.	1. Check if the EtherCAT connection status is abnormal. 2. Contact the manufacturer.
0x0008	Opening parameter configuration file failed	1. Failed to open parameter configuration file. 2. Parameter configuration file was corrupted.	System configuration file error, please contact the manufacturer.
0x0009	Decoding error	Program syntax error	Check the program for syntax errors. For details, see the output bar.
0x000A	Program line invalid	The program line number sent by the teach pendant is out of range.	Check if the line where the blue cursor is located exceeds the program range or reselect the start line number.
0x000B	Wait instruction timeout	The wait time exceeds the set range.	1. Check the Wait condition. 2. Reset the wait time.
0x000C	Instruction reading error	1. The teaching program was corrupted; 2. The teaching program does not meet the specifications.	1. Rewrite the teaching program; 2. Check the teaching program.
0x000D	Nested call of subroutine not allowed	Nested call of subroutine occurs.	Check if the Call instruction is nested to subroutines.
0x000E	Motion instruction decoding error	Motion instruction decoding error	Check the teaching program.

Error code	Description	Cause	Solution
0x000F	Configuration file operation failed	1. Fail to save configuration file 2. Configuration file restore failed.	1. Restore to the factory defaults and power on again. 2. Power off and restart.
0x0011	Creating coordinate interpolation thread failed	1. Creating coordinate interpolation thread failed. 2. The internal test function is not open.	1. Replace the hardware. 2. Update the software version.
0x0012	Jump instruction failed	Point data calculation error exists in the Jump instruction.	Reselect a teaching point.
0x0013	IR-Link initialization failed (Reboot required after error is fixed)	1. The number of IR-Link slaves was configured incorrectly. 2. The sequence of IR-Link slaves was configured incorrectly.	Reconfirm the IR-Link configurations.
0x0014	Saving teaching program failed	The memory card is loose or not recognized.	Check the memory card.
0x0015	Motion module communication error	DSP software running error	Power on the controller again.
0x0016	Enable error	DSP software running error	Power on the controller again.
0x0017	Internal program running error	DSP software running error	Power on the controller again.
0x0018	Homing failed	Homing motion failed	Perform homing again.
0x0019	Enablement lost	The enablement of the running state is lost.	Check whether the system is in enabled state.
0x001A	Servo parameter error detected	The set parameters do not match the actual servo parameters.	Change servo parameters using operating panel or software. If the actual parameters do not match the factory defaults, replace the motor or servo.
0x001B	Error getting motion state	1. The system internal processing is busy. 2. The system motion firmware is damaged.	1. Power off and restart. 2. Contact the manufacturer.

Error code	Description	Cause	Solution
0x001C	The configuration failed, the PLC configuration is in conflict with the actual cabinet type.	The PLC configuration conflicts with the actual controller model.	Check that the PLC configuration parameters match the actual controller model.
0x001D	Some functions are affected because of old version of PLC configuration.	Old version PLC configuration	The secondary development version number and the robot secondary development version number do not match, please contact the manufacturer.
0x001E	Error verifying planned servo position and feedback position	The planned servo position and the encoder feedback position are inconsistent.	System fault, please contact the manufacturer.
0x001F	Position synchronization error (Reboot required after error is fixed)	Error in synchronizing position between controller and servo.	1. Power off and restart. 2. Contact the manufacturer.
0x0020	Robot not allowed to start when in motion.	The robot is moving when started.	Restart the robot after it is stopped.
0x0021	Parameter passing in error (Reboot required after error is fixed)	The parameters passed in are incorrect.	Check the basic system settings.
0x0022	Command line not found	The line number entered is out of range.	Select a command line that is running.
0x0023	Point data not found	No point is defined.	Check whether a point is defined.
0x0024	Inverse kinematics error	This point is a singular point of the robot.	Modify the coordinate of this point.
0x0025	Point data coordinate error	The frame values exceed the range.	Get a point again.
0x0026	Armtype in MOVL/ MOVC cannot be changed.	Arm parameter cannot be changed abruptly in MOVL, MOVC, and JUMPL instructions.	Get a point again or add a joint transition point.
0x0027	V parameter out of range	The V parameter exceeds the range (1-100).	Modify the V parameter.
0x0028	Z parameter out of range	The Z parameter exceeds the range (0-5).	Modify the Z parameter.

Error code	Description	Cause	Solution
0x0029	TOOL parameter out of range	The TOOL parameter exceeds the range (0-15).	Modify the Tool parameter
0x002A	Wobj parameter out of range	The Wobj parameter exceeds the range (0-15).	Modify the User parameter.
0x002B	ACC parameter out of range	The ACC parameter exceeds the range (1-100).	Modify the Acc parameter.
0x002C	until parameter out of range	The I/O number exceeds the range (0-255).	Modify the Until In parameter.
0x002D	Pallet parameter error	PNo,i,j and k of Pallet (PNo,i,j,k) are greater than or equal to 0.	Modify the Pallet parameter.
0x002E	Pallet not defined	The pallet number is not defined.	Define the pallet before use.
0x002F	Repeat parameter error	The Repeat parameter exceeds the range.	Modify the Repeat parameter.
0x0030	Running decoding error	Error in instruction parsing during running	Check the running instructions and modify them according to the prompts.
0x0031	Inverse kinematics calculation error in CART	The jog motion destination is out of the motion space or in singularity.	1. Check the stepsize of jogging. 2. Check the direction of jogging.
0x0032	Inverse kinematics calculation error in TOOL	The jog motion destination is out of the motion space or in singularity.	1. Check the stepsize of jogging. 2. Check the direction of jogging.
0x0033	Inverse kinematics calculation error in Wobj	The jog motion destination is out of the motion space or in singularity.	1. Check the stepsize of jogging. 2. Check the direction of jogging.
0x0034	Tool load parameter out of range	The value of Tool load parameter is out of range.	Modify the Tool load parameter.
0x0035	Load parameter out of range	The value of load parameter is out of range.	Modify the load parameter.
0x0036	Arm load parameter out of range	Arm load parameter out of range	Modify the arm load parameter.
0x0038	Parameter setting error	The parameter setting value is out of range.	Check that the set parameter meets the range.

Error code	Description	Cause	Solution
0x0039	API initialization failed (Reboot required after error is fixed)	The API communication thread was not started properly.	1. Power off and restart. 2. Contact the manufacturer.
0x003A	Authentication failure, system is not available.	Fail to authenticate the hardware encryption.	Contact the manufacturer.
0x003B	The MC module process does not exist.	The MC module process does not exist.	Update the servo firmware.
0x003C	Robot system parameter reset, please restart the controller.	The robot system parameters are reset (configuration file loading/factory settings restore/memory card formatting/system upgrade/ system degrade).	Power on the controller again.
0x003F	Exception initializing soft float parameters, please restart the controller.	The soft float parameter initialization is abnormal.	Restart the controller.
0x003d	Failed to initialize balancer parameters	Balancer parameter initialization failed on startup.	Import the correct balancer parameters.
0x0040	System diagnostics trigged	System diagnostics is trigged.	No action required, clear the message.
0x0048	DSP anomaly unresponsive after emergency stop pressed in running mode	Internal system exception occurred.	1. Power off and restart. 2. Contact the manufacturer.
0x0049	DSP anomaly unresponsive after emergency stop pressed in running mode	Internal system exception occurred.	1. Power off and restart. 2. Contact the manufacturer.
0x004B	Failed to mount memory card.	Failed to mount memory card.	1. Restart the controller. 2. Reinstall the memory card. 3. Replace the memory card.
0x004C	Duplicate port or IP address	Duplicate port or IP address	Change port or IP address.

Error code	Description	Cause	Solution
0x004D	Ethernet communication error	Ethernet communication error	Check the network connection and retransmit the data.
0x004E	Network is disturbed.	Multiple Ethernet terminals are connected to one controller.	Check whether multiple terminals are connected to the same controller.
0x004F	Failed to open network communication port	Opening network communication instructions failed.	Check the network communication instructions.
0x0050	Memory card not identified	1. The memory card is not inserted. 2. The memory card has poor contact. 3. The memory card is damaged. 4. The memory card has a power-on initialization issue.	1. Check the system hardware. 2. Power off and restart.
0x0051	EtherCAT disconnected	EtherCAT communication is disconnected.	Check EtherCAT communication.
0x0052	IR-Link disconnected	IR-Link is disconnected after connection.	Check IR-Link communication status.
0x0053	Too short interval between start and pause	The interval between start and pause is too short.	Restart operation.
0x0054	Failed to acquire vision feature values	Failed to acquire vision feature values while executing the instruction.	re-acquire vision feature values after checking the vision processing.
0x0055	Decoding not finished	The program may have instruction symbols or syntax errors.	Check the command syntax of the edited program.
0x0056	Robot type error	The robot has no associated type firmware.	Restart the system or check the DSP firmware.
0x0057	FPGA firmware loading error (Reboot required after error is fixed)	FPGA firmware error or startup exception	Restart the system or check the FPGA firmware.

Error code	Description	Cause	Solution
0x0058	DSP firmware loading error (Reboot required after error is fixed)	DSP firmware error or startup exception	Restart the system or check the DSP firmware.
0x0059	System error after mode change	After switching to the play mode, it was detected that the project was not properly compiled or the configuration file was not completed properly.	Internal system error, please contact the manufacturer.
0x005A	I/O parameter setting error	The I/O related parameters are set incorrectly.	Check that the I/O parameter range set in the instruction is reasonable.
0x005B	IP address error	Error getting or setting IP address	Check the network connection.
0x005C	No IR-Link configuration authority for teach pendant	Configuration authority is currently exclusive to the secondary development platform.	Use the current configuration, or revoke the authority from the secondary development platform.
0x005D	Shared memory mapping error	Shared memory mapping error between robot and PLC.	Contact the technical support.
0x005E	(User-defined alarm 0)	The system triggered a user-defined alarm 0.	Check the user alarm 0.
0x005F	(User-defined alarm 1)	The system triggered a user-defined alarm 1.	Check the user alarm 1.
0x0060	(User-defined alarm 2)	The system triggered a user-defined alarm 2.	Check the user alarm 2.
0x0061	(User-defined alarm 3)	The system triggered a user-defined alarm 3.	Check the user alarm 3.
0x0062	(User-defined alarm 4)	The system triggered a user-defined alarm 4.	Check the user alarm 4.
0x0063	(User-defined alarm 5)	The system triggered a user-defined alarm 5.	Check the user alarm 5.
0x0064	(User-defined alarm 6)	The system triggered a user-defined alarm 6.	Check the user alarm 6.
0x0065	(User-defined alarm 7)	The system triggered a user-defined alarm 7.	Check the user alarm 7.

Error code	Description	Cause	Solution
0x0066	(User-defined alarm 8)	The system triggered a user-defined alarm 8.	Check the user alarm 8.
0x0067	(User-defined alarm 9)	The system triggered a user-defined alarm 9.	Check the user alarm 9.
0x0068	(User-defined alarm 10)	The system triggered a user-defined alarm 10.	Check the user alarm 10.
0x0069	(User-defined alarm 11)	The system triggered a user-defined alarm 11.	Check the user alarm 11.
0x006A	(User-defined alarm 12)	The system triggered a user-defined alarm 12.	Check the user alarm 12.
0x006B	(User-defined alarm 13)	The system triggered a user-defined alarm 13.	Check the user alarm 13.
0x006C	(User-defined alarm 14)	The system triggered a user-defined alarm 14.	Check the user alarm 14.
0x006D	(User-defined alarm 15)	The system triggered a user-defined alarm 15.	Check the user alarm 15.
0x006E	Interference zone 0 alarm	The robot entered the interference zone 0.	Check the position of robot and the settings of interference zone 0.
0x006F	Interference zone 1 alarm	The robot entered the interference zone 1.	Check the position of robot and the settings of interference zone 1.
0x0070	Interference zone 2 alarm	The robot entered the interference zone 2.	Check the position of robot and the settings of interference zone 2.
0x0071	Interference zone 3 alarm	The robot entered the interference zone 3.	Check the position of robot and the settings of interference zone 3.
0x0072	Interference zone 4 alarm	The robot entered the interference zone 4.	Check the position of robot and the settings of interference zone 4.
0x0073	Interference zone 5 alarm	The robot entered the interference zone 5.	Check the position of robot and the settings of interference zone 5.
0x0074	Interference zone 6 alarm	The robot entered the interference zone 6.	Check the position of robot and the settings of interference zone 6.
0x0075	Interference zone 7 alarm	The robot entered the interference zone 7.	Check the position of robot and the settings of interference zone 7.
0x0076	Interference zone 8 alarm	The robot entered the interference zone 8.	Check the position of robot and the settings of interference zone 8.

Error code	Description	Cause	Solution
0x0077	Interference zone 9 alarm	The robot entered the interference zone 9.	Check the position of robot and the settings of interference zone 9.
0x0078	Interference zone 10 alarm	The robot entered the interference zone 10.	Check the position of robot and the settings of interference zone 10.
0x0079	Interference zone 11 alarm	The robot entered the interference zone 11.	Check the position of robot and the settings of interference zone 11.
0x007A	Interference zone 12 alarm	The robot entered the interference zone 12.	Check the position of robot and the settings of interference zone 12.
0x007B	Interference zone 13 alarm	The robot entered the interference zone 13.	Check the position of robot and the settings of interference zone 13.
0x007C	Interference zone 14 alarm	The robot entered the interference zone 14.	Check the position of robot and the settings of interference zone 14.
0x007D	Interference zone 15 alarm	The robot entered the interference zone 15.	Check the position of robot and the settings of interference zone 15.
0x007F	Data stream mode not closed	The system is in data stream mode.	Close the data stream mode.
0x0080	Emergency stop alarm	The emergency stop button was pressed.	Release the emergency stop and clear the alarm.
0x0081	No reverse movement data found	The reverse movement data has been executed.	Terminate the reverse movement.
0x0082	Port number closing error	The port number is not within the range or the port has not been opened.	Check the port number.
0x0083	TCP port overflow	There are excessive peripheral TCP application connections.	Close unused TCP connections.
0x0084	API communication processing error	The API channel is occupied by other applications for a long time or the previous API application processing is blocked due to fault.	Close or reduce the previous API application processes.
0x0085	Arc trajectory uncontrollable	The arc start point is uncertain.	Add other motion commands before and after the arc command.
0x0086	Version information mismatch, use software that matches the controller version	The version of the teach pendant or InoRobotLab does not match the version of the controller.	Use a teach pendant or InoRobotLab that matches the version of the controller.

Error code	Description	Cause	Solution
0x0089	IP address conflict	IP settings conflict.	Reconfigure IP.
0x008A	File system not identified in the memory card	The file system on the memory card is incorrect.	Reformat the memory card on the teach pendant.
0x008B	Reading servo parameter failed	The controller fails to read servo parameters.	Optimize the connection between the servo and the controller.
0x008C	Parameter out of range	Parameter out of range	Set the parameters within the range.
0x008D	Invalid I/O configuration	The configured I/O is controlled by a PLC or does not exist.	Reconfigure the I/O.
0x008E	Invalid I/O setting operation	The configured I/O lacks I/O control authority.	Check the I/O control authority.
0x008F	I/O does not exist	Failed to send screw locking startup parameters to the servo.	1. Check the electric screwdriver servo firmware matches with the controller. 2. Clear parameters and restart the system.
0x0090	Failed to start tightening	Failed to send screw locking startup parameters to the servo.	1. Check the electric screwdriver servo firmware matches with the controller. 2. Clear parameters and restart the system.
0x0091	Failed to stop electric screwdriver	Failed to read screw locking status from the servo.	1. Check the electric screwdriver servo firmware matches with the controller. 2. Clear parameters and restart the system.
0x0092	Failed to read screw tightening parameters	Failed to read screw locking setting parameters from the servo	1. Check the electric screwdriver servo firmware matches with the controller. 2. Clear parameters and restart the system.
0x0093	Failed to read screw tightening parameters	Failed to read screw locking setting parameters from the servo	1. Check the electric screwdriver servo firmware matches with the controller. 2. Clear parameters and restart the system.

Error code	Description	Cause	Solution
0x0094	Failed to write screw tightening parameters	Failed to write screw locking setting parameters to the servo.	1. Check the electric screwdriver servo firmware matches with the controller. 2. Clear parameters and restart the system.
0x0095	Screw data display failed	Failed to obtain screw locking display data from the servo	1. Check the electric screwdriver servo firmware matches with the controller. 2. Clear parameters and restart the system.
0x0096	Failed to reset screw tightening count	Failed to write a screw tightening counter clearing flag to the servo.	1. Check the electric screwdriver servo firmware matches with the controller. 2. Clear parameters and restart the system.
0x0097	Failed to get thread for servo error	Failed to get thread for servo error	Restart the robot.
0x0098	Failed to start loosening	Failed to send screw removal startup parameters to the servo	1. Check the electric screwdriver servo firmware matches with the controller. 2. Clear parameters and restart the system.
0x0099	Failed to write screw loosening parameters	Failed to write screw removal setting parameters to the servo.	Check the electric screwdriver servo firmware matches with the controller.
0x009A	Failed to read screw loosening parameters	Failed to obtain screw removal setting data from the servo.	Check the electric screwdriver servo firmware matches with the controller.
0x009B	Fallback point setting out of range	Fallback point setting out of range	Set the fallback point for screw loosening within the range.
0x009C	Failed to initialize tracking process data	There are errors in the tracking process data or the system is busy.	Check for parameter errors.
0x009D	Interrupt operation ended. Failed to restore the original task trajectory	The restoration of original task trajectory was abnormally terminated, causing the restoration to fail.	Reset the start line/return to the start line or resynchronize the projects.

Error code	Description	Cause	Solution
0x009E	Interrupt queue overflow	The interrupt queue has been saturated by pending interrupt tasks.	1. Check if the interrupt triggers occur too rapidly, overwhelming task processing capacity. 2. Return to the start line or synchronize the project to clear the interrupt tasks.
0x009F	Interrupt exit position deviation	Failed to restore to the pre-interrupt waypoint upon exit, due to missing restoration instruction execution.	Restore to the pre-interrupt waypoint during interrupt exit sequence.
0x00A0	Read-write TCP port not open	The TCP port is not open.	Ensure the TCP port is properly connected.
0x00A1	Dynamic network communication port not closed	Standard network communication is used while dynamic network communication is not closed.	Use the command CNVVIOSION OFF to close the dynamic network communication.
0x00A2	Conveyor error or camera pixel error	The conveyor number is incorrect or the transfer data type of the camera is incorrect.	Reset the conveyor number or camera data transfer type.
0x00A3	Dynamic vision coordinates conversion error	Error converting dynamic vision pixels to camera frame.	Check that the pixels sent from vision equipment are correct, otherwise the camera needs to be recalibrated.
0x00A4	No reverse movement data found	There is no reverse movement data.	Clear the error.
0x00A5	Lack of motion instructions before arc instruction	The command before arc motion is not Movj or Movl. (This appears in only the single-step teaching.)	Ensure that the command before the arc command is not Movj or Movl.
0x00A6	Specified file not found	The file does not exist or the file path is incorrect.	Check that the file exists or the file path is correct.
0x00A7	Error saving diagnostic information	The relevant files do not exist during the process of saving diagnostic information.	Internal system error, please contact the manufacturer.
0x00A8	Error exporting diagnostic information	The relevant files do not exist during the process of exporting diagnostic information.	Check that the USB device is normal; otherwise it is an internal system error, please contact the manufacturer.

Error code	Description	Cause	Solution
0x00A9	Too many files or insufficient memory card space	1. There are too many files (including folders) in the /media/sdcard1/ folder. 2. There is no insufficient remaining space on the SD Card.	1. Delete excessive files. Ensure the number of files+folders is not greater than 100. 2. Clean up the SD card.
0x00AA	Pause buffer data sending failed	Failed to send buffer data when going from pause to start.	Restart from stop.
0x00AB	Multitasking number does not exist.	Multitasking number is written incorrectly or in the wrong range.	Check the task number.
0x00AC	Multitasking number incorrect or occupied	The multitasking number is incorrect or has been occupied.	Check the task number.
0x00AD	Multitasking creation failed	1. The system is busy. 2. The task is already running or has been restarted. 3. Task 0 was started under non-remote I/O permissions.	1. Re-run the XQT command to check that it runs normally. 2. Stop the current multitasking and start it again. 3. Switch to remote I/O permissions and restart task 0.
0x00AE	Failed to open or close the position latching function	The hardware is not connected or the servo is not ready.	Check the hardware connections and servo configuration.
0x00AF	Failed to call the controller debug function	The controller debug function module failed.	Check that the configuration of the controller debug switch is correct.
0x00B0	Failed to set load parameters	The system is busy or the parameter is out of range.	Reset or check the parameter.
0x00B1	Serial port or baud rate error	1. The input serial port number is out of range. 2. With the serial port not closed, another serial port with the same number but different baud rate is opened.	1. Check that the input serial port number is within the range. 2. Close the serial port via the close instruction and open a serial port with new baud rate via the close instruction.
0x00B2	Modbus parameter read-write error	Modbus read-write execution failed	Update the software version.

Error code	Description	Cause	Solution
0x00B3	Failed to modify P variable	Failed to open or write the program file where the P variable is located	Re-modify it or update the software version.
0x00B4	Abnormal network communication: Client and server connection error	The communication between the robot and the target device has been interrupted.	Re-establish the connection.
0x00B5	Abnormal network communication: Robot sending data error	The network was disconnected while the robot was sending data or there was an error while transmitting data.	Check that the network connection is normal
0x00B6	Orientation correction calculation error	Wrong tool was selected or the alignment position is a singularity.	1. Check that the tool selection is correct. 2. Check that the position that the robot needs to align with is not a singularity.
0x00B7	Error in switching station program in Modbus mode	In Modbus mode, the station program is switched when the current station program is not stopped.	Set the current station program to the stop state, and then switch the station program.
0x00B8	Supply voltage error	The system suddenly loses power during operation.	Clear the alarm.
0x00B9	Illegal InoRobShop axis configuration	The number or type of axes configured by InoRobShop does not match the actual condition.	Reconfigure the correct axis in InRobShop. CAUTION The built-in PLC program will also be lost.
0x00BA	Error selecting multiple programs in Modbus mode	In Modbus mode, multiple station programs are selected simultaneously.	Select only one program.

Error code	Description	Cause	Solution
0x00BB	Network disconnected during motion in teach mode under teach pendant control.	When the robot is under the control of teach pendant and operating in teach mode, it is detected that the network connection between the teach pendant and the controller is disconnected.	Clear the alarm after reconnecting the teach pendant and the controller.
0x00BC	Read-write serial port not open	The read-write serial port is not opened.	Open the serial port by configuring the controller parameter or by executing the open instruction.
0x00BD	P variable cannot be changed before a program is selected or when the program is running.	P variable cannot be changed before a program is selected or when the program is running.	Select a program or make sure that the robot is stopped.
0x00BE	Client-server communication timeout, network connection lost	The client-server communication is timed out, and the network connection is lost.	Check the network connection.
0x00BF	Peer server shutdown, network connection lost	Peer server shutdown, network connection lost	Open the server.
0x00C0	J1 servo parameter error detected (Reboot required after error is fixed)	One or more servo parameters of J1 are wrong.	Reboot the robot after modifying the wrong servo parameter values.
0x00C1	J2 servo parameter error detected (Reboot required after error is fixed)	One or more servo parameters of J2 are wrong.	Reboot the robot after modifying the wrong servo parameter values.
0x00C2	J3 servo parameter error detected (Reboot required after error is fixed)	One or more servo parameters of J3 are wrong.	Reboot the robot after modifying the wrong servo parameter values.
0x00C3	J4 servo parameter error detected (Reboot required after error is fixed)	One or more servo parameters of J4 are wrong.	Reboot the robot after modifying the wrong servo parameter values.

Error code	Description	Cause	Solution
0x00C4	J5 servo parameter error detected (Reboot required after error is fixed)	One or more servo parameters of J5 are wrong.	Reboot the robot after modifying the wrong servo parameter values.
0x00C5	J6 servo parameter error detected (Reboot required after error is fixed)	One or more servo parameters of J6 are wrong.	Reboot the robot after modifying the wrong servo parameter values.
0x00CB	Robot software version and servo version do not match.	Robot software version and servo version do not match.	Upgrade the servo version to match the robot software.
0x00CC	The MC failed to read the M register.	1. MC not connected. 2. The format of the feedback message from the peer device is incorrect.	Check the protocol type, IP address, and port number set by the peer PLC, and confirm the address and port number set by the robot end.
0x00CD	The MC failed to write the M register.	1. MC not connected. 2. The format of the feedback message from the peer device is incorrect.	Check the protocol type, IP address, and port number set by the peer PLC, and confirm the address and port number set by the robot end.
0x00CE	The MC failed to read the D register.	1. MC not connected. 2. The format of the feedback message from the peer device is incorrect.	Check the protocol type, IP address, and port number set by the peer PLC, and confirm the address and port number set by the robot end.
0x00CF	The MC failed to write the D register.	1. MC not connected. 2. The format of the feedback message from the peer device is incorrect.	Check the protocol type, IP address, and port number set by the peer PLC, and confirm the address and port number set by the robot end.
0x00D0	Failed to set servo acceleration parameter	Failed to set servo acceleration parameter	Check the servo module status
0x00D1	Failed to recover system configuration profile	Failed to recover system configuration profile	Re-save the configuration file or restart the robot.

Error code	Description	Cause	Solution
0x00D2	P variables are not allowed to be modified in play mode or during program execution.	P-variable was modified in play mode or during program execution under the remote Modbus control.	Stop the program and switch to teach mode before modifying the P variable.
0x00D3	Failed to modify P variable	Failed to modify P variable in memory or failed to modify P variable in file.	1. Try to modify again. 2. Check that the GP variable to be modified exists in the file. 3. Update the software version. 4. Contact the manufacturer.
0x00D4	Configured I/O program does not exist	The configured I/O program does not exist.	Check that the configured I/O program file exists.
0x00D5	The length of data sent by the peer device exceeds the limit.	The length of the data sent by the peer device exceeds the limit during Socket or serial communication.	Reduce the length of data sent by the peer.
0x00D6	BRD variables under Modbus out of range	The BRD variables entered under Modbus control are out of range.	Check the range of BRD values set via Modbus.
0x00D7	Illegal calculated value for P variable	Internal system error	1. Try to modify the P variable again. 2. Contact the manufacturer.
0x00D8	Illegal input value for P variable	The arm parameter of the input P variable is illegal.	Check the arm parameter of the input P variable.
0x00D9	Failed to get current location via GetCurPoint instruction	Failed to get current location via GetCurPoint instruction	Reduce the frequency of GetCurPoint instruction calls as appropriate.

Error code	Description	Cause	Solution
0x00DA	Start position deviates too much from the motion interrupt position.	Under the remote control, the motion was interrupted in auto mode. When the motion was restarted, it was detected that the difference between the current position and the interrupt position exceeds the following thresholds: SCARA robot: 5° 5° 500° 3° (any joint angle); 6-axis robot: 5° 5° 5° 5° 5° 5°.	1. Manually move the robot to the interrupt position. 2. Return to the start line of the program and run the program again.
0x00DB	Failed to clear trajectory data	Trajectory data clearing failed.	Contact the manufacturer.
0x00DC	Failed to write servo parameters	Failed to write the servo parameters into the controller	Optimize the connection between the servo and the controller.
0x00DD	Direct motion point not reachable	1. The target point of direct motion exceeds the limit. 2. The target point of direct motion is a singular point.	Check the range of the target point and whether the target point is a singular point.
0x00DE	Encoder multi-turn value has been cleared, please restart.	The encoder multi-turn value is too large and is cleared when the encoder is zeroed. A reboot is required.	Restart the controller.
0x00DF	Periodic synchronization error between FPGA and servo	The servo version is incorrect, please update the servo firmware.	Update the servo firmware.
0x00E0	SN mismatch between drive and encoder (motor)	1. The encoder voltage is too low. 2. The battery is not connected during power-off.	1. Replace the encoder cable; replace a new battery with matching voltage. 2. Set the parameter "Absolute encoder reset enable" to 1 to clear the fault.

Error code	Description	Cause	Solution
0x00E1	Bus undervoltage	1. The main circuit power supply voltage is low. 2. The power supply voltage drops during operation. 3. The power supply cable does not meet the specifications. 4. The main circuit power supply dropout. 5. The controller is faulty.	1. Check the voltage and stability of the power supply, replace or regulate the power supply. 2. If the voltage is normal, check the voltage at the air-break switch. If it is low, check if the power supply cable meets the specifications and replace it if necessary. 3. Reduce the operating speed.
0x00E2	Bus overvoltage	1. The main circuit input voltage is too high. 2. The power supply is unstable or affected by lightning strike. 3. The motor is in abrupt acceleration or deceleration status and the maximum braking energy exceeds the energy absorption value. 4. The bus voltage sampling value deviates greatly from the measured value. 5. The controller is faulty.	1. Replace or adjust the power supply according to the following specifications: 220V-240V±10% (198V to 264V). 2. Connect a surge protection device and then switch on the main circuit and control circuit power supplies again. If the fault persists, replace the servo drive. 3. After confirming the input voltage of the main circuit is within the specified range, increase the acceleration/deceleration time if the operating conditions allow. 4. Contact the technical support. 5. Replace the controller.
0x00E3	Main circuit open	Mains power supply is unstable with voltage fluctuations.	Check if the power supply system is stable and if there are fluctuations in the voltage range.
0x00E4	Controller and drive power undervoltage	The controller is faulty.	Replace the controller.
0x00E5	Controller and drive power overvoltage	The controller is faulty.	Replace the controller.
0x00E6	Abnormal brake and I/O power supply	The controller is faulty.	Replace the controller.

Error code	Description	Cause	Solution
0x00E7	System power detection error	A power loss was detected by the servo module, but not by the controller module, and the power detection module is faulty.	Ensure the stability of the power supply and restart the system.
0x00E8	Power under current operating conditions exceeds controller limit.	The controller power is insufficient.	Reduce the operating speed.
0x00E9	Dedicated action execution failed	1. The current dedicated action is not activated. 2. No dedicated actions are configured. 3. The executed action is not within the safety zone. 4. The dedicated I/O setup failed.	1. Check whether the dedicated action function is activated. 2. Check whether the dedicated actions are used incorrectly or whether related dedicated actions have been configured. 3. Check whether the executed action is within the safety zone. 4. Check whether the I/O related to the action is normal.
0x00EA	Power supply phase loss alarm	See servo user guide.	See servo user guide.
0x00F0	J1 axis encoder battery alarm	1. The encoder voltage is too low. 2. The battery is not connected during power-off.	1. Replace the encoder cable; replace a new battery with matching voltage. 2. Set the parameter "Absolute encoder reset enable" to 1 to clear the fault.
0x00F1	J2 axis encoder battery alarm	1. The encoder voltage is too low. 2. The battery is not connected during power-off.	1. Replace the encoder cable; replace a new battery with matching voltage. 2. Set the parameter "Absolute encoder reset enable" to 1 to clear the fault.
0x00F2	J3 axis encoder battery alarm	1. The encoder voltage is too low. 2. The battery is not connected during power-off.	1. Replace the encoder cable; replace a new battery with matching voltage. 2. Set the parameter "Absolute encoder reset enable" to 1 to clear the fault.

Error code	Description	Cause	Solution
0x00F3	J4 axis encoder battery alarm	1. The encoder voltage is too low. 2. The battery is not connected during power-off.	1. Replace the encoder cable; replace a new battery with matching voltage. 2. Set the parameter "Absolute encoder reset enable" to 1 to clear the fault.
0x00F4	J5 axis encoder battery alarm	1. The encoder voltage is too low. 2. The battery is not connected during power-off.	1. Replace the encoder cable; replace a new battery with matching voltage. 2. Set the parameter "Absolute encoder reset enable" to 1 to clear the fault.
0x00F5	J6 axis encoder battery alarm	1. The encoder voltage is too low. 2. The battery is not connected during power-off.	1. Replace the encoder cable; replace a new battery with matching voltage. 2. Set the parameter "Absolute encoder reset enable" to 1 to clear the fault.
0x00F6	J1 axis encoder overtemperature	The encoder temperature is too high.	Cooling
0x00F7	J2 axis encoder overtemperature	The encoder temperature is too high.	Cooling
0x00F8	J3 axis encoder overtemperature	The encoder temperature is too high.	Cooling
0x00F9	J4 axis encoder overtemperature	The encoder temperature is too high.	Cooling
0x00FA	J5 axis encoder overtemperature	The encoder temperature is too high.	Cooling
0x00FB	J6 axis encoder overtemperature	The encoder temperature is too high.	Cooling
0x00FC	Failed to set servo acceleration parameter	1. The servo SDO channel is not smooth. 2. The servo has a hardware or software fault.	1. Try to save the acceleration parameters again. 2. Contact the manufacturer to update the servo software.
0x00FD	Failed to recover system configuration profile	The file system is corrupted, or the internal channel of the system is damaged.	Try to save the acceleration parameters again or restart the robot.

Error code	Description	Cause	Solution
0x00FF	Failed to return to the starting line	Failed to return to the starting line	Ensure that no project compiling errors are returned and all files are present, then sync and retry.
0x01F0	E1 external axis encoder battery low alarm	The battery level of the external axis encoder is too low or the external axis is not connected to the battery.	Connect the battery to the external axis and restart the controller and external axis.
0x01F1	E2 external axis encoder battery low alarm	The battery level of the external axis encoder is too low or the external axis is not connected to the battery.	Connect the battery to the external axis and restart the controller and external axis.
0x01F2	E3 external axis encoder battery low alarm	The battery level of the external axis encoder is too low or the external axis is not connected to the battery.	Connect the battery to the external axis and restart the controller and external axis.
0x01F3	E4 external axis encoder battery low alarm	The battery level of the external axis encoder is too low or the external axis is not connected to the battery.	Connect the battery to the external axis and restart the controller and external axis.
0x01F4	E5 external axis encoder battery low alarm	The battery level of the external axis encoder is too low or the external axis is not connected to the battery.	Connect the battery to the external axis and restart the controller and external axis.
0x01F5	E6 external axis encoder battery low alarm	The battery level of the external axis encoder is too low or the external axis is not connected to the battery.	Connect the battery to the external axis and restart the controller and external axis.
0x01F6	E1 external axis encoder overtemperature alarm	The temperature of the external axis encoder is too high.	Cut off the power supply to the external axis and check whether the encoder temperature is abnormal.
0x01F7	E2 external axis encoder overtemperature alarm	The temperature of the external axis encoder is too high.	Cut off the power supply to the external axis and check whether the encoder temperature is abnormal.
0x01F8	E3 external axis encoder overtemperature alarm	The temperature of the external axis encoder is too high.	Cut off the power supply to the external axis and check whether the encoder temperature is abnormal.

Error code	Description	Cause	Solution
0x01F9	E4 external axis encoder overtemperature alarm	The temperature of the external axis encoder is too high.	Cut off the power supply to the external axis and check whether the encoder temperature is abnormal.
0x01FA	E5 external axis encoder overtemperature alarm	The temperature of the external axis encoder is too high.	Cut off the power supply to the external axis and check whether the encoder temperature is abnormal.
0x01FB	E6 external axis encoder overtemperature alarm	The temperature of the external axis encoder is too high.	Cut off the power supply to the external axis and check whether the encoder temperature is abnormal.
0x01FC	Zero point calibration sensor connection interrupted	The communication is interrupted or unstable.	Connect the sensor correctly and check the connection for any abnormalities.
0X01FD	Failed to reset zero point calibration sensor	The communication is interrupted or unstable.	Connect the sensor correctly and check the connection for any abnormalities.
0x01FE	Failed to open zero point calibration serial port	The system failed to open the file.	Re-calibrate the axis.
0x01FF	Failed to set zero point calibration serial port parameter	The system failed to set terminal attributes.	Re-calibrate the axis.
0x0200	Failed to close zero point calibration serial port	The system failed to close the file.	Re-calibrate the axis.
0x0201	Failed to send motion segment number for zero point calibration	The system failed to send information to DSP.	Re-calibrate the axis.
0x0202	Workobject and tool both in gripping state or both in non-gripping state during zero point calibration	The workobject and tool are both in clamping state or both in non-clamping state.	Avoid setting both the workobject and the tool to clamping state or non-clamping state in the settings.
0x0203	Zero point calibration function conflicted with bus function	Zero point calibration is performed when bus function is enabled.	Disable bus function before performing zero point calibration.

Error code	Description	Cause	Solution
0x0204	Zero point calibration serial port conflicted with program serial port	Serial port in program has been opened through OpenCom instruction.	Use the CloseCom instruction to disable the opened serial port.
0x0205	Program serial port conflicted with zero calibration serial port	Serial port has been opened for zero point calibration function.	Exit the zero point calibration function.
0x0206	Failed to set the soft limit during zero point calibration	The robot is in motion state.	Re-calibrate or manually set correct system joint soft limits.
0x0207	Zero point calibration serial port not opened	The host controller software misjudged that the sensor is connected.	Restart the host controller software and then calibrate.
0x1001	Duplicate directory	The directory to be created already exists.	Replace the directory name to be created.
0x1002	Memory operation error, parent directory does not exist.	The current created directory has no parent directory.	Replace a path and recreate a directory.
0x1003	Renamed directory does not exist.	The directory to be renamed does not exist.	Refresh the directory and check whether the directory exists.
0x1004	Deleted directory does not exist.	The directory to be deleted does not exist.	Refresh the directory and check whether the directory exists.
0x1005	Error sending directory (not a directory or the directory does not exist)	The directory to be sent to the host controller is illegal.	Refresh the directory and check whether the directory exists.
0x1006	Memory error and path error in creating file	Error creating path	Refresh the file and check the creation path.
0x1007	Renamed file does not exist.	The file to be renamed does not exist.	Refresh the file and check whether the file exists.
0x1008	Deleted file does not exist or the path does not exist.	The directory to be deleted does not exist.	Refresh the file and check whether the file exists.
0x1009	The given path to the file does not exist.	The given created file path is illegal.	Check the creation path for legality.

Error code	Description	Cause	Solution
0x100A	Non-file sent	What is to be sent is not a file.	Check on the handheld teach pendant whether the object to send is a file.
0x100B	Non-directory sent	What is to be sent is not a directory.	Check on the handheld teach pendant whether the object to send is a directory.
0x100C	Frame sequence error	Frame sequence error in the process of sending a large file.	Resave or open the file.
0x100D	Device malfunction, network actively disconnected	1. The Hand-held box is closed improperly without following normal operations. 2. The network is disconnected abnormally.	1. For any error due to abnormal operations, actively disconnect the network cable. 2. For abnormality, check the error causes in conjunction with the existing error codes.
0x100E	Time format setting error	Time format setting error	Set the time according to correct format.
0x100F	System time correction error	System time calculation circuit error	Check the current network connection.
0x1010	RTC time correction error	RTC external batteries do not exist or are low.	Replace batteries or check the current hardware.
0x1011	Copy error	Misoperation during file copying	Perform copy operations again by referring to the user manual.
0x1012	Cut error	Misoperation during file cutting	Perform cut operations again by referring to the user manual.
0x1014	File encryption failed	Error encrypting file	Try to encrypt the file again.
0x1015	Unknown communication code	Version mismatch	Ensure the teach pendant and the controller match in version.
0x1016	Eth1 physical link down	Eth1 line is not working.	1. Check whether the network cable is properly plugged in or has poor contact. 2. Hardware failure, contact the manufacturer.
0x1017	Eth2 physical link down	Eth2 line is not working.	1. Check whether the network cable is properly plugged in or has poor contact. 2. Hardware failure, contact the manufacturer.

Error code	Description	Cause	Solution
0x1018	Controller fan 1 error	FAN1 blocked	1. Check if there are foreign objects blocking FAN1 in the front chamber of the cabinet, which prevents the fan from running. 2. The fan itself is faulty, replace it with a new fan.
0x1019	Controller fan 2 error	FAN2 blocked	1. Check if there are foreign objects blocking FAN1 in the front chamber of the cabinet, which prevents the fan from running. 2. The fan itself is faulty, replace it with a new fan.
0x101A	The temperature on the top of the controller is too high.	ST1 ambient temperature exceeds 60℃.	1. Check whether FAN1 in the front chamber operates normally and whether the dust-proof cotton of the fan cover is seriously blocked. 2. Check whether there are any components with abnormal temperature in the front chamber. The temperature control switch automatically resets when the ambient temperature drops to 45±5℃. 3. The temperature control switch is faulty. Replace it with a new one.

Error code	Description	Cause	Solution
0x101B	Transformer overtemperature	The transformer ST2 temperature exceeds 140℃.	1. Check whether FAN2 in the rear chamber operates normally and whether the dust-proof cotton of the fan cover is seriously blocked. 2. Check whether the QF2 circuit breaker is faulty, and whether there is an abnormally large load on the secondary side of the transformer. The temperature control switch automatically resets when the B-phase coil of the transformer cools down to 105±15℃. 3. The transformer is damaged. Replace it with a new one.
0x101C	Output I/O overcurrent	The output I/O is overcurrent.	Check the hardware line.
0x101D	Fan not installed	The fan is not installed or has poor contact.	Check the fan installation.
0x101E	Controller overtemperature alarm	The temperature inside the controller is too high.	Cooling and restart.
0x101F	Controller overtemperature warning	The temperature inside the controller is too high.	Cooling and restart.
0x1020	Oscilloscope function thread startup failed	System fault	Re-power
0x1021	Mains power fluctuations detected	Mains power supply is unstable with voltage fluctuations.	Check if the power supply system is stable and if there are fluctuations in the voltage range.
0x1022	J1 brake offline	1. The power line is not connected. 2. Parameter H02-16 is incorrectly configured. 3. The brake is disconnected. 4. The power supply of the brake is abnormal.	1. Turn off the power supply, connect the power line and power up again. 2. For motors without brake function, set H02-16 to 0. 3. Replace the cable. 4. Replace the controller.

Error code	Description	Cause	Solution
0x1023	J2 brake offline	1. The power line is not connected. 2. Parameter H02-16 is incorrectly configured. 3. The brake is disconnected. 4. The power supply of the brake is abnormal.	1. Turn off the power supply, connect the power line and power up again. 2. For motors without brake function, set H02-16 to 0. 3. Replace the cable. 4. Replace the controller.
0x1024	J3 brake offline	1. The power line is not connected. 2. Parameter H02-16 is incorrectly configured. 3. The brake is disconnected. 4. The power supply of the brake is abnormal.	1. Turn off the power supply, connect the power line and power up again. 2. For motors without brake function, set H02-16 to 0. 3. Replace the cable. 4. Replace the controller.
0x1025	J4 brake offline	1. The power line is not connected. 2. Parameter H02-16 is incorrectly configured. 3. The brake is disconnected. 4. The power supply of the brake is abnormal.	1. Turn off the power supply, connect the power line and power up again. 2. For motors without brake function, set H02-16 to 0. 3. Replace the cable. 4. Replace the controller.
0x1026	J5 brake offline	1. The power line is not connected. 2. Parameter H02-16 is incorrectly configured. 3. The brake is disconnected. 4. The power supply of the brake is abnormal.	1. Turn off the power supply, connect the power line and power up again. 2. For motors without brake function, set H02-16 to 0. 3. Replace the cable. 4. Replace the controller.
0x1027	J6 brake offline	1. The power line is not connected. 2. Parameter H02-16 is incorrectly configured. 3. The brake is disconnected. 4. The power supply of the brake is abnormal.	1. Turn off the power supply, connect the power line and power up again. 2. For motors without brake function, set H02-16 to 0. 3. Replace the cable. 4. Replace the controller.
0x1028	Safety door alarm	The safety circuit is open and the safety door is open.	Close the safety circuit or the safety door.

Error code	Description	Cause	Solution
0x1029	System is not fully powered off and may cause system problems, power the system on again.	The servo's 24V drive power is cut off, causing abnormal power supply to the inverter module, resulting in an N-phase overcurrent alarm. The fault cannot be recovered and the system must be powered on again.	Re-power the system.
0x102A	Program directory file path does not exist.	The program directory is missing.	Contact the manufacturer.
0x102B	Program file path does not exist.	The specified program file is missing.	1. Check that the program exists. 2. Contact the manufacturer.
0x102C	Invalid variable setting	Incorrect variable name or value.	1. Check if the variable name or label is P, B, R, D, Str, and whether the subscript is reasonable. 2. Check that the type of the value matches and is within the range.
0x1030	Self-learning vibration suppression data file missing	The self-learning vibration suppression data file is deleted after the robot is turned on.	1. Restart the robot to automatically restore the self-learning vibration suppression data file. 2. Import the self-learning vibration suppression data file from outside.
0x1031	The print message (from the "Print" instruction) may be lost, because of too many items.	The print instruction is used to print information too frequently, resulting in too many printed information, and the buffer is full.	Reduce the frequency of message print.
0x1032	The device is busy.	The device is in use.	Close any programs or windows that may use the device, and then try again.
0x1033	Controller fan rotor locked	The fan is blocked by foreign matter.	Turn off the power and remove the foreign matter, or contact the manufacturer.
0x1034	Number of axes or slaves mismatch	The number of axes or slaves does not match.	Check the number of configured axes (slaves).

Error code	Description	Cause	Solution
0x1035	The soft float function has been enabled. The self-learning vibration suppression function cannot be enabled.	The soft float function and self-learning vibration suppression function are mutually exclusive.	Disable the soft float function and enable the self-learning vibration suppression function.
0x1038	EtherCAT slave mode status error	The EtherCAT slave module is offline or has internal module exception.	Check whether the EtherCAT communication cable is normal and try to clear the alarm.
0x1039	The number of active EtherCAT bus slaves does not match the scanned number.	The number of active EtherCAT bus slaves does not match the scanned number.	Check the number of EtherCAT slaves.
0x103A	EtherCAT bus slave alias abnormal	The EtherCAT bus slave alias is abnormal.	Check if the slave alias is correct and try re-enabling the slave management function to resolve the issue.
0x1040	Self-learning vibration suppression profile not exist	The self-learning vibration suppression profile is deleted by mistake after the robot is turned on.	1. Restart the robot to automatically restore the self-learning vibration suppression data file. 2. Import the self-learning vibration suppression data file from outside.
0x1041	Failed to clear all self-learning vibration suppression data	1. The robot system is busy. 2. The file system is corrupted.	1. Try clearing again. 2. Contact the manufacturer.
0x1042	Failed to clear self-learning vibration suppression data file for current position	1. The robot system is busy. 2. The file system is corrupted.	1. Try clearing again. 2. Contact the manufacturer.
0x1043	Self-learning vibration suppression data file corrupted	Possible cause: Mismatch in robot name, file version, file type, and verification code.	Restore default parameters on the host controller and restart the robot.
0x1044	Failed to create self-learning vibration suppression data file	File system exception or other system exception	Contact the manufacturer.

Error code	Description	Cause	Solution
0x1045	Failed to import self-learning vibration suppression data file	1. The contents of the imported file are illegal. 2. Channels are busy, causing failure in setting DSP parameters.	1. Re-import a legal file. 2. Try to restart the robot.
0x1046	Self-learning vibration suppression data file version mismatch	The elf-learning vibration suppression data file is not the latest version.	Update the data file on PC and reload the file.
0x1050	Encoder battery voltage too low	Encoder battery is about to deplete.	Replace the encoder battery and perform corresponding operations in the robot maintenance interface.
0x1051	Insufficient grease on the lead screw	Insufficient grease on the lead screw	Refill the grease and perform corresponding operations in the robot maintenance interface.
0x1052	The controller system time is wrong. To enable the maintenance reminder function, you need to set the controller time correctly and restart the controller.	The controller system time is wrong. To enable the maintenance reminder function, you need to set the controller time correctly and restart the controller.	The controller system time is wrong. To enable the maintenance reminder function, you need to set the controller time correctly and restart the controller.
0x1053	Ball screw wear ratio approaching threshold limit	The wear ratio of the ball screw is about to reach the upper threshold limit.	Replace the ball screw.
0x1054	Ball screw grease degradation threshold imminent	The degradation threshold of ball screw grease is imminent.	Refill the ball screw with lubrication grease.
0x1055	Synchronous belt remaining life approaching zero	The synchronous belt remaining life is approaching zero.	Replace the synchronous belt.
0x1056	Reducer grease degradation threshold imminent	The degradation threshold of reducer grease is imminent.	Refill the reducer with lubrication grease.
0x1057	Reducer wear ratio approaching upper limit	The reducer wear ratio is approaching the upper limit.	Replace the reducer.

Error code	Description	Cause	Solution
0x1058	Cross-model maintenance configuration file load prohibited; reset data before model change.	A maintenance configuration file for a different robot model has been loaded.	Clear maintenance data.
0x1059	Encoder battery remaining life approaching zero	The encoder battery remaining life is approaching zero.	Replace the encoder battery.
0x1060	Failed to configure mechanical unit	Mechanical unit configuration error	Reconfigure the mechanical unit based on diagnostic prompts.
0x1061	Mechanical unit (external axis) servo type mismatch or servo verification failed	The mechanical unit servo type setting does not match the configuration type, external axis EtherCAT is not connected, or external axis electronic gear ratio and other parameters are invalid.	1. Ensure that the servo type is configured correctly. 2. Ensure that the EtherCAT cable is connected properly. 3. Ensure that the electronic gear ratio of the external axis is set correctly.
0x1062	The selected mechanical unit does not exist.	The selected mechanical unit does not exist.	Check that the input mechanical unit information is correct.
0x1063	Incorrect parameter passed to the mechanical unit	Incorrect parameter passed to the mechanical unit	Contact the manufacturer.
0x1064	Incorrect serial number of the movable guide rail passed to the mechanical unit	Incorrect serial number of the movable guide rail passed to the mechanical unit	Ensure that the robot is placed on the correct guide rail or contact the manufacturer.
0x1065	Incorrect axis logic passed to the mechanical unit	Axis logic error in the mechanical unit	Enter the correct mechanical unit axis logic.
0x1066	Mismatch in type and transmission mode passed to the mechanical unit	Mismatch in type and transmission mode passed to the mechanical unit	Check whether the type of mechanical unit matches the transmission mode.
0x1067	The axis is in motion when the mechanical unit is fed with parameters.	When saving the external axis parameters, it is detected that the additional axis is in motion.	Make sure that the additional axis is at rest before saving the mechanical unit.

Error code	Description	Cause	Solution
0x1068	Mechanical unit axis attribute error	Mechanical unit axis attribute parameter error	Contact the manufacturer.
0x1070	The active tool and workobject parameters do not match	The active tool and the workobject have the same RobHold parameter.	1. Switch the active tool and workobject matching the RobHold parameter. 2. Modify the RobHold parameter of the active tool and workobject to different values.
0x1071	The wobj parameters do not match.	RobHold is True, while UFFix is False.	Modify the configuration.
0x1072	Tool number switch failed	When the system switches to a tool, the mass of the tool that is not tool0 is 0.	Set the mass of the tool to be switched before switching to that tool.
0x1073	Load number switch failed	When the system switches to a grip load, the mass of the grip load that is not load0 is 0.	Set the mass of the load to be switched before switching to that load.
0x1074	Robot installation configuration has been switched. Reboot to apply changes.	Robot installation configuration has been switched. Reboot to apply changes.	Restart the controller.
0x1075	FPGA version is too low, please upgrade.	FPGA version is too low, please upgrade.	Upgrade to the correct FPGA version.
0x1076	Soft float function disable exception	Soft float function disable exception	Disable the soft float function again.
0x1080	External axis E1 electronic gear ratio verification failed	External axis E1 electronic gear ratio verification failed	Correctly set the servo parameters of the mechanical unit and check whether the electronic gear ratio and other parameters are legal.
0x1081	External axis E2 electronic gear ratio verification failed	External axis E2 electronic gear ratio verification failed	Correctly set the servo parameters of the mechanical unit and check whether the electronic gear ratio and other parameters are legal.
0x1082	External axis E3 electronic gear ratio verification failed	External axis E3 electronic gear ratio verification failed	Correctly set the servo parameters of the mechanical unit and check whether the electronic gear ratio and other parameters are legal.

Error code	Description	Cause	Solution
0x1083	The machine unit associated with the workobject does not exist.	The machine unit associated with the workobject does not exist.	Cancel existing association or re-associate configured mechanical unit.
0x1084	Failed to set homing method of servo	Servo feedback error when setting the servo homing method.	Check that the servo exists and is in normal condition, or contact the manufacturer.
0x1085	Axis teaching is not allowed because of the presence of independent axes in the system.	Axis teaching is not allowed because of the presence of independent axes in the system.	If teaching is required, reset all axes to normal axes before operation.
0x1086	Set all axes to normal axes before using self-learning vibration suppression.	The independent axis is not supported to use self-learning vibration suppression.	Set all axes to normal axes before using self-learning vibration suppression.
0x1087	When the robot axis is moving, the start is invalid.	When the robot axis is moving, the start is invalid.	When the robot axis is moving, the start is invalid, please restart.
0x1088	Mechanical unit needs to be restarted to take effect after configuration.	Mechanical unit needs to be restarted to take effect after configuration.	Mechanical unit needs to be restarted to take effect after configuration.
0x1089	Failure to read mechanical absolute position of servo	Failure to read mechanical absolute position of servo	Optimize the connections between the servo and the controller.
0x1090	Bus I/O mapping profile verification failed	Bus I/O mapping profile content verification failed	Reconfigure the I/O mapping.
0x1091	Bus I/O mapping profile version mismatch	Bus I/O mapping profile version mismatch	Reconfigure the I/O mapping.
0x1092	Machine unit configuration file verification failed	Machine unit configuration file content verification failed	Reconfigure the mechanical unit.
0x1093	Machine unit configuration file version mismatch	Machine unit configuration file version mismatch	Reconfigure the mechanical unit.

Error code	Description	Cause	Solution
0x1094	Motor with 20-bit encoder not supported for external axis	The external axis function does not support using 20-bit encoder motors.	The external axis function does not support using 20-bit encoder motors.
0x1096	Linear motor pole pitch acquisition failure	Linear motor pole pitch acquisition failure	Check whether the slave is correctly configured or wired.
0x1097	Zero point calibration not supported for independent axis	The current calibration axis is an independent axis.	Set the current calibration axis to a non-independent axis before calibration.
0x1101	Invalid length of data exchanged internally	When requested to provide data, the DSP does not return valid data as required.	Check the DSP software version or replace the hardware.
0x1102	Verification error for data exchanged internally	When requested to provide data, the DSP does not return valid data as required.	Check the DSP software version or replace the hardware.
0x1103	Error writing block data	The GPMC channel is abnormal.	Check the DSP software version or replace the hardware.
0x1104	Error reading block data	The GPMC channel is abnormal.	Check the DSP software version or replace the hardware.
0x1105	Block data buffer full	The FPGA buffer is full and cannot accept new data.	Retry to write data after delay for a period of time.
0x1106	Channel opening error: The channel is abnormal or has been occupied.	The GPMC channel is abnormal or has been occupied.	Restart the controller.
0x1107	Channel opening error: The channel is abnormal or has been closed.	The GPMC channel is abnormal or has been closed.	Restart the controller.
0x1108	System busy	The DSP does not respond to robot instructions.	1. Retry after a while. 2. Restart the robot. 3. Check the DSP status for whether to be paused or terminated. 4. Divide long trajectory motion into multiple segments.

Error code	Description	Cause	Solution
0x1109	Error getting channel resources	The CPMC channel is frequently occupied so that currently the system cannot apply for use of CPMC resources.	Retry to use them after delay for a period of time.
0x110A	Timeout waiting for motion module response	The response time to robot instructions from the DSP exceeds the set maximum waiting time.	1. Retry after a while. 2. Restart the robot. 3. Check the DSP status for whether to be paused or terminated.
0x110B	Motion module failed to execute the command sent by ARM	The DSP failed to execute the robot instructions.	Check if the current operation is legal or check the DSP software version.
0x110C	Illegal parameters set for motion module.	Parameters set by the robot for the DSP are incorrect, e.g. out of the parameter range.	Check that function call interface parameters are correct.
0x110D	Illegal parameters set for motion module	Commands requested by the robot to the DSP are invalid.	Check the command word and ensure that there is processing on this command in the DSP.
0x110E	The number of axes configured for the system is inconsistent with the number of online scanned axes.	The model is not matched or servo is disconnected.	Check the model and servo connection to ensure that the current robot is consistent with the configured robot.
0x110F	The configured axis data is inconsistent with the axis data retrieved again from the motion module.	Data check error	Check the software version or contact the manufacturer.
0x1110	The configured I/O data is inconsistent with the data retrieved again from the motion module.	Data check error	Check the software version or contact the manufacturer.
0x1111	Error in servo entering homing mode via EtherCAT instructions	The servo cannot enter the homing mode.	Check the software version or contact the manufacturer.
0x1112	Error in servo exiting homing mode via EtherCAT instructions	Servo cannot exit the homing mode.	Check the software version or contact the manufacturer.

Error code	Description	Cause	Solution
0x1113	Error in setting homing parameters for the servo via EtherCAT instructions	The set homing parameters are not accepted by servo.	Check the software version or contact the manufacturer.
0x1114	System parameter check error	Data check error	Check the software version or contact the manufacturer.
0x1115	Error opening data channel	The GPMC channel number is incorrect.	The default channel is 0. Ensure that the channel No. is correct.
0x1116	Data channel mapping error	The GPMC data channel cannot map data to the memory.	Check the software version or contact the manufacturer.
0x1117	Data channel mapping error	The GPMC channel is abnormal.	Check the software version or contact the manufacturer.
0x1118	Data channel mapping error	The GPMC channel is abnormal.	Check the software version or contact the manufacturer.
0x1119	Device error	The GPMC channel is abnormal.	Check the software version or contact the manufacturer.
0x111A	Error opening data channel	The GPMC channel is abnormal.	Check the software version or contact the manufacturer.
0x111B	Data communication discrepancy	The GPMC channel is abnormal.	Check the software version or contact the manufacturer.
0x111C	Data communication discrepancy	The GPMC channel is abnormal.	Check the software version or contact the manufacturer.
0x111D	Error system requesting memory	The system cannot assign a required memory.	Check whether the system memory is close to the limit or whether the requested memory is too large.
0x111E	Error configuring I/O data bias information on IR-Link bus	Data check error	Check the software version or contact the manufacturer.
0x111F	Error configuring AD data bias information on IR-Link bus	Data check error	Check the software version or contact the manufacturer.
0x1120	Error configuring DA data bias information on IR-Link bus	Data check error	Check the software version or contact the manufacturer.

Error code	Description	Cause	Solution
0x1121	Error configuring encoder data bias information on IR-Link bus	Data check error	Check the software version or contact the manufacturer.
0x1122	Error configuring AD parameter (range) on IR-Link bus	Data check error	Check the software version or contact the manufacturer.
0x1123	Error configuring DA parameter (range) on IR-Link bus	Data check error	Check the software version or contact the manufacturer.
0x1124	Error configuring module number on IR-Link bus	Data check error	Check the software version or contact the manufacturer.
0x1125	Error synchronizing controller planning position with encoder feedback position	The current axis does not exist or the GPMC channel or DSP firmware is abnormal.	Check the software version or contact the manufacturer.
0x1130	Motion module failed to get the position	Internal error	Run again or contact the manufacturer.
0x1131	Motion module acquisition attempt limit exceeded	Failed to send motion data.	Run again or contact the manufacturer.
0x113A	Channel data address out of range	The channel data address is out of range.	Check the software version or contact the manufacturer.
0x1140	EtherCAT PDO initialization parameter error	PDO data error occurs.	Run again or contact the manufacturer.
0x1140	EtherCAT PDO initialization parameter error	PDO data error occurs.	Run again or contact the manufacturer.
0x11EE	Continuous collision detected and motor may overheat. Please confirm if a collision has occurred and remove any obstruction; this delayed alarm requires 10s to clear	Continuous collision is detected and motor may be overheating.	Alarm can be cleared after a 10s delay; please remove the obstruction.

Error code	Description	Cause	Solution
0x11FF	Instruction not supported by virtual controller	The virtual controller does not currently support this instruction.	Do not use this instruction in the virtual controller.
0x1200	Robot in emergency stop state when enabled	The robot is in an emergency stop when the enable command is issued.	Check the emergency stop button, clear the emergency stop status and re-enable the robot.
0x1201	Excessive fluctuation in robot joint position when the robot is enabled	The robot is currently in a vibrating or dragged state.	Reduce the joint vibration of the robot and enable it after the robot stabilizes.
0x1202	Servo alarm present when the robot is enabled.	The robot has a servo alarm when enabled.	Clear the servo alarm and re-enable the robot.
0x1203	Robot enabled too quickly after being disabled.	The robot is enabled too quickly after it is disabled.	Clear the alarm and re-enable the robot.
0x1204	Alarms other than joint overrun present when robot is enabled.	There are other alarms besides joint overrun.	Clear the alarm and re-enable the robot.
0x1205	Excessive fluctuation in robot joint position when getting zero point	The robot is currently in a vibrating or dragged state.	Reduce the joint vibration of the robot and get the zero point after the robot stabilizes.
0x1206	Soft float instruction timed out	Soft float instruction blocking timeout	Re-execute the instruction.
0x2001	Segment data overlap	The previously input target point is the same with the currently input target point.	Reteach the robot with different points.

Error code	Description	Cause	Solution
0x2002	Error calculating input arc parameters	Circular arc interpolation information cannot be calculated because: 1. At least two points are too close. 2. Three points are nearly collinear. 3. Pose change is too large. 4. Transition is performed near a singularity.	Re-teach the robot with other points to calculate the arc interpolation information.
0x2003	Error calculating input linear parameters	Unable to calculate linear interpolation information. 1. Pose change is too large. 2. Transition is performed near a singularity.	Re-teach the robot with other points to calculate the linear interpolation information.
0x2004	Inverse kinematics error	The robot is at a singularity or out of reach.	Disable the robot, switch to the joint mode and then move the robot out of the singularity, or change the target points to those that can be arrived.
0x2005	Singular position error	The robot moves to a singular position.	Switch to the joint mode and move the robot out of the singular position.
0x2006	Servo disabled during automatic running	1. Power failure of a drive occurs. 2. A drive is wired incorrectly. 3. A drive fails.	Check that the drive is normal.
0x2007	Reserved	Reserved	Reserved
0x2008	I/O index out of range	The physical I/O module does not exist.	Check that the corresponding physical I/O module exists.
0x2009	Jump parameter setting error	MH parameter is beyond the limit of J3 axis.	Modify the limit height or reselect a start or end point.
0x200A	Arm type parameter error	Three arm type parameters before the end point are not consistent with those of the start point in linear or circular motion.	Modify the motion to joint motion or reselect points to ensure consistent arm type.

Error code	Description	Cause	Solution
0x200B	Inappropriate motion characteristic parameters	The motion parameter input range is unreasonable.	Modify motion parameters such as speed and acceleration.
0x200C	DA operation error	The channel is configured as current output, but the voltage command is used. Or the channel is configured as voltage output, but the current command is used.	Operate the DA port using a command consistent with the configuration.
0x200D	The command to enable servo is sent, but the servo is not enabled actually.	1. The main power supply of the servo is not switched on. 2. The joint may be in deceleration process. 3. The joint is in motion state and does not arrive at the position.	1. Check whether the strong current button of the controller is pressed down. 2. The interval between servo stop and start or enable is too short. 3. Check whether the joint arrives at the position. Amplify the arrival error or adjust the servo parameters. 4. Status word feedback is too slow.
0x200E	Joint motion parameter input error	1. The points are beyond the space range for the Delta robots. 2. MoveJ and MOVL or MOVC performs transition near a singularity.	1. Check whether the points are beyond the space range for the Delta robots. 2. Check whether MoveJ and MOVL or MOVC performs transition near a singularity.
0x200F	Robot not returned to zero	The robot does not perform the homing operation when an incremental encoder is used.	For incremental encoder, perform the homing operation first.
0x2010	Robot radial direction out of bound	The X and Y combined radius at the end of the robot is greater than the set radius.	Under the rectangular frame, step so that the robot end moves in the direction of reduction of the X and Y combined radius.
0x2011	Robot Z positive direction out of bound.	The Z at the end of the robot is greater than the setpoint.	Under the rectangular frame, step so that the robot end moves in the negative direction of Z.
0x2012	Robot Z negative direction out of bound	The Z at the end of the robot is smaller than the setpoint.	Under the rectangular frame, step so that the robot end moves in the positive direction of Z.

Error code	Description	Cause	Solution
0x2013	Robot out of bound	The teaching point exceeds the boundary.	Change the teaching points so that they are within the work space of the robot.
0x2014	System thread priority exception	System thread priority exception detected during operation	Export the diagnostic information and contact the manufacturer.
0x2015	Real time task synchronization timeout	Real time task synchronization timeout	Export the diagnostic information and contact the manufacturer.
0x2016	The included angle between J2 and J3 axes of the palletizing robot is too small.	The included angle between J2 and J3 axes of the palletizing robot is too small.	In the manual mode, rotate the J3 axis positively or the J2 axis negatively.
0x2017	The included angle between J2 and J3 axes of the palletizing robot is too large.	The included angle between J2 and J3 axes of the palletizing robot is too large.	In the manual mode, rotate the J3 axis positively or the J2 axis negatively.
0x2018	Abnormal robot speed	The robot joint speed exceeds twice the allowable maximum speed.	1. Ensure that the settings of allowable maximum speed and acceleration of joints on the setting interface are reasonable. 2. Ensure that the robot is not near a singularity. 3. Reduce the speed of line motion.
0x2019	Motion parameter error	The motion planning parameters are abnormal.	Check that the motion parameters (speed, acceleration) are reasonable.
0x201A	The position speed or orientation speed exceeds the setting value.	The robot end motion exceeds the set position or orientation speed.	1. If the MoveJ instruction is used, set the MoveJ speed coefficient to a smaller value. 2. Check whether the set orientation speed is consistent with the J4 joint speed.
0x201D	Arm type change error	The arm type change command (specially used for SCARA) is not supported near a singular position.	Replace the point with a point far from the singular area and then call the arm type change command.

Error code	Description	Cause	Solution
0x201E	The robot acceleration is abnormal.	The robot joint speed exceeds 50 times the allowable maximum acceleration.	1. Ensure that the settings of allowable maximum speed and acceleration of joints on the setting page are reasonable. 2. Ensure that the robot is not near a singular position. 3. Reduce the speed and acceleration of line motion.
0x201F	Speed setting error	Insufficient internal storage in DSP	Decrease the Cartesian speed or increase the joint speed.
0x2020	Speed setting error	Insufficient internal spline storage in DSP	Reduce the position and orientation speed, or increase the joint speed.
0x2021	The lower worksapce boundary is too large or the stop distance is too small.	The lower worksapce boundary is too large or the stop distance is too small for the tracking process.	Decrease the lower worksapce boundary, and then increase the stop distance.
0x2022	Robot out of lower worksapce boundary	The robot exceeds the set workspace in the tracking process.	Adjust the lower worksapce boundary and the upper pickup boundary; increase the robot speed; reduce the conveyor speed.
0x2023	Conveyor speed too large	The conveyor speed exceeds a reasonable range.	The conveyor speed exceeds the maximum speed limit (1 m/s for the linear conveyor and 180°/s for the turntable conveyor).
0x2024	Conveyor speed fluctuation too large	The conveyor speed fluctuates excessively.	Check that the conveyor motor speed does not fluctuate excessively or that the conveyor is abnormal.

Error code	Description	Cause	Solution
0x2025	Vision data waiting timeout	No returned data is received for a long period of time after vision triggering signals are sent, and the vision processing cycle is greater than the photographing interval.	1. Check whether the vision trigger signal is delivered successfully. 2. If the vision trigger signal is delivered successfully every time, check whether the camera will take pictures normally every time. 3. The processing time of vision software is too long, or the delay time of data transmission is too long.
0x2026	Workobject parameter type error	A motion instruction with Wobj parameter is used in tracking motion，or a motion instruction with Cnv parameter is used in normal motion.	The workobject parameter of the motion instruction in the tracking motion shall be modified to Cnv, and the workobject parameter of the motion instruction in the normal motion shall be modified to Wobj.
0x2027	Dynamic point coordinate error	The given dynamic target position is incorrect, singular or out of bounds.	Check whether the coordinate given by vision is within a reasonable range.
0x2028	Syntax error in the conveyor tracking instruction	Refsys Convyor or Refsys Base is used continuously.	Check that RefConvyor and RefBase are jointly used.
0x2029	Failed to establish frame for grasping the work object	The Refsys Convyor instruction is executed before the GetCnvObject instruction is executed.	Call the GetCnvObject instruction first.
0x202A	Conveyor vision port error	Multiple conveyors adopting vision detection are used.	Check whether more than two vision inputs are used at the same time.
0x202B	Single-step teaching is not allowed.	The conveyor tracking-related instructions do not allow single-step teaching.	Single-step operation on instructions between Refsys Convyor and Refsys Base is not allowed.
0x202C	A disabled conveyor is used in the instruction.	A disabled conveyor is used in the instruction.	Check whether the conveyor used in the program is disabled.

Error code	Description	Cause	Solution
0x202D	PTP motion is not allowed in the tracking process.	Joint motions such as MovJ and JumP are used in the tracking process.	Check whether the MovJ or jump instruction is used between Refsys Convyor and Refsys Base.
0x202E	Conveyor direction error	Conveyor direction error	1. Check whether the encoder direction parameter on the conveyor belt interface is correct. 2. Check whether the conveyor belt shows the evidence of slipping.
0x202F	The start position for teaching in Cartesian system is at a singular position where inverse kinematic solution cannot be executed.	The start position for teaching in Cartesian system is at a singular position where inverse kinematic solution is impossible.	Switch to joint mode and move out of the singular position.
0x2030	Inconsistent latch counters	The robot position latching counters of the respective axes are inconsistent, resulting in incorrect latch state feedback from the servo. The latch I/O may have been called in static, dynamic, or xqt task.	Check that the latching signals of the respective axes are active. Check if the latch I/O has been called in static, dynamic, or xqt task. If it is called, change the latch I/O used in other tasks.
0x2031	Latching buffer full	When the robot position is latched, the latching buffer is full.	The latching speed is too fast, or there are too many latched positions that are not used.
0x2032	The motion range of J1 or J2 exceeds 180°.	When the inverted SCARA robot performs interpolation motion, the motion range of J1 or J2 axis exceeds 180°, making it impossible to ensure that the termination angle of interpolation motion is consistent with the predetermined angle.	Avoid the above situations.

Error code	Description	Cause	Solution
0x2034	The motion range of J2 passes through the singular point.	When the inverted SCARA robot performs interpolation motion, the J2 axis moves through the singularity, making it impossible to ensure that the termination angle of interpolation motion is consistent with the predetermined angle.	Avoid the above situations.
0x2035	EtherCAT communication feedback error	EtherCAT bus feedback data is lost.	Check that the EtherCAT network cable is properly connected and there is no external interference.
0x5100	Balance cylinder spring maintenance interval reached	The balance cylinder spring has reached the set lifespan threshold. Continued use may pose a risk of component damage.	Contact the manufacturer for maintenance promptly.
0x2036	Position latch timeout	1. The servo is not properly configured with probe function. 2. The correct I/O edge signal is not triggered. 3. There is hardware interference.	1. Check if a valid edge signal is output. 2. Check if the servo probe is configured. 3. Check for hardware interference.
0x2041	Limit-triggering point present in planned trajectory	The planned trajectory includes points that may trigger the limit.	Check the corresponding motion segment for points that may trigger limit.
0x2042	The constraint of velocity in trajectory is too small.	1. In the planning, the calculated spatial velocity constraint is too small, and the velocity parameters are set incorrectly. 2. The trajectory is close to a singularity.	1. Adjust the set speed parameters. 2. Adjust the trajectory away from the singularity.
0x2043	Path planning: inverse kinematics error	Inverse kinematic solution cannot be executed at an intermediate point of the planned trajectory.	Modify points in the erroneous segment.

Error code	Description	Cause	Solution
0x2044	The position goes beyond the lower operating boundary during the dynamic tracking preprocessing.	The target point is detected to go beyond the lower operating boundary during the dynamic tracking preprocessing.	1. Check that the received vision data point is within a reasonable range. 2. Check that the dynamic coordinates in the given motion instruction are not out of the lower operating boundary.
0x2045	Path planning: J1 out of range	A limit is triggered at J1 in the planned trajectory.	Check the corresponding motion segment for points that may trigger limit.
0x2046	Path planning: J2 out of range	A limit is triggered at J2 in the planned trajectory.	Check the corresponding motion segment for points that may trigger limit.
0x2047	Path planning: J3 out of range	A limit is triggered at J3 in the planned trajectory.	Check the corresponding motion segment for points that may trigger limit.
0x2048	Path planning: J4 out of range	A limit is triggered at J4 in the planned trajectory.	Check the corresponding motion segment for points that may trigger limit.
0x2049	Path planning: J5 out of range	A limit is triggered at J5 in the planned trajectory.	Check the corresponding motion segment for points that may trigger limit.
0x204A	Path planning: J6 out of range	A limit is triggered at J6 in the planned trajectory.	Check the corresponding motion segment for points that may trigger limit.
0x204B	Path planning: J7 out of range	A limit is triggered at J7 in the planned trajectory.	Change the motion points.
0x204C	Path planning: J8 out of range	A limit is triggered at J8 in the planned trajectory.	Change the motion points.
0x204D	Path planning: J9 out of range	A limit is triggered at J9 in the planned trajectory.	Change the motion points.
0x204E	Path planning: J10 out of range	A limit is triggered at J10 in the planned trajectory.	Change the motion points.
0x204F	Path planning: J11 out of range	A limit is triggered at J11 in the planned trajectory.	Change the motion points.
0X2050	Path planning: J12 out of range	A limit is triggered at J12 in the planned trajectory.	Change the motion points.

Error code	Description	Cause	Solution
0X2051	Path planning: J13 out of range	A limit is triggered at J13 in the planned trajectory.	Change the motion points.
0X2052	Path planning: J14 out of range	A limit is triggered at J14 in the planned trajectory.	Change the motion points.
0x2053	Too large orientation motion angle	J4: The orientation motion angle in the planned trajectory is greater than 360°. J6: The orientation motion angle in the planned trajectory is greater than 180°.	Re-select appropriate starting and target points for motion.
0x2054	J1 motion range exceeds 180° due to improper intermediate point in circular motion.	When the inverted SCARA robots performs circular motion, the difference in angle of the J1 axis between the starting point and the intermediate point plus the difference in angle of the J1 axis between the intermediate point and the end point exceeds 180°.	Re-select an appropriate intermediate point of the circular motion.
0x2055	Unable to reach the predetermined target angle	Unable to reach the predetermined target angle	Modify the arm parameters of "J4 range".
0x2056	During the transition movement, the robot's planned points cannot be solved inversely. Please reduce the transition or get the points again.	During the transition movement, the robot's planned points cannot be solved inversely. Please reduce the transition or get the points again.	During the transition movement, the robot's planned points cannot be solved inversely. Please reduce the transition or get the points again.
0x2057	Movc intermediate point pose mode: Intermediate point not in arc 1/4-3/4 range	In intermediate point pose mode, the intermediate point is not between 1/4 to 3/4 of the arc. Please reselect the intermediate point for arc motion.	Re-select an appropriate intermediate point of the motion.
0x2058	Movc intermediate point pose mode: Pose planning failed	Pose movement is too large in Movc intermediate point pose mode.	Re-select appropriate starting point, intermediate point, and target point of the motion

Error code	Description	Cause	Solution
0x2060	Insufficient instruction space	The joint speed is too small while the spatial speed is too large.	1. Increase the joint speed. 2. Reduce the spatial speed.
0x2061	Insufficient space for planned positions	The joint speed is too small while the spatial speed is too large.	1. Increase the joint speed. 2. Reduce the spatial speed.
0x2062	Interpolation data error	1. The joint speed and acceleration are not reasonably set. 2. The motion trajectory is too long.	1. Set the joint speed and acceleration appropriately. 2. Split long trajectory into multiple segments.
0x2063	Insufficient spline space	1. The spline distance is too large. 2. The speed is too low during transition.	1. Reduce the transition length. 2. Increase the set speed. 3. Save the error message and contact the manufacturer.
0x2064	Spline interpolation planning error	Spline interpolation planning error	Save the error message and contact the manufacturer.
0x2065	Speed fitting error	Speed fitting error	Save the error message and contact the manufacturer.
0x2066	Speed planning error	Speed planning error	Save the error message and contact the manufacturer.
0x2067	Joint transition planning error	The calculation result is incorrect during joint transition planning due to improper input parameters.	If possible, fine-tune the points or motion parameters.
0x2068	Speed look-ahead check not reasonable	It is checked that the look-ahead results are not reasonable.	Save the error message and contact the manufacturer.
0x2069	Adaptive planning results check not reasonable	Adaptive planning results check not reasonable	Save the error message and contact the manufacturer.
0x206A	General planning results check not reasonable	General planning results check not reasonable	Save the error message and contact the manufacturer.

Error code	Description	Cause	Solution
0x206B	Joint speed or acceleration exceeds the set value during normal transition	The set joint speed may be exceeded during a normal transition in CP mode.	1. Decrease the position speed and orientation speed or increase the joint speed appropriately. 2. Ensure that points are out of singular area. Reduce the transition level and transition length.
0x206C	The joint speed in tracking motion exceeds the set value.	The joint speed in tracking motion exceeds the set value.	Reduce the conveyor speed or the robot speed.
0x206D	Smooth stop exception	The deceleration distance is insufficient when a smooth stop is made.	Save the error message and contact the manufacturer.
0x206E	Joint's acceleration overlimit	When the transition trajectory between CP instructions approaches to the singular area, the acceleration of the joint exceeds the limit.	Adjust the instruction's trajectory or the transition parameters to avoid singularities.
0x206F	Robot approaching singularity	The joint speed and acceleration are too large due to the robot approaching the singular position.	1. Adjust the motion points to avoid singularities. 2. Set the motion speed near the singularities to a small value.
0x2070	Transition trajectory enters the singular area.	The transition trajectory between CP instructions enters the singular area.	Adjust the instruction's trajectory or the transition parameters to avoid singularities.
0x2071	Teach path planning: J1 out of range	A limit is triggered at J1 in the planned trajectory.	Check the corresponding motion segment for points that may trigger limit.
0x2072	Teach path planning: J2 out of range	A limit is triggered at J2 in the planned trajectory.	Check the corresponding motion segment for points that may trigger limit.
0x2073	Teach path planning: J3 out of range	A limit is triggered at J3 in the planned trajectory.	Check the corresponding motion segment for points that may trigger limit.
0x2074	Teach path planning: J4 out of range	A limit is triggered at J4 in the planned trajectory.	Check the corresponding motion segment for points that may trigger limit.

Error code	Description	Cause	Solution
0x2075	Teach path planning: J5 out of range	A limit is triggered at J5 in the planned trajectory.	Check the corresponding motion segment for points that may trigger limit.
0x2076	Teach path planning: J6 out of range	A limit is triggered at J6 in the planned trajectory.	Check the corresponding motion segment for points that may trigger limit.
0x2077	Teach path planning: inverse kinematics error	Inverse kinematic solution cannot be executed at an intermediate point of the planned trajectory.	Modify points in the erroneous segment.
0x2078	Error in calculating linear input parameters in teaching mode.	Unable to calculate linear interpolation information.	Re-teach the robot with other points to calculate the linear interpolation information.
0x2079	The arm type is inconsistent with the current point during preprocessing in teaching state and space interpolation cannot be performed.	The ARM_TYPE parameter is inconsistent with the current point during preprocessing and spatial interpolation cannot be performed.	Modify points in the erroneous segment.
0x207A	In teaching state, the orientation change is too large.	The orientation motion angle is found to be greater than 180 degrees in the planned trajectory.	1. Reduce the pose change range of a single motion. 2. Check the difference in ARM_TYPE of points between the alarming program line and the previous program line.
0x207B	The constraint of velocity in trajectory is very low in manual mode.	The constraint of velocity in trajectory is too small.	In teaching mode, the spatial velocity constraint calculated in the planning is too small due to incorrect speed parameter setting or singularities.
0x2081	Valid hardware signal for emergency stop	Hardware emergency stop is active.	A hardware emergency stop is active. Confirm safety and then dismiss the alarm.

Error code	Description	Cause	Solution
0x2082	Mode switch timeout	1. The configured robot acceleration is too small, causing excessive downtime. 2. The robot is moving while switching mode.	1. Check that the motion parameters are set correctly. 2. Stop the current motion and make sure the robot is stationary before switching the mode.
0x2083	Power error	1. The input power supply is abnormal. 2. The system hardware is faulty.	View the servo parameter H0B-45 and handle the problem according to the user guide.
0x2084	Fan error	1. The fan is not connected. 2. The fan cable has poor contact or broken wire.	View the servo parameter H0B-45 and handle the problem according to the user guide.
0x2085	Discharge tube shorted or regenerative resistor not connected.	1. The discharge resistor is not connected. 2. The discharge tube is shorted.	View the servo parameter H0B-45 and handle the problem according to the user guide.
0x2086	Discharge tube open circuit	The discharge tube is open circuit.	View the servo parameter H0B-45 and handle the problem according to the user guide.
0x2091	Trajectory recovery exception	1. Exception in line number processing for trajectory recovery. 2. Exception in data communication for trajectory recovery.	1. Reset the line number. 2. Save the error message and contact the manufacturer.
0x2092	Trajectory recovery target point error	1. Exception in line number processing for trajectory recovery. 2. Exception in data communication for trajectory recovery.	1. Reset the line number. 2. Save the error message and contact the manufacturer.
0x2093	Trajectory recovery motion state error	1. Exception in line number processing for trajectory recovery. 2. Exception in data communication for trajectory recovery.	1. Reset the line number. 2. Save the error message and contact the manufacturer.

Error code	Description	Cause	Solution
0x2094	Trajectory recovery initialization error	1. Exception in line number processing for trajectory recovery. 2. Exception in data communication for trajectory recovery.	1. Reset the line number. 2. Save the error message and contact the manufacturer.
0x2095	Issued trajectory recovery instruction line/segment number error	1. Exception in line number processing for trajectory recovery. 2. Exception in data communication for trajectory recovery.	1. Reset the line number. 2. Save the error message and contact the manufacturer.
0x2096	Trajectory recovery current point deviated from stop position	Trajectory recovery current point deviated from stop position	1. Reset the line number. 2. Save the error message and contact the manufacturer.
0x2097	Trajectory recovery target point changed	Trajectory recovery target point changed	1. Reset the line number. 2. Save the error message and contact the manufacturer.
0x20A1	Data acquisition board communication failure	Communication between robot controller and data acquisition board is disconnected.	Check in the following order: 1. Check that the encoder cable on the controller is securely connected. 2. Check that the cables and connectors on the data acquisition board are properly connected. 3. Check that the data acquisition board is installed. 4. Check that the data acquisition board light is working properly (green light flashing). 5. Check that the data acquisition board has firmware. If the alarm persists, replace the data acquisition board.

Error code	Description	Cause	Solution
0x20A2	Data acquisition board data error	The data collection board is damaged.	Check in the following order: 1. Check that the data acquisition board light is working properly (green light flashing). 2. Check whether the alarm can be cleared after restart. 3. Check that the controller's power-up timing is normal. If the alarm persists, replace the data acquisition board.
0x20A3	Data acquisition board communication error	The data acquisition board encounters data frame loss.	Check in the following order: 1. Check that the encoder cable on the controller is securely connected. 2. Check that the cables and connectors on the data acquisition board are properly connected. 3. Check that the robot motion does not result in poor contact with the data acquisition board connector due to force during robot motion. 4. Check that the data acquisition board hardware circuit works properly. 5. Check whether the alarm can be cleared after restart. If the alarm persists, replace the data acquisition board.
0x20A4	Data acquisition board data changed significantly	Robot collides with environment; robot moves with abnormal sound.	Check in the following order: 1. Check whether robot collides with environment. 2. Check whether robot speed and acceleration is too high. If the alarm persists, replace the data acquisition board.

Error code	Description	Cause	Solution
0x20A5	Data acquisition board overtemperature	Data acquisition board overtemperature	Check in the following order: 1. Check that the temperature of the robot's working environment meets the requirements. 2. If the alarm persists, reduce the speed and acceleration of the robot.
0x20A6	Excessive robot vibration	Excessive robot vibration	Check in the following order: 1. Check the robot for visible vibrations. 2. Check the robot joints for obvious transmission clearance. 3. If the alarm persists, reduce the speed and acceleration of the robot.
0x20A7	Robot positioning time too long	Residual robot vibration is significant, resulting in long positioning time.	Check in the following order: 1. Check the robot for visible vibrations. 2. Check the robot joints for obvious transmission clearance. If the alarm persists, reduce the speed and acceleration of the robot.
0x20A8	Collision detected	Robot collides with environment; robot moves with abnormal sound.	Check whether the robot collides with the environment; check whether there are abnormal noises during the robot motion.

Error code	Description	Cause	Solution
0x20A9	Vibration reduction failure	Vibration reduction failure	Check in the following order: 1. Check that the encoder cable on the controller is securely connected. 2. Check that the cables and connectors on the data acquisition board are properly connected. 3. Check whether the alarm can be cleared after restart. If the alarm persists, turn off the closed loop vibration suppression function.
0x20AA	Robot motion state error	1. The robot model in software is inconsistent with the actual one. 2. Mechanical causes: collision, loose or stuck transmission mechanism. 3. Home position deviation is too large. 4. The reduction ratio setting is incorrect. 5. The joint motion direction is opposite. 6. The data acquisition board installation direction is incorrect.	Check in the following order: 1. Make sure the robot model in software is the same as with the actual one. 2. Check for mechanical anomalies such as collision, loose or stuck transmission mechanism. 3. Check that the zero point of the robot is accurate. 4. Check that the reduction ratio parameter is correctly set. 5. Check that the direction of joint motion is correct. 6. Check that the data acquisition board is mounted in the correct direction. If the alarm persists, turn off data acquisition board alarm function.
0x20AB	Self-learning vibration reduction function calculation error	Self-learning vibration reduction function calculation error	Reduce the motion speed; turn off the self-learning vibration suppression function.
0x20AC	Self-learning vibration suppression learning timeout	Self-learning vibration suppression learning timeout	1. Change the self-learning flag in the motor instruction to SLOff. 2. Contact the manufacturer.

Error code	Description	Cause	Solution
0x20AD	Data acquisition board hardware failure	Data acquisition board hardware ID check failure	Check in the following order: 1. Check the data acquisition board for component damage. 2. Check that the data acquisition board hardware version matches with software version. If the alarm persists, replace the data acquisition board.
0x20AE	Data acquisition board self-test failure	Data acquisition board self-test failure	Check in the following order: 1. Check whether the encoder cable and data acquisition board cable are hot swapped. 2. Check whether data acquisition board chips are damaged. 3. Check whether the alarm can be cleared after restart. If the alarm persists, replace the data acquisition board.
0x20AF	Incorrect robot installation form setting or excessive zero point deviation. Check whether the installation form and zero point settings are correct; This alarm cannot be cleared. Please restart the controller.	The robot installation form setting is incorrect, or the zero point deviation is too large.	Set the robot installation form and the robot zero point position correctly.
0x20B1	Tool load mass exceeds limit	The mass setting of the tool load exceeds the limit.	Reduce the mass setting of the tool load to the limit range.
0x20B2	Tool load centroid position exceeds limit	The centroid position setting of the tool load exceeds the limit.	Reduce the centroid position setting of the tool load to the limit range.
0x20B3	Tool load centroid orientation exceeds limit	The centroid orientation setting of the tool load exceeds the limit.	Reduce the centroid orientation setting of the tool load to the limit range.
0x20B4	Tool load inertia exceeds limit	The inertia setting of the tool load exceeds the limit.	Reduce the inertia setting of the tool load to the limit range.

Error code	Description	Cause	Solution
0x20B5	Load mass exceeds the limit	The mass setting of the load exceeds the limit.	Reduce the mass setting of the load to the limit range.
0x20B6	Load centroid position exceeds limit	The centroid position setting of the load exceeds the limit.	Reduce the centroid position setting of the load to the limit range.
0x20B7	Load centroid orientation exceeds limit	The inertia setting of the load exceeds the limit.	Reduce the inertia setting of the load to the limit range.
0x20B8	Load centroid orientation exceeds limit	The inertia setting of the load exceeds the limit.	Reduce the inertia setting of the load to the limit range.
0x20B9	Arm load mass exceeds the limit	The mass setting of the arm load exceeds the limit.	Reduce the mass setting of the arm load to the limit range.
0x20BA	Arm load centroid position exceeds limit	The centroid position setting of the arm load exceeds the limit.	Reduce the centroid position setting of the arm load to the limit range.
0x20BB	Arm load centroid orientation exceeds limit	The centroid orientation setting of the arm load exceeds the limit.	Reduce the centroid orientation setting of the arm load to the limit range.
0x20BC	Arm load inertia exceeds limit	The inertia setting of the arm load exceeds the limit.	Reduce the inertia setting of the arm load to the limit range.

Error code	Description	Cause	Solution
0x20C1	Abnormality detected on J1 axis, please confirm if a collision has occurred • If no collision is found, check whether the motor of this axis has failed, and check for any interference or pulling of external cables. • If no abnormalities are found, check whether the load parameters are set correctly and whether the collision detection parameters have been adjusted. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	Check whether a collision has occurred. • If no collision is found, check whether the motor of this axis has failed. • If no abnormalities are found, check whether the load parameters are set correctly, whether the collision detection parameters have been adjusted, and whether it is caused by limited servo tracking capability. • If no abnormalities are found, check whether the robot experiences significant friction changes after prolonged cold starts or extended operation. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	Check whether a collision has occurred. If not, check the following: 1. Check that the load parameters are set correctly. 2. Check that the robot model in the controller matches the actual robot. 3. Check the robot for motor jam, brake not opened, etc. 4. If the robot starts suddenly after a long cold state, perform appropriate warm-up before running; if the robot's friction significantly decreases after long-term running, contact the manufacturer to tune the friction parameters. 5. If the robot operates at high speed and heavy load, it will cause current saturation phenomenon, which is prone to false alarms, so prevent the robot from operating under such conditions; You can also try to enter the medium precision mode through commands or manually turn on the speed feedforward of all axes to solve this problem to some extent. 6. If it is determined to be a false alarm, the collision detection sensitivity of the axis can be appropriately increased, and it is recommended to use the recommended collision detection sensitivity value. 7. If the alarm persists, turn off the collision detection switch for that axis.

Error code	Description	Cause	Solution
0x20C2	Abnormality detected on J2 axis, please confirm if a collision has occurred • If no collision is found, check whether the motor of this axis has failed, and check for any interference or pulling of external cables. • If no abnormalities are found, check whether the load parameters are set correctly and whether the collision detection parameters have been adjusted. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	Check whether a collision has occurred. • If no collision is found, check whether the motor of this axis has failed. • If no abnormalities are found, check whether the load parameters are set correctly, whether the collision detection parameters have been adjusted, and whether it is caused by limited servo tracking capability. • If no abnormalities are found, check whether the robot experiences significant friction changes after prolonged cold starts or extended operation. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	Check whether a collision has occurred. If not, check the following: 1. Check that the load parameters are set correctly. 2. Check that the robot model in the controller matches the actual robot. 3. Check the robot for motor jam, brake not opened, etc. 4. If the robot starts suddenly after a long cold state, perform appropriate warm-up before running; if the robot's friction significantly decreases after long-term running, contact the manufacturer to tune the friction parameters. 5. If the robot operates at high speed and heavy load, it will cause current saturation phenomenon, which is prone to false alarms, so prevent the robot from operating under such conditions; You can also try to enter the medium precision mode through commands or manually turn on the speed feedforward of all axes to solve this problem to some extent. 6. If it is determined to be a false alarm, the collision detection sensitivity of the axis can be appropriately increased, and it is recommended to use the recommended collision detection sensitivity value. 7. If the alarm persists, turn off the collision detection switch for that axis.

Error code	Description	Cause	Solution
0x20C3	Abnormality detected on J3 axis, please confirm if a collision has occurred • If no collision is found, check whether the motor of this axis has failed, and check for any interference or pulling of external cables. • If no abnormalities are found, check whether the load parameters are set correctly and whether the collision detection parameters have been adjusted. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	Check whether a collision has occurred. • If no collision is found, check whether the motor of this axis has failed. • If no abnormalities are found, check whether the load parameters are set correctly, whether the collision detection parameters have been adjusted, and whether it is caused by limited servo tracking capability. • If no abnormalities are found, check whether the robot experiences significant friction changes after prolonged cold starts or extended operation. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	Check whether a collision has occurred. If not, check the following: 1. Check that the load parameters are set correctly. 2. Check that the robot model in the controller matches the actual robot. 3. Check the robot for motor jam, brake not opened, etc. 4. If the robot starts suddenly after a long cold state, perform appropriate warm-up before running; if the robot's friction significantly decreases after long-term running, contact the manufacturer to tune the friction parameters. 5. If the robot operates at high speed and heavy load, it will cause current saturation phenomenon, which is prone to false alarms, so prevent the robot from operating under such conditions; You can also try to enter the medium precision mode through commands or manually turn on the speed feedforward of all axes to solve this problem to some extent. 6. If it is determined to be a false alarm, the collision detection sensitivity of the axis can be appropriately increased, and it is recommended to use the recommended collision detection sensitivity value. 7. If the alarm persists, turn off the collision detection switch for that axis.

Error code	Description	Cause	Solution
0x20C4	Abnormality detected on J4 axis, please confirm if a collision has occurred • If no collision is found, check whether the motor of this axis has failed, and check for any interference or pulling of external cables. • If no abnormalities are found, check whether the load parameters are set correctly and whether the collision detection parameters have been adjusted. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	Check whether a collision has occurred. • If no collision is found, check whether the motor of this axis has failed. • If no abnormalities are found, check whether the load parameters are set correctly, whether the collision detection parameters have been adjusted, and whether it is caused by limited servo tracking capability. • If no abnormalities are found, check whether the robot experiences significant friction changes after prolonged cold starts or extended operation. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	Check whether a collision has occurred. If not, check the following: 1. Check that the load parameters are set correctly. 2. Check that the robot model in the controller matches the actual robot. 3. Check the robot for motor jam, brake not opened, etc. 4. If the robot starts suddenly after a long cold state, perform appropriate warm-up before running; if the robot's friction significantly decreases after long-term running, contact the manufacturer to tune the friction parameters. 5. If the robot operates at high speed and heavy load, it will cause current saturation phenomenon, which is prone to false alarms, so prevent the robot from operating under such conditions; You can also try to enter the medium precision mode through commands or manually turn on the speed feedforward of all axes to solve this problem to some extent. 6. If it is determined to be a false alarm, the collision detection sensitivity of the axis can be appropriately increased, and it is recommended to use the recommended collision detection sensitivity value. 7. If the alarm persists, turn off the collision detection switch for that axis.

Error code	Description	Cause	Solution
0x20C5	Abnormality detected on J5 axis, please confirm if a collision has occurred • If no collision is found, check whether the motor of this axis has failed, and check for any interference or pulling of external cables. • If no abnormalities are found, check whether the load parameters are set correctly and whether the collision detection parameters have been adjusted. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	Check whether a collision has occurred. • If no collision is found, check whether the motor of this axis has failed. • If no abnormalities are found, check whether the load parameters are set correctly, whether the collision detection parameters have been adjusted, and whether it is caused by limited servo tracking capability. • If no abnormalities are found, check whether the robot experiences significant friction changes after prolonged cold starts or extended operation. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	Check whether a collision has occurred. If not, check the following: 1. Check that the load parameters are set correctly. 2. Check that the robot model in the controller matches the actual robot. 3. Check the robot for motor jam, brake not opened, etc. 4. If the robot starts suddenly after a long cold state, perform appropriate warm-up before running; if the robot's friction significantly decreases after long-term running, contact the manufacturer to tune the friction parameters. 5. If the robot operates at high speed and heavy load, it will cause current saturation phenomenon, which is prone to false alarms, so prevent the robot from operating under such conditions; You can also try to enter the medium precision mode through commands or manually turn on the speed feedforward of all axes to solve this problem to some extent. 6. If it is determined to be a false alarm, the collision detection sensitivity of the axis can be appropriately increased, and it is recommended to use the recommended collision detection sensitivity value. 7. If the alarm persists, turn off the collision detection switch for that axis.

Error code	Description	Cause	Solution
0x20C6	Abnormality detected on J6 axis, please confirm if a collision has occurred • If no collision is found, check whether the motor of this axis has failed, and check for any interference or pulling of external cables. • If no abnormalities are found, check whether the load parameters are set correctly and whether the collision detection parameters have been adjusted. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	Check whether a collision has occurred. • If no collision is found, check whether the motor of this axis has failed. • If no abnormalities are found, check whether the load parameters are set correctly, whether the collision detection parameters have been adjusted, and whether it is caused by limited servo tracking capability. • If no abnormalities are found, check whether the robot experiences significant friction changes after prolonged cold starts or extended operation. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	Check whether a collision has occurred. If not, check the following: 1. Check that the load parameters are set correctly. 2. Check that the robot model in the controller matches the actual robot. 3. Check the robot for motor jam, brake not opened, etc. 4. If the robot starts suddenly after a long cold state, perform appropriate warm-up before running; if the robot's friction significantly decreases after long-term running, contact the manufacturer to tune the friction parameters. 5. If the robot operates at high speed and heavy load, it will cause current saturation phenomenon, which is prone to false alarms, so prevent the robot from operating under such conditions; You can also try to enter the medium precision mode through commands or manually turn on the speed feedforward of all axes to solve this problem to some extent. 6. If it is determined to be a false alarm, the collision detection sensitivity of the axis can be appropriately increased, and it is recommended to use the recommended collision detection sensitivity value. 7. If the alarm persists, turn off the collision detection switch for that axis.
0x20D0	Collision signal received	Collision signal received	Clear the collision signal.

0x20D1	Abnormal motion state detected on J1 axis	1. Robot collision; or abnormal current due to serious errors in load parameters, model parameters, and serious deviation of zero point. 2. Servo faults, possible types including: power line UVW phase sequence error, motor angle error, encoder model mismatch, encoder wiring failure, servo gain mismatch, etc. 3. Robot motion speed and acceleration are too large, exceeding the robot's motion capability, resulting in excessive speed deviation and torque deviation.	When troubleshooting this fault, make sure that the personnel are outside the working range of the robot before enabling any axis motor. 1. First check if a violent collision has occurred, if no collision occurs, check that the load parameters are set correctly, check that the robot matches the model displayed in the controller, and check if there is a serious deviation from the zero point of the robot. 2. If there is no collision and no obvious abnormality, try increasing the servo gain and turning on the servo speed feedforward (which can be turned on through the command to enter the medium precision mode) to see if the alarm still persists. 3. If the collision and controller side factors are excluded, then investigate the servo side factors in order. 1. The UVW phase sequence is incorrect. Connect the cables in the correct U V W phase sequence. 2. An error occurs on the initial phase detection of the motor rotor due to disturbing signals upon power-on. Power on the system again. 4. The encoder model is wrong or the encoder is wired improperly. Confirm that the motor model, parameter H00-00, encoder type, and encoder wiring are correct. 5. The encoder is wired improperly, aged, or connected loosely. Re-solder, tighten or replace the encoder cable.

Error code	Description	Cause	Solution
			6. Improper parameter setting leads to excessive vibration. Set the parameters appropriately to avoid excessive vibration. For more information, see "Robot Alarms and Handling Method" in the guide.

0x20D2	Abnormal motion state detected on J2 axis	1. Robot collision; or abnormal current due to serious errors in load parameters, model parameters, and serious deviation of zero point. 2. Servo faults, possible types including: power line UVW phase sequence error, motor angle error, encoder model mismatch, encoder wiring failure, servo gain mismatch, etc. 3. Robot motion speed and acceleration are too large, exceeding the robot's motion capability, resulting in excessive speed deviation and torque deviation.	When troubleshooting this fault, make sure that the personnel are outside the working range of the robot before enabling any axis motor. 1. First check if a violent collision has occurred, if no collision occurs, check that the load parameters are set correctly, check that the robot matches the model displayed in the controller, and check if there is a serious deviation from the zero point of the robot. 2. If there is no collision and no obvious abnormality, try increasing the servo gain and turning on the servo speed feedforward (which can be turned on through the command to enter the medium precision mode) to see if the alarm still persists. 3. If the collision and controller side factors are excluded, then investigate the servo side factors in order. 1. The UVW phase sequence is incorrect. Connect the cables in the correct U V W phase sequence. 2. An error occurs on the initial phase detection of the motor rotor due to disturbing signals upon power-on. Power on the system again. 4. The encoder model is wrong or the encoder is wired improperly. Confirm that the motor model, parameter H00-00, encoder type, and encoder wiring are correct. 5. The encoder is wired improperly, aged, or connected loosely. Re-solder, tighten or replace the encoder cable.

Error code	Description	Cause	Solution
			6. Improper parameter setting leads to excessive vibration. Set the parameters appropriately to avoid excessive vibration. For more information, see "Robot Alarms and Handling Method" in the guide.

0x20D3	Abnormal motion state detected on J3 axis	1. Robot collision; or abnormal current due to serious errors in load parameters, model parameters, and serious deviation of zero point. 2. Servo faults, possible types including: power line UVW phase sequence error, motor angle error, encoder model mismatch, encoder wiring failure, servo gain mismatch, etc. 3. Robot motion speed and acceleration are too large, exceeding the robot's motion capability, resulting in excessive speed deviation and torque deviation.	When troubleshooting this fault, make sure that the personnel are outside the working range of the robot before enabling any axis motor. 1. First check if a violent collision has occurred, if no collision occurs, check that the load parameters are set correctly, check that the robot matches the model displayed in the controller, and check if there is a serious deviation from the zero point of the robot. 2. If there is no collision and no obvious abnormality, try increasing the servo gain and turning on the servo speed feedforward (which can be turned on through the command to enter the medium precision mode) to see if the alarm still persists. 3. If the collision and controller side factors are excluded, then investigate the servo side factors in order. 1. The UVW phase sequence is incorrect. Connect the cables in the correct U V W phase sequence. 2. An error occurs on the initial phase detection of the motor rotor due to disturbing signals upon power-on. Power on the system again. 4. The encoder model is wrong or the encoder is wired improperly. Confirm that the motor model, parameter H00-00, encoder type, and encoder wiring are correct. 5. The encoder is wired improperly, aged, or connected loosely. Re-solder, tighten or replace the encoder cable.

Error code	Description	Cause	Solution
			6. Improper parameter setting leads to excessive vibration. Set the parameters appropriately to avoid excessive vibration. For more information, see "Robot Alarms and Handling Method" in the guide.

0x20D4	Abnormal motion state detected on J4 axis	1. Robot collision; or abnormal current due to serious errors in load parameters, model parameters, and serious deviation of zero point. 2. Servo faults, possible types including: power line UVW phase sequence error, motor angle error, encoder model mismatch, encoder wiring failure, servo gain mismatch, etc. 3. Robot motion speed and acceleration are too large, exceeding the robot's motion capability, resulting in excessive speed deviation and torque deviation.	When troubleshooting this fault, make sure that the personnel are outside the working range of the robot before enabling any axis motor. 1. First check if a violent collision has occurred, if no collision occurs, check that the load parameters are set correctly, check that the robot matches the model displayed in the controller, and check if there is a serious deviation from the zero point of the robot. 2. If there is no collision and no obvious abnormality, try increasing the servo gain and turning on the servo speed feedforward (which can be turned on through the command to enter the medium precision mode) to see if the alarm still persists. 3. If the collision and controller side factors are excluded, then investigate the servo side factors in order. 1. The UVW phase sequence is incorrect. Connect the cables in the correct U V W phase sequence. 2. An error occurs on the initial phase detection of the motor rotor due to disturbing signals upon power-on. Power on the system again. 4. The encoder model is wrong or the encoder is wired improperly. Confirm that the motor model, parameter H00-00, encoder type, and encoder wiring are correct. 5. The encoder is wired improperly, aged, or connected loosely. Re-solder, tighten or replace the encoder cable.

Error code	Description	Cause	Solution
			6. Improper parameter setting leads to excessive vibration. Set the parameters appropriately to avoid excessive vibration. For more information, see "Robot Alarms and Handling Method" in the guide.

0x20D5	Abnormal motion state detected on J5 axis	1. Robot collision; or abnormal current due to serious errors in load parameters, model parameters, and serious deviation of zero point. 2. Servo faults, including: power line UVW phase sequence error, motor angle error, encoder model mismatch, encoder wiring failure, servo gain mismatch, etc. 3. Robot motion speed and acceleration are too large, exceeding the robot's motion capability, resulting in excessive speed deviation and torque deviation.	When troubleshooting this fault, make sure that the personnel are outside the working range of the robot before enabling any axis motor. 1. First check if a violent collision has occurred, if no collision occurs, check that the load parameters are set correctly, check that the robot matches the model displayed in the controller, and check if there is a serious deviation from the zero point of the robot. 2. If there is no collision and no obvious abnormality, try increasing the servo gain and turning on the servo speed feedforward (which can be turned on through the command to enter the medium precision mode) to see if the alarm still persists. 3. If the collision and controller side factors are excluded, then investigate the servo side factors in order. 1). The UVW phase sequence is incorrect. Connect the U/V/W cables in the correct phase sequence. 2). An error occurs on the initial phase detection of the motor rotor due to disturbing signals upon power-on. Power on the system again. 4. The encoder model is wrong or the encoder is wired improperly. Confirm the motor model, the parameter H00-00, the encoder type, and the encoder wiring are correct. 5. The encoder is wired improperly, aged, or connected loosely. Re-solder, tighten or replace the encoder cable. 6. Improper parameter setting leads to excessive vibration. Set

Error code	Description	Cause	Solution
			the parameters appropriately to avoid excessive vibration. For more information, see "Robot Alarms and Handling Method" in the guide.

0x20D6	Abnormal motion state detected on J6 axis	1. Robot collision; or abnormal current due to serious errors in load parameters, model parameters, and serious deviation of zero point. 2. Servo faults, possible types including: power line UVW phase sequence error, motor angle error, encoder model mismatch, encoder wiring failure, servo gain mismatch, etc. 3. Robot motion speed and acceleration are too large, exceeding the robot's motion capability, resulting in excessive speed deviation and torque deviation.	When troubleshooting this fault, make sure that the personnel are outside the working range of the robot before enabling any axis motor. 1. First check if a violent collision has occurred, if no collision occurs, check that the load parameters are set correctly, check that the robot matches the model displayed in the controller, and check if there is a serious deviation from the zero point of the robot. 2. If there is no collision and no obvious abnormality, try increasing the servo gain and turning on the servo speed feedforward (which can be turned on through the command to enter the medium precision mode) to see if the alarm still persists. 3. If the collision and controller side factors are excluded, then investigate the servo side factors in order. 1. The UVW phase sequence is incorrect. Connect the cables in the correct U V W phase sequence. 2. An error occurs on the initial phase detection of the motor rotor due to disturbing signals upon power-on. Power on the system again. 4. The encoder model is wrong or the encoder is wired improperly. Confirm that the motor model, parameter H00-00, encoder type, and encoder wiring are correct. 5. The encoder is wired improperly, aged, or connected loosely. Re-solder, tighten or replace the encoder cable.

Error code	Description	Cause	Solution
			6. Improper parameter setting leads to excessive vibration. Set the parameters appropriately to avoid excessive vibration. For more information, see "Robot Alarms and Handling Method" in the guide.
0x20DA	Unsupported function	Functions currently not supported (such as tracking).	Delete related instructions containing unsupported functions.
0x20E0	CnvFastCatch instruction preprocessing planning error	No horizontal planning segment exists during CnvFastCatch instruction preprocessing.	Modify the start point of the CnvFastCatch instruction appropriately to ensure the motion of the instruction includes a horizontal segment during movement.
0x20E1	Conveyor index inconsistent	Conveyor index is inconsistent between GetCnvObject instruction and RefSys Conveyor instruction, or between GetCnvObject instruction and CnvFastCatch instruction.	Modify conveyor index to be consistent between the GetCnvObject instruction and the RefSys Conveyor and CnvFastCatch instructions.
0x20E2	Conveyor instruction conflict	RefSys Conveyor instruction and CnvFastCatch instruction are used at the same time.	RefSys Conveyor instruction and CnvFastCatch instruction must be separated by RefSys Base instruction.
0x20E3	Error in tool parameter of conveyor instruction	The tool used in tracking motion instruction is not a gripping tool.	Switch the tool used in tracking motion instruction to a gripping tool.
0x20E4	Error in parameter of workobject frame associated with the active conveyor	The attribute of workobject frame associated with the active conveyor: (1) the workobject is held; or (2) the user frame is not fixed.	Switch the workobject frame connected with the active conveyor to the workobject frame whose workobject is not held and user frame is fixed.
0x20E5	The tracking process does not support the installation of robot on a movable guide rail.	The tracking process does not support the installation of robot on a movable guide rail.	Change the installation mode of the robot to "Robot at fixed position" on the mechanical unit configuration interface.

Error code	Description	Cause	Solution
0x20E6	CnvTrigVis instruction use error	The CnvTrigVis instruction is not allowed for the conveyor whose detection mode is not "vision+instruction".	Change the detection mode of the conveyor to "vision+instruction" or switch to a conveyor with the detection mode of "vision+instruction".
0x20E7	Sensor triggered too frequently	The material is fed too fast, resulting in a sensor triggering cycle of less than 0.6s, which can lead to excessively high photography frequency.	Reduce the trigger frequency of sensor.
0x20E8	The holding time of DO level is too long.	The holding time of the DO level is greater than the trigger cycle of sensor, which will cause that the sensor does not trigger to take pictures in time.	Reduce the DO level holding time, or reduce the trigger frequency of sensor.
0x20E9	Vision data sent too many times	After the robot triggers a photo, it receives vision data more than once before the next photo is taken.	Check whether the visual data is sent multiple times or whether the photographing is abnormal, and whether the camera trigger parameter is selected correctly.
0x20EA	CnvFastCatch instruction not supported for non-SCARA robots	The CnvFastCatch instruction does not support robots other than SCARA robots.	Do not use CnvFastCatch instruction for non-SCARA robots.
0x20EB	SetCnvVisOTMode use error	The SetCnvVisOTMode instruction is not allowed if the detection mode is not "Vision", "Vision+Sensor", or "Instruction+Vision".	Switch the detection method of the conveyor to "Vision", "Vision+Sensor", or "Instruction+Vision".
0x20EC	The tracking process cannot be enabled when the soft float is enabled.	The soft float function and tracking process function are mutually exclusive.	Turn off the soft float function and then turn on the tracking process.
0x20ED	The encoder resolution should not be zero.	When the virtual conveyor speed is enabled, the encoder resolution should not be set to zero.	Adjust the encoder resolution to a non-zero value, or disable the virtual conveyor speed.
0x20EE	The speed setting is exceeded in the tracking state.	The robot approaches the singular position in the tracking state.	Adjust the lower workspace boundary so that the robot's singularity is not within the workspace of the tracking process.

Error code	Description	Cause	Solution
0x20EF	The acceleration setting is exceeded in the tracking state.	The robot approaches the singular position in the tracking state.	Adjust the lower workspace boundary so that the robot's singularity is not within the workspace of the tracking process.
0x20F1	Jump/JumpL motion is forbidden in ECP frame.	Jump/JumpL motion is forbidden in ECP frame.	Change the frame or change the motion instructions.
0x20F2	Tool frame and wobj frame do not match.	Tool frame and wobj frame do not match.	Change the frame.
0x20F3	External axis motion is forbidden in ECP frame.	External axis motion is forbidden in ECP frame.	Change the frame or change the motion instructions.
0x20F4	Tracking motion is forbidden in ECP frame.	Tracking motion is forbidden in ECP frame.	Change the frame or change the motion instructions.
0x20F5	Abnormal system time fluctuation	Excessive fluctuation of system scheduling time	Export the diagnostic information and contact the manufacturer.
0x20F6	NC thread scheduling time exception	NC thread scheduling time fluctuates too much.	Export the diagnostic information and contact the manufacturer.
0x20F7	NRT thread execution time exception	The NRT thread execution timed out.	Export the diagnostic information and contact the manufacturer.
0x20F8	Real time task execution time exception	Real time task execution timeout	Export the diagnostic information and contact the manufacturer.
0x20F9	Abnormal system timer feedback	System clock_nanosleep return value exception	Export the diagnostic information and contact the manufacturer.
0x20FA	Rtbus thread execution time exception	Rtbus thread execution timeout	Export the diagnostic information and contact the manufacturer.
0x20FC	Subdivision configuration information verification error	Subdivision configuration information verification error	Export the diagnostic information and contact the manufacturer.
0x20FD	Singularity crossing function fails to reach the taught joint positions	Singularity crossing function fails to reach the taught joint positions	Turn off singularity crossing function or change the target point.

Error code	Description	Cause	Solution
0x20FE	MCU configuration information check error	MCU configuration information check error	Contact the manufacturer.
0x2101	J1 axis positive limit alarm	The joint limit position is reached.	Perform negative motion to clear the alarm.
0x2102	J1 axis negative limit alarm	The joint limit position is reached.	Perform positive motion to clear the alarm.
0x2103	J1 axis drive alarm	The drive has an alarm.	Perform troubleshooting by referring to the servo guide according to fault codes.
0x2104	J1 axis path planning out of range	The planned value exceeds the maximum calculation range (-1073741823 to +1073741824).	Check whether the absolute origin position is selected to be near the counting limit. If the zero point is near the limit, clear the drive turns.
0x2105	Excessive tracking error of J1 axis	The difference between the planned and actual positions is too large.	1. View the tracking error threshold. 2. Check for mechanical stuck. 3. Check that the power lines are properly connected. 4. Check that the load matches the servo gain. 5. Consult the troubleshooting guide.
0x2106	J1 axis overspeed alarm	The running speed is greater than the set maximum speed.	Reduce the speed of the J1 axis.
0x2107	Strong current of J1 axis not switched on	Strong current is not switched on.	Check the drive circuit for whether strong current is not switched on.
0x2108	J1 axis torque overlimit	The actual maximum current exceeds the limit.	1. Reduce the acceleration parameters appropriately. 2. Check that the load parameter settings are appropriate. 3. If the optimal trajectory is enabled, refer to the modulation guidance of the optimal trajectory.

Error code	Description	Cause	Solution
0x2109	J1 axis average load rate overlimit	Average load rate exceeds temperature rise and service life limits.	1. Troubleshoot brake, mechanical jamming, and drive. 2. Reduce the speed or acceleration percentage appropriately. 3. Extend the non-motion time. In the pure motion beat test, the average load rate may exceed the limit, which does not mean that it exceeds the limit in actual application. You can add delay such as waiting and logical interaction in actual application to the test program and test again.
0x210A	J1 axis speed overlimit	When verifying the planned output, it was found that the set joint speed was exceeded.	1. In case of CP motion, reduce position speed or increase joint speed appropriately, or adjust the points to move away from singularities. 2. Check if there are any other accompanying alarms in the alarm record.
0x210B	J1 axis acceleration overlimit	When verifying the planned output, it was found that the set joint acceleration was exceeded.	1. In case of CP motion, reduce position speed or increase joint speed appropriately, or adjust the points to move away from singularities. 2. Check if there are any other accompanying alarms in the alarm record.
0x210C	J1 axis enable synchronization error	The robot vibrates before it is enabled, or the robot moves due to external reasons during the enabling process.	Check whether there is joint movement during power on, and whether there is external interference during the enabling process.
0x210D	J1 axis arrival timeout	The servo takes too long to arrive the target position.	Check whether the arrival error threshold is too small or whether the servo gain is not proper.

Error code	Description	Cause	Solution
0x210E	J1 axis position feedback error	1. Multi-turn value changes. 2. Encoder position jumps.	Check if the multi turn value has been manually cleared. If not, check for the interference source.
0x210F	J1 axis disable status error	The robot failed to be disabled at the specified time and distance.	1. Check that servo parameter 6084 and brake parameter 0209 are correctly set. 2. Check if the load exceeds the limit.
0x2111	J2 axis positive limit alarm	The joint limit position is reached.	Perform negative motion to clear the alarm.
0x2112	J2 axis negative drive has an alarm	The joint limit position is reached.	Perform positive motion to clear the alarm.
0x2113	J2 axis drive alarm	The drive has an alarm.	Perform troubleshooting by referring to the servo guide according to fault codes.
0x2114	J2 axis path planning out of range	The planned value exceeds the maximum calculation range.	Check whether the absolute origin position is selected to be near the counting limit. If the zero point is near the limit, clear the drive turns.
0x2115	Excessive tracking error of J2 axis	The difference between the planned and actual positions is too large.	1. View the tracking error threshold. 2. Check for mechanical stuck. 3. Check that the power lines are properly connected. 4. Check that the load matches the servo gain. 5. Consult the troubleshooting guide.
0x2116	J2 axis overspeed alarm	The running speed is greater than the set maximum speed.	Reduce the speed of J2 axis.
0x2117	Strong current of J2 axis not switched on	Strong current is not switched on.	Check the drive circuit for whether strong current is not switched on.

Error code	Description	Cause	Solution
0x2118	J2 axis torque overlimit	The actual maximum current exceeds the limit.	1. Reduce the acceleration parameters appropriately. 2. Check that the load parameter settings are appropriate. 3. If the optimal trajectory is enabled, refer to the modulation guidance of the optimal trajectory.
0x2119	J2 axis average load rate overlimit	Average load rate exceeds temperature rise and service life limits.	1. Troubleshoot brake, mechanical jamming, and drive. 2. Reduce the speed or acceleration percentage appropriately. 3. Extend the non-motion time. In the pure motion beat test, the average load rate may exceed the limit, which does not mean that it exceeds the limit in actual application. You can add delay such as waiting and logical interaction in actual application to the test program and test again.
0x211A	J2 axis speed overlimit	When verifying the planned output, it was found that the set joint speed was exceeded.	1. In case of CP motion, reduce position speed or increase joint speed appropriately, or adjust the points to move away from singularities. 2. Check if there are any other accompanying alarms in the alarm record.
0x211B	J2 axis acceleration overlimit	When verifying the planned output, it was found that the set joint acceleration was exceeded.	1. In case of CP motion, reduce position speed or increase joint speed appropriately, or adjust the points to move away from singularities. 2. Check if there are any other accompanying alarms in the alarm record.

Error code	Description	Cause	Solution
0x211C	J2 axis enable synchronization error	The robot vibrates before it is enabled, or the robot moves due to external reasons during the enabling process.	Check whether there is joint movement during power on, and whether there is external interference during the enabling process.
0x211D	J2 axis arrival timeout	The servo takes too long to arrive the target position.	Check whether the arrival error threshold is too small or whether the servo gain is not proper.
0x211E	J2 axis position feedback error	1. Multi-turn value changes. 2. Encoder position jumps.	Check if the multi turn value has been manually cleared. If not, check for the interference source.
0x211F	J2 axis disable status error	The robot failed to be disabled at the specified time and distance.	1. Check that servo parameter 6084 and brake parameter 0209 are correctly set. 2. Check if the load exceeds the limit.
0x2121	J1 axis motor/reducer overheat	The motor/reducer temperature is too high, exceeding alarm threshold.	1. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 2. Reduce the duty cycle. 3. Check whether the ambient temperature of the robot is too high and reduce the temperature if necessary.
0x2122	J1 axis encoder overheat	The encoder temperature is too high, exceeding alarm threshold.	1. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 2. Reduce the duty cycle. 3. Check whether the ambient temperature of the robot is too high and reduce the temperature if necessary.

Error code	Description	Cause	Solution
0x2141	J2 axis motor/reducer overheat	The motor/reducer temperature is too high, exceeding alarm threshold.	1. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 2. Reduce the duty cycle. 3. Check whether the ambient temperature of the robot is too high and reduce the temperature if necessary.
0x2142	J2 axis encoder overheat	The encoder temperature is too high, exceeding alarm threshold.	1. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 2. Reduce the duty cycle. 3. Check whether the ambient temperature of the robot is too high and reduce the temperature if necessary.
0x2161	J3 axis motor/reducer overheat	The motor/reducer temperature is too high, exceeding alarm threshold.	1. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 2. Reduce the duty cycle. 3. Check whether the ambient temperature of the robot is too high and reduce the temperature if necessary.
0x2162	J3 axis encoder overheat	The encoder temperature is too high, exceeding alarm threshold.	1. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 2. Reduce the duty cycle. 3. Check whether the ambient temperature of the robot is too high and reduce the temperature if necessary.

Error code	Description	Cause	Solution
0x2181	J4 axis motor/reducer overheat	The motor/reducer temperature is too high, exceeding alarm threshold.	1. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 2. Reduce the duty cycle. 3. Check whether the ambient temperature of the robot is too high and reduce the temperature if necessary.
0x2182	J4 axis encoder overheat	The encoder temperature is too high, exceeding alarm threshold.	1. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 2. Reduce the duty cycle. 3. Check whether the ambient temperature of the robot is too high and reduce the temperature if necessary.
0x21A1	J5 axis motor/reducer overheat	The motor/reducer temperature is too high, exceeding alarm threshold.	1. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 2. Reduce the duty cycle. 3. Check whether the ambient temperature of the robot is too high and reduce the temperature if necessary.
0x21A2	J5 axis encoder overheat	The encoder temperature is too high, exceeding alarm threshold.	1. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 2. Reduce the duty cycle. 3. Check whether the ambient temperature of the robot is too high and reduce the temperature if necessary.

Error code	Description	Cause	Solution
0x21C1	J6 axis motor/reducer overheat	The motor/reducer temperature is too high, exceeding alarm threshold.	1. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 2. Reduce the duty cycle. 3. Check whether the ambient temperature of the robot is too high and reduce the temperature if necessary.
0x21C2	J6 axis encoder overheat	The encoder temperature is too high, exceeding alarm threshold.	1. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 2. Reduce the duty cycle. 3. Check whether the ambient temperature of the robot is too high and reduce the temperature if necessary.
0x21E1	Robot entered interference zone 0	Robot entered interference zone 0	Disable the interference zone 0, and move the robot out of interference zone 0.
0x21E2	Robot entered interference zone 1	Robot entered interference zone 1	Disable the interference zone 1, and move the robot out of interference zone 1.
0x21E3	Robot entered interference zone 2	Robot entered interference zone 2	Disable the interference zone 2, and move the robot out of interference zone 2.
0x21E4	Robot entered interference zone 3	Robot entered interference zone 3	Disable the interference zone 3, and move the robot out of interference zone 3.
0x21E5	Robot entered interference zone 4	Robot entered interference zone 4	Disable the interference zone 4, and move the robot out of interference zone 4.
0x21E6	Robot entered interference zone 5	Robot entered interference zone 5	Disable the interference zone 5, and move the robot out of interference zone 5.

Error code	Description	Cause	Solution
0x21E7	Robot entered interference zone 6	Robot entered interference zone 6	Disable the interference zone 6, and move the robot out of interference zone 6.
0x21E8	Robot entered interference zone 7	Robot entered interference zone 7	Disable the interference zone 7, and move the robot out of interference zone 7.
0x21E9	Robot entered interference zone 8	Robot entered interference zone 8	Disable the interference zone 8, and move the robot out of interference zone 8.
0x21EA	Robot entered interference zone 9	Robot entered interference zone 9	Disable the interference zone 9, and move the robot out of interference zone 9.
0x21EB	Robot entered interference zone 10	Robot entered interference zone 10	Disable the interference zone 10, and move the robot out of interference zone 10.
0x21EC	Robot entered interference zone 11	Robot entered interference zone 11	Disable the interference zone 11, and move the robot out of interference zone 11.
0x21ED	Robot entered interference zone 12	Robot entered interference zone 12	Disable the interference zone 12, and move the robot out of interference zone 12.
0x21EE	Robot entered interference zone 13	Robot entered interference zone 13	Disable the interference zone 13, and move the robot out of interference zone 13.
0x21EF	Robot entered interference zone 14	Robot entered interference zone 14	Disable the interference zone 14, and move the robot out of interference zone 14.
0x21F0	Robot entered interference zone 15	Robot entered interference zone 15	Disable the interference zone 15, and move the robot out of interference zone 15.
0x2201	J3 axis positive limit alarm	The joint limit position is reached.	Perform negative motion to clear the alarm.
0x2202	J3 axis negative limit alarm	The joint limit position is reached.	Perform positive motion to clear the alarm.
0x2203	J3 axis drive alarm	The drive has an alarm.	Perform troubleshooting by referring to the servo guide according to fault codes.

Error code	Description	Cause	Solution
0x2204	J3 axis path planning out of range	The planned value exceeds the maximum calculation range (-1073741823 to +1073741824).	Check whether the absolute origin position is selected to be near the counting limit. If the zero point is near the limit, clear the drive turns.
0x2205	Excessive tracking error of J3 axis	The difference between the planned and actual positions is too large.	1. View the tracking error threshold. 2. Check for mechanical stuck. 3. Check that the power lines are properly connected. 4. Check that the load matches the servo gain. 5. Consult the troubleshooting guide.
0x2206	J3 axis overspeed alarm	The running speed is greater than the set maximum speed.	Reduce the speed of J3 axis.
0x2207	Strong current of J3 axis not switched on	Strong current is not switched on.	Check the drive circuit for whether strong current is not switched on.
0x2208	J3 axis torque overlimit	The actual maximum current exceeds the limit.	1. Reduce the acceleration parameters appropriately. 2. Check that the load parameter settings are appropriate. 3. If the optimal trajectory is enabled, refer to the modulation guidance of the optimal trajectory.
0x2209	J3 axis average load rate overlimit	Average load rate exceeds temperature rise and service life limits.	1. Troubleshoot brake, mechanical jamming, and drive. 2. Reduce the speed or acceleration percentage appropriately. 3. Extend the non-motion time. In the pure motion beat test, the average load rate may exceed the limit, which does not mean that it exceeds the limit in actual application. You can add delay such as waiting and logical interaction in actual application to the test program and test again.

Error code	Description	Cause	Solution
0x220A	J3 axis speed overlimit	When verifying the planned output, it was found that the set joint speed was exceeded.	1. In case of CP motion, reduce position speed or increase joint speed appropriately, or adjust the points to move away from singularities. 2. Check if there are any other accompanying alarms in the alarm record.
0x220B	J3 axis acceleration overlimit	When verifying the planned output, it was found that the set joint acceleration was exceeded.	1. In case of CP motion, reduce position speed or increase joint speed appropriately, or adjust the points to move away from singularities. 2. Check if there are any other accompanying alarms in the alarm record.
0x220C	J3 axis enable synchronization error	The robot vibrates before it is enabled, or the robot moves due to external reasons during the enabling process.	Check whether there is joint movement during power on, and whether there is external interference during the enabling process.
0x220D	J3 axis arrival timeout	The servo takes too long to arrive the target position.	Check whether the arrival error threshold is too small or whether the servo gain is not proper.
0x220E	J3 axis position feedback error	1. Multi-turn value changes. 2. Encoder position jumps.	Check if the multi turn value has been manually cleared. If not, check for the interference source.
0x220F	J3 axis disable status error	The robot failed to be disabled at the specified time and distance.	1. Check that servo parameter 6084 and brake parameter 0209 are correctly set. 2. Check if the load exceeds the limit.
0x2211	J4 axis positive limit alarm	The joint limit position is reached.	Perform negative motion to clear the alarm.
0x2212	J4 axis negative limit alarm	The joint limit position is reached.	Perform negative motion to clear the alarm.
0x2213	J4 axis drive alarm	The drive has an alarm.	Perform troubleshooting by referring to the servo guide according to fault codes.

Error code	Description	Cause	Solution
0x2214	J4 axis path planning out of range	The planned value exceeds the maximum calculation range (-1073741823 to +1073741824).	Check whether the absolute origin position is selected to be near the counting limit. If the zero point is near the limit, clear the drive turns.
0x2215	Excessive tracking error of J4 axis	The difference between the planned and actual positions is too large.	1. View the tracking error threshold. 2. Check for mechanical stuck. 3. Check that the power lines are properly connected. 4. Check that the load matches the servo gain. 5. Consult the troubleshooting guide.
0x2216	J4 axis overspeed alarm	The running speed is greater than the set maximum speed.	Reduce the speed of J4 axis.
0x2217	Strong current of J4 axis not switched on	Strong current is not switched on.	Check the drive circuit for whether strong current is not switched on.
0x2218	J4 axis torque overlimit	The actual maximum current exceeds the limit.	1. Reduce the acceleration parameters appropriately. 2. Check that the load parameter settings are appropriate. 3. If the optimal trajectory is enabled, refer to the modulation guidance of the optimal trajectory.
0x2219	J4 axis average load rate overlimit	Average load rate exceeds temperature rise and service life limits.	1. Troubleshoot brake, mechanical jamming, and drive. 2. Reduce the speed or acceleration percentage appropriately. 3. Extend the non-motion time. In the pure motion beat test, the average load rate may exceed the limit, which does not mean that it exceeds the limit in actual application. You can add delay such as waiting and logical interaction in actual application to the test program and test again.

Error code	Description	Cause	Solution
0x221A	J4 axis speed overlimit	When verifying the planned output, it was found that the set joint speed was exceeded.	1. In case of CP motion, reduce position speed or increase joint speed appropriately, or adjust the points to move away from singularities. 2. Check if there are any other accompanying alarms in the alarm record.
0x221B	J4 axis acceleration overlimit	When verifying the planned output, it was found that the set joint acceleration was exceeded.	1. In case of CP motion, reduce position speed or increase joint speed appropriately, or adjust the points to move away from singularities. 2. Check if there are any other accompanying alarms in the alarm record.
0x221C	J4 axis enable synchronization error	The robot vibrates before it is enabled, or the robot moves due to external reasons during the enabling process.	Check whether there is joint movement during power on, and whether there is external interference during the enabling process.
0x221D	J4 axis arrival timeout	The servo takes too long to arrive the target position.	Check whether the arrival error threshold is too small or whether the servo gain is not proper.
0x221E	J4 axis position feedback error	1. Multi-turn value changes. 2. Encoder position jumps.	Check if the multi turn value has been manually cleared. If not, check for the interference source.
0x221F	J4 axis disable status error	The robot failed to be disabled at the specified time and distance.	1. Check that servo parameter 6084 and brake parameter 0209 are correctly set. 2. Check if the load exceeds the limit.

Error code	Description	Cause	Solution
0x22C1	An abnormality is detected in the J1 axis while both the robot and base are in motion, indicating a follow-up collision at the J1 axis. Check that the external axis is in motion and check for any collisions. • If no collision is found, check whether the motor of this axis has failed, and check for any interference or pulling of external cables. • If no abnormalities are found, check whether the load parameters are set correctly and whether the collision detection parameters have been adjusted. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	An abnormality is detected in the J1 axis while both the robot and base are in motion, indicating a follow-up collision at the J1 axis. Check that the external axis is in motion and check for any collisions. • If no collision is found, check whether the motor of this axis has failed. • If no abnormalities are found, check whether the load parameters are set correctly, whether the collision detection parameters have been adjusted, and whether it is caused by limited servo tracking capability. • If no abnormalities are found, check whether the robot experiences significant friction changes after prolonged cold starts or extended operation. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	Check that the external axis is in motion and check for any collisions. If no collision is found, check the following: 1. Check that the load parameters are set correctly. 2. Check that the robot model in the controller matches the actual robot. 3. Check the robot for motor jam, brake not opened, etc. 4. If the robot starts suddenly after a long cold state, perform appropriate warm-up before running; if the robot's friction significantly decreases after long-term running, contact the manufacturer to tune the friction parameters. 5. If the robot operates at high speed and heavy load, it will cause current saturation phenomenon, which is prone to false alarms, so prevent the robot from operating under such conditions; You can also try to enter the medium precision mode through commands or manually turn on the speed feedforward of all axes to solve this problem to some extent. 6. If it is determined to be a false alarm, the collision detection sensitivity of the axis can be appropriately increased, and it is recommended to use the recommended collision detection sensitivity value. 7. If the alarm persists, turn off the collision detection switch for that axis.

Error code	Description	Cause	Solution
0x22C2	An abnormality is detected in the J2 axis while both the robot and base are in motion, indicating a follow-up collision at the J2 axis. Check that the external axis is in motion and check for any collisions. • If no collision is found, check whether the motor of this axis has failed, and check for any interference or pulling of external cables. • If no abnormalities are found, check whether the load parameters are set correctly and whether the collision detection parameters have been adjusted. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	An abnormality is detected in the J2 axis while both the robot and base are in motion, indicating a follow-up collision at the J2 axis. Check that the external axis is in motion and check for any collisions. • If no collision is found, check whether the motor of this axis has failed. • If no abnormalities are found, check whether the load parameters are set correctly, whether the collision detection parameters have been adjusted, and whether it is caused by limited servo tracking capability. • If no abnormalities are found, check whether the robot experiences significant friction changes after prolonged cold starts or extended operation. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	Check that the external axis is in motion and check for any collisions. If no collision is found, check the following: 1. Check that the load parameters are set correctly. 2. Check that the robot model in the controller matches the actual robot. 3. Check the robot for motor jam, brake not opened, etc. 4. If the robot starts suddenly after a long cold state, perform appropriate warm-up before running; if the robot's friction significantly decreases after long-term running, contact the manufacturer to tune the friction parameters. 5. If the robot operates at high speed and heavy load, it will cause current saturation phenomenon, which is prone to false alarms, so prevent the robot from operating under such conditions; You can also try to enter the medium precision mode through commands or manually turn on the speed feedforward of all axes to solve this problem to some extent. 6. If it is determined to be a false alarm, the collision detection sensitivity of the axis can be appropriately increased, and it is recommended to use the recommended collision detection sensitivity value. 7. If the alarm persists, turn off the collision detection switch for that axis.

Error code	Description	Cause	Solution
0x22C3	An abnormality is detected in the J3 axis while both the robot and base are in motion, indicating a follow-up collision at the J3 axis. Check that the external axis is in motion and check for any collisions. • If no collision is found, check whether the motor of this axis has failed, and check for any interference or pulling of external cables. • If no abnormalities are found, check whether the load parameters are set correctly and whether the collision detection parameters have been adjusted. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	An abnormality is detected in the J3 axis while both the robot and base are in motion, indicating a follow-up collision at the J3 axis. Check that the external axis is in motion and check for any collisions. • If no collision is found, check whether the motor of this axis has failed. • If no abnormalities are found, check whether the load parameters are set correctly, whether the collision detection parameters have been adjusted, and whether it is caused by limited servo tracking capability. • If no abnormalities are found, check whether the robot experiences significant friction changes after prolonged cold starts or extended operation. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	Check that the external axis is in motion and check for any collisions. If no collision is found, check the following: 1. Check that the load parameters are set correctly. 2. Check that the robot model in the controller matches the actual robot. 3. Check the robot for motor jam, brake not opened, etc. 4. If the robot starts suddenly after a long cold state, perform appropriate warm-up before running; if the robot's friction significantly decreases after long-term running, contact the manufacturer to tune the friction parameters. 5. If the robot operates at high speed and heavy load, it will cause current saturation phenomenon, which is prone to false alarms, so prevent the robot from operating under such conditions; You can also try to enter the medium precision mode through commands or manually turn on the speed feedforward of all axes to solve this problem to some extent. 6. If it is determined to be a false alarm, the collision detection sensitivity of the axis can be appropriately increased, and it is recommended to use the recommended collision detection sensitivity value. 7. If the alarm persists, turn off the collision detection switch for that axis.

Error code	Description	Cause	Solution
0x22C4	An abnormality is detected in the J4 axis while both the robot and base are in motion, indicating a follow-up collision at the J4 axis. Check that the external axis is in motion and check for any collisions. • If no collision is found, check whether the motor of this axis has failed, and check for any interference or pulling of external cables. • If no abnormalities are found, check whether the load parameters are set correctly and whether the collision detection parameters have been adjusted. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	An abnormality is detected in the J4 axis while both the robot and base are in motion, indicating a follow-up collision at the J4 axis. Check that the external axis is in motion and check for any collisions. • If no collision is found, check whether the motor of this axis has failed. • If no abnormalities are found, check whether the load parameters are set correctly, whether the collision detection parameters have been adjusted, and whether it is caused by limited servo tracking capability. • If no abnormalities are found, check whether the robot experiences significant friction changes after prolonged cold starts or extended operation. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	Check that the external axis is in motion and check for any collisions. If no collision is found, check the following: 1. Check that the load parameters are set correctly; Check that the robot model in the controller matches the actual robot. 2. Check the robot for motor jam, brake not opened, etc. 3. If the robot starts suddenly after a long cold state, perform appropriate warm-up before running; if the robot's friction significantly decreases after long-term running, contact the manufacturer to tune the friction parameters. 4. If the robot operates at high speed and heavy load, it will cause current saturation phenomenon, which is prone to false alarms, so prevent the robot from operating under such conditions; You can also try to enter the medium precision mode through commands or manually turn on the speed feedforward of all axes to solve this problem to some extent. 5. If it is determined to be a false alarm, the collision detection sensitivity of the axis can be appropriately increased, and it is recommended to use the recommended collision detection sensitivity value. 6. If the alarm persists, turn off the collision detection switch for that axis.

Error code	Description	Cause	Solution
0x22C5	An abnormality is detected in the J5 axis while both the robot and base are in motion, indicating a follow-up collision at the J5 axis. Check that the external axis is in motion and check for any collisions. • If no collision is found, check whether the motor of this axis has failed, and check for any interference or pulling of external cables. • If no abnormalities are found, check whether the load parameters are set correctly and whether the collision detection parameters have been adjusted. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	An abnormality is detected in the J5 axis while both the robot and base are in motion, indicating a follow-up collision at the J5 axis. Check that the external axis is in motion and check for any collisions. • If no collision is found, check whether the motor of this axis has failed. • If no abnormalities are found, check whether the load parameters are set correctly, whether the collision detection parameters have been adjusted, and whether it is caused by limited servo tracking capability. • If no abnormalities are found, check whether the robot experiences significant friction changes after prolonged cold starts or extended operation. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	Check that the external axis is in motion and check for any collisions. If no collision is found, check the following: 1. Check that the load parameters are set correctly. 2. Check that the robot model in the controller matches the actual robot. 3. Check the robot for motor jam, brake not opened, etc. 4. If the robot starts suddenly after a long cold state, perform appropriate warm-up before running; if the robot's friction significantly decreases after long-term running, contact the manufacturer to tune the friction parameters. 5. If the robot operates at high speed and heavy load, it will cause current saturation phenomenon, which is prone to false alarms, so prevent the robot from operating under such conditions; You can also try to enter the medium precision mode through commands or manually turn on the speed feedforward of all axes to solve this problem to some extent. 6. If it is determined to be a false alarm, the collision detection sensitivity of the axis can be appropriately increased, and it is recommended to use the recommended collision detection sensitivity value. 7. If the alarm persists, turn off the collision detection switch for that axis.

Error code	Description	Cause	Solution
0x22C6	An abnormality is detected in the J6 axis while both the robot and base are in motion, indicating a follow-up collision at the J6 axis. Check that the external axis is in motion and check for any collisions. • If no collision is found, check whether the motor of this axis has failed, and check for any interference or pulling of external cables. • If no abnormalities are found, check whether the load parameters are set correctly and whether the collision detection parameters have been adjusted. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	An abnormality is detected in the J6 axis while both the robot and base are in motion, indicating a follow-up collision at the J6 axis. Check that the external axis is in motion and check for any collisions. • If no collision is found, check whether the motor of this axis has failed. • If no abnormalities are found, check whether the load parameters are set correctly, whether the collision detection parameters have been adjusted, and whether it is caused by limited servo tracking capability. • If no abnormalities are found, check whether the robot experiences significant friction changes after prolonged cold starts or extended operation. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	Check that the external axis is in motion and check for any collisions. If no collision is found, check the following: 1. Check that the load parameters are set correctly. 2. Check that the robot model in the controller matches the actual robot. 3. Check the robot for motor jam, brake not opened, etc. 4. If the robot starts suddenly after a long cold state, perform appropriate warm-up before running; if the robot's friction significantly decreases after long-term running, contact the manufacturer to tune the friction parameters. 5. If the robot operates at high speed and heavy load, it will cause current saturation phenomenon, which is prone to false alarms, so prevent the robot from operating under such conditions; You can also try to enter the medium precision mode through commands or manually turn on the speed feedforward of all axes to solve this problem to some extent. 6. If it is determined to be a false alarm, the collision detection sensitivity of the axis can be appropriately increased, and it is recommended to use the recommended collision detection sensitivity value. 7. If the alarm persists, turn off the collision detection switch for that axis.

Error code	Description	Cause	Solution
0x2301	J5 axis positive limit alarm	The joint limit position is reached.	Perform negative motion to clear the alarm.
0x2302	J5 axis negative limit alarm	The joint limit position is reached.	Perform positive motion to clear the alarm.
0x2303	J5 axis drive alarm	The drive has an alarm.	Perform troubleshooting by referring to the servo guide according to fault codes.
0x2304	J5 axis path planning out of range	The planned value exceeds the maximum calculation range (-1073741823 to +1073741824).	Check whether the absolute origin position is selected to be near the counting limit. If the zero point is near the limit, clear the drive turns.
0x2305	Excessive tracking error of J5 axis	The difference between the planned and actual positions is too large.	1. View the tracking error threshold. 2. Check for mechanical stuck. 3. Check that the power lines are properly connected. 4. Check that the load matches the servo gain. 5. Consult the troubleshooting guide.
0x2306	J5 axis overspeed alarm	The running speed is greater than the set maximum speed.	Reduce the speed of J5 axis.
0x2307	Strong current of J5 axis not switched on	Strong current is not switched on.	Check the drive circuit for whether strong current is not switched on.
0x2308	J5 axis torque overlimit	The actual maximum current exceeds the limit.	1. Reduce the acceleration parameters appropriately. 2. Check that the load parameter settings are appropriate. 3. If the optimal trajectory is enabled, refer to the modulation guidance of the optimal trajectory.

Error code	Description	Cause	Solution
0x2309	J5 axis average load rate overlimit	Average load rate exceeds temperature rise and service life limits.	1. Troubleshoot brake, mechanical jamming, and drive. 2. Reduce the speed or acceleration percentage appropriately. 3. Extend the non-motion time. In the pure motion beat test, the average load rate may exceed the limit, which does not mean that it exceeds the limit in actual application. You can add delay such as waiting and logical interaction in actual application to the test program and test again.
0x230A	J5 axis speed overlimit	When verifying the planned output, it was found that the set joint speed was exceeded.	1. In case of CP motion, reduce position speed or increase joint speed appropriately, or adjust the points to move away from singularities. 2. Check if there are any other accompanying alarms in the alarm record.
0x230B	J5 axis acceleration overlimit	When verifying the planned output, it was found that the set joint acceleration was exceeded.	1. In case of CP motion, reduce position speed or increase joint speed appropriately, or adjust the points to move away from singularities. 2. Check if there are any other accompanying alarms in the alarm record.
0x230C	J5 axis enable synchronization error	The robot vibrates before it is enabled, or the robot moves due to external reasons during the enabling process.	Check whether there is joint movement during power on, and whether there is external interference during the enabling process.
0x230D	J5 axis arrival timeout	The servo takes too long to arrive the target position.	Check whether the arrival error threshold is too small or whether the servo gain is not proper.

Error code	Description	Cause	Solution
0x230E	J5 axis position feedback error	1. Multi-turn value changes. 2. Encoder position jumps.	Check if the multi turn value has been manually cleared. If not, check for the interference source.
0x230F	J5 axis disable status error	The robot failed to be disabled at the specified time and distance.	1. Check that servo parameter 6084 and brake parameter 0209 are correctly set. 2. Check if the load exceeds the limit.
0x2311	J6 axis positive limit alarm	The joint limit position is reached.	Perform negative motion to clear the alarm.
0x2312	J6 axis negative limit alarm	The joint limit position is reached.	Perform positive motion to clear the alarm.
0x2313	J6 axis drive alarm	The drive has an alarm.	Perform troubleshooting by referring to the servo guide according to fault codes.
0x2314	J6 axis path planning out of range	The planned value exceeds the maximum calculation range (-1073741823 to +1073741824).	Check whether the absolute origin position is selected to be near the counting limit. If the zero point is near the limit, clear the drive turns.
0x2315	Excessive tracking error of J6 axis	The difference between the planned and actual positions is too large.	1. View the tracking error threshold. 2. Check for mechanical stuck. 3. Check that the power lines are properly connected. 4. Check that the load matches the servo gain. 5. Consult the troubleshooting guide.
0x2316	J6 axis overspeed alarm	The running speed is greater than the set maximum speed.	Reduce the speed of J6 axis.
0x2317	Strong current of J6 axis is not switched on.	Strong current is not switched on.	Check the drive circuit for whether strong current is not switched on.

Error code	Description	Cause	Solution
0x2318	J6 axis torque overlimit	The actual maximum current exceeds the limit.	1. Reduce the acceleration parameters appropriately. 2. Check that the load parameter settings are appropriate. 3. If the optimal trajectory is enabled, refer to the modulation guidance of the optimal trajectory.
0x2319	J6 axis average load rate overlimit	When verifying the planned output, it was found that the set joint speed was exceeded.	1. Troubleshoot brake, mechanical jamming, and drive. 2. Reduce the speed or acceleration percentage appropriately. 3. Extend the non-motion time. In the pure motion beat test, the average load rate may exceed the limit, which does not mean that it exceeds the limit in actual application. You can add delay such as waiting and logical interaction in actual application to the test program and test again.
0x231A	J6 axis speed overlimit	When verifying the planned output, it was found that the set joint speed was exceeded.	1. In case of CP motion, reduce position speed or increase joint speed appropriately, or adjust the points to move away from singularities. 2. Check if there are any other accompanying alarms in the alarm record.
0x231B	J6 axis acceleration overlimit	When verifying the planned output, it was found that the set joint acceleration was exceeded.	1. In case of CP motion, reduce position speed or increase joint speed appropriately, or adjust the points to move away from singularities. 2. Check if there are any other accompanying alarms in the alarm record.
0x231C	J6 axis enable synchronization error	Excessive tracking error	Check whether there is joint movement during power on.

Error code	Description	Cause	Solution
0x231D	J6 axis arrival timeout	The servo takes too long to arrive the target position.	Check whether the arrival error threshold is too small or whether the servo gain is not proper.
0x231E	J6 axis position feedback error	1. Multi-turn value changes. 2. Encoder position jumps.	Check if the multi turn value has been manually cleared. If not, check for the interference source.
0x231F	J6 axis disable status error	The robot failed to be disabled at the specified time and distance.	1. Check that servo parameter 6084 and brake parameter 0209 are correctly set. 2. Check if the load exceeds the limit.

Error code	Description	Cause	Solution
0x23C1	An abnormality is detected in the J1 axis while the robot itself is not in motion, yet the base is detected to be in motion, indicating a passive collision at the J1 axis. Check that the base is in motion (e.g., the robot is being moved by a rail) and check for any collisions. • If no collision is found, check whether the motor of this axis has failed, and check for any interference or pulling of external cables. • If no abnormalities are found, check whether the load parameters are set correctly and whether the collision detection parameters have been adjusted. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	An abnormality is detected in the J1 axis while the robot itself is not in motion, yet the base is detected to be in motion. Check that the base is in motion (e.g., the robot is being moved by a rail) and check for any collisions. • If no collision is found, check whether the motor of this axis has failed. • If no abnormalities are found, check whether the load parameters are set correctly, whether the collision detection parameters have been adjusted, and whether it is caused by limited servo tracking capability. • If no abnormalities are found, check whether the robot experiences significant friction changes after prolonged cold starts or extended operation. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	Check that the base is in motion (e.g., the robot is being moved by a rail) and check for any collisions. If no collision is found, check the following: 1. Check that the load parameters are set correctly. 2. Check that the robot model in the controller matches the actual robot. 3. Check the robot for motor jam, brake not opened, etc. 4. If the robot starts suddenly after a long cold state, perform appropriate warm-up before running; if the robot's friction significantly decreases after long-term running, contact the manufacturer to tune the friction parameters. 5. If the robot operates at high speed and heavy load, it will cause current saturation phenomenon, which is prone to false alarms, so prevent the robot from operating under such conditions; You can also try to enter the medium precision mode through commands or manually turn on the speed feedforward of all axes to solve this problem to some extent. 6. If it is determined to be a false alarm, the collision detection sensitivity of the axis can be appropriately increased, and it is recommended to use the recommended collision detection sensitivity value. 7. If the alarm persists, turn off the collision detection switch for that axis.

Error code	Description	Cause	Solution
0x23C2	An abnormality is detected in the J2 axis while the robot itself is not in motion, yet the base is detected to be in motion, indicating a passive collision at the J2 axis. Check that the base is in motion (e.g., the robot is being moved by a rail) and check for any collisions. • If no collision is found, check whether the motor of this axis has failed, and check for any interference or pulling of external cables. • If no abnormalities are found, check whether the load parameters are set correctly and whether the collision detection parameters have been adjusted. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	An abnormality is detected in the J2 axis while the robot itself is not in motion, yet the base is detected to be in motion. Check that the base is in motion (e.g., the robot is being moved by a rail) and check for any collisions. • If no collision is found, check whether the motor of this axis has failed. • If no abnormalities are found, check whether the load parameters are set correctly, whether the collision detection parameters have been adjusted, and whether it is caused by limited servo tracking capability. • If no abnormalities are found, check whether the robot experiences significant friction changes after prolonged cold starts or extended operation. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	Check that the base is in motion (e.g., the robot is being moved by a rail) and check for any collisions. If no collision is found, check the following: 1. Check that the load parameters are set correctly. 2. Check that the robot model in the controller matches the actual robot. 3. Check the robot for motor jam, brake not opened, etc. 4. If the robot starts suddenly after a long cold state, perform appropriate warm-up before running; if the robot's friction significantly decreases after long-term running, contact the manufacturer to tune the friction parameters. 5. If the robot operates at high speed and heavy load, it will cause current saturation phenomenon, which is prone to false alarms, so prevent the robot from operating under such conditions; You can also try to enter the medium precision mode through commands or manually turn on the speed feedforward of all axes to solve this problem to some extent. 6. If it is determined to be a false alarm, the collision detection sensitivity of the axis can be appropriately increased, and it is recommended to use the recommended collision detection sensitivity value. 7. If the alarm persists, turn off the collision detection switch for that axis.

Error code	Description	Cause	Solution
0x23C3	An abnormality is detected in the J3 axis while the robot itself is not in motion, yet the base is detected to be in motion, indicating a passive collision at the J3 axis. Check that the base is in motion (e.g., the robot is being moved by a rail) and check for any collisions. • If no collision is found, check whether the motor of this axis has failed, and check for any interference or pulling of external cables. • If no abnormalities are found, check whether the load parameters are set correctly and whether the collision detection parameters have been adjusted. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	An abnormality is detected in the J3 axis while the robot itself is not in motion, yet the base is detected to be in motion. Check that the base is in motion (e.g., the robot is being moved by a rail) and check for any collisions. • If no collision is found, check whether the motor of this axis has failed. • If no abnormalities are found, check whether the load parameters are set correctly, whether the collision detection parameters have been adjusted, and whether it is caused by limited servo tracking capability. • If no abnormalities are found, check whether the robot experiences significant friction changes after prolonged cold starts or extended operation. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	Check that the base is in motion (e.g., the robot is being moved by a rail) and check for any collisions. If no collision is found, check the following: 1. Check that the load parameters are set correctly. 2. Check that the robot model in the controller matches the actual robot. 3. Check the robot for motor jam, brake not opened, etc. 4. If the robot starts suddenly after a long cold state, perform appropriate warm-up before running; if the robot's friction significantly decreases after long-term running, contact the manufacturer to tune the friction parameters. 5. If the robot operates at high speed and heavy load, it will cause current saturation phenomenon, which is prone to false alarms, so prevent the robot from operating under such conditions; You can also try to enter the medium precision mode through commands or manually turn on the speed feedforward of all axes to solve this problem to some extent. 6. If it is determined to be a false alarm, the collision detection sensitivity of the axis can be appropriately increased, and it is recommended to use the recommended collision detection sensitivity value. 7. If the alarm persists, turn off the collision detection switch for that axis.

Error code	Description	Cause	Solution
0x23C4	An abnormality is detected in the J4 axis while the robot itself is not in motion, yet the base is detected to be in motion, indicating a passive collision at the J4 axis. Check that the base is in motion (e.g., the robot is being moved by a rail) and check for any collisions. • If no collision is found, check whether the motor of this axis has failed, and check for any interference or pulling of external cables. • If no abnormalities are found, check whether the load parameters are set correctly and whether the collision detection parameters have been adjusted. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	An abnormality is detected in the J4 axis while the robot itself is not in motion, yet the base is detected to be in motion. Check that the base is in motion (e.g., the robot is being moved by a rail) and check for any collisions. • If no collision is found, check whether the motor of this axis has failed. • If no abnormalities are found, check whether the load parameters are set correctly, whether the collision detection parameters have been adjusted, and whether it is caused by limited servo tracking capability. • If no abnormalities are found, check whether the robot experiences significant friction changes after prolonged cold starts or extended operation. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	Check that the base is in motion (e.g., the robot is being moved by a rail) and check for any collisions. If no collision is found, check the following: 1. Check that the load parameters are set correctly. 2. Check that the robot model in the controller matches the actual robot. 3. Check the robot for motor jam, brake not opened, etc. 4. If the robot starts suddenly after a long cold state, perform appropriate warm-up before running; if the robot's friction significantly decreases after long-term running, contact the manufacturer to tune the friction parameters. 5. If the robot operates at high speed and heavy load, it will cause current saturation phenomenon, which is prone to false alarms, so prevent the robot from operating under such conditions; You can also try to enter the medium precision mode through commands or manually turn on the speed feedforward of all axes to solve this problem to some extent. 6. If it is determined to be a false alarm, the collision detection sensitivity of the axis can be appropriately increased, and it is recommended to use the recommended collision detection sensitivity value. 7. If the alarm persists, turn off the collision detection switch for that axis.

Error code	Description	Cause	Solution
0x23C5	An abnormality is detected in the J5 axis while the robot itself is not in motion, yet the base is detected to be in motion, indicating a passive collision at the J5 axis. Check that the base is in motion (e.g., the robot is being moved by a rail) and check for any collisions. • If no collision is found, check whether the motor of this axis has failed, and check for any interference or pulling of external cables. • If no abnormalities are found, check whether the load parameters are set correctly and whether the collision detection parameters have been adjusted. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	An abnormality is detected in the J5 axis while the robot itself is not in motion, yet the base is detected to be in motion. Check that the base is in motion (e.g., the robot is being moved by a rail) and check for any collisions. • If no collision is found, check whether the motor of this axis has failed. • If no abnormalities are found, check whether the load parameters are set correctly, whether the collision detection parameters have been adjusted, and whether it is caused by limited servo tracking capability. • If no abnormalities are found, check whether the robot experiences significant friction changes after prolonged cold starts or extended operation. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	Check that the base is in motion (e.g., the robot is being moved by a rail) and check for any collisions. If no collision is found, check the following: 1. Check that the load parameters are set correctly. 2. Check that the robot model in the controller matches the actual robot. 3. Check the robot for motor jam, brake not opened, etc. 4. If the robot starts suddenly after a long cold state, perform appropriate warm-up before running; if the robot's friction significantly decreases after long-term running, contact the manufacturer to tune the friction parameters. 5. If the robot operates at high speed and heavy load, it will cause current saturation phenomenon, which is prone to false alarms, so prevent the robot from operating under such conditions; You can also try to enter the medium precision mode through commands or manually turn on the speed feedforward of all axes to solve this problem to some extent. 6. If it is determined to be a false alarm, the collision detection sensitivity of the axis can be appropriately increased, and it is recommended to use the recommended collision detection sensitivity value. 7. If the alarm persists, turn off the collision detection switch for that axis.

Error code	Description	Cause	Solution
0x23C6	An abnormality is detected in the J6 axis while the robot itself is not in motion, yet the base is detected to be in motion, indicating a passive collision at the J6 axis. Check that the base is in motion (e.g., the robot is being moved by a rail) and check for any collisions. • If no collision is found, check whether the motor of this axis has failed, and check for any interference or pulling of external cables. • If no abnormalities are found, check whether the load parameters are set correctly and whether the collision detection parameters have been adjusted. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	An abnormality is detected in the J6 axis while the robot itself is not in motion, yet the base is detected to be in motion. Check that the base is in motion (e.g., the robot is being moved by a rail) and check for any collisions. • If no collision is found, check whether the motor of this axis has failed. • If no abnormalities are found, check whether the load parameters are set correctly, whether the collision detection parameters have been adjusted, and whether it is caused by limited servo tracking capability. • If no abnormalities are found, check whether the robot experiences significant friction changes after prolonged cold starts or extended operation. • If the issue persists, see the user guide for troubleshooting or adjust the collision detection parameters.	Check that the base is in motion (e.g., the robot is being moved by a rail) and check for any collisions. If no collision is found, check the following: 1. Check that the load parameters are set correctly. 2. Check that the robot model in the controller matches the actual robot. 3. Check the robot for motor jam, brake not opened, etc. 4. If the robot starts suddenly after a long cold state, perform appropriate warm-up before running; if the robot's friction significantly decreases after long-term running, contact the manufacturer to tune the friction parameters. 5. If the robot operates at high speed and heavy load, it will cause current saturation phenomenon, which is prone to false alarms, so prevent the robot from operating under such conditions; You can also try to enter the medium precision mode through commands or manually turn on the speed feedforward of all axes to solve this problem to some extent. 6. If it is determined to be a false alarm, the collision detection sensitivity of the axis can be appropriately increased, and it is recommended to use the recommended collision detection sensitivity value. 7. If the alarm persists, turn off the collision detection switch for that axis.

Error code	Description	Cause	Solution
0x23E1	J1 axis torque overlimit in manual mode	The actual maximum current exceeds the limit.	1. Reduce the acceleration parameters appropriately. 2. Check that the load parameter settings are appropriate. 3. If the optimal trajectory is enabled, refer to the modulation guidance of the optimal trajectory.
0x23E2	J2 axis torque overlimit in manual mode	The actual maximum current exceeds the limit.	1. Reduce the acceleration parameters appropriately. 2. Check that the load parameter settings are appropriate. 3. If the optimal trajectory is enabled, refer to the modulation guidance of the optimal trajectory.
0x23E3	J3 axis torque overlimit in manual mode	The actual maximum current exceeds the limit.	1. Reduce the acceleration parameters appropriately. 2. Check that the load parameter settings are appropriate. 3. If the optimal trajectory is enabled, refer to the modulation guidance of the optimal trajectory.
0x23E4	J4 axis torque overlimit in manual mode	The actual maximum current exceeds the limit.	1. Reduce the acceleration parameters appropriately. 2. Check that the load parameter settings are appropriate. 3. If the optimal trajectory is enabled, refer to the modulation guidance of the optimal trajectory.
0x23E5	J5 axis torque overlimit in manual mode	The actual maximum current exceeds the limit.	1. Reduce the acceleration parameters appropriately. 2. Check that the load parameter settings are appropriate. 3. If the optimal trajectory is enabled, refer to the modulation guidance of the optimal trajectory.

Error code	Description	Cause	Solution
0x23E6	J6 axis torque overlimit in manual mode	The actual maximum current exceeds the limit.	1. Reduce the acceleration parameters appropriately. 2. Check that the load parameter settings are appropriate. 3. If the optimal trajectory is enabled, refer to the modulation guidance of the optimal trajectory.
0x2401	J7 axis positive limit alarm	The joint limit position is reached.	Perform negative motion to clear the alarm.
0x2402	J7 axis negative limit alarm	The joint limit position is reached.	Perform positive motion to clear the alarm.
0x2403	J7 axis drive alarm	The drive has an alarm.	Perform troubleshooting by referring to the servo guide according to fault codes.
0x2404	J7 axis path planning out of range	The planned value exceeds the maximum calculation range (-1073741823 to +1073741824).	Check whether the absolute origin position is selected to be near the counting limit. If the zero point is near the limit, clear the drive turns.
0x2405	Excessive tracking error of J7 axis	The difference between the planned and actual positions is too large.	1. View the tracking error threshold. 2. Check for mechanical stuck. 3. Check that the power lines are properly connected. 4. Check that the load matches the servo gain. 5. Consult the troubleshooting guide.
0x2406	J7 axis overspeed alarm	The running speed is greater than the set maximum speed.	Reduce the speed of J7 axis.
0x2407	Strong current of J7 axis not switched on	Strong current is not switched on.	Check the drive circuit for whether strong current is not switched on.

Error code	Description	Cause	Solution
0x2408	J7 axis torque overlimit	The actual maximum current exceeds the limit.	1. Reduce the acceleration parameters appropriately. 2. Check that the load parameter settings are appropriate. 3. If the optimal trajectory is enabled, refer to the modulation guidance of the optimal trajectory.
0x2409	J7 axis average load rate overlimit	Average load rate exceeds temperature rise and service life limits.	1. Troubleshoot brake, mechanical jamming, and drive. 2. Reduce the speed or acceleration percentage appropriately. 3. Extend the non-motion time. In the pure motion beat test, the average load rate may exceed the limit, which does not mean that it exceeds the limit in actual application. You can add delay such as waiting and logical interaction in actual application to the test program and test again.
0x240A	J7 axis planned speed error	The planned speed exceeds the limit too much.	Reduce the maximum speed in joint or Cartesian space.
0x240B	J7 axis planned acceleration error	The planned acceleration exceeds the limit too much.	Reduce the maximum acceleration or velocity in joint or Cartesian space.
0x240C	J7 axis enable synchronization error	The robot vibrates before it is enabled, or the robot moves due to external reasons during the enabling process.	Check whether there is joint movement during power on, and whether there is external interference during the enabling process.
0x240D	J7 axis arrival timeout	The servo takes too long to arrive the target position.	Check whether the arrival error threshold is too small or whether the servo gain is not proper.
0x240E	J7 axis position feedback error	1. Multi-turn value changes. 2. Encoder position jumps.	Check if the multi turn value has been manually cleared. If not, check for the interference source.

Error code	Description	Cause	Solution
0x240F	J7 axis disable status error	The robot failed to be disabled at the specified time and distance.	1. Check that servo parameter 6084 and brake parameter 0209 are correctly set. 2. Check if the load exceeds the limit.
0x2411	J8 axis positive limit alarm	The joint limit position is reached.	Perform negative motion to clear the alarm.
0x2412	J8 axis negative limit alarm	The joint limit position is reached.	Perform positive motion to clear the alarm.
0x2413	J8 axis drive alarm	The drive has an alarm.	Perform troubleshooting by referring to the servo guide according to fault codes.
0x2414	J8 axis path planning out of range	The planned value exceeds the maximum calculation range (-1073741823 to +1073741824).	Check whether the absolute origin position is selected to be near the counting limit. If the zero point is near the limit, clear the drive turns.
0x2415	Excessive tracking error of J8 axis	The difference between the planned and actual positions is too large.	1. View the tracking error threshold. 2. Check for mechanical stuck. 3. Check that the power lines are properly connected. 4. Check that the load matches the servo gain. 5. Consult the troubleshooting guide.
0x2416	J8 axis overspeed alarm	The running speed is greater than the set maximum speed.	Reduce the joint speed.
0x2417	Strong current of J8 axis not switched on	Strong current is not switched on.	Check the drive circuit for whether strong current is not switched on.
0x2418	J8 axis torque overlimit	The actual maximum current exceeds the limit.	1. Reduce the acceleration parameters appropriately. 2. Check that the load parameter settings are appropriate. 3. If the optimal trajectory is enabled, refer to the modulation guidance of the optimal trajectory.

Error code	Description	Cause	Solution
0x2419	J8 axis average load rate overlimit	Average load rate exceeds temperature rise and service life limits.	1. Troubleshoot brake, mechanical jamming, and drive. 2. Reduce the speed or acceleration percentage appropriately. 3. Extend the non-motion time. In the pure motion beat test, the average load rate may exceed the limit, which does not mean that it exceeds the limit in actual application. You can add delay such as waiting and logical interaction in actual application to the test program and test again.
0x241A	J8 axis planned speed error	The planned speed exceeds the limit too much.	Reduce the maximum speed in joint or Cartesian space.
0x241B	J8 axis planned acceleration error	The planned acceleration exceeds the limit too much.	Reduce the maximum acceleration or velocity in joint or Cartesian space.
0x241C	J8 axis enable synchronization error	The robot vibrates before it is enabled, or the robot moves due to external reasons during the enabling process.	Check whether there is joint movement during power on, and whether there is external interference during the enabling process.
0x241D	J8 axis arrival timeout	The servo takes too long to arrive the target position.	Check whether the arrival error threshold is too small or whether the servo gain is not proper.
0x241E	J8 axis position feedback error	1. Multi-turn value changes. 2. Encoder position jumps.	Check if the multi turn value has been manually cleared. If not, check for the interference source.
0x241F	J8 axis disable status error	The robot failed to be disabled at the specified time and distance.	1. Check that servo parameter 6084 and brake parameter 0209 are correctly set. 2. Check if the load exceeds the limit.
0x2501	J9 axis positive limit alarm	The joint limit position is reached.	Perform negative motion to clear the alarm.
0x2502	J9 axis negative limit alarm	The joint limit position is reached.	Perform positive motion to clear the alarm.

Error code	Description	Cause	Solution
0x2503	J9 axis drive alarm	The drive has an alarm.	Perform troubleshooting by referring to the servo guide according to fault codes.
0x2504	J9 axis path planning out of range	The planned value exceeds the maximum calculation range (-1073741823 to +1073741824).	Check whether the absolute origin position is selected to be near the counting limit. If the zero point is near the limit, clear the drive turns.
0x2505	Excessive tracking error of J9 axis	The difference between the planned and actual positions is too large.	1. View the tracking error threshold. 2. Check for mechanical stuck. 3. Check that the power lines are properly connected. 4. Check that the load matches the servo gain. 5. Consult the troubleshooting guide.
0x2506	J9 axis overspeed alarm	The running speed is greater than the set maximum speed.	Reduce the speed of J9 axis.
0x2507	Strong current of J9 axis not switched on	Strong current is not switched on.	Check the drive circuit for whether strong current is not switched on.
0x2508	J9 axis torque overlimit	The actual maximum current exceeds the limit.	1. Reduce the acceleration parameters appropriately. 2. Check that the load parameter settings are appropriate. 3. If the optimal trajectory is enabled, refer to the modulation guidance of the optimal trajectory.

Error code	Description	Cause	Solution
0x2509	J9 axis average load rate overlimit	Average load rate exceeds temperature rise and service life limits.	1. Troubleshoot brake, mechanical jamming, and drive. 2. Reduce the speed or acceleration percentage appropriately. 3. Extend the non-motion time. In the pure motion beat test, the average load rate may exceed the limit, which does not mean that it exceeds the limit in actual application. You can add delay such as waiting and logical interaction in actual application to the test program and test again.
0x250A	J9 axis planned speed error	The planned speed exceeds the limit too much.	Reduce the maximum speed in joint or Cartesian space.
0x250B	J9 axis planned acceleration error	The planned acceleration exceeds the limit too much.	Reduce the maximum acceleration or velocity in joint or Cartesian space.
0x250C	J9 axis enable synchronization error	The robot vibrates before it is enabled, or the robot moves due to external reasons during the enabling process.	Check whether there is joint movement during power on, and whether there is external interference during the enabling process.
0x250D	J9 axis arrival timeout	The servo takes too long to arrive the target position.	Check whether the arrival error threshold is too small or whether the servo gain is not proper.
0x250E	J9 axis encoder feedback error	1. Multi-turn value changes. 2. Encoder position jumps.	Check if the multi turn value has been manually cleared. If not, check for the interference source.
0x250F	J9 axis disable status error	The robot failed to be disabled at the specified time and distance.	1. Check that servo parameter 6084 and brake parameter 0209 are correctly set. 2. Check if the load exceeds the limit.
0x2511	J10 axis positive limit alarm	The joint limit position is reached.	Perform negative motion to clear the alarm.
0x2512	J10 axis negative limit alarm	The joint limit position is reached.	Perform positive motion to clear the alarm.

Error code	Description	Cause	Solution
0x2513	J10 axis drive alarm	The drive has an alarm.	Perform troubleshooting by referring to the servo guide according to fault codes.
0x2514	J10 axis path planning out of range	The planned value exceeds the maximum calculation range (-1073741823 to +1073741824).	Check whether the absolute origin position is selected to be near the counting limit. If the zero point is near the limit, clear the drive turns.
0x2515	Excessive tracking error of J10 axis	The difference between the planned and actual positions is too large.	1. View the tracking error threshold. 2. Check for mechanical stuck. 3. Check that the power lines are properly connected. 4. Check that the load matches the servo gain. 5. Consult the troubleshooting guide.
0x2516	J10 axis overspeed alarm	The running speed is greater than the set maximum speed.	Reduce the speed of J10 axis.
0x2517	Strong current of J10 axis not switched on	Strong current is not switched on.	Check the drive circuit for whether strong current is not switched on.
0x2518	J10 axis torque overlimit	The actual maximum current exceeds the limit.	1. Reduce the acceleration parameters appropriately. 2. Check that the load parameter settings are appropriate. 3. If the optimal trajectory is enabled, refer to the modulation guidance of the optimal trajectory.

Error code	Description	Cause	Solution
0x2519	J10 axis average load rate overlimit	Average load rate exceeds temperature rise and service life limits.	1. Troubleshoot brake, mechanical jamming, and drive. 2. Reduce the speed or acceleration percentage appropriately. 3. Extend the non-motion time. In the pure motion beat test, the average load rate may exceed the limit, which does not mean that it exceeds the limit in actual application. You can add delay such as waiting and logical interaction in actual application to the test program and test again.
0x251A	J10 axis planned speed error	The planned speed exceeds the limit too much.	Reduce the maximum speed in joint or Cartesian space.
0x251B	J10 axis planned acceleration error	The planned acceleration exceeds the limit too much.	Reduce the maximum acceleration or velocity in joint or Cartesian space.
0x251C	J10 axis enable synchronization error	The robot vibrates before it is enabled, or the robot moves due to external reasons during the enabling process.	Check whether there is joint movement during power on, and whether there is external interference during the enabling process.
0x251D	J10 axis arrival timeout	The servo takes too long to arrive the target position.	Check whether the arrival error threshold is too small or whether the servo gain is not proper.
0x251E	J10 axis encoder feedback error	1. Multi-turn value changes. 2. Encoder position jumps.	Check if the multi turn value has been manually cleared. If not, check for the interference source.
0x251F	J10 axis disable status error	The robot failed to be disabled at the specified time and distance.	1. Check that servo parameter 6084 and brake parameter 0209 are correctly set. 2. Check if the load exceeds the limit.
0x2601	J11 axis positive limit alarm	The joint limit position is reached.	Perform negative motion to clear the alarm.
0x2602	J11 axis negative limit alarm	The joint limit position is reached.	Perform positive motion to clear the alarm.

Error code	Description	Cause	Solution
0x2603	J11 axis drive alarm	The drive has an alarm.	Perform troubleshooting by referring to the servo guide according to fault codes.
0x2604	J11 axis path planning out of range	The planned value exceeds the maximum calculation range (-1073741823 to +1073741824).	Check whether the absolute origin position is selected to be near the counting limit. If the zero point is near the limit, clear the drive turns.
0x2605	Excessive tracking error of J11 axis	The difference between the planned and actual positions is too large.	1. View the tracking error threshold. 2. Check for mechanical stuck. 3. Check that the power lines are properly connected. 4. Check that the load matches the servo gain. 5. Consult the troubleshooting guide.
0x2606	J11 axis overspeed alarm	The running speed is greater than the set maximum speed.	Reduce the speed of J11 axis.
0x2607	Strong current of J11 axis not switched on	Strong current is not switched on.	Check the drive circuit for whether strong current is not switched on.
0x2608	J11 axis torque overlimit	The actual maximum current exceeds the limit.	1. Reduce the acceleration parameters appropriately. 2. Check that the load parameter settings are appropriate. 3. If the optimal trajectory is enabled, refer to the modulation guidance of the optimal trajectory.

Error code	Description	Cause	Solution
0x2609	J11 axis average load rate overlimit	Average load rate exceeds temperature rise and service life limits.	1. Troubleshoot brake, mechanical jamming, and drive. 2. Reduce the speed or acceleration percentage appropriately. 3. Extend the non-motion time. In the pure motion beat test, the average load rate may exceed the limit, which does not mean that it exceeds the limit in actual application. You can add delay such as waiting and logical interaction in actual application to the test program and test again.
0x260A	J11 axis planned speed error	The planned speed exceeds the limit too much.	Reduce the maximum speed in joint or Cartesian space.
0x260B	J11 axis planned acceleration error	The planned acceleration exceeds the limit too much.	Reduce the maximum acceleration or velocity in joint or Cartesian space.
0x260C	J11 axis enable synchronization error	The robot vibrates before it is enabled, or the robot moves due to external reasons during the enabling process.	Check whether there is joint movement during power on, and whether there is external interference during the enabling process.
0x260D	J11 axis arrival timeout	The servo takes too long to arrive the target position.	Check whether the arrival error threshold is too small or whether the servo gain is not proper.
0x260E	J11 axis encoder feedback error	1. Multi-turn value changes. 2. Encoder position jumps.	Check if the multi turn value has been manually cleared. If not, check for the interference source.
0x260F	J11 axis disable status error	The robot failed to be disabled at the specified time and distance.	1. Check that servo parameter 6084 and brake parameter 0209 are correctly set. 2. Check if the load exceeds the limit.
0x2611	J12 axis positive limit alarm	The joint limit position is reached.	Perform negative motion to clear the alarm.
0x2612	J12 axis negative limit alarm	The joint limit position is reached.	Perform positive motion to clear the alarm.

Error code	Description	Cause	Solution
0x2613	J12 axis drive alarm	The drive has an alarm.	Perform troubleshooting by referring to the servo guide according to fault codes.
0x2614	J12 axis path planning out of range	The planned value exceeds the maximum calculation range (-1073741823 to +1073741824).	Check whether the absolute origin position is selected to be near the counting limit. If the zero point is near the limit, clear the drive turns.
0x2615	Excessive tracking error of J12 axis	The difference between the planned and actual positions is too large.	1. View the tracking error threshold. 2. Check for mechanical stuck. 3. Check that the power lines are properly connected. 4. Check that the load matches the servo gain. 5. Consult the troubleshooting guide.
0x2616	J12 axis overspeed alarm	The running speed is greater than the set maximum speed.	Reduce the speed of J12 axis.
0x2617	Strong current of J12 axis not switched on	Strong current is not switched on.	Check the drive circuit for whether strong current is not switched on.
0x2618	J12 axis torque overlimit	The actual maximum current exceeds the limit.	1. Reduce the acceleration parameters appropriately. 2. Check that the load parameter settings are appropriate. 3. If the optimal trajectory is enabled, refer to the modulation guidance of the optimal trajectory.

Error code	Description	Cause	Solution
0x2619	J12 axis average load rate overlimit	Average load rate exceeds temperature rise and service life limits.	1. Troubleshoot brake, mechanical jamming, and drive. 2. Reduce the speed or acceleration percentage appropriately. 3. Extend the non-motion time. In the pure motion beat test, the average load rate may exceed the limit, which does not mean that it exceeds the limit in actual application. You can add delay such as waiting and logical interaction in actual application to the test program and test again.
0x261A	J12 axis planned speed error	The planned speed exceeds the limit too much.	Reduce the maximum speed in joint or Cartesian space.
0x261B	J12 axis planned acceleration error	The planned acceleration exceeds the limit too much.	Reduce the maximum acceleration or velocity in joint or Cartesian space.
0x261C	J12 axis enable synchronization error	The robot vibrates before it is enabled, or the robot moves due to external reasons during the enabling process.	Check whether there is joint movement during power on, and whether there is external interference during the enabling process.
0x261D	J12 axis arrival timeout	The servo takes too long to arrive the target position.	Check whether the arrival error threshold is too small or whether the servo gain is not proper.
0x261E	J12 axis encoder feedback error	1. Multi-turn value changes. 2. Encoder position jumps.	Check if the multi turn value has been manually cleared. If not, check for the interference source.
0x261F	J12 axis disable status error	The robot failed to be disabled at the specified time and distance.	1. Check that servo parameter 6084 and brake parameter 0209 are correctly set. 2. Check if the load exceeds the limit.
0x2701	J13 axis positive limit alarm	The joint limit position is reached.	Perform negative motion to clear the alarm.
0x2702	J13 axis negative limit alarm	The joint limit position is reached.	Perform positive motion to clear the alarm.

Error code	Description	Cause	Solution
0x2703	J13 axis drive alarm	The drive has an alarm.	Perform troubleshooting by referring to the servo guide according to fault codes.
0x2704	J13 axis path planning out of range	The planned value exceeds the maximum calculation range (-1073741823 to +1073741824).	Check whether the absolute origin position is selected to be near the counting limit. If the zero point is near the limit, clear the drive turns.
0x2705	Excessive tracking error of J13 axis	The difference between the planned and actual positions is too large.	1. View the tracking error threshold. 2. Check for mechanical stuck. 3. Check that the power lines are properly connected. 4. Check that the load matches the servo gain. 5. Consult the troubleshooting guide.
0x2706	J13 axis overspeed alarm	The running speed is greater than the set maximum speed.	Reduce the speed of J13 axis.
0x2707	Strong current of J13 axis not switched on	Strong current is not switched on.	Check the drive circuit for whether strong current is not switched on.
0x2708	J13 axis torque overlimit	The actual maximum current exceeds the limit.	1. Reduce the acceleration parameters appropriately. 2. Check that the load parameter settings are appropriate. 3. If the optimal trajectory is enabled, refer to the modulation guidance of the optimal trajectory.

Error code	Description	Cause	Solution
0x2709	J13 axis average load rate overlimit	Average load rate exceeds temperature rise and service life limits.	1. Troubleshoot brake, mechanical jamming, and drive. 2. Reduce the speed or acceleration percentage appropriately. 3. Extend the non-motion time. In the pure motion beat test, the average load rate may exceed the limit, which does not mean that it exceeds the limit in actual application. You can add delay such as waiting and logical interaction in actual application to the test program and test again.
0x270A	J13 axis planned speed error	The planned speed exceeds the limit too much.	Reduce the maximum speed in joint or Cartesian space.
0x270B	J13 axis planned acceleration error	The planned acceleration exceeds the limit too much.	Reduce the maximum acceleration or velocity in joint or Cartesian space.
0x270C	J13 axis enable synchronization error	The robot vibrates before it is enabled, or the robot moves due to external reasons during the enabling process.	Check whether there is joint movement during power on, and whether there is external interference during the enabling process.
0x270D	J13 axis arrival timeout	The servo takes too long to arrive the target position.	Check whether the arrival error threshold is too small or whether the servo gain is not proper.
0x270E	J13 axis encoder feedback error	1. Multi-turn value changes. 2. Encoder position jumps.	Check if the multi turn value has been manually cleared. If not, check for the interference source.
0x270F	J13 axis disable status error	The robot failed to be disabled at the specified time and distance.	1. Check that servo parameter 6084 and brake parameter 0209 are correctly set. 2. Check if the load exceeds the limit.
0x2711	J14 axis positive limit alarm	The joint limit position is reached.	Perform negative motion to clear the alarm.
0x2712	J14 axis negative limit alarm	The joint limit position is reached.	Perform positive motion to clear the alarm.

Error code	Description	Cause	Solution
0x2713	J14 axis drive alarm	The drive has an alarm.	Perform troubleshooting by referring to the servo guide according to fault codes.
0x2714	J14 axis path planning out of range	The planned value exceeds the maximum calculation range (-1073741823 to +1073741824).	Check whether the absolute origin position is selected to be near the counting limit. If the zero point is near the limit, clear the drive turns.
0x2715	Excessive tracking error of J14 axis	The difference between the planned and actual positions is too large.	1. View the tracking error threshold. 2. Check for mechanical stuck. 3. Check that the power lines are properly connected. 4. Check that the load matches the servo gain. 5. Consult the troubleshooting guide.
0x2716	J14 axis overspeed alarm	The running speed is greater than the set maximum speed.	Reduce the speed of J14 axis.
0x2717	Strong current of Axis 14 is not switched on.	Strong current is not switched on.	Check the drive circuit for whether strong current is not switched on.
0x2718	J14 axis torque overlimit	The actual maximum current exceeds the limit.	1. Reduce the acceleration parameters appropriately. 2. Check that the load parameter settings are appropriate. 3. If the optimal trajectory is enabled, refer to the modulation guidance of the optimal trajectory.

Error code	Description	Cause	Solution
0x2719	J14 axis average load rate overlimit	Average load rate exceeds temperature rise and service life limits.	1. Troubleshoot brake, mechanical jamming, and drive. 2. Reduce the speed or acceleration percentage appropriately. 3. Extend the non-motion time. In the pure motion beat test, the average load rate may exceed the limit, which does not mean that it exceeds the limit in actual application. You can add delay such as waiting and logical interaction in actual application to the test program and test again.
0x271A	J14 axis planned speed error	The planned speed exceeds the limit too much.	Reduce the maximum speed in joint or Cartesian space.
0x271B	J14 axis planned acceleration error	The planned acceleration exceeds the limit too much.	Reduce the maximum acceleration or velocity in joint or Cartesian space.
0x271C	J14 axis enable synchronization error	The robot vibrates before it is enabled, or the robot moves due to external reasons during the enabling process.	Check whether there is joint movement during power on, and whether there is external interference during the enabling process.
0x271D	J14 axis arrival timeout	The servo takes too long to arrive the target position.	Check whether the arrival error threshold is too small or whether the servo gain is not proper.
0x271E	J14 axis encoder feedback error	1. Multi-turn value changes. 2. Encoder position jumps.	Check if the multi turn value has been manually cleared. If not, check for the interference source.
0x271F	J14 axis disable status error	The robot failed to be disabled at the specified time and distance.	1. Check that servo parameter 6084 and brake parameter 0209 are correctly set. 2. Check if the load exceeds the limit.
0x2801	Failed to switch servo drive to speed mode	Failed to switch servo drive to speed mode	Check that the servo drive is normal or contact the manufacturer.

Error code	Description	Cause	Solution
0x2802	Failed to switch servo drive to homing mode	Failed to switch servo drive to homing mode	Check that the servo drive is normal or contact the manufacturer.
0x2803	Failed to switch servo drive to position mode	Failed to switch servo drive to position mode	Check that the servo drive is normal or contact the manufacturer.
0x2804	An inactive independent axis is used in the instruction.	An inactive independent axis is used in the instruction.	Activate the independent axis before using it.
0x2805	When switching to the independent axis mode, the axis is still in motion.	When switching to the independent axis mode, the axis is still in motion.	Switch the axis mode after the axis stops completely.
0x2806	Reference position error for independent axis homing.	Reference position error for independent axis homing.	Check that the reference position for independent axis homing is within the specification.
0x2807	Independent axis mode error	The current version has no independent axis of position mode, but the independent axis mode is switched to the position mode.	Contact the manufacturer.
0x2808	ECP function is not supported when J6 is an active independent axis.	The ECP function is used when J6 is used as an active independent axis.	Switch J6 axis to normal axis before using the ECP function.
0x2809	When J6 is an active independent axis, the tool and flange are not concentric.	When J6 is an active independent axis, the tool and flange are not concentric.	When J6 is an active independent axis, use a tool concentric with the flange.
0x280A	The use of independent axis instructions in dynamic tracking mode is not supported.	The use of independent axis instructions in dynamic tracking mode is not supported.	Do not use independent axis instructions in dynamic tracking mode.
0x280C	Independent axis function cannot be enabled when the soft float function is enabled.	The soft float function and the independent axis function are mutually exclusive.	Disable the soft float function and then enable the independent axis function.

Error code	Description	Cause	Solution
0x280D	Free curve function is not supported when J6 is an active independent axis.	Free curve function is not supported when J6 is an active independent axis.	Switch J6 axis to normal axis before using free curve function.
0x2810	When clearing the multi-turn value of the independent axis joint, it is detected that the controller is in the non-enabled state.	When clearing the multi-turn value of the independent axis joint, it is detected that the controller is in the non-enabled state.	Clear the multi-turn value of independent axis joint when the controller is in the enabled state.
0x2811	The pulse at the robot's motion point is too large in the position mode.	The pulse at the robot's motion point is too large in the position mode.	Check if the joint angles at the starting and ending points are too large. If so, clear the multi-turn value of joint and then get new points.
0x2812	When taking the current point as zero point of the external axis, the current pulse is too large.	When taking the current point as zero point of the external axis, the current pulse is too large.	Set H0536 to 0 through the backend of the external axis servo, then set H0530 to 6, and then get the zero point of the external axis.
0x2813	When taking the current point as the robot zero point, it was detected that the robot axis pulse was too large.	When taking the current point as the robot zero point, it was detected that the robot axis pulse was too large.	Set H0D20 of the axis with excessive pulse to 2 through the servo backend, restart the controller, and then take the current point as the robot zero point.
0x2814	Robot entered non-movement areas	Robot entered non-movement areas	Move the robot out of the area where movement is prohibited.
0x2816	Failed to switch servo drive to torque mode	Failed to switch servo drive to torque mode	Check that the servo drive is normal or contact the manufacturer.
0x2815	Safety mode triggers Category 1 emergency stop	The safety module detected an abnormal behavior.	Disable safety module monitoring and clear the alarm.
0x281A	The speed exceeds 250mm/s in manual low speed mode.	Speed planning error	Save error information and contact the manufacturer.
0x281B	The conveyor speed is too high in manual low speed mode.	The conveyor speed exceeds 240mm/s in manual mode.	Reduce the conveyor speed.

Error code	Description	Cause	Solution
0x281C	Manual low speed mode: CnvFastCatch instruction planning error	The conveyor speed is too high or the motion of the CnvFastCatch command is too short in manual low speed mode, causing an exception in the planning of the compensation segment in the instruction.	Reduce the conveyor speed, or add the motion distance of the CnvFastCatch instruction.
0x281D	Conveyor tracking action speed exceeded 250 mm/s in manual low speed mode	The conveyor speed is too high in manual mode.	Reduce the conveyor speed.
0x2820	Robot entered interference zone 0	Robot entered interference zone 0	Disable the interference zone 0, and move the robot out of interference zone 0.
0x2821	Robot entered interference zone 1	Robot entered interference zone 1	Disable the interference zone 1, and move the robot out of interference zone 1.
0x2822	Robot entered interference zone 2	Robot entered interference zone 2	Disable the interference zone 2, and move the robot out of interference zone 2.
0x2823	Robot entered interference zone 3	Robot entered interference zone 3	Disable the interference zone 3, and move the robot out of interference zone 3.
0x2824	Robot entered interference zone 4	Robot entered interference zone 4	Disable the interference zone 4, and move the robot out of interference zone 4.
0x2825	Robot entered interference zone 5	Robot entered interference zone 5	Disable the interference zone 5, and move the robot out of interference zone 5.
0x2826	Robot entered interference zone 6	Robot entered interference zone 6	Disable the interference zone 6, and move the robot out of interference zone 6.
0x2827	Robot entered interference zone 7	Robot entered interference zone 7	Disable the interference zone 7, and move the robot out of interference zone 7.

Error code	Description	Cause	Solution
0x2828	Robot entered interference zone 8	Robot entered interference zone 8	Disable the interference zone 8, and move the robot out of interference zone 8.
0x2829	Robot entered interference zone 9	Robot entered interference zone 9	Disable the interference zone 9, and move the robot out of interference zone 9.
0x282A	Robot entered interference zone 10	Robot entered interference zone 10	Disable the interference zone 10, and move the robot out of interference zone 10.
0x282B	Robot entered interference zone 11	Robot entered interference zone 11	Disable the interference zone 11, and move the robot out of interference zone 11.
0x282C	Robot entered interference zone 12	Robot entered interference zone 12	Disable the interference zone 12, and move the robot out of interference zone 12.
0x282D	Robot entered interference zone 13	Robot entered interference zone 13	Disable the interference zone 13, and move the robot out of interference zone 13.
0x282E	Robot entered interference zone 14	Robot entered interference zone 14	Disable the interference zone 14, and move the robot out of interference zone 14.
0x282F	Robot entered interference zone 15	Robot entered interference zone 15	Disable the interference zone 14, and move the robot out of interference zone 14.
0x2901	Invalid curve starting point	The current point is too far from the starting point in the curve instruction.	Move the robotic arm to the starting point first.
0x2902	Curve data transmission error	Internal error	Contact the manufacturer.
0x2903	Insufficient curve data space	DSP curve data storage failed for internal reason.	Save error information and contact the manufacturer.
0x2904	Insufficient queue space for speed constraints	Insufficient queue space in DSP for speed constraints.	Contact the manufacturer.
0x2905	Invalid curve starting point due to too small angle	Current point of the robot cannot be the starting point of the curve instruction.	Move the robotic arm to the starting point first.

Error code	Description	Cause	Solution
0x2906	Invalid curve starting point due to conflicting arm type	Current point of the robot cannot be the starting point of the curve instruction.	Move the robotic arm to the starting point first.
0x2907	External axis movement not allowed during free curve execution	External axis movement not allowed during free curve execution	The external axis movement is not allowed during free curve execution.
0x2908	Curve trajectory deviation exceeds tolerance	Local curve distortion causes deviation.	Adjust local path point constraints to make pose changes as uniform as possible, avoiding concentrated pose changes.
0x2A01	Soft float at singular position	The position compensated by Cartesian soft float is in the singular position.	Avoid enabling Cartesian soft float near singular position.
0x2A02	Soft float Cartesian compensation position exceeded limit	The driving distance of the robot is too large.	Increase the value of Cartesian compensation position variable or reduce the distance of soft float Cartesian motion of robot.
0x2A03	Soft float Cartesian compensation speed exceeded limit	The robot motion speed is too large.	Reduce the speed at which the robot is pushed in the Cartesian direction.
0x2A04	Soft float Cartesian compensation acceleration exceeded limit	The robot motion acceleration is too large.	Reduce the acceleration at which the robot is pushed in the Cartesian direction.
0x2A05	Soft float joint compensation angle exceeded limit	The motion angle of robot joint is too large.	Reduce the motion angle of robot joint.
0x2A06	Soft float joint compensation angular speed exceeded limit	The angular speed at which the robot is pushed is too large.	Reduce the angular speed of the robot.
0x2A07	Soft float joint compensation angular acceleration exceeded limit	The angular acceleration at which the robot is pushed is too large.	Reduce the angular acceleration of the robot.
0x2A08	Soft float joint pulse compensation value exceeded limit	The motion angle of the robot joint is too large, resulting in excessive compensation pulse.	Reduce the motion angle of robot joint.

Error code	Description	Cause	Solution
0x2A09	Soft float joint pulse compensation speed exceeded limit	The angular speed of the robot is too large, resulting in excessive compensation pulse speed.	Reduce the angular speed of the robot.
0x2A0A	Soft float joint pulse compensation acceleration exceeded limit	The angular acceleration of the robot is too large, resulting in excessive compensation pulse angular speed.	Reduce the angular acceleration of the robot.
0x2A0B	Soft float state machine error	Soft float status was incorrectly set.	Reset the soft float function.
0x2A0C	Soft float stop joint compensation angle exceeded limit	The joint angle of the robot is too large when the robot stops.	Restart the soft float function.
0x2A0D	Soft float stop joint compensation angular speed exceeded limit	The joint angular speed of the robot is too large when the robot stops.	Restart the soft float function.
0x2A0E	Soft float stop joint compensation angular acceleration exceeded limit	The joint angular acceleration of the robot is too large when the robot stops.	Restart the soft float function.
0x2A11	Error transmitting soft float enable channel parameters	Error transmitting soft float enable channel parameters	Retransmit soft float enable channel parameters.
0x2A12	Error transmitting soft float disable channel parameters	Error transmitting soft float disable channel parameters	Retransmit soft float disable channel parameter.
0x2A13	Error transmitting soft float track channel parameters	Error transmitting soft float track channel parameters	Retransmit soft float track channel parameters.
0x2A14	Error transmitting channel 1 parameters on the soft float interface	Error transmitting channel 1 parameters on the soft float interface	Retransmit channel 1 parameters on the soft float interface.
0x2A15	Error transmitting channel 2 parameters on the soft float interface	Error transmitting channel 2 parameters on the soft float interface	Retransmit channel 2 parameters on the soft float interface.

Error code	Description	Cause	Solution
0x2A16	Error transmitting soft float Cartesian configuration channel parameters	Error transmitting soft float Cartesian configuration channel parameters	Retransmit soft float Cartesian configuration channel parameters.
0x2A17	Error transmitting soft float joint configuration channel parameters	Error transmitting soft float joint configuration channel parameters	Retransmit soft float joint configuration channel parameters.
0x2A18	Error transmitting soft float blocked channel parameters	Error transmitting soft float blocked channel parameters	Retransmit soft float blocked channel parameters.
0x2A19	Error transmitting soft float blocked channel parameters of whether soft float is complete.	Error transmitting soft float blocked channel parameters of whether soft float is complete.	Switch on and off the soft float-related instructions again.
0x2A1A	Soft float stop joint pulse compensation value exceeded limit	The joint pulse is too large when the robot stops.	Restart the soft float function.
0x2A1B	Soft float stop joint pulse compensation speed exceeded limit	The joint pulse speed is too large when the robot stops.	Restart the soft float function.
0x2A1C	Soft float stop joint pulse compensation acceleration exceeded limit	The joint pulse acceleration is too large when the robot stops.	Restart the soft float function.
0x2A1D	The soft float function cannot be enabled when the tracking process is enabled.	The soft float function and tracking process function are mutually exclusive.	Disable the tracking process and then enable the soft float function.
0x2A1E	The soft float function and the independent axis function are mutually exclusive.	The independent axis function is enabled.	Disable the independent axis function and then enable the soft float function.
0x2A1F	The Soft float function cannot be enabled when the optimal trajectory function is enabled.	The soft float function and the optimal trajectory function are mutually exclusive.	Disable the optimal trajectory function and then enable the optimal trajectory function.

Error code	Description	Cause	Solution
0x2A21	The optimal trajectory function cannot be enabled when the soft float function is enabled.	The soft float function and the optimal trajectory function are mutually exclusive.	Disable the soft float function and then enable the optimal trajectory function.
0x2A22	Soft float Jacobian inverse solution error	Soft float Jacobian inverse solution error	Restart the program.
0x2A23	Reference tool number changed after the soft float function is enabled.	After the soft float function is enabled, the tool number cannot be changed.	Do not change the tool number after the soft float function is enabled.
0x2A24	Reference wobj number changed after the soft float function is enabled.	After the soft float function is enabled, the wobj number cannot be changed.	Do not change the wobj number after the soft float function is enabled.
0x2A25	The robot does not move before the soft float function is enabled.	The robot does not move before the soft float function is enabled.	Move the robot before enabling the soft float function.
0x2A26	When the soft float function is enabled, the actual position of the robot exceeds the limit value of the joint.	When the soft float function is enabled, the actual position of the robot exceeds the limit value of the joint.	Jog the robot out of the joint limit position in the teaching mode.
0x2A27	Soft float function not supported by ceiling-mounted robots	Soft float function not supported by ceiling-mounted robots	Enable the soft float function after turning off the ceiling-mount settings.
0x2A28	The use of guide rails and external axes of the positioner is not allowed in manual mode.	The use of guide rails and external axes of the positioner is not allowed in manual mode.	The use of guide rails and external axes of the positioner is not allowed in manual mode.
0x2A29	Soft float heartbeat function false alarm, causing soft float to close	A false alarm for soft float heartbeat function is triggered, causing soft float to close.	Contact the manufacturer.
0x2B01	Emergency stop signal transmission error	The emergency stop signal check result is incorrect.	If this occurs frequently, contact the manufacturer.

Error code	Description	Cause	Solution
0x2B02	Robot stop planning calculation error	The robot stop planning calculation is incorrect.	Contact the manufacturer.
0x2B03	Stop signal transmission error	The stop signal check result is incorrect.	If this occurs frequently, contact the manufacturer.
0x2B04	Collision signal transmission error	The collision signal check result is incorrect.	If this occurs frequently, contact the manufacturer.
0x2B06	Unintended motion after arriving at stop position of original trajectory	Unintended motion after arriving at stop position of original trajectory	If this occurs frequently, contact the manufacturer.
0x2B07	Planning calculation timeout triggering joint stop	Planning calculation timeout triggering joint stop	If this occurs frequently, contact the manufacturer.
0x2B08	Incorrect motion instruction count received in cache	The number of motion instructions received in cache is incorrect.	Contact the manufacturer.
0x2BB0	Mechanical lock switching not allowed due to active independent axis	The mechanical lock switching is not allowed because an active independent axis is detected.	Switch the active independent axis to normal axis before switching the mechanical lock.
0x2BB1	Axis lock mode error: Axle is enabled.	Axis lock mode error: Axle is enabled.	Ensure that the axis is not enabled and try again.
0x2BB2	Failed to lock axis	Internal error	Contact the manufacturer.
0x2C01	One-drive-many vision function parameter error	One-drive-many vision function parameter error	Contact the manufacturer.
0x2C02	One-drive-many vision primary cache full	One-drive-many vision primary cache full	Contact the manufacturer.
0x2C03	Slave encoder value abnormal in one-drive-many vision	Slave encoder value abnormal in one-drive-many vision	Check whether the EtherCAT connection is correct, ensure to clear the master EtherCAT alarm first, then clear the slave alarm.
0x2C11	6-axis zero point calibration: collision detection threshold below 100	6-axis zero point calibration: collision detection threshold below 100	Adjust the collision detection threshold to 100 or above.

Error code	Description	Cause	Solution
0x2C12	6-axis zero point calibration: torque sensor value limit exceeded	6-axis zero point calibration: torque sensor value limit exceeded	Check the torque sensor condition to see whether it is under normal force.
0x2D0A	Interpolation position limit exceeded	Interpolation position limit exceeded	Contact the manufacturer.
0x2D0B	Interpolation speed reversed	Interpolation speed reversed	Contact the manufacturer.
0x2D0C	Interpolation speed limit exceeded	Interpolation speed limit exceeded	Contact the manufacturer.
0x3000	Multiple fieldbuses active simultaneously	Multiple fieldbuses active simultaneously	Keep only one fieldbus active.
0x3001	EtherNet/IP connection actively disconnected by one party	The EtherNet/IP client or server is actively disconnected.	Re-initiate the EtherNet/IP connection.
0x3002	EtherNet/IP disconnected due to network timeout	Network communication on EtherNet/IP connection does not work.	Check whether the network cable is properly plugged in or has poor contact.
0x3003	EtherCAT (fieldbus) disconnected from master	Loose EtherCAT cable connection, EtherCAT master failure, or robot system failure.	Check whether the network cable is plugged in properly or has poor contact, check the EtherCAT master, or contact the manufacturer.
0x3004	Modbus TCP disconnected due to network timeout	Network communication on Modbus TCP connection does not work.	Check whether the network cable is properly plugged in or has poor contact.

Error code	Description	Cause	Solution
0x3005	PROFINET expansion card missing	The activated PROFINET expansion card has lost communication with the controller.	Check in the following order: 1. Check that the PROFINET expansion card is securely connected to the card slot. 2. Check that the NS indicator light on the front panel of the PROFINET expansion card is solid green or flashing green. 3. Check that the PROFINET expansion card can be activated after the controller is powered on again. If the problem persists, replace the PROFINET expansion card.
0x3032	EtherCAT (fieldbus) communication synchronization failed	The synchronization signal is not generated due to hardware errors.	Contact the manufacturer.
0x3033	EtherCAT (fieldbus) communication IRQ loss overlimit	1. The data frame has been lost or discarded at an upstream station. 2. The performance of the host station is poor, the jitter of the IRQ exceeds the set value (H0E-22) * communication cycle.	1. Check the CPU usage of the master. 2. Reduce the communication time. 3. Check whether link loss occurs on the upstream slave.
0x3034	EtherCAT (fieldbus) EEPROM loading error	During program start-up, the first 8 bytes of data in the EEPROM were wrong, causing EtherCAT slave to fail to start.	Contact the manufacturer to re-burn the XML file.
0x3035	EtherCAT (fieldbus) initialization error	EtherCAT (fieldbus) initialization error	The hardware is faulty, contact the manufacturer.
0x3036	EtherCAT (fieldbus) state switching error	Bad state switching due to incorrect operation of the master or human error.	Contact the manufacturer.
0x3037	DC not enabled for EtherCAT (fieldbus) slave	In DC mode, there is no synchronization signal causing abnormal movement due to master fault or improper master operation.	Contact the manufacturer.

Error code	Description	Cause	Solution
0x3038	EtherCAT (fieldbus) PDO overlimit	The number of PDOs configured for EtherCAT (fieldbus) exceeds the limit.	Check whether the number of PDOs configured for the master exceeds the limit.
0x3039	EtherCAT (fieldbus) link lost	The physical connection of the data link is unstable or the process data is lost due to plug-in/plug-out of the network cable.	Check whether the network cable is connected properly and whether the application site suffers from strong vibration.
0x303A	Link interference existed in EtherCAT (fieldbus) connection	The data is lost due to EMC interference, poor quality of the network cable or improper connection.	Check whether proper grounding is performed and rectify EMC measures. Check whether the network cable used is provided by Inovance and whether the network cable is reliable.
0x303B	EtherCAT (fieldbus) data forward error	The upstream station detects that the data frame has been corrupted, leading to a data transfer error.	Check the upstream station to locate the fault cause.
0x303C	EtherCAT (fieldbus) no data received	1. The data frame has been lost or discarded at an upstream station. 2. The performance of the host station is poor, the jitter of the IRQ exceeds the set value (H0E-22) * communication cycle.	1. Check the CPU usage of the master. 2. Reduce the communication time. 3. Check whether link loss occurs on the upstream slave.
0x303D	EtherCAT (fieldbus) mailbox configuration error (preop)	1. The master station is configured improperly. 2. The slave XML file is incorrect.	1. Check the master configuration. 2. Check the slave xml file.
0x303E	EtherCAT (fieldbus) SM2 configuration error	1. The master station is configured improperly. 2. The slave XML file is incorrect.	1. Check the master configuration. 2. Check the slave xml file.
0x303F	EtherCAT (fieldbus) SM3 configuration error	1. The master station is configured improperly. 2. The slave XML file is incorrect.	1. Check the master configuration. 2. Check the slave xml file.

Error code	Description	Cause	Solution
0x3040	Illegal MAC address of PROFINET expansion card	The MAC address of the PROFINET expansion card was not burned, or an illegal MAC address was detected.	Burn a unique MAC address for the PROFINET expansion card.
0x3041	No network connection on PROFINET expansion card	PROFINET expansion card is not connected to the master.	Check the cable between the PROFINET expansion card and the master.
0x3042	PROFINET expansion card configuration parameter error	The configuration parameters for the PROFINET expansion card are incorrectly set.	Check the configuration parameter settings for the PROFINET expansion card.
0x3043	PROFINET expansion card disconnected from master	PROFINET expansion card is disconnected from master.	Check the cable between the PROFINET expansion card and the master.
0x3044	PROFINET expansion card chip not detected	No PROFINET expansion card slave chip is detected.	Reinstall the PROFINET expansion card or restart the controller.
0x3045	PROFINET expansion card master shutdown	It is detected that the IOPS state in the PNIO data frame is bad, indicating that the data frame is invalid and the master is stopped.	Check if the PROFINET master is in the STOP state. If yes, switch it to the RUN state.
0x3046	PROFINET expansion card name not exist	No PROFINET expansion card device name is detected.	Assign a device name to the PROFINET expansion card through configuration software.
0x3047	PROFINET expansion card PDO length mismatch	A verification error has occurred in the length of the Inobus data frame.	Check that the set length of the data area in the Inobus standard frame matches the actual length.
0x3048	PROFINET expansion card Inobus CRC error	A CRC error has been detected in the received Inobus data frame.	Check and eliminate Inobus link interference.
0x3049	PROFINET expansion card master-slave data address length mismatch	The length of the PROFINET bus data configured on the host controller of the robot does not match the data length configured on the PLC.	Modify the bus communication data length at the robot or PLC end to ensure consistency between the two ends.
0x3050	EtherCAT (fieldbus) watchdog configuration error	The master station is configured improperly.	Check the master configuration.

Error code	Description	Cause	Solution
0x3051	EtherCAT (fieldbus) sync signal lost	1. The DC start time in register 0x990 and the DC active time in register 0x981 configured in the master are less than the local system time of the slave. 2. The DC startup time in the 0x990 register and the DC active time in the 0x981 register of the master settings are too large, exceeding the local system time of the slave by 17.2s. The sync signal pulse width in the 0x982 register is 0.	Check the master configuration.
0x3052	EtherCAT (fieldbus) PHY initialization configuration failed	1. The ESC is not working properly before reading the PHY ID 2. The PHY chip is damaged. 3. The data is disturbed. 4. The model configured is different from the actual model of the PHY.	Replace the servo drive. Contact Inovance for maintenance.
0x3053	EEPROM sync signal pulse width length abnormal	1. The master committed an incorrect operation while writing EEPROM configuration data. 2. The ESI file is abnormal.	Reset the ESI file using a parameter or re-program the ESI file.
0x3100	Fail to save configuration file	The system is busy or there is a problem with the file system.	1. Try to save again. 2. Contact the manufacturer.
0x3101	Fail to Load the DSP0 Firmware	DSP0 firmware error or startup abnormality	1. Restart the robot. 2. Contact the manufacturer.
0x3102	Fail to Load the EtherCAT slave Xml File	Xml file error or startup exception	1. Restart the robot. 2. Contact the manufacturer.
0x3110	CANopen: CANopen slave offline	The CANopen slave is in offline status.	1. Check the CAN network connection. 2. Check and eliminate CAN network interference.

Error code	Description	Cause	Solution
0x3111	CANopen: Unknown status of CANopen slave	The CANopen master status is abnormal.	1. Restart the robot. 2. Contact the manufacturer.
0x3112	CANopen: Unknown configuration error	The JSON file error or parsing exception occurs.	1. Restart the robot. 2. Contact the manufacturer.
0x3113	CANopen: Configuration file missing	JSON file is missing, or the path is incorrect.	1. Restart the robot. 2. Contact the manufacturer.
0x3114	CANopen: Invalid configuration file	JSON file is invalid or incorrect.	1. Update the configuration file. 2. Contact the manufacturer.
0x3115	CANopen: Empty slave list in the configuration file	CANopen configuration has no slave or configuration failed.	Update the configuration file.
0x3116	CANopen: Empty SDO configuration in the configuration file	JSON file generation failed or error	Update the configuration file.
0x3117	CANopen: Empty PDO mapping configuration in the configuration file	JSON file generation failed or error	Update the configuration file.
0x3118	CANopen: Memory initialization failed	The memory usage or operational load of the robot is too high.	1. Reduce memory usage or operational load. 2. Contact the manufacturer.
0x3119	CANopen: Protocol stack initialization failed	JSON file error or protocol stack operation failed	1. Update the configuration file. 2. Restart the robot.
0x311A	CANopen: Synchronization configuration error	The CANopen master is abnormal.	1. Restart the robot. 2. Contact the manufacturer.
0x311B	CANopen: Heartbeat producer initialization error	The CANopen master is abnormal.	1. Restart the robot. 2. Contact the manufacturer.
0x311C	CANopen: SDO initialization error	The CANopen master is abnormal.	1. Restart the robot. 2. Contact the manufacturer.

Error code	Description	Cause	Solution
0x311D	CANopen: Heartbeat consumer initialization error	The CANopen master is abnormal.	1. Restart the robot. 2. Contact the manufacturer.
0x311E	CANopen: Online SDO error	Online SDO client generation or transmission failed	1. Check the CAN network connection. 2. Restart the robot.
0x311F	CANopen: SDO configuration error in initializing slave	Failed to write CANopen slave configuration information.	1. Check the CAN network connection. 2. Update the configuration file. 3. Restart the robot.
0x3120	CANopen: SDO busy	Online SDO client is currently unavailable.	Retry later.
0x3121	CANopen: SDO internal error	Unknown	Unknown
0x3122	CANopen: Emergency message	An emergency message is received from CANopen slave.	1. Check for anomalies in the slave operational state. 2. Check and eliminate CAN network interference.
0x3123	CANopen: Error setting INITIALIZATION status	Unknown	Unknown
0x3124	CANopen: Error setting PRE_OPERATIONAL status	Unknown	Unknown
0x3125	CANopen: Error setting OPERATIONAL status	Unknown	Unknown
0x3126	CANopen: Error setting STOPPED status	Unknown	Unknown
0x3127	CANopen: NMT RESET execution error	NMT reset frame transmission failed.	1. Check the CAN network connection. 2. Check and eliminate CAN network interference. 3. Restart the robot.
0x3128	CANopen: NMT START execution error	NMT start frame transmission failed.	Unknown
0x3129	CANopen: NMT STOP execution error	Unknown	Unknown

Error code	Description	Cause	Solution
0x313A	CANopen: Error switching to PRE_OPERATIONAL status	A CANopen master status switchover error occurs.	1. Restart the robot. 2. Contact the manufacturer.
0x313B	CANopen: Error setting OPERATIONAL status	Failed to reconnect slave.	1. Check the CAN network connection. 2. Check and eliminate CAN network interference. 3. Restart the robot.
0x3200	DeviceNet: Master initialization failed	Master initialization failed.	1. Check whether the CAN interface is properly enabled. 2. Check for duplicated MAC ID.
0x3201	DeviceNet: Master MAC ID duplicated	There is other device in the network with a conflicting MAC ID to the master.	Change the MAC ID of the conflicting device in the network or change the MAC ID of the master.
0x3202	DeviceNet: Master CAN drive initialization failed	CAN drive initialization failed.	Check whether the CAN drive is normal.
0x3203	DeviceNet: Master CAN read failure	CAN drive read failed.	Check whether the CAN interface is properly enabled.
0x3204	DeviceNet: Master CAN write failure	CAN drive write failed.	Check whether the CAN interface is properly enabled.
0x3205	DeviceNet: Slave MAC ID occupied	The MAC ID set for the slave is already occupied.	Change the slave MAC ID to an unused one.
0x3206	DeviceNet: MAC ID unavailable	The set MAC ID exceeds limit.	Change to a valid MAC ID.
0x3207	DeviceNet: Slave input I/O data space allocation error	Protocol stack space allocation error	Troubleshoot the space allocation module.
0x3208	DeviceNet: Slave configured input I/O data too long	The input I/O data configured for slave is too long.	Reconfigure the output I/O data to a reasonable length.
0x3209	DeviceNet: Slave output I/O data allocation space error	Protocol stack space allocation error	Troubleshoot the space allocation module.

Error code	Description	Cause	Solution
0x320A	DeviceNet: Slave configured output I/O data too long	The output I/O data configured for slave is too long.	Reconfigure the input I/O data to a reasonable length.
0x320B	DeviceNet: Slave configured MAC ID conflicts with master MAC ID	The MAC ID in slave settings conflicts with the master MAC ID.	Reconfigure the MAC ID of the slave.
0x320C	DeviceNet: Master transmit data pointer null	The master transmit data pointer is null.	Troubleshoot the data transmit module.
0x320D	DeviceNet: MAC ID unavailable	The set MAC ID exceeds limit.	Reconfigure a valid MAC ID.
0x320E	DeviceNet: Error in setting scan range for slave MAC ID	Error in setting the MAC ID for scanning slaves	Reconfigure the MAC ID range for scanning slaves.
0x320F	DeviceNet: Node error mask not cleared before scanning slaves	Node error mask not cleared before scanning slaves	Check the slave scanning module.
0x3210	DeviceNet: An explicit message is currently being processed	An explicit message is currently in transmission.	Resend pending explicit message later.
0x3211	DeviceNet: Slave inactive. Cannot send explicit message.	The explicit message is sent while slave is not active.	Troubleshoot the slave active status.
0x3212	DeviceNet: Master already in slave scan status	Master executed slave scan action again while scanning slaves.	Abort the current pending slave scanning action and wait for the completion of the ongoing slave scanning.
0x3213	DeviceNet: Slave offline	The slave is offline.	Check the slave status and physical connection reliability.
0x3214	DeviceNet: Slave I/O connection timeout	The slave I/O connection timed out.	Check the slave status and physical connection reliability.
0x3215	DeviceNet: Slave configured output I/O data length mismatch with actual length	The configured length of the slave output data does not match the actual length.	Reconfigure the length of the slave input I/O data.

Error code	Description	Cause	Solution
0x3216	DeviceNet: Slave configured input I/O data length mismatch with actual length	The configured length of the slave input data does not match the actual length.	Reconfigure the length of the slave output I/O data.
0x3217	DeviceNet: Slave configured vendor ID mismatch with actual value	The configured vendor ID of the slave does not match the actual value.	Configure the vendor ID of the slave to the actual value.
0x3218	DeviceNet: Configured slave type mismatch with actual value	The configured type of the slave does not match the actual value.	Configure the type of the slave to the actual value.
0x3219	DeviceNet: Slave configured product code mismatch with actual value	The configured product code of the slave does not match the actual value.	Configure the product code of the slave to the actual value.
0x321A	DeviceNet: Slave CCV configuration value mismatch with actual value	The configured CCV of the slave does not match the actual value.	Configure the CCV of the slave to the actual value.
0x321B	DeviceNet: Slave I/O data volume too large	The slave I/O data volume is too large.	Check the configuration of the slave I/O data length.
0x321C	DeviceNet: Configuration parameter parsing failed	The configuration file format is incorrect.	Import a correctly formatted configuration file.
0x3300	Failed to start EtherCAT module data interaction thread	Failed to start EtherCAT module data interaction thread	Reboot after resolving the error for recovery.
0x3301	Configured number of EtherCAT analog module ADs out of range	The configured number of EtherCAT analog module ADs is out of range.	Check the configured number of ADs.
0x3302	Configured number of EtherCAT analog module DAs out of range	The configured number of EtherCAT analog module DAs is out of range.	Check the configured number of DAs.

Error code	Description	Cause	Solution
0x3303	EtherCAT master station configuration completed. Please reboot.	EtherCAT master station configuration completed. Please reboot.	Restart the controller.
0x3304	EtherCAT master configuration file parsing failed	EtherCAT master configuration file parsing failed	Try to reconfigure.
0x3305	GL20_RTU_ECT slave status error	GL20_RTU_ECT slave status error	Check whether the corresponding slave is faulty.
0x3306	EtherCAT I/O slave status error	EtherCAT I/O slave status error	Check whether the corresponding slave is faulty.
0x3307	EtherCAT AD slave status error	EtherCAT AD slave status error	Check whether the channel input value exceeds limit or the channel is disconnected.
0x3308	EtherCAT DA slave status error	EtherCAT DA slave status error	Check whether the wiring is incorrect or the external device impedance is too high.
0x3309	Excessive input-type PDOs configured in EtherCAT master	Excessive input-type PDOs configured in EtherCAT master	Check the number of PDOs on the slave, and configure fewer.
0x330A	Excessive output-type PDOs configured in EtherCAT master	Excessive output-type PDOs configured in EtherCAT master	Check the number of PDOs on the slave, and configure fewer.
0x3311	EtherNet/IP transfer type or trigger method not supported	EtherNet/IP transfer type or trigger method not supported	Check transfer type and trigger method.
0x3312	EtherNet/IP connection ownership conflict - duplicate connection to target device	EtherNet/IP connection ownership conflict - duplicate connection to target device	Ensure the target device is not connected by other masters.
0x3313	EtherNet/IP connection size invalid	EtherNet/IP connection size invalid	Ensure the connection size is valid.

Error code	Description	Cause	Solution
0x3314	EtherNet/IP device not configured. Configuration data not set for the device requiring configuration.	EtherNet/IP device not configured. Configuration data not set for the device requiring configuration.	Configure the EtherNet/IP correctly.
0x3315	Unsupported RPI of EtherNet/IP or issues with timeout multiplier coefficient and inhibit time	Unsupported RPI of EtherNet/IP or issues with timeout multiplier coefficient and inhibit time	Check whether the RPI and timeout multiplier coefficient and inhibit time are valid.
0x3316	EtherNet/IP RPI not accepted	EtherNet/IP RPI not accepted	Check whether the RPI is valid.
0x3317	EtherNet/IP local connection is full, the connection manager cannot support more connections	EtherNet/IP local connection is full, the connection manager cannot support more connections	Delete unnecessary connections.
0x3318	EtherNet/IP Vendor ID or Product Code mismatch with device	EtherNet/IP Vendor ID or Product Code mismatch with device	Check whether the Vendor ID or Product Code of EtherNet/IP matches the device.
0x3319	EtherNet/IP product type mismatch with device	EtherNet/IP product type mismatch with device	Check whether the EtherNet/IP product type matches the device.
0x331A	EtherNet/IP primary or secondary version number mismatch with device	EtherNet/IP primary or secondary version number mismatch with device	Check whether the EtherNet/IP primary or secondary version number matches the device.
0x331B	EtherNet/IP connection request failed due to no enabled connection	EtherNet/IP connection request failed due to no enabled connection	Check the connection configuration.
0x331C	EtherNet/IP target application cannot support more connections	EtherNet/IP target application cannot support more connections	Delete unnecessary connections.
0x331D	EtherNet/IP RPI less than inhibit time	The RPI of EtherNet/IP is less than the inhibit time.	Ensure the RPI is greater than or equal to the inhibit time.

Error code	Description	Cause	Solution
0x331E	EtherNet/IP connection timeout	EtherNet/IP connection timeout	Re-download the configuration.
0x331F	Invalid segment path or segment value in EtherNet/IP path	Invalid segment path or segment value in EtherNet/IP path	Check the connection path.
0x3320	EtherNet/IP supplier-specific error	EtherNet/IP supplier-specific error	Contact the supplier.
0x3321	Unsupported transfer type in EtherNet/IP transfer type/trigger method parameters	Unsupported transfer type in EtherNet/IP transfer type/ trigger method parameters	Check the transfer type.
0x3322	Unsupported trigger method in EtherNet/IP transfer type/trigger method parameters	Unsupported trigger method in EtherNet/IP transfer type/ trigger method parameters	Check the trigger method.
0x3323	Unsupported request direction in EtherNet/ IP transfer type/trigger method parameters	Unsupported request direction in EtherNet/IP transfer type/ trigger method parameters	Check the request direction in transfer type/trigger method parameters.
0x3324	EtherNet/IP O->T fixed/variable flag not supported	EtherNet/IP O->T fixed/variable flag not supported	Check the O->T fixed/variable flag.
0x3325	EtherNet/IP T->O fixed/variable flag not supported	EtherNet/IP T->O fixed/variable flag not supported	Check the T->O fixed/variable flag.
0x3326	EtherNet/IP O->T priority not supported	EtherNet/IP O->T priority not supported	Re-check the O->T priority.
0x3327	EtherNet/IP T->O priority not supported	EtherNet/IP T->O priority not supported	Re-check the T->O priority.
0x3328	EtherNet/IP O->T connection type not supported	EtherNet/IP O->T connection type not supported	Reconfigure the O->T connection type.
0x3329	EtherNet/IP T->O connection type not supported	EtherNet/IP T->O connection type not supported	Reconfigure the T->O connection type.

Error code	Description	Cause	Solution
0x332A	Invalid data area length	The data segment provided in the EtherNet/IP connection path parameters does not contain the requested configuration of required 16-bit word count for the application path.	Check the connection configuration.
0x332B	Unsupported size requested in EtherNet /IP O->T network connection parameters	Unsupported size requested in EtherNet/IP O->T network connection parameters	Reconfigure the size requested in the O->T network connection parameters.
0x332C	Unsupported size requested in EtherNet /IP T->O network connection parameters	Unsupported size requested in EtherNet/IP T->O network connection parameters	Reconfigure the size requested in the T->O network connection parameters.
0x332D	EtherNet/IP specified configuration application path mismatch with valid path	The EtherNet/IP specified configuration application path does not match the valid configuration application path.	Reconfigure a valid configuration application path.
0x332E	EtherNet/IP specified consumption application path mismatch with valid path	The EtherNet/IP specified consumption application path does not match the valid consumer application path.	Reconfigure a valid consumption application path.
0x332F	EtherNet/IP specified production application path mismatch with valid path	The EtherNet/IP specified production application path does not match the valid production application path.	Reconfigure a valid production application path.
0x3330	EtherNet/IP specified configuration and/or consumption and/or production application path combinations inconsistent	The EtherNet/IP specified configuration and/or consumption and/or production application path combinations are inconsistent.	Reconfigure the configuration and/or consumption and/or production application path combinations.

Error code	Description	Cause	Solution
0x3331	Information in EtherNet/IP data segment inconsistent with consumer data format	The information in the EtherNet/IP data segment is inconsistent with the consumer data format.	Reconfigure and ensure the information in the EtherNet/IP data segment is consistent with the consumer data format.
0x3332	Information in EtherNet/IP data segment inconsistent with producer data format	The information in the EtherNet/IP data segment is inconsistent with the producer data format.	Reconfigure and ensure the information in the EtherNet/IP data segment is consistent with the producer data format.
0x3333	Null Forward Open request not supported in EtherNet/IP target	Null Forward Open request is not supported in the EtherNet/IP target.	Reconfigure the Forward Open request in the EtherNet/IP target.
0x3334	EtherNet/IP connection timeout multiplier value reserved or resulting timeout value exceeded limit supported by device	The EtherNet/IP connection timeout multiplier value is reserved or the resulting timeout value exceeded the upper limit supported by the device.	Reconfigure the connection timeout multiplier value.
0x3335	EtherNet/IP connection size mismatch with existing multicast connection	EtherNet/IP connection size mismatch with existing multicast connection	Check the connection configuration.
0x3336	EtherNet/IP fixed/variable connection type mismatch with existing multicast connection	EtherNet/IP fixed/variable connection type mismatch with existing multicast connection	Reconfigure the EtherNet/IP fixed/variable connection type.
0x3337	EtherNet/IP fixed/variable connection priority mismatch with existing multicast connection	EtherNet/IP fixed/variable connection priority mismatch with existing multicast connection	Reconfigure the EtherNet/IP fixed/variable connection priority.
0x3338	EtherNet/IP fixed/variable transfer type mismatch with existing multicast connection	EtherNet/IP fixed/variable transfer type mismatch with existing multicast connection	Reconfigure the EtherNet/IP fixed/variable transfer type.

Error code	Description	Cause	Solution
0x3339	EtherNet/IP fixed/ variable production trigger mismatch with existing multicast connection	EtherNet/IP fixed/variable production trigger mismatch with existing multicast connection	Reconfigure the EtherNet/IP fixed /variable production trigger type.
0x333A	EtherNet/IP fixed/ variable production inhibit time mismatch with existing multicast connection	EtherNet/IP fixed/variable production inhibit time mismatch with existing multicast connection	Reconfigure the EtherNet/IP fixed /variable production inhibit time.
0x333B	EtherNet/IP duplicate connection attempt to active session	The EtherNet/IP device is attempting to reconnect to a target that is already connected.	Waiting for the connection result.
0x333C	EtherNet/IP master failed to send data	EtherNet/IP communication error	Check whether the communication between master and slave is normal.
0x333D	EtherNet/IP master failed to receive data	EtherNet/IP communication error	Check whether the communication between master and slave is normal.
0x333E	EtherNet/IP protocol stack startup failed	EtherNet/IP protocol stack status error	1. Restart the protocol stack. 2. Check the connection status of the dynamic network port. If the network port is loose or not connected, connect the port properly and restart for it to take effect. 3. To clear the alarm, save and disable the EtherNet/IP slave switch.
0x333F	Connection start failed	Connection configuration error	Check the connection configuration.
0x3340	Offline of target slave connected to master	The target slave connected to the master has gone offline.	Check whether the slave switch is On and the link is in normal operation.
0x3341	Network port declaration failed	Network port declaration failed	Reissue configuration.

Error code	Description	Cause	Solution
0x3342	Configured slave IP conflicted with local master IP	Slave IP cannot be the same as the master network port IP	Fill in the correct slave IP.
0x3350	File read and write size exceeds 40MB	File read and write size exceeds 40MB	Stop writing oversized file.
0x3351	File read and write size exceeds 50MB	File read and write size exceeds 50MB	Stop writing oversized file.
0x3352	EMMC remaining memory less than 25%	EMMC remaining memory less than 25%	Check EMMC memory, delete some program files and some user files appropriately.
0x3353	EMMC remaining memory less than 5%	EMMC remaining memory less than 5%	Check EMMC memory, delete some program files and some user files appropriately.
0x3354	EMMC life exceeded 90%	The usage of EMMC has exceeded 90% of its estimated lifespan.	Reduce file read/write operations, including SavePoints, FWrite, FWriteALine, etc. Replace the core board.
0x3355	EMMC life exhausted, please contact the manufacturer	The EMMC life has been exhausted, please contact the manufacturer.	Check if the EMMC life has been exhausted. If yes, replace the core board.
0x4A00	The event list in the event mechanism is full.	The host controller does not read the events in the event list in time.	Read the event list in time.
0x4A01	In the event mechanism, when several events sent by the controller to the host controller are compared with the same events sent back by the host controller, the event information is inconsistent.	The communication between the controller and host controller is incorrect.	Check whether the communication between the controller and host controller is normal.
0x4A02	The range of event IDs in the event mechanism exceeds the limit.	The initialization array in the event mechanism was not properly filled, an internal error.	Save the error message on the system diagnosis page and contact the manufacturer.

Error code	Description	Cause	Solution
0x4A10	One-drive-many vision: Master bus configuration updated, restart required	Update the parameters of one-drive-many vision.	Restart.
0x4A11	One-drive-many vision: EtherCAT online slave mismatch with configuration	The EtherCAT communication is disconnected.	Check whether the EtherCAT channel is connected properly.
0x4A12	One-drive-many vision: Slave TCP connection disconnected	TCP communication connection is abnormal.	1. Check whether the EtherNet port network cable is connected normally. 2. Check for any anomalies in the local area network.
0x4A13	One-drive-many vision: TCP visual data cache queue full	Vision data transfer load is high.	1. Reduce the conveyor speed. 2. Decrease the number of workobjects on the conveyor.
0x4A14	One-drive-many vision: Monitoring cache queue full	Vision data transfer load is high.	1. Reduce the conveyor speed. 2. Decrease the number of workobjects on the conveyor.
0x4A15	One-drive-many vision: Slave device index value error	1. The index value is occupied. 2. The index value is invalid.	Use the correct slave device index value.
0x4A16	One-drive-many vision: Master server exception	Master device vision data server exception	Check whether the master device server is properly enabled
0x4A17	One-drive-many vision: Data processing error	Workobject pixel coordinate and conveyor coordinate calibration matrix are incorrect.	Check whether the calibration matrix is correct.
0x4A18	One-drive-many vision: Shared memory vision data cache full	Vision data transfer load is high.	1. Check whether the motion module is functioning normally. 2. Reduce the number of workobjects on the conveyor belt (if the motion module is functioning normally).

Error code	Description	Cause	Solution
0x4A19	One-drive-many vision: Slave connections not fully established	Slave connections are not fully established, while the master device is already executing one-drive-many related API.	Ensure that all slaves are successfully connected to the master.
0x4A1A	One-drive-many vision: TCP vision data verification error	TCP data transmission error	Check whether the local area network is interfered.
0x4A1B	One-drive-many vision: Slave EtherCAT-(IN) connection status abnormal	EtherCAT-IN connection is disconnected.	Check whether the EtherCAT connection is abnormal.
0x4A1C	One-drive-many vision: Slave EtherCAT-(OUT) connection status abnormal	EtherCAT-OUT connection is disconnected.	Check whether the EtherCAT connection is abnormal.
0x4A1D	One-drive-many vision: Slave EtherCAT-(IN)-(OUT) connection status abnormal	EtherCAT-IN-OUT connection is disconnected.	Check whether the EtherCAT connection is abnormal.
0x4A1E	One-drive-many vision: Slave ESC status abnormal (master EtherCAT connection normal)	The EtherCAT slave status is abnormal.	Check whether the EtherCAT connection is abnormal.
0x4A1F	One-drive-many vision: Configured/ Online slave quantity mismatch	The number of configured slaves does not match the number of online slaves.	Check the configuration of one-drive-many vision, fix the error and restart the controller.
0x5001	J1 axis torque overlimit warning	The actual maximum current exceeds the limit for warning.	Check whether the speed or acceleration is too large.
0x5002	J1 axis average load rate overlimit warning	The average load rate exceeds the limit for warning.	Check whether speed and acceleration are too large. Troubleshoot the brake, mechanical parts and drive.
0x5003	J2 axis torque overlimit warning	The actual maximum current exceeds the limit for warning.	Check whether the speed or acceleration is too large.

Error code	Description	Cause	Solution
0x5004	J2 axis average load rate overlimit warning	The average load rate exceeds the limit for warning.	Check whether speed and acceleration are too large. Troubleshoot the brake, mechanical parts and drive.
0x5005	J3 axis torque overlimit warning	The actual maximum current exceeds the limit for warning.	Check whether the speed or acceleration is too large.
0x5006	J3 axis average load rate overlimit warning	The average load rate exceeds the limit for warning.	Check whether speed and acceleration are too large. Troubleshoot the brake, mechanical parts and drive.
0x5007	J4 axis torque overlimit warning	The actual maximum current exceeds the limit for warning.	Check whether the speed or acceleration is too large.
0x5008	J4 axis average load rate overlimit warning	The average load rate exceeds the limit for warning.	Check whether speed and acceleration are too large. Troubleshoot the brake, mechanical parts and drive.
0x5009	J5 axis torque overlimit warning	The actual maximum current exceeds the limit for warning.	Check whether the speed or acceleration is too large.
0x500A	J5 axis average load rate overlimit warning	The average load rate exceeds the limit for warning.	Check whether speed and acceleration are too large. Troubleshoot the brake, mechanical parts and drive.
0x500B	J6 axis torque overlimit warning	The actual maximum current exceeds the limit for warning.	Check whether the speed or acceleration is too large.
0x500C	J6 axis average load rate overlimit warning	The average load rate exceeds the limit for warning.	Check whether speed and acceleration are too large. Troubleshoot the brake, mechanical parts and drive.
0x500D	J7 axis torque overlimit warning	The actual maximum current exceeds the limit for warning.	Check whether the speed or acceleration is too large.
0x500E	J7 axis average load rate overlimit warning	The average load rate exceeds the limit for warning.	Check whether speed and acceleration are too large. Troubleshoot the brake, mechanical parts and drive.
0x500F	J8 axis torque overlimit warning	The actual maximum current exceeds the limit for warning.	Check whether the speed or acceleration is too large.

Error code	Description	Cause	Solution
0x5010	J8 axis average load rate overlimit warning	The average load rate exceeds the limit for warning.	Check whether speed and acceleration are too large. Troubleshoot the brake, mechanical parts and drive.
0x5011	J1 axis servo warning	Warning occurs on the servo of J1 axis.	View the servo parameter H0B-45 and handle the problem according to the user guide.
0x5012	J2 axis servo warning	Warning occurs on the servo of J2 axis.	View the servo parameter H0B-45 and handle the problem according to the user guide.
0x5013	J3 axis servo warning	Warning occurs on the servo of J3 axis.	View the servo parameter H0B-45 and handle the problem according to the user guide.
0x5014	J4 axis servo warning	Warning occurs on the servo of J4 axis.	View the servo parameter H0B-45 and handle the problem according to the user guide.
0x5015	J5 axis servo warning	Warning occurs on the servo of J5 axis.	View the servo parameter H0B-45 and handle the problem according to the user guide.
0x5016	J6 axis servo warning	Warning occurs on the servo of J6 axis.	View the servo parameter H0B-45 and handle the problem according to the user guide.
0x5017	J7 axis servo warning	Warning occurs on the servo of J7 axis.	View the servo parameter H0B-45 and handle the problem according to the user guide.
0x5018	J8 axis servo warning	Warning occurs on the servo of J8 axis.	View the servo parameter H0B-45 and handle the problem according to the user guide.
0x5019	J9 axis servo warning	Warning occurs on the servo of J9 axis.	View the servo parameter H0B-45 and handle the problem according to the user guide.
0x5020	The number of vision data points of tracking process exceeds 50, the excessive points will be ignored.	The number of vision data points is out of range.	Limit the number of vision data points to 50 or fewer.

Error code	Description	Cause	Solution
0x5021	J1 axis motor/reducer overloaded.	Motor/reducer temperature is too high, exceeding warning threshold.	1. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 2. Reduce the duty cycle. 3. Check whether the ambient temperature of the robot is too high and reduce the temperature if necessary.
0x5022	J1 axis encoder overheated	Encoder temperature is too high, exceeding warning threshold.	1. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 2. Reduce the duty cycle. 3. Check whether the ambient temperature of the robot is too high and reduce the temperature if necessary.
0x5023	J1 axis motor/reducer overloaded	The steady-state predicted temperature of the motor/ reducer is high, exceeding the warning threshold.	1. Check that the ambient temperature is set correctly. 2. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 3. Reduce the duty cycle.
0x5041	J2 axis motor/reducer overheated	Motor/reducer temperature is too high, exceeding warning threshold.	1. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 2. Reduce the duty cycle. 3. Check whether the ambient temperature of the robot is too high and reduce the temperature if necessary.

Error code	Description	Cause	Solution
0x5042	J2 axis encoder overheated	Encoder temperature is too high, exceeding warning threshold.	1. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 2. Reduce the duty cycle. 3. Check whether the ambient temperature of the robot is too high and reduce the temperature if necessary.
0x5043	J2 axis motor/reducer overloaded	The steady-state predicted temperature of the motor/ reducer is high, exceeding the warning threshold.	1. Check that the ambient temperature is set correctly. 2. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 3. 3. Reduce the duty cycle.
0x5061	J3 axis motor/reducer overheated	Motor/reducer temperature is too high, exceeding warning threshold.	1. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 2. Reduce the duty cycle. 3. Check whether the ambient temperature of the robot is too high and reduce the temperature if necessary.
0x5062	J3 axis encoder overheated	Encoder temperature is too high, exceeding warning threshold.	1. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 2. Reduce the duty cycle. 3. Check whether the ambient temperature of the robot is too high and reduce the temperature if necessary.

Error code	Description	Cause	Solution
0x5063	J3 axis motor/reducer overloaded	The steady-state predicted temperature of the motor/ reducer is high, exceeding the warning threshold.	1. Check that the ambient temperature is set correctly. 2. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 3. 3. Reduce the duty cycle.
0x5081	J4 axis motor/reducer overheated	Motor/reducer temperature is too high, exceeding warning threshold.	1. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 2. Reduce the duty cycle. 3. Check whether the ambient temperature of the robot is too high and reduce the temperature if necessary.
0x5082	J4 axis encoder overheated	Encoder temperature is too high, exceeding warning threshold.	1. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 2. Reduce the duty cycle. 3. Check whether the ambient temperature of the robot is too high and reduce the temperature if necessary.
0x5083	J4 axis motor/reducer overloaded	The steady-state predicted temperature of the motor/ reducer is high, exceeding the warning threshold.	1. Check that the ambient temperature is set correctly. 2. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 3. Reduce the duty cycle.

Error code	Description	Cause	Solution
0x50A1	J5 axis motor/reducer overheat	Motor/reducer temperature is too high, exceeding warning threshold.	1. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 2. Reduce the duty cycle. 3. Check whether the ambient temperature of the robot is too high and reduce the temperature if necessary.
0x50A2	J5 axis encoder overheated	Encoder temperature is too high, exceeding warning threshold.	1. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 2. Reduce the duty cycle. 3. Check whether the ambient temperature of the robot is too high and reduce the temperature if necessary.
0x50A3	J5 axis motor/reducer overloaded	The steady-state predicted temperature of the motor/ reducer is high, exceeding the warning threshold.	1. Check that the ambient temperature is set correctly. 2. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 3. Reduce the duty cycle.
0x50C1	J6 axis motor/reducer overheated	Motor/reducer temperature is too high, exceeding warning threshold.	1. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 2. Reduce the duty cycle. 3. Check whether the ambient temperature of the robot is too high and reduce the temperature if necessary.

Error code	Description	Cause	Solution
0x50C2	J6 axis encoder overheated	Encoder temperature is too high, exceeding warning threshold.	1. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 2. Reduce the duty cycle. 3. Check whether the ambient temperature of the robot is too high and reduce the temperature if necessary.
0x50C3	J6 axis motor/reducer overloaded	The steady-state predicted temperature of the motor/ reducer is high, exceeding the warning threshold.	1. Check that the ambient temperature is set correctly. 2. Check whether the speed, acceleration, and load values are too large, and reduce the motion speed, acceleration, or load if necessary. 3. Reduce the duty cycle.
0x50D1	Wrong object type for visual recognition	The object type of the sent visual data is not within the range [0,15].	Modify the object type number sent by vision to make it within the range [0,15].
0x50D2	Vision data waiting timeout	No returned data is received for a long period of time after vision triggering signals are sent, and the vision processing cycle is greater than the photographing interval.	1. Check whether the vision trigger signal is delivered successfully. 2. If the vision trigger signal is delivered successfully every time, check whether the camera will take pictures normally every time. 3. The processing time of vision software is too long, or the delay time of data transmission is too long.
0x50D3	One-drive-many vision: Shared address 2 full	The shared address 2 of one-drive-many vision is full.	Check the TCP connection status of the slave.
0x50D4	One-drive-many vision: Missed object data cache full	The shared address 2 of one-drive-many vision is full.	Check the TCP connection status of the slave.

Error code	Description	Cause	Solution
0x50E1	Lookahead error	Lookahead error	Save the error message on the system diagnostics page and contact the manufacturer.
0x50E2	Adaptive speed planning verification error	Adaptive speed planning verification error	Save the error message on the system diagnostics page and contact the manufacturer.
0x50E3	General speed planning verification error	General speed planning verification error	Save the error message on the system diagnostics page and contact the manufacturer.
0x50E4	Trajectory time too long	The time of a single trajectory exceeds 1000s.	It does not affect operation, but may cause unexpected problems. Increase the speed and acceleration to reduce the execution time of a single instruction.
0x50E5	No smooth curve at a fly point	Position and motion parameters constrain the smooth length.	Normal work is not affected, but be aware that the cycle time can be optimized.
0x50E6	Stop time too long	Stop time exceeded 500 ms during pause or stop due to reasons such as excessively low acceleration.	Normal work is not affected, but be aware that the robot cannot be operated again before it is completely stopped.
0x50E7	The load mass setting is too small when the optimal trajectory mode is turned on, the robot will move at low speed. Check whether the load setting is correct.	The load mass setting is too small when the optimal trajectory mode is turned on.	Check that the load parameters are set properly.
0x50EE	EtherCAT communication feedback error	The speed is abnormal when the robot coasts to stop.	Save the error message on the system diagnostics page and contact the manufacturer.
0x50F8	Real time task execution time too long	Real time task execution timeout	Export the diagnostic information and contact the manufacturer.
0x5101	Robot entered interference zone 0	Robot entered interference zone 0	Disable the interference zone 0, and move the robot out of interference zone 0.

Error code	Description	Cause	Solution
0x5102	Robot entered interference zone 1	Robot entered interference zone 1	Disable the interference zone 1, and move the robot out of interference zone 1.
0x5103	Robot entered interference zone 2	Robot entered interference zone 2	Disable the interference zone 2, and move the robot out of interference zone 2.
0x5104	Robot entered interference zone 3	Robot entered interference zone 3	Disable the interference zone 3, and move the robot out of interference zone 3.
0x5105	Robot entered interference zone 4	Robot entered interference zone 4	Disable the interference zone 4, and move the robot out of interference zone 4.
0x5106	Robot entered interference zone 5	Robot entered interference zone 5	Disable the interference zone 5, and move the robot out of interference zone 5.
0x5107	Robot entered interference zone 6	Robot entered interference zone 6	Disable the interference zone 6, and move the robot out of interference zone 6.
0x5108	Robot entered interference zone 7	Robot entered interference zone 7	Disable the interference zone 7, and move the robot out of interference zone 7.
0x5109	Robot entered interference zone 8	Robot entered interference zone 8	Disable the interference zone 8, and move the robot out of interference zone 8.
0x510A	Robot entered interference zone 9	Robot entered interference zone 9	Disable the interference zone 9, and move the robot out of interference zone 9.
0x510B	Robot entered interference zone 10	Robot entered interference zone 10	Disable the interference zone 10, and move the robot out of interference zone 10.
0x510C	Robot entered interference zone 11	Robot entered interference zone 11	Disable the interference zone 11, and move the robot out of interference zone 11.
0x510D	Robot entered interference zone 12	Robot entered interference zone 12	Disable the interference zone 12, and move the robot out of interference zone 12.

Error code	Description	Cause	Solution
0x510E	Robot entered interference zone 13	Robot entered interference zone 13	Disable the interference zone 13, and move the robot out of interference zone 13.
0x510F	Robot entered interference zone 14	Robot entered interference zone 14	Disable the interference zone 14, and move the robot out of interference zone 14.
0x5110	Robot entered interference zone 15	Robot entered interference zone 15	Disable the interference zone 15, and move the robot out of interference zone 15.
0x6000	Robot initialization parameter error	Robot initialization parameters are illegal.	Reconfigure robot initialization parameters according to diagnostic prompts.
0x6001	Incorrect initialization status	Robot initialization failed	Reinitialize the robot according to diagnostic prompts.
0x6002	Mechanical unit initialization parameter error	Mechanical unit initialization parameters are illegal.	Reinitialize the mechanical unit initialization parameters according to diagnostic prompts.
0x6003	External axis initialization parameter error	External axis initialization parameters are illegal.	Reinitialize the external axis initialization parameters according to diagnostic prompts.
0x6004	Incorrect mechanical unit or external axis count for initialization	Incorrect mechanical unit or external axis count for initialization	Reconfigure the number of mechanical units or external axes according to diagnostic prompts.
0x6010	Illegal position points	The passed in point parameters are illegal.	Check if the Euler angle value of the point exceeds 180 degrees.
0x6011	Illegal tool parameters	The passed in tool parameters are illegal.	1. Check that the input tool installation position Euler angle is correct. 2. Check that the load parameters are legal.
0x6012	Illegal wobj parameters	The passed in wobj parameters are illegal.	1. Check that the input workobject installation position Euler angle is correct. 2. Check that the fixing properties match the mechanical unit.
0x6013	Illegal load parameters	The passed in load parameters are illegal.	Check if there is a negative value of mass inertia in the input load parameters.

Error code	Description	Cause	Solution
0x6014	Illegal joint position	The passed in joint position is illegal.	Check if the input joint position exceeds the limit.
0x6016	The active tool and workobject parameters do not match (kinematics).	The active tool and the workobject have the same RobHold parameter.	1. Switch the active tool and workobject matching the RobHold parameter. 2. Modify the RobHold parameter of the active tool and workobject to different values.
0x6017	Illegal pose parameter	The passed in pose parameters are illegal.	Check if the Euler angle value of the point exceeds 180 degrees.
0x6018	Pose deviation calculation type flag error	The flag bit of the pose deviation calculation function is incorrect.	System error, please contact the manufacturer.
0x6030	Wrong world frame type	The world frame type passed in is incorrect when converting the world frame to a reference frame.	System error, please contact the manufacturer.
0x6031	Wrong mechanical unit type	The mechanical unit type passed in is incorrect when converting the world frame to a reference frame.	System error, please contact the manufacturer.
0x6032	Wrong reference frame type	The reference frame type passed in is incorrect when converting the reference frame type to world frame.	System error, please contact the manufacturer.
0x6033	Wrong mechanical unit type	The mechanical unit type passed in is incorrect when converting the reference frame to world frame.	System error, please contact the manufacturer.
0x6034	The mechanical unit associated with the workobject is not calibrated.	The mechanical unit associated with the workobject is not calibrated.	Calibrate the mechanical unit associated with the workobject.
0x6050	Tool or wobj parameter error	The RobHold parameter is the same for both the tool and the workobject currently in use.	Ensure that the RobHold parameter is set to true for only one of tool and workobject.
0x6051	End parameter calculation error	The RobHold parameter is the same for both the tool and the workobject currently in use.	Ensure that the RobHold parameter is set to true for only one of tool and workobject.

Error code	Description	Cause	Solution
0x6052	Error getting external reference frame	The RobHold parameter is the same for both the tool and the workobject currently in use.	Ensure that the RobHold parameter is set to true for only one of tool and workobject.
0x6053	Wobj frame conversion error	The RobHold parameter is the same for both the tool and the workobject currently in use.	Ensure that the RobHold parameter is set to true for only one of tool and workobject.
0x6054	Reference frame conversion error	The RobHold parameter is the same for both the tool and the workobject currently in use.	Ensure that the RobHold parameter is set to true for only one of tool and workobject.
0x6060	Tool calibration error	Calibration calculation errors caused by non-standard tool calibration points or other reasons	1. Check that the calibration points comply with the calibration rules. 2. Check that the robot parameter initialization is correct.
0x6061	Guide rail-robot calibration error	Calibration calculation errors caused by non-standard guide rail calibration points or other reasons	Check that the calibration points comply with the calibration rules.
0x6062	Positioner calibration error	Calibration calculation errors caused by non-standard position calibration points or other reasons.	1. Check that the calibration points comply with the calibration rules. 2. Check that the positioner calibration sequence is correct.
0x6063	User frame calibration error	Calibration calculation errors caused by non-standard user frame calibration points or other reasons.	Check that the calibration points comply with the calibration rules.
0x6064	Improper tool calibration points	Improper tool calibration points	Re-select the calibration points having significant difference in orientation.
0x6065	Calibration points too close	Calibration points too close	Re-select the calibration points.
0x6066	Calibration points collinear	Calibration points collinear	Re-select the calibration points.

Error code	Description	Cause	Solution
0x6067	Excessive discrepancy between dual-arm parameters, resulting in deviations in target joint angles during inverse calculation	Excessive discrepancy between dual-arm parameters, resulting in deviations in target joint angles during inverse calculation	Modify the arm parameters of the point.
0x6070	Same calibration points input	Same calibration points input	Reselect calibration points.
0x6071	In TCP mode, points relative to flange cannot be converted to Wobj points.	In TCP mode, points relative to flange cannot be converted to Wobj points.	Get a point again.
0x6080	Error in inverse solution of target position in orientation calibration.	The target position for orientation calibration is singular or unreachable.	Check whether the target position for orientation calibration is a singular position.
0x6081	Singular pose present in the orientation calibration path	Orientation calibration causes the robot to go through singular pose.	Adjust the original points so that the robot does not go through singular pose during the orientation calibration process.
0x6090	Incorrect arm type parameter	The passed in arm type parameter is incorrect.	Check that the arm type parameter configuration corresponding to the point is correct.
0x6092	Singular or unreachable position	The points are at singular or unreachable positions.	Check whether the input points are at singular or unreachable positions, and modify the points if necessary.
0x6094	SCARA singularity	SCARA elbow singularity	Adjust points to avoid singularity.
0x6096	Shoulder singularity	Shoulder singularity	Adjust points to avoid singularity.
0x6097	Elbow singularity	Elbow singularity	Adjust points to avoid singularity.
0x6098	Wrist singularity	Wrist singularity	Adjust points to avoid singularity.
0x7030	Curve path angle too small	Position or orientation in the free curve path has sharp turning.	Modify the points in curve path to avoid sharp turning positions or orientations.
0x7031	Excessive spacing between trajectory points	Position or orientation spacing exceeds the specification.	Adjust the spacing between trajectory points or add intermediate points.

Error code	Description	Cause	Solution
0x7032	Illegal number of curve points	The number of curve points input in the curve instruction is less than 4 or more than 500.	Adjust the number of curve points.
0x7033	No valid data within the curve group	No valid curve instruction within the curve group instruction.	Modify curve group instruction.
0x7034	Points with too small spacing in curve	Repetitive or small spacing of input points on the curve results in the inability to construct the curve.	Remove duplicate or too close points.
0x7035	Excessive pose variation caused by Euler degeneracy	When the absolute value of Euler angle B approaches 90°, excessive fluctuation in A/C angles configuration leads to excessive pose spacing.	Avoid degenerate positions or manually adjust A/C values to prevent excessive spacing in curve point pose.
0x7038	Joints move too large	Excessive joint motion between adjacent points in the path.	Check that the arm parameters are set correctly or select appropriate SlewMode.
0x7039	Joint value overlimit	A point in the curve path exceeds the joint angle limit in the set arm parameters.	Modify the point position, modify the point arm parameters, select SlewMode, depending on the situation.
0x7040	Conflict state	Internal error	Contact the manufacturer.
0x7041	Conflict symbol	Internal error	Contact the manufacturer.
0x7042	Conflict resume state	Internal error	Contact the manufacturer.
0x7043	Conflict line number	Internal error	Contact the manufacturer.
0x7050	Too many I/Os bound to the same point on the curve	The number of I/Os bound to the same or similar point exceeds the limit.	Reduce or adjust the number of I/Os.
0x7051	Insufficient effective points on the curve	Insufficient effective points on the curve.	Add curve points.
0x7052	Curve too long	The total length of the curve exceeds the specification.	Execute the curve by multiple instructions.
0x7053	I/O not bound to the point of curve	I/O is bound to a point outside the curve.	Modify the I/O number.

Error code	Description	Cause	Solution
0x7060	Positioner coordination not supported by free curve	The wobj frame is not fixed.	Use the free curve instruction in a fixed wobj frame.
0x8001	No network device	The FPGA module is not functioning properly.	Contact the manufacturer.
0x8002	No master	Failed to request EtherCAT master	Restart the controller. If the problem persists, it is a hardware device error, contact the manufacturer.
0x8003	Invalid domain	Failed to apply for EtherCAT master domain resource	Restart the controller. If the problem persists, it is a hardware device error, contact the manufacturer.
0x8004	Slave not found	The slave could not be found when configuring it.	Restart the controller or contact the manufacturer to check the EtherCAT slave device.
0x8005	Invalid process data	Error configuring slave PDO	Restart the controller or contact the manufacturer to check the EtherCAT slave device.
0x8006	Invalid service data	Error configuring slave SDO	Restart the controller or contact the manufacturer to check the EtherCAT slave device.
0x8007	Invalid entry object	Error configuring slave SDO entry	Restart the controller or contact the manufacturer to check the EtherCAT slave device.
0x8008	Domain memory address allocation error	Failed to apply for EtherCAT master domain memory	Restart the controller. If the problem persists, contact the manufacturer.
0x8009	Failed to activate master	Failed to apply for activation of EtherCAT master	Restart the controller. If the problem persists, contact the manufacturer.
0x800A	Service data common error	Length mismatch or slave not found while configuring slave SDO	Restart the controller. If the problem persists, contact the manufacturer to check the EtherCAT slave deice.

Error code	Description	Cause	Solution
0x800B	Registration cycle callback error	Failed to create timed interrupt task	Restart the controller. If the problem persists, contact the manufacturer.
0x800C	Process communication configuration error	Error configuring PDO buffer	Restart the controller. If the problem persists, contact the manufacturer.
0x800D	Module initialization error	EtherCAT Module resource initialization error	Restart the controller. If the problem persists, contact the manufacturer.
0x800E	Error parsing configuration	Error parsing the slave configuration	Check the secondary development configuration and reconfigure if necessary.
0x800F	Channel parameter configuration error	Failed to configure the interval between channel synchronization signals and interrupts	Restart the controller. If the problem persists, contact the manufacturer.
0x8010	Domain registration error	Domain registration failure detected	Restart the controller. If the problem persists, contact the manufacturer.
0x8011	Error creating timer	Failed to create timer due to system reasons	Restart the controller. If the problem persists, contact the manufacturer.
0x8012	Error starting timer	Failed to start timer due to system reasons	Restart the controller. If the problem persists, contact the manufacturer.
0x8013	EtherCAT communication cycle configuration error	EtherCAT communication cycle is not an integral number of 250us.	Check the secondary development configuration and reconfigure if necessary.
0x8014	EtherCAT version selection error	Wrong version of EtherCAT is used.	EtherCAT version error, contact the manufacturer.
0x8015	EtherCAT servo slave quantity error	The number of configured ECAT servo slaves is less than 1.	Check whether the configured number of EtherCAT servo slaves is correct.
0x8016	EtherCAT I/O slave quantity error	The number of configured EtherCAT I/O slaves is less than 0.	Check whether the configured number of EtherCAT I/O slaves is correct.

Error code	Description	Cause	Solution
0x8017	EtherCAT I/O module quantity error	The number of configured EtherCAT I/O modules is less than 1.	Check whether the number of configured EtherCAT I/O modules is correct.
0x8018	EtherCAT I/O type error	The EtherCAT I/O type is unknown.	Check the secondary development configuration and reconfigure if necessary.
0x8019	EtherCAT I/O not supported	The type of configured EtherCAT I/O is currently not supported.	Check the secondary development configuration and reconfigure if necessary.
0x801A	EtherCAT memory request error	Current memory request failed due to excessive system resource usage.	Restart the controller. If the problem persists, contact the manufacturer.
0x801B	EtherCAT alarm shared memory configuration error	EtherCAT alarm shared memory resource request failed.	Restart the controller. If the problem persists, contact the manufacturer.
0x801C	EtherCAT Servo operation mode configuration error	ECAT servo control mode is not CSP-8,CSV-9 or CST-10.	Check the secondary development configuration and reconfigure if necessary.
0x801D	EtherCAT register configuration error	Failed to access slave register	Check that the slave registers to be accessed are correct or allowed to be accessed.
0x801E	Mismatch between the number of configured EtherCAT I/Os and the number of online I/Os	The number of configured EtherCAT I/Os does not match the number of online I/Os.	Check the number of configured EtherCAT I/Os and the number of online I/Os.
0x801F	Mismatch between the number of configured EtherCAT servos and the number of online servos	The number of configured EtherCAT servos does not match the number of online servos.	Check the number of configured EtherCAT servos and the number of online servos.
0x8020	Configured EtherCAT servo supplier codes not supported	The slave device is not supported.	Check that all slave devices in all EtherCAT networks are supported.
0x8021	Mismatch between configured slave device ID and online slave device ID	The configured slave device ID does not match the online slave device ID.	Check whether the configured slave device ID matches the online slave device ID.

Error code	Description	Cause	Solution
0x8022	FPGA received a non-EtherCAT frame	FPGA received a non-EtherCAT frame	Check whether the slave wiring is correct, clear and reboot after recovery.
0x8024	EtherCAT slave function exception (configured sequence does not match the actual sequence or invalid SDO exists)	The configured sequence does not match the actual one or invalid SDO exists.	Check whether the configured sequence of the slaves matches the actual sequence or the SDO is valid.
0x8028	Error writing buffer	Failed to write buffer due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x8029	Error writing start command	Failed to write start command error due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x802A	Error reading status register	Failed to read status register due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x802B	Error reading data link status	Failed to read the data link status due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x802C	Error reading service data channel	Failed to read the service data channel due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x802D	Error reading service data length	Failed to read the service data length due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x802E	Service data length error	Service data length error due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x802F	Service data reception error	Service data reception error due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x8030	Service data channel busy	Service data channel busy error due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.

Error code	Description	Cause	Solution
0x8031	Service data message error	Service data message error due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x8032	Communication error in reading process data	Error reading process data due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x8033	Error reading process data length	Failed to read the process data length due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x8034	Process data length error	Failed to read the process data length due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x8035	Process data reception error	Failed to receive process data due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x8036	Error opening network device	Failed to open network device due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x8037	Underlying communication error	The underlying communication control error is caused by a hardware error.	Restart the controller. If the problem persists, contact the manufacturer.
0x8038	Underlying communication error	Underlying communication read error due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x8039	Underlying communication error	Underlying communication write error due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x803A	Error reading send time stamp	Failed to read send time stamp due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x803B	Error reading receive time stamp	Failed to read receive time stamp due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x803C	Error reading remaining process data	Failed to read the remaining process data due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.

Error code	Description	Cause	Solution
0x803D	Error reading application time stamp	Failed to read application time stamp due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x803E	Error opening fieldbus LED	Failed to open fieldbus LED due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x803F	Fieldbus LED IOCTRL error	Fieldbus LED IOCTRL error due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x8040	Invalid underlying hardware	The underlying hardware module 1 is invalid due to a hardware error.	Restart the controller. If the problem persists, contact the manufacturer.
0x8041	Invalid underlying hardware	The underlying hardware module 2 is invalid due to a hardware error.	Restart the controller. If the problem persists, contact the manufacturer.
0x8042	EtherCAT slave disconnected (slave offline/alias error)	EtherCAT slave is disconnected or the alias is incorrect.	Check the EtherCAT device/Alias.
0x8043	Non-EtherCAT slave device connected	A non-EtherCAT slave device is connected.	Check that the connected device meets the requirements.
0x8044	EtherCAT network port 0 not connected	The EtherCAT network port 0 is not connected.	Check the status of EtherCAT network port 0.
0x8045	EtherCAT network port 1 not connected	The EtherCAT network port 1 is not connected.	Check the status of EtherCAT network port 1.
0x8046	EtherCAT startup time setting error	EtherCAT startup time is incorrectly set due to a hardware error.	Restart the controller. If the problem persists, contact the manufacturer.
0x8047	EtherCAT slave status error	The read ECAT slave status word indicates that the status is abnormal, possibly due to mismatch with servo module.	Check the error codes to troubleshoot and restart the system, or upgrade the servo module to the matched version.
0x8048	Failed to open EOE virtual NIC device	System driver loading is not normal.	Check the system software version and restart the system.
0x8050	Slave IN/OUT reverse connection	The IN and OUT of external axis are connected in reverse.	Check the wiring of external axis.
0x805C	Toggle bit not changed	The slave feeds back that the toggle bit is not changed.	Re-initiate a SDO access, or contact the manufacturer.

Error code	Description	Cause	Solution
0x805D	SDO protocol timeout	SDO protocol timeout	Re-initiate a SDO access, or contact the manufacturer.
0x805E	Client/Server command specifier not valid or unknown	The slave feeds back that Client/Server command specifier is not valid or unknown.	Internal system error, please contact the manufacturer.
0x805F	Object inaccessible	The slave feeds back that the object is not accessible.	This operation is prohibited.
0x8060	Error reading an write-only object	The slave feeds back that an error occurs while trying to read a write-only object.	This operation is prohibited.
0x8061	Error writing an read-only object	The slave feeds back that an error occurs while trying to write a read-only object.	This operation is prohibited.
0x8062	Object does not exist in the object directory.	The slave feeds back that the object does not exist in the object directory.	Check that the object exists in the object dictionary.
0x8063	Object cannot be mapped into PDO.	The slave feeds back that the object cannot be mapped to the PDO.	Check that the object to be mapped is correct.
0x8064	Length of object to be mapped exceeds PDO length	The slave feeds back that the length of the object to be mapped exceeds the PDO length.	Check that the length of the object to be mapped is correct.
0x8065	Basic parameter incompatible	The slave feeds back that the base parameters are incompatible.	Check that the basic parameters of the slave are compatible.
0x8066	Device internal incompatibility	The slave feeds back that the device is internally incompatible.	Restart the controller and slave. If the problem persists, contact the manufacturer.
0x8067	Access failure due to a hardware error	The slave feeds back that the access failed due to hardware causes.	Restart the controller and slave. If the problem persists, contact the manufacturer.
0x8068	Service parameter length mismatch	The slave feeds back that the service parameter length mismatches.	Check that the length of the service parameter of the accessed slave is correct.

Error code	Description	Cause	Solution
0x8069	Service parameter too long	The slave feeds back that the service parameter length is too long.	Check that the length of the service parameter of the accessed slave is correct.
0x806A	Service parameter too short	The slave feeds back that the service parameter length is too short.	Check that the length of the service parameter of the accessed slave is correct.
0x806B	Subindex does not exist.	The slave feeds back that the subindex does not exist.	Check that the object subindex exists.
0x806C	Parameter value out of range	The slave feeds back that the parameter value is out of range.	Check that the parameters are set correctly.
0x806D	Written parameter value too large	The slave feeds back that the written parameter value is too large.	Check that the parameters are set correctly.
0x806E	Written parameter value too small	The slave feeds back that the written parameter value is too small.	Check that the parameters are set correctly.
0x806F	Maximum value smaller than minimum value.	The slave feeds back that the maximum value is less than the minimum value.	Check that the set values are correct.
0x8070	Error transferring or storing data	The slave feeds back that the data failed to be transferred or stored.	Restart the controller and slave. If the problem persists, contact the manufacturer.
0x8071	Error storing data due to local control	The slave feeds back that the data failed to be stored due to local control.	Restart the controller and slave. If the problem persists, contact the manufacturer.
0x8072	Error storing data due to device status	The slave feeds back that the data failed to be stored due to device status.	Restart the controller and slave. If the problem persists, contact the manufacturer.
0x8073	Object dictionary dynamic generation failure or object dictionary not found	The slave feeds back that the object dictionary dynamic generation fails or no object dictionary is present.	Check that the object dictionary exists.
0x8079	IR-Link initialization error	Failed to initialize IR-LINK due to IR-Link configuration or hardware connection error.	Check the IR-Link configuration or hardware connection.
0x807A	IR-Link communication cycle configuration error	The IR-Link communication cycle is not an integral number of 250us.	Check the IR-Link communication cycle configuration.

Error code	Description	Cause	Solution
0x807B	IR-Link version selection configuration error	Wrong version of IR-Link is used.	Contact the manufacturer.
0x807C	IR-Link servo slave quantity configuration error	The number of configured IR-Link servos is less than 0.	Check the number of configured IR-Link servos.
0x807D	IR-Link slave quantity configuration error	The number of configured IR-Link slaves is less than 1.	Check the number of configured IR-Link slaves.
0x807E	IR-Link module quantity configuration error	The number of configured IR-Link modules is less than 1.	Check the number of configured IR-Link modules.
0x807F	IR-Link type configuration error	The type of configured IR-Link I/O is unknown.	Check the type of configured IR-Link I/O.
0x8080	IR-Link configuration not supported	IR-Link configuration is not supported.	Check the IR-Link configuration.
0x8081	IR-Link memory request configuration error	Current memory request failed due to excessive system resource usage.	Restart the controller. If the problem persists, contact the manufacturer.
0x8082	IR-Link alarm shared memory configuration error	IR-Link alarm shared memory resource request failed	Restart the controller. If the problem persists, contact the manufacturer.
0x8083	IR-Link slave operation mode configuration error	IR-Link operation mode is not MODE-8.	Check the IR-Link operation mode.
0x8084	IR-Link register configuration error	Failed to access slave register	Check that the slave registers to be accessed are correct or allowed to be accessed.
0x8085	Mismatch between the number of configured IR-Link I/Os and the number of online I/Os	The number of configured IR-Link I/Os does not match the number of online I/Os.	Check the number of configured IR-Link I/Os and the number of online I/Os.
0x8086	Mismatch between the number of configured IR-Link slaves and the number of online slaves	The number of configured IR-Link slaves does not match the number of online slaves.	Check the number of configured IR-Link slaves and the number of online slaves.

Error code	Description	Cause	Solution
0x8087	IR-Link servo supplier codes not supported	The IR-Link servo supplier codes are not supported.	Check the IR-Link configuration.
0x8090	Error writing buffer	Failed to write buffer due to hardware error.	Restart the controller. If the problem persists, contact the manufacturer.
0x8091	Error reading service data channel	Failed to read the service data channel due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x8092	Error reading service data length	Failed to read the service data length due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x8093	Service data length error	Service data length error due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x8094	Service data reception error	Service data reception error due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x8095	Service data channel busy	Service data channel busy error due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x8096	Service data message error	Service data message error due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x8097	Communication error in reading process data	Error reading process data due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x8098	Error reading process data length	Failed to read the process data length due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x8099	Process data length error	Process data length error due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x809A	Process data reception error	Failed to receive process data due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x809B	Error opening network device	Failed to open network device due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.

Error code	Description	Cause	Solution
0x809C	GPMC IR-Link read error	GPMC IR-Link read error due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x809D	GPMC IR-Link write error	GPMC IR-Link write error due to hardware error	Restart the controller. If the problem persists, contact the manufacturer.
0x809E	IR-Link slave disconnected	IR-Link slave is disconnected.	Check the status of IR-Link device.
0x809F	Non-IR-Link slave device connection error	A non-IR-Link slave device is connected.	Check that the connected device meets the requirements.
0x80A0	IR-Link port 0 not connected	The IR-Link port 0 is not connected.	Check the connection of IR-Link port 0.
0x80A1	IR-Link port 1 not connected	The IR-Link port 1 is not connected.	Check the connection of IR-Link port 1.
0x80A2	IR-Link startup time setting error	IR-Link startup time is incorrectly set due to a hardware error.	Restart the controller. If the problem persists, contact the manufacturer.
0x8142	EtherCAT slave 1 disconnected	The EtherCAT slave 1 is disconnected.	Check the EtherCAT device.
0x819E	IR-Link slave 1 disconnected	The IR-Link slave 1 is disconnected.	Check the connection status of IR-Link device.
0x8242	EtherCAT slave 2 disconnected	The EtherCAT slave 2 is disconnected.	Check the EtherCAT device.
0x829E	IR-Link slave 2 disconnected	The IR-Link slave 2 is disconnected.	Check the connection status of IR-Link device.
0x8342	EtherCAT slave 3 disconnected	The EtherCAT slave 3 is disconnected.	Check the EtherCAT device.
0x839E	IR-Link slave 3 disconnected	The IR-Link slave 3 is disconnected.	Check the connection status of IR-Link device.
0x8442	EtherCAT slave 4 disconnected	The EtherCAT slave 4 is disconnected.	Check the EtherCAT device.
0x849E	IR-Link slave 4 disconnected	The IR-Link slave 4 is disconnected.	Check the connection status of IR-Link device.
0x8542	EtherCAT slave 5 disconnected	The EtherCAT slave 5 is disconnected.	Check the EtherCAT device.

Error code	Description	Cause	Solution
0x859E	IR-Link slave 5 disconnected	The IR-Link slave 5 is disconnected.	Check the connection status of IR-Link device.
0x8642	EtherCAT slave 6 disconnected	The EtherCAT slave 6 is disconnected.	Check the EtherCAT device.
0x869E	IR-Link slave 6 disconnected	The IR-Link slave 6 is disconnected.	Check the connection status of IR-Link device.
0x8742	EtherCAT slave 7 disconnected	The EtherCAT slave 7 is disconnected.	Check the EtherCAT device.
0x879E	IR-Link slave 7 disconnected	The IR-Link slave 7 is disconnected.	Check the connection status of IR-Link device.
0x8842	EtherCAT slave 8 disconnected	The EtherCAT slave 8 is disconnected.	Check the EtherCAT device.
0x889E	IR-Link slave 8 disconnected	The IR-LINK slave 8 is disconnected.	Check the connection status of IR-Link device.
0x8942	EtherCAT slave 9 disconnected	The EtherCAT slave 9 is disconnected.	Check the EtherCAT device.
0x899E	IR-Link slave 9 disconnected	The IR-Link slave 9 is disconnected.	Check the connection status of IR-Link device.
0x8A42	EtherCAT slave 10 disconnected	The EtherCAT slave 10 is disconnected.	Check the EtherCAT device.
0x8A9E	IR-Link slave 10 disconnected	The IR-Link slave 10 is disconnected.	Check the connection status of IR-Link device.
0x8B42	EtherCAT slave 11 disconnected	The EtherCAT slave 11 is disconnected.	Check the EtherCAT device.
0x8B9E	IR-Link slave 11 disconnected	The IR-Link slave 11 is disconnected.	Check the connection status of IR-Link device.
0x8C42	EtherCAT slave 12 disconnected	The EtherCAT slave 12 is disconnected.	Check the EtherCAT device.
0x8C9E	IR-Link slave 12 disconnected	The IR-LINK slave 12 is disconnected.	Check the connection status of IR-Link device.
0x8D42	EtherCAT slave 13 disconnected	The EtherCAT slave 13 is disconnected.	Check the EtherCAT device.
0x8D9E	IR-Link slave 13 disconnected	The IR-Link slave 13 is disconnected.	Check the connection status of IR-Link device.

Error code	Description	Cause	Solution
0x8E42	EtherCAT slave 14 disconnected	The EtherCAT slave 14 is disconnected.	Check the EtherCAT device.
0x8E9E	IR-Link slave 14 disconnected	The IR-Link slave 14 is disconnected.	Check the connection status of IR-Link device.
0x8F42	EtherCAT slave 15 disconnected	The EtherCAT slave 15 is disconnected.	Check the EtherCAT device.
0x8F9E	IR-Link slave 15 disconnected	The IR-Link slave 15 is disconnected.	Check the connection status of IR-Link device.
0x9016	J1 exceeds Group 1 speed limit	J1 exceeds Group 1 speed limit	Clear the alarm.
0x9017	J2 exceeds Group 1 speed limit	J2 exceeds Group 1 speed limit	Clear the alarm.
0x9018	J3 exceeds Group 1 speed limit	J3 exceeds Group 1 speed limit	Clear the alarm.
0x9019	J4 exceeds Group 1 speed limit	J4 exceeds Group 1 speed limit	Clear the alarm.
0x901A	J5 exceeds Group 1 speed limit	J5 exceeds Group 1 speed limit	Clear the alarm.
0x901B	J6 exceeds Group 1 speed limit	J6 exceeds Group 1 speed limit	Clear the alarm.
0x901C	J1 exceeds Group 2 speed limit	J1 exceeds Group 2 speed limit	Clear the alarm.
0x901D	J2 exceeds Group 2 speed limit	J2 exceeds Group 2 speed limit	Clear the alarm.
0x901E	J3 exceeds Group 2 speed limit	J3 exceeds Group 2 speed limit	Clear the alarm.
0x901F	J4 exceeds Group 2 speed limit	J4 exceeds Group 2 speed limit	Clear the alarm.
0x9020	J5 exceeds Group 2 speed limit	J5 exceeds Group 2 speed limit	Clear the alarm.
0x9021	J6 exceeds Group 2 speed limit	J6 exceeds Group 2 speed limit	Clear the alarm.
0x9022	J1 exceeds Group 3 speed limit	J1 exceeds Group 3 speed limit	Clear the alarm.

Error code	Description	Cause	Solution
0x9023	J2 exceeds Group 3 speed limit	J2 exceeds Group 3 speed limit	Clear the alarm.
0x9024	J3 exceeds Group 3 speed limit	J3 exceeds Group 3 speed limit	Clear the alarm.
0x9025	J4 exceeds Group 3 speed limit	J4 exceeds Group 3 speed limit	Clear the alarm.
0x9026	J5 exceeds Group 3 speed limit	J5 exceeds Group 3 speed limit	Clear the alarm.
0x9027	J6 exceeds Group 3 speed limit	J6 exceeds Group 3 speed limit	Clear the alarm.
0x9028	J1 exceeds Group 4 speed limit	J1 exceeds Group 4 speed limit	Clear the alarm.
0x9029	J2 exceeds Group 4 speed limit	J2 exceeds Group 4 speed limit	Clear the alarm.
0x902A	J3 exceeds Group 4 speed limit	J3 exceeds Group 4 speed limit	Clear the alarm.
0x902B	J4 exceeds Group 4 speed limit	J4 exceeds Group 4 speed limit	Clear the alarm.
0x902C	J5 exceeds Group 4 speed limit	J5 exceeds Group 4 speed limit	Clear the alarm.
0x902D	J6 exceeds Group 4 speed limit	J6 exceeds Group 4 speed limit	Clear the alarm.
0x902E	J1 exceeds Group 5 speed limit	J1 exceeds Group 5 speed limit	Clear the alarm.
0x902F	J2 exceeds Group 5 speed limit	J2 exceeds Group 5 speed limit	Clear the alarm.
0x9030	J3 exceeds Group 5 speed limit	J3 exceeds Group 5 speed limit	Clear the alarm.
0x9031	J4 exceeds Group 5 speed limit	J4 exceeds Group 5 speed limit	Clear the alarm.
0x9032	J5 exceeds Group 5 speed limit	J5 exceeds Group 5 speed limit	Clear the alarm.
0x9033	J6 exceeds Group 5 speed limit	J6 exceeds Group 5 speed limit	Clear the alarm.

Error code	Description	Cause	Solution
0x9034	J1 exceeds Group 6 speed limit	J1 exceeds Group 6 speed limit	Clear the alarm.
0x9035	J2 exceeds Group 6 speed limit	J2 exceeds Group 6 speed limit	Clear the alarm.
0x9036	J3 exceeds Group 6 speed limit	J3 exceeds Group 6 speed limit	Clear the alarm.
0x9037	J4 exceeds Group 6 speed limit	J4 exceeds Group 6 speed limit	Clear the alarm.
0x9038	J5 exceeds Group 6 speed limit	J5 exceeds Group 6 speed limit	Clear the alarm.
0x9039	J6 exceeds Group 6 speed limit	J6 exceeds Group 6 speed limit	Clear the alarm.
0x903A	J1 exceeds Group 7 speed limit	J1 exceeds Group 7 speed limit	Clear the alarm.
0x903B	J2 exceeds Group 7 speed limit	J2 exceeds Group 7 speed limit	Clear the alarm.
0x903C	J3 exceeds Group 7 speed limit	J3 exceeds Group 7 speed limit	Clear the alarm.
0x903D	J4 exceeds Group 7 speed limit	J4 exceeds Group 7 speed limit	Clear the alarm.
0x903E	J5 exceeds Group 7 speed limit	J5 exceeds Group 7 speed limit	Clear the alarm.
0x903F	J6 exceeds Group 7 speed limit	J6 exceeds Group 7 speed limit	Clear the alarm.
0x9040	J1 exceeds Group 8 speed limit	J1 exceeds Group 8 speed limit	Clear the alarm.
0x9041	J2 exceeds Group 8 speed limit	J2 exceeds Group 8 speed limit	Clear the alarm.
0x9042	J3 exceeds Group 8 speed limit	J3 exceeds Group 8 speed limit	Clear the alarm.
0x9043	J4 exceeds Group 8 speed limit	J4 exceeds Group 8 speed limit	Clear the alarm.
0x9044	J5 exceeds Group 8 speed limit	J5 exceeds Group 8 speed limit	Clear the alarm.

Error code	Description	Cause	Solution
0x9045	J6 exceeds Group 8 speed limit	J6 exceeds Group 8 speed limit	Clear the alarm.
0x9046	J1 exceeds Group 1 position limit	J1 exceeds Group 1 position limit	Close the joint position monitoring function or deactivate Group 1, and move the robot to a safe position.
0x9047	J2 exceeds Group 1 position limit	J2 exceeds Group 1 position limit	Close the joint position monitoring function or deactivate Group 1, and move the robot to a safe position.
0x9048	J3 exceeds Group 1 position limit	J3 exceeds Group 1 position limit	Close the joint position monitoring function or deactivate Group 1, and move the robot to a safe position.
0x9049	J4 exceeds Group 1 position limit	J4 exceeds Group 1 position limit	Close the joint position monitoring function or deactivate Group 1, and move the robot to a safe position.
0x904A	J5 exceeds Group 1 position limit	J5 exceeds Group 1 position limit	Close the joint position monitoring function or deactivate Group 1, and move the robot to a safe position.
0x904B	J6 exceeds Group 1 position limit	J6 exceeds Group 1 position limit	Close the joint position monitoring function or deactivate Group 1, and move the robot to a safe position.
0x904C	J1 exceeds Group 2 position limit	J1 exceeds Group 2 position limit	Close the joint position monitoring function or deactivate Group 2, and move the robot to a safe position.
0x904D	J2 exceeds Group 2 position limit	J2 exceeds Group 2 position limit	Close the joint position monitoring function or deactivate Group 2, and move the robot to a safe position.
0x904E	J3 exceeds Group 2 position limit	J3 exceeds Group 2 position limit	Close the joint position monitoring function or deactivate Group 2, and move the robot to a safe position.

Error code	Description	Cause	Solution
0x904F	J4 exceeds Group 2 position limit	J4 exceeds Group 2 position limit	Close the joint position monitoring function or deactivate Group 2, and move the robot to a safe position.
0x9050	J5 exceeds Group 2 position limit	J5 exceeds Group 2 position limit	Close the joint position monitoring function or deactivate Group 2, and move the robot to a safe position.
0x9051	J6 exceeds Group 2 position limit	J6 exceeds Group 2 position limit	Close the joint position monitoring function or deactivate Group 2, and move the robot to a safe position.
0x9052	J1 exceeds Group 3 position limit	J1 exceeds Group 3 position limit	Close the joint position monitoring function or deactivate Group 3, and move the robot to a safe position.
0x9053	J2 exceeds Group 3 position limit	J2 exceeds Group 3 position limit	Close the joint position monitoring function or deactivate Group 3, and move the robot to a safe position.
0x9054	J3 exceeds Group 3 position limit	J3 exceeds Group 3 position limit	Close the joint position monitoring function or deactivate Group 3, and move the robot to a safe position.
0x9055	J4 exceeds Group 3 position limit	J4 exceeds Group 3 position limit	Close the joint position monitoring function or deactivate Group 3, and move the robot to a safe position.
0x9056	J5 exceeds Group 3 position limit	J5 exceeds Group 3 position limit	Close the joint position monitoring function or deactivate Group 3, and move the robot to a safe position.
0x9057	J6 exceeds Group 3 position limit	J6 exceeds Group 3 position limit	Close the joint position monitoring function or deactivate Group 3, and move the robot to a safe position.
0x9058	J1 exceeds Group 4 position limit	J1 exceeds Group 4 position limit	Close the joint position monitoring function or deactivate Group 4, and move the robot to a safe position.

Error code	Description	Cause	Solution
0x9059	J2 exceeds Group 4 position limit	J2 exceeds Group 4 position limit	Close the joint position monitoring function or deactivate Group 4, and move the robot to a safe position.
0x905A	J3 exceeds Group 4 position limit	J3 exceeds Group 4 position limit	Close the joint position monitoring function or deactivate Group 4, and move the robot to a safe position.
0x905B	J4 exceeds Group 4 position limit	J4 exceeds Group 4 position limit	Close the joint position monitoring function or deactivate Group 4, and move the robot to a safe position.
0x905C	J5 exceeds Group 4 position limit	J5 exceeds Group 4 position limit	Close the joint position monitoring function or deactivate Group 4, and move the robot to a safe position.
0x905D	J6 exceeds Group 4 position limit	J6 exceeds Group 4 position limit	Close the joint position monitoring function or deactivate Group 4, and move the robot to a safe position.
0x905E	J1 exceeds Group 5 position limit	J1 exceeds Group 5 position limit	Close the joint position monitoring function or deactivate Group 5, and move the robot to a safe position.
0x905F	J2 exceeds Group 5 position limit	J2 exceeds Group 5 position limit	Close the joint position monitoring function or deactivate Group 5, and move the robot to a safe position.
0x9060	J3 exceeds Group 5 position limit	J3 exceeds Group 5 position limit	Close the joint position monitoring function or deactivate Group 5, and move the robot to a safe position.
0x9061	J4 exceeds Group 5 position limit	J4 exceeds Group 5 position limit	Close the joint position monitoring function or deactivate Group 5, and move the robot to a safe position.
0x9062	J5 exceeds Group 5 position limit	J5 exceeds Group 5 position limit	Close the joint position monitoring function or deactivate Group 5, and move the robot to a safe position.

Error code	Description	Cause	Solution
0x9063	J6 exceeds Group 5 position limit	J6 exceeds Group 5 position limit	Close the joint position monitoring function or deactivate Group 5, and move the robot to a safe position.
0x9064	J1 exceeds Group 6 position limit	J1 exceeds Group 6 position limit	Close the joint position monitoring function or deactivate Group 6, and move the robot to a safe position.
0x9065	J2 exceeds Group 6 position limit	J2 exceeds Group 6 position limit	Close the joint position monitoring function or deactivate Group 6, and move the robot to a safe position.
0x9066	J3 exceeds Group 6 position limit	J3 exceeds Group 6 position limit	Close the joint position monitoring function or deactivate Group 6, and move the robot to a safe position.
0x9067	J4 exceeds Group 6 position limit	J4 exceeds Group 6 position limit	Close the joint position monitoring function or deactivate Group 6, and move the robot to a safe position.
0x9068	J5 exceeds Group 6 position limit	J5 exceeds Group 6 position limit	Close the joint position monitoring function or deactivate Group 6, and move the robot to a safe position.
0x9069	J6 exceeds Group 6 position limit	J6 exceeds Group 6 position limit	Close the joint position monitoring function or deactivate Group 6, and move the robot to a safe position.
0x906A	J1 exceeds Group 7 position limit	J1 exceeds Group 7 position limit	Close the joint position monitoring function or deactivate Group 7, and move the robot to a safe position.
0x906B	J2 exceeds Group 7 position limit	J2 exceeds Group 7 position limit	Close the joint position monitoring function or deactivate Group 7, and move the robot to a safe position.
0x906C	J3 exceeds Group 7 position limit	J3 exceeds Group 7 position limit	Close the joint position monitoring function or deactivate Group 7, and move the robot to a safe position.

Error code	Description	Cause	Solution
0x906D	J4 exceeds Group 7 position limit	J4 exceeds Group 7 position limit	Close the joint position monitoring function or deactivate Group 7, and move the robot to a safe position.
0x906E	J5 exceeds Group 7 position limit	J5 exceeds Group 7 position limit	Close the joint position monitoring function or deactivate Group 7, and move the robot to a safe position.
0x906F	J6 exceeds Group 7 position limit	J6 exceeds Group 7 position limit	Close the joint position monitoring function or deactivate Group 7, and move the robot to a safe position.
0x9070	J1 exceeds Group 8 position limit	J1 exceeds Group 8 position limit	Close the joint position monitoring function or deactivate Group 8, and move the robot to a safe position.
0x9071	J2 exceeds Group 8 position limit	J2 exceeds Group 8 position limit	Close the joint position monitoring function or deactivate Group 8, and move the robot to a safe position.
0x9072	J3 exceeds Group 8 position limit	J3 exceeds Group 8 position limit	Close the joint position monitoring function or deactivate Group 7, and move the robot to a safe position.
0x9073	J4 exceeds Group 8 position limit	J4 exceeds Group 8 position limit	Close the joint position monitoring function or deactivate Group 8, and move the robot to a safe position.
0x9074	J5 exceeds Group 8 position limit	J5 exceeds Group 8 position limit	Close the joint position monitoring function or deactivate Group 8, and move the robot to a safe position.
0x9075	J6 exceeds Group 8 position limit	J6 exceeds Group 8 position limit	Close the joint position monitoring function or deactivate Group 8, and move the robot to a safe position.
0x9077	Overspeed in manual mode	The TCP speed of the robot in manual mode exceeds 250 mm/s.	Clear the alarm.
0x9078	TCP speed exceeds Group 1 limit	The TCP speed of the robot exceeds the Group 1 limit.	Clear the alarm.

Error code	Description	Cause	Solution
0x9079	TCP speed exceeds Group 2 limit	The TCP speed of the robot exceeds the Group 2 limit.	Clear the alarm.
0x907A	TCP speed exceeds Group 3 limit	The TCP speed of the robot exceeds the Group 3 limit.	Clear the alarm.
0x907B	TCP speed exceeds Group 4 limit	The TCP speed of the robot exceeds the Group 4 limit.	Clear the alarm.
0x907C	TCP speed exceeds Group 5 limit	The TCP speed of the robot exceeds the Group 5 limit.	Clear the alarm.
0x907D	TCP speed exceeds Group 6 limit	The TCP speed of the robot exceeds the Group 6 limit.	Clear the alarm.
0x907E	TCP speed exceeds Group 7 limit	The TCP speed of the robot exceeds the Group 7 limit.	Clear the alarm.
0x907F	TCP speed exceeds Group 8 limit	The TCP speed of the robot exceeds the Group 8 limit.	Clear the alarm.
0x9080	Robot entered interference zone 1	The robot enters the interference zone 1.	Close the TCP position monitoring function or deactivate the interference zone 1.
0x9081	Robot entered interference zone 2	The robot enters the interference zone 2.	Close the TCP position monitoring function or deactivate the interference zone 2.
0x9082	Robot entered interference zone 3	The robot enters the interference zone 3.	Close the TCP position monitoring function or deactivate the interference zone 3.
0x9083	Robot entered interference zone 4	The robot enters the interference zone 4.	Close the TCP position monitoring function or deactivate the interference zone 4.
0x9084	Robot entered interference zone 5	The robot enters the interference zone 5.	Close the TCP position monitoring function or deactivate the interference zone 5.
0x9085	Robot entered interference zone 6	The robot enters the interference zone 6.	Close the TCP position monitoring function or deactivate the interference zone 6.
0x9086	Robot entered interference zone 7	The robot enters the interference zone 7.	Close the TCP position monitoring function or deactivate the interference zone 7.

Error code	Description	Cause	Solution
0x9087	Robot entered interference zone 8	The robot enters the interference zone 8.	Close the TCP position monitoring function or deactivate the interference zone 8.
0x9088	Robot entered interference zone 9	The robot enters the interference zone 9.	Close the TCP position monitoring function or deactivate the interference zone 9.
0x9089	Robot entered interference zone 10	The robot enters the interference zone 10.	Close the TCP position monitoring function or deactivate the interference zone 10.
0x908A	Robot entered interference zone 11	The robot enters the interference zone 11.	Close the TCP position monitoring function or deactivate the interference zone 11.
0x908B	Robot entered interference zone 12	The robot enters the interference zone 12.	Close the TCP position monitoring function or deactivate the interference zone 12.
0x908C	Robot entered interference zone 13	The robot enters the interference zone 13.	Close the TCP position monitoring function or deactivate the interference zone 13.
0x908D	Robot entered interference zone 14	The robot enters the interference zone 14.	Close the TCP position monitoring function or deactivate the interference zone 14.
0x908E	Robot entered interference zone 15	The robot enters the interference zone 15.	Close the TCP position monitoring function or deactivate the interference zone 15.
0x908F	Robot entered interference zone 16	The robot enters the interference zone 16.	Close the TCP position monitoring function or deactivate the interference zone 16.
0x9090	J1 position check error for master chip	J1 position check error for master chip	Contact the manufacturer.
0x9091	J2 position check error for master chip	J2 position check error for master chip	Contact the manufacturer.
0x9092	J3 position check error for master chip	J3 position check error for master chip	Contact the manufacturer.
0x9093	J4 position check error for master chip	J4 position check error for master chip	Contact the manufacturer.
0x9094	J5 position check error for master chip	J5 position check error for master chip	Contact the manufacturer.

Error code	Description	Cause	Solution
0x9095	J6 position check error for master chip	J6 position check error for master chip	Contact the manufacturer.
0x9096	J1 position check error for slave chip	J1 position check error for slave chip	Contact the manufacturer.
0x9097	J2 position check error for slave chip	J2 position check error for slave chip	Contact the manufacturer.
0x9098	J3 position check error for slave chip	J3 position check error for slave chip	Contact the manufacturer.
0x9099	J4 position check error for slave chip	J4 position check error for slave chip	Contact the manufacturer.
0x909A	J5 position check error for slave chip	J5 position check error for slave chip	Contact the manufacturer.
0x909B	J6 position check error for slave chip	J6 position check error for slave chip	Contact the manufacturer.
0x909C	Category 1 emergency stop time too long	The Category 1 emergency stop exceeds the limit.	1. Set the delay for Category 1 emergency stop to a larger value 2. Contact the manufacturer.
0x909D	STO stop time too long	The STO stop time exceeds the limit.	1. Set the delay for Category 0 emergency stop to a larger value 2. Contact the manufacturer.
0x909E	J1 CRC failed for master chip data frame	J1 CRC failed for master chip data frame	Contact the manufacturer.
0x909F	J2 CRC failed for master chip data frame	J2 CRC failed for master chip data frame	Contact the manufacturer.
0x90A0	J3 CRC failed for master chip data frame	J3 CRC failed for master chip data frame	Contact the manufacturer.
0x90A1	J4 CRC failed for master chip data frame	J4 CRC failed for master chip data frame	Contact the manufacturer.

Error code	Description	Cause	Solution
0x90A2	J5 CRC failed for master chip data frame	J5 CRC failed for master chip data frame	Contact the manufacturer.
0x90A3	J6 CRC failed for master chip data frame	J6 CRC failed for master chip data frame	Contact the manufacturer.
0x90A4	J1 CRC failed for master chip calculated data frame	J1 CRC failed for master chip calculated data frame.	Contact the manufacturer.
0x90A5	J2 CRC failed for master chip calculated data frame	J2 CRC failed for master chip calculated data frame.	Contact the manufacturer.
0x90A6	J3 CRC failed for master chip calculated data frame	J3 CRC failed for master chip calculated data frame.	Contact the manufacturer.
0x90A7	J4 CRC failed for master chip calculated data frame	J4 CRC failed for master chip calculated data frame.	Contact the manufacturer.
0x90A8	J5 CRC failed for master chip calculated data frame	J5 CRC failed for master chip calculated data frame.	Contact the manufacturer.
0x90A9	J6 CRC failed for master chip calculated data frame	J6 CRC failed for master chip calculated data frame.	Contact the manufacturer.
0x90AA	J1 CRC failed for slave chip data frame	J1 CRC failed for slave chip data frame	Contact the manufacturer.
0x90AB	J2 CRC failed for slave chip data frame	J2 CRC failed for slave chip data frame	Contact the manufacturer.
0x90AC	J3 CRC failed for slave chip data frame	J3 CRC failed for slave chip data frame	Contact the manufacturer.
0x90AD	J4 CRC failed for slave chip data frame	J4 CRC failed for slave chip data frame	Contact the manufacturer.
0x90AE	J5 CRC failed for slave chip data frame	J5 CRC failed for slave chip data frame	Contact the manufacturer.
0x90AF	J6 CRC failed for slave chip data frame	J6 CRC failed for slave chip data frame	Contact the manufacturer.

Error code	Description	Cause	Solution
0x90B0	J1 CRC failed for slave chip calculated data frame	J1 CRC failed for slave chip calculated data frame.	Contact the manufacturer.
0x90B1	J2 CRC failed for slave chip calculated data frame	J2 CRC failed for slave chip calculated data frame.	Contact the manufacturer.
0x90B2	J3 CRC failed for slave chip calculated data frame	J3 CRC failed for slave chip calculated data frame.	Contact the manufacturer.
0x90B3	J4 CRC failed for slave chip calculated data frame	J4 CRC failed for slave chip calculated data frame.	Contact the manufacturer.
0x90B4	J5 CRC failed for slave chip calculated data frame	J5 CRC failed for slave chip calculated data frame.	Contact the manufacturer.
0x90B5	J6 CRC failed for slave chip calculated data frame	J6 CRC failed for slave chip calculated data frame.	Contact the manufacturer.
0x90B6	J1 SN code fluctuates in master chip data frame	J1 SN code fluctuates in master chip data frame	Contact the manufacturer.
0x90B7	J2 SN code fluctuates in master chip data frame	J2 SN code fluctuates in master chip data frame	Contact the manufacturer.
0x90B8	J3 SN code fluctuates in master chip data frame	J3 SN code fluctuates in master chip data frame	Contact the manufacturer.
0x90B9	J4 SN code fluctuates in master chip data frame	J4 SN code fluctuates in master chip data frame	Contact the manufacturer.
0x90BA	J5 SN code fluctuates in master chip data frame	J5 SN code fluctuates in master chip data frame	Contact the manufacturer.
0x90BB	J6 SN code fluctuates in master chip data frame	J6 SN code fluctuates in master chip data frame	Contact the manufacturer.

Error code	Description	Cause	Solution
0x90BC	J1 SN code fluctuates in master chip calculated data frame	J1 SN code fluctuates in master chip calculated data frame	Contact the manufacturer.
0x90BD	J2 SN code fluctuates in master chip calculated data frame	J2 SN code fluctuates in master chip calculated data frame	Contact the manufacturer.
0x90BE	J3 SN code fluctuates in master chip calculated data frame	J3 SN code fluctuates in master chip calculated data frame	Contact the manufacturer.
0x90BF	J4 SN code fluctuates in master chip calculated data frame	J4 SN code fluctuates in master chip calculated data frame	Contact the manufacturer.
0x90C0	J5 SN code fluctuates in master chip calculated data frame	J5 SN code fluctuates in master chip calculated data frame	Contact the manufacturer.
0x90C1	J6 SN code fluctuates in master chip calculated data frame	J6 SN code fluctuates in master chip calculated data frame	Contact the manufacturer.
0x90C2	J1 SN code fluctuates in slave chip data frame	J1 SN code fluctuates in slave chip data frame	Contact the manufacturer.
0x90C3	J2 SN code fluctuates in slave chip data frame	J2 SN code fluctuates in slave chip data frame	Contact the manufacturer.
0x90C4	J3 SN code fluctuates in slave chip data frame	J3 SN code fluctuates in slave chip data frame	Contact the manufacturer.
0x90C5	J4 SN code fluctuates in slave chip data frame	J4 SN code fluctuates in slave chip data frame	Contact the manufacturer.
0x90C6	J5 SN code fluctuates in slave chip data frame	J5 SN code fluctuates in slave chip data frame	Contact the manufacturer.
0x90C7	J6 SN code fluctuates in slave chip data frame	J6 SN code fluctuates in slave chip data frame	Contact the manufacturer.

Error code	Description	Cause	Solution
0x90C8	J1 SN code fluctuates in slave chip calculated data frame	J1 SN code fluctuates in slave chip calculated data frame	Contact the manufacturer.
0x90C9	J2 SN code fluctuates in slave chip calculated data frame	J2 SN code fluctuates in slave chip calculated data frame	Contact the manufacturer.
0x90CA	J3 SN code fluctuates in slave chip calculated data frame	J3 SN code fluctuates in slave chip calculated data frame	Contact the manufacturer.
0x90CB	J4 SN code fluctuates in slave chip calculated data frame	J4 SN code fluctuates in slave chip calculated data frame	Contact the manufacturer.
0x90CC	J5 SN code fluctuates in slave chip calculated data frame	J5 SN code fluctuates in slave chip calculated data frame	Contact the manufacturer.
0x90CD	J6 SN code fluctuates in slave chip calculated data frame	J6 SN code fluctuates in slave chip calculated data frame	Contact the manufacturer.
0x90CE	J1 multi-turn value in master chip overflows	J1 multi-turn value in master chip overflows	Contact the manufacturer.
0x90CF	J2 multi-turn value in master chip overflows	J2 multi-turn value in master chip overflows	Contact the manufacturer.
0x90D0	J3 multi-turn value in master chip overflows	J3 multi-turn value in master chip overflows	Contact the manufacturer.
0x90D1	J4 multi-turn value in master chip overflows	J4 multi-turn value in master chip overflows	Contact the manufacturer.
0x90D2	J5 multi-turn value in master chip overflows	J5 multi-turn value in master chip overflows	Contact the manufacturer.
0x90D3	J6 multi-turn value in master chip overflows	J6 multi-turn value in master chip overflows	Contact the manufacturer.
0x90D4	J1 estimated multi-turn value in master chip overflows	J1 estimated multi-turn value in master chip overflows	Contact the manufacturer.
0x90D5	J2 estimated multi-turn value in master chip overflows	J2 estimated multi-turn value in master chip overflows	Contact the manufacturer.

Error code	Description	Cause	Solution
0x90D6	J3 estimated multi-turn value in master chip overflows	J3 estimated multi-turn value in master chip overflows	Contact the manufacturer.
0x90D7	J4 estimated multi-turn value in master chip overflows	J4 estimated multi-turn value in master chip overflows	Contact the manufacturer.
0x90D8	J5 estimated multi-turn value in master chip overflows	J5 estimated multi-turn value in master chip overflows	Contact the manufacturer.
0x90D9	J6 estimated multi-turn value in master chip overflows	J6 estimated multi-turn value in master chip overflows	Contact the manufacturer.
0x90DA	J1 multi-turn value in slave chip overflows	J1 multi-turn value in slave chip overflows	Contact the manufacturer.
0x90DB	J2 multi-turn value in slave chip overflows	J2 multi-turn value in slave chip overflows	Contact the manufacturer.
0x90DC	J3 multi-turn value in slave chip overflows	J3 multi-turn value in slave chip overflows	Contact the manufacturer.
0x90DD	J4 multi-turn value in slave chip overflows	J4 multi-turn value in slave chip overflows	Contact the manufacturer.
0x90DE	J5 multi-turn value in slave chip overflows	J5 multi-turn value in slave chip overflows	Contact the manufacturer.
0x90DF	J6 multi-turn value in slave chip overflows	J6 multi-turn value in slave chip overflows	Contact the manufacturer.
0x90E0	J1 estimated multi-turn value in slave chip overflows	J1 estimated multi-turn value in slave chip overflows	Contact the manufacturer.
0x90E1	J2 estimated multi-turn value in slave chip overflows	J2 estimated multi-turn value in slave chip overflows	Contact the manufacturer.
0x90E2	J3 estimated multi-turn value in slave chip overflows	J3 estimated multi-turn value in slave chip overflows	Contact the manufacturer.
0x90E3	J4 estimated multi-turn value in slave chip overflows	J4 estimated multi-turn value in slave chip overflows	Contact the manufacturer.

Error code	Description	Cause	Solution
0x90E4	J5 estimated multi-turn value in slave chip overflows	J5 estimated multi-turn value in slave chip overflows	Contact the manufacturer.
0x90E5	J6 estimated multi-turn value in slave chip overflows	J6 estimated multi-turn value in slave chip overflows	Contact the manufacturer.
0x90E6	Joint position state abnormal	The joint position state is checked inconsistent between master and slave chips.	Contact the manufacturer.
0x90E7	Joint speed state abnormal	The joint speed state is checked inconsistent between master and slave chips.	Contact the manufacturer.
0x90E8	TCP speed state abnormal	The TCP speed state is checked inconsistent between master and slave chips.	Contact the manufacturer.
0x90E9	Low-speed safety function state abnormal	The low-speed state is checked inconsistent between master and slave chips.	Contact the manufacturer.
0x90EA	TCP position monitoring function state abnormal	The TCP position state is checked inconsistent between master and slave chips.	Contact the manufacturer.
0x90EB	Category 1 emergency stop state abnormal	The emergency stop state is checked inconsistent between master and slave chips.	Contact the manufacturer.
0x90EC	Running state abnormal	The running state is checked inconsistent between master and slave chips.	Contact the manufacturer.
0x90ED	System initialization failed	System initialization failed	Contact the manufacturer.
0x90EE	Abnormal J1 in master chip data frame	Abnormal J1 in master chip data frame	Contact the manufacturer.
0x90EF	Abnormal J2 in master chip data frame	Abnormal J2 in master chip data frame	Contact the manufacturer.
0x90F0	Abnormal J3 in master chip data frame	Abnormal J3 in master chip data frame	Contact the manufacturer.
0x90F1	Abnormal J4 in master chip data frame	Abnormal J4 in master chip data frame	Contact the manufacturer.

Error code	Description	Cause	Solution
0x90F2	Abnormal J5 in master chip data frame	Abnormal J5 in master chip data frame	Contact the manufacturer.
0x90F3	Abnormal J6 in master chip data frame	Abnormal J6 in master chip data frame	Contact the manufacturer.
0x90F4	Abnormal J1 in slave chip data frame	Abnormal J1 in slave chip data frame	Contact the manufacturer.
0x90F5	Abnormal J2 in slave chip data frame	Abnormal J2 in slave chip data frame	Contact the manufacturer.
0x90F6	Abnormal J3 in slave chip data frame	Abnormal J3 in slave chip data frame	Contact the manufacturer.
0x90F7	Abnormal J4 in slave chip data frame	Abnormal J4 in slave chip data frame	Contact the manufacturer.
0x90F8	Abnormal J5 in slave chip data frame	Abnormal J5 in slave chip data frame	Contact the manufacturer.
0x90F9	Abnormal J6 in slave chip data frame	Abnormal J6 in slave chip data frame	Contact the manufacturer.
0x90FA	Abnormal J1 in slave chip calculated data frame	Abnormal J1 in slave chip calculated data frame	Contact the manufacturer.
0x90FB	Abnormal J2 in slave chip calculated data frame	Abnormal J2 in slave chip calculated data frame	Contact the manufacturer.
0x90FC	Abnormal J3 in slave chip calculated data frame	Abnormal J3 in slave chip calculated data frame	Contact the manufacturer.
0x90FD	Abnormal J4 in slave chip calculated data frame	Abnormal J4 in slave chip calculated data frame	Contact the manufacturer.
0x90FE	Abnormal J5 in slave chip calculated data frame	Abnormal J5 in slave chip calculated data frame	Contact the manufacturer.
0x90FF	Abnormal J6 in slave chip calculated data frame	Abnormal J6 in slave chip calculated data frame	Contact the manufacturer.

Error code	Description	Cause	Solution
0x9100	Abnormal J1 in master chip calculated data frame	Abnormal J1 in master chip calculated data frame	Contact the manufacturer.
0x9101	Abnormal J2 in master chip calculated data frame	Abnormal J2 in master chip calculated data frame	Contact the manufacturer.
0x9102	Abnormal J3 in master chip calculated data frame	Abnormal J3 in master chip calculated data frame	Contact the manufacturer.
0x9103	Abnormal J4 in master chip calculated data frame	Abnormal J4 in master chip calculated data frame	Contact the manufacturer.
0x9104	Abnormal J5 in master chip calculated data frame	Abnormal J5 in master chip calculated data frame	Contact the manufacturer.
0x9105	Abnormal J6 in master chip calculated data frame	Abnormal J6 in master chip calculated data frame	Contact the manufacturer.
0x9106	Master chip triggers Category 0 stop	Master chip triggers Category 0 stop	Clear the alarm.
0x9107	Slave chip triggers Category 0 stop	Slave chip triggers Category 0 stop	Clear the alarm.
0x9200	Batch point modification communication data error	Batch point modification communication data error	Clear the alarm.
0x9201	Batch point modification initialization error	Batch point modification initialization error	Clear the alarm.
0x9202	Error reading point file for batch point modification	Error reading point file for batch point modification	Clear the alarm.
0x9203	Error modifying point cache for batch point modification, please restart the controller.	Error modifying point cache for batch point modification, please restart the controller.	Restart the controller. If the problem persists, contact the manufacturer.

Error code	Description	Cause	Solution
0x9204	Error modifying point file for batch point modification, please restart the controller.	Error modifying point file for batch point modification, please restart the controller.	Restart the controller. If the problem persists, contact the manufacturer.
0x9301	MCUA failed to read EEPROM	Communication error between chip and EEPROM	Contact the manufacturer.
0x9302	MCUB failed to read EEPROM	Communication error between chip and EEPROM	Contact the manufacturer.
0x9303	MCUA failed to write EEPROM	Communication error between chip and EEPROM	Contact the manufacturer.
0x9304	MCUB failed to write EEPROM	Communication error between chip and EEPROM	Contact the manufacturer.
0x9305	MCUA has a DMA transmission error	Communication error between chip and EEPROM	Contact the manufacturer.
0x9306	MCUB has a DMA transmission error	Communication error between chip and EEPROM	Contact the manufacturer.
0x9307	MCUB discharge voltage collection failed	Discharge voltage abnormal	Contact the manufacturer.
0x9308	MCUA 3.3 V abnormal	The power supply voltage of the safety controller is abnormal.	Check the power supply voltage.
0x9309	MCUB 3.3 V abnormal	The power supply voltage of the safety controller is abnormal.	Check the power supply voltage.
0x930A	MCUA board voltage abnormal	The board voltage of safety controller is abnormal.	Check the power supply voltage.
0x930B	MCUB board voltage abnormal	The board voltage of safety controller is abnormal.	Check the power supply voltage.
0x930C	CRC check failed during MCUA initialization	EEPROM parameter in MCUA is abnormal.	Contact the manufacturer.
0x930D	CRC check failed during MCUB initialization	EEPROM parameter in MCUB is abnormal.	Contact the manufacturer.

Error code	Description	Cause	Solution
0x930E	CRC check failed during MCUA mutual check	EEPROM parameter in MCUA is abnormal.	Contact the manufacturer.
0x930F	CRC check failed during MCUB mutual check	EEPROM parameter in MCUB is abnormal.	Contact the manufacturer.
0x9310	MCUA chip self-test failed	The chip is damaged.	Contact the manufacturer.
0x9311	MCUB chip self-test failed	The chip is damaged.	Contact the manufacturer.
0x9312	MCUA failed to parse mailbox data of the host controller	Communication error	Contact the manufacturer.
0x9313	MCUA failed to process mailbox data of the host controller	Communication error	Contact the manufacturer.
0x9314	MCUA failed to parse MCUB mailbox data	Communication error	Contact the manufacturer.
0x9315	MCUA failed to process MCUB mailbox data	Communication error	Contact the manufacturer.
0x9316	MCUB failed to parse MCUA mailbox data	Communication error	Contact the manufacturer.
0x9317	MCUB failed to process MCUA mailbox data	Communication error	Contact the manufacturer.
0x9318	MCUA detects disconnection from MCUB	Communication error	Contact the manufacturer.
0x9319	MCUA detects disconnection from FPGA	Communication error	Contact the manufacturer.
0x931A	MCUA failed to compare DIs	DI abnormal	Contact the manufacturer.
0x931B	MCUB failed to compare DIs	DI abnormal	Contact the manufacturer.

Error code	Description	Cause	Solution
0x931C	MCUA timed out while waiting for a message from MCUB	Communication error	Contact the manufacturer.
0x931D	MCUA interrupt execution timeout	Communication error	Contact the manufacturer.
0x931E	MCUB interrupt execution timeout	Communication error	Contact the manufacturer.
0x931F	MCUA timing abnormal	Communication error	Contact the manufacturer.
0x9320	MCUB timing abnormal	Communication error	Contact the manufacturer.
0x9321	MCUA failed to compare DOs	The DO is abnormal.	Contact the manufacturer.
0x9322	MCUB failed to compare DOs	The DO is abnormal.	Contact the manufacturer.
0x9323	MCUA parameter initialization failed	Parameter saving error	Contact the manufacturer.
0x9324	MCUB parameter initialization failed	Parameter saving error	Contact the manufacturer.
0x9325	MCUA failed to save parameters	Parameter saving error	Contact the manufacturer.
0x9326	MCUB failed to save parameters	Parameter saving error	Contact the manufacturer.
0x9327	MCUA output pin read-back failed	Pin read-back error	Contact the manufacturer.
0x9328	MCUB output pin read-back failed	Pin read-back error	Contact the manufacturer.
0x9329	DI comparison failed for three-position enable	The DI inputs are inconsistent.	Ensure simultaneous DI inputs.
0x932A	Clock self-test failed	The chip or external clock source is abnormal.	Contact the manufacturer.
0x932B	CPU self-test failed	Chip exception	Contact the manufacturer.
0x932C	RAM self-test failed	Chip exception	Contact the manufacturer.

Error code	Description	Cause	Solution
0x932D	Flash self-test failed	Chip exception	Contact the manufacturer.
0x932E	MCUA triggers safety door	The safety door properties have been configured and a DI input is received.	Change the safety parameter DI configuration properties or remove the DI input
0x932F	MCUA triggers an emergency stop	The emergency stop properties have been configured and a DI input is received.	Change the safety parameter DI configuration properties or remove the DI input
0x9330	MCUB triggers safety door	The safety door properties have been configured and a DI input is received.	Change the safety parameter DI configuration properties or remove the DI input
0x9331	MCUB triggers an emergency stop	The emergency stop properties have been configured and a DI input is received.	Change the safety parameter DI configuration properties or remove the DI input
0x9332	MCUA parameter read-back failed	Safety parameter saving error	Contact the manufacturer.
0x9333	MCUB parameter read-back failed	Safety parameter saving error	Contact the manufacturer.
0x9334	Safety parameters initialized, reboot the system	The safety parameters are initialized.	Restart the controller.
0x9345	Main chip of the safety module failed to read the parameter version number	Software update caused a safety parameter compatibility issue.	Contact the manufacturer.
0x9346	Slave chip of the safety module failed to read the parameter version number	Software update caused a safety parameter compatibility issue.	Contact the manufacturer.
0x9347	Main chip of the safety module failed to read the parameter length	Software update caused a safety parameter compatibility issue.	Contact the manufacturer.
0x9348	Slave chip of the safety module failed to read the parameter length	Software update caused a safety parameter compatibility issue.	Contact the manufacturer.

Error code	Description	Cause	Solution
0x9349	CRC write error on the main chip of the safety module during software update	Software update caused a safety parameter compatibility issue.	Contact the manufacturer.
0x934A	CRC write error on the slave chip of the safety module during software update	Software update caused a safety parameter compatibility issue.	Contact the manufacturer.
0x934B	Length check error on the main chip of the safety module during program update	Software update caused a safety parameter compatibility issue.	Contact the manufacturer.
0x934C	Length check error on the slave chip of the safety module during program update	Software update caused a safety parameter compatibility issue.	Contact the manufacturer.
0x934D	Parameter version number write error on the main chip of the safety module during software update	Software update caused a safety parameter compatibility issue.	Contact the manufacturer.
0x934E	Parameter version number write error on the slave chip of the safety module during software update	Software update caused a safety parameter compatibility issue.	Contact the manufacturer.
0x934F	Length info write error on the main chip of the safety module during software update	Software update caused a safety parameter compatibility issue.	Contact the manufacturer.
0x9350	Length info write error on the slave chip of the safety module during software update	Software update caused a safety parameter compatibility issue.	Contact the manufacturer.
0x9351	Parameter error on the main chip of the safety module during software update	Software update caused a safety parameter compatibility issue.	Contact the manufacturer.

Error code	Description	Cause	Solution
0x9352	Parameter error on the slave chip of the safety module during software update	Software update caused a safety parameter compatibility issue.	Contact the manufacturer.
0x9353	Parameter write error on the main chip of the safety module during software update	Software update caused a safety parameter compatibility issue.	Contact the manufacturer.
0x9354	Parameter write error on the slave chip of the safety module during software update	Software update caused a safety parameter compatibility issue.	Contact the manufacturer.
0x9355	Incompatible version parameters detected in the main chip of the safety module	Software update caused a safety parameter compatibility issue.	Contact the manufacturer.
0x9356	Incompatible version parameters detected in the slave chip of the safety module	Software update caused a safety parameter compatibility issue.	Contact the manufacturer.
0x9361	CRC error detected while writing safety parameters to the main chip of the safety module	Software update caused a safety parameter compatibility issue.	Contact the manufacturer.
0x9362	CRC error detected while writing safety parameters to the slave chip of the safety module	Software update caused a safety parameter compatibility issue.	Contact the manufacturer.
0x9363	Error writing safety parameters during the initialization phase of the main chip of the safety module	Software update caused a safety parameter compatibility issue.	Contact the manufacturer.

Error code	Description	Cause	Solution
0x9364	Error writing safety parameters during the initialization phase of the slave chip of the safety module	Software update caused a safety parameter compatibility issue.	Contact the manufacturer.
0x9365	Error writing flag bit during the initialization phase of the main chip of the safety module	Software update caused a safety parameter compatibility issue.	Contact the manufacturer.
0x9366	Error writing flag bit during the initialization phase of the slave chip of the safety module	Software update caused a safety parameter compatibility issue.	Contact the manufacturer.
0x9500	Failed to issue robot body parameters	The communication between A40i and MCU is abnormal.	Check the communication status between A40i and MCU.
0x9501	MCU alarm clearance failed	The communication between A40i and MCU is abnormal.	Check the communication status between A40i and MCU.
0x9502	No MCU board detected	MCU is loose or has been removed.	Check if the MCU has poor contact.
0x9503	Communication between MCU and FPGA disconnected	The communication between MCU and FPGA is disconnected.	Check the MCU communication status.
0x9504	Failed to obtain MCU version	The communication between A40i and MCU is abnormal.	Check the MCU communication status.
0x9505	Safety MCU and controller version mismatch	Safety MCU and controller version mismatch	Reflash the safety MCU version or controller version.
0x9702	Excessive difference in robot joint positions before and after power-on	Excessive deviation is detected in robot Jx-axis joint position before and after power-on.	1. Run the program in manual slow mode. If the point has shifted, check whether a collision occurred on-site. 2. If no collision is found, contact Inovance Technical Support for troubleshooting. If the trajectory is normal, production can resume normally.

Error code	Description	Cause	Solution
0x9703	Excessive difference in robot encoder values before and after power-on	Excessive deviation is detected in robot Jx-axis encoder values before and after power-on.	1. Run the program in manual slow mode. If the point has shifted, check whether a collision occurred on-site. 2. If no collision is found, contact Inovance Technical Support for troubleshooting. If the trajectory is normal, production can resume normally.
0x9704	One-turn discrepancy at power cycle	There is a discrepancy of one turn in the encoder multi-turn count between the values recorded during the last power-down and the current power-on.	1. Run the program in manual slow mode. If the point has shifted, check whether a collision occurred on-site. 2. If no collision is found, contact Inovance Technical Support for troubleshooting. If the trajectory is normal, production can resume normally.
0x9705	Multi-turn discrepancy at power cycle	There is a discrepancy of multiple turns in the encoder multi-turn count between the values recorded during the last power-down and the current power-on.	1. Run the program in manual slow mode. If the point has shifted, check whether a collision occurred on-site. 2. If no collision is found, contact Inovance Technical Support for troubleshooting. If the trajectory is normal, production can resume normally.
0x9706	Abnormal power-off during motion, encoder data verification failed	Abnormal power-off during motion, encoder data verification failed	Avoid abnormal power-off operations while the robot is in motion, as this can affect the normal operation of some verification functions.
0x9707	EtherCAT communication error, unable to perform encoder data verification	1. 1. Configuration file error. 2. 2. EtherCAT slave does not match system configuration.	1. Check if the EtherCAT connection status is abnormal. 2. Contact the manufacturer.
0x9708	Encoder data read error, unable to perform encoder data verification	Encoder data read error, unable to perform encoder data verification	Optimize the connections between the servo and the controller.

Error code	Description	Cause	Solution
0x9901	Station reservation module scheduling failed	The station reservation service thread was not started properly.	1. After clearing the alarm, switch to "Disable station reservation" on the "Station configuration" page and click "Save". Then switch to "Station reservation > Bind one by one" and check for alarms. 2. Restart the controller and check if the error is fixed. If not, please contact the manufacturer.
0x9902	Station reservation configuration file validation failed	The content of the station configuration file in the project failed validation; the file may be corrupted or manually altered.	Switch to the "Station Configuration" page, reconfigure the station, then click "Save".
0x9903	Station reservation configuration file version mismatch	The station configuration file version does not match in the project.	Switch to the "Station configuration" page, reconfigure the station, and click "Save".
0x9904	Station program failed to start, please check the station configuration	The station configuration or the station program is abnormal.	1. Check if the triggered station is configured with a station program. 2. Check if the program corresponding to the station includes the StationEntry function, and that this function does not contain return values or parameters.
0x9905	The input port for the program stop function is high, preventing the workstation from starting	The input port corresponding to the program stop function is at a high level.	Check if the input port corresponding to the program stop function is at a high level. When starting the station, ensure the input port corresponding to the program stop function remains at a low level.
0x9906	Program start function invalid in one-by-one binding mode	When switching to the sequential binding mode for workstation reservation on the workstation configuration page, use the program start function to initiate the main task	When switching to the workstation reservation - individual binding mode on the workstation configuration page, use the workstation reservation input signal to start the workstation

Error code	Description	Cause	Solution
0x9907	When enabling the workstation reservation individual binding mode, the program reset function is invalid	When switching to the workstation reservation individual binding mode on the workstation configuration page, the program reset function is used	When switching to the workstation reservation individual binding mode on the workstation configuration page, the program reset function is not allowed
0x9908	When station reservation is disabled, it is not possible to start the station via station input signals	In the station configuration page, when switching to the station reservation off mode, it is not allowed to start the station using station input signals	In the station configuration page, switch to the station reservation - one-by-one binding mode
0x9909	Restart station failed, the current running program does not match the station program	When restarting the station, the running line number is not in the station program	When restarting the station, the running line number must be in the station program or within the subroutine function called before the station program paused
0xE001	Normal operation	Normal operation	No action is required.
0xE002	Firmware loading failed	Unable to get firmware from SD card	Check whether it is in test mode, check the SD card content.
0xE003	Firmware loading failed	Unable to get the correct firmware from Flash	Re-upgrade the system.
0xE004	Firmware loading failed	FPGA failed to reset	Restart the controller. If the problem persists, contact the manufacturer.
0xE005	Firmware loading failed	FPGA transmission error	Restart the controller. If the problem persists, contact the manufacturer.
0xE006	Firmware loading failed	Handshake with FPGA failed	Restart the controller. If the problem persists, contact the manufacturer.
0xE007	Control channel 1 running properly	Control channel 1 running properly	No action is required.
0xE008	Failed to allocate memory to control channel 1	No enough memory	Restart the controller. If the problem persists, contact the manufacturer.

Error code	Description	Cause	Solution
0xE009	Firmware loading failed	Firmware not found or information not available	Restart the controller. If the problem persists, contact the manufacturer.
0xE00A	Firmware loading failed	Firmware not found or information not available	Restart the controller. If the problem persists, contact the manufacturer.
0xE00B	Control channel 1 firmware length out of range	Control channel 1 firmware length out of range	Restart the controller. If the problem persists, contact the manufacturer.
0xE00C	Firmware loading failed	Unable to communicate with FPGA	Restart the controller. If the problem persists, contact the manufacturer.
0xE00D	Failed to reset control channel 1	Failed to reset control channel 1	Restart the controller. If the problem persists, contact the manufacturer.
0xE00E	SPI communication failed	Failed to send start word	Restart the controller. If the problem persists, contact the manufacturer.
0xE00F	SPI communication failed	POS action failed	Restart the controller. If the problem persists, contact the manufacturer.
0xE010	Failed to regulate communication rate of channel 1	Failed to regulate communication rate of channel 1	Restart the controller. If the problem persists, contact the manufacturer.
0xE011	Control channel 1 data loading timeout	Control channel 1 data loading timeout	Restart the controller. If the problem persists, contact the manufacturer.
0xE012	Control channel 1 running response is abnormal.	Control channel 1 running response is abnormal.	Restart the controller. If the problem persists, contact the manufacturer.
0xE013	Control channel 0 running properly	Control channel 0 running properly	No action is required.
0xE014	Failed to allocate memory to control channel 0	No enough memory	Restart the controller. If the problem persists, contact the manufacturer.
0xE015	Firmware loading failed	Firmware not found or information not available	Restart the controller. If the problem persists, contact the manufacturer.

Error code	Description	Cause	Solution
0xE016	Firmware loading failed	Firmware not found or information not available	Restart the controller. If the problem persists, contact the manufacturer.
0xE017	Control channel 0 firmware length out of range	Control channel 0 firmware length out of range	Restart the controller. If the problem persists, contact the manufacturer.
0xE018	Firmware loading failed	Unable to communicate with FPGA	Restart the controller. If the problem persists, contact the manufacturer.
0xE019	Failed to reset control channel 0	Failed to reset control channel 0	Restart the controller. If the problem persists, contact the manufacturer.
0xE01A	SPI communication failed	Failed to send start word	Restart the controller. If the problem persists, contact the manufacturer.
0xE01B	SPI communication failed	POS action failed	Restart the controller. If the problem persists, contact the manufacturer.
0xE01C	Failed to regulate communication rate of channel 0	Failed to regulate communication rate of channel 0	Restart the controller. If the problem persists, contact the manufacturer.
0xE01D	Control channel 0 data loading timeout	Control channel 0 data loading timeout	Restart the controller. If the problem persists, contact the manufacturer.
0xE01E	Control channel 0 running response is abnormal.	Control channel 0 running response is abnormal.	Restart the controller. If the problem persists, contact the manufacturer.
0xE101	Error loading controller parameters	Error loading controller parameters	Contact the manufacturer.
0xE102	Error loading model parameter	Error loading model parameter	Import the parameters of the corresponding model.
0xE103	Error loading machine parameter patch	Error loading machine parameter patch	Import the parameters of the corresponding model.
0xE104	The machine parameter version and software version are incompatible.	The machine parameter version and software version are incompatible.	Import machine parameters compatible with the software.

Error code	Description	Cause	Solution
0xE105	Error writing servo parameter	Error writing servo parameter	Contact the manufacturer.
0xE107	Servo parameters updated automatically, please reboot the controller.	Servo parameters updated automatically, please reboot the controller.	Restart the controller.
0xE109	The controller software version is incompatible with the manipulator parameter version.	The controller software version is incompatible with the manipulator parameter version.	Turn off the universality switch, reprogram or upgrade the system.
0xE10A	When the controller is first connected to the robot body, or after replacing the robot control cabinet or body, if the controller parameters do not match the body parameters, perform a wild match operation	The controller parameters are inconsistent with the manipulator parameters.	Perform universality configuration under the control authority of teach pendant or InoRobotLab. Or ensure to turn off the wildcard switch after one wildcard operation
0xE10B	The controller failed the hardware condition check with the main body. Please verify whether there are errors in the IRLINK or ECAT communication links, and whether the controller is paired with the main body	The verification of hardware conditions (power, brake, DB value) between the controller and the main body did not pass.	Please check if there are any errors in the IRLINK and ECAT communication links, and if the controller is paired with the main body
0xE10D	Version information mismatch, use software that matches the controller version	The version of the teach pendant or InoRobotLab does not match the version of the controller.	Use a teach pendant or InoRobotLab that matches the version of the controller.

16.3 Servo Debug

Communication links and ports

Standard EtherNet link, and EtherNet port on the controller



If communication between the servo debug software and the servo drive is lost for a long time, communication may be disconnected. In this case, re-establish the connection.

Module functions

Servo data supports all functions; DSP data supports only functions 1-4.

No.	Function	Description
1	Import and export of waveform files	Supports the export of waveform data and simultaneous export of multi-channel waveform data. Supports import of data file with suffix .inoparam and display of waveforms.
2	Continuous oscilloscope	Supports the continuous data acquisition function of the oscilloscope as well as the waveform display function.
3	Waveform analyzer	Supports functions such as waveform selection, dragging, scaling, adaptation, horizontal and vertical scale adjustment, FFT analysis, waveform comparison, coordinate value measurement, and cursor value measurement.
4	Trigger oscilloscope	Supports the setting of channel data acquisition conditions. Trigger condition: rising edge/falling edge/edge change/equal to Trigger level: Numerical setting Pre-trigger setting: (%) Number of conditions: 2 Conditional relation: single condition, two conditions (AND/OR)
5	Parameters and operating status monitoring	Supports the monitoring of commonly used servo parameters and operating status of robots (error, ready, run, no ready, EtherCAT state machine).
6	Servo debugging	Supports servo usability adjustment, tuning, fault management, I/O settings, servo parameter list, speed JOG, position JOG, homing, bus motor parameters, data monitoring and other functions.
7	Running status monitoring	Displays running status on the interface: servo alarm error, servo drive status (ready, run, no ready), servo EtherCAT state machine (1248).

No.	Function	Description
8	Log	Log file in xxx format.
9	Historical fault records	Reads and displays the last 10 fault records for each axis.

Description of the module function

- Continuous oscilloscope

The oscilloscope includes servo oscilloscope and DSP oscilloscope.

- Servo continuous oscilloscope

The range of data that a servo continuous oscilloscope can display is as follows.

No.	Data Name	Source
1	Bit definition	Servo
2	Position reference	Servo
3	Position feedback	Servo
4	Position following error	Servo
5	Speed reference	Servo
6	Speed feedback	Servo
7	Speed feedback filter	Servo
8	Current reference	Servo
9	Current feedback	Servo
10	Bus voltage	Servo
11	Phase U feedback current	Servo
12	Phase V feedback current	Servo
13	Phase W feedback current	Servo
14	d-axis feedback current	Servo
15	Q-axis feedback current instantaneous value	Servo
16	D-axis current instantaneous value	Servo
17	Position reference absolute value	Servo
20	Position feedback absolute value	Servo

No.	Data Name	Source
21	Auto-tuned resonance frequency	Servo
22	Auto-tuned inertia value	Servo
23	Torque reference (compensated)	Servo
24	Torque feedback (compensated)	Servo
25	Position following deviation-reference unit	Servo
27	Real-time target absolute position	Servo
28	Real-time target speed	Servo
29	Real-time target torque	Servo
30	Control word	Servo
31	Status word	Servo
32	Position reference unfiltered	Servo
33	Load rate of braking resistor	Servo
42	Main cyclic program execution time	Servo
43	Interrupt program execution time of torque loop	Servo
44	Interrupt program execution time of position loop	Servo
45	Scheduling time of main cyclic interrupt program	Servo
46	Torque interrupt scheduling time	Servo
47	Soft interrupt scheduling time of position loop	Servo
48	SYNC signal cycle detection value (us)	Servo
49	SYNC and IRQ phase position (us)	Servo

The indicators of the servo continuous oscilloscope are as follows.

No.	Indicator	Description
1	Sampling interval	1 ms to 100 ms

No.	Indicator	Description
2	Max. number of supported channels	8
3	Transfer data type	16-bit, 32-bit
4	Channel configuration	Each channel can be configured with any sampling data for any axis (repeatable sampling data configuration options for multiple channels). Supports channel settings: The visibility, scale, color, vertical scale, etc. for each channel can be set. The longitudinal movement of waveform is also supported.
5	Waveform acquisition start and stop	Start and stop buttons are provided. When the stop button is pressed, the waveform acquisition stops, and the waveform is displayed automatically.
6	Waveform display time range	The horizontal timeline can be set, with a range of 300 ms to 30000 ms.

- DSP continuous oscilloscope

The range of data that a DSP continuous oscilloscope can display is as follows.

No.	Data Name	Source
1	Running line number (float)	DSP
2	Running status	DSP
3	Common DI	DSP
4	Common DO	DSP
5	Torque feedback (per axis)	DSP
6	Planed angle (per axis)	DSP
7	Feedback angle (per axis)	DSP
8	Control word (planned by DSP) (per axis)	DSP
9	Status word (received by DSP) (per axis)	DSP
10	Internal planning status	DSP
11	Stop mode (per axis)	DSP
12	System DI	DSP
13	System DO	DSP
14	Emergency stop mode	DSP

No.	Data Name	Source
15	Positions in Cartesian space (6)	DSP
16	Speeds in Cartesian space (6)	DSP
17	DSP alarm code	DSP
18	Servo gain (x3 per axis)	DSP

The indicators of the DSP continuous oscilloscope are as follows.

No.	Indicator	Description
1	Sampling interval	1 ms to 100 ms
2	Max. number of supported channels	8
3	Transfer data type	16-bit, 32-bit

- Import and export of waveform files

Supports the export of waveform data and simultaneous export of multi-channel waveform data.

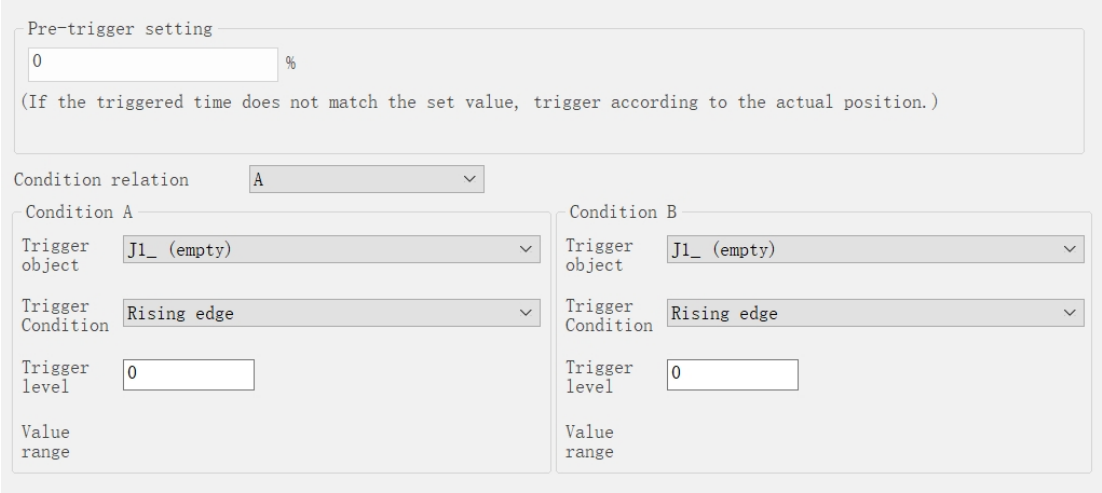
Supports import of data file with suffix .inoparam and display of waveforms.

- Waveform analyzer

Supports functions such as waveform selection, dragging, scaling, adaptation, horizontal and vertical scale adjustment, FFT analysis, waveform comparison, coordinate value measurement, and cursor value measurement.

- Trigger oscilloscope

The trigger setting supports single or two trigger conditions, as shown below.



The trigger condition can be set to condition A, condition B, both conditions A and B, or condition A or B.

  - Trigger object: Channel variables, DI bit variables, and DO bit variables.
  - Trigger condition: rising edge/falling edge/edge change/equal to
  - Trigger level: Numerical setting Pre-trigger setting: (%)

- Parameter monitoring

The parameter monitoring function is divided into servo parameter and DSP parameter monitoring, and their specifications are as follows.

- Servo parameter monitoring

Supports the monitoring of servo parameter data commonly used by robots. The data contents are as follows.

No.	Description	Parameter	Dictionary Object	Unit	Range	Length
1	Servo software version (J1 axis to J6 axis)	H0100	2001h-01h	1	0 to 65535	Uint16
2	Servo software non-standard version (J1 axis to J6 axis)	H0002	2000h-03h	1	0 to 65535	Uint32
3	Encoder software version (J1 axis to J6 axis)	H0004	2000h-05h	1	0 to 65535	Uint16
4	Motor model (J1 axis to J6 axis)	H0000	2000h-01h	1	0 to 65535	Uint16
5	Servo model (J1 axis to J6 axis)	H0102	2001h-03h	1	0 to 65535	Uint16
6	Absolute position mode (J1 axis to J6 axis)	H0201	2002-02h	1	0 to 2	Uint16
7	Actual motor speed (J1 axis to J6 axis)	H0B00	200Bh-01h	rpm	-9000 to +9000	int16
8	Average load rate (J1 axis to J6 axis)	H0B12	200Bh-0Dh	0.1%	-3000 to +3000	Uint16
9	Absolute position feedback (J1 axis to J6 axis)	H0B17	200Bh-12h	Encoder unit	-2^{31} to $+2^{31}$	int32

No.	Description	Parameter	Dictionary Object	Unit	Range	Length
10	Encoder multi-turn data (J1 axis to J6 axis)	H0B70	200Bh-47h	Turn	-32767 to +32767	Uint16
11	Encoder single-turn data (J1 axis to J6 axis)	H0B71	200Bh-48h	Encoder unit	-2^{31} to $+2^{31}$	int32
12	Servo fault code (J1 axis to J6 axis)	H0B34	200Bh-23h	1	065535	Uint16
13	Servo Sub-fault code (J1 axis to J6 axis)	H0B45	200Bh-2Eh	1	0 to 65535	Uint16
14	Encoder sub-fault code (J1 axis to J6 axis)	H0B28	200Bh-1Dh	1	0 to 65535	Uint16
15	Torque feedback (J1 axis to J6 axis)	-	6077h	0.1%	-5000 to +5000	int16
16	Target position (J1 axis to J6 axis)	-	607Ah	Reference unit	-2^{31} to $+2^{31}$	int32
17	Position feedback (J1 axis to J6 axis)	-	6064h	Encoder unit	2^{31} to $+2^{31}$	int32
18	Gear ratio (J1 axis to J6 axis)	-	6091-01h 6091-02h	1	0 to 2^{32} - 1	Uint32 Uint32
19	Position deviation (J1 axis to J6 axis)	-	60F4	Reference unit	-2^{31} to $+2^{31}$	int32
20	Excessive position deviation threshold (J1 axis to J6 axis)	-	6065	Reference unit	0 to 2^{32} - 1	Uint32

- DSP parameter monitoring

Supports the monitoring of commonly used DSP parameter data. The data contents are shown in the table below.

No.	Description	Parameter	Unit	Min.	Max.	Data Type	Change Method
1	Running line number	R0010	1	1	-	short	Read-only
2	Running status	R0020	1	0	1	short	Read-only
3	DSP alarm code	R0030	1	0x2000	0x6000	unsigned int	Read-only
4	Emergency stop mode	R0040	1	0	2	short	Read-only
5	DSP internal planning status	R0050	1	0	1	short	Read-only
6	J1 axis torque	R0101	0.10%	-5000	5000	float	Read-only
12	J1 axis planned angle	R0102	°	-250	250	float	Read-only
18	J1 axis actual angle	R0103	°	-250	250	float	Read-only
24	J1 axis stop mode	R0104	1	0	9	short	Read-only
30	J1 axis servo control word	R0105	1	0	65535	unsigned short	Read-only
36	J1 axis servo status word	R0106	1	0	0xFFFF	unsigned short	Read-only
42	J1 axis servo position loop gain	R0107	0.1 Hz	0	20000	unsigned short	Read-only
43	J1 axis servo speed loop gain	R0108	0.1 Hz	1	20000	unsigned short	Read-only

No.	Description	Parameter	Unit	Min.	Max.	Data Type	Change Method
44	J1 axis servo current loop gain	R0109	0.10%	0	2000	unsigned short	Read-only
7	J2 axis torque feedback	R0201	0.10%	-5000	5000	float	Read-only
13	J2 axis planned angle	R0202	°	-250	250	float	Read-only
19	J2 axis actual angle	R0203	°	-250	250	float	Read-only
25	J2 axis stop mode	R0204	1	0	9	short	Read-only
31	J2 axis servo control word	R0205	1	0	65535	unsigned short	Read-only
37	J2 axis servo status word	R0206	1	0	0xFFFF	unsigned short	Read-only
45	J2 axis servo position loop gain	R0207	0.1 Hz	0	20000	unsigned short	Read-only
46	J2 axis servo speed loop gain	R0208	0.1 Hz	1	20000	unsigned short	Read-only
47	J2 axis servo current loop gain	R0209	0.10%	0	2000	unsigned short	Read-only
8	J3 axis torque feedback	R0301	0.10%	-5000	5000	float	Read-only
14	J3 axis planned angle	R0302	°	-6000	360	float	Read-only
20	J3 axis actual angle	R0303	°	-6000	360	float	Read-only

No.	Description	Parameter	Unit	Min.	Max.	Data Type	Change Method
26	J3 axis stop mode	R0304	1	0	9	short	Read-only
32	J3 axis servo control word	R0305	1	0	65535	unsigned short	Read-only
38	J3 axis servo status word	R0306	1	0	0xFFFF	unsigned short	Read-only
48	J3 axis servo position loop gain	R0307	0.1 Hz	0	20000	unsigned short	Read-only
49	J3 axis servo speed loop gain	R0308	0.1 Hz	1	20000	unsigned short	Read-only
50	J3 axis servo current loop gain	R0309	0.10%	0	2000	unsigned short	Read-only
9	J4 axis torque feedback	R0401	0.10%	-5000	5000	float	Read-only
15	J4 axis planned angle	R0402	°	-540	540	float	Read-only
21	J4 axis actual angle	R0403	°	-540	540	float	Read-only
27	J4 axis stop mode	R0404	1	0	9	short	Read-only
33	J4 axis servo control word	R0405	1	0	65535	unsigned short	Read-only
39	J4 axis servo status word	R0406	1	0	0xFFFF	unsigned short	Read-only
51	J4 axis servo position loop gain	R0407	0.1 Hz	0	20000	unsigned short	Read-only

No.	Description	Parameter	Unit	Min.	Max.	Data Type	Change Method
52	J4 axis servo speed loop gain	R0408	0.1 Hz	1	20000	unsigned short	Read-only
53	J4 axis servo current loop gain	R0409	0.10%	0	2000	unsigned short	Read-only
10	J5 axis torque feedback	R0501	0.10%	-5000	5000	float	Read-only
16	J5 axis planned angle	R0502	°	-180	180	float	Read-only
22	J5 axis actual angle	R0503	°	-180	180	float	Read-only
28	J5 axis stop mode	R0504	1	0	9	short	Read-only
34	J5 axis servo control word	R0505	1	0	65535	unsigned short	Read-only
40	J5 axis servo status word	R0506	1	0	0xFFFF	unsigned short	Read-only
54	J5 axis servo position loop gain	R0507	0.1 Hz	0	20000	unsigned short	Read-only
55	J5 axis servo speed loop gain	R0508	0.1 Hz	1	20000	unsigned short	Read-only
56	J5 axis servo current loop gain	R0509	0.10%	0	2000	unsigned short	Read-only
11	J6 axis torque feedback	R0601	0.10%	-5000	5000	float	Read-only
17	J6 axis planned angle	R0602	°	-540	540	float	Read-only

No.	Description	Parameter	Unit	Min.	Max.	Data Type	Change Method
23	J6 axis actual angle	R0603	°	-540	540	float	Read-only
29	J6 axis stop mode	R0604	1	0	9	short	Read-only
35	J6 axis servo control word	R0605	1	0	65535	unsigned short	Read-only
41	J6 axis servo status word	R0606	1	0	0xFFFF	unsigned short	Read-only
57	J6 axis servo position loop gain	R0607	0.1 Hz	0	20000	unsigned short	Read-only
58	J6 axis servo speed loop gain	R0608	0.1 Hz	1	20000	unsigned short	Read-only
59	J6 axis servo current loop gain	R0609	0.10%	0	2000	unsigned short	Read-only
60	Cartesian spatial position x	R1001	mm	-4000	4000	float	Read-only
61	Cartesian spatial position y	R1002	mm	-4000	4000	float	Read-only
62	Cartesian spatial position z	R1003	mm	-4000	4000	float	Read-only
63	Cartesian space orientation Rz	R1101	°	-180	180	float	Read-only
64	Cartesian space orientation Ry	R1102	°	-180	180	float	Read-only

No.	Description	Parameter	Unit	Min.	Max.	Data Type	Change Method
65	Cartesian space orientation Rx	R1103	°	-180	180	float	Read-only
66	Cartesian spatial position speed Vx	R1201	mm/s	-4000	4000	float	Read-only
67	Cartesian spatial position speed Vy	R1202	mm/s	-4000	4000	float	Read-only
68	Cartesian spatial position speed Vz	R1203	mm/s	-4000	4000	float	Read-only
69	Cartesian spatial orientation speed Wx	R1301	°/s	-100	100	float	Read-only
70	Cartesian spatial orientation speed Wy	R1302	°/s	-100	100	float	Read-only
71	Cartesian spatial orientation speed Wz	R1303	°/s	-100	100	float	Read-only
72	First 32 bits of common DI	R2001	1	0	0xffffffff	unsigned int	Read-only
73	Last 32 bits of common DI	R2002	1	0	0xffffffff	unsigned int	Read-only
74	First 32 bits of common DO	R3001	1	0	0xffffffff	unsigned int	Read-only

No.	Description	Parameter	Unit	Min.	Max.	Data Type	Change Method
75	Last 32 bits of common DO	R3002	1	0	0xffffffff	unsigned int	Read-only
76	System DI	R2000	1	0	0xffff	unsigned short	Read-only
77	System DO	R3000	1	0	0xffff	unsigned short	Read-only

The DSP parameters are defined as follows:

There are 77 DSP parameters RXXXX, with R indicating robot, the first 8 digits of XXXX indicating item category, and the last 8 bits indicating function type.

The item category is divided into four major categories: miscellaneous, axis (J1-J6), Cartesian, and DI/DO.

Description:

- The miscellaneous (00) contains four function types such as DSP running line number (10), running state (20), etc.
- The axis (01-06) represents J1-J6 axes. The functions are subdivided into 9 function types such as torque feedback (01), planning angle (02), etc.
- Cartesian (10-13) and DI/DO (20, 30) are shown in the table.
- Servo debugging

Description:

Static attributes of the servo parameters and upload and download of current values. After the servo parameters are uploaded, the background color is highlighted (gray) when the current value is different from the default value.

When downloading the servo parameters (not yet downloaded), the background color is highlighted (purple) if the current input value is different from the default value.

Parameter list:

Conforms to the parameter list in the general servo background software.

- JOG

Description: The robot jogs according to the preset speed. Verify that the motor rotates correctly.

Specifications:

1. The servo is automatically disabled if no operation is performed. The protection time is 2s.
2. Speed range: 0 to the maximum motor speed.

- Bus motor parameters

Description:

- Static attributes of the motor parameters and upload and download of current values.
- After the servo parameters are uploaded, the background color is highlighted (gray) when the current value is different from the default value.

- When downloading the servo parameters (not yet downloaded), the background color is highlighted (purple) if the current input value is different from the default value.
- Mechanical characteristic analysis

Description:

Locates the mechanical resonance point by scanning the waveform.

Specifications:

1. The servo is automatically disabled if no operation is performed. The protection time is 2s.
2. Bode plot is output by background.

- Running status monitoring

The following running status is displayed:

- Servo error
- Servo drive status (ready, run, no ready)
- Servo EtherCAT state machine (1, 2, 4, 8)
- Log

The error log is saved to the local directory.

- Fault management

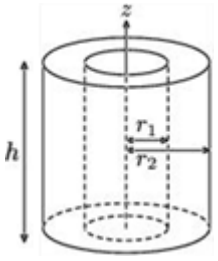
Description:

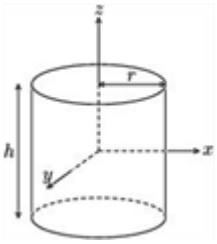
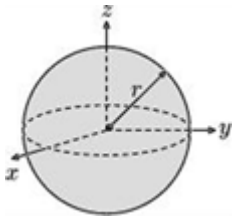
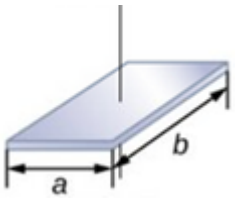
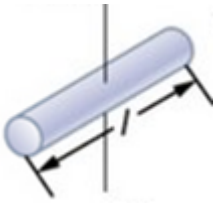
Reports current and historical equipment faults, gives all possible causes and solutions, and provides a reset function.

A total of 10 current and historical fault records of the servo, including the fault code, time stamp, fault name, the cause of the fault and the action to be taken.

16.4 Simple Calculation of Load Parameters

When the load is a simple geometry, or is close to a simple geometry, its inertia parameters can be calculated using a simple scheme. The formula for calculating the inertia of common geometry is as follows.

Description	Moment of inertia	Remarks	Illustration
Thick cylinder open at both ends, with inner diameter r_1, outer diameter r_2, height h, and mass m.	$I_z = \frac{1}{2}m(r_1^2 + r_2^2)$ $I_x = \frac{1}{12}m[3(r_1^2 + r_2^2) + h^2]$ $I_y = \frac{1}{12}m[3(r_1^2 + r_2^2) + h^2]$	-	

Description	Moment of inertia	Remarks	Illustration
Solid cylinder, with radius *r*, height *h*, mass *m*	$I_z = \frac{1}{2}mr^2$ $I_x = I_y = \frac{1}{12}m(3r^2 + h^2)$	The special case when the cylinder satisfies r1 = 0.	
Solid sphere, with radius *r*, and mass *m*.	$I = \frac{2}{5}mr^2$	-	
Solid cuboid, with height *h*, width *a*, length *b*, mass *m*	$I_h = \frac{1}{12}m(a^2 + b^2)$ $I_w = \frac{1}{12}m(h^2 + b^2)$ $I_d = \frac{1}{12}m(a^2 + h^2)$	Moment of inertia of a cube with side length *s* $I_{CM} = \frac{ms^2}{6}$	
Thin rod, with length *l*, and mass *m*.	$I_{center} = \frac{ml^2}{12}$	The special case when the solid cuboid has *w*= *L*, *a* = *d* = *0*.	

Service and Support

Should you encounter a safety accident during the use or operation of the product, or face challenges in operating and maintaining the equipment, which remain unresolved after the relevant documentation is consulted, we provide multiple channels to ensure prompt resolution:

- Channel #1: Contact service@inovance.com.
- Channel #2: Visit https://www.inovance.com/global to access document downloads, after-sales support, spare parts ordering, repair applications, and authenticity verification services.
- Channel #3: Download My Inovance app (https://zshc-eu.inovance.com/download-pc/) where you can access products info and documentation, and query product parameters.

We are committed to providing you with quick and professional technical support, and we look forward to your satisfaction and trust.

19012861A01

Copyright © Shenzhen Inovance Technology Co., Ltd.

Shenzhen Inovance Technology Co., Ltd.

www.inovance.com

Add.: Inovance Headquarters Tower, High-tech Industrial Park,
Guanlan Street, Longhua New District,
Shenzhen 518000, P.R. China

Tel: (0755) 2979 9595 Fax: (0755) 2961 9897

Suzhou Inovance Technology Co., Ltd.

www.inovance.com

Add.: No. 52, Tian E Dang Road, Wuzhong District, 215104,
Suzhou City, Jiangsu Province, P.R. China

Tel: (0512) 6637 6666 Fax: (0512) 6285 6720