JHY ROBOT

Control System User Manual

For other instructions, please refer to

- JHY Robot Body User Manual
- JHY Robot Control System Welding Instructions
- JHY Robot External Axis Control Instructions

Wuxi Jihoyen Industrial Automation Co., Ltd.

www.jhyrobotic.com

2023.05

Table of Contents

1 Terms and Definitions

The following terms and definitions apply to this document.

1.1 Teach Pendant

A handheld unit that can be used to program or move a robot and connect to a control system.

1.2 Key Indication

1. The keys on the handheld teach pendant are represented by [ ]: for example, the emergency stop key is represented by the [emergency stop] key.
2. The keys on the display are represented by [ ]: for example, the clear key is represented by the [clear] key.

2 Security

2.1 Safety Regulations

The requirements of robots and other mechanical equipment are usually different, such as large range of motion in space, fast operation, rapid movement of arms, etc., which will cause safety hazards.

Read and understand the instruction manual and related documents, and follow various rules and procedures to avoid personal, injury or equipment accidents. It is the user's responsibility to ensure that the robot system is always in a safe operating environment and complies with and abides by national and local laws, regulations and related regulations on safety.

Compliance with Regulations

The teaching and maintenance of robots must comply with the following regulations

1. Laws on industrial safety and health
2. Mandatory orders concerning industrial safety and health laws
3. Corresponding regulations on industrial safety and health laws
4. Safe operation of industrial robots (ISO10218).

Teaching and repairing robots is listed as a “hazardous operation” under industrial safety and

health laws.

Safety Management


Attention

1. Personnel who teach and maintain the robot must be trained in advance.

2. Carry out safety management according to specific policies in compliance with relevant regulations. Designate authorized operators and safety management personnel, and give further safety education.

2.2 Safety Precautions For Operators

The robot is potentially dangerous within the maximum range of motion.

All relevant personnel in the robotics industry (safety managers, installers, operators and maintenance personnel, etc.)

The idea of **Safety First** must be established at all times, and relevant operations must be carried out on the premise of ensuring the safety of all personnel.


Attention

1. Any dangerous operations are prohibited in the installation area of the robot.

2. Strictly abide by the following terms in the work area:

- Wear work clothes (do not wear loose clothes)
- Do not wear gloves when operating the robot.
- Underwear, shirts and ties should not be exposed from work clothes.
- Do not wear large jewelry such as earrings, rings or pendants.
- Wear appropriate safety equipment when necessary, such as hard hats, safety shoes (with non-slip soles), face shields, protective glasses and gloves.

3. Unauthorized personnel are not allowed to approach the robot and its peripheral auxiliary equipment.

4. During the operation of the equipment, the robot appears to have stopped, which may also be because the robot is waiting for the start signal to be in a state of imminent action. Even in such a state, the robot should be regarded as being in action. In order to ensure the safety of the operator, the display or sound of warning lights, etc. should be able to effectively inform (the operator) that the robot is in action.

5. Never force the shaft of the robot, otherwise it may cause personal injury or equipment

damage.

6. Never lean against the robot electrical control cabinet; do not press the operation button at will. Otherwise, the robot may cause unexpected actions, resulting in personal injury and equipment damage.

7. During operation, non-staff are never allowed to touch the robot electrical control cabinet. Otherwise, the robot may make unexpected movements, resulting in personal injury or equipment damage.

2.3 Safety Precautions During Operation

2.3.1 General Principles

Please follow the following principles for safe operation of the robotic system:

1. If there are staff in the protected space, please operate the robot system manually. When entering the protected space, be sure to carry a control device (teaching device) with you to control the robot at any time.

2. Pay attention to rotating or moving tools, such as cutting tools and saws. Make sure these tools stop moving before approaching the robot.

3. Pay attention to the high temperature surface of the work-piece and the robot system. The temperature of the robot motor is very high after long-term operation.

4. Pay attention to the fixture and make sure to clamp the work-piece. If the fixture is opened, the work-piece will fall off and cause personal injury or equipment damage. The fixture is very powerful and can also cause personal injury if not operated correctly.

5. Pay attention to hydraulic and pneumatic systems and live components. Even if the power is cut off, the residual power on these circuits is dangerous.

6. When installing a tool on the robot, be sure to CUT OFF the power supply on the electrical control cabinet and the installed tools, lock the power switch, and hang a warning sign. During installation and maintenance, for example, when the power is turned on, it may cause electric shock or cause the robot to move abnormally, resulting in injury.

7. Never exceed the allowable range of the robot (see the technical specifications section of the robot manual for the allowable range of the robot). Failure to do so may result in personal injury or equipment damage.

8. Whenever possible, the teaching work should be carried out outside the work area.

9. The following cautions shall be observed when performing teaching work within the robot motion range:

10. Always observe from the front of the robot;

- Always follow pre-established operating procedures
- Always have an idea of dodging in case the robot makes an unexpected move, making sure you have a way out in an emergency.

2.3.2 Emergency Treatment

Attention

Press the emergency stop button immediately when:

1. When the robot is operated, there are obstacles in the operating space of the robot that may collide, or there are other workers in the operating space;

2. Robots hurt people or machines.

2.3.3 Operating The Handheld Teach Pendant

2.3.3.1 Operating the Handheld Teach Pendant

1. The handheld teaching pendant is a high-quality terminal equipped with high-sensitivity advanced electronic equipment. To avoid failure or damage caused by improper operation, please follow the following instructions when operating.

2. Use the handheld teach pendant only for the purposes specified in this manual. The handheld teach pendant is developed, manufactured and tested in accordance with applicable safety standards. If you follow the safety and use instructions in this manual, the product will not cause personal injury or damage to machines and equipment under normal conditions.

2.3.3.2 Handling and Cleaning

1. Handle with care. Do not drop, throw or hit hard, as this may cause damage or malfunction.

2. In the event of an impact, it is important to verify that the safety functions (actuators and emergency stops) are working properly and are not damaged.

3. When the equipment is not in use, please place it on the vertical wall shelf to prevent accidental falling off.

4. Always use and store the handheld teach pendant in such a way that the cable does not trip

over the person.

5. Do not use sharp objects such as a screwdriver or nib to operate the touchscreen. This may damage the touch screen. Use your finger or stylus (depending on the model, some models do not have a stylus, usually on the back of the handheld pendant).

6. Clean the touch screen regularly. Dust and small particles may block the touch screen and cause malfunction.

7. Do not use solvents, detergents, or scrubbing sponges to clean the handheld teach pendant. Use a soft cloth with a small amount of water or a mild detergent.

8. Always cover the USB port when no USB device is connected. If the port is exposed to dust, it can break or malfunction.

2.3.3.3 Cables and Power Supplies

1. Before opening the cable entry area of the handheld tech pedant, please turn off the power. Otherwise, the component may be damaged or an unknown signal may appear.

2. Make sure that no one is caught by the cable so that the equipment does not fall to the ground.

3. Avoid extrusion of other objects to avoid damage to the cable.

4. Do not place the cable above the sharp edge, so as not to damage the outer skin of the cable.

2.3.4 Security Tools

2.3.4.1 Security Mechanism

Robot systems can be equipped with various safety protection devices, such as door interlock switches, safety light curtains and safety pads, etc. The most commonly used is the door interlock switch of the robot unit, which can be turned on to suspend the robot. When customers use it, they need to configure the corresponding switch signal in the robot control system, and correspond the I/O port connected by the hardware to the dedicated signal. Please refer to the I/O settings in this manual.

2.3.4.2 Safety Monitoring

Emergency stops and safety guards are monitored so the controller can detect any failures and the robot stops until the problem is resolved.

2.3.4.3 Built-in Safety Stop Function

The controller continuously monitors hardware and software functions. If any problems or errors are detected, the robot will stop, operating until the problem is resolved.

2.3.4.4 Limiting Robot Work Range

Robot operating range can be limited by mechanical stops, software functions, or a combination of both. Please refer to the documentation related to the functions of this manual for how to configure the robot system.

2.3.5 Safety Considerations For Manual Mode

2.3.5.1 Introduction to Manual Mode

In manual mode, the movement of the manipulator is under manual control. The three-position actuator must be pressed to start the motor of the manipulator, that is, to allow movement. Manual mode is suitable for creating and verifying programs and debugging the manipulator system.

2.3.5.2 Operating Speed

1. The movement of the manipulator in manual deceleration mode is limited to 250 mm/s (this can be set by setting whether the function is enabled or not).
2. When operating the manipulator in manual mode, the operator is very close to the manipulator. Manipulating industrial manipulators can be dangerous and should therefore be done in a controlled manner.
3. In manual mode, the manipulator moves at a preset speed, but must be under manual control. Manual mode should only be used when all personnel are outside the safe protected space, and operators must be specially trained to be aware of potential hazards.

2.3.5.3 Three-Position Actuator

1. In manual mode, the robot’s motor is activated by a three-position actuator on the handheld teaching pendant. In this way, the robot can only move by pressing the actuator.
2. This “stop-move” function is also applicable when stepping the debugging program in manual full speed mode.
3. The three-position actuator is uniquely designed, you must press the button halfway to start

the robot motor. If the button is not pressed or fully pressed, the robot will not move.

2.3.6 Safety Considerations For Automatic Mode

2.3.6.1 Introduction to Automatic Mode

1. Automatic mode is used to run robot programs in production.
2. In automatic mode, the safety function of the three-position actuating device is disabled so that the manipulator can move without human intervention.

2.3.6.2 Interference in Treatment Process

1. Process disturbances affect not only a specific manipulator unit, but the entire system chain, even if the problem is caused by a specific unit.
2. The chain of events may result in unknowable hazardous operations when operating a single manipulator unit, so special attention should be paid to this interference. Personnel performing all remedial actions must be familiar with the entire production line, not just the failed manipulator.

3 Install Wiring

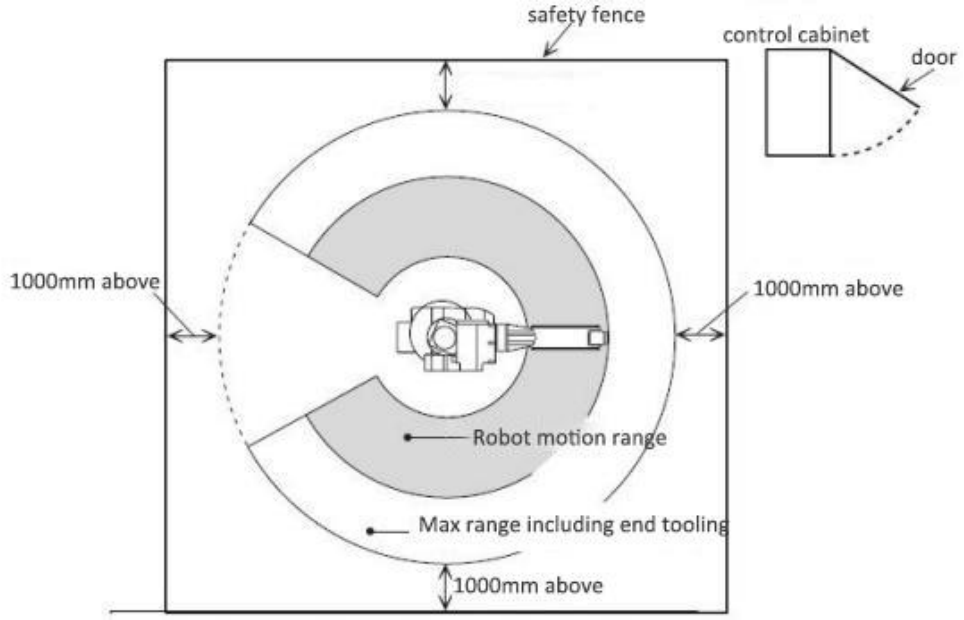
3.1 Installation Site and Environment

3.1.1 Installation Environment Requirements

The installation site before installing the electrical control cabinet must meet the following conditions:

1. The ambient temperature during operation should be between 0° and 45° C (32° to 113° F).
2. The handling and maintenance period should be -10° to 60° C (14° to 140° F).
3. The humidity must be lower than the dew point (relative humidity below 10%).
4. Places with less dust, dust, oil fume and water.
5. No flammable products and corrosive liquids and gases are allowed in the operation area.
6. The vibration or shock energy of the electrical control cabinet is small (the vibration is below 0.5G).
7. There should be no large electrical noise sources nearby (such as Tungsten Inert Gas (TIG) Welding, etc.).
8. There is no potential hazard of collision with mobile equipment such as forklifts.

3.1.2 Installation Location Requirements



1. The **electrical control cabinet** should be installed outside the robot's action range (outside the safety bar).

2. The **electrical control cabinet** should be installed in a position where the robot can be seen clearly.

3. The **electrical control cabinet** should be installed in a position that is convenient for opening the door for inspection.

4. Install the **electrical control cabinet** at least 500mm away from the wall to keep the maintenance channel open.

3.1.3 Precautions for Installation and Wiring

1. The system must be electrically grounded. If the equipment is not grounded, a fire alarm or electric shock will occur, resulting in personal injury.

2. Before the system is grounded, the power should be turned off and the main power switch should be locked. Otherwise, electric shock and personal injury may be caused.

3. Do not touch any substrate in the electrical control cabinet within 5 minutes after the power is cut off. Capacitors will store electrical energy after the power is cut off, so be careful whenever you operate the substrate. Failure to follow this warning may cause electric shock.

4. If the door is not closed, the power supply cannot be turned on, that is, the safety interlock device prevents the power supply from being turned on. Otherwise, a fire alarm or electric shock

may be caused.

5. During the wiring period, the electrical control cabinet is in emergency stop mode, and the user is responsible for anything that happens. Once the wiring is completed, an operation inspection is required. Otherwise, personal injury or mechanical failure may be caused.

Attention

6. Do not climb on top of the electrical control cabinet, otherwise it may cause personal injury or mechanical failure.

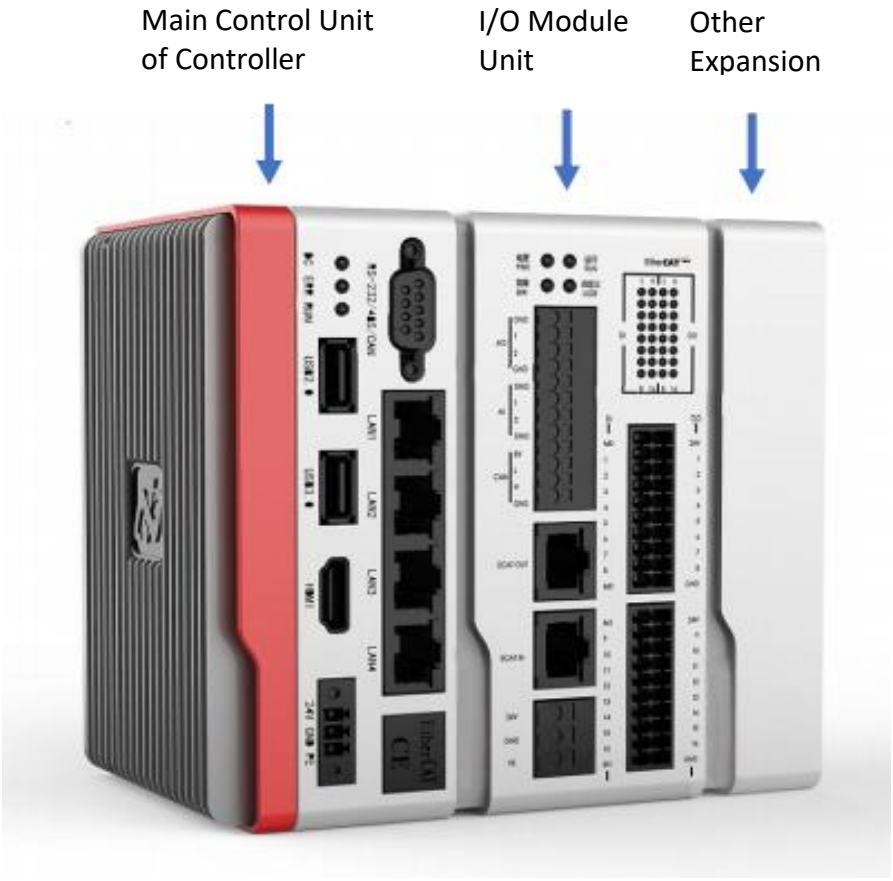
7. Wiring must be carried out by authorized staff. Incorrect wiring may cause fire or electric shock.

8. Please wire according to the rated capacity specified in the manual. Incorrect wiring may cause fire alarm or mechanical damage.

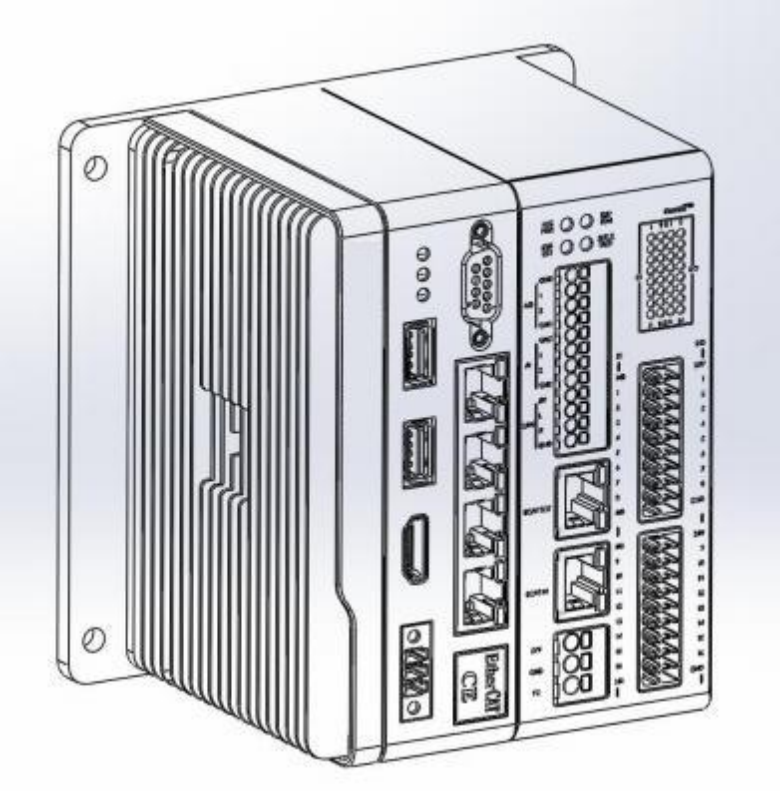
9. Confirm that the wiring of each circuit is safe and firm. If the circuit wiring is not firm, it will cause a fire alarm or electric shock.

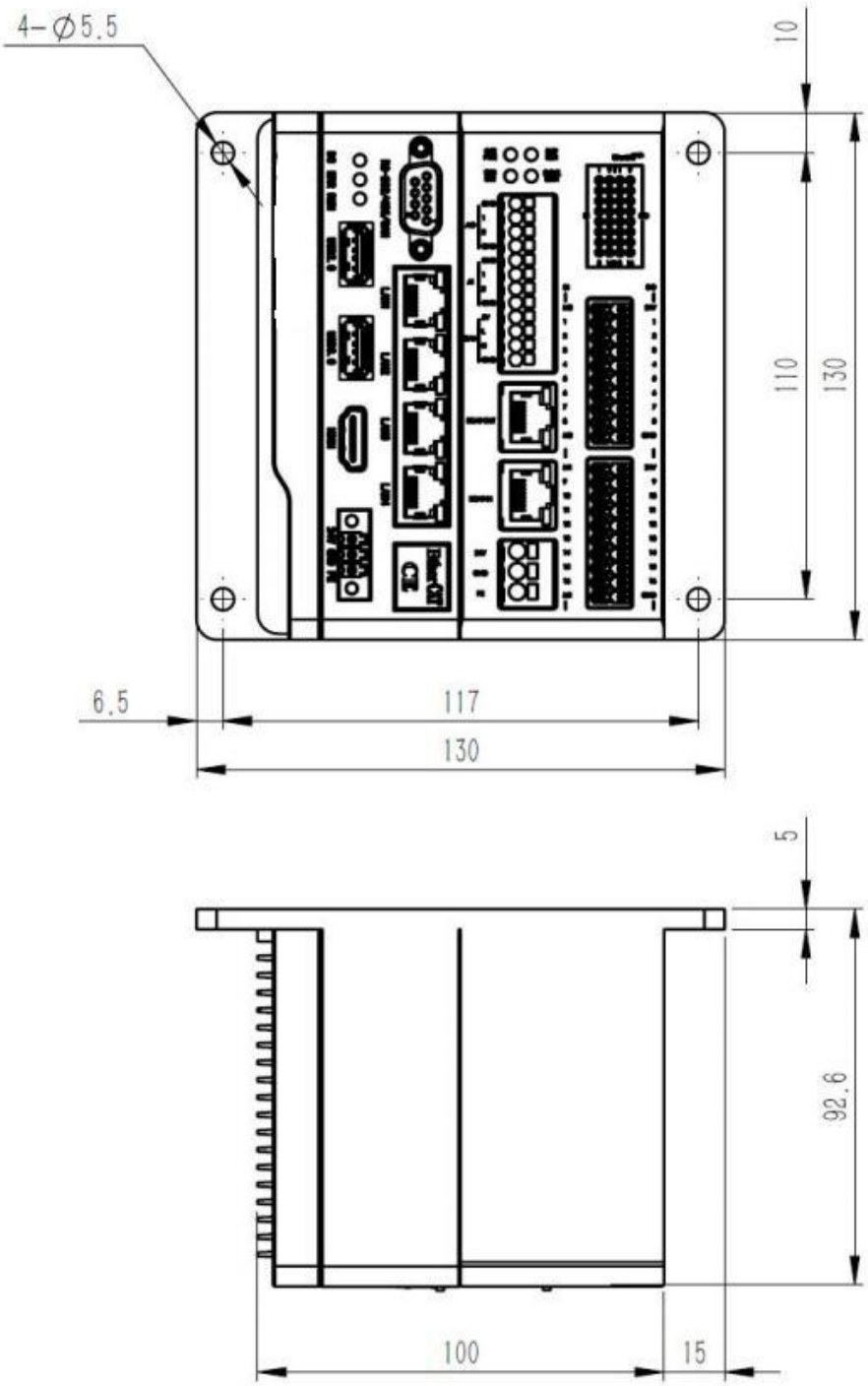
10. Do not touch the substrate directly with your hands. Integrated circuit (IC) substrates may fail due to static electricity.

3.2 Controller Installation Instructions



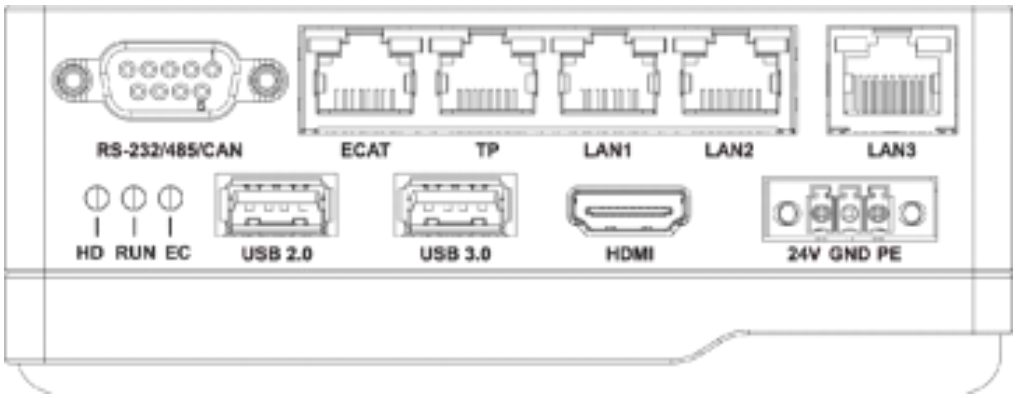
3.2.1 Installation Mode of Main Control and I/O Unit




4-Ø5.5
10
110
130
6.5
117
130
5
92.6
100
15

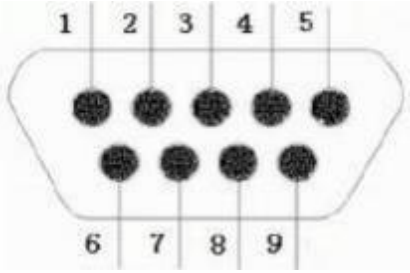
3.2.2 Main Control Unit Wiring Description

3.2.2.1 Interface Description of Main Control Unit



Interface Name	Description
+ 24V (must be connected)	Power supply to 24V
GND	DC power ground
PE	Connect reliability grounding and shielded terminals
TP (to teach pendant) (must be connected)	The other end is connected to the handheld box network port, the default IP: 192.168.100.3
Ecat (must be connected)	Connect with EtherCAT driver cable (standard, network cable interface)
LAN 1 (optional)	Default IP : 192. 168.101.3
LAN 2 (optional)	Default IP : 192. 168.102.3
LAN 3 (optional)	Default IP : 192. 168.103.3
RS 232/ CAN /485 combination port (optional)	For external communication
USB 2.0 / USB3.0 (optional)	External USB keyboard, mouse or U disk
HDMI (optional)	External HDMI interface display

3.2.2.2 RS-232/485/ CAN Interface Definition



1	VCCS_CAN	CAN power supply 24V
2	RS232_RX	Serial port 232 communication receiving
3	CAN_L	CAN low level (the new version of the controller cancels CAN and designs the CAN interface on the I/O module)
4	RS485-	Serial port 485 communication low level
5	GND	Ground
6	CAN_GND	CAN grounding (the new version of the controller cancels CAN and designs the CAN interface on the I/O module)
7	RS232_TX	Serial port 232 communication sending
8	CAN_H	CAN high level (the new version of the controller cancels CAN, and the CAN interface is designed on the I/O module)
9	RS485+	Serial port 485 communication high level

3.2.2.3 Controller Status Indicator Description

3.2.2.3.1 Status Indicator HD:

Displays the read and write status of the hard disk. If the status indicator light is on, it indicates that the hard disk write operation is in progress. The status indicator light is controlled by the hardware system of the controller and is not operated by the upper-level program.

3.2.2.3.2 Status Indicator RUN:

The status indicator RUN remains normally on after power on (EtherCAT driver is not loaded.) When the EtherCAT driver starts, the driver will control the status indicator. The status indicator RUN under EtherCAT drive operation displays the status as follows:

1. The driver is in an idle state (waiting for the application to request the master:) on for 1s + off for 1s, alternately, that is, slow flash.

2. Bus communication is disconnected: on 300ms + off 200ms + on 300ms + off 1200ms,

alternately, that is, fast flash twice.

3. The application requests the master station to enter the OP mode (PDO interactive communication:) on 100ms + off 100ms, alternately, that is, fast flash.

3.2.2.3.3 Status Indicator EC:

The status indicator EC remains normally on after being powered on. After the EtherCAT interface program is started, the status indicator will be automatically operated. In the automatic operation mode:

1. **System not registered**: on 200ms + off 200ms + on 800ms + off 800ms, alternately, that is, fast flash + slow flash.

2. **Network disconnection or abnormal network interface card status**: on 200ms + off 200ms + on 200ms + off 1400ms, alternately, that is, flash twice.

3. **The master request is busy (the master is scanning the slave):** on for 1s + off for 1s, alternately, that is, slow flash.

4. **Master Station Ready (PDO communication established)**: on 100ms + off 100ms, alternately, that is,fast flash.

5. **Other states (during the configuration of the master station)**: on 200ms + off 200ms + on 200ms + off 200ms + on 200ms + off 1000ms, alternately, that is, the application exits by flashing 3 times, after that, the status indicator ERR display is uncertain; it may be always on, or it may be often off.

3.2.3 Teaching Pendant Installation Wiring

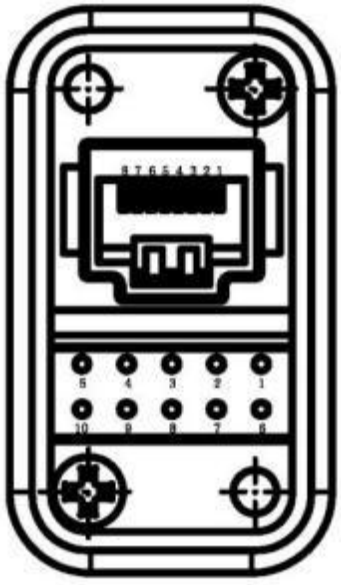


Figure 3 Teaching Pendant Aerial Insertion

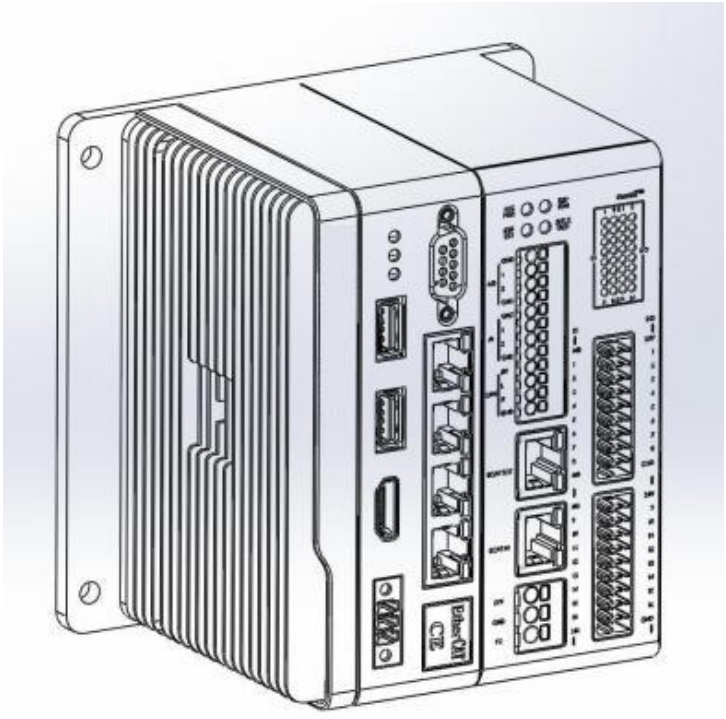
Definition of External Cable of Teach Pendant:

1	Red	24V+	Connect 24V power supply +
2	White & Red	0V	Connect 24V power supply 0V
3	Black	PE	Grounding
4	Green	EMG1-1	Emergency stop terminal 1
5	Black & Green	EMG1-2	Emergency stop terminal 1
6	Blue	EMG2-1	Emergency stop terminal 2
7	Black & Blue	EMG2-2	Emergency stop terminal 2

(The specific line number shall prevail)

The hand-held box must be connected with 24 V power supply wiring, and the emergency stop terminal can be selected and used according to the needs. The emergency stop terminal provides two sets of wiring output, and connects the emergency stop signal of the hand-held teaching pendant. It is normally closed by default. Press the emergency stop button of the hand-held teaching pendant to open it.

3.2.4 IO Module Wiring Definition



Ethercat dedicated to welded robots from the station I/O module				
IO type	**Num. of channels**	**Signal type**	**Signal level**	**Remark**

DI	16	compatible PNP/NPN	0V/24V,500mA	"1"signal level:15-24VDC；"0"signal level,0-5VDC
DO	16	NPN	0V,500mA	"1"signal level: 24VDC；"0"signal level,max 1.5VDC,10kΩ load
AI	2	voltage	0-10V	single grade,12BIT,two-wire system, input resistance 500kΩ
AO	2	voltage	0-10V	single grade,12BIT,two-wire system, min. load resistance 1kΩ,max. current 10mA
CAN	1	signal	5V	CAN-5V,CAN-L,CAN-H,CAN-GND

4 Hand-held Teach Pendant

This chapter describes the layout and key functions of the robot handheld teach pendant.

4.1 Key Function of Teach Pendant

No.	Name	Key	Function
1	Emergency Stop		**Press this button and the servo power will be cut off.** After cutting off the servo power supply, the [servo ready indicator] of the hand-held teach pedant is off, and the emergency stop information is displayed on the screen. After the alarm is removed, the emergency stop button can be opened, and the servo power can be continued after the emergency stop button is opened. After this button is pressed, the servo power will not be turned on. Method of opening the [emergency stop button]: rotate clockwise until the emergency stop button pops up. With the sound of “click”, it indicates that the emergency stop button has been opened.

2	Pause		**Press this button and the robot will stop running.** **This button can be used in any mode.** In teach mode: When this button is pressed, the light will be on. At this time, the robot cannot perform axis operation. In play mode: Press this button one time to enter the pause mode. At this time, the pause indicator lights up and the robot is in the pause state. Press the [Start] button on the hand-held teach pedant to make the robot continue to work.
3	Start		**Press this button and the robot starts play.** This indicator lights up when play mode is running. Start the robot by a special input start signal. This indicator lights up when play is running. This indicator lights up when the robot starts a play run from a dedicated input start signal. The mode knob must be set to play mode before pressing this button; Make sure the[servo ready indicator]is on.
4	Mode		**Teach mode,Play mode, and remote mode** TEACH: Teach mode Axis manipulation and editing can be done with a hand-held teach pedant (in this mode, working signals from external devices are invalid). PLAY: play mode You can run the program after teaching it. REMOTE: Remote mode The external TCP/IP protocol and IO can be used to start the teaching program.

5	Three-stage Switch		**Press this key, servo power on.** Before operation, the mode knob must be set in Teach mode → click the [Servo Ready] key on the hand-held teach pedant (the indicator light of [Servo Ready] is blinking) → gently hold the three-stage switchand the servo power is switched on (the indicator light of [Servo Ready] is steady on). At this time, if the grip, the servo power cut off. If you do not press the “Servo Ready” button on the hand-operated teach pedant, the servo power will not be switched on even if you gently hold the “three-step switch”.
6	Hand Pulse Knob		Rotate the scroll up and down to switch the drop-down box contents, project/program and program line. Press down to confirm the selection.
7	Serve Ready		**Press this key and the servo power will be switched on effectively.** When the servo power is cut off due to emergency stop or other reasons, use this key to effectively switch on the servo power. In play mode and remote mode, after pressing this key, the indicator light will be on and the servo power will be switched on. In teach mode, after pressing this key, the indicator light of [Servo Ready] flashes. At this time, press to the 1st stage of the [three-step switch] on the teach pedant. The indicator light of [Servo Ready] lights up, indicating that the servo power is connected.
8	Upshift		**Can be used with other keys.**

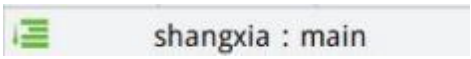
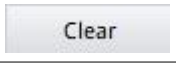
9	Menu		**Press this key to open the system application menu** Program: program management, file editing Variables: global variable, global location, global increment, project variable, project location, project increment Robot: coordinate system management, zero calibration, exception processing Settings: Permission management and system parameters Custom: Process
10	Chain		Assist key, used in conjunction with other keys. This key must be used in teach mode.
11	Fn1 Status Display		**Press this key to open the status display window.** View the robot's current position, digital volume IO, analog volume, multitasking, speed, variables, workspace and some process parameters.
12	Fn2 Floating Menu		**Press this key to open the floating menu** The floating menu can switch coordinate system, switch play mode, switch robot/external axis, switch speed and part of the process menu.
13	Axis Key		**Key to operate each axis of the robot.** **This key group must be used in Teach mode.** Multiple axes can be operated by holding down two or more keys. The robot runs according to the selected coordinate system and manual speed. Before operating the axis, please confirm whether the set coordinate system and speed are appropriate. Before operation, it is necessary to confirm that the [Servo Ready Indicator] on the hand-held operation teach pendant of the robot is on.
14	Forward and		When the servo power is turned on, the robot will run step by step (in reverse direction) according to the

	Backward		programming points trajectory once this key is pressed. This key must be used in Teach mode.
15	Speed+ or Speed-		The key to set the running speed of the robot in teaching mode. The low speed gears are 1%, 2%, 5%, and +/-5% each time the key is pressed. The set speed is displayed in the status bar. [Upper] + [V+] / [V-] can adjust the speed under play mode.

4.2 Teach Pendant Interface

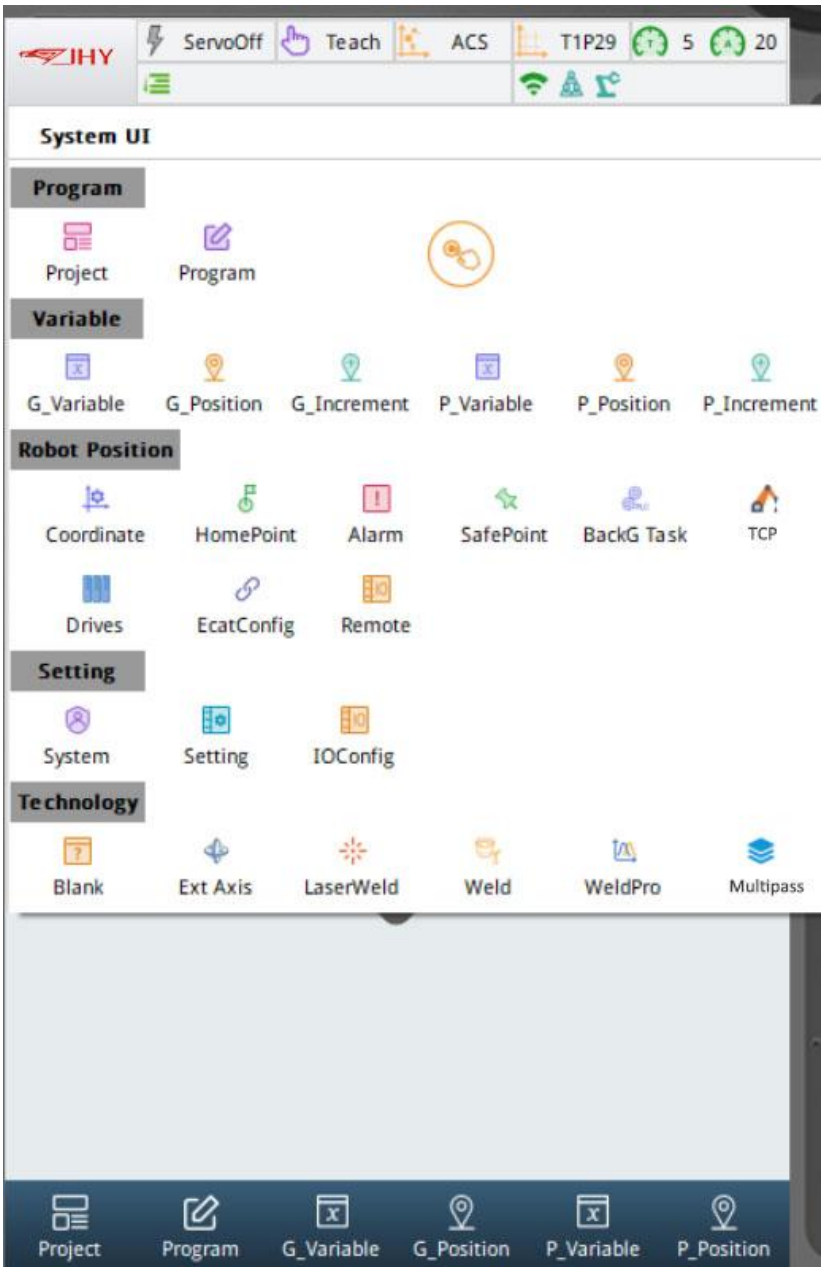
Figure 7 Teach Pendant Interface

Menu, button and identification in the interface:

	Each menu and submenu will be displayed in the system application menu. Press the “Menu” key on the handheld teach pedant or click the LOGO button in the upper left corner of the interface to display the system application menu.
	Display current servo; Operation mode; Coordinate system status.
	Display the number of the current tool coordinate system or the piece coordinate system; Display speed, the former is the speed of Teach mode, the latter is the speed of Play mode
	Display execution cycle mode (cycle, single, single step), current project name and program name
	Teach pedant connection state; Current permissions; Current coordinate system; Other process states
	You can use this key to view log and historical alarm information.
	Error and operation prompt or alarm, no alarm display “Standby”, display movement speed
	Key for clearing alarm

	Common application toolbar (can be customized)

4.3 System Application Menu



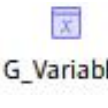
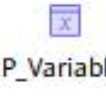
4.3.1 Common Toolbar Customization

The system menu bar has two operation modes. Click  or  to switch between them. In the drag mode, you can select common interface ICONS and drag them to the common toolbar for quick interface switching.

Icon	Function	Description

	Click Mode	In this mode, click other small ICONS on the menu bar to enter the corresponding interface
	Drag Mode	In this mode, you can drag a small menu bar icon to the toolbar of a common application Operation method: 1. Switch to drag mode 2. Hold down the icon and drag it to the toolbar of a common application to release the icon

4.3.2 Description of Menu Options

Icon	Name	Function
	Project Management	1. Create the new project 2. Project management: import, export, rename, save as, delete the selected project 3. New program: create a new program under the selected project 4. Program management: import, export, rename, save as, delete the selected program 5. Open the program
	Program Editing	Edit the selected program under the current project.
	Global Variable	Global numeric variables and global strings can be used in different projects for program editing.
	Global Position	The position variable can be calibrated, and can be used in different projects for program file editing.
	Global Increment	Can write positional increments, can be used in different projects for program file editing.
	Project Variable	Project value variables and project strings can only be edited and used in the currently opened project

 P_Position	Project Position	Position variables can be calibrated and can only be edited and used in the currently opened project
 P_Increment	Project Increment	You can write positional increments, which can only be edited and used in the currently open project
 Coordinate	Coordinate System Management	Calibrate and modify the world coordinate system, tool coordinate system and piece Coordinate System. Set the current coordinate system, and the corresponding serial number is displayed in the status bar at the upper right corner of the interface
 HomePoint	Zero Point Calibration	Calibrate the zero point of the robot, manually modify it, and clear the absolute encoder
 Alarm	Alarm	Axis Simulation Setting Limit switch of Axis motion
 SafePoint	HOME Coordinate	Set home position coordinates (return to zero)
 BackG Task	Backstage Task	Background task setting, reservation function
 System	System Management	Set administrator permissions. Different permissions have different operation contents. Version Information
 Setting	Setting	Settings: System Parameters, Hardware Parameters, Joint, External Axis, Cartesian, Planning Parameters, DH Parameters
 Blank	Undefined	Process parameter setting

4.4 Status Bar

4.4.1 Servo Display Status

	Servo off
	Servo on

4.4.2 Working Mode

Display the working mode of the robot, and switch it by the mode knob on the teaching pendant.

	Teach: The robot is in teach mode.
	Play: The robot is in play mode.
	Remote: The robot is in remote working mode.

4.4.3 Coordinate System

Displays the selected coordinate system, selected from {coordinate system} in the floating menu of the interface. The floating menu pops up using the [Fn2] key on the hand-held teach pedant.

	Joint coordinate system
	Robot coordinate system
	piece Coordinate System
	World coordinate system
	Tool coordinate system

4.4.4 Current Tool No. and Workpiece No.

In order to facilitate the user to determine the serial number of the tool (workpiece) currently in use, a tool (workpiece) coordinate system data queue with 32 elements is used inside the program. The default number 1 is that no tool (workpiece) is used, and 2--32 coordinate system queues element is an editable queue element.

4.4.5 Speed

Display the current speed by pressing the [V+] or [V-] key on the hand-held teach pendant. The former is the speed of teach mode and the latter is the speed of play mode.

4.4.6 Teaching File play Execution Mode

	Infinite cycle: the teaching program is executed in a non-stop cycle
	Single execution: only perform the teaching program once
	Single-line execution: only one line teaching program is executed at a time.

4.4.7 Current Permissions and Network Status

4.4.7.1 Permission Level

Change in the System Application Menu - Permission Management

 Operation Mode	 Administration Mode	 Reset

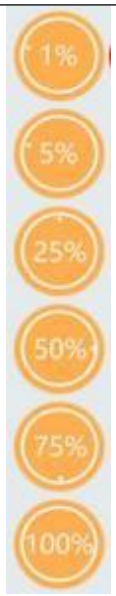
4.4.7.2 Network Status

 Connected	 Disconnected

4.5 Floating Menu

The floating menu pops/closes by using the [Fn2] key on the hand-held teach pedant with quick switching function.

Level 1	Level 2	Function
		1. Switch to joint coordinate system

		2. Switch to robot coordinate system
		3. Switch to piece Coordinate System
		4. Switch to world coordinate system
		5. Switch to tool coordinate system
	play program execution mode 	Infinite cycle
		Single execution
		Single-line execution
	jog axis switch 	Switch to robot axis (hold [J1] - [J8] represents robot axis) Switch to external axis (hold [J1] - [J8] for external axis)
	Speed percentage switching 	switch the speed percentage of the current mode

4.6 Robot Status and Alarm History Viewing

4.6.1 Robot Status

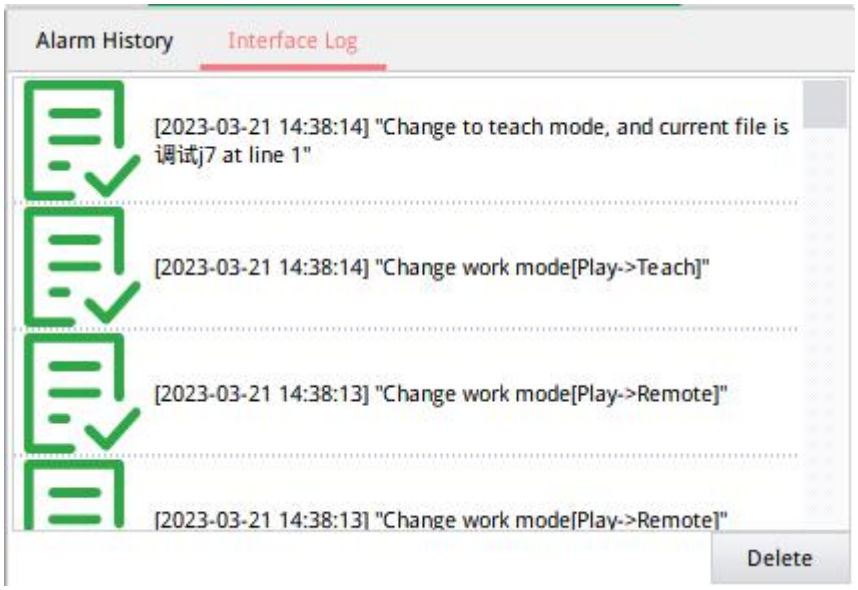
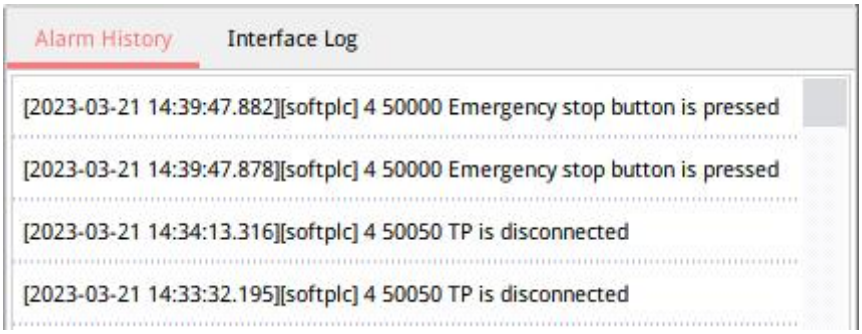
When there is an error message or alarm, the middle display area will turn yellow or red and display the error alarm information. Press the {clear} key on the right to clear the error.

4.6.2 Robot Alarm History and Action Log View

Status	Description
Normal	 When the robot is in normal operation state, it can operate the servo motion on the robot and execute the teaching program.
Servo Ready	 The robot is in servo-powered state
In execution	 The robot is moving or performing an instruction program, and the following number indicates the current linear velocity at the end of the robot
Yellow Alarm	 The robot is in the alarm state, which will not lead to servo down or emergency stop of the robot. The operator needs to check according to the alarm content and rule out the alarm cause
Red Alarm	 The robot is in the serious alarm state, which will cause the robot to stop abruptly and get off the private server. Before the alarm is lifted, the robot cannot enter the Servo Ready state

You can view all alarm information records and interface log through the left button (as shown below).

Item	Description

Interface Log	Record history interface action and alarm 
Alarm History	Record system alarm history 

5 Get Started Quickly

5.1 Servo Power On

When opening the main power switch on the electrical control cabinet, make sure that there is no person within the robot action range. Ignoring this warning may result in personal injury due to accidental contact with the robot. If any problem occurs, immediately press the emergency stop button, which is located on the upper left of the front door of the electrical control cabinet.

5.1.1 Switch on the Main Power Supply

Switch the main power knob on the electrical control cabinet to the ON position,the power indicator displays as green,now the main power is turned on.

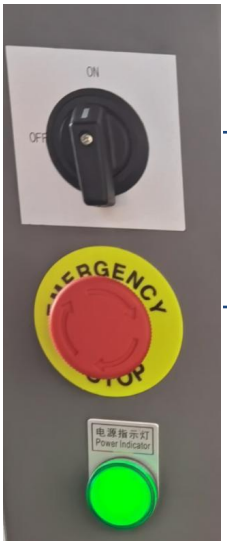


Figure 8 electrical control cabinet Panel

5.1.2 Switch on the Servo Power Supply

The servo power-on procedure is different under the teach mode, the play mode, and the remote mode.

1. In teach mode:

Press the [Servo Ready] button on the hand-held teach pendant, and the [Servo Ready] indicator is flashing; press the 1st stage of the [Three- stage Switch] on the back of the hand-held teach pendant, and the [Servo Ready] indicator on the hand-held teach pendant is always on, indicating that the servo power is on.

2. In play and remote modes:

Release the [Servo Ready] button on the hand-held teach pendant, and the [Servo Ready] indicator light on the hand held teach pendant is always on, indicating that the servo power is on.

5.2 Servo Power Cut Off

5.2.1 Cut Off the Servo Power Supply

The servo power cutoff procedure is different under the teach mode, the play mode, and the remote mode.

1. In teach mode:

Release or press to the 2rd stage [Three-Stage Switch] on the back of the handheld teach pendant, and the [Servo Ready] indicator on the handheld teach pendant goes out, indicating that the servo power supply is cut off.

2. play and remote mode

Press the [Servo Ready] button on the handheld teach pendant again, and the [Servo Ready] indicator on the handheld teach pendant goes out, indicating that the servo power supply is cut off.

3. Press the emergency stop button on the panel of the electrical control cabinet

Once the servo power supply is cut off, the braking device is activated and the robot is braked and cannot perform any further operation. Emergency stop can be entered at any time in any mode (teach mode, play mode or remote mode).

5.2.2 Cut Off the Main Power Supply

First cutting off the servo power then cut off the main power.

Switch the main power knob on the electrical control cabinet to the OFF position to cut off the main power supply, as shown in the figure below:

Figure 9 Rotary Switch

5.3 Selection of Coordinate System

In teach mode, the robot motion coordinate system is selected from the floating menu.

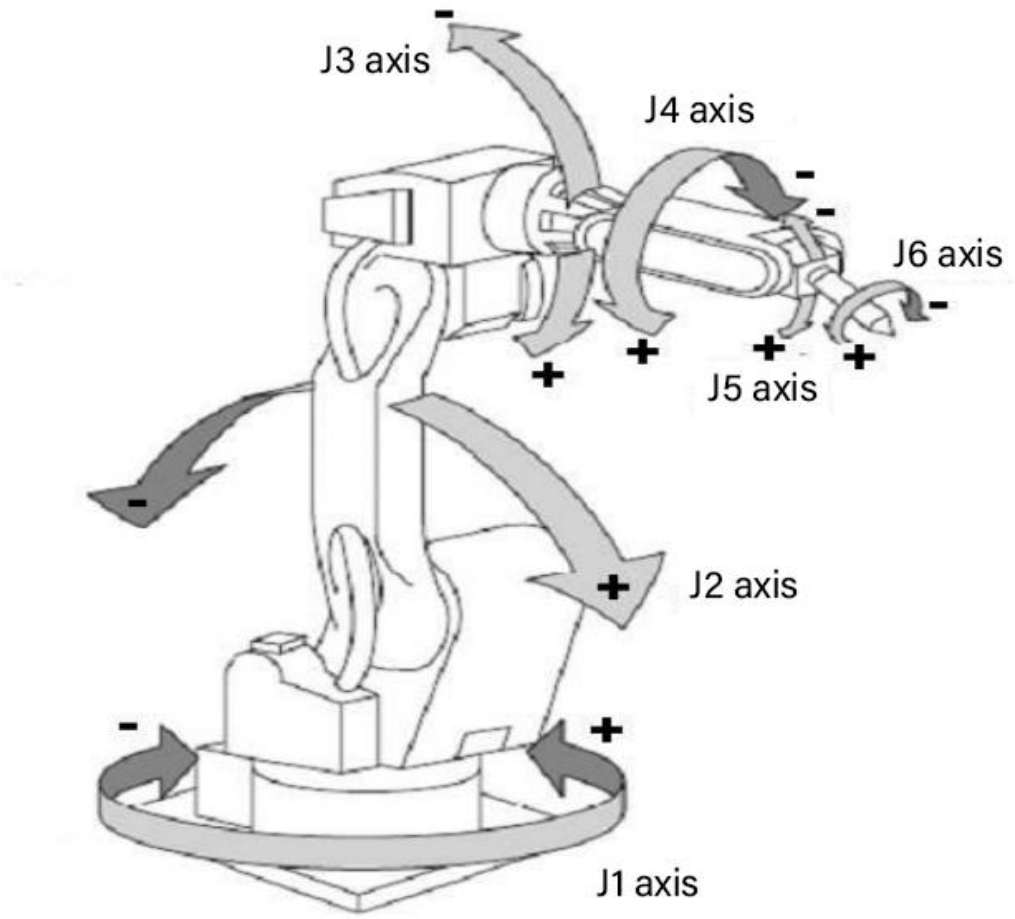
5.4 Manual Speed Adjustment

In Teach mode, modify the motion speed of the JOG robot:

Press the [V+] key or [V-] key on the hand-held teach pedant. Each time, the manual speed +/-5,will be confirmed by the speed display in the upper right corner of the status area. (The first value is the teaching speed)

5.5 Axis Operation

In teach mode, each axis of the robot can be moved to the desired position by pressing the **axis keys**, and the motion of each axis changes according to the selected coordinate system.



6r robot rotation direction（overall）

Each axis can only be moved when the **axis key** is pressed.

Danger

1. Before operating the robot, please confirm that the emergency stop button can work normally: Press the emergency stop button on the electrical control cabinet, the servo indicator light goes out and alarm appears on the teach pendant, indicating that the emergency stop key is normal. If the robot cannot be stopped in an emergency, it may cause mechanical damage.

2. The following cautions shall be observed when performing teaching work within the robot motion range:

- Always observe from the front of the robot
- Always operate according to the pre-established operating procedures
- Always have an idea of avoiding the robot in case of unexpected actions
- Make sure you have a fallback in case of an emergency

Improper and careless handling of the robot can cause injury.

3. Before performing the following operations, make sure that there are no people in the robot's range of motion and that you are in a safe position area.

- Switch on the power supply of the electrical control cabinet.
- When operating the robot with a hand-held teach pendant.
- On play motion
- When remote operation

If the robot collides with any person who enters the motion range, will result in personal injury.

4. Perform the following inspection steps before teaching the robot, correct any problems found, and verify that all other necessary work has been completed.

- Check whether the motion of the robot is abnormal;
- Check if the insulation and covering of the external cables damage.

After the servo power is turned on (press the [Servo Ready] key, hold the [Three-Stage Switch], and the servo indicator light is always on), press the operation key of each axis on the handheld teach pendant to make each axis of the robot generate the required action. The above illustration shows how each axis behaves in the axis coordinate system.

Please pay attention to the joint movement speed status before operating the robot, and adjust it to the appropriate speed by pressing the high/low speed key.

5.6 Flush

The flush function can only be used in teach mode.

The alignment function is to automatically align the rotation component of the rotation around a certain axis to an integer multiple of 90 degrees, such as -180°, -90°, 0°, 90°, 180°, etc. The constraint is that the Euler angles must be within 15 degrees.

When the angle is less than 15 degrees, press the [Ctrl] key + [A+] key, and the rotation component around the X axis of the selected coordinate system is aligned to an integer multiple of 90°; [Ctrl] key + [B +] key The rotation component of the selected coordinate system around the Y axis is equal to the integer multiple of 90°; the [Ctrl] key + [C+] key is used to align the rotation component of the selected coordinate system around the Z axis to an integer multiple of 90°.

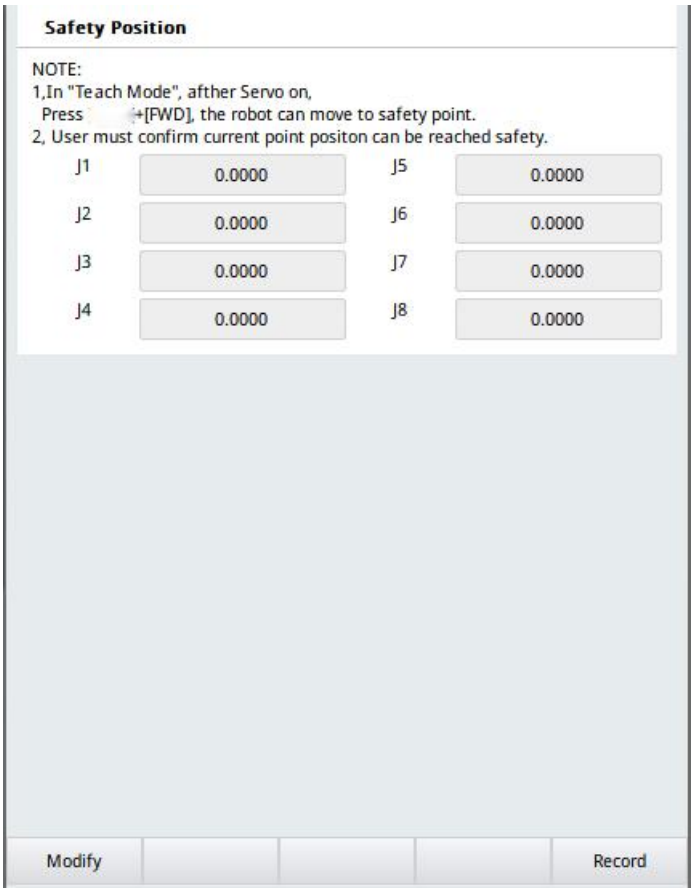
Finally, select the robot motion coordinate system through the floating menu to confirm whether

it has reached the position that is an integer multiple of 90°

5.7 Return to Zero position

In teach mode, set the original position coordinates,after the servo on, press the [Ctrl] key+ [Forward] key can move to this point.

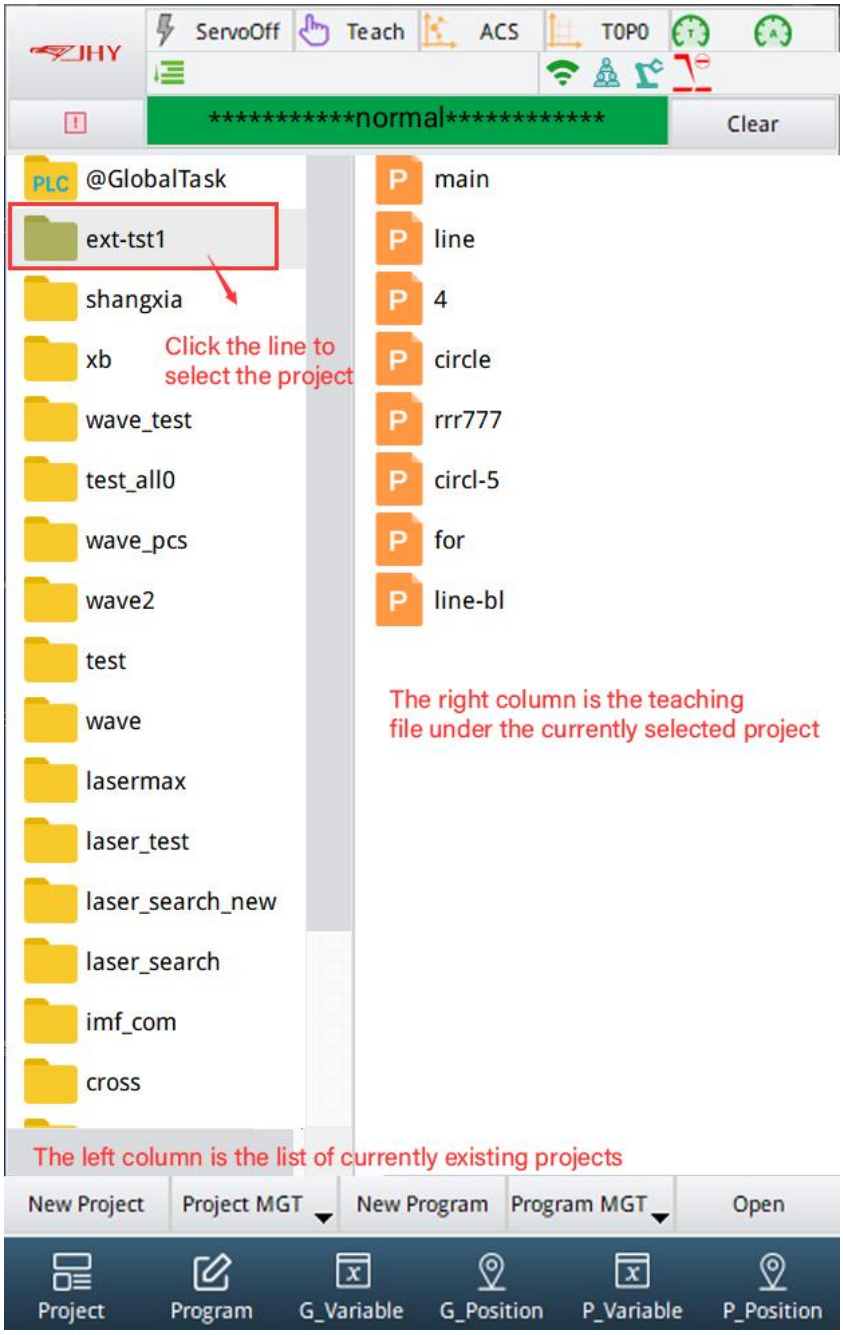
This point should be recorded as a position that the robot can safely reach from any position.



If the HOME point position is not set, it will default to the factory zero point position of the robot.

6 Engineering Management Interface

Select  from the main menu or click “Project Management” on the common toolbar to enter the project management interface.



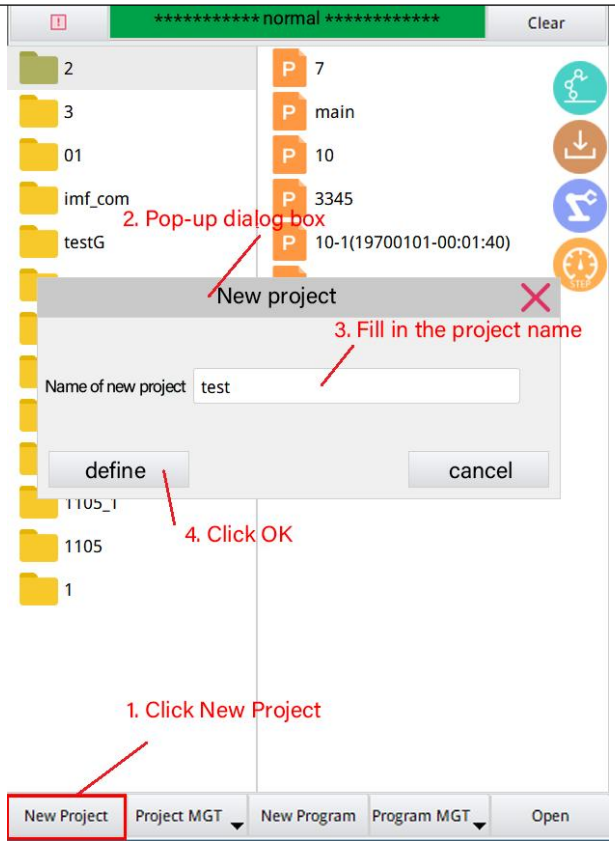
6.1 Description of Robot Engineering Structure

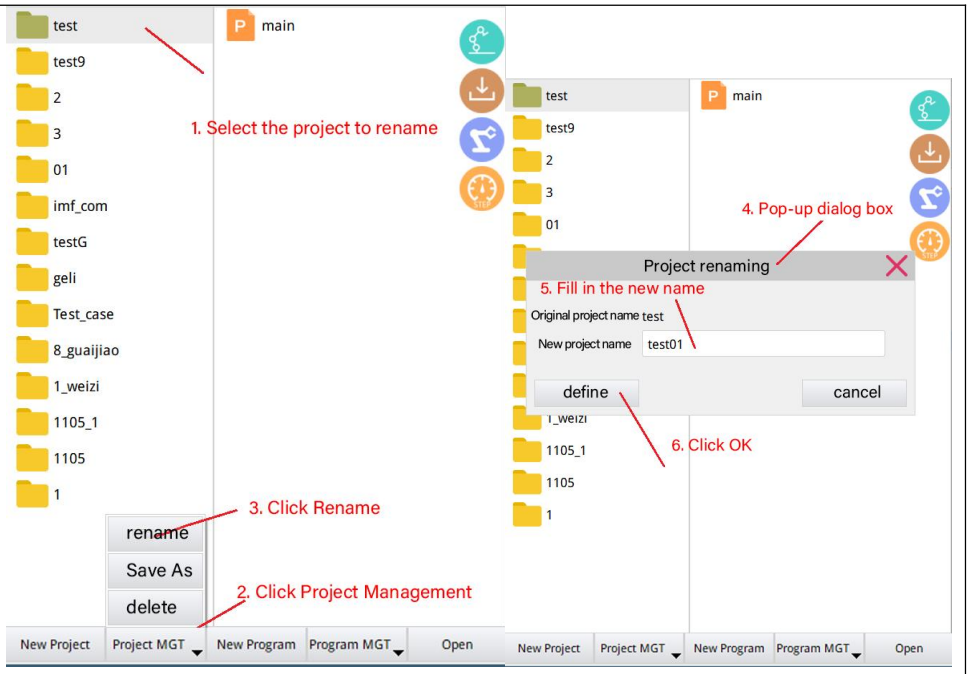
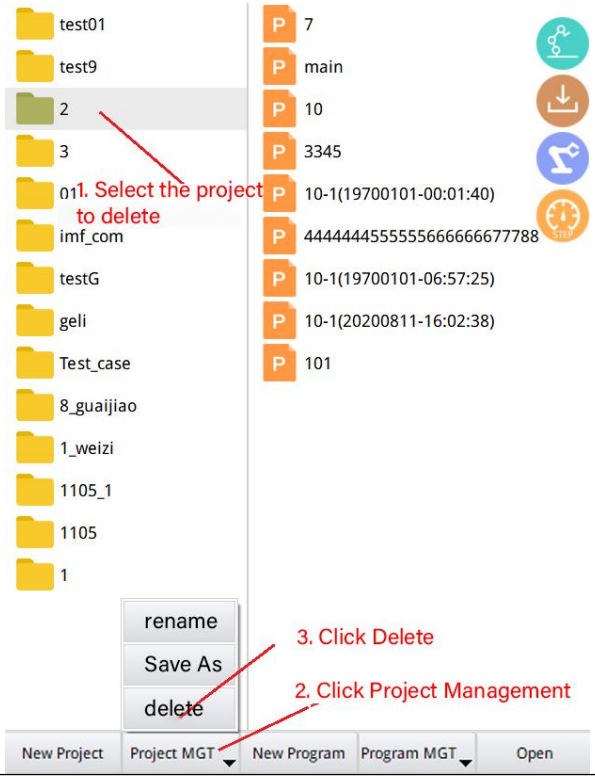
The robot program is based on the project, each project can have multiple teaching files, and each project contains variables that can only be used by the teaching files within the project. Take the project named “test” as an example. The following figure shows the structure of the teaching file main and other teaching files can be executed.

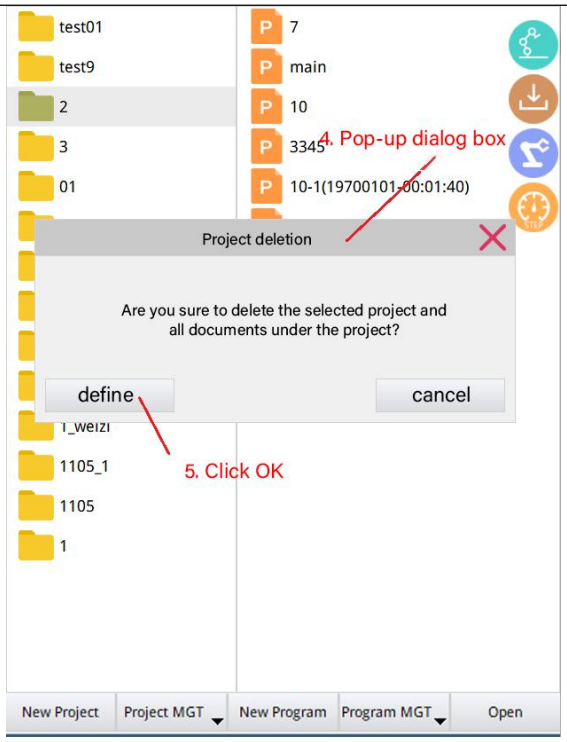
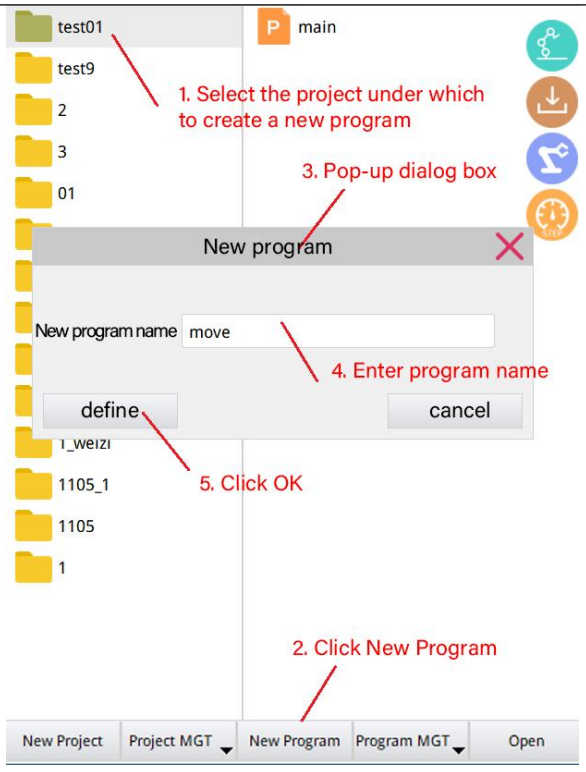
- The difference between the teaching file *main* and other teaching files: Other teaching files can be called in the teaching file main; other teaching files can call each other, but the teaching file *main* cannot be called.

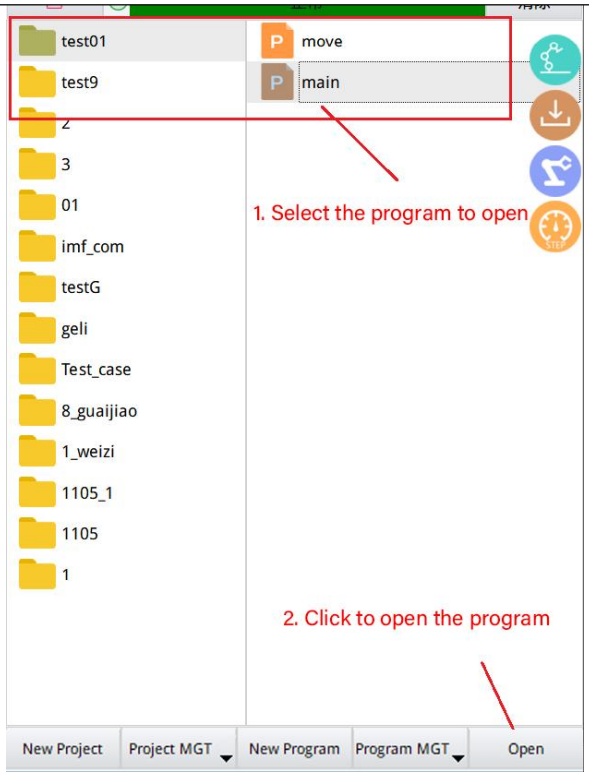
Engineering Test	*Teaching file main* Note: When creating a project, the project will automatically add a main program named *main*, this program cannot be deleted, renamed.
	Other teaching files Note: Users need to create their own. Subprogram name specification: Contains only alphanumeric underscores and cannot begin with a digit
	Engineering variable, engineering positional variable, engineering incremental positional variable Note: When creating a project, each project will automatically create these three types of variables. These variables can only be called by the teaching file under the project

6.2 Engineering Operations

Item	Description
1. New Project	

2. Project Rename	
3. Project Save As	[Save as] is to make a new copy of the entire project, and the operation steps refer to the project rename.
4. Project Delete	Deleting a project will delete all teaching files and variable files under the project. 

	
5. New Program	
6. Program Rename	To rename a program, first select the program that you want to operate, then select [Program Management], and click [Rename] other operations to view the **program rename**
7. Program Save As	First select the program you want to operate, then select [Program Management], click [Save As] other steps to view the **program rename**
8. Program	To delete a program, first select the program to be operated, then select

<table>
<tr><td>Delete</td><td>[Program Management], click [Delete] other operation steps to view program delete</td></tr>
<tr><td>Open the Program</td><td> 

Click Open to enter the program editing interface</td></tr>
</table>

7 Program Editing Interface

Click  on the menu bar or click “File Editing” on the common toolbar to enter the program editing interface

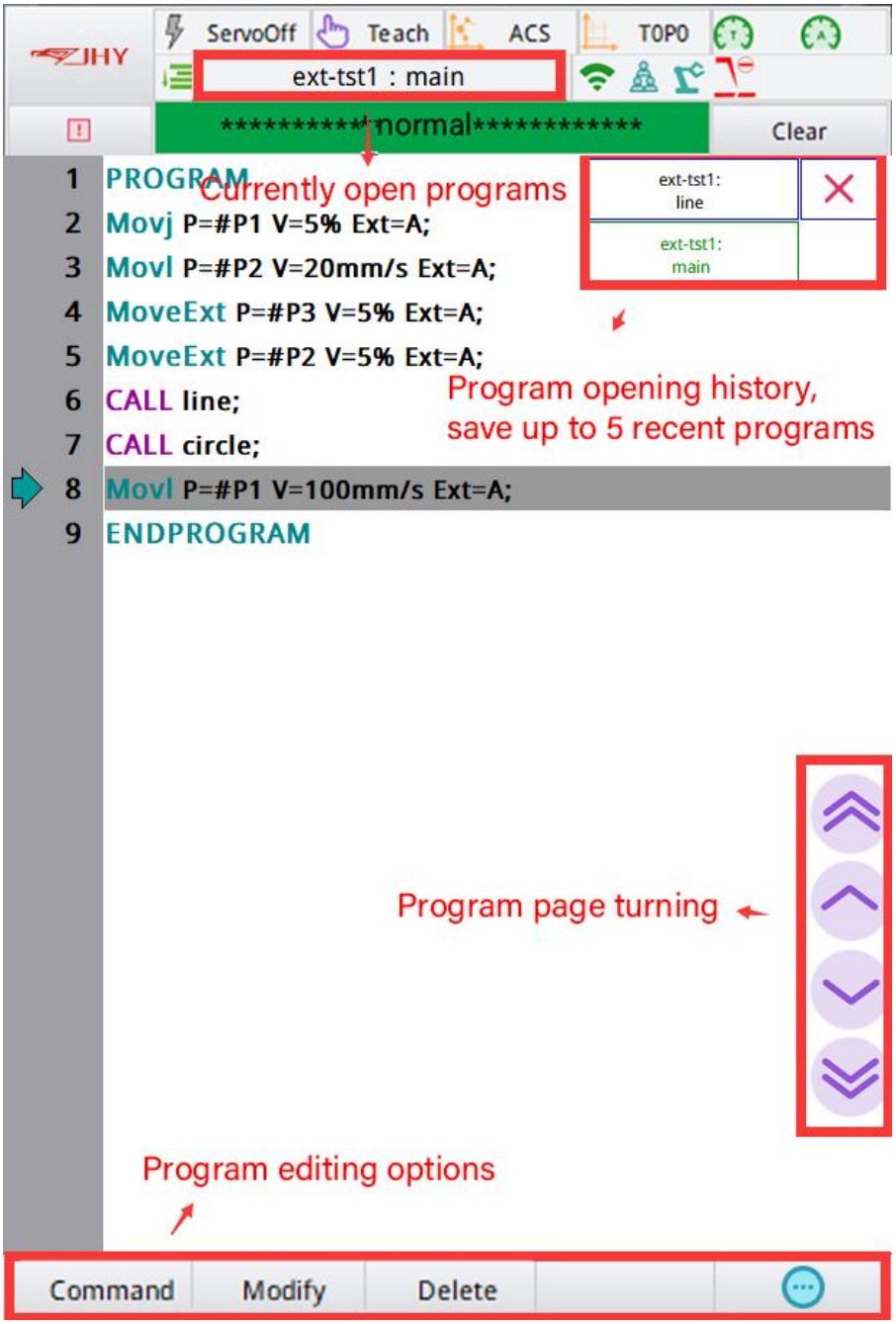


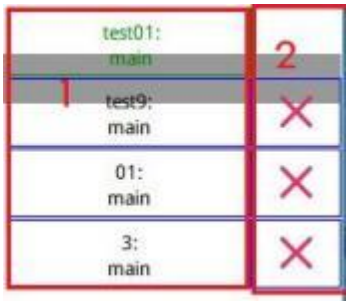
Figure 11 Program Editing Interface

7.1 Currently Open Programs

The name of the currently opened program is displayed in the status bar. For example, *test01: main, test01* is the project name, and main is the name of the teaching file in the project.

7.2 History Program

The interface will display the names of the five files that have been opened recently.



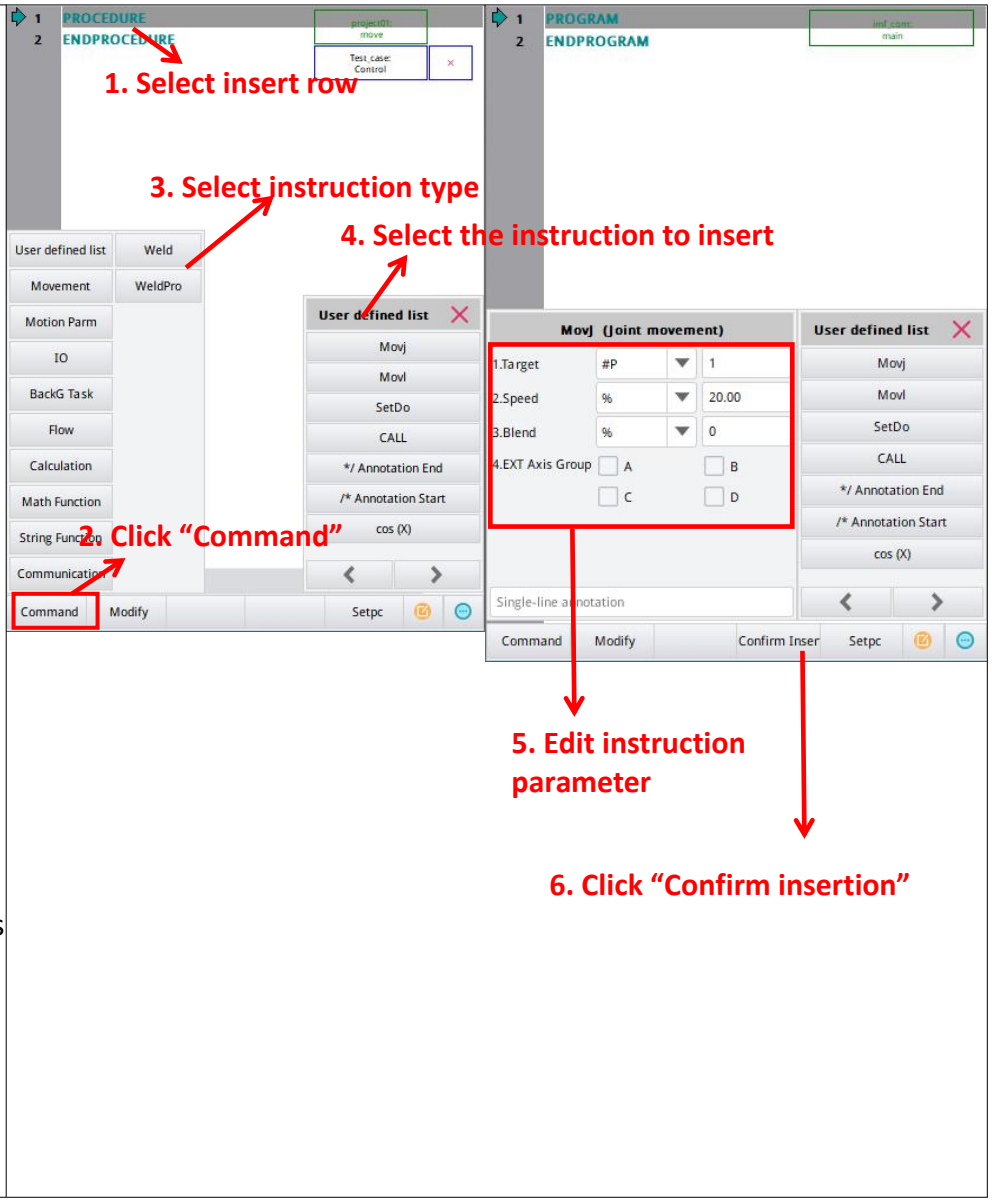
Item	Description
1. Quickly switch the history program	Click the corresponding row in the left column to quickly switch to the corresponding program. The green one is the current open program.
2. Delete history records	Click corresponding X logo to the two columns on the right to delete the program history of the corresponding row

7.3 Program Page Turning

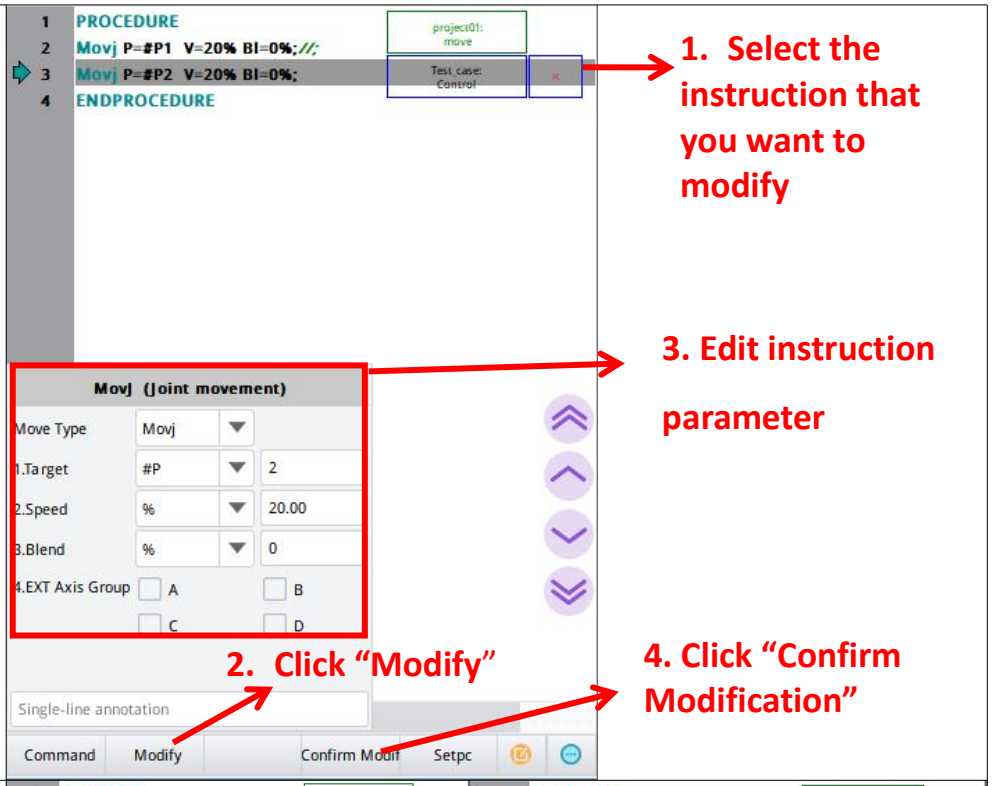
Icons	Description
	Get to the top
	Page up
	Page down
	End of program

7.4 Program Editing

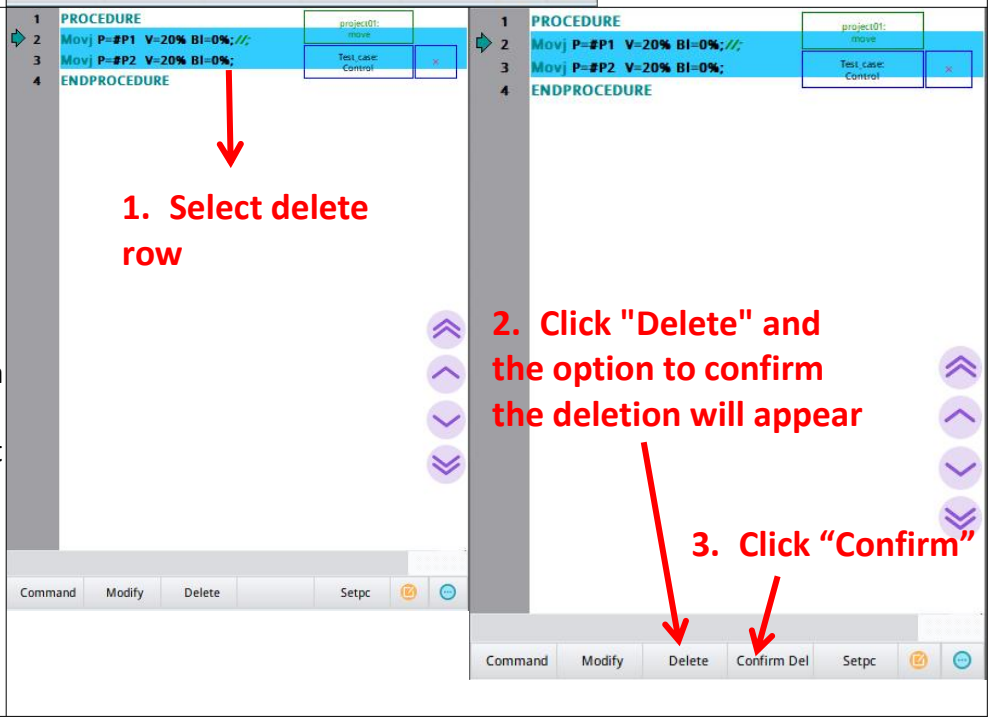
Function	Description

1. Teach instruction insertion (inserts a new instruction to the next line of the selected line)	

2. Instruction modification



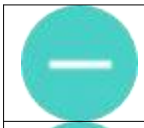
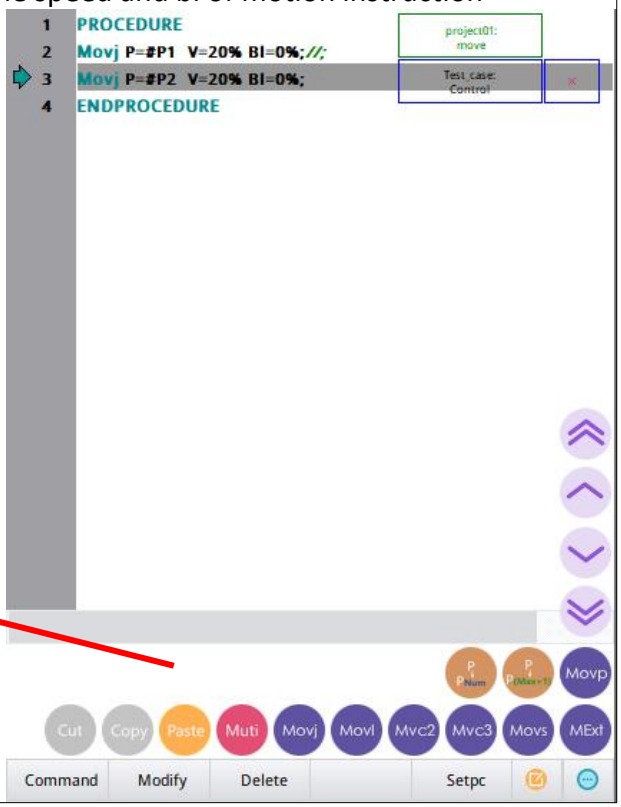
4. Delete instruction (when deleting a single line, click the line to select it)



7.5 Extension Bar

Click  to open the extended operation bar. The functions of each button in the extended bar are described as follows.

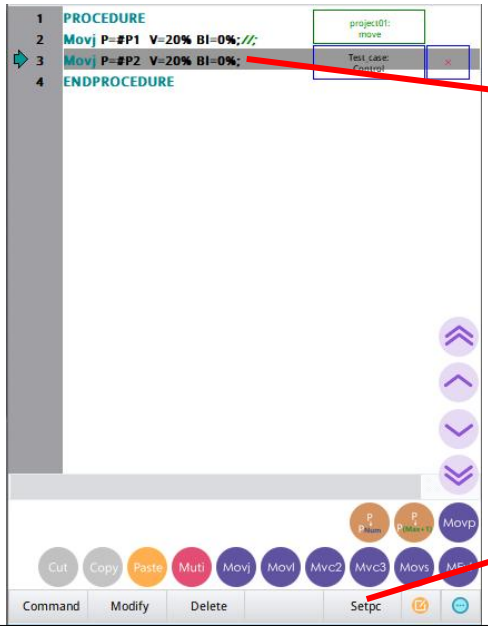
Icon	Description

	Program display font reduction
	Program display font enlargement
	Cut Selected Rows
	Copy Selected Rows
	Paste the program to the end of the selected line
	Batch modify, batch modify the speed and bl of motion instruction  1. Click the icon

2. Modify parameter

Batch modification allows you to modify the speed and bl of the motion instruction between the start and end lines

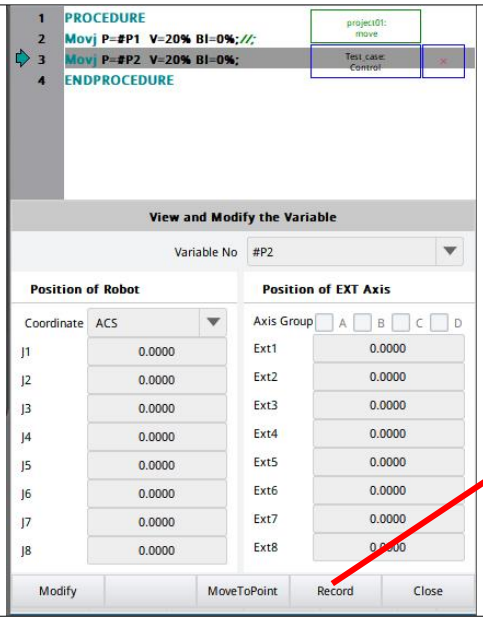
Variable View and Modify: This function is used to quickly view and edit all variables (including global variables and engineering variables) used in the currently selected line instruction.



1.Select view row

1. Click the icon

2.Click “Setpc”

		**Record the robot's current position to the position-type variable, and the coordinate system is the current JOG coordinate system**
	Quick Insert Movl: 1. In the servo on state, click this button to add a position point at the end of the engineering position variable. The position of this point is the current position of the robot, and the coordinate system is the current jog coordinate system. 2. At the same time, a Movl instruction is inserted in the currently selected line of the teaching file. The target location of the instruction is the engineering location variable recorded in 1	
	Quick insert Movj: The operation and steps are the same as above, and a Movj motion command will be inserted	
	Quick insertion Mvc2: The operation and steps are the same as above, and a Movc2 motion command is inserted, that is the middle point of the arc	
	Quick insertion Mvc3: The operation and steps are the same as above, and a Movc3 motion command will be inserted, that is the arc target point.	

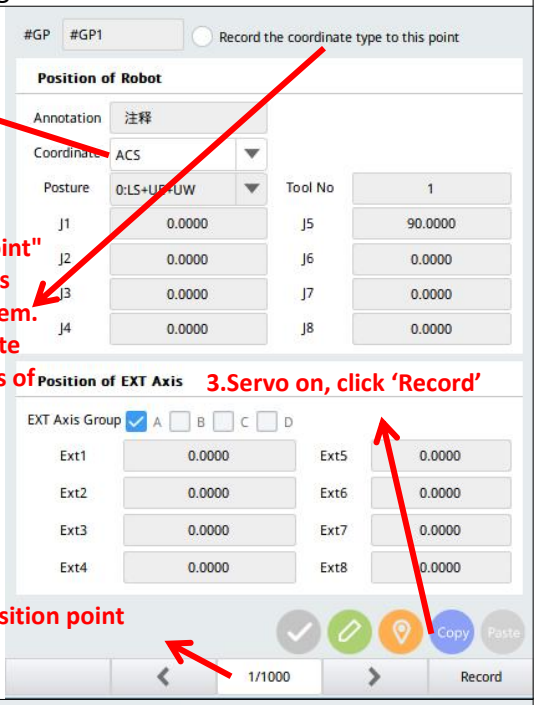
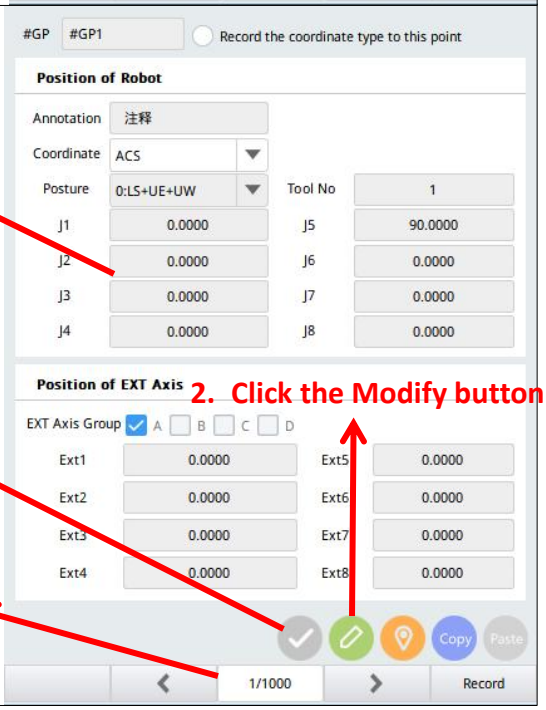
8 Variables

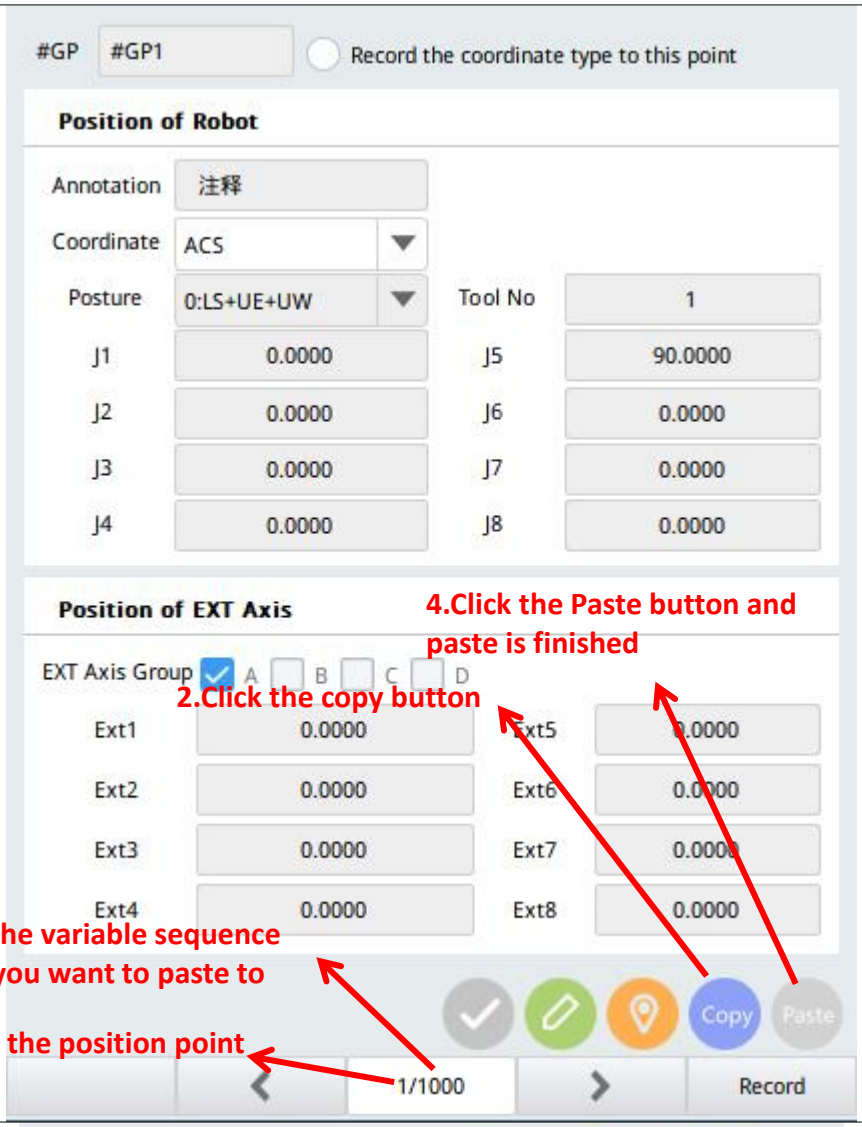
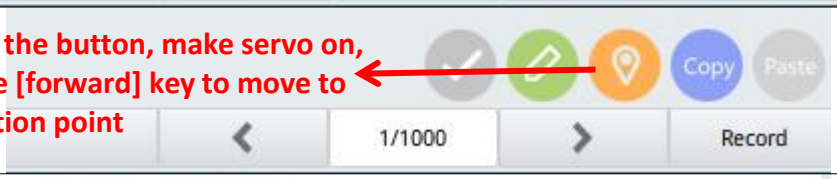
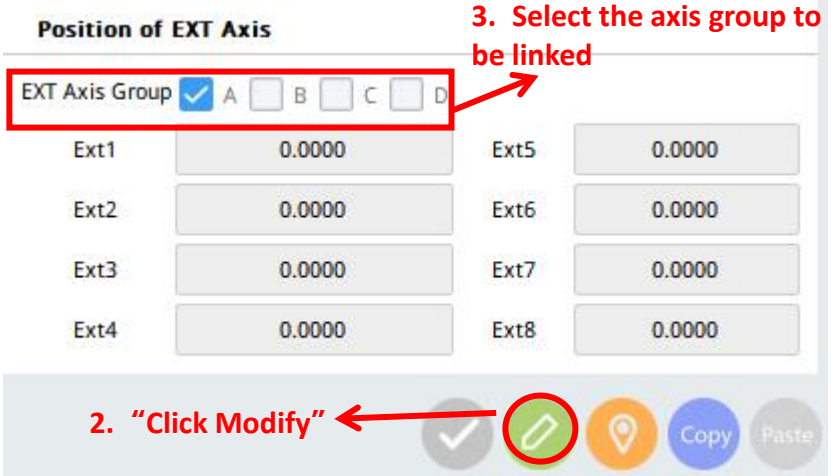
8.1 Global Variables

Global variables, including global position variables (#GP), global incremental positional variables (#GIncP), global numeric variables (#GVar), and global string variables (#GStr), are accessible to all teach files in the project.

8.1.1 Global Position Variables

The global position points are called in the teach file as #GPX (X indicates the sequence number 1-1000)

Item	Operating Instructions
1. Position points record	
2. Manually modify a location point	

3. Copy Paste	
4. Move to point	
5. Extended shaft set	 When the extended axis (external axis group) is selected, the jog operation starts and stops at the same time.

8.1.2 Global Incremental Location Points

The global increment position points are called in the teach file as #GIncPX (X indicates the sequence number 1-1000): Global incremental position points operate in the same way as global position points, except that they do not have the function of recording the current position and moving to a point.

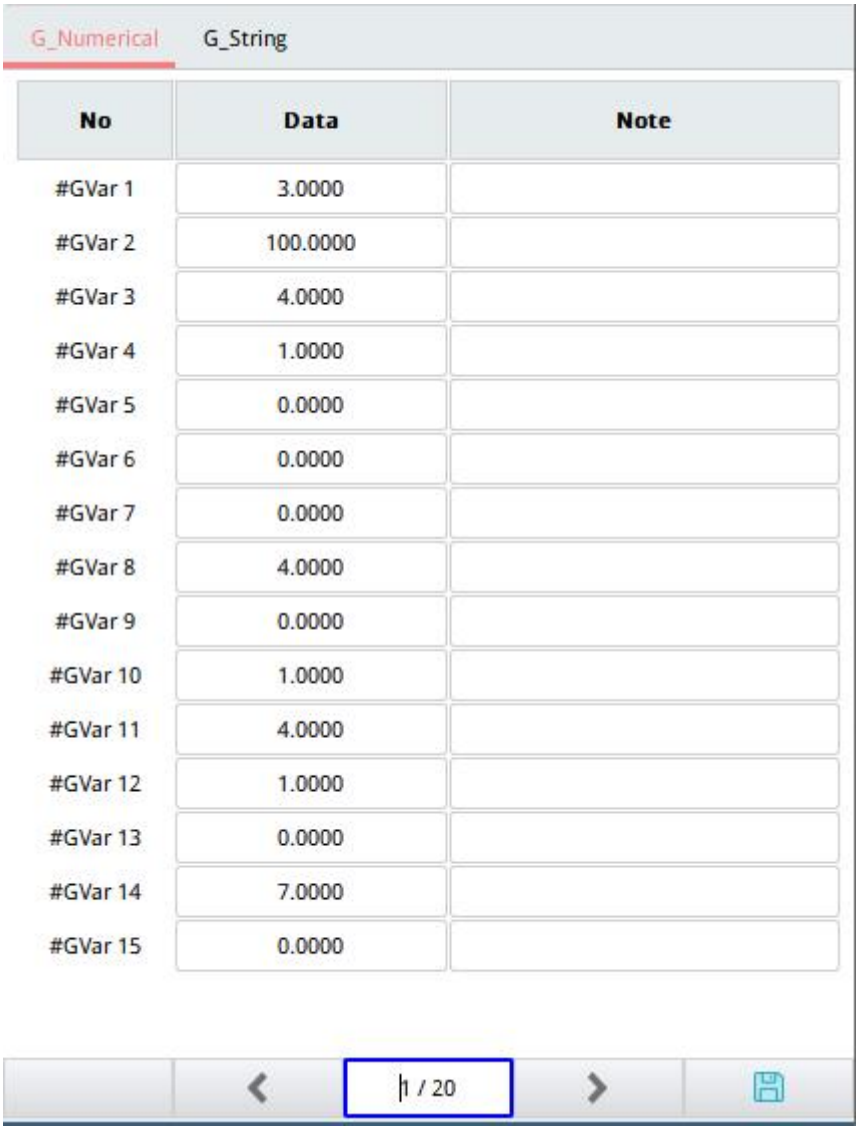
8.1.3 Global String Variable

The global string variable is called in the teach file as #GStrX (X indicates the sequence number 1-300):

Click  to save the modification after modifying the variable value.

8.1.4 Global Numeric Variables

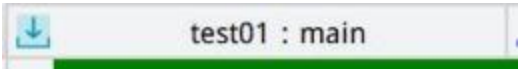
The global numeric variables are called in the teach file as #GVarX (X indicates the sequence number 1-300):



Click  to complete the modification after modifying the variable value.

8.2 Engineering Variables

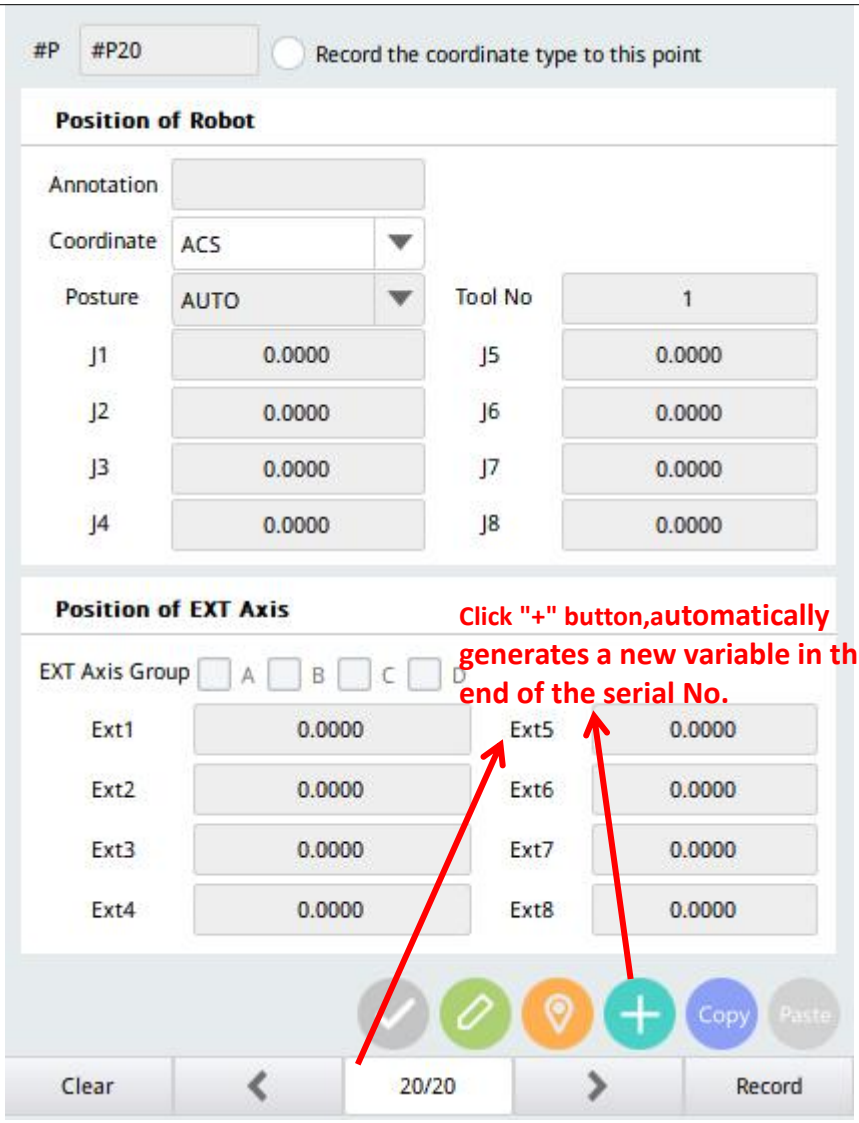
Each project has a set of project variables that can only be accessed by the teach file under that project. Including engineering position variable (#P), engineering increment position variable (#IncP), engineering value variable (#Var), engineering string variable (#Str).

Before viewing project variables, first you must open a project. The currently opened project can be obtained in the status bar. 

For example, the currently selected project is test01. In this case, the variables displayed in the project variable interface are the variables under this project.

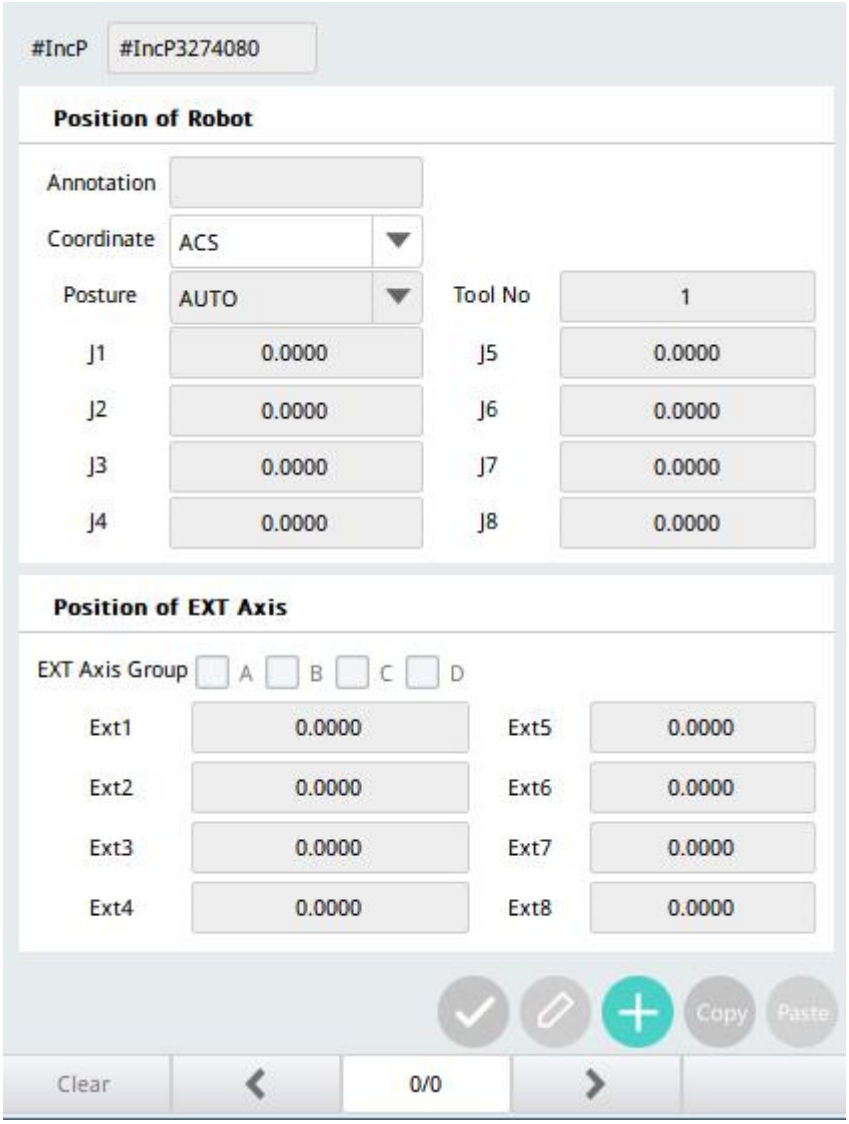
8.2.1 Engineering Positional Variable

The project position point is called in the form of #PX (X indicates the serial number) in the teaching file.

Item	Operating Instructions
1. New variable	
2. Other operations	Refer to global position variable

8.2.2 Engineering Incremental Positional Variable

The project location point is called in the form of #IncpPX (X represents the serial number) in the teaching file. The specific operation refers to the project location variable.



8.2.3 Engineering Numerical Variables

The engineering value variable is called in the teach file as #VarX (X is the sequence number).

Compared with global numerical variables, engineering numerical variables need to be added by users themselves click “ ” key, and other operations are the same.

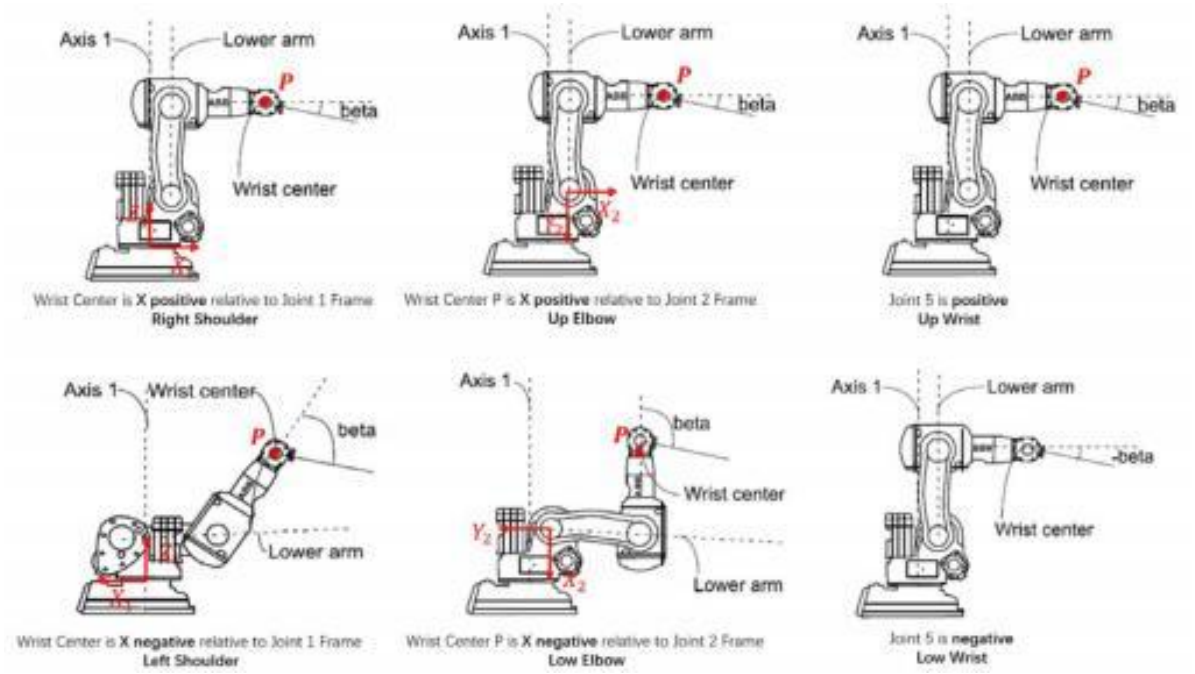
8.2.4 Engineering String Variables

The engineering string variable is called in the teach file as #StrX (X for ordinal).

Compared with the global string variable, the project string variable needs to be added by the user to click “ ” key, and other operations are the same.

8.3 Robot Posture

The robot pose can be used in position variables and incremental position variables. For the 6-axis robot pose model, the following definitions are used:



9 Coordinate System

This chapter describes the

- ACS - Axis Coordinate System
- KCS - Kinematic Coordinate System
- WCS - World Coordinate System
- TCS - Tool Coordinate System
- PCS - piece Coordinate System

9.1 Introduction to Coordinate System

The following coordinate systems can be used for axis manipulation of the robot.(ACS,KCS,WCS,TCS,PCS)

9.1.1 Axis Coordinate System – ACS

Axis Coordination System (ACS). The axis coordinate system is a pure rotation coordinate system which is established by taking the mechanical zero point of each axis as the origin. Each axis of the robot can rotate independently or together.

9.1.2 Kinematic Coordinate System – KCS

Kinematic Coordination System (KCS). The kinematics coordinate system is a coordinate system used to perform forward and reverse kinematics modeling on the robot, and is a basic Cartesian coordinate system of the robot, which may also be referred to as a robot base coordinate system (BCS) or a kinematics coordinate system. In this coordinate system, the robot end TCP may perform movement along the X axis, Y axis, and Z axis of the coordinate system, and rotation around the X axis, Y axis, and Z axis of the coordinate system.

9.1.3 Tool Coordinate System – TCS

Tool Coordination System (TCS)

Tool installation: the effective direction of the tool held by the robot wrist flange is taken as the Z axis of the tool coordinate system, and the origin of the tool coordinate system is defined at the tip point (or center point) TCP (TOOL CENTER POINT) of the tool.

Tool not installed: At this time, the tool coordinate system is established on the center point of the end face of the robot flange, and the Z axis direction is perpendicular to the end face of the flange and points to the front of the flange face.

When the robot moves, the tool coordinate system moves with the movement of the tool tip point TCP. The user can choose to perform the teach motion in the tool coordinate system TCS. The teaching motion in the TCS coordinate system includes a moving motion along the X, Y, and Z axes of the tool coordinate system and a rotating motion around the X, Y, and Z axes of the tool coordinate system. The robot system supports the user to save 32 user-defined tool coordinate systems.

9.1.4 World Coordinate System – WCS

World Coordinate System (WCS). The world coordinate system is also a spatial Cartesian coordinate system. The world coordinate system is a reference coordinate system of other Cartesian coordinate systems (the kinematics coordinate system KCS and the piece coordinate system PCS), and the kinematics coordinate system KCS and the piece Coordinate System PCS are

established with reference to the world coordinate system WCS. In the case where the world coordinate system is not configured for teaching by default, there is no positional offset or posture transformation between the world coordinate system and the robot kinematic coordinate system, and therefore the world coordinate system WCS and the kinematic coordinate system KCS coincide at this time. Users can teach the world coordinate system WCS through the Coordinate System Management interface. The end of the robot tool can move along the X axis, Y axis and Z axis of the coordinate system and rotate around the X axis, Y axis and Z axis of the coordinate system in the world coordinate system. The robot system supports the user to save 32 user-defined world coordinate systems.

9.1.5 Piece Coordinate System – PCS

Piece Coordination System (PCS). The piece coordinate system PCS is a Cartesian coordinate system based on the world coordinate system - WCS. The piece coordinate system is mainly used to facilitate the user to switch multiple identical workpieces under the world coordinate system WCS in one application. In addition, after teaching the piece coordinate system, the movement and rotation of the robot tool end TCP in the piece coordinate system can reduce the difficulty of teaching. The piece coordinate system supports the user to save 32 user-defined piece coordinate systems.

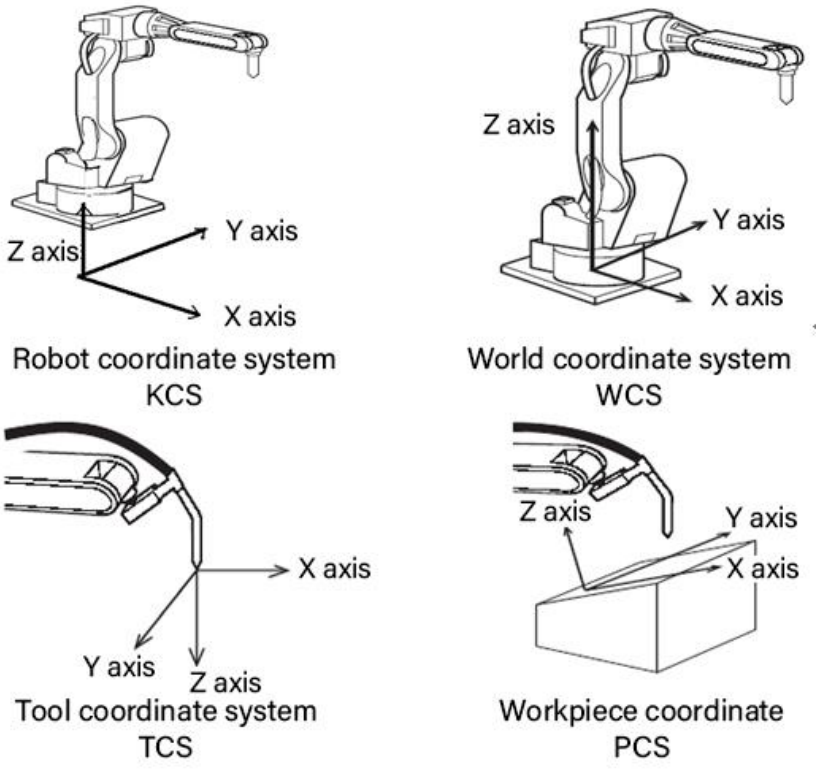


Figure 12 Schematic diagram of four coordinate systems

9.2 Coordinate System Management Interface

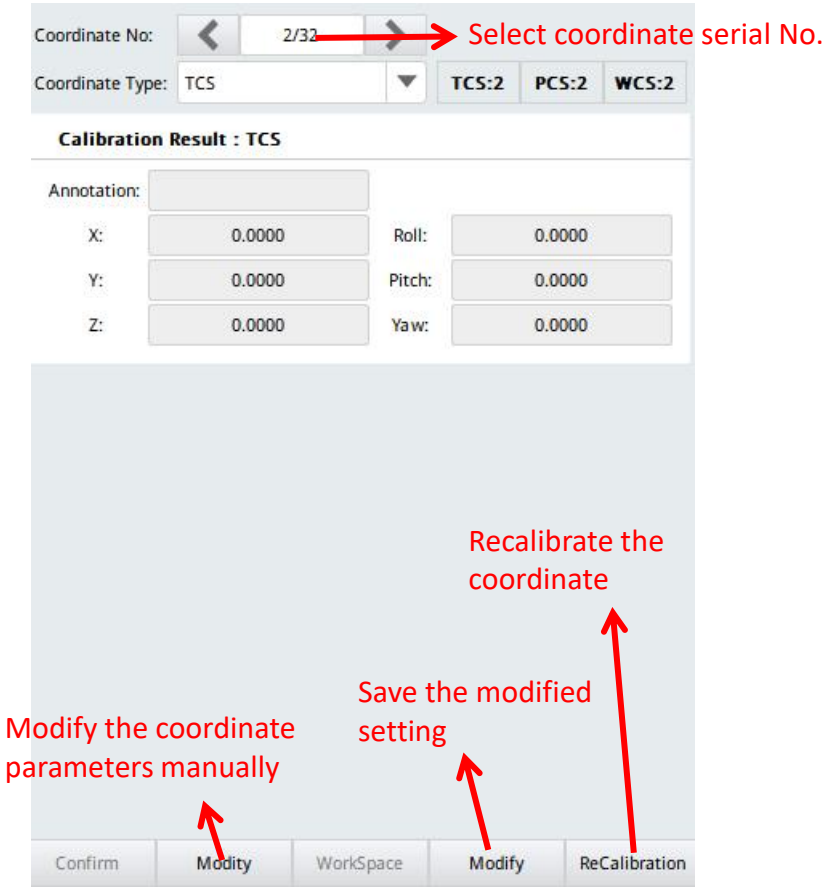


Figure 13 Coordinate System Interface

9.3 Axis Coordinate System (ACS)

In teach mode, when the coordinate system is set to the Axis coordinate system ACS, the J1, J2, J3, J4, J5 and J6 axes of the robot move separately. Please refer to the following table for the movement of each axis when the axis operation key is pressed:

Axis Name		Axis Operating Key	Action
Basic axis	J1 axis		Body left and right rotation
	J2 axis		Lower arm back and forth movement
	J3 axis		upper arm up and down movement

Wrist axis	J4 axis		upper arm forward and reverse rotation
	J5 axis		Wrist up and down movement
	J6 axis		Wrist forward and reverse rotation

9.4 Kinematic Coordinate System (KCS)

In the teach mode, when the coordinate system is set to the kinematic coordinate system KCS, the robot tool end TCP moves in parallel along the X, Y and Z axes of the KCS coordinate system and rotates around the X, Y and Z axes of the KCS coordinate system. When the axis operation key is pressed, refer to the following table for the movement of each axis:

Axis Name		Axis Operating Key	Action
Moving axis	X axis		Translational movement along the X axis of the KCS coordinate system
	Y axis		Translational movement along the Y axis of the KCS coordinate system
	Z axis		Translational movement along the Z axis of the KCS coordinate system
Axis of rotation	About the X axis		X-axis rotational movement around the KCS coordinates
	Around Y axis		Y-axis rotational movement around KCS coordinates
	Around Z axis		Z-axis rotational movement around KCS coordinates

9.5 Tool Coordinate System

The tool management interface is mainly used to manage the tools installed on the end flange of the robot.

9.5.1 Tool Coordinate System Axis Action

In the tool coordinate system, the effective direction of the tool held by the robot wrist flange is defined as the Z axis, and the origin of the coordinate system is defined at the tool center point (TCP), so the pose of the tool coordinate system changes with the movement of the wrist.

The default tool coordinate system is at the center of the robot flange until a tool is assigned, as shown in the following illustration.

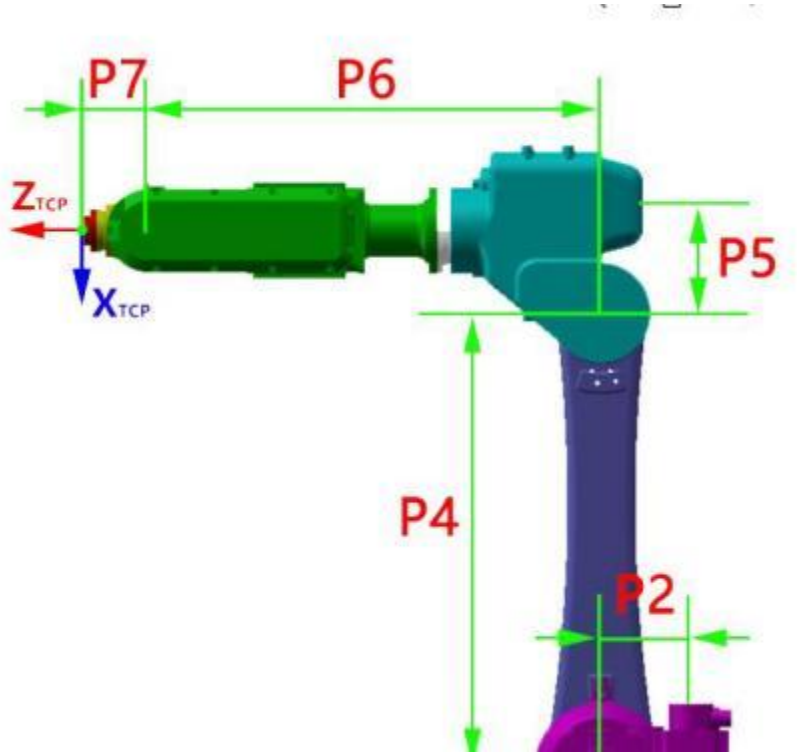


Figure 14 Default Tool Coordinate System

The movement along the tool coordinate system is based on the effective direction of the tool and is independent of the position and posture of the robot, so it is most suitable for parallel movement operation without changing the posture of the tool with respect to the workpiece.

Refer to the following table for the action of each axis:

Axis Name		Axis Operating Key	Action
Moving axis	X axis		Translational movement along the X axis of the TCS coordinate system
	Y-axis		Translational movement along the Y axis of the TCS coordinate system
	Z axis		Translational movement along the Z axis of the TCS coordinate system

Axis of rotation	About the X axis		X-axis rotational motion around TCS coordinates
	Around Y axis		Y-axis rotational motion around TCS coordinates
	Around Z axis		Z-axis rotational motion around TCS coordinates

9.5.2 Calibration of Tool Coordinate System

The user can enter the calibration management interface through the submenu {Coordinate System Management} under the system application menu. The main interface of tool coordinate system TCS management is the same as that of world coordinate system WCS management.

9.5.2.1 Parameter Interpretation

Calibration plane:

XY: Calibration selects three points P1-P3 in the XY plane, and the direction of the Z axis is obtained by cross multiplication of the X axis and the Y axis according to the right-hand rule.

YZ: Three points P1-P3 are selected to be located in the YZ plane for calibration, and the X-axis direction is obtained by the cross product of the X-axis and the Y-axis according to the right-hand rule.

ZX: Calibration selects three points P1-P3 to be located in the ZX plane, and the Y-axis direction is obtained by cross multiplication of the Z-axis and the X-axis according to the right-hand rule.

9.5.2.2 Four-Point Method: Calibration Position Offset(现在的版本偏移是 6 点法示教?)

When using {4-point method} for calibration, use the tip (center) point (i.e. TCP point) of the tool to be tested to approach the same reference point from four different directions. The reference point can be selected arbitrarily, but it must be the same fixed reference point. The robot controller calculates the TCP from four different flange positions. The four flange positions from the robot TCP point movement to the reference point must be spread out by a sufficient distance to make the calculated TCP point as accurate as possible.

The {4-point method} diagram is shown below:

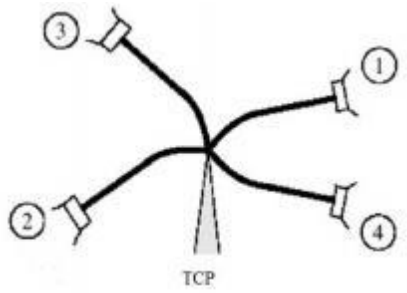
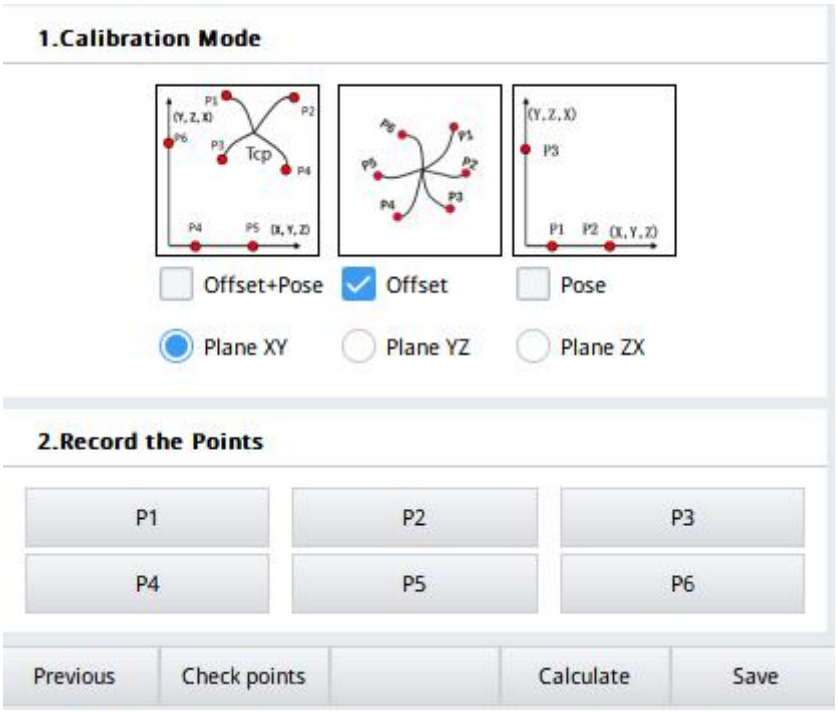


Figure 15 Diagram of four-point method

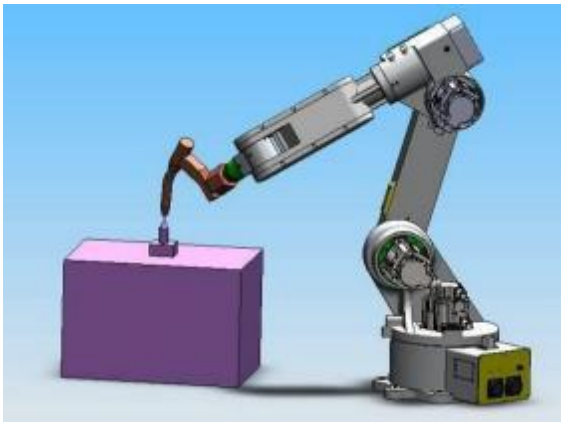
The steps of {4-Point Method} to teach and calculate the location of the tool center point TCP are as follows:

Step 1: Select the index number of the coordinate system to be refreshed and select the four-point teach mode.

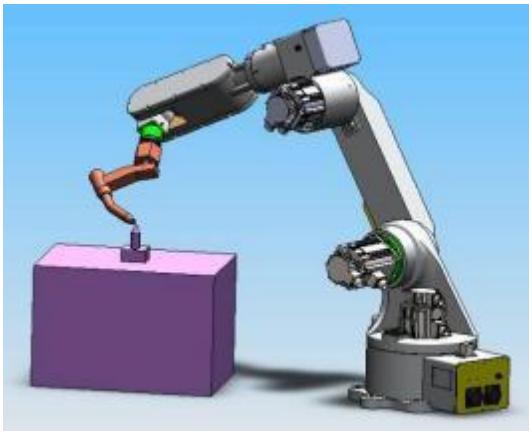


Four points are required to be recorded using the four-point method.

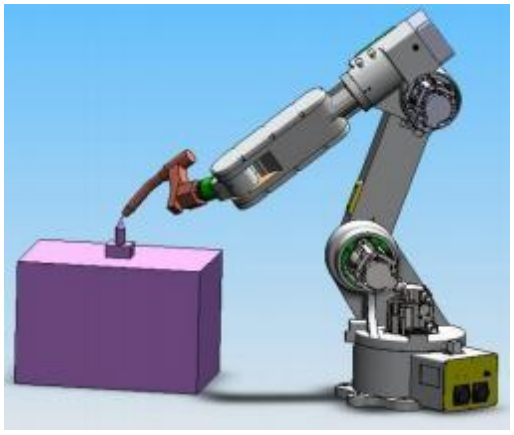
Step 2: Bring the tip point TCP of the tool under test closer to a fixed reference point from the first direction. Click the [Record P1] button with the servo power on to record the first position point. The background color of the [Record P1] button changes to green when the P1 point recording is completed.



Step 3: Bring the tip point TCP of the tool under test closer to the same fixed reference point from the second direction. Click the [Record P2] button with the servo power on to record the second position point. The background color of the [Record P2] button changes to green when the P2 point recording is completed.

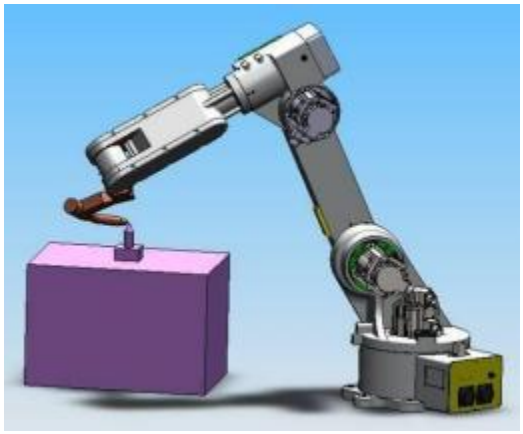


Step 4: Bring the tip point TCP of the tool under test closer to the same fixed reference point from a third direction. With the servo power on, click the [Record P3] button to record the third position point. The background color of the [Record P3] button changes to green when the P3 point recording is completed.



Step 5: Bring the tip point TCP of the tool under test closer to the same fixed reference point from the fourth direction. With the servo power on, click the [Record P4] button to record the fourth

position point. The background color of the [Record P4] button changes to green when the P4 point recording is completed.



Step 6: Click the [Calculate] button after recording the four position points required by the {4-point method} to automatically calculate the TCP position point data and display the calculation result.

Note that if two or more identical location points are recorded in the four-point method, the calculation will not succeed, and the program will report an error.

Step 7: Click the [Save] button to save the recorded coordinates of teaching position points and calculated coordinate system data and return to the main interface of coordinate system management.

Step 8: Click the [Set Current] button to use the calculated TCP tool as the flange end tool. At this point, you have completed all the steps from the tool coordinate system calculation to switching the newly calculated tool to the currently used tool. The tool coordinate system is calculated and switched successfully, and various movements of the robot can now be performed under the new tool.

Note that the four-point method can only determine the position offset value of the tool tip (center) point TCP relative to the mounting surface of the robot end flange. When the user needs to teach to determine the tool posture component, the three-point method needs to be additionally used, or the six-point method is directly used.

9.5.2.3 Three-Point Method: Calibrated posture Offset

{3-point method} The steps to teach and calculate the pose components of the tool coordinate system TCS are as follows:

Step 1: Select the serial number of the tool coordinate system to be modified or refreshed, and select the three-point working mode.

In the three-point method working mode, it is necessary to record three position points, i.e., P1 point, P2 point and P3 point. In addition, you need to select the plane in which the teach point is located, as shown below, select the XY plane. That is, the taught points P1 and P2 are used to determine the direction of the X-axis of the tool coordinate system, and the point P3 is on the positive Y axis side of the XY plane of the tool coordinate system. Since the three-point method only determines the posture component of the tool coordinate system TCP, the XY plane to be taught is only required to be parallel to the XY plane of the actual tool coordinate system TCS, and is not necessarily required to be the XY plane of the TCS. Point P1 is a point on the X axis of the coordinate system of the selected XY teaching plane, and is not required to be the tool tip (center) point TCP(the origin of the tool coordinate system), and so are points P2 and P3.

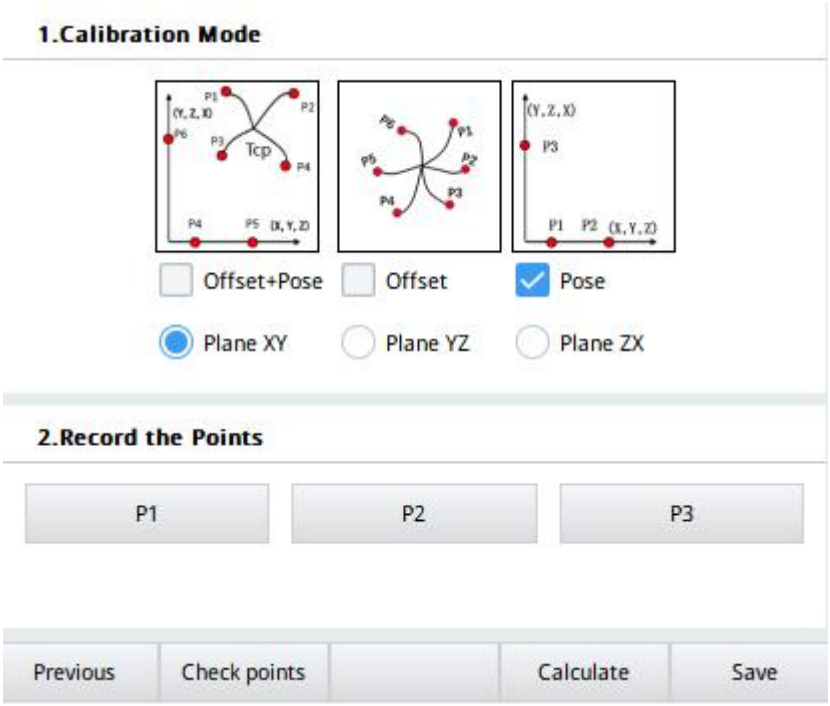
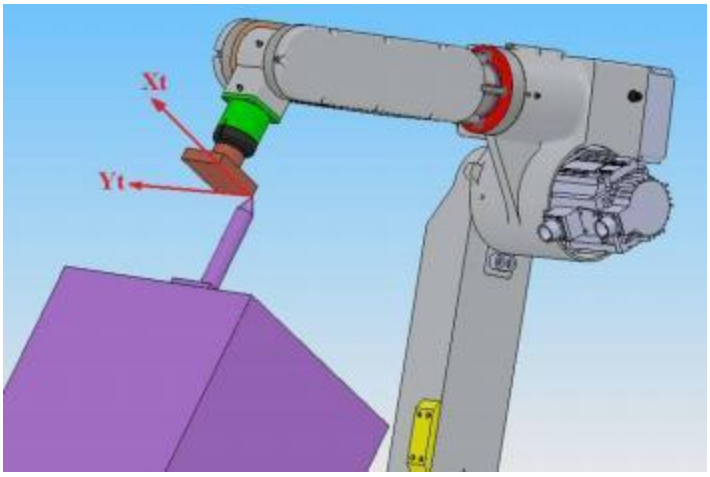


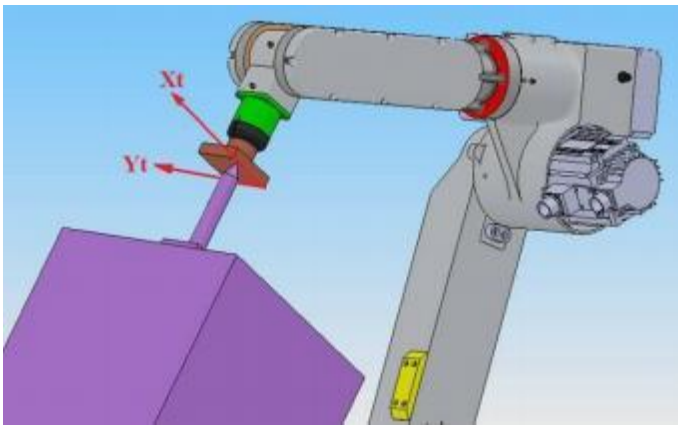
Figure 16 Select the three-point method operation mode

Step 2: As shown in the following figure, first record the first point in the X-axis direction on the tool coordinate system, i.e. P1 point.



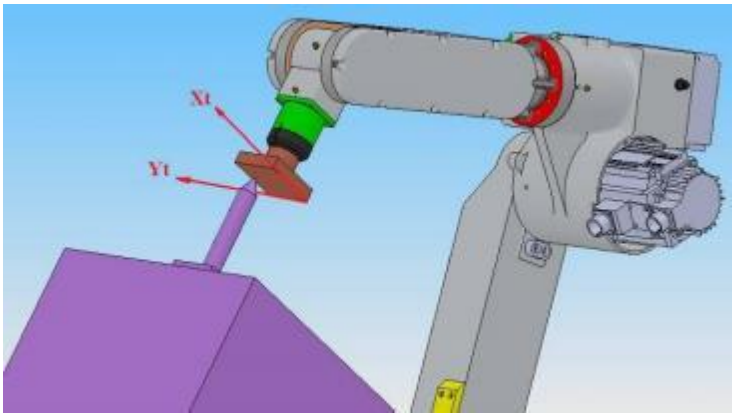
The figure 17 records the P1 point (the first point P1 in the positive direction of the X-axis)

Step 3: Record the second point in the X-axis direction on the tool coordinate system, point P2.



The figure 18 records the point P2 (the second point P2 in the positive direction of the X-axis)

Step 4: Record a point in the positive direction of the Y axis on the XY plane of the tool coordinate system, i.e., point P3.



The figure 19 records the P3 point (point in the positive direction of the Y axis on the XY plane)

Step 5: Click the {Calculate} button, and the program generates the tool coordinate system posture component data according to the recorded P1, P2 and P3 points, and click the [Save] button to update the posture component of the tool coordinate system TCS of the selected coordinate system serial number.

It should be noted that, when the three-point method (three position points to be recorded) is used to determine the tool posture, the three position points can only be taught by moving motion in Cartesian space (that is, only XYZ moving motion in KCS, WCS, PCS and TCS can be taught, but not XYZ rotating motion or single joint rotating motion in ACS), and cannot be taught by rotating motion with any posture, otherwise, the posture component of the tool coordinate system cannot be calculated, and an error warning is given.

9.5.2.4 Six-Point Method: Offset of Calibration Position and Posture

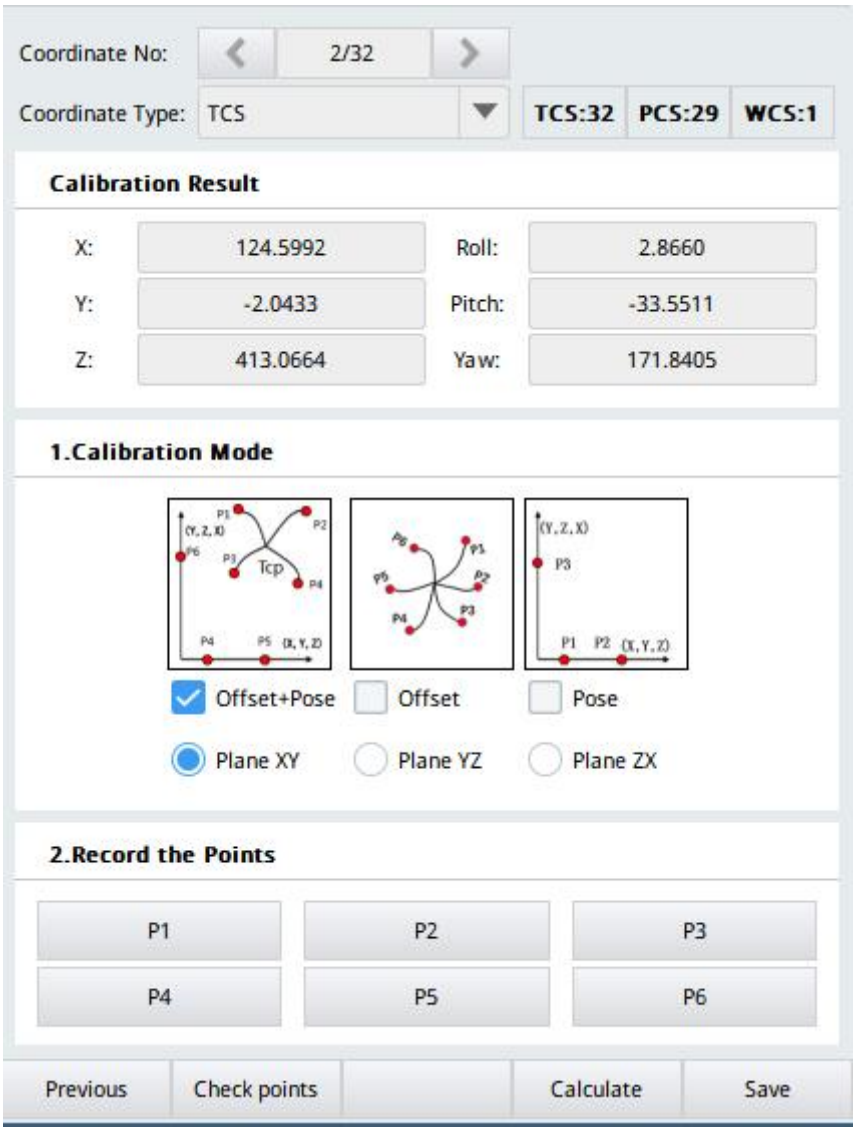


Figure 20 Tool Coordinate System Calibration Interface

There are three calibration methods for tool coordinate system: {6-point method}, {4-point method} and {3-point method}. {6-point method} can teach the position offset and pose vector of the end tool TCP (TOOL CENTER POINT) of the six-axis joint robot synthetically; {4-point method} can only calculate the TCP position offset value of the end of the six-axis robot tool, but can not calculate the pose vector of the tool; {3-point method} can only calculate the TCP pose vector of the tool end of the six-axis robot, but cannot calculate the position offset value of the TCP at the tool end. The {6-point method} is actually a comprehensive use of the {4-point method} and the {3-point method}. The first four position points recorded in the {6-point method} use the {4-point method} to calculate the position offset of the TCP at the end of the tool, and the last three position points use the {3-point method} to calculate the TCP pose vector of the tool.

9.6 Piece Coordinate System (PCS)

9.6.1 Piece Coordinate System Axis Action

In the teach mode, when the coordinate system is set to the piece Coordinate System PCS, the robot tool end TCP moves in translation along the X, Y and Z axes of the PCS coordinate system and rotates around the X, Y and Z axes of the PCS coordinate system. When the axis operation key is pressed, the actions of each axis are shown in the following table:

Axis Name		Axis Operating Key	Action
Moving axis	X axis		Translational movement along the X axis of the PCS coordinate system
	Y-axis		Translational movement along the y-axis of the PCS coordinate system
	Z axis		Translational movement along the Z axis of the PCS coordinate system
Rotation Axis	About the X axis		X-axis rotational motion around PCS coordinates
	Around Y axis		Y-axis rotational motion around PCS coordinates

	Around Z axis		Z-axis rotational motion around PCS coordinates

9.6.2 Calibration of Piece Coordinate System

The user can enter the calibration management interface through the submenu {Coordinate System Management} under the system application menu.

9.6.2.1 Parameter Interpretation of Calibration Method

Calibration plane:

XY: Three points P1-P3 are selected to be located in the XY plane for calibration, and the direction of the Z axis is obtained by cross multiplication of the X axis and the Y axis according to the right-hand rule.

YZ: Three points P1-P3 are selected to be located in the YZ plane for calibration, and the X-axis direction is obtained by the cross product of the X-axis and the Y-axis according to the right-hand rule.

ZX: Three points P1-P3 are selected to be located in the ZX plane for calibration, and the Y-axis direction is obtained by cross multiplication of the Z-axis and the X-axis according to the right-hand rule.

Mode:

Mode 1 of 3-points: Take the XY calibration plane as an example, P1 is the origin position, P2 is a point in the positive direction of the X axis, and P3 is a point on the XY plane. The origin of the coordinate system taught in this way is located at the point P1, the positive direction of the X axis is directed from the point P1 to the point P2, and the point P3 is located on one side of the positive direction of the Y axis. YZ plane and the ZX plane are the same.

Mode 2 of 3-points: Take the XY calibration plane as an example, one point P1 and another point P2 on the X axis, and teach the third point P3 on the Y axis. Make a vertical line perpendicular to the line P1-P2 through point P3, and the position of the vertical foot is the origin of the coordinate system.The coordinate system taught in this way has a positive X-axis direction from point P1 to point P2, and point P3 is on the positive semi-axis of the Y-axis.

Offset of home point: the coordinate system taught by the user using the above two methods is offset to the O0 position point in the teaching record.

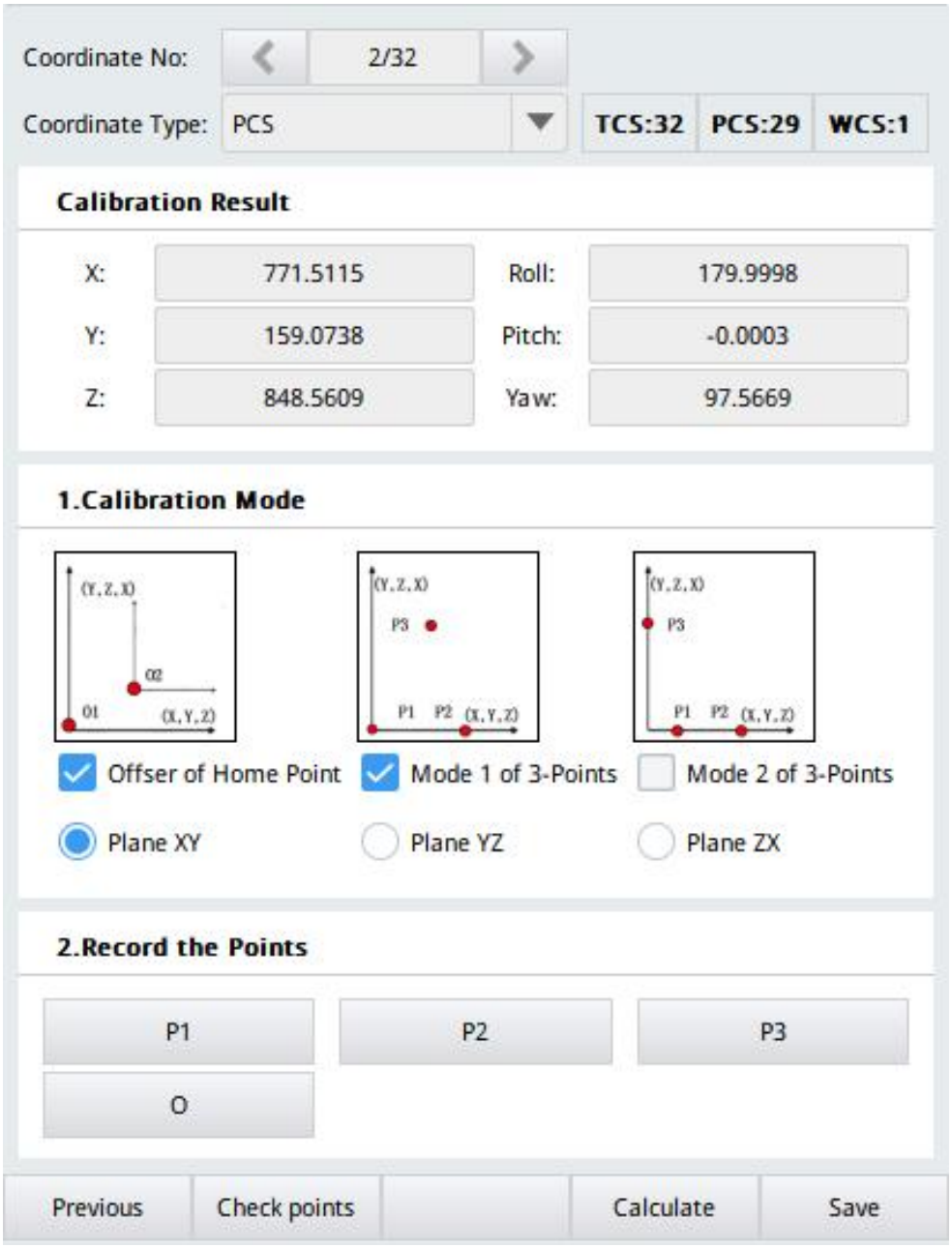


Figure 21 Calibration interface of piece coordinate system

9.6.2.2 How to Calibrate a Piece Coordinate System

Step 1: Select the coordinate system number and type

The default (1) coordinate system cannot be modified, please select the calibration starting from No. 2.

Step 2: Select Calibration Method

Select the calibration method [Mode 1 of 3-points], [Mode 2 of 3-points], refer to the above “Calibration Method Parameter Explanation” for details.

In addition, the user can also add and record a coordinate origin offset position point O0. This position point is optional. When the user uses this function, the coordinate system taught by the user using the above two methods can be offset to the O0 position point in the teaching record, while the pose of the coordinate system remains unchanged.

The user can select the taught coordinate system plane by pressing the {XY}, {YZ}, {ZX} keys on the interface.

Step 3: Location Point Recording

Move to the position point, click the corresponding button, and the color of the button changes to green (as shown in the figure below), indicating that the recording is successful. Note that the calibrated position points cannot be too close to each other.

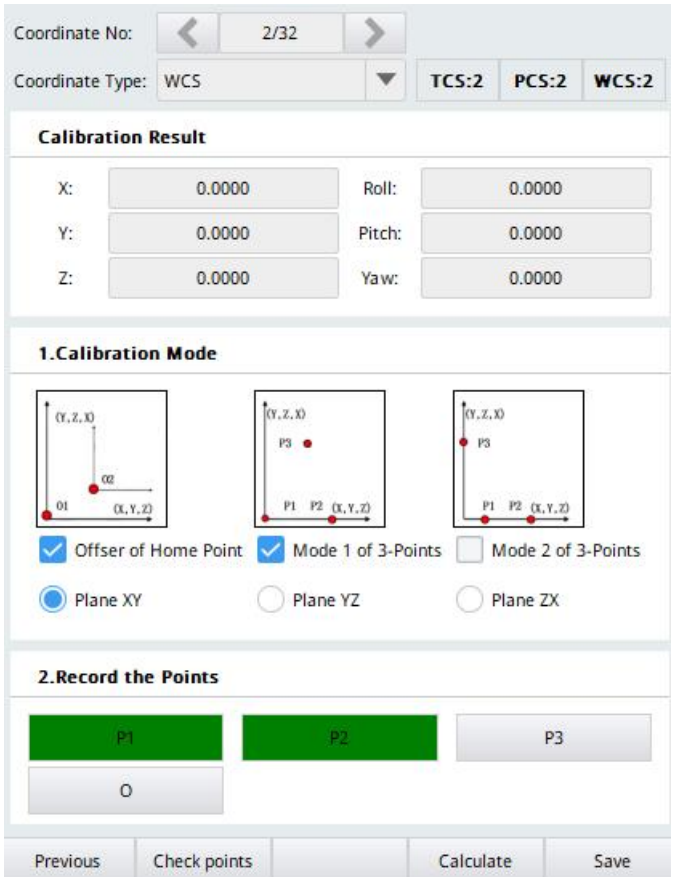


Figure 22 Piece Coordinate System Calibration Interface 2

Step 4: Calculate and Save Results

Click [Calculate], and click [Save] if the result is displayed successfully; if it fails, an alarm will be displayed in the status bar,modify or re-calibrate according to the alarm prompt.

In addition, on the position point recording interface, the user clicks the [Check Points] button to enter the teaching point management interface. On this interface, the user can view the recorded position data and make the robot move to the specified recording point. First select the recorded

position point, then make the robot in the servo enable state, select points P1, P2, P3 and [MoveTo], press and hold [Forward], and the robot moves towards the specified position point in the linear motion mode. Click [Previous] to return to the calibration screen.

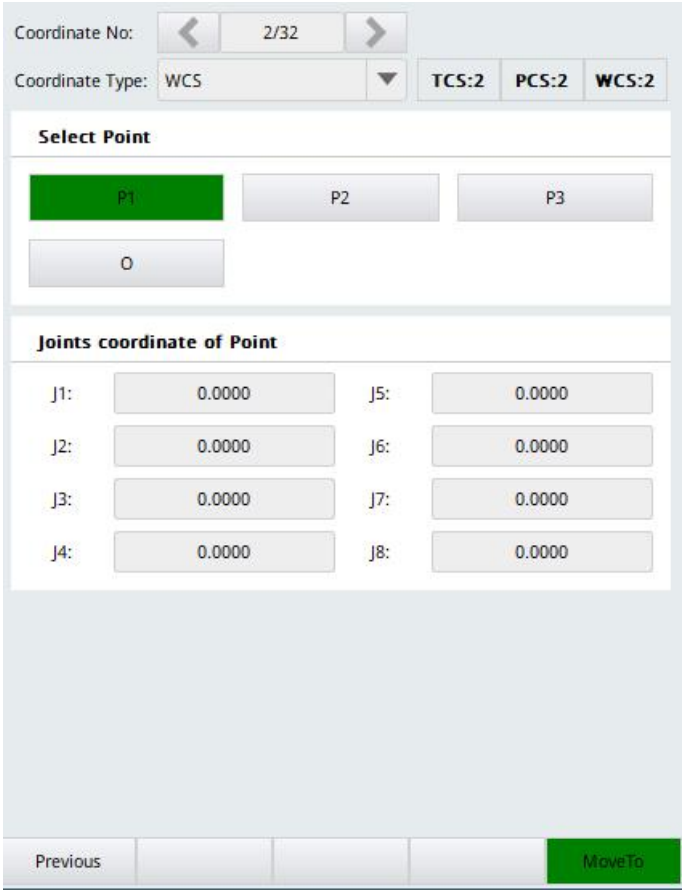
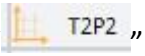


Figure 23 View interface of teaching points

Step 5:Set as current

After calibrating a piece Coordinate System, return to the coordinate system management interface, and set the calibrated coordinate system as the currently used coordinate system.

Check whether " 

" on the status bar shows the set coordinate system serial number, T is the tool coordinate system, and P is the piece Coordinate System.

Step 6: Verify Coordinate System

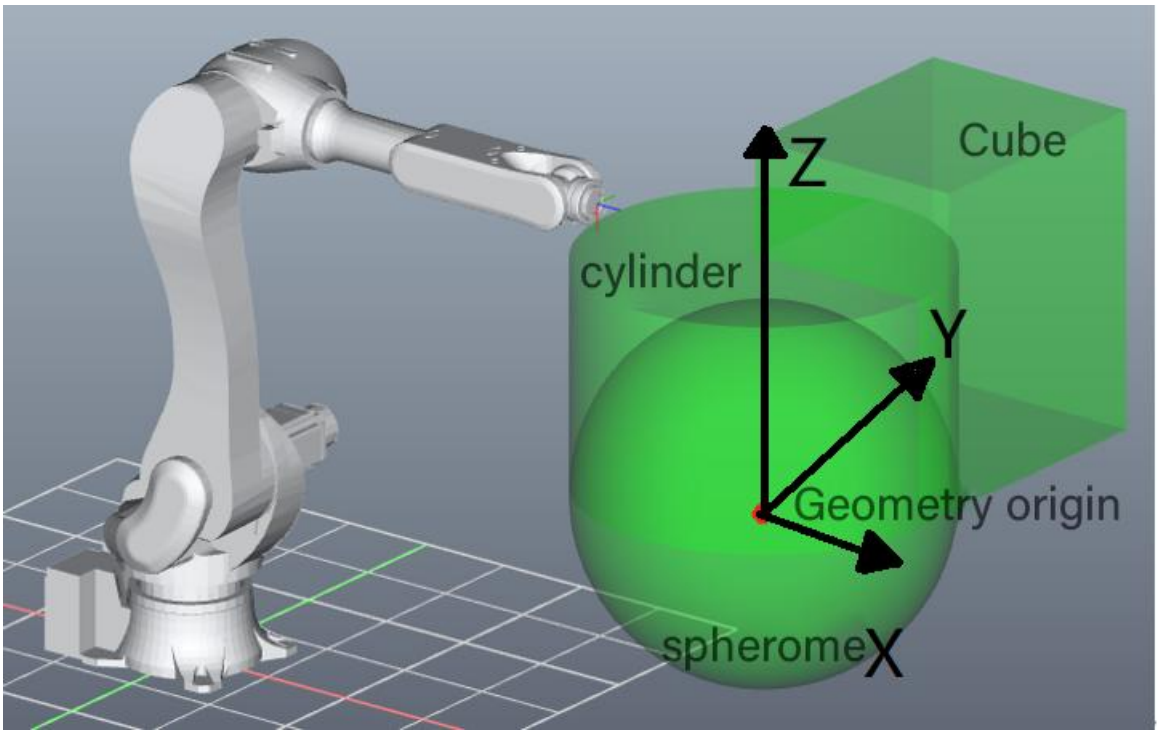
The directional accuracy of the calibrated coordinate system is verified by the translational motion of the robot tool end TCP along the X, Y, Z axes of the PCS coordinate system and the rotational motion about the X, Y, Z axes of the PCS coordinate system of the point (tip).

9.7 Workspace Monitoring

In a practical application scenario, a workspace of a robot often needs to be relatively limited, for example, a range of the workspace is set, the robot is limited to move within a certain range, or an overrun range of a working area is set, the robot is limited to move into a certain range, or a mutual interference range of work is set, for example, multiple devices need to share the same space area, and when the robot runs to the interference range, a signal can be output to a peripheral device working in the same interference range.so as to restrict each other and avoid collision.

The robot system develops the function of workspace monitoring for three types, including working area, obstacle area and monitoring area, and up to 32 workspace monitoring areas can be set.

9.7.1 Description of Working Space Monitoring

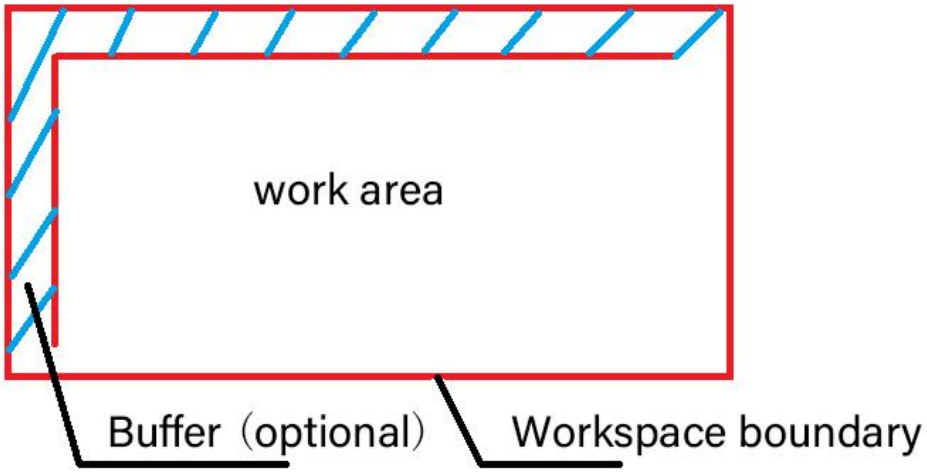


9.7.1.1 Work Area

Set a workspace of a geometric body (sphere, cube, cylinder), the current TCP point of the robot can only run in the set geometric body, once the scope of the workspace is exceeded, the robot will perform emergency braking and stop and trigger the alarm prompt of exceeding the workspace, or the workspace buffer boundary has been preset, when the current TCP point of the

robot enters the buffer boundary area, the robot will automatically decelerate to a safe speed to move within the buffer boundary.

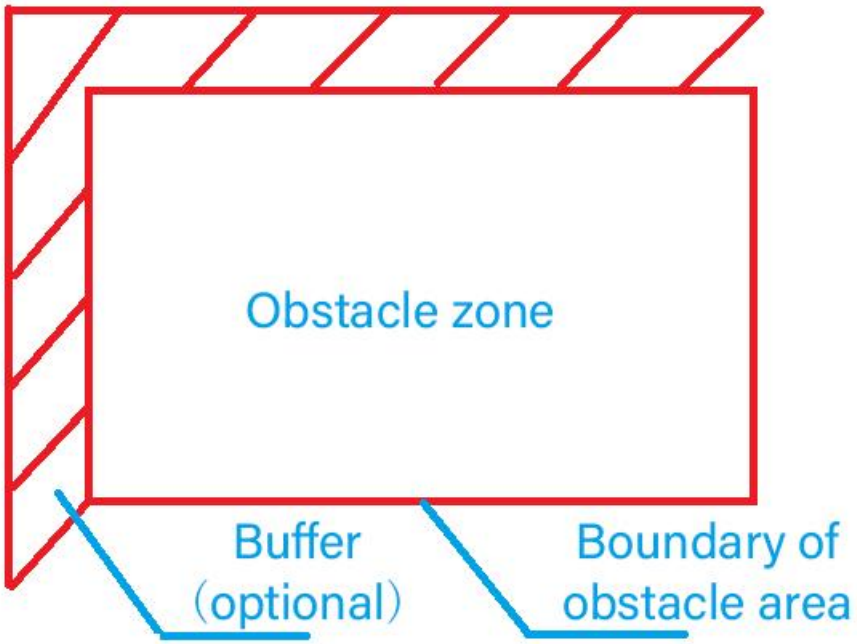
If the current TCP point of the robot does not exceed all the activated work areas, it is still determined that the robot is operating normally in the work area, i.e., does not exceed the work area. if that current TCP end of the robot exceed the range of the activated work area, i. e., exceed the work area, the robot will perform emergency braking and stop, and trigger an alarm prompt of exceeding the work space, wherein the alarm contains a corresponding work space serial number.



9.7.1.2 Obstacle Area

An obstacle space of a geometric body (sphere, cube, cylinder) is set, the current TCP point of the robot can only operate outside the set geometric body, once entering the obstacle area range, the robot will perform emergency braking and stop and trigger an alarm prompt that the robot has entered the obstacle area, or the obstacle buffer boundary has been preset, and when the current TCP point of the robot enters the buffer boundary area, the robot will automatically decelerate to a safe speed to move in the buffer boundary.

If the current TCP point of the robot enters any activated obstacle area, it will be judged that the robot has entered the obstacle area, the robot will perform emergency braking and stop, and trigger an alarm prompt for entering the obstacle area, and the alarm contains the corresponding workspace serial number.



9.7.1.3 Monitoring Area

Setting a monitoring space of a geometric body (sphere, cube, cylinder), when the current TCP point of the robot enters the set monitoring interval, the robot will perform input and output IO signal triggering according to user's setting, for example, setting the input as low/high level triggering, and outputting IO signals or values; at the same time, a warning prompt that the robot has entered the monitoring area will pop up, or the obstacle buffer boundary has been preset. When the current TCP point of the robot enters the buffer boundary area, the robot will automatically decelerate to a safe speed to move within the buffer boundary.
If the current TCP point of the robot enters any activated monitoring area, it will be judged that the robot has entered the monitoring area, and a warning prompt of entering the obstacle area will be triggered; set the monitoring area in the teach mode, the robot will not stop the movement in an emergency; in the play mode, the robot will stop the movement in an emergency after entering the monitoring area, and the alarm contains the corresponding workspace serial number.

9.7.2 Working Space Monitoring Setting

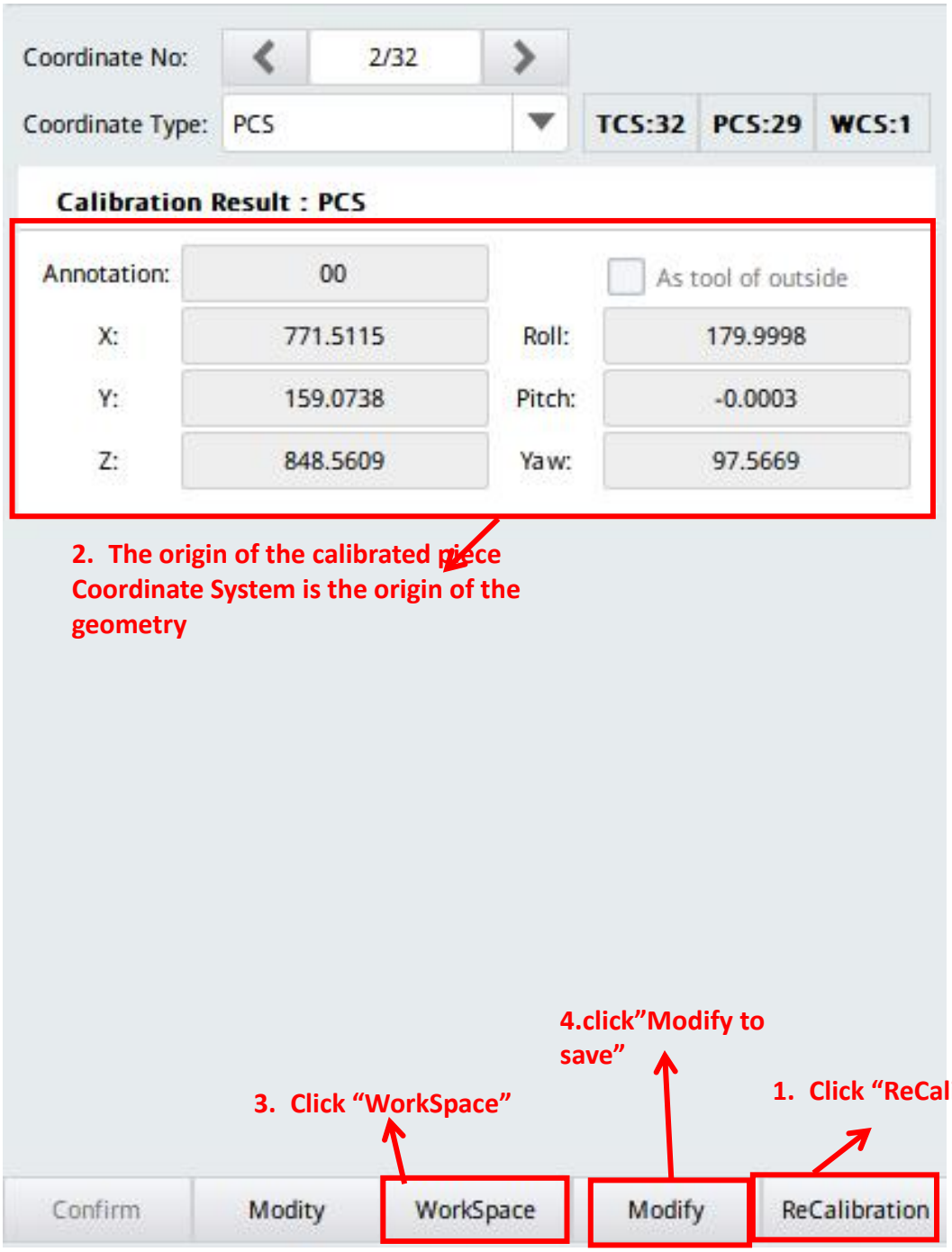
The geometry model of the work monitoring space is set based on the currently bound piece Coordinate System, that is, the set piece Coordinate System origin is the origin of the set

geometry, and the origin offset can also be set functionally, that is, the offset is performed based on the geometry origin and the piece Coordinate System origin.

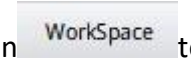
9.7.2.1 Geometry Origin Setting

In the [menu]  , select and click [Coordinate] 

to enter the coordinate system calibration interface, and select the coordinate system serial number. Select [piece Coordinate System/PCS] as the coordinate system type, and then perform coordinate calibration. For the description of calibration operation steps, please skip to 9.6.2 calibration of piece Coordinate System.

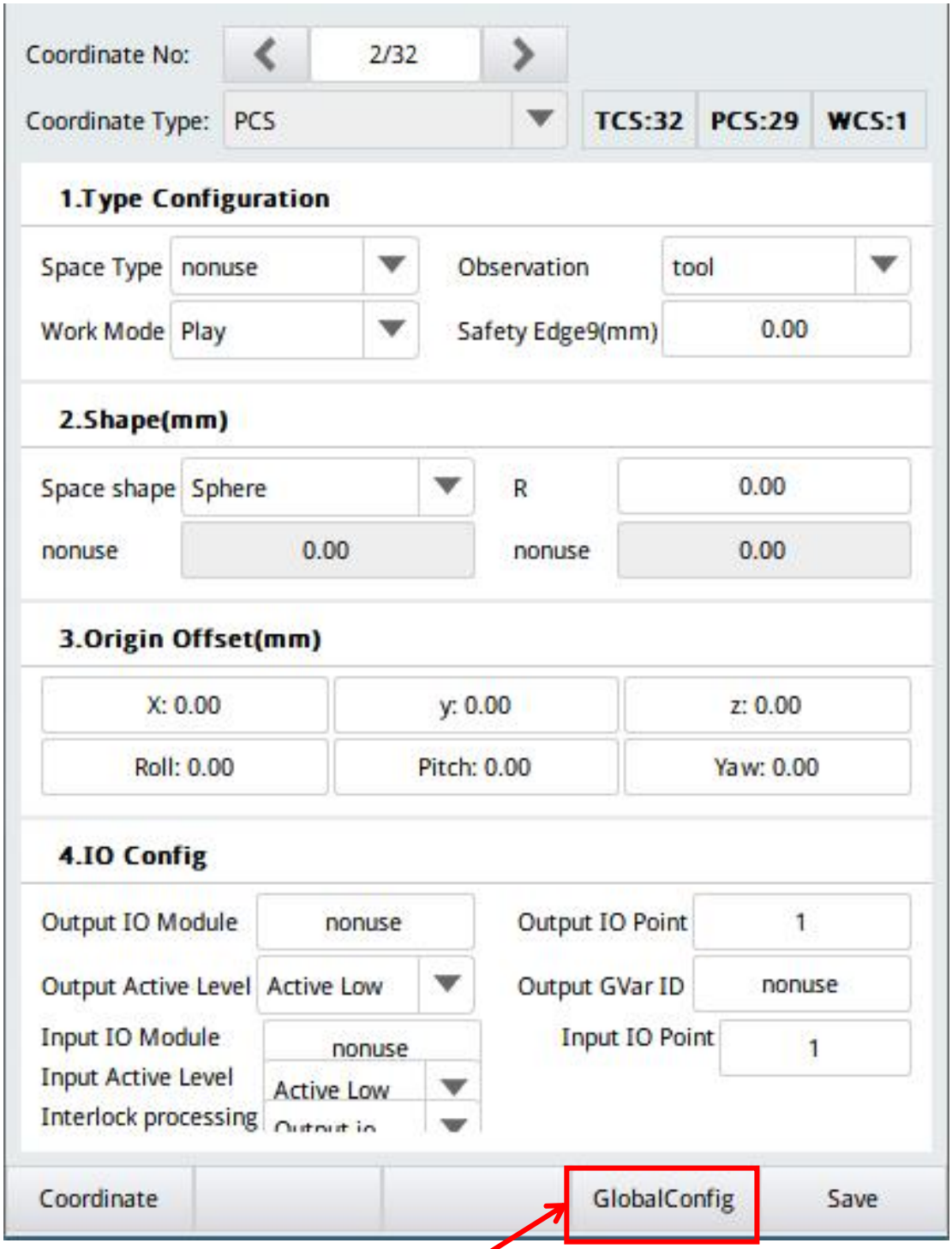


9.7.2.2 Global Monitoring Settings

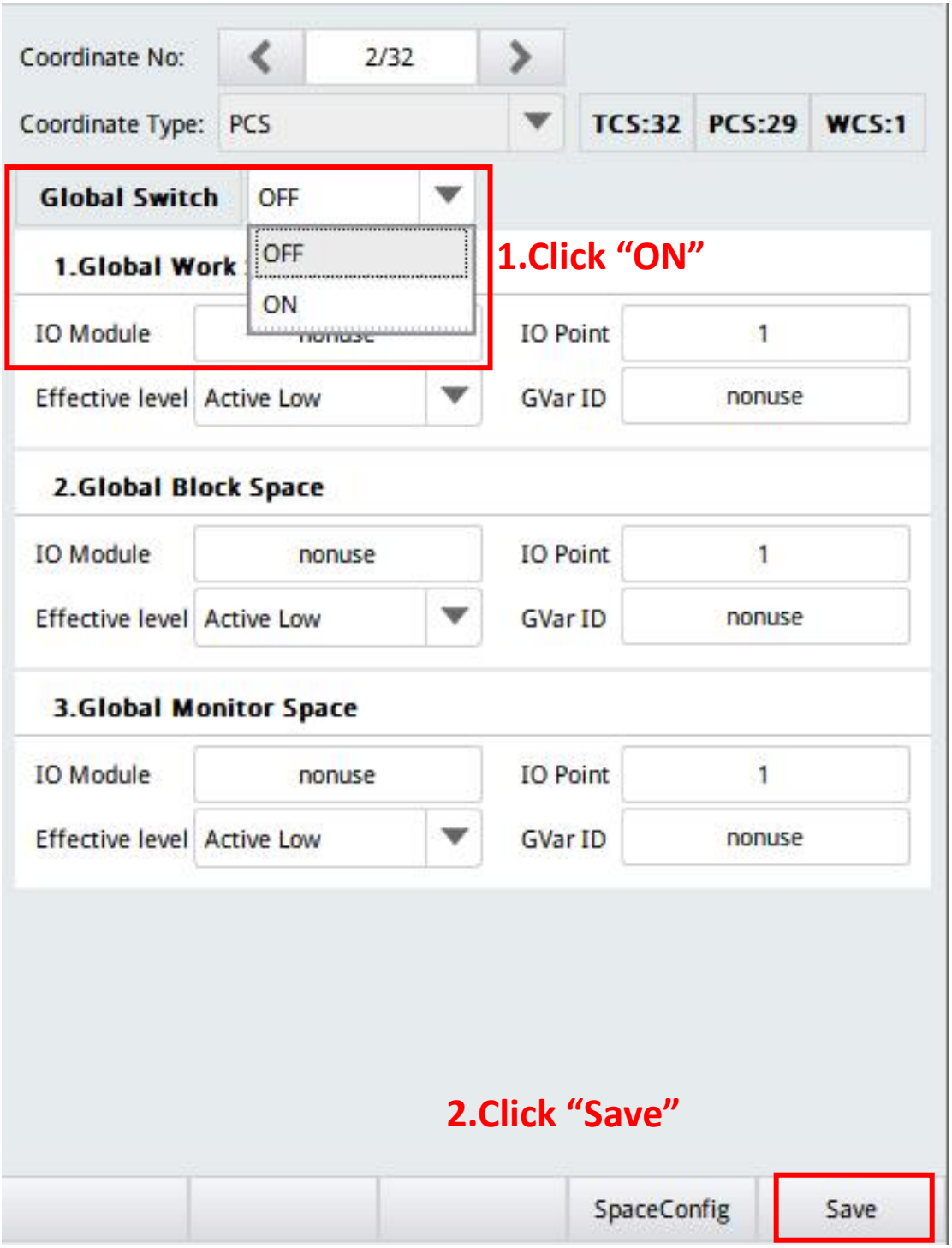
After calibration of piece Coordi nate System, select and click {Workspace} in the bottom column 

to enter the system as shown in the figure.

Note: When using workspace monitoring for the first time, you need to click to enter {Global Configuration} to enable settings.



Go to "GlobalConfig" for the first time



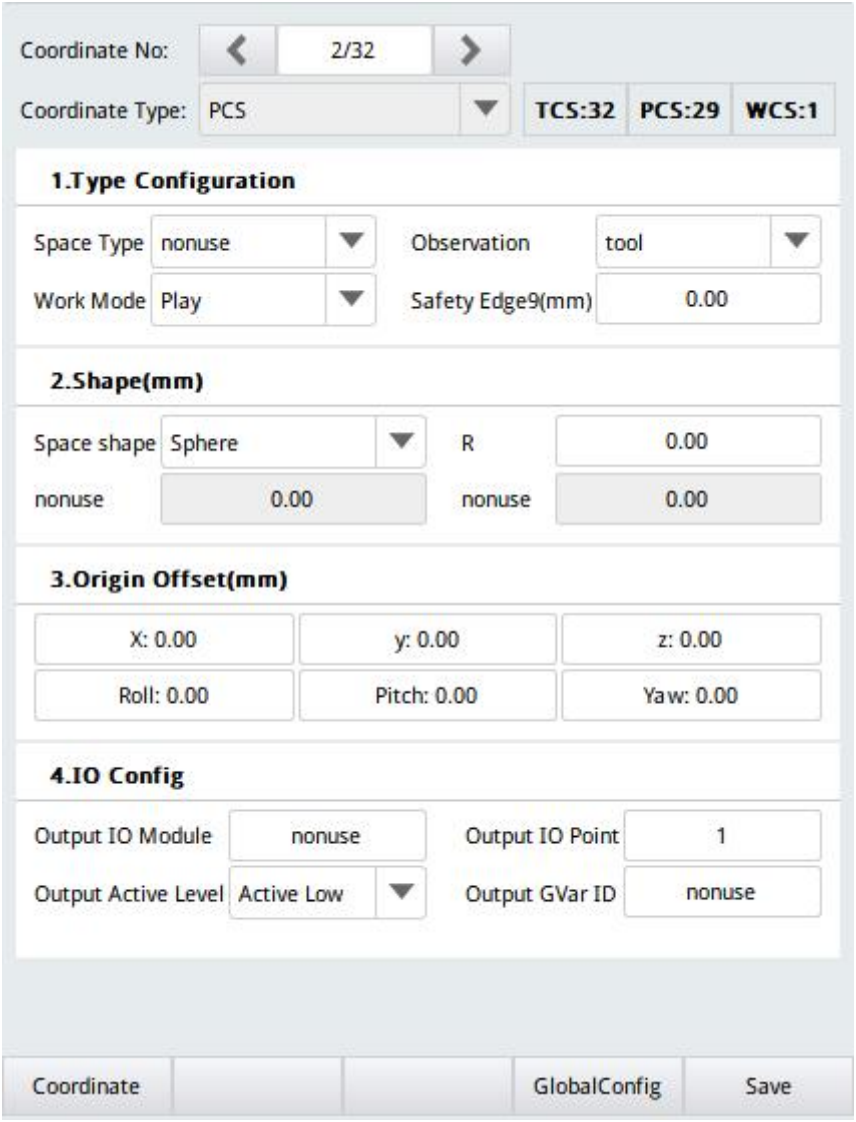
Name		Definition
Global monitoring switch		Enable/disable switch for workspace monitoring
Global workspace	IO Module	When the current TCP point of the robot exceeds all active working areas, the corresponding IO signal is output. If not, enter -1 in the input box
	Output Gvar Sequence	When the current TCP point of the robot exceeds all active working areas, the corresponding Gvar sequence number is

	Number	assigned 1. If not, enter -1 in the input box
Global obstacle area	IO Module	When the current TCP point of the robot enters any activated obstacle area, the corresponding IO signal is output. If not, enter -1 in the input box
	Output Gvar Sequence Number	When the current TCP point of the robot enters any activated obstacle area, the corresponding Gvar sequence number is assigned 1. If not, enter -1 in the input box
Global interference region	IO Module	When the current TCP point of the robot enters any activated interference area, the corresponding IO signal is output. If not, enter -1 in the input box
	Output Gvar Sequence Number	When the current TCP point of the robot enters any activated interference area, the corresponding Gvar sequence number is assigned 1. If not, enter -1 in the input box

9.7.2.3 Global Monitoring Settings

Note: 1. The coordinate system serial number can be selected from 1 to 32; During workspace configuration, please pay attention to the selected coordinate system serial number.

2. Click Save, and the data will take effect.



Name		Definition
Type Configuration	Area Type	There are four options: 1. Not effective; 2. work area; 3. obstacle area; 4. Surveillance areas;
	Mode of operation	Three modes are available: 1. Auto mode: only effective in play mode 2. teach mode: effective only in teach mode 3. Teach + Auto mode: effective in play and teach mode
	Monitoring point	Two monitoring points are included;1. Tool end;2. Flange center
	Security boundary (mm)	That is, the buffer boundary, when the robot runs to the safety boundary, the running speed will drop to the safe speed.
Shape	Sphere	The origin of the current corresponding piece Coordinate System is

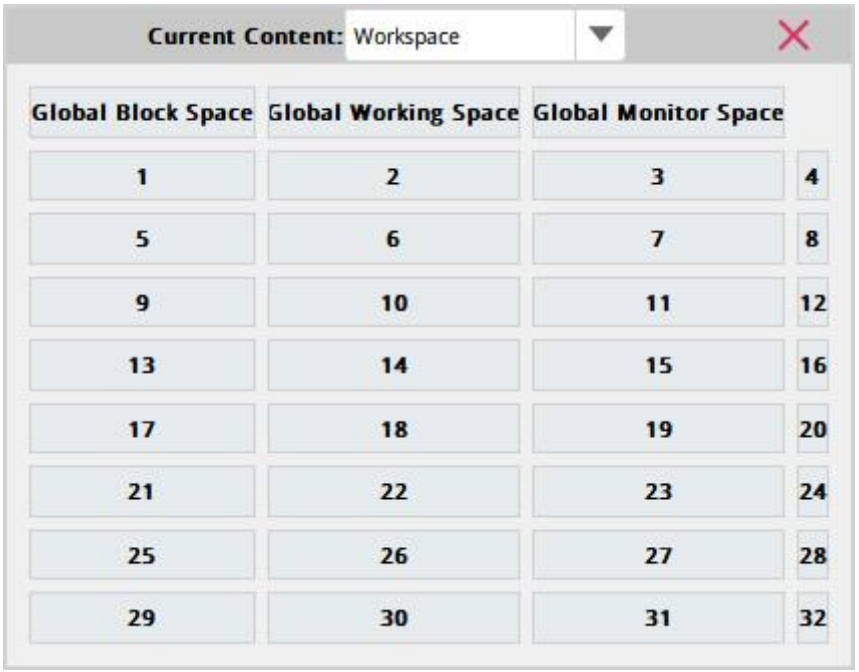
configuration (mm)		taken as the center of the sphere, and the radius is automatically set to limit a space area.	
	Cube	The origin of the current corresponding piece Coordinate System is taken as the vertex of the bottom surface of the cube, and the length, width and height are set by themselves to limit a space area.	
	Cylinder	The origin of the current corresponding piece Coordinate System is taken as the center of the bottom circle of the cylinder, and the radius and the height are automatically set to limit a space area.	
Area origin offset (mm)		Based on the origin of the current corresponding piece Coordinate System, the origin of the geometric body is offset in the directions of X, Y, Z, around X, around Y and around Z.	
IO Configuration	Output IO module	Working area	When the current TCP point of the robot exceeds the activated work area, the corresponding IO signal is output. If not, enter -1 in the input box
		Obstacle area	When the current TCP point of the robot enters the activated obstacle area, the corresponding IO signal is output. If not, enter -1 in the input box
		Interference region	When the current TCP point of the robot enters the activated interference area, the corresponding IO signal is output. If not, enter -1 in the input box
	Output Gvar Sequence Number	Working area	When the current TCP point of the robot exceeds the active working area, the corresponding Gvar sequence number is assigned 1. If not, enter -1 in the input box
		Obstacle area	When the current TCP point of the robot enters the activated obstacle area, the corresponding Gvar sequence number is assigned 1. If not, enter -1 in the input box
		Interference region	When the current TCP point of the robot enters the activated interference area, the corresponding Gvar sequence number is assigned 1. If not, enter -1 in the input

			box
	Input IO module (only valid in interference area)	Interference region	When the corresponding input IO signal is triggered, the current TCP point of the robot enters the activated interference area, and the robot stops moving urgently
	Interlock processing (only valid in interference area)		There are two options: 1. Continuous output: when the current TCP point of the robot enters the activated interference area, the corresponding output IO signal continues to be output; 2. Ignore output: If the input signal has been triggered, the current TCP point of the robot enters the activated interference area, and the corresponding output IO signal is not output.

9.7.2.4 Work Space Monitoring

Manually press the viewing interface Fn1 of the handheld box 

and select {Workspace Interface} to view the activated and triggered workspace monitoring.



10 System Parameter Setting

Click the Logo in the upper left corner to open [System Application], and select [System Parameters] to set parameters. After the parameter setting is completed, click [Save] below to save the parameters.

Select the setting parameter, and you can use the [Copy][Paste] button for shortcut operation.

10.1 System Parameters

Figure 24 System Parameter Interface

No.	Parameter Name	Parameter Description	
1	Remote Protocol	Modebus TCP	Remote external IO to control the robot
		Io Mode	The upper computer communicates with the robot through Modbus TCP
		TCP Remote	The upper computer communicates with the robot through Tcp/ip
		Modbus RTU	
2	Teach pendant Inspection	Disconnection And Ignore	There is no prompt after the handheld box is disconnected or has been disconnected
		Disconnection and Alarm	After the hand-held box is offline, it will prompt the alarm of "Hand-held box is offline or has been offline"
3	Auto Annotation for CMD	Close	Indicates that the function of automatically adding comments to commands is not enabled;
		Open	Indicates that the command is enabled to automatically add comments; when writing a program, it will automatically add comments after setting the command line

4	Movc2 teach check	linearMotion	When teaching and running the circle command program, it is set to run to the midpoint of the circle in a straight line
		CircularMotion	When teaching and running the circle command program, it is set to run to the midpoint of the circle in the form of an arc
5	Auto break off	Close	Indicates that the brake release function is not used;
		Open	Indicates that the brake release function is used; in an emergency, if the robot cannot be operated through the teaching pendant, this function can be turned on at this time to perform brake release operations on the 4, 5, and 6 axes of the robot (note: safety protection must be done)
6	Setpc check	Close	--
		Open	--
7	Main craft	--	--

10.2 Hardware Parameters

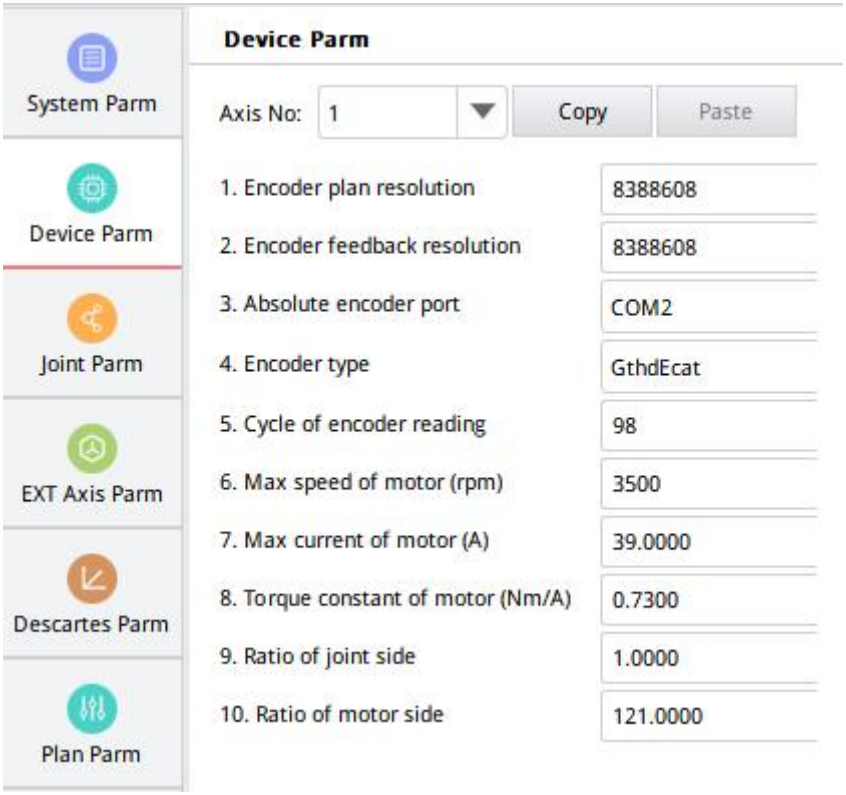


Figure 25 Hardware Parameter Interface

Parameter list:

No.	Parameter Name	Parameter Description
1	Encoder plan resolution	Example of encoder lines: The resolution of a 17-bit encoder is 131072
2	Encoder feedback resolution	
3	Absolute Encoder Port	Bus-based control value This parameter is invalid
4	Encoder Type	Default GthdEcat
5	Cycle of encoder reading	The absolute encoder data is read regularly to refresh the current absolute position of the robot
6	Max speed of motor (rpm)	The maximum speed of the motor, it is not allowed to exceed this speed during actual operation
7	Max current of motor (A)	The maximum current of the motor,it is not allowed to be exceeded this value during actual operation.
8	Torque constant of motor (Nm/A)	Torque produced by unit current
9	Ratio of joint side	The ratio of the rotation angle of the motor end to the rotation angle of the execution end (the allowable value is
10	Ratio of motor side	

		positive or negative to change the direction of joint movement)

10.3 Axis

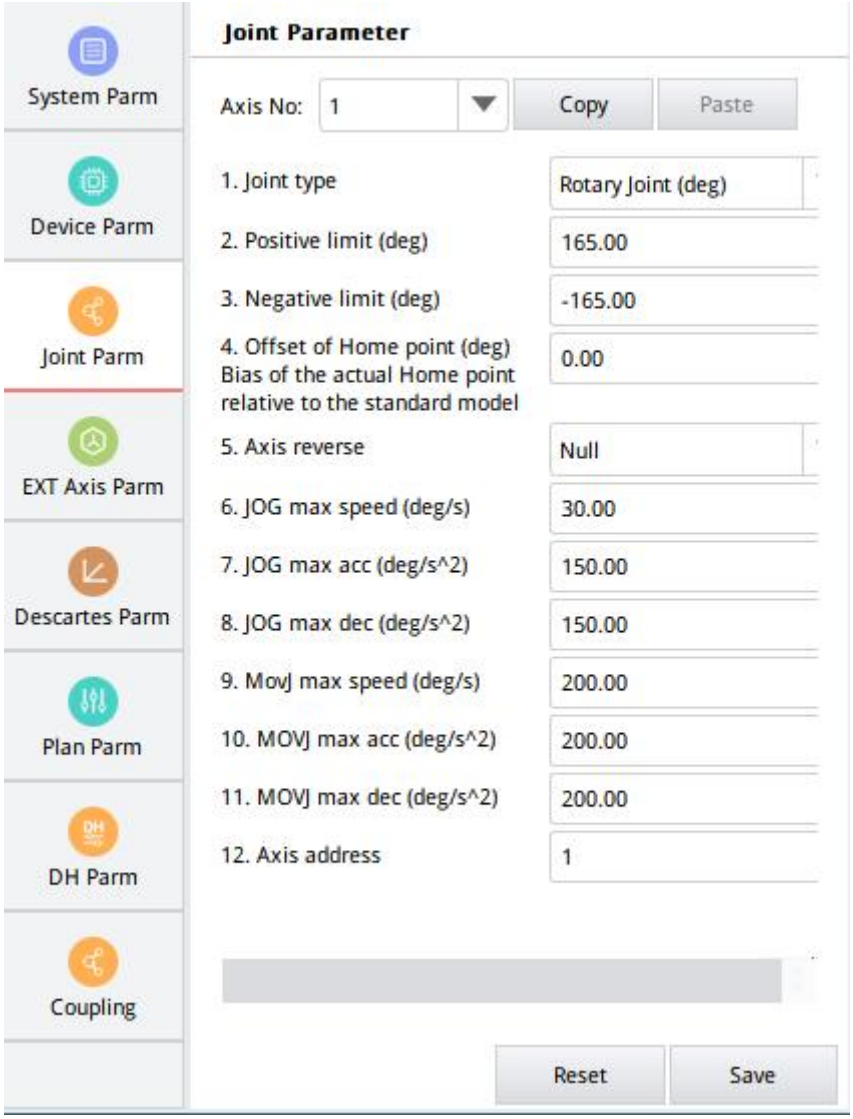


Figure 26 Axis Parameter Interface

Parameter list:

No.	Parameter Name	Parameter Description
1	Joint Type	Null; Rotary joint (deg);
		Translational Joint (mm);
2	Positive limit(deg)	Maximum joint range of motion
3	Negative limit(deg)	Minimum joint range of motion
4	Offset of home point(deg)	The deviation between the user-defined joint zero

	(Bias of the actual home point relative to the standard model)	point and the system internal robot model zero point
5	Axis Reverse	Whether the rotation direction of the user-defined joint is opposite to the direction of the robot model inside the system
6	JOG Max Speed(X/s)	The maximum value of the movement speed of the jog axis operation keys in the joint coordinate system
7	JOG Max Acc(X/s^2)	The maximum value of the movement acceleration of the jog axis operation keys in the joint coordinate system
8	JOG Max Dec(X/s^2)	The maximum value of the movement deceleration of the jog axis operation keys in the joint coordinate system
9	Movj max speed(X/s)	MOVJ command joint movement speed maximum
10	Movj Max Acc(X/s^2)	MOVJ command joint movement acceleration maximum value
11	Movj Max Dec(X/s^2)	MOVJ command joint motion deceleration maximum value
12	Axis address	The station number used in the bus communication must match the setting of the host computer

10.4 External axis

External axis parameters refer to the [Axis] parameter setting, and external axis that are not used have parameter 1 [Axis Type] selected as “Null”.

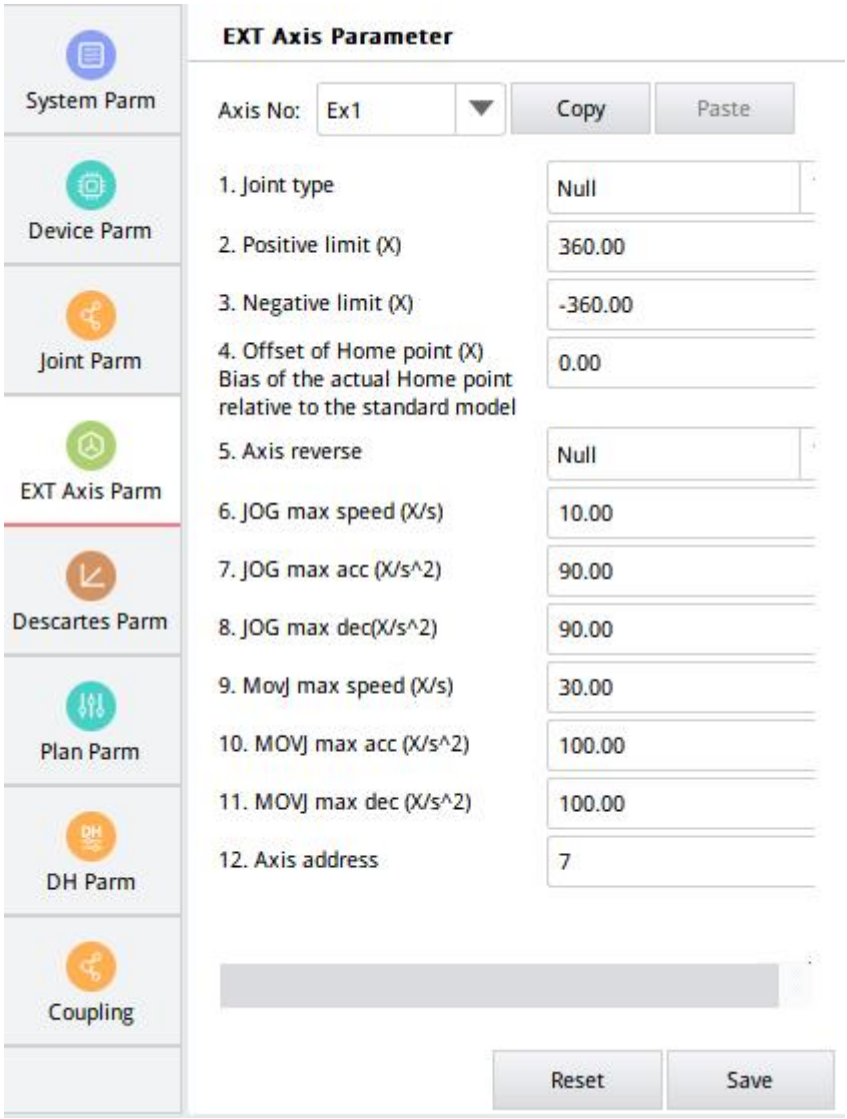
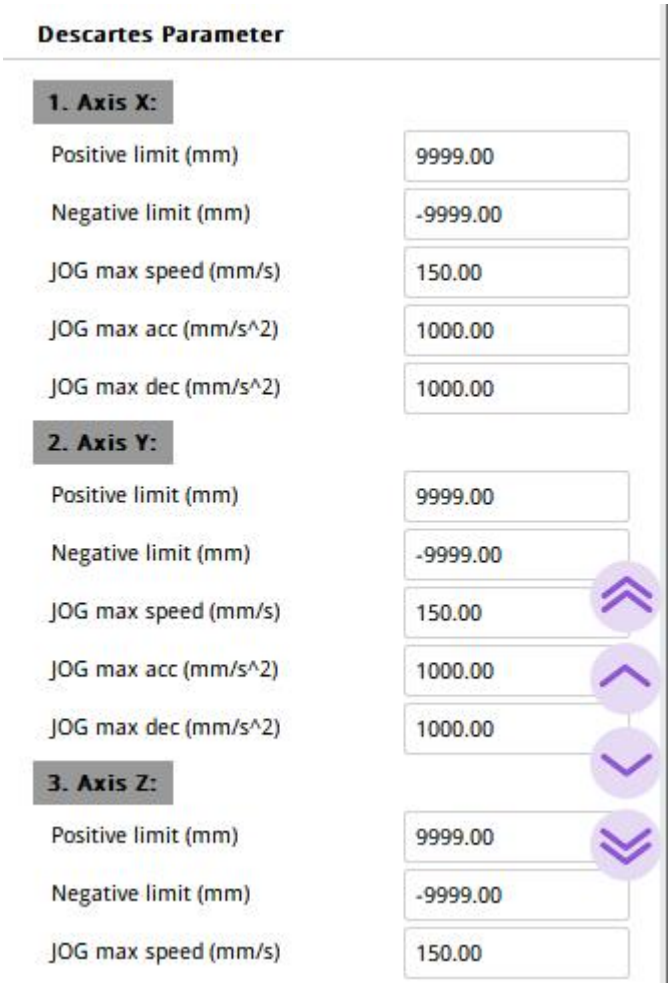


Figure 27 Auxiliary Shaft Parameter Interface

10.5 Descartes



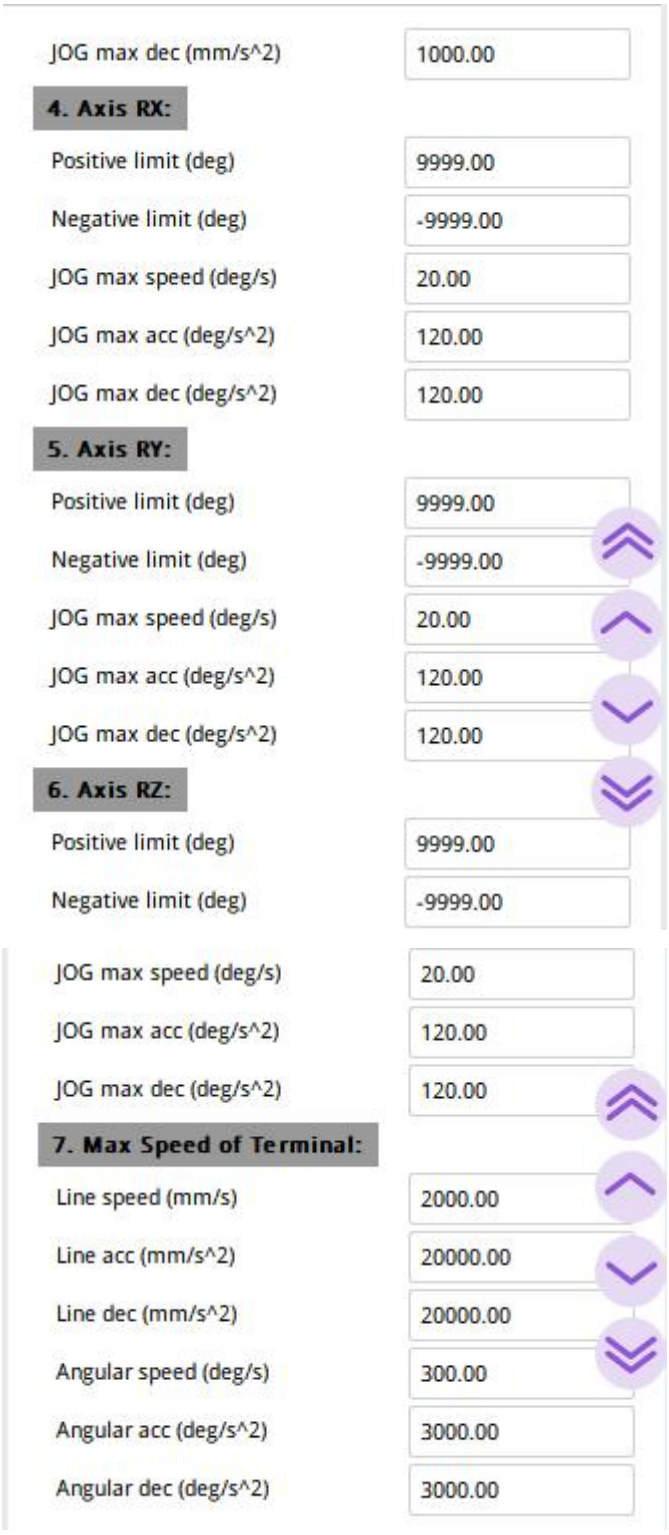


Figure 28 Descartes Parameter Interface

Parameter list:

No.	Parameter Name	Parameter Description
1	Positive limit(mm)	Maximum motion range in Cartesian coordinates
2	Negative limit(mm)	Minimum motion range in Cartesian coordinates
3	JOG Max	Maximum speed of jog axis operation key in Cartesian

	Speed(mm/s)	coordinate system
4	JOG Max Acc(X/s^2)	The maximum value of the movement acceleration of the jog axis operation key in the Cartesian coordinate system
5	JOG max dec(X/s^2)	The maximum value of the movement deceleration of the jog axis operation key in the Cartesian coordinate system
6	Line speed(mm/s)	The maximum value of the synthetic linear velocity at the end of the point in the Cartesian coordinate system
7	Line Max Acc(X/s^2)	The maximum value of the synthetic linear acceleration at the end of the point in the Cartesian coordinate system
8	Line Max dec(X/s^2)	The maximum value of the synthetic linear deceleration at the end of the point in the Cartesian coordinate system
9	Angular speed(deg/s)	Maximum resultant angular velocity at the end of a point in Cartesian coordinates
10	Angular acc(deg/s^2)	The maximum value of the synthetic angular acceleration at the end of the point in the Cartesian coordinate system
11	Angular dec(deg/s^2)	The maximum value of the synthetic angular deceleration at the end of the point in the Cartesian coordinate system

10.6 Plan Parameters

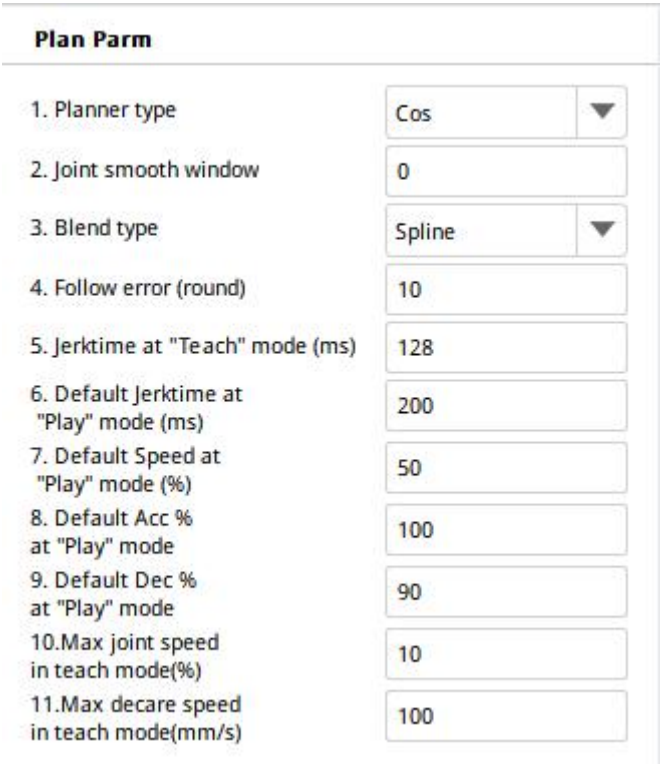
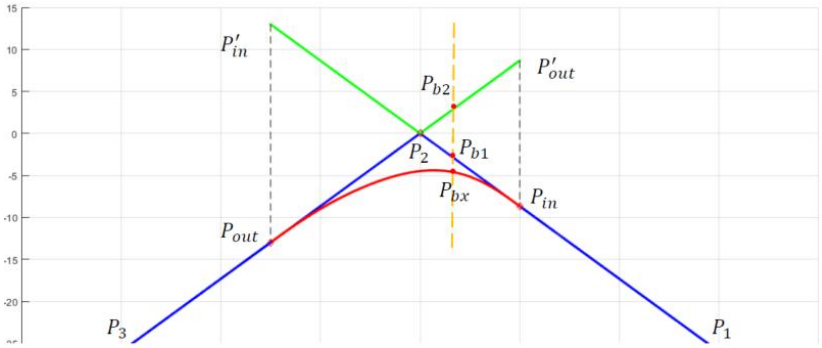


Figure 29 Planning Parameter Interface

Parameter list:

No.	Parameter Name	Parameter Description
1	Planner Type	Cos: Jerk is the trajectory planning of the Cos function Quintic: Jerk is a trajectory planner for degree 5 curves Door: Jerk is a trajectory planner for square waves SawTooth: Jerk is a zigzag trajectory planner Exp: Jerk is a trajectory planner for exponential curves Bell: Jerk is a quadratic trajectory planner Refer to the figure below for the planning curve
2	Joint Smooth Window	Output Averaging Filter Length
3	Blend Type	Fusion: The path is mixed and transitioned, the speed and acceleration do not change suddenly, the trajectory is controllable, but the speed cannot be constant, the default option Arc: Circular transition ensures that the speed of the transition section is constant and the trajectory is controllable, but it will cause a sudden change in acceleration speed. Not suitable for joint

		trajectory and pose planning. Spline: Spline transition ensures that the speed of the transition section is constant and the trajectory is controllable, but it will cause a sudden change in acceleration speed, which is smaller than the Arc transition. Not suitable for joint trajectory and pose planning VelBlend: Speed transition, that is, two motion commands are planned at the same time, and the speed is superimposed, get the new location. The trajectory shape is related to the speed when entering the Blend segment  Figure: End position path blending
4	Follow Error (round)	The difference between the position command and the actual position in the time period from the start of the movement to the actual position
5	Jerktime at “teach” mode(ms)	The acceleration time used in manual jog, if the jog value is increased, it will easily cause shake
6	Default Jerktime at “play” mode(ms)	When playing back, if the program does not call SetDyn, the default jerk time is this value, which is the jerk time in SetDyn
7	Default speed at “play” mode(%)	Default running speed of play mode when startup
8	Default acc% at “play” mode	Default acceleration speed of play mode when startup
9	Default Dcc% at “play” mode	Default deceleration speed of play mode when startup
10	Max joint speed in teach mode(%)	Limit the maximum joint movement speed in teach mode

11	Max declare speed in teach mode(mm/s)	Cartesian space movement maximum speed in teach mode

Planner Type Legend:

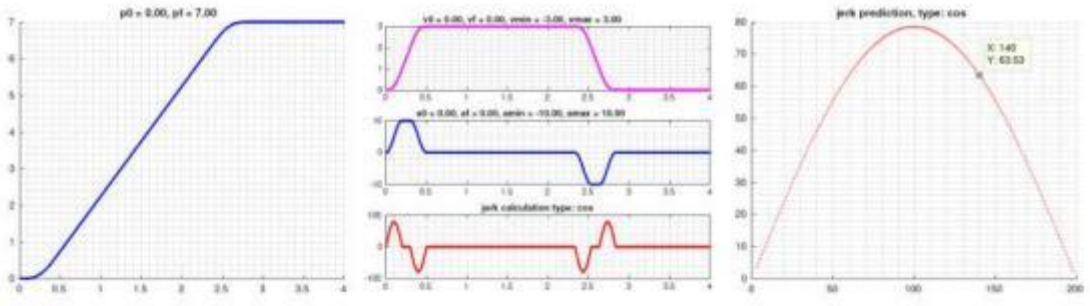


Figure 4: cos-shaped jerk

Figure 5: quintic-shaped jerk

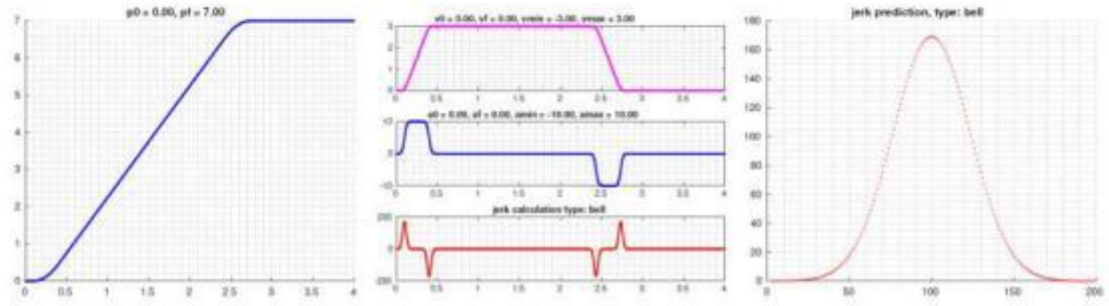


Figure 6: bell-shaped jerk

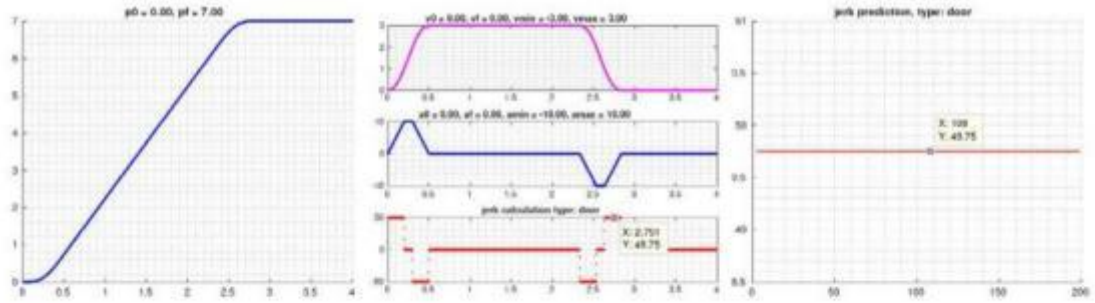


Figure 1: door-shaped jerk

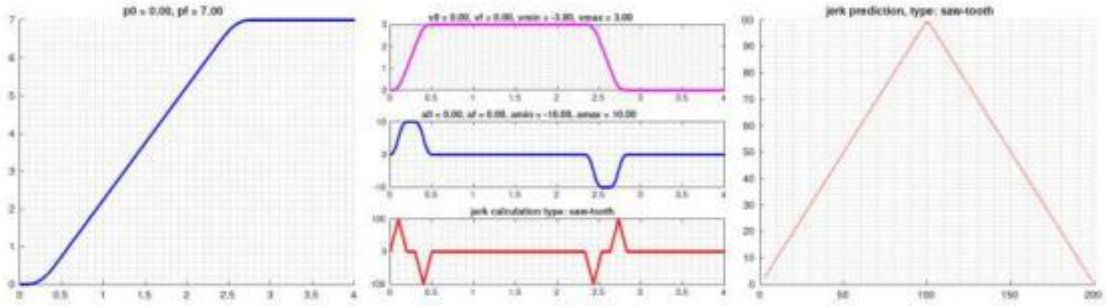


Figure 2: saw-tooth-shaped jerk

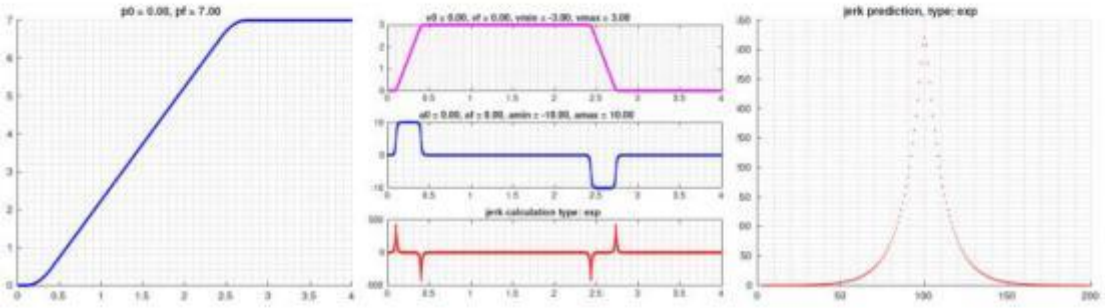
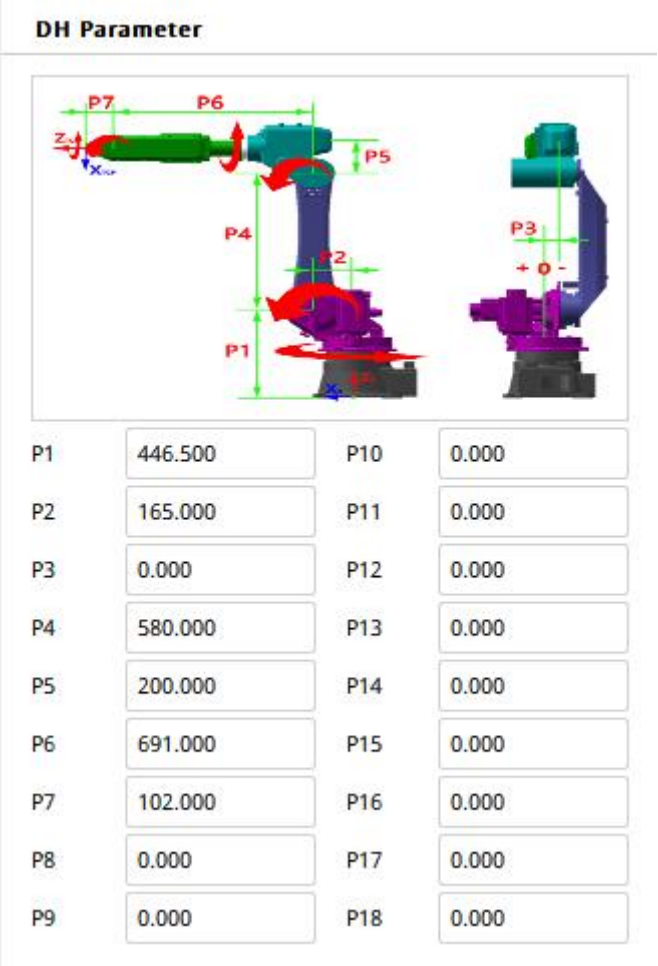


Figure 3: exp-shaped jerk

10.7 DH Parameters



P1	446.500	P10	0.000
P2	165.000	P11	0.000
P3	0.000	P12	0.000
P4	580.000	P13	0.000
P5	200.000	P14	0.000
P6	691.000	P15	0.000
P7	102.000	P16	0.000
P8	0.000	P17	0.000
P9	0.000	P18	0.000

Figure 30 DH Parameter Interface

No.	Parameter Name	Parameter Description
1	P1~P7	Please fill in the parameter information according to the diagram
2	P8/P9	Coupling ratio between axis 4 and axis 5
3	P10/P11	Coupling ratio between axis 6 and axis 4
4	P12/P13	Coupling ratio between axis 6 and axis 5
5	P14~P18	Reserved

10.8 Coupling

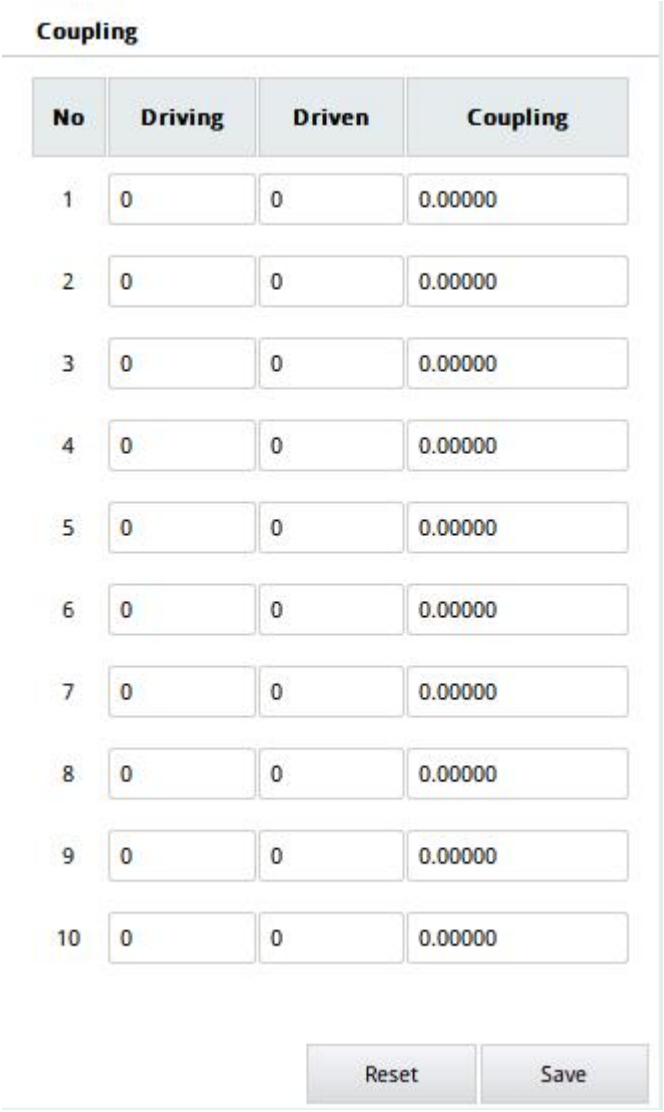


Figure 31 Coupling Ratio Setting Interface

11 General IO

General IO is mainly used to configure general signals such as anti-collision, pause, restart, alarm reset and safety door.

11.1 Input Signal

11.1.1 Interface

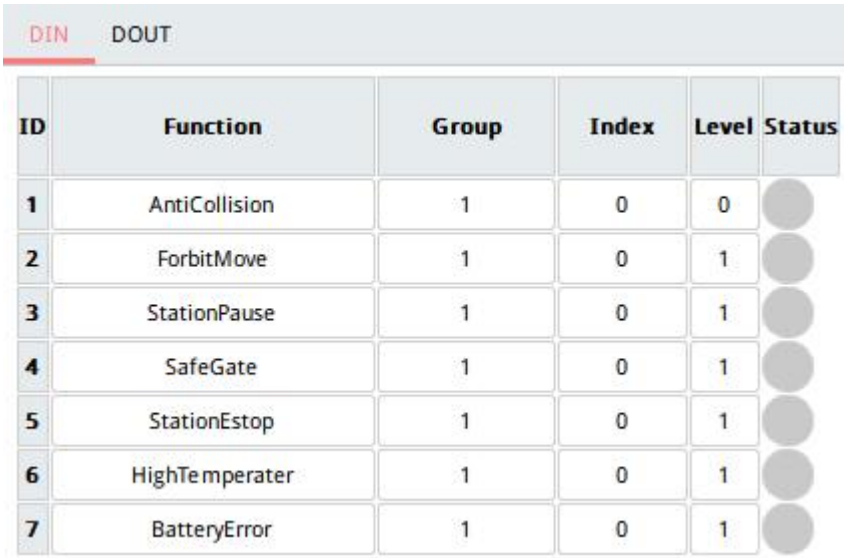


Figure 32 IO Input Interface

Group: indicates the address ID of IO module. The value of the extended IO slave address ID starts from 1.

Index: indicates the number of digits of the IO module to be used, 0 indicates not to be used, and the value of point starts from 1.

Level: Valid value, normally 1 is valid and can be negated.

Status: Points input status indicator, shows as green light when signal is triggered,shows as gray when no signal is displayed.

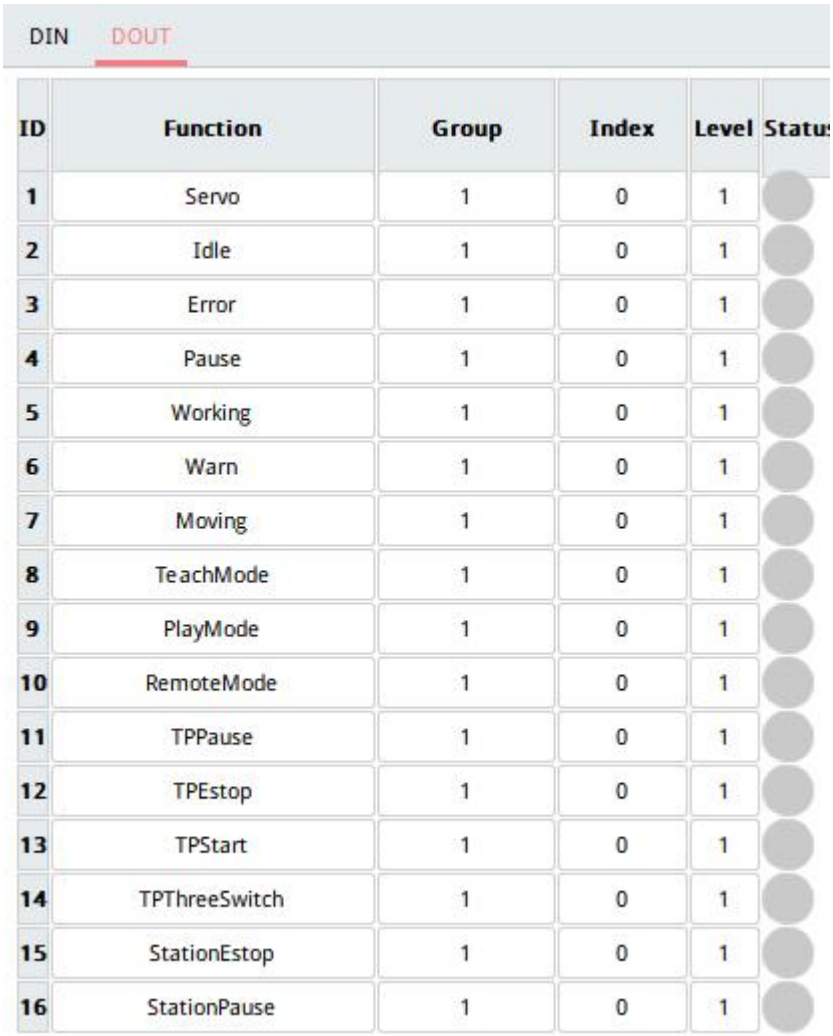
11.1.2 Function Description

No.	Input/output type	Parameter Name	Description
1	Digital input	Anti-collision	After detecting that the anti-collision signal is triggered, the system stops the movement in an emergency and prompts the anti-collision signal to trigger the alarm.
2	Digital input	ForbitMove	After the system detects that the external robot prohibition signal is triggered, the yellow warning prompts that the robot is prohibited

			from moving.
3	Digital input	SationPause	When the system detects that the pause signal is triggered, the system pauses the execution of the teach file.
4	Digital input	Safe gate	After the system detects that the emergency door signal is triggered, it will stop the movement in an emergency and prompt the emergency door to give an alarm.
5	Digital input	StationEstop	After the system detects that the emergency stop signal is triggered, the system stops the movement in an emergency and prompts the emergency stop alarm.
6	Digital input	High temperature	After the system detects that the high temperature alarm signal is triggered, it stops moving and prompts the high temperature alarm.
7	Digital input	BatteryError	After the system detects that the battery alarm signal is triggered, it stops moving and prompts the battery alarm.

11.2 Output Signal

11.2.1 Interface



ID	Function	Group	Index	Level	Status
1	Servo	1	0	1	
2	Idle	1	0	1	
3	Error	1	0	1	
4	Pause	1	0	1	
5	Working	1	0	1	
6	Warn	1	0	1	
7	Moving	1	0	1	
8	TeachMode	1	0	1	
9	PlayMode	1	0	1	
10	RemoteMode	1	0	1	
11	TPPause	1	0	1	
12	TPEstop	1	0	1	
13	TPStart	1	0	1	
14	TPThreeSwitch	1	0	1	
15	StationEstop	1	0	1	
16	StationPause	1	0	1	

Figure 33 IO Output Interface

Group: indicates the address ID of IO module. The value of the extended IO slave address ID starts from 1.

Index: indicates the number of digits of the IO module to be used, 0 indicates not to be used, and the value of point starts from 1.

Level:valid value,normally 1 is valid and can be negated.

Status: Point input status indicator, shows as green light when signal is triggered,shows as gray when no signal is displayed.

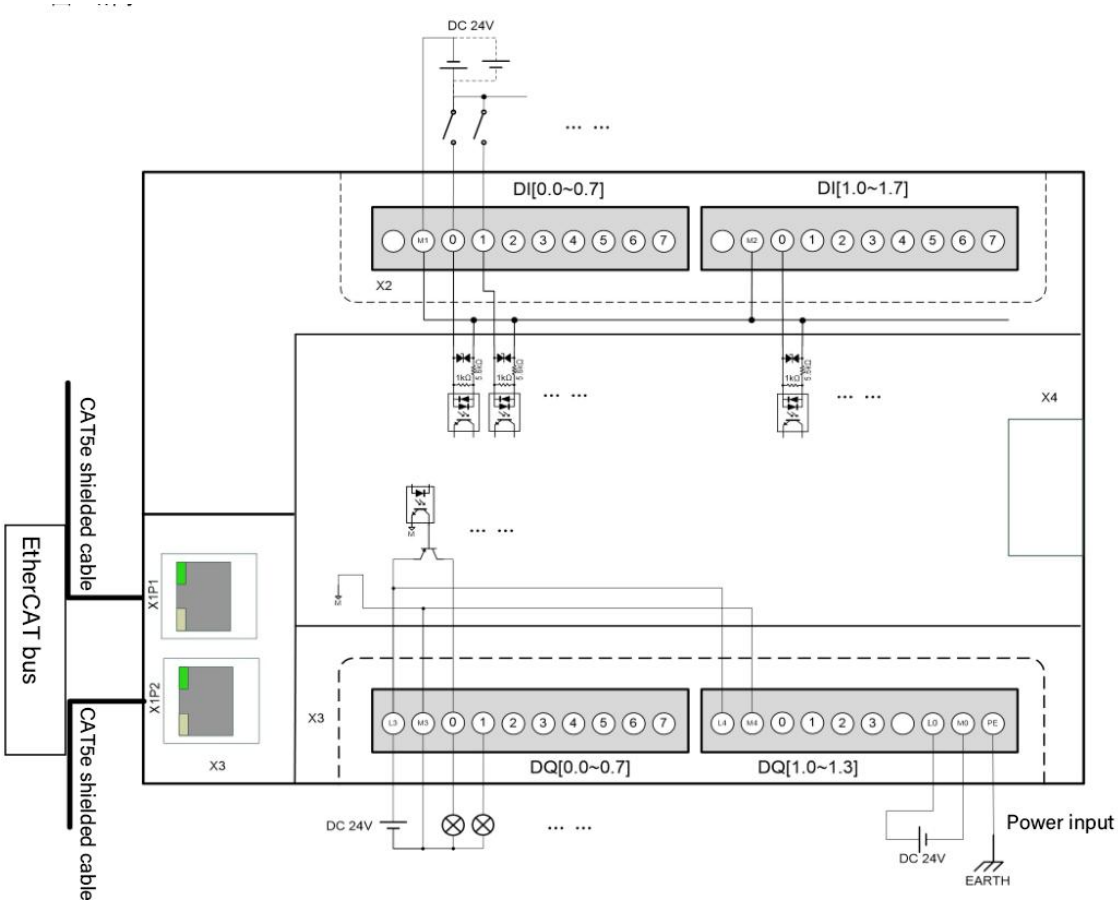
11.2.2 Function Description

No.	Input/output type	Parameter Name	Description
1	Digital output	Servo	Output signal after robot is enabled
2	Digital output	Idle	Enable in standby state
3	Digital output	Error	The robot is in a red alarm abnormal state
4	Digital output	Pause	The robot is in a paused state
5	Digital output	Working	The robot is in the state of executing the teach file.
6	Digital output	Warn	The robot is in a warning prompt state
7	Digital output	Moving	The robot is in moving state
8	Digital output	TeachMode	Output signal when handheld teach pendant is in teach mode
9	Digital output	PlayMode	Output signal when handheld teach pendant is in play mode
10	Digital output	RemoteMode	Output signal when handheld teach pendant is in remote mode
11	Digital output	TPpause	Output signal when emergency stop signal of hand-held box is triggered.
12	Digital output	TPEstop	Output signal when handheld box pause signal is triggered.
13	Digital output	TPSTART	Output signal when hand-held box start signal is triggered.
14	Digital output	TPthreeSwitch	Output signal when trigger enable button signal of three-stage switch of hand-held box.
15	Digital output	StationEstop	The workstation emergency stop input signal is received and the system is in the workstation emergency stop state.
16	Digital output	StationPause	A workstation pause input signal is received and the system is in the workstation pause state

11.3 IO Wiring

Take R51C1-EA/pro as an example, refer to the instruction manual for specific IO wiring.

R51C1-EA/pro wiring diagram



Ecat: Network cable connection

Power supply: 24V power supply, as shown in the lower right corner

Output: L3/L4(QL0/QL1) to 24V, M3/M4(QM0/QM1) to 0V, output is 24V

Input: If M1/M2(IM0/IM1) is connected to 0V, then input shall be 24V;

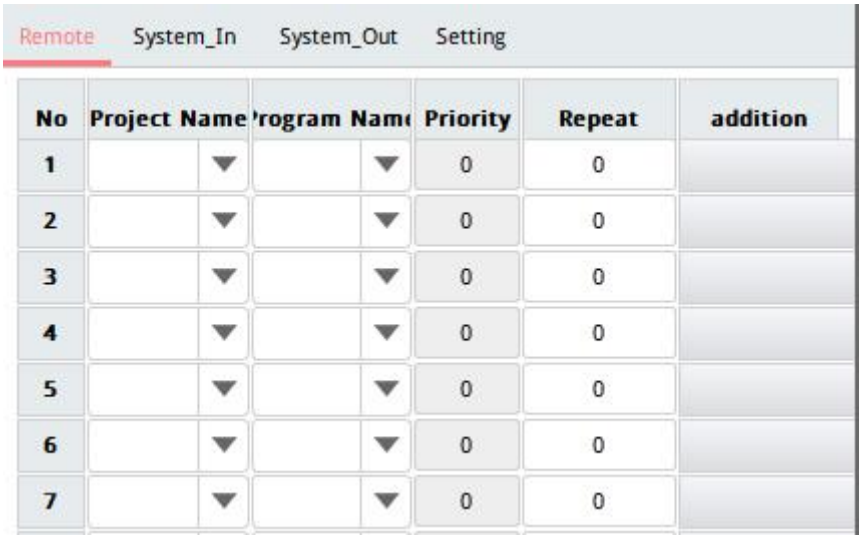
If M1/M2(IM0/IM1) is connected to 24V, then input shall be 0V;

12 Remote IO

Remote IO is mainly used in remote mode, such as remote operation of external startup file, reservation file, external startup confirmation and other signals related to remote operation.

12.1 Introduction to the Interface

12.1.1 File Settings



No.: The file number obtained by combining the remote input file bits. In binary mode, only file numbers 1~8 are valid, and in combination mode, files 1~255 are valid.

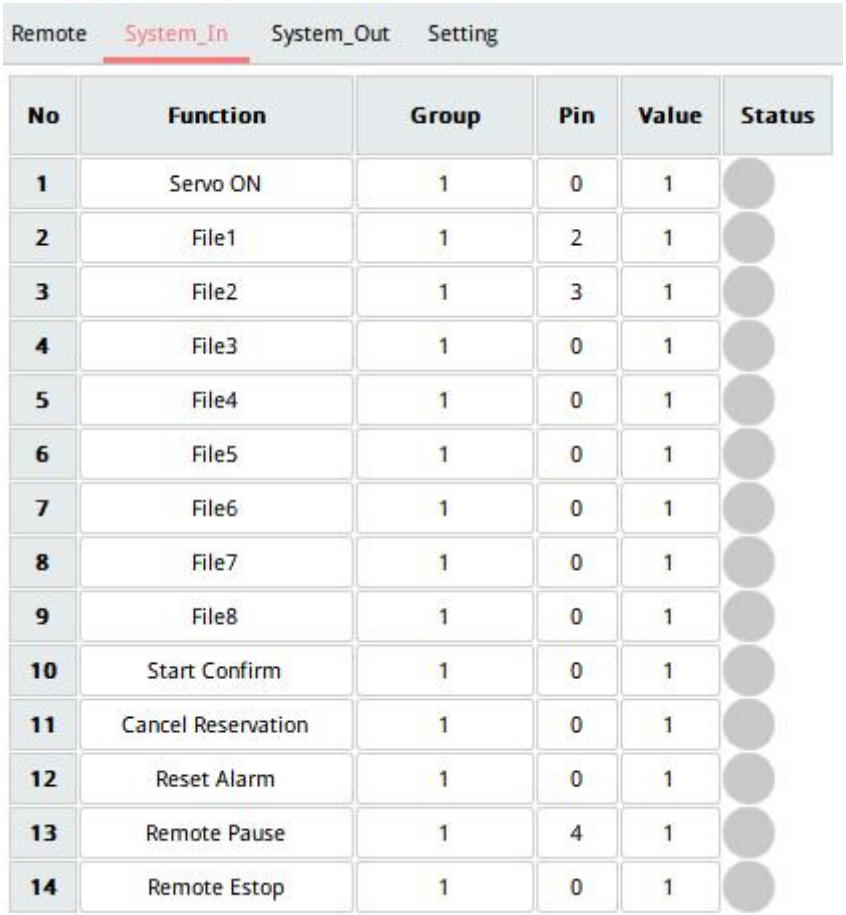
Project name and program name: the project name and program name corresponding to the file number. If the file is started, the program of the project name and program name corresponding to the file number will be executed.

Priority: display the order of file execution. Priority is 1, which is the program being executed. In single mode, priority 2~8 are the programs being reserved; Only programs with priority 1 are in combined mode.

Repeat: The number of execution times of the reserved file program is mainly used for piece counting statistics and can be manually modified.

Additional: When the number of completions of the corresponding file number reaches an integral multiple of the set number of executions, it will automatically trigger an additional task to be executed.

12.1.2 Remote IO Input



No	Function	Group	Pin	Value	Status
1	Servo ON	1	0	1	
2	File1	1	2	1	
3	File2	1	3	1	
4	File3	1	0	1	
5	File4	1	0	1	
6	File5	1	0	1	
7	File6	1	0	1	
8	File7	1	0	1	
9	File8	1	0	1	
10	Start Confirm	1	0	1	
11	Cancel Reservation	1	0	1	
12	Reset Alarm	1	0	1	
13	Remote Pause	1	4	1	
14	Remote Estop	1	0	1	

Group: Indicates the address ID of the IO module, and the value of the address ID of the extended IO slave station starts from 1.

Pin: Indicates the number of the module used, 0 means not used, and the value of the dot starts from 1.

Value: valid value,normally 1 is valid, can be negated.

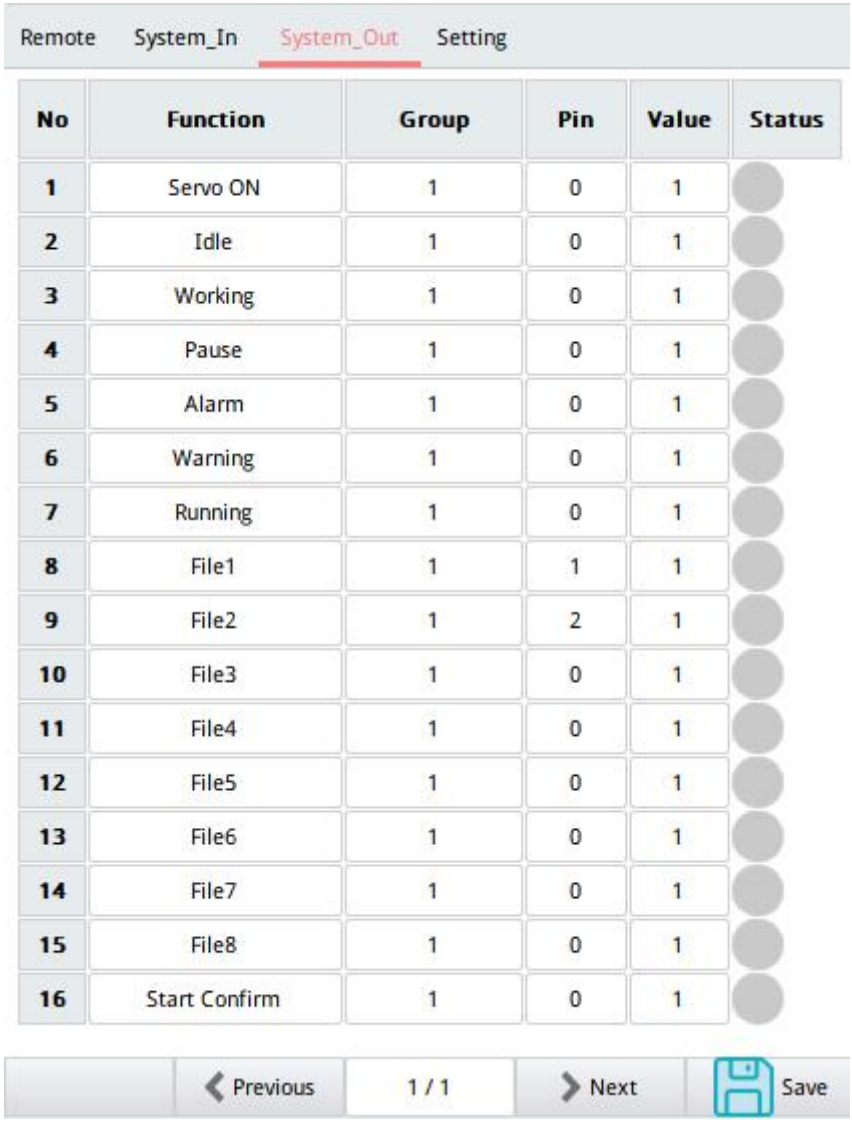
Status: Point input status indicator, the light will be green when the signal is triggered, and it will be black when there is no signal.

Function Description:

No.	Input/output type	Parameter Name	Description
1	Remote IO Input	Servo On	In the remote IO control mode, the system will enable the robot when it receives this signal.
2	Remote IO Input	File bit 1	File number input bit 1, it represents the file bit 1 if it is a single mode; it represents the first bit of 8421 encoding if it is a combination mode.

3	Remote IO Input	File Bit 2	File number input bit 2, it represents the file bit 2 if it is a single mode; it represents the second bit of 8421 encoding if it is a combination mode.
4	Remote IO Input	File bit 3	File number input bit 3, it represents the file bit 3 if it is a single mode; it represents the third bit of 8421 encoding if it is a combination mode.
5	Remote IO Input	File bit 4	File number input bit 4, it represents the file bit 4 if it is a single mode; it represents the 4th bit of 8421 encoding if it is a combination mode.
6	Remote IO Input	File Bit 5	File number input bit 5, it represents the file bit 5 if it is a single mode; it represents the 5th bit of 8421 encoding if it is a combination mode.
7	Remote IO Input	File bit 6	File number input bit 6, it represents the file bit 6 if it is a single mode; it represents the 6th bit of 8421 encoding if it is a combination mode.
8	Remote IO Input	File bit 7	File number input bit 7, it represents the file bit 7 if it is a single mode; it represents the 7th bit of 8421 encoding if it is a combination mode.
9	Remote IO Input	File bit 8	File number input bit 8, it represents the file bit 8 if it is a single mode; it represents the 8th bit of 8421 encoding if it is a combination mode.
10	Remote IO Input	Startup Confirmation	File start signal in combination mode. This signal is invalid in single mode.
11	Remote IO Input	Cancel an appointment	Cancel all appointments (including additional tasks corresponding to execution or appointment files)
12	Remote IO Input	Alarm Reset	Clear alarm signal
13	Remote IO Input	Remote Pause	Pause robot motion
14	Remote IO Input	remote scram	Emergency stop signal

12.1.3 Remote IO Output



Group: Indicates the address ID of the IO module, and the value of the address ID of the extended IO slave station starts from 1.

Pin: Indicates the number of the module used, 0 means not used, and the value of the dot starts from 1.

Value: valid value,normally 1 is valid, can be negated.

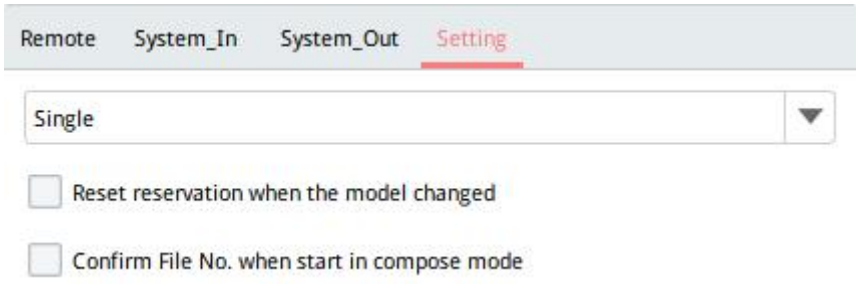
Status: Point input status indicator, the light will be green when the signal is triggered, and it will be black when there is no signal.

Description:

No.	Input/output type	Parameter Name	Description
1	Remote IO Output	Servo On	Output signal after robot is enabled
2	Remote IO Output	Idle	Enable in standby state
3	Remote IO Output	Working	The robot is in the state of executing the

			teach file.
4	Remote IO Output	Pause	The robot is in a paused state
5	Remote IO Output	Alarm	The robot is in a red alarm abnormal state
6	Remote IO Output	Warning	The robot is in a warning prompt state
7	Remote IO Output	Running	the state in which that robot is in motion
8	Remote IO Output	File bit 1	File bit 1 is in reservation
9	Remote IO Output	File Bit 2	File bit 2 is in reservation
10	Remote IO Output	File bit 3	File bit 3 is in reservation
11	Remote IO Output	File bit 4	File bit 4 is in reservation
12	Remote IO Output	File Bit 5	File bit 5 is in reservation
13	Remote IO Output	File bit 6	File bit 6 is in reservation
14	Remote IO Output	File bit 7	File bit 7 is in reservation
15	Remote IO Output	File bit 8	File bit 8 is in reservation
16	Remote IO Output	Start Confirm	Remote input enable confirm is pressed

12.1.4 Setting

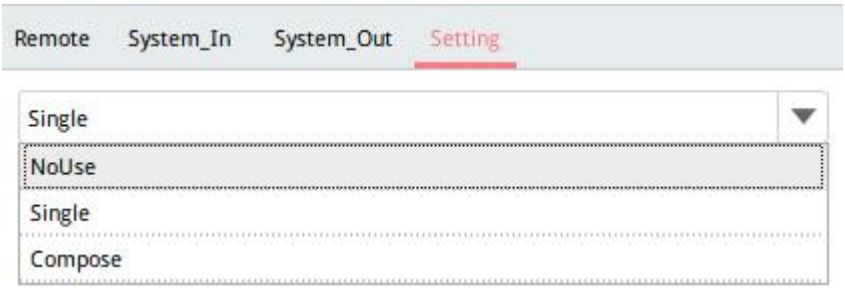


Reset reservation when the model changed and **Confirm File No. When start in compose mode** are backup functions.

If **Reset reservation when the model changed** is checked, all reservations will be cleared when the mode of the handheld box is switched; the new reservation is required when switching back to the remote mode.

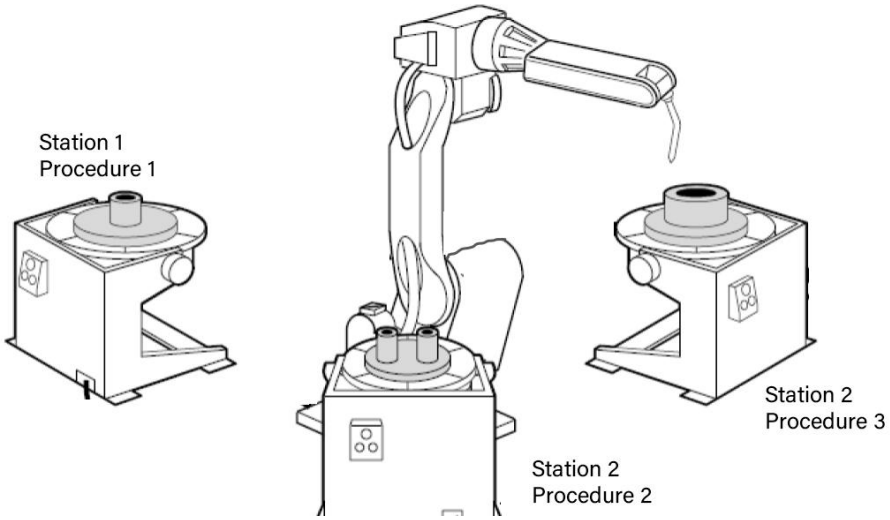
If the function of **Confirm File No. When start in compose mode** is checked, the file position signal must be detected when the combination mode is started (that is, the corresponding file signal (combination) is continuously input), the confirmation signal is activated, and the file position input signal can only be canceled after the station file starts to execute.

There are two working modes of remote IO: 1. No used 2. Single 3. Compose (combination mode).



12.2 Use in Single Mode

12.2.1 Application



The three stations shown in the above figure are used to process different workpieces respectively, and the following preparations shall be made in advance:

On the station 1, execute program 1 of workpiece 1.

On the station2, execute the program 2 of the workpiece 2.

On the station 3, execute the program 3 of the workpiece 3.

During reproduction, switch to “REMOTE” mode, after preparing workpiece 1, press the start button on station 1, and the robot executes program 1. When executing Procedure 1, prepare Workpiece 2 and Workpiece 3, and then press the start buttons on station 2 and station 3 in turn. At this time, even if the program 1 is still running, since the start buttons have been pressed, the program 2 and the program 3 have been reserved in sequence, and the programs will be executed in sequence according to the reserved order.

The system can realize the reservation of 8 stations at most.

12.2.2 Operation Steps

12.2.2.1 Setting Remote Work Mode

Enter [Remote] - [Setting] - set it to [single] mode in [Remote IO Mode].

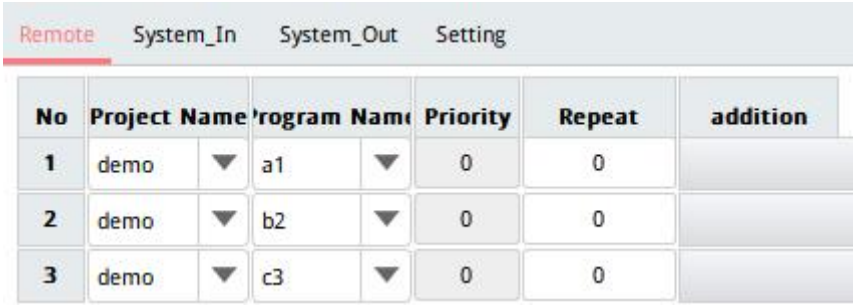
12.2.2.2 Setting Remote File Names

Step 1 Create a workstation program.

enter [menu] - [project]-create a demo project, and create a new station program with program names of a1, b2and c3.

Step 2 Set up remote files.

Enter the [Remote] interface. The settings are as follows:



12.2.2.3 Setting Remote IO Input

Connect to ethercat IO and use input points 1.1~1.15 for configuration.

Remotely operate the robot to perform 3 stations' work, enable signal must be set for file 1, file2, and file 3. Cancel reservation, alarm reset, remote pause and remote emergency stop can be set as required.

Remote System_In System_Out Setting

No	Function	Group	Pin	Value	Status
1	Servo ON	1	0	1	
2	File1	1	1	1	
3	File2	1	2	1	
4	File3	1	3	1	
5	File4	1	0	1	
6	File5	1	0	1	
7	File6	1	0	1	
8	File7	1	0	1	
9	File8	1	0	1	
10	Start Confirm	1	0	1	
11	Cancel Reservation	1	0	1	
12	Reset Alarm	1	0	1	
13	Remote Pause	1	4	1	
14	Remote Estop	1	0	1	

12.2.2.4 Setting Remote IO Output

Connect ethercat IO, and use output points 1.1~1.16 for configuration.

The remote IO output signal is set according to the external status.

Remote System_In System_Out Setting

No	Function	Group	Pin	Value	Status
1	Servo ON	1	0	1	
2	Idle	1	0	1	
3	Working	1	0	1	
4	Pause	1	0	1	
5	Alarm	1	0	1	
6	Warning	1	0	1	
7	Running	1	0	1	
8	File1	1	1	1	
9	File2	1	2	1	
10	File3	1	0	1	
11	File4	1	0	1	
12	File5	1	0	1	
13	File6	1	0	1	
14	File7	1	0	1	
15	File8	1	0	1	
16	Start Confirm	1	0	1	

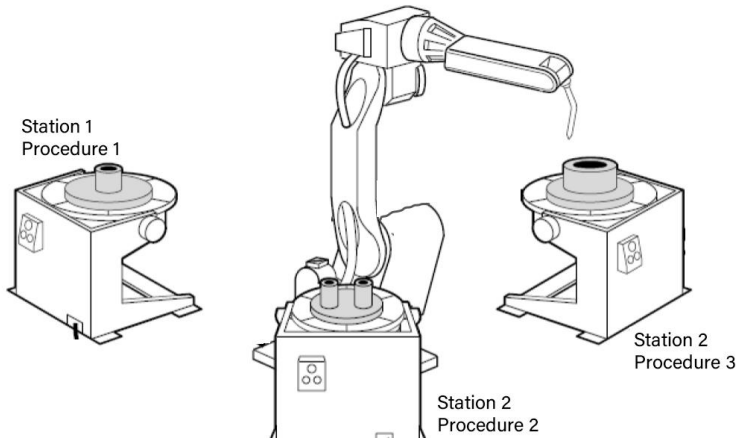
12.2.2.5 Start Reservation

For example, the procedures of station 1, station 2 and station 3 are executed in sequence, and the workflow is as follows:

After completing the setting of step 1-4, switch to remote mode—input signal 1.4 to enable—wait for normal enabling of robot—input signal 1.1 to station 1—station 1 is working, wait for execution—input signal 1.2 to station 2—station 2 is reserved, wait for execution—input signal 1.3 to station 3—station 3 is reserved, wait for execution.

12.3 Combined Mode Usage

12.3.1 Application



The three stations shown in the above figure are used to process different workpieces respectively, and the following preparations shall be made in advance:

On the tool 1, execute the program 1 of the workpiece 1.

On the tool 2, execute the program 2 of the workpiece 2.

On the tool 3, execute the program 3 of the workpiece 3.

Only two file bits are available.

The combined mode system enables up to 255 process files.

12.3.2 Operation Steps

12.3.2.1 Setting Remote Work Mode

Enter [Remote] - [Setting] - [Remote IO] Mode to set to [Compose] mode.

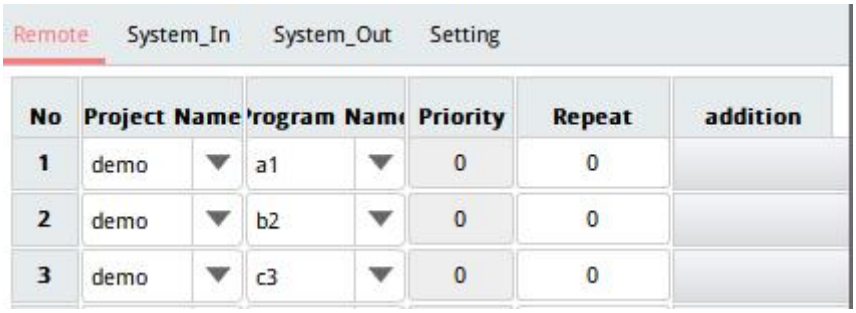
12.3.2.2 Set Remote File Names

Step 1 Create a workstation program

enter [menu] - [project] - create a demo project, and create a new station program with program names of a1, b2and c3.

Step 2 Set up remote files.

Enter the [Remote] interface and select the [Remote] property page. The settings are as follows:



12.3.2.3 Set remote IO Input

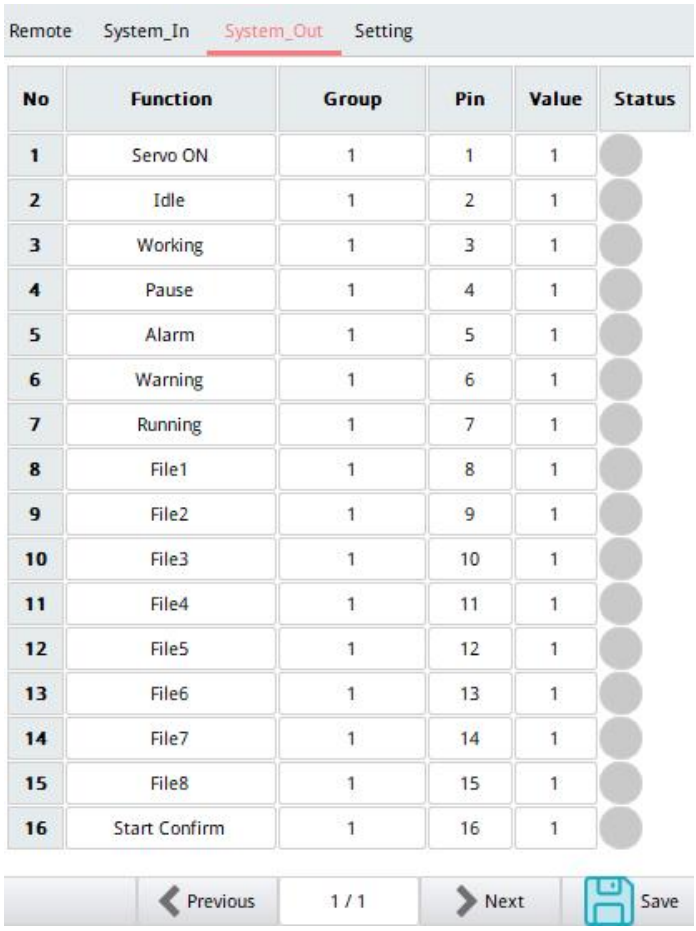
Operating the robot to work remotely at 3 stations . **File 1**, **File 2**, **File 3** and **Servo ON** signal must be set. **Cancel reservation**,**Reset alarm, Remote Pause** and **Remote Estop** can be set as required.

No	Function	Group	Pin	Value	Status
1	Servo ON	1	4	1	
2	File1	1	1	1	
3	File2	1	2	1	
4	File3	1	3	1	
5	File4	1	0	1	
6	File5	1	0	1	
7	File6	1	0	1	
8	File7	1	0	1	
9	File8	1	0	1	
10	Start Confirm	1	5	1	
11	Cancel Reservation	1	6	1	
12	Reset Alarm	1	7	1	
13	Remote Pause	1	8	1	
14	Remote Estop	1	9	1	

12.3.2.4 Setting remote IO Output

Connect ethercat IO, and use output points 1.1~1.16 for configuration.

The remote IO output signal is set according to the external status.



12.3.2.5 Start an appointment

For example, the procedures of station 1, station 2 and station 3 are executed in sequence, and the workflow is as follows:

Start the work flow of station 1: after completing the setting of steps 1-4, switch to remote mode--give the enable input signal 1.4--wait for the robot to be enabled normally--set the file 1 signal 1.1--the reservation of station 1 is completed--give the start confirmation signal 1.5--the robot starts to work in station 1(demo.a1), waiting for execution--the work is completed.

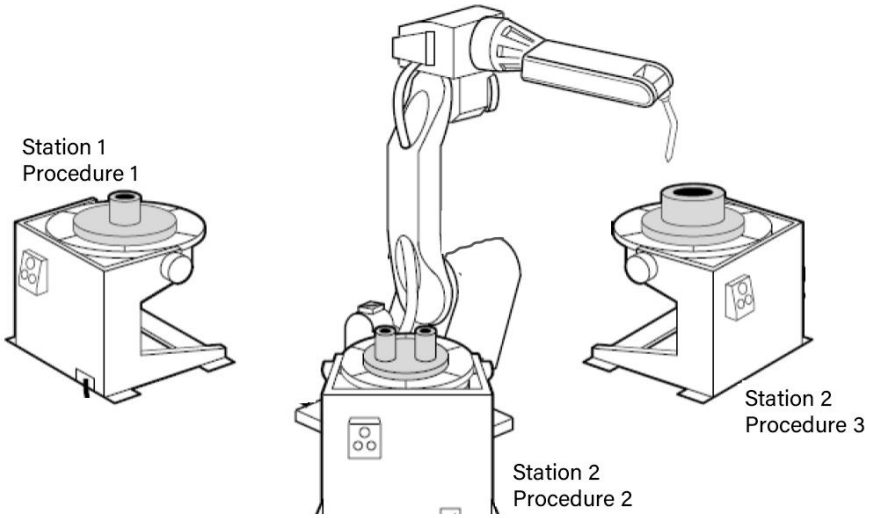
Start the work flow of station 2: after completing the setting of steps 1-4, switch to remote mode--input enable signal 1.4--wait for normal enable of robot--input signal 1.2 to file 2--reservation of station 2 is completed--start confirmation signal 1.5--robot starts working at station 2 (demo.b2), waiting for execution--work is completed.

Start the work flow of station 3: after completing the setting of steps 1-4, switch to remote mode--input enable signal 1.4--wait for normal enabling of robot--input signal 1.1 to file 1, input signal 1.2 to file 2 (file 1 and file 2 are combined to form station 3 file)--reservation of station 3 is

completed--send start confirmation signal 1.5--robot starts working at station 3(demo.c3), wait for execution--work is completed.

12.4 Use of Additional Tasks

12.4.1 Application Scenarios



The two stations shown in the above figure are used to process different workpieces respectively, and the following preparations shall be made in advance:

On tool 1, execute program 1 of workpiece 1

On the tool 2, execute the program 2 of the workpiece 2

Additional tasks can be performed based on individual mode and combination mode, and the system can add up to 256 additional tasks.

12.4.2 Operation Steps

12.4.2.1 Set Remote Work Mode

Enter [Remote] - [Setting] - [Remote IO Mode], which can be set to [Single] or [Compose]mode as required.

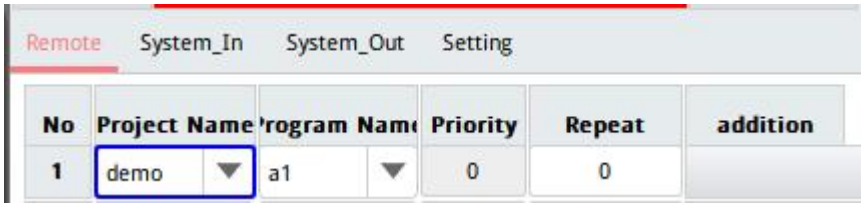
12.4.2.2 Set Remote File Names

Step 1 Create a workstation program

Enter [menu]-[project] -create a demo project, and create a new station program with program names of a1,b2and c3.

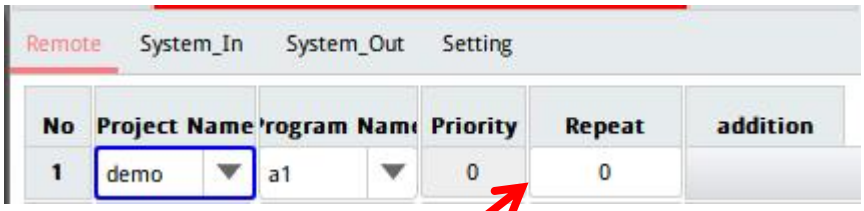
Step 2 Set up remote files

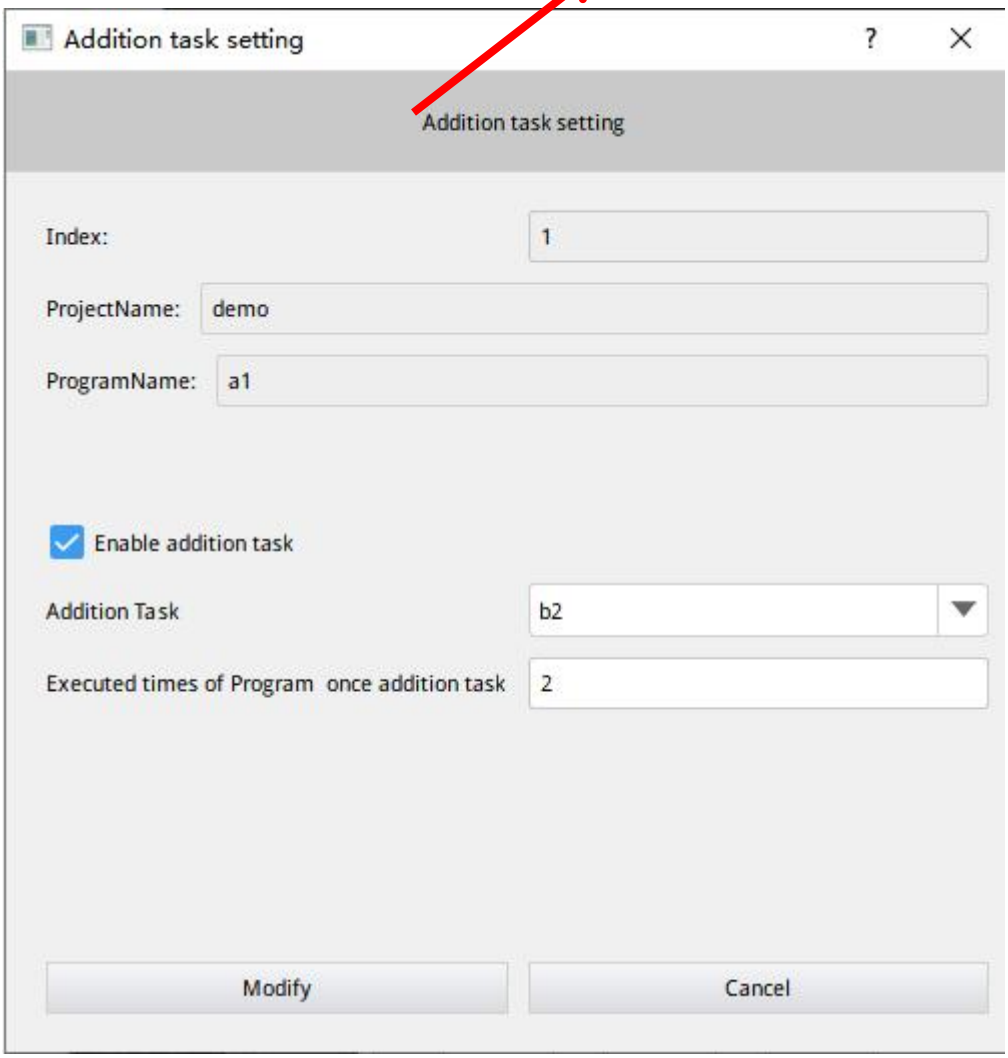
Enter the [Remote] interface and select the [remote] Settings property page. The settings are as follows:



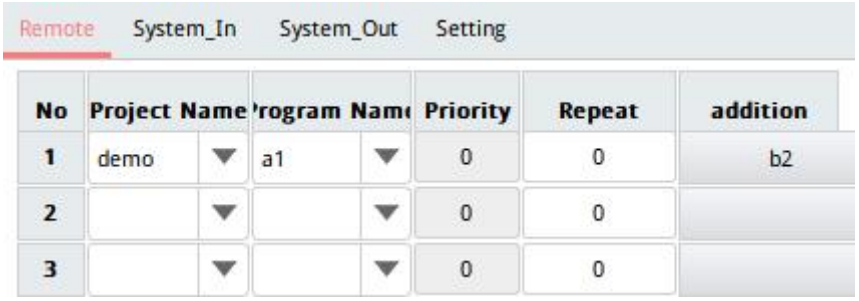
Step 3 Set up additional tasks

Click the [Additional] button in the line corresponding to the document number to enter the setting page, and the settings are as follows:





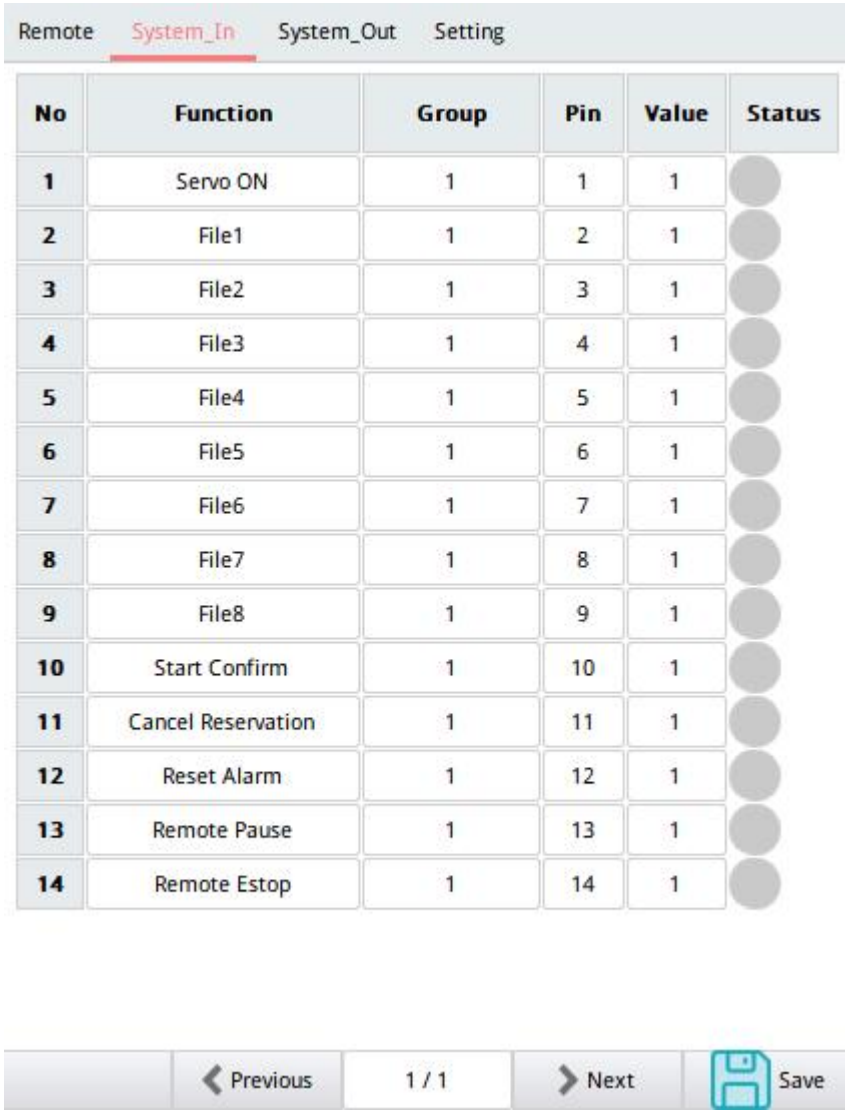
To enable the additional task function, you need to check before [enable additional task] option, select the [Addition task] to be added, input the [executed times of program once addition task], and click [Modify] to complete the configuration.



12.4.2.3 Set Remote IO Input

Connect to ethercat IO and use input points 1.1~1.15 for configuration.

Operating the robot to work remotely at 3 stations . **File 1**, **File 2**, **File 3** and **Servo ON** signal must be set. **Cancel reservation**,**Reset alarm, Remote Pause** and **Remote Estop** can be set as required.



12.4.2.4 Set Remote IO Output

Connect ethercat IO, and use output points 1.1~1.16 for configuration.

The remote IO output signal is set according to the external status.

12.4.2.5 Start Additional Task Operation

Example: workflow of additional task (file number b2):

In the individual mode/continuous mode, the station 1(file number a1) is started twice, and the system automatically starts the additional task station 2(file number b2) after the completion of the accumulated two jobs.

Note: The additional task is automatically started each time the number of times the corresponding file number is executed is an integer multiple.

13 Other Functions

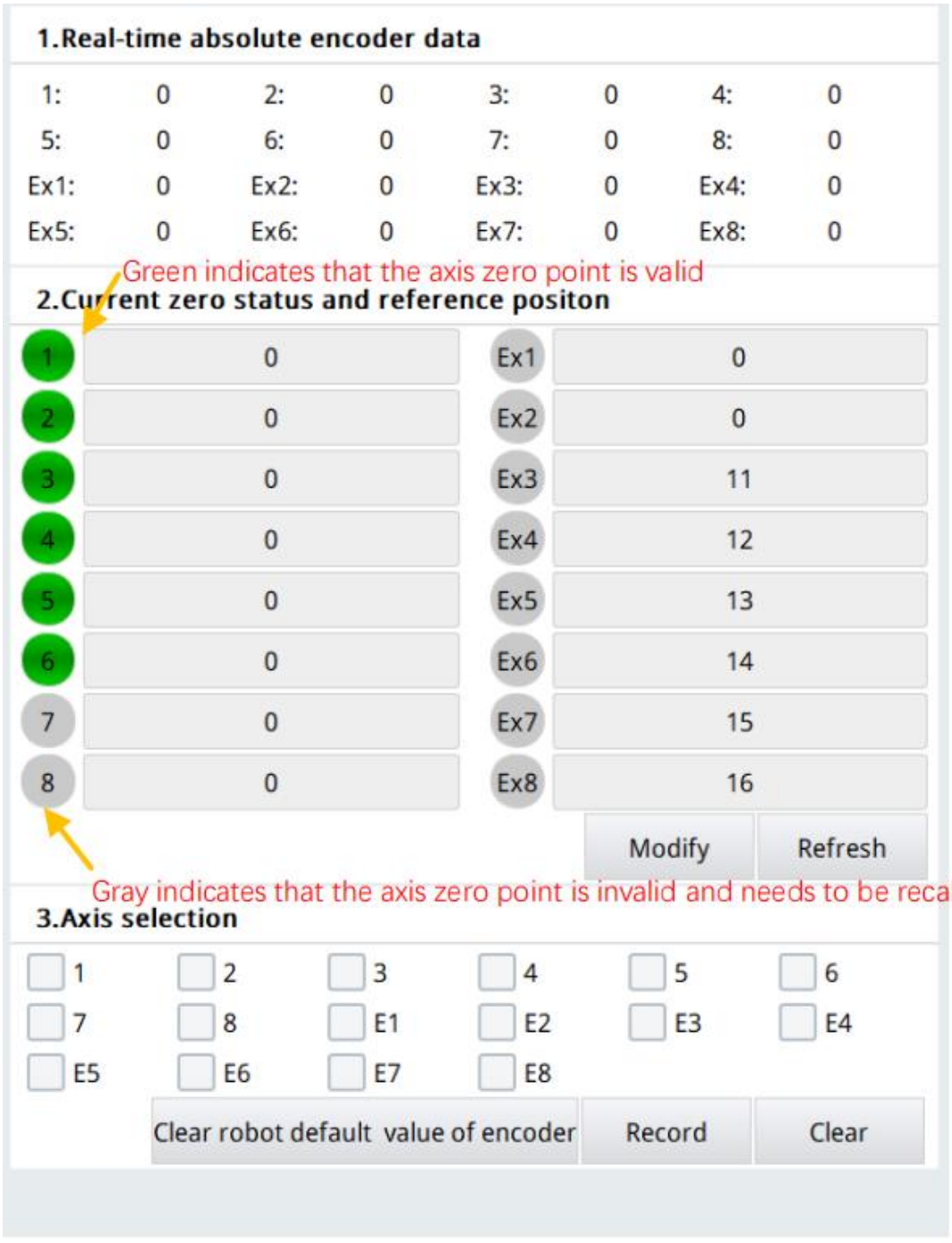
13.1 Zero Calibration

Attention
1. Without the calibration of the origin position, the teaching and playback operations cannot be performed. In a system using multiple robots, each robot must perform home position calibration. 2. When there is a coupling relationship between the joint axes, for example, there is a coupling relationship between the fifth axis and the sixth axis of common robots. When the fifth axis must be at the zero position, the zero point data recorded by the sixth axis will be valid; otherwise, the zero point data recorded by the six-axis is invalid. Therefore, the zero position data of the sixth axis must be recorded when the fifth axis is at zero position. If there is no coupling relationship, each axis can calibrate the zero position independently, and the respective zero positions will not affect the zero positions of other joints. 3. When all the used axes (main body axis and auxiliary extension axis) have completed the zero calibration, the "all" indicator light on the zero calibration interface turns green, indicating that the robot has completed the zero data calibration, and the robot can Carry out motion in Cartesian space.

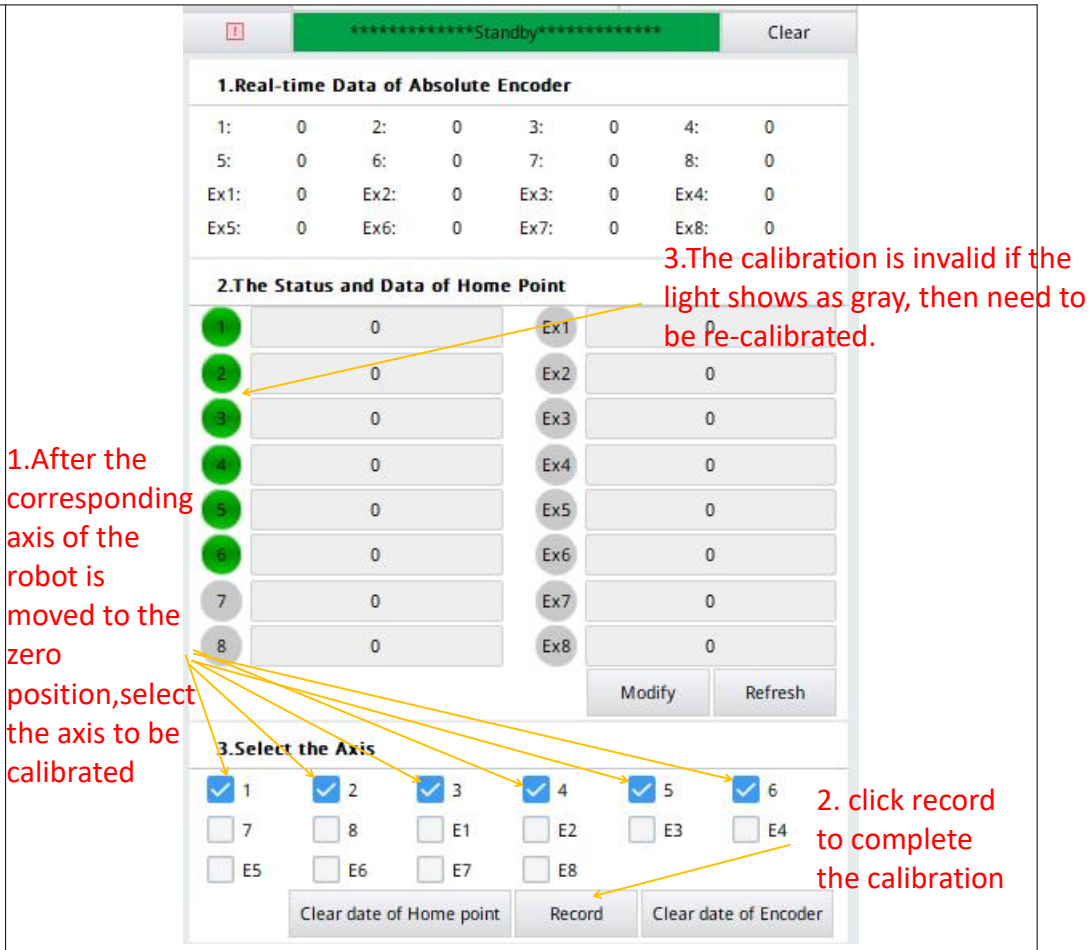
After clearing the encoder zero drift alarm, the mechanical zero position calibration and software record calibration must be required for each axis of the robot immediately.

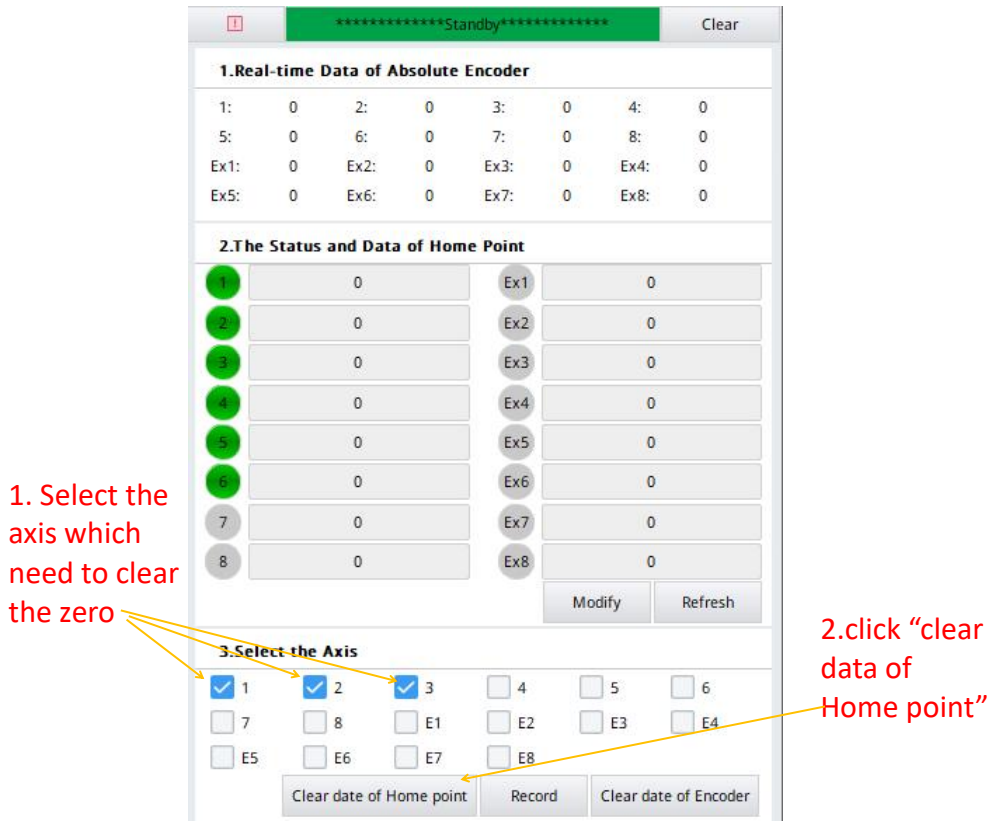
Mechanical zero calibration: perform calibration according to the following method, i.e., make each axis run to mechanical reference zero point through single-axis movement.

Software record calibration: When all 6 axes have returned to the mechanical reference zero point through single-axis movement, it is necessary to enter the factory setting of the software and re-record the zero-point position. Ensure that the zero positions of the software and hardware correspond. After that, each time the robot returns to zero, it will return to that position.

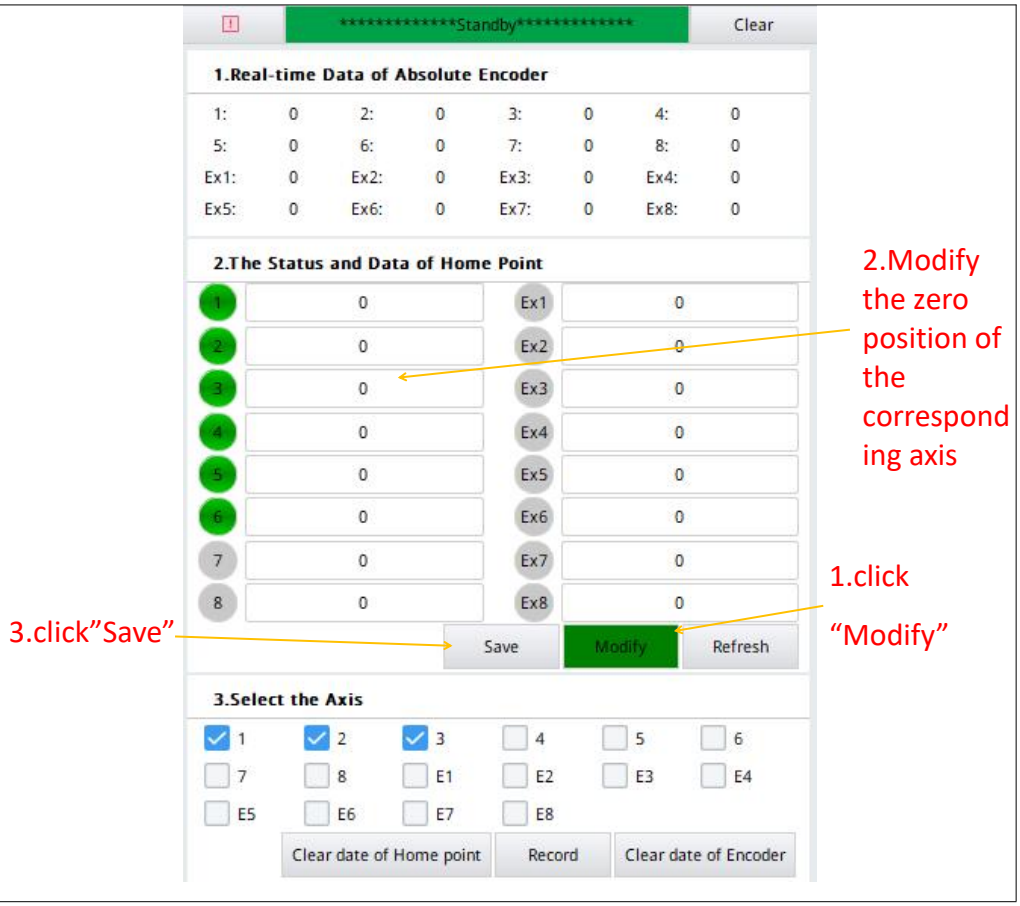


Item	Procedure

<table>
<tr>
<td>Record zero</td>
<td>


</td>
</tr>
<tr>
<td>Absolute encoder reset (the selected axis of the zero data to 0)</td>
<td>


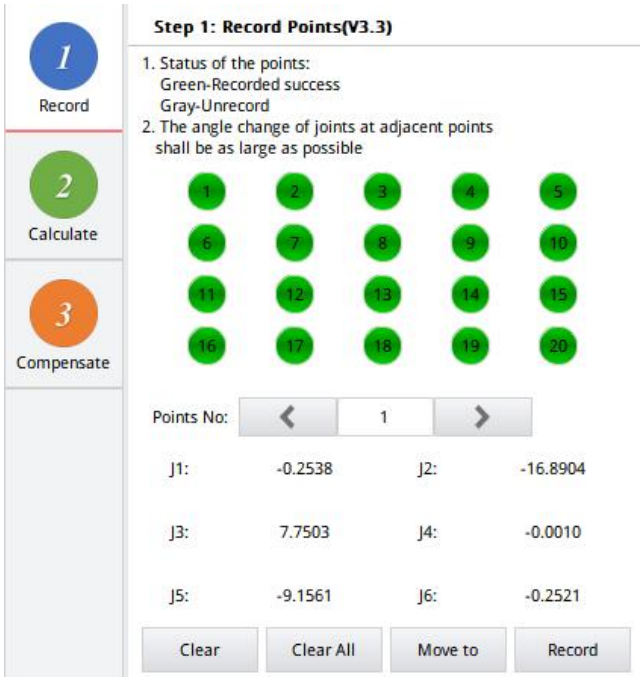
</td>
</tr>
<tr>
<td>Manually modify</td>
<td></td>
</tr>
</table>

zero point (not recommended)	

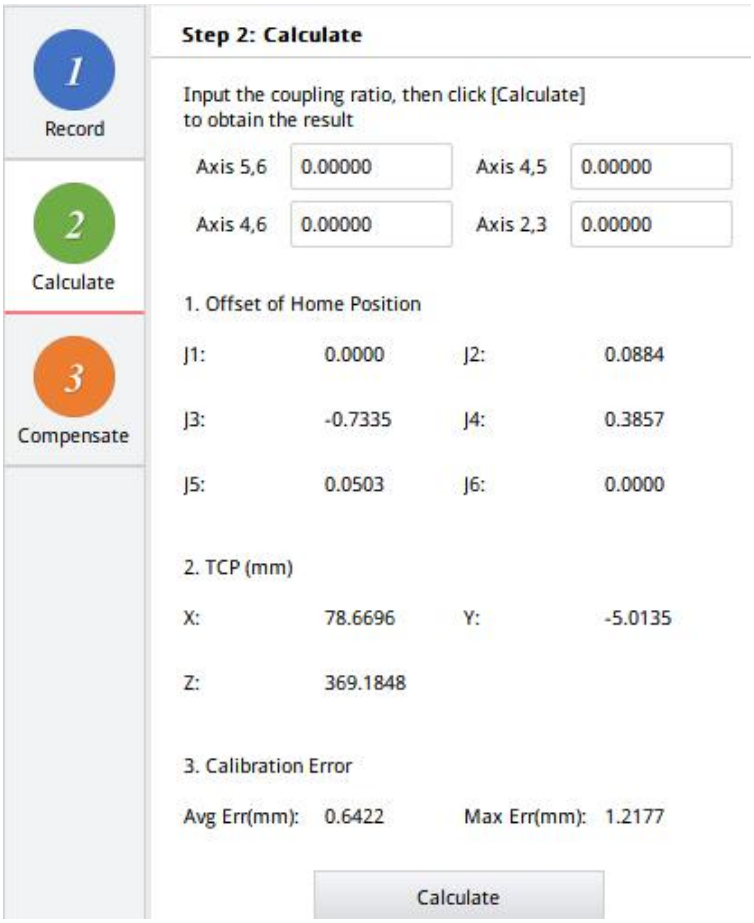
13.2 20-Point Method

The 20-point method is used to compensate for the zero point and the tool coordinate system.

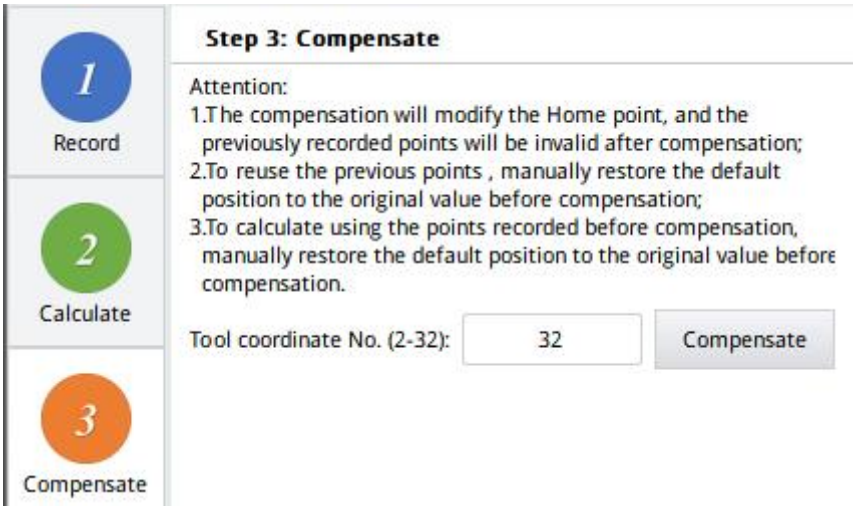
step 1: Step 1: Change the different poses of the robot and use the end of the tool to align the tip and record the position points. The joint angle of each position point changes as much as possible.


1
Record
2
Calculate
3
Compensate
Step 1: Record Points(V3.3)
1. Status of the points:
Green-Recorded success
Gray-Unrecord
2. The angle change of joints at adjacent points
shall be as large as possible
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Points No:
1
J1:
-0.2538
J2:
-16.8904
J3:
7.7503
J4:
-0.0010
J5:
-9.1561
J6:
-0.2521
Clear
Clear All
Move to
Record

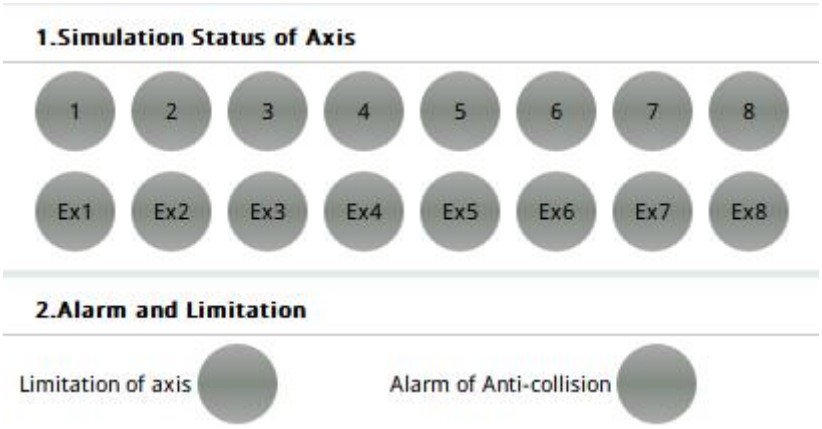
Step 2: Click [Calculate]



Step 3: Click [Compensate]



13.3 Exception Handling



Green light means active, gray means normal.

13.4 Authority Management

13.4.1 User Switching

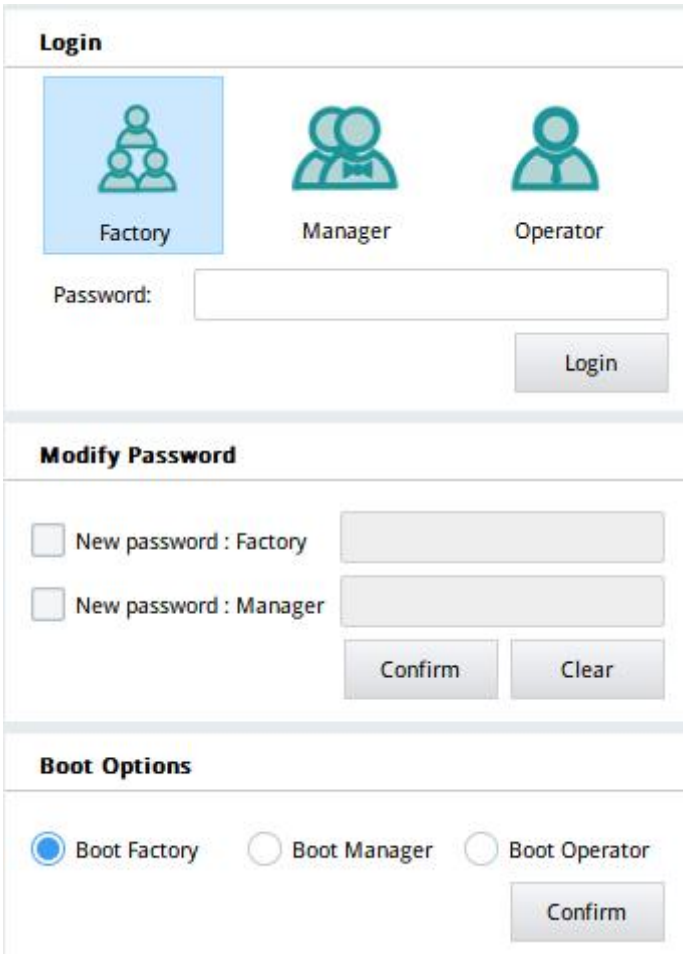


Figure 34 User Switching Interface

13.4.2 Hardware Testing

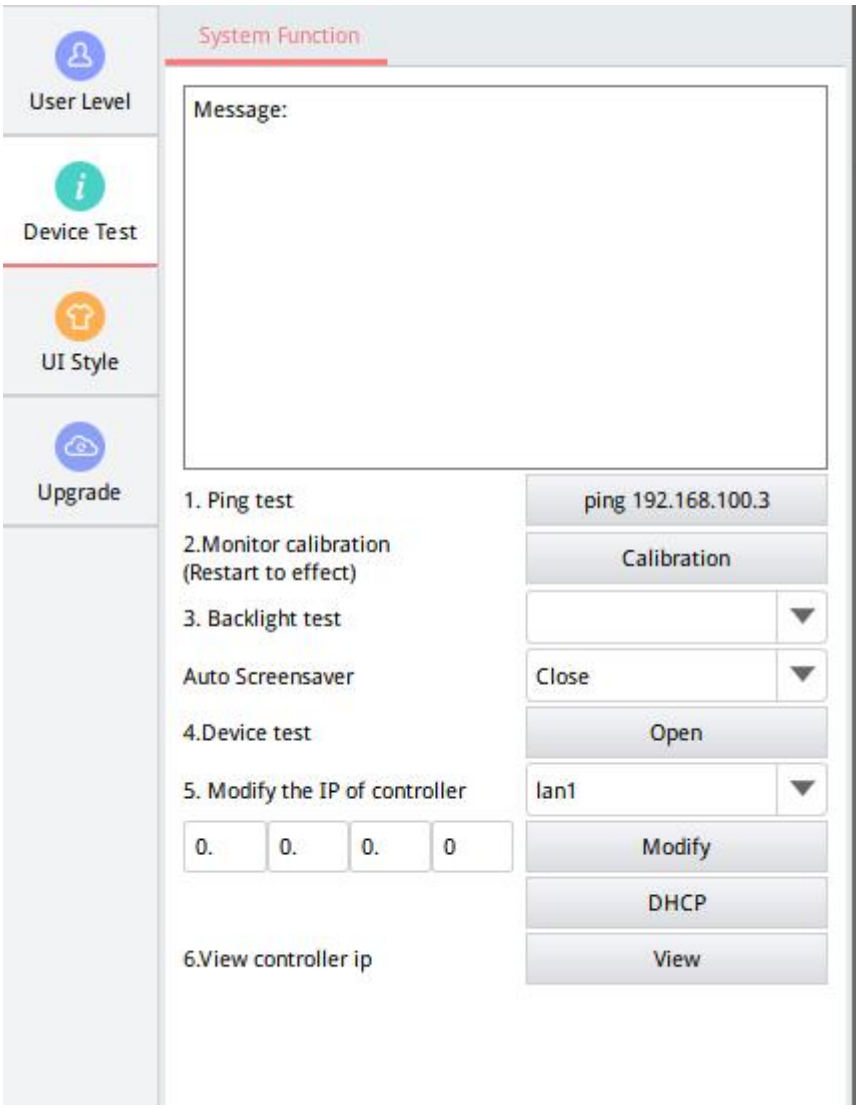


Figure 35 System Version Interface

Parameter list:

No.	Parameter Name	Parameter Description
1	Ping Test	Check whether the communication between the teach pendant and the controller is normal
2	Monitor calibration(restart to effect)	Calibrate the handheld touch screen, and the handheld will automatically restart after the calibration procedure is completed.
3	Backlight Test Auto Screensaver	Adjust the backlight brightness of hand-held lcd Automatic screensaver (on/off)
4	Device test	[Open hand-held Hardware Test]
5	Modify the IP of	Modify LAN1, LAN2(LAN3) network port IP

	controller	
6	View controller IP	View controller IP

13.4.3 Interface Style

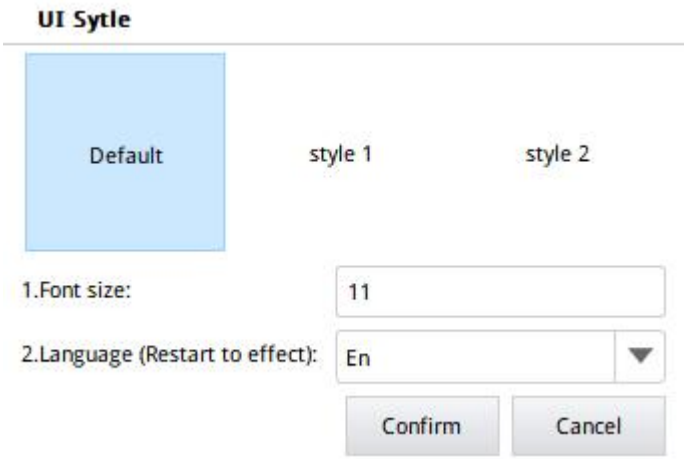
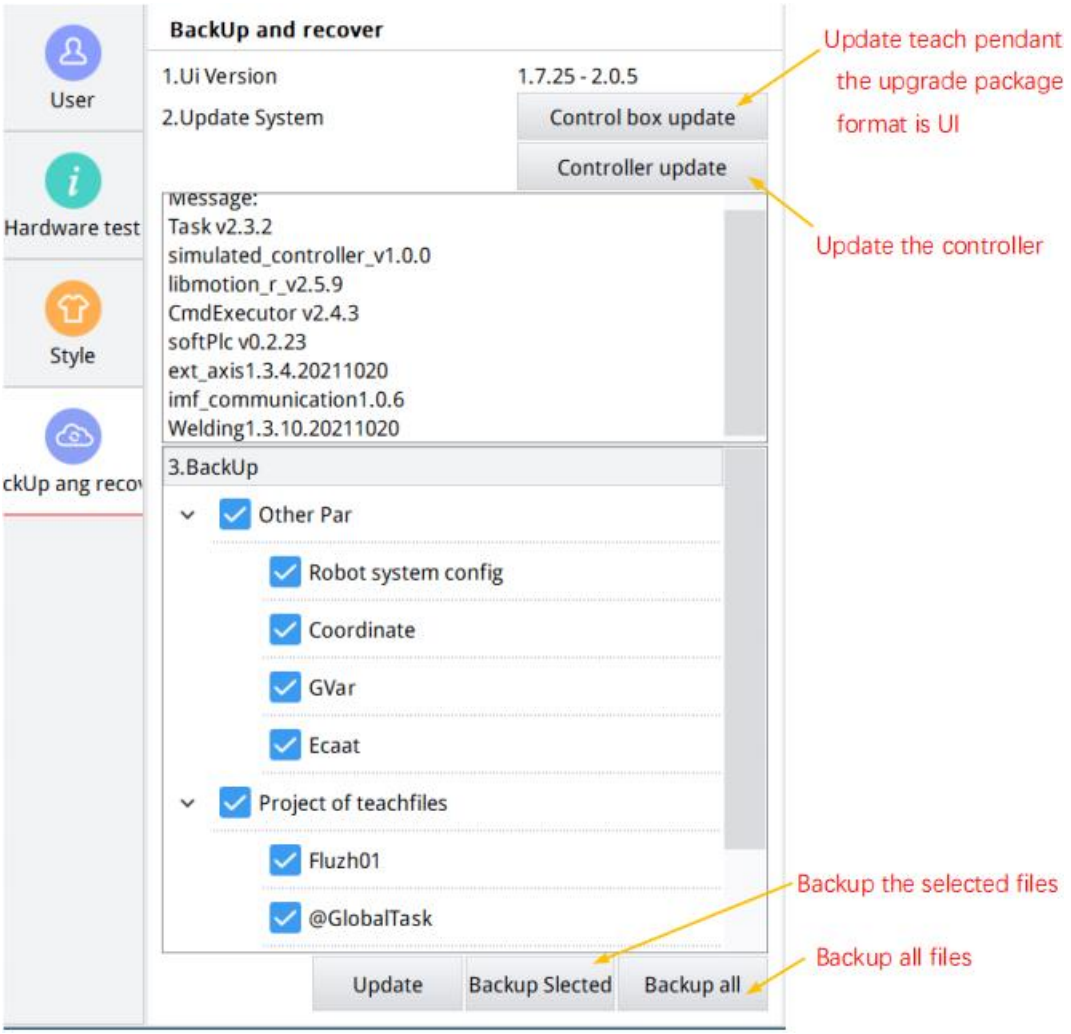


Figure 36 Interface Style Setting Interface

Set the interface font, font size and display language.

13.4.4 Upgrade and Backup



14 Instruction System

14.1 List of Orders

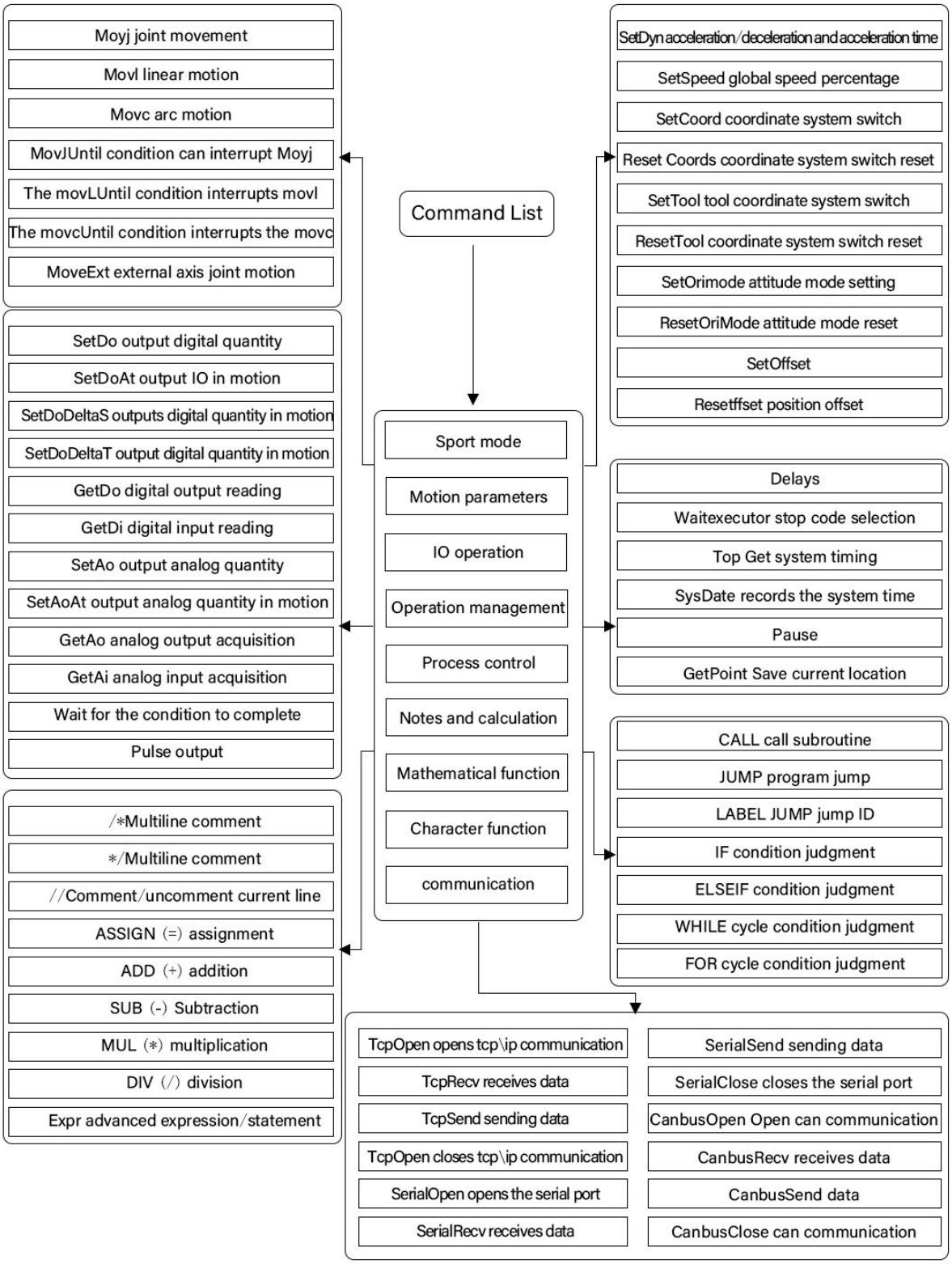


Figure 37 List of Commands

14.2 Directive Specification

Representation rules:

·**[]**: indicates that the content in brackets appears 0 or 1 times, indicating optional

·**|** : means "or", either of the two

·**=**: indicates the input assignment

·**=>**: Indicates output assignment

·**#**represents a numeric value, **Var** represents a variable, and **Str** represents a string

·The light blue italic font indicates the values or variables that need to be filled in or selected during parameter setting

14.3 Motion Instructions

14.3.1 Processing Mechanism of Motion Instruction

14.3.1.1 Motion Command Preprocessing

The robot does not move a position by executing a line of motion instructions. In order to improve the smoothness of the movement, the teaching program analysis module of the control system will compile all the instructions in advance and perform preprocessing, and then send several movement instructions to the robot's motion stack according to the logic of the program itself. At this time none of the positions were reached. It is important to understand that the execution of the application program is independent of the movement of the robot. At the same time, the system provides a pause analysis command to facilitate the user to interrupt the ahead-of-time compilation mode of the program according to the actual situation.

14.3.1.2 Motion Command Trajectory Transition

At the junction of two movement commands, because the speed change of the two movement commands before and after is not smooth, that is, the speed of the previous trajectory must be reduced to 0 before continuing to the next trajectory. This increases cycle time unnecessarily. By setting the transition section, the exercise time can be greatly reduced. When this Blend value is set to "Not used", the robot strictly arrives at every point of the trajectory. However when the Blend value is set to #Var or %, the trajectory will blend around each point and the robot does not strictly reach these points.

14.3.2 Movj Axis Motion

Movj trajectories and examples:

Movj commands are used to program point-to-point motions (axis motions). To optimize the speed of movement, the center of the tool follows a system-defined curve between the start and end points. Positions between points are interpolated by axis.

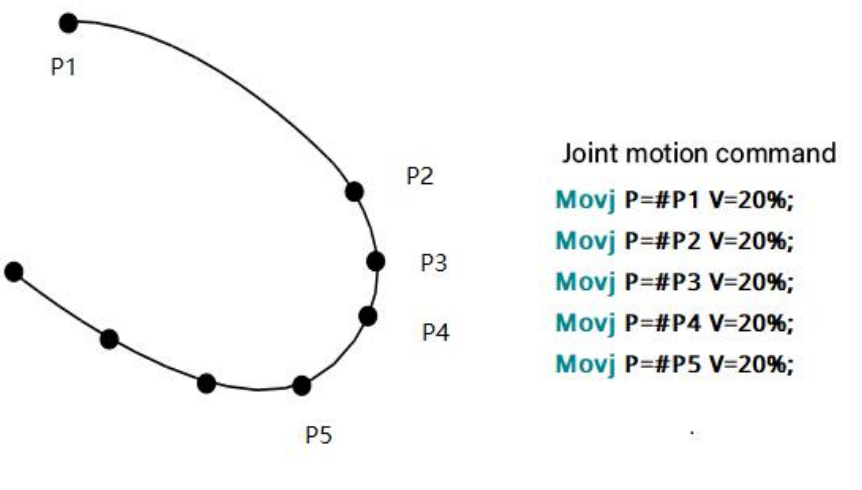


Figure 38 Movj trajectory and command

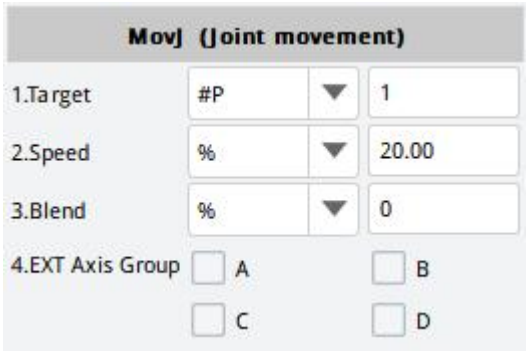


Figure 39 Movj Parameter Settings

Instruction format	MovjP=PointV=Sped%[Bl=Blend%][Ext=[A-D]]	
Examples	MovjP=#P1V=20%Bl=20%Ext=A;	
Parameter Description	target	Point: can be axis space or cartesian space point
	speed	Speed:Synthetic speed at the end of joint space, speed = % * maximum synthetic speed at the end of joint space. The value is from 1 to 100, and the default value is 20
	Blend	Blend: axis space blend radius, blend radius =%* total length of axis space trajectory.
	EXT Axis	Ext: External axis group number of synchronous motion

	Group	required, multiple selection is allowed. For Movj, there is only the synchronous motion planning of the robot and the external axis, and there is no strict position following planning. The speed of the external axis is the smaller of both the robot movement speed and the maximum speed allowed for the external axis, and there is no need to set the speed of the external axis itself separately.
Precautions		

14.3.3 Movs joint spline motion

Movs trajectory and example:

The Movs command performs b-spline motion in joint space.

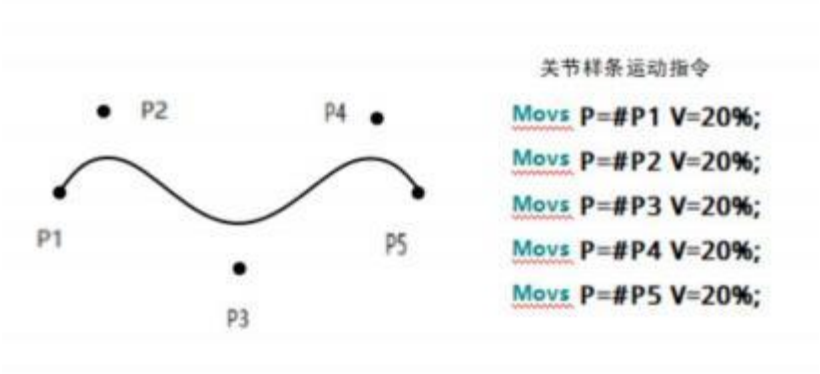
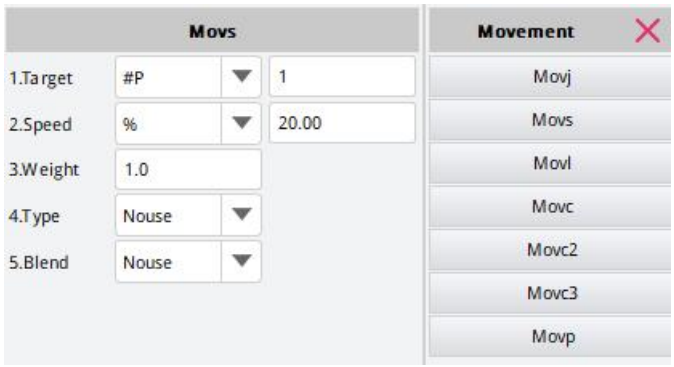


Figure 40 Movs trajectory and instructions



14.3.4 Movl Linear Motion

Movl Motion Trajectories and Examples

Rectilinear motion in Cartesian space, including pose following and planning.

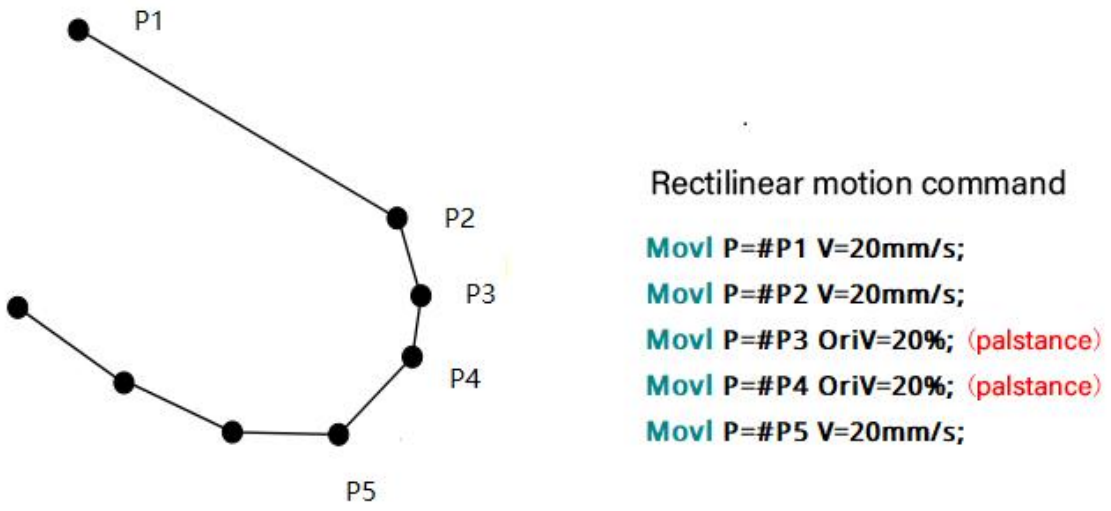


Figure 41 Movl trajectory and command

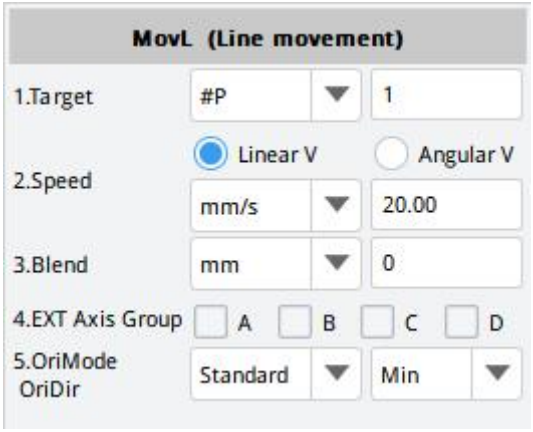


Figure 42 Movl Parameter Settings

Instruction format	MovlP=*Point*(V=*Speed*mm/s[Bl=*Blend*mm])|(OriV=*OriSped*%[OriBl=%*OriBlend]*)[Ext=[A-D]]	
Examples	MovlP=#P1V=20mm/sBl=10mmExt=A; MovlP=#P1OriV=20%OriBl=10%Ext=A;	
Parameter Description	Point	*Point*: Can be an axis space or Cartesian space point.
	Speed	*Speed*: Cartesian-space end path resultant speed, in:mm/sOriSped：Cartesian Space Posture End Resultant Velocity Percentage, Velocity =%* Posture End Resultant Maximum Velocity.
	Blend	*Blend*: Cartesian space blend radius, units:mm; *OriBlend*: Cartesian space end pose blend percentage, blend angle =%* pose space trajectory total angle.
	EXT axis	The number of the external axis group that needs to move

	group	synchronously can be selected multiple times. Whether the synchronous motion is position following depends on whether the corresponding PCS recorded by the position point moves with the corresponding external axis.
	OriMode	Standard, PathFrame, JointSpace
	OirDir	Posture planning, default Min, Max as required
Precautions	1. After setting Speed and Blend, the path is planned and the posture is followed. 2. After OriSpeed and OriBlend are set, the pose is planned and the path is followed.	

14.3.5 Movc Circular Motion

Movc trajectory and examples

Arc motion in Cartesian space, including posture following. In a circular motion, the tool center point moves along an arc defined by 3 points, with circular interpolation between the start point, middle point and end point.

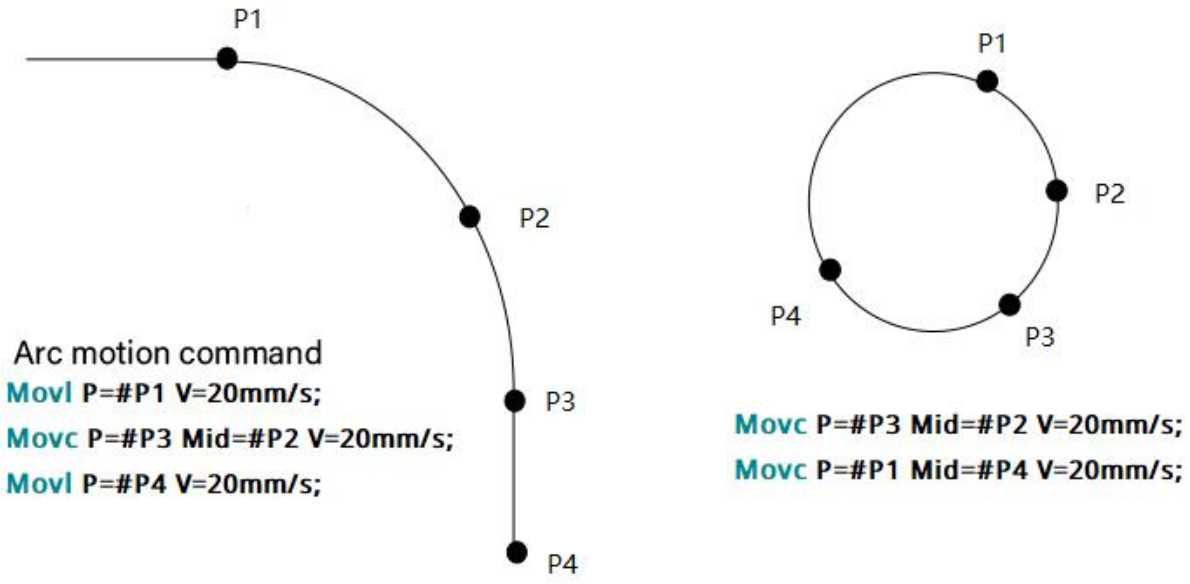


Figure 43 Movc trajectory and command

The arc command needs 2 position points, which represent the middle point and the end point of the arc respectively, and the current position of the robot or the end position of the previous action is the starting point of the arc. A complete circle requires 4 position points, realized by 2 Movc instructions.

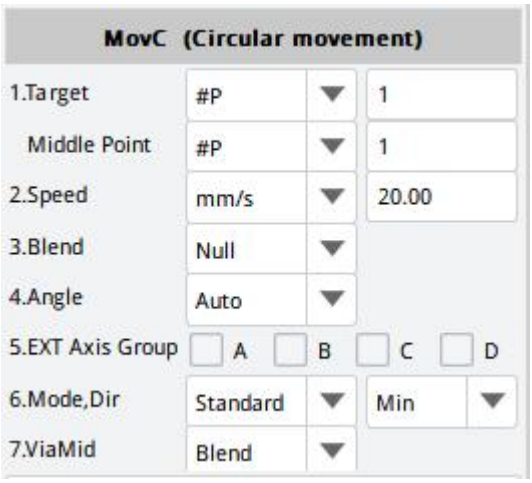


Figure 44 Movc Parameter Settings

Instruction	Movc P=*EndPoint* Mid=*MidPoint* V=*Speed* mm/s[Bl=*Blend* mm]

format	[Deg=Degree][Ext=[A-D]]	
Examples	MovcP=#P1Mid=#P2V=20mm/sBl=10mmDeg=108Ext=A;	
	Target	*EndPoint:*The end point of the arc, can be joint space or Cartesian space point;
	Middle point	*MidPoint:*The middle point of the arc,can be joint space or Cartesian space point.
	Speed	*Speed*: Cartesian-space end path resultant speed, in:mm/s。
	Blend	*Blend:* Cartesian space blend radius, units:mm。
	Angle	Degree: the arc angle of the actual motion, for example, a full circle, by default it is an arc angle formed by three points, supports negative numbers, optional.
	EXT axis group	Ext: external axis group number to be synchronized, multiple choices are allowed, same as MovL.
	Mode,Dir	Standard, PathFrame, JointSpace
	Viamid	Posture planning, default Min, Max as required 应为blend和ignore
Precautions	Arc trajectory, pose planning, path following situation is unreasonable, not supported.	

14.3.6 MovjUntil Conditional Interrupt Movj

Joint movement, if the custom Boolean expression becomes True during the movement, stop the current instruction or directly execute the next instruction; Otherwise the movement is in place. This type of command does not support the setting of transition parameters and does not support synchronous motion.

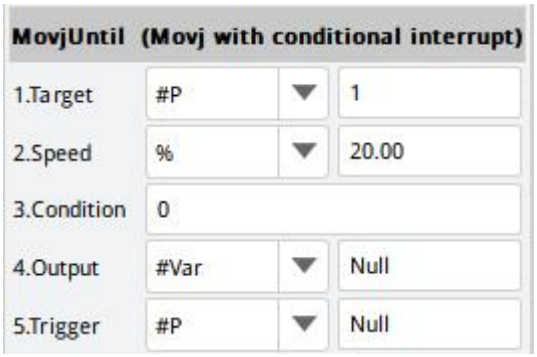


Figure 45 11.3.5 MovjUntil Parameter Settings

Instruction format	MovjUntil P=*Point* V=*Speed*% Cond=*Condition?* =>ReachEnd	
Examples	MovjUntil P=#P1V=20%Cond=(#Var1<<5)=>#Var10;	
Parameter Description	Target	*Point* : Target point, which can be joint space or Cartesian space point.
	Speed	*Speed*: Joint space end resultant velocity, unit:degree/s。
	Condition	*Condition?* : Custom Boolean expression. For example, #Var1<<5.
	Output	ReachEnd: If Condition is False, the target point is reached, ReachEnd=True; If Condition is True before reaching the target point, it means to leave or stop halfway, the target point is not reached, and ReachEnd=False . ReachEnd is a BoolVar.
	Trigger	
Precautions		

14.3.7 MovlUntil Conditional Interrupt Movl

MovlUntil Cartesian space linear motion, if the Boolean expression becomes True during the motion, stop the current instruction or directly execute the next instruction; Otherwise the

movement is in place. This type of command does not support the setting of transition parameters and does not support synchronous motion.

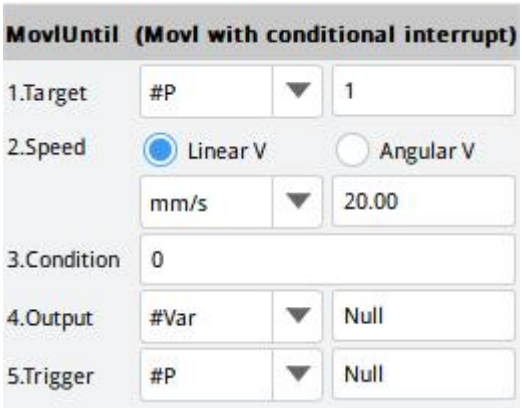


Figure 46 MovlUntil Parameter Settings

Instruction format	MovlUntil P=*Point*(V=*Speed*mm/s)|(Ori=*OriSpeed*%) Cond=*Condition?*=>ReachEnd	
Examples	MovlUntil P=#P1V=20mm/sCond=(#Var1<<5)=>#Var10; MovlUntil P=#P1OriV=20%Cond=(#Var1<<5)=>#Var10;	
Parameter Description	Target	*Point*: Target point, which can be joint space or Cartesian space point.
	Speed	*Speed*: Cartesian-space end path resultant speed, unit:mm/s; *OriSpeed*: Cartesian space end pose resultant velocity percentage, velocity =%* pose end resultant maximum velocity.
	Condition	*Condition?*： Custom Boolean expression. For example, #Var1<<5.
	Output	ReachEnd: If Condition is False, the target point is reached, ReachEnd=True: If Condition is True before reaching the target point,it means leaving or stopping halfway and not reaching the target point,ReachEnd=False. ReachEnd is a BoolVar.
	Trigger	
Precautions	Select one between *Speed* and *OriSpeed.* The former represents path planning and posture following;The latter is pose planning and path following.	

14.3.8 MovcUntil Conditional Interrupt Movc

MovcUntil Cartesian space arc motion, if the Boolean expression becomes True during the motion, stop the current instruction or directly execute the next instruction; Otherwise the movement is in place. This type of command does not support the setting of transition parameters and does not support synchronous motion.

Figure 47 MovcUntil Parameter Settings

Instruction format	MovcUntil P=*EndPoint* Mid=*MidPoint* V=*Speed* mm/s[Deg=*Degree*] Cond=*Condition?* =>ReachEnd	
Examples	MovcUntil P=#P1Mid=#P2V=20mm/sDeg=108 Cond=(#Var1<<5)=>#Var10;	
Parameter Description	Target	*EndPoint:* the end point of the arc, which can be a joint space point or a Cartesian space point;
	Middle point	*MidPoint* : The middle point of the arc, either in joint space or Cartesian space.
	Speed	*Speed* : Cartesian-space end path resultant speed, unit:mm/s。
	Angle	*Degree*: the arc angle of the actual motion, such as a full circle. The default is the arc angle formed by three points. Negative value is supported and optional.
	Condition	*Condition?* : Custom Boolean expression. For example, #Var1<<5.
	Output	ReachEnd: If Condition is False, the target point is reached,

		ReachEnd=True: If Condition is True before reaching the target eachEnd=False if the target point is not reached. ReachEnd is a BoolVar.
	Trigger	
	ViaMid	
Precautions		

14.3.9 MoveExt Outer Axis Set Articulation

The MoveExt external axis group joint movement, and the robot chooses whether to follow the motion or not according to the set parameters.

Figure 48 MoveExt Parameter Settings

Instruction format	MoveExt P=*Point* V=*Speed*%[Bl=*Blend*%][Ext=[*A-D*]]	
Examples	MoveExt P=#P1V=20%Ext=A	
Parameter Description	Target	*Point*: Target point, using only the relevant part of the external axis.
	Speed	*Speed*: Joint space end resultant velocity percentage, velocity =%* Joint space end resultant maximum velocity.
	Blend	*Blend*: Percentage of joint space blend radius, blend radius =%* total axis space trajectory length.
	EXT Axis Group	Ext: the external axis group number which requiring synchronous motion, with values of A,B,C and D. If Ext is not selected, all external axes move together. Whether the synchronous motion of the robot is position following the external axis depends on whether the PCS recorded by the position point moves together with the external axis.
Precautions		

14.4 Motion Parameters

14.4.1 SetDyn Acceleration/Deceleration and Acceleration Time

Acceleration/deceleration magnification setting and jerk time setting.

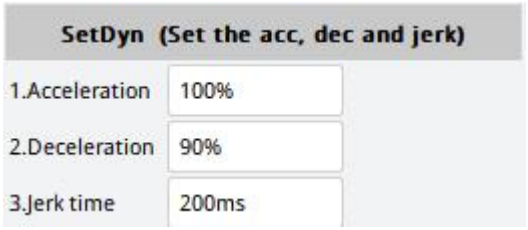


Figure 49 SetDyn Setting Interface

Instruction format	SetDyn Acc=#%Dec=#% JerkTime=#	
Examples	SetDyn Acc=1%Dec=1% JerkTime=128ms;	
Parameter Description	Acceleration	Acc: Global acceleration percentage, all instructions apply next.
	Deceleration	Dec: Global deceleration percentage, all subsequent commands apply.
	jerk time	JerkTime: jerk time, unit:ms, ranging from 32 to 512 milliseconds, and the default value is 128.
Precautions		

14.4.2 SetSpeed Global Speed Percentage

Adjust the speed percentage of the motion command following this statement.

Figure 50 SetSpeed Parameter Setting

Instruction format	SetSpeed Vel=#%	
Examples	SetSpeed Vel=20%;	
Parameter Description	global speed	Global speed percentage.
Precautions	If you do not call the SetSpeed command, the program defaults is 20%.	

14.4.3 SetCoord Coordinate System Switching

Coordinate system switching is applicable to all subsequent motion commands, and coordinate points in Cartesian space are recorded until the next SetCoord command is encountered, or the switching is canceled when ResetCoord is encountered.

Figure 51 SetCoord Settings

Instruction format	SetCoord Type= *Type*Idx=#	
Examples	SetCoord Type=PCSIdx=1;	
Parameter Description	coordinate	*Type*: Coordinate system type, optional KCS,PCS,WCS; Idx: Coordinate system serial number.
Precautions	This instruction takes effect for the location points between SetCoord and ResetCoord.	

14.4.4 ResetCoord Coordinate System Switching Reset

The coordinate system stops switching and returns to the coordinate system recorded in the command itself. Refer to SetCoord command.

14.4.5 SetTool Coordinate System Switching

The tool coordinate system switch is applicable to all subsequent motion commands until the next SetTool or ResetTool is encountered to cancel the switching.

Chart 52 SetTool Settings

Instruction format	SetTool Index=#	
Examples	SetTool Index=1	
Parameter Description	Tool coordinate	Index: Tool coordinate system serial number.

	No.	
Precautions	This command is valid for the position points between SetTool and ResetTool.	

14.4.6 ResetTool Coordinate System Switching Reset

The tool stops switching and returns to the tool number recorded in the command itself, refer to SetTool command.

14.4.7 SetOriMode Posture planning mode setting

Set the posture planning mode, and all the following Cartesian space motion instructions will take effect until the next SetOriMode or ResetOriMode is encountered.

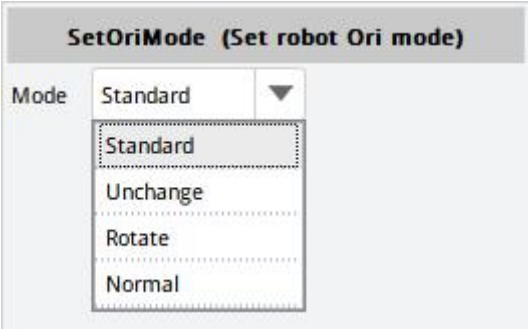


Figure 53 SetOriMode Settings

Instruction format	SetOriMode Mode=*Standard*|*Unchange*|*Rotate*|*Normal*	
Examples	SetOriMode Mode=Standard;	
Parameter Description	Mode	Mode: Standard,Unchange,Rotate,Normal.
Precautions	This instruction is valid for the position points between SetOriMode and ResetOriMode.	

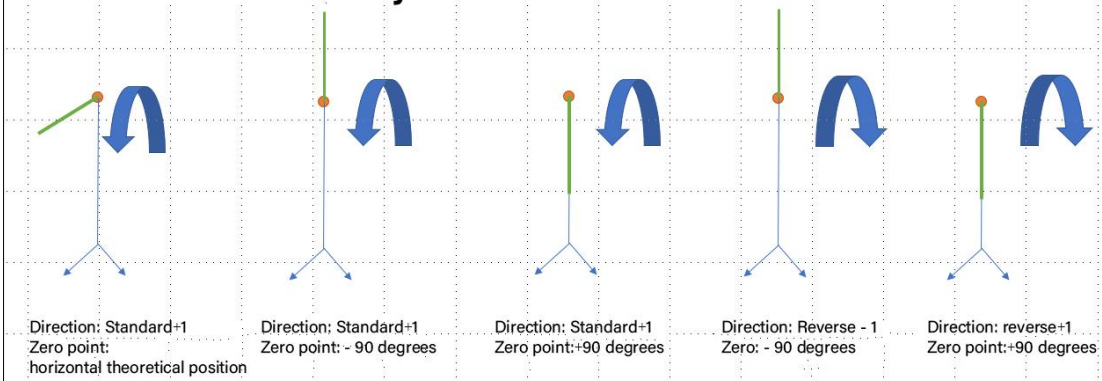
14.4.8 ResetOriMode posture Mode Reset

Resumes the posture standard follow mode Standard.

14.4.9 SetOffset Position Offset Setting

Set axis space position point, Cartesian position point and external axis position point offset.

Figure 54 SetOffset Parameter Setting

Instruction format	SetOffsetType="Axis"|"*Pose*"(J1=#J2=#J3=#J4=#J5=#J6=#J7=#J8=#)||(X=#Y=#Z=#[Roll=#Pitch=#Yaw=#])	
Examples	SetOffset Type=AxisJ1=5.00J2=10.00J3=0.00J4=0.00J5=0.00J6=0.00J7=0.00J8=0.00; SetOffset Type=PoseX=5.00Y=10.00Z=0.00Roll=0.00Pitch=0.00Yaw=0.00; SetOffset Type=PoseX=0Y=0Z=0Roll=10Pitch=10Yaw=10	
Parameter Description	Type	Type: offset type, robot joint offset, Cartesian position point offset 1. J1, J2, J3, J4, J5, J6, J7, J8 axis position offset or external axis offset; 2.X, Y, Z: position offset; 3.Roll，Pitch，Yaw：Posture offset.
	offset value	offset value
Precautions	Joint zero offset+joint direction  Direction: Standard+1 Zero point: horizontal theoretical position Direction: Standard+1 Zero point: - 90 degrees Direction: Standard+1 Zero point:+90 degrees Direction: Reverse - 1 Zero: - 90 degrees Direction: reverse+1 Zero point:+90 degrees Joint direction and zero offset relative to standard case Encoder value → theoretical joint angle → displayed joint angle=(theoretical joint angle - zero offset) * joint direction Display joint angle → theoretical joint angle=display joint angle * joint direction+zero point offset The input of forward kinematics is the theoretical joint angle, and the output of inverse kinematics is the theoretical joint angle The current angle is also the theoretical joint angle calculated from the data obtained by the encoder This instruction is valid for the position points between SetOffset and ResetOffset.	

14.4.10 ResetOffset Position Offset Cancellation

Joint Position Point Offset no longer works and is often paired with SetOffset.

14.5 Operation of IO

14.5.1 SetDo Output Digital Quantity

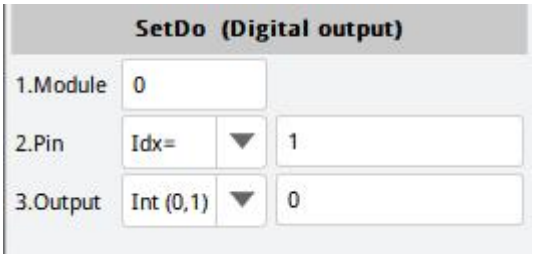
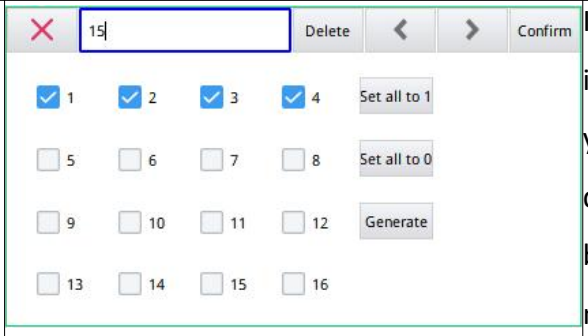


Figure 55 SetDo Settings

Instruction format	SetDo Val=*Value*Grp=*Group* Idx=#|*ALL*	
Examples	SetDo Val=1Grp=0Idx=1;	
	SetDo Val=0b1111Grp=1Idx=All;	
Parameter Description	Module	*Group*:Group: Indicates which group of digital outputs, the controller software considers 16 IOs as a group.
	Pin	Idx.: Indicates the IO sequence number in the group. If *ALL* is selected, it indicates that the group is output together.
	Output	*Value*: Indicates that the digital output is 0 or 1. If *ALL* is selected, *Value* is the binary number 0b01100110.
Precautions		If *ALL* is selected, Val is interpreted as binary. When you modify the output value, a decimal number is generated but is displayed as a 0b binary number. The same below

14.5.2 SetDoAt Output IO in Motion

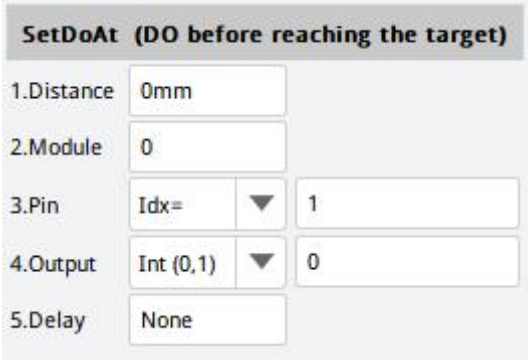


Figure 56 SetDoAt

Instruction format	SetDo AtDis=*Distance* Val=*Value* Grp=*Group* Idx=#|*ALL* [Td=*Delay* ms]	
Examples	SetDo AtDis=7mm Val=1 Grp=1 Idx=1 Td=5ms; SetDo AtDis=7mm Val=0b1111 Grp=1 Idx=All Td=5ms;	
Parameter Description	distance	*Distance*: indicates the distance from the target point, unit:mm.
	Module	*Group*: indicates which group of digital outputs, and the controller software considers 16 IOs as a group.
	Pin	Idx: Indicates the IO sequence number in the group. If *ALL* is selected, the group is output together.
	Output	*Value* : Indicates that the digital output is 0 or 1. If *ALL* is selected, value is the binary number 0b01100110.
	Delay	*Delay* : indicates the delay time, unit ms.
Precautions	This instruction is valid only for the previous instruction.	

14.5.3 SetDoDeltaS Output Digital Quantity in Motion

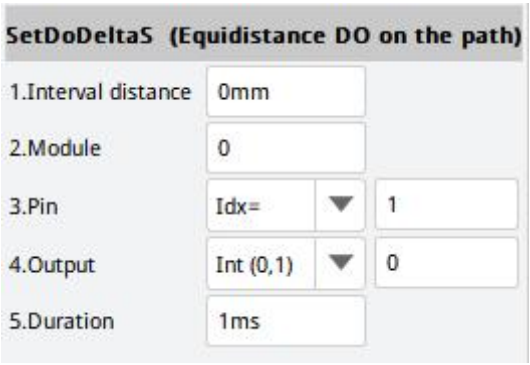


Figure 57 SetDoDeltaS Settings

Instruction	SetDoDeltaS Dis=*DeltaS* Val=*Value* Grp=*Group* Idx=#|*ALL* Width=#ms

format		
Examples	SetDoDeltaS Dis=50mm Val=1 Grp=1 Idx=1 Width=2ms;	
	SetDoDeltaS Dis=50mm Val=0b1111 Grp=1 Idx=All Width=2ms;	
Parameter Description	Interval distance	*DeltaS*: indicates the separation distance, in mm
	Module	*Group*: indicates the group of digital outputs, and the controller software considers 16 IO as a group.
	Pin	Idx: Indicates the IO sequence number in the group. If *ALL* is selected, the group is output together.
	Output	*Value:* Indicates that the digital output is 0 or 1. If *ALL* is selected, value is the binary number *0b01100110*.
	Delay	*Width:* Indicates the signal duration in ms.
Precautions	This instruction is valid only for the previous instruction.	

14.5.4 SetDoDeltaT Output Digital Quantity in Motion

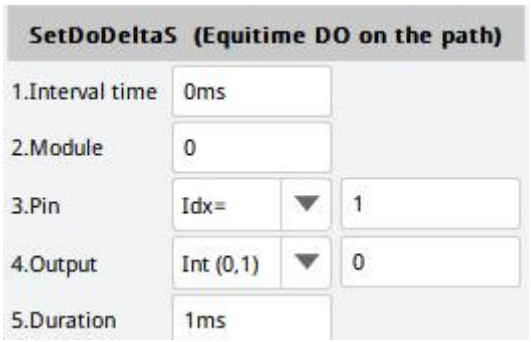


Figure 58 SetDoDeltaT Settings

Instruction format	SetDoDeltaT Time=*DeltaT* Val=*Value* Group=# Idx=*#*|*ALL* Width=*#*ms	
Examples	SetDoDeltaT Time=5ms Val=1 Group=1 Idx=1 Width=1ms; SetDoDeltaT Time=5ms Val=0b1111 Group=1 Idx=All Width=1ms;	
Parameter Description	Interval time	DeltaT: represents the time interval in ms.
	Module	*Group:* indicates which group of digital outputs, and the controller software considers 16 IOs as a group.
	Pin	Idx: Indicates the IO sequence number in the group. If *ALL* is selected, the group is output together.
	Output	*Value:* Indicates that the digital output is 0 or 1. If *ALL* is selected, *value* is the binary number *0b01100110.*

	Delay	Width: Indicates the signal duration in ms.
Precautions	This instruction is valid only for the previous instruction.	

14.5.5 GetDo Digital Output Reading

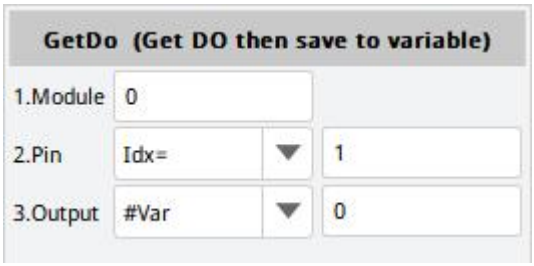


Figure 59 GetDo Parameters

Instruction format	GetDo Grp=*Group* Idx=*#|ALL*=>*#Var#*	
Examples	GetDo Grp=1 Idx=1=>#Var2; GetDo Grp=1 Idx=All=>#Var2;	
Parameter Description	Module	*Group*: indicates which group of the digital outputs, and the controller software considers 16 IOs as a group.
	Pin	Idx: Indicates the IO sequence number in the group. If *ALL* is selected, the group is output together.
	Output	The digital output value is stored in a numeric variable.
Precautions		

14.5.6 GetDi Digital Quantity Input Acquisition

Chart 60 GetDi Parameters

Instruction format	GetDi Grp=*Group* Idx=*#*=>*#Var#*	
Examples	GetDi Grp=1 Idx=1=>#Var2; GetDi Grp=1 Idx=All=>#Var2;	
Parameter Description	Module	*Group*: indicates the group of digital inputs, and the controller software considers 16 IO as a group.

	Pin	Idx: Indicates the IO sequence number in the group. If *ALL* is selected, the group is imported together.
	Output	Digital input values are stored in numeric variables.
Precautions		

14.5.7 SetAo Output Analog Quantity

Figure 61 SetAo Parameter Setting

Instruction format	SetAo Val=*Value* Idx=#	
Examples	SetAo Val=7.00 Idx=1;	
Parameter Description	output	*Value*: Indicates the analog output value.
	Pin	Idx: Indicates the IO sequence number in the group.
Precautions		

14.5.8 SetAoAt Output Analog Quantity in Motion

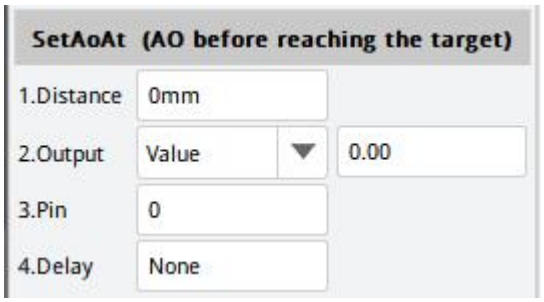


Figure 62 SetAoAt Parameter Setting

Instruction format	SetAoAt Dis=*Distance* Val=*Value* Idx=# [Td=*Delay*ms]	
Examples	SetAoAtDis=10mmVal=8.00Idx=25ms;	
Parameter Description	Distance	*Distance:* Indicates the distance from the target point, unit mm.
	Output	*Value:* Represents the analog value.
	Pin	Idx: Indicates the IO sequence number in the group.
	Delay	*Delay*: indicates the delay time,unit:ms.

Precautions	This instruction is valid only for the previous instruction.

14.5.9 GetAo Analog Output Acquisition

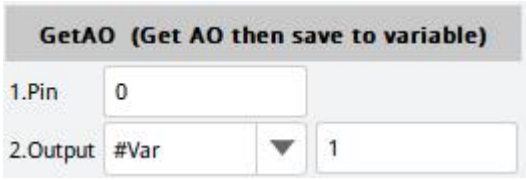


Figure 63 GetAo Parameter Setting

Instruction format	GetAo Idx = #=>#Var#	
Examples	GetAo Idx=1 =>#Var2;	
Parameter Description	Pin	Idx: Indicates the IO sequence number in the group.
	Output	Analog input values are stored in variables.
Precautions		

14.5.10 GetAi Analog Input Acquisition

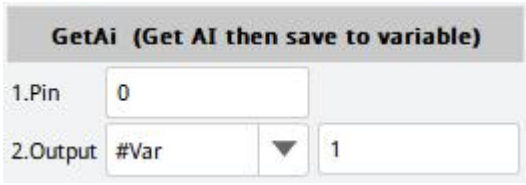


Figure 64 GetAi Parameter Setting

Instruction format	GetAi Idx = #=>#Var#	
Examples	GetAi Idx=1 =>#Var2;	
Parameter Description	Pin	Idx: Indicates the IO sequence number in the group.
	Output	Analog input values are stored in variables.
Precautions		

14.5.11 Wait for the condition to complete

Figure 65 Wait Parameter Settings

Wait for a Boolean expression

Instruction format	Wait Cond=*Condition?* [Time = #ms => *#Var#*]	
Examples	Wait Cond=(#Var1>>5) Time=5ms =>#Var2;	
Parameter Description	condition	*Condition?* ： Custom Boolean expression.
	Time(ms) 0-Keep waiting	Time: indicates the upper limit of the waiting time. If not set, it means to wait all the time (in s).
	timeout output 0-No output	*#Var#* is True if timeout occurs, False otherwise.
	Wait in advance(mm) 0-invalid	
Precautions		

14.5.12 Pulse Output

The digital output is set to 1 and held for a period of time, then set to zero

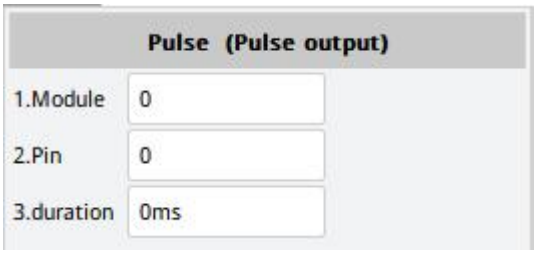


Chart 66 Pulse Parameter Settings

Instruction format	Pulse Grp=*Group* Idx = #Width = #ms

Examples	Pulse Grp=1 Idx=1 Width=5ms;	
Parameter Description	Module	***Group***: indicates which group of the digital inputs, and the controller software considers 16 IOs as a group.
	Pin	Idx: Indicates the IO sequence number in the group.
	Duration	Width indicates the setting time in ms.
Precautions		

14.6 Motion Management

14.6.1 Delay

Figure 67 Delay Parameter Setting

Instruction format	Delay Time=#ms;	
Examples	Delay Time=2000ms;	
Parameter Description	Delay time	Timing time, in ms.
Precautions		

14.6.2 WaitExecutor Stops Decoding

Stop teaching instruction decoding in advance, wait for the robot to actually execute the current instruction line and then automatically start decoding.

Examples	WaitExecutor;

14.6.3 tod Get System Timing

Figure 68 tod parameter setting

Instruction format	*#Var#*= tod();
Examples	#Var1= tod();

	#Var2=tod(); #Var3=#Var2-#Var1; Record a time at time T1 and put it into variable #Var1;Record a time at T2 and put it in variable #Var2;The time difference between T1 and T2 is equal to the value of #Var2 minus the value of #Var1.	
Parameter Description	Result	The recorded time is stored in *#Var#.*

14.6.4 SysDate Record system timing

Figure 69 SysDate parameter setting

Instruction format	SysDate =>***#Var#***	
Examples	SysDate =>#Var1;	
Parameter Description	Variable	Date: The system time is recorded in a string variable.

14.6.5 Pause

Stop execution and wait for external signal to restart

Examples	Pause;

14.6.6 GetPoint Save Current Position

Get Tool End Position

Figure 70 GetPoint Parameter Settings

Instruction	GetPoint ***Type*** => P=***Point***

format		
Examples	GetPoint KCS=>#P1;	
Parameter Description	coordinate	Type Coordinate system model, KCS,WCS,PCS,ACS.
	Output to	Point stores the location point.
Precautions		

14.7 Control Process

14.7.1 Call Subroutine

Examples	1 PROCEDURE 2 Movj P=#P1 V=20%; 3 CALL call; 4 Movl P=#P2 V=20mm/s; 5 ENDPROCEDURE	When the robot end moves to point P1, start to execute the third line of program: Call the subroutine named "call", after the execution of the subroutine "call", return to the original program to continue execution of the fourth line of the program to move to point P4. Note: The calling and called programs must be in the same project. Subprogram name: can only contain alphanumeric underscores and cannot start with a number.

14.7.2 JUMP-LABEL Flag

Examples	2 Movj P=#P1 V=20%; 3 JUMP 1; 4 Movj P=#P2 V=20%; 5 LABEL 1; 6 Movj P=#P3 V=20%;	After the end of the robot moves to point P1, it jumps to the fifth line labeled with"1"and moves down to the sixth line to point P3

14.7.3 IF Conditional Judgment

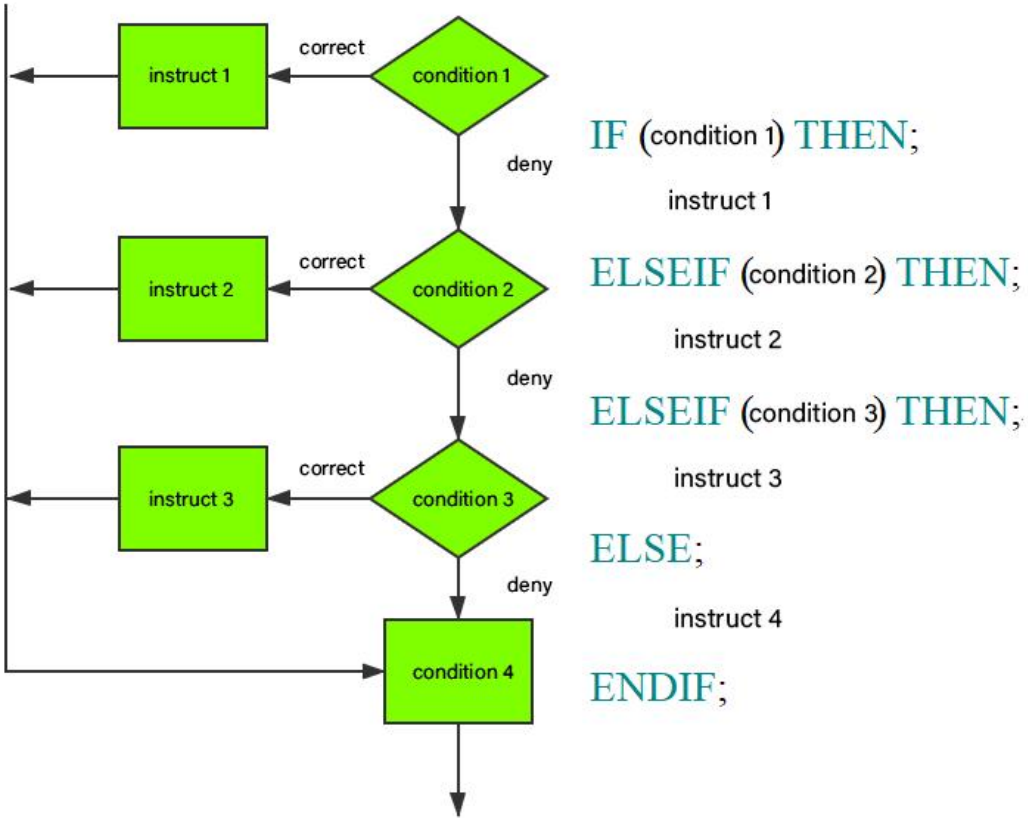


Figure 71 IF Flow Chart

Examples	2 Movj P=#P1 V=20%; 3 IF (#Var1>=5) THEN; 4 Movl P=#P2 V=20mm/s; 5 ELSE; 6 Movl P=#P3 V=20mm/s; 7 ENDIF;	After the end of the robot moves to the point P1, the condition (whether the value of #Var1 is greater than or equal to 5) is judged, if the value is greater than or equal to 5, the fourth row is run, and the end of the robot moves to the point P2; otherwise, the end of the robot moves to the point P3.

14.7.4 ELSELF Condition Judgment

Refer to IF flow chart.

Examples	2 Movj P=#P1 V=20%; 3 IF (#Var1=5) THEN; 4 Movl P=#P2 V=20mm/s; 5 ELSEIF (#Var1 = 8) THEN; 6 Movl P=#P3 V=20mm/s; 7 ELSE; 8 Movl P=#P4 V=20mm/s; 9 ENDIF;	After the robot end moves to the point P1, judging the condition 1(whether the value of #Var1 is equal to 5), if the value is equal to 5, run the fourth row, move to the point P2, and the judgment end; judging condition 2(whether the value of #Var1 is equal to 8), if the value is equal to 8, run the sixth line, move to point P3, and the judgment end; otherwise move to point P4.

14.7.5 WHILE Cycle Condition Judgement

WHILE loop: conditional loop, first judge and then execute, if the condition is not established, exit the loop.

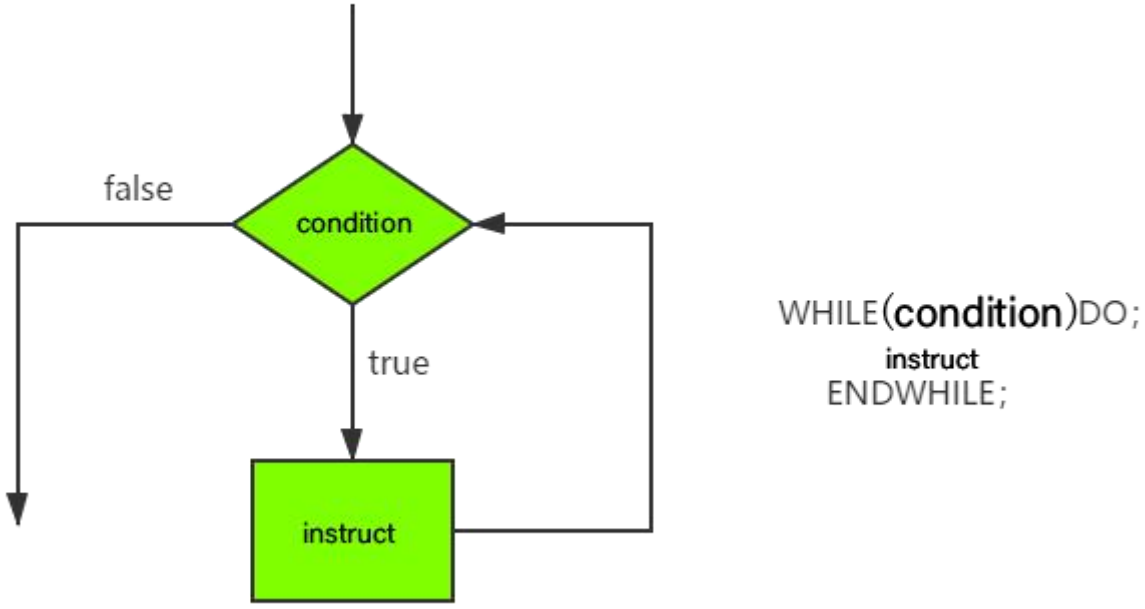


Figure 72 WHILE Flow Chart

Examples	2 Movj P=#P1 V=20%; 3 #Var1 = 1; 4 WHILE (#Var1>=10) DO; 5 Movj P=#P2 V=20%; 6 Movj P=#P3 V=20%; 7 #Var1 = #Var1 + 1; 8 ENDWHILE;	The robot end runs to P1 (transition point), sets the value of #Var1 to 1, determines that the value of condition 1 is False (condition 1 is not satisfied), and executes the program to move to points P2, P3, and #Var1+1 in a loop until condition 1 is satisfied when #Var1=10, and exits the loop. The end of the robot makes a total of 9 round trips between P2 and P3.

14.7.6 FOR Loop Condition Judgment

Figure 73 For Loop

Instruction format	FOR#Var#FROM#1TO#2STEP#3DO; Procedure 1; ENDFOR;	By default, the value of #Var#is set to #1, and when the value of #Var#is in the interval [#1, #2], the program 1 is executed in a loop, and the number of loop steps is #3, that is,#Var#is added by #3 once in each loop (#Var#=#Var#+#3).
Example 1	2 Movj P=#P1 V=20%; 3 FOR #Var1 FROM 1 TO 10 STEP 1 DO; 4 Movl P=#P2 V=20mm/s; 5 Movl P=#P3 V=20mm/s; 6 ENDFOR;	The end of the robot first runs to P1(transition point), and then runs 10 times between points P2 and P3 before exiting the FOR loop. (The value of #Var1 is set to 1 at the first cycle, +1 each time, until #Var1=11>10 after 10 cycles, exit the cycle)

Example 2	9 Movj P=#P1 V=20%; 10 #Var1 = 1; 11 FOR #Var1 FROM 1 TO 10 STEP 2 DO; 12 Movl P=#P2 V=20mm/s; 13 Movl P=#P3 V=20mm/s; 14 Movc P=#P1 Mid=#P1 V=20mm/s; 15 ENDFOR;	The end of the robot first runs to P1(transition point), and then loops between points P2 and P3 for 5 times before exiting the FOR loop. (The value of #Var1 is set to 1 in the first cycle, +2 each time, until #Var1=11>10 after 5 cycles, exit the cycle)

14.8 Notes and Calculations

14.8.1 /**/ Multi-line Note

Examples	4 /*FOR #Var1 FROM 1 TO 10 STEP 1 DO; 5 Movl P=#P2 V=20mm/s; 6 Movl P=#P3 V=20mm/s; 7 Movc P=#P1 Mid=#P1 V=20mm/s; 8 ENDFOR;*/	Insert {/* multiline comment} at the start line and {*/multiline comment} at the end line. Uncomment and repeat.

14.8.2 Comment/Uncomment Current Line

Examples	//Movj P=#P1 V=20%;	Click {// Comment / Uncomment Current Line }tocomment or uncomment

14.8.3 ASSIGN (=) Assignment

ASSIGN (=) Assignment

Figure 74 Assignment

Instruction format	A=B
Examples	#GVar1=#Var1; #Var1=1.58; #GStr1=#Str5; #Str1="jjkkll"; #GP1=#GP5;

	#GIncP1=#IncP2;
Parameter Description	assigning the value of B to A; Numeric variables , string variables , positional variables, and positional increment variables are supported.

14.8.4 ADD (+) Addition

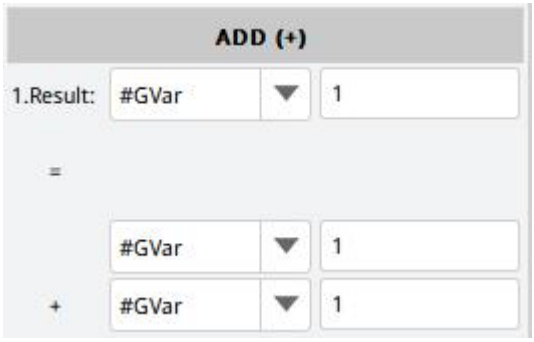


Figure 75 Addition

Instruction format	C=A+B
Examples	#GVar1=#GVar1+1; #GVar1=#Var1+#Var2; #GVar1=1+2;
Parameter Description	adding the value of A and the value of B to C;Support for numeric variables

14.8.5 SUB (-) Subtraction

Figure 76 Subtraction

Instruction format	C=A-B
Examples	GVar1=#GVar1-1; #GVar1=#Var2-#Var1;

	#Var1=5-1;
Parameter Description	subtracting the value of A from the value of B and assigning the value to C;Support for numeric variables

14.8.6 MUL(*) Multiplication

Figure 77 Multiplication

Instruction format	C=A*B
Examples	#GVar1=#GVar1*#Var2; #GVar1=#GVar1*2; #GVar1=2*3;
Parameter Description	multiplying the value of A by the value of B and assigning the result to C;Support for numeric variables

14.8.7 DIV(/) Division

Figure 78 Addition

Instruction format	C=A/B
Examples	#GVar1=#GVar1/#Var2; #GVar1=#GVar1/2; #GVar1=2/3;

Parameter Description	dividing the value of A by the value of B and assigning the value to C;Support for numeric variables

14.8.8 Expr Advanced Expressions/Statements

Expr advanced expression support:

1. Four arithmetic operations
2. Bit operation
3. Boolean operation
4. Mathematical function
5. String Handling

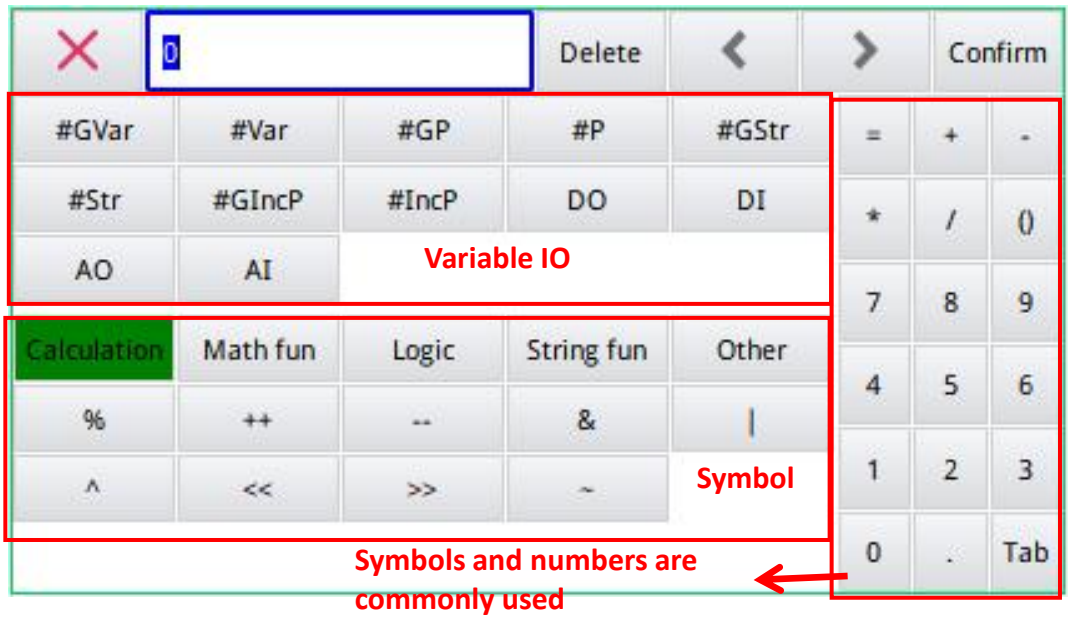


Figure 79 Advanced Expression Interface

Extended Description:

Operation

%	modulo (remainder operation)
++	self-increment
--	Auto-decrement
&	bitwise AND
|	bitwise OR
^	bitwise XOR

<<	left shift
>>	Right shift
~	negation

Mathematical function

ceil()	Rounded up
trunc(,)	variable rounding
exp()	e exponential function
log(,)	10 Logarithmic function
mod(,)	remainder function
sqrt(,)	square root
pow(,)	exponential function
sin()	sine
cos()	cosine
tan()	tangent
acos()	Arc cosine
asin()	Arc sine
atan()	Arc tangent
atan2()	arctangent (2 variables)
abs()	absolute value

Logic

&&	logical AND
||	logical OR
!	logical not
>=	Greater than or equal to
<=	less than or equal to
>	greater than
<	less than
==	equal to
!=	Not equal to

Character function

strtovar()	Converts a string to a numeric type
vartostr()	Converts a numeric type to a string

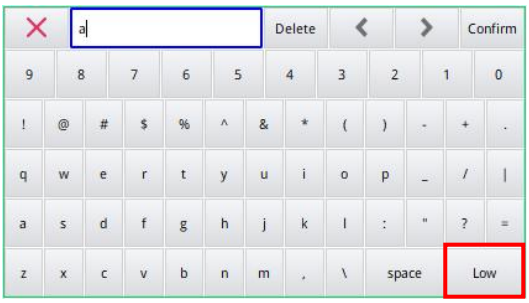
strcat(,)	String concatenation
strlen()	Calculate string length
strcmp(,)	String comparison
strtok(,)	Separates a string into fragments based on the delimiter

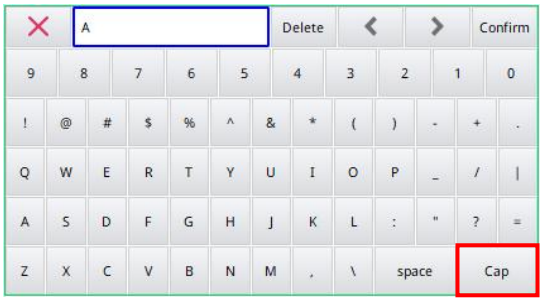
Other

tod()	Timing instruction

Chinese and upper and lower case switching

Click [Switch] in the lower right corner, and the left side will switch different input keys.





Examples:

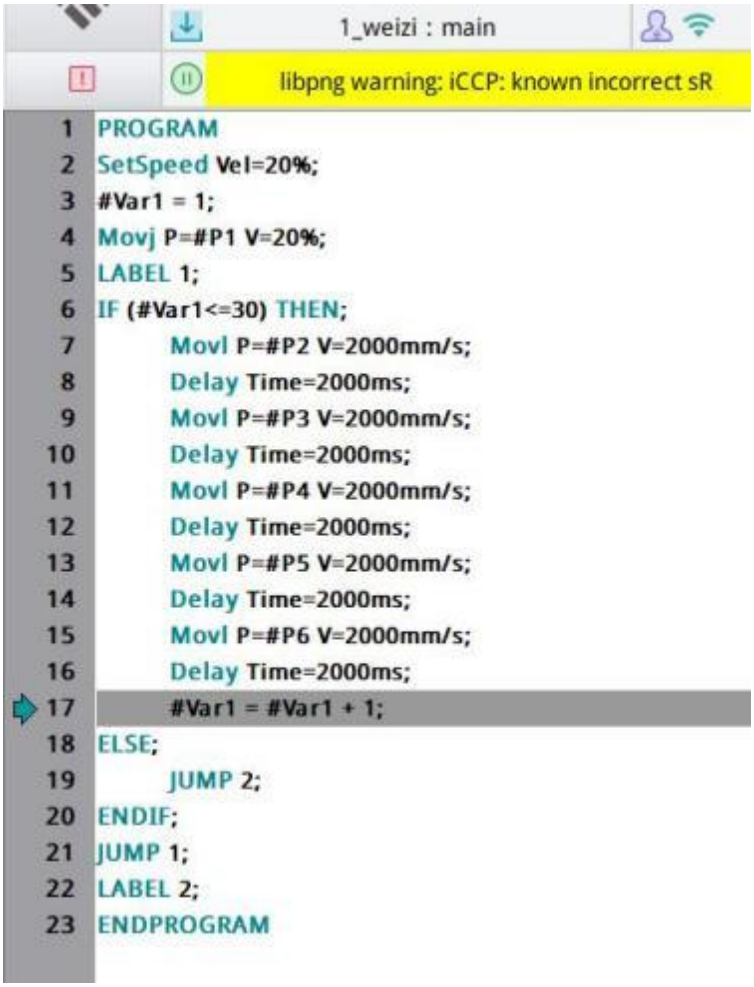


Figure 80 Example of an Advanced Expression

As in line 3: assign a value to the variable #Var

Line 6: IF statement (custom Boolean expression)

Can be entered through advanced expressions.

14.9 Mathematical Functions

abs(X)	take absolute value
sqrt(X)	root sign
exp(X)	natural index
In(X)	natural logarithm
sin(X)	sine
cos(X)	cosine
tan(X)	tangent
asin(X)	arcsine
acos(X)	arccosine
floor(X)	Rounding down

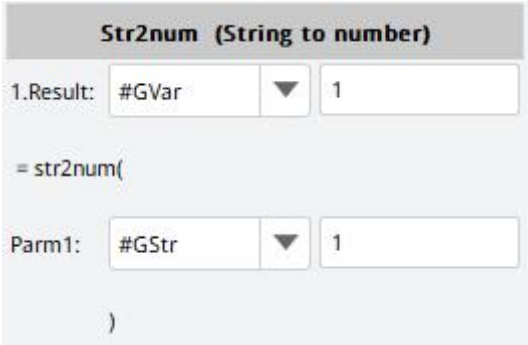
ceil(X)	Rounding up
round(X)	Rounding off
trunc(X)	rounding
pow(X,Y)	X to the power of Y
mod(X,Y)	remainder of X to Y
atan2(X,Y)	arctangent

14.10 Character Functions

14.10.1 strlen (X) Finding the Length of a String

Parameter Description	Calculates the length of a string and stores it in a numeric variable
Examples of instructions	#Str1=asfsdfasg #Var1=strlen(#Str1)
operation result	#Var1=9;

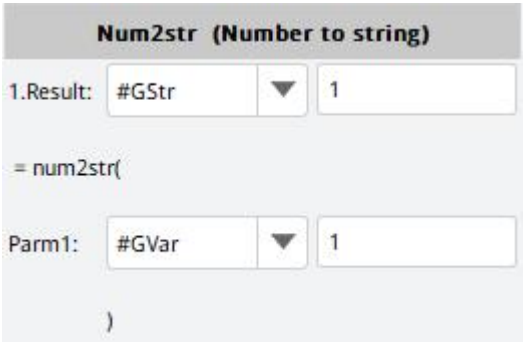
14.10.2 str2num(X) String to Real



Parameter Description	Converts a string to a real variable

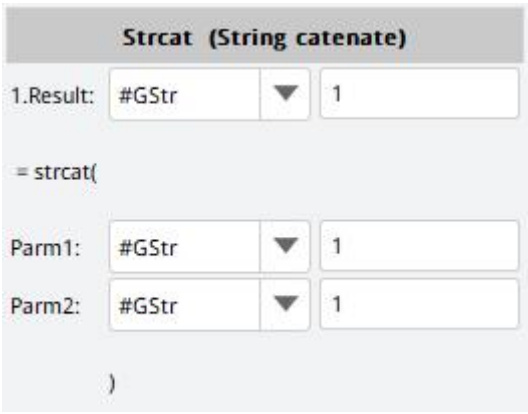
Examples of instructions	#Var1=str2num(#Str1)
Parameter Description	It is generally used to convert a numeric string sent and received into a real number for calculation or assignment

14.10.3 num2str(X) Real to String



Parameter Description	Converts a real variable to a string
Examples of instructions	#Str1=num2str(#Var1)
Parameter Description	It is generally used to transfer the x,y,z and other values of the location point out of the string and send it

14.10.4 strcat (X, Y) String Splicing



Parameter Description	Concatenate 2 strings into 1 string
Examples of instructions	#Str2=asd #Str3=fgh

	#Str1=strcat(#Str2,#Str3)
operation result	#Str1=asdfgh

14.10.5 strleft (X, N) Takes N characters to the left of the string X

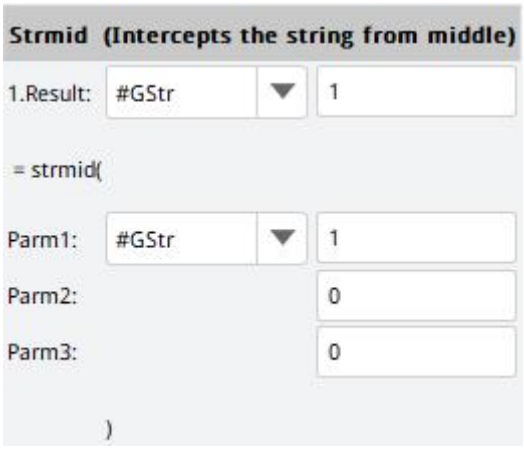
Parameter Description	Take N characters from the left of string X
Examples of instructions	#Str2=asdfgh #Str1=strleft(#Str2,2)
operation result	#Str1=st

14.10.6 strright (X, N) Takes N characters from the right of a string X

Parameter Description	Take N characters to the right of string X
Examples of instructions	#Str2=asdfgh #Str1=strright(#Str2,2)
operation	#Str1=gh

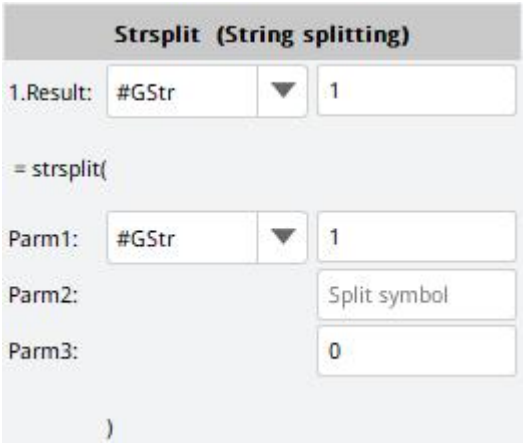
result	

14.10.7 strmid (X, M, N) Takes the M-th to N-th characters of the string X



Parameter Description	Take the M-th to N-th characters of the string X
Examples of instructions	#Str2=asdfgh #Str1=strmid(#Str2,3,4)
operation result	#Str1=df

14.10.8 strsplit (X, Y, N) String Split

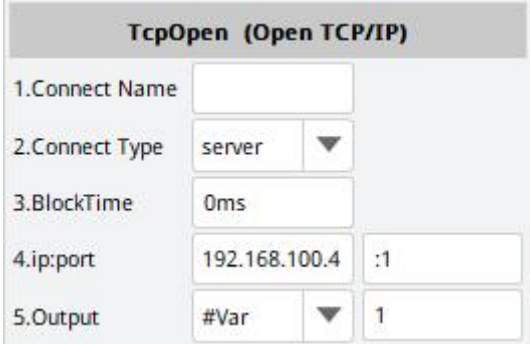


Parameter Description	Split the string X according to the separator Y, and take the Nth character
Examples of instructions	#Str2=a,s,d,f,g,h #Str1=strsplit(#Str2,",",3);

operation result	#Str1=d

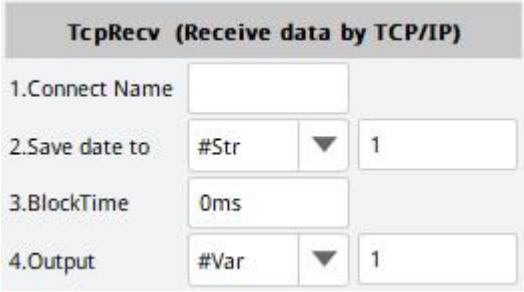
14.11 Communications

14.11.1 TcpOpen opens tcp/ip communication



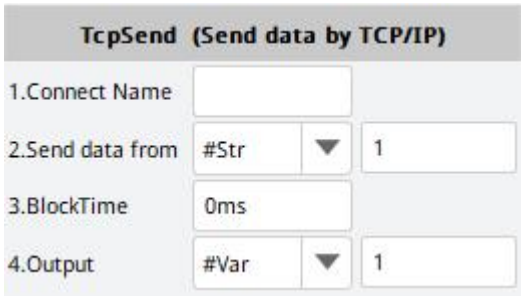
Examples of instructions	TcpOpen name="imf" type=""server"" block=1 addr="192.168.100.33:22" =>#Var1;	
Parameter Description	Connect Name	an identifier of the link, with which subsequent receive and send operations are associated;
	Connect type	link type, server: Create a server; client: Create a client and link to the server;
	block time	if that instruction is bloc, 1: Blocking time (ms); waiting time for instruction execution 0: non-blocking;
	ip:port	IP address of the link: port; the server can listen to the whole network port, so no ip is needed to create the server
	output	output values, >=0: Number of links maintained by the socket; -1: The server is starting, and the client is linking; -2: Failure;

14.11.2 TcpRecv Receiving Data



Examples of instructions	TcpRecv name="imf" dest=#Str1 block=0 =>#Var1;	
Parameter Description	Connect Name	receiving a data link identifier, created by TcpOpen;
	Save data to	a character string variable for storing the received data;
	Block time	if that instruction is bloc, 1: Blocking time (ms); waiting time for instruction execution 0: non-blocking;
	output	output values, >=0: Number of bytes received; -2: no corresponding identifier;

14.11.3 TcpSend Sending Data



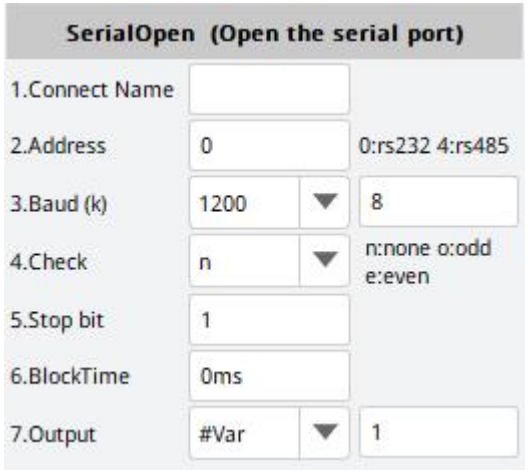
Examples of instructions	TcpSend name="imf" src=#Str1 block=0 =>#Var1;	
Parameter Description	Connect Name	Link identifier of the sending data, created by TcpOpen;
	Send data from	String variable for storing the sent data;
	block time	if that instruction is bloc, 1: Blocking time (ms); waiting time for instruction execution

		0: non-blocking;
	output	output values, >=0: Number of bytes sent; -3: no corresponding identifier;

14.11.4 TcpClose Closing a TCP/IP Connection

Examples of instructions	TcpClose name="imf";	
Parameter Description	Connect Name	Close link identifier, created by TcpOpen;

14.11.5 SerialOpen Receiving Data



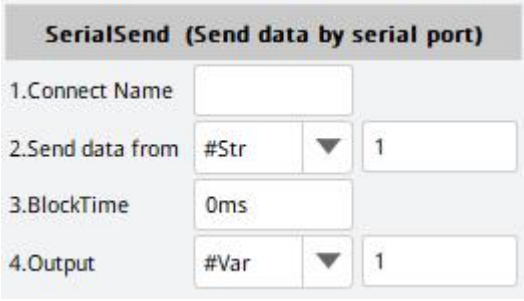
Examples of instructions	SerialOpen name="imf" addr=4 par="1200,8,n,1" block=0 =>#Var1;	
Parameter Description	Connect Name	Open the serial port identifier, and the subsequent received and sent data is related to this identifier;
	Address	Open the serial port address to 0, 1, 2 ..... To mark; 0: rs232; 4: rs485;
	Baud(k)	The number of symbols transmitted per second
	check	Select error detection mode, n: no check detection, o odd

		check, e even check
	stop bit	Indicates the end of a transfer and gives the computer an opportunity to correct clock synchronization. The more bits available for stop bits, the greater the tolerance for different clock synchronizations, but the slower the data transfer rate.
	Block time	if that instruction is bloc, 1: Blocking time (ms); waiting time for instruction execution 0: non-blocking;
	output	0: The serial port is opened normally; -1: failure;

14.11.6 SerialRecv Receiving Data

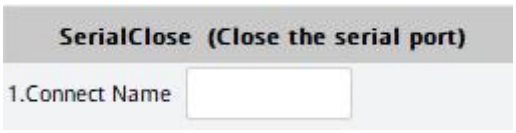
Examples	SerialRecv name="imf" dest=#Str1 block=0 =>#Var1;	
Parameter Description	Connect Name	Serial port identifier for receiving data, created by SerialOpen;
	Save data to	a character string variable for storing the received data;
	Block time	if that instruction is bloc, 1: Blocking time (ms); waiting time for instruction execution 0: non-blocking;
	output	output values, >=0: Number of bytes received; -1: Serial port is not open;

14.11.7 SerialSend Sending Data



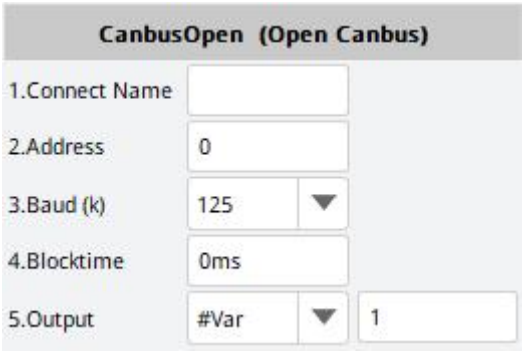
Examples of instructions	SerialSend name="imf" src=#Str1 block=0 =>#Var1;	
Parameter Description	Connect Name	Serial port identifier for sending data, created by SerialOpen;
	Send data from	String variable for storing the sent data;
	Block time	if that instruction is bloc, 1: Blocking time (ms); waiting time for instruction execution 0: non-blocking;
	output	output values, >=0: Number of bytes sent; -1 : Serial port is not open;

14.11.8 SerialClose Closing a TCP/IP Connection



Examples of instructions	SerialClose name="imf";	
Parameter Description	Connect Name	Close serial port identifier, created by SerialOpen;

14.11.9 CanbusOpen Receiving Data



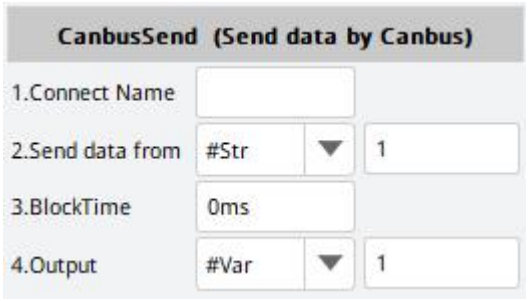
Examples of instructions	CanbusOpen name="imf" addr=1 par=125 block=0 =>#Var1;	
Parameter Description	Connect Name	Open the canbus identifier, and the subsequent received and sent data is related to this identifier;
	Address	Canbus address, marked with 0, 1, 2, currently there is only one canbus, fill in 0;
	Baud(k)	Canbus baud rate: 125 means 125k, in addition to the commonly used 1000k, 500k, 250k and so on;
	Blocktime	if that instruction is bloc, 1 : Blocking time(ms); waiting time for instruction execution 0 : non-blocking;
	Output	0: canbus is opened normally; -1: failure;

14.11.10 CanbusRecv Receiving Data

Examples of instructions	CanbusRecv name="imf" dest=#Str1 block=0 =>#Var1;	
Parameter Description	Connect Name	canbus identifier, created by CanbusOpen;
	Save data to	The received data frame will be converted into a string and stored in the form of"frame id, frame payload, standard

		frame/extended frame, data frame/remote frame, data;", for example: 000000FF,08,std,d,09,0A,0B,0C,0D,0E,0F,10;000000FE,8,ext,r,09,0A,0B,0C,0D,0E,0F,10, multiple frames of data received at one time, with ";"(a) Division;
	Block time	if that instruction is bloc, 1: Blocking time (ms); waiting time for instruction execution 0: non-blocking;
	output	output values, >=0: the number of received can frames; -1 : failure;

14.11.11 CanbusSend Sending Data



Examples of instructions	CanbusSend name="imf" src=#Str1 block=0 =>#Var1;	
Parameter Description	Connect Name	canbus identifier, created by CanbusOpen;
	Send data from	The string stored in the sent data frame, the format is the same as CanbusRecv
	Block time	if that instruction is block, 1: Blocking time (ms); waiting time for instruction execution 0: non-blocking;
	Output	output values, >=0: the number of received can frames; -1: failure;

14.11.12 CanbusClose Closing a TCP/IP Connection

Examples of instructions	CanbusClosename="imf";	
Parameter Description	Connect Name	Close the canbus identifier, created by CanbusOpen;

15 Error Messages

Error Code	Error message	Solution
0001	Hardware slave configuration fails, path is invalid, error code %1 (1: null; 2: length<3; 3: length>4096	Internal error, please upgrade or restore the program
0002	Hardware initialization fails, error code %1	Check that the hardware is powered on
0003	Hardware not ready, error code %1 (-1: ethercat disconnected; -2: slave number change; -13: hardware not register; -23: NIC driver abnormal; -33: cable disconnected)	Check whether the EtherCAT network communication is normal. Check if the number of slaves has changed. Check for registration.
0004	Axis %1 is not ready for interpolation, error code %2	Check whether the EtherCAT communication is normal, power off and restart
0005	Hardware slave %1 error, driver error code %2, ref to driver manual	Slave station error, refer to the handling method of specific slave station brand
0006	Read encoder (0 -based) %1 fails, error code %2	Check whether the EtherCAT communication is normal and the network cable is disconnected
0007	Axis %1 servo on fails, error code %2 (-1: ethercat not ready; -4: no power supply; -5:	Check whether the EtherCAT communication is normal, whether

	slave error)	the network cable is disconnected and whether it is powered on
0008	Axis %1 servo off fails, error code %2	Check whether the EtherCAT communication is normal and whether the network cable is disconnected and whether it is powered on
0009	Axis %1 clear error fails, error code %2	Check whether the EtherCAT communication is normal, whether the network cable is disconnected and whether it is powered on
000A	Axis %1 get slave error code fails, error code %2	Check whether the EtherCAT communication is normal, whether the network cable is disconnected and whether it is powered on
000B	Axis %1 get slave state fails, error code %2	Check whether the EtherCAT communication is normal, whether the network cable is disconnected and whether it is powered on
000C	Read profiler (0 -based) %1 fails, error code %2	Check whether the EtherCAT communication is normal and whether the network cable is disconnected and whether it is powered on
000D	IO is not ready, error code %1	Check IO
1001	Software configuration is invalid	Check the software setting parameters to see which parameter setting failed error message
1002	Software initialization fails	Check software setup parameters
1003	Axis reference calibration is invalid	Re-calibrate zero
1004	Axis %1 positive limit error, current %2 out of	Warning clears automatically after

	[%3, %4]	jog back to range of motion
1005	Axis %1 negative limit error, current %2 out of [%3, %4]	Warning clears automatically after jog back to range of motion
1006	Cartesion (XYZABC) %1 positive limit error, current %2 out of [%3, %4]	Warning clears automatically after jog back to range of motion
1007	Cartesion (XYZABC) %1 negative limit error, current %2 out of [%3, %4]	Warning clears automatically after jog back to range of motion
1008	Axis %1 overspeed, current %2, beyond limit %3	Manually Clear Errors
1009	%1 linear velocity overspeed, current %2, beyond limit %3	Manually Clear Errors
100A	%1 angular velocity overspeed, current %2, beyond limit %3	Manually Clear Errors
100B	Motion time out fatal error, current(ns) %1, beyond motion cycle(ns) %2	Please contact technical support or modify the planning cycle and EtherCAT cycle parameters
100C	Motion time out warning, current(ns) %1, beyond 80% of motion cycle(ns) %2	Please contact technical support or modify the planning cycle and EtherCAT cycle parameters
100D	Cannot open premotion log file	No treatment required
100E	Inverse kinematic fatal error, target point (KCS without tool, UUID=%1) is X: %2, Y: %3, Z: %4, A: %5, B: %6, C%7, pose idx: %8, error code %9	Check teaching point
100F	Target point is not reachable, target point (KCS without tool, UUID=%1) is: X: %2, Y: %3, Z: %4, A: %5, B: %6, C%7， pose idx: %8, error code %9	Check teaching point
1010	Movc points are co-linear, command UUID is %1	Check teaching point
1011	External cannot follow robot, as robot trajectory is too small, UUID = %1	Reteach robot trajectory

1012	Start point pose index %1! = End point pose index %2, UUID = %3	Check teach point posture
1013	Start point pose index %1! = Aux point pose index %2, UUID = %3	Check teach point posture
2001	Invalid parameter: profile ts %1, out of [0.0005, 0.016]	Check setup parameters
2002	Invalid parameter: profile size %1, out of [32, 512]	Check setup parameters
2003	Invalid parameter: smooth size %1, out of [0, 128]	Check setup parameters
2004	Invalid parameter: %1 encoder actual resolution %2	Check setup parameters
2005	Invalid parameter: %1 encoder feedback resolution %2	Check setup parameters
2006	Invalid parameter: %1 encoder sign %2	Check setup parameters
2007	Invalid parameter: %1 motor max speed %2	Check setup parameters
2008	Invalid parameter: %1 motor max current %2	Check setup parameters
2009	Invalid parameter: %1 motor max torque %2	Check setup parameters
200A	Invalid parameter: %1 axis positive limit %2, negative limit %3	Check setup parameters
200B	Invalid parameter: %1 axis max velocity %2, max jog velocity %3	Check setup parameters
200C	Invalid parameter: %1 axis, gear ratio in axis side %1 > gear ratio in motor side %2	Check setup parameters
200D	Invalid parameter: %1 axis sign %2	Check setup parameters
200E	Invalid parameter: Cartesion %1 positive limit %2, negative limit %3	Check setup parameters

200F	Invalid parameter: Cartesian %1 max velocity %2, max jog velocity %3	Check setup parameters
2010	Invalid parameter: Cartesian %1 sign %2	Check setup parameters
2011	Invalid parameter: Cartesian end linear max velocity %1, max acceleration %2, max deacceleration %3	Check setup parameters
2012	Invalid parameter: Cartesian end angular max velocity %1, max acceleration %2, max deacceleration %3	Check setup parameters
2013	Invalid parameter: Axis space end max velocity %1, max acceleration %2, max deacceleration %3	Check setup parameters
2014	Invalid parameter: External group %1, axis space end max velocity %2, max acceleration %3, max deacceleration %4	Check setup parameters
2015	Invalid parameter: %1 coordinate, index %2	Check setup parameters
2016	Invalid parameter: Axis %1 reference zero pulse, based on reference current axis is %2	Check setup parameters
3001	%1: Only works in %2 state	Check Controller Status
3002	%1: Not works in %2 state	Check Controller Status
3003	%1: Only works in Teach Mode	Check Controller Status
3004	%1: Hardware is not ready	Check the hardware
3005	%1: Command buffer is not ready (full or finished)	Internal error Contact Technical Support
3006	%1: Parameter %2 = %3 is invalid	Check setup parameters
3007	%1: Parameter %2 = %3 is invalid，valid value is between [%4,%5]	Check setup parameters
3008	%1: Parameter struct is invalid	Check setup parameters
3009	InitMotionController: Controller has been initialized	Internal error No treatment required

300A	StartMotionController: Controller has not been initialized, call imf::InitMotionController first	Internal error No treatment required
300B	StartMotionController: Controller has been started	Internal error No treatment required
300C	%1: Controller has not been started	Internal error No treatment required
300D	StartMotionController: Robot model dll path is invalid	Internal error Upgrade or restore your system
300E	StartMotionController: Robot model dll dose not contain specific robot type %1	Internal error Upgrade or restore your system
300F	StartMotionController: Cannot open %1 controller dll, system error info %2	Internal error Upgrade or restore your system
3010	StartMotionController: Creating controller with specific type (real/virtual/simulated) fails, system error info %1	Internal error Upgrade or restore your system
3011	StartMotionController: Creating %1 posix thread fails， system error info %2	Internal error Contact Technical Support
3012	LogMotionDataOn: Motion log path %1 is invalid	Internal error Upgrade or restore your system
3013	Jog: Only works in Teach Mode and Servo On state	Check Controller Status
3014	Jog: Jog coordinate is different from current working coordinate	Check the coordinate system parameters sent by Jog
3015	%1: Immediate motion command only works in Teach Mode, in Standby State and no immediate motion command is executing	Check the conditions for immediate command sending
3016	%1: Immediate command only works in Teach Mode	Check the conditions for immediate command sending
3017	%1: Immediate command only works in Teach Mode,	Check the conditions for immediate command sending
3018	%1: Velocity %2 is out of range (%3, %4] with	Check command parameters

	unit %5	
3019	%1: Blend radius %2 is out of range [%3, %4] with unit %5	Check command parameters
301A	%1: Target point is not reachable, UUID = %2	Check command target point
301B	%1: The first coordinate is system reserved, not allowing modification	Manual Clear No treatment required
301C	%1: Only suitable for Real Controller Mode	Internal error No treatment required
301D	%1: Open dll %2 fails, %3	Internal error Upgrade or restore your system
301E	%1: %2 dll is not loaded	Internal error Upgrade or restore your system
301F	%1: %2 function is not registered, cannot find specified function	Internal error Upgrade or restore your system
50000	Emergency stop button is pressed	Release the emergency stop
50100	system IO [%1] is triggered	
50101	system IO [%1] is triggered	
50102	system IO [%1] is triggered	
20000	undefined data type %1	Internal error, cannot be resolved after restart, contact technical support
20001	no project has been loaded	Please open a project first
21000	mysql exec error %1	Database Internal error, cannot be resolved after restart, contact technical support
21001	mysql data format error, missing col %1 or data type not compatible	Database Internal error, cannot be resolved after restart, contact technical support
21002	get failed, id %1 not exist in %2	Database Internal error, cannot be resolved after restart, contact technical support
21003	set key %1 of table %2 failed, must in	Database Internal error, cannot be

	range([%3, %4])	resolved after restart, contact technical support
21004	set key %1 of table %2 failed, address out of bounds: offset + size = %3, size of target = %4	Database Internal error, cannot be resolved after restart, contact technical support
21005	set key %1 of table %2 failed, key not exist	Database Internal error, cannot be resolved after restart, contact technical support
21006	error loading project %1, not exist	Database Internal error, cannot be resolved after restart, contact technical support
21007	error loading project %1, missing main	Database Internal error, cannot be resolved after restart, contact technical support
21008	user data table %1 not found	Database Internal error, cannot be resolved after restart, contact technical support
22000	modbus error %1	Communication error, cannot be resolved after restart, contact technical support
30000	system init ...	No treatment required
30001	system init timeout, over %1 s	No treatment required
30002	unknown motion mode %1, real will be used.	Configuration file error, cannot be resolved after restart, contact technical support
30003	no ecat config file, init abort	Configuration file error, cannot be resolved after restart, contact technical support
31000	api calling error %1	Failed to call API. Unable to solve after restarting. Contact technical support
31001	motion Internal error %1	Unable to resolve after restart,

		contact technical support
32000	command %1, missing keyword %2	Instruction format error, check program
32001	command %1, unexpected value of keyword %2	Instruction format error, check program
32002	command %1, invalid value of keyword %2	Instruction format error, check program
32003	unknown robot command: %1	Instruction format error, check program
40000	%1: error parsing IO variable(%2)	Instruction format error, check program
40001	%1: error parsing String(%2)	Instruction format error, check program
40002	%1: error parsing operator(%2)	Instruction format error, check program
40003	%1: error parsing number(%2)	Instruction format error, check program
40004	%1: error parsing integer(%2)	Instruction format error, check program
40005	%1: error parsing system variable(%2)	Instruction format error, check program
40006	%1: unbalanced multi-line comment	Instruction format error, check program
41000	%1: invalid symbol(%2) in expression	Instruction format error, check program
41001	%1: unknown built-in func %2	Instruction format error, check program
41002	%1: expression terminated unexpectedly	Instruction format error, check program
41003	%1: unbalanced parenthesis in expression	Instruction format error, check program
41004	%1: invalid ++/ - - in expression	Instruction format error, check

		program
41005	%1: undeclared symbol %2	Instruction format error, check program
41100	%1: invalid statement declaration with %2	Instruction format error, check program
41101	%1: %2 statement missing symbol %3	Instruction format error, check program
41102	%1: %2 statement missing keyword %3	Instruction format error, check program
41180	%1: break/continue only works with in for/while	Instruction format error, check program
41181	%1: for statement loop variable not left -value	Instruction format error, check program
41182	%1: assignment left term not left -value	Instruction format error, check program
41200	%1: %2 all robot cmd admits only one output term	Instruction format error, check program
41201	%1: %2 output term must be placed in the end for all robot cmd	Instruction format error, check program
41202	%1: %2 missing symbol %3	Instruction format error, check program
41203	%1: %2 missing keyword %3	Instruction format error, check program
41204	%1: %2 incomplete key-value pair detected	Instruction format error, check program
41205	%1: %2 key -value pair count check fail, %3 required	Instruction format error, check program
41300	%1: program declaration missing keyword %2	Instruction format error, check program
41301	%1: program declaration missing symbol %2	Instruction format error, check program
41302	%1: program declaration missing name	Instruction format error, check

		program
41303	%1: function name (%2) differs from file name (%3)	Instruction format error, check program
41304	%1: function declaration, missing return type	Instruction format error, check program
41305	%1: function declaration, invalid return type %2	Instruction format error, check program
41306	%1: function declaration, duplicate argument %2	Instruction format error, check program
41307	%1: function declaration, invalid argument name %2	Instruction format error, check program
41308	%1: function declaration, invalid argument type %2	Instruction format error, check program
41309	%1: wrong argument number: provided = %2, expected = %3	Instruction format error, check program
41304	%1: function argument type check failure	Instruction format error, check program
42000	%1: operands type check fail %2	Instruction format error, check program
42001	%1: division by 0, %2	Instruction format error, check program
42002	%1: invalid operand value %2	Instruction format error, check program
42003	%1: symbol %2 not exist	Instruction format error, check program
42004	%1: index (%2) exceeds array bounds %3	Instruction format error, check program
42005	%1: field (%2) not exist in struct	Instruction format error, check program
42100	%1: function %2 arg count check failure, %3 provided, %4 needed	Instruction format error, check program
42101	%1: function %2 with the %3 -th input null	Instruction format error, check

		program
42102	%1: function %2 division by 0	Instruction format error, check program
42103	%1: function %2 arg type check faliure (%3)	Instruction format error, check program
42104	%1: function %2 arg range check faliure %3	Instruction format error, check program
42200	%1: %2 passing array as parameter is forbidden, key = %3	Instruction format error, check program
42201	%1: %2 group and index of DIO must be integers %3	Instruction format error, check program
42202	%1: %2 index of AIO must be integers %3	Instruction format error, check program
42203	%1: %2, key %3 must be %4 type	Instruction format error, check program
42300	%1: unknown struct field %2	Instruction format error, check program
42301	%1: manipulation of struct field %2 is forbidden	Instruction format error, check program
42302	%1: IO variable %2 out of bounds	Instruction format error, check program
42303	%1: undefined coord type %2	Instruction format error, check program
43000	file %1 not exist	Program does not exist, check if program exists
43001	duplicate command name %1, last found in %2	Instruction format error, check program
43002	duplicate command type %1, last found in %2	Instruction format error, check program
43003	execute %1 isnforbidden, type = FUNCTION	Instruction format error, check program
43004	program %1, start line %2 exceeds limit	Startup line number out of limit

60000	invalid registration file	Error in registration file, which cannot be solved after restarting. Contact technical support
60001	invalid registration code	Error in registration file, which cannot be solved after restarting. Contact technical support
61000	app id = %1 is registered, but cannot be loaded	The program file is missing and cannot be solved after restarting. Contact technical support
61001	apps %1 and %2 have duplicate id = %3	Process plug-in error, cannot be resolved after restart, contact technical support
61002	apps %1 and %2 accepts common command %3	Process plug-in error, cannot be resolved after restart, contact technical support