iManifold

INSTRUCTION

iManifold Robot Control System

User Manual

Copyright

Statement

Shenzhen iManifold Robot Technology Ltd. reserves the right to modify the product or its features without prior notice.

Shenzhen iManifold Robot Technology Ltd. does not assume the responsibility for direct or indirect injury or damage caused by improper use of the product.

Contact Us

Shenzhen iManifold Robot Technology Ltd.

Address: F508, 5th Floor, Building F, Huafeng International Robot Industrial Park, Bao'an District, Shenzhen

iManifold Technology (Jiangyin) Ltd.

Address : Room 102, Building 60, Tianan Digital City, No. 55 Changshan Avenue, Jiangyin City, Wuxi City, Jiangsu Province

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 robots and is connected to the 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.

1.3 DH Model

The Denavit Hartenberg (DH) model is a modeling technique applied in robot kinematics. It establishes a coordinate system on each connecting rod and uses homogeneous coordinate transformation to achieve coordinate transformation between the connecting rods.

The DH model is used to describe the different configurations of industrial robots supported by the robot system. The following are various robot configurations supported by the system.

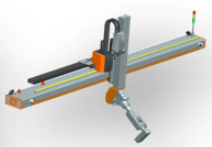
机器人构型 RobotType	关节配置Axes			应用工艺 (Applications)						
	Ground	Hand	Aux	Tracking	Arc Welding	Laser Cutting	Palletize	CNC	Vision	Bending
Gantry 	LL-	R - -	8	-	✓	✓	-	-	-	-
	LL-	RR -	8	-	✓	✓	-	-	-	-
	LL-	RRR	8	-	✓	✓	-	-	-	-
	LLL	- - -	8	✓	✓	✓	✓	✓	✓	-
	LLL	R - -	8	✓	✓	✓	✓	✓	✓	-
	LLL	RR -	8	✓	✓	✓	✓	✓	✓	✓
	LLL	RRR	8	✓	✓	✓	✓	✓	✓	✓
SCARA 	RRL	R - -	8	✓	✓	✓	✓	✓	✓	-
	LRR	R - -	8	✓	✓	✓	✓	✓	✓	-
	RLR	R - -	8	✓	✓	✓	✓	✓	✓	-
Articulate 	RRR	RRR	8	✓	✓	✓	✓	✓	✓	✓
	RRR	R - R	8	✓	✓	✓	✓	✓	✓	✓
	RRR	- RR	8	✓	✓	✓	✓	✓	✓	✓
	RRR	- R -	8	✓	✓	✓	✓	✓	✓	✓
	RLL	- - R	-	✓	✓	✓	✓	✓	✓	-
	RRR	- - R	-	✓	✓	✓	✓	✓	✓	-
	RL-	LRR	8	-	-	-	✓	-	-	-
	LRL	- - -	8	-	-	-	✓	-	-	-
	LRL	R - -	8	-	-	-	✓	-	-	-

机器人构型 RobotType	关节配置Axes			应用工艺 (Applications)						
	Ground	Hand	Aux	Tracking	Arc Welding	Laser Cutting	Palletize	CNC	Vision	Bending
Delta 	RR-	- - -	-	✓	✓	✓	-	-	-	-
	RRR	- - -	-	✓	✓	✓	-	-	-	-
	RRR	- RR	-	✓	✓	✓	-	-	-	-
	RRR	- R -	-	✓	✓	✓	✓	✓	✓	-
Tripod 	LLL	- - -	-	✓	✓	✓	✓	✓	✓	-
	RRR	- - R	-	✓	✓	✓	✓	✓	✓	-
Customize 	LL-	RR-	8	✓	✓	✓	✓	✓	✓	✓
	LLR	RR -	8	✓	✓	✓	✓	✓	✓	✓
	LLR	RRR	8	✓	✓	✓	✓	✓	✓	✓
Painting 	RRR	RRR	8	✓	✓	✓	✓	✓	✓	-
	RRR	RRR (R)	8	✓	✓	✓	✓	✓	✓	-
Steward 	6-SPS		0	✓	-	-	-	-	✓	-
	6-UPU		0	✓	-	-	-	-	✓	-

0







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

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.

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 electric 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 electric 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.

1. When installing a tool on the robot, be sure to CUT OFF the power supply on the electric 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.

2. 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.
3. Whenever possible, the teaching work should be carried out outside the work area.
4. The following cautions shall be observed when performing teaching work within the robot motion range:
5. 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

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 **electric 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 electric 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

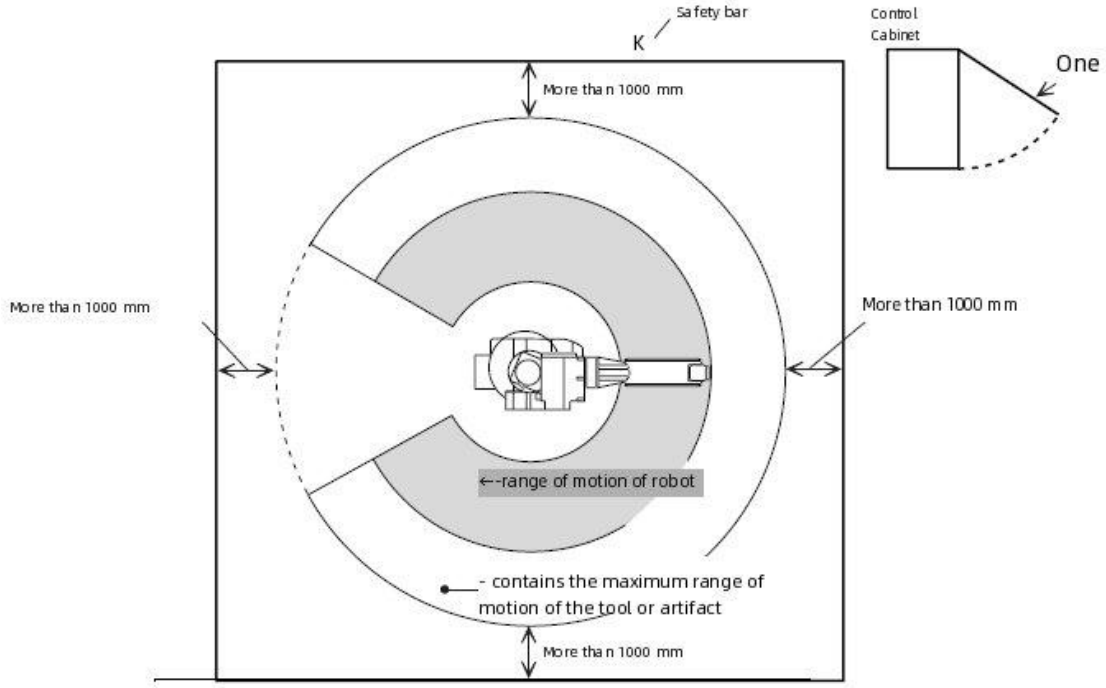


Diagram 1 Installation Location Requirements

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

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

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

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

3.1.3 Precautions for Installation and Wiring

危险

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 electric 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 electric 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.

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

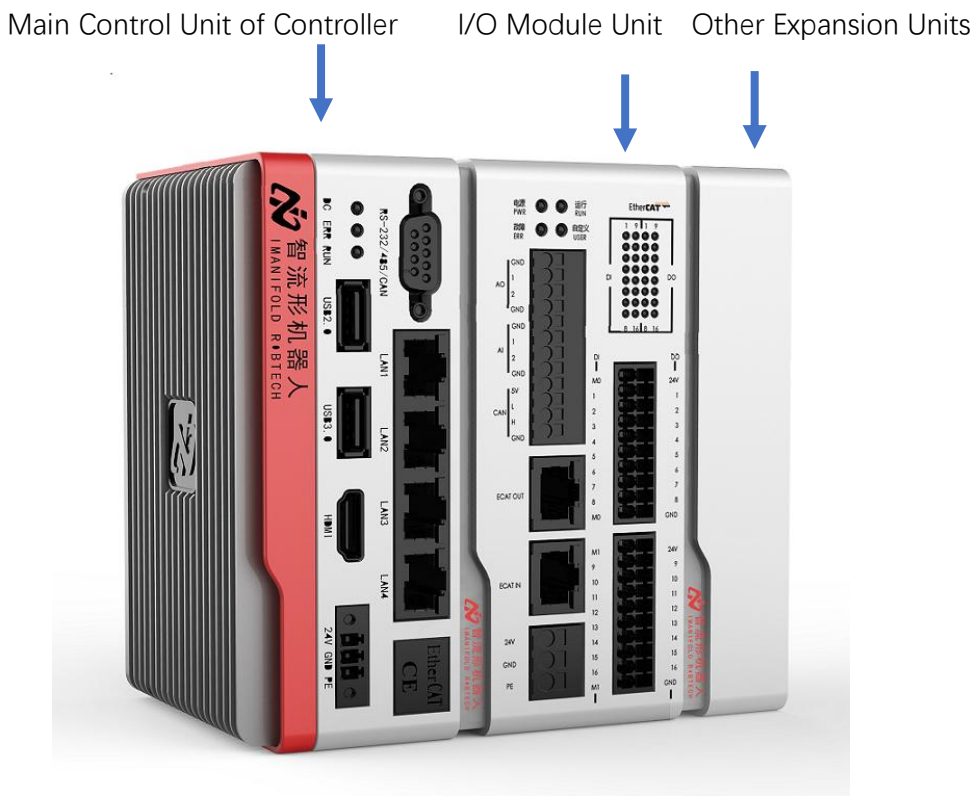
2. Wiring must be carried out by authorized staff. Incorrect wiring may cause fire or electric shock.

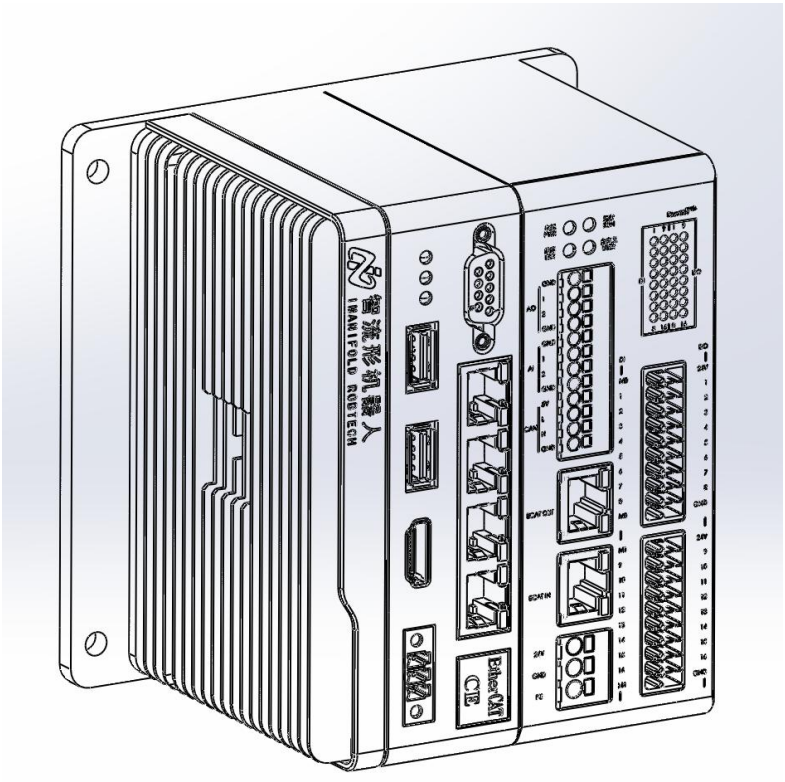
3. Please wire according to the rated capacity specified in the manual. Incorrect wiring may cause fire alarm or mechanical damage.

4. 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.

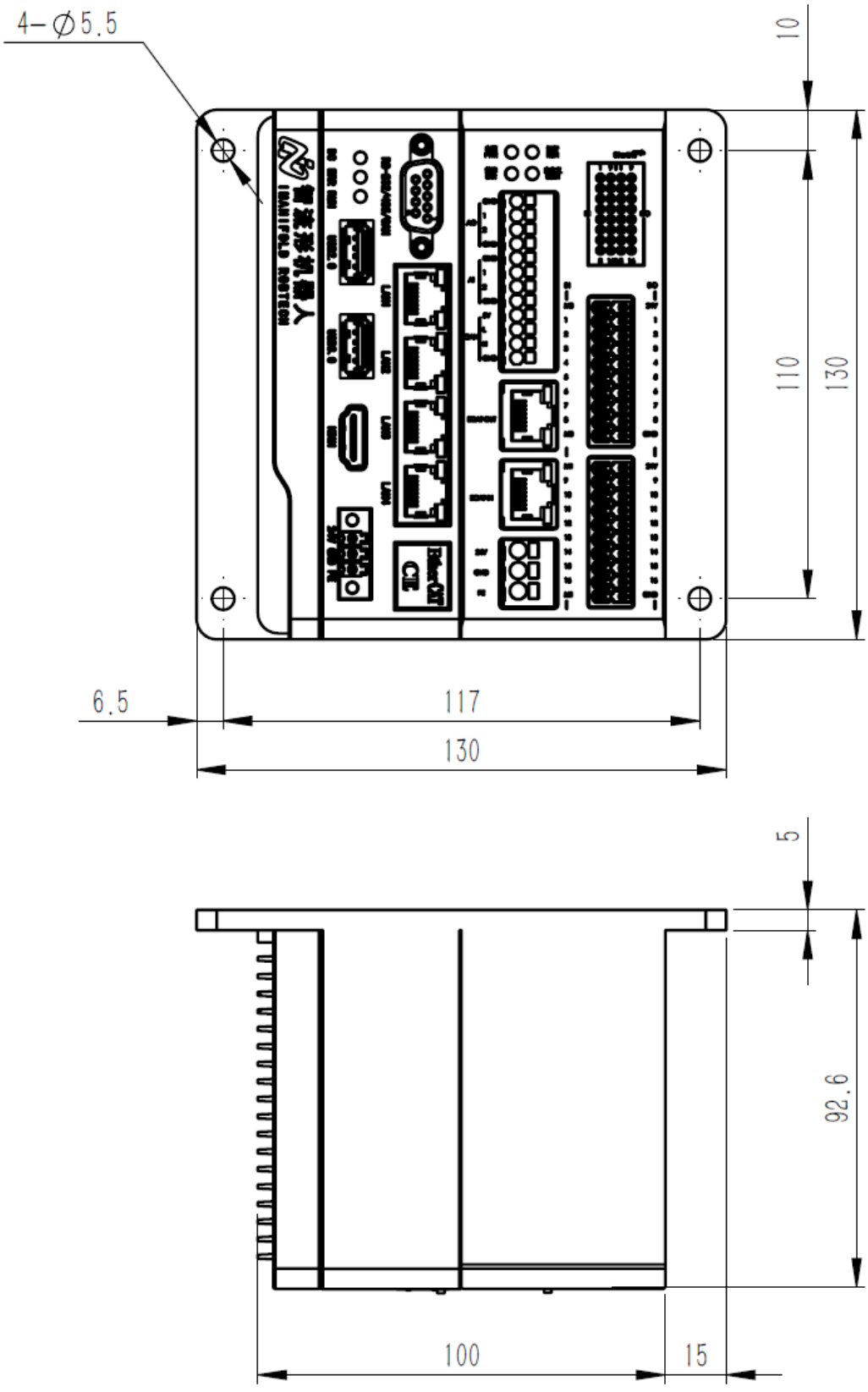
5. Do not touch the substrate directly with your hands. Integrated circuit (IC) substrates may fail due to static electricity.

3.2 Controller Installation Instructions



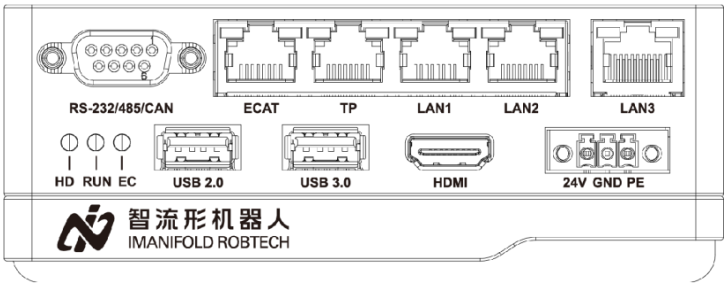


3.2.1 Installation Mode of Main Control and I/O Unit



3.2.2 Main Control Unit Wiring Description

3.2.2.1 Interface Description of Main Control Unit

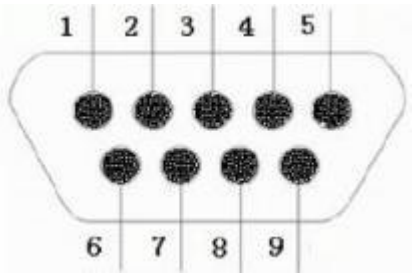


Interface Name	Description
+24V (must be connected)	给 24V 供电
GND	DC power ground
PE	Connect reliability grounding and shielded terminals
TP (to teach pendant) (must be connected)	The other end is connected to the handheld box network port, the default IP: 192.168.100.3
Ecat（must be connected）	Connection with EtherCAT driver cable (standard, network cable interface)
LAN1（optional）	Default IP：192.168.101.3
LAN2（optional）	Default IP：192.168.102.3
LAN3（optional）	Default IP：192.168.103.3
RS232/CAN/485 combination port （optional）	For external communication
USB2.0/USB3.0（optional）	External USB keyboard, mouse or U disk

HDMI（optional）	External HDMI interface display

Diagram 2 Controller interface

3.2.2.2 RS-232/485/CAN 接口定义



1	**VCCS_CAN**	**CAN 电源 24V**
2	RS232_RX	CAN power supply 24V
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 receiving
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

Diagram 3 Serial Port

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, flash twice.

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

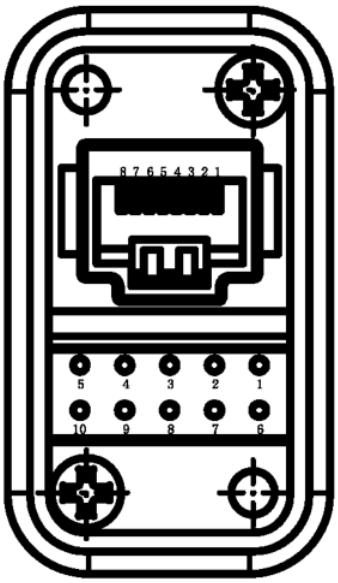


Diagram 4 Teaching Pendant Aerial Insertion

Definition of External Cable of Teach Pendant:

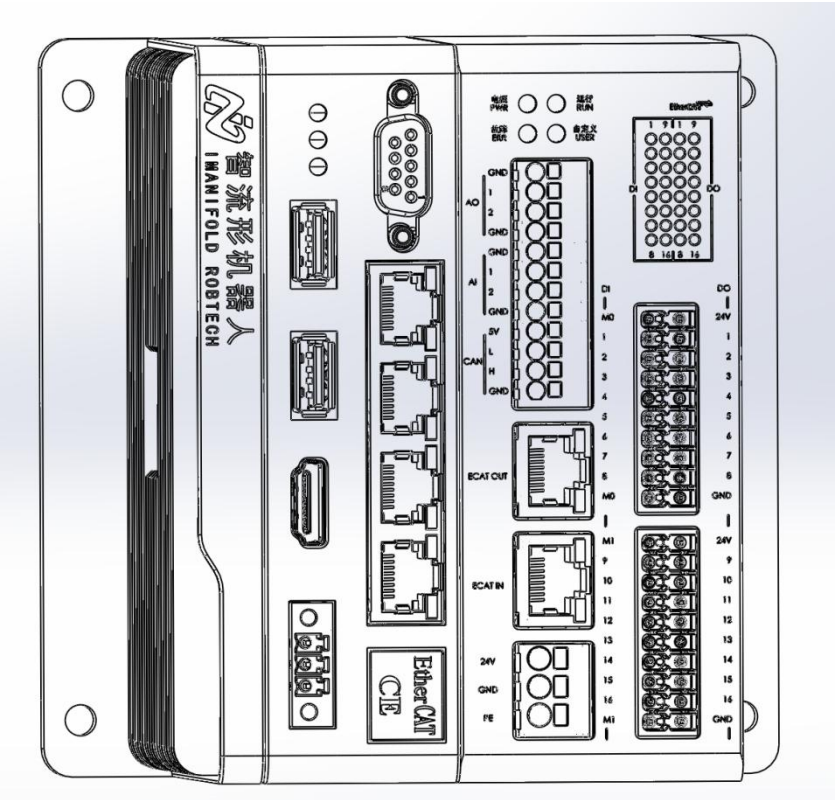
1	**Red**	**24V+**	**Connect 24V power supply +**
2	White & Red	0V	Connect 24V power supply 0V
3	Black	PE	Grounding
4	Green	EMG1-1	Emergency stop terminal 1
5	Black&Green	EMG1-2	Emergency stop terminal 1
6	Blue	EMG2-1	Emergency stop terminal 2
7	Black&Blue	EMG2-2	Emergency stop terminal 2

(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



焊接机器人专用EtherCAT从站I/O模块

额定电源：24VDC,+-20%

IO类别	通道数	信号类型	信号电平	备注
DI	16	兼容PNP/NPN	0V/24V,500mA	“1”信号电平：15~24VDC； “0”信号电平：0-5VDC；
DO	16	NPN	0V,500mA	“1”信号电平：24VDC； “0”信号电平：Max:1.5VDC,10kΩ负载；
AI	2	电压	0-10V	单极、12BIT、二线制，输入阻抗500K欧；
AO	2	电压	0-10V	单极、12BIT、二线制，最小负载阻抗1K欧，最大电流10mA；
CAN	1	信号	5V	CAN-5V,CAN-L,CAN-H,CAN-GND

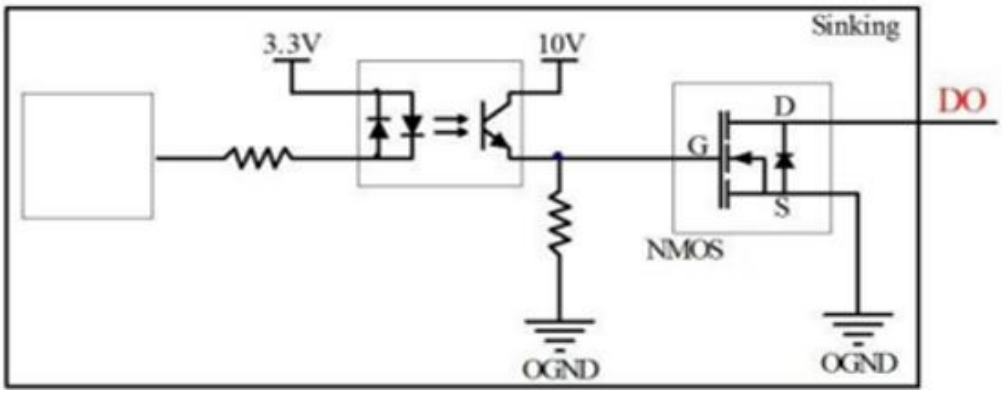


Diagram 5 Schematic Diagram Of Leakage Type (bottom edge) Output Circuit

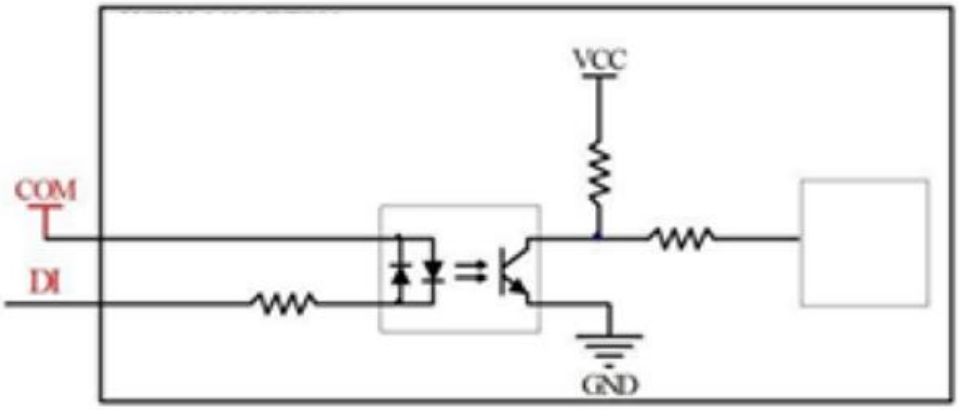


Diagram 6 General Input Circuit Diagram

3.3 Power Supply

3.3.1 Three-Phase Power Supply

The three-phase power supply is composed of AC 200V, 50Hz or AC 220V, 50/60Hz.

When there is a temporary power supply frequency interruption or voltage drop, the power outage processing circuit will operate such that the servo power supply is cut off.

Connect the power supply of the electric control cabinet to a stable input power supply with small voltage fluctuations.

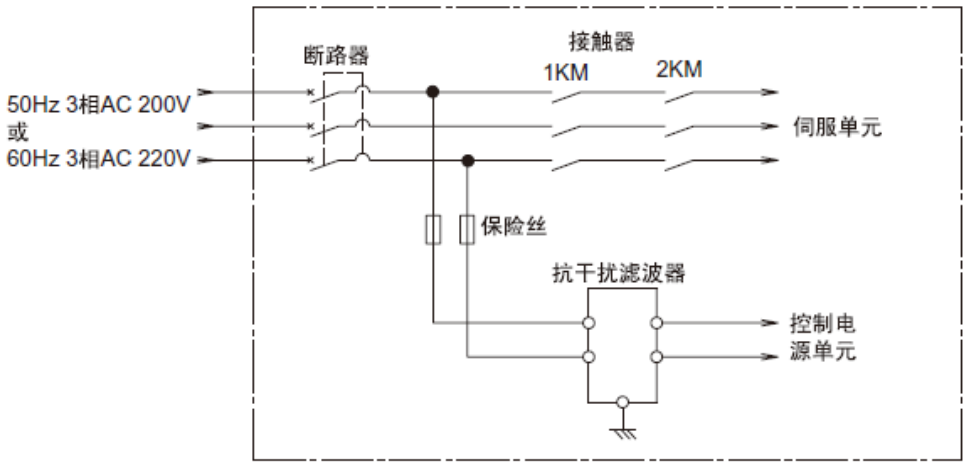


Diagram 7 Input Connections For Power Supply

3.3.2 Installation of Noise Filter

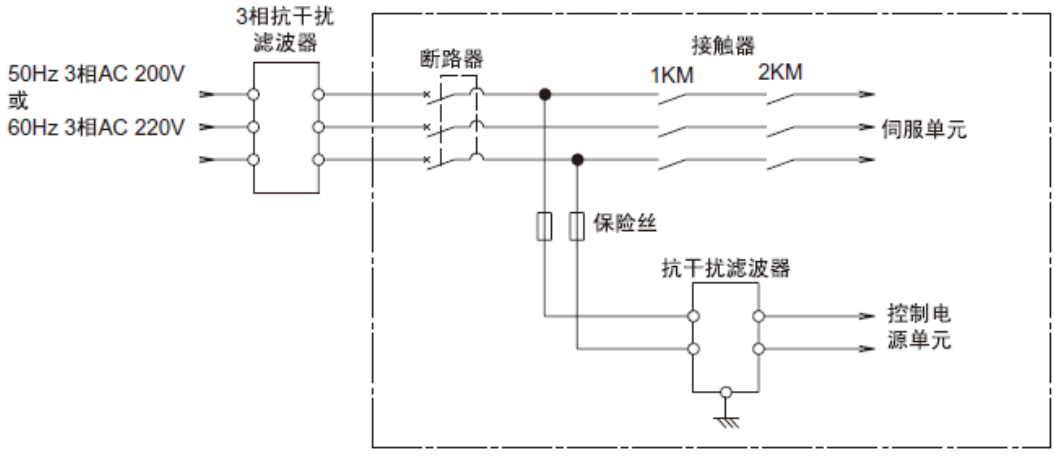


Diagram 8 Connection Of Three-phase Noise Filter

3.3.3 Installation of Leakage Circuit Breaker

If you connect a leakage circuit breaker to the power supply of the **electric control cabinet**, you need to use a leakage circuit breaker that can prevent high frequency, which can prevent the malfunction caused by the high frequency leakage current of the rectifier.

Even with a leakage circuit breaker installed, the NX100's rectifier still has the possibility of high-frequency current leakage, but there is no danger of such current leakage.

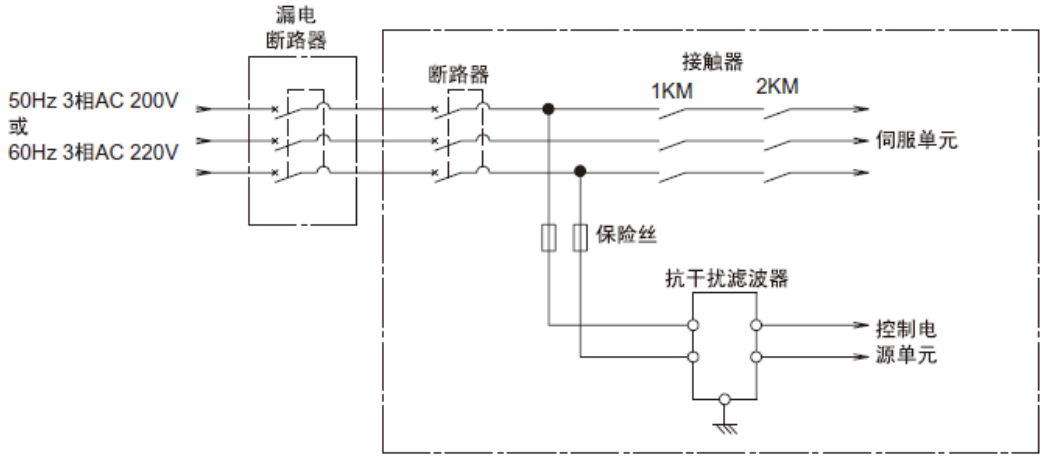


Diagram 9　Connection Of Leakage Circuit Breakers

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 key 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 fault 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 key is pressed, the servo power will not be turned on. Method of opening the emergency stop button: rotate clockwise until the emergency stop button is activated. With the sound of “click”, it indicates that the emergency stop button has been opened.
2	Pause		**Press this key and the robot will stop running.** **This key can be used in any mode.** In teach mode: When this light is pressed, the light will be on. At this time, the robot cannot perform axis

			operation. Playback mode: Press this key indicator for the next 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 playback.** This indicator lights up when playback mode is running. Start the robot by a special input start signal. This indicator lights up when playback is running. The mode knob must be set to playback mode before pressing this button; Make sure the servo Ready indicator light is on.
4	Mode		**Playback mode, teach mode, or 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: Playback mode

			You can run the program back 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 teaching mode → click the [Servo Ready] key on the hand-held teach pedant (the indicator light of [Servo Ready] is blinking) → gently hold the three-step switch and 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 playback 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, gently shake hands and hold the [three-step switch] on the operation teach pedanta. 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		**Secondary 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 playback mode, switch robot/transformer, 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 manipulated by holding down two or more keys. The robot runs according to the selected coordinate system and manual speed.

			Please make sure that the set coordinate system and speed are appropriate before axis operation. Before operation, it is necessary to confirm that the Servo Ready indicator light on the hand-held teach pedant of the robot is on.
14	Forward and Backward		When the servo power is on, press and hold this key, and the robot will run step by step (in reverse) according to the program point track shown. This key must be used in Teach mode.
15	Speed Increased or Reduced		**Set the speed of robot Teach mode.** The low speed gears are 1%, 2%, 5%, and +/-5% each time you press up. The set speed is displayed in the status bar. [Upper] + [V+] / [V-] can adjust the playback mode speed.

4.2 Teach Pendant Interface

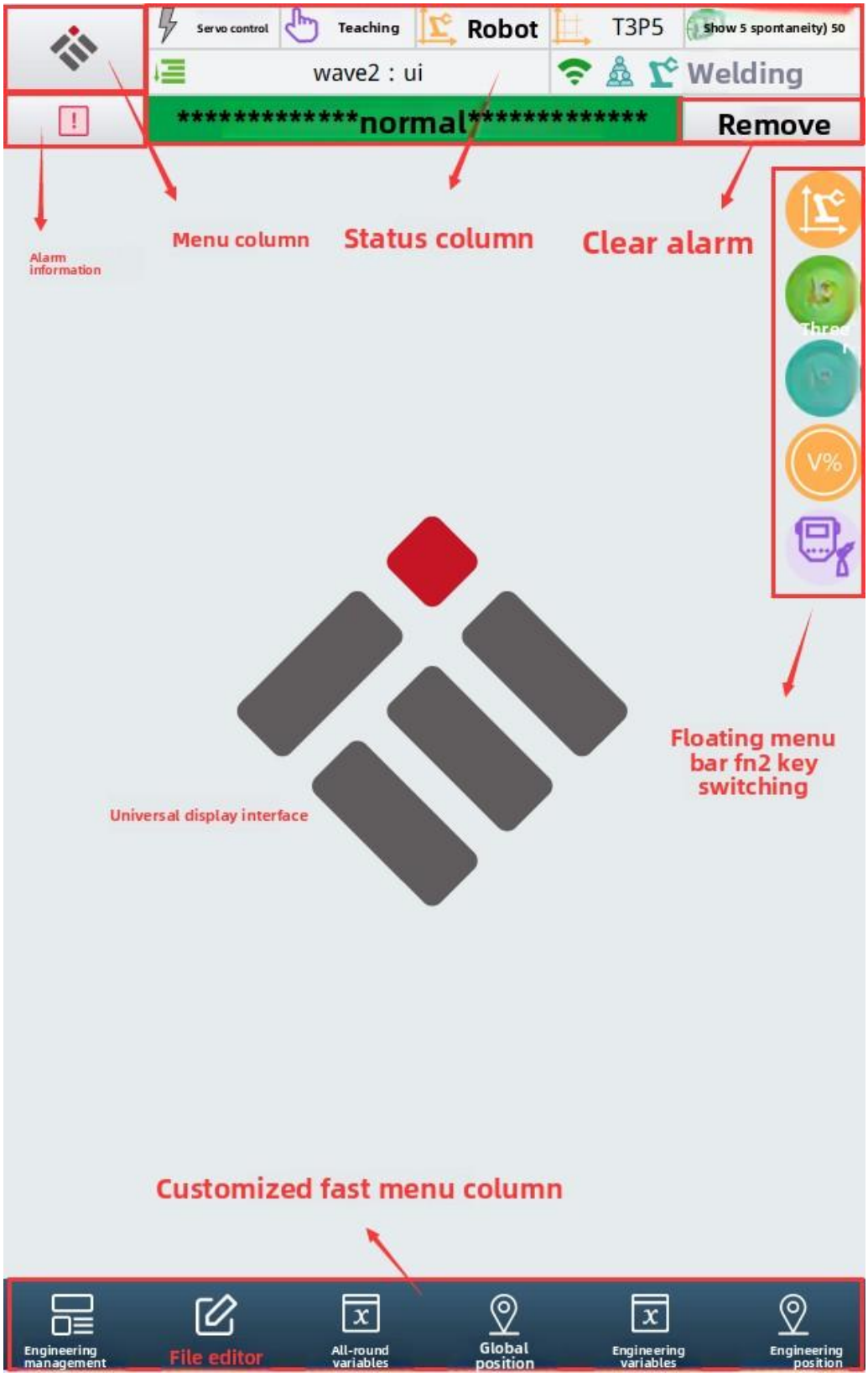
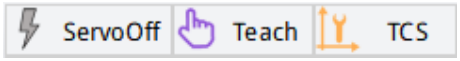
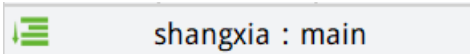
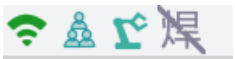
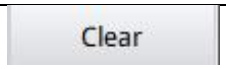
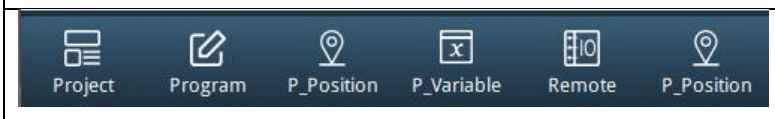


Diagram 10 Teaching Pendant Interface

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 and the piece coordinate system; Display speed, the former is the speed of teach pedant mode, the latter is the speed of playback mode
	Display execution cycle mode (cycle, single, single step), current project name and program name
	Teach pedant connection state, current permissions, current coordinate system, and other process states
	You can use this button to view log and historical alarm information.
	Error and operation prompt or alarm, no alarm display normal, display movement speed
	Button for clearing alarm
	Common application toolbar (customizable)

4.3 System Application Menu

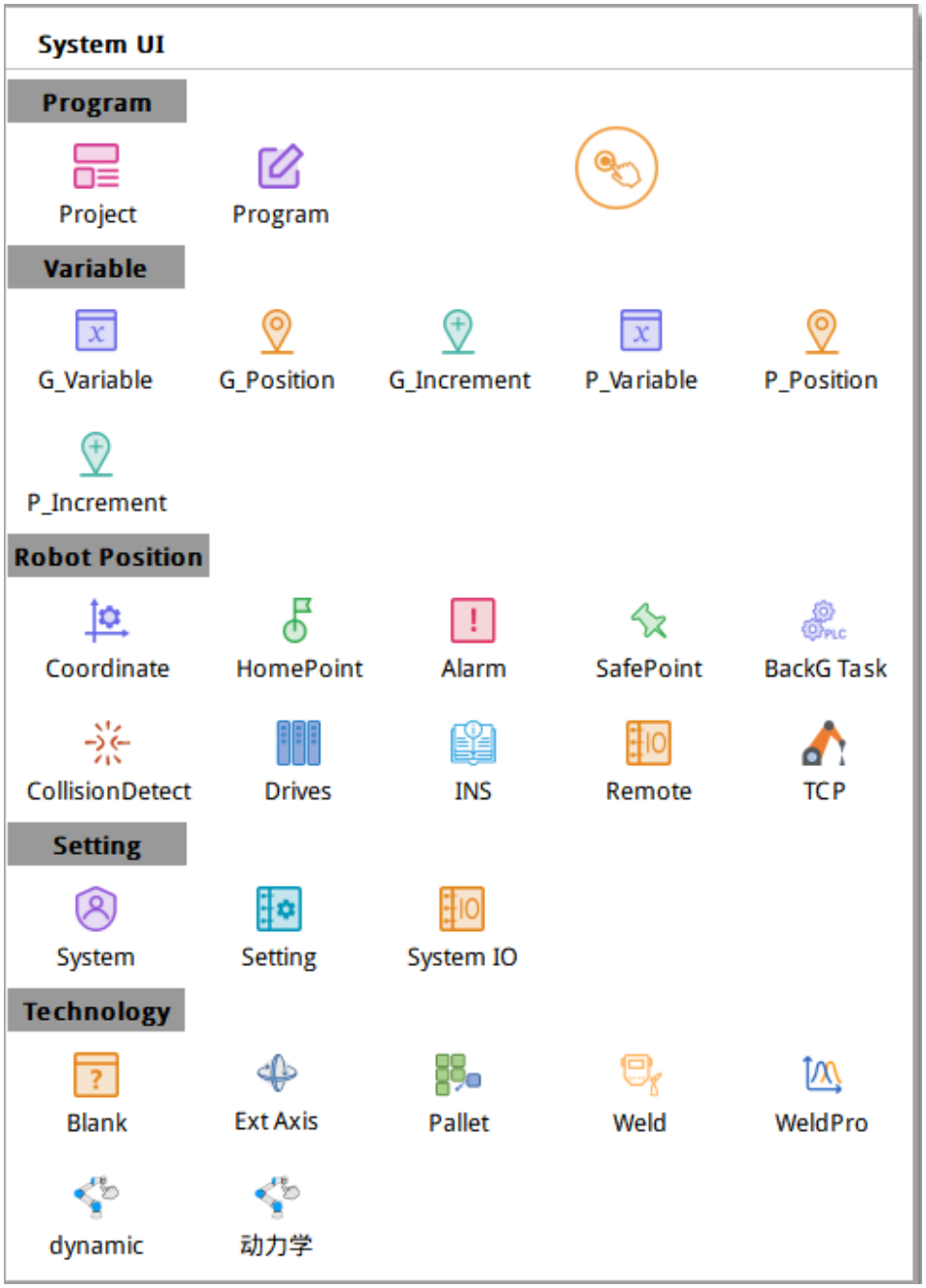


Diagram 11 Menu Interface

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	Instruction
	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
	Engineering Management	1. Launch 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
	Document 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.

	All-Office Position	The position variable can be calibrated, and can be used in different projects for program file editing.
	All-Office Increment	Can write positional increments, can be used in different projects for program file editing.
	Engineering Variable	Project numeric variables and project strings can only be edited and used in the currently opened project
	Engineering Position	Position variables can be calibrated and can only be edited and used in the currently opened project
	Engineering Increment	You can write positional increments, which can only be edited and used in the currently open project
	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
	Zero Point Calibration	The zero point of the robot is calibrated, manually modified, and the absolute encoder is cleared
	Abnormal Treatment	Axis Simulation Setting Axis motion limit switch
	HOME Coordinate	Set home position (return to zero)
	Backstage Task	Background task setting, reservation function
	Rights Management	Set administrator permissions. Different permissions have different operation contents. Version Information

	System Parameter	Settings: System Parameters, Hardware Parameters, Joint, Auxiliary Axis, Cartesian, Planning Parameters, DH Parameters
	Undefined	Process parameter setting

4.4 Status Bar

4.4.1 Display Servo Status

	Servo off
	Servo on

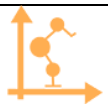
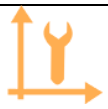
4.4.2 Working Mode

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

	Teach: The robot is in teach mode.
	Playback: The robot is in playback 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.

	Axis coordinate system
	Kinematic coordinate system
	Piece coordinate system
	World coordinate system
	Tool coordinate system

4.4.4 Current Tool#and Workpiece#

It is convenient for the user to determine the number of the tool (workpiece) currently used. The program uses a tool (piece) coordinate system data queue with 32 elements. The default number 1 is no tool (workpiece), and the coordinate system queue element 2-32 is the editable queue element.

4.4.5 Speed

Display the current speed by pressing the [V+] or [V-] key on the hand-operated instruction device. The former is the speed of teach mode and the latter is the speed of playback mode.

4.4.6 Teaching File Playback Execution Mode

	Unlimited cycle: the teaching program is executed in a non-stop cycle
	Single cycle: only perform the teaching program once
	Single operation: only one 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

		
Action Mode	Administration Mode	Reset

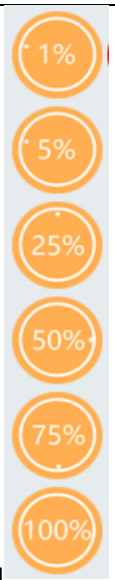
4.4.7.2 Network Status

	
Connected	Disconnected

4.5 Floating Menu

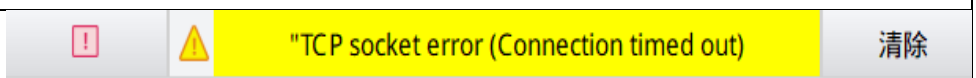
The floating menu pops/closes by using the [Fn2] key on the hand-operated teach pedant with quick switching function.

Level 1	Level 2	Function
	Jog coordinate system switching 	Switch to axis coordinate system
		Switch to kinematic coordinate system
		Switch to piece coordinate system
		Switch to world coordinate system
		Switch to tool coordinate system
	Playback program execution mode 	Infinite cycle
		Single execution
		One-line execution
	Jog axis switch 	Switch to robot axis (hold [J1] - [J8] represents robot axis) Switch to auxiliary axis (hold [J1] - [J8] for external auxiliary axis)
		Switch to robot axis (hold [J1] - [J8] represents robot axis) Switch to auxiliary axis (hold [J1] - [J8] for external auxiliary axis)

	 Speed percentage switching	Percentage of the speed at which the current mode is switched

4.6 Robot Status and Alarm History Viewing

4.6.1 Robot Status

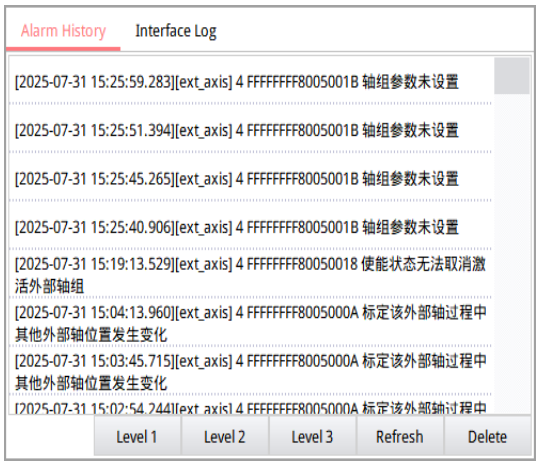
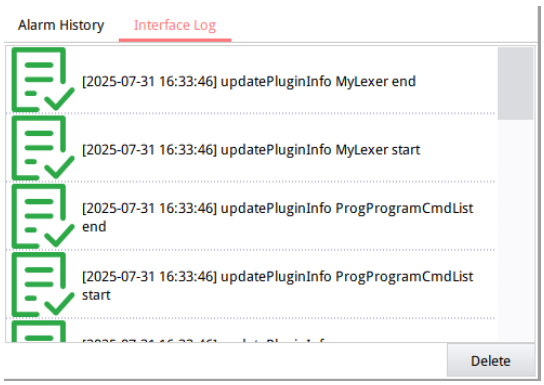
Status	Description
Normal	 When the robot is in normal operation state, it can operate the servo motion on the robot and execute the teaching program.
Servo Ready	 The robot is in servo-powered state
In execution	 The robot is moving or performing an instruction program, and the following number indicates the current linear velocity at the end of the robot
Yellow Alarm	 The robot is in the 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
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

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

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

危险

When opening the main power switch on the upper electric control cabinet,make sure that there is no person within the robot action range.Neglect of this prompt may result in accidental contact with the robot and personal injury.In case of any problem,press the emergency

5.1.1 Switch on the Main Power Supply

Turn the main power switch ON the side panel of the electric control cabinet to the ON position, then the main power is on. Press the green servo start button on the panel of the electric control cabinet.

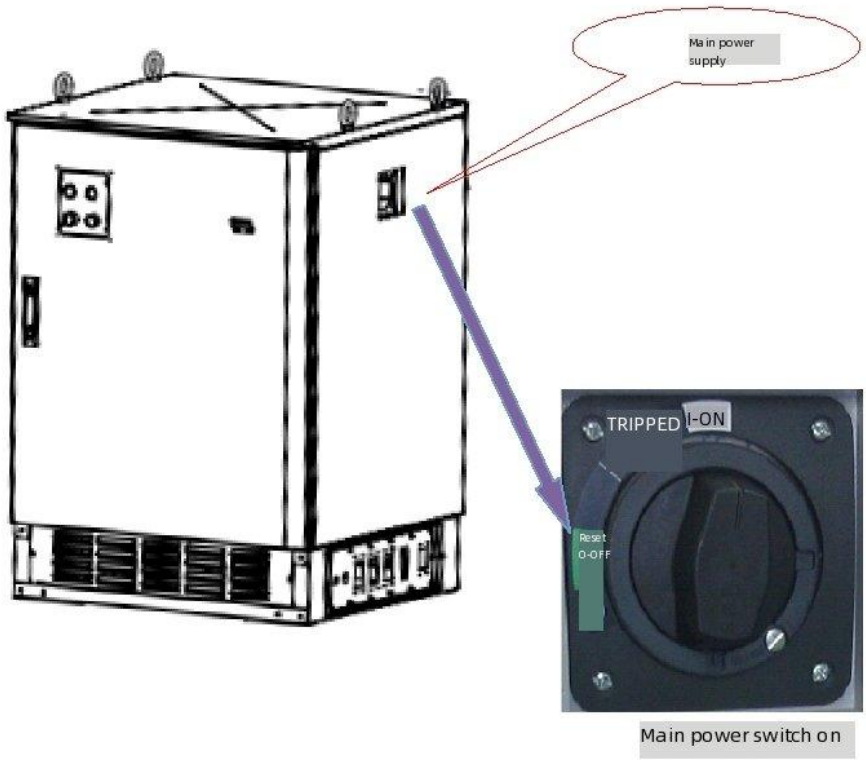


Diagram 12 Main Power Switch

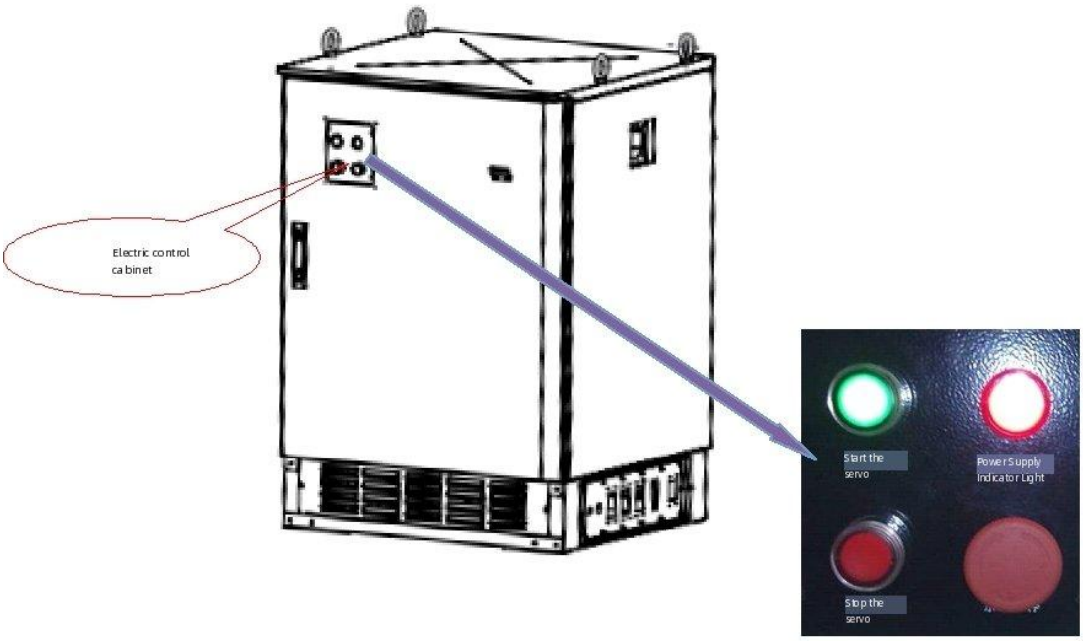


Diagram 13 Electric Control Cabinet Panel

5.1.2 Switch on the Servo Power Supply

The servo power-on procedure is different for the teach mode, the playback mode, and the remote mode.

1.In teach mode:

Press the [Servo Ready] key on the hand-held operation teach pendant, and the [Servo Ready] indicator is flashing; gently hold the [Three-Step Switch] on the back of the hand-held operation teaching pendant, and the [Servo Ready] indicator on the hand-held operation teach pendant is on for a long time, indicating that the servo power is on.

2.In playback and remote modes:

Press the [Servo Ready] key on the handheld operation teach pendant, and the [Servo Ready] indicator light on the handheld operation teaching pendant is on for a long time, 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 for the teach mode, the playback mode, and the remote mode.

1.In teach mode:

Release or firmly hold the [Three-Segment Switch] on the back of the handheld operation teach pendant, and the [Servo Ready] indicator on the handheld operation teach pendant goes out, indicating that the servo power supply is cut off.

2.Playback and remote mode

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

3.Press the emergency stop key on the panel of the electric 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, playback mode or remote mode).

5.2.2 Cut Off the Main Power Supply

Cut off the main power after cutting off the servo power.

Turn the main power switch on the side door of the electric control cabinet to the OFF position to cut off the main power supply, as shown in the figure below:

Diagram 14 Rotary Switch

5.3 Selection of Coordinate System

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

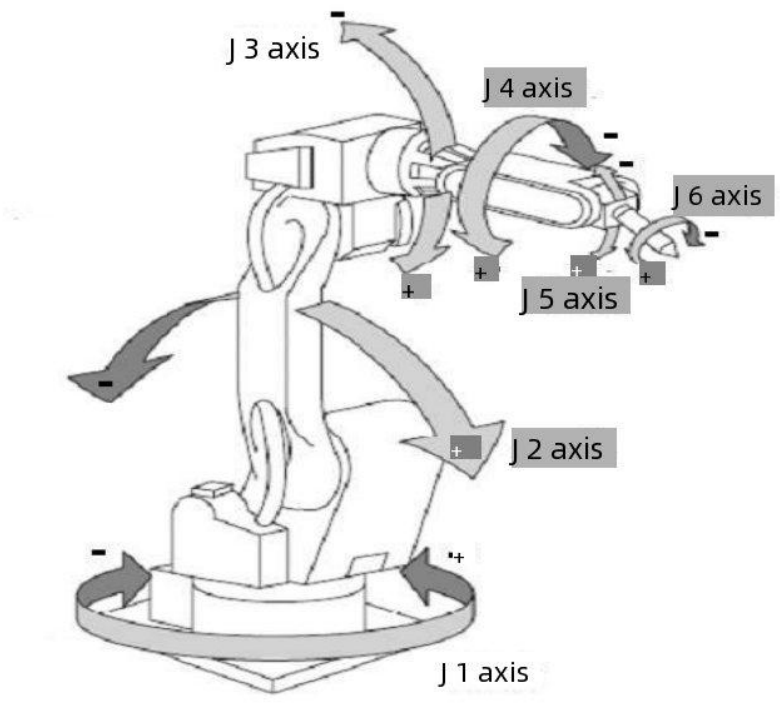
5.4 Manual Speed Adjustment

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 Action

In teach mode, each axis of the robot can be moved to the desired position by pressing the **axis key**, 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.

危险

1. Before operating the robot, please confirm that the emergency stop key can work normally: Press the emergency stop key on the electric control cabinet, and the indicator light of [Servo Ready] goes out, 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 location area.
   - Switch on the power supply of the electric control cabinet.
   - When operating the robot with a hand-held teach pendant.
   - On playback.
   - When remote.

   If the robot collides with any person who enters the range of motion, personal injury will result.
4. Before **teaching the robot**, perform the following inspection steps. If any problems are found, correct them immediately and confirm that all other necessary work has been completed.

   - Check for any abnormal movements of the robot;
   - Check if the insulation and coverings of the external cables are damaged.

After the servo power is turned on (press the [Servo Preparation] button, hold the [Three Section Switch], and the servo indicator light will stay on), press each axis operation button on the handheld operation teaching device to generate the required action for each axis of the robot. The following diagram illustrates the motion of each axis in the joint coordinate system.

Please pay attention to the joint movement speed status before operating the robot, and adjust it to the appropriate speed through the high and low speed buttons.

5.6 Flush

The leveling function can only be used in teaching mode.

The leveling function is to automatically level the rotational component of the rotational motion 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 angle must be within the range of 15 degrees.

When the angle is less than 15 degrees, press the **chain key+【A+】** key on the handheld teaching pendant to select the rotation component of the coordinate system around the X-axis that is level to an integer multiple of 90 degrees; **Chain key+[B+]** key selects the rotation component of the coordinate system around the Y-axis that is aligned to an integer multiple of 90 °; **Chain key+[C+]** key selects the rotation component of the coordinate system around the Z-axis that is aligned to an integer multiple of 90 °.

Finally, select the robot's motion coordinate system through the floating menu to confirm whether it has reached an integer multiple position of 90 °.

5.7 Return Zero

In teaching mode, set the original position coordinates, and after the servo is turned on, press the **chain key and forward key** on the handheld teaching pendant to move to that point.

This point should be recorded as a location that the robot can safely reach from

any position.

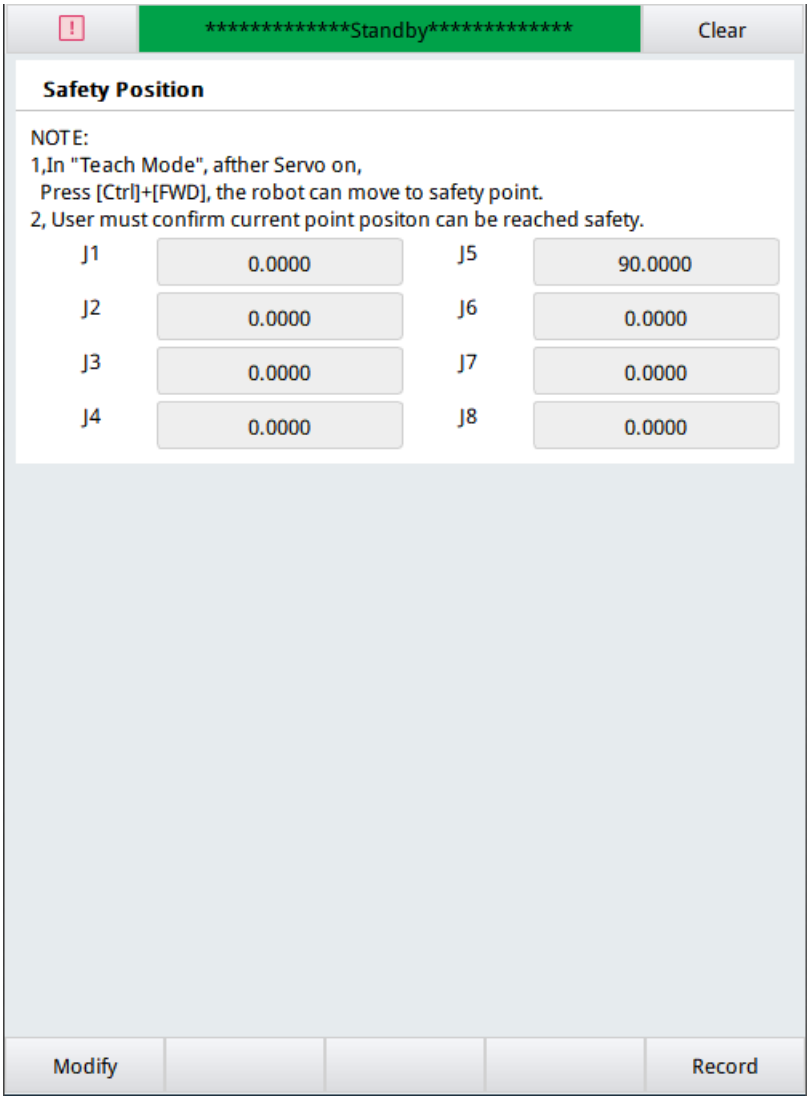


Diagram 15 HOME Point Setting

If the HOME point position is not set, it defaults to the robot's factory zero position.

6 Engineering Management Interface

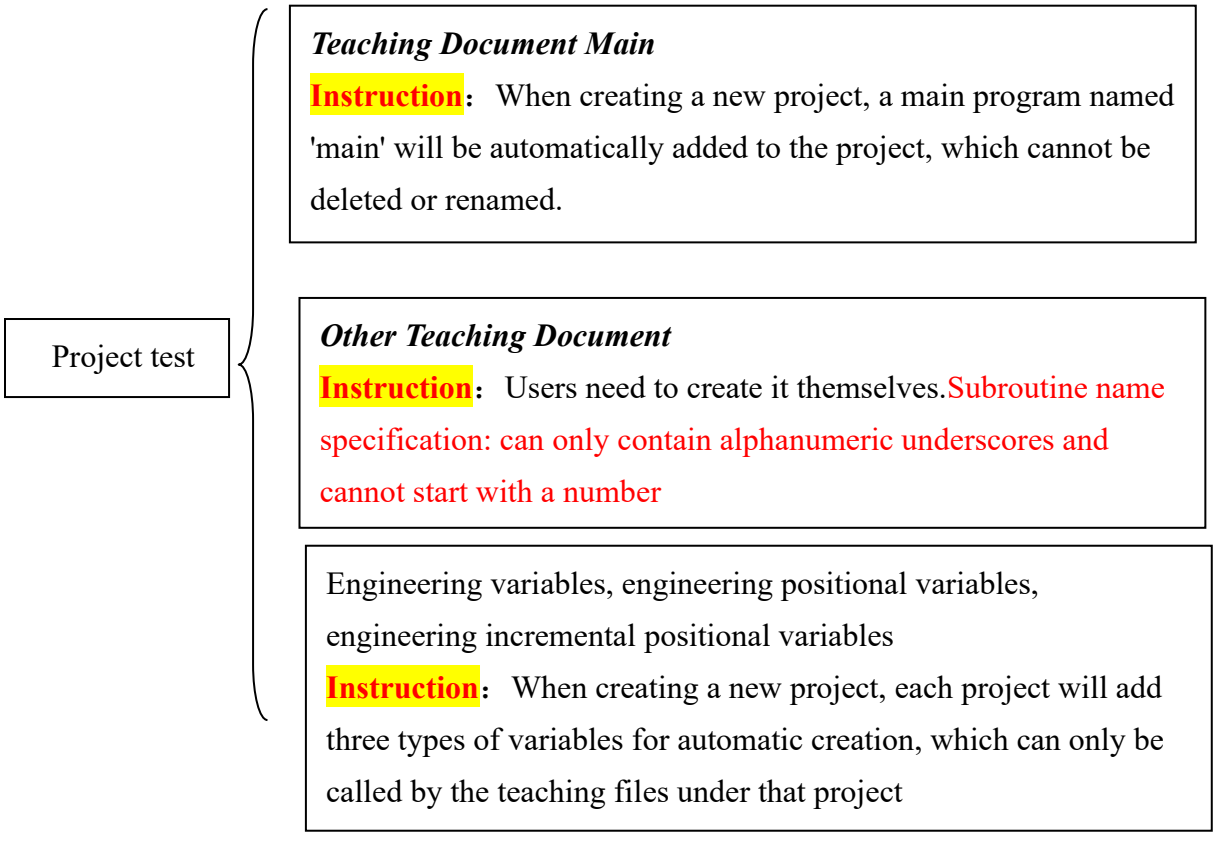
Select  from the main menu or click “Project Management” on the common toolbar to enter the project management interface.

Diagram 16 Project Management Interface

6.1 Description of Robot Engineering Structure

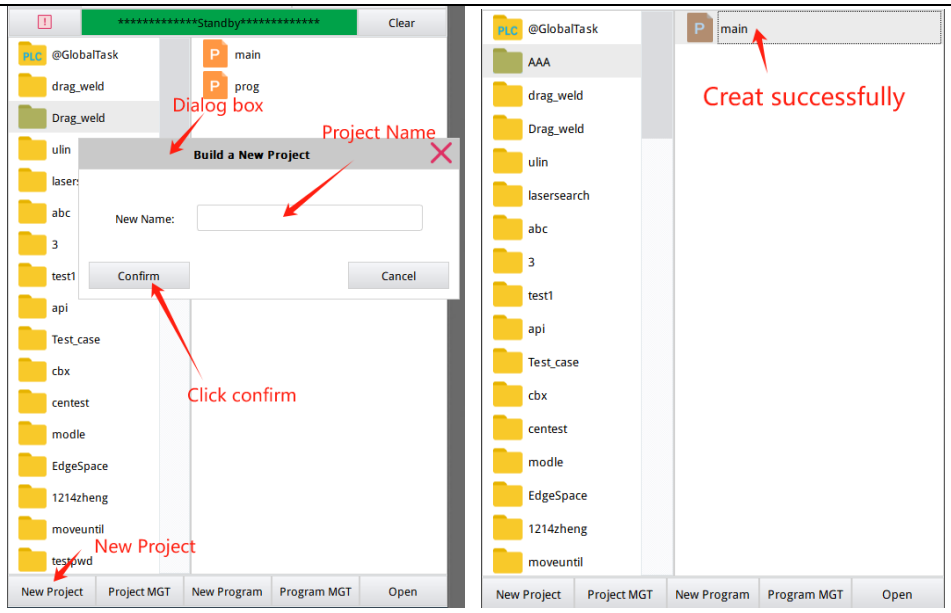
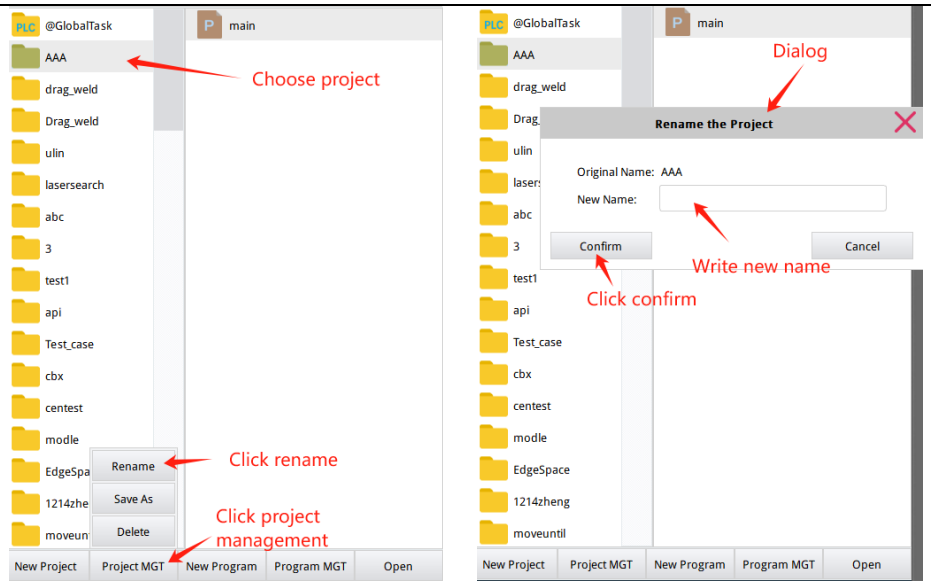
The robot program is based on the project, each project can have multiple teaching files, and each project contains variables that can only be used by the teaching files within the project. Take the project named "test" as an example. The following figure shows the structure of the teaching file main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

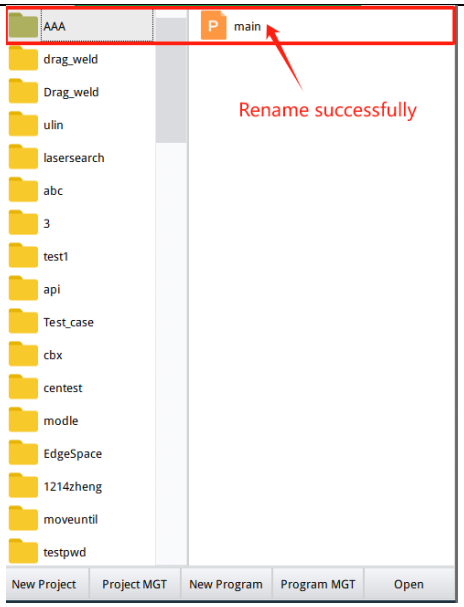
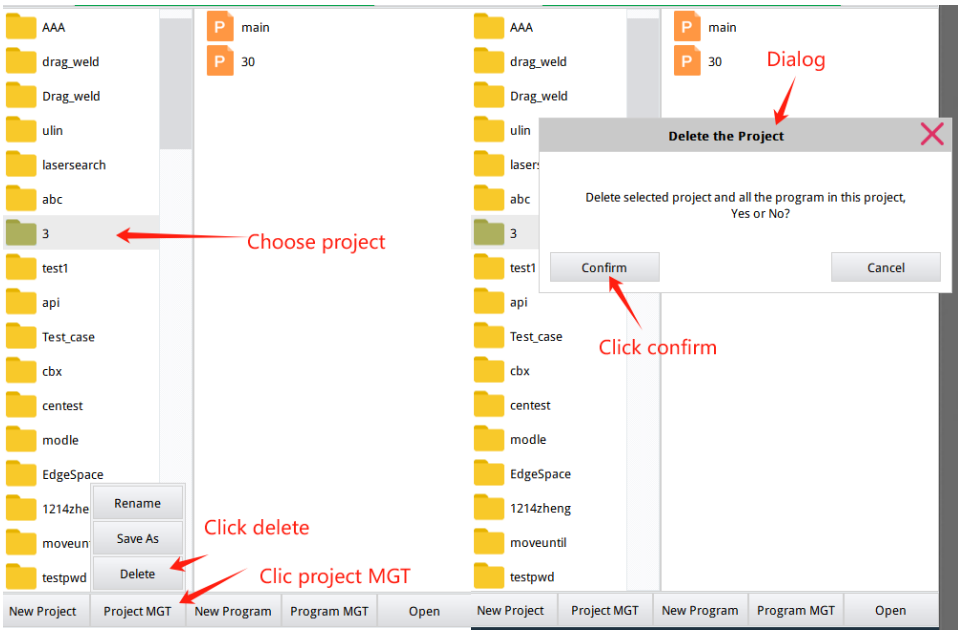
- The main teaching file and other teaching files can be executed;
- The difference between the main teaching file and other teaching files: Other teaching files can be called in the main teaching file; Other teaching files can call each other, but cannot call the teaching file main.

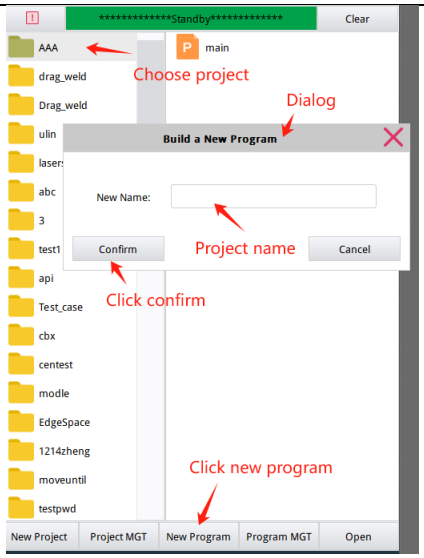
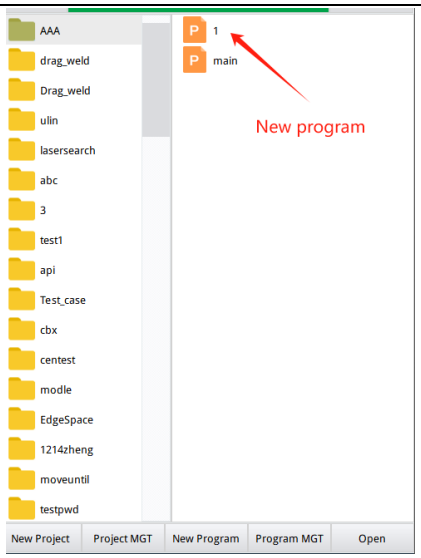


6.2 Engineering Operations

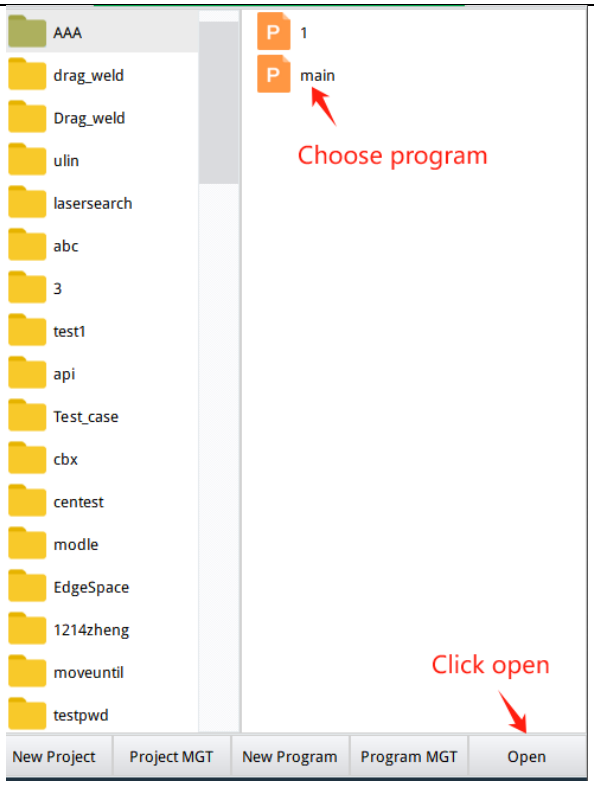
以下所讲的是能够在机器人示教模式下进行的程序编辑操作。

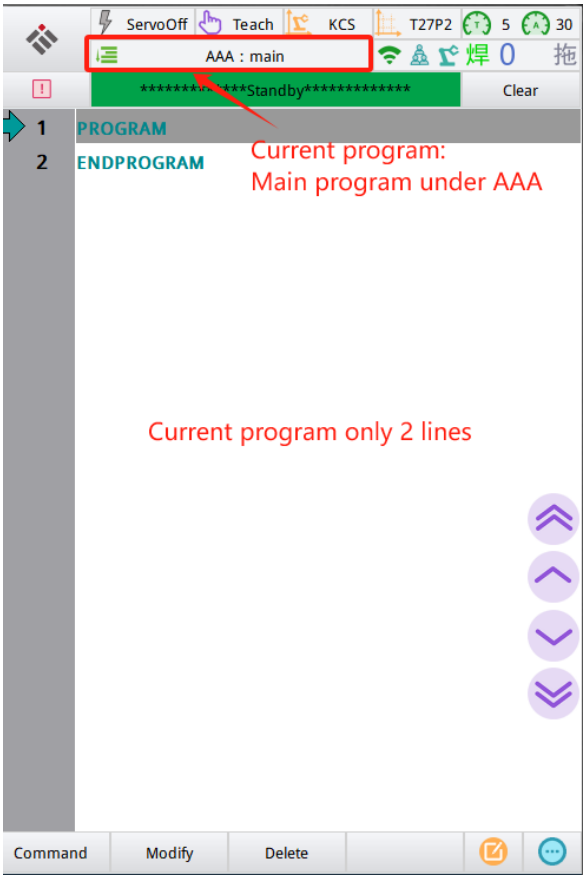
Item	Description
1. New Project	
2.Project Rename	

<table>
<tr><td></td><td> 
</td></tr>
<tr><td>3.Save Project As</td><td>Save As is to copy the entire project again. Please refer to the steps for renaming the project</td></tr>
<tr><td>4.Project Deletion</td><td>Deleting a project will delete all teaching and variable files under the project
 
</td></tr>
</table>

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.Delete Program	To delete a program, first select the program to be operated, then select [Program Management], click [Delete] other operation steps to view **program delete**

Open the Program





Click Open to enter the **program editing** interface

7 Program Editing Interface

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

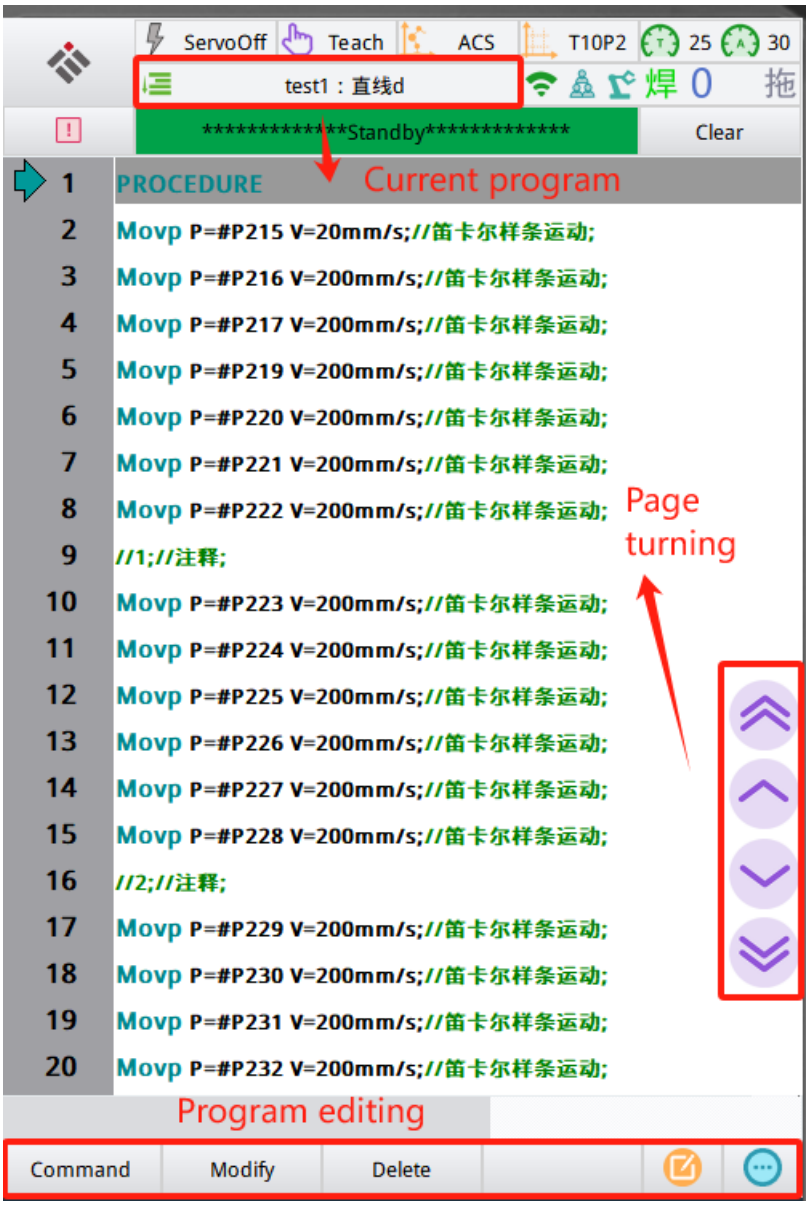


Diagram 17 Program Editing Interface

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.

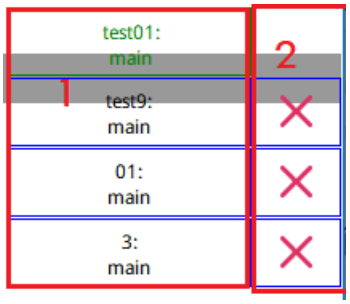


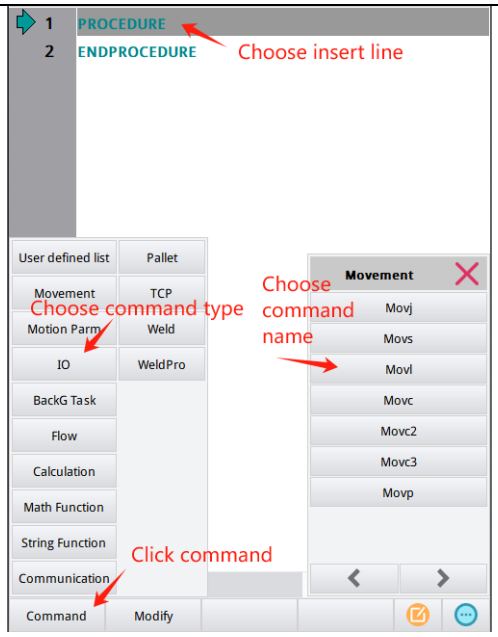
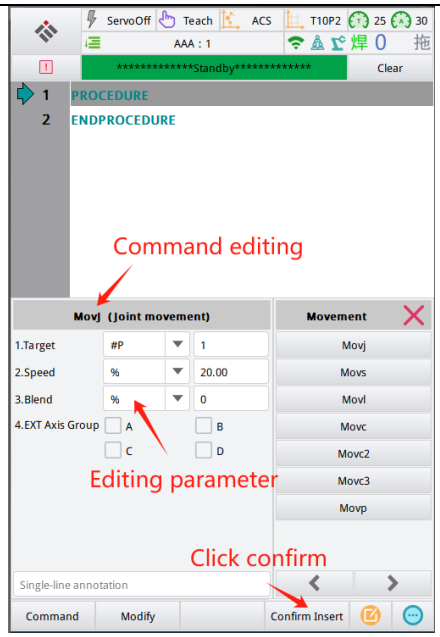
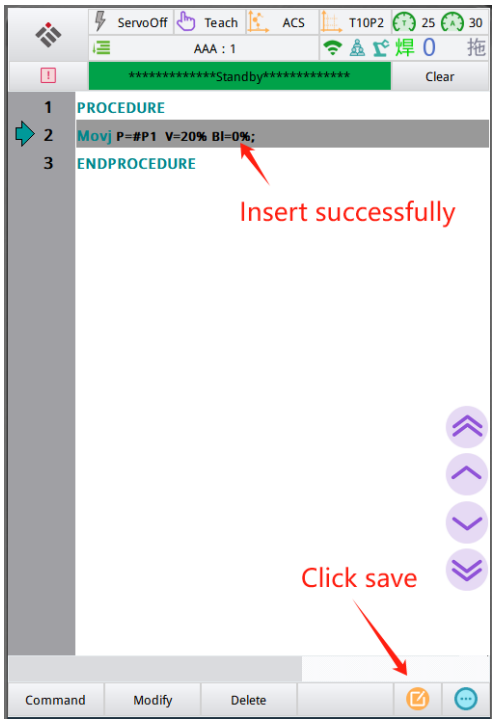
Diagram 18 Historical Program Interface

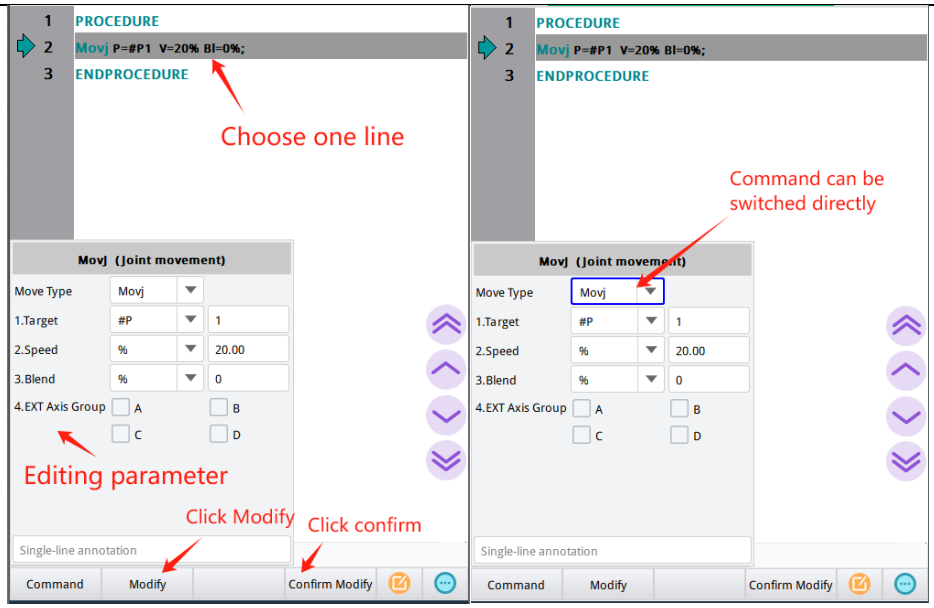
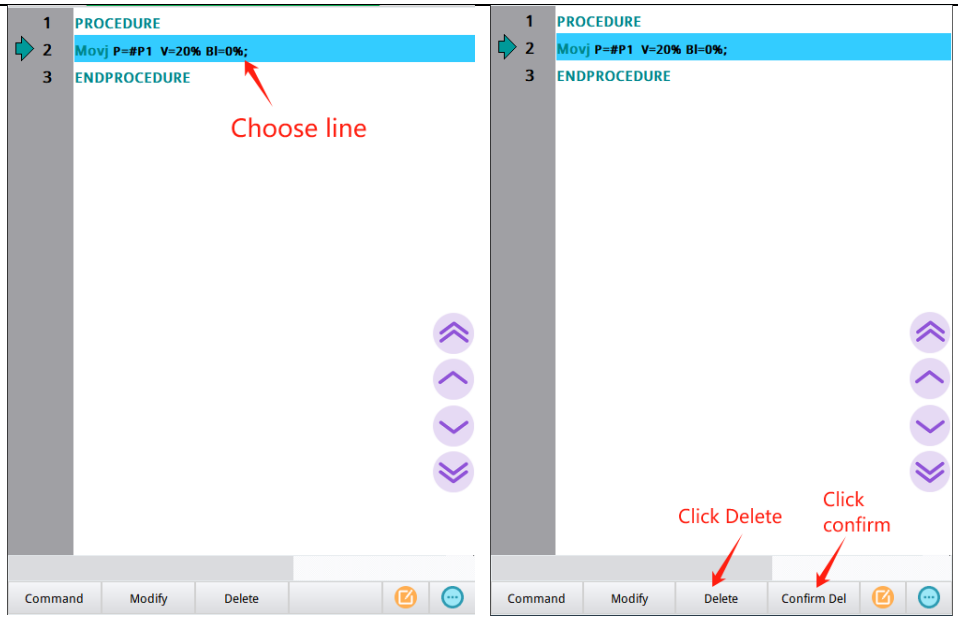
Item	Description
1.Quickly switch the history program	Click the corresponding row in the left column to quickly switch to the corresponding program. The green line is the current open program.
2.Delete historical records	Click corresponding ✕, to the two columns on the right to delete the program history 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

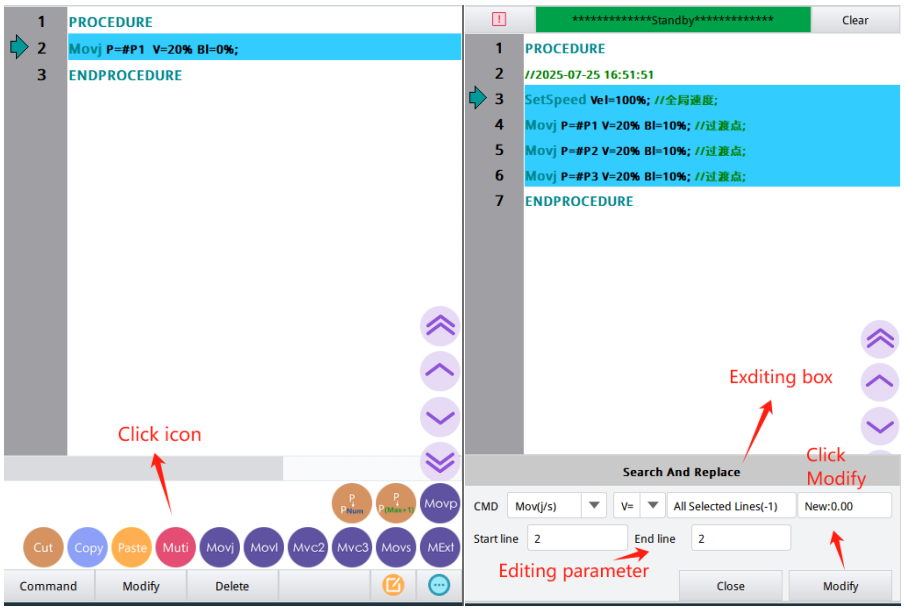
Function	Description
1. Teach instruction insertion (inserts a new instruction to the next line of the selected line)	  

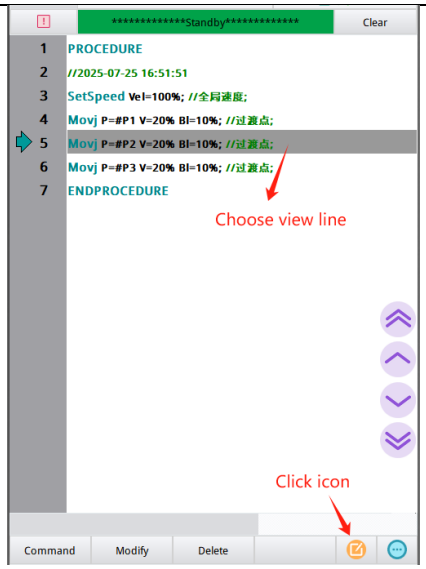
2.Instruction modification	
3.Delete instruction (when deleting a single line, click the line to select it)	

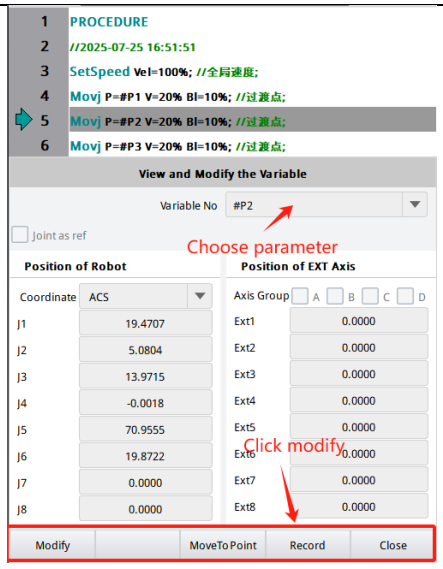
7.5 Extension Bar

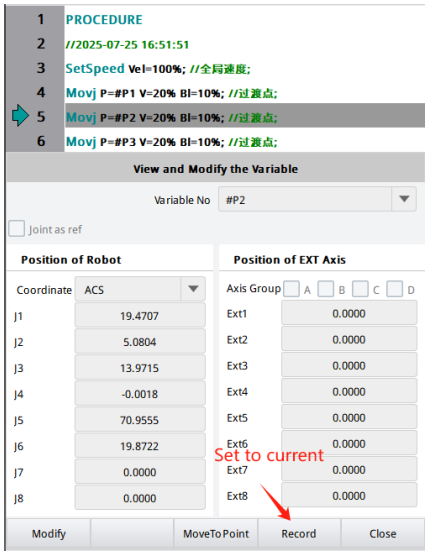
Click  , to open the extended operation bar. The functions of each button in the extended bar are described as follows.

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 
	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.







Quick Insert MovI:

1. In the upper servo 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 one-day MovI 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, i.e., the middle point of the arc
	Quick insertion Mvc3: The operation and steps are the same as above, and a Movc3 motion command, i.e. arc target point, will be inserted
	Quickly insert Movs: The operation and steps are the same as above, and a Movs motion command will be inserted
	Quickly insert Movp: The operation and steps are the same as above, and a Movp motion command will be inserted
	Quickly insert MoveExt: The operation and steps are the same as above, and a MovceExt motion command will be inserted
	Quickly modify the current location point: In the servo state, clicking this button will re record the position point of the existing 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
	Quickly modify the current location point: In the servo state, clicking this button will re record the position point of the existing 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

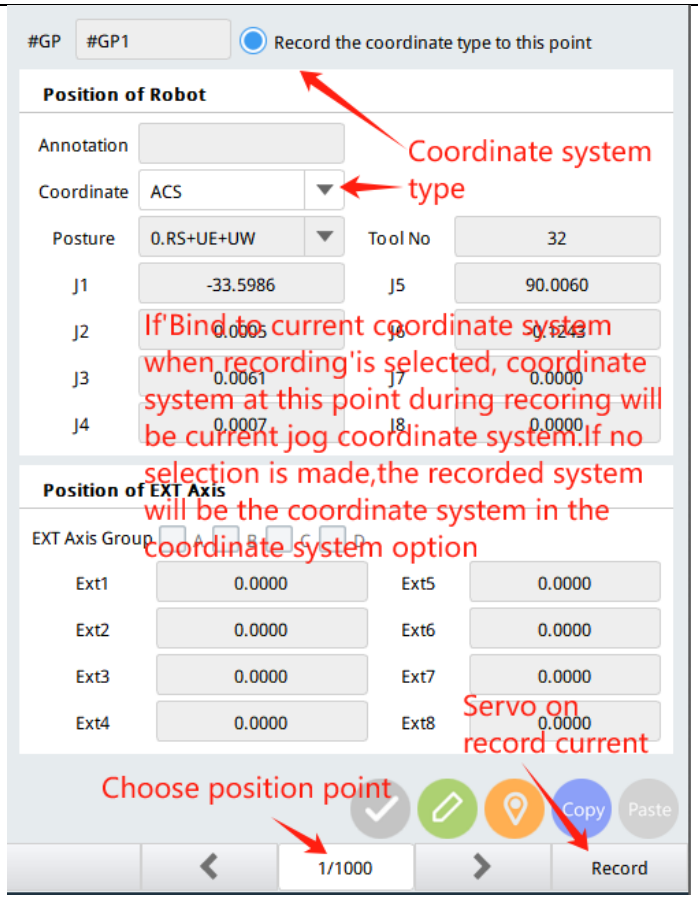
8 Variables

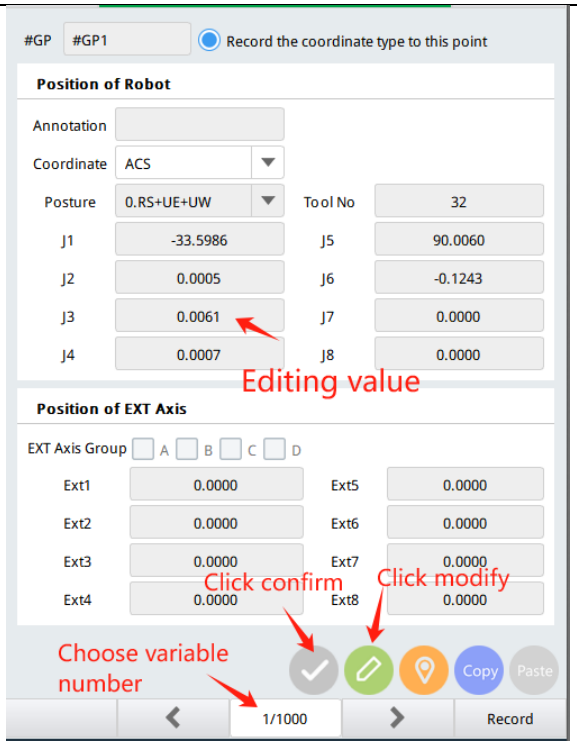
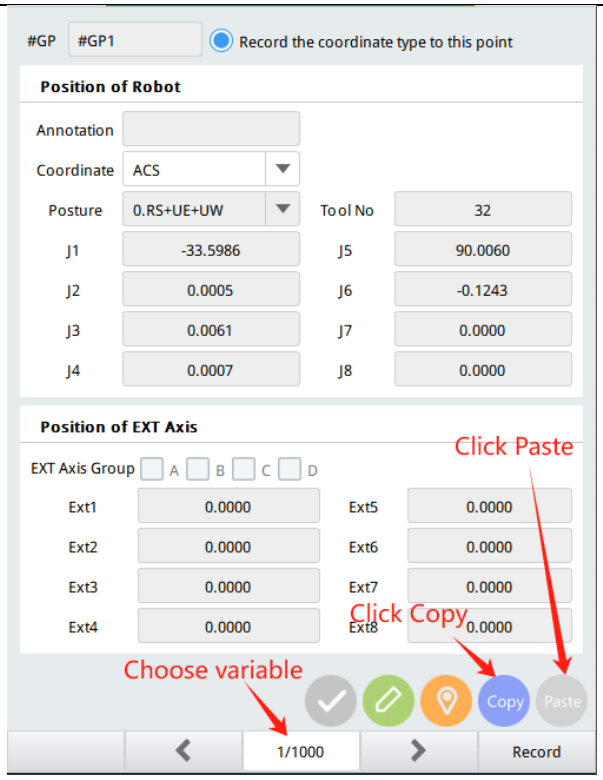
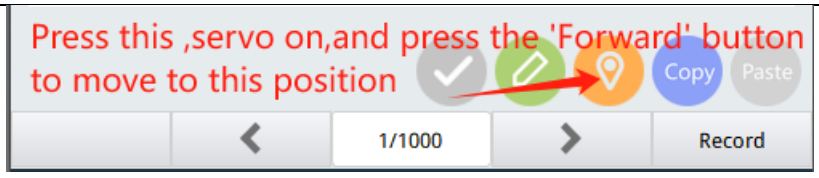
8.1 Global Variable

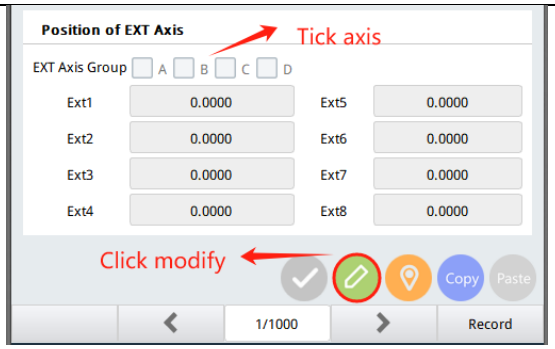
Global variables, including global positional variables (#GP), global incremental positional variables (#GIncP), global numeric variables (#GVar), and global string variables (#GStr), are accessible to all teach files in the project.

8.1.1 Global Positional Variables

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

Item	Operating Instruction
1.Location point 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):

G_Numerical G_String

No	Data	Note
#GVar 1	1.0000	BaseHorValueAvg
#GVar 2	57787184.0000	BaseVerValueAvg
#GVar 3	1.0000	weld_point_count
#GVar 4	10.0000	
#GVar 5	-24.0000	
#GVar 6	19121.0000	
#GVar 7	0.0000	
#GVar 8	0.0000	
#GVar 9	0.0000	
#GVar 10	4387.9974	
#GVar 11	0.0000	
#GVar 12	0.0000	
#GVar 13	2.0000	
#GVar 14	1.0000	
#GVar 15	0.0000	

1 / 20

Diagram 19 Global string variable

Click  to complete 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):

G_Numerical G_String

No	Data	Note
#GVar 1	1.0000	BaseHorValueAvg
#GVar 2	57787184.0000	BaseVerValueAvg
#GVar 3	1.0000	weld_point_count
#GVar 4	10.0000	
#GVar 5	-24.0000	
#GVar 6	19121.0000	
#GVar 7	0.0000	
#GVar 8	0.0000	
#GVar 9	0.0000	
#GVar 10	4387.9974	
#GVar 11	0.0000	
#GVar 12	0.0000	
#GVar 13	2.0000	
#GVar 14	1.0000	
#GVar 15	0.0000	

1 / 20

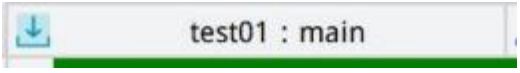
Diagram 20 Global Numeric Variables

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).

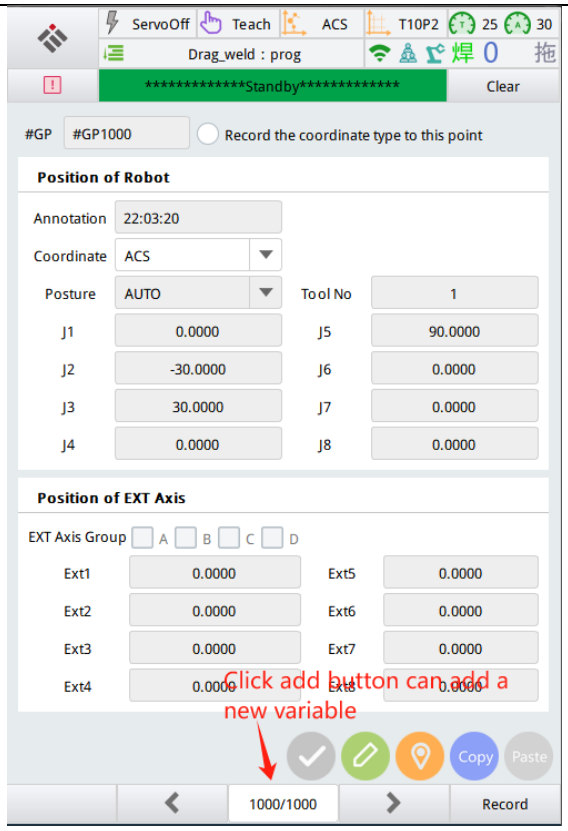
Before viewing project variables, you must first open a project. The currently opened project can be obtained in the status bar.



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 Instruction
1. New variable	
2.Other operations	Reference global position variable

8.2.2 Engineering Incremental Positional Variable

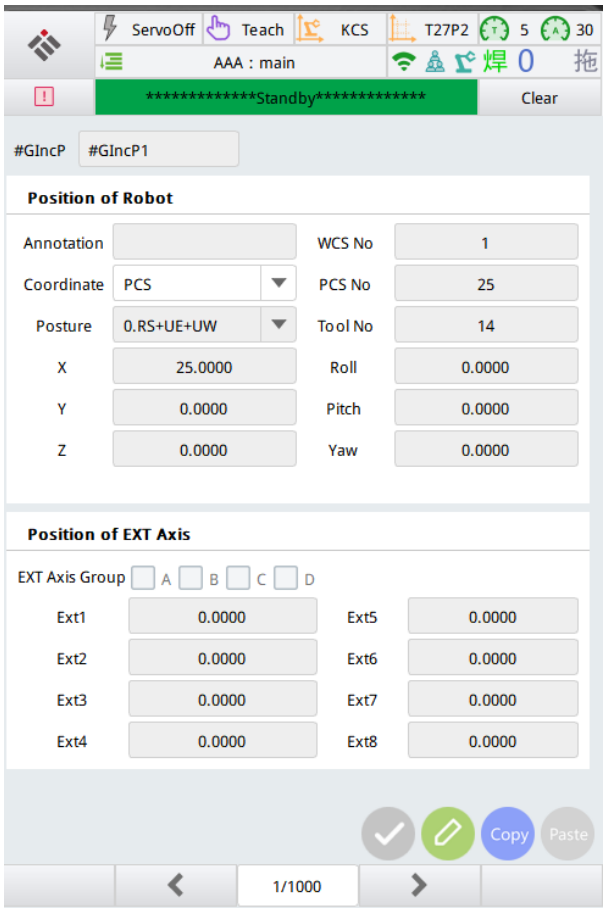


Diagram 21　Engineering incremental position

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 , 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 , and other operations are the same.

8.3 Robot Posture

The robot pose is used in position variables and incremental position variables.

For the 6-axis robot pose model, the following definitions are used:

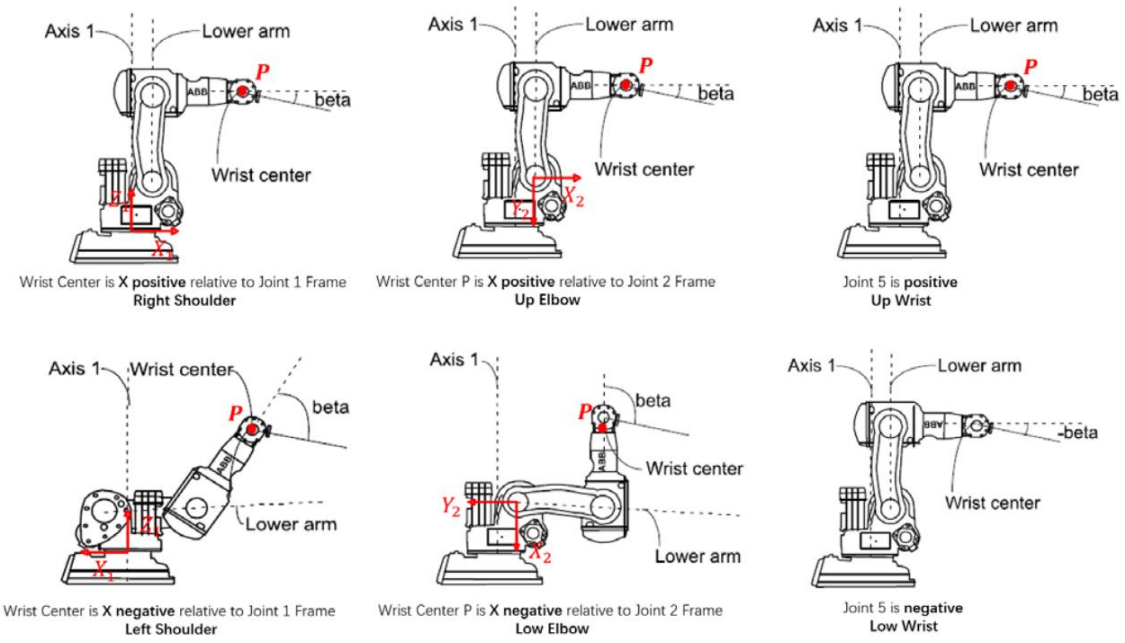


Diagram 22 6-Axis Robot Attitude Definition

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.

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 tool 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 workpiece 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. You 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.

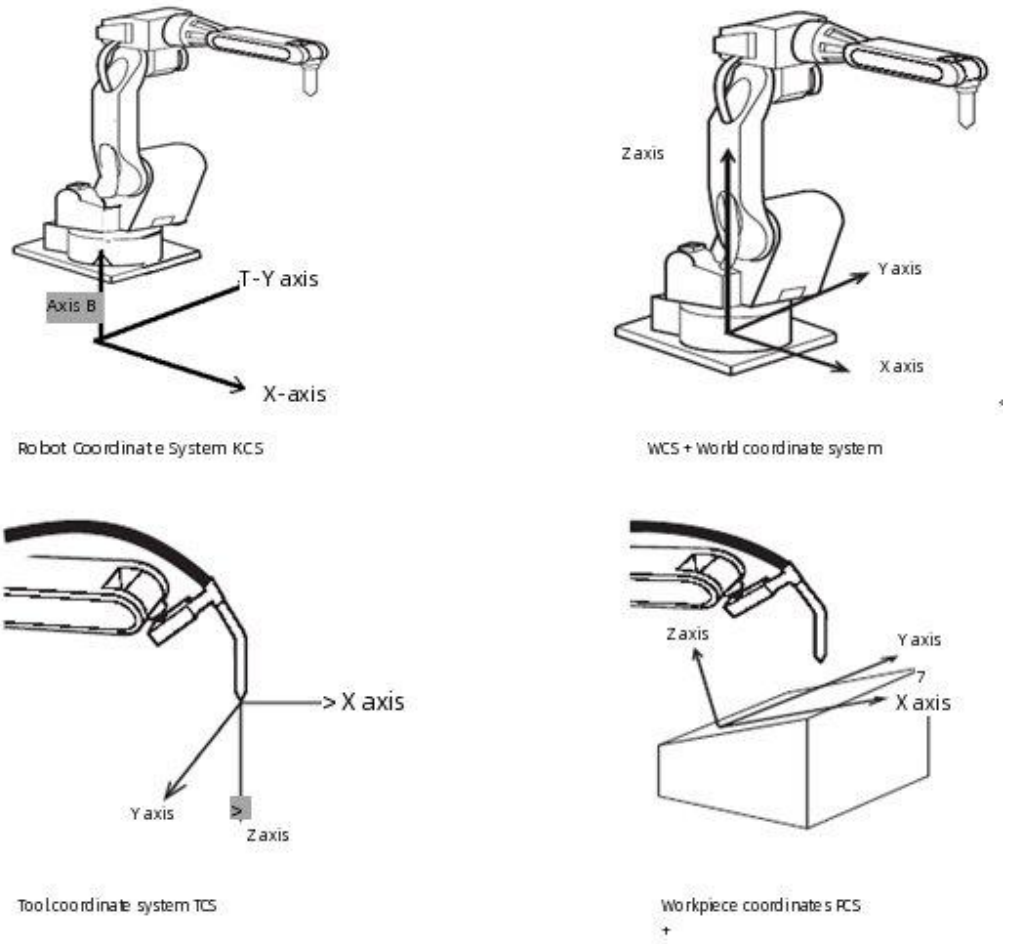


Diagram 23 Schematic diagram of four coordinate systems

9.2 Coordinate System Management Interface

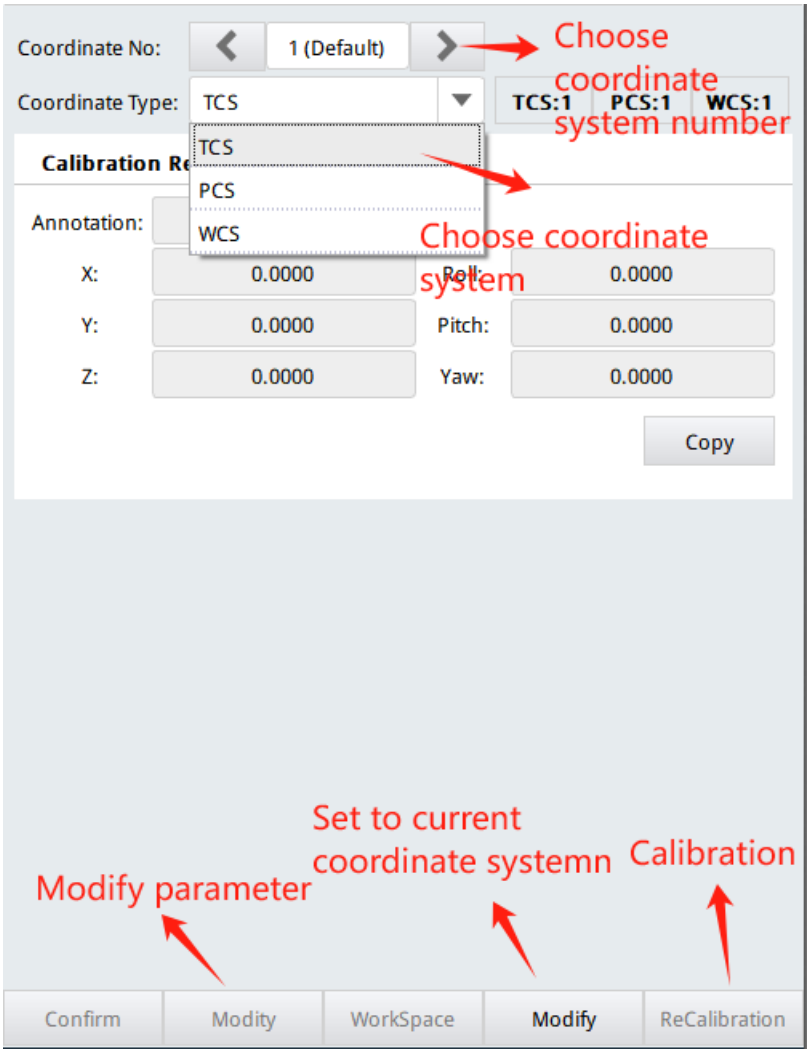


Diagram 24 Coordinate System Interface

9.3 Axis Coordinate System (ACS)

In the teaching mode, when the coordinate system is set to the joint coordinate system ACS, the J1, J2, J3, J4, J5 and J6 axes of the robot move respectively. Please refer to the following table for the action of each axis when pressing the axis operation key:

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		Up-down movement of upper arm
Wrist axis	J4 axis		Wrist rotation with upper arm strap
	J5 axis		Wrist up and down movement
	J6 axis		Wrist rotation

9.4 Kinematic Coordinate System (KCS)

In the teaching 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 and pressed, refer to the following table for the actions 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	Around X axis		Rotational motion about the X-axis of the KCS coordinate
	Around Y axis		Y-axis rotational motion around KCS coordinates
	绕 Z 轴		Y-axis rotational motion 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.

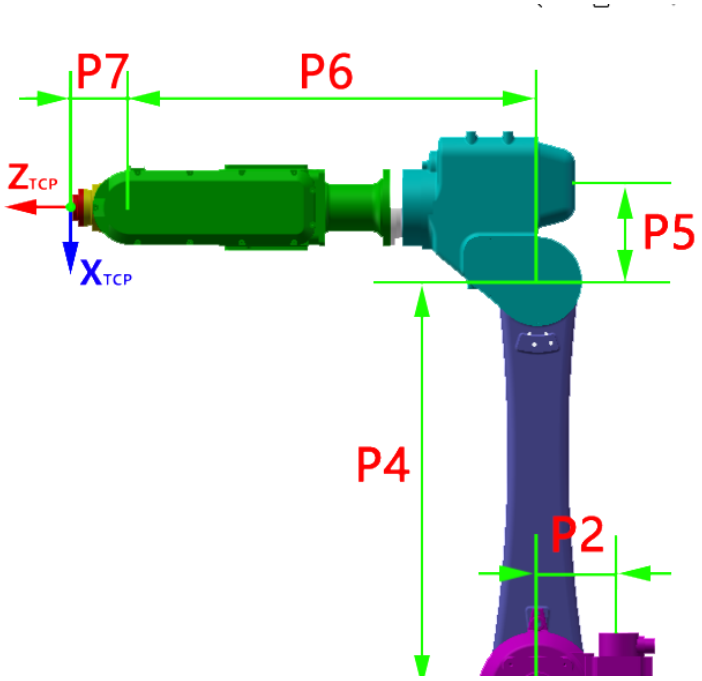


Diagram 25 Default Tool Coordinate System

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	Around 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 Offset 6-point method: calibration position offset

When using the {offset 6-point method} calibration, approach the same

reference point from six different directions using the tip (center) point (i.e. TCP point) of the test tool. The reference point can be selected arbitrarily, but it must be the same fixed and unchanging reference point. The robot controller calculates TCP from six different flange positions. The six flange positions of the robot TCP point moving to the reference point must be dispersed by sufficient distance to make the calculated TCP points as accurate as possible. The schematic diagram of the bias 6-point method is shown below:

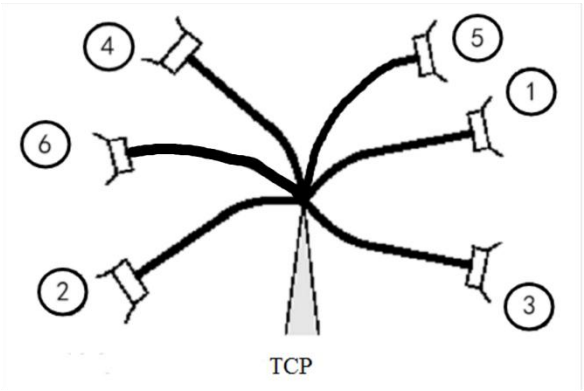


Diagram 26 Diagram of four-point method

{offset 6-Point Method} The steps 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 teaching mode.。

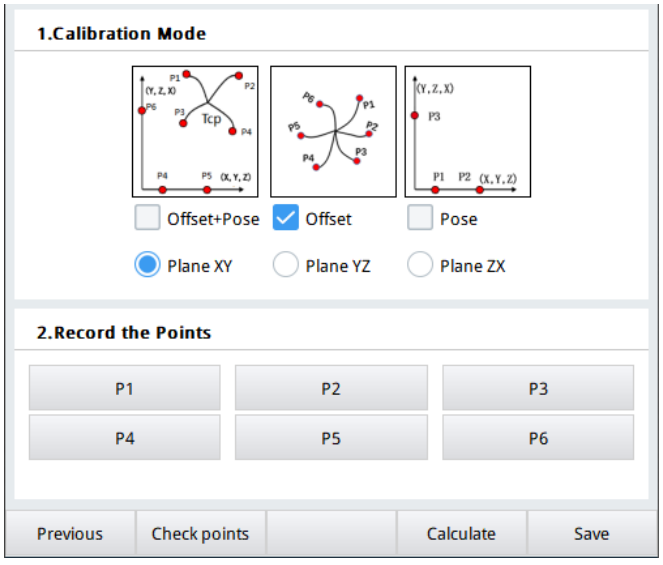


Diagram 27 Offset six point method records six position points

Step 2: Move the tip point TCP of the test tool from the first direction towards a

fixed reference point. Click the {Record P1} button with the servo power on to record the first position point. After completing the P1 point recording, the background color of the {Record P1} button will change to green.

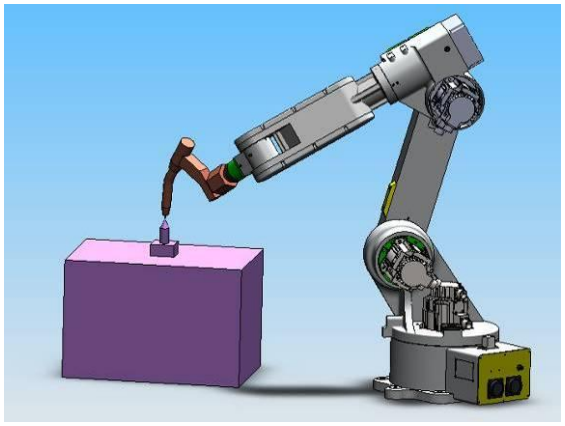


Diagram 28 Record the P1 position point

Step 3: Move the tip point TCP of the test tool from the second direction towards the same fixed reference point. Click the {Record P2} button with the servo power on to record the second position point. After completing the P2 point recording, the background color of the {Record P2} button will change to green.

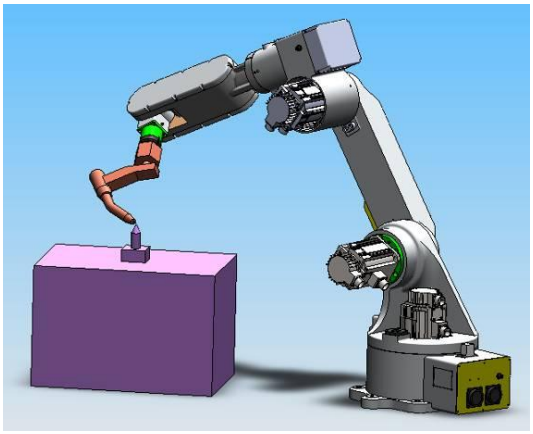


Diagram 29 Record the P2 position point

Step 4: Move the tip point TCP of the test tool from the third direction towards the same fixed reference point. Click the {Record P3} button with the servo power on to record the third position point. After completing the P3 point recording, the background color of the {Record P3} button will change to green.

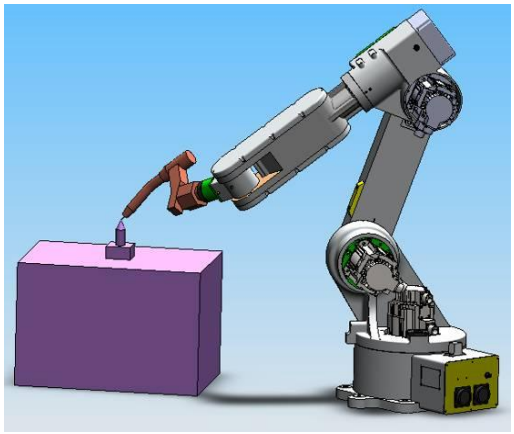


Diagram 30 Record the P3 position point

Step 5: Move the tip point TCP of the test tool from the fourth direction towards the same fixed reference point. Click the {Record P4} button with the servo power on to record the fourth position point. After the P4 point recording is completed, the background color of the {Record P4} button will change to green.

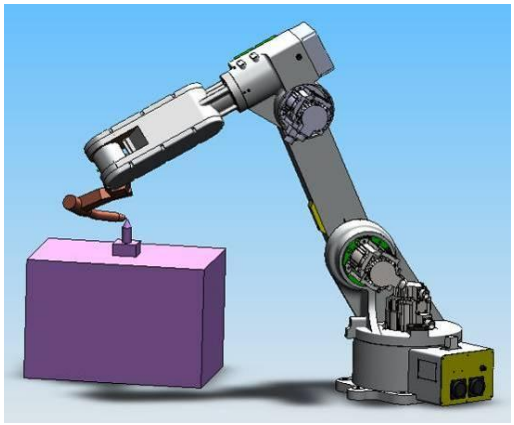


Diagram 31 Record the P4 position point

Step 6: Continue recording the fifth and sixth position points according to the above steps. After recording the P5 and P6 points, the background color of the {Record P5, P6} button will change to green.

Step 7: After completing the recording of the six required position points for the {offset 6-point method}, click the {calculate} button to automatically calculate the TCP position point data and display the calculation results.

Note that if two or more identical position points are recorded in the offset six point method, the calculation cannot be successful and the program will report an error.

Step 8: Click the {Save Results} button to save the recorded teaching position point coordinates and calculated coordinate system data, and return to the coordinate system management main interface.

Step 9: Click the {Set as Current} button to use the calculated TCP tool as the flange end tool. So far, all steps from calculating the tool coordinate system to switching the newly calculated tool to the currently used tool have been completed. The tool coordinate system was calculated and successfully switched, and now various robot movements can be performed under the new tool.

Note that using the offset six point method can only determine the offset value of the tool tip (center) point TCP relative to the installation surface of the robot end flange. When users need to teach to determine the tool posture component, they need to use the additional posture three-point method or directly use the offset+posture six point method.

9.5.2.3 Attitude 3-point method: calibrating attitude offset

The steps for teaching and calculating the attitude component of the tool coordinate system TCS using the attitude 3-point method are as follows:

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

In the three-point posture method working mode, three position points need to be recorded, namely P1 point, P2 point, and P3 point. In addition, users also need to select the plane where the teaching point is located, as shown below, by choosing the XY plane. The P1 and P2 points of the teaching are used to determine the direction of the X-axis of the tool coordinate system, and the P3 point is on the positive side of the Y-axis of the XY plane of the tool coordinate system. Since the three-point pose method only determines the pose component of the tool coordinate system TCP, the XY plane used for teaching only needs to be parallel to the XY plane of the actual tool coordinate system TCS, and does not necessarily

need to be the XY plane of TCS. Point P1 is a point on the X-axis of the selected XY teaching plane coordinate system, and does not necessarily need to be the tool tip (center) point TCP (the origin of the tool coordinate system). The same applies to points P2 and P3.

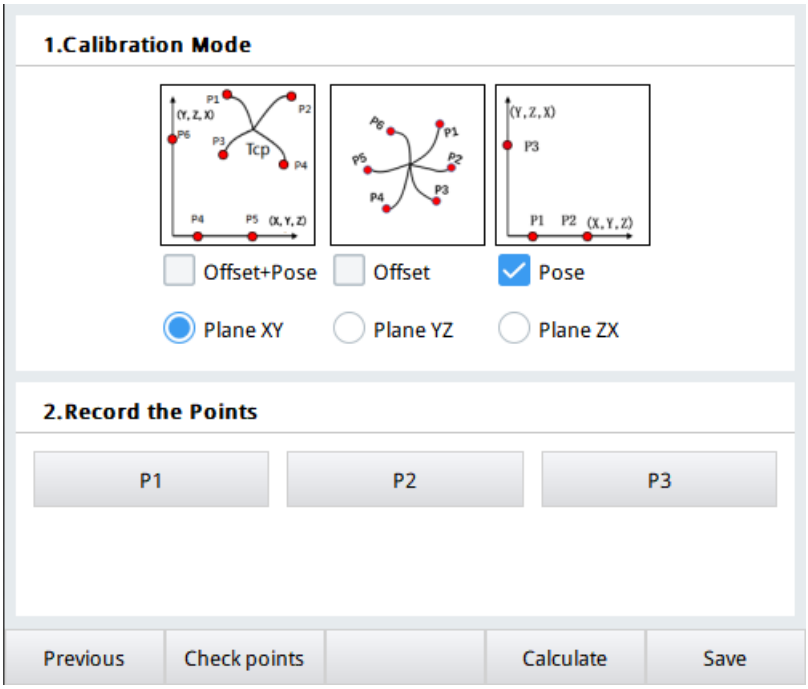


Diagram 32 Choose the three-point posture method working mode

Step 2: As shown in the figure below, first record the first point in the X-axis direction on the tool coordinate system, which is point P1.

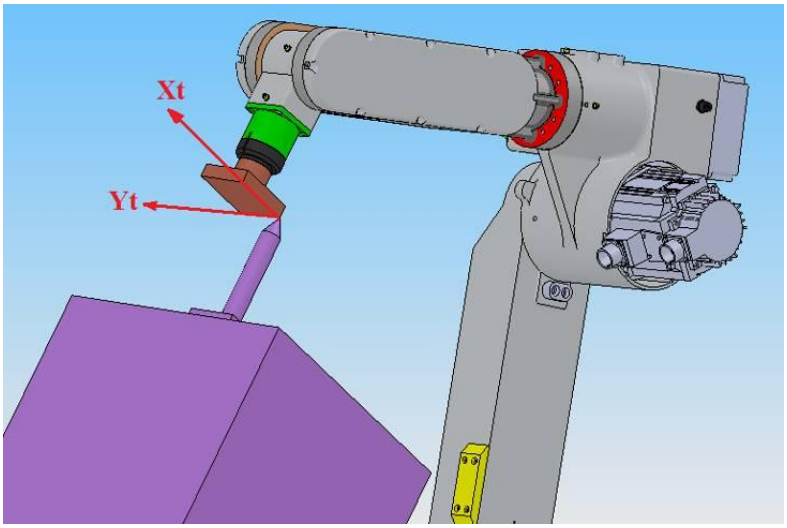


Diagram 33 Record point P1 (the first point P1 in the positive X-axis direction)

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

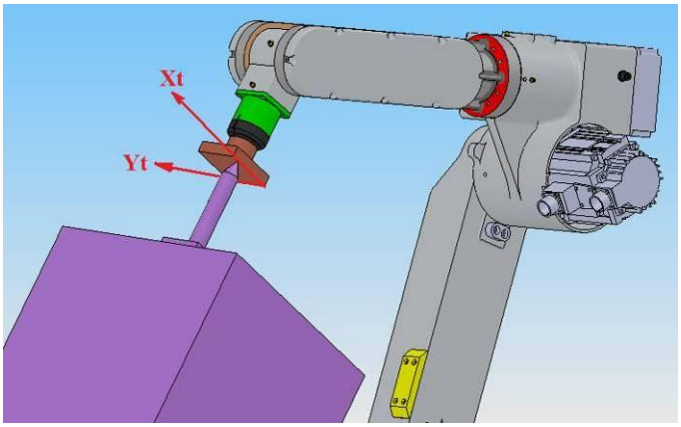


Diagram 34 Record point P2 (the second point P2 in the positive X-axis direction)

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

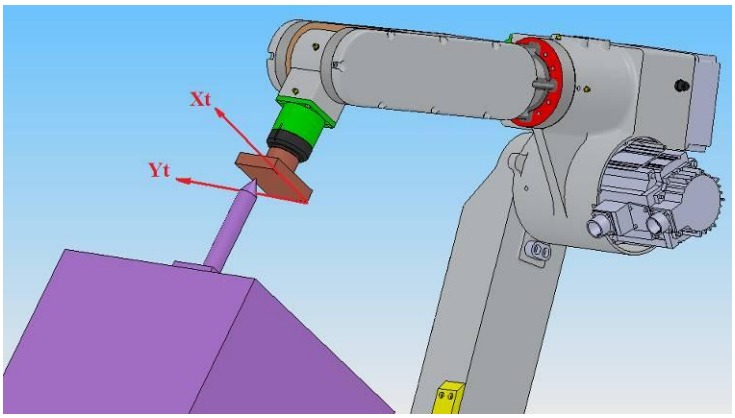


Diagram 35 Record point P3 (point in the positive Y-axis direction on the XY plane)

Step 5: Click the {Calculate} button, and the program generates tool coordinate system pose component data based on the recorded P1, P2, and P3 points. Click {Save Results} to update the pose component of the tool coordinate system TCS with the selected coordinate system number.

It should be noted that when using the three-point pose method (the three position points that need to be recorded) to determine the tool pose, these three position points can only be taught using Cartesian space movement (i.e. only XYZ movement movements under KCS, WCS, PCS, TCS can be used for teaching, and XYZ rotation movements or single joint rotation movements under ACS cannot be used for teaching), and rotation movements with any pose cannot be used for teaching. Otherwise, the pose components of the tool coordinate system cannot be

calculated and an error warning will be given.

9.5.2.4 Offset+Attitude 6-point Method: Calibrate the offset of position and attitude

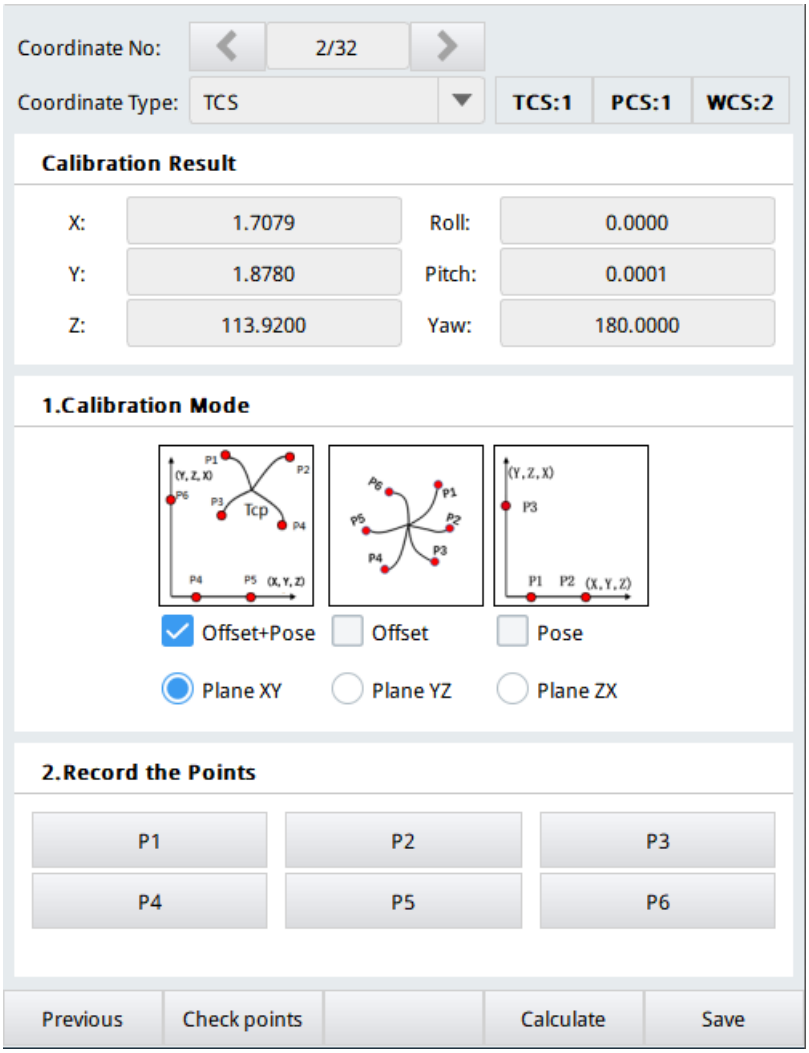


Diagram 36 Tool coordinate system calibration interface

There are three methods for calibrating the tool coordinate system: {offset 6-point method}, {attitude 3-point method}, and {offset+attitude 6-point method}. The offset+attitude 6-point method can comprehensively teach the position offset and attitude vector of the end tool TCP (TOOL CENTER POINT) of a six axis articulated robot; The offset 6-point method can only calculate the TCP position offset value at the end of the six axis robot tool, but cannot calculate the tool's attitude vector; The attitude 3-point method can only calculate the TCP attitude

vector at the end of the six axis robot tool, and cannot calculate the position offset value of the TCP at the end of the tool. The {offset+attitude 6-point method} is actually a combination of {4-point method} and {3-point method}. The first four position points recorded in the {offset+attitude 6-point method} are calculated using the {4-point method} to determine the position offset of the TCP at the end of the tool, while the last three position points are calculated using the {3-point method} to determine the TCP attitude vector of the tool.

9.6 Piece Coordinate System (PCS)

9.6.1 Piece Coordinate System Axis Action

In the teaching mode, when the coordinate system is set to the workpiece 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

Axis of rotation	Around 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: 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.

Mode:

Three-point method mode 1: 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. The same is true for the YZ plane and the ZX plane.

Three-point method mode 2: 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 perpendicular line to the line P1-P2 through point P3, and the origin of the coordinate system is at the position of the foot. 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.

Origin offset: the coordinate system taught by the user using the above two methods is offset to the 00 position point in the teaching record.

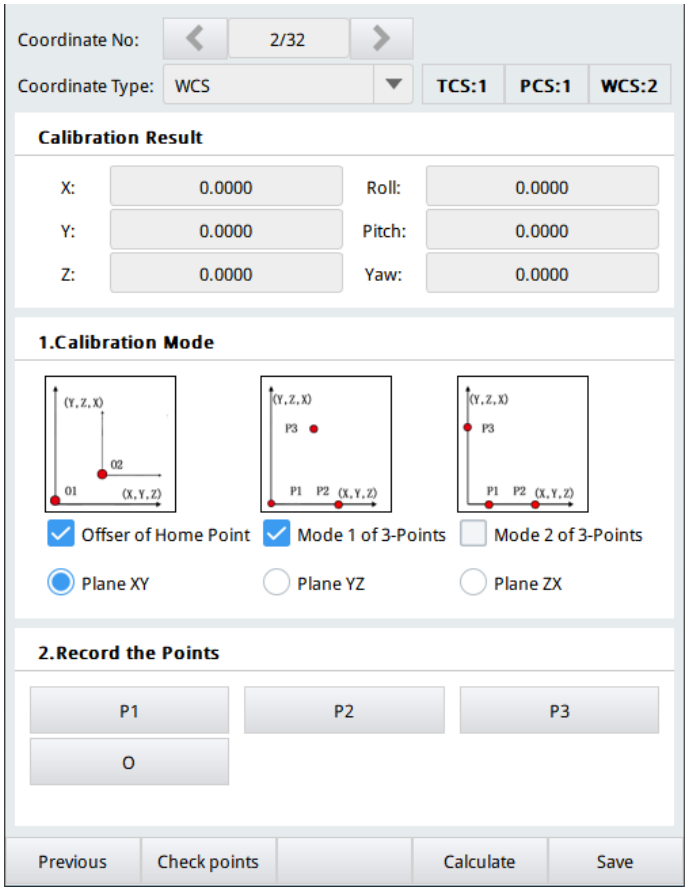


Diagram 37 Calibration interface of piece coordinate system

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 at 2.

Step 2: Select Calibration Method

Select the calibration method {Three-point Method Mode 1}, {Three-point Method Mode 2}, 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 00. 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 attitude of the coordinate system remains unchanged.

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

Step 3: Location Point Recording

Move to the position point, click the corresponding button, and the background 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.

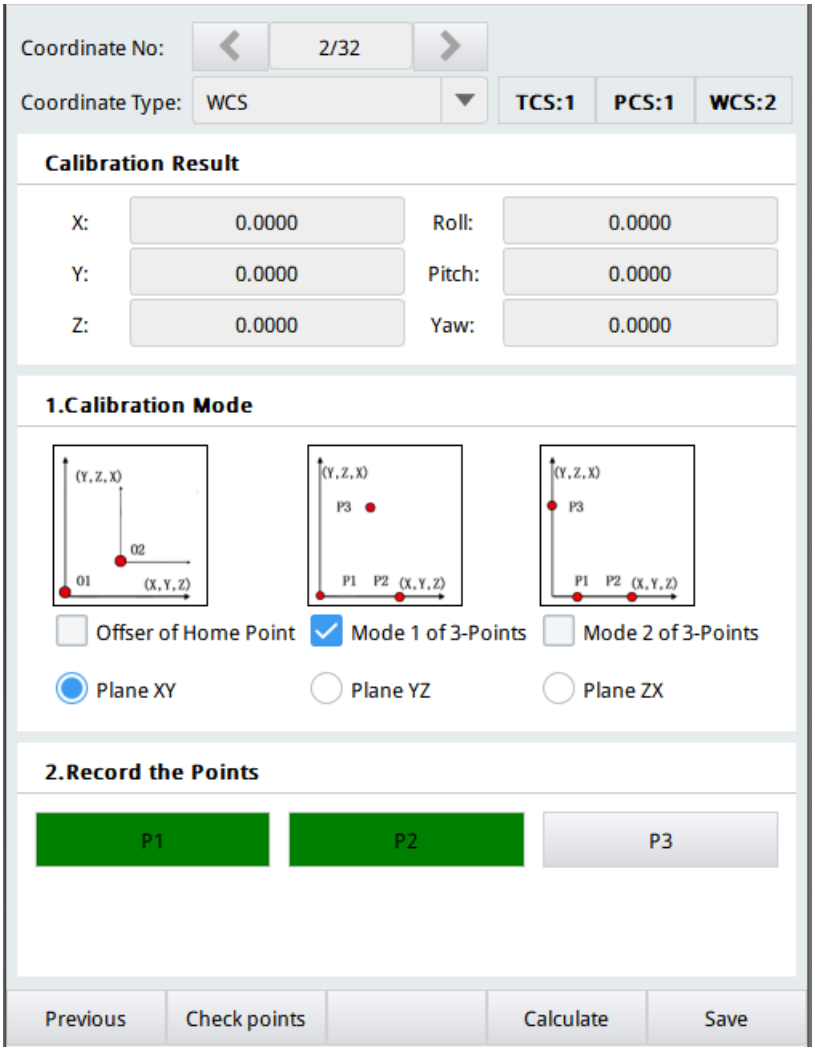


Diagram 38 Piece Coordinate System Calibration Interface 2

Step 4: Calculate and Save Results

Click {Calculate}, and click {Save Result} if the result is displayed successfully; if it fails, an alarm will be displayed in the status bar, and the calibration will be modified or recalibrated according to the alarm prompt.

In addition, on the position point recording interface, the user clicks the {Teaching Point} 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 {move to point}, 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.

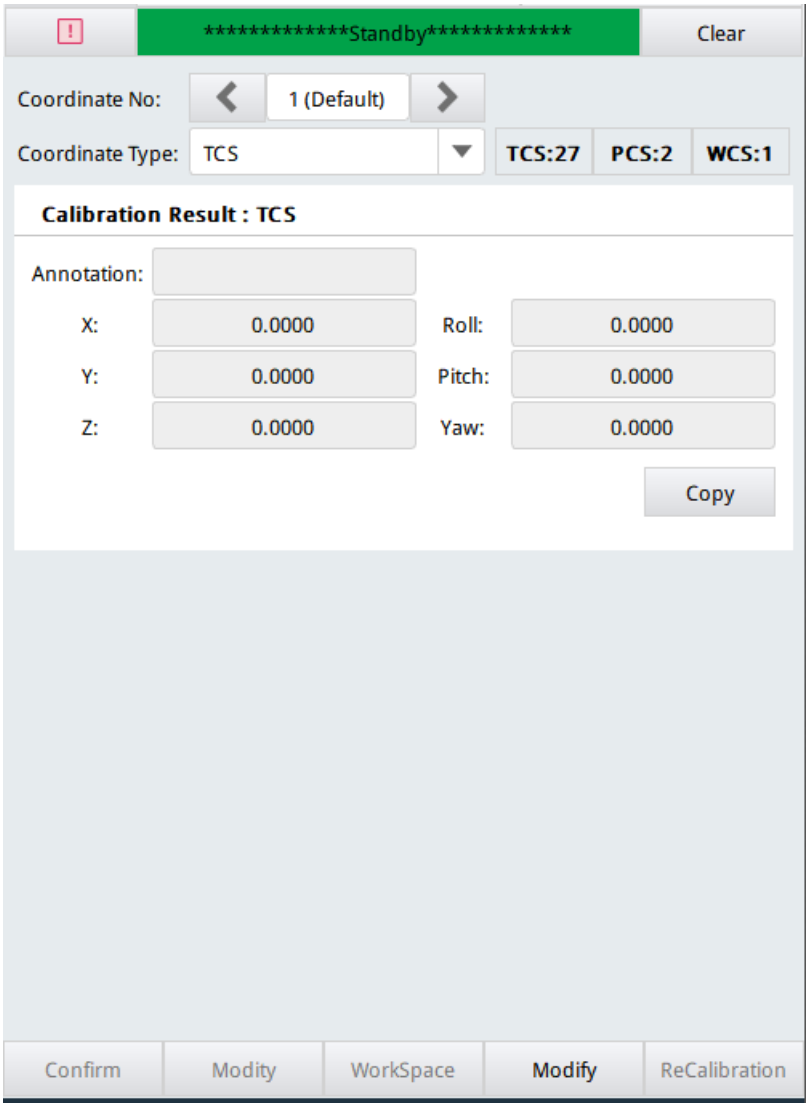


Diagram 39 View interface of teaching points

Step 5: After calibrating a workpiece 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 workpiece coordinate system.

Step 6: Verify Coordinate System

The orientation 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

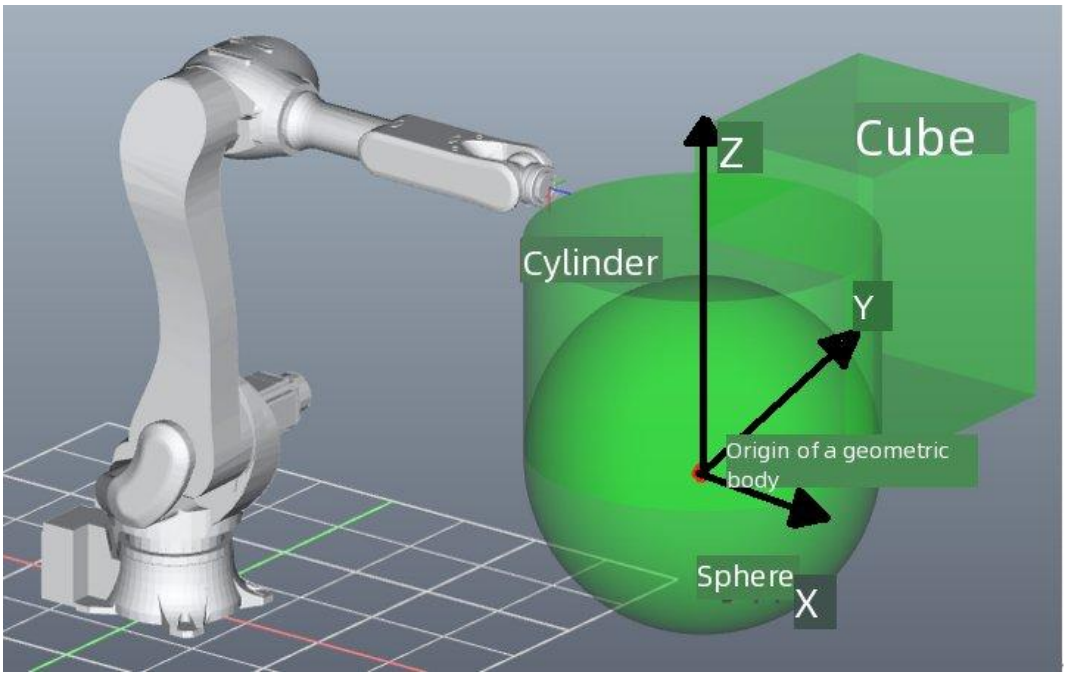


Diagram 40 Schematic diagram of workspace monitoring

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 in 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.

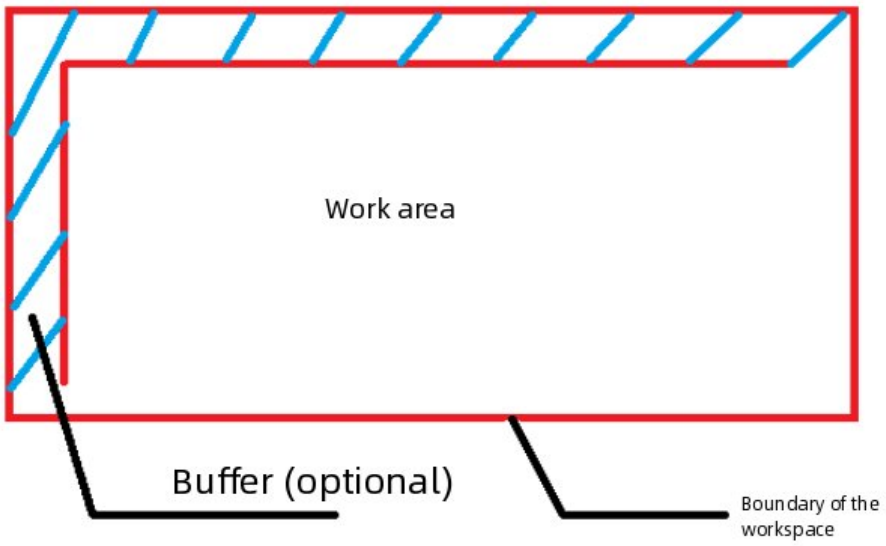


Diagram 41 Schematic diagram of work area

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.

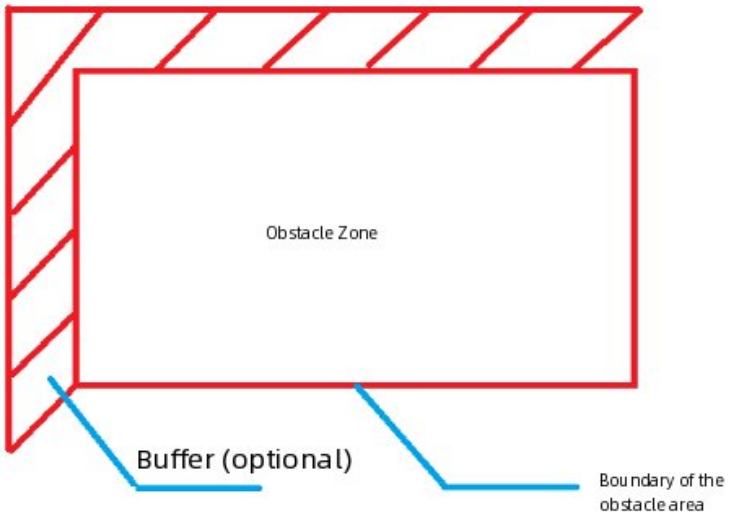


Diagram 42 Obstacle Area Diagram

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 the user 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 teaching mode, the robot will not stop the movement in an emergency; in the playback 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 workpiece coordinate system, that is, the set workpiece 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 workpiece coordinate system origin.

9.7.2.1 Geometry Origin Setting

In the {menu}  ,select and click {coordinate system management} 

to enter the coordinate system calibration interface, and select the coordinate system serial number. Select {Workpiece Coordinate System/PCS} as the coordinate system type, and then perform coordinate calibration. For the description of calibration operation steps, please skip to calibration of workpiece coordinate system.

.

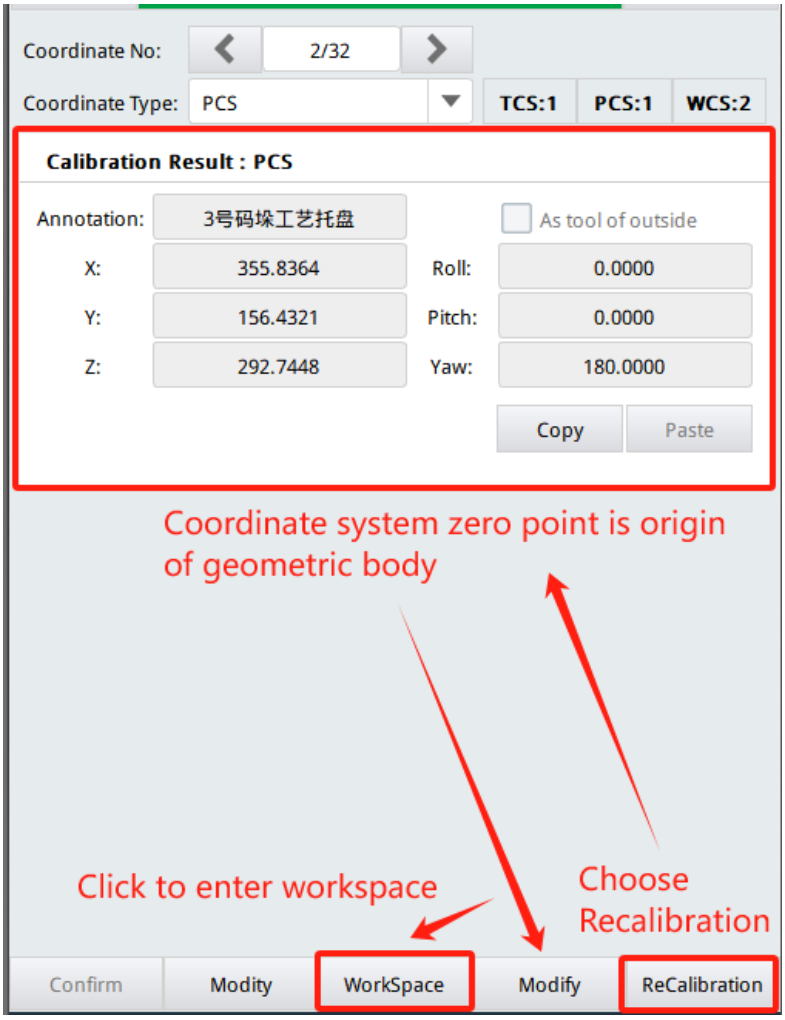
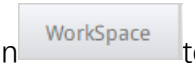


Diagram 43 Geometric origin setting

9.7.2.2 Global Monitoring Settings

After calibration of workpiece coordinate 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.

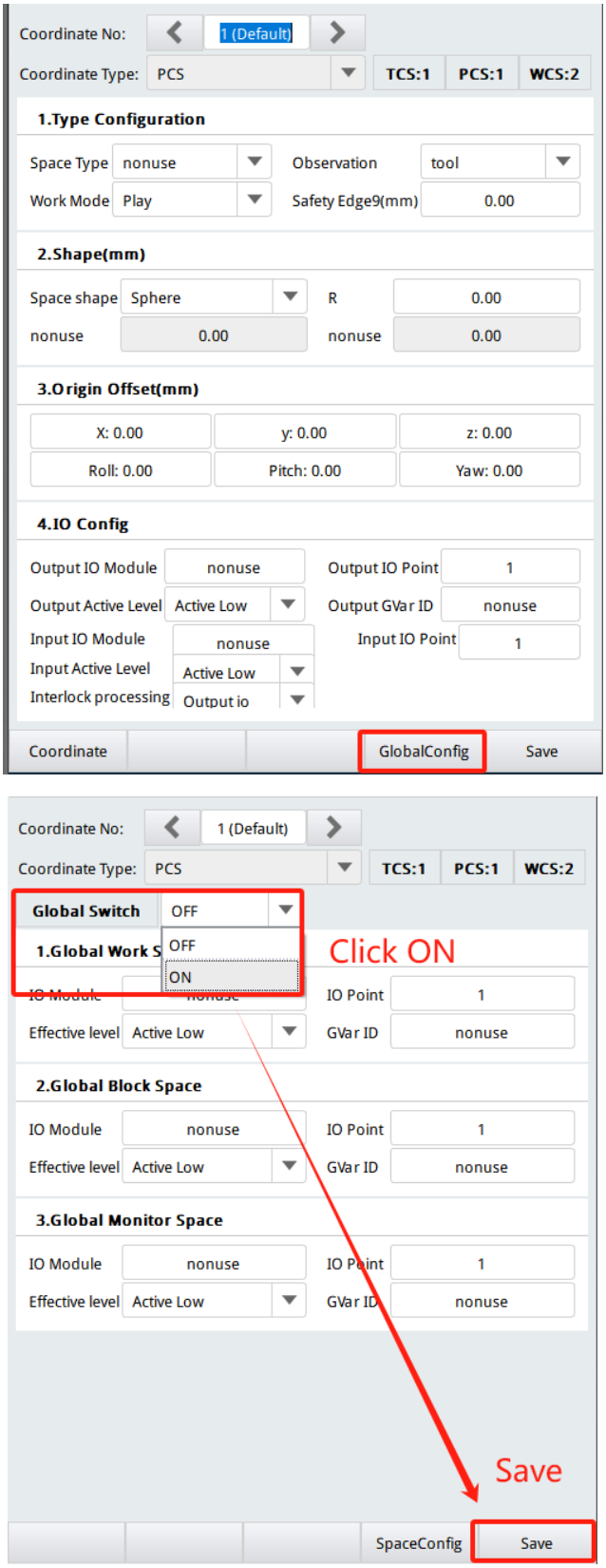


Diagram 44 Global monitoring settings

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 Number	When the current TCP point of the robot exceeds all active working areas, the corresponding Gvar sequence number is assigned 1. If not, enter -1 in the input box
	Active level	You can choose high-level trigger or low-level trigger
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
	Active level	You can choose high-level trigger or low-level trigger
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

	Active level	You can choose high-level trigger or low-level trigger

9.7.2.3 Global Monitoring Settings

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.

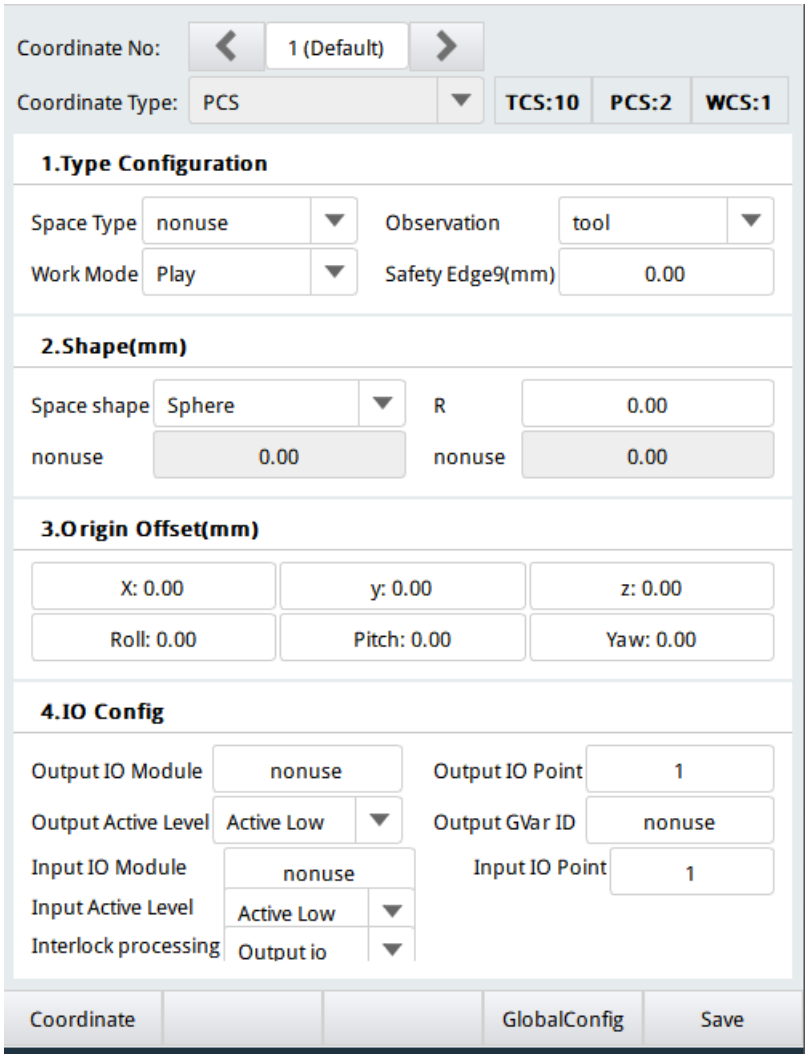


Diagram 45 Workspace Monitoring Settings

Name		Definition
Type	**Area Type**	There are four options: 1. Not effective; 2. work area;

Configuration		3. obstacle area; 4. Surveillance areas;
	Mode of operation	Three modes are available: 1. Auto mode: only effective in playback mode 2. Teaching mode: effective only in teaching mode 3. Teach + Auto mode: effective in playback 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 configuration（mm）	**Sphere**	The origin of the current corresponding workpiece coordinate system is 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 workpiece 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 workpiece 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 workpiece 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
	Inpur 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.

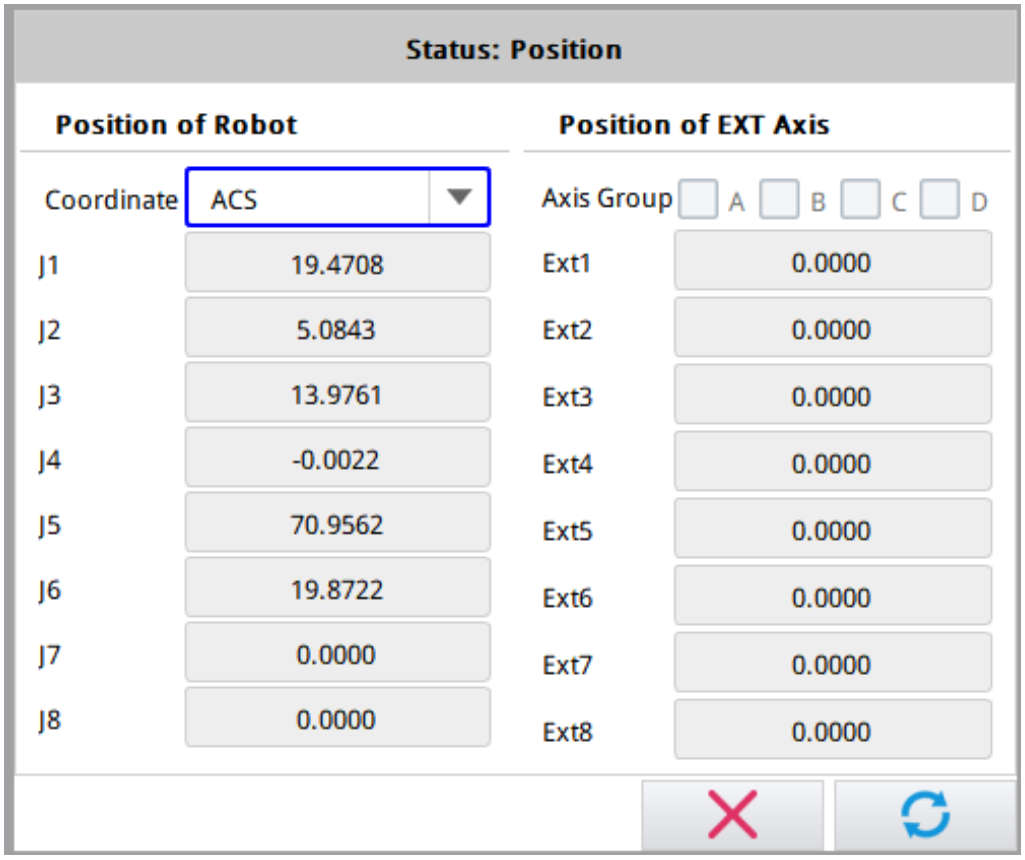


Diagram 46 Workplace Monitoring Interface

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

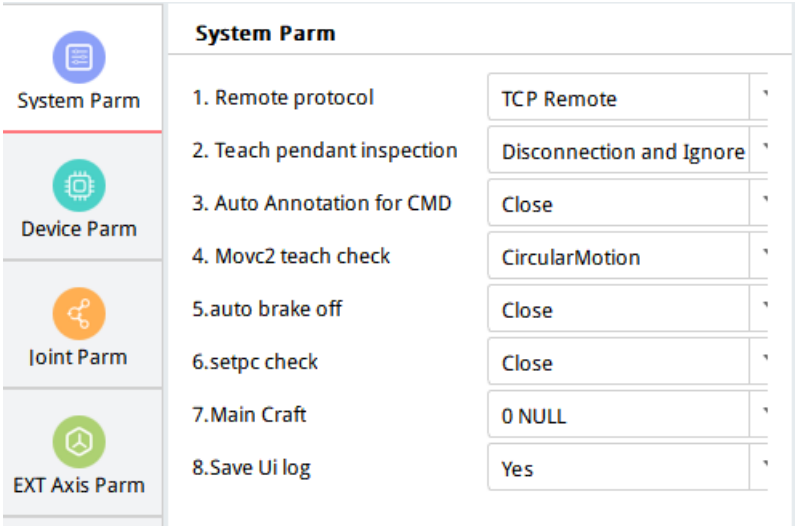


Diagram 47 System Parameter Interface

No.	Parameter Name	Parameter Name	Parameter Instruction
1	Remote Mode Type	IO control	Remote external IO for command control of robots
		Modbus TCP control	The upper computer communicates and interacts with the robot through Modbus TCP

		Remote TCP control	The upper computer communicates and interacts with the robot through TCP/IP
2	Handheld connection status check	Handheld disconnection not handled	Handheld box disconnected or previously disconnected without any prompts
		Handheld disconnection alarm	After the handheld box is disconnected, it will prompt an alarm of "handheld box disconnected or previously disconnected"
3	Automatic annotation of instructions	Close	Off - Indicates that the automatic annotation function for instructions is not enabled; Enable - means to enable automatic annotation of instructions; When writing a program, setting the instruction line will automatically add comments
		Open	
4	Movc2 Forward Teaching Inspection Trajectory	Linear motion	When teaching to run the circle instruction program, set it to run in a straight line to the midpoint of the circle
		Circular motion	When teaching to run the circle instruction program, set it to run in a circular arc manner to the midpoint of the

			circle
5	Automatic release of the brake during collision	Close	Close - indicates not using the release brake function; Open - indicates the use of the release brake function; In emergency situations, if the robot cannot be operated through the teaching pendant, this function can be activated to release the brake on the four, five, and six axes of the robot（Attention: Safety precautions must be taken）
		Open	
6	Setpc check	Close Open	Close - indicates not using Setpc function; Enable - indicates the use of Setpc for checking; After each mode switch, you need to click the Setpc button on the file editing interface to specify the startup line
7	Specify the acceptance of physical button technology	Select the corresponding process for use	Which process controls the expansion of button functions

10.2 Hardware Parameters

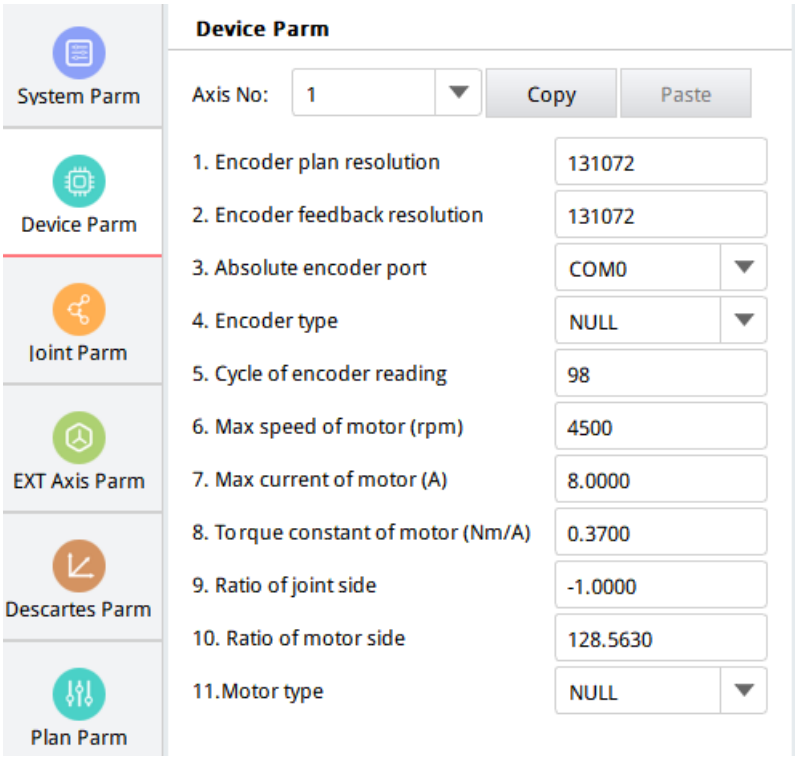


Diagram 48 Hardware Parameter Interface

No.	Parameter Name	Parameter Description
1	Encoder planning resolution	Example of encoder line count: The resolution of a 17 bit encoder is 131072
2	Encoder feedback resolution	
3	Read absolute value encoder port	Bus-based control value This parameter is invalid
4	Encoder Type	Default GthdEcat
5	Periodic reading absolute value encoder	The absolute encoder data is read regularly to refresh the current absolute position of the robot
6	Maximum motor speed(rpm)	The maximum speed of the motor is not allowed to be exceeded during actual operation
7	Maximum motor current(A)	The maximum current of the motor is not allowed to be exceeded during actual operation.

8	Motor torque constant (Nm/A)	Torque per unit current
9	Reduction ratio joint end	The ratio of the motor end rotation angle to the actuator end rotation angle (the allowable value is positive or negative to change the joint motion direction)
10	Reduction ratio motor end	
11	Drive type	Default GthdEcat

10.3 Axis

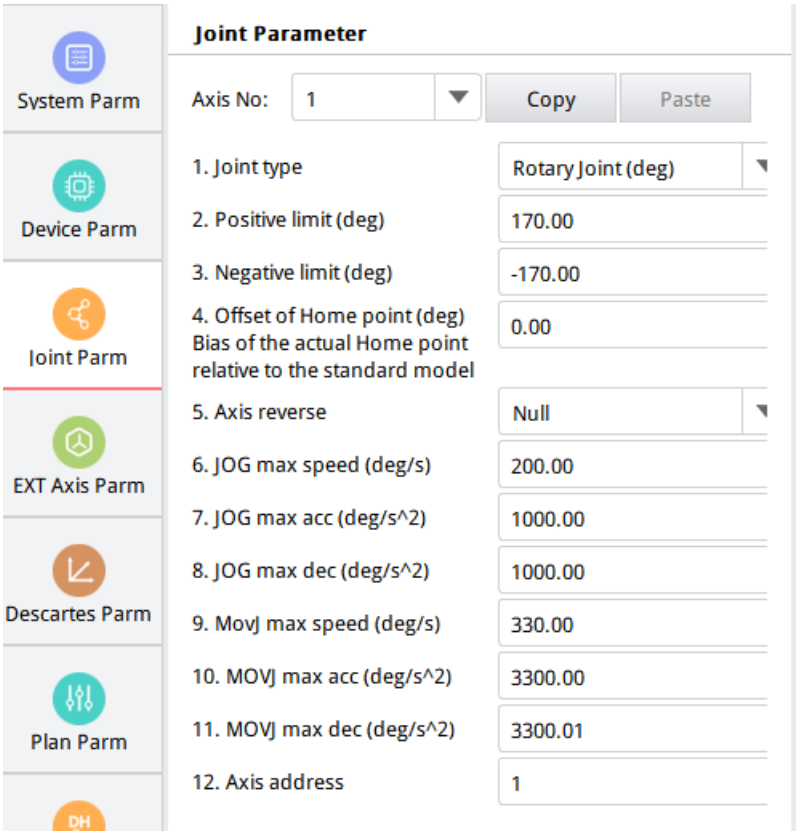


Diagram 49Axis Parameter Interface

No.	Parameter Name	Parameter Instruction
1	Axis Type	Not used; Rotating joint (deg); Translational joint (mm);
2	Joint positive limit	Maximum range of joint motion

3	Joint negative limit position	Minimum range of joint motion
4	Joint zero offset	The deviation value between the user-defined joint zero point and the zero point of the internal robot model in the system
5	Reverse joint	Whether the user-defined joint rotation direction is opposite to the direction of the internal robot model in the system
6	Axis JOG Max Speed	Maximum velocity of jog axis operation key in axis coordinate system
7	Axis JOG Max Acceleration	Maximum acceleration of jog axis operation key in axis coordinate system
8	Axis JOG Maximum Deceleration	Maximum value of motion deceleration of jog axis operating key in axis coordinate system
9	Maximum velocity of axis motion	MOVJ Command Axis Speed Max
10	Maximum axis motion acceleration	MOVJ Command Axis Acceleration Max
11	Maximum axis motion deceleration	MOVJ commanded axis deceleration maximum
12	Axis hardware address	The station number used for bus communication shall match the setting of the upper computer

10.4 Auxiliary Axis

Auxiliary axis parameters refer to the {Axis} parameter setting, and auxiliary axes that are not used have parameter 1{Axis Type} selected as "Not Used".

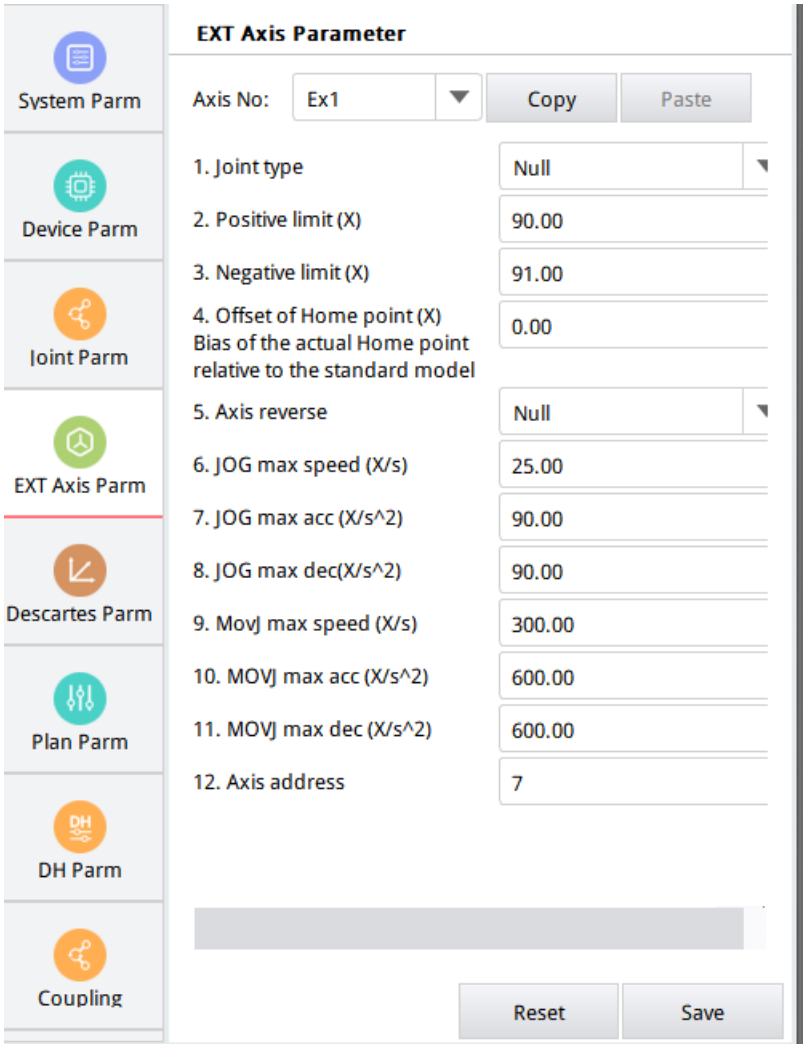


Diagram 50 Auxiliary Shaft Parameter Interface

10.5 Descarts

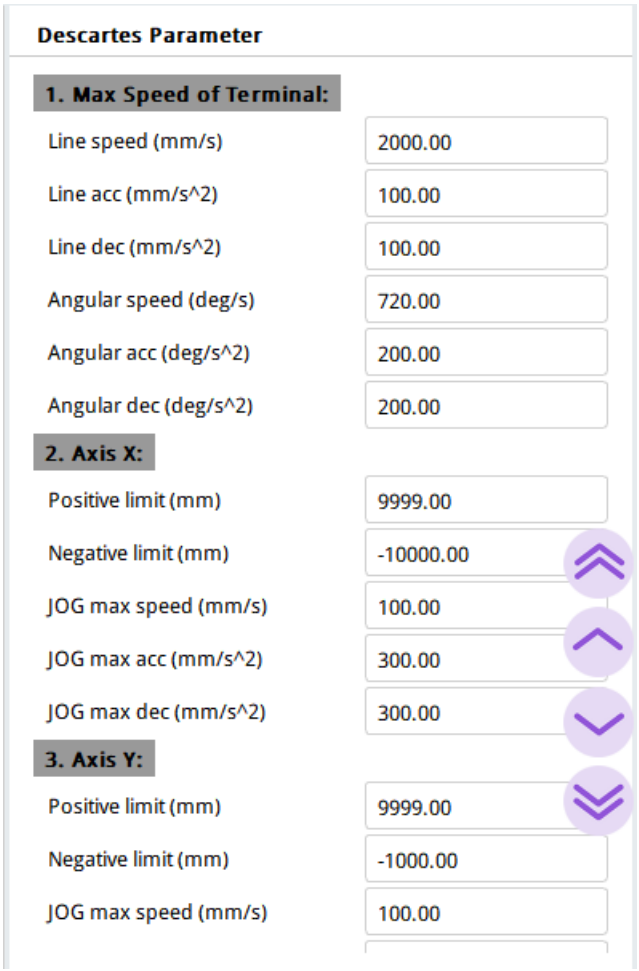


Diagram 51 Cartesian Parameter Interface 1

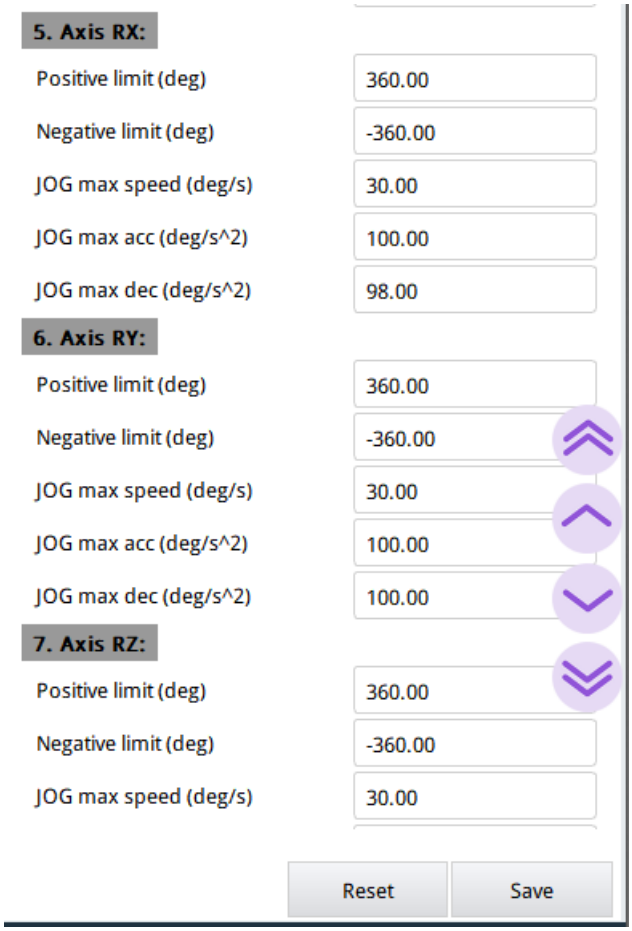


Diagram 52 Cartesian Parameter Interface 2

No.	Parameter Name	Parameter Instruction
1	Positive limit	Maximum range of motion in Cartesian coordinates
2	Negative limit	Minimum range of motion in Cartesian coordinates
3	JOG Max Speed	Maximum speed of jog axis operation key in Cartesian coordinate system
4	JOG Max Acceleration	Maximum acceleration of jog axis operating key in Cartesian coordinate system
5	JOG maximum deceleration	Maximum deceleration of jog axis operating key in Cartesian coordinate system
6	Linear velocity	Maximum resultant linear velocity at the end of a point in Cartesian coordinates
7	Linear acceleration	Maximum resultant linear acceleration at the end of a point in Cartesian coordinates

8	Linear deceleration	Maximum resultant linear deceleration at the end of a point in Cartesian coordinates
9	Angular velocity	Maximum resultant angular velocity at the end of a point in Cartesian coordinates
10	Angular acceleration	Maximum resultant angular acceleration at the end of a point in Cartesian coordinates
11	Angular deceleration	Maximum value of resultant angular deceleration at the end of a point in Cartesian coordinates

10.6 Planning Parameters

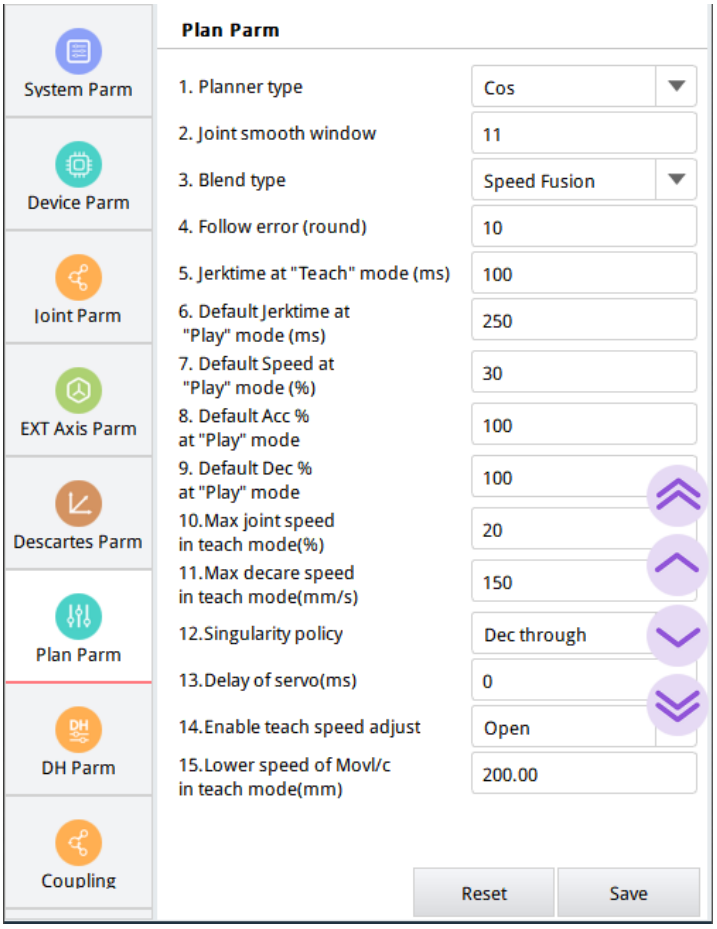
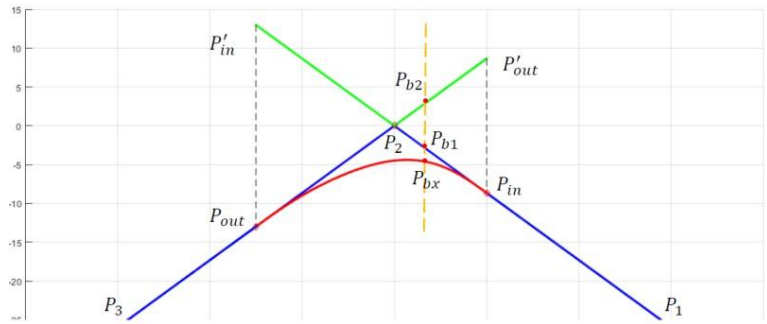


Diagram 53 Planning Parameter Interface

No.	Parameter Name	Parameter Instruction
1	Planning Type	Cos:Jerk is the trajectory planning of the Cos function Quintic:Jerk is a trajectory planner for quintic curves Door:Jerk is the trajectory plan for the square wave SawTooth:Jerk is a trajectory planner for sawteeth Exp:Jerk is the trajectory planning for exponential curves Bell:Jerk is the trajectory planning of a conic Refer to the following figure for planning curve
2	Planner Smoothing Window	Output Averaging Filter Length
3	Trajectory Transition Type	Fusion: Blended path transition, no abrupt change in velocity and acceleration, controllable trajectory, but not constant velocity, default option Arc: arc transition, to ensure that the transition section speed is constant, the trajectory is controllable, but it will lead to acceleration mutation. Not suitable for axis trajectory and pose planning. Spline: spline transition, which ensures constant velocity and controllable trajectory of transition section, but will cause acceleration mutation, which is smaller than Arc transition. Not suitable for axis trajectory and pose planning VelBlend: Velocity transition, i.e. two motion commands are planned at the same time, and the velocity is superimposed to obtain a new position. the shape of that trajectory is related to the speed at which it enters the Blend segment

		 **Fig: mixed end slot path**
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	Acceleration time of planner in teach mode	Manual jog time with the acceleration time, adjust the jog will be more fierce, easy to shake
6	Auto Run Default Acceleration Time	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	Auto Run Default Speed Percentage	Default playback mode running speed at startup
8	Auto Run Default Acceleration Percent	Startup default playback mode running acceleration
9	Auto Run Default Deceleration Percentage	Startup default playback mode running deceleration

10	Maximum joint speed in teaching mode (%)	Limit maximum joint movement speed in teaching mode
11	Maximum Cartesian velocity in teaching mode (mm/s)	Maximum velocity of Cartesian space motion in teaching mode
12	Singular point processing	The strategy chosen by the robot when passing through a singularity can be to give an alarm and stop when passing through the singularity, or to slow down and pass through the singularity slowly
13	Upper servo delay (ms)	Increase the delay of the servo
14	Enable automatic adjustment of teaching speed	Teaching speed automatic adjustment switch, used in conjunction with 15 items
15	Minimum speed adjustment in Movl/teaching mode (mm)	When setting the instruction speed to low in Movl/c, use the lowest speed value under the forward check

Planner type legend:

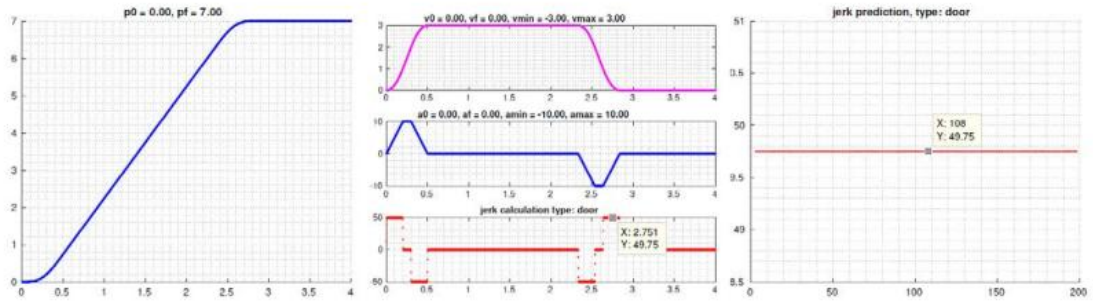


Figure 1: door-shaped jerk

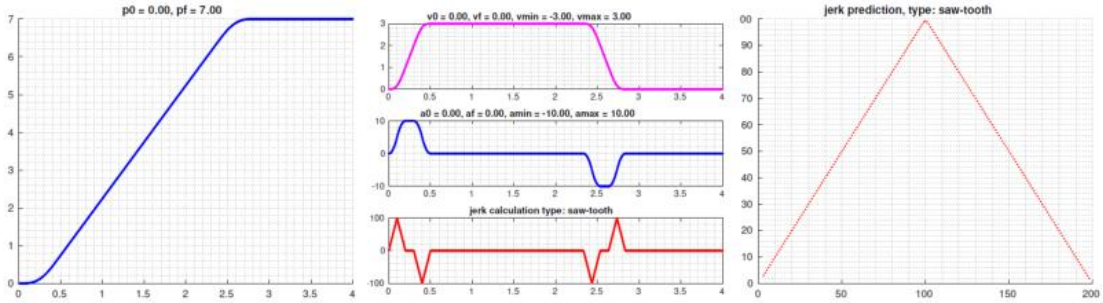


Figure 2: saw-tooth-shaped jerk

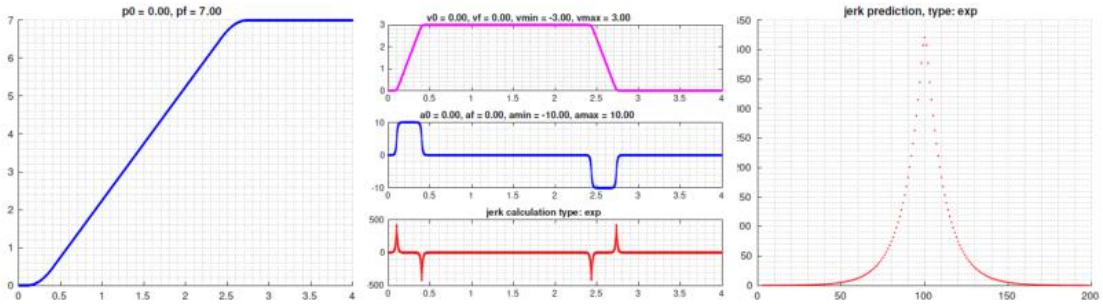


Figure 3: exp-shaped jerk

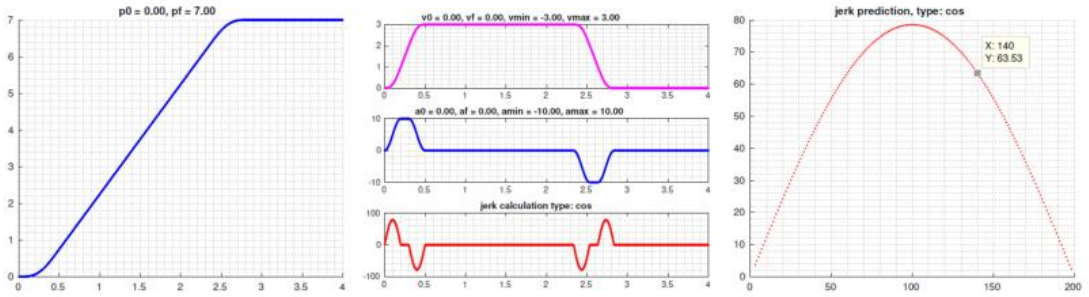


Figure 4: cos-shaped jerk

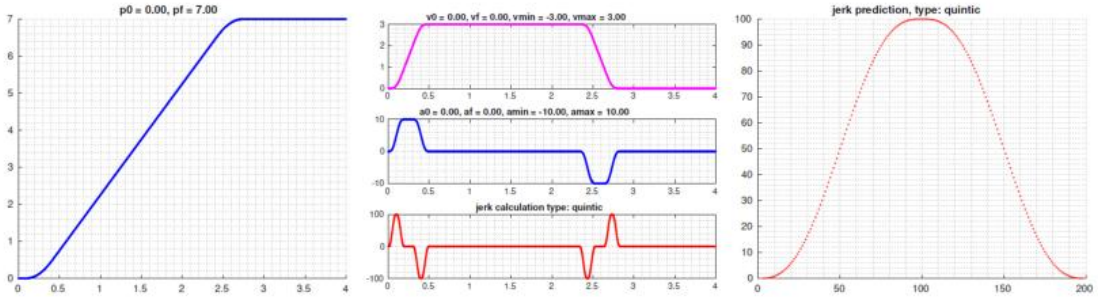


Figure 5: quintic-shaped jerk

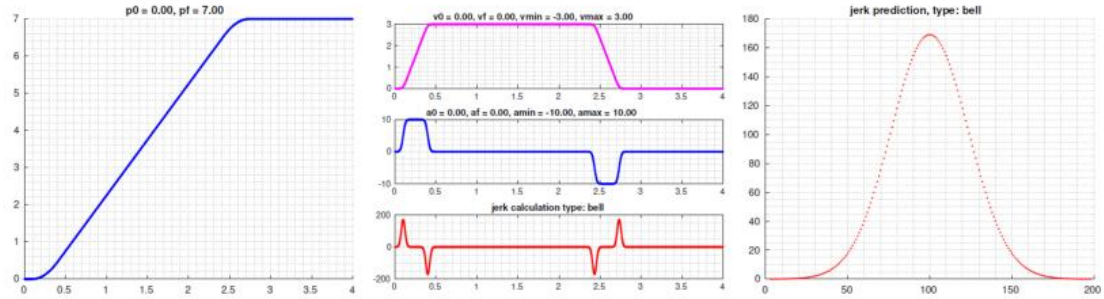


Figure 6: bell-shaped jerk

Diagram 54 Planning Curve

10.7 DH Parameters

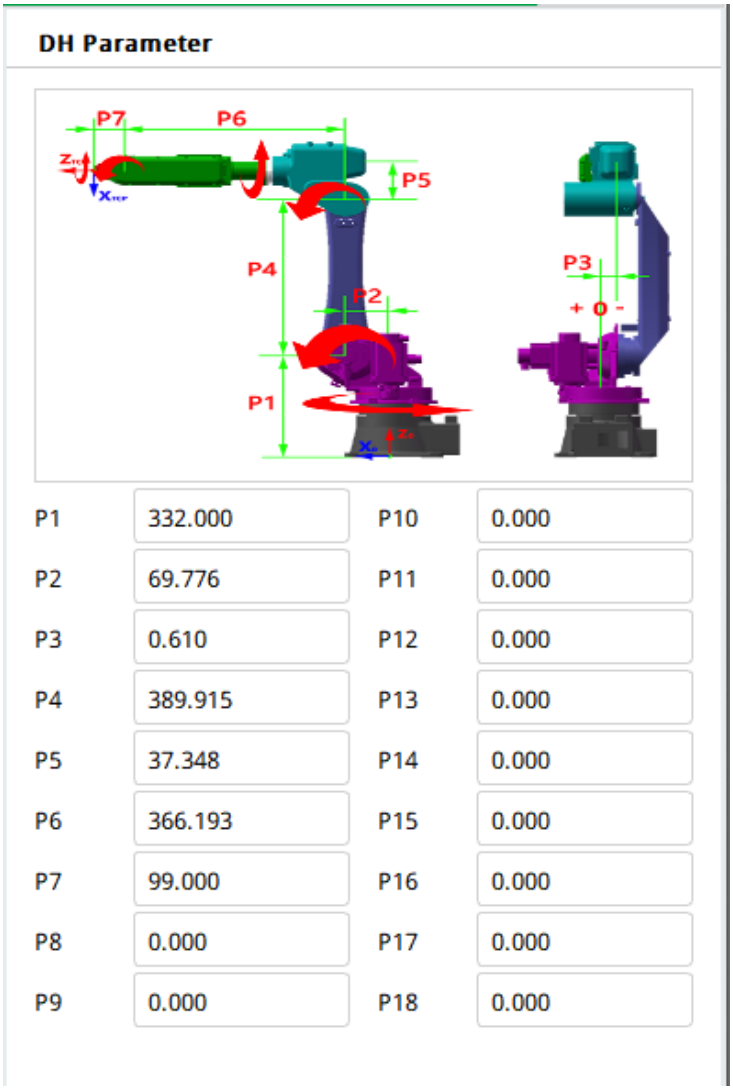


Diagram 55 DH Parameter Interface

No.	Parameter Name	Parameter Instruction
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	Denotes the coupling ratio between axis 6 and axis 4
4	P12/P13	Denotes the coupling ratio between axis 6 and axis 5
5	P14~P18	Reserved

10.8 Coupling

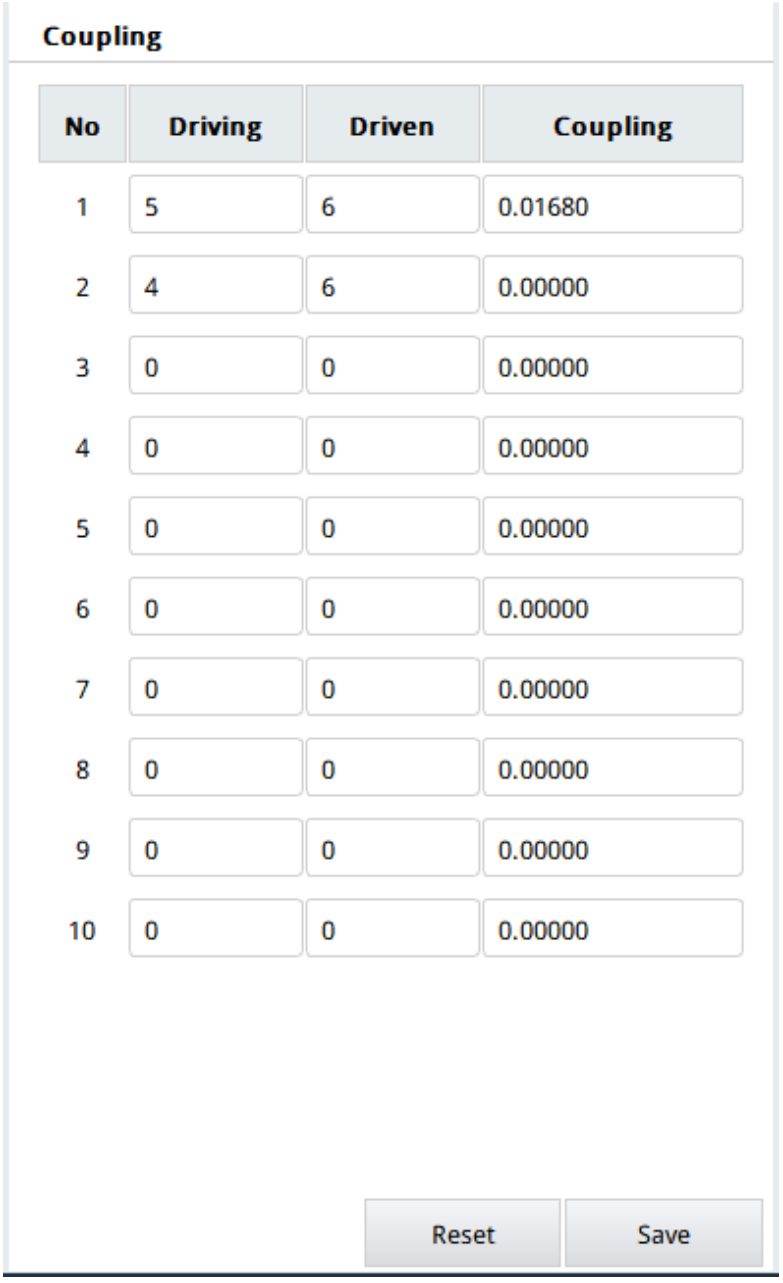


Diagram 56 Coupling Ratio Setting Interface

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

No	Function	Group	Pin	Value	Status
1	Anti-Collision	1	0	1	
2	Forbit Move	1	0	1	
3	Station Pause	1	0	1	
4	Safe Gate	1	0	1	
5	Station Estop	1	0	1	
6	Alarm of High Tem	1	0	1	
7	Alarm of Battery	1	0	1	
8	ExtEstop1	1	0	1	
9	ExtEstop2	1	0	1	
10	ExtEstop3	1	0	1	
11	ExtEstop4	1	0	1	

Diagram 57 IO Input Interface

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

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

Valid value: Normal 1 is valid and can be negated.

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

11.1.2 Function Description

No.	Input/Output type	Parameter Name	Instruction
1	Digital Input	Anti-collision signal	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	External Disable Robot	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	Pause	When the system detects that the pause signal is triggered, the system pauses the execution of the teach file.
4	Digital Input	Safety 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	Scram	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

		alarm	stops moving and prompts the high temperature alarm.
7	Digital Input	Battery alarm	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

No	Function	Group	Pin	Value	Status
1	Servo ON	1	0	1	
2	Idle	1	0	1	
3	Alarm	1	0	1	
4	Pause	1	0	1	
5	Working	1	0	1	
6	Warning	1	0	1	
7	Running	1	0	1	
8	Teach Mode	1	0	1	
9	Play Mode	1	0	1	
10	Remote Mode	1	0	1	
11	TP Pause	1	0	1	
12	TP Estop	1	0	1	
13	TP Start	1	0	1	
14	TP Switch	1	0	1	
15	Station Estop	1	0	1	
16	Station Pause	1	0	1	

Diagram 58 IO Output Interface

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

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

Valid value: Normal 1 is valid and can be negated.

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

11.2.2 Function Instruction

No.	Input/Output type	Parameter Name	Description
1	Digital output	Servo enabling	Output signal after robot is enabled
2	Digital output	Idle	Enable in standby state
3	Digital output	Alarm (red alarm)	The robot is in a red alarm abnormal state
4	Digital output	Pause	The robot is in a paused state
5	Digital output	At work	The robot is in the state of executing the teach file.
6	Digital output	Alarm（yellow warning）	The robot is in a warning prompt state
7	Digital output	In motion	the state in which that robot is in motion
8	Digital output	Teach Mode Status	Output signal when handheld teach pendant is in teach mode
9	Digital output	Playback Mode Status	Output signal when handheld teach pendant is in teach mode
10	Digital output	Remote Mode Status	Output signal when handheld teach pendant is in teach mode
11	Digital output	Hand-held box emergency stop output	Output signal when emergency stop signal of hand-held box is triggered.

12	Digital output	Hand-held box emergency stop output	Output signal when emergency stop signal of hand-held box is triggered.
13	Digital output	Hand-held box emergency stop output	Output signal when emergency stop signal of hand-held box is triggered.
14	Digital output	Three-step switch of hand-held box output	Output signal when trigger button signal of three-step switch of hand-held box.
15	Digital output	Workstation scram	The workstation emergency stop input signal is received and the system is in the workstation emergency stop state.
16	Digital output	Workstation scram	The workstation emergency stop input signal is received and the system is in the workstation emergency stop 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

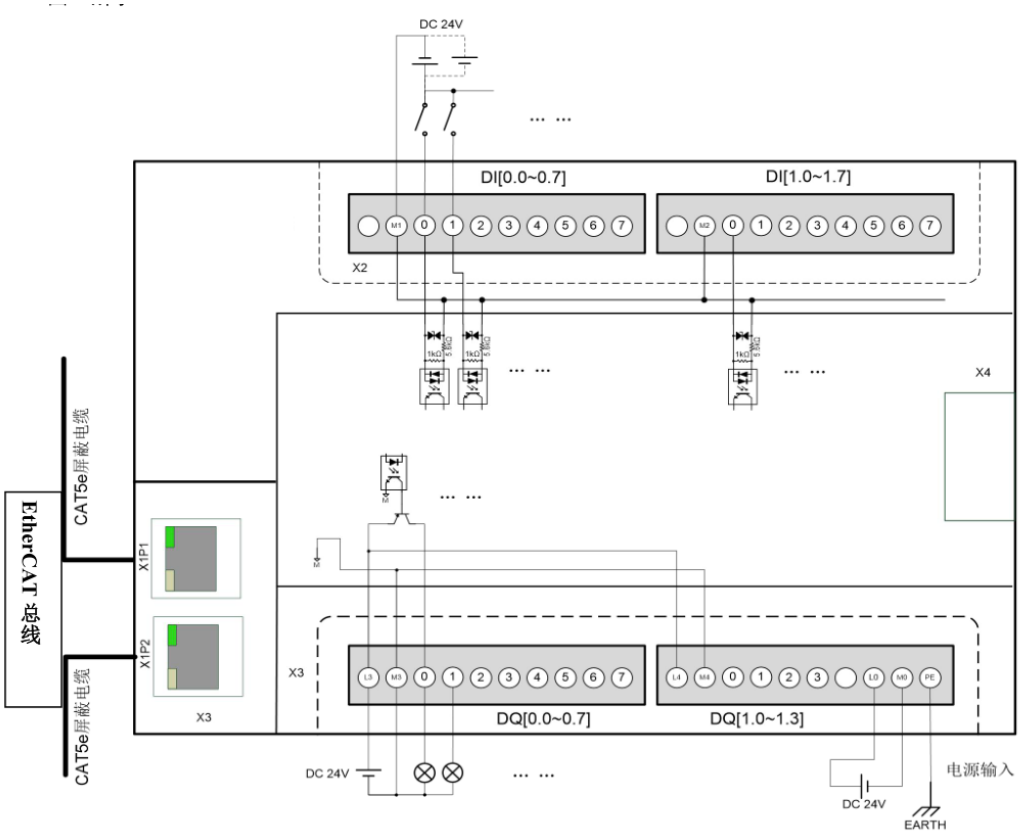


Diagram 59 IO Wiring

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, input 24V;

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

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 Instruction to the Interface

12.1.1 File Settings

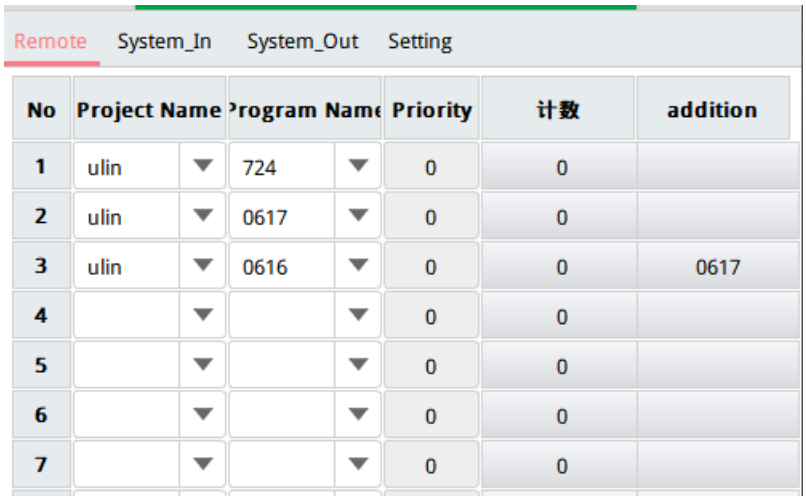


Diagram 60 IO File Settings

File No.: The file number combined by remote input file bits. In binary mode, only file numbers 1 to 8 are valid, and in combination mode, file numbers 1 to 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.

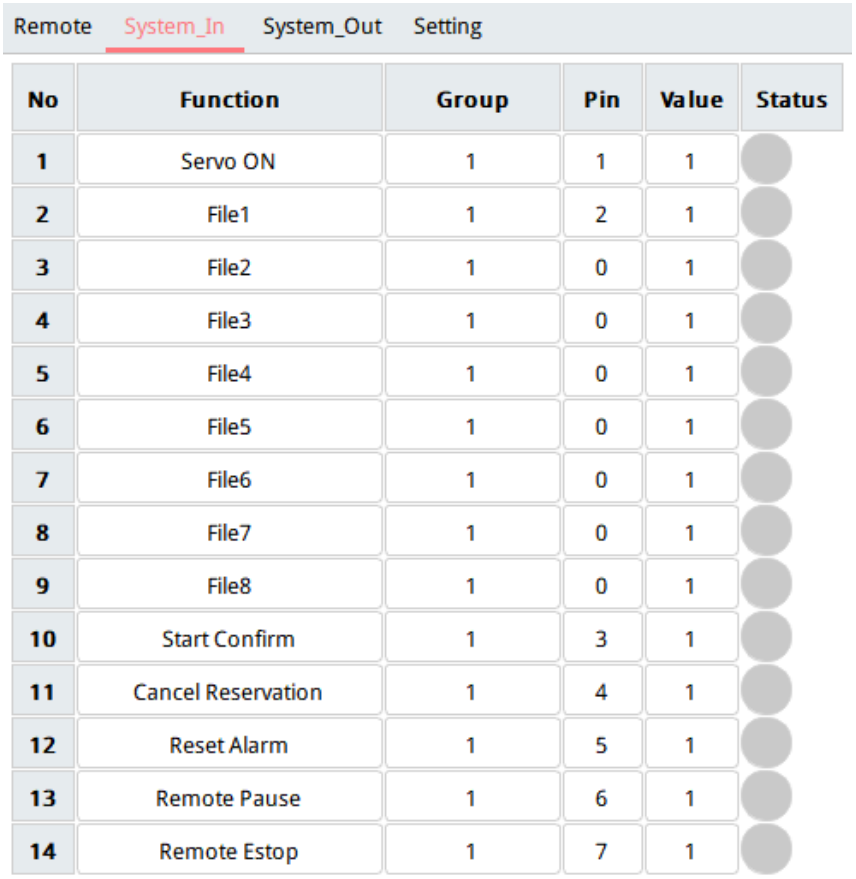
Priority: Display the order of file execution. Priority is 1, which is the program being executed. In single mode, priority 2~8 are the programs being reserved; Only programs with priority 1 are in combined mode.

Completion times: The number of times the appointment file program is executed. It is mainly used for piecework statistics and can be modified manually.

Additional task: When the completion times of the corresponding file number

reach the integer multiple of the set execution times, an additional task is automatically triggered to be executed.

12.1.2 Remote IO Input



No	Function	Group	Pin	Value	Status
1	Servo ON	1	1	1	
2	File1	1	2	1	
3	File2	1	0	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	3	1	
11	Cancel Reservation	1	4	1	
12	Reset Alarm	1	5	1	
13	Remote Pause	1	6	1	
14	Remote Estop	1	7	1	

Diagram 61 Remote IO Input

Module: Indicates the address ID of IO module. The value of the address ID of extended IO slave station starts from 1.

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

Valid value: Normal 1 is valid and can be negated.

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

Description:

No.	Input/output type	Parameter Name	Description
1	Remote IO Input	Servo Enabling	In remote IO control mode, the system receives this signal Robot Enable.
2	Remote IO Input	File bit 1	File number input bit 1, if it is a single mode to represent file bit 1; If it is combined mode, it indicates the first bit of 8421 coding.
3	Remote IO Input	File bit 2	File number input bit 2, if it is a single mode representative file bit 2; If it is combined mode, it indicates the second bit of 8421 coding.
4	Remote IO Input	File bit 3	File number input bit 3, if it is a single mode representative file bit 3; If it is combined mode, it indicates the third bit of 8421 coding.
5	Remote IO Input	File bit 4	File number input bit 4, if it is a single mode representative file bit 4; If it is combined mode, it indicates the fourth bit of 8421 coding.
6	Remote IO Input	File bit 5	File number input bit 5, if it is a single mode representative file bit 5; If it is combined mode, it indicates the fifth bit of 8421 coding.
7	Remote IO Input	File bit 6	File number input bit 6, if it is a single mode representative file bit 6; If it is combined mode, it indicates

			the sixth bit of 8421 coding.
8	Remote IO Input	File bit 7	File number input bit 7, if it is a single mode representative file bit 7; If it is combined mode, it indicates the seventh bit of 8421 coding.
9	Remote IO Input	File bit 8	File number input bit 8, if it is a single mode representative file bit 8; If it is combined mode, it indicates the eighth bit of 8421 coding.
10	Remote IO Input	Startup Confirmation	File start signal in combination mode. This signal is inactive in standalone mode.
11	Remote IO Input	Conceal 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

Remote System_In System_Out Setting

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

Diagram 62 Remote IO Output

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

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

Valid value: Normal 1 is valid and can be negated.

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

Description:

No.	Input/Output type	Parameter Name	Description
1	Remote IO Output	Servo Enabling	Output signal after robot is enabled
2	Remote IO Output	Idle	Enable in standby state
3	Remote IO Output	Alarm (red alarm)	The robot is in a red alarm abnormal
4	Remote IO Output	Pause	The robot is in a paused state
5	Remote IO Output	At work	The robot is in the state of executing the teach file.
6	Remote IO Output	Warning(yellow warning)	The robot is in a warning prompt state
7	Remote IO Output	In motion	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	Startup Confirmation	Remote input enable confirm pressed

12.1.4 Basic Settings

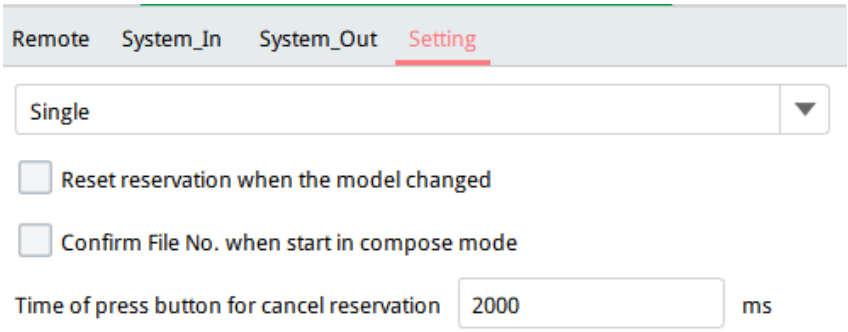


Diagram 63 Basic Settings

There are two working modes of remote IO: 1. Not used, 2. Single mode, 3. Combined mode.

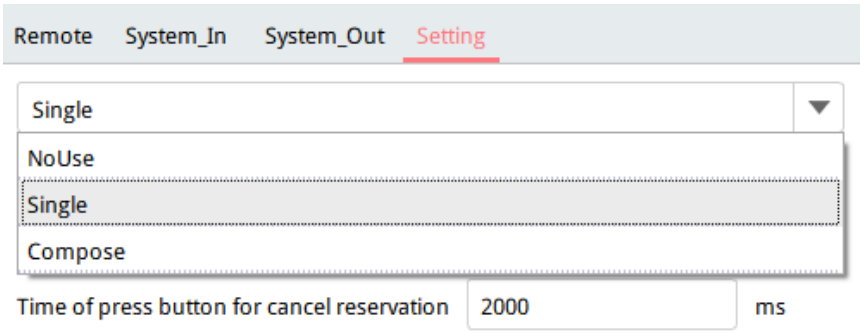


Diagram 64 Remote IO Work Mode

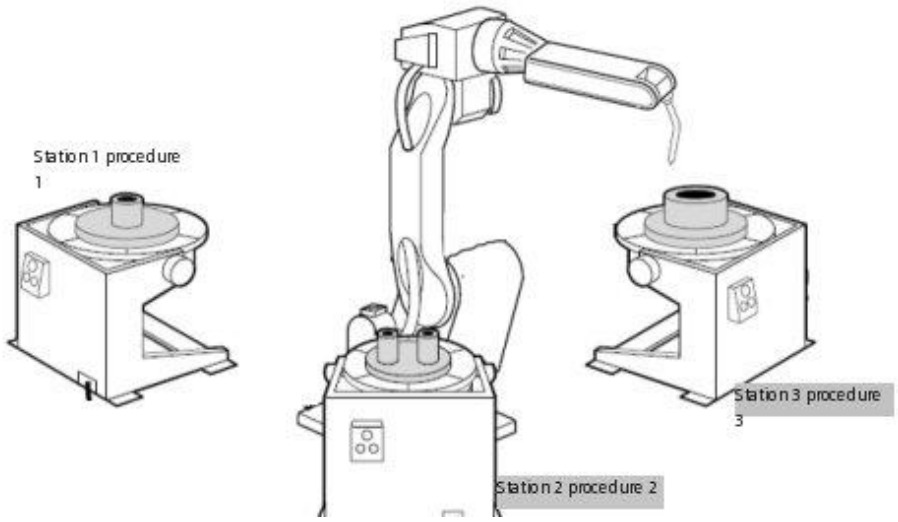
The mode switch reset reservation and **the confirmation signal when the combination mode is started** are standby functions.

Mode switch reset reservation If checked, all reservations will be cleared when the handheld box mode is switched, and re-reservation is required when switching back to remote mode.

If **the confirmation signal is checked when the combination mode is started,** the file position signal (i.e., the corresponding file signal (combination) is continuously input) shall be detected when the combination mode is started, the confirmation signal shall be started, and the file position input signal can be cancelled after the station file starts to be executed.

12.2 Use of Standalone Mode

12.2.1 Application Scenarios



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 program 1 of workpiece 1.

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

On the tool 3, the program 3 of the workpiece 3 is executed.

During reproduction, switch to "REMOTE" mode, after preparing workpiece 1, press the start button on tool 1, and the robot executes program 1. When executing Procedure 1, prepare Workpiece 2 and Workpiece 3, and then press the start buttons on Tooling 2 and Tooling 3 in turn. At this time, even if the program 1 is still running, since the start button has been pressed, the program 2 and the program 3 are scheduled in order, and the programs are executed in the order of the reservation.

The system can reserve up to 8 stations.

12.2.2 Operation Steps

12.2.2.1 Setting Remote Work Mode

Enter {Remote IO} - {Basic Settings} - {Remote IO Mode} to set to {Separate} mode.

12.2.2.2 Setting Remote File Names

Step 1 Create a workstation program.

enter {main menu} - {project management}-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 IO} interface and select the {File Settings} property page. The settings are as follows:

Remote | System_In | System_Out | Setting

No	Project Name	Program Name	Priority	计数	addition
1	ulin	724	0	0	
2	ulin	0617	0	0	
3	ulin	0616	0	0	0617
4			0	0	
5			0	0	
6			0	0	
7			0	0	

12.2.2.3 Setting Remote IO Input

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

The remote operation robot can work at 3 stations. File position 1, file position 2, file position 3 and enable signal must be set. 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	1	1	
2	File1	1	2	1	
3	File2	1	0	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	3	1	
11	Cancel Reservation	1	4	1	
12	Reset Alarm	1	5	1	
13	Remote Pause	1	6	1	
14	Remote Estop	1	7	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	1	1	
2	Idle	1	2	1	
3	Working	1	3	1	
4	Pause	1	4	1	
5	Alarm	1	5	1	
6	Warning	1	6	1	
7	Running	1	7	1	
8	File1	1	8	1	
9	File2	1	9	1	
10	File3	1	10	1	
11	File4	1	11	1	
12	File5	1	12	1	
13	File6	1	13	1	
14	File7	1	14	1	
15	File8	1	15	1	
16	Start Confirm	1	16	1	

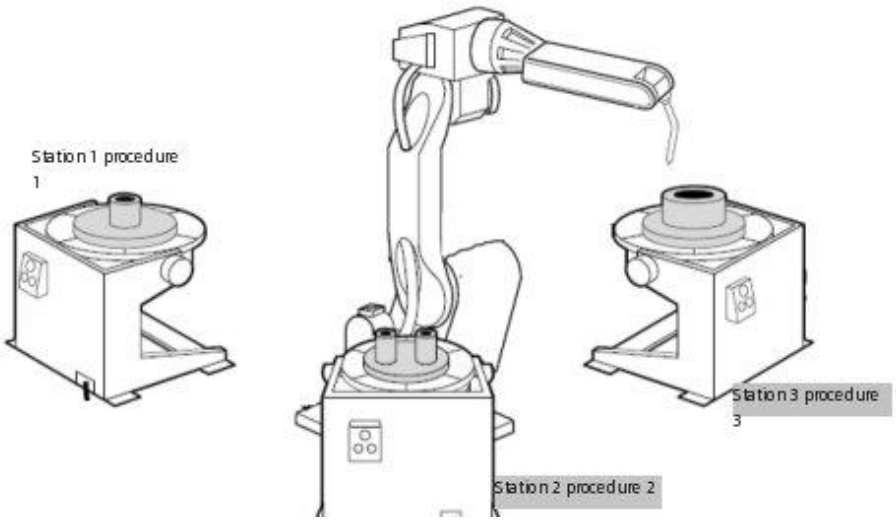
12.2.2.5 Start Reservation

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 steps 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 Scenarios



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 program 1 of workpiece 1.

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

On the tool 3, the program 3 of the workpiece 3 is executed.

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 IO} - {Basic Settings} - {Remote IO Mode} to set to {Combined} mode.

12.3.2.2 Setting Remote File Names

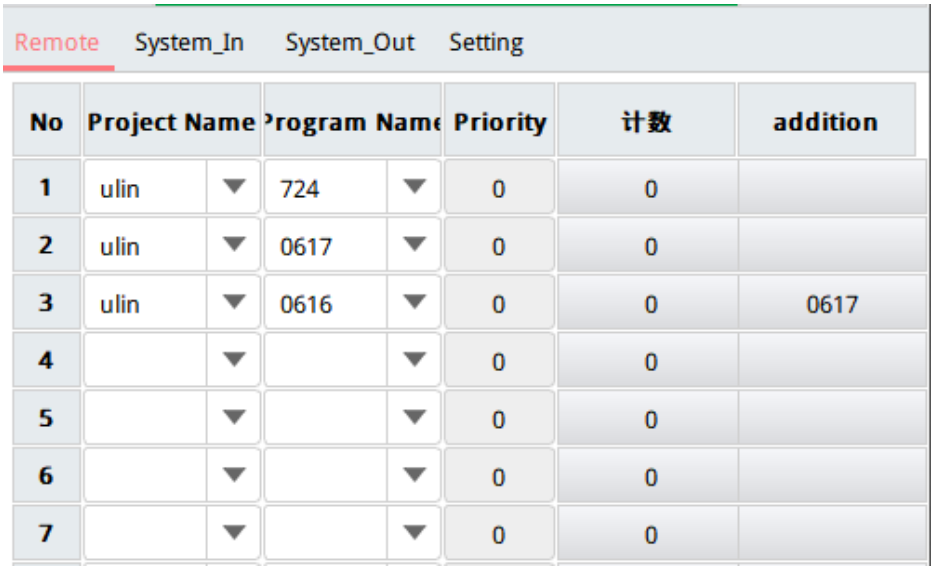
Step 1 Create a workstation program

enter {main menu} - {project management} - 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 IO} interface and select the {File Settings} property page. The settings are as follows:



12.3.2.3 Setting remote IO Input

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

The remote operation robot can work at 3 stations. File position 1, file position 2, file position 3 and enable signal must be set. 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	1	1	
2	File1	1	2	1	
3	File2	1	0	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	3	1	
11	Cancel Reservation	1	4	1	
12	Reset Alarm	1	5	1	
13	Remote Pause	1	6	1	
14	Remote Estop	1	7	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.

Remote System_In System_Out Setting

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

12.3.2.5 Starting Booking

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 bit 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.

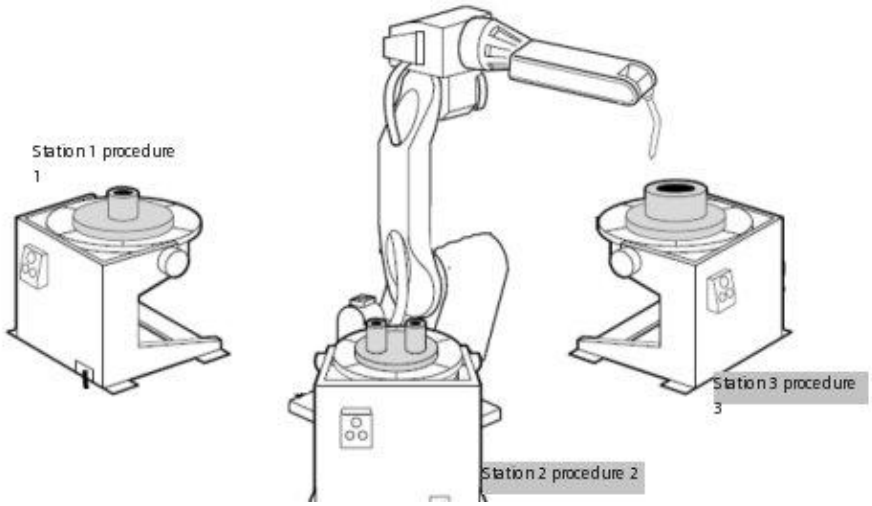
Start the work flow of station 2: after completing the setting of steps 1-4, switch to remote mode--input enable signal 1.4--wait for normal enable of robot--input signal 1.2 to file position 2--reservation of station 2 is completed--start confirmation signal 1.5--robot starts working at station 2 (demo.b2),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

Enter {Remote IO} - {Basic Settings} - {Remote IO Mode}, which can be set to

{Single} or {Combined} mode as required.

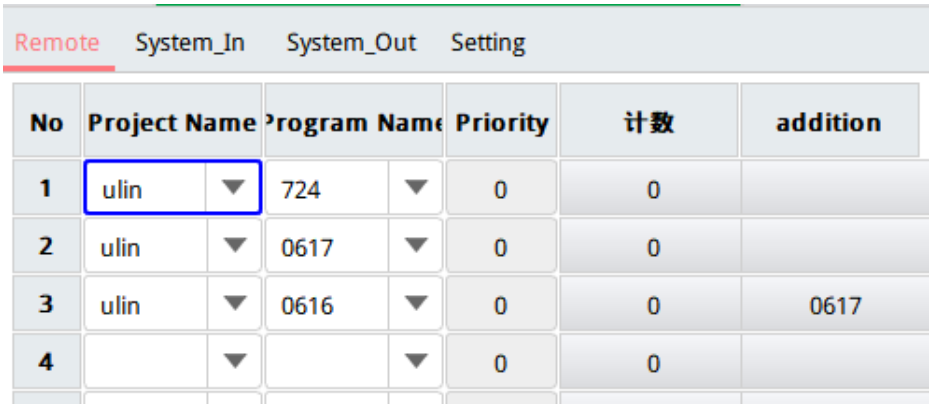
12.4.2.2 Setting Remote File Names

Step 1 Create a workstation program

enter {main menu}-{project management}-create a demo project, and create a new station program with program names of a1,b2and c3.

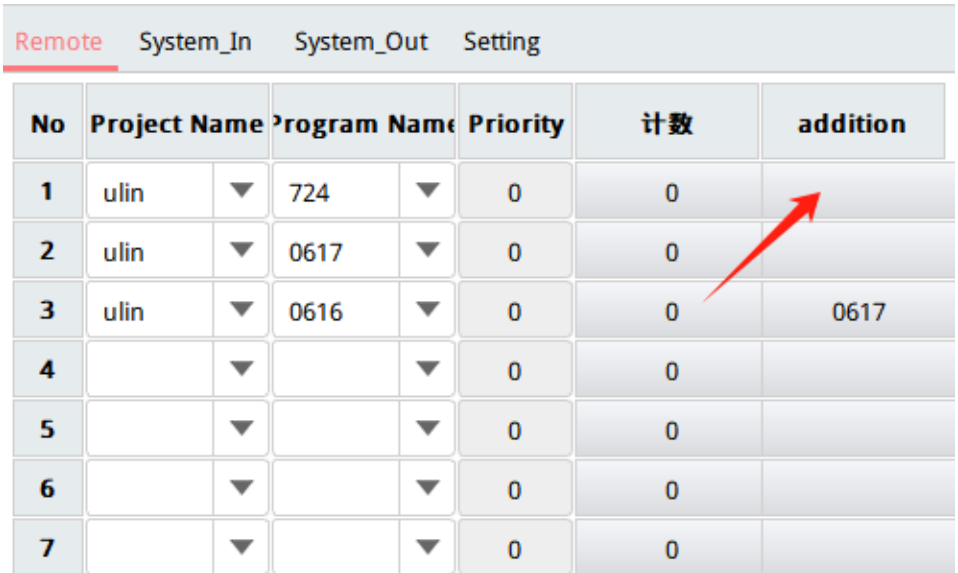
Step 2 Set up remote files

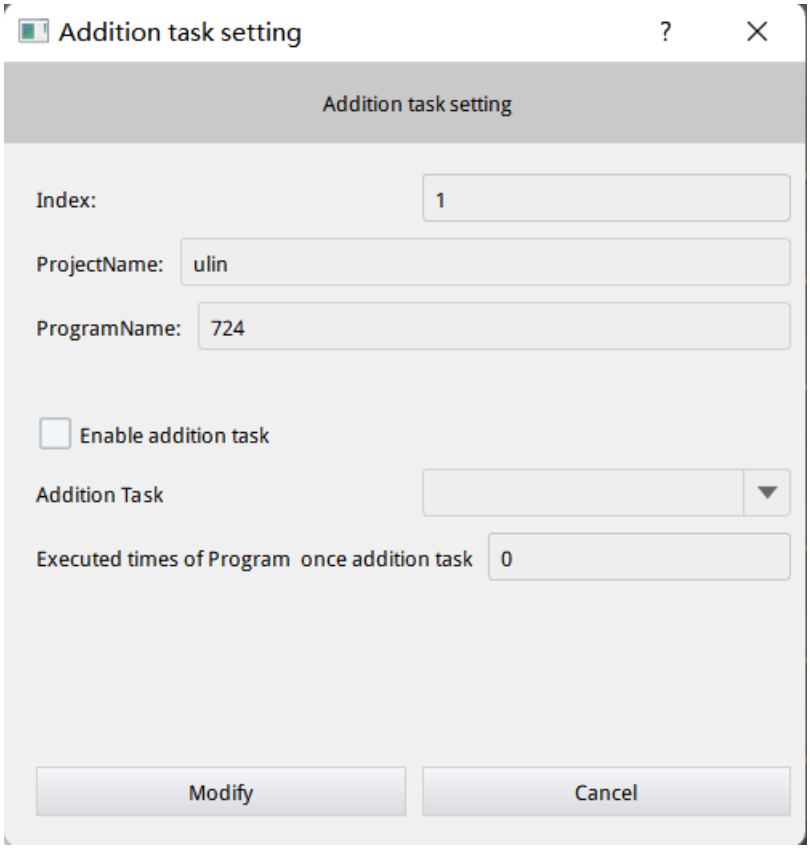
Enter the {Remote IO} interface and select the {File Settings} property page. The settings are as follows:



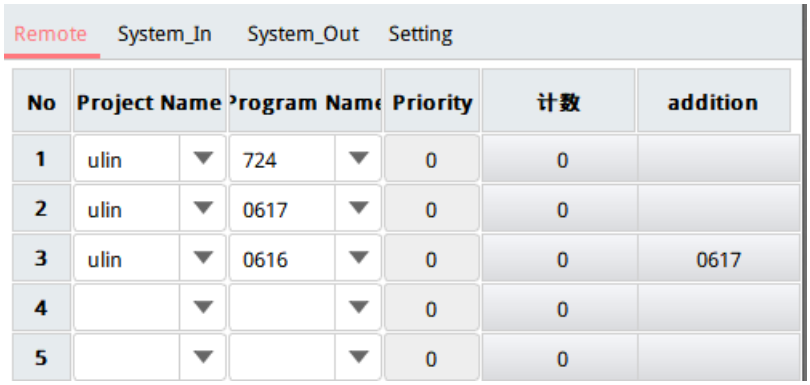
Step 3 Set up additional tasks

Click the {Additional Task} 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 **whether to enable the additional task option**, select the **additional task** to be added, **input the execution times of the file number corresponding to the additional task**, and click Modify to complete the configuration.。

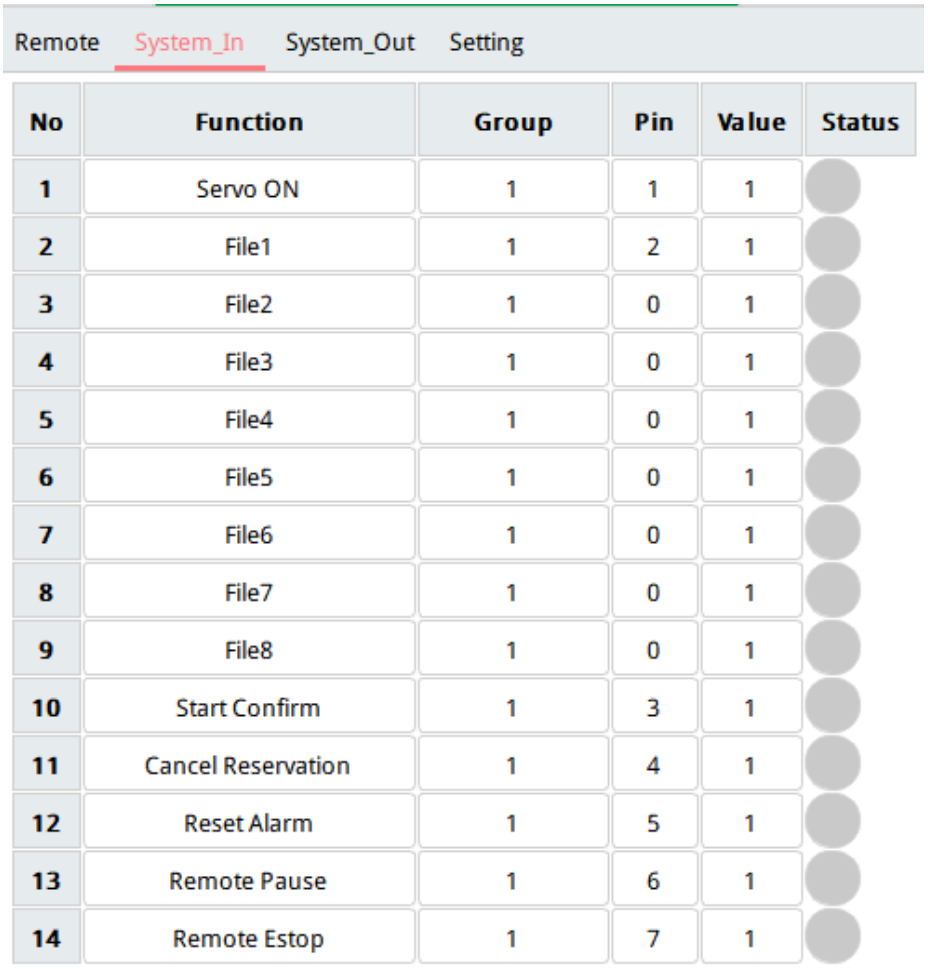


12.4.2.3 Setting Remote IO Input

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

The remote operation robot can work at 3 stations. File position 1, file position 2, file position 3 and enable signal must be set. Cancel reservation, alarm reset,

remote pause and remote emergency stop can be set as required.

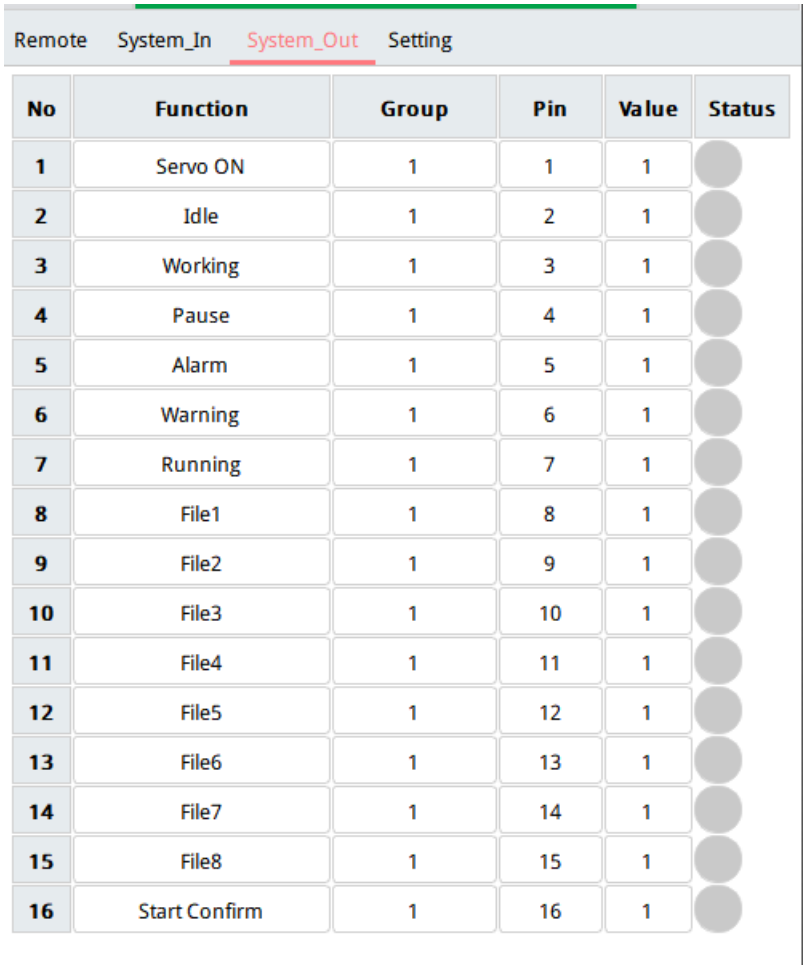


No	Function	Group	Pin	Value	Status
1	Servo ON	1	1	1	
2	File1	1	2	1	
3	File2	1	0	1	
4	File3	1	0	1	
5	File4	1	0	1	
6	File5	1	0	1	
7	File6	1	0	1	
8	File7	1	0	1	
9	File8	1	0	1	
10	Start Confirm	1	3	1	
11	Cancel Reservation	1	4	1	
12	Reset Alarm	1	5	1	
13	Remote Pause	1	6	1	
14	Remote Estop	1	7	1	

12.4.2.4 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.



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

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 Backend Tasks

13.1 Introduction to Backend Task Functions

The backend task function is a virtual PLC built into the controller, which can write logical instructions but cannot process robot motion instructions. The controller has a total of 10 built-in background programs, and background tasks can run in parallel with the main program. The two can interact with IO signals, bus data, and custom data.

The start and stop buttons on the control panel of the teaching pendant can only be used to control the start and stop of loading programs in tasks. The start, pause, and single step of background tasks must be done on the task management page.

13.2 Edit Backend Tasks

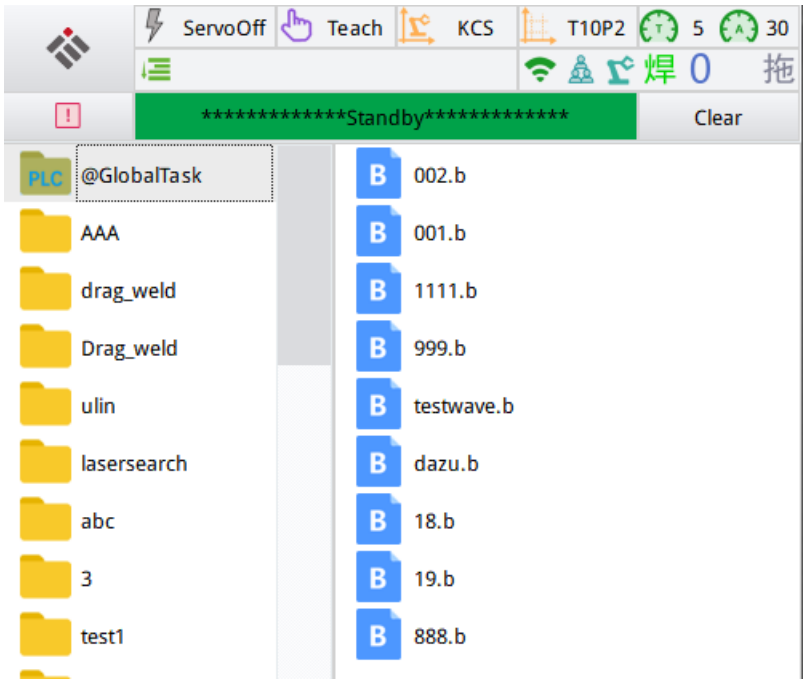


Diagram 65 Background Task File Interface

In the engineering management interface, @ GlobalTask is the backend task editing folder. The files created in this folder are backend task files and do not support adding robot motion commands. Other operations are consistent with

normal program editing.

13.3 Background Task Configuration

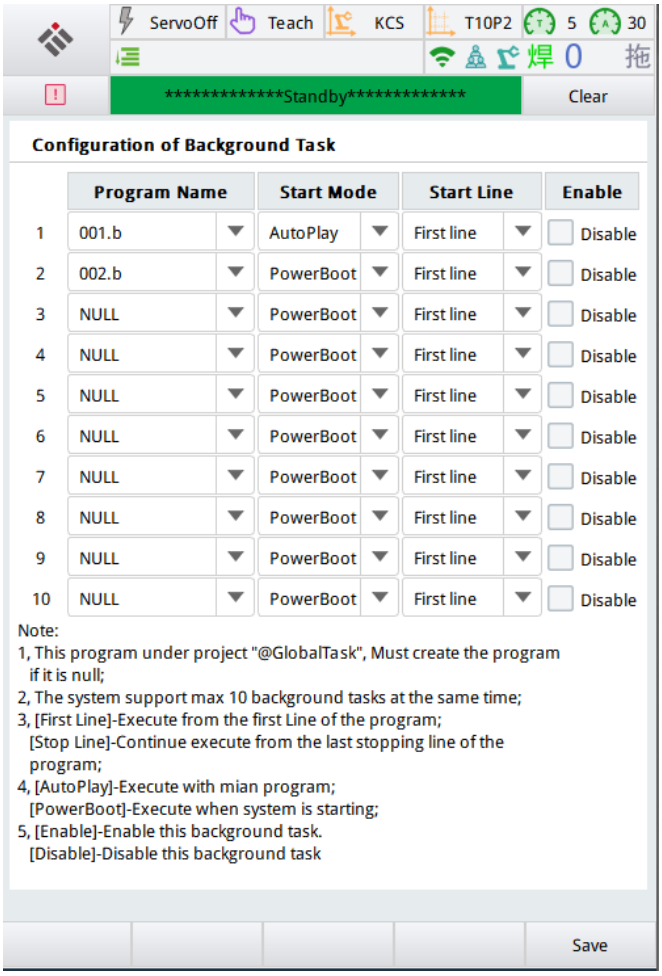


Diagram 66 Background Task Configuration Interface

1. The program for the backend task is located in the project "@ GlobalTask". If the project does not exist, please create it first.

2. Supports up to 10 background tasks running simultaneously.

3. [First line] - Start running from the first line at startup [Stop line] - Continue running from the last stop line.

4. [Handheld synchronization] - Run together with the main program [Power on startup] - Run after startup.

5. [Enable] - Use this backend task [Do not enable] - Do not use.

14 Dynamics Settings

The dynamics settings page is used to set parameters related to robot dynamics. Dynamics settings are related to robot drag teaching, collision detection, and other functions. Please ensure that the robot dynamics settings are reasonable, otherwise the above functions may not function properly or the robot may shake abnormally.

14.1 Global Parameter

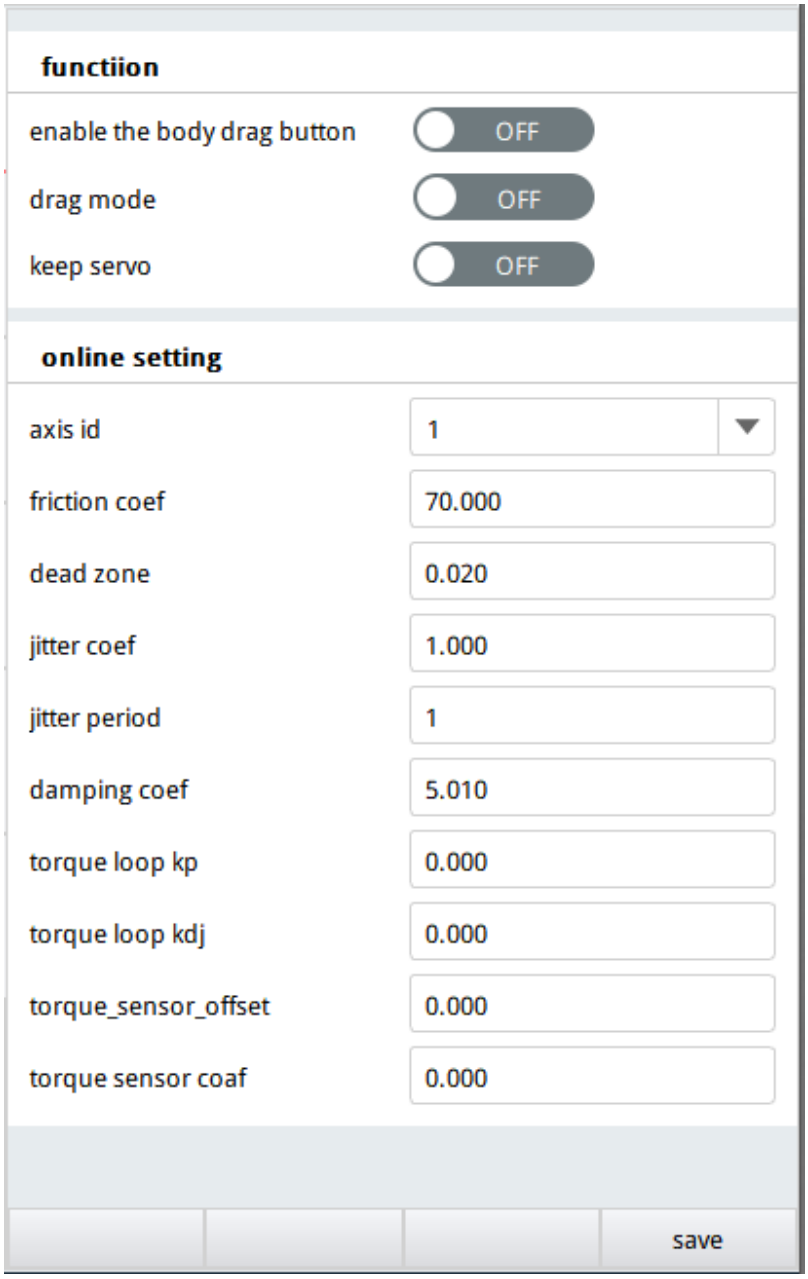


Diagram 67 Global Parameter Interface

No.	Parameter Name	Parameter Instruction
1	Input type	Different types of input feedback parameters in dynamics
2	Gravity direction (XYZ)	It determines the force situation during the robot's movement, such as inclined installation or inverted installation, and requires the correct setting of gravity parameters. Incorrect settings can lead to abnormal movement and alarms
3	Feedback smoothing window	Output mean filter length
4	Drag maximum speed	Maximum speed of movement in drag mode
5	Output maximum torque	Maximum torque for movement in drag mode
6	Whether to activate dynamics	Activate the dynamic function
7	Feedforward activation	Activate the dynamic feedforward function
8	Log activation	Enable dynamic log
9	Enable the drag button on the main body	Activate the force control sensor button
10	Drag mode	Enter drag mode
11	Maintain servo	Servo hold

14.2 Dynamic Joint Parameters

The motor parameters include robot body related data such as rated current, maximum current, rated torque, maximum torque, torque constant, reduction ratio, and zero point of force sensor. The parameters on this page directly affect the accuracy of the motion. Please ensure that the robot motor parameters are set reasonably, otherwise there may be issues with the function not functioning properly or the robot parameter recognition failing. Do not modify without the permission and assistance of the robot manufacturer.

The dynamic parameters represent the mass, center of mass, and inertia of each joint, and can be automatically obtained through ontology identification.

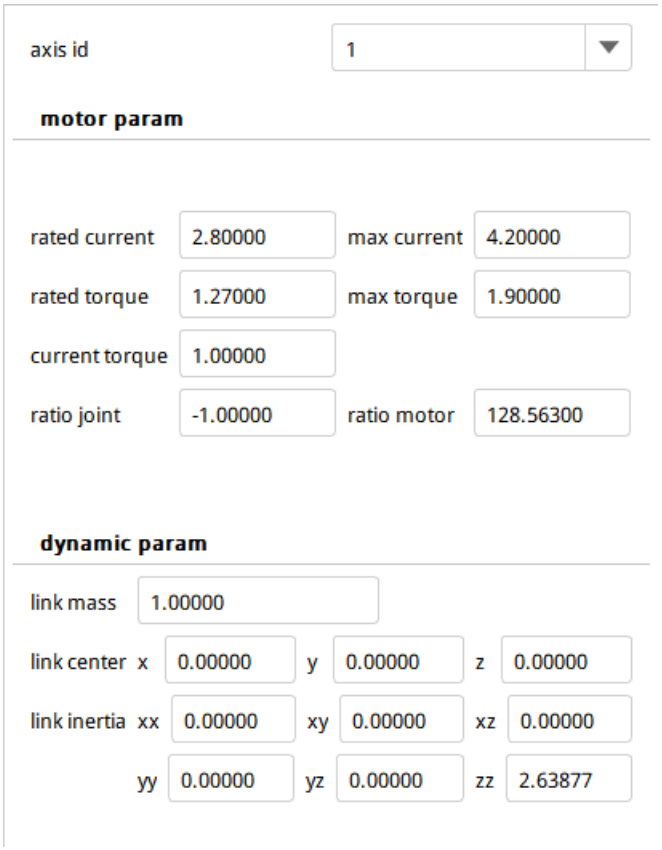


Diagram 68 Motor Parameter Interface

14.3 Identification of Dynamic Parameters

14.3.1 Identification of the Main Body

The ontology recognition operation is to collect information during the motion process by having the robot execute a series of preset trajectories. Support obtaining robot dynamic parameters from zero, including: end load of each joint, inertia parameters of each joint load, and first-order friction model parameters of each joint.

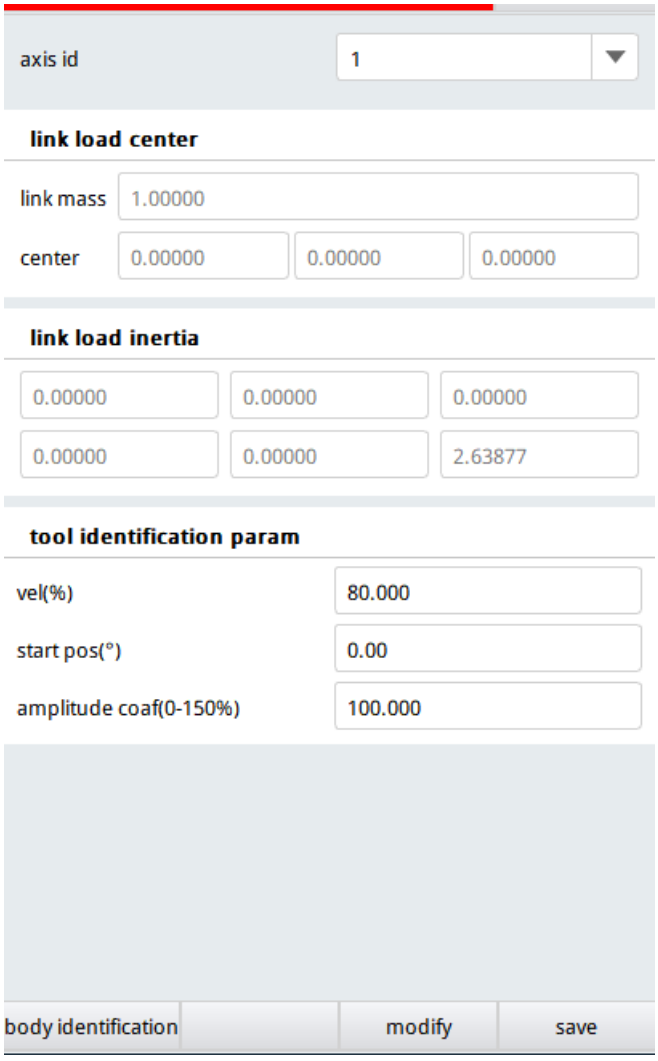


Diagram 69 Body Identification Parameter Interface

No.	Step	Instruction
1	Set frequency	Control the motion speed of the identification process

	coefficient (0-1.5)	Value range: 0.1 (low-speed mode) to 1.5 (high-speed mode) It is recommended to use a 0.3 frequency verification trajectory for the first time to avoid interference space in the motion trajectory
2	Set initial bias (degrees)	Safe obstacle avoidance starting position can be set for each axis When there is interference space, joint starting bias can be set to avoid interference space (pay attention to positive and negative directions)
3	Set amplitude coefficient (0-1.5)	Define the range of joint motion during the identification process Value range: 0.1 (conservative mode) to 1.5 (maximum travel) It is recommended to use a 1-amplitude verification trajectory for the first time, ensuring that there is no interference in the workspace
4	Click on full parameter identification	Pinch the three-stage switch and click on the full parameter recognition button to automatically run the preset trajectory and obtain the body parameters

14.3.2 Friction Identification

After the identification of the body or friction force, the parameters in the friction coefficient will be updated to the identified parameters. If you need to manually modify this parameter, please click the "Save" button after modification to save it. However, it is not recommended for users to make modifications on their own, as unauthorized modifications may result in abnormal dynamics related functions. At the same time, the drag efficiency of each joint can be adjusted by adjusting the friction coefficient.

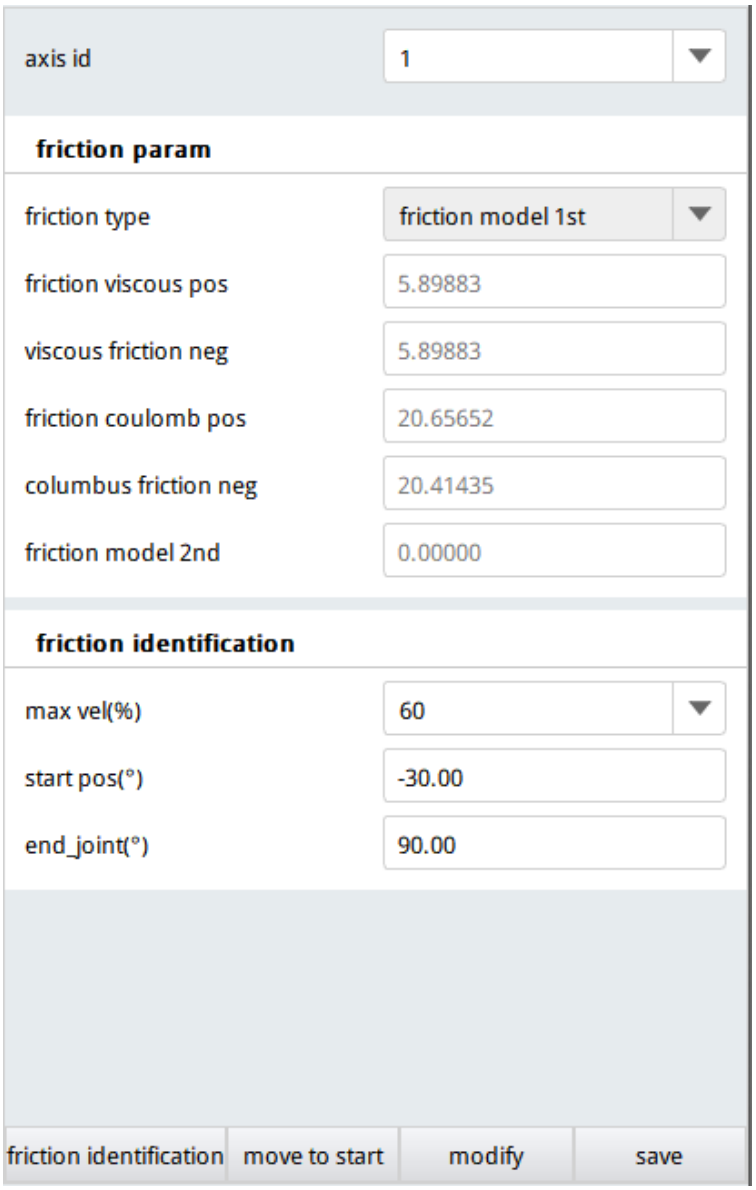


Diagram 70 Friction Identification Parameter Interface

No.	Parameter Name	Parameter Instruction
1	coefficient of friction（%）	The percentage compensation value of friction compensation=coefficient x theoretical value
2	Friction dead zone (deg/s)	Friction compensation does not take effect within the dead zone range
3	Jitter control coefficient	Compensation value for jitter control within the dead zone=coefficient x theoretical value
4	Shake control flip cycle	The period of jitter control=flipping period x planning period x2
5	Friction type	The actual effective friction model is not recommended to be modified
6	Positive viscous friction force	Coefficient of frictional force in the direction of forward motion
7	Negative viscous friction force	Coefficient of dynamic friction in the negative direction of motion
8	Forward Columbus frictional force	Coefficient of static friction force in the direction of forward motion
9	Negative Columbus Friction Force	Coefficient of static friction force in the direction of forward motion
10	Second-order viscous friction force	Acceleration coefficient of friction force

No.	Step	Description
1	Set frequency	Control the motion speed of the identification

	(HZ)	process Value range: 0.1 (low-speed mode) to 1 (high-speed mode) It is recommended to use a 0.5 frequency verification trajectory for the first time to avoid interference space in the motion trajectory
3	Set initial bias (°)	Set the starting position for safe obstacle avoidance When there is interference space, joint starting bias can be set to avoid interference space (pay attention to positive and negative directions)
4	Set amplitude (°)	Define the range of joint motion during the identification process Value range: 0 ° -360 ° It is recommended to use a 20 ° amplitude verification trajectory for the first use, ensuring that there is no interference in the workspace
5	Empty load identification	Pinch the three-stage switch and select the joint that requires friction identification. Click to run the identification program and wait for it to complete
6	Calculate and save the results	After completing the calculation, a recognition result pop-up window appears. Click confirm and save.

14.3.3 Load Identification

The load identification function can conveniently calculate the dynamic parameters of the tool

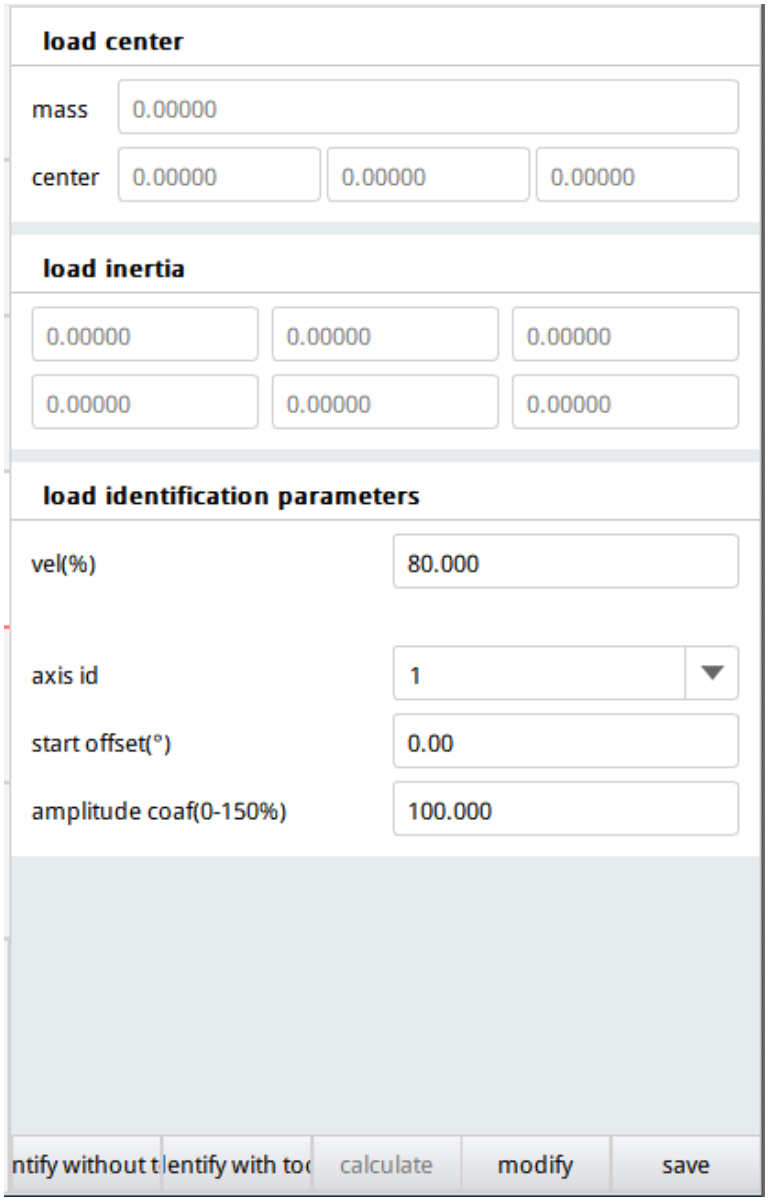


Diagram 71 Load Identification Parameter Interface

No.	Step	Description
1	Set frequency coefficient (0-1.5)	Control the motion speed of the identification process Value range: 0.1 (low-speed mode) to 1.5 (high-speed mode) It is recommended to use a 0.3 frequency verification trajectory for the first time to avoid interference space in the motion trajectory
2	Axis number	Amplitude and initial bias can be set for each joint

3	Set initial bias (degrees)	Set the starting position for safe obstacle avoidance When there is interference space, joint starting bias can be set to avoid interference space (pay attention to positive and negative directions)
4	Set amplitude coefficient (0-1.5)	Define the range of joint motion during the identification process Value range: 0.1 (conservative mode) to 1.5 (maximum travel) It is recommended to use a 1-amplitude verification trajectory for the first time, ensuring that there is no interference in the workspace
5	Empty load identification	Pinch the three-stage switch and run the identification program without tools in an unloaded state, waiting for the identification program to complete
6	Load identification	Pinch the three-stage switch and install the tool load, run the tool identification program, and wait for the identification program to complete
7	Calculate and save the results	After completing the calculation, a recognition result pop-up window appears. Click confirm and save.

- Please be sure to accurately define the dynamic parameters of the new tool, otherwise it will affect the motion performance of the robot, and in severe cases, it may even cause overload and damage to the robot.
- Support load identification for robots in both upright and inverted positions (inverted position requires adjusting the direction of gravity).
- If the following situations occur during the identification process, the identification process will be terminated, and all obtained identification data will

be lost, requiring re identification:

1. When the identification operation is in the middle, selecting other tools or switching to other interfaces;
2. Identify the triggering of emergency stop or external safety stop during program operation;
3. During the operation of the recognition program, switching from automatic mode to manual mode will result in recognition failure.

- The identification program needs to be executed in automatic mode, so all protective measures should be in an effective state. External control signals may start the robot at any time, so please switch to automatic mode after installation is completed and all personnel have retreated to a safe area.

14.4 Zero Calibration

Zero calibration of force sensor

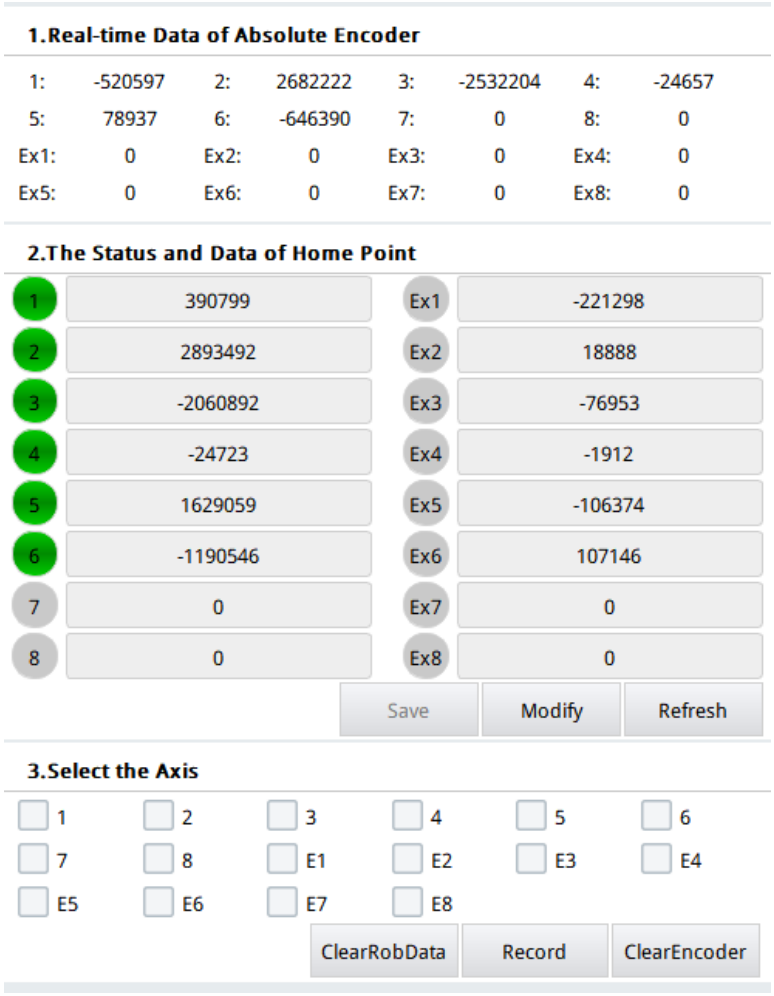


Diagram 72 Zero Calibration Interface

When the force sensor data is abnormal or needs to be re recorded as zero position, it is necessary to calibrate the force sensor data and software recording for each axis of the robot.

Mechanical zero calibration: By moving along a single axis, each axis runs to the mechanical reference zero point.

Software record calibration: After all six axes have returned to the mechanical reference zero point through single axis motion, it is necessary to enter the software's factory settings and re record the zero point position. Ensure that the zero positions of software and hardware correspond. Afterwards, every time the robot returns to zero, it will return to that position.

14.5 Document Generation

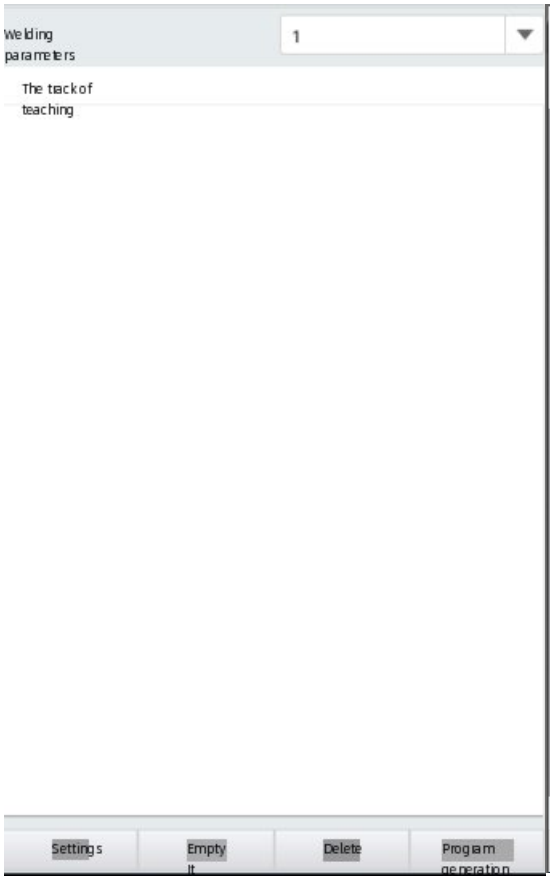


Diagram 73 Document Generation Interface

Diagram 74 File Interface

Before recording the points in drag mode, it is recommended to configure the welding parameters in advance. After configuring the parameters, drag the teaching points.

Record point information through force sensors (IO signals) in drag mode and select appropriate motion commands. After recording, click on 'Program Generation' to enter the program interface and execute the program.

14.5.1 Parameter Setting

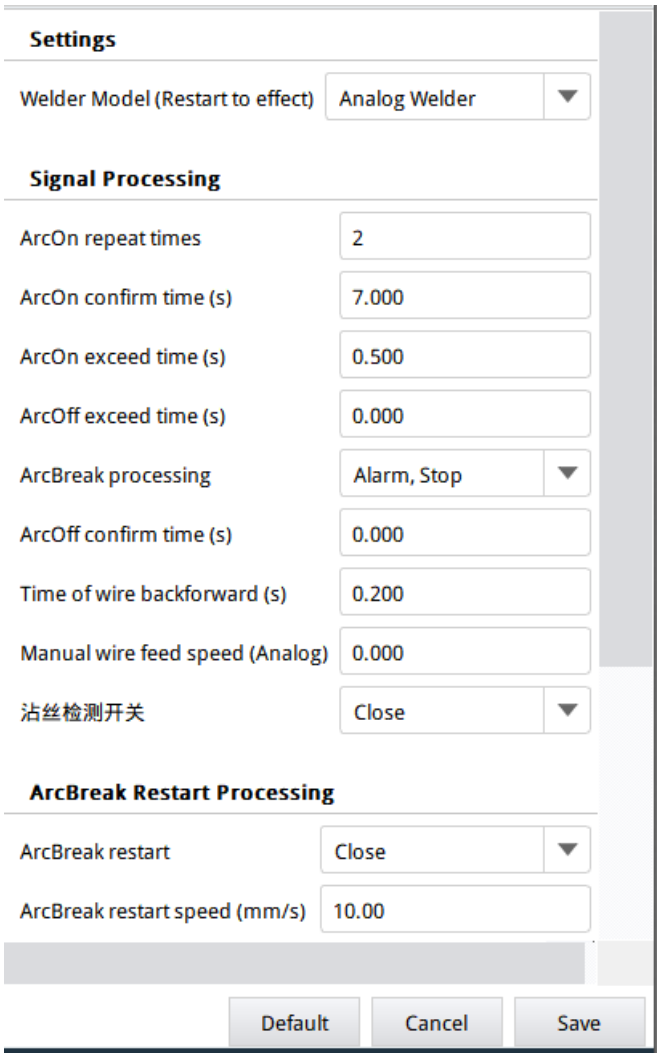


Diagram 75 Welding Parameter Configuration Interface

The welding parameter configuration includes pages with numbers 1-10 that can be configured, and different numbers can set different welding parameter contents. The configuration includes: welding file number, project name, program name, movj speed, movj transition section, movl welding speed, movl transition section, automatic insertion of Arcon, arc start delay time, arc stop delay time, automatic insertion of WaveOn, swing arc file number, and can save and call welding parameters on the main interface after modification. For specific welding instructions, please refer to the welding operation manual.

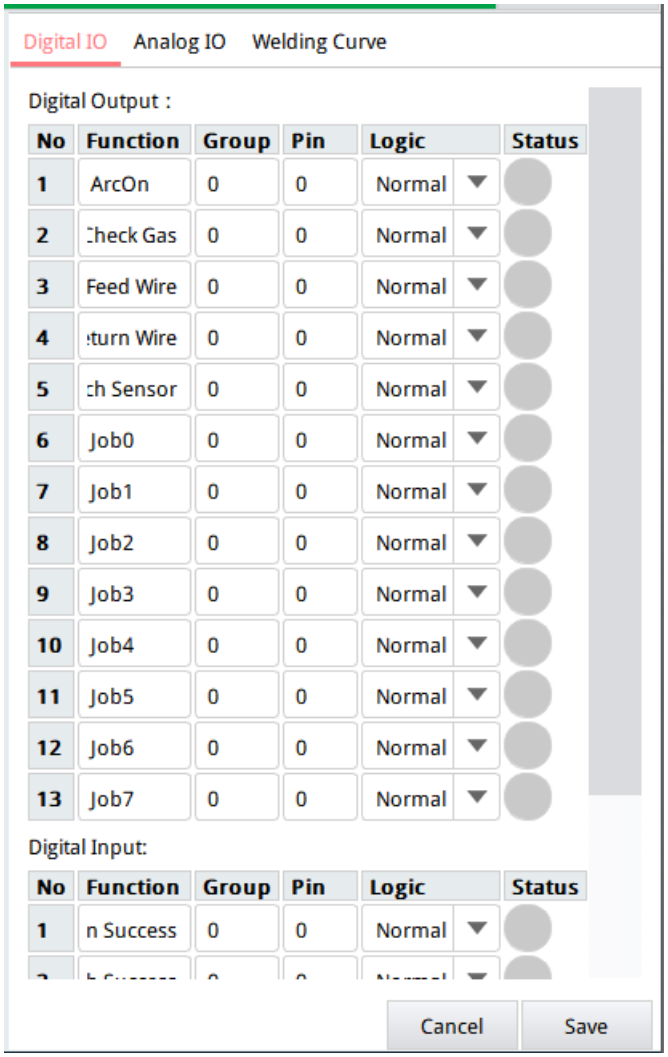


Diagram 76 Drag IO Configuration Interface

Module: Represents the address ID of the IO module, and the value of the extended IO slave address ID starts from 1.

Point: indicates the position of the module to be used, 0 indicates not to be used, and point values start from 1.

Valid value: Normal 1 is valid and can be reversed.

No.	Parameter Name	Valid Value	Description
1	Insert_movj	0-Input	Insert movj command
2	Insert_movl	0-Input	Insert movl command
3	Insert_movc	0-Input	Insert movc command

4	delete_last	0-Input	Delete the last instruction
5	arc_on	0-Input	Insert welding start command
6	arc_off	0-Input	Insert welding closure command
7	wave_on	0-Input	Insert the instruction to start the swing arc
8	wave_off	0-Input	Insert the instruction to close the swing arc
9	enable_drag	0-Input	Drag the button using the main body
10	led_green	1-Output	Turn on the green LED indicator light
11	led_red	1-Output	Turn on the red LED indicator light
12	led_blue	1-Output	Turn on the blue LED indicator light

15 Other Functions

15.1 Zero Calibration

注意
1. The teaching and playback operations cannot be performed without the origin position calibration. For systems using multiple robots, each robot must be calibrated for the origin position. 2. When there is a coupling relationship between the axis axes, for example, there is a coupling relationship between the fifth axis and the sixth axis of a common robot, the zero-point data recorded by the sixth axis can be valid only when the fifth axis is at the zero-point position, otherwise, the zero point data recorded by the sixth axis is invalid. it is necessary to record the zero-position data of the sixth axis in a state where the fifth axis is at the zero-position. If there is no coupling relationship, each axis can be calibrated independently, and their respective zero positions will not affect the zero positions of other axis. 3. When all used axes (body axis and auxiliary extension axis) have completed zero calibration, the "All" indicator on the zero-calibration interface turns green, indicating that the robot has completed the calibration of zero data, and the robot can move in Cartesian space.

After clearing the encoder zero drift alarm, the mechanical zero position calibration and software record calibration are 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 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, every time the robot returns to zero, it will return to that position.

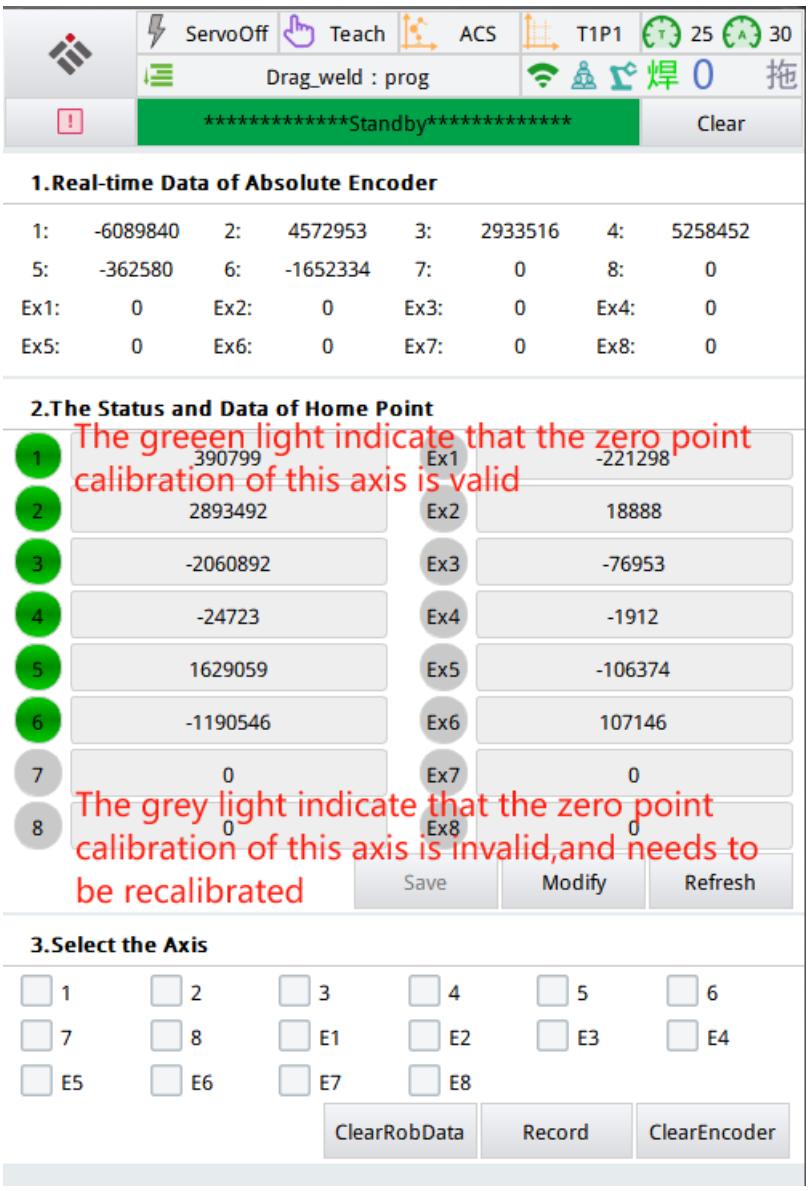
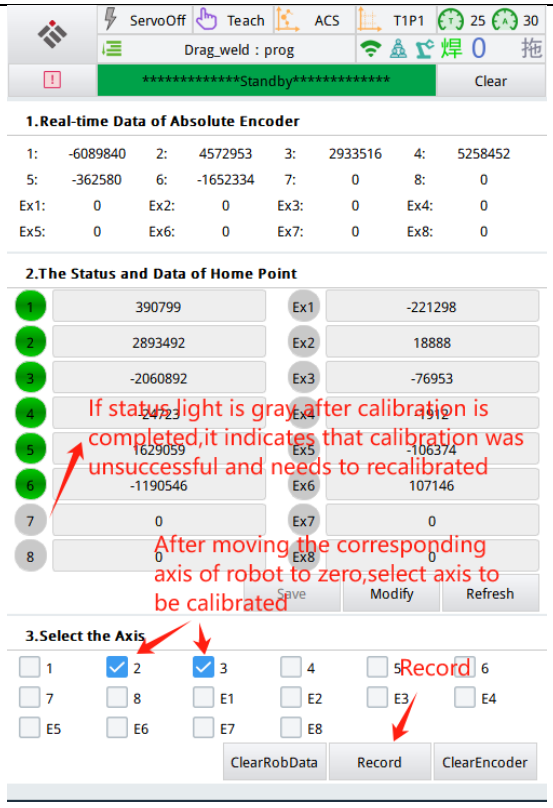
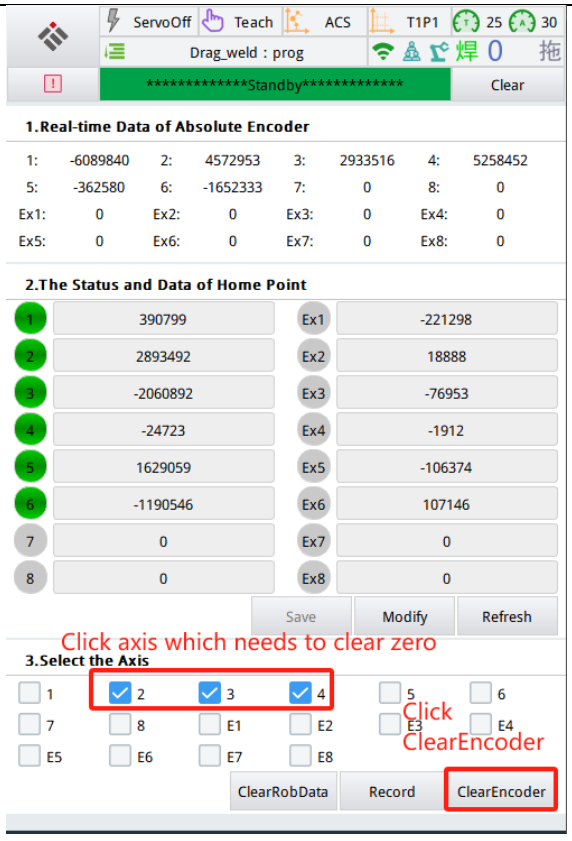
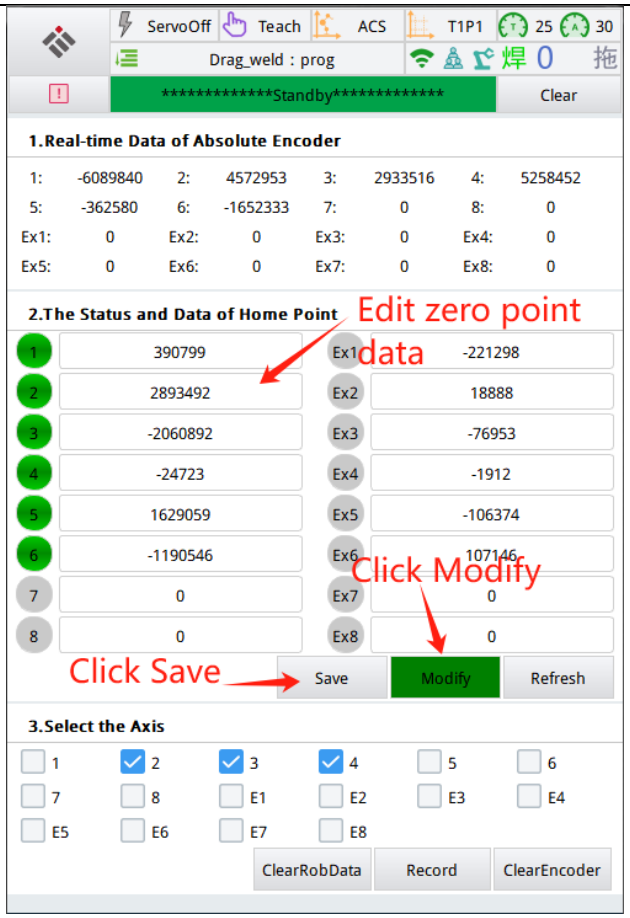


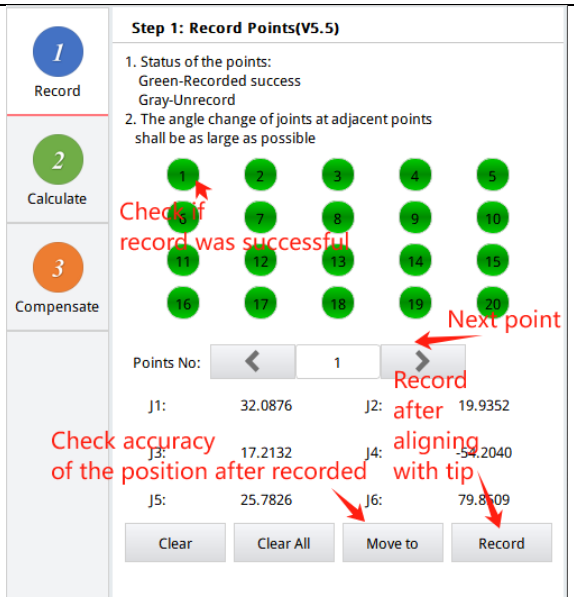
Diagram 77 Zero point calibration configuration interface

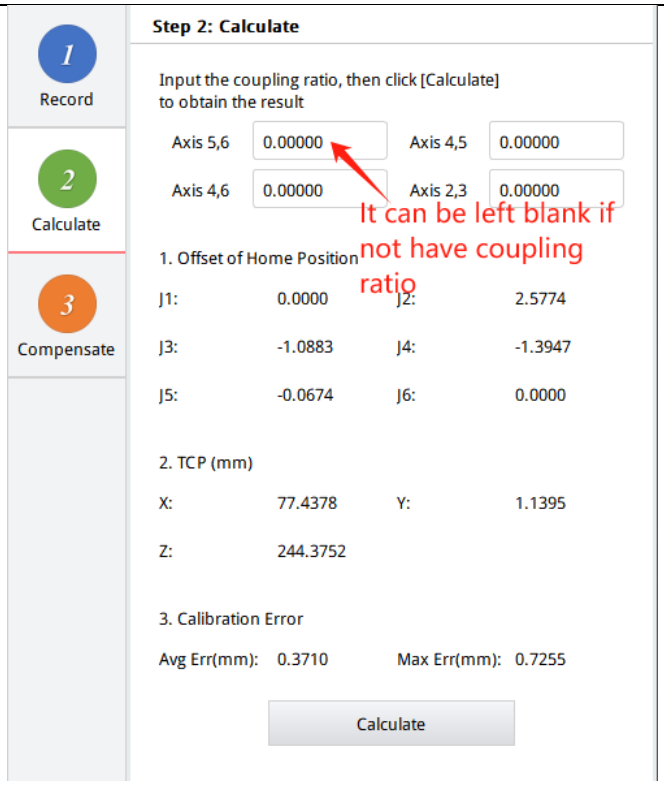
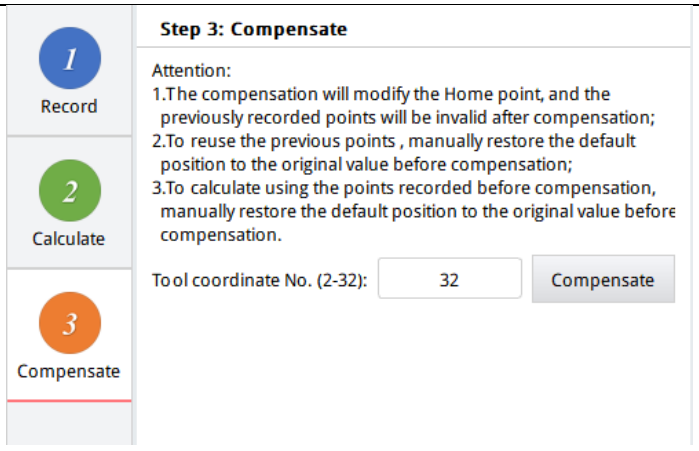
Item	Procedure
Record zero	
Absolute encoder reset (the selected axis of the zero data to 0)	

Manually modify zero point (Not recommended)	

15.2 20-Point Method

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

Item	Procedure
Record the location	

	Use robot tools to align the tip of the robot with different postures and record the position points. The joint angles of each position point should change as much as possible
Calculation results	
Select compensation location (Manual Compensation)	

15.3 Exception Handling

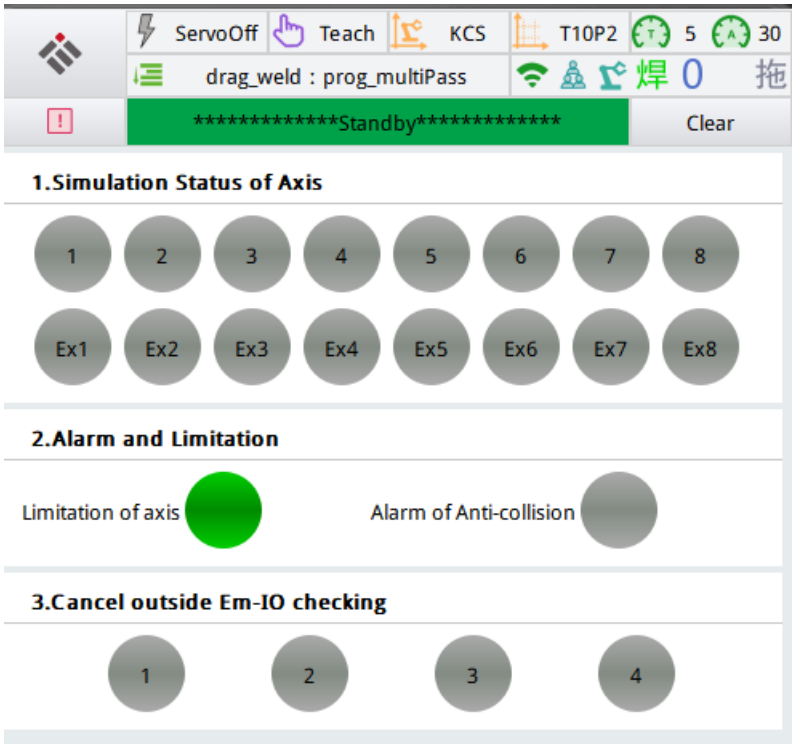


Diagram 78 Abnormal configuration interface

No.	Parameter Name	Parameter Instruction
1	Current axis simulation status	Set the corresponding axis of the robot to simulation state, green light indicates effectiveness, gray indicates normal.
2	Axis motion limitation	Limit the movement of the robot axis, with a green light indicating effectiveness and a gray light indicating normal operation.
3	Cancel collision alarm	Cancel collision alarm, green light indicates effectiveness, gray indicates normal.
4	Cancel external IO emergency stop check	When the external IO emergency stop of the robot is triggered, the external IO emergency stop of the robot is blocked for 10 seconds. If the external IO emergency stop is not restored, the alarm will continue, with a green light indicating effectiveness and a gray light indicating

		normal operation.

15.4 Authority Management

15.4.1 User Switching

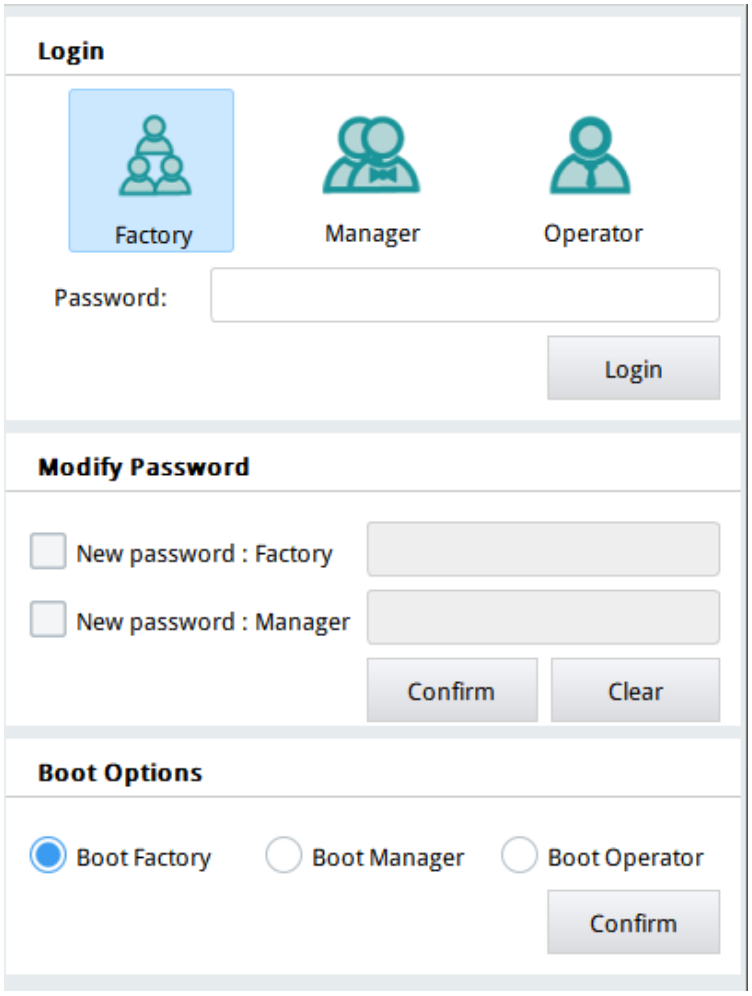


Diagram 79 User Switching Interface

15.4.2 Hardware Testing

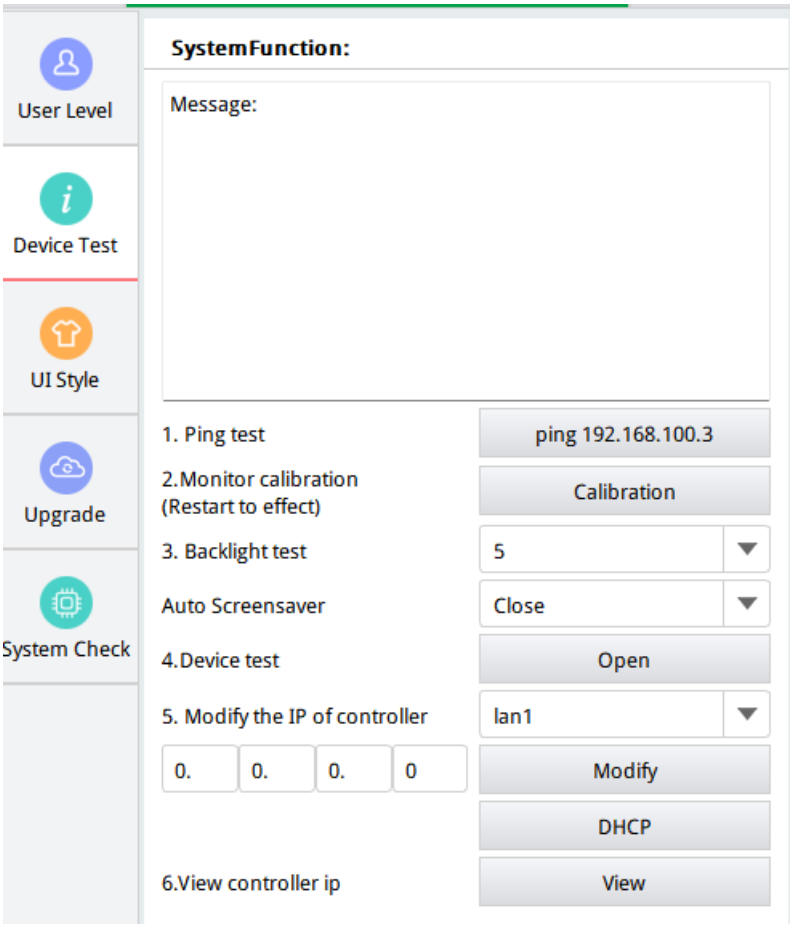


Diagram 80 System Version Interface

No.	Parameter Name	Parameter Instruction
1	Ping test	Check if the communication between the teaching pendant and the controller is normal.
2	Touch calibration	Hand hold the touch screen for calibration,and the calibration program automatically restart after execution
3	Backlight test	Adjust the brightness of the handheld LCD backlight Automatic screensaver (on/off)
4	Handheld hardware test	{Open hardware test}
5	Modify controller IP	Modify network port IP of lan1,lan2(lan3)
6	View controller IP	View network port IP of lan1,lan2(lan3)

15.4.3 Interface Style

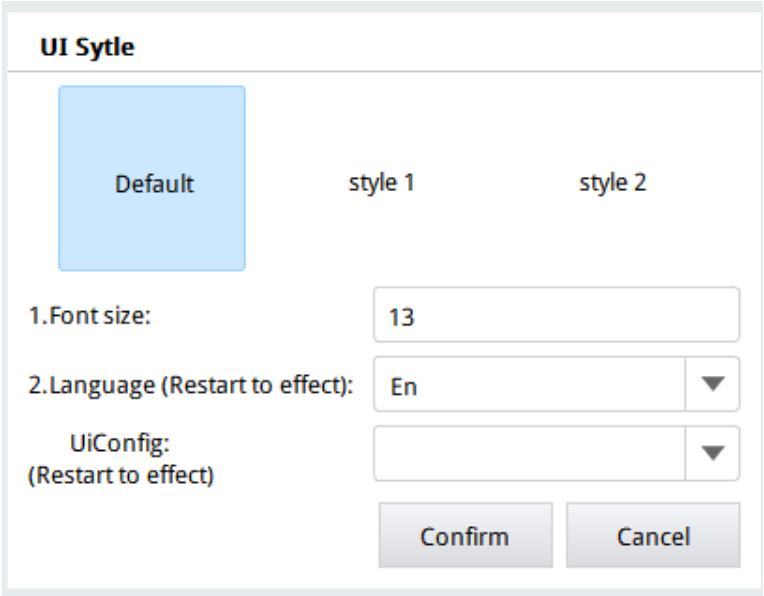
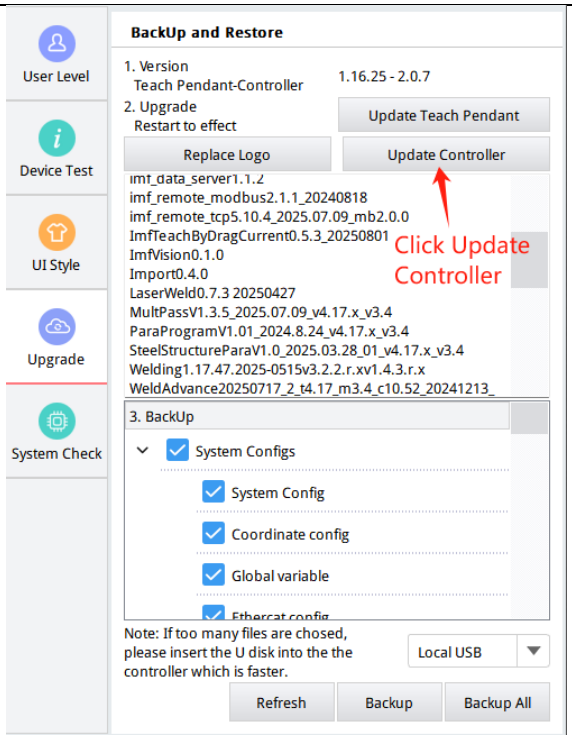
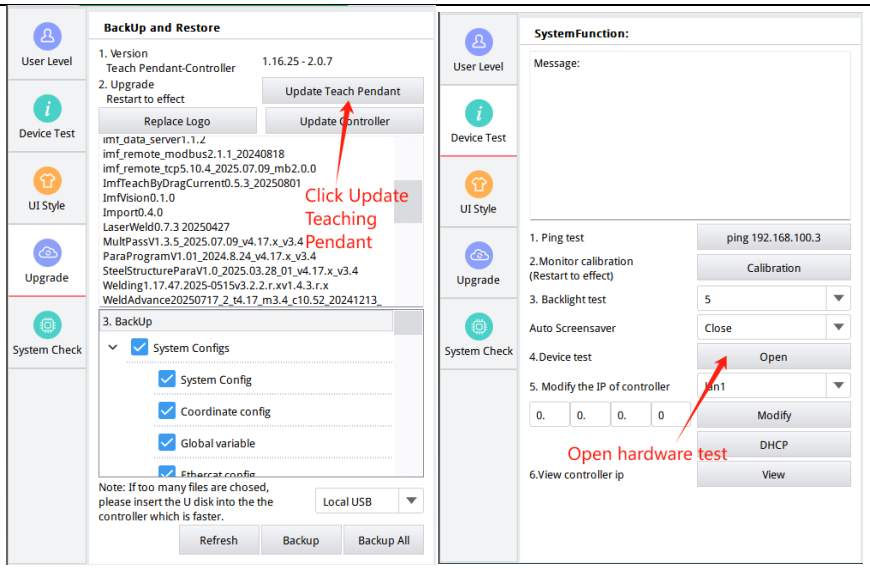


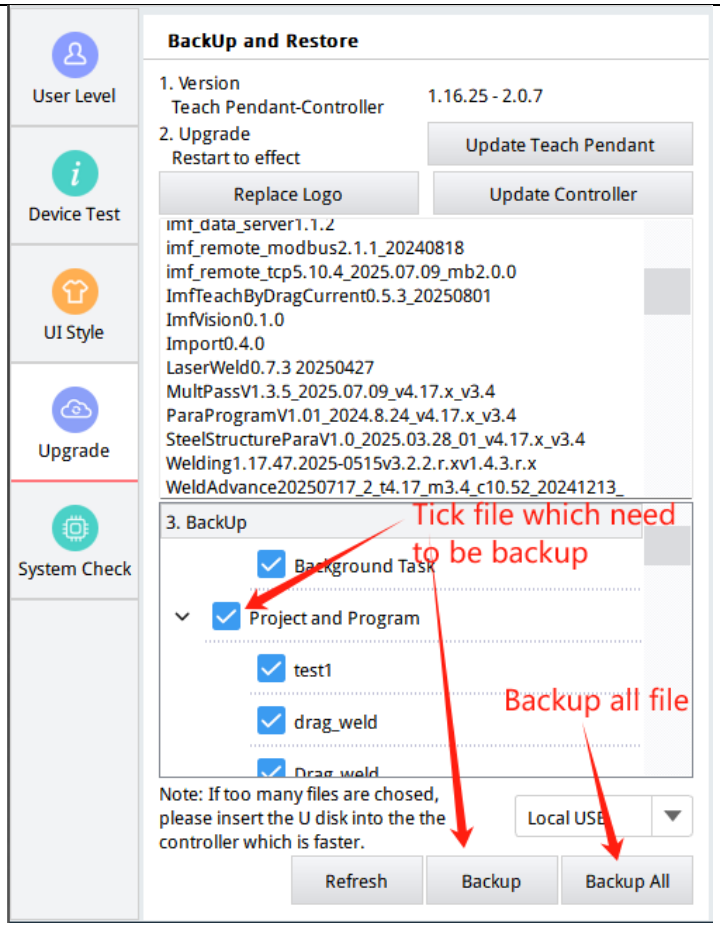
Diagram 81 Interface Style Setting Interface

Set the interface font, font size and display language.

15.4.4 Upgrade and Backup

Item	Procedure
System upgrade/import teaching files	 To upgrade the system, you need to click on "Controller

	Update" in the "Upgrade and Backup" interface, insert the USB drive, and select the corresponding upgrade file. (The operation of importing teaching files is the same as above)
Interface upgrade	 To upgrade the UI of the handheld box interface, click on "Handheld Box Update" in the "Upgrade and Backup" interface, insert the USB drive and select the corresponding upgrade file. After the upgrade is complete, switch to the "Hardware Testing" interface, click on "Open Hardware Testing", and then select "Return to Robot Interface";

Backup files	 1. Click 'Backup All' to backup all files; 2. Check the files that need to be backed up and click "Backup Selected Files" to backup the selected files

16 Instruction System

16.1 List of Orders

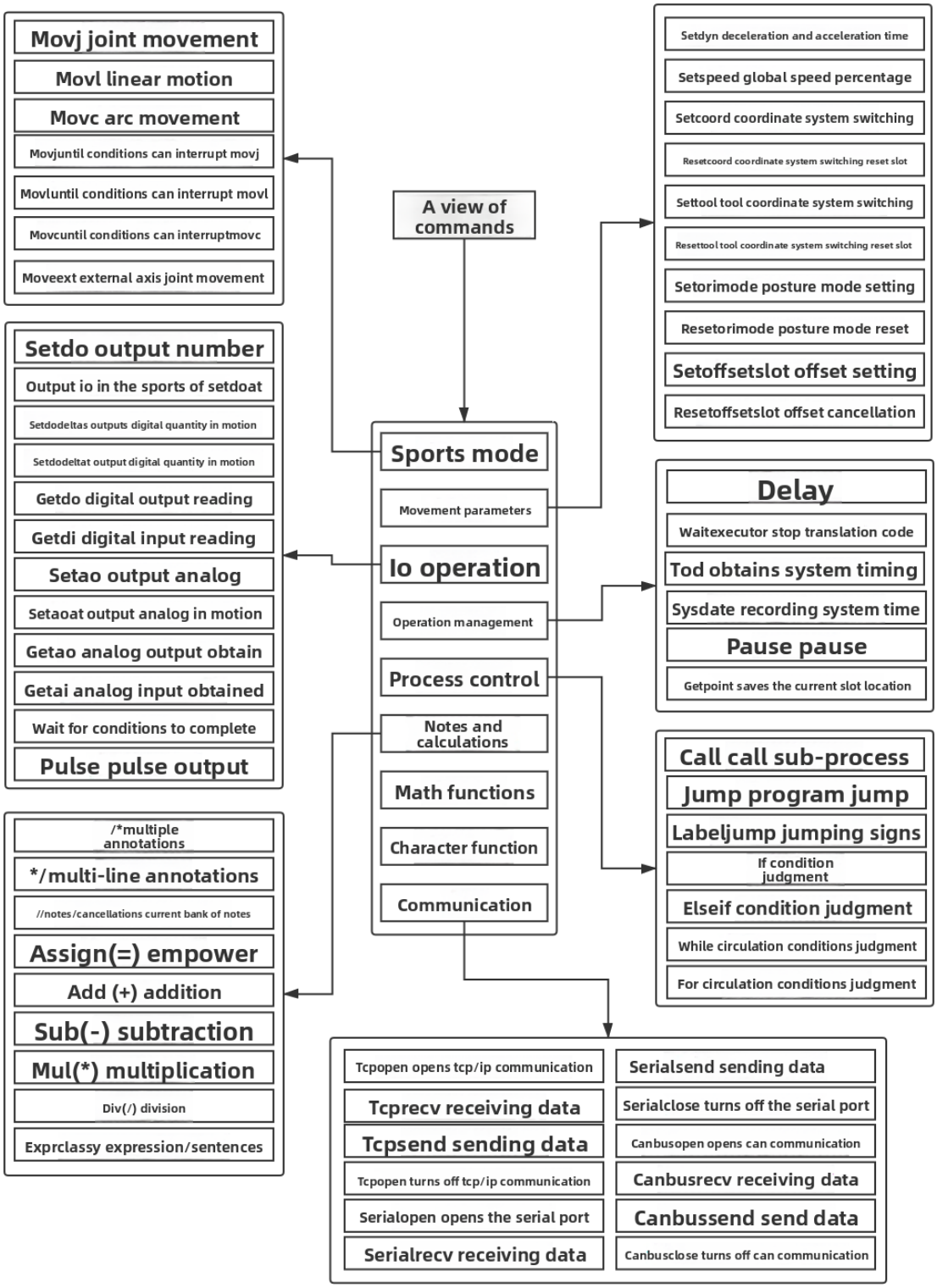


Diagram 82 List of Commands

16.2 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*

16.3 Motion Instructions

16.3.1 Processing mechanism of Motion commands

16.3.1.1 Processing Mechanism of Motion Instruction

The robot does not execute a line of motion commands, the robot moves a position. In order to improve the fluency of motion, the teaching program parsing module of the control system will compile all instructions in advance and preprocess them, and then send several motion instructions to the motion stack of the robot according to the logic of the program itself. At this time, the robot may not even reach the first position. It is to be understood that the execution of the application is independent of the robot motion. At the same time, the system provides a pause parsing instruction to facilitate the user to interrupt the pre-compilation mode of the program according to the actual situation.

16.3.1.2 Motion Command Trajectory Transition

At the junction of two motion 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 to continue the next trajectory. this unnecessarily increases the cycle time. By providing the transition section, the movement time can be greatly reduced. When this Blend value is set to Not Used, the robot is strictly reached 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.

16.3.2 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.

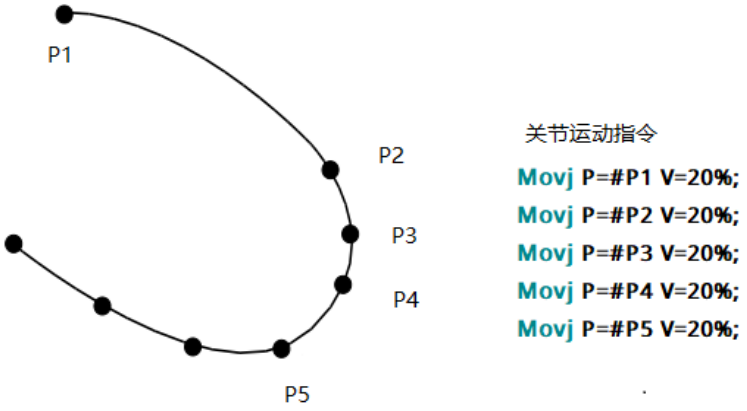


Diagram 83 Movj Trajectory and Command

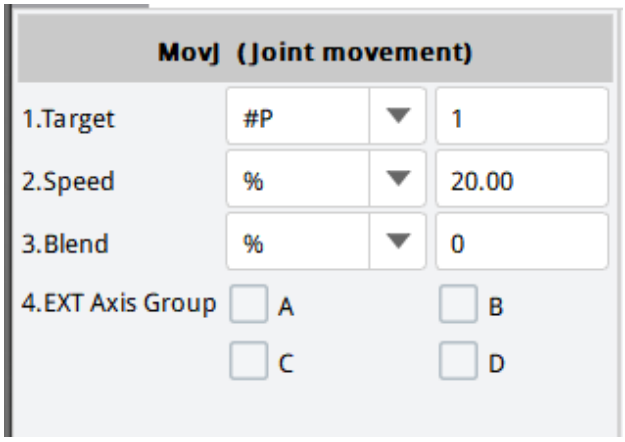


Diagram 84 Movj Parameter Settings

Instruction format	Movj P=*Point* V=*Speed*% [Bl= *Blend*%] [Ext= [*A-D*]]	
Eaxmples	Movj P=#P1 V=20% Bl=20% Ext=A;	
Parameter Description	Target point	*Point*:can be axis space or cartesian space point
	Speed	*Speed*: Axis space end synthesis speed, Speed =%* Axis space end synthesis maximum speed. Values range from 1 to 100, with a default of 20.
	Transition	*Blend*: axis space blend radius, blend radius =%* total

	section	axis space trajectory length.
	Axle set	Ext External axis group number of synchronous motion 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		

16.3.3 Movs Linear Motion

Movs Motion Trajectories and Examples

The Movs Motion performs B-spline motion in joint space.

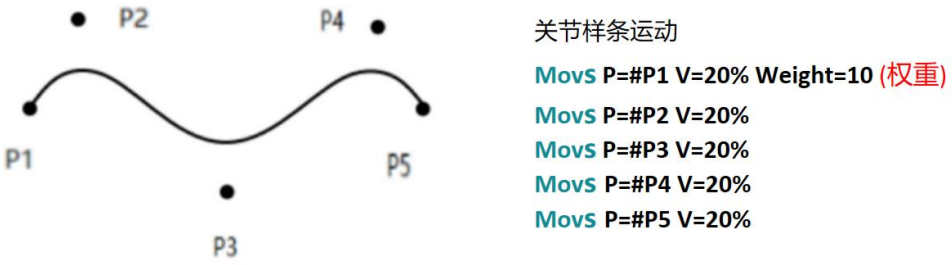


Diagram 85 Movs trajectory and command

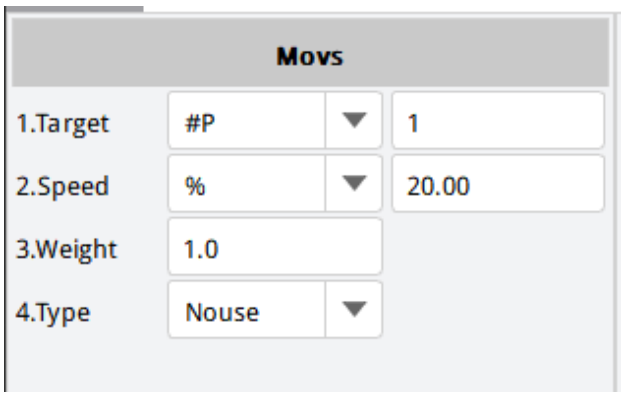


Diagram 86 Movs Parameter Settings

Instruction format	Movs P=*Point* V=*Speed*% [Weight= [*1-10*]][Type=*end*,Bl=*Blend*%]	
Examples	Movs P=#P1 V=20% Weight=3 Type=end Bl=20%;	
Parameter Description	Target 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.
	Weight	*Weight*: Point accuracy, the higher the value, the higher the point accuracy, ranging from 1 to 10, with a default value of 1.
	Type	End Cancel the spline motion control point and change it to reach the target point
	Transition section	*Blend*: Transition radius of joint space, transition radius=% * total length of joint space trajectory.
Precautions	After selecting 'end' as the Type, Movs Sports will be the same as Movj Sports	

16.3.4 Movl Linear Motion

Movl Motion Trajectories and Examples

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

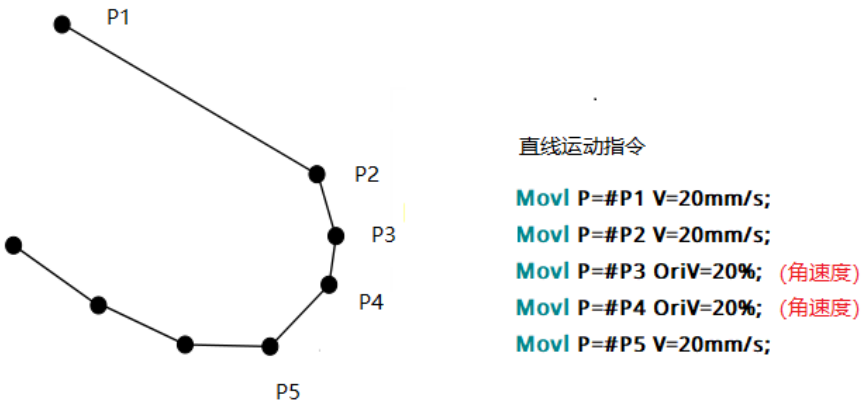


Diagram 87 Movl Trajectory and Command

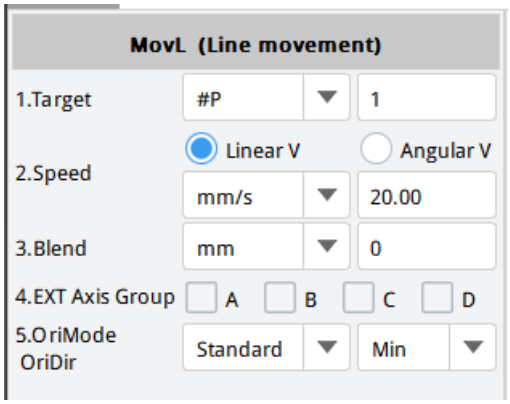


Diagram 88 Movl Parameter Settings

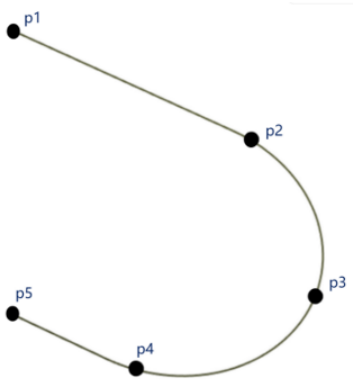
Instruction format	Movl P=*Point* (V=*Speed* mm/s [Bl=*Blend* mm]) |(OriV = *OriSpeed* % [OriBl =% *OriBlend*]) [Ext= [*A-D*]]	
Eaxmples	Movl P=#P1 V=20mm/s Bl=10mm Ext=A; Movl P=#P1 OriV=20% OriBl=10% Ext=A;	
Parameter Description	Target point	*Point*:Can be an axis space or Cartesian space point.
	Speed	*Speed*: Cartesian-space end path resultant speed, in:mm/s *OriSpeed*: Cartesian Space Posture End Resultant Velocity Percentage, Velocity =%* Posture End Resultant Maximum Velocity.
	Transition	*Blend*: Cartesian space blend radius, units:mm;

	section	*OriBlend*: Cartesian space end pose blend percentage, blend angle =%* pose space trajectory total angle.
	Axle set	The group number of the external shaft that needs synchronous movement, multiple choices are allowed. Whether the synchronous motion is position following or not depends on whether the corresponding PCS of the position point record is moving 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.	

16.3.5 Movp Circular Motion

Movc Motion Trajectories and Examples

Cartesian spatial spline motion, including position weights, attitude tracking, and planning.



笛卡尔样条运动

Movp P=#P1 V=100mm/s PosWeight=10 OriWeight=10(权重)
Movp P=#P2 V=100mm/s
Movp P=#P3 V=100mm/s
Movp P=#P4 V=100mm/s
Movp P=#P5 V=100mm/s

Diagram 89 Movp Trajectory and Command

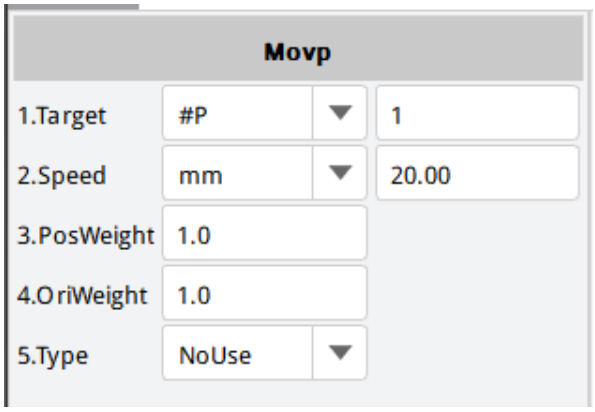


Diagram 90 Movp Parameter Settings

Instruction format	Movp P=*Point* (V=*Speed* mm/s [Bl=*Blend* mm]) |(PosWeight=*PosBlend* % [OriWeight=% *OriBlend*])	
Example	Movp P=#P1 V=20mm/s PosWeight=2 OriWeight=2 Movp P=#P1 V=20mm/s PosWeight=2 Type=End Bl=10mm	
Parameter Description	Target point	*Point*: It can be joint space or Cartesian space points.
	Speed	*Speed*: Cartesian space final path synthesis velocity, unit: mm/s
	Transition section	*Blend*: Transition radius of Cartesian space, unit: mm;
	Position weight	*PosWeight*: The precision of point positioning at the end of Cartesian space is higher, and the higher the value, the higher the precision of point positioning. The values range from 1 to 10, with a default value of 1.
	Attitude weight	*OriWeight*: Cartesian space end pose to point accuracy, the higher the value, the higher the point accuracy, ranging from 1 to 10, with a default value of 1.
	Type	End cancel spline motion control points and switch to reaching the target point

	Transition section	*Blend*: Transition radius of Cartesian space, unit: mm;
Precautions	After selecting 'end' as the Type, Movp Sports will be the same as Movl Sports	

16.3.6 Movc Circular Motion

16.3.6.1 Movc Circular Motion

Movc motion trajectory and examples

Cartesian space circular motion, including pose following. In circular motion, the center point of the tool moves along an arc defined by 3 points, and circular interpolation is used between the starting point, midpoint, and endpoint.

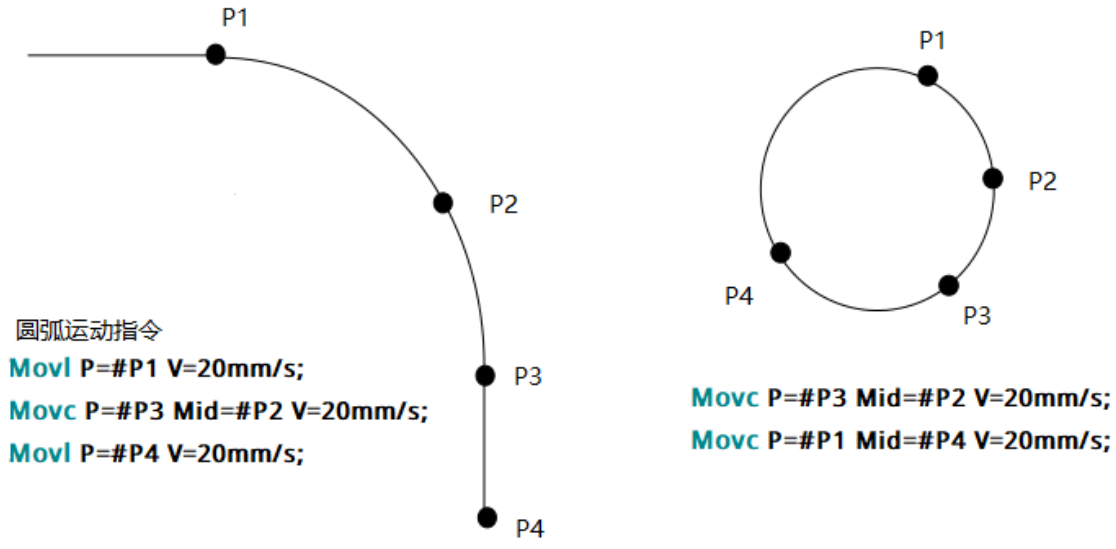


Diagram 91 Movc Tajectory and Command

The arc command requires two 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, which are implemented by 2 Movc instructions.

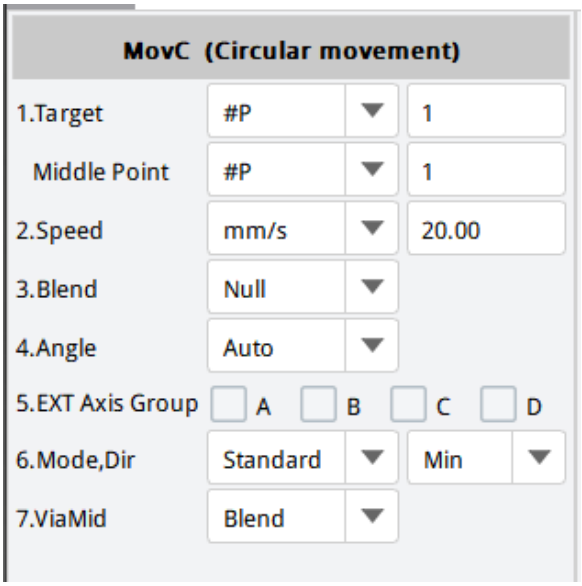


Diagram 92 Movc Parameter Settings

Instruction format	Movc P=*EndPoint* Mid=*MidPoint* V=*Speed* mm/s [Bl = *Blend* mm] [Deg = *Degree*] [Ext = [*A-D*]]	
Examples	Movc P=#P1 Mid=#P2 V=20mm/s Bl=10mm Deg=108 Ext=A;	
Parameter description	Target point Middle point	*EndPoint*: The endpoint of an arc can be a joint space or a Cartesian space point; *MidPoint*: The midpoint of an arc can be a joint space or a Cartesian space point.
	Speed	*Speed*: Cartesian-space end path resultant speed, in:mm/s。
	Transition section	*Blend*: Cartesian space blend radius, units:mm。
	Arc angle	*Degree*: the arc angle of the actual motion, such as a full circle, default It is the arc angle formed by three points. Negative value is supported. Optional.
	Axle set	Ext: external axis group number to be synchronized, multiple choices are allowed, same as MovL.

	OriMode	Standard,PathFrame,JointSpace
	OirDir	Attitude planning, default Min, Max as required
Precautions	Arc trajectory, pose planning, path following situation is unreasonable, not supported.	

16.3.6.2 Movc2 Movc3 Combination Arc Motion

Movc2 Movc3 trajectory and command

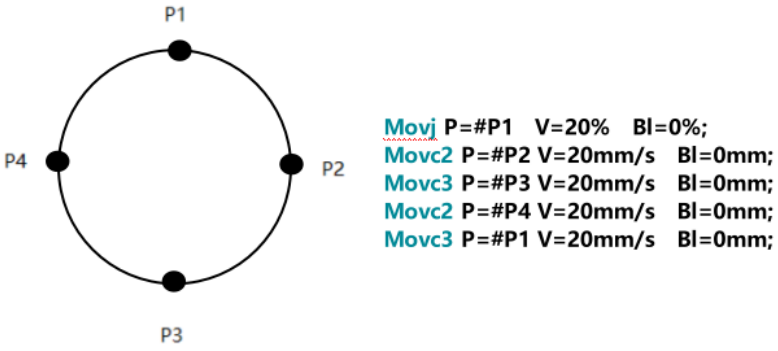


Diagram 93 Movc2 Movc3 Trajectory and Command

Movc2 Movc3The instruction serves as the two position points of the arc, representing the midpoint and endpoint of the arc, respectively

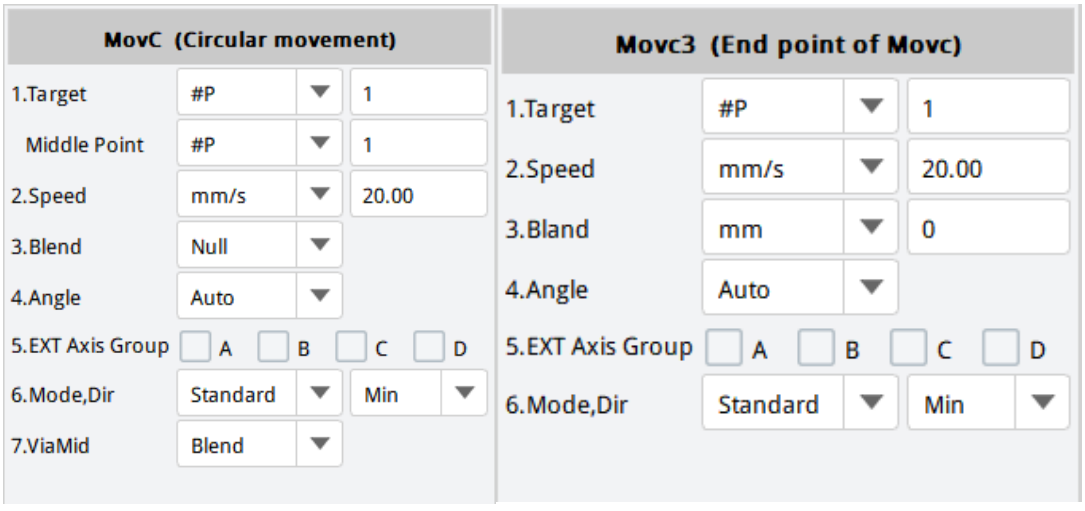


Diagram 94 Movc2 Movc3 Parameter Settings

Instruction format	Movc2 P=*Point* V=*Speed* mm/s [Bl = *Blend* mm] [Deg = *Degree*] [Ext = [*A-D*]]

	Movc3 P=*Point* V=*Speed* mm/s [Bl = *Blend* mm] [Deg = *Degree*] [Ext = [*A-D*]]	
Examples	Movc2 P=#P1 V=20mm/s Bl=10mm Deg=108 Ext=A; Movc3 P=#P2 V=20mm/s Bl=10mm Deg=108 Ext=A;	
Parameter description	Target point	*Point*: The target point can be a joint space or a Cartesian space point;
	Speed	*Speed*: Cartesian space final path synthesis velocity, unit: mm/s.
	Transition section	*Blend*: Transition radius in Cartesian space, unit: mm.
	Arc angle	*Degree*: The actual arc angle of motion, such as a full circle, defaults to the arc angle formed by three points, supports negative numbers, and is optional.
	Axle set	Ext: The external axis group number that requires synchronous motion can be selected multiple times, the same as MovL.
	OriMode	Standard,PathFrame,JointSpace
	OirDir	Attitude planning, default Min, given Max according to requirements
Precaution	The situation of arc trajectory, attitude planning, and path following is unreasonable and not supported. The posture of the starting point of the arc rotates around the normal and cannot move to the endpoint. The *Speed* and *Blend* of Movc2 and Movc3 only take effect when modified in Movc2, and modifying parameters in Movc3 is meaningless.	

16.3.7 MoveExt Outer Axis Set Articulation

The MoveExt external axis set is articulated, and the robot chooses whether to follow

the motion or not according to the set parameters.

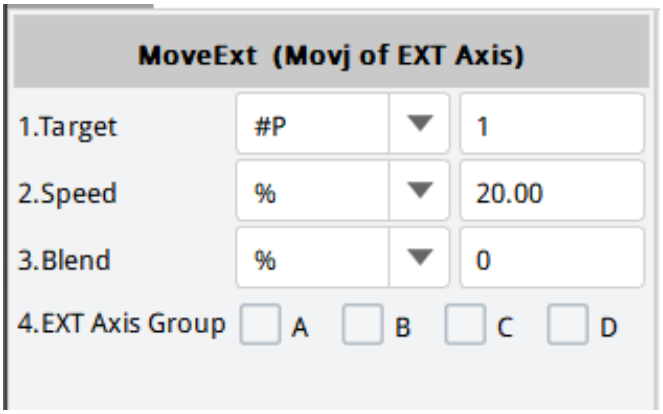


Diagram 95 MoveExt Parameter Settings

Instruction format	MoveExt P=*Point* V=*Speed*% [Bl = *Blend* %] [Ext = [*A-D*]]	
Examples	MoveExt P=#P1 V=20% Ext=A	
Parameter description	Target point	*Point*: Target point, using only the relevant part of the external axis.
	Speed	*Speed*: Axis space end resultant velocity percentage, velocity =%* Axis space end resultant maximum velocity.
	Transition section	*Blend*: Percentage of axis space blend radius, blend radius =%* total axis space trajectory length.
	Axle set	Ext: External axis group number 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 with the external axis.
Precaution	Supports up to 4 external axis groups moving simultaneously	

16.3.8 MovjUntil Conditional Interrupt Movj

Axis 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.

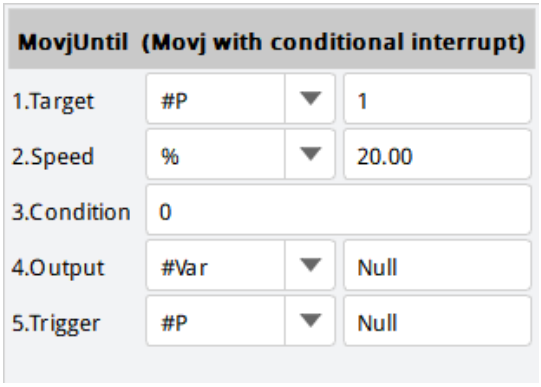


Diagram 96 MovjUntil Parameter Settings

Instruction format	MovjUntil P=*Point* V=*Speed* % Cond=*Condition?* =>ReachEnd	
Examples	MovjUntil P=#P1 V=20% Cond=(#Var1<<5) =>#Var10;	
Parameter description	Target point	*Point*: The target point can be a joint space or a Cartesian space point. Support distance conditions from the target point, (distance is based on the motion of the target coordinate system, without overlapping oscillations)
	Speed	*Speed*: Axis space end resultant velocity, in:degree/s。
	Interrupt condition	*Condition?* : Custom Boolean expression. For example, #Var1<<5.
	Output variable	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, and ReachEnd=False if the target point is not reached. ReachEnd is a BoolVar.

Precaution	

16.3.9 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.

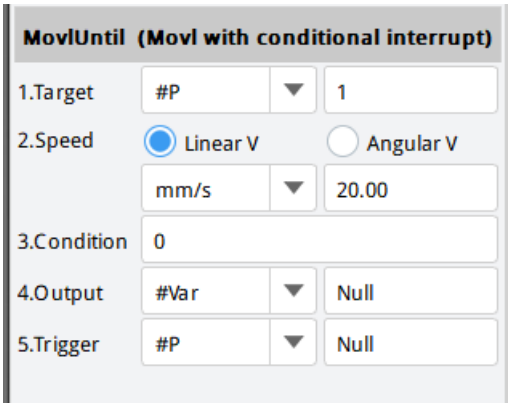


Diagram 97 MovlUntil Parameter Settings

Instruction format	MovlUntil P=*Point* (V=*Speed*mm/s) | (Ori= *OriSpeed* %) Cond=*Condition?* =>ReachEnd	
Examples	MovlUntil P=#P1 V=20mm/s Cond=(#Var1<<5) =>#Var10; MovlUntil P=#P1 OriV=20% Cond=(#Var1<<5) =>#Var10;	
Parameter description	Target point	*Point*: The target point can be a joint space or a Cartesian space point. Support distance conditions from the target point, (distance is based on the motion of the target coordinate system, without overlapping oscillations)
	Speed	*Speed*: Cartesian-space end path resultant speed, in:mm/s; *OriSpeed*: Cartesian space end pose resultant velocity

		percentage, velocity =%* pose end resultant maximum velocity.
	Interrupt condition	*Condition?*： Custom Boolean expression. For example, #Var1<<5.
	Output variable	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.
Precaution	*Speed* and *OriSpeed* can be selected. The former represents path planning and posture following;The latter is pose planning and path following.	

16.3.10 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.

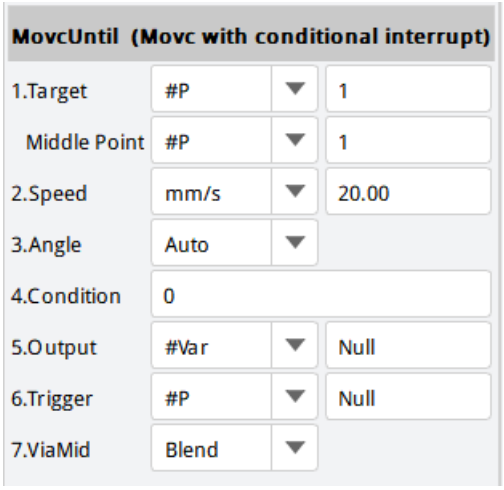


Diagram 98 MovcUntil Parameter Settings

<table>
<tr><td>Instruction format</td><td colspan="2">MovcUntil P=EndPoint Mid=MidPoint V=Speed mm/s [Deg = Degree] Cond=Condition? => ReachEnd</td></tr>
<tr><td>Examples</td><td colspan="2">MovcUntil P=#P1 Mid=#P2 V=20mm/s Deg=108
Cond=(#Var1<<5) =>#Var10;</td></tr>
<tr><td rowspan="5">Parameter description</td><td>Target point

Middle point</td><td>MidPoint: The midpoint of the arc, either in axis space or Cartesian space.
EndPoint: The endpoint of an arc can be a joint space or a Cartesian space point; Support distance conditions from the target point, (distance is based on the motion of the target coordinate system, without overlapping oscillations)</td></tr>
<tr><td>Speed</td><td>Speed: Cartesian-space end path resultant speed, in:mm/s。</td></tr>
<tr><td>Arc angle</td><td>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.</td></tr>
<tr><td>Interrupt</td><td>Condition? : Custom Boolean expression. For example, #Var1<<5.</td></tr>
<tr><td>Output variable</td><td>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, and ReachEnd=False if the target point is not reached. ReachEnd is a BoolVar.</td></tr>
<tr><td>Precaution</td><td colspan="2"></td></tr>
</table>

16.3.11 MoveExtUntil Conditional Interrupts MoveExt

MoveExtUntil external axis joint movement, if the Boolean expression changes to True during the movement, stops the current instruction or directly executes the next instruction; Otherwise, the movement will be in place. It can lock the triggering

position, including the target coordinate system position and joint position. This type of instruction does not support setting transition parameters and synchronous motion.

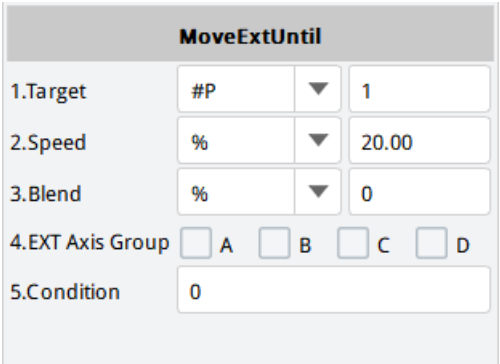


Diagram 99 MoveExtUntil Parameter Settings

Instruction format	MoveExtUntilMoveExt P=*Point* V=*Speed*% [Bl = *Blend* %] [Ext = [*A-D*]]Cond=*Condition?* =>ReachEnd	
Examples	MoveExtUntil P=#P1 V=20% Ext=A Cond=(#Var1<<5) =>#Var10;	
Parameter description	Target point	*Point*: The target point can be a joint space or a Cartesian space point. Support distance conditions from the target point, (distance is based on the motion of the target coordinate system, without overlapping oscillations)
	Speed	*Speed*: Joint space end synthesis velocity, unit: degree/s.
	Axle set	Ext: The external axis group number that requires synchronous motion has values of A, B, C, and D. If Ext is not selected, it means that all external axes move together. Whether the synchronous motion of the robot follows the external axis depends on whether the PCS recorded at the position point moves together with the external axis.

	Interrupt condition	*Condition?*： Customize Boolean expressions. For example, # Var1<<5.
Precaution	Supports up to 4 external axis groups moving simultaneously	

16.4 Motion Parameters

16.4.1 SetDyn Acceleration/Deceleration/Acceleration time

Acceleration/deceleration ratio setting and acceleration time setting.

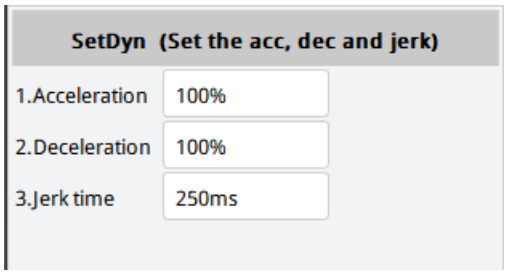


Diagram 100 SetDyn Setting Interface

Instruction format	SetDyn Acc = #% Dec = #% JerkTime = #	
Examples	SetDyn Acc=1% Dec=1% JerkTime=128ms;	
Parameter description	Acceleration	Acc: Global acceleration percentage, all instructions below apply.
	Deceleration	Dec: The global deceleration percentage applies to all subsequent instructions.
	Acceleration time	JerkTime: The acceleration time, in milliseconds, ranges from 32 to 512 milliseconds, with a default value of 128.
Precaution		

16.4.2 SetSpeed Global speed percentage

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

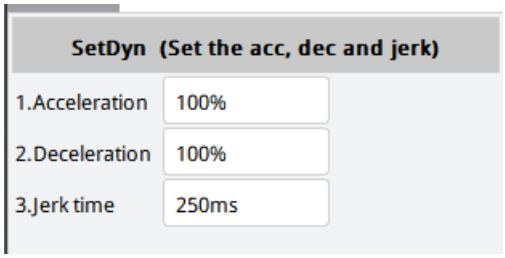


Diagram 101 SetSpeed Parameter Setting

Instruction format	SetSpeed Vel= #%	
Examples	SetSpeed Vel=20%;	
Parameter description	Global speed	Global speed percentage.
Precaution	If the Set Speed instruction is not called, the program defaults to 20%.	

16.4.3 SetCoord Coordinate System Switching

Coordinate system switching, applicable to all subsequent motion commands, records the coordinate points in Cartesian space until the next AKS oord command is encountered, or the switch is canceled when ResetCoord is encountered.

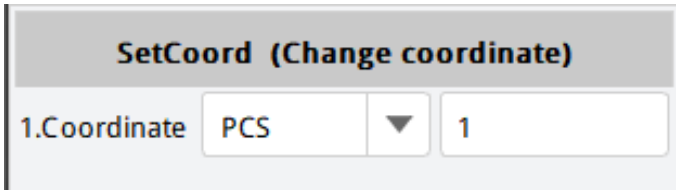


Diagram 102 SetCoord Settings

Instruction format	SetCoord Type= *Type* Idx=#	
Examples	SetCoord Type=PCS Idx=1;	
Parameter description	Coordinate system	*Type* :Coordinate system type, optional KCS, PCS, WCS; Idx : Coordinate system number.

Precaution	This instruction is effective for the position points between AKS oord and ResetCoord.

16.4.4 ResetCoord Coordinate System Switching Reset

Stop switching the coordinate system and restore it to the coordinate system recorded by the instruction itself, refer to the AKS oord instruction.

16.4.5 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 switch.

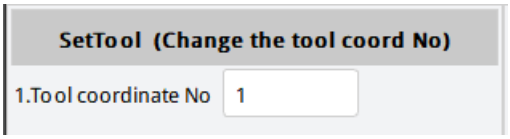


Diagram 103 SetTool Settings

Instruction format	SetTool Index=#	
Examples	SetTool Index=1	
Parameter description	No.	Index: Tool coordinate system ordinal number.
Precaution	This command is valid for the position points between SetTool and ResetTool.	

16.4.6 ResetTool Coordinate System Switching Reset

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

16.4.7 SetOriMode Posture 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.

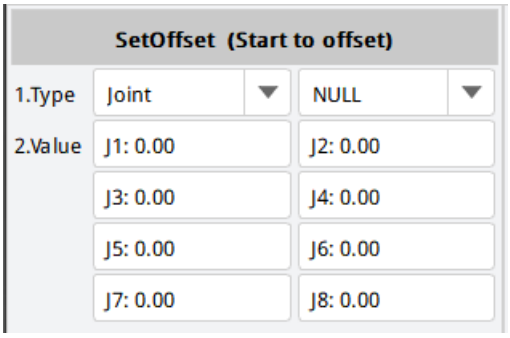


Diagram 104 SetOriMode Settings

Instruction format	SetOriMode Mode=*Standard*|*Unchange*|*Rotate*|*Normal*	
Examples	SetOriMode Mode=Standard;	
Parameter description	Mode	Mode: *Standard*, *Unchange*, *Rotate*, *Normal*。
Precaution	This instruction takes effect for the position points between SetOriMode and ResetOriMode.	

16.4.8 ResetOriMode Posture Mode Reset

Resumes the attitude standard follow mode Standard.

16.4.9 SetOffset Position Offset Setting

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

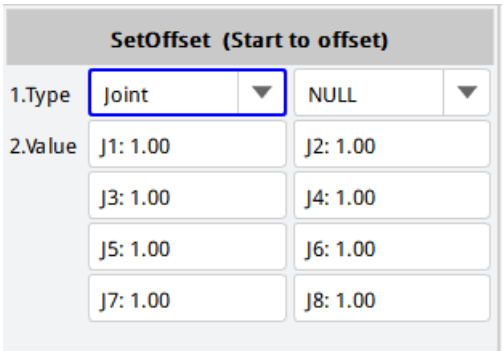
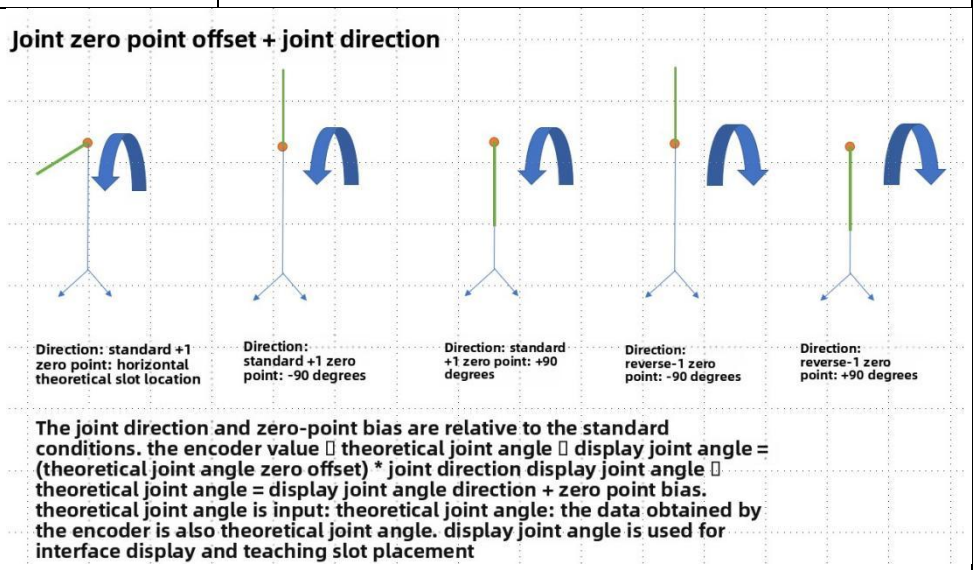


Diagram 105 SetOffset Parameter Settings

<table>
<tr><td>Instruction format</td><td colspan="2">SetOffset Type = “Joint” | “Pose” （ J1 =#J2 =#J3 =#J4 =#J5 =#J6 =#J7 =#J8 =#） || （X = #Y = #Z = #[Roll = #Pitch = #Yaw = #]）</td></tr>
<tr><td>Examples</td><td colspan="2">SetOffset Type=Joint J1=5.00 J2=10.00 J3=0.00 J4=0.00 J5=0.00 J6=0.00 J7=0.00 J8=0.00;
SetOffset Type=Pose X=5.00 Y=10.00 Z=0.00 Roll=0.00 Pitch=0.00 Yaw=0.00;
SetOffset Type=Pose X=0 Y=0 Z=0 Roll=10 Pitch=10 Yaw=10
SetOffset Type=Pose X=#Gvar1 Y=#Gvar2 Z=#Gvar3 Roll=#Gvar4 Pitch=#Gvar5 Yaw=#Gvar6</td></tr>
<tr><td rowspan="2">Parameter description</td><td>Type</td><td>Type: Bias type, robot joint bias, Cartesian position point bias
NULL: Input type, supports direct input of points or variable input to modify points
1. J1, J2, J3, J4, J5, J6, J7, J8 joint position offset or external axis offset;
2. X, Y, Z: Position offset;
3. Roll, Pitch, Yaw: attitude offset.</td></tr>
<tr><td>Offset value</td><td>Offset value</td></tr>
<tr><td>Precaution</td><td colspan="2"> 

This instruction takes effect for the position points between SetOffset</td></tr>
</table>

	and ResetOffset.

16.4.10 ResetOffset Position Offset Cancellation

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

16.4.11 SetBlendType Setting Transition Section Type

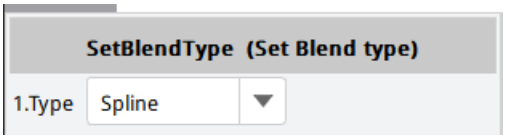


Diagram 106 SetBlendType Transition Section Type Setting

Instruction format	SetBlendType Type=*Fusion*|*Spline*|*VelBlend*	
Examples	SetBlendType Type=*Fusion*;	
Parameter description	Mode	Mode: *Fusion*, *Spline*, *VelBlend*。
Precaution		

16.4.12 ResetBlendType Setting Transition Section Type

Setting the transition type is no longer effective and is used in conjunction with SetBlendType.

16.4.13 GetAxisInput Get Drive Input

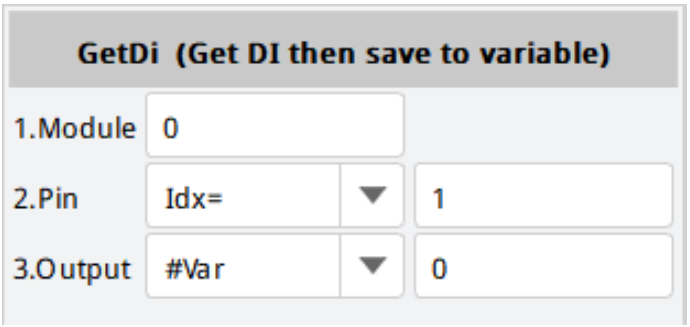


Diagram 107 GetAxisInput Get Drive Input

Instruction format	GetAxisInput Axis=*Axis Number* Bit=*Bit* Value:Representing variable output*#GVar* 或*#Var*。	
Examples	GetAxisInput Axis=1 Bit=1=>#Var1;	
Parameter description	Axle set	*Axis*: Indicate the joint axis number of the robot, supporting up to 16 axes
	Bit	Bit:
	Variable	The numerical input value is stored in a numerical variable.
Precaution		

16.5 IO Operation

16.5.1 SetDo Output Digital Quantity

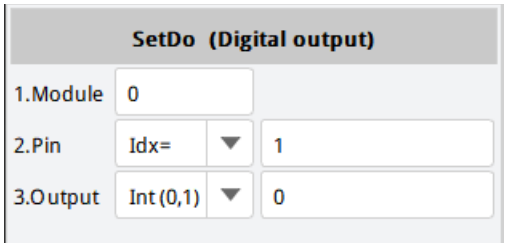
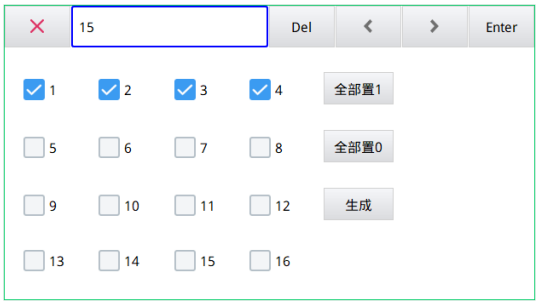


Diagram 108 SetDo Setting

Instruction format	SetDo Val=*Value* Grp=*Group* Idx = *#* | *ALL*	
Examples	SetDo Val=1 Grp=0 Idx=1; SetDo Val=0b1111 Grp=1 Idx=All;	
Parameter description	Module	*Group* : Indicates the group of digital outputs, and the controller software considers 16 IO as a group.
	Point location	Idx. : Indicates the IO sequence number in the group. If *ALL* is selected, it indicates that the group is output together.

	Output value	*Value*： Indicates that the digital output is *0* or *1.* If *ALL* is selected,*value* is the binary number *0b01100110*。
Precaution		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

16.5.2 SetDoAt Output IO in Motion

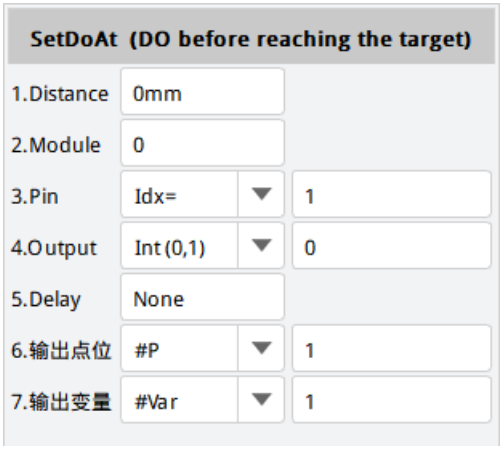


Diagram 109 SetDoAt

Instruction format	SetDoAt Dis=*Distance* Val=*Value* Grp=*Group* Idx = *#* | *ALL* [Td=*Delay* ms]	
Examples	SetDoAt Dis=7mm Val=1 Grp=1 Idx=1 Td=5ms; SetDoAt Dis=7mm Val=0b1111 Grp=1 Idx=All Td=5ms;	
Parameter Description	Distance	*Distance* : distance from the target point, in mm.
	Module	*Group* : Group: indicates the group of digital outputs,

		and the controller software considers 16 IO as a group.
	Point location	Idx : Indicates the IO sequence number in the group. If *ALL* is selected, the group is output together.
	Output value	*Value* : Indicates that the digital output is *0* or *1.* If *ALL* is selected,*value* is the binary number *0b01100110.*
	Delay time	*Delay* : indicates the delay time in ms.
Precaution	This instruction is valid only for the previous instruction.	

16.5.3 SetDoDeltaS Output Digital Quantity in Motion

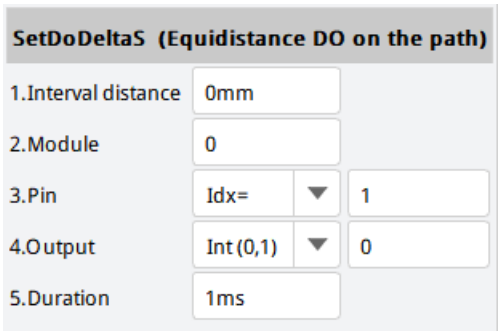


Diagram 110 SetDoDeltaS Settings

Instruction format	SetDoDeltaS Dis=*DeltaS* Val=*Value* Grp=*Group* Idx = *#*| *ALL* Width= *#*ms	
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	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.
	Point location	Idx : Indicates the IO sequence number in the group. If *ALL* is selected, the group is output together.
	Output value	*Value* : Indicates that the digital output is *0* or *1.* If *ALL* is selected,*value* is the binary number *0b01100110.*

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

16.5.4 SetDoDeltaT Output Digital Quantity in Motion

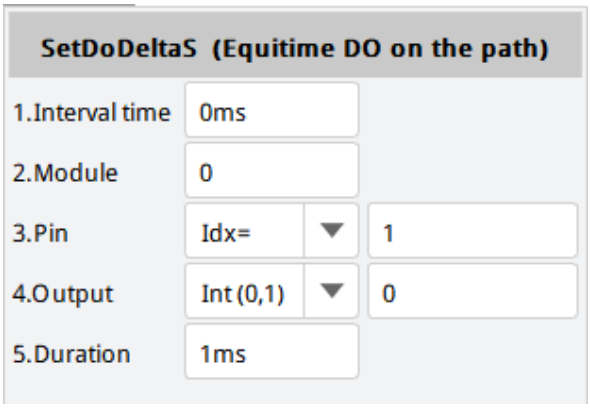


Diagram 111 SetDoDeltaT Settings

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	Time interval	*DeltaT* : Represents the time interval in ms.
	Module	*Group* : Indicates the group of digital outputs, and the controller software considers 16 IO as a group.
	Point location	Idx : Indicates the IO sequence number in the group. If *ALL* is selected, the group is output together.
	Output value	*Value* : Indicates that the digital output is *0* or *1.* If *ALL* is selected,*value* is the binary number *0b01100110.*
	Delay time	Width : Indicates the signal duration in ms.
Precaution	This instruction is valid only for the previous instruction.	

16.5.5 GetDo Digital Output Reading

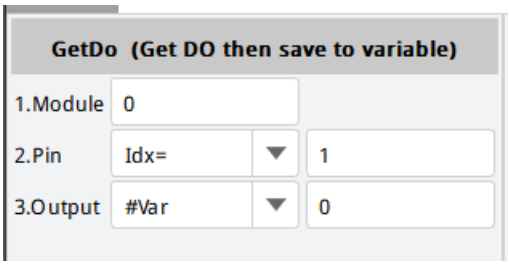


Diagram 112 GetDo Parameter

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 the group of digital outputs, and the controller software considers 16 IO as a group.
	Point location	Idx : Indicates the IO sequence number in the group. If *ALL* is selected, the group is output together.
	Variable	The digital output value is stored in a numeric variable.
Precaution		

16.5.6 GetDi Digital Quantity Input Acquisition

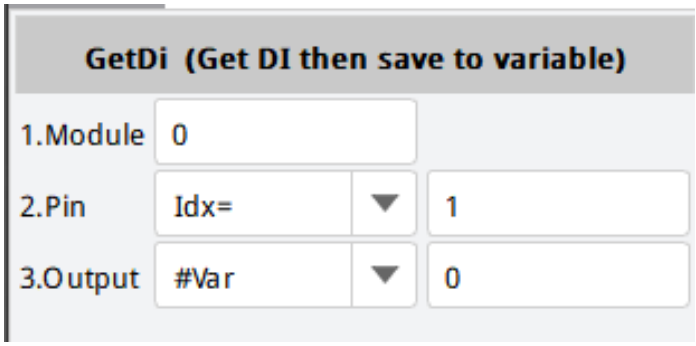


Diagram 113 GetDi Parameter

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.
	Point location	Idx : Indicates the IO sequence number in the group. If ***ALL*** is selected, the group is imported together.
	Variable	Digital input values are stored in numeric variables.
Precaution		

16.5.7 SetAo Output Analog Quantity

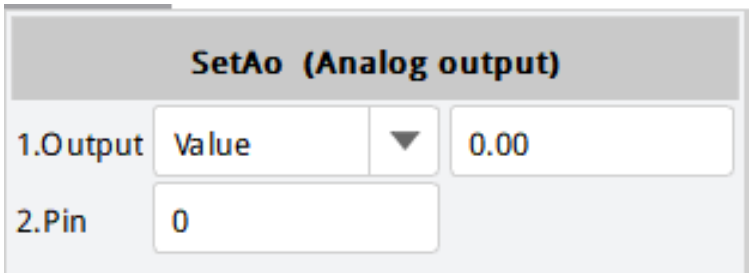


Diagram 114 SetAo Parameter Settings

Instruction format	SetAo Val=*Value* Idx = *#*	
Examples	SetAo Val=7.00 Idx=1;	
Parameter description	Output value	***Value*** : Indicates the analog output value.
	Point location	Idx : Indicates the IO sequence number in the group.
Precaution		

16.5.8 SetAoAt Output Analog Quantity in Motion

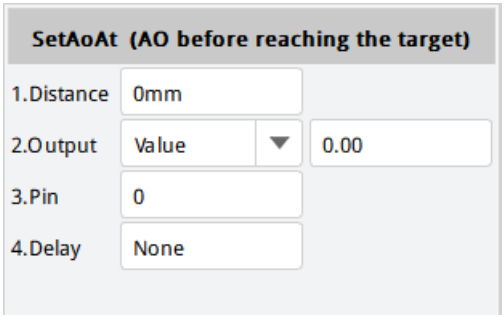


Diagram 115 SetAoAt Parameter Setting

Instruction format	SetAoAt Dis=*Distance* Val=*Value* Idx = *#* [Td=*Delay* ms]	
Examples	SetAoAt Dis=10mm Val=8.00 Idx=25ms;	
Parameter description	Distance	*Distance* : distance from the target point, in mm.
	Output value	*Value* : Represents the analog value.
	Point location	Idx : Indicates the IO sequence number in the group.
	Delay time	*Delay* : indicates the delay time in ms.
Precaution	This instruction is valid only for the previous instruction.	

16.5.9 GetAo Analog Output Acquisition

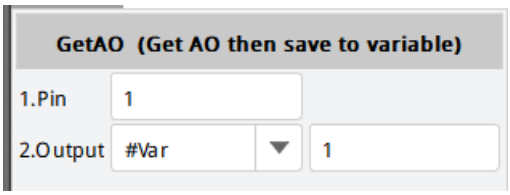


Diagram 116 GetAo Parameter Setting

Instruction format	GetAo Idx = *#* =>*#Var#*
Examples	GetAo Idx=1 =>#Var2;

Parameter description	Point location	Idx : Indicates the IO sequence number in the group.
	Variable	Analog output values are stored in variables.
Precaution		

16.5.10 GetAi Analog Input Acquisition

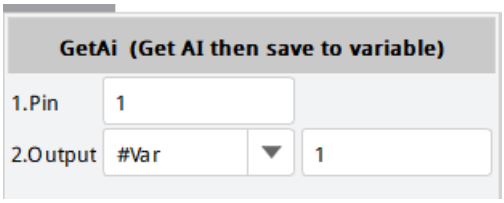


Diagram 117 GetAi Parameter Setting

Instruction format	GetAi Idx = *#* =>*#Var#*	
Examples	GetAi Idx=1 =>#Var2;	
Parameter description	Point location	Idx : Indicates the IO sequence number in the group.
	Variable	Analog input values are stored in variables.
Precaution		

16.5.11 Wait for the condition to complete

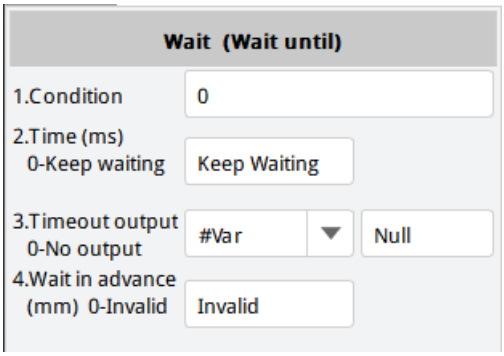


Diagram 118 Wait Parameter Setting

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.
	Waiting time	Time : indicates the upper limit of the waiting time. If not set, it means to wait all the time (in s).
	Timeout output variable	*#Var#* is True if timeout occurs, False otherwise.
Precaution		

16.5.12 Pulse Output

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

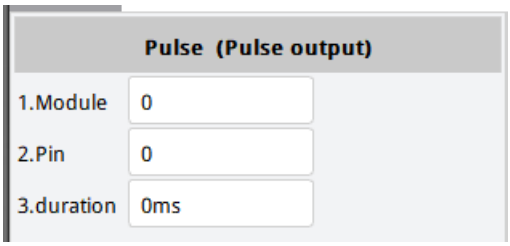


Diagram 119 Pulse Parameter Setting

Instruction format	Pulse Grp=*Group* Idx = *#* Width = *#* ms	
Examples	Pulse Grp=1 Idx=1 Width=5ms;	
Parameter description	Module	*Group* : Indicate the group of digital inputs,and the controller software considers 16 IO as a group.
	Point location	Idx : Indicates the IO sequence number in the group.
	Duration	Width indicates the setting time in ms.
Precaution		

16.6 Motion Management

16.6.1 Delay Time

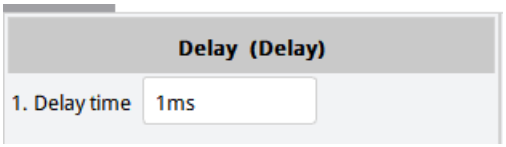
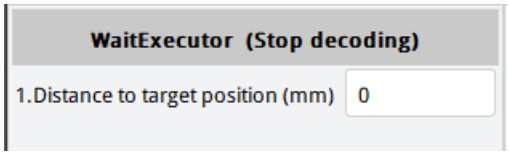


Diagram 120 Delay Parameter Setting

Set a delay to increase the robot's movement time.

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

16.6.2 WaitExecutor Stops Decoding



图表 121 WaitExecutor 停止译码设置

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

Instruction format	Delay Dis=#mm;	
Examples	WaitExecutor Dis=100mm;	
Parameter description	Distance from the target location point.	Distance from the target location point, in mm.
Precaution		

16.6.3 tod Gets System Timing

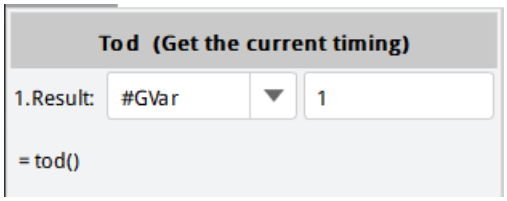


Diagram 122 tod Parameter Settings

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	Variable	The recorded time is stored in *#Var#.*

16.6.4 SysDate Recording System Time

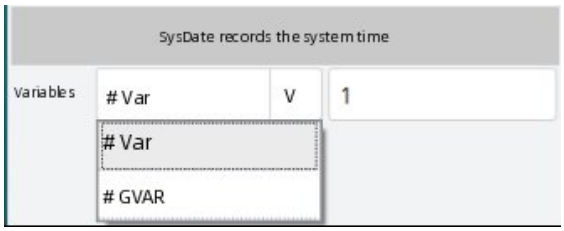


Diagram 123 SysDate Parameter Setting

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

description		

16.6.5 Pause

Stop execution and wait for external signal to restart

Examples	Pause;

16.6.6 User Alarm

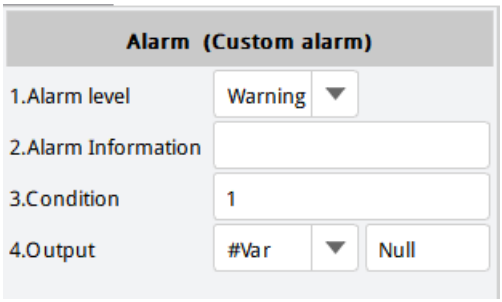


Diagram 124 Alarm Custom Alarm Parameter Settings

By using the teaching program to actively trigger the servo under alarm, the triggering conditions can be set

Examples	Alarm Level=Warning Message="Alarm content"Cond=(Condition);

16.6.7 GetPoint Save Current Location

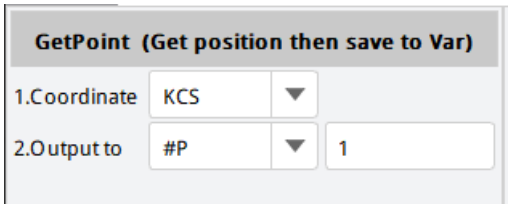


Diagram 125 GetPoint Parameter Setting

Instruction format	GetPoint *Type* => P=*Point*	
Examples	GetPoint KCS =>#P1;	
Parameter description	Coordinate system	*Type* Coordinate system model, KCS,WCS,PCS, ACS.

	Position point	*Point* Storage location point.
Precaution		

16.7 Control Flow

16.7.1 Call Subprogram

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. 注意：The calling and called programs must be in the same project. Subprogram name: can only contain alphanumeric underscores and cannot start with a number.

16.7.2 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 marked with"1"and moves down to the sixth line to point

		P3

16.7.3 IF Condition Judgement

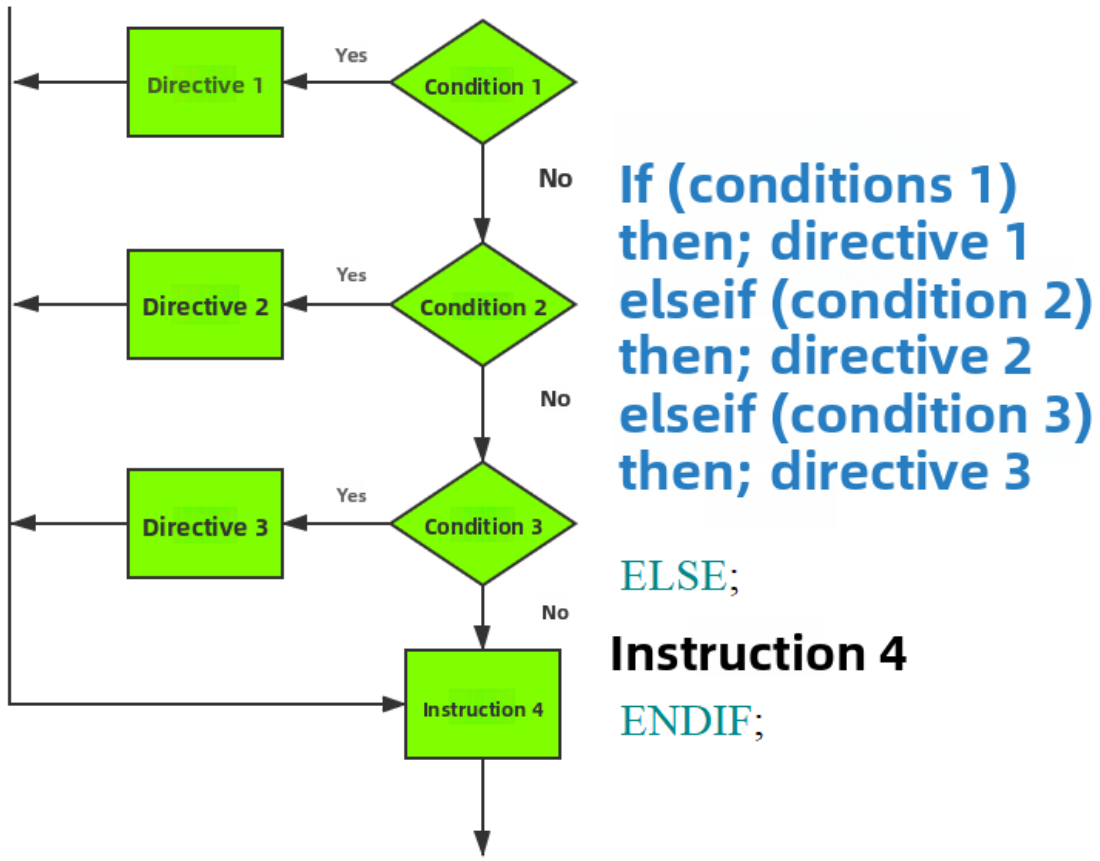


Diagram 126 IF Flow Chart

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 pointP2; otherwise, the end of the robot

		moves to the pointP3.

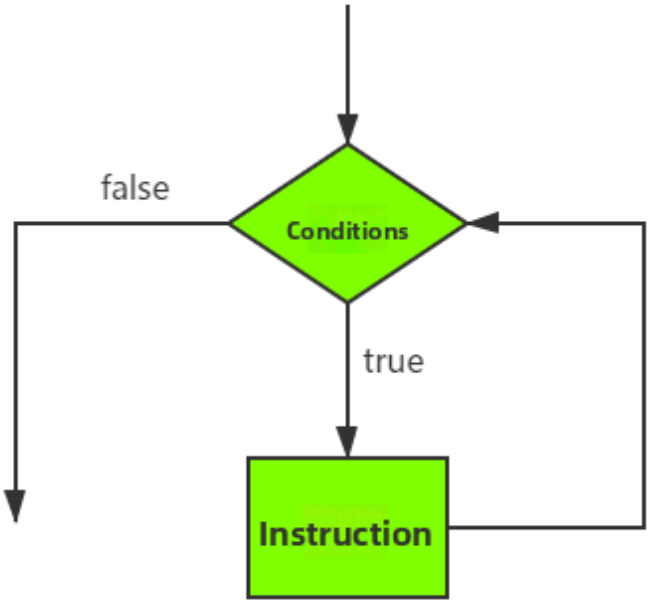
16.7.4 ELSELF Condition Judgement

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, running the fourth row, moving to the point P2, and ending the judgment; judging condition 2(whether the value of #Var1 is equal to 8), if the value is equal to 8, running the sixth line, moving to point P3, and ending the judgment; otherwise move to point P4.

16.7.5 WHILE Cycle Conditional Judgement

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



WHILE (条件) DO
指令
ENDWHILE;

Diagram 127 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.

16.7.6 FOR Loop Conditional Judgement

Diagram 128 FOR Loop

Instruction format	FOR #Var# FROM #1TO #2 STEP #3 DO; Program 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 loops between points P2 and P3 for 10 times 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)

16.8 Notes and Calculations

16.8.1 /**/ Muti-lines 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.

16.8.2 Comment/Uncomment Current Line

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

16.8.3 ASSIGN(=)Assignment

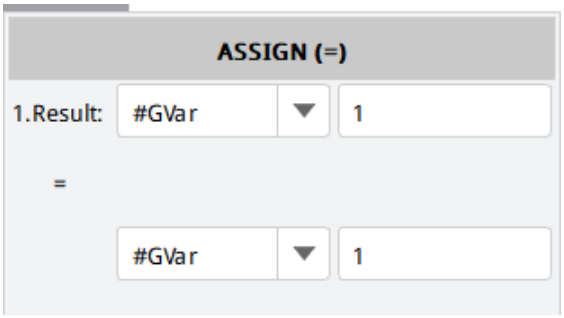


Diagram 129 Assignment

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; Support for numeric, string, positional, and positional increment variables.

16.8.4 ADD(+) Addition

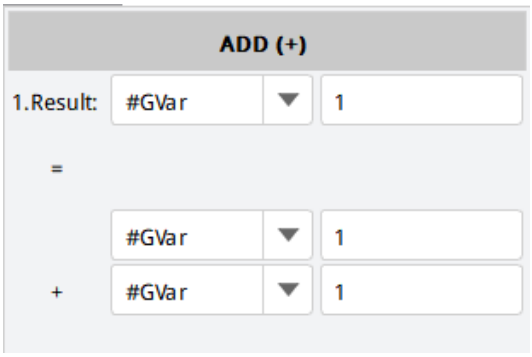


Diagram 130 Addition

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.

16.8.5 SUB(-) Subtraction

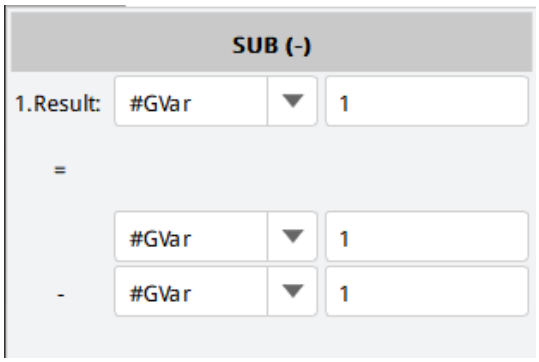


Diagram 131 Subtraction

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

16.8.6 MUL(*) Multiplication

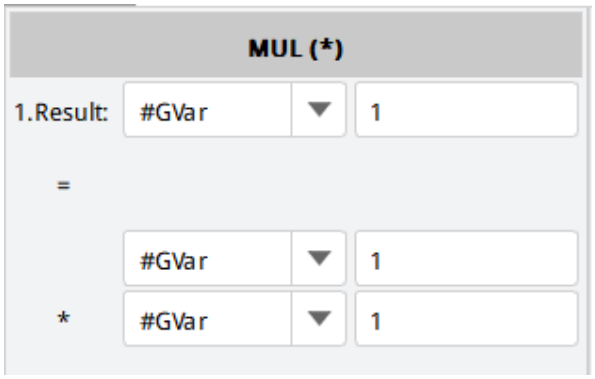


Diagram 132 Multiplication

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

16.8.7 DIV(/) Division

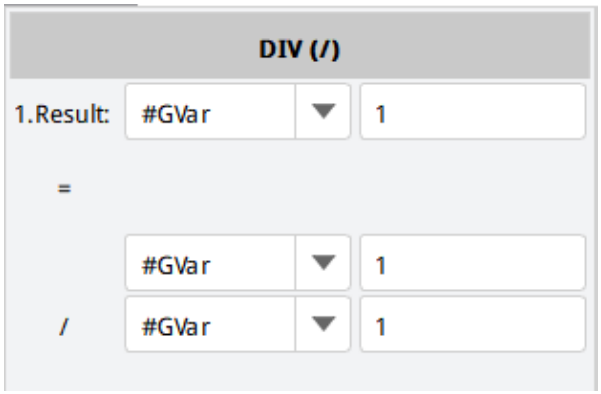


Diagram 133 Division

Instruction format	C=A/B
Examples	#GVar1 = #GVar1/#Var2; #GVar1 = #GVar1/2; #GVar1 = 2/3;
Parameter description	Diving the value of A by the value of B and assigning the value to C; Support for numeric variables

16.8.8 Expr Advanced Expressions/Statements

Expr advanced expression support:

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

Diagram 134 Advanced Expression Interface

Extended description

Operation

%	Modulo(Remainder operation)
++	Self-increment
--	Autodecrement
&	Bitewise AND
|	Bitewise OR
^	Bitewise XOR
<<	Left shift
>>	Right shift
~	Negation

Mathematical function

ceil()	Rounding up
trunc(,)	Variable
exp()	e exponential function
log(,)	10 Logarithmic function
mod(,)	Complementary function
sqrt(,)	Square root
pow(,)	Exponential function
sin()	Sine
cos()	Cosine
tan()	Tangent
acos()	Arccosine
asin()	Arcsine
atan()	Arctanent
atan2()	Arctanent（bivariate）
abs()	Absolute value

Logic

&&	Logical AND
||	Logical OR
!	Logical NOT
>=	Greater than or equal to
<=	Less than or equal to
>	Greater than
<	Lesser 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

Chinse and upper and lower case switching

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

Examples:

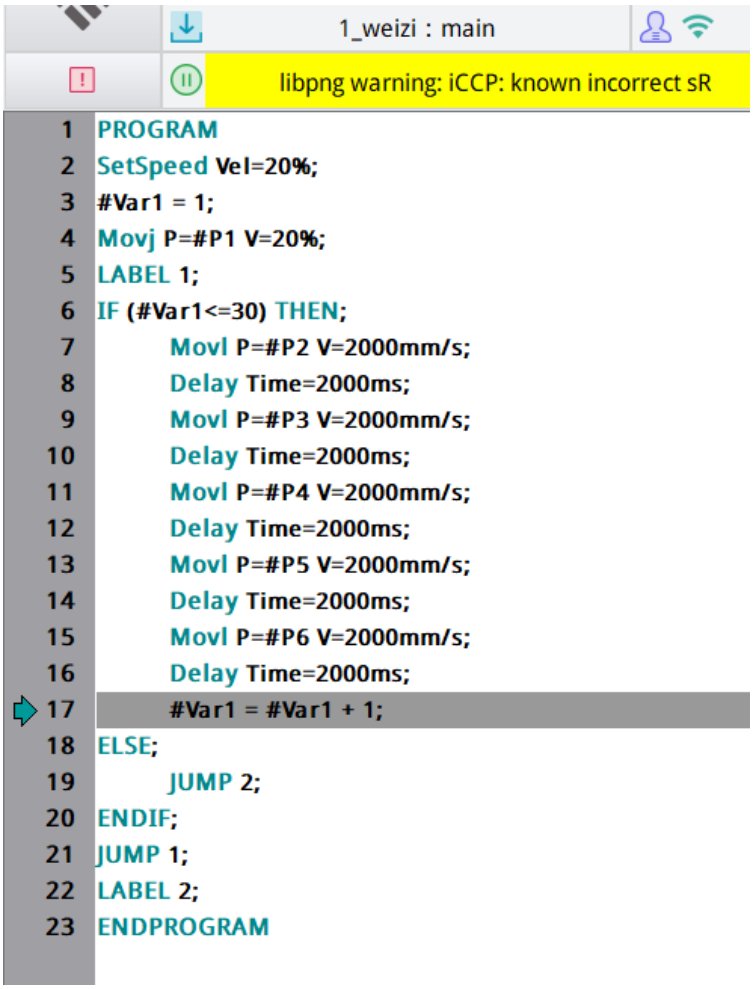


Diagram 135 Example of an Advanced Expression

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

Line 6: IF statement (custom Boolean expression)

Can be entered through advanced expressions.

16.9 Mathematically Function

abs(X)	Take absolutely value
sqrt(X)	Root sign
exp(X)	Natural index
ln(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
trunc(X)	Rounding
pow(X,Y)	X to the power of Y
mod(X,Y)	Remainder of X to Y
atan2(X,Y)	Arctangent

16.10 Character Functions

16.10.1 strlen(X) Finding the Length of a String

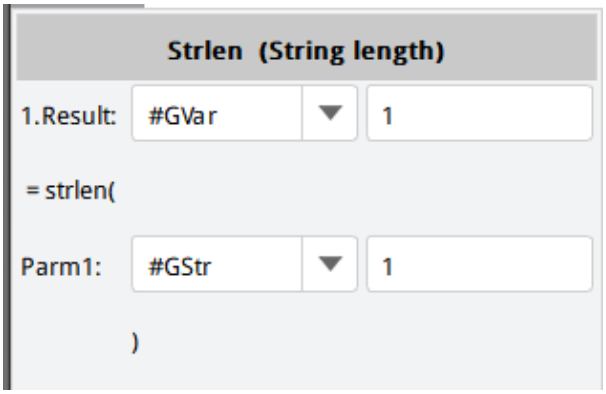


Diagram 136 strlen（X）Find the Length of the String

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

16.10.2 str2num(X) String to Real

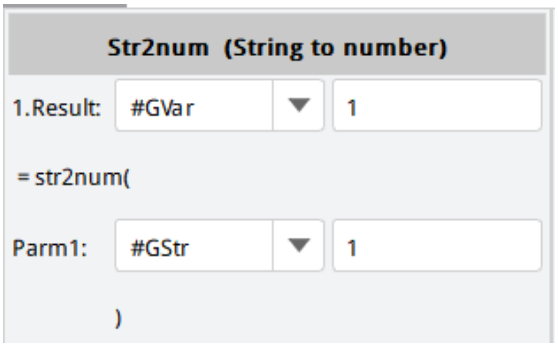


Diagram 137 str2num（X）String to Real

Parameter description	Converts a string to a real variable
Examples of instruction	#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

16.10.3 num2str(X) Real to String

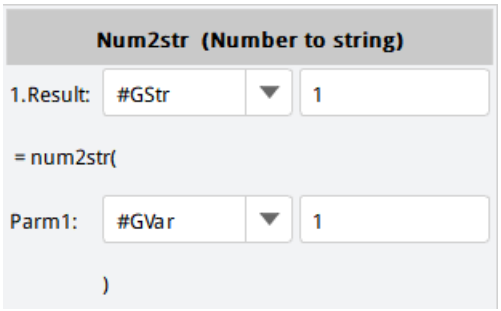


Diagram 138 num2str（X）Real to String

Parameter description	Converts a string to a real variable
Examples of instruction	#Str1=num2str(#Var1)
Parameter description	It's generally used to convert a numeric string sent and received into a real number for calculation or assignment

16.10.4 strcat(X,Y) String Splicing

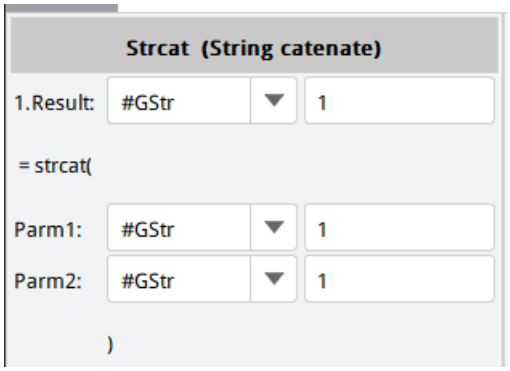


Diagram 139 strcat(X,Y) String Splicing

Parameter description	Concatenate 2 strings into 1 string
Examples of	#Str2=asd

instruction	#Str3=fgh #Str1=strcat(#Str2,#Str3)
Operation result	#Str1=asdfgh

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

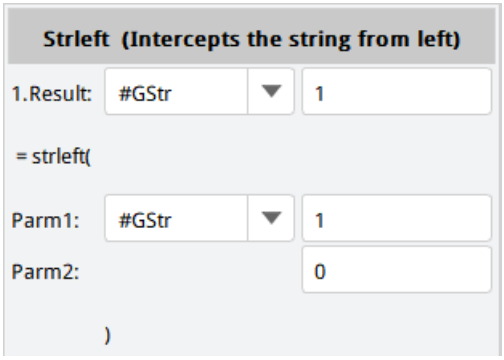


Diagram 140 strleft(X,N) Take the Left Character

Parameter description	Takes N characters to the left of the string X
Examples of instruction	#Str2=asdfgh #Str1=strleft(#Str2,2)
Operation results	#Str1=st

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

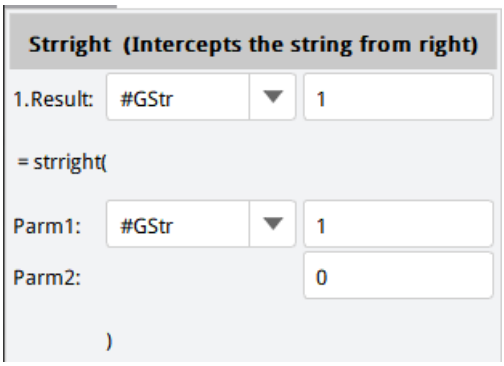


Diagram 141 strright(X,N) Take the Right Character

Parameter description	Takes N characters from the right of a string X
Examples of instruction	#Str2=asdfgh #Str1=strright(#Str2,2)
Operation results	#Str1=gh

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

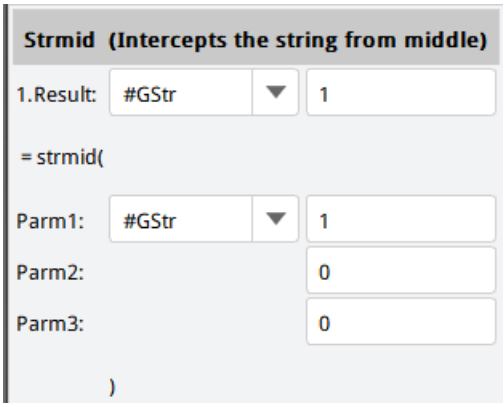


Diagram 142 strmid(X,M,N) Take Character

Parameter description	Takes the M-th to N-th characters of the string X

Examples of instruction	#Str2=asdfgh #Str1=strmid(#Str2,3,4)
Operation results	#Str1=df

16.10.8 strsplit(X,Y,N) String Split

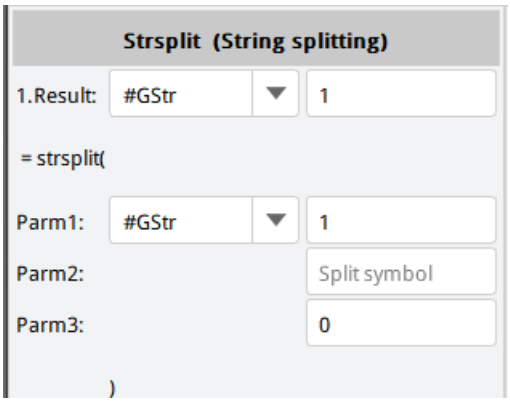


Diagram 143 strsplit(X,Y,N) String Split

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

16.10.9 Buildstr String Construction

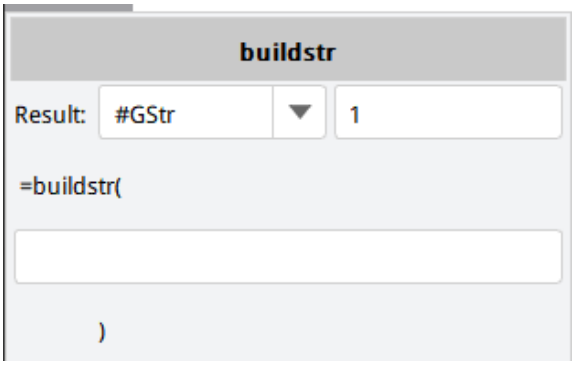


Diagram 144 buildstr String Construction

Parameter description	Replace the # Str, # GStr, # Var, and # GVar in the format string with actual variables and write them into the message
Examples of instruction	#Str1 = buildstr("value = #GVar1")
Operation results	#Str1 = "value = 1"(假设#GVar1 = 1)

16.10.10 Parsestr String Parsing

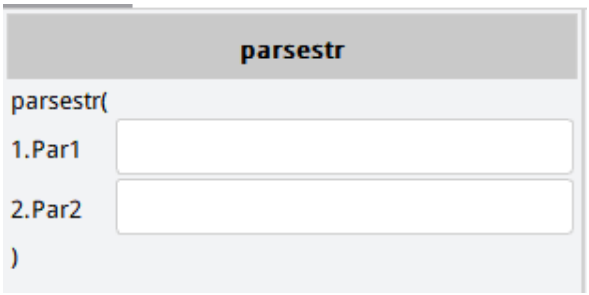


Diagram 145 parsestr String Parsing

Parameter description	Parse the message in the specified format and convert the specified characters in the message into variables
Examples of instruction	parse("value = #GVar1", "value = 3.14")
Operation results	#GVar1 = 3.14

16.11 Communication

16.11.1 TcpOpen Enables tcp/ip Communication

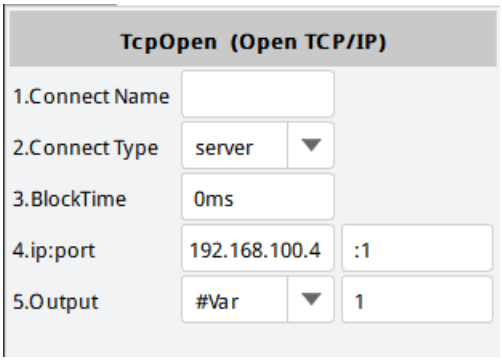


Diagram 146 TcpOpen Enables tcp/ip Communication

Examples of instruction	TcpOpen name="imf" type=""server"" block=1 addr="192.168.100.33:22" =>#Var1;	
Parameter description	Connection name	an identifier of the link, with which subsequent receive and send operations are associated;
	Connection type	link type, server: Create a server; client: Create a client and link to the server;
	Blocking 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 value	output values, >=0: Number of links maintained by the socket; -1: The server is starting, and the client is linking; -2: Failure;

16.11.2 TcpRecv Receiving Data

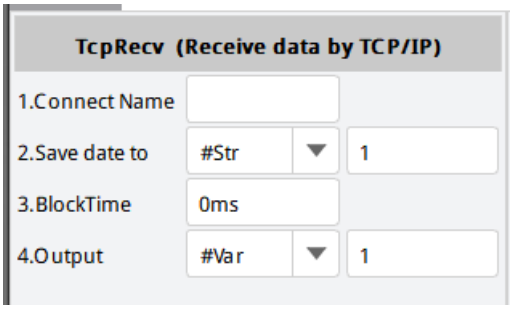


Diagram 147 TcpRecv Receiving Data

Examples of instruction	TcpRecv name="imf" dest=#Str1 block=0 =>#Var1;	
Parameter description	Connection Name	Receiving a data link identifier, created by TcpOpen;
	Data retention	A character string variable for storing the received data;
	Blocking time	If that instruction is bloc, 1: Blocking time (ms); waiting time for instruction execution 0: non-blocking;
	Output value	Output values, >=0: Number of bytes received; -2: no corresponding identifier;

16.11.3 TcpSend Sending Data

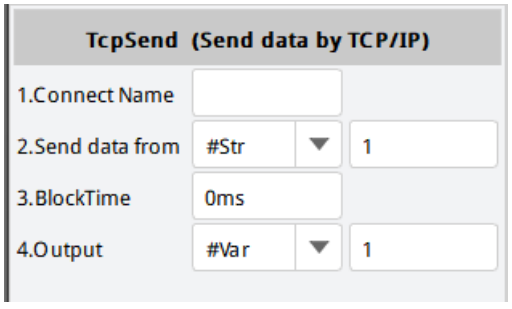


Diagram 148 TcpSend Sending Data

Examples of instructions	TcpSend name="imf" src=#Str1 block=0 =>#Var1;	
Parameter description	Connection name	Link identifier of the sending data, created by TcpOpen;
	Data retention	String variable for storing the sent data;
	Blocking	if that instruction is bloc, 1: Blocking time (ms); waiting time for instruction execution 0: non-blocking;
	Output value	output values, >=0: Number of bytes sent; -3: no corresponding identifier;

16.11.4 TcpClose Closing a TCP/IP Connection

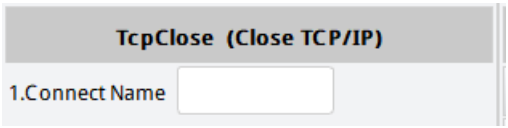


Diagram 149 TcpClose Closing a TCP/IP Connection

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

16.11.5 SerialOpen Open Serial Port

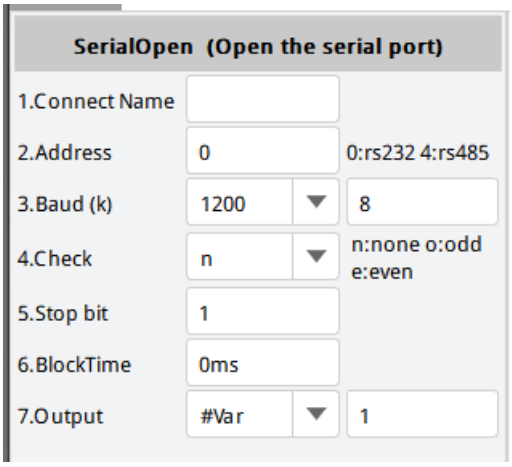


Diagram 150 SerialOpen Open Serial Port

Examples of instruction	SerialOpen name="imf" addr=4 par="1200,8,n,1" block=0 =>#Var1;	
Parameter description	Connection 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 rate	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 transmission and provides the computer with an opportunity to correct clock synchronization. The greater the number of bits available for the stop bit, the greater the tolerance for different clock synchronizations, but at the same time the slower the data transfer rate.
	Blocking time	If that instruction is bloc, 1: Blocking time (ms); waiting time for instruction

		execution 0: non-blocking;
	Output value	0: The serial port is opened normally; -1: failure;

16.11.6 SerialRecv Receiving Data

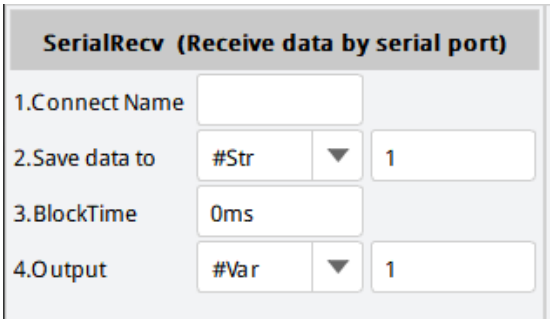


Diagram 151 SerialRecv Receiving Data

Examples	SerialRecv name="imf" dest=#Str1 block=0 =>#Var1;	
Parameter description	Connection name	Serial port identifier for receiving data, created by SerialOpen;
	Data retention	A character string variable for storing the received data;
	Blocking time	If that instruction is bloc, 1: Blocking time (ms); waiting time for instruction execution 0: non-blocking;
	Output value	Output values, >=0: Number of bytes received; -1: Serial port is not open;

16.11.7 SerialSend Sending Data

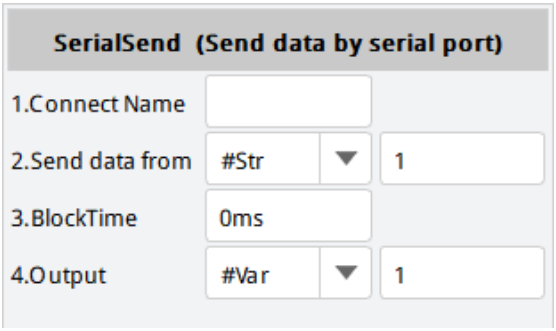


Diagram 152 SerialSend Sending Data

Examples of instruction	SerialSend name="imf" src=#Str1 block=0 =>#Var1;	
Parameter description	Connection name	Serial port identifier for sending data, created by SerialOpen;
	Data retention	String variable for storing the sent data;
	Blocking time	If that instruction is bloc, 1: Blocking time (ms); waiting time for instruction execution 0: non-blocking;
	Output value	Output values, >=0: Number of bytes sent; -1 : Serial port is not open;

16.11.8 SerialClose Closing a TCP/IP Connection

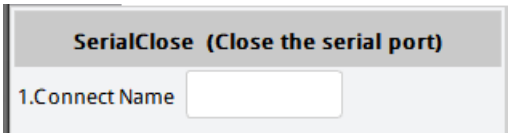


Diagram 153 SerialClose Closing a TCP/IP Connection

Examples of instruction	SerialClose name="imf";	
Parameter	Connection	Close serial port identifier, created by SerialOpen;

description	name	

16.11.9 CanbusOpen Canbus Communication

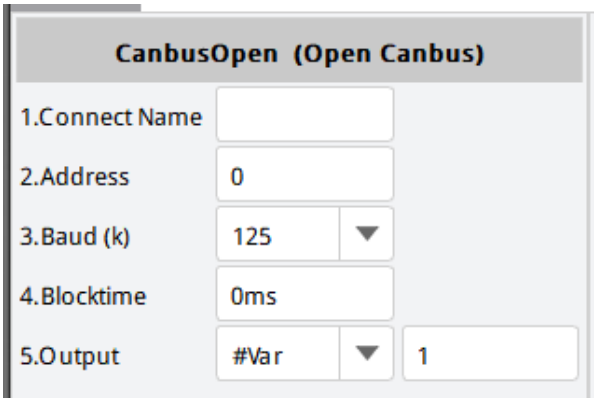


Diagram 154 Open Canbus Communication

Examples of instruction	CanbusOpen name="imf" addr=1 par=125 block=0 =>#Var1;	
Parameter description	Connection 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 rate	Canbus baud rate: 125 means 125k, in addition to the commonly used 1000k, 500k, 250k and so on;
	Blocking time	If that instruction is bloc, 1 : Blocking time(ms); waiting time for instruction execution 0 : non-blocking;
	Output value	0: canbus is opened normally; -1: failure;

16.11.10 CanbusRecv Receiving Data

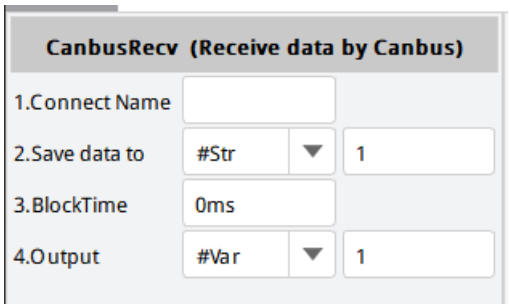


Diagram 155 CanbusRecv Receiving Data

Examples of instruction	CanbusRecv name="imf" dest=#Str1 block=0 =>#Var1;	
Parameter description	Connection name	Canbus identifier, created by CanbusOpen;
	Data retention	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;
	Blocking time	If that instruction is bloc, 1: Blocking time (ms); waiting time for instruction execution 0: non-blocking;
	Output time	Output values, >=0: the number of received can frames; -1 : failure;

16.11.11 CanbusSend Sending Data

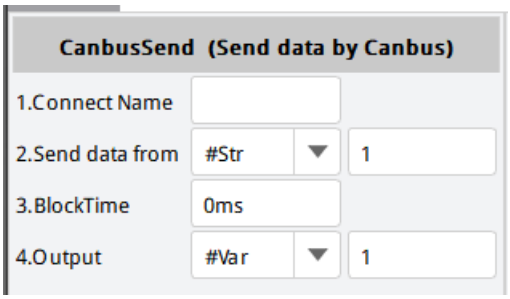


Diagram 156 CanbusSend Sending Data

Examples of instruction	CanbusSend name="imf" src=#Str1 block=0 =>#Var1;	
Parameter description	Connection name	Canbus identifier, created by CanbusOpen;
	Data retention	The string stored in the sent data frame, the format is the same as CanbusRecv
	Blocking time	If that instruction is bloc, 1: Blocking time (ms); waiting time for instruction execution 0: non-blocking;
	Output value	Output values, >=0: the number of received can frames; -1: failure;

16.11.12 CanbusClose Closing a TCP/IP Connection

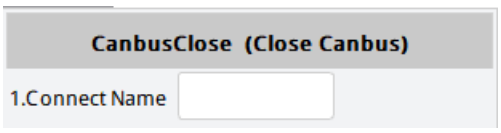


Diagram 157 CanbusClose Closing a TCP/IP Connection

Examples of instruction	CanbusClose name="imf";

Parameter description	Connection name	Close the canbus identifier, created by CanbusOpen;

16.11.13 MBTcpOpen Enable ModbusTcp Communication

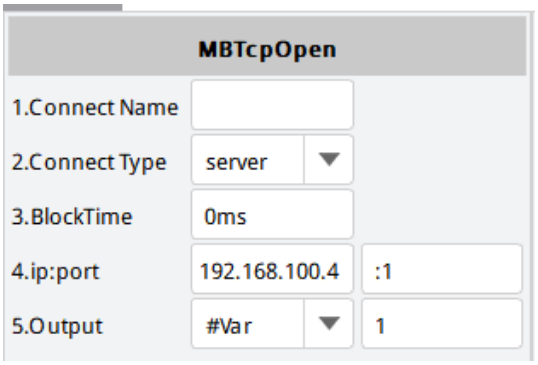


Diagram 158 MBTcpOpen Enable ModbusTcp Communication

Examples of instruction	MBTcpOpen name="imf" type="server" block=0 addr"192.168.100.3:1"=>#Var1;	
Parameter description	Connection name	The identifier of this link is associated with subsequent receiving and sending operations;
	Connection type	Link type, Server: Create a server; Client: Create a client and link it to the server;
	ip:port	Link IP address: End port; Among them, the server-side can listen to all network ports, so creating a server does not require an IP address
	Blocking time	Is the instruction blocked, 1: Blocking time (ms); The waiting time for instruction execution 0: Non blocking;
	Output value	0: Canbus opens normally; -1: Failure;

16.11.14 MBTcp Read/Write Register

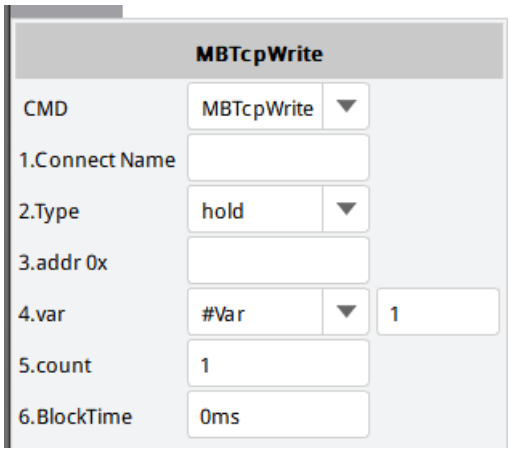


Diagram 159 MBTcp Read/Write Register

Examples of instruction	MBTcpWrite name="imf" type="hold" addr=0x10001 var=#Var1 count=1 block=0;	
Parameter description	Connection name	The identifier of this link is associated with subsequent receiving and sending operations;
	Instruction type	Write: Write register; Read: Read register;
	Start address	The starting address of the register is 0x: 10001;
	Type	Register type 1：hold；2：input；3：coil；4：discrete；
	Starting variable	The numerical variables stored in the sent data;
	Quantity	The number of variables written;
	Blocking time	Is the instruction blocked, 1: Blocking time (ms); The waiting time for instruction execution 0: Non blocking;

16.11.15 MBTcpOpen Close modbusTcp Communication

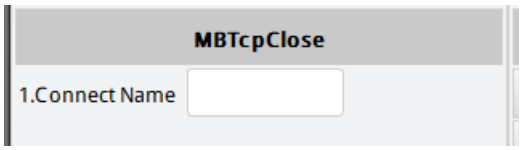


Diagram 160 MBTcpOpen Close modbusTcp Communication

Examples of instruction	MBTcpClose name="imf";	
Parameter description	Connection name	Close the link identifier, created by MBTcpOpen;

17 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 disconneted; -2: slave number change; -13: hardware not register; -23: NIC driver abnormal; -33: cable disconneted)	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 interploation, 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: slave error)	Check whether the EtherCAT communication is normal, whether 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	Joint reference calibration is invalid	Re-zero
1004	Joint %1 positive limit error, current %2 out of [%3, %4]	Warning clears automatically after jog back to range of motion
1005	Joint %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	Joint %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 attitude
1013	Start point pose index %1 != Aux point pose index %2, UUID = %3	Check teach point attitude
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	Check setup parameters

	resolution %2	
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 joint postive limit %2, negative limit %3	Check setup parameters
200B	Invalid parameter: %1 joint max velocity %2, max jog velocity %3	Check setup parameters
200C	Invalid parameter: %1 joint, gear ratio in joint side %1 > gear ratio in motor side %2	Check setup parameters
200D	Invalid parameter: %1 joint sign %2	Check setup parameters
200E	Invalid parameter: Cartesion %1 postive limit %2, negative limit %3	Check setup parameters
200F	Invalid parameter: Cartesion %1 max velocity %2, max jog velocity %3	Check setup parameters
2010	Invalid parameter: Cartesion %1 sign %2	Check setup parameters
2011	Invalid parameter: Cartesion end linear max velocity %1, max	Check setup parameters

	acceleration %2, max decceleration %3	
2012	Invalid parameter: Cartesion end angular max velocity %1, max acceleration %2, max decceleration %3	Check setup parameters
2013	Invalid parameter: Joint space end max velocity %1, max acceleration %2, max decceleration %3	Check setup parameters
2014	Invalid parameter: External group %1, joint space end max velocity %2, max acceleration %3, max decceleration %4	Check setup parameters
2015	Invalid parameter: %1 coordinate, index %2	Check setup parameters
2016	Invalid parameter: Joint %1 reference zero pulse, based on referemce current joint 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	Internal error

	not been initialized, call imf::InitMotionController first	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 unit %5	Check command parameters
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	Manual Clear 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

Error Code	Error Message	Solution
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	

Error Code	Error Message	Solution
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 range([%3, %4])	Database Internal error, cannot be 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	Instruction format error,

	for/while	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 declartion missing keyword %2	Instruction format error, check program
41301	%1: program declartion 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	Instruction format error,

	type %2	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 is forbidden, type =	Instruction format error,

	FUNCTION	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	Error in registration file, which cannot be solved after restarting. Contact technical support
61001	apps %1 and %2 have duplicate id = %3	Error in registration file, which cannot be solved after restarting. Contact technical support
61002	apps %1 and %2 accepts common command %3	Error in registration file, which cannot be solved after restarting. Contact technical support