System Operation Manual

Catalogue

System Operation Manual

> Safety Precautions and Product Assembly Instructions

> Safety Precautions

Robot owners and operators are responsible for their own safety, and iNexBot is not responsible for the safety of the use of robot. iNexBot reminds users that they must be aware of the need to use safety equipment when using the robot and must comply with the safety terms.

Note: Occasions when robots may not be used:

1.Burning environment

2.Explosive environment

3.Environment of radio interference

4.In water or other liquids

5.Transport people or animals

6.No climbing

7.Others

Safety operation procedures:

Ⅰ. Manual and jog robots

1.Please do not operate the teach pendant and operation panel with gloves

2.When jogging the robot, use a lower speed ratio to increase the chance of controlling the robot

3.Consider the robot's motion trend before pressing the jog button on the teach pendant

4.Consider in advance the trajectory that can avoid the robot's movement, and confirm that the route is free from interference

5.The area around the robot must be clean and free of oil, water and impurities

II . Production and operation

1.Before starting the operation, be sure to know all the tasks that the robot will perform according to the written programs

2.Be sure to know the location and status of all switches, sensors and control signals that will control the movement of the robot

3.Be sure to know the location of the E-stop button on the robot control cabinet and peripheral control devices and be prepared to use them in case of emergency

Warnings

Never assume that just because the robot is not moving means that the program is complete, because the robot is probably waiting for an input signal to keep moving.

Product Assembly

Teach pendant installation

The figures below show the interface at the end of the teach pendant cable, and the connection interface at the bottom of the control cabinet

Control cabinet installation

Installation environment

1.Ambient temperature: The ambient temperature has a great impact on the life of the controller, the operating ambient temperature of the controller is not allowed to exceed the allowable temperature range (-10℃~50℃)

2.Install the controller vertically on the surface of the flame-retardant object in the installation cabinet, and there should be enough space around to dissipate heat

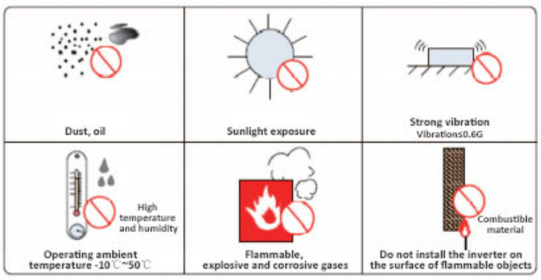
3.Please install it in a place that is not easy to vibrate. Vibration should not be greater than 0.6G. Pay special attention to keep away from equipments such as punches

4.Avoid installing in places with direct sunlight, humidity and water droplets

5.Avoid installing in places with corrosive, flammable and explosive gases in the air

6.Avoid installing in places with oil and dust. Installation site's pollution degree is PD2

7.NRC series products are installed in the cabinet and need to be installed in the final system for use. The final system should provide the corresponding fireproof enclosure, electrical enclosure and mechanical enclosure, etc., and comply with local laws and regulations and relevant IEC standard requirements, as shown in the figure



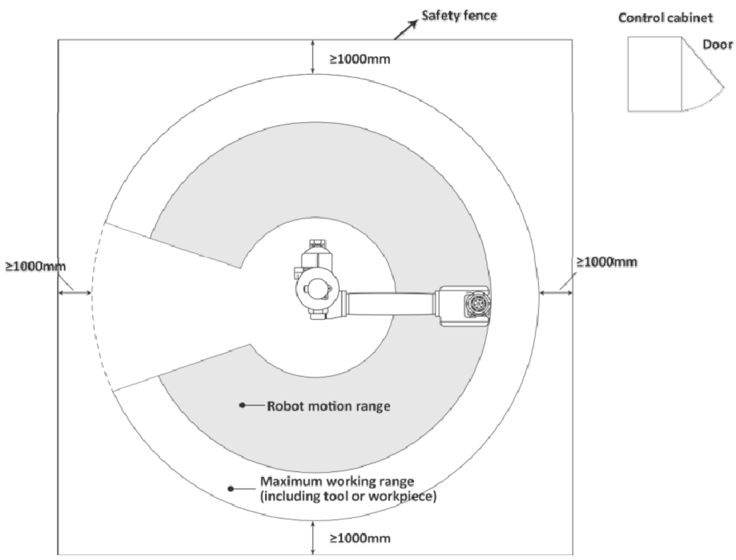
Installation location

1.The control cabinet should be installed outside the robot's motion range (outside the safety fence).

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

3.The control cabinet should be installed in a location where it is easy to open the door for inspection.

4.The control cabinet should be at least 500mm away from the wall to keep the maintenance channel unobstructed.



Cable requirements

Cable classification:

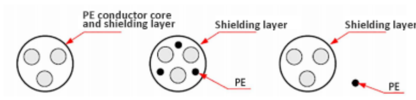
Level 1: sensitive signals (low-voltage analog signals, high-speed encoder signals, high-speed communication signals, ±10V analog signals, low-speed 422&485 signals, digital input and output signals)

Level 2: interference signals (low-voltage power supply, contactor control line, motor line with recorder, high-voltage AC power line, motor line without recorder)

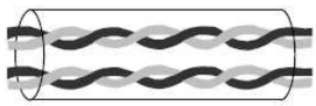
1.In the process of cable selection, it is recommended to use symmetrical shielded cables for input and output main circuit cables. Compared with four-core cables, the use of symmetrical shielded cables can reduce the electromagnetic radiation of the entire conduction system

2.Recommended power cable type - symmetrical shielded cable

Recommended signal cable type - shielded twisted pair cable



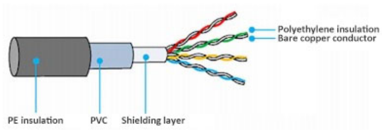
Schematic diagram of symmetrical shielded cable



Schematic diagram of shielded twisted pair cable

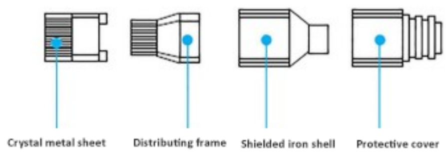
Note: Shielded twisted pair cable is recommended for digital signal lines

Recommended communication cable type - shielded communication cable, as shown in the figure



Schematic diagram of shielded communication cable

Note: The crystal head used must has a shielding metal shell. The shielding layer of the communication cable and the shielding iron shell of the crystal head are crimped together, as shown in the figure.



Schematic diagram of crystal head with shielding metal shell

Wiring requirements

1.Power cables should be routed away from all signal cables.

2.Motor cables, input power cables and control circuit cables should not be routed in the same raceway as much as possible.

3.Avoid electromagnetic interference caused by coupling when the motor cable and the control circuit are routing in parallel for a long distance.

4.Keep a minimum distance of 100mm between cables of different grades in the same raceway.

Note:

1.Cables of different grades are arranged separately. When long-distance cables are routed in the same direction, a distance of at least 100mm should be maintained between cables of different grades

2.Use the conductor as the backplane (using a zinc plate that has not been sprayed) and connect the metal part of the controller directly to the backplane

3.Keep the cables separated according to the grade, and if cables of different grades must be crossed, they should be kept 90° crossed

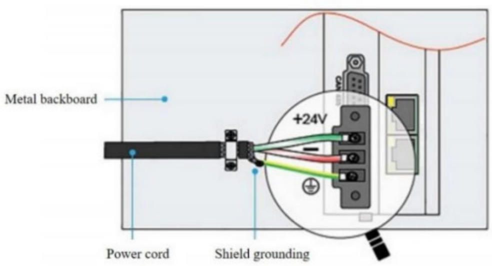
Grounding requirements

Warnings

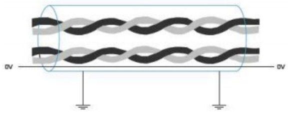
⚠

Be sure to ground the ground terminal, otherwise there may be the danger of electric shock or malfunction due to interference.

Power cable grounding requirements, as shown in the figure



The differential signal line (CAN/RS485/RS422) adopts shielded twisted pair cable, and the shielding layer must be connected to 0V at both ends of the cable, as shown in the figure



Wiring notes

1.Personnel participating in wiring and inspection must be professionals with corresponding skills.

2.The product must be grounded reliably, the grounding resistance should be less than 4 ohms, and the neutral wire (zero wire) cannot be used instead of the ground wire.

3.Wiring must be correct and secure to avoid product failure or unintended consequences.

4.The surge absorbing diode connected to the product must be connected in the specified direction, otherwise the product will be damaged.

5.Before plugging or unplugging or opening the product chassis, the product must be disconnected from the power supply.

6.Try to avoid the signal line and the power line going through the same pipe, the distance should be more than 30mm.

7.For the signal line and encoder (PG) feedback line, please use multi-stranded stranded wire and multi-core stranded shielded wire. For the wiring length, the maximum length is 3m for the instruction input line and 20m for the PG feedback line. The signal line of the encoder is a set of twisted pair wires, the power line is a set of twisted pair wires, and the battery line is a set of twisted pair wires.

8.Do not turn the power on/off frequently. If you need to turn the power on/off repeatedly and continuously, limit it to less than one time in one minute. Since the power supply part of the servo unit has capacitors, frequent ON/OFF may cause degradation of the performance of the main circuit components inside the servo unit.

9.Confirm the power and voltage of the switching power supply in the control system. Ensure that the power of the controller, teach pendant and IO module is not less than 50W, the specific power supply power depends on the IO module load size.

10.It is recommended to use the servo switching power supply separately from the switching power supply of the controller system to prevent the servo from interfering with the control system.

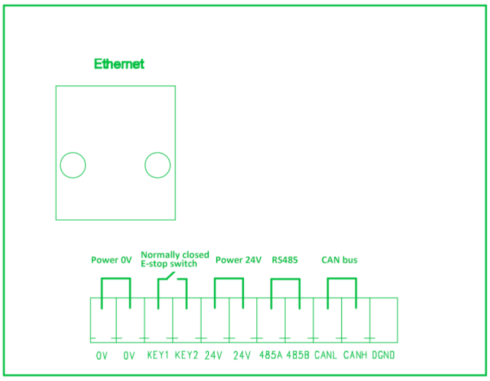
Note:

1.The network cable connecting the control system and the servo needs to use the super six shielded network cable

2.If one axis corresponds to one servo, the network cables need to be connected in the order of the axes

3.Please follow the order of controller-servo-IO board when wiring

Teach Pendant Adapter Box Wiring Definition Diagram



Teach Pendant Buttons and Interface Introduction

T30 teach pendant physical buttons

Left side

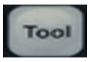
	Switch current servo status

Robot	Switch current robot (only available in multi-robot mode)
EXT	Switch between current robot and external axes (only available when there are external axes)
Home	Click the button to return to zero point
PSP	Click the button to return to reset point
Clear	Clear the error after the servo reports an error

	Switch drag method (reserved)

Downside

	Step through a program sequentially or in reverse order in teach mode
	Step through a program in teach mode
	Reduce teaching or running speed
	Increase teaching or running speed

	
	Switch tool hand
	Switch between four coordinate systems

Right side

	Pause the program in run mode
	Start the program in run mode

	The corresponding axis runs in the reverse direction when teaching
	The corresponding axis runs in the positive direction when teaching

Key switch

	Left, switch to teach mode Middle, switch to auto mode Right, switch to remote mode

E-stop button

	Press the button for emergency stop

Wheel knob

	Switch to the previous line and the next line by rotating the knob in the program interface

Deadman button

	Three-stage button Press to the middle to power on the robot Press to the bottom to power off the robot Release the button to power off the robot

Operating System Introduction

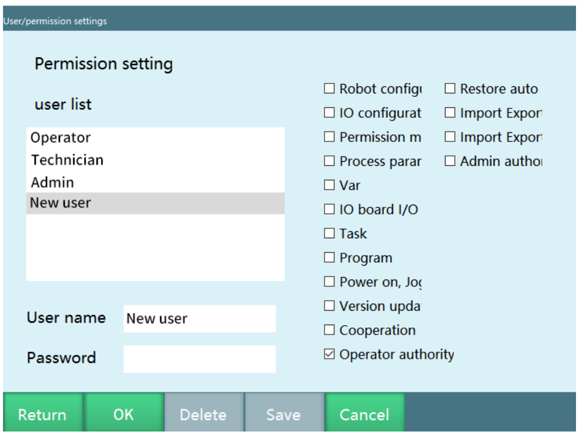
Basic instructions

The left side of the interface shows the function keys

Admin	Open the admin/technician/ operator switch interface
Settings	Open the robot function setting interface
Function	Open the robot process selection interface
Var	Open the robot variable setting interface
Status	Open the robot status view interface
Project	Open the project preview interface
Job	Open the program instruction interface
Log	Open the error log interface
Monitor	Open the robot monitor display interface
12:30 Thursday 2016/08/30	Date and time display

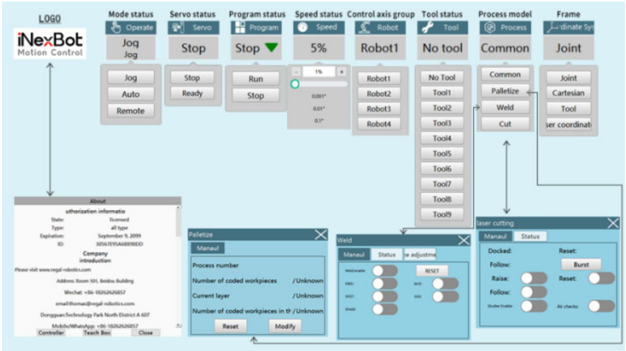
Permission settings:

Switch user to "Admin", select [Permission settings], create a new user, and customize the permissions



Status introduction

The status bar at the top of the program shows the various states of the robot



Mode status: teach mode, remote mode, run mode; you can switch the mode by rotating the external knob

Servo status : stop, ready, run, alarm

1.Switch between "stop" and "ready" status: Press the left "Servo" button

2.Switch from "ready" to "run" status:

Press the enable button in teach mode

Press the "Start" button in run mode

Give start signal in remote mode

3.If you press the [E-stop button] on the control cabinet/teach pendant, the servo status will switch to "alarm"

Notes

The E-stop button needs to be connected to the servo

Program status: run, stop

Run status:

1.When stepping through the program in "Teach mode"

2.When running the program in "Run mode" or "Remote mode", the program status switches to "run"

Jog speed: 0.001°, 0.01°, 0.1°, 1%, 5%, 10%, 15%, 20%, 25%, 30%, 35%, 40%, 45%, 50%, 55%, 60%, 65%, 70%, 75%, 80%, 85%, 90%, 95%, 100%

Adjust teaching or running speed by pressing [V+]&[V-] at the bottom of the teach pendant

Note: 0.01mm, 0.1mm, 1mm in Cartesian coordinate system & tool coordinate system

Robot status: " Robot 1", " Robot 2", " Robot 3", " Robot 4"

Switch the robot by pressing the [Robot] button on the left of the teach pendant

Note: This system supports only up to four robots

Tool status: "Tool 1", "Tool 2", "Tool 3", "Tool 4", "Tool 5", "Tool 6", "Tool 7", "Tool 8", "Tool 9", "No tool"

Switch the tool by pressing the [Tool] button at the bottom of the teach pendant

Process mode : "General", "Welding", "Palletizing", "Cutting", "Stamping"

1."General", "Welding", "Palletizing", "Cutting": make pop-up call through the process in the upper right corner

2."Stamping process": switch through [Settings - Operation parameters - Process selection], and directly change the operation interface

Coordinate system: "Joint", "Cartesian", "Tool", "User"

Switch the coordinate system by pressing the [Coord] button on the left side of the teach pendant

Robot Coordinate Systems and Axis Operations

Control groups and coordinate systems

Coordinate systems

For axis operations on the robot body, the coordinate system has the following forms:

Joint coordinate system:

Each joint axis of the robot moves independently. When a single axis is jogged under joint coordinate system, the robot coordinates of the jogged axis on the "Monitor-Robot coordinates" interface will change.

Cartesian coordinate system:

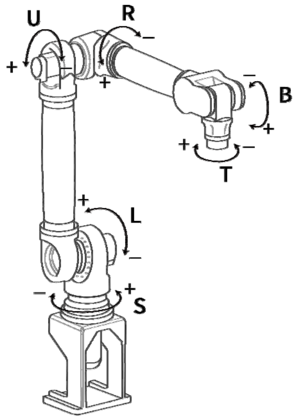
The front end of the robot moves in parallel along the X, Y and Z axes of the base. A, B and C rotate around the X, Y and Z axes respectively. The Euler angle rotational sequence used in this system is X'Y'Z' and the fixed angle rotational sequence is ZYX.

Tool coordinate system:

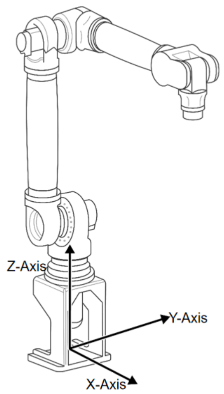
The tool coordinate system takes the effective direction of the robot's wrist tool as the Z-axis, defines the origin of the coordinate system at the tip point of the tool, and the tip point of the body moves in parallel according to the coordinates. TA, TB and TC rotate around the TX, TY, TZ axes respectively.

User coordinate system:

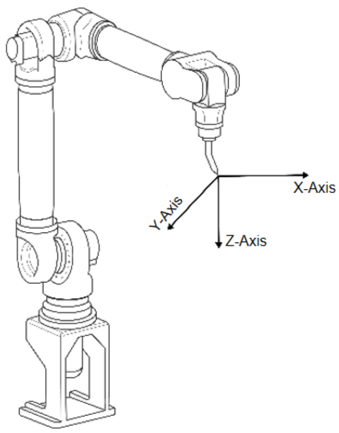
XYZ Cartesian coordinates are defined anywhere. The body tip point moves in parallel according to the coordinates.



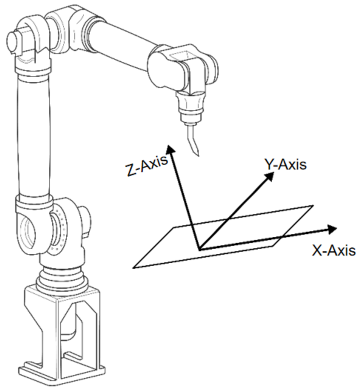
Joint coordinate system



Cartesian coordinate system



Tool coordinate system

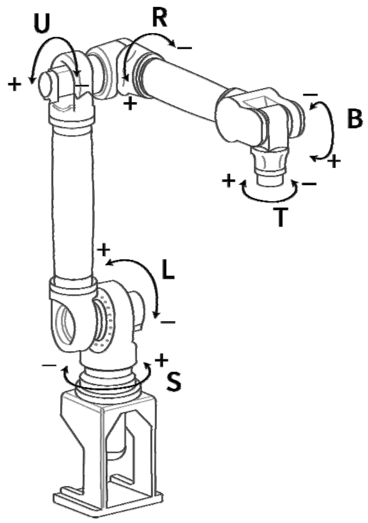


User coordinate system

Coordinate systems and axis operations

Joint coordinate system

In the joint coordinate system, each axis of the robot can operate independently.

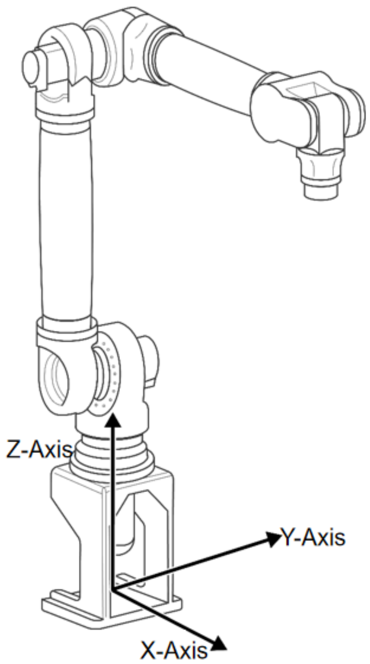


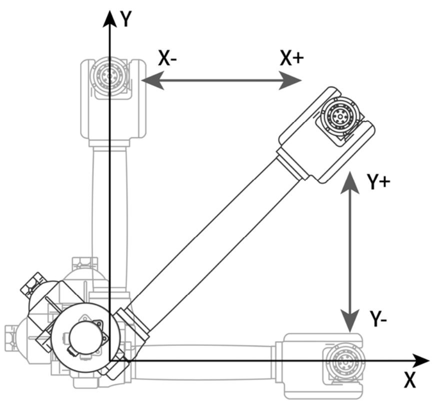
Axis operations in joint coordinate system

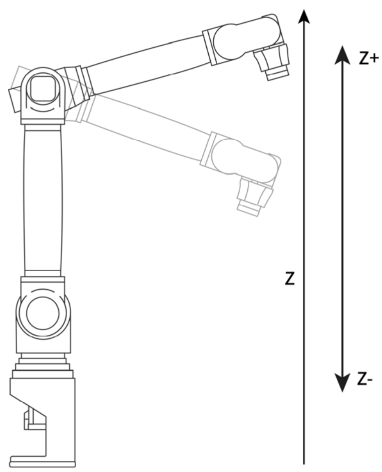
Axis name		Axis operation	Action
Basic axis	S axis	S+/S-	Body rotates left and right
	L axis	L+/L-	Lower arm moves forward and backward
	U axis	U+/U-	Upper arm moves up and down
Wrist axis	R axis	R+/R-	Wrist rotates
	B axis	B+/B-	Wrist moves up and down
	T axis	T+/T-	Wrist rotates

Cartesian coordinate system

In the Cartesian coordinate system, the robot moves parallel to the X, Y and Z body axes, as shown in the figure below.


Z-Axis
Y-Axis
X-Axis


Y
X-
X+
Y+
Y-
X


Z+
Z
Z-

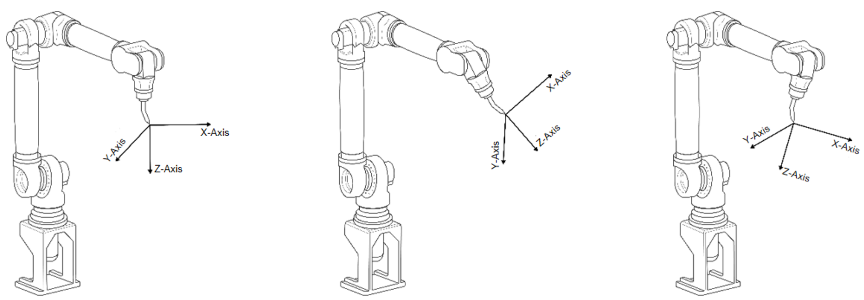
Axis operations in Cartesian coordinate system

Axis name		Axis operation	Action
Basic axis	X axis	X+/X-	Move in parallel along the X axis
	Y axis	Y+/Y-	Move in parallel along the Y axis
	Z axis	Z+/Z-	Move in parallel along the Z axis
Attitude axis	A axis	A+/A-	Rotate around the X axis
	B axis	B+/B-	Rotate around the Y axis
	C axis	C+/C-	Rotate around the Z axis

Tool coordinate system

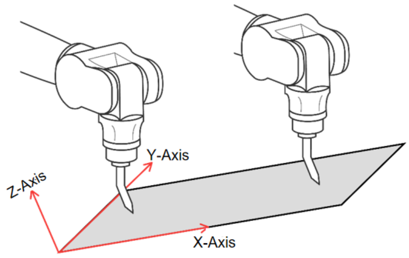
In the tool coordinate system, the robot moves in parallel along the X, Z and Y axes defined at the the tool tip point.

The tool coordinate system takes the effective direction of the tool installed on the robot wrist flange as the Z axis, and defines the coordinates at the tool tip point, so the orientation of the tool coordinate axis changes with the movement of the wrist, as shown in the figure below.



The movement of the tool coordinates is not affected by changes in robot position or posture and is primarily based on the effective direction of the tool.

Therefore, tool coordinate movements are best suited to applications where the tool posture is always constant and moving parallel to the workpiece, as shown below.

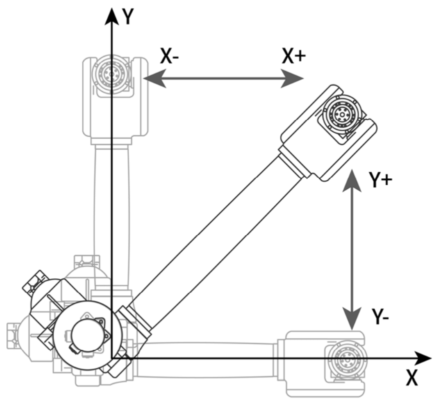


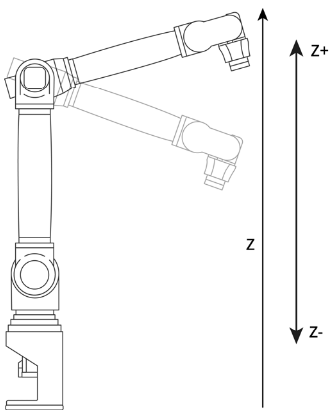
Axis operations in tool coordinate system

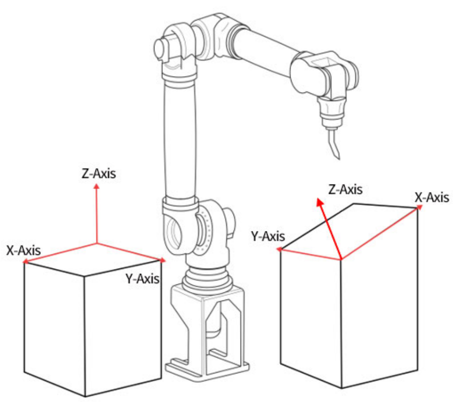
Axis name		Axis operation	Action
Basic axis	TX axis	TX+/TX-	Move in parallel along the TX axis
	TY axis	TY+/TY-	Move in parallel along the TY axis
	TZ axis	TZ+/TZ-	Move in parallel along the TZ axis
Attitude axis	TA axis	TA+/TA-	Rotate around TX axis
	TB axis	TB+/TB-	Rotate around TY axis
	TC axis	TC+/TC-	Rotate around TZ axis

User coordinate system

In the user coordinate system, the X, Y and Z axes are set at any position in the robot's range of motion at any angle, and the robot moves parallel to these set axes, as shown below.







Axis operations in user coordinate system

Axis name		Axis operation	Action
Basic axis	UX axis	UX+/UX-	Move in parallel along the UX axis
	UY axis	UY+/UY-	Move in parallel along the UY axis
	UZ axis	UZ+/UZ-	Move in parallel along the UZ axis
Attitude axis	UA axis	UA+/UA-	Rotate around UX axis
	UB axis	UB+/UB-	Rotate around UY axis
	UC axis	UC+/UC-	Rotate around UZ axis

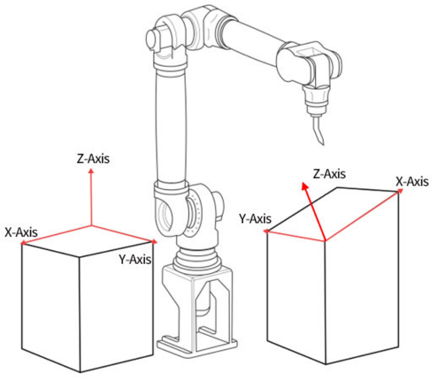
Use case of user coordinate system

The use of the user coordinate system makes various teaching operations easier.

Here, we will illustrate this with a few examples.

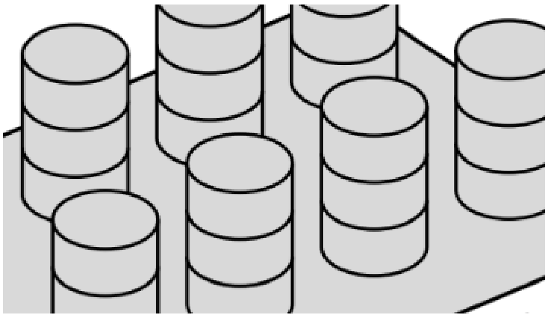
1.When there are multiple fixture tables:

Manual operations can be made easier by using the user coordinates set for each fixture table.



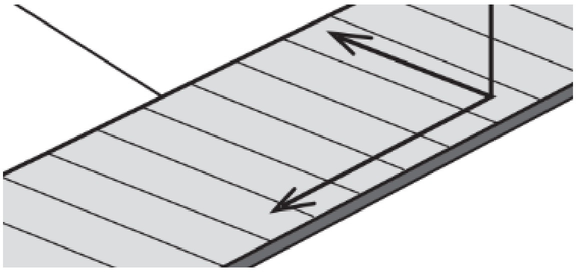
2.When engaged in arranging and stacking operations:

Perform user left calibration, if the user coordinates are set on the pallet, it becomes easier to set the displacement increment during parallel movement.



3.When running synchronously with the conveyor belt:

In the conveyor belt process, it is necessary to calibrate the user coordinates and specify the movement direction of the conveyor belt.



External axis

Use the [External axis] button to switch to the external axis, then you can jog and teach the external axis; the external axis only supports joint jog operations.

Axis name	Axis operation	Action
O1 axis	J1+/J1-	External axis 1 rotates
O2 axis	J2+/J2-	External axis 2 rotates
O3 axis	J3+/J3-	External axis 3 rotates
O4 axis	J4+/J4-	External axis 4 rotates
O5 axis	J5+/J5-	External axis 5 rotates

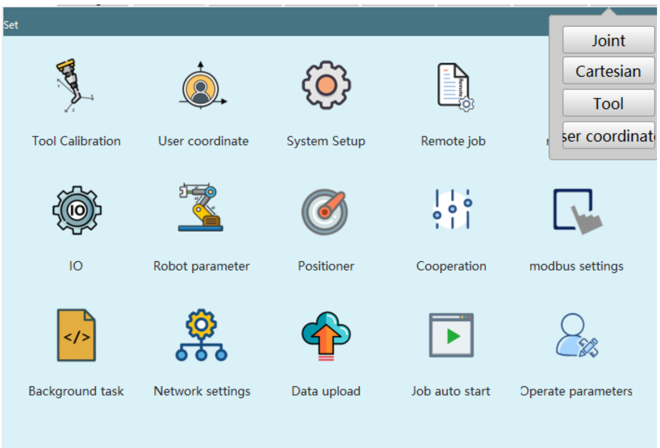
Coordinate system description and switching

There are four coordinate systems in this product, namely joint coordinate system, Cartesian coordinate system, tool coordinate system and user coordinate system.

- All points in the joint coordinate system are the angle values of the robot joint axis relative to the mechanical zero point of the axis;
- The Cartesian coordinate system is also called the "base coordinate system", and all its points are the coordinate values (unit mm) of the robot end (flange center) relative to the center of the robot base;
- All points in the tool coordinate system are the coordinate values (unit mm) of the end (TCP) of the tool carried by the robot relative to the center of the robot base. For its definition and usage, please refer to the chapter of "Tool hand and user coordinates";
- The user coordinate system is also called "workpiece coordinate system", and all its points are the coordinate values (unit mm) of the end of the tool carried by the robot (the center of the flange when no tool is attached) relative to the origin of the user coordinate system. For its definition and usage, please refer to the chapter of "Tool hand and user coordinates".

Teach mode

Press the [Coordinate] button in the physical button area at the bottom of the teach pendant. Each time you press this button, the coordinate system switches in the following order, you can confirm this by the display in the status bar at the top. You can also click on the coordinate system column in the status bar to bring up the coordinate system selection menu, and click on the corresponding coordinate system to switch between Joint → Cartesian → Tool → User, as shown below

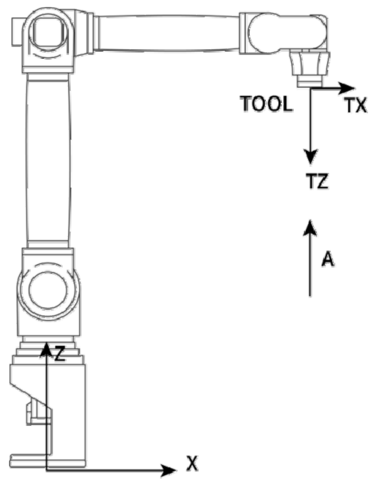


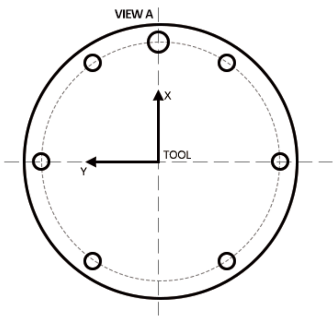
Tool Hand and User Coordinates

Tool hand calibration

Tool coordinate system

Center of flange: the origin of the default tool coordinate system; the direction in which the center of the flange points towards the flange locating hole is the +X direction, the direction perpendicular to the flange and outwards is the +Z direction and finally the Y direction can be determined by the right hand rule. The new tool coordinate system is a change from the default tool coordinate system.





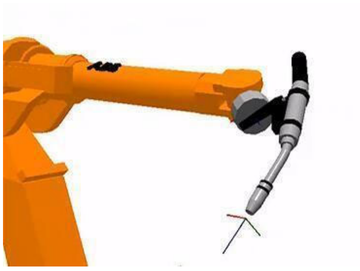
TCP: TOOL CENTER POINT

Robot trajectory and speed: the trajectory and speed of TCP points.

The TCP is generally set in the center of the gripper, at the end of the wire, at the front end of the spot welding static arm, etc.

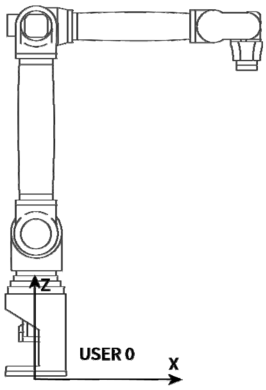
In order to describe the position of an object in space, it is necessary to fix a coordinate system on the object, and then determine the pose of the coordinate system (origin position and three coordinate axis attitudes), i.e., seven DOFs (degrees of freedom) are needed to completely describe the pose of the rigid body. For industrial robots, a tool (Tool) needs to be mounted on the end flange to perform the operation. In order to determine the pose of the tool (Tool), it is necessary to bind a tool coordinate system (TCS) to the Tool, the origin of the TCS is the TCP (Tool Center Point). When programming the robot trajectory, it is necessary to record the pose of the TCS in other coordinate systems into the program for execution.

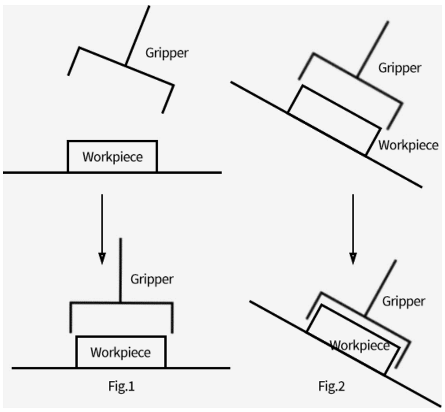
Industrial robots generally have a TCS defined in advance, with the XY plane of the TCS bound to the flange plane of the robot's sixth axis, and the origin of the TCS coinciding with the center of the flange. Obviously the TCP is in the center of the flange. The ABB robot calls the TCP tool0 and the REIS robot calls it _tnull. Although the default TCP can be used directly, in practice, for example, when welding, the user usually defines the TCP point to be the tip of the wire (actually the pose of the coordinate system of the torch tool in the tool0 coordinate system), then the position recorded in the program is the position of the tip of the wire, and the attitude recorded is the attitude of the torch as it rotates around the tip of the wire.



Thinking:

We know that the tool coordinate system is an object of study in motion, but what role does it play in the actual debugging process? Think about how the attitude and position of the gripper in Figure 1 and Figure 2 are obtained through adjustment?





Two conjectures can be drawn from the thinking:

Conjecture 1: If the gripper in Figure 1 has a rotation point, then the gripper can select the workpiece directly around this rotation point.

Conjecture 2: If the gripper in Figure 2 can move to the workpiece in a forward direction, then the gripper will move directly to the workpiece.

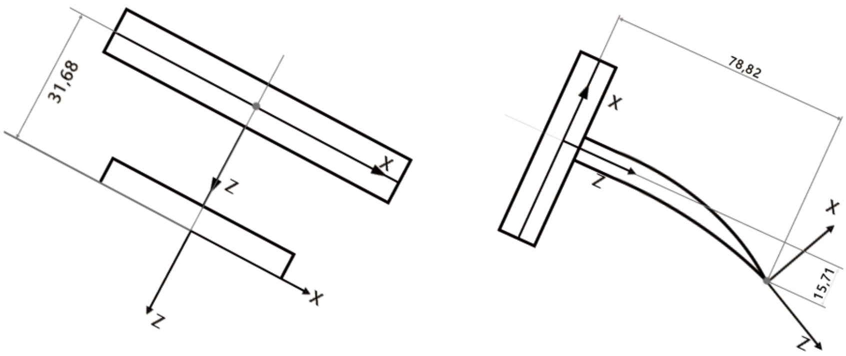
Conclusion: The role of establishing the tool coordinate system:

1.Determine the TCP point (i.e. tool center point) of the tool to facilitate the adjustment of the tool state.

2.Determine the tool feed direction to facilitate the adjustment of the tool position.

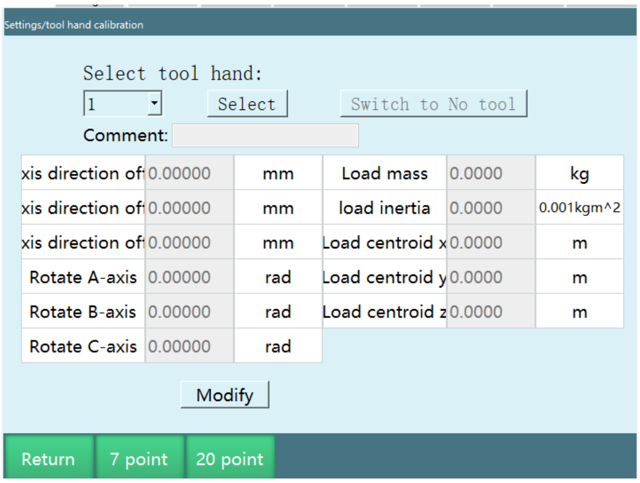
Tool coordinate system characteristics

The new tool coordinate system is a change from the default tool coordinate system. The position and orientation of the new tool coordinate system always maintain the absolute position and attitude relationship with the flange, but it is always changing in space.



Tool hand parameter setting

Click "Settings - Tool hand calibration" to enter the "Tool hand calibration" interface, as shown in the following figure



If there are detailed parameters of the tool, in this interface, the user can directly fill in the relevant parameters of the tool end offset, without the need for 7-point calibration.

When entering this interface, the saved tool hand size parameters in the controller will be read automatically (each item is 0 by default), if you change the tool hand, please fill in again.

Detailed parameter setting steps are as follows:

1.Open the "Tool hand calibration" interface, the following table is the introduction of each parameter:

Parameter	Function
X-axis offset	Length of offset of the tool end relative to the center of the flange along the X-axis of the Cartesian coordinate system (mm).
Y-axis offset	Length of offset of the tool end relative to the center of the flange along the Y-axis of the Cartesian coordinate system (mm).
Z-axis offset	Length of offset of the tool end relative to the center of the flange along the Z-axis of the Cartesian coordinate system (mm)
Rotate around A-axis	The rotation angle of the tool end relative to the center of the flange around the X-axis of the Cartesian coordinate system (°)
Rotate around	The rotation angle of the tool end relative to the center of the

B-axis	flange around the Y-axis of the Cartesian coordinate system (°)
Rotate around C-axis	The rotation angle of the tool end relative to the center of the flange around the Z-axis of the Cartesian coordinate system (°)

2.Click on the [Modify] button.

3.Fill in the parameters corresponding to the tool, the function of each parameter is shown in the table above.

4.Confirm that there is no error and click the [Save] button to set successfully.

Warnings

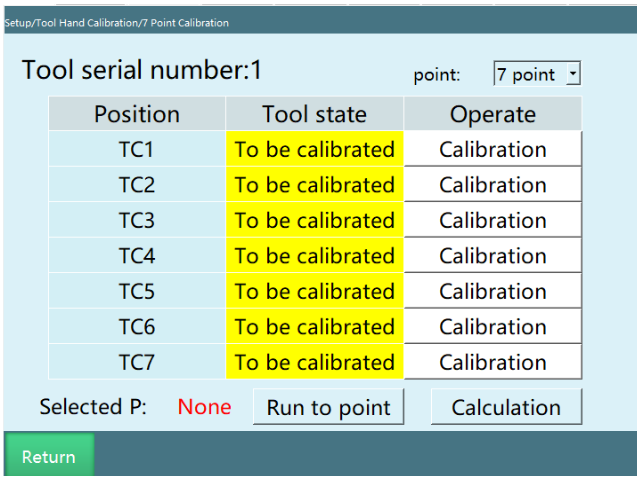
Keep the flange parallel to the horizontal plane before measuring the data

Click the [Clear] button to clear the filled parameters.

If you click the [Return] or [7-point calibration] button in the bottom operation area during parameter setting, it will jump to the corresponding interface, and the unsaved setting parameters will not be retained.

7-point calibration

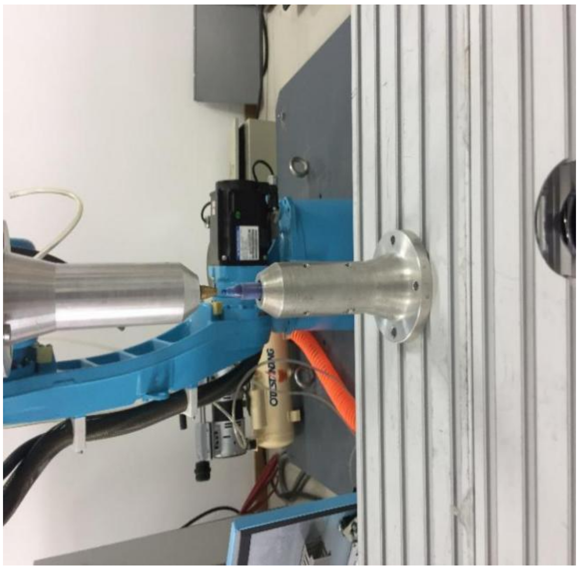
Click the [7-point calibration] button at the bottom to enter the "7-point calibration" interface, as shown in the figure



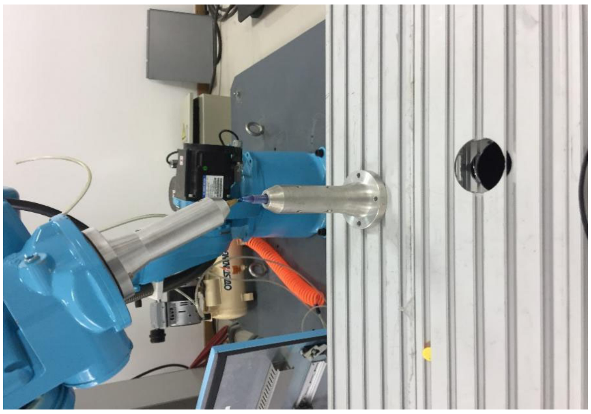
If the detailed parameters of the tool are not available, TCP calibration can be performed to automatically calculate each dimensional parameter of the tool. The specific calibration steps are as follows:

1.Now use the pen tip as a reference point and make sure this reference point is fixed, as shown in the figure below.

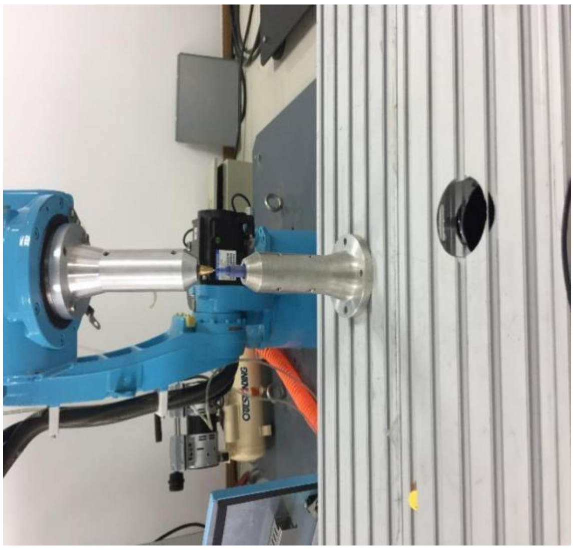
2.With the tool end vertical and facing the reference point, click the [Calibrate] button corresponding to "TC1" in the interface, as shown in the figure below.



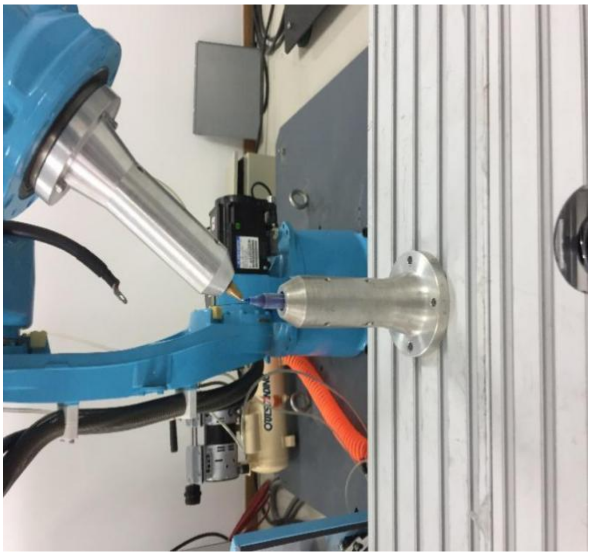
3.TC2 calibration: Switch the robot to a posture with the tool end facing the reference point, and click the [Calibrate] button corresponding to that line, as shown in the figure below.



4.TC3 calibration: Switch the robot to a posture with the tool end facing the reference point, and click the [Calibrate] button corresponding to that line, as shown in the figure below.



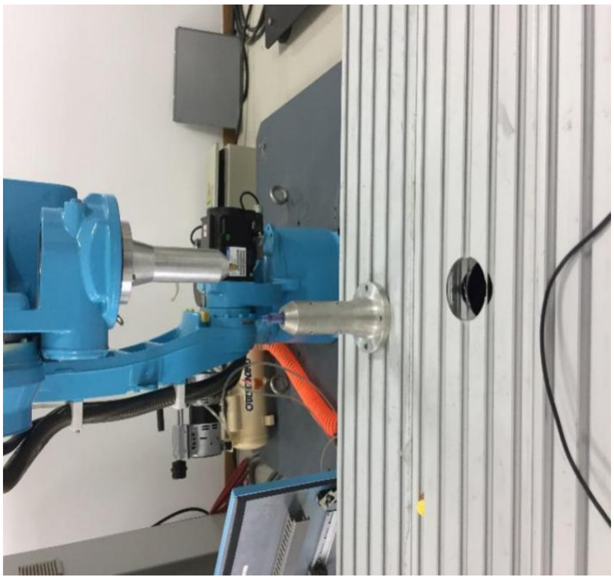
5.TC4 calibration: Switch the robot to a posture with the tool end facing the reference point, and click the [Calibrate] button corresponding to that line, as shown in the figure below.



6.TC5 calibration: With the tool end vertical and facing the reference point (same as TC1), click the [Calibrate] button corresponding to that line, as shown in the figure below.

7.TC6 calibration: On the basis of TC5, move any distance along the negative direction of the X-axis of the Cartesian coordinate system, and click the [Calibrate] button corresponding to that line, as shown in the figure below.

8.TC7 calibration: On the basis of TC6, move any distance along the positive direction of the Y-axis of the Cartesian coordinate system, and click the [Calibrate] button corresponding to that line, as shown in the figure below.



9.Click [Run to this point] to see if the calibration is accurate.

10.Click the [Calculate] button, the calibration is successful.

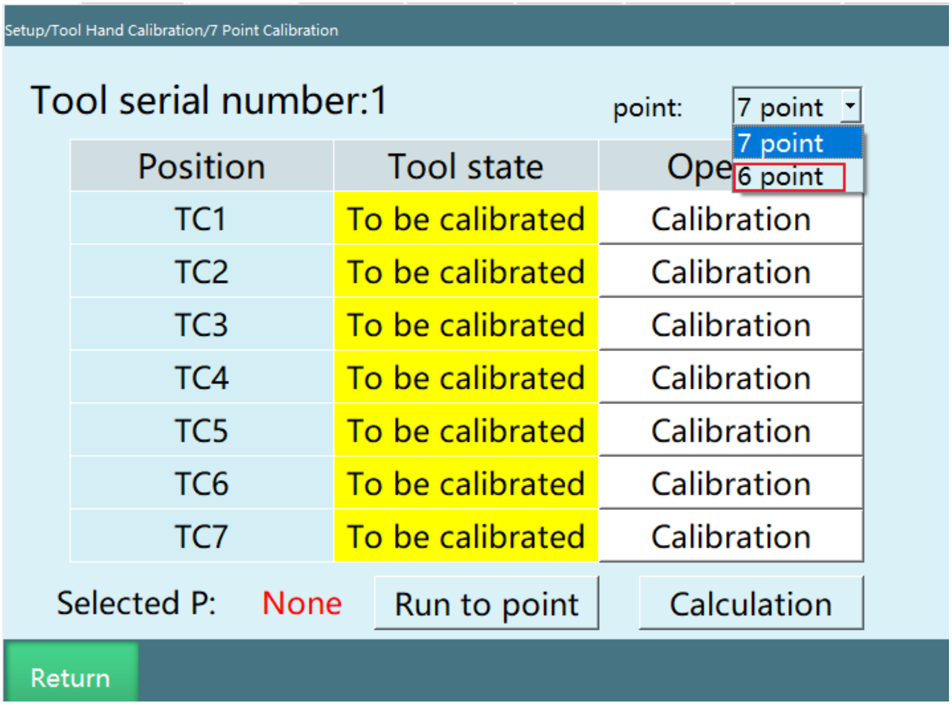
If you are not satisfied with a point that has been calibrated during the calibration process, you can click the [Cancel calibration] button corresponding to that line to cancel the calibration and then calibrate the point again.

Click the [Demo] button at the bottom to open the "Demo" interface, which explains how to perform the tool calibration.

Click the [Return] button at the bottom to return to the "Tool hand calibration" interface.

6-point calibration

Enter the "Settings - Tool hand calibration - 7-point calibration" interface, you can choose "6-point calibration" for the "Calibration method", as shown in the figure below.



Calibration method:

Point 1: The robot's J5 is vertically down.

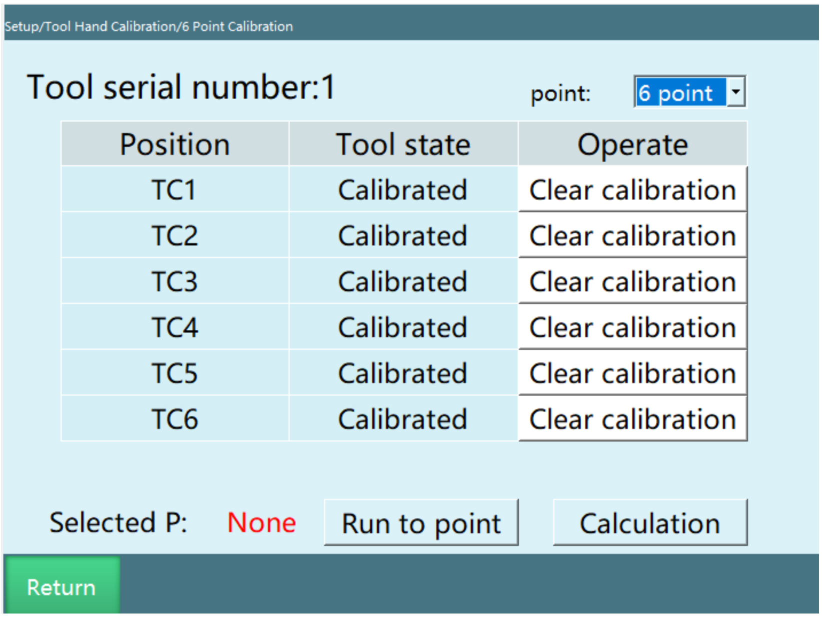
Point 2: The robot rotates 180° around the C-axis on the basis of the first point

Point 3: The robot rotates 35°around the B-axis on the basis of the first point

Point 4: The robot returns to zero point with the tool hand end vertical

Point 5: The robot performs X- movement on the basis of the fourth point

Point 6: The robot performs Y+ movement on the basis of the fifth point



1.After the 6-point calibration is completed, select any point that has been calibrated, and click [Run to this point] to check whether the calibration is accurate.

2.Click the [Calculate] button, the calibration is successful. Click the [Return] button at the bottom to return to the "Tool hand calibration" interface, rotate around ABC to verify the calibration error.

3.If you are not satisfied with a point that has been calibrated during the calibration process, you can click the [Cancel calibration] button corresponding to that line to cancel the calibration and then calibrate the point again.

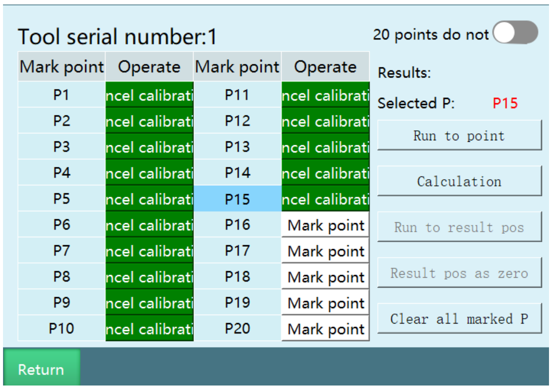
4.Click the [Return] button at the bottom to return to the "Tool hand calibration" interface.

12/15-point calibration

The 12/15/20-point calibration shares a calibration interface, and calibrating the first 15 points means using the 15-point calibration method.

The 12-point calibration means that the 15-point calibration does not mark the last three points (13-15). The calibration result is only the offset of the XYZ axis of the tool hand, and there is no value of rotation around ABC.

Click the [20-point calibration] button at the bottom of the "Tool hand calibration" interface to enter the calibration interface, as shown in the figure.

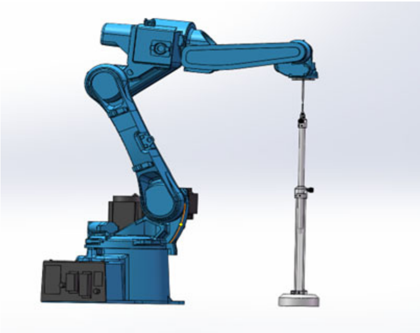


1.Find a reference point (the tip of the calibration cone is the reference point) and make sure this reference point is fixed.

2.Start inserting position points, click [Mark this point] for each point inserted, and insert fifteen points.

The specific steps are as follows:

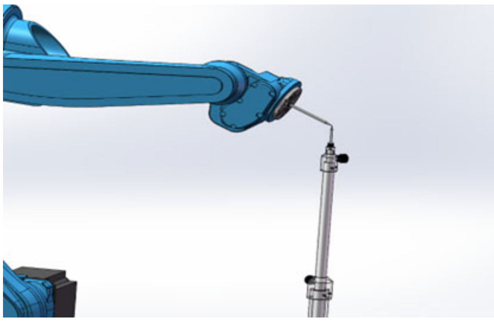
1.Point 1: The robot returns to the zero point, align the tip of the robot with the tip of the calibration cone through Cartesian coordinate system, and calibrate the first point;



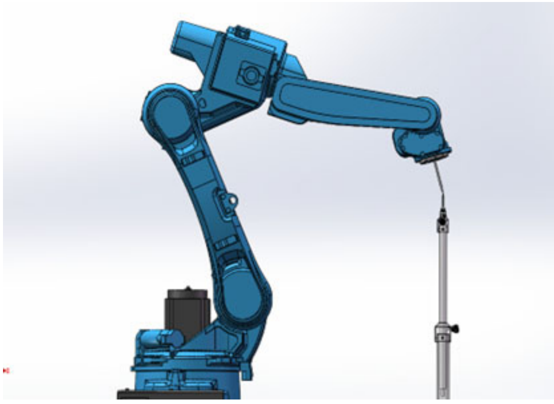
2.Point 2: On the basis of the first point, the robot rotates 180 degrees around the C-axis through the Cartesian coordinate system, align the tip of the robot with the tip of the calibration cone, and calibrate the second point;

3.Point 3: The robot returns to the zero point, align the tip of the robot with the tip of the calibration cone through the Cartesian coordinate system, and calibrate the third point; (same as the first point)

4.Point 4: On the basis of the third point, perform B- movement through the Cartesian coordinate system with rotation angle between 30°and 60°, align the tip of the robot with the tip of the calibration cone, and calibrate the fourth point;

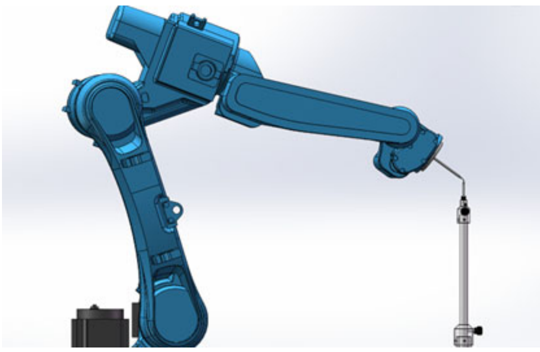


5.Point 5: On the basis of the fourth point, perform B+ movement through the Cartesian coordinate system, make J5>-90°, align the tip of the robot with the tip of the calibration cone, and calibrate the fifth point;

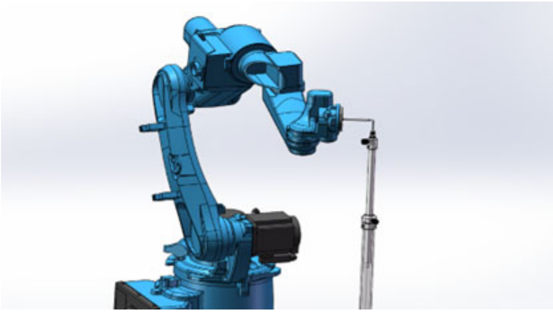


6.Point 6: Select the first point and move the robot to the first point, and on the basis of the first point, perform B+ movement through the Cartesian coordinate system, make J5>-90°, align the tip of the robot with the tip of the calibration cone, and calibrate the sixth point;

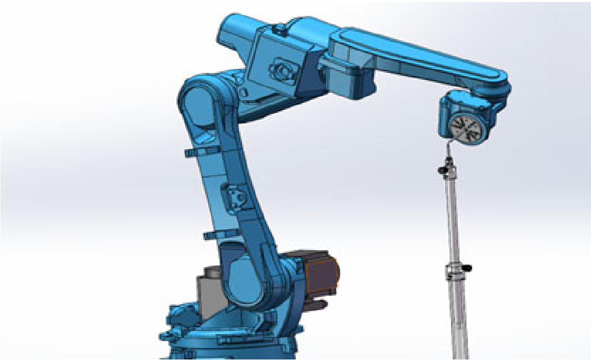
7.Point 7: On the basis of the first point, perform B- movement through the Cartesian coordinate system, make J5>-90°, align the tip of the robot with the tip of the calibration cone, and calibrate the seventh point;



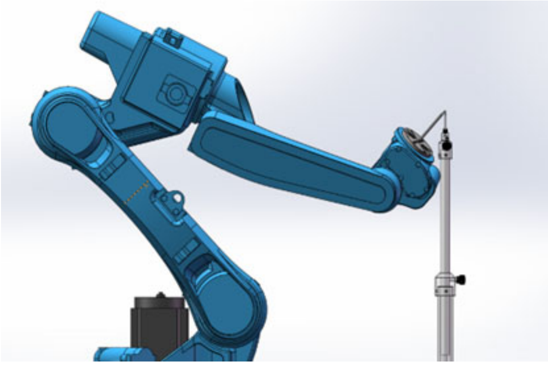
8.Point 8: On the basis of the seventh point, perform A+ movement through the Cartesian coordinate system, rotate by 90°and make J5>-90°, align the tip of the robot with the tip of the calibration cone, and calibrate the eighth point;



9.Point 9: On the basis of the seventh point, perform A- movement through the Cartesian coordinate system, rotate by 90°and make J5>-90°, align the tip of the robot with the tip of the calibration cone, and calibrate the ninth point;

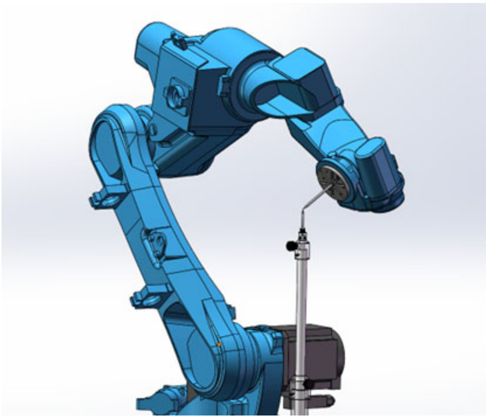


10.Point 10: The robot returns to the first point, jog axis 5 through the joint coordinate system to make axis 5 up and J5<-90°, align the tip of the robot with the tip of the calibration cone, and calibrate the tenth point;



11.Point 11: On the basis of the tenth point, the robot performs A+ movement through the Cartesian coordinate system, rotate by 90°and make J5<-90°, align the tip of the robot with the tip of the calibration cone, and calibrate the eleventh point;

12.Point 12: On the basis of the tenth point, the robot performs A- movement through the Cartesian coordinate system, rotate by 90°and make J5<-90°, align the tip of the robot with the tip of the calibration cone, and calibrate the twelfth point;



13.Point 13: The robot returns to the zero position, adjust the robot attitude so that the end tool tip of the robot is facing downwards, align the tip of the robot with the tip of the calibration cone, and calibrate the thirteenth point;

14.Point 14: On the basis of the thirteenth point, perform X- movement through the Cartesian coordinate system, move the robot by a distance, click directly to calibrate the fourteenth point;

15.Point 15: On the basis of the fourteenth point, perform Y+ movement through the Cartesian coordinate system, move the robot by a distance, and click directly to calibrate the fifteenth point.

16.Click [Calculate] when you finish marking.

[Cancel calibration]: If you are not satisfied with a point that has been calibrated during the calibration process, you can click the [Cancel calibration] button corresponding to that line to cancel the calibration and then calibrate the point again.

[Run to this point]: You can click [Run to this point] after each point is calibrated, then the robot will run to that point.

[Mark the result position as zero]: Set the position after calibration compensation as the current robot's zero position.

[Clear all mark points]: The calibration points will be saved in the controller, and the calibration results will be cleared only after clicking "Cancel calibration", "Clear all mark points", and switching tool hands to enter the calibration interface.

Notes

For the posture of each point, please try to select the posture in any direction. If the posture selected is rotated in a certain direction, the accuracy is sometimes inaccurate.

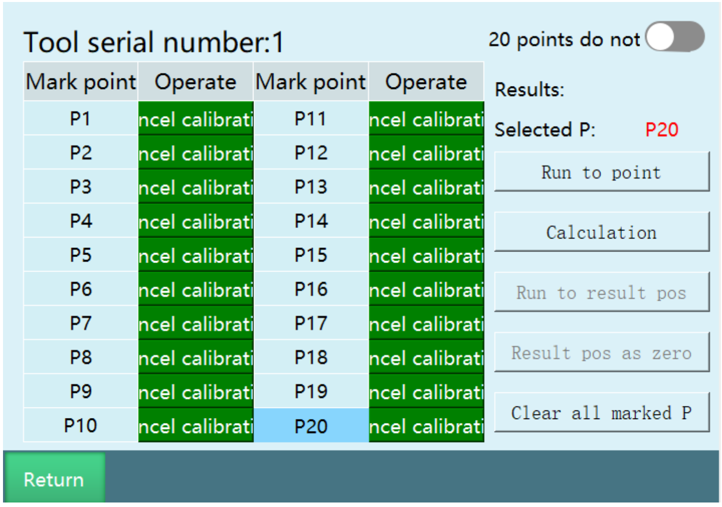
Please keep the reference point fixed during the calibration process, otherwise the calibration error will increase.

Click the [Return] button at the bottom to return to the "Tool hand calibration" interface.

20-point calibration

The 12/15/20-point calibration share a calibration interface, and calibrating all 20 points means using the 20-point calibration method.

Click the [20-point calibration] button at the bottom of the "Tool hand calibration" interface to enter the "20-point calibration" interface, as shown in the figure.



1.Find a reference point (the pen tip is the reference point) and make sure this reference point is fixed.

2.Start inserting position points, click [Mark this point] for each point inserted, and insert 20 points, the greater the difference between the poses of each point, the better.

Manufacturers recommended calibration steps: point 1: tool hand vertical down; point 2: go A+; point 3: go A+; point 4: go A+; point 5: go A-; point 6: go A-; point 7: go A-; point 8: go B+; point 9: go B+; point 10: go B+; point 11: go B-; point 12: go B-; point 13: go B-, the rest points are mainly calibrated by moving the robot around C axis to make a metre-shaped arrangement

The specific calibration steps are as follows:

Point 1: Make the robot tool hand end perpendicular to the reference point

Point 2: Do A+ on the basis of the first point

Point 3: Do A+ on the basis of the first point, rotate 40°

Point 4: Do A+ on the basis of the first point, rotate 60°

Point 5: Do A- on the basis of the first point, rotate 20°

Point 6: Do A- on the basis of the first point, rotate 40°

Point 7: Do A- on the basis of the first point, rotate 60°

Point 8: Do B+ on the basis of the first point, rotate 20°

Point 9: Do B+ on the basis of the first point, rotate 30°

Point 10: Do B+ on the basis of the first point, rotate 40°

Point 11: Do B- on the basis of the first point, rotate 20°

Point 12: Do B- on the basis of the first point, rotate 30°

Point 13: Do B- on the basis of the first point, rotate 40°

Point 14: Do C+ on the basis of the first point, rotate 30°

Point 15: Do C+ on the basis of the first point, rotate 50°

Point 16: Do C+ on the basis of the first point, rotate 70°

Point 17: Do C+ on the basis of the first point, rotate 90°

Point 18: Do C- on the basis of the first point, rotate 30°

Point 19: Do C- on the basis of the first point, rotate 60°

Point 20: Do C- on the basis of the first point, rotate 90°

Click [Calculate] when you completing the 20-point calibration.

[Cancel calibration]: If you are not satisfied with a point that has been calibrated during the calibration process, you can click the [Cancel calibration] button corresponding to that line to cancel the calibration and then calibrate the point again.

[Run to this point]: You can click [Run to this point] after each point is calibrated, then the robot will run to that point.

[Mark the result position as zero]: Set the position after calibration compensation as the current robot's zero position.

[Clear all mark points]: The calibration points will be saved in the controller, and the calibration results will be cleared only after clicking "Cancel calibration", "Clear all mark points", and switching tool hands to enter the calibration interface.

[20 points without zero calibration]: When this button is turned on, only the size + attitude is calibrated; "Run to calculation result position" is always grayed out, "Mark result position as zero" becomes "Save calculation result". When this button is turned on, the calibration method is that we make the tool hand perpendicular to the calibration rod at point 1, do X- and Y+ at last two points, and calibrate the other points according to the original 20-point calibration method. When this button is turned off, mark 20 points according to the original 20-point calibration method, and you can mark the result position as zero point.

Notes

For the posture of each point, please try to select the posture in any direction. If the posture selected is rotated in a certain direction, the accuracy is sometimes inaccurate.

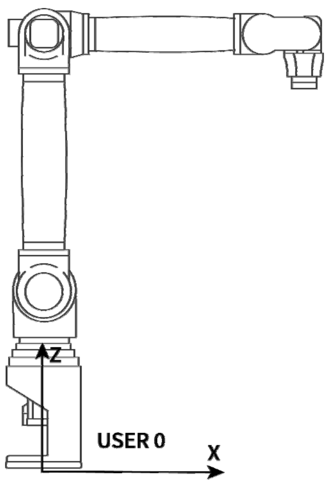
Please keep the reference point fixed during the calibration process, otherwise the calibration error will increase.

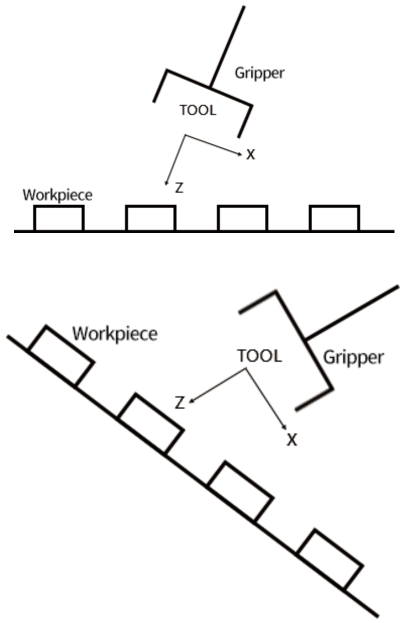
User coordinate system

The role of the user coordinate system

Definition: Default user coordinate system: The default user coordinate system User0 coincides with the Cartesian coordinate system. The new user coordinate system is a change from the default user coordinate system.

Thinking: We know that the user coordinate system is a reference object in motion, but what role does it play in the actual debugging process?





Conjecture: As you can see from the figure, it would be difficult to debug each workpiece position using the default user coordinate system User0 or Cartesian coordinate system, but it would be much easier if there was a coordinate system with two directions exactly parallel to the work surface.

The role of the user coordinate system

1.Determine the reference coordinate system.

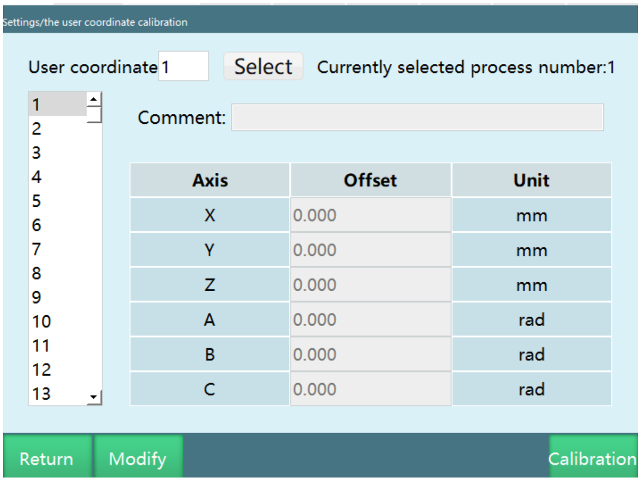
2.Determine the movement direction on the workbench for easy debugging.

User coordinate system characteristics

The new user coordinate system is a change from the default user coordinate system User0. The position and attitude of the new user coordinate system are unchanged in space.

User coordinate parameter setting

Click the [User Coordinate Calibration] button on the "Settings" interface to enter the user coordinate interface, as shown in the figure.



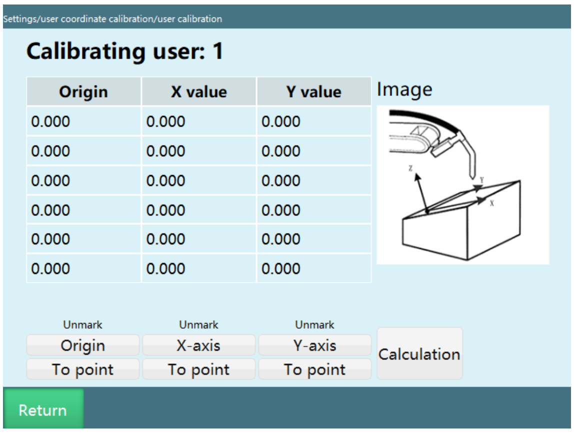
The parameters of the user coordinates are as follows

Parameter	Function
X value	The offset of the origin of the user coordinate relative to the origin of the robot base in the X-axis direction
Y value	The offset of the origin of the user coordinate relative to the origin of the robot base in the Y-axis direction
Z value	The offset of the origin of the user coordinate relative to the origin of the robot base in the Z-axis direction
A value	The angle (radians) that the user coordinate system rotates around the X-axis relative to the Cartesian coordinate system
B value	The angle (radians) that the user coordinate system rotates around the Y-axis relative to the Cartesian coordinate system
C value	The angle (radians) that the user coordinate system rotates around the Z-axis relative to the Cartesian coordinate system

If there is an exact value, please fill in directly. Note that the three values of ABC are radians.

User coordinate system calibration

Click the [User calibration] button at the bottom of the "User coordinate calibration" interface to enter the "User calibration" interface, as shown in the figure.



To calibrate the user coordinate system, please follow these steps:

1.Move the end of the robot to the position that is expected to be the origin of the user coordinate system and click "Calibrate origin" button.

2.Move the robot any distance relative to the origin of the user coordinate system to the position expected to be the positive direction of the X-axis of the user coordinate system, and click the "Calibrate X-axis" button.

3.Move the robot any distance relative to the origin of the user coordinate system to the position expected to be the positive direction of the Y-axis of the user coordinate system, and click the "Calibrate Y-axis" button.

Notes

If the Y-axis of the user coordinate system is not accurately calibrated, the system will automatically compensate

Click the [Return] button at the bottom of the interface to return to the "User coordinate calibration" interface.

Numerical variables

This chapter mainly describes the variables related to this control system.

	Type	Quantity	Example
Global numerical variables	Global Integer GINT	990	GI001....GI990
	Global Double GDOUBLE		GD001....GD990
	Global Boolean GBOOL		GB001....GB990
	Global String GSTRING		GS001....GS990
Local numerical variables	Local Integer INT	999	I001....I999
	Local Double DOUBLE		D001....D999
	Local Boolean BOOL		B001....B999
	Local String STRING		S001....S999

Variable name

Global numerical variables

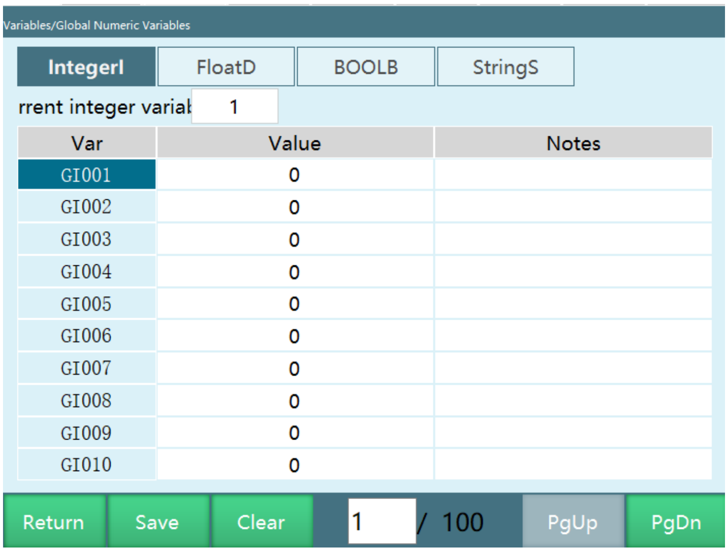
Global numerical variables are variables that can act on all programs of all robots. For example, program AA of robot 1 and program BB of robot 2 can use the same global numerical variable at the same time. This section will mainly explain the use of the global variable interface, as well as the use of position and numerical variables.

The robot needs so many instructions to complete a process, and if we insert the instructions and set the variables each time, it is such a tedious task, based on this, we added numerical variables for calling.

For example, there are many instructions such as "WHILE (INTI001=10)...END (WHILE)" in the program of the robot to complete a certain process, we can directly call the preset numerical variables.

Global numerical variables can also be used to transfer information between the main program, the called subprogram and the background program for logical judgments.

Numerical variables store numerical values and contain four types of variables: integer variables, double variables, boolean variables and string variables.



Note: Global variables will be saved directly to the parameter after assignment

Global boolean variable GBOOL

Global boolean variable saves bytes, and the value and comment of each variable can be modified in this interface. The meaning of each parameter is as follows:

- The "Variable name" is the number of the variable, and the name of the global boolean variable is GBxxx.
- The "Value" is the value of the variable, and the range of the value of the Boolean variable is "0/1".
- The "Comment" is the comment defined by the user for the variable, which is convenient for the user to mark the function of the variable. The range is any value, which can be Chinese.

Global integer variable GINT

The global integer variable saves integers, and the value and comment of each variable can be modified in this interface. The meaning of each parameter is as follows:

- The "Variable name" is the number of the variable, and the name of the global integer variable is GIxxx.
- The "Value" is the value of the variable, and the range of integer variables is integer.
- The "Comment" is the comment defined by the user for the variable, which is convenient for the user to mark the function of the variable. The range is any value, which can be Chinese.

Global double variable GDOUBLE

The global double variable saves real numbers, you can modify the value, content and comment of each variable in this interface. The meaning of each parameter is as follows:

- The "Variable name" is the number of the variable, and the name of the global double variable is GDxxx.
- The "Value" is the value of the variable, and the range of double variables is real numbers.

- The "Comment" is the comment defined by the user for the variable, which is convenient for the user to mark the function of the variable. The range is any value, which can be Chinese.

Click the data type you want to modify, then select the variable name and click [Modify] to modify the value and comment and then click [Save]. You can also click [Clear] to clear the data you have selected.

Global string variable GSTRING

Global string variable can save all variable types and non-variable types, such as: numbers, symbols, letters (including case), Chinese characters

- The "Variable name" is the number of the variable, and the name of the global string variable is GSxxx.
- The "Value" is the value of the variable, and the range of string variables is all variable types and non-variable types.

Use of global numerical variables

Defining global numerical variables

Please define the variables before using them, and define them as follows:

1.Click "Variables - Global numerical" to enter the "Global numerical variables" interface;

2.Select the corresponding global numerical variable type;

3.Select the corresponding variable number and click the "Modify" button;

4.Fill in the required values in the "Value" and "Comment" parts;

5.For variables that have not been manually defined, the default value is 0.

Direct variable assignment

The assignment instructions SETBOOL, SETINT, SETDOUBLE, and SETSTRING allow you to change the value of a variable directly while running the program.

Click the "Insert" button in the "Program" interface;

Select "Variable class";

To change a global BOOL variable, select the SETBOOL instruction and click "OK".

Select "GBOOL" for the variable type; select the previously defined global BOOL variable for the variable name; select "Custom" for the source of the variable value; fill in the value to be changed for the new parameter, if the variable value needs to be changed to 1, then Fill in 1 here;

For example, to change the value of the GB001 variable to 1 when running the program, you can insert the instruction GB001=1

Count with global numerical variable

During the running of the program, all calculations and assignments are made to the values in the cache, but not to the values in the "Variables - Global numerical" interface. If you want to count a loop process (such as WHILE inner loop), you can use the SET instruction.

Usage scenarios:

There is a process between WHILE and ENDWHILE instructions, and there is an ADD GI001 1 instruction inside the process, that is, every time it loops between WHILE and ENDWHILE, the value of GI001 variable is added one, that is, the number of times the process is executed is added one, after the program stops, the value of GI001 is restored to 0, and it is impossible to check the number of times the process is run.

Solution: Insert a SET GI001 instruction after the Add GI001 1 instruction. When the program is finished running, you can see the value of GI001 in the "Variables - Global numerical" interface, which represents the number of times the process has been run.

Insertion method:

Click the [Insert] button in the "Program" interface;

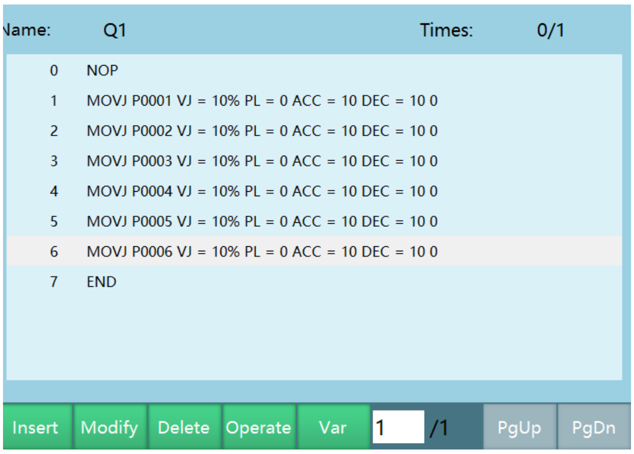
Select "Variable class" - "SET" and click "OK";

Select the variable type, and if you want to change the global integer variable, select GINT and the variable name "GI001";

Click the [Insert] button to finish the operation.

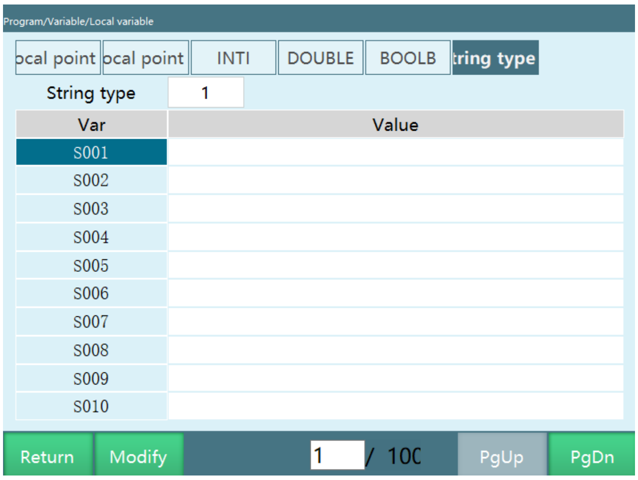
Local numerical variables

Local numerical variables can only be used in the defined program itself, for example, variables of program A cannot be used in program B.



Numerical variables store numerical values and contain four types of variables: integer variables, double variables, boolean variables and string variables.

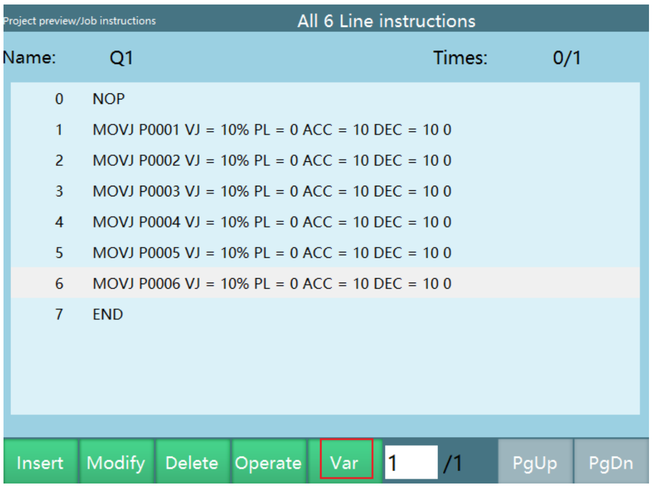
All the local numerical variables defined can only be used in the current program and cannot be used by other programs or background programs.

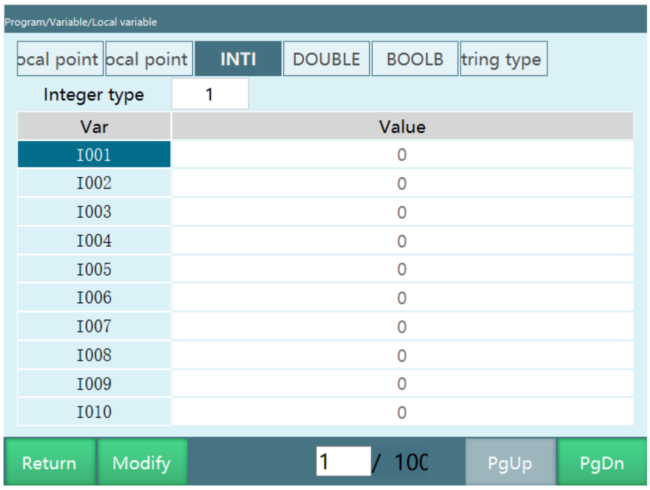


Use of local variables

Defining local numerical variables

Defining a local variable is different from defining a global variable. To define local variables, you need to set them by clicking [Variables] button on the "Program" page.





Integer INT

Local integer variables are used to store integer variables. The variable name is Ixxx.

The default is 0. Select the variable name that needs to be modified and click "Modify", enter the value and click "Save".

DOUBLE

Local double variables are used to store double variables. The variable name is Dxxx.

The default is 0. Select the variable name that needs to be modified and click "Modify", enter the value and click "Save".

Boolean BOOL

Local boolean variables are used to store boolean variables. The variable name is Bxxx.

The default is 0. Select the variable name that needs to be modified and click "Modify", enter the value and click "Save".

STRING

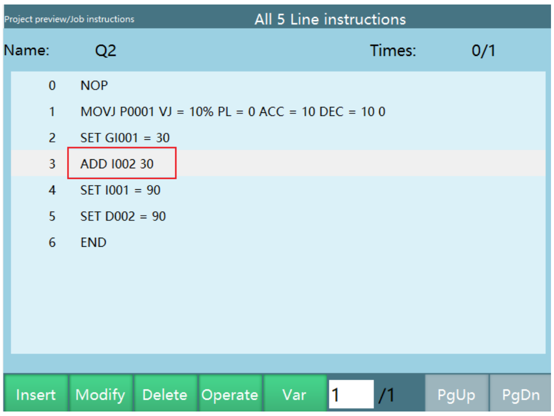
Local string variable can store all variable types and non-variable types, **such as: numbers, symbols, letters (including case), Chinese characters**

Local string variables are used to store string variables. The variable name is Sxxx

The "Value" is the value of the variable, and the range of string variables is all variable types and non-variable types.

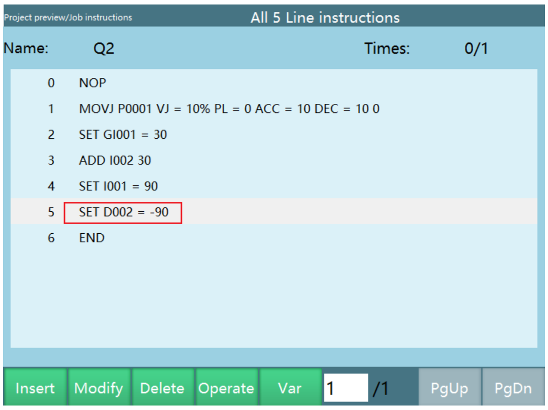
Assignment of values to local variables by calculation instructions

Calculating and assigning values to local variables using ADD, SUB, MUL, DIV, and MOD instructions is done in the same way as for global variables. For example, I003 add 20, as shown in the figure



Direct assignment of values to local variables

Direct assignment of values to local variables using SETINT, SETDOUBLE, SETBOOL instructions is the same as direct assignment of values to global variables. For example: D002=90, as shown in the figure



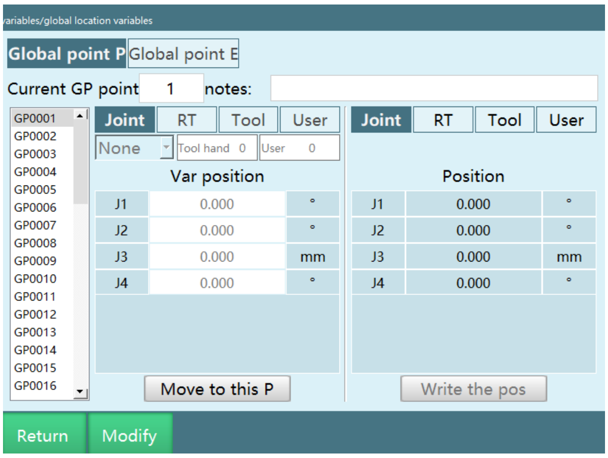
Position variables

This chapter mainly describes the variable settings of this control system.

	Type	Quantity	Example
Global position variable	Global GP point	9999	GP0001......GP9999
	Global GE point	9999	GE0001......GE9999
Local position variable	Local P point	9999	P0001.........P9999
	Local E point	9999	E0001.........E9999

Global position variables

Global GP points are available in all job files of a robot. You can define the global position variables in the "Variables - Global position" interface.



The global position variable is defined as follows:

1.Enter the "Variables" - "Global position" interface;

2.Select the variable to be defined, e.g. GP0001;

3.Teach the robot to the position to be defined and switch the coordinate system to the desired coordinate system, e.g. Cartesian coordinate system;

4.Click the [Modify] button;

5.Click the [Record current point] button;

6.Click the [Save] button.

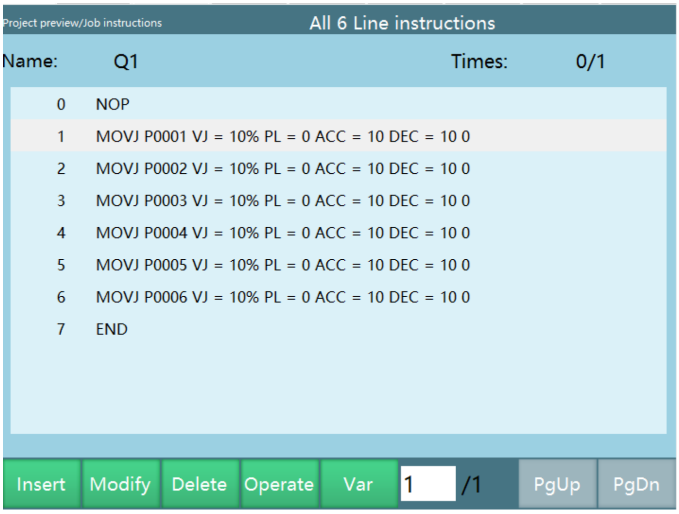
Local position variables

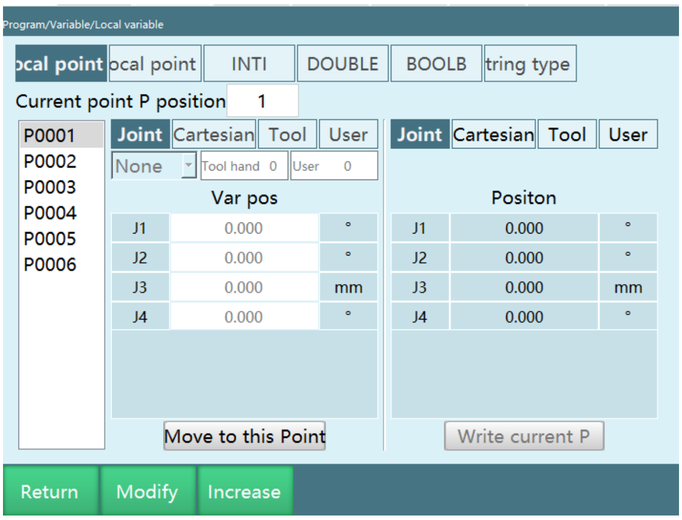
The local position variable (P000X) can only be used for a single job file and cannot be used across all job files.

Local position variables can be defined only when inserting MOVJ, MOVL, MOVC, and other motion instructions, you can define the local position variables in the "Program instruction" interface-"Variables".

Local position variable setting method 1

1.Click "Program"-"Variables"-"Local variables" to enter the local variables view interface

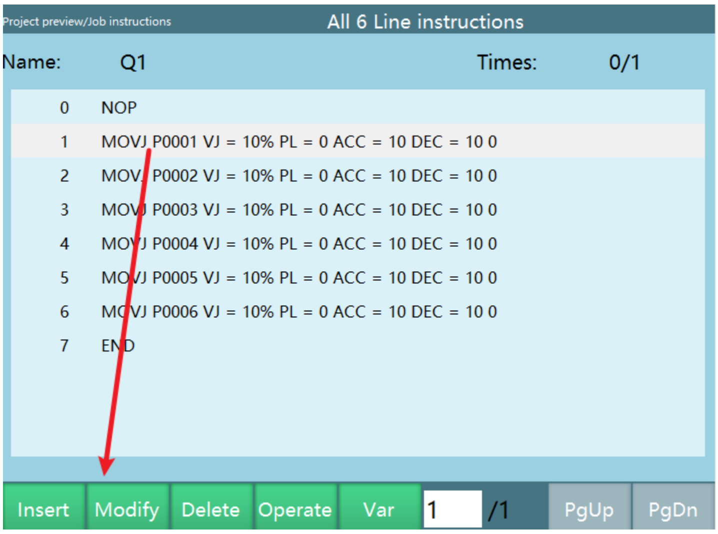


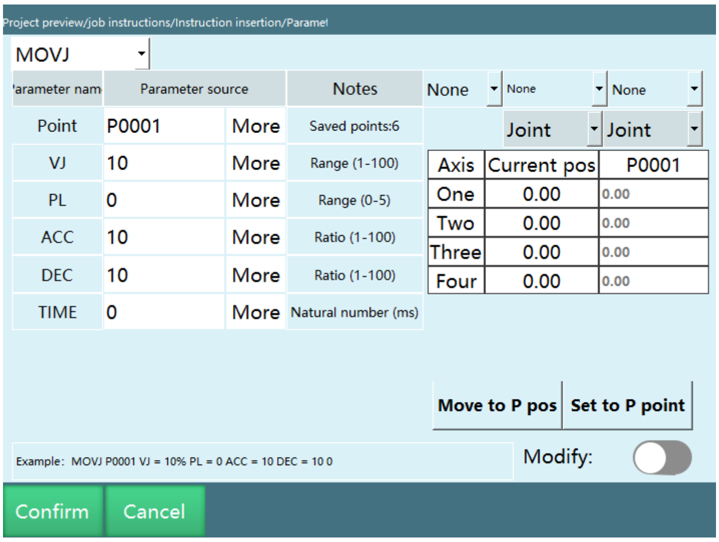


2.You can perform functions such as "Modify points", "Add points", "Run to this point", and "Write to current position" for local position variables

Local position variable setting method 2

Create or modify the MOVJ instruction, enter the instruction interface





2.The "Current position" column shows the robot position in the currently selected coordinate system; the "P0001" column shows the robot position in the selected coordinate system at point P

3.Move the robot to point P: this requires powering up and jogging the robot in the teach mode;

Set the current position as point P: Click to save the current point to the local point P;

Manual modification: turn on to manually fill in the coordinates of point P

Position variable parameters

Form parameters

Form parameters are only available for 6-axis tandem multi-joint robots.

The form value is the binary conversion value of the robot's axis 1, axis 3, and axis 5 positions

Conversion method

Take a 6-axis robot for example, axis 1 is 59 degrees, axis 2 is 69 degrees, axis 3 is 79 degrees, axis 4 is 89 degrees, axis 5 is 99 degrees, and axis 6 is 109 degrees;

Select axis 1, 3 and 5, if the point range is between -90 and +90, then the binary value is 1, if not, then the value is 0;

So the result is as follows:

Axis	Axis 1	Axis 3	Axis 5
Binary value	1	1	0

Binary 110 = Decimal 6

The form value is the decimal result plus 1, so the form value of this point is 7.

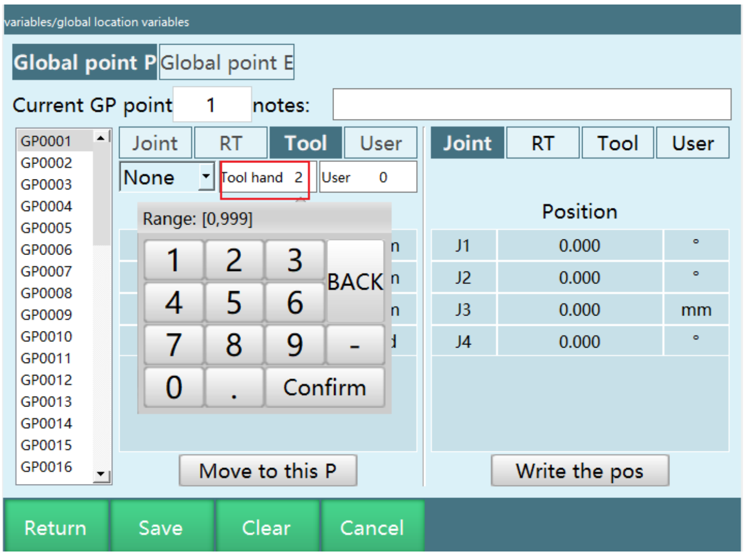
When the current point is selected, the robot will automatically calculate the form of the current point, and the form value corresponds to the interval in which the robot's 135 axes are located. For example: Form 3=010 (axis 1, axis 3, axis 5)+1=011, axis 1 is not within -90°~90°, axis 3 is within the interval, axis 5 is not within the interval.

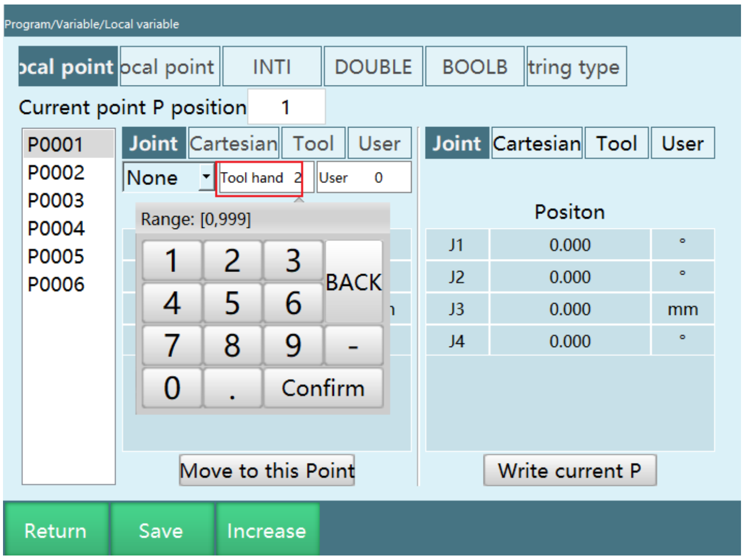
Tool hand parameters

If you want to bind the point to the tool hand, select the corresponding tool hand, if not, then select "No"; if the tool hand used during the operation and the tool hand selected by the point parameter are different, it will not work.

For example, bind tool hand 2 and use tool hand 1 to step the instruction using that point,

Controller reports an error (robot 1 tool coordinate is used incorrectly, point coordinate is 2, actual coordinate is 1)



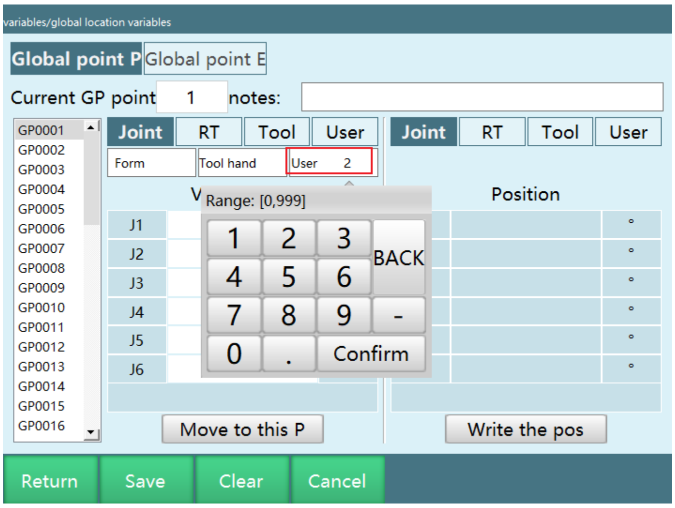


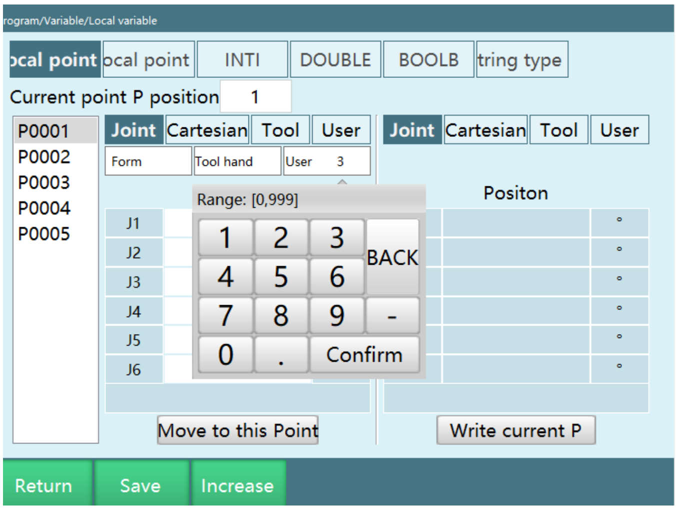
User coordinate parameters

Set the user coordinate point to bind the user coordinate, if not, then select "No"; if the user coordinate used during the operation and the user coordinate system bound to the point parameter are different, it will not work.

For example, bind user coordinate 1 and use user coordinate 5 to step the instruction using that point;

Controller reports an error (robot 1 user coordinate is used incorrectly, point user is 1, actual user is 5)





Description of program local point parameters

This function introduces the point saving format in the program.

```
//DIR
//JOB
//NAME XXX
//POS
///NPOS 2,0,0,0,0,0
///POSTYPE PULSE
///PULSE
P001 = 0,0,0,0,0,0,0,11,22,33,44,55,66,0
P002 = 1,1,0,0,0,0,0,815,0,1297,3.1416,0,0,0
```

For example, P0002 = 1,1,0,0,0,0,0,815,0,1297,3.1416,0,0,0

The breakdown of the point data is as follows:

P0002	Point name	P0001-P9999
1	Coordinate system	0: Joint; 1: Cartesian; 2: Tool; 3: User
1	Angle/radian	0: Angle (joint point); 1: Radian (Cartesian point, tool point, user point)
0	Form/Left and right hand	Form parameters for 6-axis, left and right hand parameters for 4-axis SCARA
0	Tool	Tool hand number
0	User	User coordinate number
0	Reserved	Reserved
0	Reserved	Reserved
815	Axis 1	Point axis 1 coordinate
0	Axis 2	Point axis 2 coordinate
1297	Axis 3	Point axis 3 coordinate
3.1416	Axis 4	Point axis 4 coordinate
0	Axis 5	Point axis 5 coordinate
0	Axis 6	Point axis 6 coordinate
0	Axis 7	Point axis 7 coordinate

Robot Teaching and Running

Robot preparation

Startup and safety confirmation

Startup process:

Check whether the connecting lines of the servo, controller and teach pendant components are well connected

Turn the main power switch on the cabinet panel to the ON position, the main power is connected

Press the green servo start button on the cabinet panel

Warnings

Before teaching, please confirm that the E-stop button is normal

Confirmation of the use of the E-stop button:

Before using the robot, please check the E-stop button on the control cabinet and the teach pendant respectively: whether the servo power is disconnected when the E-stop button is pressed

1.Press the E-stop button on the control cabinet and teach pendant

2.Confirm that the servo power is turned off, the teach pendant shows servo error, and the servo error light on the control cabinet is on

3.Clear the servo error, the servo error light on the control cabinet goes out, and "servo stop" is displayed on the teach pendant

4.。Lightly press the [DEADMAN] button on the teach pendant (the button on the back of the teach pendant), the robot is powered on, and the teach pendant displays "servo running", indicating that the servo power is successfully connected

Teach pendant preparation

Check parameters

Select the robot type:

1.Enter [Settings] - [Robot parameters] - [Slave configuration] - [Robot settings]

2.Click [Modify] and select the robot type

Adjust the servo:

1.Enter [Settings] - [Robot parameters] - [Slave configuration] - [Robot settings]

2.Jog the robot to see if J1 controls the axis 1, J2 controls the axis 2, and so on, if not, modify it yourself

Note: Some servo slave stations are all-in-one, and the robot axes in the slave configuration may not be 1234567 in order

Adjust the actual direction of the robot:

1.Enter [Settings] - [Robot parameters] - [DH parameters]

2.Refer to the robot example picture (the direction marked is: the positive direction of the jogging joint axis), jog the positive direction of the joint axis of the robot, if not consistent, enter [Settings] - [Robot parameters] - [Joint parameters] to adjust the model direction

Adjust model direction:

If the actual direction of the jogging robot is consistent with the direction of the robot example picture, keep the value of the model direction unchanged

If the actual direction of the jogging robot is opposite to the direction of the robot example picture, reverse the value of the model direction

Adjust the zero position:

The scale of each axis on the robot body is the mechanical zero point, adjust each axis of the robot to the mechanical zero point

1.Enter [Settings] - [Robot parameters] - [Zero position]

2.Click [Set all joints to zero]

Notes

If you jog the robot joint axis 90 degrees actually, but the display on the teach pendant is not 90 degrees, then you need to adjust the reduction ratio or confirm with the manufacturer

If the robot cannot walk straight when jogging the coordinate axis in Cartesian coordinate system, then you need to adjust the DH parameters or confirm with the manufacturer

Jogging robot

1.The teach pendant and the controller are connected properly

2.Servo and robot parameters are normal

3.In teach mode, press the [Servo] button on the teach pendant to switch the the state from "servo stop" to "servo ready"

4.Lightly press and hold the [DEADMAN] button on the teach pendant (the button on the back of the teach pendant), you will hear the sound of the robot being powered on, and the "Servo status" column will display green "servo running"

5.Control the movement of the robot by operating the physical buttons on the right side of the teach pendant

Basic operation of the project interface

1.Switch to "Admin" account

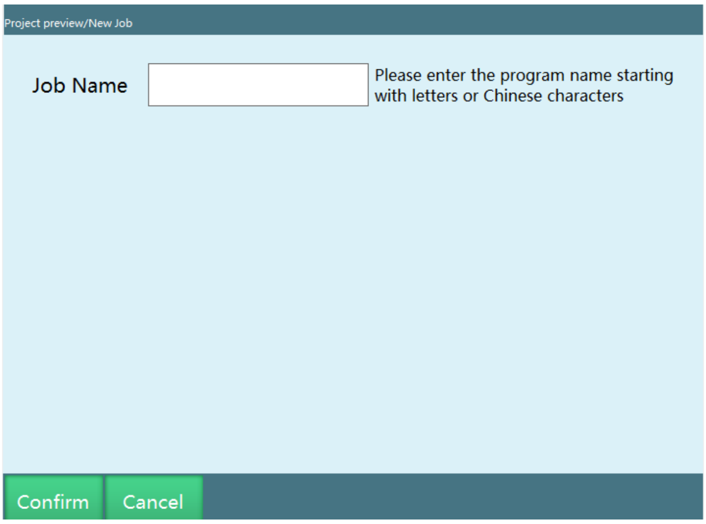
2.Click [Project] on the left

Create new program:

To create a new foreground program, the user needs to perform the following steps:

1.Enter the [Project] interface and click [New]

2.Enter the program name in the "New program" window that pops up



3.Click the [OK] button at the bottom, the program is created successfully, you will jump to the interface of the newly created program; if you want to cancel the new operation of the program, then click the [Cancel] button

Notes

The program name must be a string of two or more characters starting with a letter/Chinese character

The new program name cannot be the name of an existing program

Open program

To open an existing program, the user needs to perform the following steps:

1.Enter [Project] interface

2.Select the program you want to open

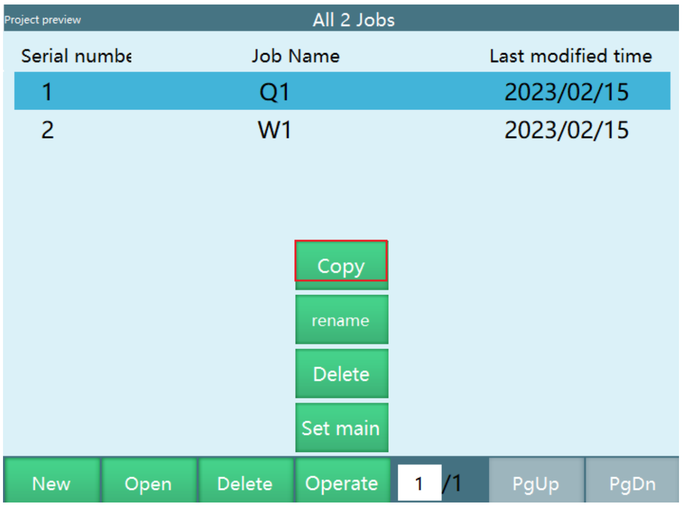
3.Click the [Open] button at the bottom, the program opens successfully

Copy program

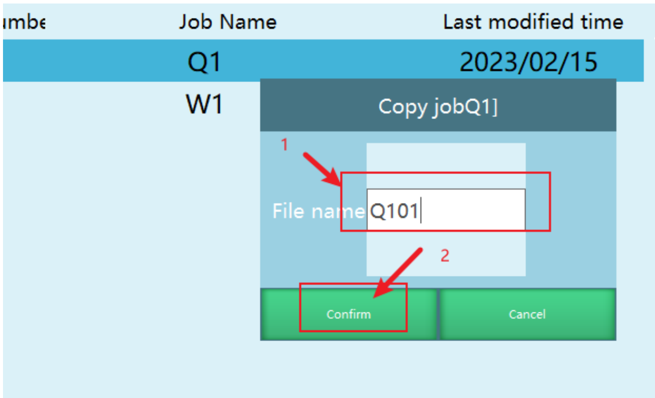
To copy an existing program, the user needs to perform the following steps:

1.Enter [Project] interface

2.Select the program you want to copy



3.Click the [Operation] button at the bottom, and then click [Copy]



4.Click [OK], you can also modify the program; if you want to cancel the copy, click [Cancel]

Rename program

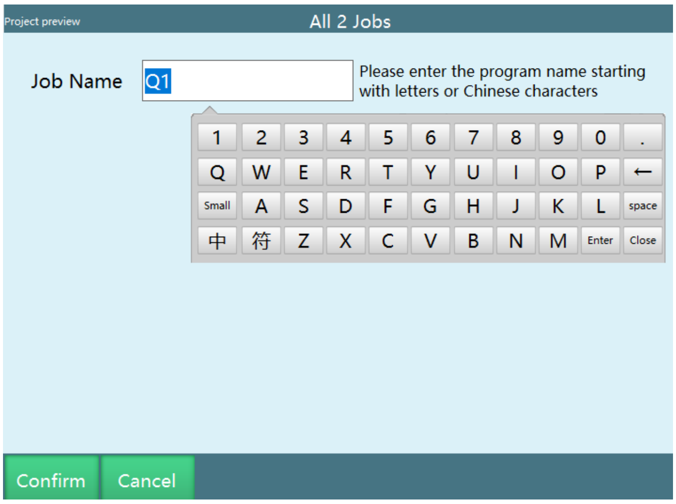
The rename operation can change the name of the selected program

The operation steps are as follows:

1.Click [Project], select the program you want to rename

2.Click [Operation], then click [Rename]

3.In the pop-up window, enter the name you want to change



4. Click the [OK] button; if you want to cancel the rename operation, click the [Cancel] button

Notes

The program name of the renamed program cannot be the name of an existing program

The program names of the programs in the foreground and background cannot be repeated

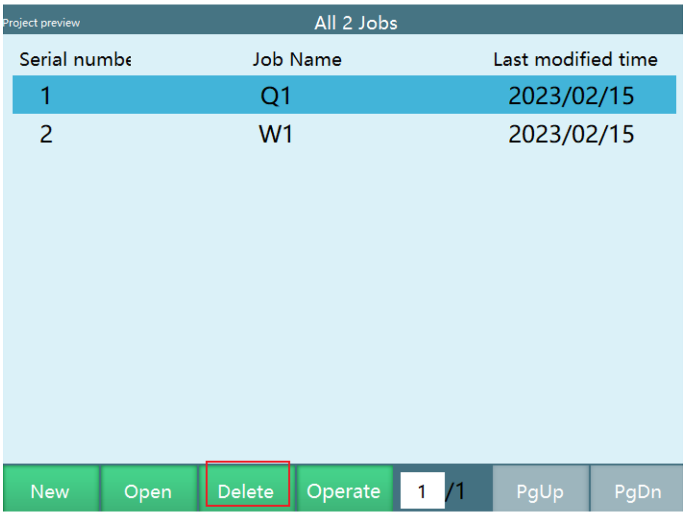
Delete program

The delete operation can delete the selected program

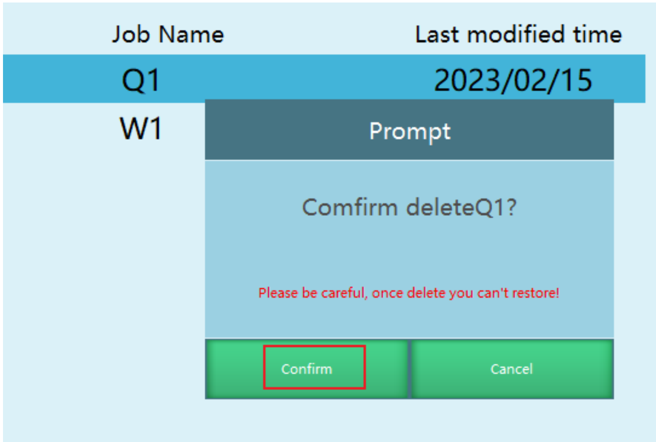
The relevant operation steps are as follows:

1.Click [Project], select the program you want to delete

2.Click the [Delete] button at the bottom



3.Click the [OK] button in the pop-up window; if you want to cancel the delete operation, click the [Cancel] button

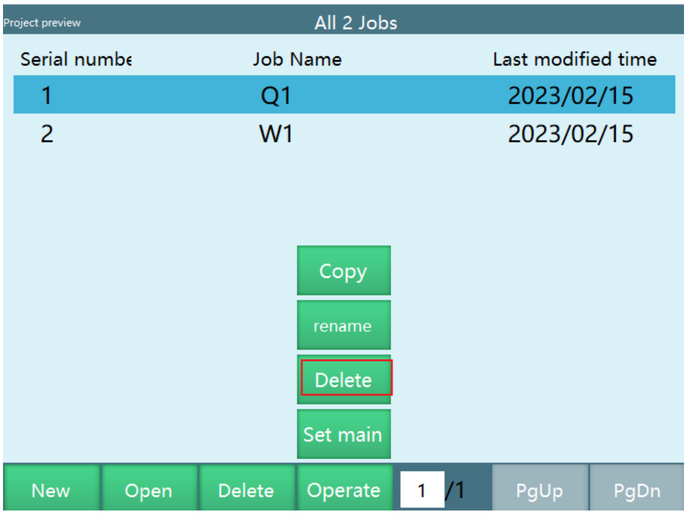


Batch delete

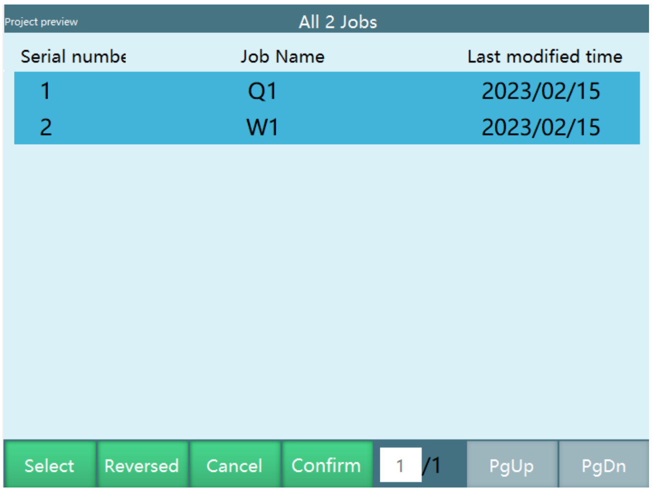
The batch delete function can delete multiple programs at one time. The method of use is as follows:

1.Click [Project]

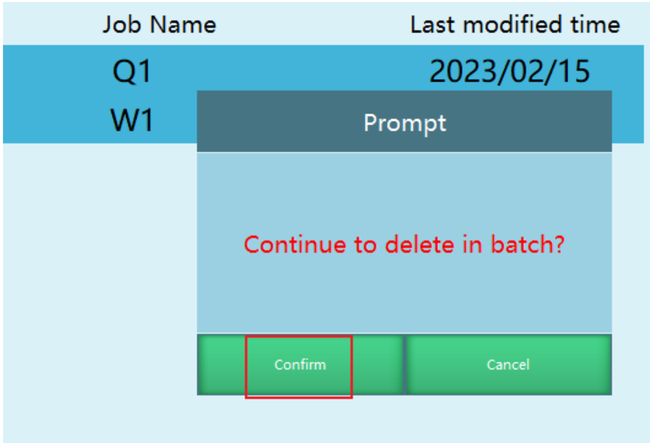
2.Click [Operation] on the bottom menu bar and select the [Batch delete] button



3.Select the program to be deleted, and click the [Select all] button to select all programs on this page



4.Click the [OK] button, a confirmation box will pop up, click the [OK] button to delete the batch successfully



Notes

The batch select operation can only select the files on the current page, but cannot enter the previous or next page

Program instruction writing

Instruction operation

If the user wants to perform some operations related to instructions, such as insert/modify/delete/operate, he needs to enter the program instruction interface, and use the buttons at the bottom to perform related operations

Insert instruction

The insertion of instructions needs to be performed by using the [Insert] button at the bottom of the program instruction interface

The inserted instruction is below the selected instruction line, you can insert 9999 points

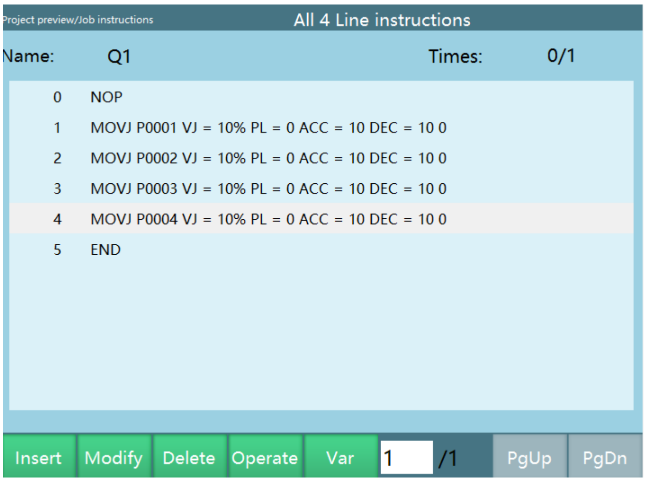
The relevant steps are as follows:

1.Switch to "Admin" account

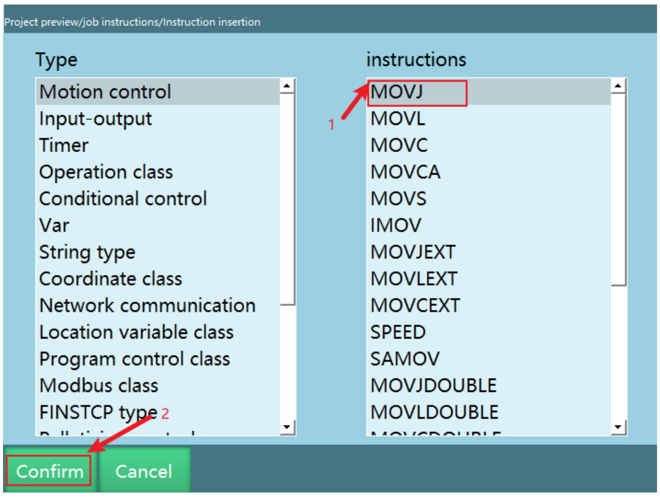
2.Click [Project] on the left

3.Click [New]

4.Enter the program instruction interface

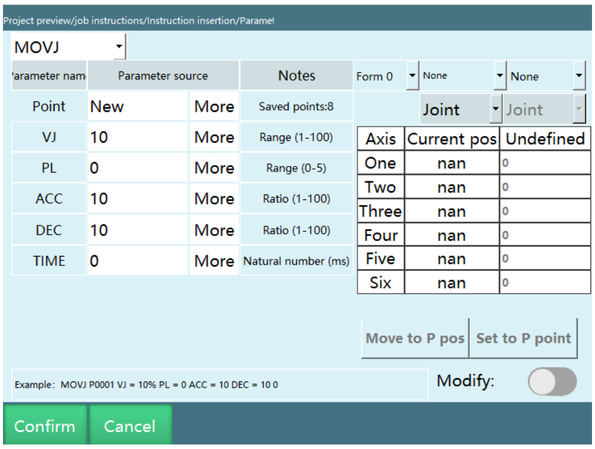


5.Click the [Insert] button, the "Instruction type" menu will pop up



6.Click on the instruction type of the instruction to be inserted, e.g. motion control class

7.Click the instruction to be inserted, such as MOVJ, as shown in the figure:



8.Set the relevant parameters of the inserted instruction

9.Click the [OK] button at the bottom

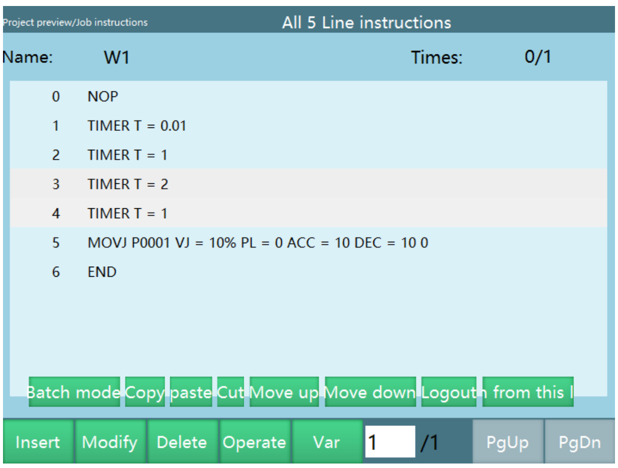
Modify instructions in batch mode or single-line mode

Batch mode: You can copy, paste, cut, delete, modify, log out, move up, move down multiple instructions at the same time

Ⅰ. If the user wants to batch copy, paste, cut, delete, modify, log out, move up, move down the instructions in this job file, take batch copy as an example, the steps are as follows:

1.Click [Operation] - [Batch mode] at the bottom to enter batch mode

2.Select one or more instructions to be copied



3.Select the [Copy] button

4.Select the instruction above the target position

5.Click the [Paste] button

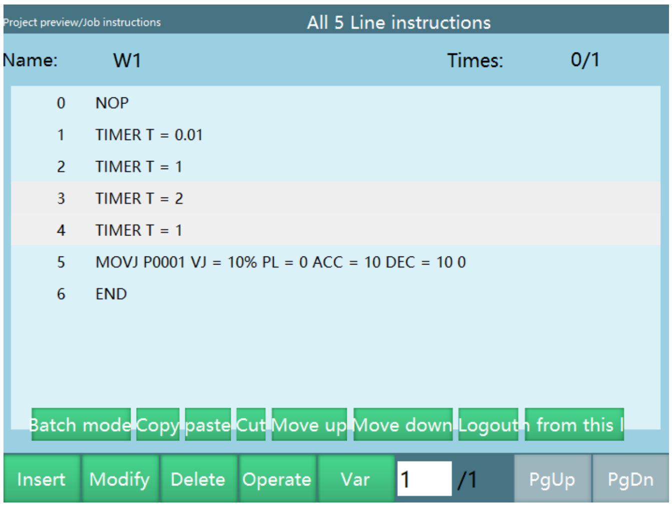
Ⅱ. If the user wants to batch copy, paste, cut, delete, modify, log out, move up, and move down the instructions across job files, take batch copy as an example, the steps are as follows:

1.Enter the [Project] interface

2.Open the program to be copied

3.Click [Operation] - [Batch mode] at the bottom to enter batch mode

4.Select one or more instructions to be copied



5.Select the [Copy] button

6.Open the job file to which you want to copy the instruction

7.Select the instruction above the target position

8.Click [Paste]

Single-line mode: exit batch mode

Click [Operation] - [Batch mode] - [Single-line mode] at the bottom

Notes

Foreground program instructions cannot be copied to background programs

Basic operation of each mode

The user can switch between three modes ("Teach", "Run", "Remote") by using the [Mode selection key] in the upper right corner of the teach pendant, and the program can run in these three modes

Teach mode

In the teach mode, you can perform some operations related to the robot, such as system parameter setting, jogging operation, job file programming. In the process of the job file programming, you can use the [Step] button to perform step operations on the job file

Trajectory confirmation with Step button

After selecting the inserted instruction line, the user can perform step operations on the programmed job file by holding down the [DEADMAN] button while clicking the [Step] button in the physical button area at the bottom of the teach pendant (do not release the [DEADMAN] button while the robot is in motion). Step operation can run only the selected instruction line

The specific steps are as follows:

1.Select the instruction line to be stepped

2.Press the [DEADMAN] button, the robot is powered on

3.Press the [Step] button, the robot executes the instruction of the selected line, and stops after completing the execution

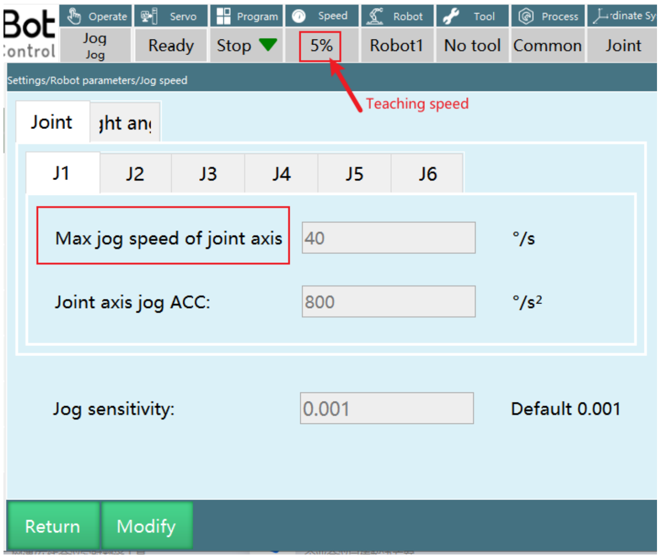
4.The selected line will move down automatically. If you want to step the next line of instruction, press the [Step] button again

Teach mode speed

1.In the teach mode, the actual jog speed of the robot is calculated as follows:

	Actual jog speed of the robot	Maximum speed limit
Jog the robot in the joint coordinate system	Maximum jog speed of joint axis*teach speed	Custom
Jog the robot in the Cartesian coordinate system	Cartesian coordinate maximum jog speed*teach speed	250mm/s
Return to zero	Rated speed*teach speed	Rated speed*30%
The speed of returning to the safety point by joint	Rated speed*teach speed	Rated speed*30%
The speed of returning to the safety point in a straight line	100mm/s*teach speed	Omitted
The speed of running to this point	Rated speed*teach speed	Omitted
Step joint speed	Rated speed*(teach speed*instruction speed)	Rated speed*30%
Step Cartesian speed	Teach speed*instruction speed	Omitted

2.Take the calculation of the actual jog speed of the robot in the joint coordinate system as an example:



The actual jog speed of the robot is: VJ=40°/s * 50%=20°/s

Maximum speed limit of the robot: If the maximum jog speed of the joint axis is 40°/s, then the actual jog speed of the robot will not be greater than 20°/s according to the formula of maximum speed limit (maximum jog speed of the joint axis*50%), regardless of the teach speed

Maximum Cartesian speed limit in the joint coordinate system: adjust "stepMaxDecareSpeed" in the controller configuration file Robot_A.json: 300 (300 is the default speed value, in mm/s)

Commissioning function

1.The commissioning function is to use the [Start] button as the commissioning button in the teach mode, press and hold the [Start] button to keep running when power on, and release it to stop

2.The commissioning mode supports all instructions

3.The commissioning function does not support reverse order and background programs

Running mode

In the running mode, you can click the [Running times] button in the lower left corner to set the running times of the program, the default is [Single]

Click the [Cycle] button in the lower left corner to make the program run in an infinite loop

In the running mode, the upper part of the program displays the already running times and the total set running times, the format is "already running times/total set running times", in the process of running, the user can modify the running times. After the modification, the robot stops after running the set times. For example, the original running times setting is 200, and the robot has run 156 times. At this time, if you set the running times to 3, the robot will continue to run three times and then stop

Running mode speed

Running speed = instruction speed*speed ratio in the status bar above

For the default speed of the running mode when startup, the user can set it in **[Operation parameters]**

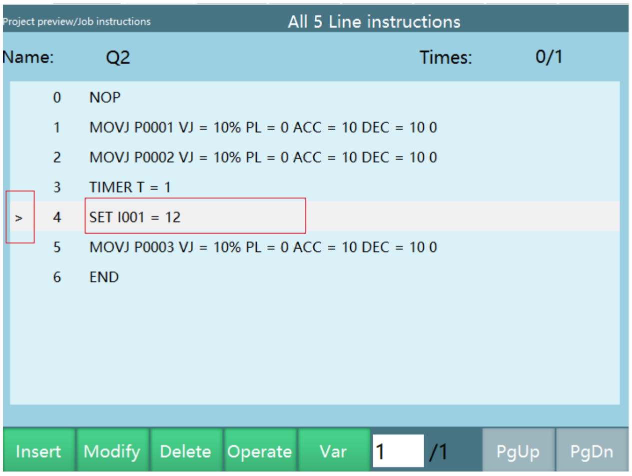
Notes

The instruction speed set during welding is the actual speed. Suppose the linear speed is set to 50mm/s, then the actual welding speed is 50mm per second

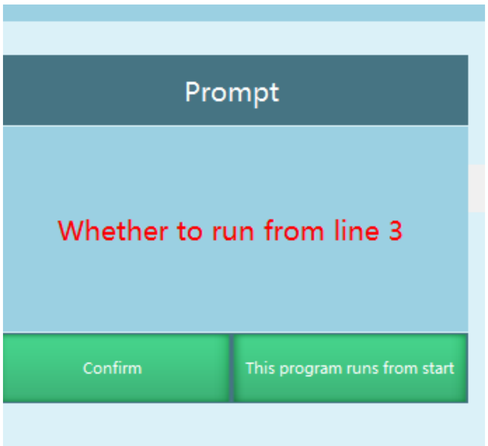
The running speed after using the global speed is: teach speed*instruction speed*global speed

Running from current line

Ⅰ. Open the job file in teach mode, select a line, click the [Operation] button, click [Run from here], a > symbol will appear in the job file



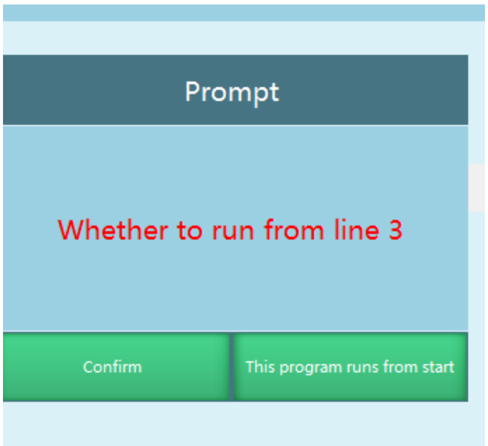
1.Switch to running mode, click [Start], there will be a prompt pop-up window when running



2.Click the [OK] button to run from the selected line, and click [Run this program from the beginning] to run from the first line of the program

Ⅱ. In the running mode, when the program runs into the subprogram, switch to the teach mode, select a line, click on the [Operation] button, click on [Run from here], a > symbol will appear in the job file

1.Switch to running mode, click [Start], there will be a prompt pop-up window when running



2.Click the [OK] button to run from the selected line. After the subprogram is completed, it will return to the main program and continue to execute the next instruction

If you click [Run this program from the beginning], it will start from the first line of the subprogram and will not return to the main program

Breakpoint operation

Teach mode breakpoints

There is also a "breakpoint" in the teach mode. If there is an instruction to change the local variable in the step process of the program, you can check the local variable value at the "breakpoint" by turning the power off and then on

If you want to clear the "breakpoint", you can perform operations such as returning to zero, reset, power off during the step process of instruction, running other programs, running to this point, modifying local values/local position variables and performing step operation on instructions, restarting the controller, and modifying robot parameters

Running mode breakpoints

During the operation (except the first instruction), if the operation is interrupted when switching to other modes, the variable status and program running position at the time of interruption will be saved as a breakpoint, and when running again, a prompt pop-up box will ask "whether to continue running the current program", select "Execute at breakpoint" to continue running from the breakpoint, select "Rerun" to run again from the first instruction and the breakpoint disappears

Cases where breakpoints will not be cleared:

1.IO E-stop/servo alarm/output information instruction

2.Exit the current program and re-enter to run it again

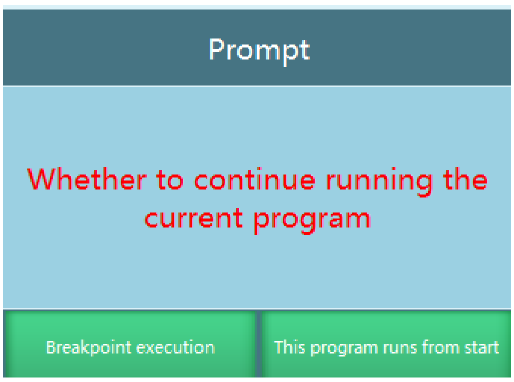
Jog the robot

Go to other pages to modify non-robot parameters

Switch to running mode, select "Cycle", modify the running times

Cases where breakpoints occur:

1.Select "Run this program from the beginning" in the pop-up window



2.Insert/delete/move/cut/copy instructions

3.Modify local value/local position variable/program instruction

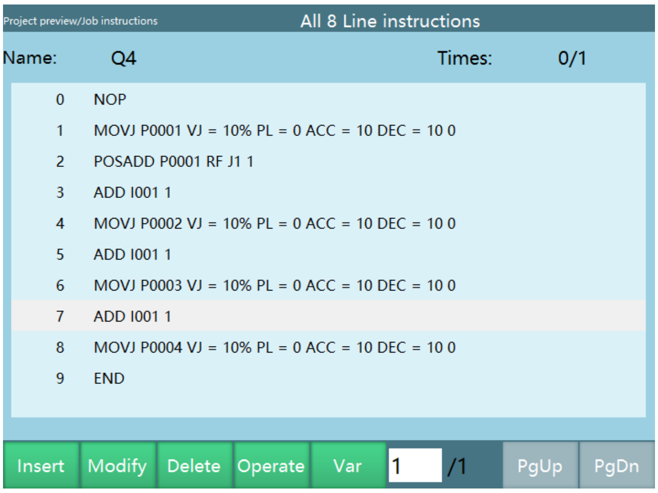
4.Error when running program instruction and power off

5.Restart the controller, modify the robot parameters

Breakpoint status check:

After switching to teach mode when breakpoint occurs, you can check the position/value variable status at breakpoint by power-on.

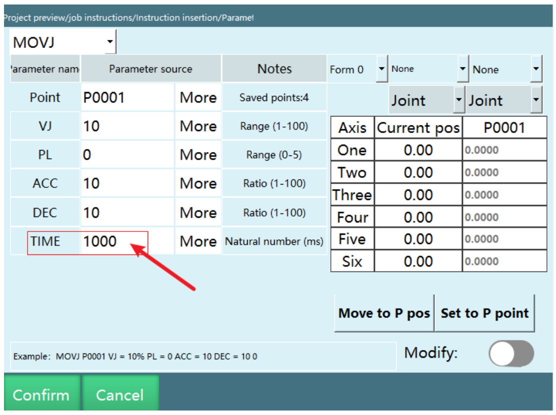
Example: The initial state of P0001 and I001 is shown in the figure, and it changes as follows during the operation: P0001 J1+1, I001+1.

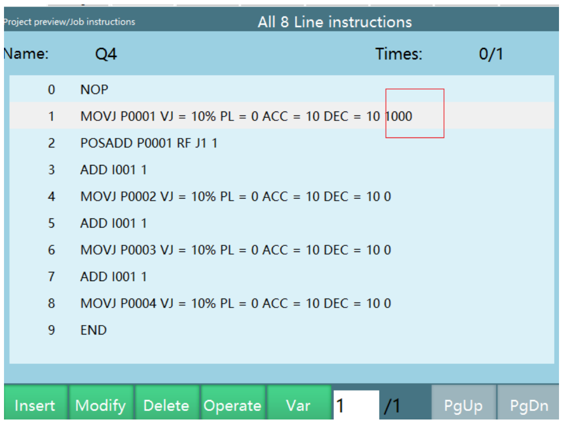


When running to the 6th line, P0001 J1=1, I001=2, a breakpoint occurs when switching to teach mode, after switching to teach mode, P0001, I001 is displayed as the initial value, at this time press [DEADMAN] to power on, it will be displayed as P0001 J1=1, I001=2, the initial value is restored after power off

Early execution function

It takes effect when the motion instruction's time parameter is set, and the unit of the parameter is ms, as shown in the figure:





Insert the DOUT instruction after the MOVJ instruction; fill in the TIME parameter of the MOVJ instruction with 1000ms, then the next instruction will be executed 1s ahead of time during running. For example, if the MOVJ instruction is to be executed for 3s, then the MOVJ instruction will run for 2s before the DOUT is executed, after the execution of DOUT, the MOVJ continues to run to P0001.

Remote mode

Remote mode supports two control methods: digital IO and Modbus slave

The device priority is: Modbus > digital IO. When two external devices are connected, the enabling of digital IO can be controlled through Modbus touch screen

When the teach pendant is unplugged, trigger the remote IO signal, it will automatically enter the remote mode

Modbus & digital IO can be used at the same time

The open method is as follows:

Open the modbus file in the Addr.jsonconfig file

Change "false" after coexistIOControl to "true"

Notes

When Modbus & digital IO are used at the same time: Modbus controls the start and stop of the program

When Modbus & digital IO are used at the same time: the program setting needs to be done in the remote program setting interface

When Modbus & digital IO are used at the same time: whether the program supports current line or breakpoint execution needs to be set in [Remote IO breakpoint execution] and [Remote IO current line execution] on the operation parameter page

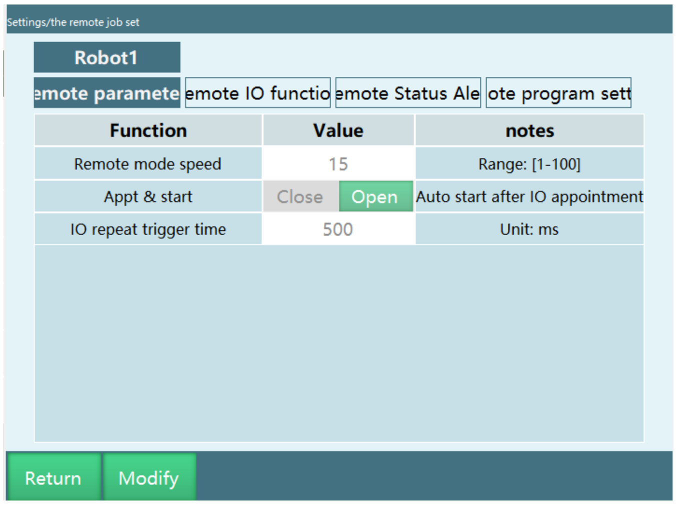
Remote mode speed

Remote point-to-point speed = rated speed*remote speed*instruction speed

Remote linear speed = Remote speed*instruction speed

Remote IO speed modification method

1.Enter [Settings] - [Remote program settings] - [Remote parameters] interface

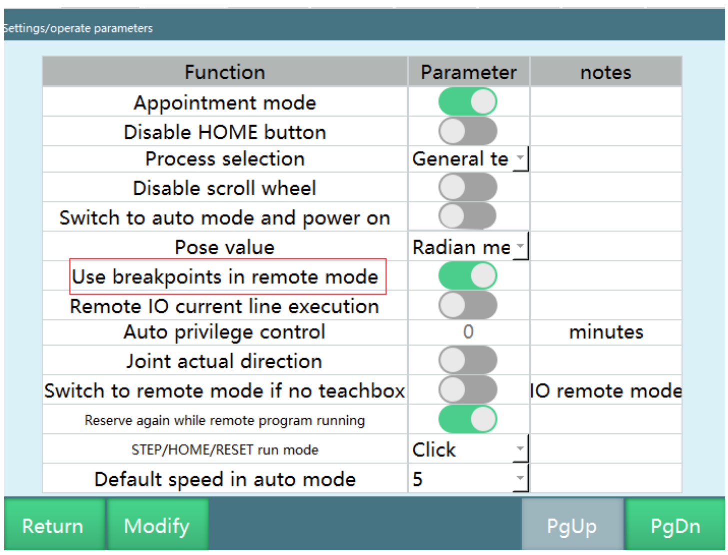


2..Click [Modify] to modify the remote mode speed

3.Click [Save], you can switch to remote mode to view it

Remote mode breakpoints

When using the io reservation program, the breakpoint will be executed by default. If you do not need to execute the breakpoint remotely, you can turn it off in [Settings] - [Operation parameters] - [Remote IO breakpoint execution]



Notes

The teach pendant is prohibited from modifying the speed in remote mode. The remote speed needs to be set in advance in the teach mode, the default remote speed is 15%

Acceleration adjustment

Function: Increase robot efficiency, the larger the acceleration multiple, the faster the robot runs to its maximum speed

Enter [Settings - Robot parameters - Joint parameters] to adjust the acceleration multiple

When the acceleration multiple is set to 1, it takes 1s for the robot to reach the maximum rated positive speed, but if the acceleration multiple is set to 2, it takes 0.5s for the robot to reach the maximum rated positive speed, reducing the time by 1/2

1.Time to run to rated speed = (running speed*instruction speed) /

(acceleration multiple*instruction acceleration*running speed)

Example 1: The running speed is 50%, the instruction speed is 40%, the instruction acceleration is 10%, the rated positive speed is 4000 rpm, and the maximum acceleration is 4 times. (point-to-point instruction)

2.Instruction maximum speed=rated speed*running

speed*instruction speed=4000r/min*50%*40%=800r/min

The time required for the robot to run from 0r/min to 800r/min = (rated speed*running speed*instruction speed) / (rated speed*acceleration multiple*running speed*instruction acceleration) = (4000r/min*40%*50%) / (4000 r/min*4*50%*10%)=1s

Example 2: The running speed is 30%, the instruction speed is 1000mm/s, the instruction acceleration is 50%, the Cartesian maximum speed is 2000mm/s, and the Cartesian maximum acceleration is 2 times. (straight line instruction)

3.Instruction maximum speed = running speed * instruction speed =

1000mm/s * 30% = 300mm/s

The time required for the robot to run from 0mm/s to 300mm/s = (running speed * instruction speed) / (Cartesian maximum speed*Cartesian acceleration multiple*instruction acceleration*running speed) = (1000mm/s*30%)/(2000mm/s*2*50%*30%)=0.5s

Debugging Manual

‹‹‹

Catalogue

Debugging Manual

Safety Instructions

Warnings

- This manual covers robot parameters, external axis parameters, system settings and other operations.
- You must read carefully and fully understand the instructions in the manual before using this system.
- Please consider anything not described in the manual as "not allowed" or "prohibited".

Cautions

- The diagrams in the manual are only to demonstrate some detailed parts, you need to install the complete equipment before using this system
- Due to system function improvement, the manual will be revised appropriately, and the revised manual will update the manual version number

Safety tips level

This manual includes safety precautions to ensure the personal safety of operators and prevent machine damage, and describes them in the main text as "Warnings" and "Cautions" according to their importance in terms of safety. The relevant supplementary notes are described as "Notes". Before using, users must read carefully the items described in "Warnings", "Cautions" and "Notes".

Warnings

- Used in situations where there may be a danger of death or serious injury to the user due to incorrect operation

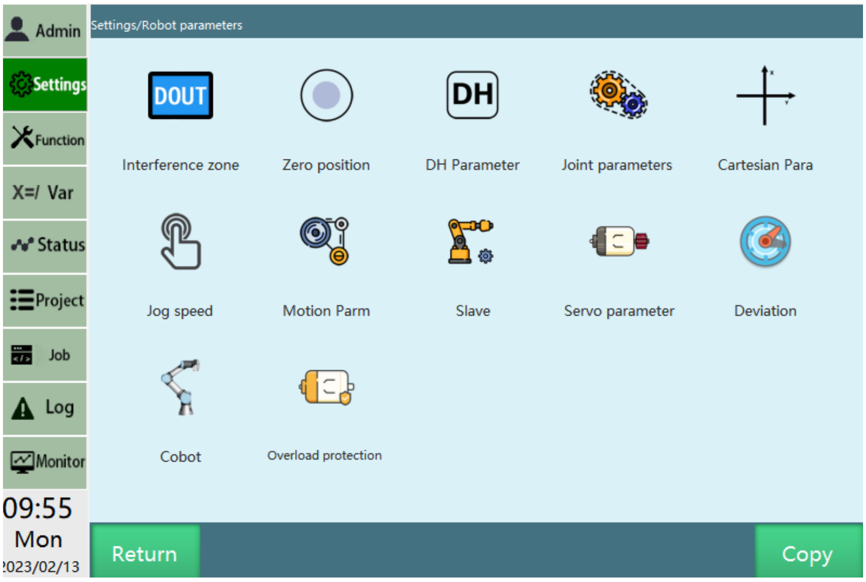
Cautions

- Used in situations where there may be dangers such as minor or moderate injury, damage to objects, etc. due to incorrect operation

Notes

- Used to describe items other than warnings or cautions

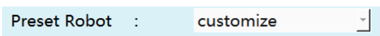
Robot and External Axis Parameters



Robot Parameters

In the "DH parameters" interface, we provide the "Preset robot" function. If this drop-down list contains the robot model you are using, you can use this function to quickly and easily set up the parameters of your robot.

1. Click [Preset robot] in the upper left corner of the "DH Parameter" interface, you can select the robot model that has been adapted, and the DH parameters and joint parameters of the robot will be filled in automatically after selection.



2. You need to modify the zero point manually after selecting the preset robot.

Notes

- The configuration method of the preset robot can be obtained by contacting the system manufacturer

Slave configuration

Warnings

- Please do not switch to servo-ready mode, power on, run and do other operations

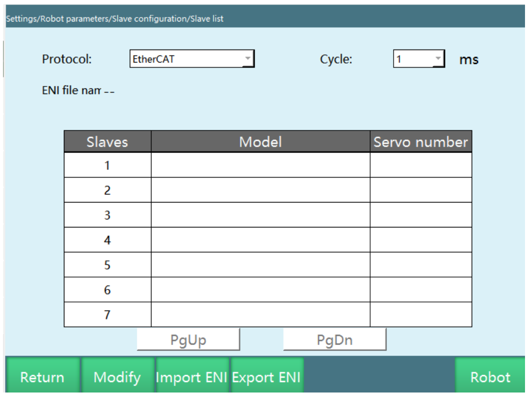
You can go to [Settings] - [Robot parameters] - [Slave configuration] to modify the robot settings.

The relevant steps are as follows:

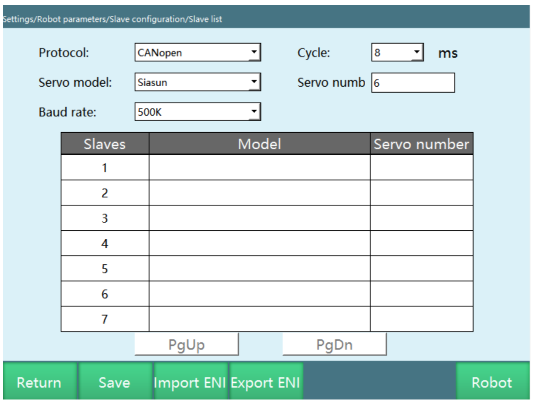
Enter [Settings] - [Robot parameters] - [Slave configuration] interface;

This interface displays the number of slaves currently connected to the controller; you can modify the communication cycle and bus type which is divided into EtherCAT and CANopen, and the modification takes effect after restarting.

Bus type: EtherCAT

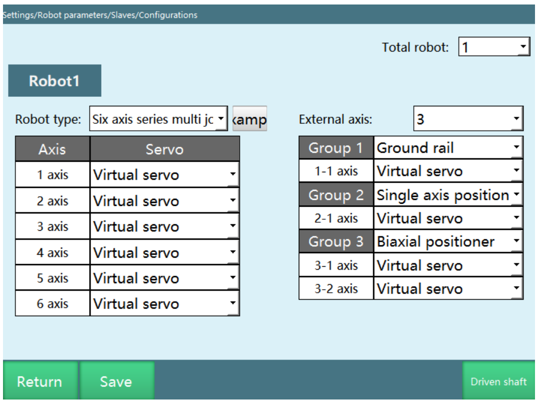


Bus type: CANopen

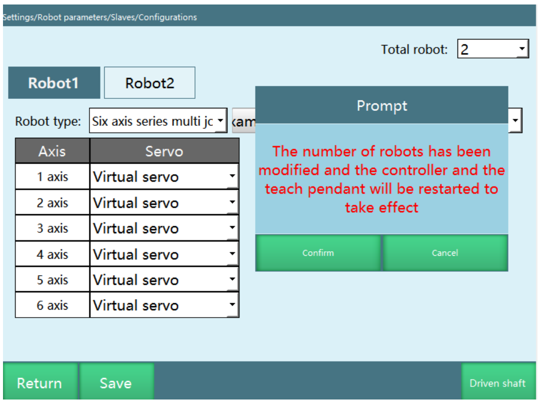


Enter [Robot] interface

You can set the number of robots, robot type and number of external axis groups, and do servo selection in this interface;

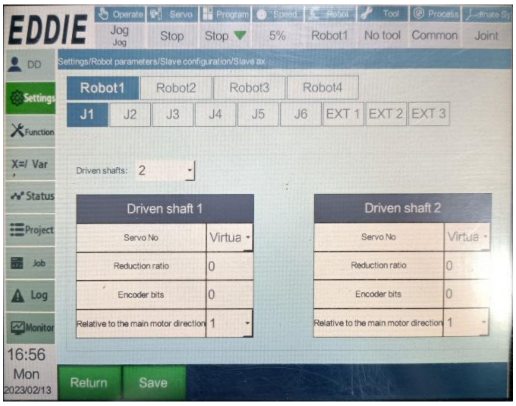


After the number of robots is modified, there will be a prompt box indicating a restart is required to make the modifications effective, other parameter modifications take effect immediately. *_*The parameters will be reset after modifying the robot type, so be careful when changing the robot type*_*



Enter [Slave axis] interface

You can set the number of slave axes, reduction ratio, number of encoder bits and direction relative to the master motor, and do servo selection in this interface.



Meaning of each parameter

- **Robot communication cycle**

1ms, 2ms, 4ms or 8ms

- **Number of robots**

One controller supports up to 4 robots

- **Robot type**

6-axis tandem multi-joint, 6-axis collaborative, 6-axis painting, 6-axis special-shaped one, 5-axis tandem multi-joint, 4-axis SCARA, 4-axis SCARA special-shaped one, 4-axis linkage palletizing, 4-axis palletizing screw, 4-axis parallel robot, 4-axis tandem multi-joint, 4-axis Cartesian, 4-axis polar coordinate special-shaped, 3-axis SCARA, 3-axis Cartesian, 3-axis Cartesian special-shaped one, 2-axis SCARA, 7-axis tandem multi-joint, single-axis, gantry welding model, wine tank model

- **Number of external axis groups**

External axis type supports ground rail, single/dual axis positioner, and supports up to 3 groups of external axes, the total number of axes is up to 5, and there can only be one ground rail

- **Servo serial number**

Servo's corresponding serial number, model

- **The currently supported servo types are as follows**

  Note: This table is no longer updated after 20.06 because the function of adding servo files has been added, and users can configure themselves

- **The currently supported IO types are as follows**

IO board manufacturers	IO model
HUATAI	HUATAI
	HUATAI PWM
Metrotech	Metrotech
	Metrotech old
ITEGVA	ITEGVA
GPG	GPG
INEXBOT	R1
	R1_PWM
	R2
	R2A
	R2B
	R3
	R4
	R4P
	High precision clock

Xiling	Xiling EJ1861
	Xiling EJ1862
MT	MT
Leadshine	Leadshine
YENWARE	YENWARE CATIOA

Joint parameter setting

Warnings

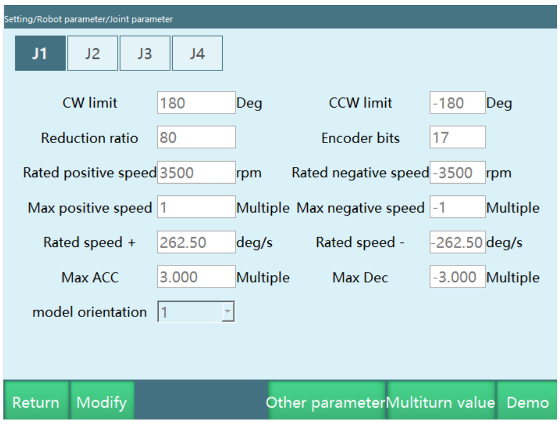
⚠

- Please do not switch to servo-ready mode, power on, run and do other operations without configuring this parameter

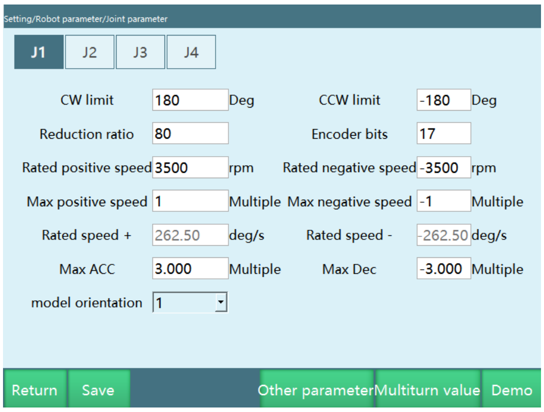
You can go to [Settings] - [Robot parameters] - [Joint parameters] to modify the joint parameters.

The relevant steps are as follows:

1. Enter [Settings] - [Robot parameters] - [Joint parameters] interface.
2. At this time, the input box is grayed out and no value can be entered.



3. After clicking "Modify", the "Modify" button becomes "Save", the input box becomes white, and you can enter the value after the respective parameter.



4. Click "Save" and the modification is successful.

Meaning of each parameter

- **CW limit**

Maximum range of robot joints in positive direction.

- **CCW limit**

Maximum range of robot joints in reverse direction. (This value must be negative)

- **Reduction ratio**

Reduction ratio of the reducer.

- **Encoder bits**

Number of bits of the encoder.

- **Rated positive RPM**

The rated rotation speed of the motor in the positive direction.

- **Rated reverse RPM**

The rated rotation speed of the motor in the reverse direction. (This value must be negative)

- **Maximum positive RPM**

The maximum rotation speed of the motor in the positive direction; its value is a multiple of the rated positive RPM. If the rated positive RPM is 3000 rpm and the maximum positive RPM needs to be 6000 rpm, then fill in 2 times here.

- **Maximum reverse RPM**

The maximum rotation speed of the motor in the reverse direction; its value is a multiple of the rated reverse RPM. If the rated reverse RPM is -4000 rpm and the maximum reverse RPM needs to be -6000 rpm, then fill in -1.5 times here. (This value must be negative)

- **Rated positive speed**

The rated positive speed of the robot joint; it is automatically calculated from the rated positive RPM, encoder bits and the reduction ratio (the axis 3 of the 4-axis SCARA and axis 1 of the 4-axis SCARA special-shaped robot also need to add the pitch), no need to fill in.

- **Rated reverse speed**

The rated reverse speed of the robot joint; it is automatically calculated from the rated reverse RPM, encoder bits and the reduction ratio, no need to fill in. (This value must be negative)

- **Maximum acceleration**

The maximum acceleration of the robot joint movement; its value is a multiple of the rated positive (reverse) speed. If the rated positive speed is 300 degrees/s, the maximum acceleration needs to be 1500 degrees/s^2, then fill in 5 times here.

- **Maximum deceleration**

The maximum deceleration of the robot joint movement; its value is a multiple of the rated positive (reverse) speed. If the rated positive speed is 300 degrees/s, the maximum acceleration needs to be 1200 degrees/s^2, fill in -4 times here. It is recommended that the maximum acceleration and maximum deceleration values be the same. (This value must be negative)

- **Model direction**

The model direction should be set by referring to the joint positive direction diagram below, and the direction of the jogging "+" key of each axis should be the same as the joint positive direction diagram (choosing 1 for the same and -1 for the opposite)

- **Actual joint direction**

Set the actual direction of the joint relative to the model direction (joint positive direction legend below), choose 1 for the same direction as the model direction and -1 for the opposite direction, set it according to the actual needs.

Note:

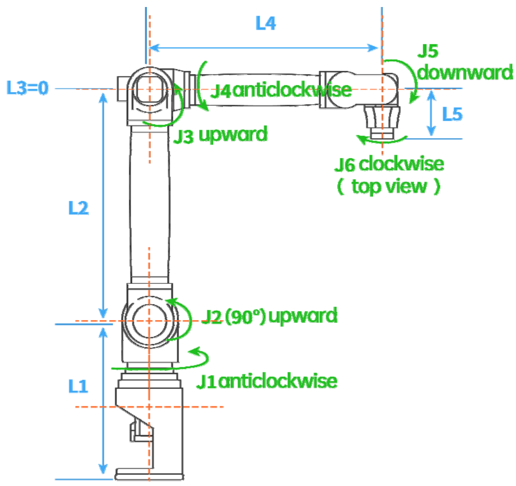
1. Please configure the model direction first and then configure the actual direction of the joint
2. To use the "Actual joint direction" function, you need to go to [Settings] - [Operation parameters] interface, turn on the button after the "Actual joint direction"
3. The "Actual joint direction" is in the [Settings] - [Robot parameters] - [Joint parameters] - [Other parameters] interface

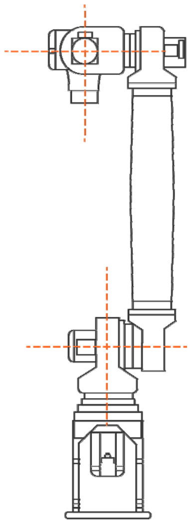
- **Gear backlash**

The angle to compensate for the filled value whenever the joint moves in the opposite direction

Note: The "Gear backlash" is in the [Settings] - [Robot parameters] - [Joint parameters] - [Other parameters] interface

Joint positive direction diagram:





Robot type	Axis	Positive direction (top view or left view)
6-axis	J1	anticlockwise
	J2	upward
	J3	upward
	J4	anticlockwise
	J5	downward
	J6	clockwise
4-axis SCARA	J1	anticlockwise
	J2	anticlockwise
	J3	upward
	J4	clockwise

4-axis palletizing	J1	anticlockwise
	J2	upward
	J3	upward
	J4	anticlockwise
4-axis joint	J1	anticlockwise
	J2	upward
	J3	upward
	J4	upward
5-axis joint	J1	anticlockwise
	J2	upward
	J3	upward
	J4	anticlockwise
	J5	downward
2-axis SCARA	J1	anticlockwise

	J2	anticlockwise
3-axis SCARA	J1	anticlockwise
	J2	anticlockwise
	J3	downward
single-axis	J1	anticlockwise

Multi-turn value overflow counting

You can go to [Settings] - [Robot parameters] - [Joint parameters] to modify the joint parameters.

This function is used to eliminate the effect of jumps between encoder max/min values.

For example, the encoder multi-turn value range is [-2147483648,2147483647], the current encoder multi-turn value position is 2147483647, if you rotate 1 unit in the positive direction, the position will be -2147483648, if the system does not know the encoder multi-turn value range, it will think that the robot suddenly jumps, and will not know that only 1 unit was actually rotated, then the robot is prone to be out of control.

Warnings

⚠

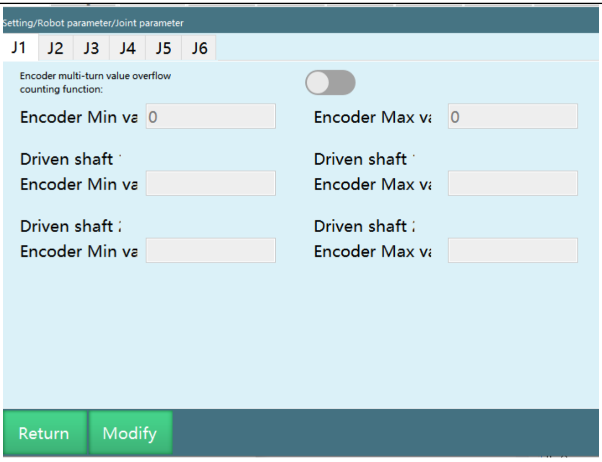
- This parameter must be filled in, otherwise the following problems may occur

  ——Large jumps in points, such as a sudden change from 4 degrees to 40 degrees

  ——Out of control

- If the slave axis is configured, the encoder max/min value of the slave axis

must also be filled in



Encoder multi-turn value overflow counting function: turn on the button for this joint to use this function.

Minimum value of multi-turn value: calculate and fill in by yourself according to the body servo parameters (negative value is required).

Maximum value of multi-turn value: calculate and fill in by yourself according to the body servo parameters.

Calculation method:

Maximum value of multi-turn value=$2^{\text{digits of single-turn value+digits of multi-turn value}}-1$

(The maximum value should not exceed 2^{31}-1=2,147,483,647, if it exceeds, fill in the maximum value)

Minimum value of multi-turn value=$-2^{\text{digits of single-turn value+digits of multi-turn value}}$

(The minimum value should not exceed -2^{31}=-2,147,483,648, if it exceeds, fill in the minimum value)

Notes

- There are two encoder multi-turn value ranges

——$[-2^{31}, 2^{31}-1]$

——$[0, 2^{32}-1]$

Robot zero position

Warnings

⚠

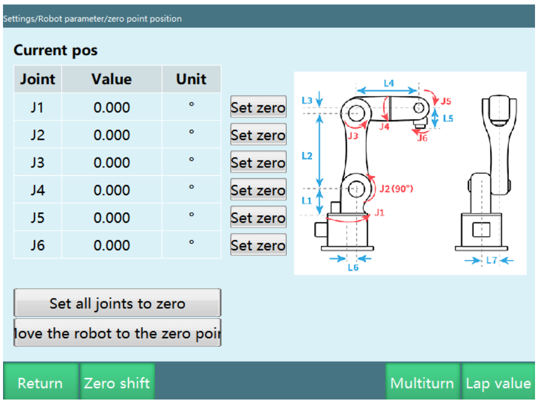
- After modifying the zero position, all job programs are unavailable, please do not use
- Zero point calibration requires correct slave configuration and joint parameters configuration

Zero calibration

If the robot zero position is a non-standard zero position, users can align the robot according to the robot's alignment hole, and then set the current robot position coordinates to the zero position on the robot zero position interface.

The specific operation steps are as follows:

1. Open the [Settings] - [Robot parameters] - [Zero position] interface.
2. When the "joint coordinate mode" is set, the attitude of each joint of the robot at the zero position is as shown in the following figure, in which the lower arm is in a vertical state, the forearm is in a horizontal state, and the wrist (fifth joint) is also in a horizontal state. In general robots, the zero position interface (such as grooves, scribe lines, rulers, etc.) has been considered in the body design process.
3. Click the [Set to zero] button corresponding to the axis you want to set the zero point, or set all joint coordinates to zero point at once by clicking the [Set all joints to zero] button.
4. In the modification prompt box that pops up, click [OK] to set the robot zero points as shown in the figure.



5. The zero position of this axis (all axes) is set successfully.

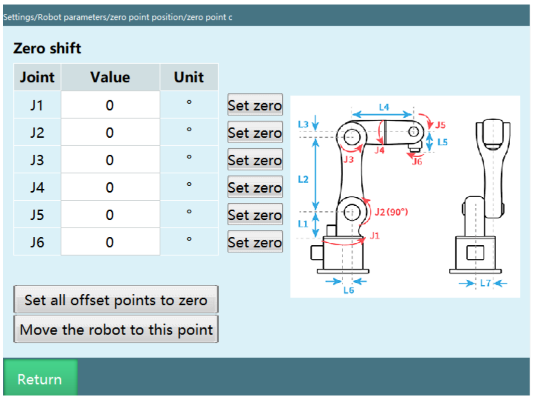
- In the servo-ready state, press the DeadMan button and then press [Move robot to the zero point] to ensure that the robot is safe.
- The speed value is automatically adjusted to 5% and can be manually adjusted to increase the movement speed.
- After setting the current position to zero, the axis coordinates of the current position become (0,0,0,0,0,0).
- You can set the current position coordinates of one or more axes to the zero point coordinates, at this time, the zero point coordinates of unset axes are the original zero point coordinates.

Warnings

- Teaching and playback operations cannot be performed without calibration of the origin position.
- For systems that use multiple robots, each robot must perform origin position calibration.
- When there is a coupling relationship between joint axes, such as the common coupling relationship between the fifth axis and the sixth axis of a robot, the fifth axis must be at the zero position, then the zero data recorded for the sixth axis will be valid, otherwise, the zero data recorded for the sixth axis will be invalid. So the zero data of the sixth axis must be recorded with the fifth axis at the zero position. If there is no coupling relationship, each axis can calibrate the zero position individually, and the respective zero position will not affect the zero position of other joints.

Zero offset

The zero offset can be used when the user needs to adjust the zero point, you can enter the value manually and the operation method is similar to the zero point calibration.



Clear multi-turn value

Warnings

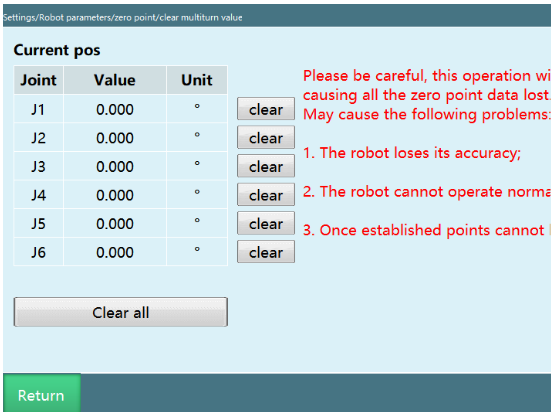
- Please operate with caution. This operation will cause the robot encoder value to be cleared, resulting in the factory zero data being cleared.

  May cause the following problems:

  i. The robot loses accuracy;

  ii. The robot can't operate properly;

  iii. The points ever established are not working.



Clear all axes' multi-turn values: clear all axes' multi-turn values for the robot at once (excluding external axes)

"Clear" after each joint: clear the multi-turn value for that axis

Single-turn value

This function can modify the single-turn value corresponding to each axis

Warnings

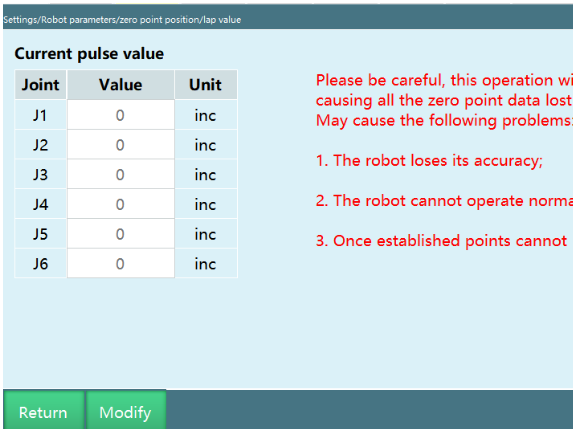
- Please operate with caution. This operation will cause the robot encoder value to be cleared, resulting in the factory zero data being cleared.

  May cause the following problems:

  i. The robot loses accuracy;

  ii. The robot can't operate properly;

  iii. The points ever established are not working.



How to retrieve the lost zero point

Prerequisites:

1. The robot loses the zero point just because of an operation error, and the lost zero point cannot be retrieved in the event of a collision
2. Recorded the single-turn value before the zero point was lost (when the multi-turn value is not cleared, the value displayed in the single-turn value interface is the data of the last zero calibration)

Steps:

1. Find the single-turn value recorded before the zero point is lost for backup
2. Teach the robot to the mechanical zero position
3. Clear the multi-turn value of all axes of the robot (this operation will clear the multi-turn values and single-turn values, please operate with caution)
4. Calibrate the zero point of all axes of the robot
5. Enter the single-turn value data prepared in step 1 in the single-turn value interface
6. Return the robot to zero point
7. Confirm that the zero point is correct

DH parameters

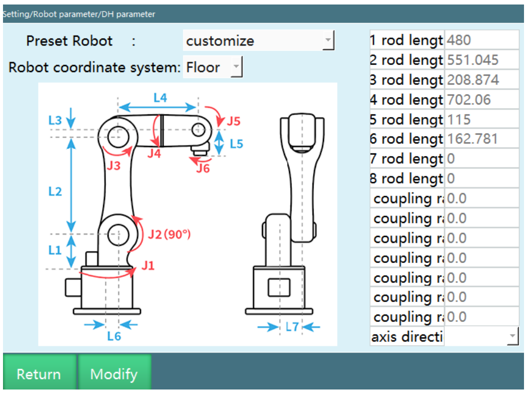
Warnings

- Please do not switch to servo-ready mode, power on, run and do other operations without configuring this parameter
- Zero calibration needs to be configured before configuring DH parameters

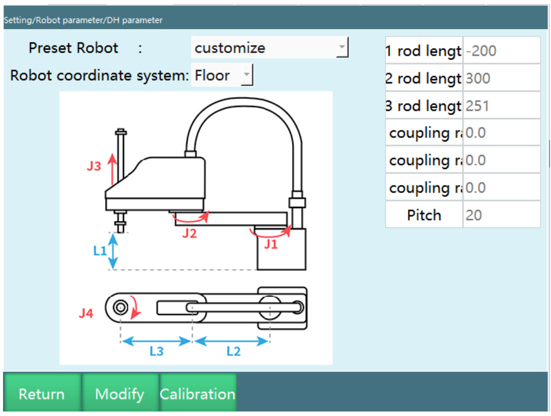
The relevant steps are as follows:

1. Enter the [Settings] - [Robot parameters] - [DH parameters] interface;

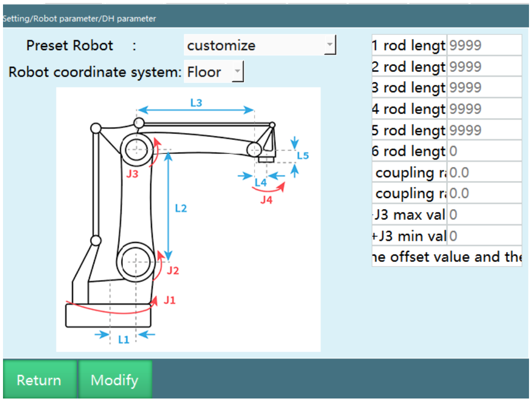
6-axis tandem multi-joint:



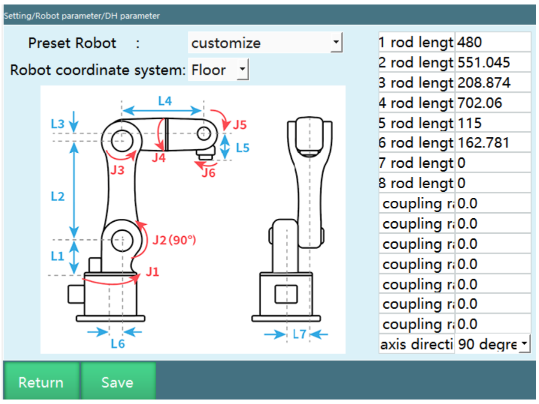
4-axis SCARA/4-axis SCARA special-shaped 1:



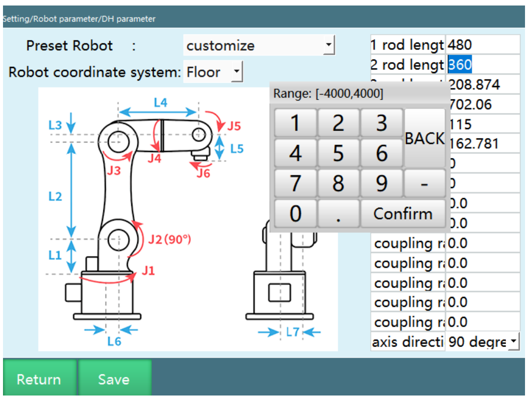
4-axis linkage palletizing:



2. Click the [Modify] button at the bottom.
3. Fill in the parameter values according to your robot.



4. Click the parameter value you want to modify (such as L2), then the soft keyboard will appear, enter the number you want to replace, and click "OK".



5. Click the [Save] button to complete the parameter modification.

Meaning of each parameter

- **Preset robot**

By importing the robot joint parameters and DH parameters into the controller in advance, you can eliminate the need to fill in the parameters repeatedly

Note: For specific usage, please contact the manufacturer

- **Robot coordinate system**

floor mounting

ceiling mounting

- **Rod length**: robot size
- **Coupling ratio**: some robot bodies are designed so that the motor spans many axes to drive a particular axis, which creates a coupling between the two axes. For example, if we rotate axis 2, axis 3 follows, which is axis coupling. To counteract this coupling effect, a coupling ratio is needed.

The calculation formula for the coupling ratio is:

$$\text{coupling ratio}=\frac{following\ axis\ rotation\ angle}{main\ axis\ rotation\ angle}$$

For example, if we rotate axis 2 by 10° and find that axis 3 follows the rotation by 15°, then the coupling ratio is

$$\frac{15}{10}=1.5$$

- **Axis 5 direction**: direction of axis 5 at zero point calibration

Horizontal direction

Vertical direction

- **Pitch**

Pitch of the link responsible for up and down movement in 4-axis SCARA (axis 3 of 4-axis SCARA and axis 1 of 4-axis SCARA special-shaped robot)

- **J2+J3 min/max**

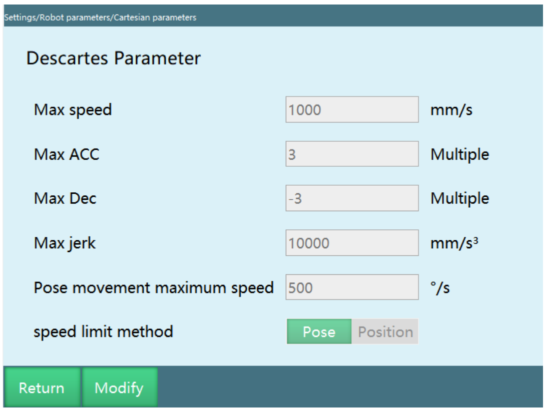
Let the axis 2 and axis 3 of the 4-axis palletizing robot move to J2max/J3max, J2min/J3max, J2max/J3min and J2min/J3min respectively, and record the values of J2+J3 in the four cases. Remove the highest and lowest values of the four, and the remaining two are the **J2+J3 min/max values**.

Cartesian parameters

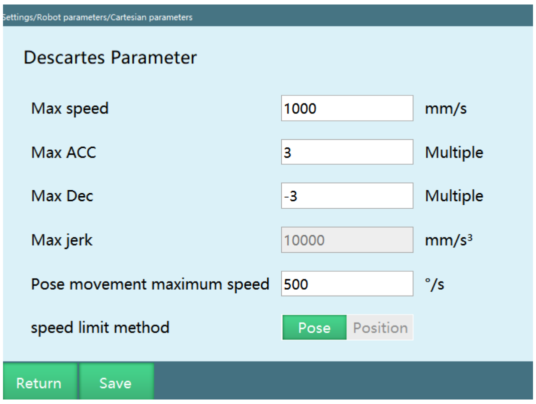
You can go to [Settings] - [Robot parameters] - [Cartesian parameters] to modify the Cartesian parameters.

The relevant steps are as follows:

1. Enter [Settings] - [Robot parameters] - [Cartesian parameters] interface.
2. The input box is grayed out and no value can be entered.



3. After clicking "Modify", the "Modify" button becomes "Save" and the input box turns white, allowing you to enter values after the respective parameters.



4. Click "Save" and the modification is successful.

Meaning of each parameter

- **Maximum speed**

The maximum linear speed of the robot during operation.

- **Maximum acceleration**

The maximum acceleration of the robot during operation; its value is a multiple of the maximum speed. If the maximum speed is 1000mm/s and the maximum acceleration needs to be 3000mm/s², then fill in 3 times here.

- **Maximum deceleration**

The maximum deceleration of the robot during operation; its value is a multiple of the maximum speed. If the maximum speed is 1000 mm/s and the maximum deceleration needs to be -3000 mm/s^2, then fill in -3 times here. It is recommended that the maximum acceleration and maximum deceleration values be the same and the same as the maximum acceleration and maximum deceleration in the joint parameters. (This value must be negative)

- **Maximum jerk**

This parameter is a reserved parameter and is currently invalid.

- **Pose movement maximum speed**

The maximum speed of the robot during attitude movement, if the instruction speed exceeds this value, it will be decelerated

- **Speed limit method**

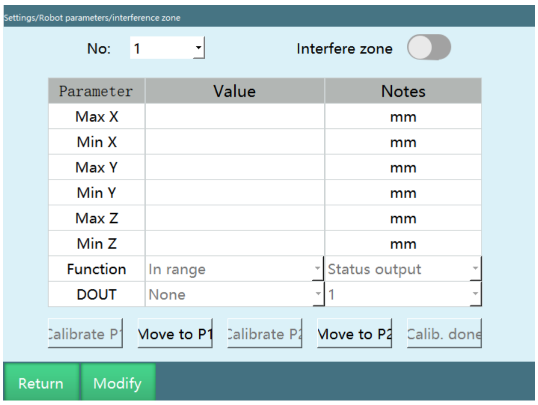
Pose: The robot's linear interpolation motion is limited by both the maximum speed and the pose movement maximum speed

Position: The robot's linear interpolation motion is limited only by the maximum speed

Interference area range setting

The robot range limit is used to limit the robot's range of motion. It can be set in two ways: "manual fill" and "calibration".

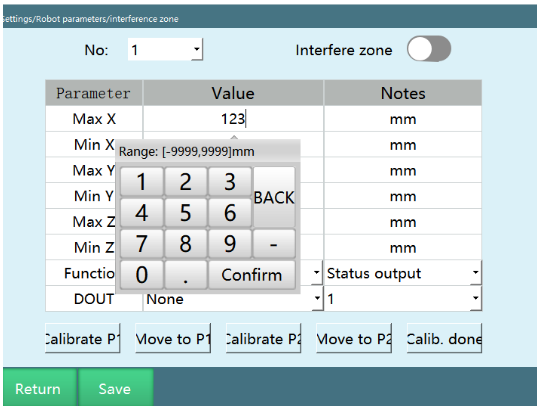
Interface:



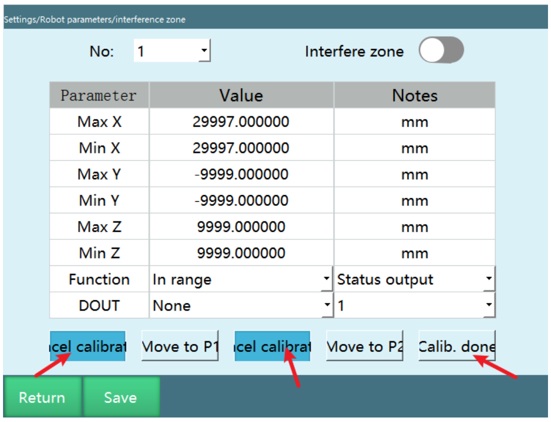
Button	Meaning
Min X	Motion range X-axis minimum value
Max X	Motion range X-axis maximum value
Min Y	Motion range Y-axis minimum value

Max Y	Motion range Y-axis maximum value
Min Z	Motion range Z-axis minimum value
Max Z	Motion range Z-axis maximum value
Process number	9 process numbers can be selected, you can use multiple process numbers at the same time
Interference area enable	When this switch is turned on, the interference area takes effect
Function	When the robot is inside\outside the area, prohibit the robot from running\trigger status output Inside the area-state output: when the robot is inside the interference area, trigger status output Outside the area-state output: when the robot is outside the interference area, trigger status output Inside the area-no motion: when the robot is inside the interference area, prohibit it from running Outside the area-no motion: when the robot is outside the interference area, prohibit it from running
Output IO	Valid when the "Function" option is "state output"; output the interference area state
"Calibrate P1" and "Calibrate P2"	Calibrate the maximum and minimum values of the range
"Move to P1" and "Move to P2"	Move the robot to the calibrated position
Calibration completed	Automatic calculation of maximum and minimum values

When setting the range using the manual fill method, the maximum and minimum coordinate values that the robot can move in the X, Y, and Z axes can only be set after clicking the "Modify" button.



When setting the range using the range calibration method, you can move the robot and click on [Calibrate P1] and [Calibrate P2] on the interface to determine the maximum and minimum values, and then click on "Calibration completed" after the calibration is completed.

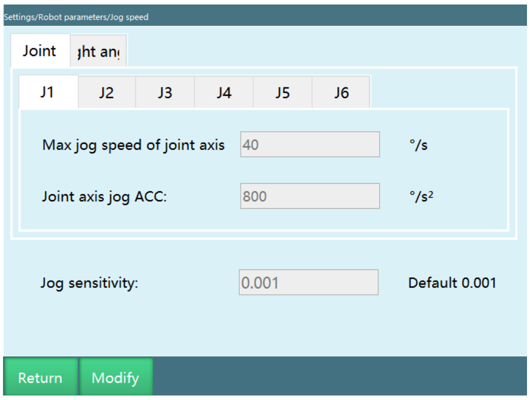


Jog speed

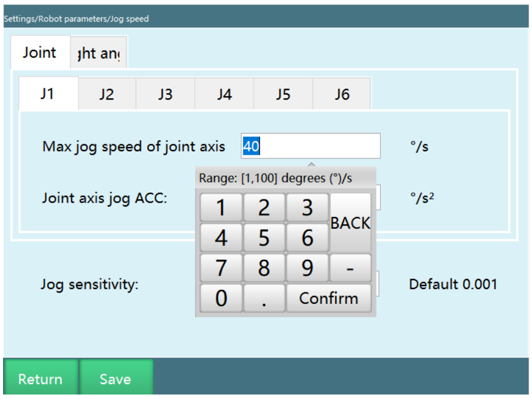
You can go to [Settings] - [Robot parameters] - [Jog speed] to modify the jog speed.

The relevant steps are as follows:

1. Enter the [Settings] - [Robot parameters] - [Jog speed] interface.
2. The input box is grayed out and no value can be entered.



3. After clicking "Modify", the "Modify" button becomes "Save" and the input box turns white, allowing you to enter values after the respective parameters.



4. Click "Save" and the modification is successful.

Meaning of each parameter

- **Joint axis maximum jog speed**

omitted

- **Joint axis jog acceleration**

omitted

- **Cartesian coordinate maximum jog speed**

omitted

- **Cartesian coordinate jog acceleration**

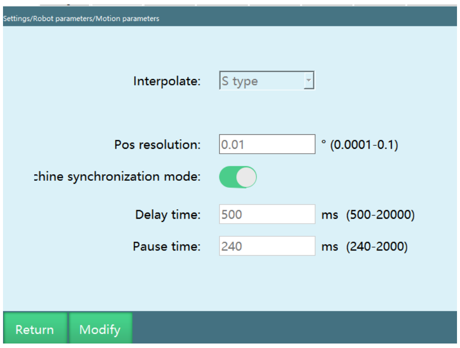
omitted

- **Jog sensitivity**

After power on, the jog operation is invalid when the jitter range of the robot is greater than the jog sensitivity

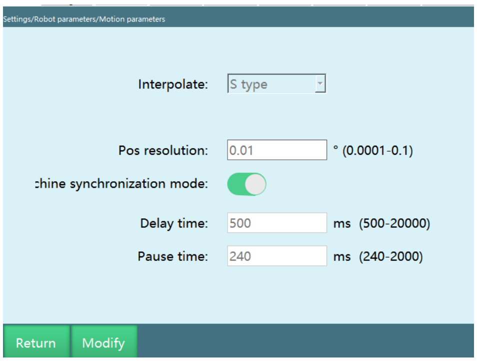
Motion parameters

[Motion parameters] interface provides two robot motion interpolation methods, as shown in the following figure.



**"Enable dual-robot synchronization mode" switch is only valid when both robots are 6-axis tandem multi-joint robots*

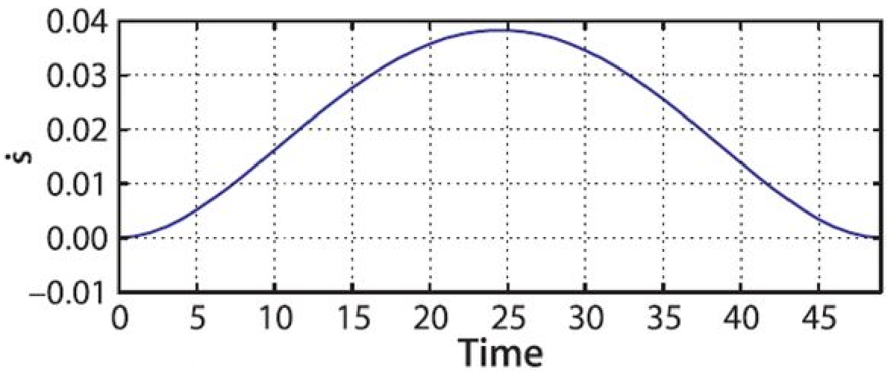
1. Click the [Modify] button;
2. Select robot interpolation method;

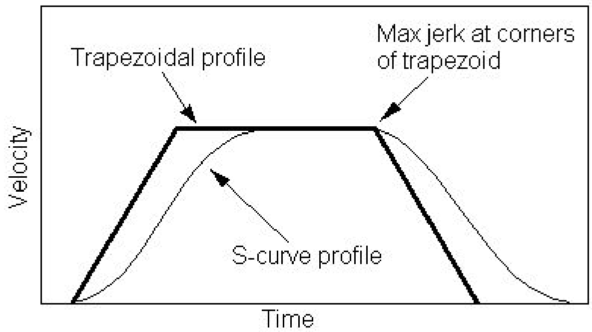


3. Click the [Save] button.

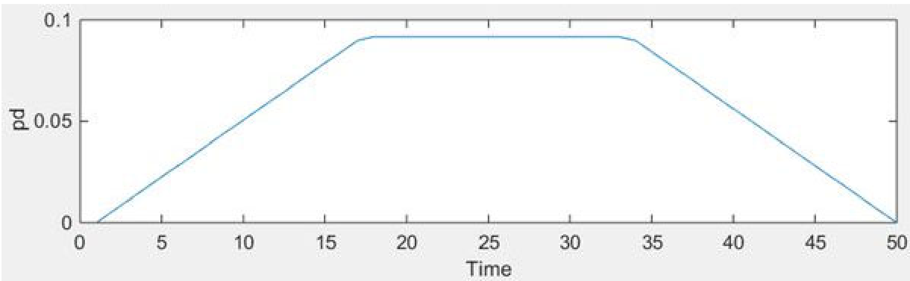
Meaning of each parameter

- **S-shaped interpolation**





- **Trapezoidal interpolation**



- **Remote mode speed**

Remote mode actual running speed = remote mode speed * global speed

- **Absolute position resolution**

When the operating point is 2 points whose difference is less than the resolution, it will be executed as 1 point

- **Enable dual-robot synchronization mode**

Dual-robot mode switch, when off, is multi-robot mode, two 6-axis tandem multi-joint robots are independent of each other; when on, is dual-robot mode, can use dual-robot instruction in robot 1 program to control robot 2 collaboration. Turn off dual-robot collaboration and restart the controller to take effect

Note: This function is displayed when two 6-axis tandem multi-joint robots are configured in the slave station.

- **Run delay time**

Run delay at program startup

- **Pause time**

The time taken from running to stop when the program is stopped due to mode switch, paused due to mode switch, remotely stopped, or remotely paused during operation

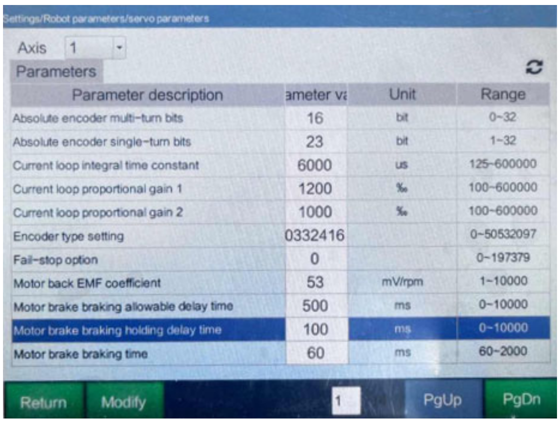
Servo parameters

Warnings

- Please modify carefully and test in a safe area after modification

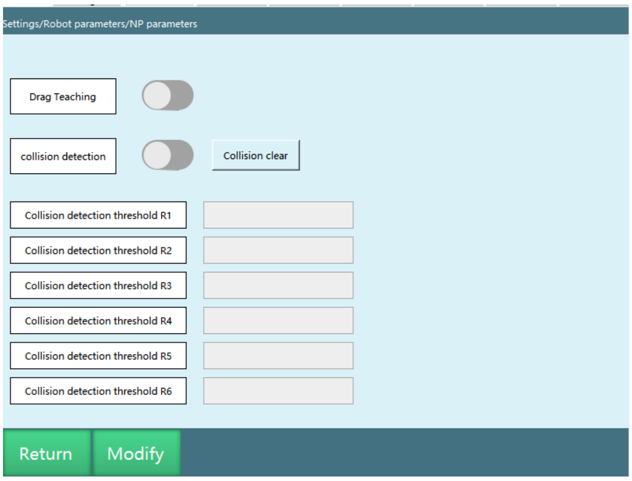
Open [Settings] - [Robot parameters] - [Servo parameters] interface, you can view and modify the parameters of the servo in this interface.

At present, we can only read and modify the parameters of Tsino-Dynatron and HUACHENG servos.



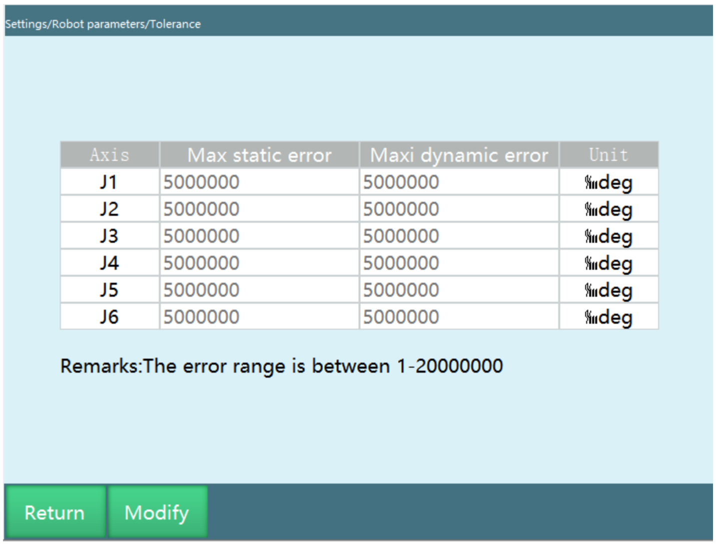
NP parameters

Open [Settings] - [Robot parameters] - [NP parameters] interface, there are functions such as drag teaching and collision detection for human-machine collaborative robots



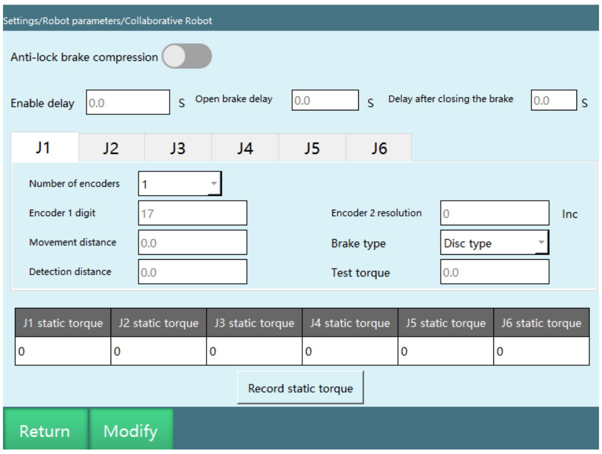
Following error

Open [Settings] - [Robot parameters] - [Following error] interface, you can set the maximum static error, maximum dynamic error, etc., unit □ , range 1 - 20000000.



Collaborative robot

This interface is the parameter setting interface for collaborative robots, other types of robots do not need to set.



Enable delay: The delay time after pressing the enable key before issuing the enable instruction to the servo

Brake open delay: The delay time after issuing the enable instruction before issuing the brake open instruction to the servo

Delay after the brake is closed: The delay time elapsed from the closing of the brake until the servo responds to the next operation

Number of encoders: The number of encoders in single joint

Encoder 1 bits: The same as the encoder bits in the joint parameters

Encoder 2 resolution: The inc value of another encoder in single joint

Movement distance: The jogging distance of the joint before the brake is opened, generally 20; the value is the encoder value, the unit is inc.

Brake type: Brake disc brake and pin-type brake; the value is the encoder value, the unit is inc.

Detection distance: The joint movement distance used to detect whether the brake is open after opening the brake

Detection torque: After opening the brake, if the torque exceeds the detection torque when the joint runs detection distance, it is considered that the brake has failed to open.

External Axis Parameters

Joint parameters

Same as robot joint parameters configuration, please refer to robot joint parameters for configuration

Warnings

- Please do not switch to servo-ready mode, power on, run and do other operations without configuring this parameter

Zero position

Same as the robot zero position configuration, please refer to the robot zero position for configuration

Warnings

- Please do not switch to servo-ready mode, power on, run and do other operations without configuring external axis joint parameters.

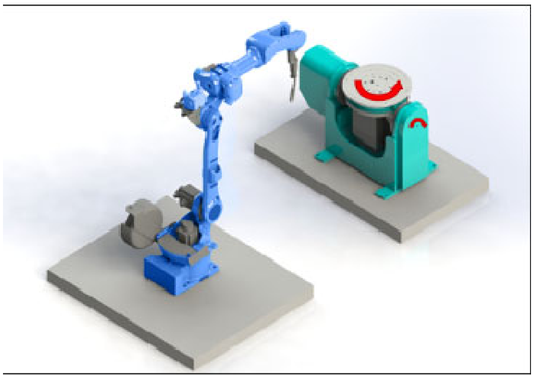
External axis calibration

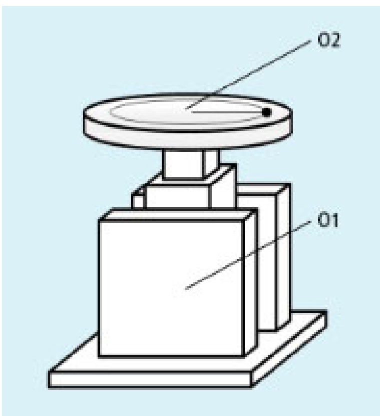
Notes

- Make sure the robot tool hand is calibrated before calibrating the external axes
- Jog to verify that the external axis parameters are accurate

The [External axis calibration] interface contains the calibration status of the set external axis group and the current collaborative external axis group number.

External axis positive direction:





O1 (lower flip axis): The positive direction is the direction facing away from the robot

O2 (upper rotation axis): The positive direction is anticlockwise from top to bottom

External axis calibration:

- **Ground rail calibration**: After setting the ground rail joint parameters and rack and pinion ratio, turn on the collaboration switch and consider it calibrated.

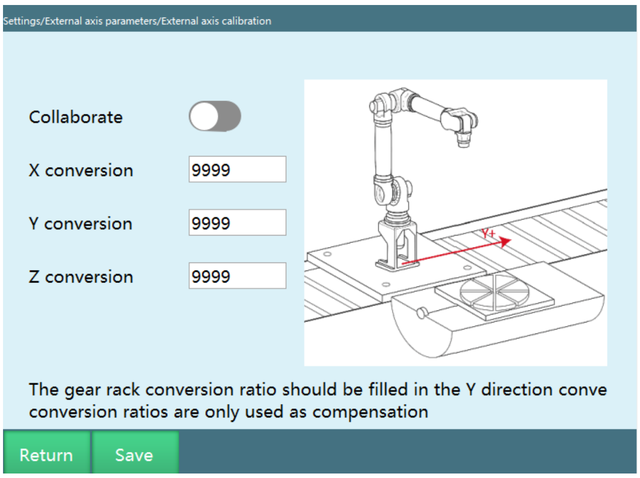
X-direction conversion ratio: error value; rotate the ground rail by 360º, fill in the X-axis direction error here. Fill in 0 if there is no error.

Y-direction conversion ratio (rack and pinion ratio): rotate the ground rail by 360º,fill in the Y-axis movement value here.

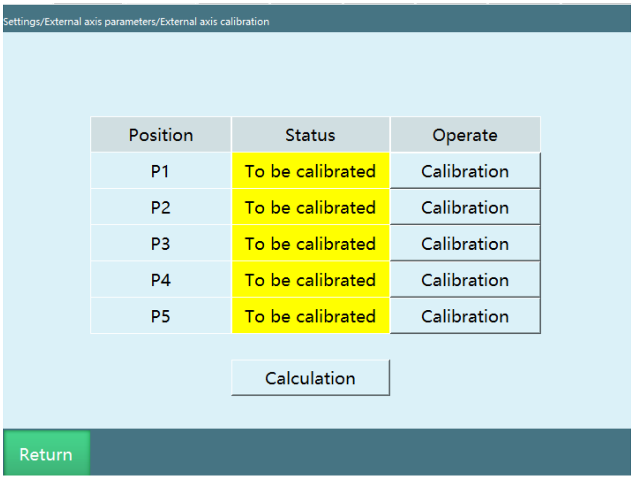
Z-direction conversion ratio: error value; rotate the ground rail by 360º, fill in the Z-axis direction error here. Fill in 0 if there is no error.

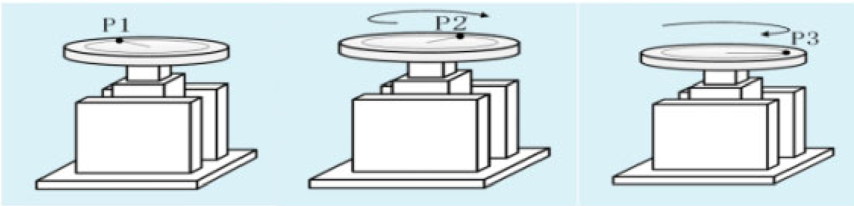
Notes

- The positive direction of the ground rail is parallel to the Y+ direction of the robot
- Y-direction conversion ratio calculation method: first fill in 360 for the pitch, measure the actual movement distance (mm) when the axis moves 360º, and fill in this distance into the pitch.
- The robot can't be powered on when the XYZ direction conversion ratio is 9999

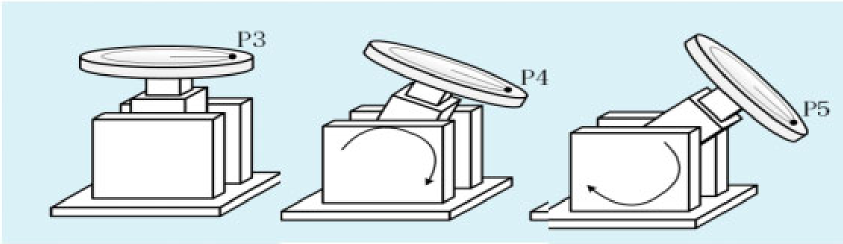


- Dual-axis positioner calibration:





1. Return the external axis to zero point and find a point on the turntable as the reference point A
2. P1: Rotate the external axis 2 about 100 degrees in the positive direction. At this time, the reference point A is P1 in the figure, move the robot end to P1, and calibrate P1
3. P2: Rotate the external axis 2 about 50 degrees in the reverse direction. At this time, the reference point A is P2 in the figure, move the robot end to P2, and calibrate P2
4. P3: Return the external axis to zero point. At this time, the reference point A is P3 in the figure, move the robot end to P3, and calibrate P3



5. P4: Rotate the external axis 1 about 25 degrees in the positive direction. At this time, the reference point A is P4 in the figure, move the robot end to P4, and calibrate P4
6. P5: Rotate the external axis 1 about 25 degrees in the positive direction again. At this time, the reference point A is P5 in the figure, move the robot end to P5, and calibrate P5
7. Click "Calculate"

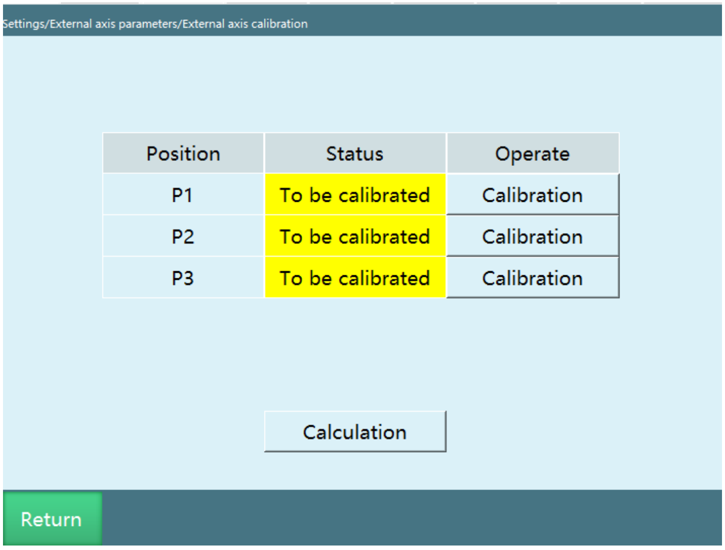
- Single-axis flip positioner calibration:

1. Return the external axis to zero point and find a point on the platform as the reference point A
2. Rotate the external axis 50 degrees in the positive direction, at this time, A is P1, move the robot end to point P1, and calibrate P1
3. Rotate the external axis 25 degrees in the opposite direction, at this time, A is P2, move the robot end to point P2, and calibrate P2
4. Return the external axis to zero point, at this time, A is P3, move the robot end to point P3, and calibrate P3

- Single-axis rotary positioner calibration:

1. Return the external axis to zero point and find a point on the platform as the reference point A
2. Rotate the external axis 100 degrees in the positive direction, at this time, A is P1, move the robot end to the P1 point, and calibrate P1
3. Rotate the external axis 50 degrees in the opposite direction, at this time, A is P2, move the robot end to point P2, and calibrate P2

4. Return the external axis to zero point, at this time, A is P3, move the robot end to point P3, and calibrate P3



Notes on external axis instructions:

-When there are multiple groups of positioners, the robot can only cooperate with one group of positioners at the same time, and the currently coordinated external axis group can be switched through the coordinate system class - switch external axis.

-MOVJEXT (external axis point-to-point): Select two points on the external axis, and insert E001 and E002 for robot alignment point (the E point coordinates include the position data of the robot and the external axis).

-MOVLEXT (external axis straight line): Select two points on the external axis, and insert E001 and E002 for robot alignment point (the E point coordinates include the position data of the robot and the external axis); when inserting, select "Yes" for "SYNC" synchronization, if the external axis is not calibrated or collaboration group numbers is not selected, then Synchronous operation cannot be enabled.

-MOVCEXT (external axis arc): Select three points on the external axis, insert MOVJEXT or MOVLEXT at the first point; insert E001, E002, E003 for robot alignment point (E point coordinates include the position data of robot and external axis); when inserting, select "Yes" for "SYNC" synchronization, if the external axis is not calibrated or collaboration group numbers is not selected, then Synchronous operation cannot be enabled.

Jog speed

Same as robot jog speed configuration, please refer to robot jog speed for configuration

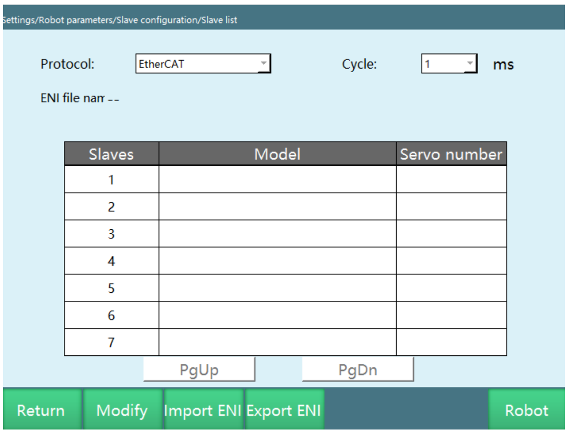
New Robot Configuration Steps

- When you get a new control system, please first configure the number of robots, robot type, robot servo type, external axis type, external axis servo type and IO model, otherwise the error message "Cannot connect to servo" will appear after power on, and the servo cannot be used.
- Please configure the number of robots, robot type, robot servo type, external axis type, external axis servo type and IO model strictly according to your actual wiring. If you confirm that you have strictly followed the actual wiring, but the error message "Cannot connect to servo" still appears, please contact our technical support staff and provide the servo model and IO model you are using.
- When the servo type and IO model are not configured correctly, it will take a while to connect the controller and the teach pendant after the system is started. Therefore, if "Disconnected" is displayed on the top of the teach pendant after the system is started, this is a normal phenomenon.

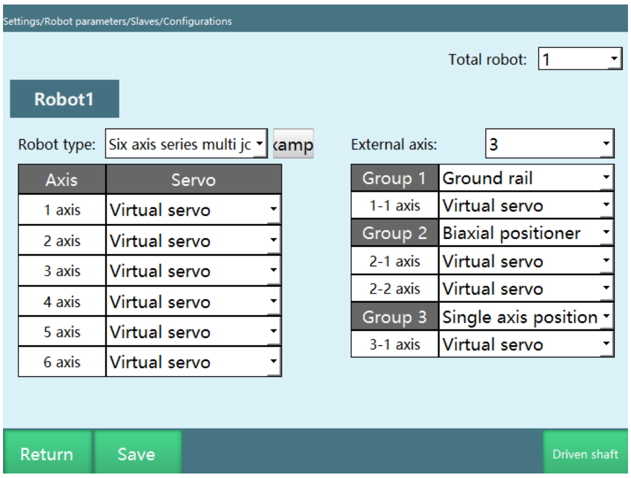
The following are the complete parameter configuration steps:

1. Switch permissions to "Admin", the default password is 123456;
2. Configure the number of robots, robot communication cycle, robot type, and servo model in "Settings - Robot parameters - Slave configuration"; (please select the correct robot model, otherwise the robot will not be able to move normally!)

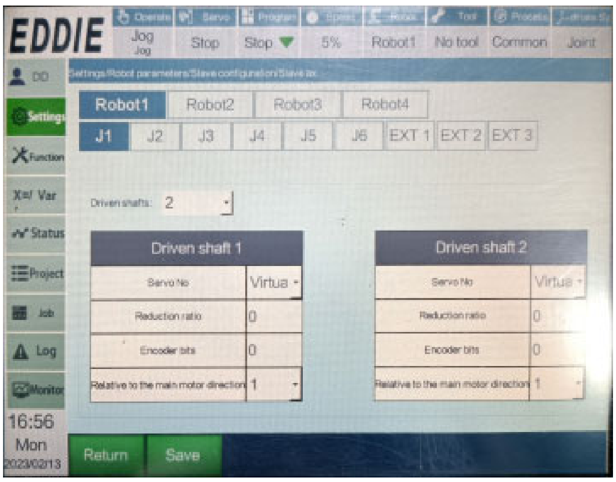
The servo list displays the number and model of servos read after the current controller is turned on, and the communication cycle can be set in this interface.



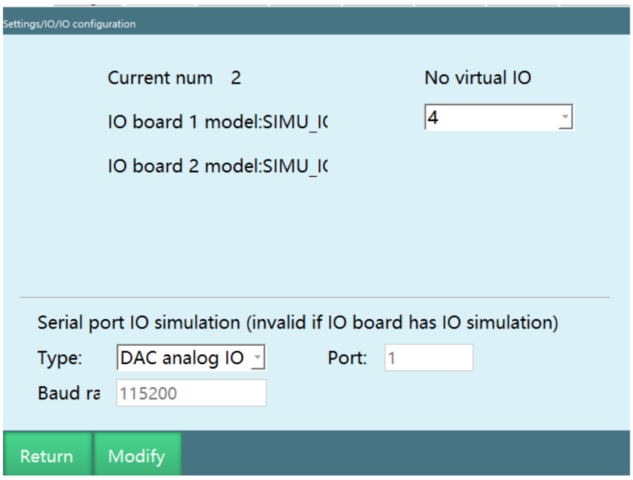
In the robot servo configuration interface, you can configure the number of robots, robot type, number of external axis groups and do servo selection



In the slave axis setting interface, you can set the number of slave axes and the slave axis servo

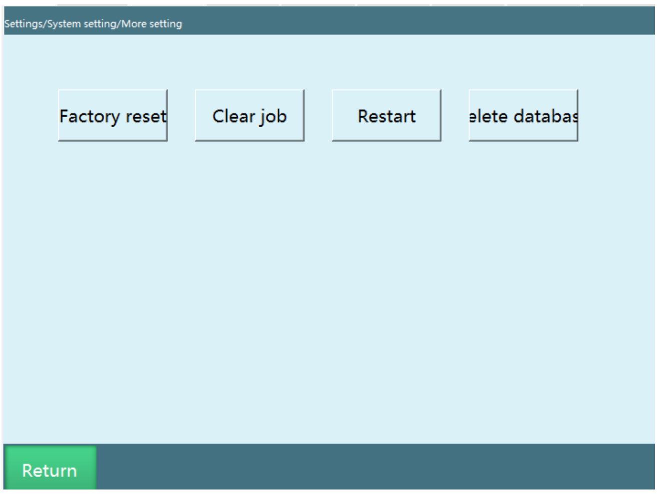


3. Configure the serial port analog IO type and the number of virtual IO in "Settings-IO-IO configuration", no settings are needed for normal EtherCAT IO;



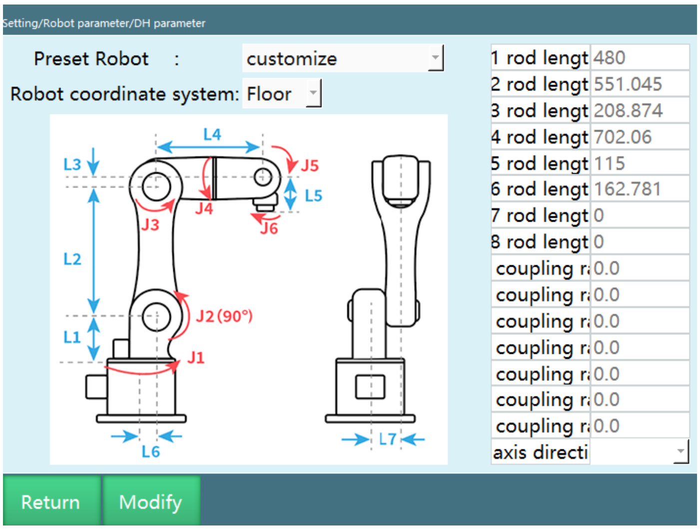
Note: The ENI file is slightly different when using HUATAI IO

4. Restart the system (After the robot configuration is modified, it will take effect after restarting);



5. In the "DH parameters" interface, we provide "Preset robot" function. If this drop-down list contains the robot model you are using, you can use this function to set up the parameters of the robot quickly and easily.

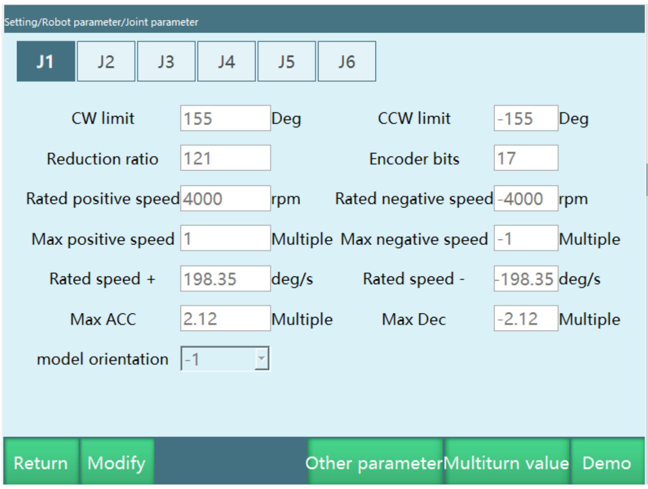
6. Select the robot coordinate system according to the actual assembly (ceiling mounting: Cartesian coordinates, tool coordinates, user coordinates; same operating habits as floor mounting)



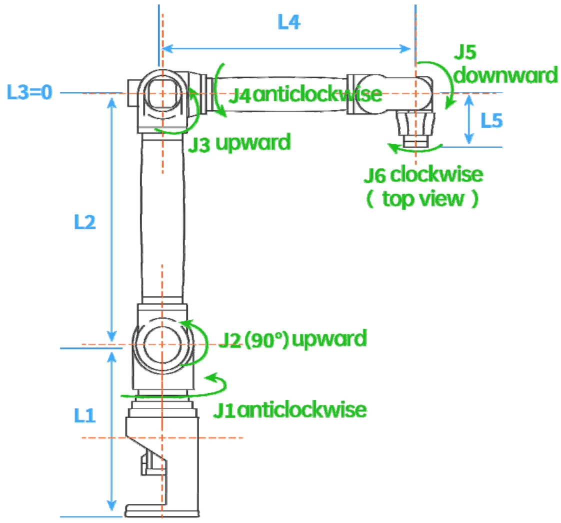
7. Click [Preset robot] in the upper left corner of the "DH parameters" interface, you can select the robot model that has been adapted, and the DH parameters and joint parameters of the robot will be automatically filled in after selection.

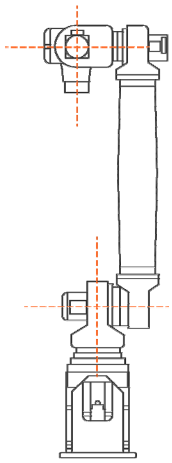
8. You need to modify the zero point manually after selecting the preset robot.

9. If your robot is not listed in this option, please follow the steps below to fill in the parameters manually;



10. Fill in the parameters in "Settings-Robot parameters-Joint parameters", and set the limit of each joint to (-3000, 3000); (Please jog each axis of the robot individually to check whether the positive direction of each axis of the robot is correct!)





Robot Type	Axis	Positive direction
6-axis	J1	anticlockwise
	J2	upward
	J3	upward
	J4	anticlockwise
	J5	downward
	J6	clockwise

4-axis SCARA	J1	anticlockwise
	J2	anticlockwise
	J3	upward
	J4	clockwise
4-axis palletizing	J1	anticlockwise
	J2	upward
	J3	upward
	J4	anticlockwise
4-axis joint	J1	anticlockwise
	J2	upward
	J3	upward
	J4	upward

5-axis joint	J1	anticlockwise
	J2	upward
	J3	upward
	J4	anticlockwise
	J5	downward
2-axis SCARA	J1	anticlockwise
	J2	anticlockwise
3-axis SCARA	J1	anticlockwise
	J2	anticlockwise
	J3	downward
single-axis	J1	anticlockwise
4-axis SCARA special-shaped	J1	upward
	J2	anticlockwise
	J3	anticlockwise
	J4	clockwise

11. Set the robot zero point in "Settings - Robot parameters - Zero position". If the zero position of the robot is axis 5 vertically downward, please select "90º" for "Axis 5 direction" in the last line of the "DH parameters" interface. If it is horizontal, then select "0º" for "Axis 5 direction" in the "DH parameters" interface;

12. Set the joint limits of each axis according to the actual operating environment in "Settings - Robot parameters - Joint parameters".

13. Fill in the "Settings - Robot parameters - Joint parameters" according to the actual parameters of the robot, where the acceleration and deceleration can be set to 4-6 times the maximum positive and negative speed;

14. Check whether the parameters in the "Cartesian parameters", "Jog speed", and "Motion parameters" interfaces are correct.

System settings

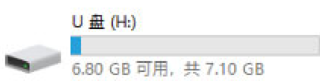
This chapter will mainly introduce how to check and upgrade the software version, set the system date and time, and set the controller IP.

Make a U disk in FAT32 format

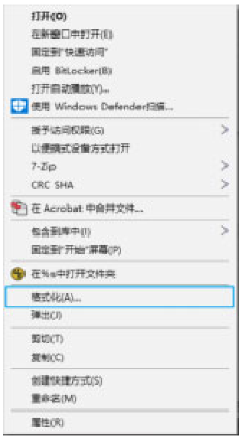
To upgrade programs, import and export parameters and procedures in this system, a U disk in FAT32 format is required. The steps to make a U disk in FAT32 format are as follows:

1. Prepare a computer and a U disk, please note that the production process will empty all the contents of the U disk, this is an irreversible step, so please back up the contents of the U disk;
2. Insert the U disk into the USB port of the computer;
3. Open "My Computer" on your computer or "This PC" interface in Windows 10 system;

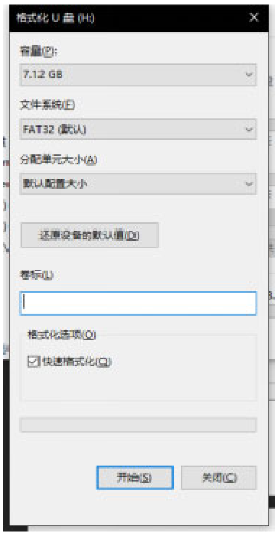
4. At this point there should be a U disk drive letter , if not, please re-plug the U disk, if it still does not appear, please replace the U disk;



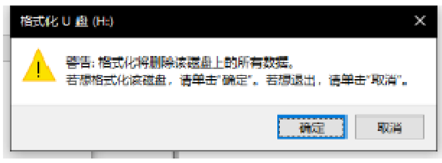
5. Right-click the drive letter and click "Format" in the menu that appears;



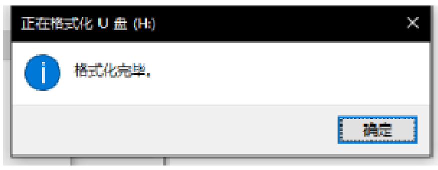
6. The settings in the pop-up interface are as shown below;



7. Click the [Start] button and click the [OK] button in the confirmation box that pops up;



8. When the "Format Complete" window pops up, the U disk in FAT32 format is created.

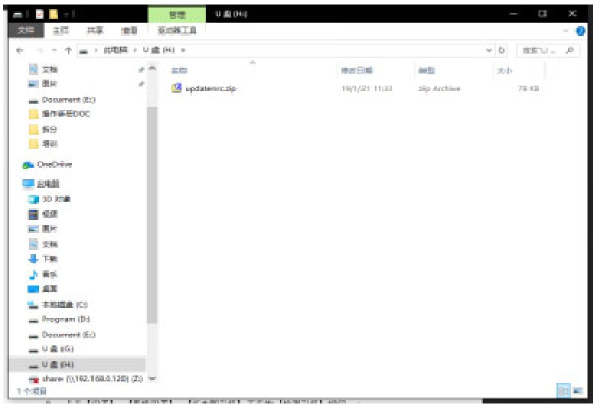


Version check and upgrade

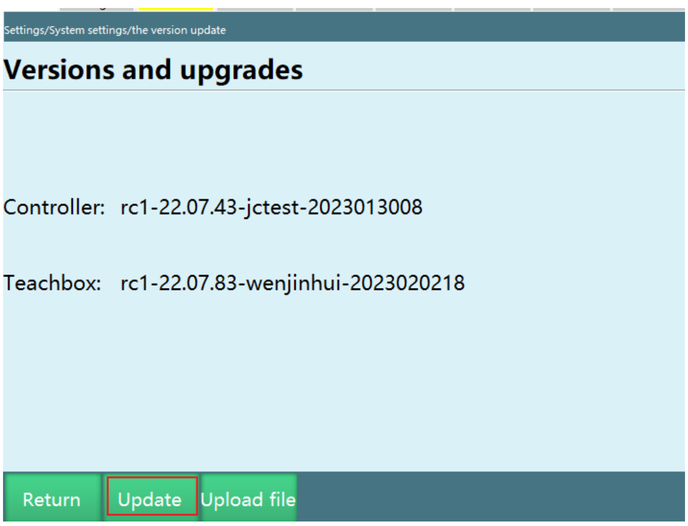
In the "Settings-System settings-Version upgrade" interface, you can check the software version of the teach pendant and the controller, and upgrade the teach pendant software.

The steps for upgrading the teach pendant software are as follows:

1. Put the upgrade file (.zip format, no unzip required, and no special characters such as brackets in the file name) into the root directory of the U disk, (the U disk must be in FAT32 format) and insert the U disk into the USB port of the teach pendant.



2. Click the [Detect upgrade] button under [Settings] - [System settings] - [Versions and upgrades].



3. Select the automatically detected upgrade file from the list.

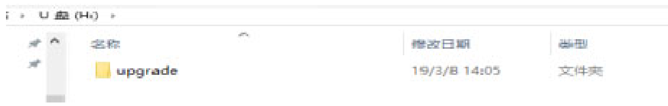
4. Click the [OK] button.

5. After successful upgrade, the teach pendant will restart automatically, and the upgrade will be successful after restarting.

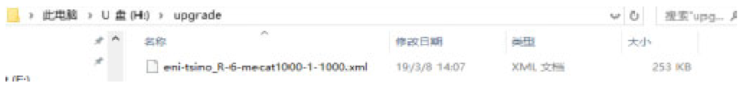
Upload file

To upload a file such as an ENI file to the controller, follow these steps:

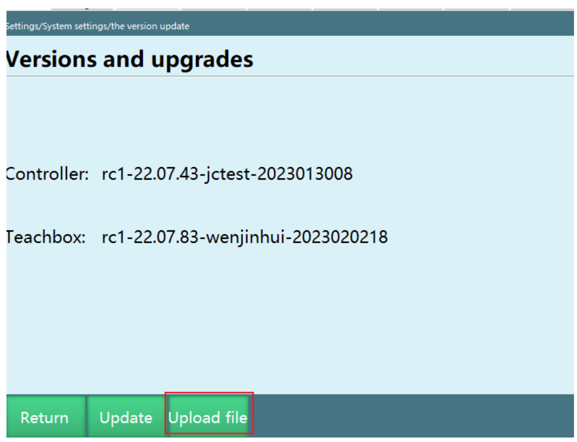
1. Prepare a computer and a U disk;
2. Create a new folder named "upgrade" in the U disk;



3. Put the files to be uploaded into the "upgrade" folder;



4. Insert the U disk into the USB port of the teach pendant;
5. Open "Settings - System settings - Version upgrade" interface;
6. Click the "Upload file" button;



7. Select the file you want to upload from the detected files pop-up and click the "OK" button.

System date and time setting

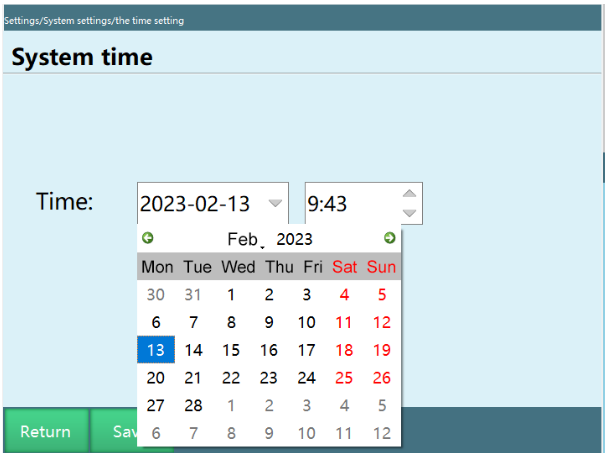
You can set the system date and time in the "System settings" interface.

The specific steps are as follows:

1. Open the "Settings - System settings - Time setting" interface.

2. Click the [Modify] button.

3. Select the year, month, day, hour and minute in the date and time setting.



4. Click the [Save] button.

IP setting

- You can modify the controller IP, teach pendant IP, and the IP to which the teach pendant is connected in the "Settings - System settings - IP setting" interface.
- Please do not modify the IP under unnecessary circumstances, so as not to cause malfunctions.
- If you modify the controller IP to a non-default value (192.168.1.13), please record the IP of the controller by yourself.
- The teach pendant connection IP is used for switching when one teach pendant is connected to multiple controllers at the same time.
- The "Reset network configuration" function is only available for T20.
- The specific steps to modify the current connection IP are as follows:

1. Click [System settings] - [IP setting].
2. Click the [Modify] button corresponding to "Connection IP".
3. Modify it to the required IP address and it will take effect immediately.
4. Modify the "Connection IP" to the IP filled in step 3.

The specific steps to modify the IP of the current controller are as follows:

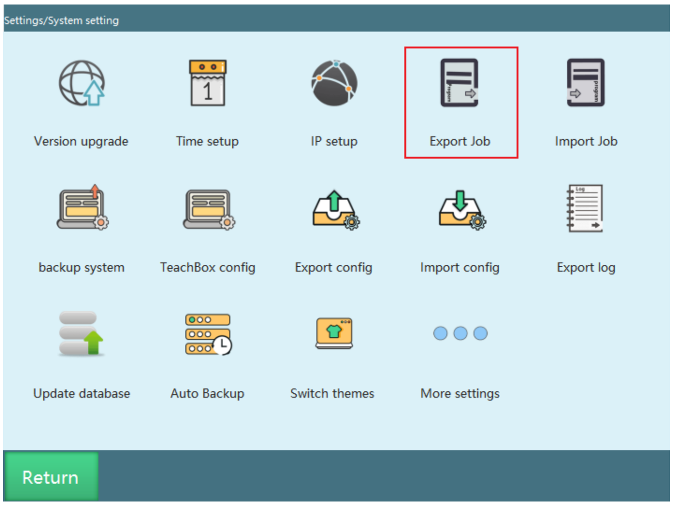
1. Click [System settings] - [IP setting].

2. Click the [Modify] button corresponding to "Modify controller IP".
3. Modify it to the required IP address and it will take effect immediately.
4. Modify the "Connection IP" to the IP filled in step 3.

The specific steps to modify the IP of the teach pendant are as follows:

1. Click [System settings] - [IP setting].
2. Click the [Modify] button corresponding to "Teach pendant IP".
3. Modify it to the required IP address, and restart the teach pendant to take effect.
4. Modify the "Connection IP" to the IP filled in step 3.

Export program



Click the [Export program] button in the "System settings" interface to export the program to a U disk.

The specific steps are as follows:

1. Insert the U disk (must be in FAT32 format) into the USB port of the teach pendant.
2. Click [Settings] - [System settings] - [Export program].
3. The exported programs are separated by date and type. Export the program to the "robotJobxx-xx-xx-xx (current date and time)" directory in the root directory of the U disk.

Import program

Click the [Import program] button in the "System settings" interface to import the program into the teach pendant.

The specific steps are as follows:

1. Create a new folder named "robotJobxxx (digital)" in the U disk, and create a new folder named "R1" in this folder;
2. Put the program with the suffix ".JBR" in the R1 folder;
3. Insert the U disk (must be in FAT32 format) into the USB port of the teach pendant;
4. Click [Settings] - [System settings] - [Import program];
5. **The system will pop up all the relevant directories in the U disk, select the directory of the program you need to import, then R1-R4 will be displayed according to the number of robots configured on the actual teach pendant, the corresponding robot job files in the U disk will be displayed in the white dialog box below, click [Select All] to select all robot job files, or you can customize the files you need and click [OK] to import the job files.**

One-click backup system

Click the [One-click backup system] button on the "System settings" interface to back up all related files such as job files, teach pendant program, controller program, robot configuration to a U disk at one time.

The specific steps are as follows:

1. Insert the U disk into the USB port of the teach pendant.
2. Click [Settings] - [System settings] - [One-click backup system].

Modify teach pendant configuration

Click the [Modify teach pendant configuration] button in the "System settings" interface to modify some function parameters saved on the teach pendant.

The specific steps are as follows:

Click the "Modify" button, modify the parameters, and click "Save"

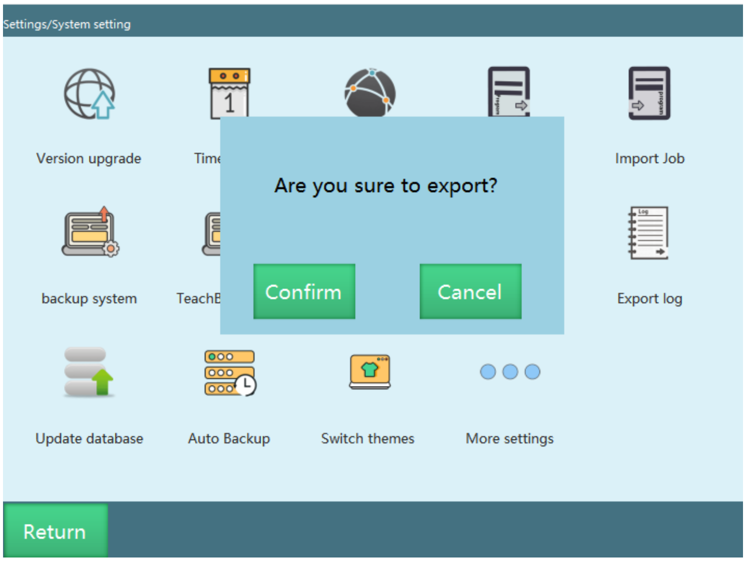
Export controller configuration

Click the [Export configuration parameters] button in the "System settings" interface to export the controller configuration parameters to a U disk.

The controller configuration parameters save configuration parameters such as robot, IO, external axis, and process parameters.

The specific steps are as follows:

1. Insert the U disk into the USB port of the teach pendant.
2. Click [Settings] - [Export configuration parameters] button.



3. Click the [OK] button.
4. Wait for the export operation to complete.

Import controller configuration

Click the [Import configuration parameters] button in the "System settings" interface to import the local configuration parameters into the teach pendant.

The specific steps are as follows:

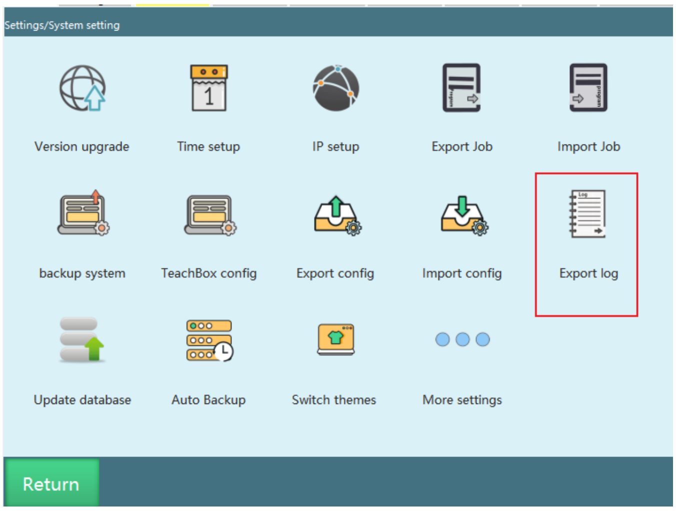
1. Insert the U disk into the USB port of the teach pendant.
2. Click [Settings] - [Import configuration parameters] button.
3. The system will pop up all the relevant directories in the U disk, select the directory of the program you need to import, and the configuration file of the corresponding robot in the U disk will be displayed in the white dialog box below. Click the configuration file of the desired robot and then click [OK]

to enter the detailed configuration parameters selection interface, you can select all or the required configuration parameters. Click the [OK] button.

4. Wait for the import operation to complete.

Export log

The export of the log involves the teach pendant log and the controller log;



Click the [Export log] button in the "System settings" interface/[Export] button in the "Log" interface to import the log, crash logs, robot parameter configurations and job files to the U disk. *The controller log is the most commonly used when we look for the cause of robot errors. The specific steps are as follows:

Insert a U disk in FAT32 format into the USB port of the teach pendant;

Enter the "Settings - System settings" interface/"Log" interface of the teach pendant;

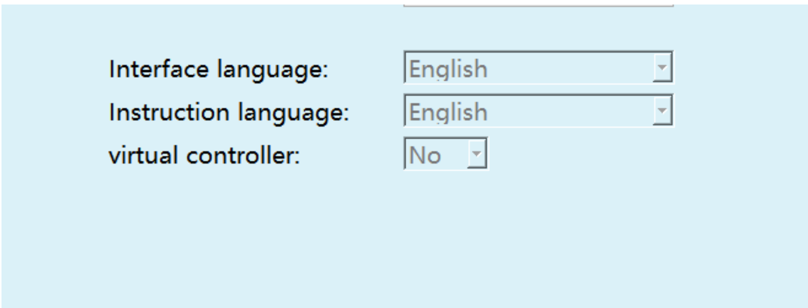
Click the [Export controller log] button in the "System settings" interface/the [Export] button in the "Log" interface, you can choose to export 5/30/100/500 logs;

After the export is complete, there will be four folders in the U disk. The controller logs, configuration, job files and the crash log (dumplog) generated by the program when it crashes are saved in the "controllerLogs-current date and time", "configFile-version information-current date and time", "robotJob-current date and time" and "dumplog-current date and time" directories of the U disk respectively; the logs of the teach pendant are saved in the teachbox.db file in the "controllerLogs-current date and time" directory.

Language change

The language of instructions and interface of this system can be switched between Chinese, English and Korean respectively. To switch the language, please follow the steps below:

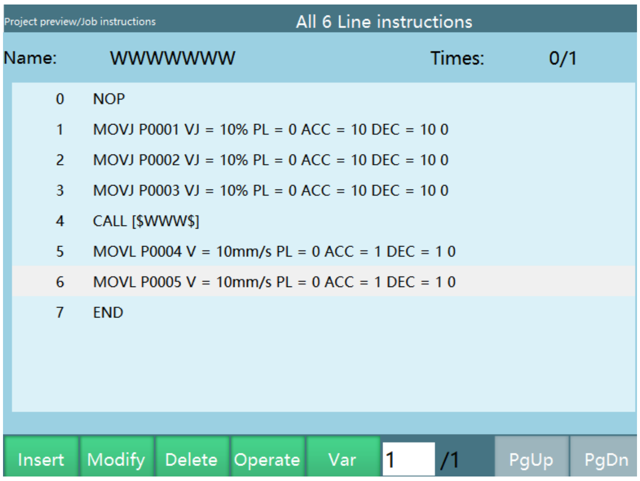
1. Enter the "Settings - System settings - Modify teach pendant configuration" interface;



2. Click the "Modify" button;
3. Select the required instruction language or interface language;
4. Click "Save". After saving, the instruction language takes effect immediately, and the interface language requires a reboot to take effect.

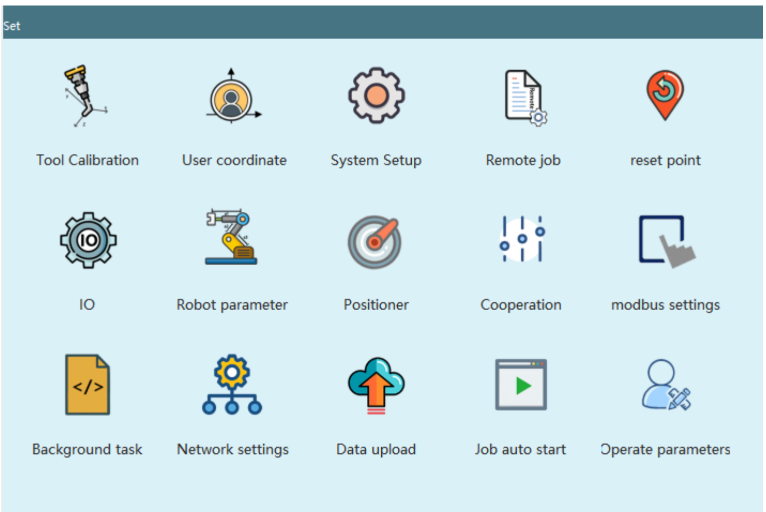
Chinese instruction

English instruction



Chinese interface

English interface



Korean interface

Database upgrade

It is used to upgrade the config.db file, which saves the configuration of the teach pendant, such as ip, some parameters in the operation parameters, etc.

Import/export ENI

When the robot eni file does not exist, it is necessary to import the eni file. When importing ENI, it is necessary to ensure that there is an eni file under the USB disk.

1. Create a new upgrade folder in the U disk, and upload the eni file to the upgrade folder.
2. Insert the U disk into the USB interface of the teach pendant.
3. Click [Settings] - [Robot parameters] - [Slave configuration] button.
4. Click [Import ENI] and select the ENI file to be imported in the USB disk directory to directly import it.
5. Wait for the import.

[Export ENI] is used to export the eni used by the current system. When you are ready to replace a new eni or replace the controller, you can use this function to export the eni in use and make a good backup. To import/export eni, please follow the steps below:

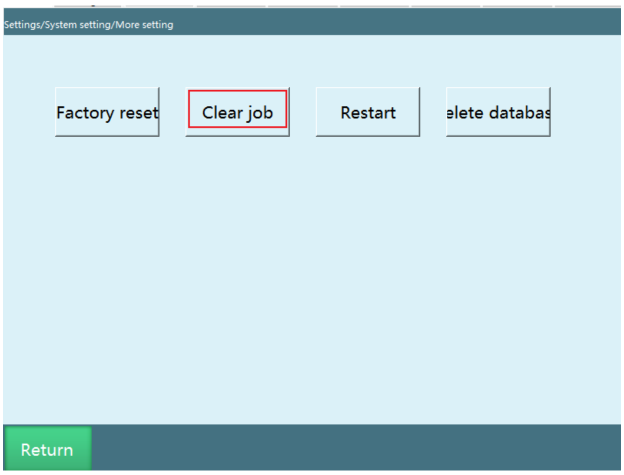
1. Insert the USB disk into the USB port of the teach pendant.
2. Click [Settings] - [Robot parameters] - [Slave configuration] button.
3. Click [Export ENI] to start the export work directly.
4. Wait for the export.

Clear program

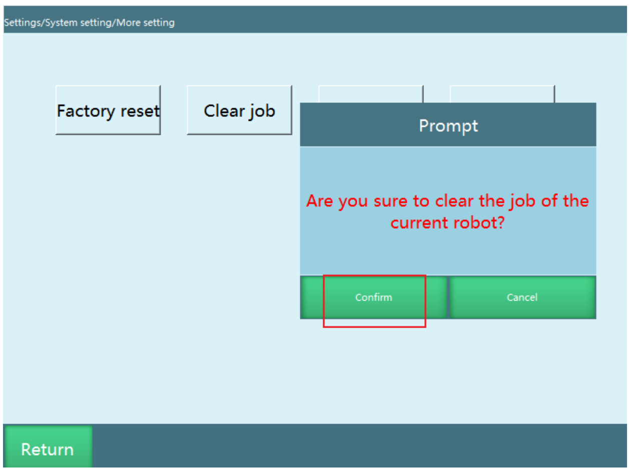
The "Clear program" function can clear all the programs in the system at one time, which is used when there are many useless programs.

The clear steps are as follows:

1. Enter the "Settings - System settings - More settings" interface;
2. Click the "Clear program" button;



3. Click the "OK" button in the pop-up dialog box.

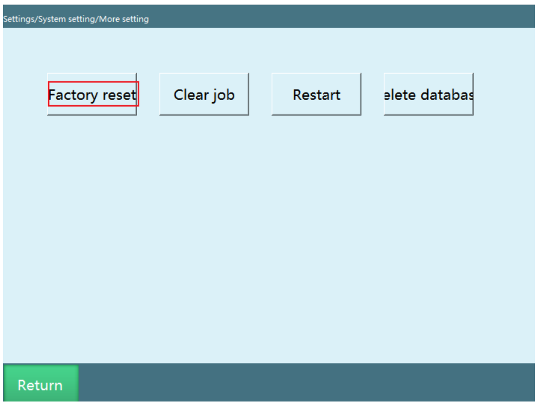


Restore factory settings

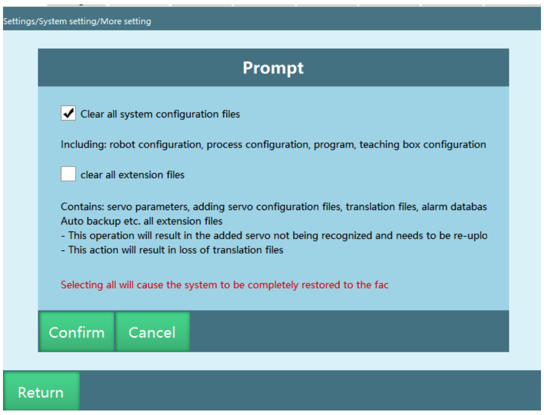
Restoring factory settings will clear all robot parameters and programs, so please be careful! Please be sure to back up all parameters and program files before performing this operation!

The steps are as follows:

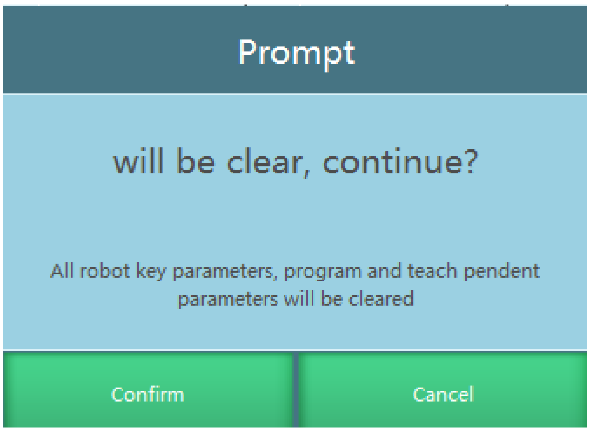
1. Enter the "Settings - System settings - More settings" interface;
2. Click the [Restore factory settings] button;



3. There are two options in the pop-up prompt window, which can clear the system configuration files and all extension files. You can choose the configuration file you want to clear.



4. Click the [OK] button and the selected configuration files will be restored to its factory settings.

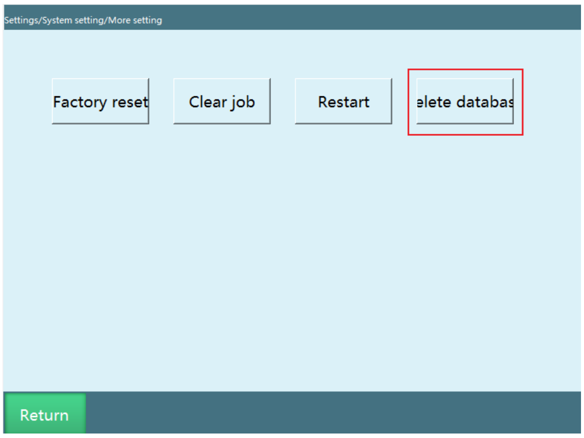


Delete database

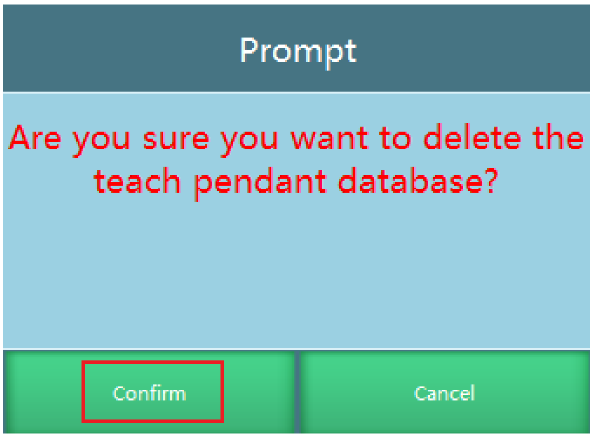
Function: After clicking, the database of the teach pendant will be deleted, which generally contains information on the user side, the created user and all passwords will be reset, the connection controller IP will be reset, the currently set voice will be reset, and the color theme of the teach pendant will be reset.

The steps are as follows:

1. Enter the "Settings - System settings - More settings" interface.
2. Click the [Delete database] button.



3. Click the [OK] button and the data will be reset.



Screen calibration

The screen calibration function is available for the T30 teach pendant.

The steps are as follows:

1. In the power-on state, press the left [O] + middle [Coordinate] + right [STOP] physical buttons at the same time, and the teach pendant will pop up a message, prompting "Calibration file has been deleted, restart the teach pendant to take effect", restart the teach pendant manually to enter the calibration interface.
2. Follow the example and click on the cross center of each of the 1-5 points with the stylus to complete the calibration.

Automatic backup

Controller automatic backup function

Backup content: program, parameter, software (nrc.out)

Number of backups: maximum 10, the newest replaces the oldest

Backup naming: by prerequisite, version, time

Example: If you modify parameters at 13:10 on September 10, 2020, then the backup name will be "parameter-20.04-3.3.7-202009101310"

Prerequisites for triggering backup: power on, parameter modification, program modification, upgrade

Backup frequency:

Back up once if the version and parameters are confirmed to be normal when starting up;

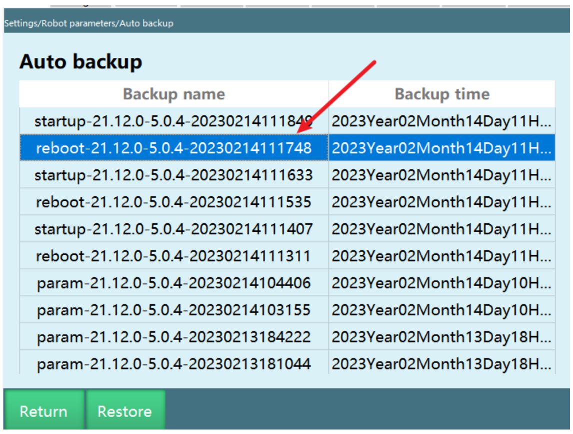
Back up once if the parameters are not modified again within 5 minutes after the parameters are modified;

Back up once if the program is not modified again within 5 minutes after the program is modified (insert instruction, modify instruction);

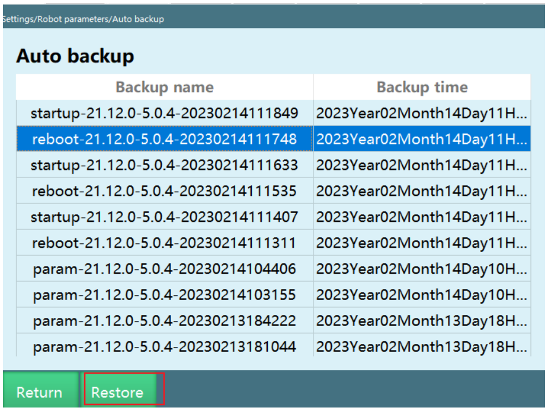
Back up once before upgrading

Backup restore method

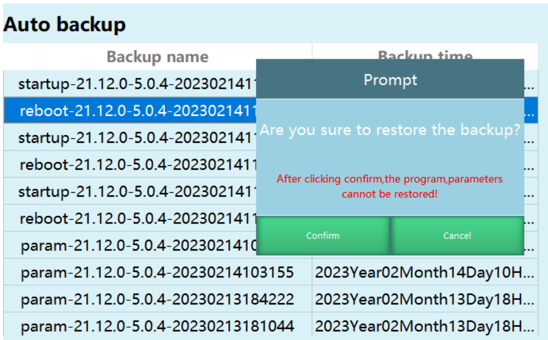
1. Select the backup you want to restore, the cursor will be displayed after selection



2. Click the "Restore backup" button.



3. A pop-up prompt will appear, click "OK".

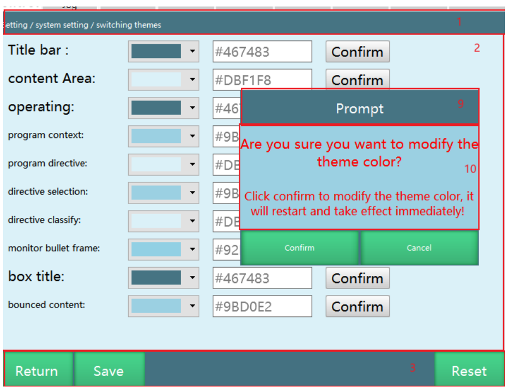


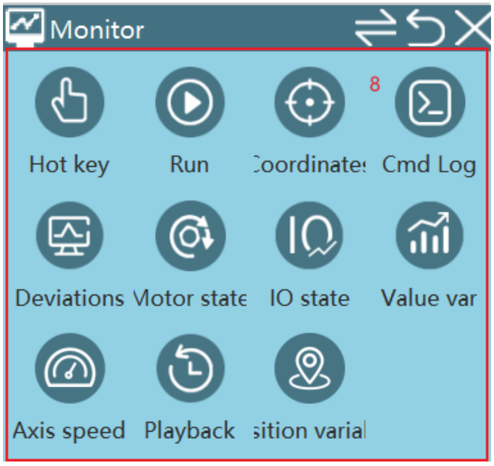
4. Please do not power off during the restoring process.

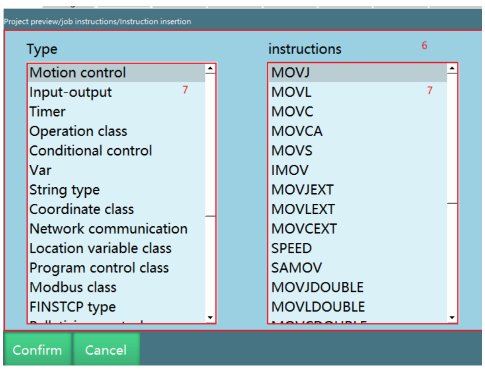
Switch topics

Customize the background color of different areas (suppporting RGB).

After setting, the teach pendant needs to be restarted to take effect. Click "OK " after modification, and the teach pendant will restart automatically. For the PC version, you need to manually restart the teach pendant.







1. Title bar
2. Content area
3. Operation area
4. Program background
5. Program instruction background
6. Instruction selection background
7. Instruction selection classification background

8. "Monitor" pop-up box
9. Pop-up box title
10. Pop-up box content

Operation parameters

This chapter will mainly introduce the usage and precautions of each parameter in the operation parameters.

Reservation mode

The reservation mode uses digital IO to control the running of programs. This mechanism is to set (reserve) in advance the programs to be started by IO and their running times and number them in the remote mode. After switching to the remote mode, sort the set programs by the IO signals. After pressing the "Run" button, the programs will be running according to the scheduled order and the running times. After all programs have been run, the operation stops. If you want to run again, you need to resort the programs.

If you need to make a single program run in an infinite loop, then set the running times of the program to 0 when making a reservation.

Enable description

When the "Reservation mode" is turned on, the operation process is: trigger remote IO program 1 signal → trigger start signal → robot running; if remote IO program 2 signal is triggered at this time, remote IO program 2 will be queued up, and remote IO program 2 will be executed after remote IO program 1 is executed.

When the "Reservation mode" is turned off, the operation process is: trigger remote IO program 1 signal → robot running; if remote IO program 2 signal is triggered at this time, it is invalid; you can run remote IO program 2 only after executing remote IO program 1.

Remote mode IO control is in reservation mode when turned on, and in non-reservation mode when turned off

The default is on

Setup steps

The steps of the reservation process are as follows:

1. Go to "Settings - Remote program settings";
2. Set 5 reserved programs and the running times;
3. Set the function of each IO input port in the "Settings-Remote program settings-Robot 1-Remote IO function", wherein program 1-program 5 correspond to the sorting function of the five programs in the "Remote program settings" interface;
4. Switch to remote mode;
5. Give the IO corresponding to the program serial number a high level for 2 seconds (set to active high) and release it, the program will enter the queue;
6. After the sorting is completed, if you want to cancel the sorting of a program, then give the IO corresponding to the program serial number a high level for 2 seconds again (set to active high) and release it;
7. Give the IO port corresponding to the program start signal a rising edge (set to active high) and the system starts running according to the running times of the programs in the queue;
8. You can also sort and cancel the queue during operation.

*If you turn on the "Run on reservation" switch, the first reserved program will start running as soon as it is reserved

*After the "Reservation mode" is turned off in the "Settings-Operation parameters", there is no reservation queue in the remote mode, and only one program can be run at the same time

Disable "Return to zero" button

Turn on to disable the "Return to zero" button

The default is off

Process selection

You can set general process, special process, palletizing process, welding process, cutting process

The default is general process

Disable wheel button

Turn on to disable the wheel button

The default is off

Switch to run mode for automatic power on

Turn on to switch to run mode for automatic power on

The default is off

Attitude value

Radian measure, degree measure

The default is radian measure

Remote IO breakpoint execution

Turn on to use breakpoint execution, turn off to not use it

The default is on

Remote IO current line execution

Turn on to use current line execution, turn off to not use it

The default is off

Switch back to user rights after running

Switch to operator privileges at the set time during running.

The default value of the parameter is 0, which means no switching.

Joint actual direction

After turning on, the robot and external axis joint parameter setting interfaces will display the joint actual direction parameter.

The default is off.

Switch to remote mode without teach pendant

After turning on, triggering the remote IO control signal will automatically switch to remote mode when the teach pendant is not connected

The default is on.

Reserve again while the remote IO program is running

After turning on, the reserved program can be reserved again during running;

After turning off, the reserved program cannot be reserved again, only the program in "reserved" and "unreserved" status can be reserved

The default is on.

Step/return to zero/reset point operation mode

Click to run: press the corresponding button, the robot will perform the corresponding function

Press to run: keep pressing the corresponding button, the robot will execute the corresponding function

Run mode startup default speed

After turning on, this refers to the initial speed set by the system every time you turn on the system and switch to the run mode

Synchronize operation modes when connecting controller

Function: Synchronize the operation modes when the controller and the teach pendant are connected for the first time

Follow controller: When the controller and the teach pendant are connected normally, the teach pendant will follow the mode sent by the controller

Follow mode knob: When the controller and the teach pendant are connected normally, switch the operation modes through the knob

Special case: When the program is running and the teach pendant is reconnected to the controller normally, you can also use the knob directly to switch the modes

Note:

1. After the teach pendant and the controller are reconnected normally, there will be a pop-up prompt: the robot is running, please press the "OK" button to confirm the synchronization. After clicking the "OK" button, the teach pendant will synchronize the knob operation mode
2. Before the pop-up window disappears: **only the OK button, stop button and knob are available, the rest are inoperable**

The default is "Follow controller".

Safety light curtain teach mode invalid

Function: After it is turned on, the safety light curtain limit is shielded in the teach mode, and the alarm can be used normally (default is off)

The default is off

Disable start button in run mode

Function: When turned on, the teach pendant is switched to the run mode, and the run status and physical start buttons cannot be clicked, including the START button of the PC version (default is off)

Note: The START button will also be disabled on the PC version

The default is off

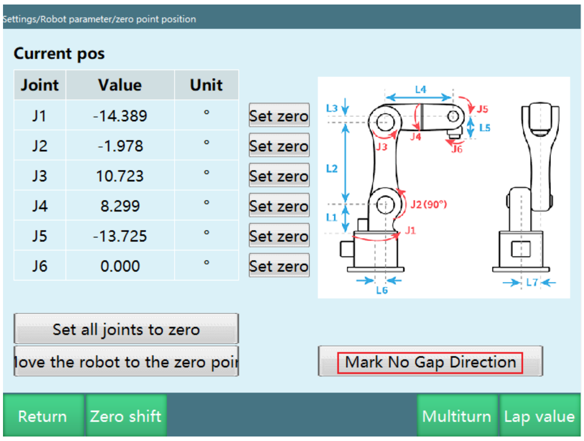
NP parameters

Function: Display [Settings] - [Robot parameters] - [NP parameters] when turned on; hide when turned off

The default is off

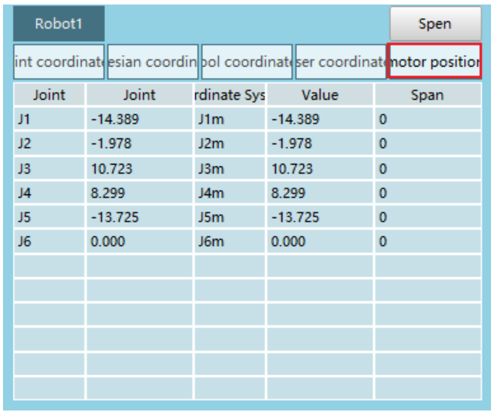
Display motor coordinate position and calibration button

Function: Display [Settings] - [Robot parameters] - [Zero position] - [Mark no gap direction]



and [Monitor] - [Robot coordinates] - [Motor position] when turned on; hide when turned off

The default is off



Robot log

The system log is divided into teach pendant log and controller log. The teach pendant log mainly stores the log generated by the operations on the teach pendant. The controller log saves all logs of robot operations, parameter modifications, etc.

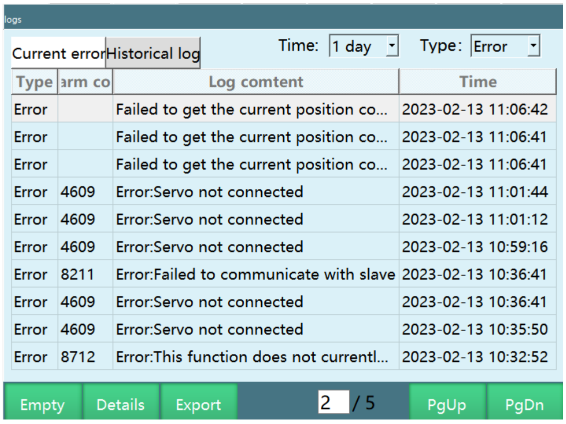
Note: If you want to export logs, please refer to "System settings - Export log".

Teach pendant log view

You can view operation and error logs in the "Log" interface.

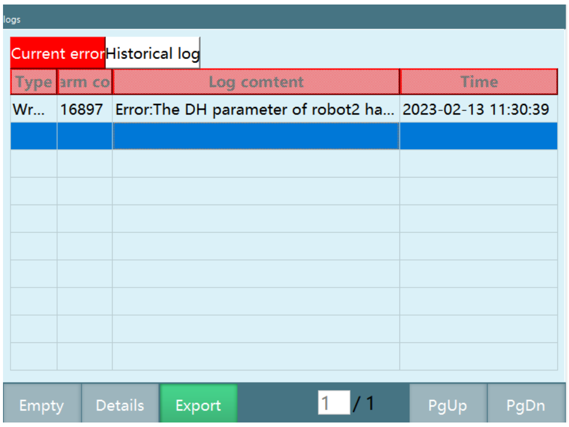
The specific viewing steps are as follows:

1. Click [Log] to open the log view interface. If there is no error message, you will directly enter the "History log", otherwise you will see logs of error type by default;



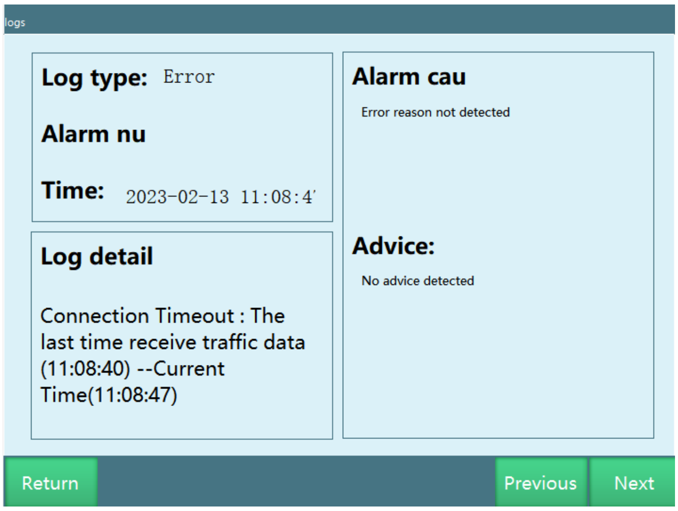
Note: The logs of "Error" type are displayed first when you enter the "Log" interface

If there is currently an error message, click [Log] to enter the "Current error" interface

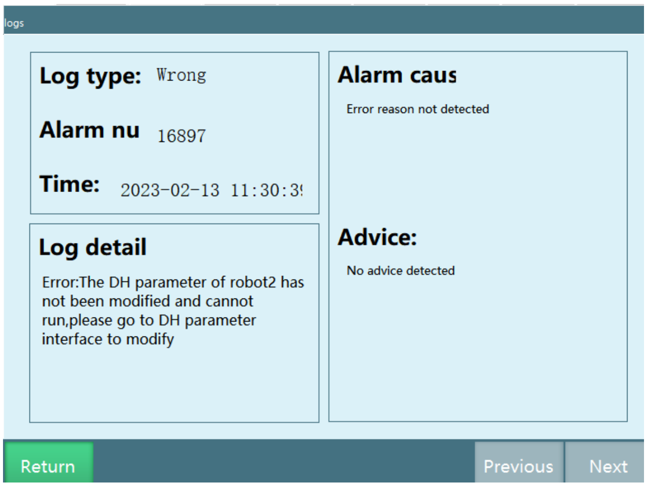


2. Click the "Type" tab above to switch the type of logs to be viewed.
3. After selecting a log, click the [Details] button below to view the log details;

History log details interface:



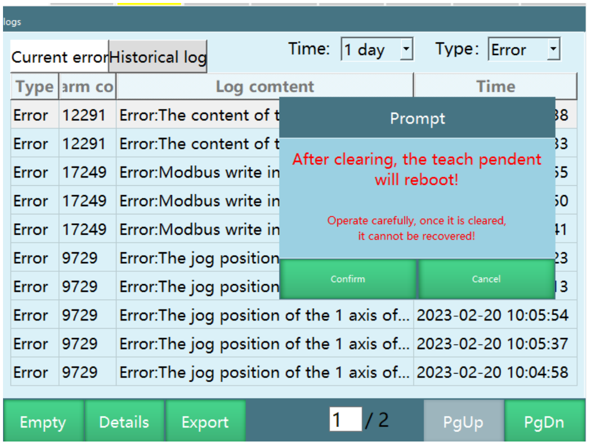
Current error details interface:



Note: The fault cause and handling suggestions need to be uploaded to the corresponding database

Clear log:

Click the "Clear" button, then there will be a prompt box prompting that the teach pendant will be restarted after clearing and the logs cannot be restored after clearing



Log type

Log type includes "All", "Message", "Operation", "Warning", "Error". We mainly view "operation log" and "error log"

Operation log: This type of log saves the user's basic operations, such as creating a new program, renaming a program, inserting instructions, etc.

Error log: This type of log saves all system error and servo error information, including error code, error time, error type, error content, solutions and other information.

Troubleshooting

Encoder battery undervoltage error

Note: The following operations will cause the zero point to be lost, you must reset the robot points

Steps:

Disconnect the power supply of the controller, control cabinet and other equipment of the undervoltage robot, replace robot batteries by professionals while ensuring safety, after the replacement, reconnect the power supply and start the controller system

After the teach pendant and the controller are started normally, there is still a pop-up box reporting errors

Click the "Clear error" button on the teach pendant, and the "OK" button will appear in the pop-up box

Click the "OK" button to enter the zero calibration interface

After re-calibrating the zero point, it returns to normal

Check all the points that can be used by the program, make sure that the positions of the points are normal, and ensure that the points set in the process are normal

IO, Modbus& Remote Function User Manual

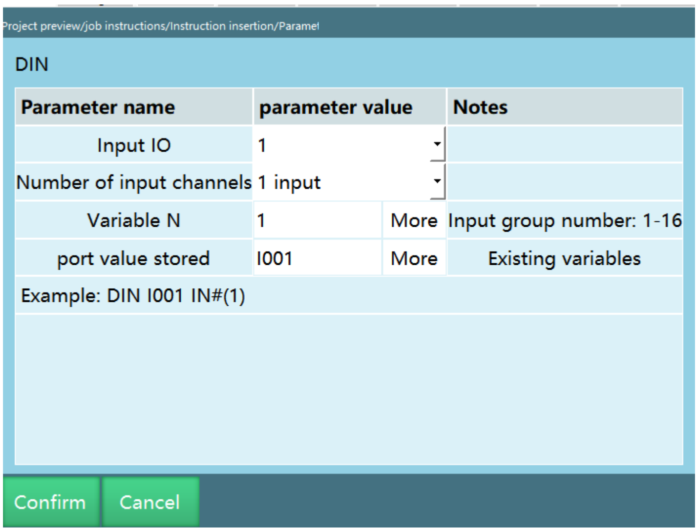
Catalogue

IO, Modbus and Remote Program

IO

Input/Output instructions

DIN-IO input



This instruction is used to read the digital input status into a variable that can be local/global integer (INT, GINT) variable or local/global Boolean (B, GB) variable.

Input IO board: You can select from the IO boards 1-4.

Number of input channels: IN#-1 channel input, at this time, 1 channel is 1 group, and groups 1-16 correspond to ports 1-16 respectively;

IGH#-4 channel input, at this time, every 4 channels is a group, that is, 1-4 channel ports, 5-8 channel ports, 9-12 channel ports, and 13-16 channel ports are 1-4 groups respectively. The group number can be filled with 1-4. If you want to read the input status of 5-8 channel ports at the same time, you can fill in the group number 2.

IG#-8 channel input, at this time, every 8 channels is a group, that is, 1-8 is the 1st group, and 9-16 is the 2nd group. If you want to read the input status of ports 9-16 at the same time, fill in the group number 2.

If multiple channel ports are read at the same time, the port status will be converted to decimal and saved into the variables. And the group number being read can be obtained from the corresponding variable.

For example, read 5-8 channel ports while there are 4 channels, their status are as follows, and will be stored in I001

1	2	3	4
0	1	1	0

Then the binary value is 0110, which is converted to 6 in decimal.

Then it is saved in the system as DIN I001 IGH#(1) 6

For example, read 9-16 channel ports while there are 8 channels, their status are as follows, and will be stored in GI001

16	15	14	13	12	11	10	9
0	1	1	0	1	0	0	1

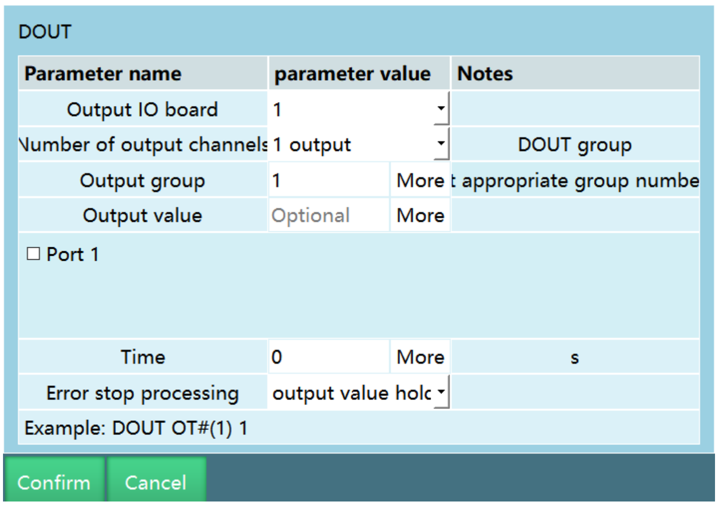
Then the binary value is 01101001, which is converted to 105 in decimal.

Then it is saved in the system as DIN GI001 IG#(2) 105

Input group number: It can be set to read the 1/4/8 channel input status at the same time, or set through the variable values of the binding variable.

Port value storage: Store the value read from the IO input into the selected variable.

DOUT-IO output



This instruction is used to output digital signals through the digital IO board.

Output IO board: Select the IO board you need to output, you can choose from 1-4.

Number of output channels: OT#-1 channel output, at this time, 1 channel is a group, and groups 1-16 correspond to ports 1-16 respectively.

OGH#-4 channel output, at this time, every 4 channels is a group, that is, 1-4 channel ports, 5-8 channel ports, 9-12 channel ports, and 13-16 channel ports are 1-4 groups respectively. The group number can be filled with 1-4. If you want to output the 5-8 channel ports at the same time, you can fill in the group number 2.

OG#-8 channel output, at this time, every 8 channels is a group, that is, 1-8 is the 1st group, and 9-16 is the 2nd group. If you want to output the ports 9-16 at the same time, fill in the group number 2.

Output group number: It can be set to output 1/4/8 channel IO at the same time or set through the variable values of the binding variable.

Output value: You can select "Optional" or choose to output through variables, or set through the variable values of the binding variable.

If "Optional" is selected, then check the status of each port in each group of IOs, and the output is 1 if checked and 0 if unchecked.

If you choose to output through variable, the variable value will be converted from decimal to binary at the time of output, as shown in DIN.

Time: Wait for the specified time after the instruction is executed, and then invert the output.

Error stop processing: In the process of IO signal output, an error alarm is generated. The IO signal will make different processing methods.

If you select "Output value hold", then while the program is running, the port output will remain as is and the timing time will be suspended when an alarm is triggered or in case of other unexpected situations. When the alarm error is cleared and the program begins to run normally, the IO output timing continues for the remaining time before the pause, and the port will be inverted when the timing is over.

If you select "Time-out stop", no matter what the situation is, as long as the port timing is over, the port value will be inverted, and it will not be affected by the pause or error.

AIN-Analog input

This instruction is used to read a single port input value from an analog IO board into a variable.

Analog input port: Select the input port to be read.

Variable name: Please select the variable name of the variable you need to read into, such as GD001.

AOUT-Analog output

This instruction is used to set the output value of a single port of the analog IO board. The output value can be a floating point number.

Analog output port: Select the output port whose value needs to be set.

Variable value source: Please select global floating point GDOUBLE or local floating point DOUBLE variable or hand-filled value

PULSEOUT-Pulse output

This instruction is used to control the pulse output of the IO board that supports PWM.

Number: Total number of pulses output.

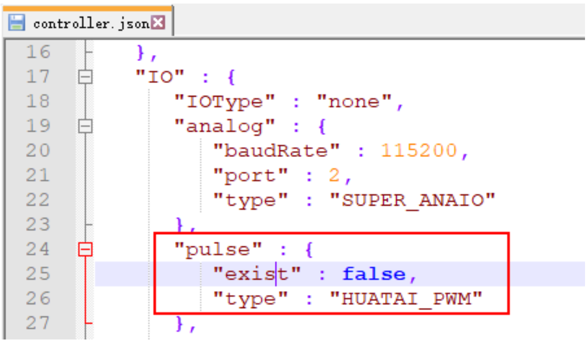
Frequency: Pulse output frequency; for example, the default value is 100, then 100 pulses are output in 1s

The IO boards that support this function are as follows:

HUATAI IOPWM

INEXBOT R1PWM

Usage:


controller.json

```
    },
    "IO" : {
        "IOType" : "none",
        "analog" : {
            "baudRate" : 115200,
            "port" : 2,
            "type" : "SUPER_ANAIO"
        },
        "pulse" : {
            "exist" : false,
            "type" : "HUATAI_PWM"
        },
```

Modify profile controller.json;

Find the "exist" parameter in "IO" - "pulse" and change it to "turn";

turn: function on;

false: function off;

Find the "type" parameter in "IO" - "pulse" and change it to the corresponding IO board;

HUATAI_PWM: HUATAI IO

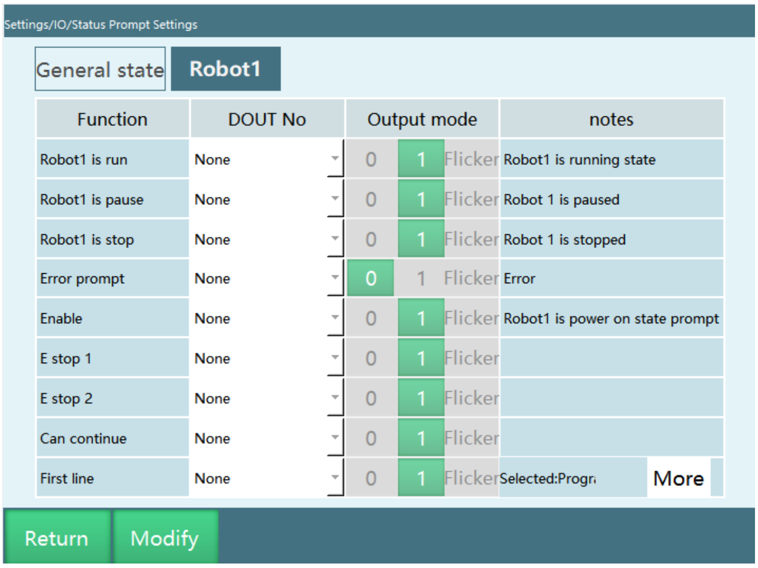
INEXBOT_PWM: INEXBOT R1

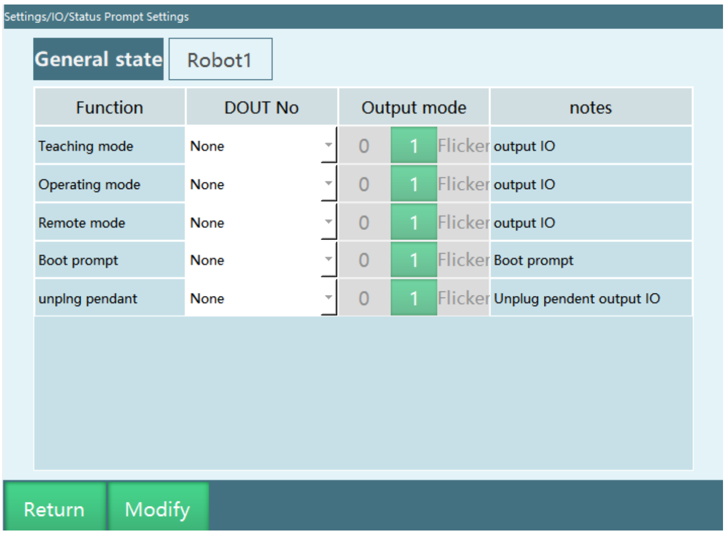
READ_DOUT-Read output

This instruction is used to read the output status of the current digital IO board into a variable. It is used in the same way as DIN, except that the reading is the status of the output.

IO status prompt settings

In the "Status prompt settings" interface, you can set the I/O port corresponding to the "Boot prompt", robot running status, "Error prompt", "Enable", mode status, emergency stop, and other functions and the level corresponding to that port.





Robot1 run: The corresponding DOUT port outputs a high level when Robot1 is running

Robot1 pause: The corresponding DOUT port outputs a high level when Robot1 is paused

Robot1 stop: The corresponding DOUT port outputs a high level when Robot1 stops

Error prompt: When the robot servo reports an error, the corresponding DOUT port outputs the corresponding signal which can be set to "Bright" or "Flashing"

Enable: Output high level when the robot is powered on

E-stop 1: Output high or low level after "E-stop" signal is triggered, this can be set by yourself

E-stop 2: Output high or low level after "E-stop" signal is triggered, this can be set by yourself

Main program first line: Output a signal with a high level parameter of 1 and the program cursor jumps to the first line of the main program

Continuable: Output a signal with a high level parameter of 1, you can run a paused program

Boot prompt: Controller power-on output status, output high level when power on

Teach mode: Output high level when in teach mode

Run mode: Output high level when in run mode

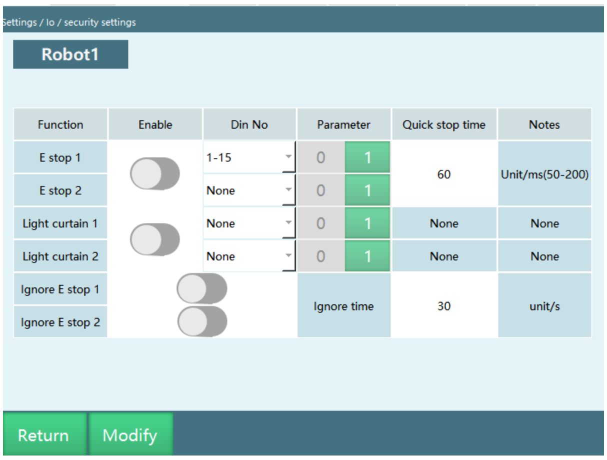
Remote mode: Output high level when in remote mode

Unplug teach pendant: Output high or low level after unplugging the teach pendant, this can be set by yourself

IO safety settings

In the "Security settings" interface, you can set the I/O port corresponding to the emergency stop, safety light curtain and other functions and the level corresponding to that port.

After the IO E-stop has been lifted, you need to click the "Clear error" button to clear the error before you can perform other operations.



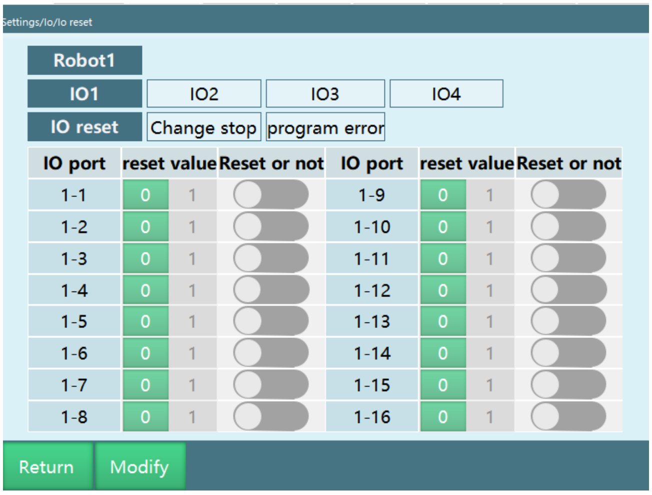
E-stop: The robot is powered off and switched to servo stop status after the emergency stop signal is triggered

Safety light curtain: The robot pauses after the safety light curtain is triggered, you can press the start button again to resume operation

Block E-stop: After turning on, the emergency stop signal is blocked during the blocking time

IO reset

IO reset function can restore the output port of IO to the initial status when the program is stopped or error is reported. IO reset is divided into three types: remote IO reset, switch mode stop, and program error stop.



Remote IO reset: In remote mode, when a reset signal is given, the robot will execute the reset program to return to the reset point, which will reset the IO port set in this interface to the reset value. If the reset program is stopped in the middle, the IO reset operation will not be performed.

Switch mode stop: When running a program, switching the mode to teach or remote mode will cause the program to stop, which will reset the IO port set in this interface to the reset value.

Program error stop: Program error causes the program to stop, which will reset the IO port set in this interface to the reset value. Specific types of errors: servo error, error of IO setting, error in system operation

Usage steps:

Enter the "IO reset" interface;

Select the robot;

Click to enter the reset scenario (IO reset, switch mode stop, program error stop);

Select the IO board;

Turn on the "Reset" switch corresponding to the IO port to be reset;

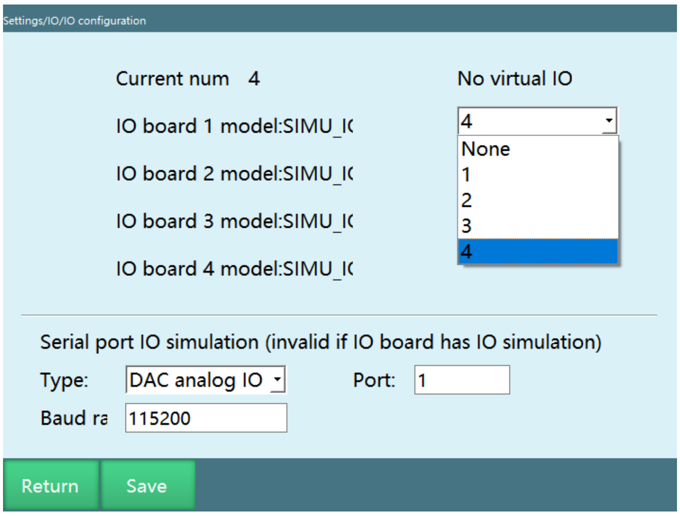
Select the reset value (0/1), 0 is low level and 1 is high level.

IO configuration

The system will automatically identify the IO model according to the hardware connection order, no need to set; it can be used to view the number and model of IO boards.

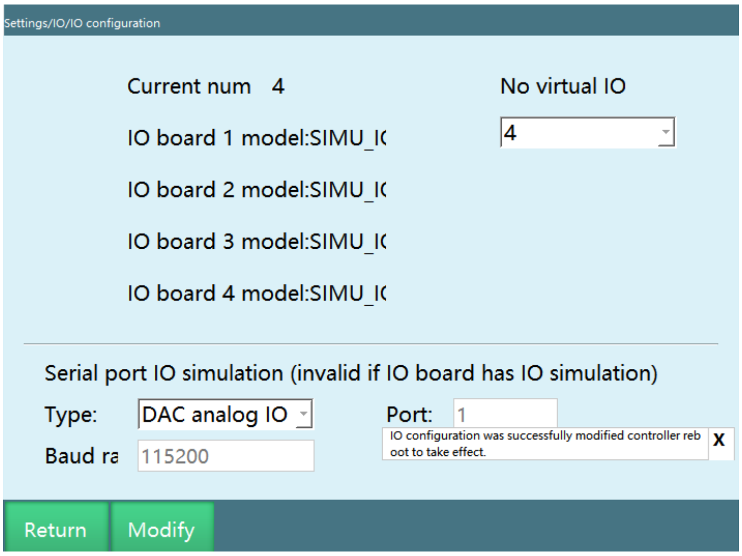
Enter the [Settings] - [IO] - [IO configuration].

The input box is grayed out and no value can be entered.



After clicking "Modify", the "Modify" button becomes "Save", then select the desired virtual IO from the "Number of virtual IO" drop-down box

Note: Virtual IO is only for program debugging and program demonstration, and does not have any IO signal access

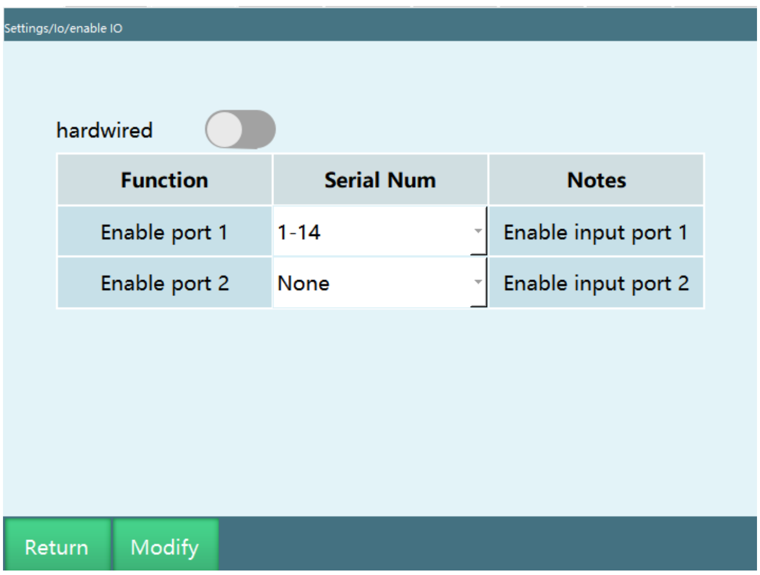


Click "Save", restart to take effect, and the modification is successful.

Enable IO

If you use the enable hardwired teach pendant, you need to select the corresponding DIN port and turn on the enable switch in this interface after connecting the cable. The power-on enable function is controlled by the IO board input signal; for the non-enable hardwired teach pendant, no need to set.

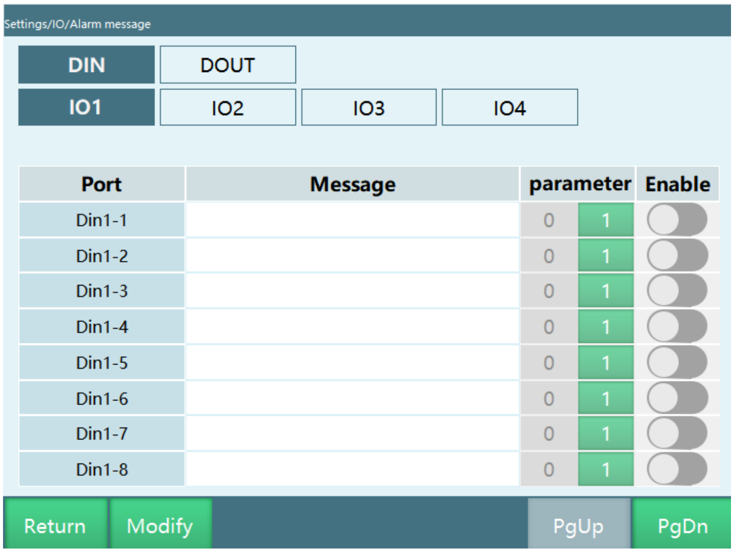
When this function is turned on, the teach pendant enable button is disabled and is not available for use.



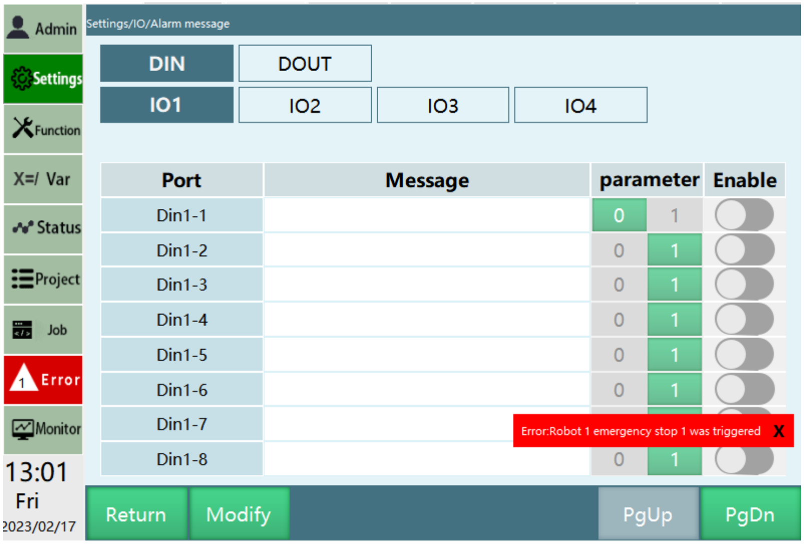
Enable port 1 is enabled for power-on, and enable port 2 is enabled for power-off. To power on, you only need to turn on the enable port 1. In any case, as long as the enable port 2 is turned on, it will be powered off.

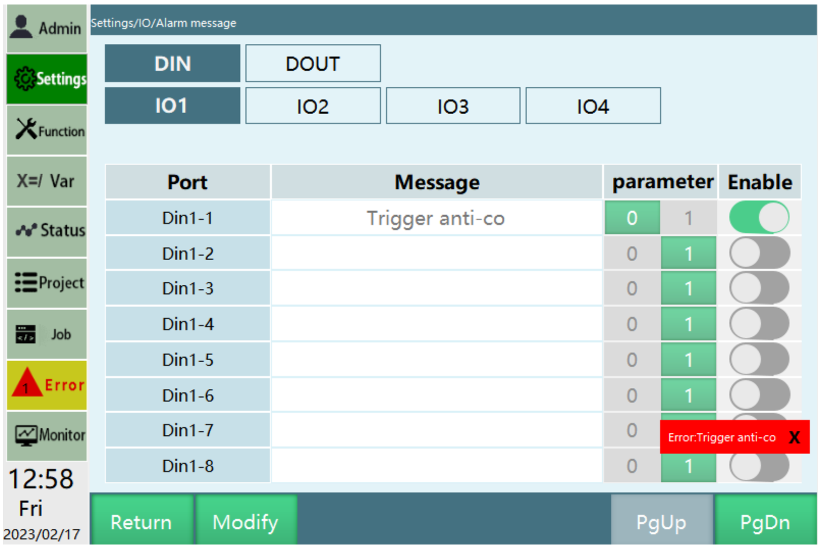
Alarm message

This function allows you to customize the alarm content of IO input and output ports, and the priority of alarm message is higher than that of other types of IO alarm messages.



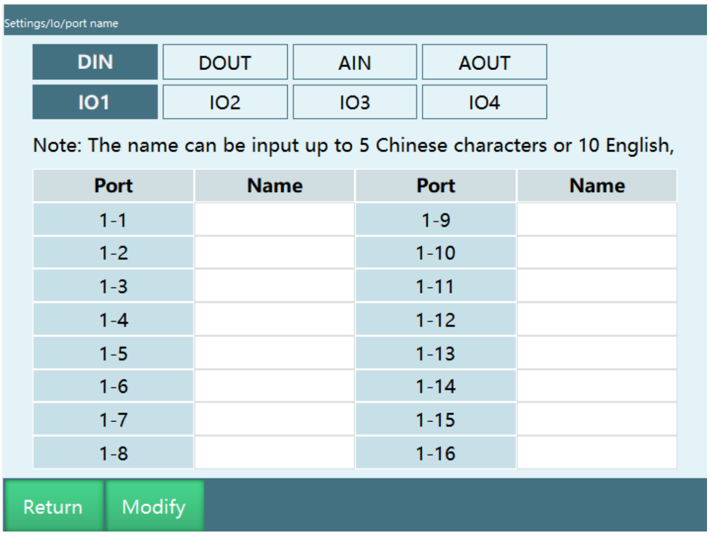
For example: set the IO emergency stop signal port to 15 to connect to the anti-collision IO, 1 to trigger and 0 to release; if DIN1 is triggered, it will report "Robot 1 IO emergency stop is triggered"; at this time, find DIN1 on the alarm message interface, and enter " Trigger anti-collision" in the message field, then triggering DIN15 again will report an error "Trigger anti-collision" instead of "Robot 1 IO emergency stop is triggered".



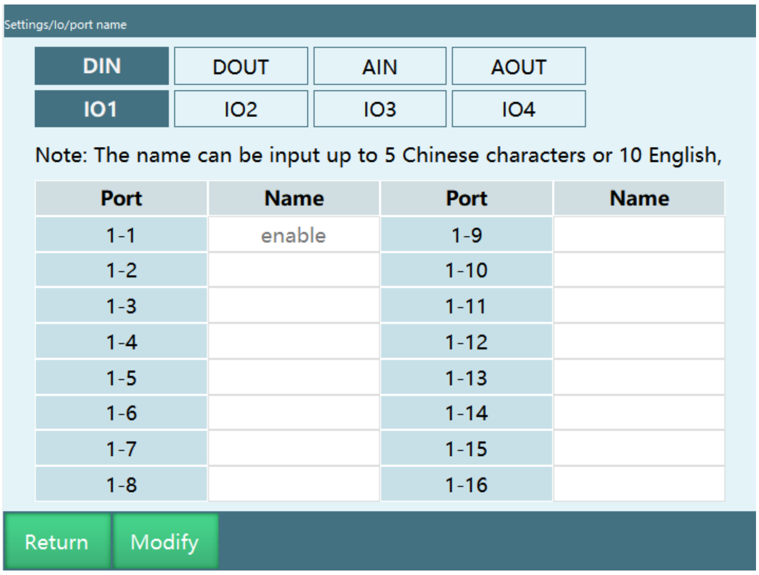


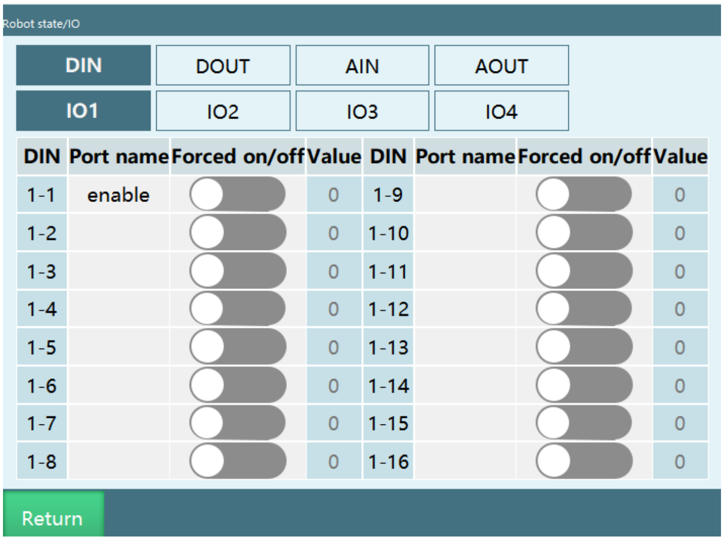
Port name

The port name supports a maximum of 5 Chinese characters or 10 English characters. After the setting is successful, the name will be automatically displayed when using the IO port related drop-down box option.



If the name of DIN1-1 is set as "enable ", the DIN1 name "enable " will be displayed in the [IO] of [Status]





Brief description of remote mode IO reservation

Signal description

	Function	Support mode	Trigger/output method	Description
Digital IO input	Start	remote	rising edge	When the parameter is 1, the signal is valid when it changes from 0 to 1
	Stop	remote	continuously valid	When the parameter is 1, the signal is continuously valid
	Pause	remote	continuously valid	When the parameter is 1, the signal is continuously valid
	Clear error	remote	rising edge	When the parameter is 1, the signal is valid when it

				changes from 0 to 1
	Start as reservation	remote	no	When it is turned on, the robot will be powered on when the reservation is successful
	I/O program 1-10	remote	pulse (period 0.6s)	When the parameter is 1, the signal is valid at 0-1-0, and the program needs to be triggered for at least 0.6 seconds for successful reservation.
	E-stop 1	teach, run, remote	high level	Scan once in 1 ms, trigger when scanned
	E-stop 2	teach, run, remote	high level	
	Safety light curtain 1	run (running), remote (running)	high level	
	Safety light curtain 2	run (running), remote (running)	high level	
	Block E-stop	used in conjunction	The emergency stop function is blocked when the button is turned on,	

	1	with E-stop	and the emergency stop signal is re-detected after the set time is up	
	Block E-stop 2	used in conjunction with E-stop		
Digital IO output	Boot prompt	no mode limit	bright, flashing, output only at power on	Output high level
	Robot1 run	teach, run, remote	bright, flashing	Output high level when the program is running
	Robot1 pause	teach, run, remote	bright, flashing	Output high level when the program is paused
	Robot1 stop	teach, run, remote	bright, flashing	Output high level when the program is stopped
	Error prompt	no mode limit	bright, flashing	If bright, then output high level; if flashing, then output pulse (period 1s, 0.5s on, 0.5s off)
	Enable	no mode limit	bright, flashing	Output high level
	IO program 1-10 reservation output	remote	bright, flashing	Not bright when already reserved/not reserved; Flashing during reservation,

				period 1.2s, 0.6s on, 0.6s off; Bright when in operation, output high level
	E-stop 1	when signal is triggered	high level, low level, flashing	When the parameter is 1, output high level
	E-stop 2	when signal is triggered		
	Unplug teach pendant	no mode limit	high level, low level, flashing	Click "Unplug teach pendant", output 1 or 0
	Continuable	when signal is triggered	high level, low level, flashing	Output a signal with a high level parameter of 1, you can run the paused program
	Main program first line	teach, run, remote	high level, low level, flashing	Output a signal with a high level parameter of 1 and the program cursor jumps to the first line of the main program

Note: In this description, output 1 means output high level

Description of remote mode status

Not reserved: After entering remote mode, if no reservation has been made for the program, or the reservation was made and then cancelled, then "Not reserved" will be displayed.

Reserving: If the reservation is successful, "Reserving" will be displayed.

Running: If the program is running, "Running" will be displayed.

Reserved: If the program finishes running or is triggered to stop, "Reserved" will be displayed.

The speed cannot be modified in the remote mode, and the speed needs to be modified in advance in [Settings-Remote program setting]

Program reservation

Trigger the IO port corresponding to the program to successfully reserve the program. To cancel the reservation, you need to trigger the IO port corresponding to the program again.

Start: Directly trigger the IO port corresponding to the trigger

Start as reservation: signal 0-1 (press the button), 0.6 seconds or more time later, signal 1-0 (release the button), the program runs directly; when "Start as reservation" is selected, the start signal may not be set.

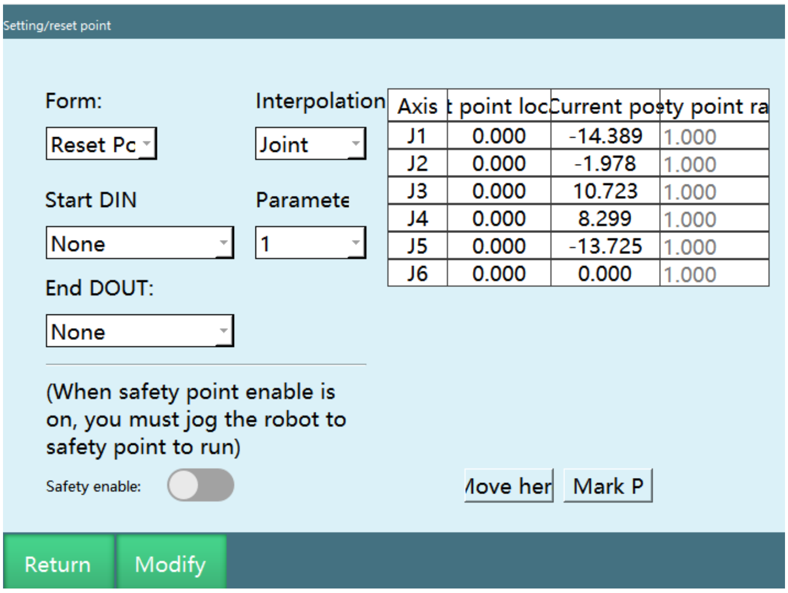
You can reserve again after the reserved program runs

Troubleshooting

After the IO function is successfully set, please go to "Status"-"IO function status" to check whether the setting is successful or whether there are conflicting functions.

Reset point setting

The reset point function supports movement to a safe point by means of joint and linear interpolation, and you can also use the reset program instruction to customize the reset trajectory and position.



Form: reset point, reset program;

Interpolation method: joint, linear; the movement speed is 10% of global speed when joint interpolation is selected, and 100mm/s when linear interpolation is selected; the running speed is equal to instruction speed x speed in status bar when reset program is selected.

Safety enable: When turned on, program will run to determine whether the robot is in the reset point (safety point) position, and it must be in the reset point position to continue running the program.

Start DIN: Reset point trigger signal;

Parameter: Reset point trigger signal 0 valid or 1 valid.

End DOUT: Status signal output after returning to the reset point;

Safety point range: The safe range error of each axis, if within the range, the robot will be judged to be at the reset point (safety point);

Mark this point: Set the current robot coordinate as the reset point, and click "OK" to set successfully;

Move here: Move to the reset point by joint interpolation;

Description of remote mode control right

When there are teach pendant, touch screen and I/O control device in the control system, the priority of the control right is teach pendant>touch screen>I/O control device.

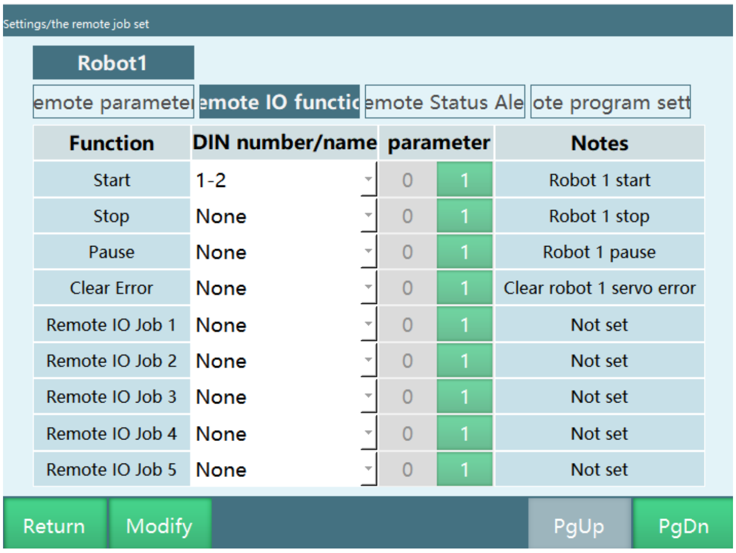
After switching to the remote mode, the control right is switched to the touch screen. If there is no touch screen, switch to I/O control device. At this time, the interface of the teach pendant only displays the connection status of the Modbus module and the I/O module and the I/O program.

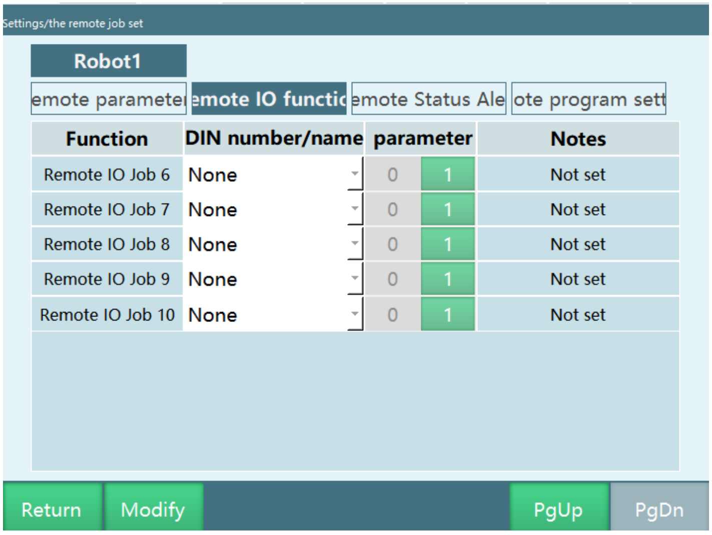
When there are touch screen and I/O module at the same time, set the I/O module enable in the touch screen.

Remote IO control

Remote I/O function selection setting

In "Remote program setting-Remote IO function", you can set the I/O port corresponding to the remote IO control (start, stop, pause, emergency stop, clear alarm, etc.) and the level corresponding to that port, and you can also set the program run by the I/O module remote control.



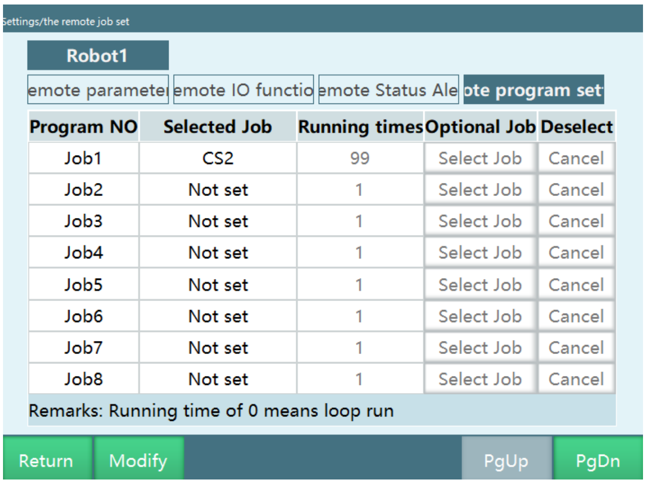


The program of the set I/O module can only be selected from the program set in the "Remote program setting" interface.

There can be up to 10 remote reservation programs

Start as reservation: After it is turned on, the robot will be powered on and run the first reserved program immediately after the reservation is successful, and other programs can be reserved at this time.

Remote program setting



The programs used by the touch screen and the I/O control module can be set in the "Remote program setting" interface.

If there are multiple robots, you can select the robot to be set at the robot section, and set each program of the robot.

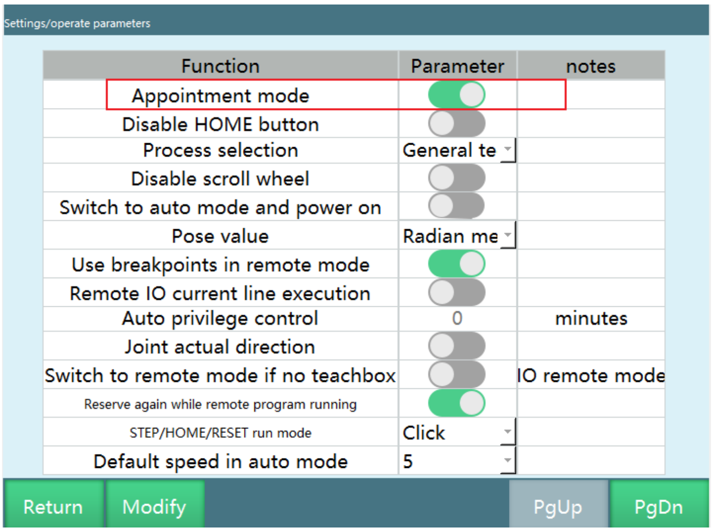
The program used by the I/O control module needs to be set in the I/O function interface.

The selected program on the remote program interface can be canceled by clicking the "Cancel" button.

Just fill in the corresponding number for the running times, 0 means cycle running.

Reservation mode

In "Settings/Operation parameters":



After the "Reservation mode" is enabled, if the remote IO program signal is triggered, the program reservation is successful, and if the start signal is triggered, the robot runs;

After the "Reservation mode" is disabled, if the remote IO program signal is triggered, the robot runs directly. At this time, triggering other remote IO program signals is invalid. After the robot runs, the remote IO program signal can be triggered again. There is no need to set a start signal.

Use of remote function (IO)

Remote function overview

Set 10 remote programs and the running times of each program, queue the 10 programs before running, run them according to the order in the queue and the running times, and stop and wait to queue again after the queue is finished.

Steps to use remote function

Write program——Set remote program——Set IO——Switch to remote mode——Reserve and sort——Run

1.Write program

Create a new program and insert the instruction, please make sure the program can run properly.

2.Set remote program

Enter the "Settings-Remote program setting" interface, set the program name and running times for program 1 to program 10, if you want the single program to run in an infinite loop, then set the running times for the program to 0. The program name here refers to the program in the "Project" interface, and the remote program will be modified automatically after the instruction in the program is modified, so there is no need to reset the remote program.

If the program name of the program is modified, please reset the program in the "Remote program setting" interface.

3.Set IO

In the "IO-IO function" interface, set the corresponding IO port and effective value of each function. When the effective value is 1, the high level is valid, and when the effective value is 0, the low level is valid.

The function of the IO port corresponding to program 1-program 10 is not to select the program to run, but to queue the program in the remote mode.

4.Switch to remote mode

Turn the mode selection key to the remote mode position or click the mode status in the program to select the remote mode.

When the teach pendant is not connected to the controller, the controller will automatically enter the remote mode when it is started.

When the controller is connected to IO, Modbus device, and teach pendant at the same time, the priority of the three devices is teach pendant>Modbus device>IO device. After switching to the remote mode, the Modbus device is valid and the IO device is invalid. At this time, if you turn off the enable button in the Modbus device, the IO device will be valid.

5.Reserve and sort

For example: The IO function in the IO function is set to

run port 1 valid value 1

stop port 2 valid value 1

pause port 3 valid value 1

clear alarm port 4 valid value 1

program 1 port 5 valid value 1

program 2 port 6 valid value 1

program 3 port 7 valid value 1

program 4 port 8 valid value 1

program 5 port 9 valid value 1

program 6 port 10 valid value 1

program 7 port 11 valid value 1

program 8 port 12 valid value 1

program 9 port 13 valid value 1

program 10 port 14 valid value 1

Then the sorting is such that if you give port 6 a high level for 1 second and then release it, program 2 will be the first in line, if you give port 8 a high level for 1 second and then release it, program 4 will be the second in line, and so on. If you want to dequeue a program in the queue, give the corresponding IO port a high level for 1 second, the program will be dequeued in the queue.

There can only be 10 programs in the queue, and the same program cannot be queued repeatedly.

When a program is running, it can be re-added to the end of the queue.

6.Run

Give a high level to the port with running function, and the robot will start to run according to the order in the queue and running times. After the operation is completed, the servo will not be powered off. At this time, add the program to the queue, and the robot will run the program immediately.

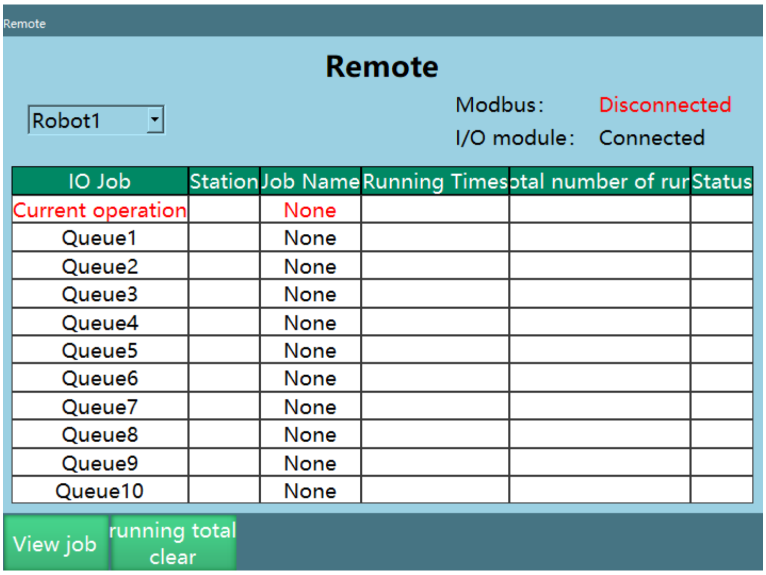
When there is no program in the queue, if you let the robot run, it will power on but not operate. At this time, if you place the program in the queue, the robot will execute the program immediately.

View operation

To view the details of program operation through remote IO control, click the "View program" button in the remote mode interface, modbus can also be viewed through this function.

Total running clear

Clear the total running times of the currently running program, only the total running times can be cleared, but the running times cannot be cleared.



MODBUS

Modbus instructions

Open MODBUS connection

This instruction is used to open the modbus communication connection in run mode, the process number bound is the modbus master process number

Disconnect modbus connection

This instruction is used to disconnect the modbus communication connection in run mode, the process number bound is the modbus master process number

Get modbus connection status

This instruction stores the connection status of modbus in the bool variable, and determines the connection status of modbus by getting the value of the variable. This instruction gets status every time it is run, and the status is often placed under "Open modbus connection".

Modbus read

This instruction is used to read the address code of the corresponding location in modbus, and the address types that can be set are 3x, 4x-bit, 3x-bit, 0x.

Slave register first address: the first address to read

Number of reads: the total number of addresses to be read

First variable type: get the variable where the data is stored

First variable name: get the first variable name where the data is stored

Modbus write

This instruction is used to write variables into the address code of the corresponding location in the slave register via modbus. The address types that can be set are 4x, 4x-bit, and 0x, and 3x and 3x-bit are missing compared to Modbus read.

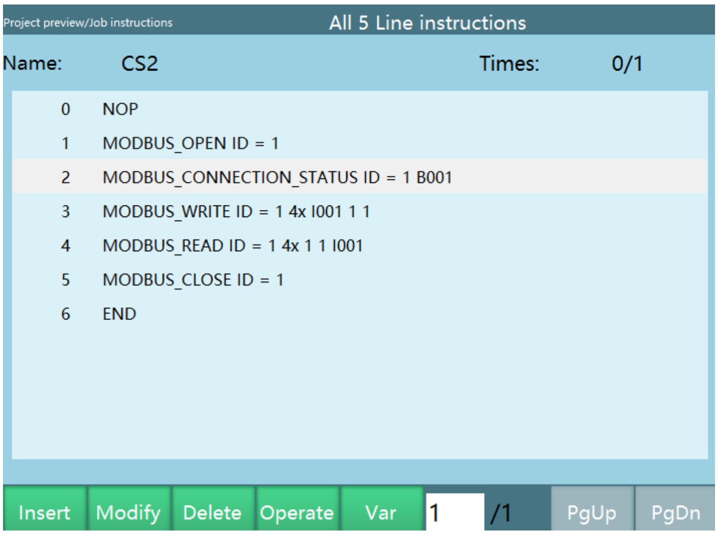
First variable type: the type of the first variable written

First variable name: the first written variable name

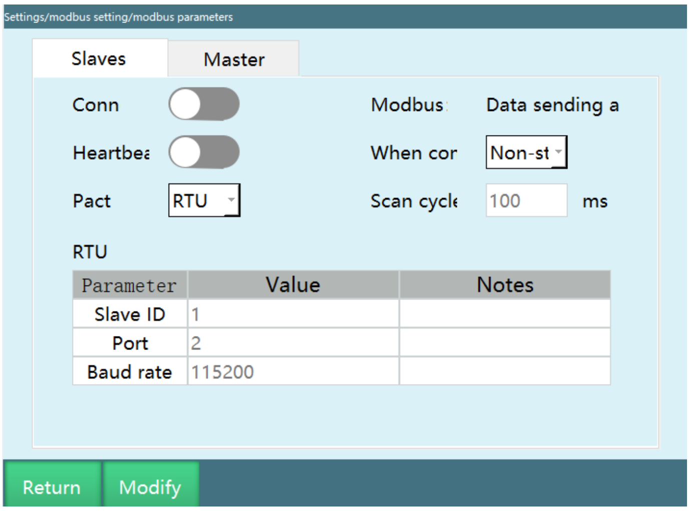
Slave register first address: the first address to write

Number of writes: the total number of addresses to be written

The specific usage is as follows:



Modbus slave



When acting as slave, you can set heartbeat detection to confirm the communication status with the master station, and whether to stop when the communication is disconnected. You can choose RTU or TCP protocol, and its corresponding port settings.

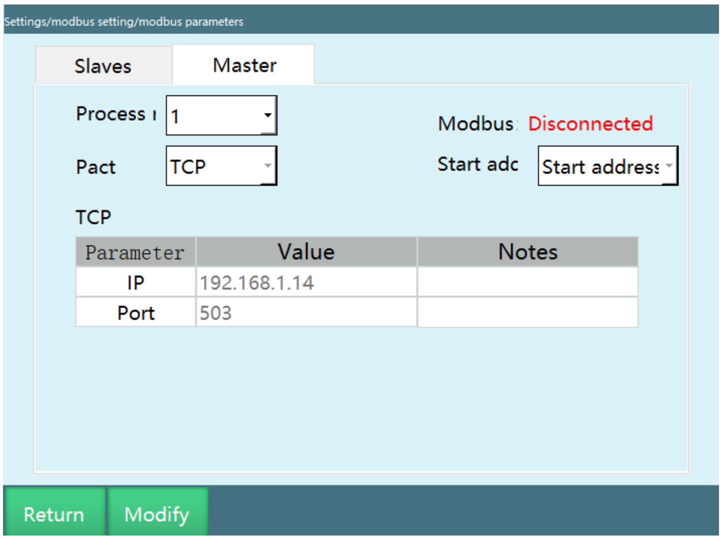
Heartbeat detection: When turned on, the system will determine the status of communication with the modbus, and when turned off, it will not monitor the communication connection with the modbus, which is generally used for RTU protocol.

When communication is disconnected: If "Stop" is selected, the robot is powered off when modbus communication is disconnected. Conversely, if "Non-stop" is selected, the robot does not power off when the modbus communication is disconnected.

Protocol: RTU or TCP.

Scan period: Refers to how often the system scans for data in range in the modbus

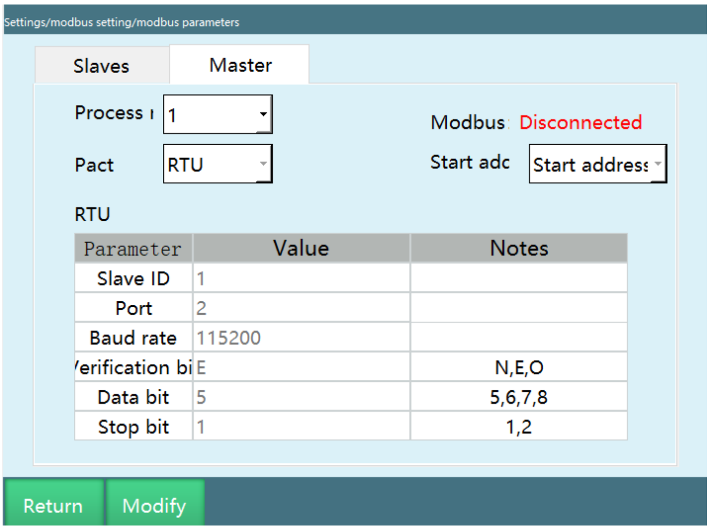
Modbus master



When acting as master, only the communication method and its corresponding port settings can be selected.

"Start address" can be set to "Start address is 0" or "Start address is 1"

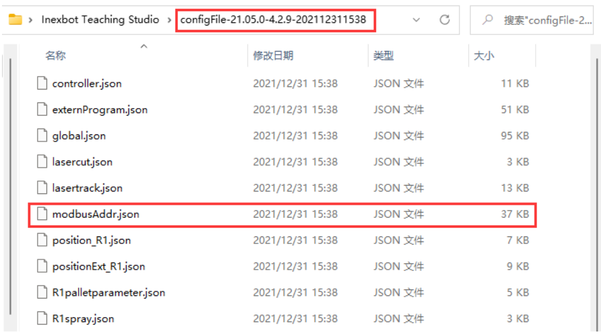
When the master protocol is set to RTU, the "Check bit", "Data bit", and "Stop bit" need to be set.



Modbus address code modification

1.Insert the U disk, export the controller configuration

2.Find the configuration file modbusAddr.json in the configFile+date folder



3.Open it with a text editor such as Notepad++

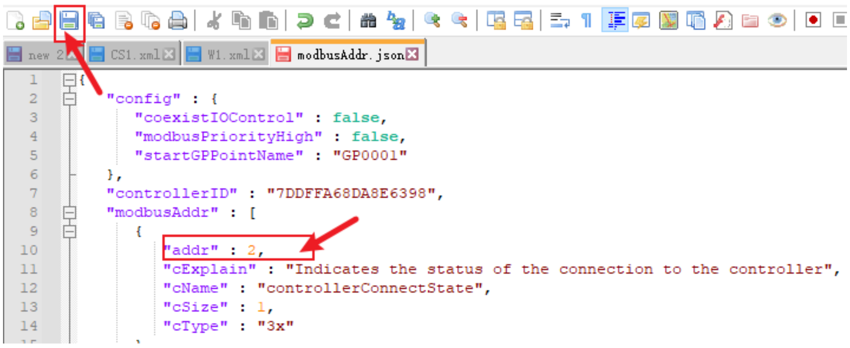
4.When opened, you can see a {...} containing a set of address code parameters (the system will automatically generate a copy of the original address code)

```
"modbusAddr" : [
    {
        "addr" : 1,
        "cExplain" : "Indicates the status of the connection to the controller",
        "cName" : "controllerConnectState",
        "cSize" : 1,
        "cType" : "3x"
    },
```

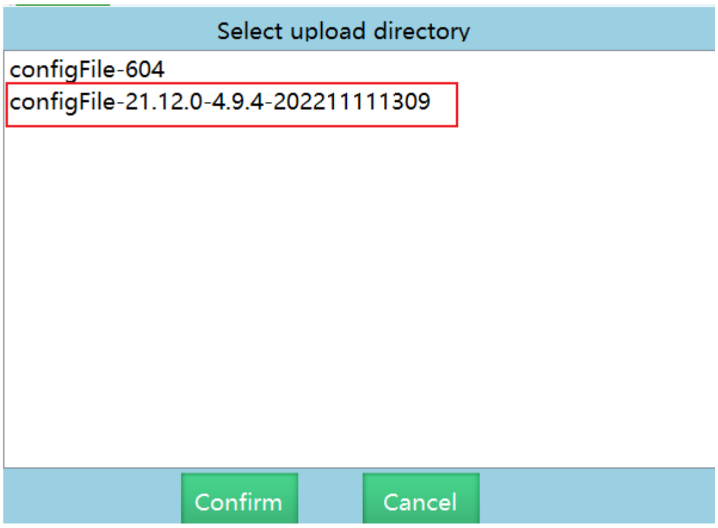
5.To modify the address, you only need to directly change the number after addr. When the number is 0, the function of the address code is invalid

```
"modbusAddr" : [
    {
        "addr" : 1,
        "cExplain" : "Indicates the status of the connection to the controller",
        "cName" : "controllerConnectState",
        "cSize" : 1,
        "cType" : "3x"
```

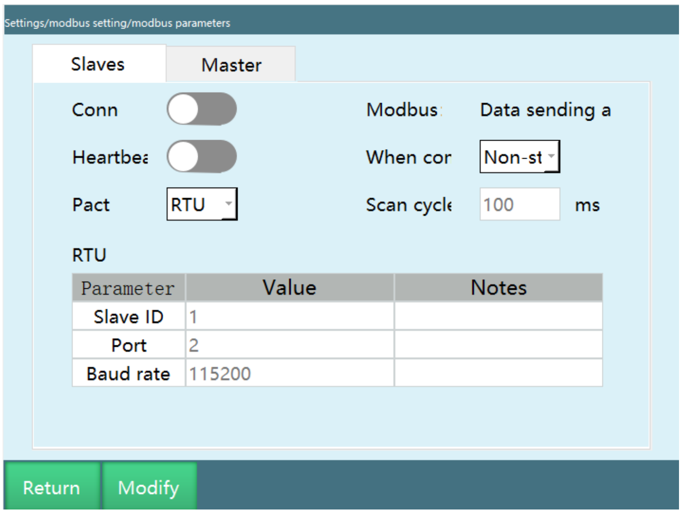
6.Click "Save" after modification



7.Then re-import the parameter into the controller, restart to take effect



8.After modifying the parameters, restart or reopen the connection to take effect (The system will automatically restart if the configuration file is imported)



Use of Modbus

Function overview

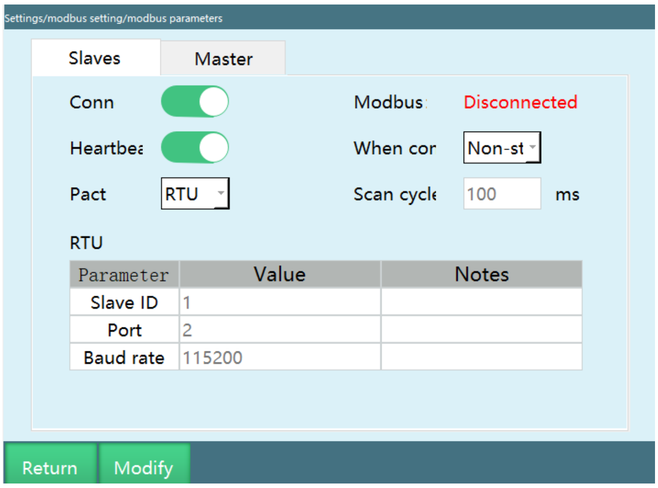
The Modbus function can replace some functions of the teach pendant, remotely control robot operation, teach, view status, etc.

Modbus supports modbusTCP and modbusRTU protocols.

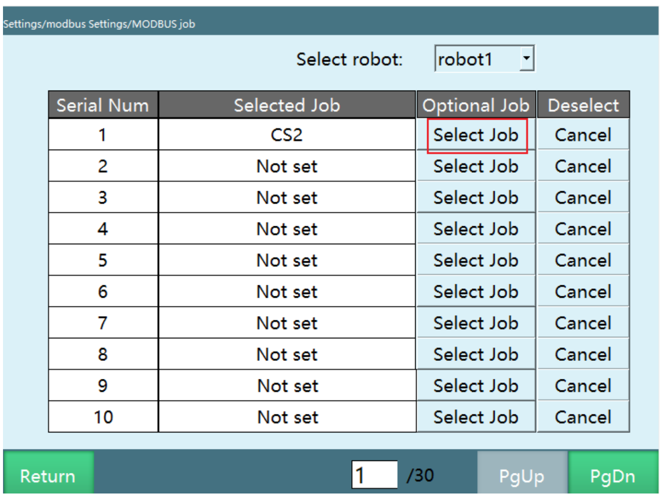
Modbus has two modes: teach and run. For address codes, see "MODBUS Address Code List.xls" for details.

Modbus poll connection method

1.Enter the "Settings/modbus settings/modbus parameters" interface and turn on the connection switch

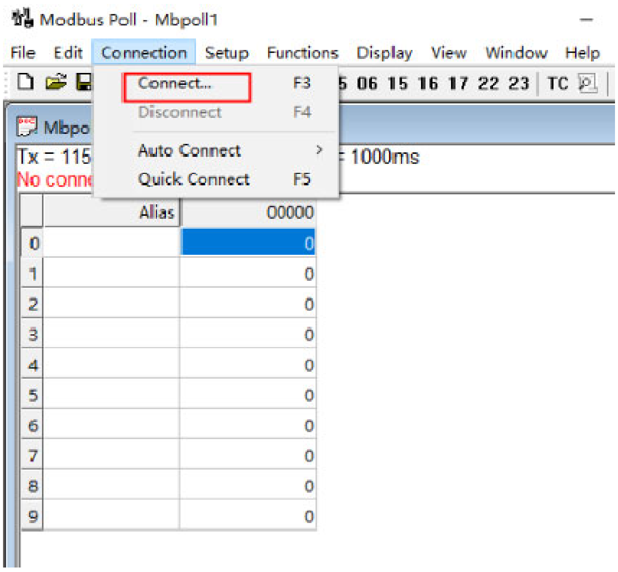


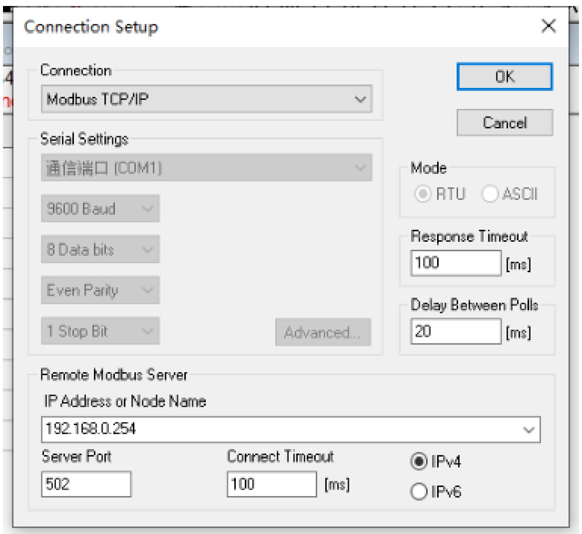
2.In the "Settings/modbus settings/modbus program" interface, select the program



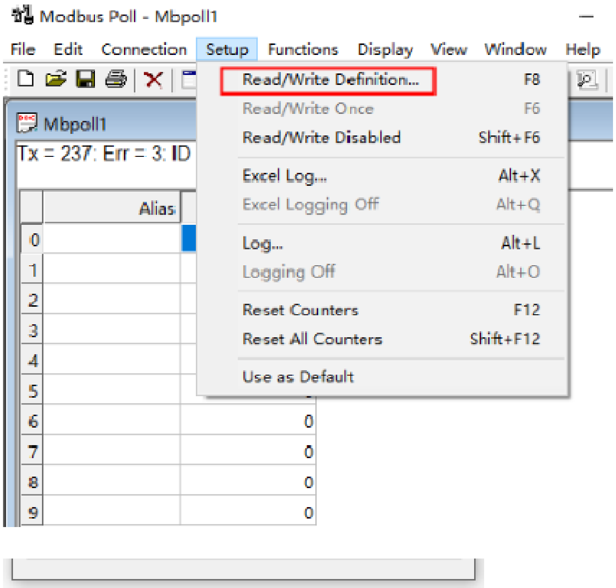
3.Open the ModbusPoll software

4.After opening the software, we need to connect and set (Connection——Concection Setup), set the required parameters (the parameters in the picture are just examples), click the "OK" button, and the teach pendant page will display connected, if connected and unconnected blinking screen is displayed, you need to change the Scan Rate parameter from 1000ms to 100ms in Setup——Read/Write Definition

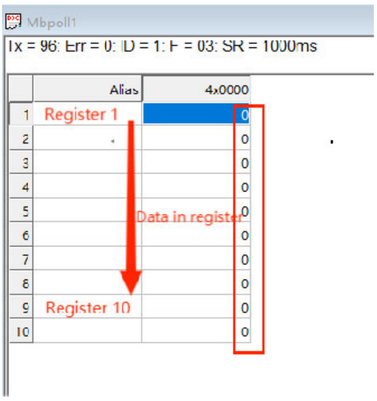


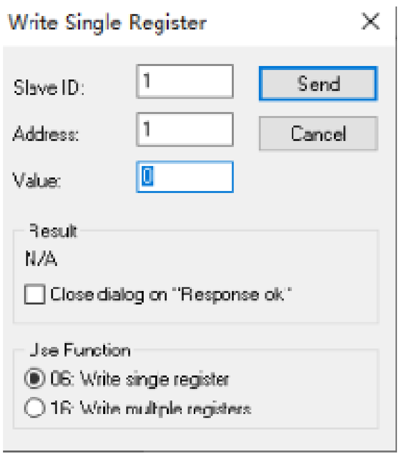


5.Set the parameters under Setup-----Read/Write Definition (the parameters in the picture are only examples). If the address code does not take effect, you can troubleshoot: change the start address to 1, check the "PLC Addresses" option, and click "OK"



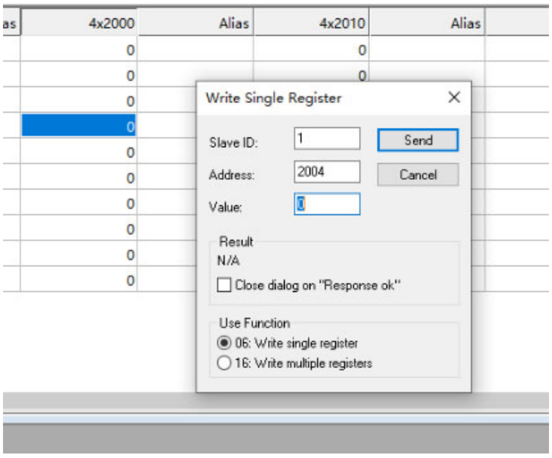
6.Double-click the register data and fill in the relevant address code



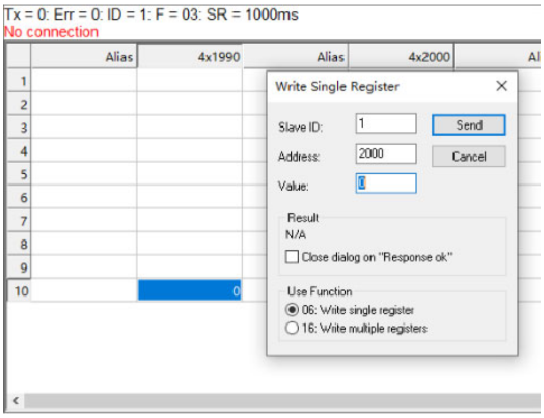


Modbus readable global positions in any mode

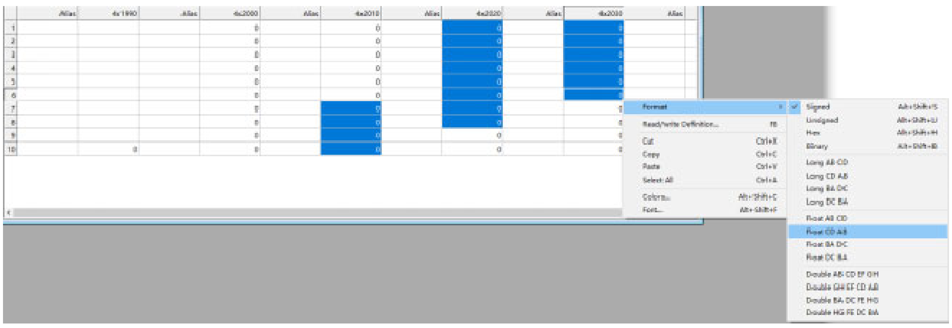
1.Select the global type to be read according to Modbus address 2004: 0 means GP point, 1 means GE point.



2.Select the number of the global point to be read according to Modbus address 2000, numbered from 1 to 999.



3.Change the Format of 2017-2028 and 2031-2036 to FloatCD AB



4.The Modbus address code starts from 2017 and includes 2017, and every 2 address codes represent the value of one axis of the robot, that is, 2017 and 2018 represent axis 1.

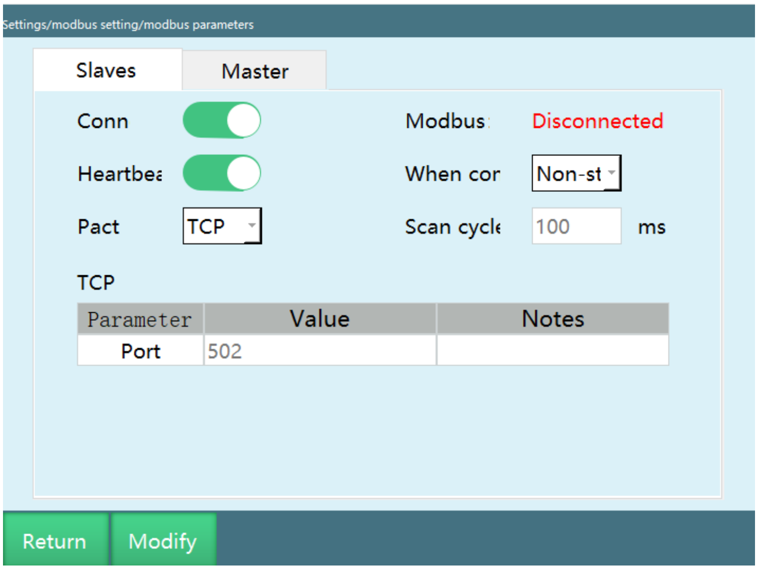
5.Address codes 2031-2036 represent external axes, and only support 3 external axes.

6.After the Modbus is successfully connected, select the global point number through the address code 2000, and select the global point type through the address code 2004, then you can view the global point in address code 2017.

Modbus multi-master connection

1.Connect the computer and one or more touchpads to the controller through the switch.

2.The controller acts as a slave station, and the modbus poll and touchpad act as the master station. You can open multiple modbus polls to act as multiple master stations. Currently, the controller supports up to 9 master stations to be connected at the same time.



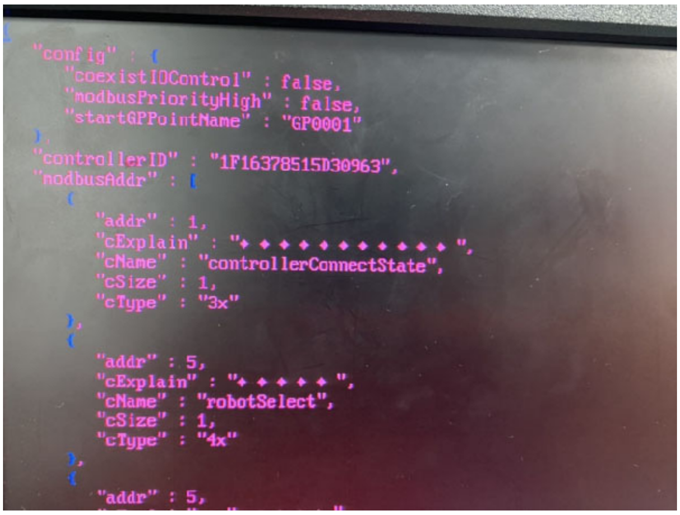
3.In the modbus poll, click "Connection" and select "Connect", select TCP as the connection type, keep the same IP address and port number as the teach pendant, and keep the same scan cycle as the teach pendant.

4.Modbus Poll and touchpad can control the robot at the same time.

Modbus and IO priority

1.Enter the teach pendant through the display, and find modbusAddr.json in the robot/config/ directory.

2.Open modbusAddr.json with vi editor.



3.coexistIOControl: false means modbus and IO are not used at the same time, i.e. IO cannot control the robot when modbus is connected; true means modbus and IO are used at the same time, i.e. modbus and IO can control the robot at the same time.

4.When coexistIOControl is false, whether modbusPriorityHigh is false or true, modbus always has a higher priority than IO by default, and IO cannot control the robot when modbus is connected.

5.When coexistIOControl is true, if modbusPriorityHigh is false, it means that modbus and IO are used at the same time and IO has a higher priority than modbus, i.e. modbus and IO can control the robot at the same time and modbus runs according to the IO settings (settings on the teach pendant), e.g. breakpoint and current line run.

6.When coexistIOControl is true, if modbusPriorityHigh is true, it means that modbus and IO are used at the same time and modbus has a higher priority than IO, that is, modbus and IO can control the robot at the same time, but modbus and IO each run according to their own settings, for example, if you close the breakpoint execution in the teach pendant, then inputting 0 (stop) and then 3 (breakpoint execution) in modbus address code 19 will start breakpoint execution, but IO control cannot.

Modbus touch screen usage process

This section uses Weiluntong touch screen and modbusTCP protocol as examples; the touch screen model is MT6071iP.

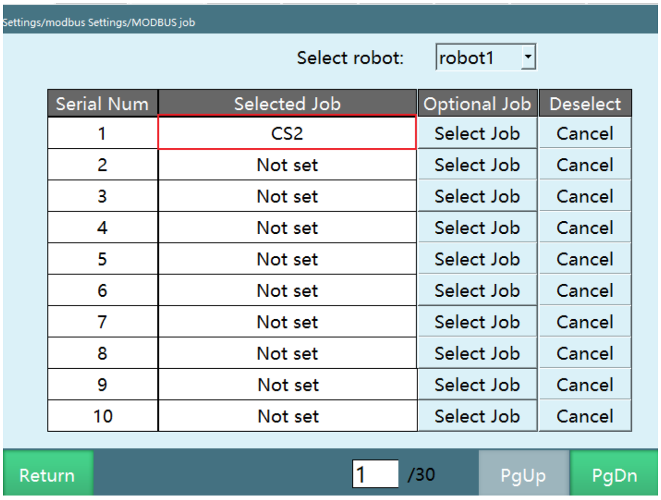
Write program——Set Modbus program——Set Modbus parameters——Switch to remote mode——Touch screen preparation——Select program——Run

(1)Write program

Write the program with the teach pendant, and make sure it can run normally.

(2)Set Modbus program

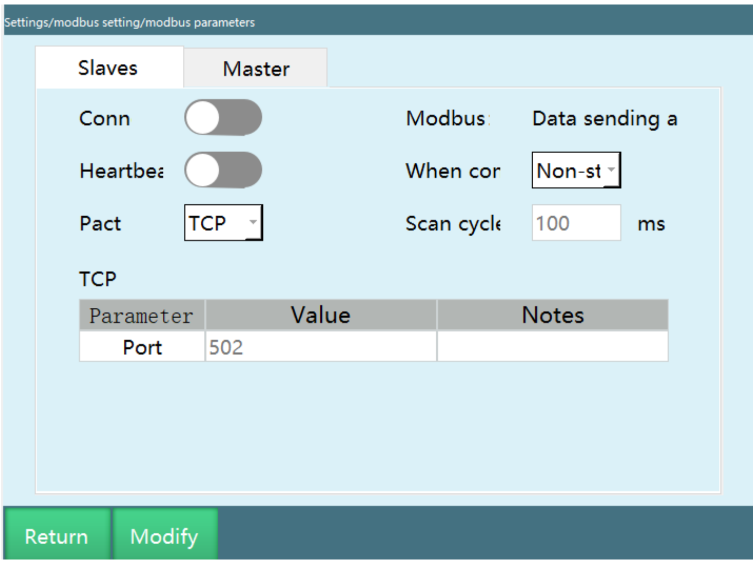
Set the program in "Settings-Modbus setting-Modbus program", if the setting is successful, the selected program list will display the program name.



You can set up to 1000 programs in total

(3)Set Modbus parameters

Set the protocol to TCP in "Settings-Modbus settings-Modbus parameters", set the controller as slave, leave the IP unchanged, set the port to 502, and enable the connection; it will take effect after restarting the controller.



Modbus parameter description

Connect: You need to turn on the connection button after Modbus setup, and you can check the connection status on the right side.

Heartbeat detection: turn on to detect the frequency of sending and receiving between Modbus and the controller, and heartbeat detection shows that data sending and receiving is off after the Modbus connection is disconnected

Protocol: TCP, RTU.

Master/Slave: master station, slave station.

TCP parameters

IP: Modbus device IP address, only valid when it is set as the master station.

Port: Modbus device port

RTU parameters

Slave ID: the default is 1

Port: serial port number of the controller

Baud rate: fill in the baud rate corresponding to the touch screen

1.Switch to remote mode

Turn the mode selection key to the remote mode position or click the mode status in the program and select the remote mode.

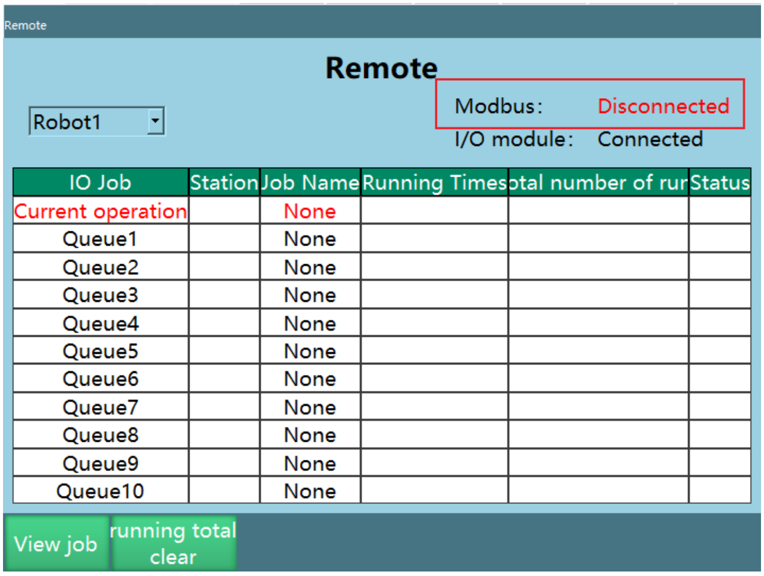
Note: When the controller is connected to IO, Modbus device and teach pendant at the same time, the priority of the three devices is teach pendant>Modbus device>IO device. When switching to remote mode, the Modbus device is valid and the IO device is invalid. If the enable button in the Modbus device is turned off at this time, the IO is valid.

2.Touch screen preparation

Connect the RJ45 network port of the touch screen, the network port of the teach pendant, and the network port of the teach pendant on the controller to the same switch.

Connect the touch screen to the controller: IP: 192.168.1.13; port: 502.

After the touch screen program is edited and run, the modbus will change from unconnected to connected status on the remote interface of the teach pendant.



3.Select program

Use the touch screen to write 1 to address code 45 of type 4x, select demo program 1 for robot 1.

Use the touch screen to write 5 to address code 61 of type 4x, set running times 5 for robot 1 (not valid); use the touch screen to write 1 to address code 71 of type 4x, confirm the modification of running times (running times 5 takes effect).

Run

Use the touch screen to write 1 to address code 29 of type 4x, switch to servo-ready status.

Use the touch screen to write 1 to address code 19 of type 4x, run the job file.

Network Function User Manual

<<<

Catalogue

Network Function User Manual

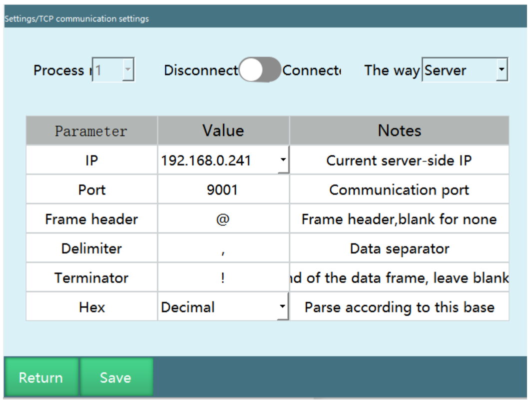
TCP External Communication

TCP Communication

When communicating with external devices, TCP communication can be selected.

Parameter settings

You can set TCP communication on the "Settings - TCP communication settings" interface.



Process number: 9 process numbers are supported.

Connection switch: When the mode is Client, white means disconnected, green means connected; when the mode is Server, white means closed, green means open. (The interface will indicate successful communication connection when connected)

Mode: Use the controller as a server or client. (When the controller is a client or a server, it can use different process numbers to communicate with multiple external devices and send and receive data; Note: The IP and port between the process numbers cannot be the same when the controller is a client with multiple process numbers connected to multiple server devices, and the port cannot be the same when the controller is a server with multiple process numbers connected to multiple client devices)

IP: When the controller is used as a server (the mode is Server), the IP here is the controller IP, no modification is required. When the controller is used as a client, the IP here needs to be set to the IP of the external communication device.

Port: When the mode is Server, it is the local listening port for client connection; when the mode is Client, it is the port for server connection.

Frame header: The frame header that is used when the controller receives messages from external devices during data communication; it can be modified.

Separator: The separator that is used when the controller receives messages from external devices during data communication; it can be modified.

Terminator: The terminator that is used when the controller receives messages from external devices during data communication; it can be modified.

Base: Select the corresponding base for the decimal or hexadecimal data to be received, then parse it in decimal and output.

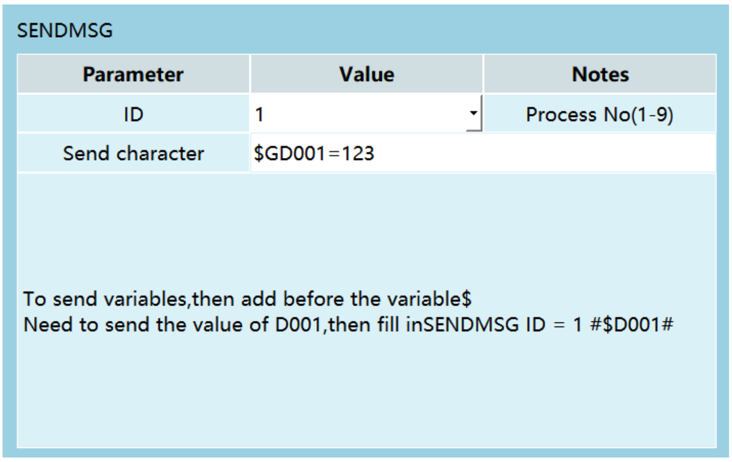
Note: When connecting through TCP communication, first set the IP of the controller and the IP of the external device to the same network segment, such as 192.168.1.xxx. If the controller is set as the client, the external device is the server, then set the IP and port in the network settings to be the same as those in the external device network debugging software, turn on the connection switch, the system will prompt that the connection is successful

Network communication class instructions

SENDMSG-Send data

This instruction is used to send data to the connected external device. Strings and variables can be sent by selecting the corresponding process number. Strings and variables can be mixed and sent. The frame header, separator, terminator and base set in the "Settings - TCP communication settings" interface are not used when sending data to external devices.

If you want to send a variable, add $ before the variable.



For example:

Prerequisite: GD001=123, I001=10

Need to send data "The value of GD001 is 123, and the value of I001 is 10" to the upper computer whose network setting process number is 3

Insert instruction SENDMSG:

ID=1

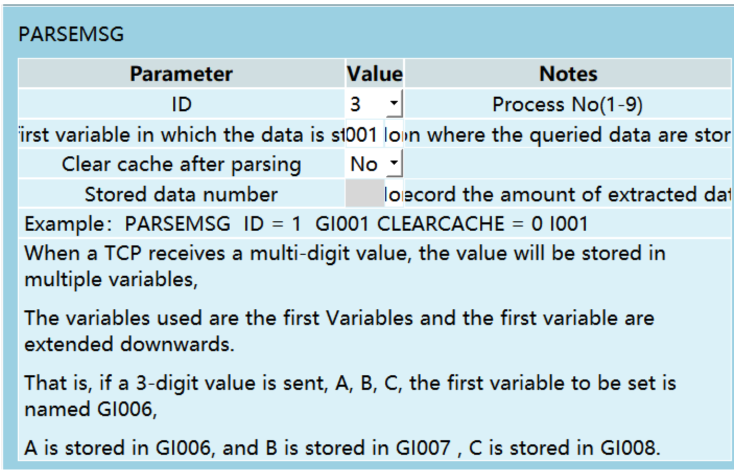
Send characters: The value of GD001 is $GD001, and the value of I001 is $I001

PARSEMSG-Parse data

This instruction is used to parse a set of data from the external device.

This instruction will store the data from the external device in several global variables, and it need to set the first variable.

Not clear the cache after parsing: The data sent by the external device will be temporarily stored in the controller's cache; a. Not clear the cache: Data remains in the cache until the next set of data is sent after the first parsing; b. Clear the cache: The data in the cache will be cleared after the first parsing is completed.



PARSEMSG

Parameter	Value	Notes
ID	3	Process No(1-9)
irst variable in which the data is s	I001	lo n where the queried data are stor
Clear cache after parsing	No	
Stored data number		lo ecord the amount of extracted dat

Example: PARSEMSG ID = 1 GI001 CLEARCACHE = 0 I001

When a TCP receives a multi-digit value, the value will be stored in multiple variables,

The variables used are the first Variables and the first variable are extended downwards.

That is, if a 3-digit value is sent, A, B, C, the first variable to be set is named GI006,

A is stored in GI006, and B is stored in GI007 , C is stored in GI008.

For example:

Frame header: @

Separator: ,

Terminator: ！

The first variable type of the PARSEMSG instruction is GDOUBLE, and the first variable name is GD003.

External device sends data: @,12,6,47,102,77.88,！

Then the EXPLAIN instruction stores these five values in GD003, GD004, GD005, GD006, and GD007 respectively.

GD003=12, GD004=6, GD005=47, GD006=102, GDOO7=77.88

GD003=12

GD004=6

GD005=47

GD006=102

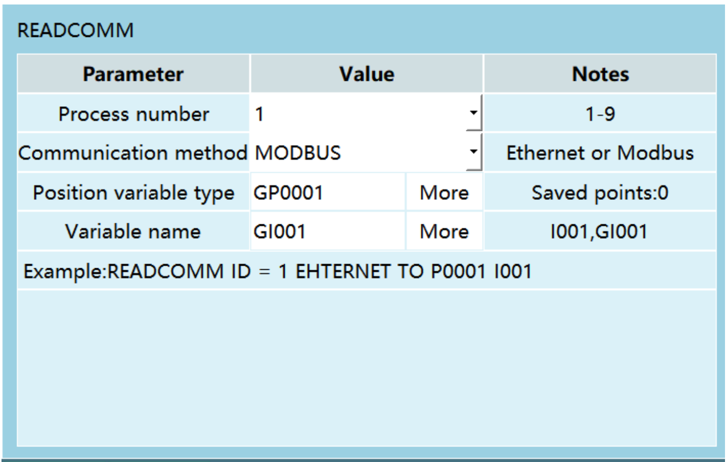
GD007=77.88

Clear cache after parsing: Yes or No

READCOMM-Read

Read the points sent by Ethernet or Modbus and store them in the position variable, and store the number in the numerical variable.

Note: The method of use is the same as "External Point Function", this instruction currently only supports Modbus



READCOMM

Parameter	Value		Notes
Process number	1		1-9
Communication method	MODBUS		Ethernet or Modbus
Position variable type	GP0001	More	Saved points:0
Variable name	GI001	More	I001,GI001

Example:READCOMM ID = 1 EHTERNET TO P0001 I001

Process number: The process number of the network communication to open the communication.

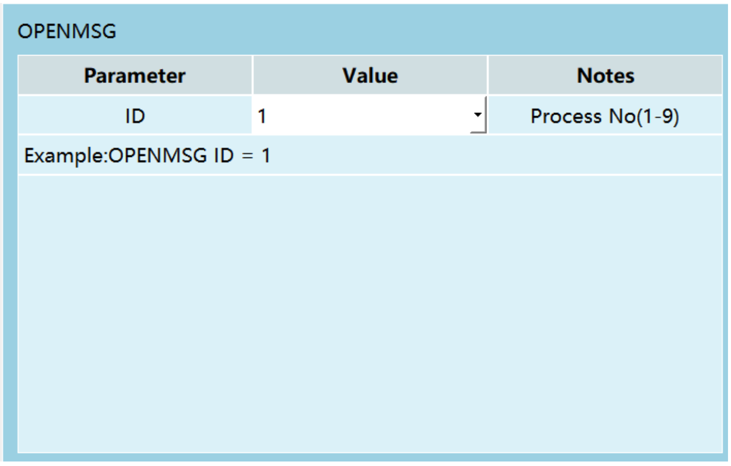
Communication method: Use Ethernet communication or Modbus communication.

Position variable type: Global position variable and local position variable can be selected.

Variable name: Store the number of received points.

OPENMSG-Open data

Open the network communication corresponding to the process number. Run the OPENMSG instruction to open the communication. (Connect communication)



OPENMSG

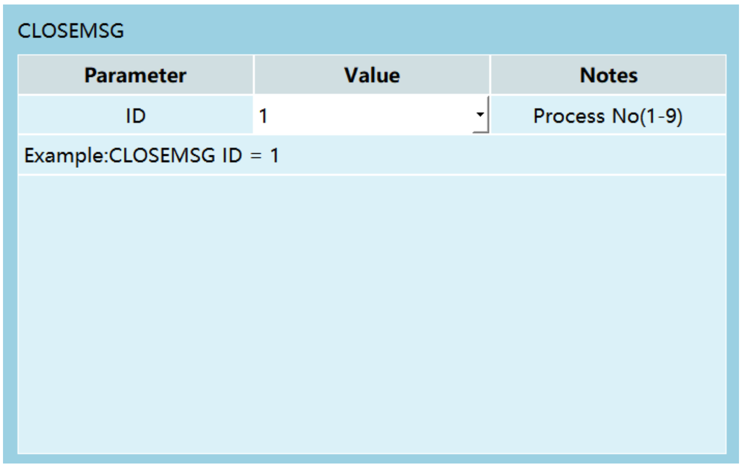
Parameter	Value	Notes
ID	1	Process No(1-9)

Example:OPENMSG ID = 1

Process number: The process number of the network communication to open the communication.

CLOSEMSG-Close data

Close the network communication corresponding to the process number. Run the CLOSEMSG instruction to close the communication. (Disconnect communication)



CLOSEMSG

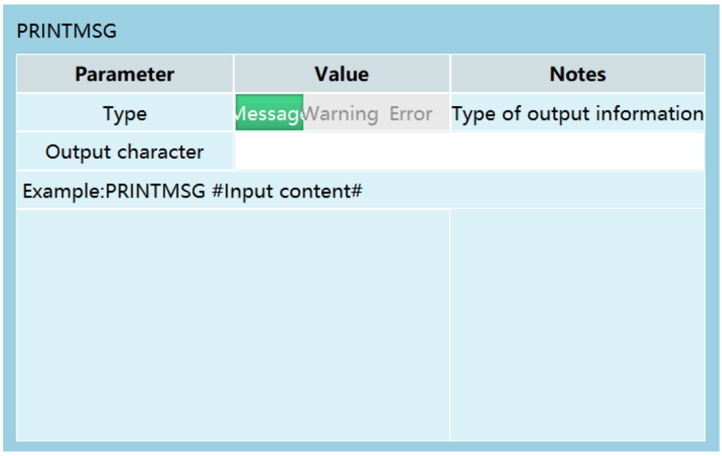
Parameter	Value	Notes
ID	1	Process No(1-9)

Example:CLOSEMSG ID = 1

Process number: The process number of the network communication to close the communication.

PRINT-Output information

Screen output instruction; display the content on the teach pendant in three forms. Can output the data of custom characters or variables


PRINTMSG

Parameter	Value	Notes
Type	Messag Warning Error	Type of output information
Output character		

Example:PRINTMSG #Input content#

Output information is now divided into three types: message, warning and error.

Output character: Output characters. Can input any character (support escape character), also can output variables, such as GD001 variable, GD001=10;

In the output information's message, warning and error instructions, enter $GD001 separately

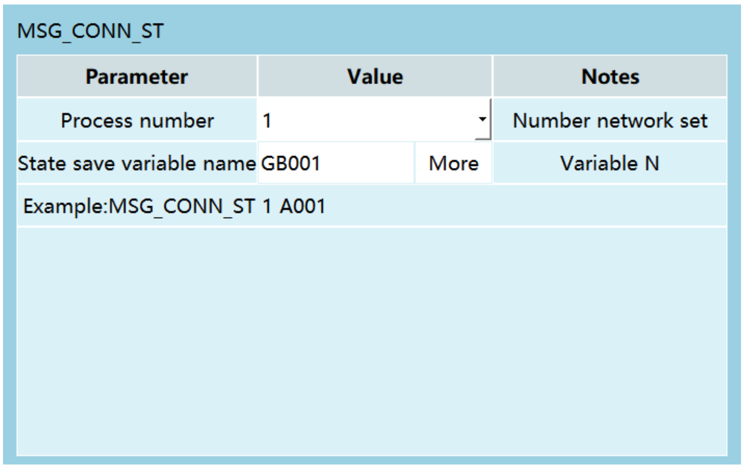
Now when running or stepping this instruction, the lower right corner of the teach pendant will display like this:

The message is a small white bar with the following content: 10;

The warning is a small yellow bar with the following content: 10;

The error is a small red bar with the following content: 10;

MSG_CONN_ST-Get information connection status



Process number: The process number for judging the connection status of network communication.

Stored variable type: Store the communication status into the local BOOL variable or the global GBOOL variable.

Stored variable name: The variable name of the variable that stores the communication status.

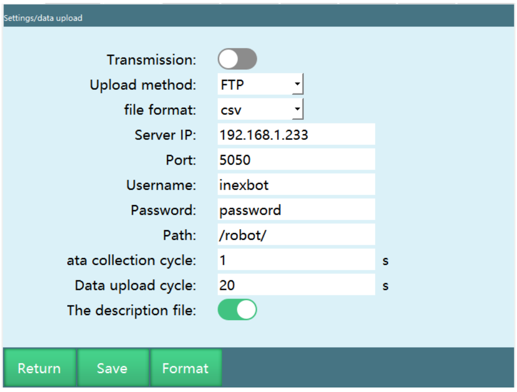
Read the current process number network communication status into the corresponding global Boolean or local Boolean variable. If the communication is normal, the stored value is 1, and if the communication fails, the stored value is 0.

Data Upload

Basic settings

The data upload function can automatically collect and upload the current robot operating status and parameters at regular intervals, integrate the data into csv and txt files and upload them to the designated server.

Click the [Modify] button in the "Settings - Data upload" to set the parameters required to connect to the ftp server.



Transmission: Once turned on, it starts to connect to the ftp server and upload data. After all parameters are filled in, turn on this switch. After this switch is turned on, the controller will automatically start collecting and uploading data when it is started up.

Upload method: Currently, only ftp protocol is supported. So please have an ftp server before using this function.

File format: Currently supports csv and txt formats. The file content is same, but the file format is different. The csv format is more convenient for data statistics.

Server IP: The IP address of the ftp server. Please ensure that the controller and the ftp server are in the same network, and ensure that their gateways are the same (the controller gateway can be viewed and modified in "Settings - System settings - IP settings").

Port: The port used by the ftp protocol of the ftp server. The default port used by the general ftp protocol is 21.

Username: The username used to log in to the ftp server. You need to create a user on the ftp server first.

Password: The password used to log in to the ftp server.

Path: The path used when the file is uploaded to the ftp server. This path is relative to the ftp root directory.

Data collection cycle: According to the set time, the controller collects the current data once and stores it in the file to be sent at regular intervals.

Data upload cycle: According to the set time, the controller sends the file with collected data to the specified directory of the ftp server at regular intervals.

Send description file: The description file is sent before the first sending of the data file after starting up the controller or turning on the "Transmission" enable switch. The content is customizable and is generally used to describe the current robot's serial number and other information. If this switch is turned off, the description file will not be sent.

Data format

Once you have configured the connection parameters for ftp, you need to configure the data format of the data file to be sent. When setting the data format, use a special string to represent the parameters to be sent. For example, if you want to send the current date in the following format "2019-03-07", you need to fill in the data format as follows: "Y%-m%-d%" (excluding quotation marks).

If the generated file is in csv format, each item should be separated by English commas (,).

The parameters represented by special strings are as follows:

Example of generating a csv file

The desired results are as follows:

Description document file name: Robot-R1_Year-Month-Day_Hour:Minute:Second_INFO

Description document content: Robot-R1, year-month-day, hour: minute: second, local IP, local MAC, technical department, machining parts, 1-axis motor speed, 2-axis motor speed, 3-axis motor speed, 4-axis motor speed, 5-axis motor speed, 6-axis motor speed, 1-axis motor torque, 2-axis motor torque, 3-axis motor torque, 4-axis motor torque, 5-axis motor torque, 6-axis motor torque, 1-axis motor load, 2-axis motor load, 3-axis motor load, 4-axis motor load, 5-axis motor load, 6-axis motor load, current controller status, current error code

Data document file name: Robot-R1_Year-Month-Day_Hour:Minute:Second_DATA

Data content: Robot-R1, year-month-day, hour: minute: second, local IP, local MAC, 1-axis motor speed, 2-axis motor speed, 3-axis motor speed, 4-axis motor speed, 5-axis motor Speed, 6-axis motor speed, 1-axis motor torque, 2-axis motor torque, 3-axis motor torque, 4-axis motor torque, 5-axis motor torque, 6-axis motor torque, 1-axis motor load, 2-axis motor load, 3-axis motor Load, 4-axis motor load, 5-axis motor load, 6-axis motor load, current controller status, current error code

The format of the data written is as follows:

Description document file name: Robot-R1_ $Y%-$m%-$d%_ $H%:$M%:$S%_INFO

Description content: Robot-R1,$Y%-$m%-$d%,$H%:$M%:$S%, $IP%,$MAC%,Technology Department,Machining Parts,$RPM_J1%,$RPM_J2%,$RPM_J3%,$RPM_J4%,$RPM_J5%,$RPM_J6%,$Torsion_J1%,$Torsion_J2%,$Torsion_J3%,$Torsion_J4%,$Torsion_J5%,$Torsion_J6%,$Load_J1%,$Load_J2%,$Load_J3%,$Load_J4%,$Load_J5%,$Load_J6%,$StatusCode%,$ErrorCode%

Data document file name: Robot-R1_ $Y%-$m%-$d%_ $H%:$M%:$S%_DATA

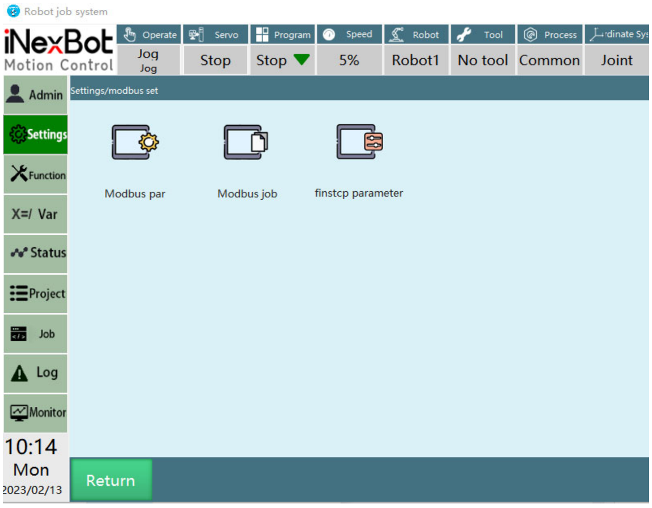
Data content: Robot-R1,$Y%-$m%-$d%,$H%:$M%:$S%, $IP%,$MAC%,$RPM_J1%,$RPM_J2%,$RPM_J3%,$RPM_J4%,$RPM_J5%,$RPM_J6%,$Torsion_J1%,$Torsion_J2%,$Torsion_J3%,$Torsion_J4%,$Torsion_J5%,$Torsion_J6%,$Load_J1%,$Load_J2%,$Load_J3%,$Load_J4%,$Load_J5%,$Load_J6%,$StatusCode%,$ErrorCode%

*For the parameters related to the axis, you need to manually input that axis, such as 1 axis speed: $RPM_J%, you needs to write 1 after J

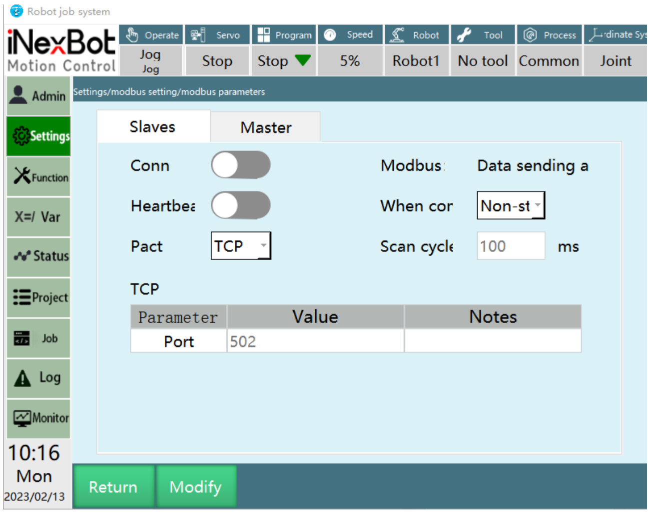
External Transmission Point

Parameter settings

External communication can use modbus. To set the parameters, you need to enter the "Settings - modbus settings - modbus parameters" interface. (You can also check modbus-related manual)



Parameters for detection and status display of switches



Connection: modbus-related switch, check the modbus signal after it is turned on.

Heartbeat detection: Turn on to detect the sending and receiving frequency between modbus and the controller. After disconnecting the modbus connection, the heartbeat detection shows that the data sending and receiving is closed.

Modbus: Display the connection status between modbus and the controller.

When the communication is disconnected:

No shutdown: when the Modbus slave is disconnected and the communication is disconnected, the controller will not stop running or power off;

Shutdown: When the Modbus slave is disconnected and the communication is disconnected, the controller will stop running or power off.

Scan cycle: The time required to perform a scan operation.

In this interface, you can set whether modbus is connected, the protocol used for modbus connection, the controller as a modbus master/slave, and each parameter when connected.

Controller is the master

TCP (port): the connection port of the slave

RTU (port): the port to which the slave modbus is connected

RTU (ID): ID of the slave

RTU (baud rate): baud rate of modbus, need to set

Controller is the slave

TCP (port): the port used by the controller to connect, need to set

RTU (ID): ID used by the controller to connect, need to set

RTU (port): the port used by the controller to connect, need to set

RTU (baud rate): the baud rate used by the controller to connect, need to set

Communication method

Due to the limitation of address code, if there are too many points, then they need to be sent in batches, and a maximum of 30 points can be sent each time.

As long as the controller is connected to the PLC, the points can be sent, and the controller will automatically store them.

Purpose	Address code	Process
All points sending flag	1001	Set it to 1 when PLC needs to send points, set it to 2 after sending all points, and set it to 0 after the controller receives them.

The sending flag for sending once	1002	Set it to 1 when PLC needs to send points, set it to 0 after the controller receives them, and PLC sets it to 1 again for the next sending process.
The number of points sent once	1003	The number of points sent by PLC at one time, up to 30
Data stored at points	According to the number	See below for detailed explanation
Frame number for each frame of data	1004	The number must be changed every time the points are sent, and it cannot be the same as the last time
Clear controller point queue flag	1005	To discard the point queue that has been sent to the controller, the PLC will set it to 1, and the controller will set it to 0 after the queue has been cleared.

Data Stored At Points

One point data contains the values of one coordinate system and six axes (if it is a 4-axis robot, it contains the values of one coordinate system and four axes).

I-th point	Address code	Notes
Coordinate system	1010+20*(i-1)	1⩽i⩽32
Whether to use	1011+20*(i-1)	1⩽i⩽32; use: send 0; not use: send 1
The value of the j-th axis	1010+2+20*(i-1)+2*(j-1)	1⩽i⩽32, 1⩽j⩽9, the value of the axis uses float type, so it occupies two addresses

Examples

Need to send 88 points. Since only 32 points can be sent each time, they need to be divided into 3 transmissions, and the number of transmissions is 32, 32, and 24 respectively.

The process is as follows:

PLC sets 1003 to 32, sets the value of each address code used by the point to store data, sets 1001 to 1, and sets 1002 to 1;

The controller detects that 1002 is 1,1001 is 1, then takes out the data of the point storage address code according to the value of 1003, and then sets 1002 to 0;

PLC detects that 1002 is 0, sets 1003 to 32, sets the data of the point storage address code, and then sets 1002 to 1;

The controller detects that the value of 1002 is 1, and the value of 1001 is 1, takes out the data of the point storage address code according to the value of 1003, and then sets 1002 to 0;

PLC detects that the value of 1002 is 0, sets 1003 to 24, sets the data of the point storage address code, sets 1001 to 2, and sets 1002 to 1;

The controller determines that 1002 is 1 and 1001 is 2, takes out the data of the point storage address code according to the value of 1003, and then sets the value of 1002 to 0, and sets the value of 1001 to 0.

Instructions

MOVCOMM-External point

This instruction is used to move the points stored in the controller in accordance with the set interpolation method.

MOVCOMM

Parameter name	Parameter		Notes
Runin	Joint		
VJ	10	More	Speed range 1-100
PL	0	More	Smooth transition(0-5)
ACC	20	More	Motion ACC
DEC	20	More	Motion DEC
TIME	0	More	Early execution,N(ms)

Example: MOVCOMM MOVL VJ = 10% PL = 0 ACC = 10 DEC = 10

Interpolation method:

a. Joint

b. Linear

c. Curve

The interpolation method used during movement, all points are moved in this interpolation method.

VJ: The maximum speed during movement. Joint interpolation: 1-100; Other interpolation methods: 2-1000

PL: Position level, 0-5, 0 can be filled when the interpolation method is curve.

ACC: The maximum acceleration during movement.

DEC: The maximum deceleration during movement.

TIME: Early execution time; The next early executable instruction can be executed early.

Parameter source: Can be customized or bound to variables.

Welding Process

Catalogue

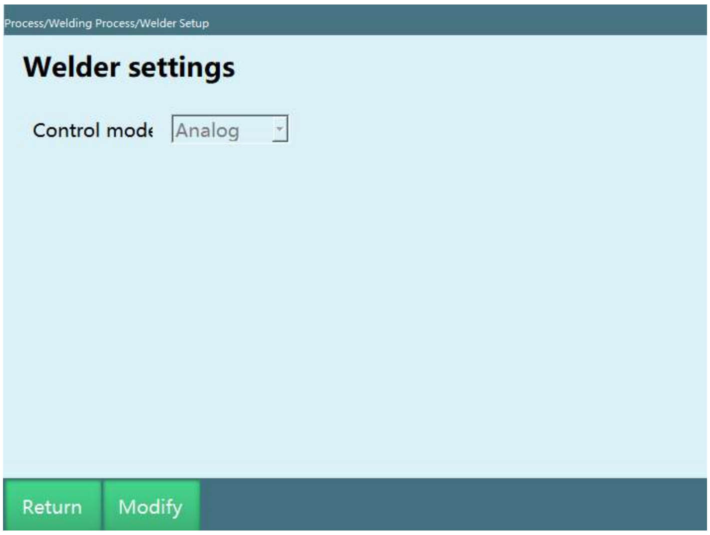
Welding Process

Welder setting

Enter "Process/Welding process/Welder setting" to modify the welder settings.

The steps are as follows:

1. Enter the "Process/Welding process/Welder setting" page

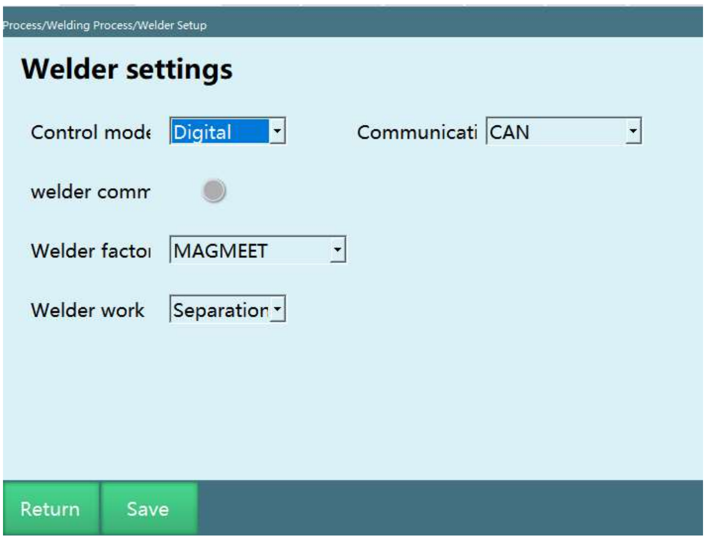


Two ways to control the welder:

Analog control: The full name is analog welder, which refers to the welder controlled by io analog quantity.

Digital control: Set according to the actual needs of the industrial site.

2. Click "Modify" to select the welder control method



Four communication modes of digital welder: CAN, ModBus RTU, EtherCAT, ModBus TCP

When selecting "**ModBus RTU**", you need to fill in the slave ID, port number, and baud rate;

When selecting "**ModBus TCP**", you need to fill in the IP and port number.

Welder communication status: Gray means communication failure, green means communication success.

Welder power supply manufacturers: General, MEGMEET, Shenwei Intelligent, Aotai, Meganice, Ruiling

When selecting "**Ruiling**", you need to select and fill in the parameters in [Material/Wire diameter/Gas].

Welder working mode: "Unified", "Separate".

3. Click "Save" to save successfully

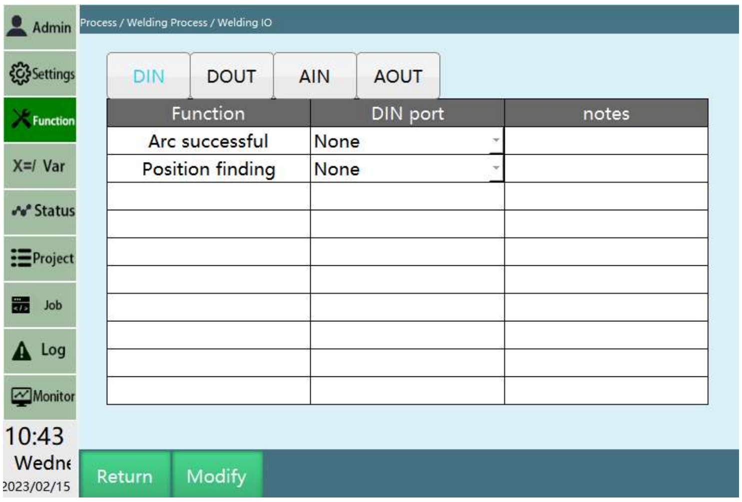
Welding IO setting

Enter "**Process/Welding process/Welding IO**" to modify the welding IO settings. The relevant steps are as follows:

Enter the "Process/Welding process/Welding IO" page.

After clicking "Modify", the "Modify" button becomes "Save", you can select the corresponding IO port after the respective function.

DIN



Ignition success signal: This signal is set to detect whether the ignition is successful. When the ARCON instruction is executed, an ignition signal is required. If the ignition signal is not given in the set welding detection time, an error will be reported (the welding ignition signal has timed out)

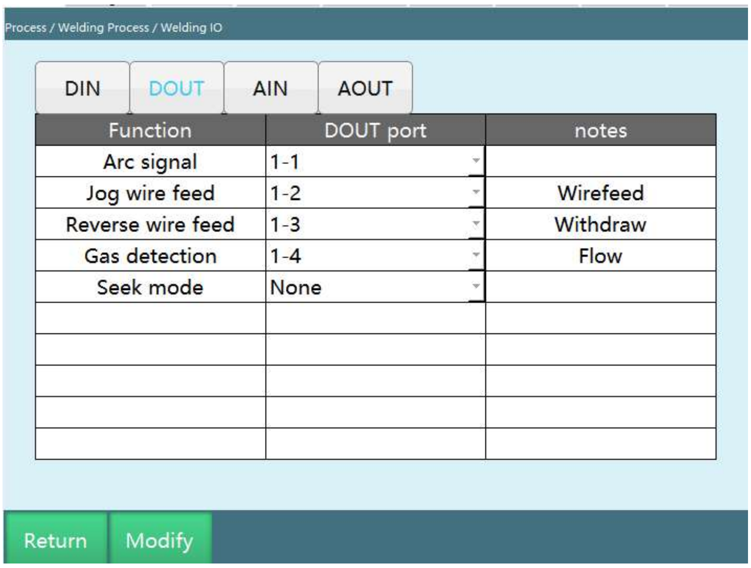
Search success signal: In the arc search, it is necessary to set the search success signal. (You can choose the port for the signal you need)

How to use: 1. In the arc search, find two single-core wires, one end of one wire connects the IO output end 1-5 (search mode signal), and the other end is connected to the iron plate

2. One end of the other wire connects to the IO input end 1-6 (search success signal), and the other end is connected to the end of the tool hand.

3. In the arc search, open the output port 1-5, when the end of the tool hand touches the iron plate, the set 1-6 input signal will change from low level to high level

DOUT



Ignition signal: When ready to ignite the arc, the system will send the output signal to the welder

Inching wire feeding signal: Welder wire feeding. When the corresponding signal port is opened, the welding monitoring window will display simultaneously as follows: Manual operation - Wire feeding switch on

Reverse wire feeding signal: The IO board gives the corresponding output signal when the welder retracts the wire

Gas detection signal: The IO board gives the corresponding output signal when the gas is supplied by the gas pump

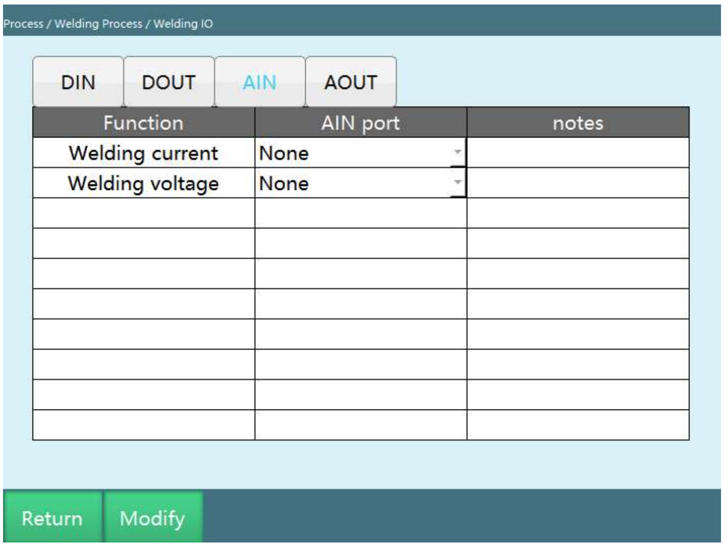
Search mode: It means that the welder enters the search mode, when the robot is moving, if the welding wire touches the workpiece, the welder will give a search success signal

How to use: 1. In the arc search, find two single-core wires, one end of one wire connects the IO output end 1-5 (search mode signal), and the other end is connected to the iron plate

2. One end of the other wire connects to the IO input end 1-6 (search success signal), and the other end is connected to the end of the tool hand.

3. In the arc search, open the output port 1-5, when the end of the tool hand touches the iron plate, the set 1-6 input signal will change from low level to high level

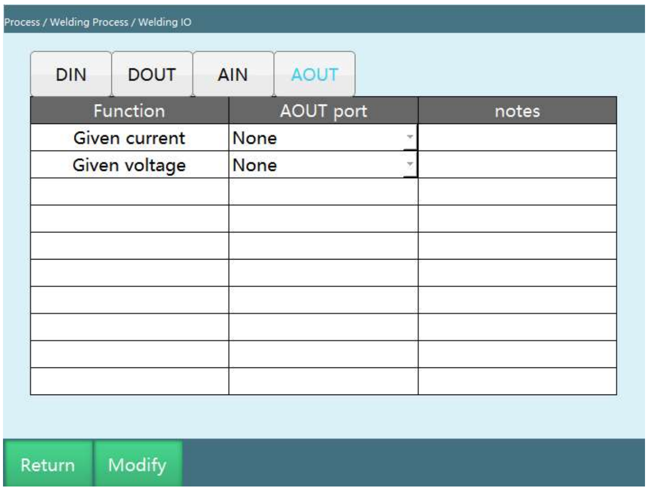
AIN



Welding current signal: input signal of analog welder current

Welding voltage signal: input signal of analog welder voltage

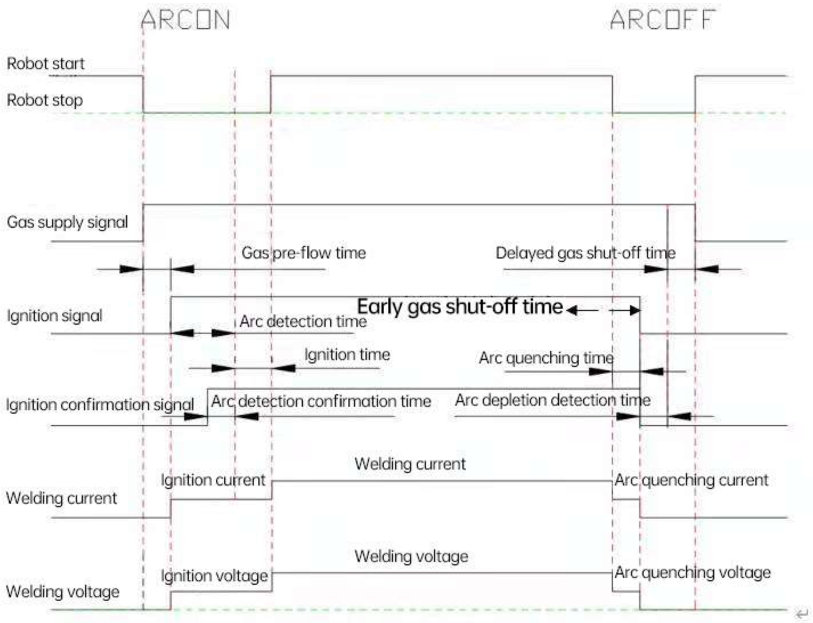
AOUT



Given current signal: the signal of a given current

Given voltage signal: the signal of a given voltage

Welding sequence diagram

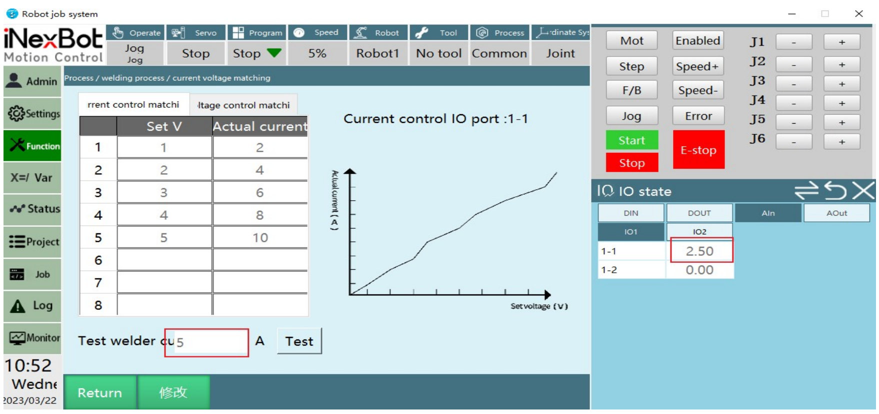


Current-voltage matching

Enter "Process/Welding process/Current-voltage matching" to modify the welding voltage and current. The relevant steps are as follows:

1. Enter the "Process/Welding process/Current-voltage matching" page. Note: This page will be hidden when you select digital welder.
2. At this time, no value can be entered in the current and voltage input box. After clicking "Modify", the "Modify" button becomes "Save", you can enter the value after the respective parameter.

The parameter setting steps of the current control matching interface are as follows:



Connect the controller to the welder, open the teach pendant interface as shown in the figure.

Set voltage: the value of the analog output in IO monitoring

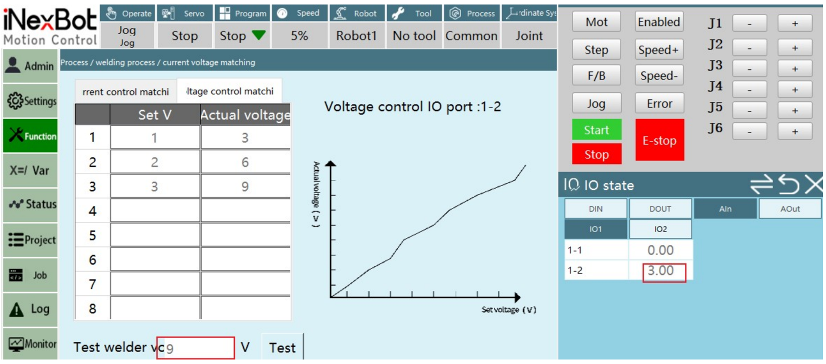
Actual welder current: the actual output current of the welder, which is displayed on the welder

Welder test current: fill in the values in the "Set voltage" column and the "Actual welder current" column, enter the value in the "Welder test current" box, click "Test", and a current value will be calculated.

As shown in the figure, a proportional coefficient 2 is calculated by the filled voltage and the actual current value of the welder. At this time, the "Welder test current" is 5A. After clicking the "Test", value 2.5 will be calculated through the proportional coefficient in the analog output port.

Note: The output upper limit of welder's current AOUT port is 10, if the test current is greater than 10, then perform the testing according to the upper limit; if the lower limit of welder's current AOUT port is less than 0, then perform the testing according to the lower limit

The parameter setting steps of the voltage control matching interface are as follows:



(Here is to modify the proportional relationship between the voltage and current sent by the controller to the welder and the actual voltage and current of the welder.)

Connect the controller to the welder, open the teach pendant interface as shown in the figure:

Set voltage: the value of the analog output in IO monitoring

Actual welder voltage: the actual output voltage of the welder, which is displayed on the welder

Welder test voltage: fill in the values in the "Set voltage" column and the "Actual welder voltage" column, enter the value in the "Welder test voltage" box, click "Test", and a voltage value will be calculated.

As shown in the figure, a proportional coefficient 3 is calculated by the filled voltage and the actual voltage value of the welder. At this time, the "Welder test voltage" is 9V. After clicking the "Test", value 3 will be calculated through the proportional coefficient in the analog output port.

The output upper limit of welder's voltage AOUT port is 10, if the test voltage is greater than 10, then perform the testing according to the upper limit; if the lower limit of welder's voltage AOUT port is less than 0, then perform the testing according to the lower limit

Operation steps of current-voltage matching when connecting welder

Current-voltage multi-stage matching: The current-voltage matching is divided into multiple stages, which can be any number of stages from 1 to 8

The operation steps are as follows:

1. Select "Current control matching"
2. Fill in 1 in the first line of "Set voltage", check the present current value on the welder, and fill in the value in the first line of "Actual welder current"
3. Fill in 3 in the second line of "Set voltage", check the present current value on the welder, and fill in the value in the second line of "Actual welder current"
4. Repeat the above operations until the 8 lines are set (if you only do 1 stage matching, just set the 1st and 2nd lines)
5. Fill in 220 for the welder test current, check whether the current of the welder is 220

Click "Save" and the modification is successful. This function parameter can be saved in 1 copy without process number.

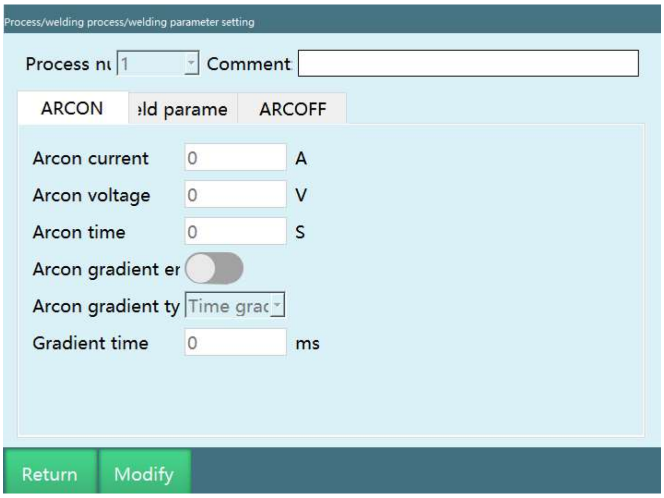
Welding parameter setting

Enter "**Process/Welding process/Welding parameter**" to modify the welding parameters settings. The relevant steps are as follows:

1. Enter the "Process/Welding process/Welding parameter" page.
2. Click "Modify", the "Modify" button becomes "Save", you can select the process number and modify the values of the ignition parameters, welding parameters and arc quenching parameters.

For example: ignition current=10, ignition voltage=8, welding current=15, welding voltage=20, arc quenching current=10, arc quenching voltage=15.

- Turn on "Ignition gradient enable" switch, select [Time gradient] for "Ignition gradient mode", set "Gradient time" to 1s.
- Turn on "Arc quenching gradient enable" switch, select [Time gradient] for "Arc quenching gradient mode", set "Gradient time" to 1s.
- Execution effect: The ignition current reaches 10A and the ignition voltage reaches 8V after the ignition signal is given, and then the current and voltage values gradually change from the the ignition current and voltage to the welding current (15A) and welding voltage (20V) within the set ignition gradient time (1s) for welding, and at the end of the welding, the current and voltage values gradually change from the the welding current and voltage to the arc quenching current and voltage within the set arc quenching gradient time (1s)



Process number: There are many choices of welding wire: carbon steel welding wire, low alloy structural steel welding wire, alloy structural steel welding wire, stainless steel welding wire and non-ferrous metal welding wire; ignition voltage, ignition current, ignition time, welding voltage, welding current, arc quenching voltage, arc quenching current and arc quenching time required by different welding wires are all different, so 1-99 different welding parameters can be set, you only need to call them later

Notes: You can add a note to this process number to indicate its function

Ignition current: the current applied from the time the wire is heated

Ignition voltage: the voltage applied from the time the wire is heated

Ignition time: The time to maintain the set ignition current and voltage values after the ignition signal is given.

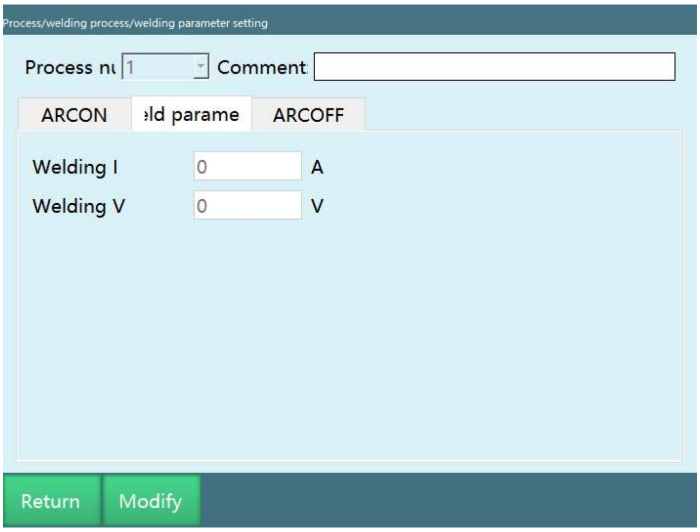
For example, ignition current = 20A, ignition voltage = 10V, and the waiting time is 1 second, which means that after reaching the ignition current and voltage values, it will maintain for one second before reaching the welding current and voltage values.

Ignition gradient enable: Control the time or distance for the gradual change from the ignition current and voltage to the welding current and voltage

Ignition gradient mode: time gradient

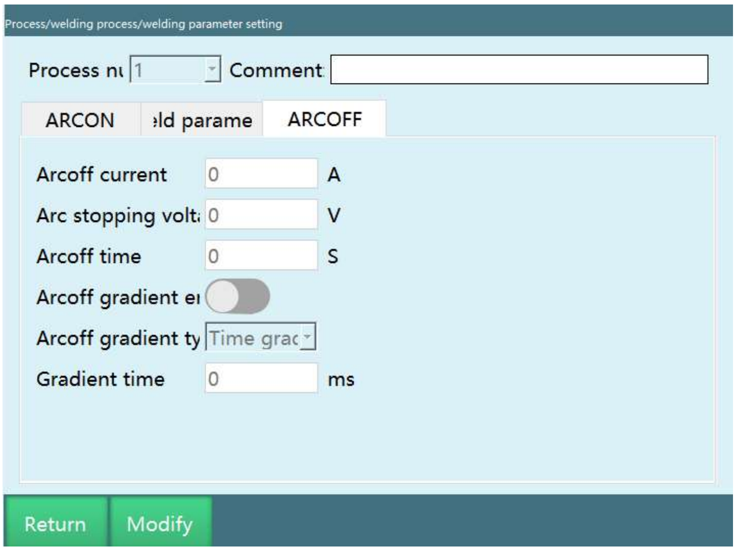
Gradient time: The time required for the gradual change from the ignition current and voltage to the welding current and voltage

For example, if the gradient time is set to 2s, then the current and voltage values will gradually change from the ignition current and voltage to the welding current and voltage within two seconds, instead of directly reaching the set welding current and voltage.



Welding current: The current applied during welding. When welding, the current flowing through the welding circuit is the result of the balance between the wire feed speed and the melting speed

Welding voltage: Welding voltage is the arc voltage, it can provide welding energy and ensure welding quality



Arc quenching current: The current given by the quencher when the arc needs to be quenched during welding

Arc quenching voltage: Refers to the highest power frequency voltage that is allowed to be applied to the arrester under the condition that the arrester can quench the arc when the power frequency freewheeling current crosses the zero value for the first time. The arc quenching voltage should be greater than the highest power frequency voltage that may appear on the working bus of the arrester, otherwise the arrester may explode due to the inability to quench the arc

Arc quenching time: The time for the robot to maintain welding with the arc quenching current and voltage after reaching the arc quenching point.

For example: the arc quenching time is 1s, which means that after the robot reaches the arc quenching point, it will maintain the welding for 1s with the arc quenching current and voltage, and then the welding ends. Different arc quenching mediums have different arc quenching time, generally in seconds.

Arc quenching gradient enable: Control the time for the gradual change from the welding current and voltage to the arc quenching current and voltage. Note: The following gradient parameters will take effect only after the gradient enable is turned on

Arc quenching gradient mode: time gradient

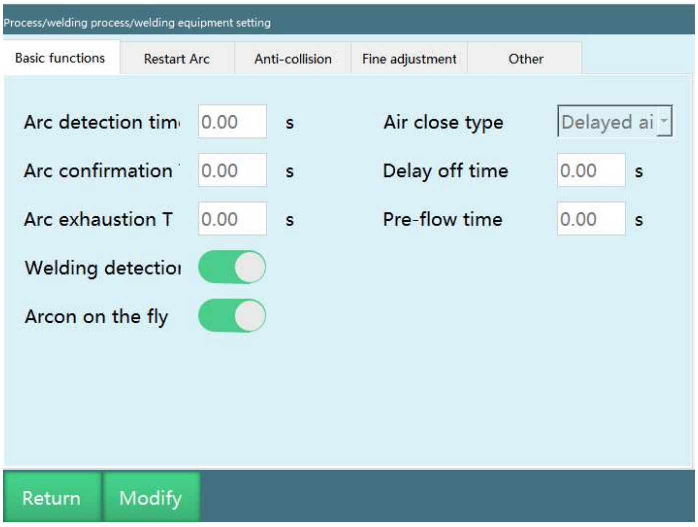
Gradient time: The time required for the gradual change from the welding current and voltage to the arc quenching current and voltage.

For example, if the gradient time is set to 2s, then the current and voltage values will gradually change from the welding current and voltage to the arc quenching current and voltage within two seconds, instead of directly changing from the welding current and voltage to the arc quenching current and voltage.

Welding equipment setting

Enter "**Process/Welding process/Welding equipment setting**" to modify the welding equipment settings. The relevant steps are as follows:

1. Enter the "Process/Welding process/Welding equipment setting" page
2. Click "Modify", the "Modify" button becomes "Save", click on the selection box below and select the functions you nee



Basic functions

Arc detection time: the time from the controller sending the ignition signal to the system receiving the ignition success signal from the welder! If the system does not receive ignition success signal within this time, the system will issue an ignition signal timeout error.

Arc detection confirmation time: In order to prevent disturbing signals due to obstacles such as dust, delay for a period of time to ensure that the arc has signal transmission, and only start welding after the successful ignition signal is continuously detected during this period.

Note: The arc detection time should be greater than the arc detection confirmation time

Arc depletion detection time: The time from the start of arc depletion to the actual end of arc depletion.

For example, if the arc depletion detection time is set to 2 seconds, it means that the time from the start of arc depletion to the actual end of arc depletion is 2 seconds. If the ignition signal is still on after the welding is completed, it will report arc depletion failure.

Delayed gas shut-off time: After the welding is finished and the arc quenching signal is sent, the welding wire has not cooled down, if the protective gas is stopped at this time, oxidation will still occur, so the gas needs to be shut off after a delay, and this operation also has the function of cooling the welding torch.

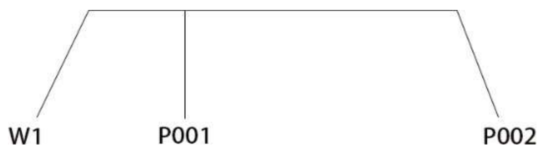
Set the "Delayed gas shut-off time" to 1s, and you can see that the set gas supply signal will be delayed by 1s before shutdown in the [Monitor] - [IO status - DOUT] interface after the welding is completed.

Early gas shut-off time: the time parameter to terminate the gas supply before arc quenching.

Set the "Early gas shut-off time" to 1s, and you can see that the set gas supply signal will be shut down 1s in advance in the [Monitor]-[IO status - DOUT] interface after the welding is completed.

Gas pre-flow: Start gas supply in advance when the robot moves from the safety point to the welding start point.

Gas pre-flow time: When welding, in order to prevent the wire from being oxidized by air, it may be necessary to supply air to blow off the air around the torch in advance to reduce the appearance of porosity in the welded seam and make the welded seam look flatter and smoother



W1 means safety point, P001 means welding start point, P002 means welding end point, P001-P002 means welding distance

Turn on "Gas pre-flow"

When the set air supply time is less than the time from the safety point to the welding start point

For example: Set the gas pre-flow time to 4s, and it takes 10s for the robot to move from W1 to the welding start point P001

Execution effect: It takes 10s for the robot to move from W1 to P001, the robot starts to supply air at 6s and reaches P001 at 10s, and starts ignition at the same time.

When the set air supply time is greater than the time from the safety point to the welding start point

For example: Set the gas pre-flow time to 4s, and it takes 2s for the robot to move from W1 to the welding start point P001

Execution effect: It takes 2s for the robot to move from W1 to P001, the robot will stay at P001 for 2s, and the ignition will start only after 4s.

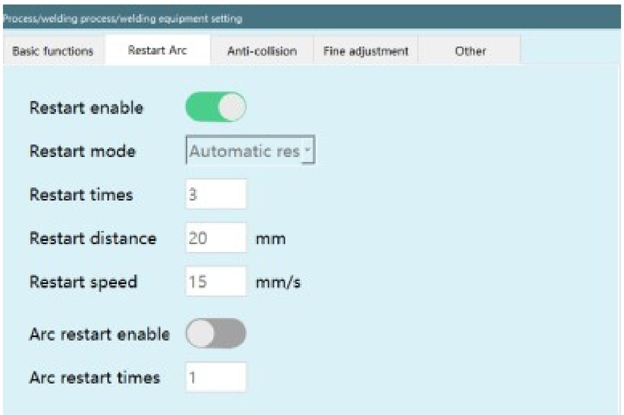
Turn off "Gas pre-flow"

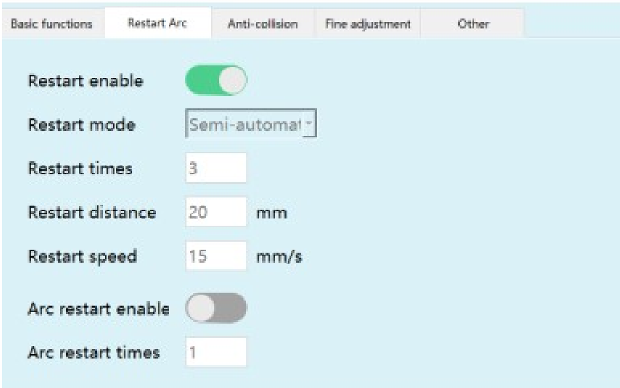
If "Gas pre-flow" is not turned on: start the gas supply after moving from safety point to the welding start point

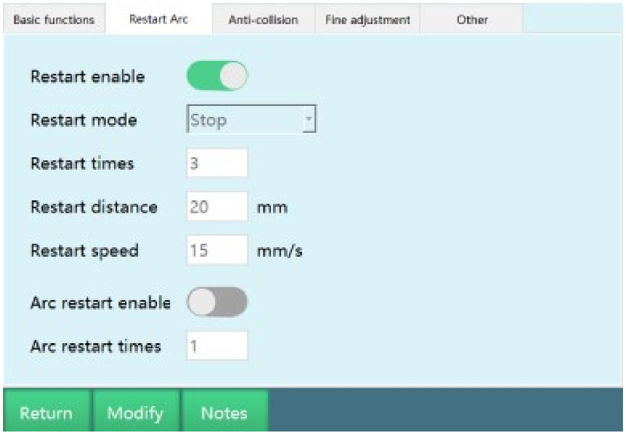
For example: turn off "Gas pre-flow", set gas pre-flow time to 4s

Execution effect: start the gas supply after the robot moves from point W1 (safety point) to welding start point P001, and the robot will start the ignition only after 4s.

Reignition/Restart







Restart enable: Restart enable, valid only when the ignition signal is given after the arc is broken.

Automatic restart: After detecting the occurrence of arc break, servo and program are running, within the set arc detection time, the ignition signal will be given again, and the program will continue to run

Semi-automatic restart: After detecting the occurrence of arc break, the servo is running and the program is paused. At this time, you need to manually click the "Start" button to give the ignition signal again within the set arc detection time, and the program will continue to run.

Stop: After detecting the occurrence of arc break, the servo is in the ready state and the program is in the stop state. After an arc break occurs, you need to clear the error and then manually click the "Start" button.

Restart distance: The back-off distance of the restart action. During the welding process, when the breakpoint is run again, it is possible to go back a distance (to prevent empty welding).

Restart speed: The back-off speed of the restart action. The welder will not back off when the speed is 0.

Reignition enable: First send a signal to let the welder start the ignition, if the ignition fails, then execute the ignition action again in place; if the ignition is successful, execute the welding action normally; if the ignition is not successful within the set number of times, it will stop and report an error.

Reignition times: The maximum number of times to perform reignition actions within the current welding start and end interval, beyond which no restart will be performed

For example, if you set the reignition times to 2, then after an arc break occurs, if the ignition is not successfully started after giving the ignition signal 2 times, the controller will report an error

<table>
<tr><td colspan="2">Restart function (Note: To use this function, you need to turn on the "Welding interruption detection" in the "Basic functions")</td></tr>
<tr><td colspan="2">welding track straight line P001-P002

welding start point P001, welding end point P002,</td></tr>
<tr><td>Automatic restart

Restart distance 20mm

Restart speed 15mm/s</td><td>Execution effect:

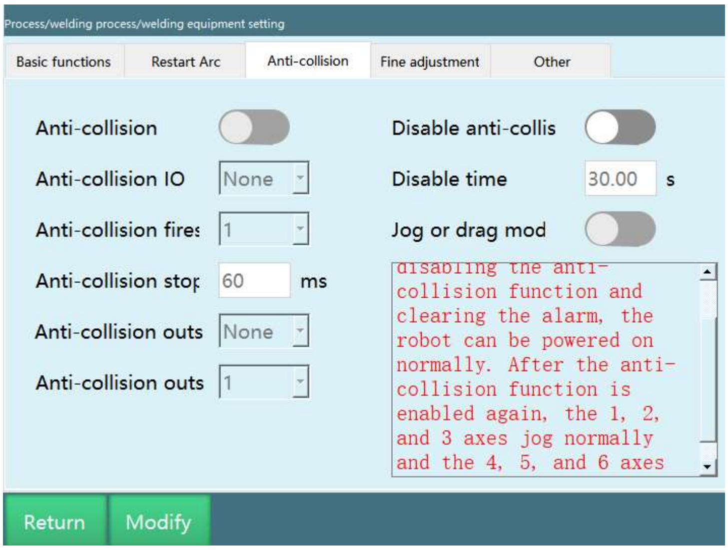
After the welding starts, the robot moves from P001 to P002. After an arc break occurs, the controller will issue a warning (arc break detected in welding), and the servo and the program are both in the running state at this time.

When the arc is broken during the</td></tr>
</table>

	movement from P001 to P002, the robot will move 20mm at the speed of 15mm/s at the arc break point according to the restart distance and restart speed parameters you set, and give the ignition signal again after reaching the retraction distance, the robot performs the welding operation again
Semi-automatic restart Restart distance 20mm Restart speed 15mm/s	Execution effect: After the welding starts, the robot moves from P001 to P002. After an arc break occurs, the controller will issue a warning (arc break is detected in welding) . At this time, the servo is running, the program is in the pause state, and there will be a pop-up window prompting that the arc breaks Arc break occurs during the movement from P001 to P002, click the "Confirm" button in the pop-up prompt box, and then click the "Start" button, the robot will move 20mm at the speed of 15mm/s at the arc break point according to the restart distance and restart speed parameters you set, and give the ignition signal again after reaching the retraction distance, the robot performs the welding operation again
Stop Restart distance 20mm Restart speed 15mm/s	Execution effect: After the welding starts, the robot moves from P001 to P002. When the arc breaks, the controller reports an error (arc break is detected in welding). At this time, the servo is ready, the program is in the stop state, and there will be a pop-up window prompting that the arc breaks. An arc break occurs during the

	movement from P001 to P002. After an error is reported, click the "Clear" button first, then click the "Confirm" button in the pop-up window prompt box, and then click the "Start" button, and there will be a pop-up window prompt again (breakpoint execution, first line execution) 1. The effect when you select the "Breakpoint execution": the robot will move 20mm at the speed of 15mm/s at the arc break point according to the restart distance and restart speed parameters you set, and give the ignition signal again after reaching the retraction distance, the robot performs the welding operation again 2. The effect when you select the "First line execution": the robot will perform the welding operation from the beginning
Restart times	The number of times the ignition signal can be given when the arc is broken Execution effect: set the restart times to 3, then the ignition signal can be given up to three times after an arc break occurs, and the controller will report an error when the ignition signal is given for the fourth time (arc break is detected in welding)

Anti-collision



Anti-collision enable: Turn on the enable switch to detect the anti-collision signal.

Anti-collision IO: IO input signal when a collision occurs.

Anti-collision trigger level: 1/0 corresponds to high level/low level.

Anti-collision quick stop time: The time required for the robot to stop after anti-collision is triggered.

If the set anti-collision quick stop time is 60ms, then the time from work to stop of the robot after a collision is 60ms

Anti-collision status output port: The specified value output port outputs a signal when anti-collision is triggered.

If the anti-collision status output level is 1, the IO output port is set to port 1-2, when a collision occurs, the output port 1-2 will change from low level 0 to high level 1.

If the anti-collision status output level is 0, the IO output port is set to port 1-2, when a collision occurs, the output port 1-2 will change from high level 1 to low level 0.

Anti-collision status output level: 1/0 corresponds to high level/low level.

Shield anti-collision enable:

When the welding torch collision occurs, the controller reports an error (torch anti-collision is triggered), you can not clear the error at this time, you need to turn on the shielding anti-collision enable, set the shielding time, the anti-collision signal will not be detected within the shielding time, if the anti-collision signal is released, the "Shield anti-collision enable" will be turned off immediately.

Shielding time: the time parameter for shielding anti-collision.

Turn on "Shield anti-collision enable", set the shielding time to 10s, when a collision occurs,

it will be shielded for 10s in order to to move the torch to a safe position.

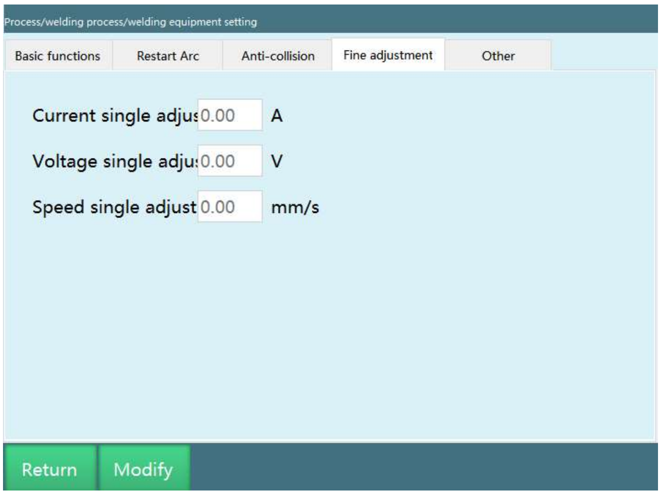
After reaching the shielding time, the controller reports an error (shielding has ended and the torch anti-collision is triggered)

Jogging or dragging mode:

Turn on "Shield anti-collision enable" after a collision occurs and set the shielding time

Turn on the jogging or dragging mode enable, then you can drag 4, 5, 6 axis after the collision (at this time, 4, 5, 6 axis can only drag, 1, 2, 3 axis can jog)

Fine adjustment



Welding current single adjustment amount: single adjustment range of welding current during welding

For example: Welding current single adjustment amount is 5A, if you want to increase or decrease the current value during the welding process, you can click on the process bar - [Welding process] - [Fine adjustment]

Click "**Increase given value**", the current value will increase 5A during the welding process; click "**Decrease given value**", the current value will decrease 5A during the welding process

Note: The increased or decreased value is adjusted according to the welding current single adjustment amount

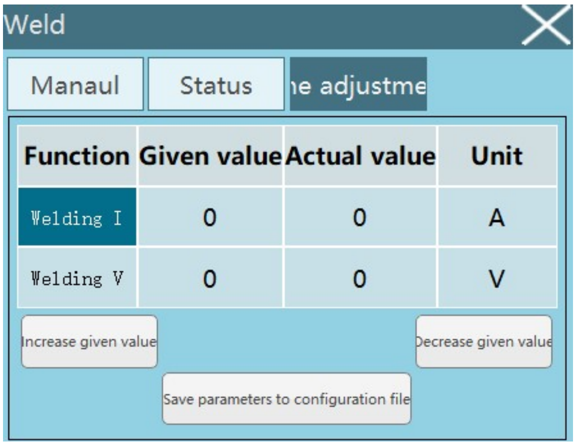
Welding voltage single adjustment amount: single adjustment range of welding voltage during welding

For example: Welding voltage single adjustment amount is 6V, if you want to increase or decrease the voltage value during the welding process, you can click on the process bar - [Welding process] - [Fine adjustment]

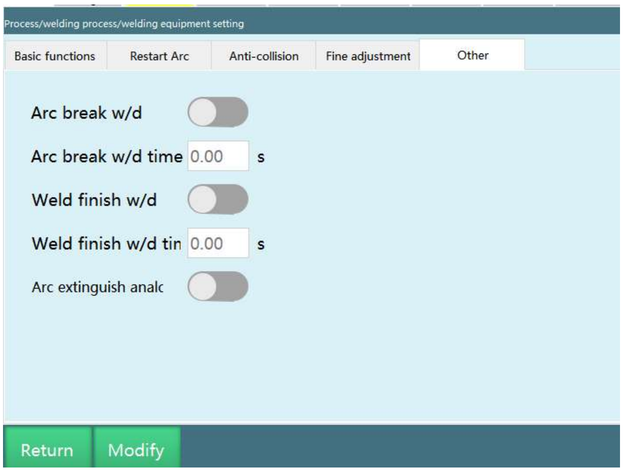
Click "**Increase given value**", the voltage value will increase 6V during the welding process; click "**Decrease given value**", the voltage value will decrease 6V during the welding process

Note: The increased or decreased value is adjusted according to the set welding voltage single adjustment amount

Welding speed single adjustment amount: The increment and subtraction range of a single micro-adjustment can be set in the welding equipment parameters



Other



Retraction after welding enable: When the welding is over, the welding torch will receive a signal, and the welding wire will be retracted to prevent collision with the workpiece when going to the next welding point.

Retraction time after welding: the time for retracting the welding wire after completion of welding.

Turn on "Retraction after welding enable" and set the "Retraction time after welding" to 3 seconds, then at the end of welding, it will take 3 seconds in total from receiving the wire retraction signal to the end of the wire retraction

Arc-break retraction enable: If the welding current exceeds the rated load rate of the welder, the welder will have a short-term protection, the arc will be broken, the welding wire will be retracted to prevent adhesion to the workpiece.

Arc break retraction time: the time for retracting the welding wire after the welding arc is broken.

Turn on "Arc break retraction enable" and set the "Arc break retraction time" to 2 seconds, then in order to prevent the welding wire from sticking to the workpiece, the welding wire retraction time needs 2 seconds.

Arc quenching analog zero-setting function: The analog voltage and current signals are reset to zero (analog output) at the end of welding.

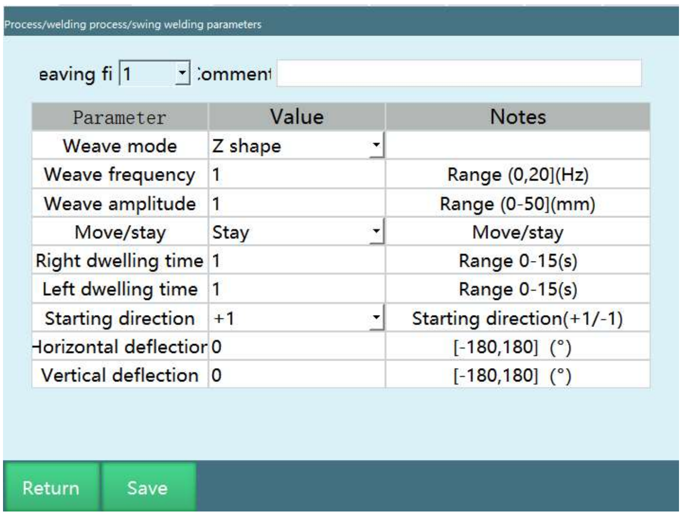
Weaving parameters

Enter "**Process/Welding process/Weaving parameters**" to modify the weaving parameters. The relevant steps are as follows:

1. Enter "Process/Welding process/Weaving parameters" page. The weaving file has 9 process numbers to choose from. Select the weaving welding

parameters to be modified and click on the "Modify" button at the bottom, all input boxes become available for input.

2. Click the "Save" button to finish saving after the input is completed.



Parameter	Value	Notes
Weave mode	Z shape	
Weave frequency	1	Range (0,20](Hz)
Weave amplitude	1	Range (0-50](mm)
Move/stay	Stay	Move/stay
Right dwelling time	1	Range 0-15(s)
Left dwelling time	1	Range 0-15(s)
Starting direction	+1	Starting direction(+1/-1)
Horizontal deflectior	0	[-180,180] (°)
Vertical deflection	0	[-180,180] (°)

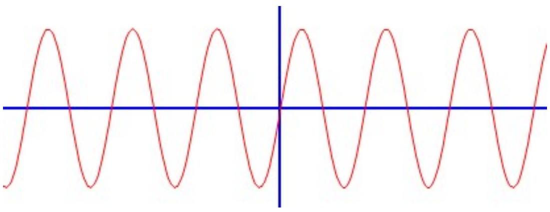
iNexBot supports four swing modes: sine pendulum, Z-shaped pendulum, circular pendulum, and external axis fixed-point pendulum.

Swing frequency, swing amplitude, starting direction, horizontal declination, vertical declination and other parameters are adjustable, which can be set according to the actual needs of the industrial site.

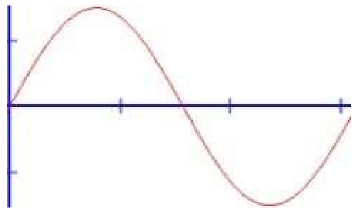
Swing amplitude: the greater the amplitude, the greater the robot swings;

Swing frequency: the greater the frequency, the faster the robot swings.

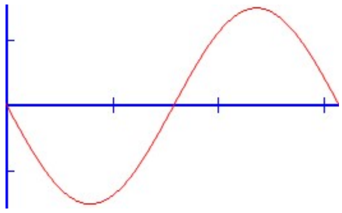
Starting direction: +1, start from a certain point and go up first; -1, start from a certain point and go down first



Original figure

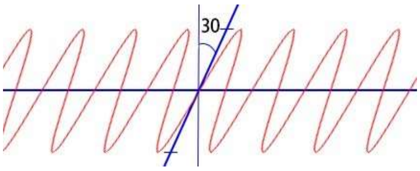


+1 figure



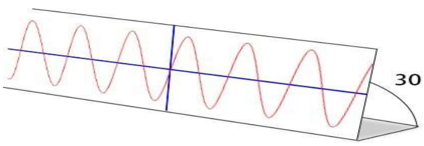
-1 figure

Horizontal declination: The figure shows the weaving trajectory with a horizontal deflection angle of 30 degrees



Horizontal declination figure

Vertical declination: The figure shows the weaving trajectory with a vertical deflection angle of 30 degrees



Vertical declination figure

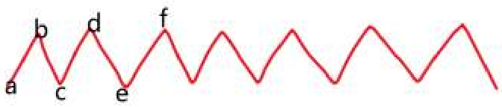
Move: The robot moves forward for the set time every time it swings, and then enters the next swing;

Stay: The robot stays for the set time every time it swings

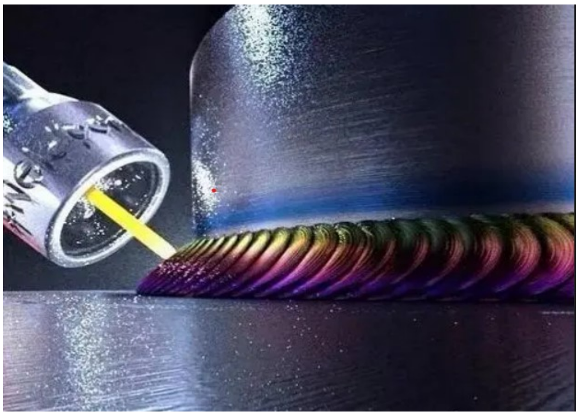
Left stay time/Right stay time

When the weaving welding method is Z-shaped pendulum and external axis fixed-point pendulum, there will be left stay time/right stay time; here it refers to the time to stay at a target point during Z-shaped pendulum and fixed-point weaving welding (as shown below)

The red trajectory indicates the Z-shaped weaving welding trajectory. If the left and right stay time is set to 1 second, the robot will stay at point a for 1 second and then run to point b, and stay at point b for 1 second and then run to point c, and follow this operating logic to complete the entire weaving track.



Weaving welding is a welding operation in which the welding seam heat source performs regular lateral swinging on the weldment during welding. The weaving welding effect figure is shown below.

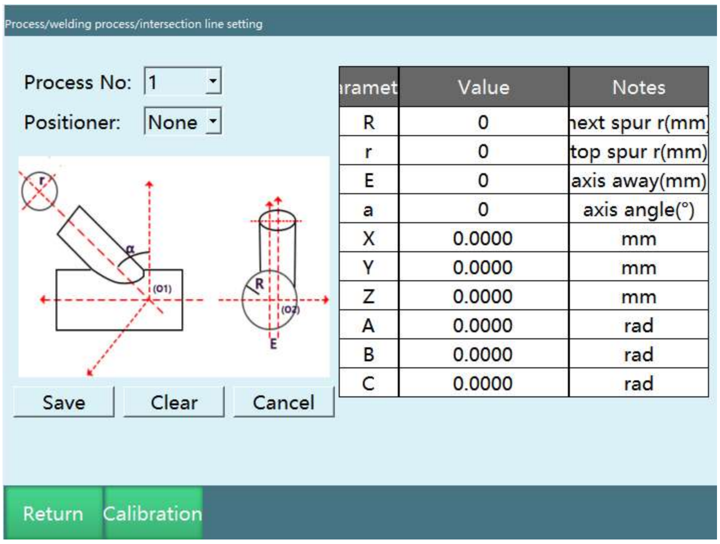


Intersecting line setting

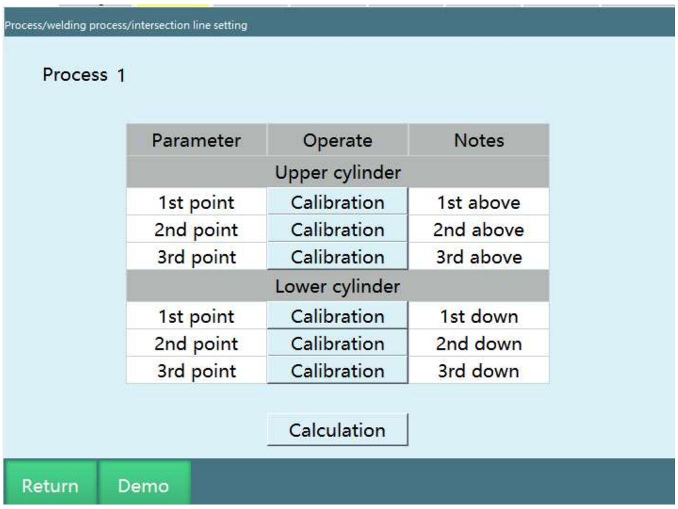
Enter "**Process/Welding process/Intersecting line setting**" to modify the intersecting line settings.

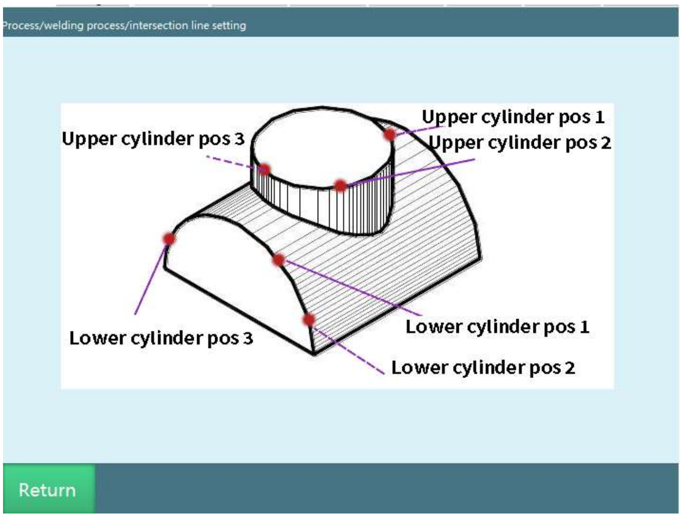
The relevant steps are as follows:

1. Enter "Process/Welding process/Intersecting line setting" page



2. Calibrating before use can reduce the error. Click "Calibrate" to enter the calibration interface, if you do not know how to calibrate, there is a "Demo" button in the interface, you can check it, as shown in the figure

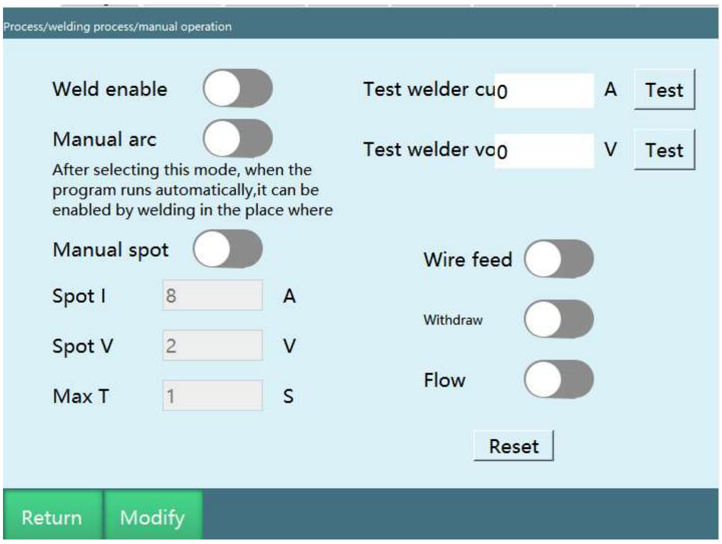




Manual operation

Enter "**Process/Welding process/Manual operation**" to modify the manual operation settings. The relevant steps are as follows:

1. Enter "Process/Welding process/Manual operation" page.



Welding enable: When the "Welding enable" is turned on, the robot will perform the welding function, otherwise it will just walk the track.

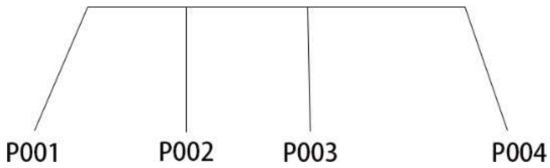
• After the program of the welding trajectory is written, you can first confirm that the running trajectory is correct in the teach mode

• Switch to the running mode and turn on "Welding enable", and the robot will perform the welding function. When the program is in the running mode, press "Stop", and then press "Start", after the program restarts, the welding function will no longer be performed.

Manual ignition mode: If you select this mode, during the automatic operation of the program, you can manually control the ignition or quenching of arc through the welding enable switch

•When the robot moves from the welding start point P001 to the welding end point P004, if the "Welding enable" is turned on, the robot will start the ignition; if the "Welding enable" is turned off, the robot will quench the arc;

•For example, in the running mode, during the movement of the robot, the "Welding enable" is turned on at P002 and turned off at P003, then during the movement from P002 to P003, the robot keeps the ignition state, and when moving from P003 to P004, the robot keeps the quenching state;



•When the robot moves from W1 (safety point) to the welding start point P001, if the "Welding enable" is turned on, the robot will not start ignition; the robot will only start the ignition when it reaches P001;

"Manual ignition mode" is not turned on: When the robot moves from the welding start point P001 to the welding end point P004, the "Welding enable" button is invalid (the robot will not start ignition even if the "Welding enable" button is turned on)

Manual spot welding: Set spot welding current, spot welding voltage and maximum time, click "Save"

Long press the "Manual spot welding" button (valid when pressed and hold, invalid when released), the robot will perform welding, if you release the button, the robot will stop welding

Spot welding current: spot welding output current

Spot welding voltage: spot welding output voltage

Maximum time: The maximum time the "Manual spot welding" button is allowed to be held down.

•For example, the maximum time is set to 5s, then if you press and hold the "Manual spot welding", the robot will weld for 5s. If the time exceeds 5s, even if you press the "Manual spot welding" button, the robot will not weld

Fault reset: Valid when using digital welder. This function can be used to reset the fault of the welder

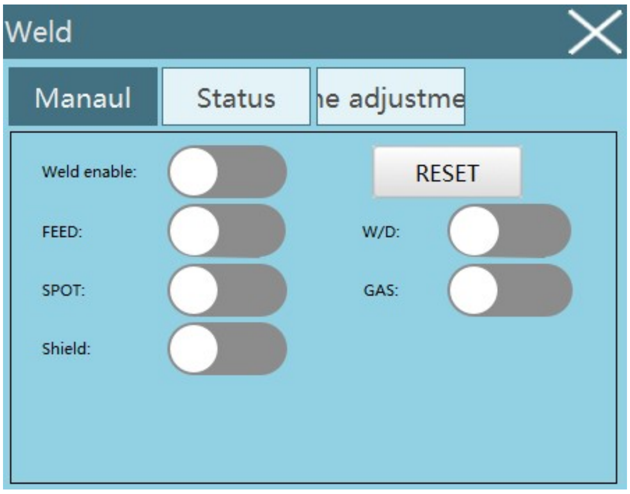
Wire feeding: Feed wire at the start of welding

Wire retraction: Retract the wire after the welding is completed

Gas supply: Turn on to supply gas

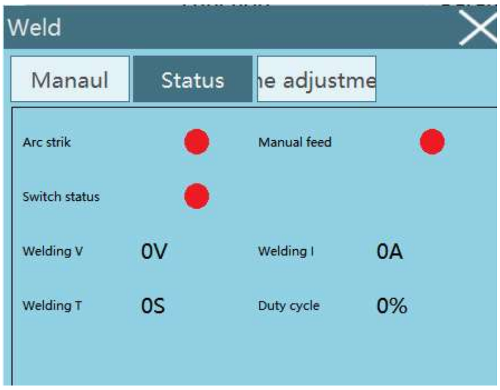
For the convenience of welding, we now add [Process]/[Welding] to the status bar.

Note: You need to select [Welding process] in "Settings/Operation parameters - Process selection", modify and save. The status bar displays [Welding]. Click [Welding], a manual operation window of welding will pop up



[Manual Operation] has the same effect as the manual operation in "Process/Welding process/Manual operation" page. In this status bar, it is more convenient to see the signal during the welding process, the change of current and voltage values and other effects.

Shield anti-collision: After triggering the anti-collision, turn on the "Shield anti-collision" switch, and shield the anti-collision according to the parameters in "Welding equipment setting - Anti-collision". After turning on the "Shield anti-collision" switch, the anti-collision signal will be released within the shielding time, which helps to move the welding torch to a safe position

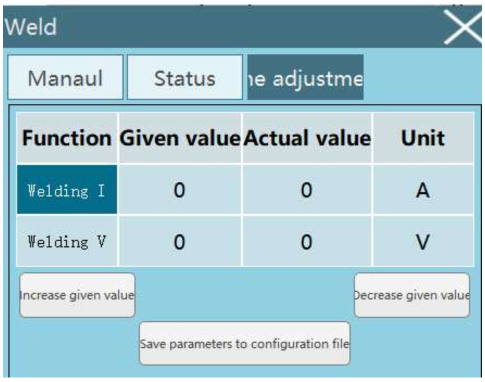


[Status]

Ignition success/Manual wire feeding/Torch switch status: green is on, red is off.

Welding current/voltage: Input current and voltage.

Welding time: The time from the start of welding to the end of welding. Record the welding time after start-up.



[Fine adjustment]

Save parameters to configuration file: Save the parameters during welding, click the "Save" button, and the parameters will be automatically overwritten into the instructions or parameters;

• When the instruction uses custom parameters, the parameters will be saved to instructions;

• When using the welding process number parameter, the parameters will be saved to the welding parameters;

Increase given value/Decrease given value: Select the parameter to be adjusted, click "Increase given value" and "Decrease given value", the adjustment will take effect immediately

• For example: In the "Welding process-Welding equipment setting-fine adjustment" interface, set the welding current single adjustment amount to 5A, if you want to increase or decrease the current value during the welding process, you can click [Increase given value], [Decrease given value]

• Click [Increase given value], the current value will increase 5A during the welding process, click [Decrease given value], the current value will decrease 5A during the welding process

Note: The increased or decreased value is adjusted according to the set welding current single adjustment amount

• For example: In the "Welding process-Welding equipment setting-fine adjustment" interface, set the welding voltage single adjustment amount to 6V, if you want to increase or decrease the voltage value during the welding process, you can click [Increase given value], [Decrease given value]

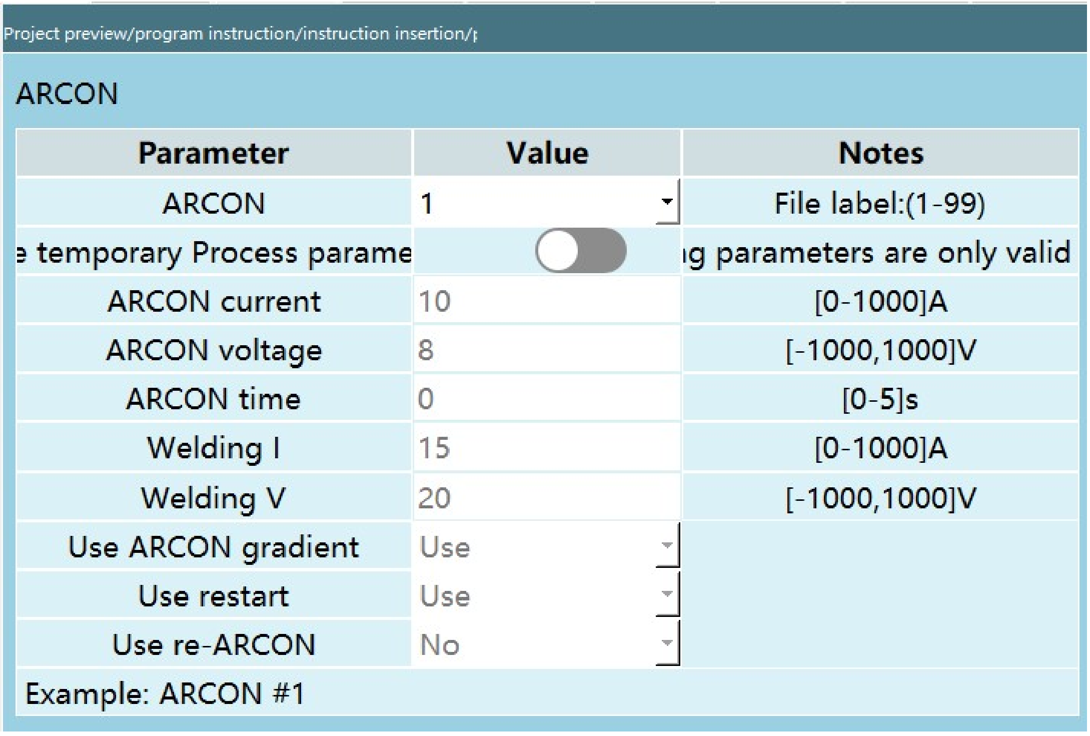
• Click [Increase given value], the voltage value will increase 6V during the welding process, click [Decrease given value], the voltage value will decrease 6V during the welding process

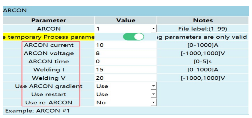
Note: The increased or decreased value is adjusted according to the set welding current single adjustment amount

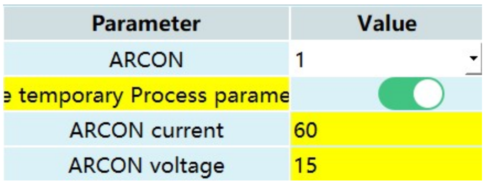
Welding instructions description

ARCON instruction - Welding start

ARCON instruction - Welding start
Function: This instruction can perform ignition operation
Parameter interface

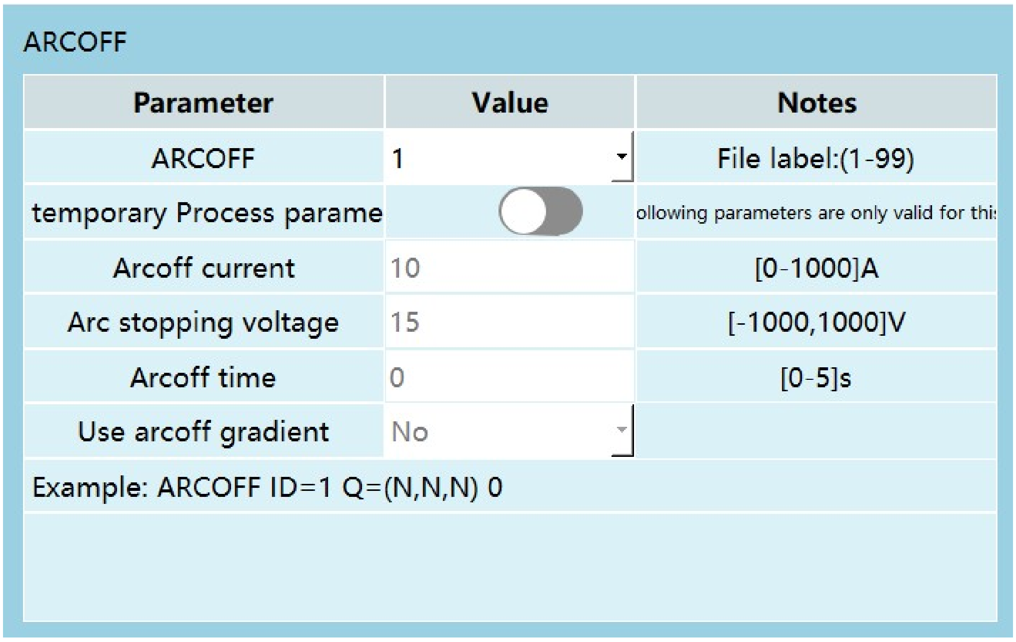


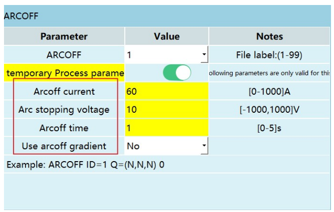
Parameter	Value	Note
ARCON	1-99	Welding process supports 99 file numbers
Use temporary process parameters	Turn on the "Use temporary process parameters" switch, the current and voltage values at the beginning of welding will depend on the temporary process parameters you set For example: the ignition current set in the "Welding process-Welding parameter" interface is 50A, if the "Use temporary process parameters" switch is turned on in the ARCON instruction parameter interface, and the set ignition current is 60A, then the ignition current when performing welding operation is 60A Turn on the "Use temporary process	Turn on the "Use temporary process parameters" switch, the parameters below will take effect, as shown in the figure below  Note: The parameters below are only valid for this instruction

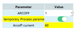
	parameters" switch, and the modified parameter values will be displayed with a yellow fill color, as shown below  If you do not turn on the "Use temporary process parameters" switch, the current and voltage values during welding operation are the current and voltage values set in the "Welding process-Welding parameter" interface	

ARCOFF instruction - welding end

ARCOFF instruction - welding end
Function: Execute the arc quenching operation, select the process number corresponding to the ARCON
Parameter interface



Parameter	Value	Note
ARCOFF	1-99	Welding process supports 99 file numbers
Use temporary process parameters	Turn on the "Use temporary process parameters" switch, the current and voltage values at the beginning of welding will depend on the temporary process parameters you set For example: the quenching current set in the "Welding process-Welding parameter" interface is 50A, if the "Use temporary process parameters" switch is turned on in the	Turn on the "Use temporary process parameters" switch, the parameters below will take effect, as shown in the figure below  Note: The parameters below are only valid for this instruction

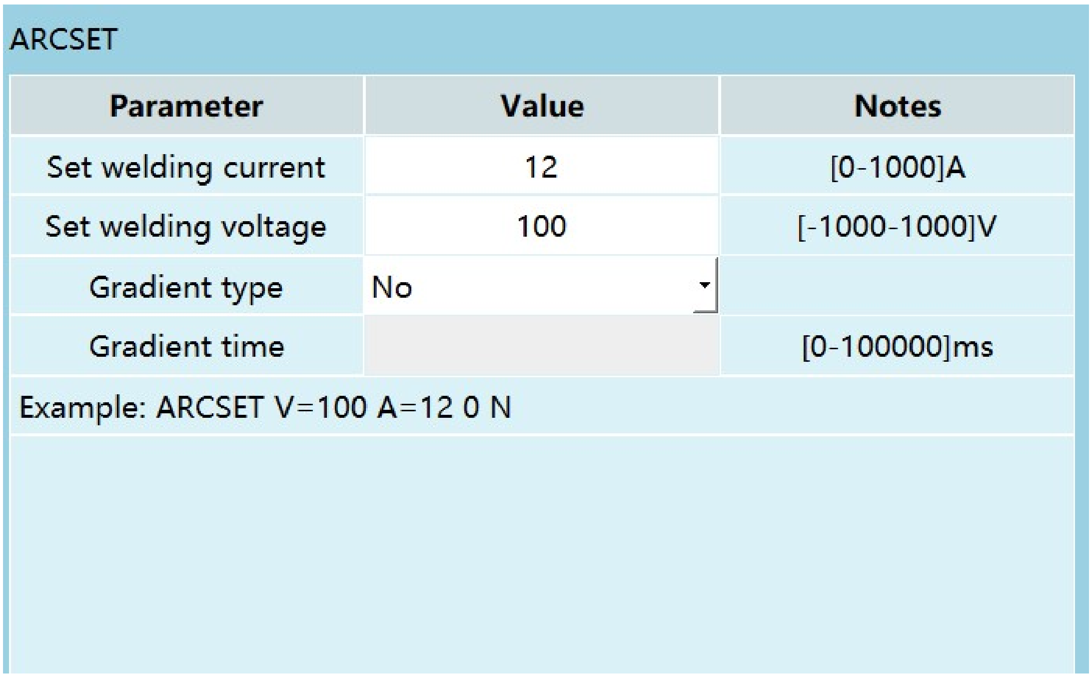
	ARCOFF instruction parameter interface, and the set quenching current is 60A, then the quenching current when performing welding operation is 60A Turn on the "Use temporary process parameters" switch, and the modified parameter values will be displayed with a yellow fill color, as shown below  If you do not turn on the "Use temporary process parameters" switch, the current and voltage values during welding operation are the current and voltage values set in the "Welding process-Welding parameter" interface	

ARCSET instruction - welding setting

ARCSET instruction - welding setting
Function: This instruction can set the current and voltage during welding For example: set welding current 50A, voltage 15V in "Welding process-Welding parameter" interface Set welding current 45A, voltage 20V in the ARCSET instruction parameter interface

If the ARCSET instruction is inserted after the ARCON instruction, the current and voltage values during welding are the parameter values filled in the ARCSET instruction interface

Parameter interface:



ARCSET

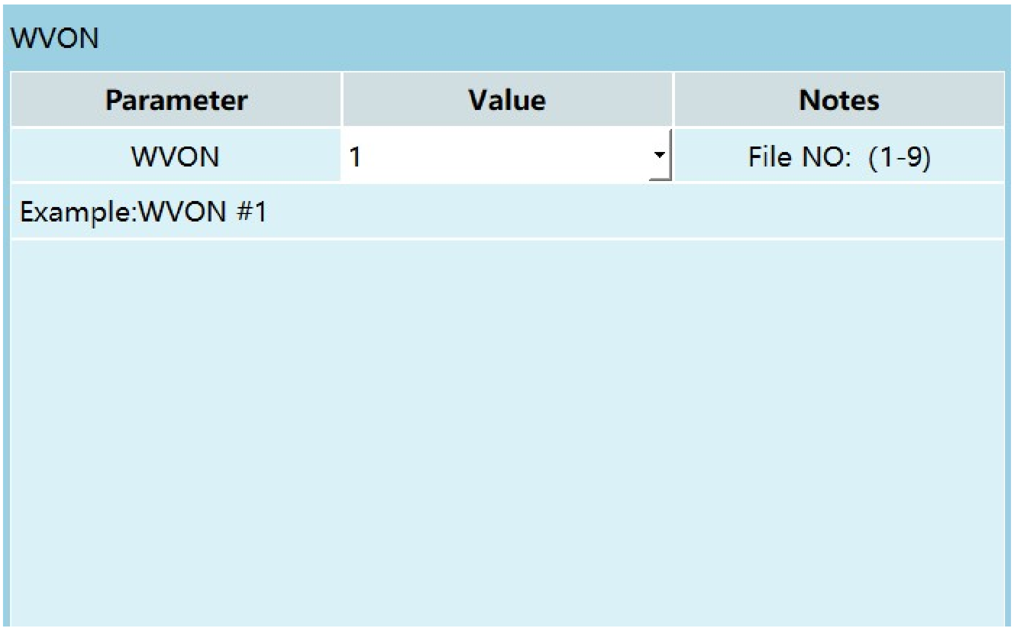
Parameter	Value	Notes
Set welding current	12	[0-1000]A
Set welding voltage	100	[-1000-1000]V
Gradient type	No	
Gradient time		[0-100000]ms

Example: ARCSET V=100 A=12 0 N

Parameter	Value	Parameter range
Set welding current	Fill in the welding current value during welding operation	[0-1000]A
Set welding voltage	Fill in the welding voltage value during welding operation	[-1000-1000]V
Gradient method	Time gradient No	
Gradient time	The time from the ignition current and voltage to the welding current and voltage For example: the set	[0-100000]ms

	gradient time is 1s, then the time from the ignition current and voltage to the welding current and voltage is 1s If you not use gradient method, the ignition current and voltage will reach the welding current and voltage immediately	

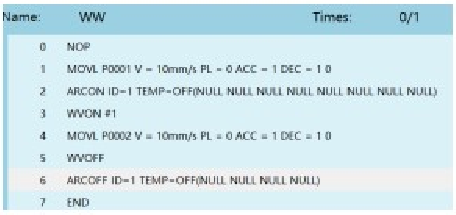
WVON instruction-weaving start

WVON instruction-weaving start
Function: Execute this instruction to start weaving welding, please run the welding start ARCON instruction before executing this instruction.
Parameter interface 

WVON

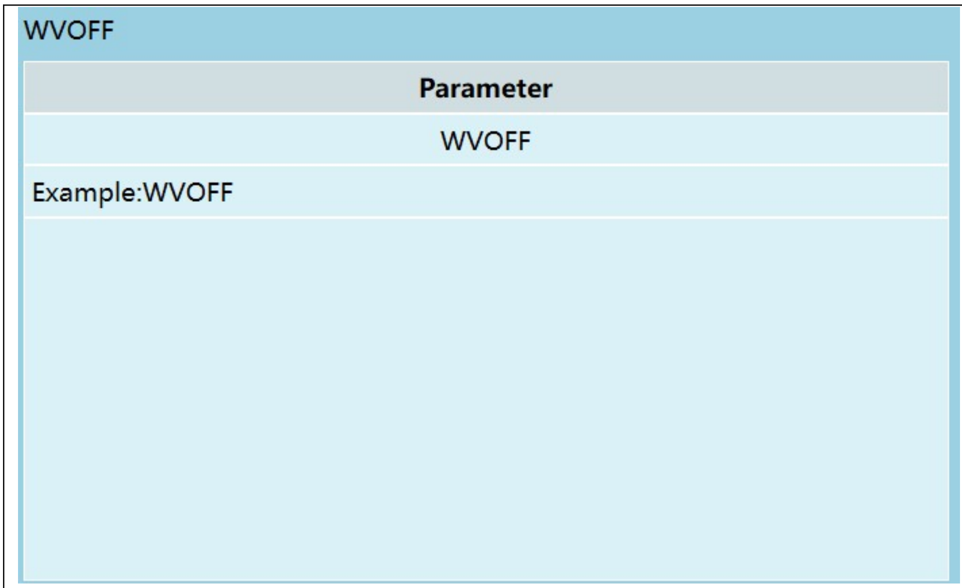
Parameter	Value	Notes
WVON	1	File NO: (1-9)

Example:WVON #1

Parameter	Value	Note
WVON	1-9	Different file numbers can be selected during weaving welding, as shown below:  Figure 1 For example: in the "Welding process-Weaving parameter" interface, the selected weaving file is 1, and the weaving mode is sine weaving, as shown in Figure 1, select 1 as the weaving start process number, then Sine weaving will be performed while welding.  For example: in the "Welding process - Weaving parameter" interface, the selected weaving file is 2, and the weaving mode is Z-shaped, as shown in Figure 2, select 2 as the weaving start process number, then Z-shaped weaving will be performed during welding.

WVOFF instruction - weaving welding end

WVOFF instruction - weaving welding end
Function: Execute this instruction to end weaving welding.
Parameter interface



How to use: Execute the WVOFF instruction as shown in Figure 1, and the robot will end weaving welding

Note: It is necessary to insert the ARCON instruction before the WVON instruction, and insert the ARCOFF instruction after the WVOFF instruction

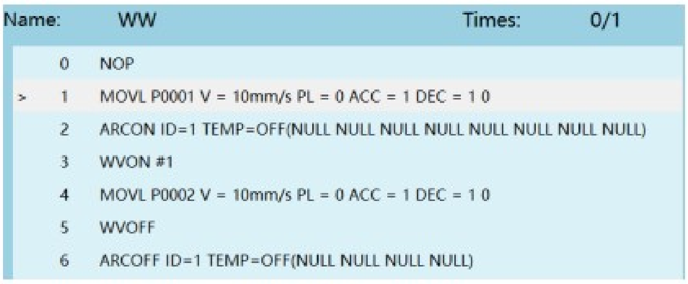
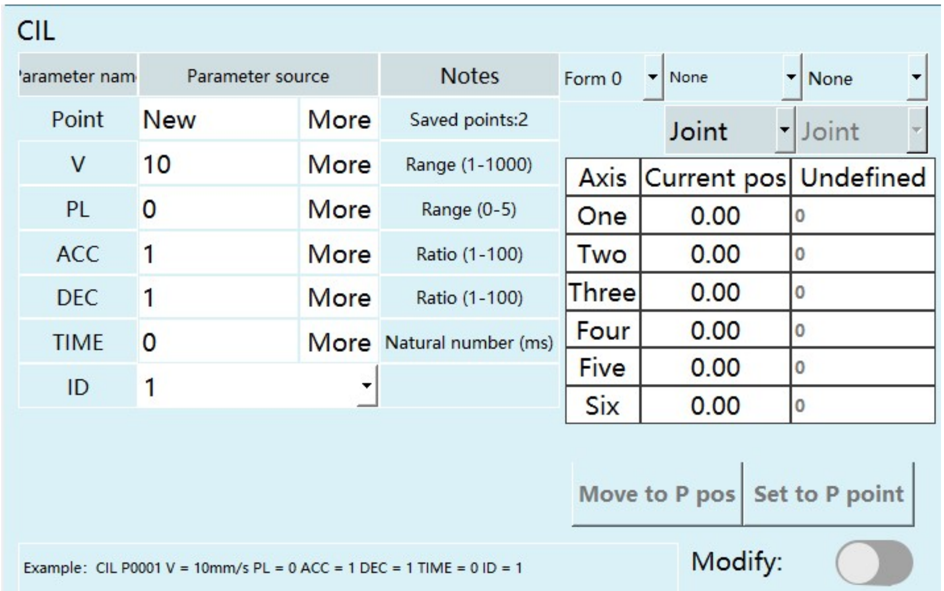
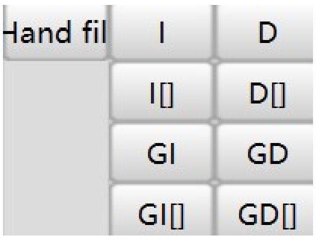


Figure 1

CIL instruction - intersecting line

CIL instruction - intersecting line
Parameter interface

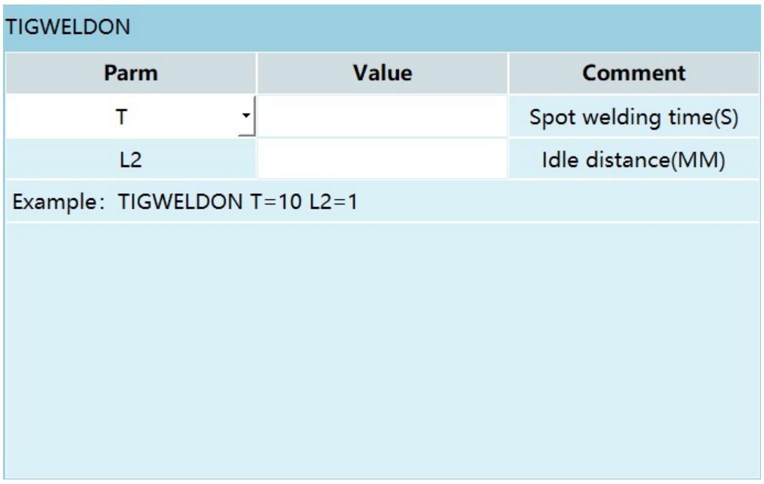


Parameter name	Parameter source	Note
Point: Select "Local position variable"/"Global position variable"/"Bind variable". When the value is "New", inserting this instruction will create a new P variable, and record the current position of the robot to this P variable.	P GP P[] CP[]	New position variable
V: speed, unit mm/s	These parameters can bind variables, as the figure shown below:  	Indicates the range and unit of the parameter
PL: position level For example: create two points, P001 and P002, and walk a straight line. If the PL is set, the degree of connection between the two points will be fine.		
ACC: acceleration, unit percentage		
DEC: deceleration, unit percentage		
TIME: early execution time, unit		

ms		
ID: intersecting line process number		

TIGWELDON INSTRUCTIONS - T.I.G welding start

TIGWELDON INSTRUCTIONS - T.I.G welding start
Function: Execute this instruction to start the T.I.G welding. Please run the welding start ARCON instruction before executing this instruction.
Parameter interface

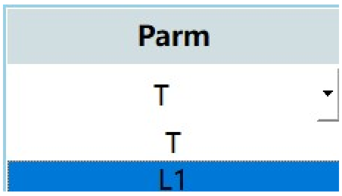
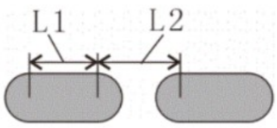
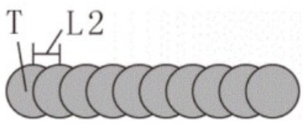


TIGWELDON

Parm	Value	Comment
T		Spot welding time(S)
L2		Idle distance(MM)

Example: TIGWELDON T=10 L2=1

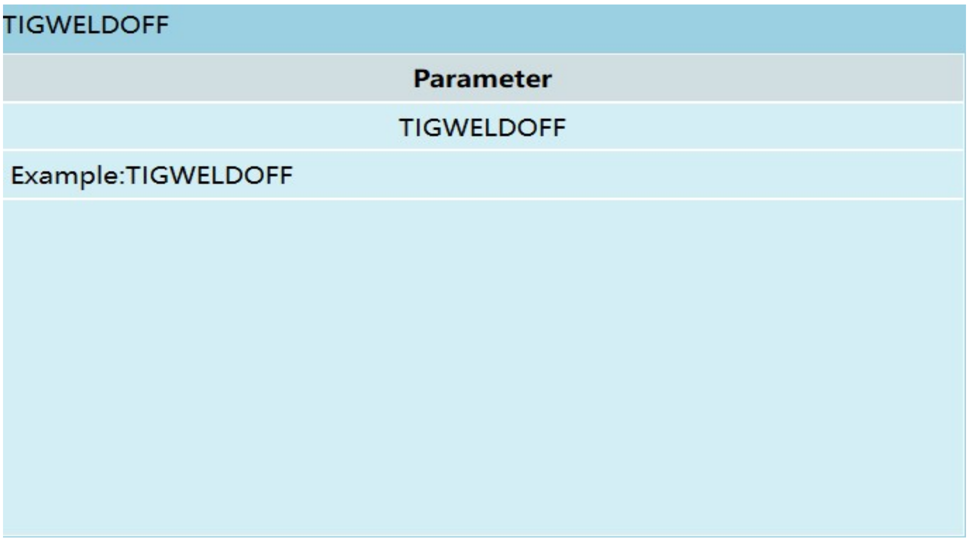
Picture of T.I.G welding

Parameter	Value	Note
T/L1   Figure 1  Figure 2	Figure 1 is an example diagram when setting the parameters of welding distance and idling distance. welding distance The robot continues to weld while moving For example: the set welding distance is 5mm, as shown in Figure 1, the L1 is 5mm L2 means idling distance spot welding time The robot does not move within the set time and keeps welding continuously T means spot welding time L2 means idling distance	L1 means welding distance (mm), T means spot welding time (s)
Idling distance L2	Fill in the idling distance parameter For example: the	Fill in the idling distance parameter, the unit is mm

	filled distance is 5mm, as shown in Figure 1, the L2 is 5mm	

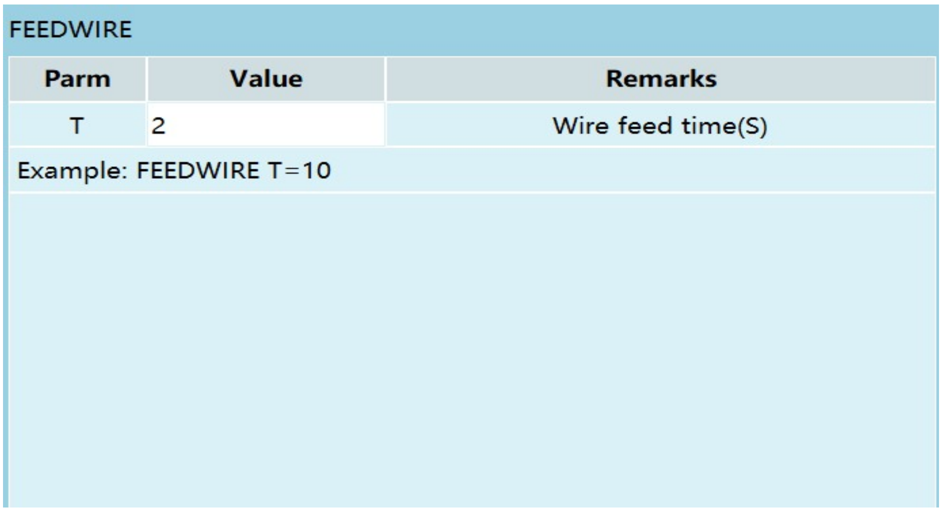
TIGWELDOFF instruction - T.I.G welding end

TIGWELDOFF instruction - T.I.G welding end
Function: Execute this instruction to end the T.I.G welding.
Instruction interface:



FEEDWIRE instruction - Wire feeding

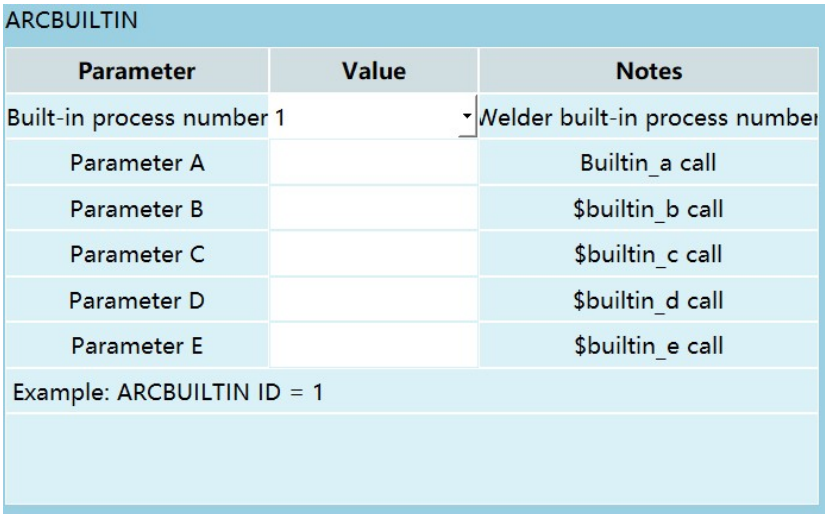
FEEDWIRE instruction - Wire feeding
Function: Feed the wire within the set time at the start of welding or after the end of welding, and then cut off the welding wire when it reaches the designated position, in order to make the welding wire uniform during welding
Parameter interface:



Parameter	Value	Note
T	Represents wire feeding time	Wire feeding time, unit s

ARCBUILTIN instruction - welder built-in process

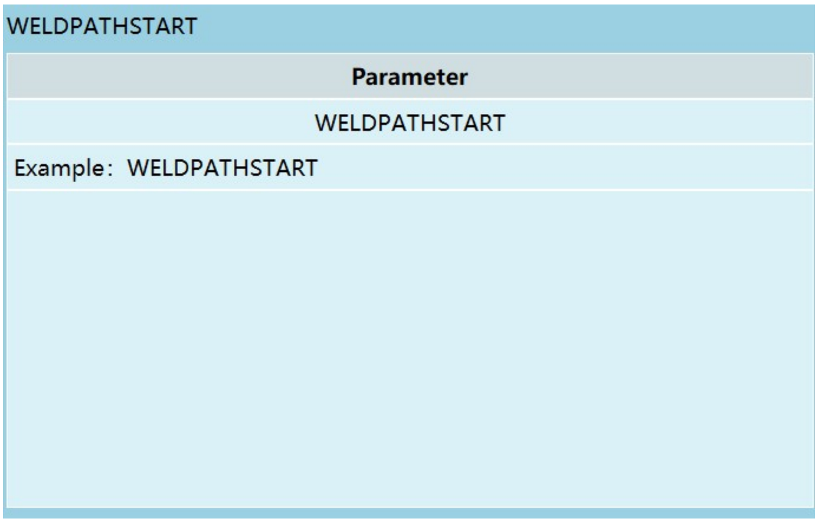
ARCBUILTIN instruction - welder built-in process
Instruction interface This instruction can be used with Aotai welder at present



Parameter	Value	Note

Built-in process number	1-9	Welder built-in process number
Parameter A		$builtin_a call
Parameter B		$builtin_b call
Parameter C		$builtin_c calls
Parameter D		$builtin_d call
Parameter E		$builtin_e call

WELDPATHSTART instruction - initial weld path recording start

WELDPATHSTART instruction - initial weld path recording start
Instruction interface The WELDPATHSTART instruction needs to be used in conjunction with the WELDPATHSTOP instruction. Between the two instructions, only MOVL, MOVCA and MOVC instructions are supported. 
How to use the instruction:

```
0   NOP
> 1   MOVJ P0002 VJ = 10% PL = 0 ACC = 10 DEC = 10 0
2   MOVL P0003 V = 10mm/s PL = 0 ACC = 1 DEC = 1 0
3   ARCON ID=1 TEMP=OFF(NULL NULL NULL NULL NULL NULL NULL NULL)
4   WELDPATHSTART
5   MOVL P0003 V = 10mm/s PL = 0 ACC = 1 DEC = 1 0
6   MOVL P0004 V = 10mm/s PL = 0 ACC = 1 DEC = 1 0
7   WELDPATHSTOP
8   ARCOFF ID=1 TEMP=OFF(NULL NULL NULL NULL)
9   END
```

As shown in the figure: P2 is the safety point, P3 is the welding start point, and P4 is the welding end point, P3-P4 represents the initial weld path of welding

WELDPATHOFFSET instruction - weld path offset calculation

WELDPATHOFFSET instruction - weld path offset calculation

Function: calculate the weld path after offset and run it in the order of calculation

Instruction parameter interface:

WELDPATHOFFSET

Parameter	Value	Notes
Original weld bead	WELDPATH0	
X-axis offset	0	
Y-axis offset	0	
Z-axis offset	0	
A-axis offset	0	
B-axis offset	0	
C-axis offset	0	
Store calculation result	WELDPATH1	

Example: WELDPATHOFFSET WELDPATH0 11 22 33 2 2 2 WELDPATH1

Original weld path	initial weld path, generally starts from weld path 0
X-axis offset	Taking the original weld path as the reference weld path, by setting offsets for

Y-axis offset Z-axis offset A-axis offset B-axis offset C-axis offset	the X, Y, Z, A, B, and C axes, the weld path will be offset on the basis of the original weld path. For example: on a flat workpiece, use the original weld path as the reference weld path, if you want to offset the weld bead to the left or right by 10mm, then it is necessary to set the offset of the Y axis (10mm), the offset axis and offset of the specific weld path can be set by yourself.
Calculation results storage	WELDPATH1-WELDPATH21 Use the original weld path as the reference weld path, and store the results after setting the offsets on the X, Y, Z, A, B, and C axes into the new weld path

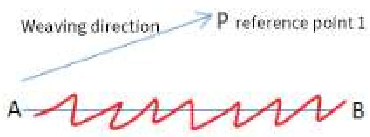
STARTOFFSETWELD instruction - run offset weld path

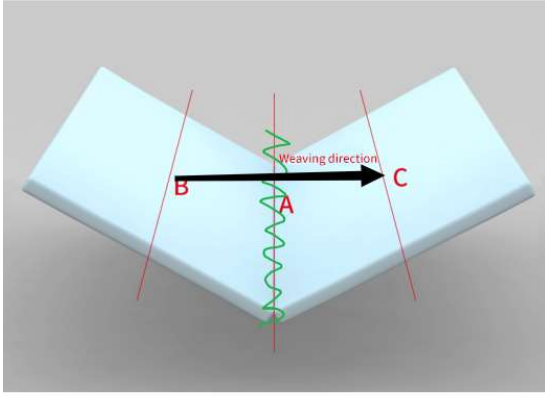
STARTOFFSETWELD instruction - run offset weld path
The trajectory, speed and other parameters of offset weld path are the same as the initial weld path. Take the initial weld path as the reference weld path, set the offset weld path after the offset set by WELDPATHOFFSET instruction, the length of the offset weld bead is the same as the initial weld path, only the angle and position of the offset weld path are different
Instruction interface

STARTOFFSETWELD
Parameter
STARTOFFSETWELD
Example: STARTOFFSETWELD

REFP instruction-weaving reference point

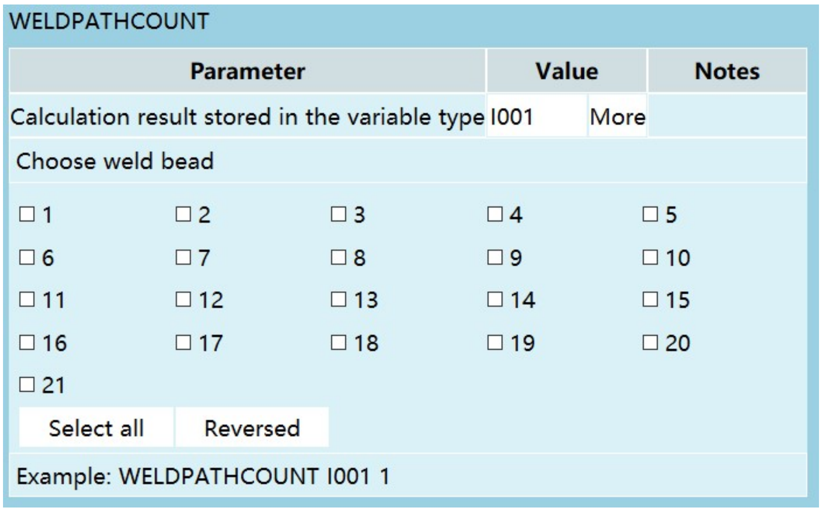
REFP instruction-weaving reference point		
Instruction interface:		
REFP		
Parameter	**:ference point (Cartesia**	**Notes**
Reference point	Reference point1	
track automatic correctic	No	run to the starting point
track speed automatic c	0	2-9999mm/s
'erence point variable na	New	
X		
Y		
Z		
A		
B		
C		
	rent position as referenc	he robot to the referenc
Example: REFP 1 P0001		
Parameter	Reference point (Cartesian)	
Reference point	The red trajectory indicates the trajectory of weaving welding by determining the weaving welding direction Only select reference point 1 or	

	reference point 2  Select both reference point 1 and reference point 2  	
Weaving track automatic correction enable	No/Yes	Calculate and automatically run to the starting point of the correction trajectory
Weaving track automatic correction speed	After the automatic correction enable is turned on, the starting point of the weld path will be offset to the middle of the two reference points The green trajectory indicates the trajectory and direction of weaving welding after correcting the starting point of weaving welding	Speed range 2-9999mm/s

	 Figure 1 For example: as shown in the figure, B is reference point 1, C is reference point 2, and the starting point of the weld path is the center point between the two reference points after the automatic correction is enabled.	

WELDPATHCOUNT instruction - weld path number calculation

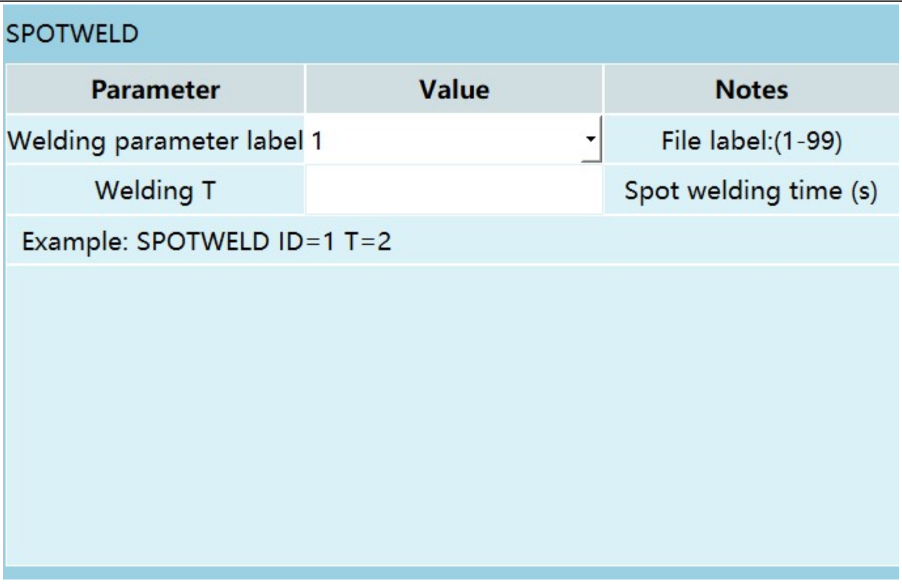
WELDPATHCOUNT instruction - weld path number calculation
Instruction interface



Calculation result storage variable type	Variable and bind variable types can be selected INT, GINT, I[], GI[]
Select weld path	1-21

SPOTWELD instruction - spot welding

SPOTWELD instruction - spot welding
When this instruction is executed, the robot starts to perform spot welding.
Instruction interface:



Parameter	Value	Note
Welding parameter number	1-99	Welding parameter process number used in spot welding (1-99)
Welding time	For example: if the spot welding time is set to 2 seconds, it means that the robot will keep welding within 2 seconds.	The robot does not move within the set time and keeps welding, unit (s)
Usage	Set welding signal, current-voltage matching parameters and welding equipment parameters Set the current, voltage and time parameters of spot welding in the "Welding process-Manual operation" interface After setting the required parameters, click the spot welding enable button on the manual operation interface or the status interface of the welding process, you will	

	find that the set ignition and air supply signal ports are open (because the welder is not connected during the test, the wire feeding and wire retracting signal ports did not respond)

Use cases

Normal ignition welding

Parameter setting

All parameter values are set without specific meaning and are used as examples only

Enter "**Process/Welding process/Welder setting**", set the "Welder control mode" to "Analog control"

Enter "**Process/Welding process/Welding IO**"- Digital input: ignition success signal 1-1; Digital output: ignition signal 1-2; Analog input: welding current signal DIN1-1, welding voltage signal DIN1-2; Analog output: given current signal DOUT1-1, given voltage signal DOUT1-2

Enter "**Process/Welding process/Current-voltage matching**", fill in [1] for welding current in the first line of "Set current", fill in [10] for actual welding current; fill in [1] for welding voltage in the first line of "Set voltage", fill in [10] for actual welding voltage

Enter "**Process/Welding process/Welding parameter setting**", set the ignition parameters: ignition current [8] A, ignition voltage [8] V, ignition time [2] S; welding parameters: welding current [10] A, welding voltage [10] V; arc quenching current [7] A, arc quenching voltage [7] V, arc quenching time [2] S

Enter "**Process/Welding process/Welding equipment setting**" - Basic functions: arc detection time [2] S, arc detection confirmation time [1] S

Enter "**Process/Welding process/Manual operation**", turn on the "Welding enable" switch and the "Manual ignition mode" and "Manual spot welding", set the manual spot welding current to 8A, spot welding voltage to 8V, maximum time to 2S

Use case

```
0  NOP
1  MOVL P001 V = 10 mm/s  PL = 0 ACC = 1 DEC = 1
2  ARCON #1
3  MOVL P002 V = 10 mm/s  PL = 0 ACC = 1 DEC = 1
4  ARCOFF
5  END
```

Instruction meaning

1. The robot moves to the **welding start point P001**

2. ARCON#1 welding start

- Set 4s gas pre-flow time and 1s arc detection time, start ignition and execute welding parameter number 1
- 4s gas pre-flow time (the welder feeds gas 4 seconds in advance, detects the gas, the gas detection signal port outputs high level, after 4s, the ignition starts, and the set ignition signal output port outputs high level)
- 1s arc detection time (If the ignition success signal is detected within 1s, the program continues to run, if not, the error "Waiting for the welding ignition success signal timed out" will be reported)

3. The robot moves to the welding end point P002

- The robot starts to weld, the ignition voltage is 60V, the ignition current is 10A, the ignition time is 1s, the welding voltage is 80V, the welding current is 20A, the arc quenching voltage is 50V
- During the movement from P001 to P002, if the welder starts the ignition successfully, the welder will start the ignition with the ignition current and voltage values, and maintain the set ignition current and voltage values for 1 second (ignition time) until the current and voltage reach the welding current and voltage, and then start welding

4. ARCOFF welding end

- Set arc quenching voltage 50V, arc quenching current 10A and arc quenching time 1s on the "Process-Welding parameters" interface
- Start arc quenching after welding is completed, and keep it for 1 second (arc quenching time) after reaching the arc quenching current and voltage values, then the welding ends, the set air supply and ignition signal output ports are changed from high level 1 to low level 0

Programming

Click "Project", click "New", enter the program name, click "OK"

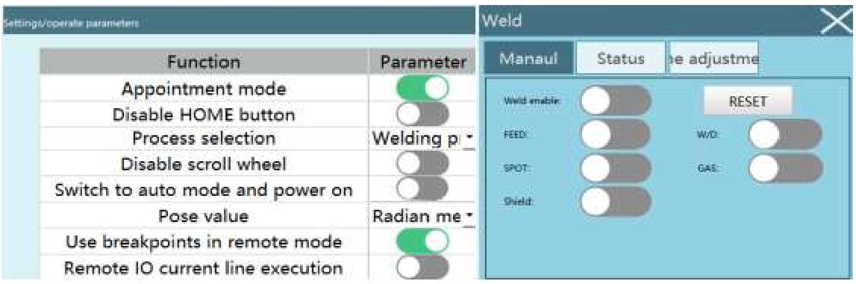
1. Move the robot to the welding start point, click "Insert", select "Motion Control Class", select "MOVL", click "OK", modify the speed value, click "OK"
2. Click "Insert", select "Welding control class", select "ARCON", click "OK", enter the file number (the file number corresponds to the value in the welding parameter setting interface), and click "OK"
3. Move the robot to the welding end point, click "Insert", select "Motion control class", select "MOVL", click "OK", modify the speed value, click "OK"
4. Click "Insert", select "Welding control class", select "ARCOFF", click "OK", and then click "OK" again

Trajectory confirmation: After the program is written, turn the key to switch the teach pendant from the teach mode to the running mode, and click "start" to confirm whether the running trajectory of the robot is correct and whether it meets the needs

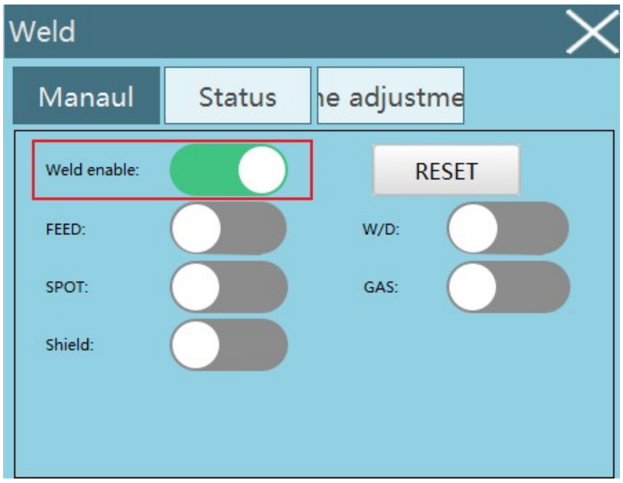
Welding: After confirming that the running trajectory is correct, the robot will perform the welding function only when the "**Welding enable**" is turned on;

When the program is in the running mode, if you press "stop" and then press "start", then after the program restarts, the welding function will no longer be performed.

Welding enable turn-on method: switch the teach pendant to teach mode, click the "Welding" button in the upper right corner, the following picture will appear:



Select "Welding enable" to turn it on;



Weaving welding use case

Parameter setting

All parameter values are set without specific meaning and are used as examples only

- Enter "Process/Welding process/Welder setting", set the "Welder control mode" to "Analog control"
- Enter "Process/Welding process/Welding IO"- Digital input: ignition success signal 1-1; Digital output: ignition signal 1-2; Analog input: welding current signal DIN1-1, welding voltage signal DIN1-2; Analog output: given current signal DOUT1-1, given voltage signal DOUT1-2
- Enter "Process/Welding process/Current-voltage matching", fill in [1] for welding current in the first line of "Set current", fill in [10] for actual welding current; fill in [1] for welding voltage in the first line of "Set voltage", fill in [10] for actual welding voltage
- Enter "Process/Welding process/Welding parameter setting", set the ignition parameters: ignition current [8] A, ignition voltage [8] V, ignition time [2] S; welding parameters: welding current [10] A, welding voltage [10] V; arc quenching current [7] A, arc quenching voltage [7] V, arc quenching time [2] S
- Enter "Process/Welding process/Welding equipment setting" - Basic functions: arc detection time [2] S, arc detection confirmation time [1] S

- Enter "Process/Welding process/Manual operation", turn on the "Welding enable" switch and the "Manual ignition mode" and "Manual spot welding", set the manual spot welding current to 8A, spot welding voltage to 8V, maximum time to 2S

Enter "Process/Welding process/Weaving welding parameters" to set parameters

Use case

```
0  NOP
1  MOVL P001 V = 10 mm/s  PL = 0 ACC = 1 DEC = 1
2  ARCON #1
3  WVON #1
4  MOVL P002 V = 10 mm/s  PL = 0 ACC = 1 DEC = 1
5  WVOFF
6  ARCOFF
7  END
```

Instruction meaning

1. The robot moves to point P001 (weaving welding start point)
2. **ARCON#1 welding start**

- During the movement from P001 to P002, if the welder starts the ignition successfully, the welder will start the ignition with the ignition current and voltage values, and maintain the set ignition current and voltage values for 1 second (ignition time) until the current and voltage reach the welding current and voltage, and then start welding

3. WVON#1 weaving welding start

- Execute the parameters in the weaving file 1; (if it is WVON#2, the weaving starts, and execute the parameters in the weaving file 2)
- After executing this instruction, the robot performs the weaving operation according to the weaving parameters set in **Process/Welding process/Weaving parameters**

4. P002 end point of weaving welding

- P001-P002 is the weaving trajectory that needs to be executed
- The amplitude, frequency, direction/horizontal and vertical deflection angle of the weaving track is performed according to the settings in Figure 1.

5. WVOFF#1 weaving end

- The robot completes the weaving welding operation

6. ARCOFF welding end

- Set arc quenching voltage 50V, arc quenching current 10A and arc quenching time 1s on the "Process-Welding parameters" interface
- Start arc quenching after welding is completed, and keep it for 1 second (arc quenching time) after reaching the arc quenching current and voltage values, then the welding ends, the set air supply and ignition signal output ports are changed from high level 1 to low level 0

Programming

Click "Project", click "New", enter the program name, click "OK"

Move the robot to the **welding start point**, click "Insert", select "Motion control class", select "MOVL", click "OK", modify the speed value and click "OK"

Click "Insert", select "Welding control class", select "ARCON", click "OK", enter the file number (the file number corresponds to the value in the **welding parameter setting** interface), and click "OK"

Click "Insert", select "Welding control class", select "WVON", click "OK", and enter the file number (the file number corresponds to the value in the **weaving welding parameters** interface)

Move the robot to the **welding end point**, click "Insert", select "Motion control class", select "MOVL", click "OK", modify the speed value, click "OK"

Click "Insert", select "Welding control class", select "WVOFF", click "OK", and then click "OK" again

Click "Insert", select "Welding control class", select "ARCOFF", click "OK", and then click "OK" again

Trajectory confirmation: After the program is written, turn the key to switch the teach pendant from the teach mode to the running mode, and click "start" to confirm whether the running trajectory of the robot is correct

Welding: After confirming that the running trajectory is correct, the robot will perform the welding function only when the "**Welding enable**" is turned on; **the welding enable turn-on method has been described in the ignition welding case**

T.I.G welding use case

All parameter values are set without specific meaning and are used as examples only

- Enter "**Process/Welding process/Welder setting**", set the "Welder control mode" to "Analog control"
- Enter "**Process/Welding process/Welding IO**"- Digital input: ignition success signal 1-1; Digital output: ignition signal 1-2; Analog input: welding current signal DIN1-1, welding voltage signal DIN1-2; Analog output: given current signal DOUT1-1, given voltage signal DOUT1-2
- Enter "**Process/Welding process/Current-voltage matching**", fill in [1] for welding current in the first line of "Set current", fill in [10] for actual welding current; fill in [1] for welding voltage in the first line of "Set voltage", fill in [10] for actual welding voltage
- Enter "**Process/Welding process/Welding parameter setting**", set the ignition parameters: ignition current [8] A, ignition voltage [8] V, ignition time [2] S; welding parameters: welding current [10] A, welding voltage [10] V; arc quenching current [7] A, arc quenching voltage [7] V, arc quenching time [2] S
- Enter "**Process/Welding process/Welding equipment setting**" - Basic functions: arc detection time [2] S, arc detection confirmation time [1] S
- Enter "**Process/Welding process/Manual operation**", turn on the "Welding enable" switch and the "Manual ignition mode" and "Manual spot welding", set the manual spot welding current to 8A, spot welding voltage to 8V, maximum time to 2S

Enter "**Process/Welding process/Weaving welding parameters**" to set parameters

Use case

```
0  NOP
1  MOVL P001 V = 10 mm/s  PL = 0 ACC = 1 DEC = 1
2  ARCON #1
3  TIGWELDON L1 = 2 L2 = 3
4  MOVL P002 V = 10 mm/s  PL = 0 ACC = 1 DEC = 1
5  TIGWELDOFF
6  ARCOFF
7  END
```

Instruction meaning

1. P001 (welding start point)
2. ARCON#1 welding start

- The robot starts to weld, the ignition voltage is 60V, the ignition current is 10A, the ignition time is 1s, the welding voltage is 80V, the welding current is 20A, the arc quenching voltage is 50V
- During the movement from P001 to P002, if the welder starts the ignition successfully, the welder will start the ignition with the ignition current and voltage values, and maintain the set ignition current and voltage values for 1 second (ignition time) until the current and voltage reach the welding current and voltage, and then start welding

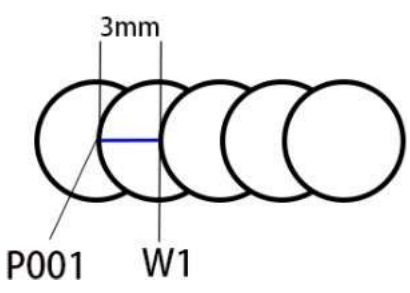
3. TIGWELDON T=2 L2=3 (T.I.G welding start)

Set spot welding time and idling distance

TIGWELDON

Parm	Value	Comment
T	2	Spot welding time(S)
L2	3	Idle distance(MM)
Example：TIGWELDON T=10 L2=1		

- The robot starts ignition, the robot welds at point P001 for 2s (i.e. T=2s), then the robot quenches the arc, and idly walks 3mm (i.e. L2=3mm) to point W1
- The robot starts the ignition at point W1, welds at point W1 for 2s, quenches the arc, and idly walks 3mm to point W2
- ① Start the ignition, ② Weld for 2s, ③ Quench the arc, ④ Idly walk 3mm, and cycle the previous 4 steps until running to the welding end point (P002).

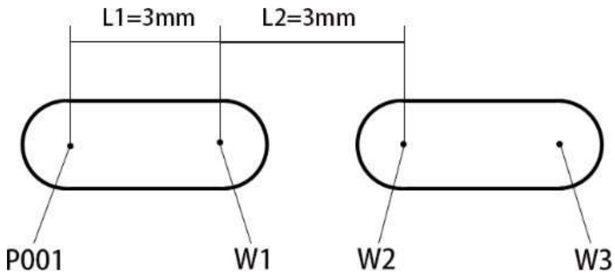


Set welding distance and idling distance

TIGWELDON

Parm	Value	Comment
L1	3	Welding distance(MM)
L2	3	Idle distance(MM)
Example：TIGWELDON T=10 L2=1		

The robot starts the ignition, the robot starts at P001 and runs L1 to W1 in a welding state (the distance between P001 and W1 is 3mm, i.e. the welding distance), then the robot quenches the arc and idly walks 3mm (i.e. L2=3mm) to W2, the robot starts the ignition at W2, the robot runs from W2 to W3 while welding, then the robot quenches the arc and idly walks 3mm to W4. ① Start the ignition, ② Weld for 2s, ③ Quench the arc, ④ Idly walk 3mm, and cycle the previous 4 steps until running to the welding end point (P002).



4. P002 (welding end point)
5. TIGWELDOFF (T.I.G welding end)
6. **ARCOFF welding end**

- Set arc quenching voltage 50V, arc quenching current 10A and arc quenching time 1s on the "Process-Welding parameters" interface
- Start arc quenching after welding is completed, and keep it for 1 second (arc quenching time) after reaching the arc quenching current and voltage values, then the welding ends, the set air supply and ignition signal output ports are changed from high level 1 to low level 0

Programming

- Click "Project", click "New", enter the program name, click "OK"

Move the robot to the **welding start point**

- Click "Insert", select "Motion control class", select "MOVL", click "OK", modify the speed value and click "OK"
- Click "Insert", select "Welding control class", select "ARCON", click "OK", enter the file number (the file number corresponds to the value in the **welding parameter setting** interface), and click "OK"
- Click "Insert", select "Welding control class", select "TIGWELDON", click "OK", select T.I.G welding type: Option 1: select T for the first line of parameter // Option 2: select L1 for the first line of parameter and enter the corresponding value
- Move the robot to the **welding end point**, click "Insert", select "Motion Control Class", select "MOVL", click "OK", modify the speed value, click "OK"
- Click "Insert", select "Welding control class", select "TIGWELDOFF", click "OK", and then click "OK" again, click "Insert", select "Welding control class", select "ARCOFF", click "OK", and then click "OK" again

Trajectory confirmation: After the program is written, turn the key to switch the teach pendant from the teach mode to the running mode, and click "start" to confirm whether the running trajectory of the robot is correct

Welding: After confirming that the running trajectory is correct, the robot will perform the welding function only when the "**Welding enable**" is turned on; **the welding enable turn-on method has been described in the ignition welding case**

Multi-layer multi-pass welding use case (Two-layer three-pass welding)

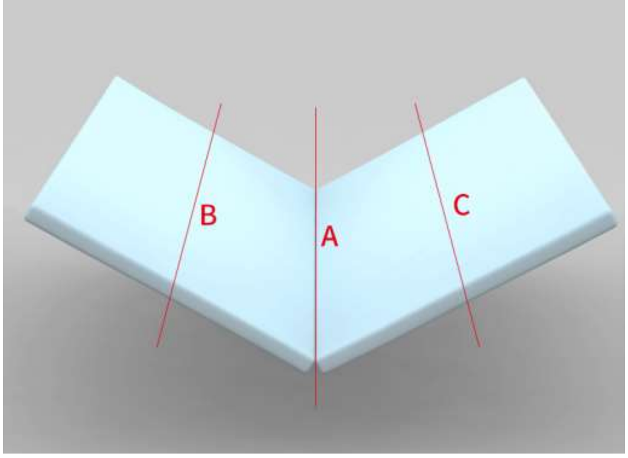
Parameter setting

All parameter values are set without specific meaning and are used as examples only

- Enter "**Welding process/Welder setting**", set the "Welder control mode" to "Analog welder"
- Enter "Welding process/Welding IO"- Digital input: ignition success signal 1-1; Digital output: ignition signal 1-2; Analog input: DIN1-1; Analog output: NOUT1-1

- Enter "Welding process/Current-voltage matching", fill in [1] for welding current in the first line of "Set current", fill in [10] for actual welding current; fill in [1] for welding voltage in the first line of "Set voltage", fill in [10] for actual welding voltage
- Enter "Welding process/Welding parameter setting", set the ignition parameters: ignition current [8] A, ignition voltage [8] V, ignition time [2] S; welding parameters: welding current [10] A, welding voltage [10] V; arc quenching current [7] A, arc quenching voltage [7] V, arc quenching time [2] S
- Enter "Welding process/Welding equipment setting" - Basic functions: arc detection time [2] S, arc detection confirmation time [1] S, arc depletion detection time [1] S, turn on "Gas pre-flow". Select "Delayed gas shut-off" for "Gas shut-off mode", delayed gas shut-off time [2] S
- Enter "Welding process/Manual operation", turn on the "Welding enable" switch, turn on the "Manual ignition mode" and "Manual spot welding", set the manual spot welding current to 8A, spot welding voltage to 8V, maximum time to 2S

Welding trajectory diagram



图一

Programming

```
Name:    WW                                   Times:    0/1
0   NOP
1   MOVJ P0001 VJ = 10% PL = 0 ACC = 10 DEC = 10 0
2   MOVL P0002 V = 10mm/s PL = 0 ACC = 1 DEC = 1 0
3   ARCON ID=1 TEMP=OFF(NULL NULL NULL NULL NULL NULL NULL NULL)
4   WELDPATHSTART
5   MOVL P0002 V = 10mm/s PL = 0 ACC = 1 DEC = 1 0
6   MOVL P0003 V = 10mm/s PL = 0 ACC = 1 DEC = 1 0
7   WELDPATHSTOP
8   ARCOFF ID=1 TEMP=OFF(NULL NULL NULL NULL)
9   WELDPATHOFFSET WELDPATH0 0 100 50 0 0 0 WELDPATH1
10  WELDPATHOFFSET WELDPATH0 0 -100 50 0 0 0 WELDPATH2
11  WELDPATHCOUNT I001 2097151
12  WHILE (GI001 < 4)
13  MOVJ P0001 VJ = 10% PL = 0 ACC = 10 DEC = 10 0
14  GOTO_WELD_START_POS
15  ARCON ID=1 TEMP=OFF(NULL NULL NULL NULL NULL NULL NULL NULL)
16  STARTOFFSETWELD
17  ADD GI001 1
18  ARCOFF ID=1 TEMP=OFF(NULL NULL NULL NULL)
19  ENDWHILE
20  END
```

Instruction meaning

Line 1: robot safety point

Line 2: start point of initial weld path

Line 3: welding start, start the ignition according to the parameters set in the "Process-Welding process" interface, start to prepare for welding

Lines 4-7: Indicates that P002-P003 is recorded as the initial weld path, as shown in Figure 1 above, the initial weld path is A

Line 8: welding end, initial path welding complete

Line 9: The offset weld bead that needs to be welded. According to Figure 1, B is the offset weld bead 1 that needs to be welded. The Y and Z axes offsets are set on the basis of the initial weld bead

Line 10: the offset weld bead to be welded, according to Figure 1, C is the offset weld bead 2 to be welded, and the Y and Z axis offsets are set on the basis of the initial weld bead

Line 11: Calculation of the number of welds

Lines 12-18: complete the welding of the offset weld bead through a cycle, give the cycle instruction a judgment condition, when a section of weld bead is welded, add 1 to GI001. When GI004<4, jump out of the loop and complete multi-pass welding

The use of spot welding

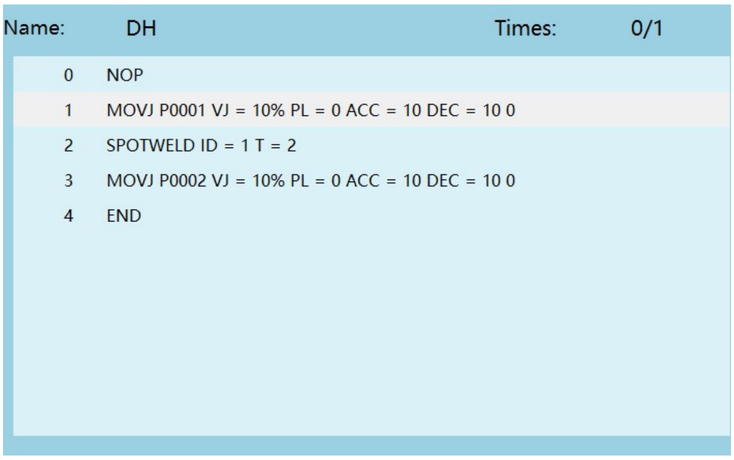
All parameter values are set without specific meaning and are used as examples only

1. Enter "Process/Welder setting", set the welder control mode to "Analog"
2. Set welding signal

Digital input	Ignition success signal 1-1
Digital output	Ignition signal 1-1
	Wire feed signal 1-2
	Wire retract signal 1-3
	Air supply signal 1-4
Analog input	AIN1-1
Analog output	AOUT1-1

3. Enter "Process/Current-voltage matching", fill in [1] for welding current in the first line of setting current, and [10] for actual welding current; fill in [1] for welding voltage in the first line of setting voltage, and fill in [10] for actual welding voltage
4. Enter "Process/Manual operation", turn on the welding enable switch, set the manual spot welding current to 8A, spot welding voltage to 8V, and the maximum time to 2S

Example of use of spot welding instructions:



Instruction meaning

Line 1: The point where the robot needs to perform spot welding

Line 2: set the spot welding time, set the spot welding time to two seconds according to the instruction parameters, the robot will spot weld at P001 point continuously for two seconds, supply air during the welding process, and the ignition signal port changes from low level 0 to high level 1, when the spot welding time is reached, gas will be supplied after welding, and the ignition signal port will change from high level 1 to low level 0

Line 3: the safety point where the robot moves to after the spot welding is completed

Weaving welding reference point use case

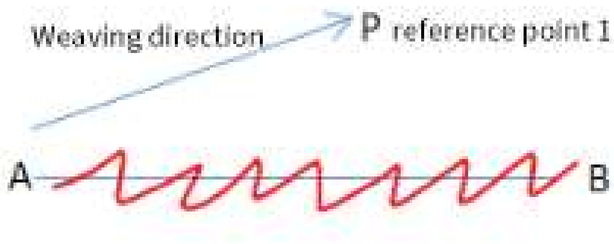
Parameter setting

All parameter values are set without specific meaning and are used as examples only

- Enter "**Welding process/Welder setting**", set the "Welder control mode" to "Analog welder"
- Enter "**Welding process/Welding IO**" - Digital input: ignition success signal 1-1; Digital output: ignition signal 1-2; Analog input: DIN1-1; Analog output: NOUT1-1
- Enter "**Welding process/Current-voltage matching**", fill in [1] for welding current in the first line of "Set current", fill in [10] for actual welding current;

fill in [1] for welding voltage in the first line of "Set voltage", fill in [10] for actual welding voltage

- Enter "**Welding process/Welding parameter setting**", set the ignition parameters: ignition current [8] A, ignition voltage [8] V; welding parameters: welding current [10] A, welding voltage [10] V; arc quenching current [7] A, arc quenching voltage [7] V
- Enter "**Welding process/Welding equipment setting**" - Basic functions: arc detection time [2] S, arc detection confirmation time [1] S, arc depletion detection time [1] S
- Enter "**Welding process/Manual operation**", turn on the "Welding enable" switch
- Insert a REFP (weaving reference point) instruction, set the current position as the the reference point of weaving welding, the MOVL trajectory and the points calibrated by REFP (weaving reference point) form the weaving welding plane, and determine the weaving direction. As shown in Figure 1 below, point A and reference point 1 determine the weaving welding direction, and the red trajectory is the weaving welding trajectory through the weaving welding direction.
- Weaving welding plane: used with the weaving start instruction to determine the coordinate system of the weaving welding



Programming

Name:	BH	Times: 0/1
0	NOP	
1	MOVJ P0001 VJ = 10% PL = 0 ACC = 10 DEC = 10 0	
2	ARCON ID=1 TEMP=OFF(NULL NULL NULL NULL NULL NULL NULL NULL)	
3	WVON #1	
4	MOVL P0002 V = 10mm/s PL = 0 ACC = 1 DEC = 1 0	
5	REFP 1 P0003 0 0	
6	MOVL P0004 V = 10mm/s PL = 0 ACC = 1 DEC = 1 0	
7	WVOFF	
8	ARCOFF ID=1 TEMP=OFF(NULL NULL NULL NULL)	
9	MOVJ P0005 VJ = 10% PL = 0 ACC = 10 DEC = 10 0	
10	END	

Instruction meaning:

Line 1: safety point when welding starts

Line 2: start welding

Line 3: start weaving welding

Lines 4-6: As shown in Figure 1 above, the direction of weaving welding is determined from point A to reference point 1, P2-P4 is the trajectory of weaving welding, and the trajectory of weaving welding is carried out through the direction of weaving welding.

Line 7: Weaving welding is completed

Line 8: welding done

Line 9: Safety point where the robot moves after welding

External axis fixed point weaving welding

1. After connecting the external axis, click on the robot on the "Settings-Robot parameters-Slave configuration" interface to see if the external axis has been connected successfully
2. After the external axis is successfully connected, calibrate the external axis and parameter settings on the "Settings-External axis Parameters" interface.
3. Welding process parameter setting

All parameter values are set without specific meaning and are used as examples only

- Enter "**Welding process/Welder setting**", set the "Welder control mode" to "Analog welder"

- Enter "**Welding process/Welding IO**" - Digital input: ignition success signal 1-1; Digital output: ignition signal 1-2; Analog input: DIN1-1; Analog output: NOUT1-1
- Enter "**Welding process/Current-voltage matching**", fill in [1] for welding current in the first line of "Set current", fill in [10] for actual welding current; fill in [1] for welding voltage in the first line of "Set voltage", fill in [10] for actual welding voltage
- Enter "**Welding process/Welding parameter setting**", set the ignition parameters: ignition current [20] A, ignition voltage [10] V, ignition time [1] S; welding parameters: welding current [50] A, welding voltage [20] V; arc quenching current [28] A, arc quenching voltage [15] V
- Enter "**Welding process/Welding equipment setting**" - Basic functions: arc detection time [2] S, arc detection confirmation time [1] S
- Enter "**Welding process/Manual operation**", turn on the "Welding enable" switch, turn on the manual ignition mode, set the manual spot welding current to 8A, the spot welding voltage to 8V, and the maximum time to 2S.
- Enter "**Process/Welding process/Weaving welding parameters**" and set the parameters, as shown in the figure below

Process/welding process/swing welding parameters

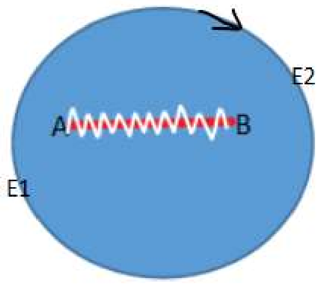
eaving fi 1 :omment

Parameter	Value	Notes
Weave mode	External axis fixed	
Weave frequency	1	Range (0,20](Hz)
Weave amplitude	1	Range (0-50](mm)
Move/stay	Mobile	Move/stay
Right dwelling time	1	Range 0-15(s)
Left dwelling time	1	Range 0-15(s)
Starting direction	+1	Starting direction(+1/-1)
Horizontal deflectior	0	[-180,180] (°)
Vertical deflection	0	[-180,180] (°)

Trajectory of fixed-point weaving welding

- Calibrate two points on the external axis: E1 (start point of external axis rotation axis), E2 (end point of external axis rotation axis); P1-P2 determines the direction of rotation of the external axis

- The A-B red line segment indicates the linear trajectory of the weaving welding on the external axis: A (the start point of the linear trajectory, the robot needs to move) and B (the end point of the linear trajectory, the robot needs to move)
- The white curve represents the weaving trajectory when weaving welding is performed after setting the weaving welding parameters.
- While the external axis is rotating, the robot performs fixed-point weaving welding, and the weaving welding trajectory is the part indicated by the white curve in the figure below



Programming:

```
Name:    Q12                         Times:    0/1
0   NOP
1   MOVLEXT E0001 V = 10 mm/s PL = 0 ACC = 1 DEC = 1 SYNC = 0 0
2   ARCON ID=1 TEMP=OFF(NULL NULL NULL NULL NULL NULL NULL NULL)
3   MOVLEXT E0002 V = 10 mm/s PL = 0 ACC = 1 DEC = 1 SYNC = 1 0
4   ARCOFF ID=1 TEMP=OFF(NULL NULL NULL NULL)
5   END
```

Instruction meaning:

Line 1: Insert the MOVLEXT instruction (both the external axis and the robot need to move), determine the coordinate point and external axis point of the robot, i.e. point E1 and point A in the above figure

Line 2: start welding operation

Line 3: Insert the MOVLEXT instruction (both the external axis and the robot need to move), determine the coordinate point and the external axis point of the robot, i.e. point E2 and point B in the above figure. when inserting the second MOVLEXT instruction, you need to turn on the synchronization function (when the synchronization function is turned on, the robot performs weaving welding on the external axis while the external axis is rotating)

Line 4: welding end

Searching and Tracking Process

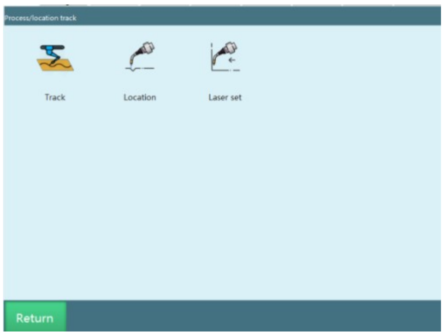
Catalogue

Searching and Tracking Process

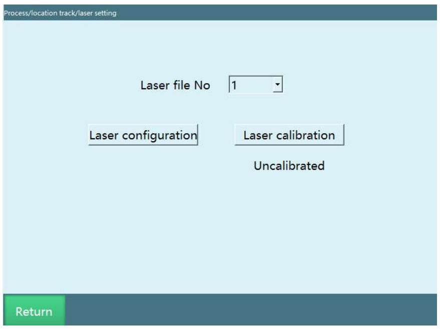
Laser tracking process

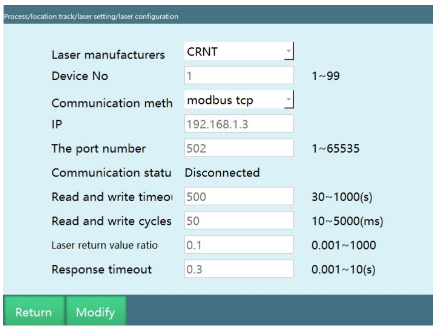
Laser setting

Parameter setting: Enter the "Process/Searching and tracking process" to set parameters, the file number corresponds to the file number in the instruction, and the laser is selected according to the actual use



Enter "Laser setting/Laser configuration" to set the communication between the laser and the controller





Laser manufacturer: Select the corresponding laser device name.

Device number: The corresponding upper computer.

Communication method: Modbus or Ethernet

IP: The IP of the connected upper computer. It is necessary to ensure that the controller, the upper computer, and the teach pendant are in the same network segment before they can be connected.

Port number: The port numbers of the teach pendant and the upper computer need to be the same.

Communication status: "Connected" will be displayed when the laser is turned on.

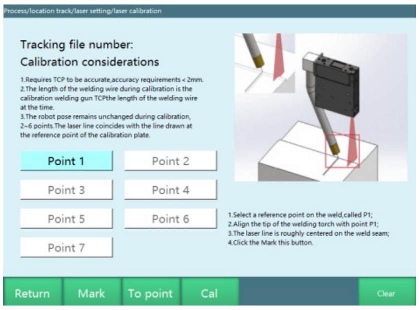
Read and write timeout: If the laser does not receive data after reading and writing for such a long time (s), it will time out

Read and write cycle: The time interval between each data reading and writing of the upper computer (ms).

Laser return value scale factor: The ratio of the actual coordinate value to the coordinate value returned by the laser.

Response timeout: In communication with the laser, the timeout period between the robot query instruction and the laser response instruction.

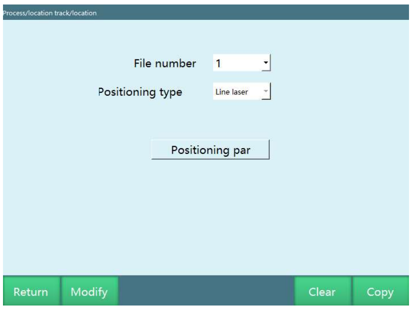
Enter "Laser setting/Laser calibration" to calibrate the laser

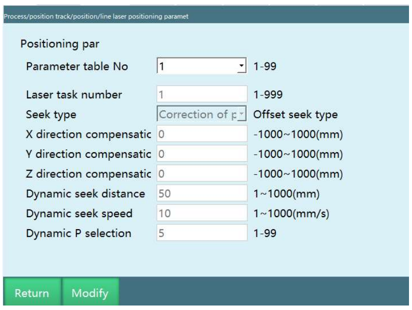


Calibrate seven points according to the diagram. When you first enter, a small white bar will pop up in the lower right corner to indicate that the laser is connected successfully. If it prompts that the initialization fails or the connection fails, you must check whether the manufacturer, ip, and port number in the laser configuration are set correctly. When calibrating, it is necessary to ensure that the weld surface is parallel to the laser, and the laser must be perpendicular to the weld. During the calibration process, the attitude needs to be kept unchanged. At the same time, it is necessary to make sure that the intersection of the weld seam and the laser can been seen in the corresponding manufacturer's debugging software for each calibrated point without shaking. After you have calibrated the seven points, you can click "Move here" to check, and click "Calculate" if it is correct. If you find that the point is not accurate during the searching process, you need to re-calibrate the laser or tool hand.

Searching process

Enter "**Searching/[Line laser] Searching parameters**" for parameter setting





Parameter table number: Similar to the process number of other processes, it can save the parameters of different users and can be selected in the instruction.

Laser task number: Corresponding to the previous device number.

Searching type: (1) Reference searching: After the searching point is calibrated, the robot will convert the searched point into a variable, insert the variable by instructions and walk to the point; (2) Correction searching: Based on the reference searching, on the requirements of the workpiece or weld, select 1-4 points for reference searching, then the weld can be translated left and right on the plane and rotated according to the number of points, and the robot tool hand can still find it and follow the weld. This method is usually used in the welding of a large number of identical workpieces in the same batch.

X-direction compensation: Compensate a certain length in the tool coordinate system of the welding seam position recognized by the laser

Y-direction compensation: Compensate a certain length in the tool coordinate system of the welding seam position recognized by the laser

Z-direction compensation: Compensate a certain length in the tool coordinate system of the welding seam position recognized by the laser

Dynamic searching distance: The distance of the robot's dynamic searching, the robot needs to visually measure how far it can reach the weld, otherwise it cannot find the weld

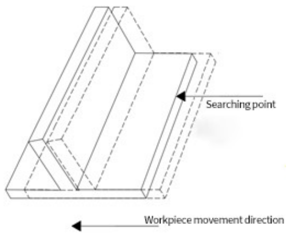
Dynamic searching speed: The speed of dynamic searching

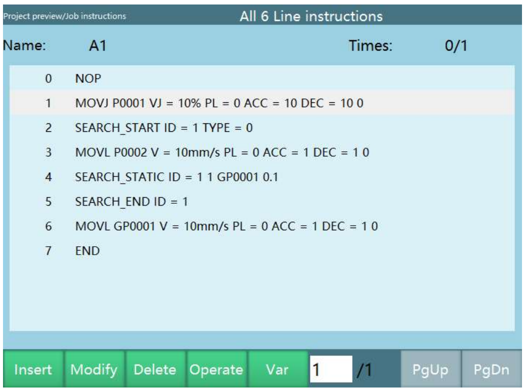
Dynamic searching point selection: Calculate how many points the laser will read within the distance according to the read and write cycle and dynamic searching distance. When the laser just touches the weld, there will be height errors or interference of non-weld gaps in other directions, so these points must be filtered out to ensure the points can be accurately found by dynamic searching.

Types of Laser Searching&Tracking and Use Cases

Single-point searching

Single-point searching (two-point, three-point, four-point searching) is to insert the corresponding number of SEARCH_STATIC instructions between the SEARCH_START and SEARCH_END instructions, and ensure that there is a moving point before each SEARCH_STATIC and the laser can find the weld on the upper computer. The single-point searching function is mainly used to check the calibration accuracy after the robot and the laser are calibrated; the realization method is to send the data of the point taken by the laser to the robot and then the robot moves to the point.





SEARCH_START: Turn on the laser

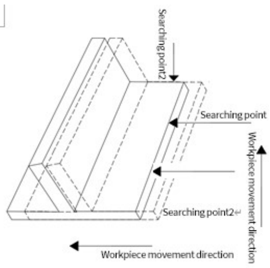
SEARCH_STATIC: Save the weld seam found by the laser into a variable for later calculation or direct movement to the point

SEARCH_END: Turn off the laser, the file number should be the same as the beginning

MOVL: Run to the point of the previous searching

Two-point searching

The two-point searching function is mainly used for intermittent welding and straight weld applications. Two points are taken by laser and the point data is sent to the robot, then the robot walks two points to form a straight line. Two static searching points are required in the instruction



SEARCH_START: Turn on the laser

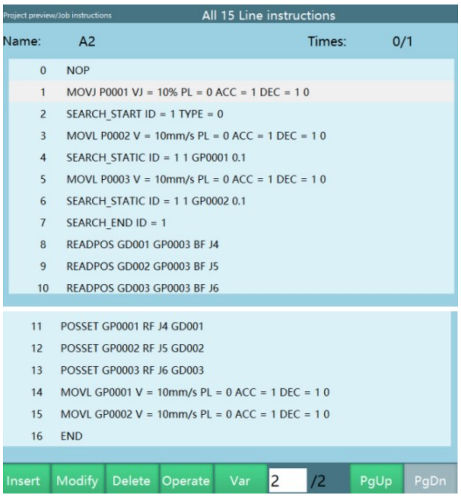
SEARCH_STATIC: Save the weld seam found by the laser into a variable for later calculation or direct movement to the point

SEARCH_END: Turn off the laser, the file number should be the same as the beginning

MOVL: Run to the point of the previous searching

Two-point searching attitude change function

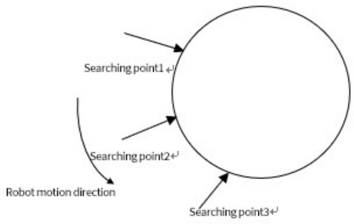
The two-point searching attitude change means searching with one attitude and welding with one attitude, which is mainly used to solve the problem that the searching attitude interferes with the workpiece during welding by changing the attitude. The instructions are the same as the two-point searching, except that the robot attitude is different during the searching, or it can be run by customizing the attitude as follows

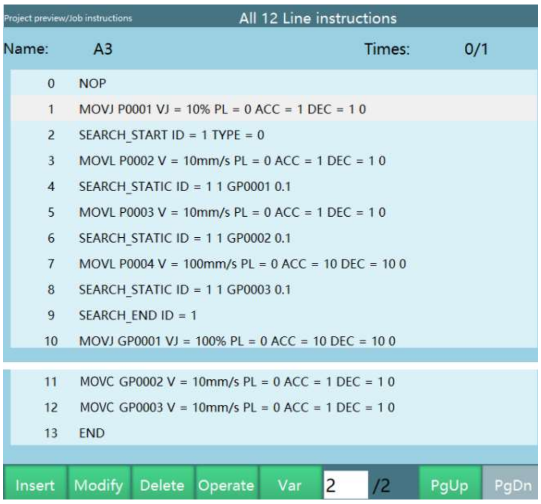


Note: The robot's attitude change path is as follows (Variables>Global position variables>find the global position variable parameter GP0003 you set>adjust to the attitude you want to apply>click "Write to current position"), the global position used here does not conflict with the searching point, the ABC attitude value of GP0003 is taken out and assigned to the running point GP0001 or GP0002.

Three-point arc function

The three-point arc function means that the laser finds three points on the arc, and then uses the MOVC instruction to form an arc with three points. This function is mainly used in the arc workpiece welding scenarios;





SEARCH_START: Turn on the laser

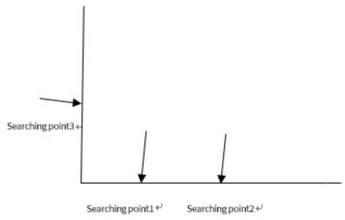
SEARCH_STATIC: Save the weld seam found by the laser into a variable for later calculation or direct movement to the point

SEARCH_END: Turn off the laser, the file number should be the same as the beginning

MOVC: Substitute the variables saved in the previous three-point searching into the MOVC instruction, so that the robot will walk the arc according to the searching points

Three-point searching to calculate coordinate system

Three-point searching is to take three points on the two sides where the workpiece intersects, and calculate the user coordinate system through these three points. This method is used in most welding situations. If the calculated user coordinate system is different from the original user coordinate system, then the points or welds in the original user coordinate system become the points or welds in the calculated user coordinate system. Three-point offset supports one-point offset, two-point offset and rotation offset;



SEARCH_START: Turn on the laser

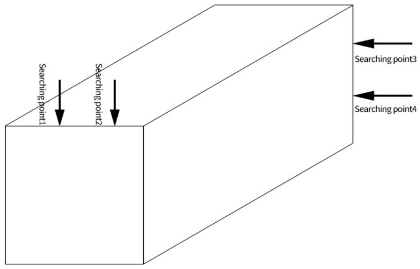
SEARCH_STATIC: Save the weld seam found by the laser into a variable for later calculation or direct movement to the point

SEARCH_END: Turn off the laser, the file number should be the same as the beginning

SEARCH_CALC: Select "3-points to calculate the user coordinate system", use the three variables found before to calculate the user coordinate system 1

Four-point searching to calculate coordinate system

The four-point searching function means to take four points on the workpiece with two points on any one side, and calculate the user coordinate system, so that each four-point searching will result in a new user coordinate system, but the trajectory within the user coordinate system will not change. Three-point searching for intersection is to search for three points on both sides of the workpiece, which can also calculate the intersection point. During four-point searching, if every two points found are not in the same plane of the workpiece, then the overall size of the workpiece can be calculated and the overall user coordinate system of the workpiece can be calculated too;



SEARCH_START: Turn on the laser

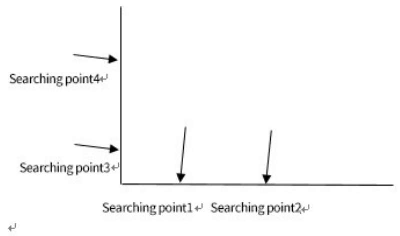
SEARCH_STATIC: Save the weld seam found by the laser into a variable for later calculation or direct movement to the point

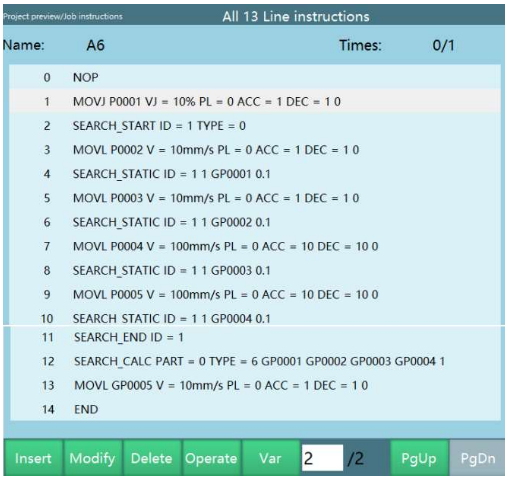
SEARCH_END: Turn off the laser, the file number should be the same as the beginning

SEARCH_CALC: Select "4 points to calculate the user coordinate system", use the four variables found before to calculate the user coordinate system 2

4 points to determine the two lines to calculate intersection

"4 points to determine the two lines to calculate intersection" is to take four points on both sides of the intersection of the workpiece, two points on each side determine a straight line, and then the two lines intersect to determine the intersection point.




Project preview/Job instructions   All 13 Line instructions

Name: A6   Times: 0/1

```
0   NOP
1   MOVJ P0001 VJ = 10% PL = 0 ACC = 1 DEC = 1 0
2   SEARCH_START ID = 1 TYPE = 0
3   MOVL P0002 V = 10mm/s PL = 0 ACC = 1 DEC = 1 0
4   SEARCH_STATIC ID = 1 1 GP0001 0.1
5   MOVL P0003 V = 10mm/s PL = 0 ACC = 1 DEC = 1 0
6   SEARCH_STATIC ID = 1 1 GP0002 0.1
7   MOVL P0004 V = 100mm/s PL = 0 ACC = 10 DEC = 10 0
8   SEARCH_STATIC ID = 1 1 GP0003 0.1
9   MOVL P0005 V = 100mm/s PL = 0 ACC = 10 DEC = 10 0
10  SEARCH_STATIC ID = 1 1 GP0004 0.1
11  SEARCH_END ID = 1
12  SEARCH_CALC PART = 0 TYPE = 6 GP0001 GP0002 GP0003 GP0004 1
13  MOVL GP0005 V = 10mm/s PL = 0 ACC = 1 DEC = 1 0
14  END
```

Insert Modify Delete Operate Var 2 /2 PgUp PgDn

SEARCH_START: Turn on the laser

SEARCH_STATIC: Save the weld seam found by the laser into a variable for later calculation or direct movement to the point

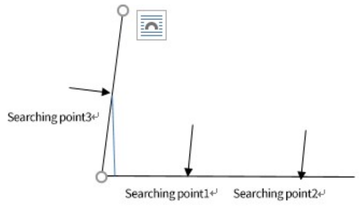
SEARCH_END: Turn off the laser, the file number should be the same as the beginning

SEARCH_CALC: Select "4 points to determine the two lines to calculate intersection", and use the GP0001, GP0002, GP0003, GP0004 points data to calculate the intersection GP0005

MOVL: Run to the calculated intersection point

3 points to calculate the projection point

"3 points to calculate the projection point" is to take three points on both sides of the intersection of the workpiece, two points on one side determine a straight line, and the vertical foot is determined by the projection point of a point on the other side on the straight line. The value is recorded in the global variable.



SEARCH_START: Turn on the laser

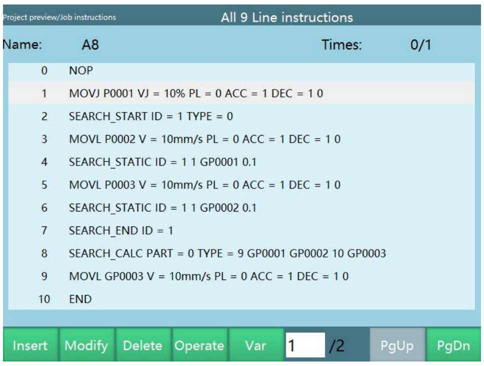
SEARCH_STATIC: Save the weld seam found by the laser into a variable for later calculation or direct movement to the point

SEARCH_END: Turn off the laser, the file number should be the same as the beginning

SEARCH_CALC: Select "3 points to calculate the projection point", and calculate the projection point GP0004 through the point data of GP0001, GP0002, and GP0003

MOVL: Run to the calculated projection point

Vector calculation



SEARCH_START: Turn on the laser

SEARCH_STATIC: Save the weld seam found by the laser into a variable for later calculation or direct movement to the point

SEARCH_END: Turn off the laser, the file number should be the same as the beginning

SEARCH_CALC: Select "Vector calculation", select 10mm from GP0001 to GP0002 to calculate GP0003

MOVL: Run to the calculated vector point

4 points to calculate plane user coordinate system

The purpose and effect of the test:

Calculate a new user coordinate system based on the shadow by reflecting the shadow of the object on the plane.

Testing process:

First, you need to find a fixed plane and calibrate a minimum user coordinate system 1, as shown in figure ①;

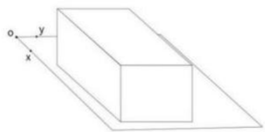


Figure ①

Then find two adjacent and intersecting sides on the workpiece, and use laser to mark two points on each side, and mark four points a1, b1, c1, d1 in total (calibrate each point with laser single-point searching), as shown in figure ②; after four points have been calibrated, calculate and obtain the user coordinate system 2; it is necessary to find a weld L1 on the current workpiece for calibration (using MOVL);

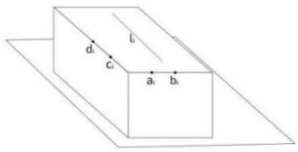
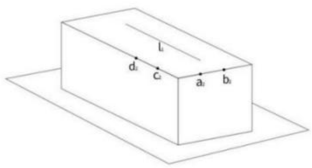


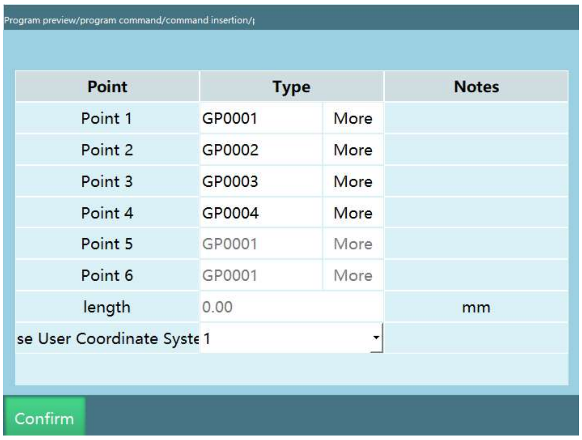
Figure ②

After the above operations are completed, offset or rotate the workpiece and perform the second calibration (a2,b2,c2,d2) at the position of the four points previously calibrated; calculate the user coordinate system 3.



Note: The above operations must be performed with a tool hand.

Instruction application for testing:



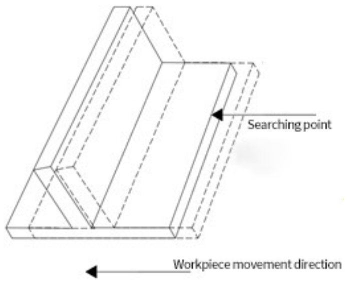
The above figure is the instruction insertion process of user coordinate system 2;

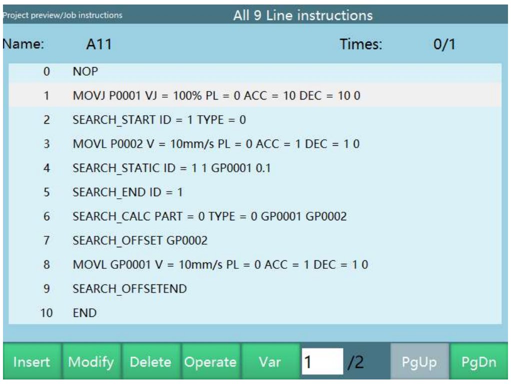
The above figure is the instruction writing process of user coordinate system 3.

Searching offset

1D offset

Use case: After single-point searching, the workpiece can only move in one direction, and the searching direction must be the same as the offset direction





SEARCH_START: Turn on the laser

SEARCH_STATIC: Save the weld seam found by the laser into a variable for later calculation or direct movement to the point

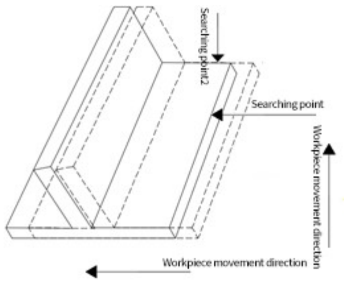
SEARCH_END: Turn off the laser, the file number should be the same as the beginning

SEARCH_CALC: Select a x-dimensional offset according to the searching points and the actual situation, the point to be offset is GP0001, and the offset is GP0002

SEARCH_OFFSET: Offset is to use the offset instruction to compensate the error when a large number of workpieces are welded. There are different usages of single-point to four-point offset, you can use them according to the actual situation. Use the calculated GP0002 offset to calculate the offset point of GP0001, GP0001 can be replaced with the required weld.

2D offset

After two-point searching, only the XY direction offset occurs when the workpiece is not rotated



SEARCH_START: Turn on the laser

SEARCH_STATIC: Save the weld seam found by the laser into a variable for later calculation or direct movement to the point

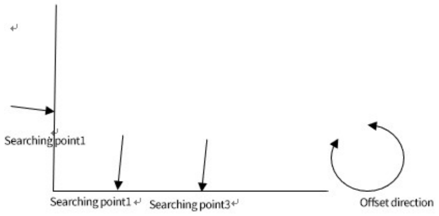
SEARCH_END: Turn off the laser, the file number should be the same as the beginning

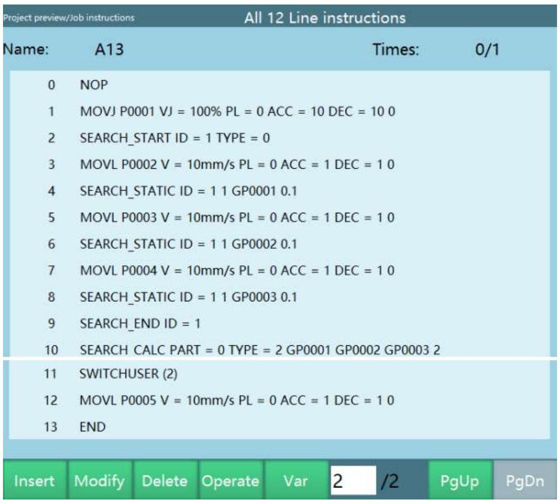
SEARCH_CALC: Select a x-dimensional offset according to the searching points and the actual situation, the point to be offset is GP0001, and the offset is GP0003

SEARCH_OFFSET: Offset is to use the offset instruction to compensate the error when a large number of workpieces are welded. There are different usages of single-point to four-point offset, you can use them according to the actual situation. Use the calculated GP0003 offset to calculate the offset point of GP0001, GP0001 can be replaced with the required weld.

2D offset + rotation

After the three-point searching, the workpiece can be rotated as a whole and offset in XY directions, the first time for reference searching and the second time for correction searching when the offset occurs.





SEARCH_START: Turn on the laser

SEARCH_STATIC: Save the weld seam found by the laser into a variable for later calculation or direct movement to the point

SEARCH_END: Turn off the laser, the file number should be the same as the beginning

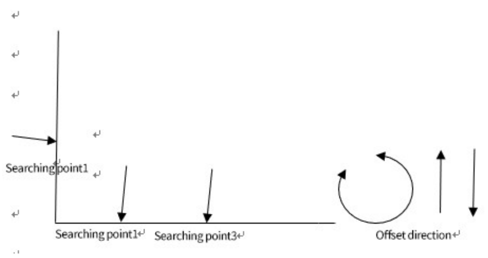
SEARCH_CALC: Select "2D offset + rotation", the point to be offset is P0005, calculate user coordinate system with three points

SWITCHUSER (2): Switch to the calculated user coordinate system

MOVL: At this time, P0005 is a user point that is taught in advance and will be offset according to the different user coordinate system calculated each time, P0005 can be replaced with the required weld seam

3D offset (retain base user coordinate system)

After three-point searching or four-point searching, the workpiece can be rotated as a whole and offset in XY directions, and two job files are required



Project preview/Job instructions — All 13 Line instructions

Name: A14 Times: 0/1

```
0   NOP
1   MOVJ P0001 VJ = 100% PL = 0 ACC = 10 DEC = 10 0
2   SEARCH_START ID = 1 TYPE = 0
3   MOVL P0002 V = 10mm/s PL = 0 ACC = 1 DEC = 1 0
4   SEARCH_STATIC ID = 1 1 GP0001 0.1
5   MOVL P0003 V = 10mm/s PL = 0 ACC = 1 DEC = 1 0
6   SEARCH_STATIC ID = 1 1 GP0002 0.1
7   MOVL P0004 V = 10mm/s PL = 0 ACC = 1 DEC = 1 0
8   SEARCH_STATIC ID = 1 1 GP0003 0.1
9   SEARCH_END ID = 1
10  SEARCH_CALC PART = 0 TYPE = 5 GP0001 GP0002 GP0003 2
```

At this point, you can copy the above program and then insert the following instructions

```
11  SWITCHUSER (3)
12  MOVL P0005 V = 10mm/s PL = 0 ACC = 1 DEC = 1 0
13  SWITCHUSER (2)
14  END
```

SEARCH_START: Turn on the laser

SEARCH_STATIC: Save the weld seam found by the laser into a variable for later calculation or direct movement to the point

SEARCH_END: Turn off the laser, the file number should be the same as the beginning

SEARCH_CALC: Select "Three points to calculate user coordinate system", and output the user coordinate system 2

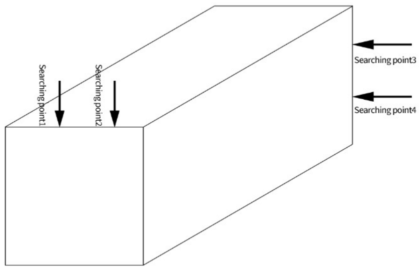
At this time, the second job file is needed, because the previous job file is the basic user coordinate system, and the next job file is the user coordinate system after calculating the offset. You can copy this file and add the following instructions:

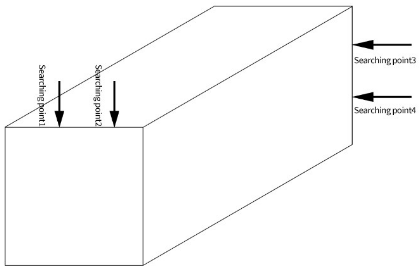
```
11  SWITCHUSER (3)
12  MOVL P0005 V = 10mm/s PL = 0 ACC = 1 DEC = 1 0
13  SWITCHUSER (2)
14  END
```

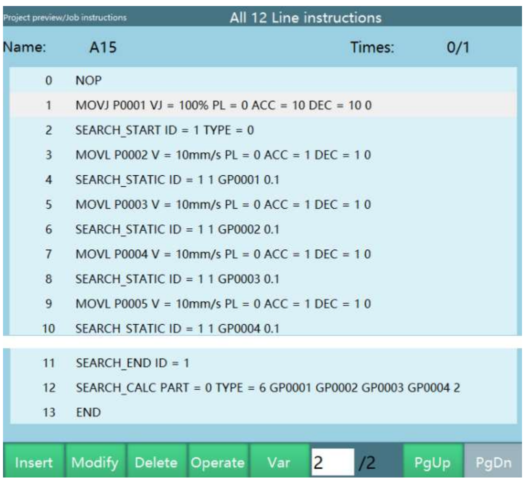
In the second job file, it is necessary to calculate and switch to the new user coordinate system to offset according to the difference between the user coordinate systems. P0005 must be taught in the user coordinate system 3 calculated after the first run, and must be a user point. P0005 can be replaced with the required weld seam. After that, no matter how the workpiece rotates, as long as the laser can find the 3 points, the P0005 after offset can be calculated. After the operation is completed, the coordinate system needs to be restored to the initial coordinate system 2, so as not to affect the subsequent user points

3D offset + rotation

After 4-point searching, the workpiece can be rotated as a whole and offset in XYZ directions, and two job files are required







SEARCH_START: Turn on the laser

SEARCH_STATIC: Save the weld seam found by the laser into a variable for later calculation or direct movement to the point

SEARCH_END: Turn off the laser, the file number should be the same as the beginning

SEARCH_CALC: Select "Four points to calculate user coordinate system", and output the user coordinate system 5

At this time, the second job file is needed, because the previous job file is the basic user coordinate system, and the next job file is the user coordinate system after calculating the offset. You can copy this file and add the following instructions:

```
SEARCH_CALC PART = 0 TYPE = 6 GP0001 GP0002 GP0003 GP0004 5
SWITCHUSER (5)
MOVL P0006 V = 10mm/s PL = 0 ACC = 1 DEC = 1 0
SWITCHUSER (2)
END
```

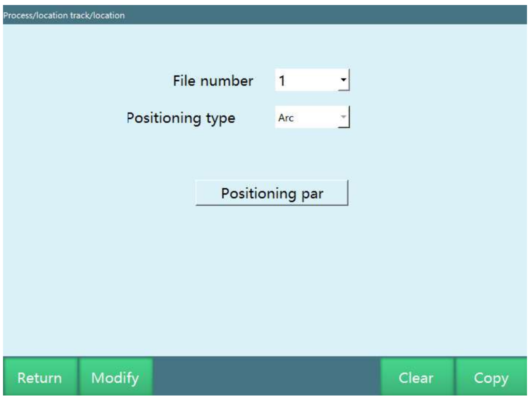
In the second job file, it is necessary to calculate and switch to the new user coordinate system to offset according to the difference between the user coordinate systems. P0006 must be taught in the user coordinate system 2 calculated after the first run, and must be a user point. P0006 can be replaced with the required weld seam.

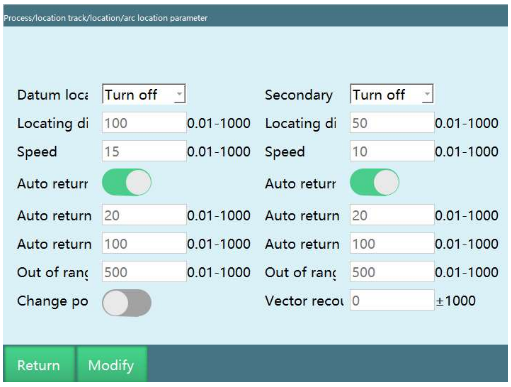
If the ground rail is used in the searching, the device number must be selected from 1 to 4 in the laser configuration.

Arc searching process

Arc searching process

Enter the "**Process/Searching and tracking/Searching/[Arc] Searching parameters**" to set the arc parameters





Searching file number: Corresponds to instruction file number

Reference searching: First searching

Secondary searching: In some cases, the reference searching is not very accurate or some manufacturers' reference searching is too fast, so use the secondary searching

Searching distance: The moving distance from the instruction searching start point

Speed: Searching speed

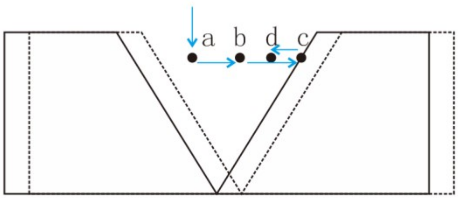
Auto return: Return after the torch touches the searching points

Auto return distance: The distance to go back from touching the workpiece

Change attitude: Turn on when performing two-point simple touch searching calculation.

Motion vector compensation: When performing two-point simple touch searching calculation, the reverse compensation is 0~5mm to prevent the welding wire from poking into the weld seam.

Introduction of arc searching points

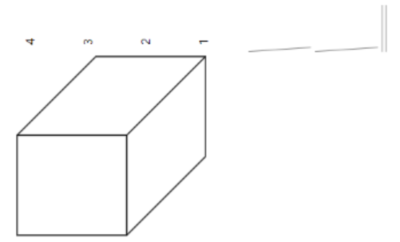


As shown in the figure: point a is the preparation point of dynamic searching; point b is the start point of dynamic searching; the robot moves along the direction of vector ab to find the position, and the welding wire stops immediately when it touches the workpiece, indicating that the position has been found; the searching distance (point b is the start point) and the speed are set in the process parameters.

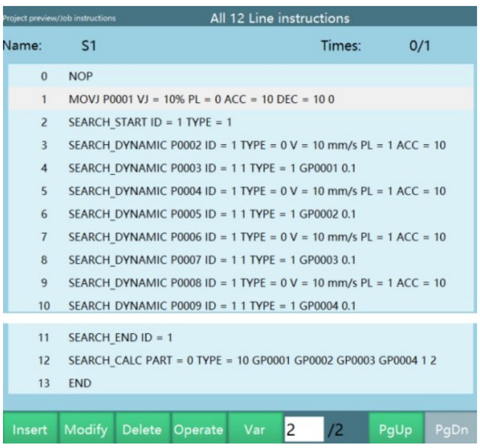
If it is required to return automatically after searching, the robot will return from c to d automatically (return distance and speed are set in the parameters). In the arc searching process parameters, select "Reference searching", configure other parameters; run the program, the program will stop at the SEARCH_CALC instruction (normal), turn off the reference searching switch in the parameters. Run the program again

Arc searching types and use cases

Calculate the new user coordinate system with two points on the plane



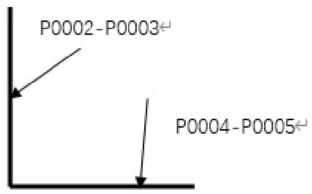
Operation process: Use the SEARCH_DYNAMIC instruction to find a total of 4 points. Select the two intersecting sides and find two points on each side according to the required plane user coordinate system, as shown in the figure above. You can adjust the specific points according to the actual situation.

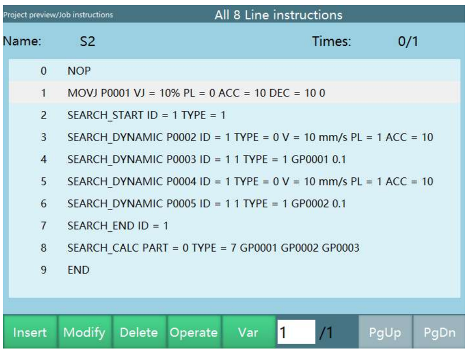


In the middle, you need to add MOVJ or MOVL as path auxiliary points according to the actual situation. After finding 4 points by dynamic searching, calculate the new user coordinate system by "SEARCH_CALC-two points in plane to calculate the required coordinate system", if offset is required, the method can be the same as "three points to calculate the user coordinate system"

2-point easy touch searching

Operation process: First dynamically search the touch point GP0001 in the direction perpendicular to one side of the fillet weld, then dynamically search the touch point GP0002 in the direction perpendicular to the other side of the fillet weld, and then use the SEARCH_CALC instruction to calculate the weld point GP0003 through the two points GP0001,GP0002, and the attitude of the weld point is the same as GP0001.





SEARCH_START: Turn on the arc signal

SEARCH_DYNAMIC: Save the point found by the arc into a variable for later calculation or direct movement to the point

SEARCH_END: Turn off the arc signal, the file number should be the same as the beginning

SEARCH_CALC: Save the found points GP0001 and GP0002 to the variable GP0003, and finally calculate the fillet weld point GP0003.

3D offset + rotation

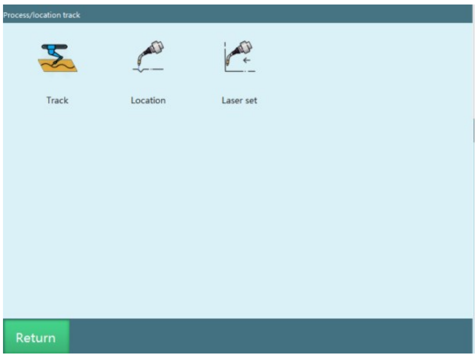
Operation process: You need to touch a plane to find three points and then touch along the edge according to the actual needs of the user coordinate system to find the points, a total of six points, as shown in the figure above

In the middle, you need to add MOVJ or MOVL as path auxiliary points according to the actual situation. After finding 6 points by dynamic searching, calculate the user coordinate system by "SEARCH_CALC-3D offset + rotation" instruction, if offset is required, the follow-up method is the same as that of 4-point searching, and then create a same job file and teach out the weld, offset by switching the user coordinate system

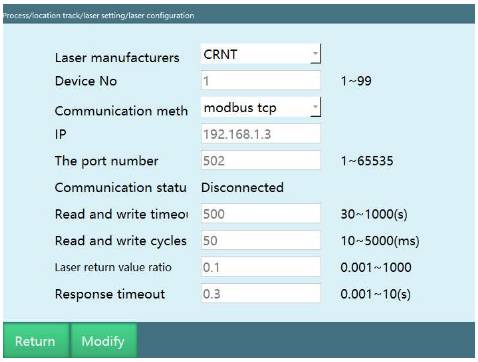
Laser tracking process

Searching and tracking process

Parameter setting: Enter "**Process/Searching and tracking**" to set parameters, the file number corresponds to the file number in the instruction, and the laser is selected according to the actual use



Enter "Searching and tracking/Laser setting/Laser configuration" to set the communication between the laser and the controller



Laser manufacturer: Select the manufacturer according to the laser model

Device number: The corresponding upper computer.

IP: The IP of the connected upper computer. It is necessary to ensure that the controller, the upper computer, and the teach pendant are in the same network segment before they can be connected.

Port number: The port number of the teach pendant and the upper computer need to be the same.

Communication status: "Connected" will be displayed when the laser is turned on.

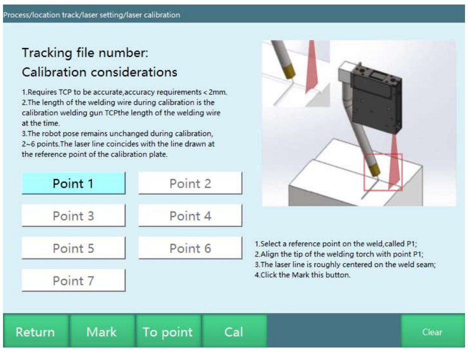
Read and write timeout: If the laser does not receive data after reading and writing for such a long time (s), it will time out

Read and write cycle: The time interval between each data reading and writing of the upper computer (ms).

Laser return value scale factor: The ratio of the actual coordinate value to the coordinate value returned by the laser.

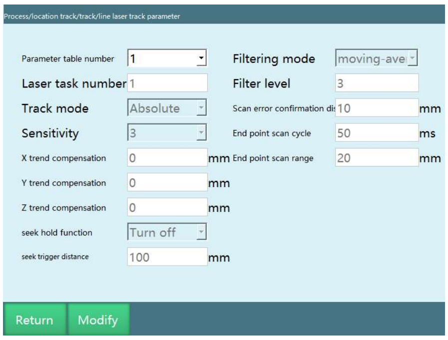
Response timeout: In communication with the laser, the timeout period between the robot query instruction and the laser response instruction.

Enter "Searching and tracking/Laser setting/Laser calibration" to calibrate the laser



Calibrate seven points according to the diagram. When calibrating, it is necessary to ensure that the weld surface is parallel to the laser, and the laser must be perpendicular to the weld. During the calibration process, the attitude needs to be kept unchanged. At the same time, it is necessary to make sure that the intersection of the weld seam and the laser can been seen in the corresponding manufacturer's debugging software for each calibrated point without shaking. After you have calibrated the seven points, you can click "Move here" to check, and click "Calculate" if it is correct. If you find that the point is not accurate during the searching process, you need to re-calibrate the laser or tool hand.

Enter "Searching and tracking/Tracking/Line laser tracking parameters" for parameter setting



Parameter table number: Similar to the process number of other processes, it can save the parameters of different users and can be selected in the instruction.

Laser task number: Corresponds to the previous device number.

Tracking mode: (1) Absolute, that is, precise tracking, in the case of known weld seam, track accurately by searching the start point or moving directly to the vicinity of the weld seam. Precise tracking can ensure that when the weld seam offsets or tool hand changes attitude in the process of tracking, as long as the laser can identify the weld seam, it can accurately make the tool hand move along the weld seam (temporarily only supports linear motion) (2) Incremental, that is, fuzzy tracking, fuzzy tracking is used when precise tracking is not required. As long as the robot tool hand remains stationary on the weld after calibration, it will only move in the direction perpendicular to the weld according to the movement of the weld.

Sensitivity: Laser sensitivity during incremental tracking.

X-direction compensation: Compensate a certain length in the tool coordinate system of the welding seam position recognized by the laser

Y-direction compensation: Compensate a certain length in the tool coordinate system of the welding seam position recognized by the laser

Z-direction compensation: Compensate a certain length in the tool coordinate system of the welding seam position recognized by the laser

Searching hold function: Compensate to a certain position according to the taught weld seam during fuzzy tracking, and then keep tracking at this position.

Searching hold trigger distance: The complementary distance during fuzzy tracking, applicable to short weld seams.

Filtering mode: The filtering algorithm method for smoothing sensor data.

Filtering level: The lower the level, the smoother, the more lagging.

Scan error confirmation distance: During the tracking process, if the sensor fails to scan continuously and the robot moves a certain distance, the error will be reported and the robot will stop.

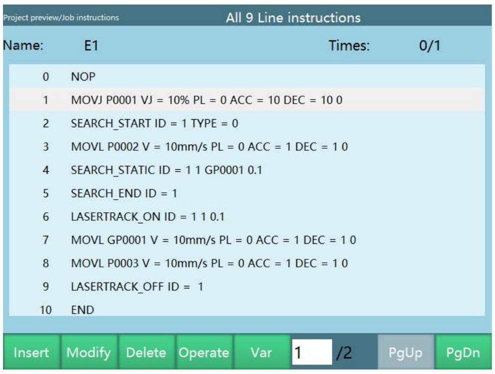
End point scan cycle: the scan cycle, the cycle is smaller in general, less than 30ms.

End point scan interval: Set a distance before and after the end point of teaching as the scan interval.

Tracking use cases

Linear tracking (absolute)

Determine the direction of the weld seam by teaching a straight line, and then scan and track it in real time through the laser to ensure that the weld torch can be kept on the identified weld seam for welding operations. Similar to the principle of searching, the torch can also change its attitude during tracking. If you need to change the attitude, you only need to change the attitude at the teach point



SEARCH_START: Turn on the laser

SEARCH_STATIC: Save the weld seam found by the laser into a variable for later calculation or direct movement to the point

SEARCH_END: Turn off the laser, the file number should be the same as the beginning

LASERTRACK_ON: Turn on the laser

MOVL: Run to the point of the previous searching

MOVL: G001 is the start point, P0003 is the end point, PL must be 5, if there is obvious acceleration and deceleration, please go to the laser configuration to modify the read and write cycle until it does not stop

LASERTRACK_OFF: Turn off the laser, the file number should be the same as the beginning

Arc/arc voltage tracking process

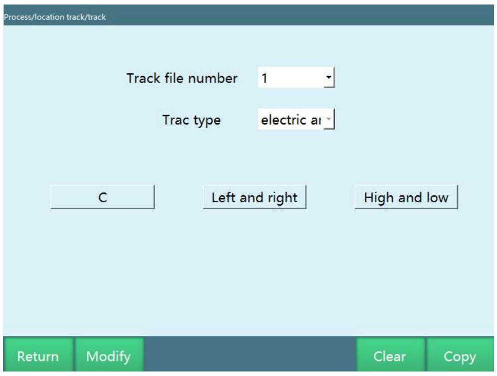
Arc tracking

1. Overview

Arc tracking is mainly suitable for the correction of the weaving welding trajectory of fillet welds and V-groove welds. It is often used in medium and thick plate welding in robot welding processes to correct workpiece deformation and partial workpiece alignment error caused by high-current welding.

2. Parameter configuration

Enter the "Process/Searching and tracking/tracking" interface, as shown in the figure below, you need to set the communication parameters, left and right compensation parameters and high and low compensation parameters in turn.

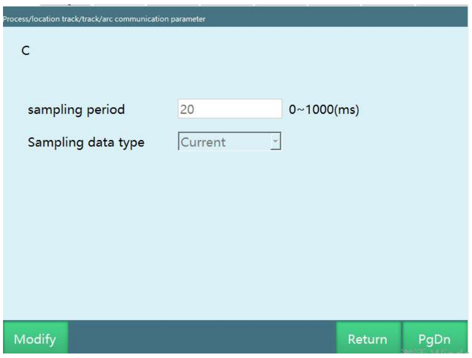


Communication parameters

Enter the "Communication parameters" interface, as shown in the figure below, and the meaning of each parameter is as follows:

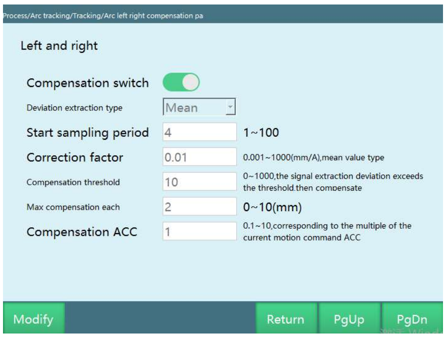
Sampling period: The time period for collecting current and voltage signals during weaving welding, the recommended period is 2ms~20ms.

Sampling data type: During the weaving welding process, choose the voltage/current with larger fluctuations. For Aotai and Megmeet welders, it is recommended to select the current.



Left and right compensation parameters

Left and right compensation refers to the left and right compensation of the swing arc trajectory on the swing plane. Enter the "Left and right compensation parameters" interface, as shown in the figure below, and the meanings of each parameter are as follows:



Compensation switch: Indicates whether left and right correction is performed during weaving welding. For V-groove welds weaving welding with only high and low deviations, this switch can be turned off.

Deviation extraction type: Only the mean value algorithm is currently supported.

Start sampling cycle number: The current signal has no obvious change in the first few cycles of weaving welding, and it is invalid. Generally, sampling starts from the 3rd to 5th cycle.

Correction factor: The compensation length of the current deviation per 1A. Generally, the welding current is large and the deviation value is large, so the value should be set small; otherwise, the value should be set large. Suggested value: 0.01~0.5.

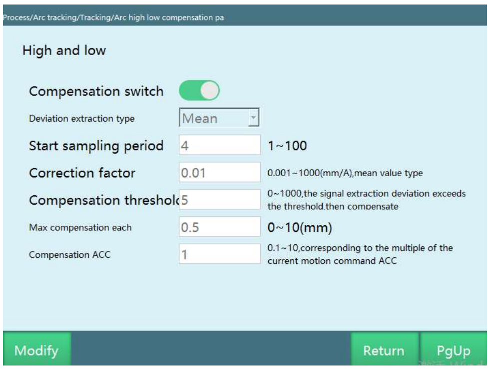
Compensation threshold: Compensate if the current signal deviation exceeds the threshold value, otherwise no compensation. The recommended value is 10. If the welding current is large, adjust it larger, and if the welding current is small, set it smaller.

Maximum compensation amount per time: Compensate once when deviation is extracted in a single cycle, maximum compensation amount per time refers to the maximum length of each compensation, and the maximum compensation amount is to prevent overcompensation caused by excessive current in sampling.

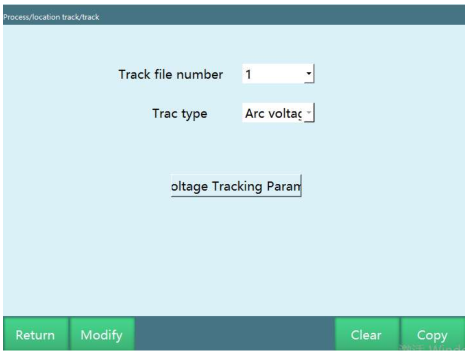
Compensation acceleration multiple: The acceleration of the left and right compensation amount, you can set it according to the deviation correction factor, if the deviation correction factor is large, set it to a large value, and if the deviation correction factor is small, set it to a small value. The recommended value is 1.

High and low compensation parameters

High and low compensation refers to the compensation in the normal direction of the swing plane. Enter the "High and low compensation parameters" interface, as shown in the figure below, the meanings of each parameter are the same as the left and right compensation parameters. In general, the left and right compensation is large, and the high and low compensation is small, so the compensation related parameters can be adjusted appropriately.



Arc pressure tracking

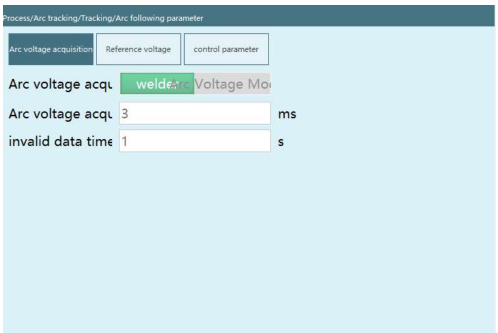


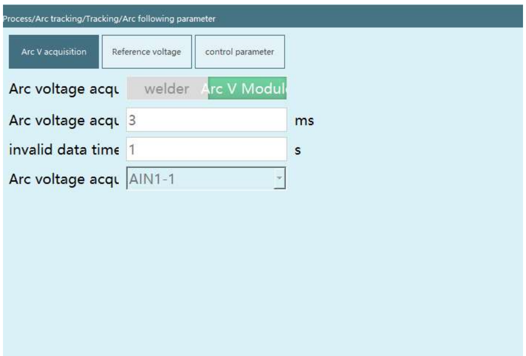
Clear parameters: Clear the parameter values set in the current tracking file number

Copy parameters: Copy the parameter values set in the current tracking file number to the process number you want

Arc pressure tracking parameters

Arc voltage acquisition





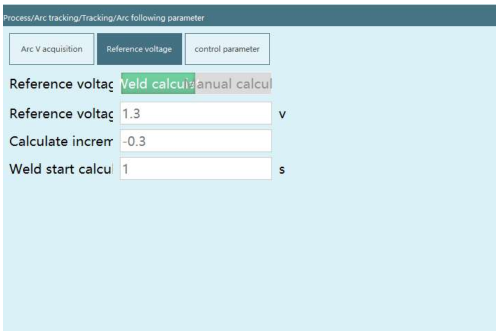
Arc voltage acquisition equipment: Welder and arc voltage module

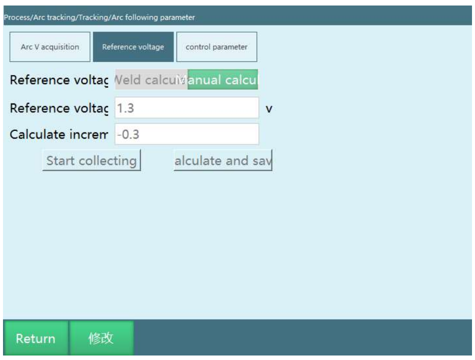
Arc voltage acquisition cycle: Refers to how long to collect a voltage, the unit is ms

Invalid data time: There is a period of time when the welding arc has a extra large voltage, and the data cannot be collected for calculation

Arc voltage acquisition analog port: The analog input port that needs to be connected, only available when the arc voltage module is selected

Reference voltage





Reference voltage acquisition method: It is divided into welding calculation and manual calculation. When welding calculation is selected, after the voltage is set, the reference voltage will be calculated after the welding start calculation time. If manual calculation is selected, the entire track needs to be run

Reference voltage: The user sets the voltage he wants, and when the voltage of the welder exceeds or falls below this set voltage value, deviation correction is required

Calculation increment: The voltage value that needs to be compensated, generally the target value is controlled to be a value that is a fraction of a V smaller than the calculated value

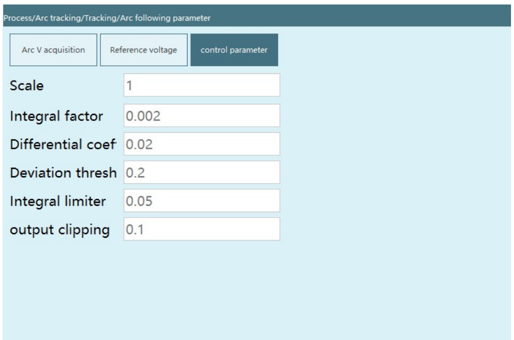
Welding start calculation time: The time required to calculate the reference voltage after welding starts, only available for welding calculation

Start collection: Only available for manual calculation. When manual calculation is selected, the arc voltage pop-up interface will be displayed. Click "Start collection" when starting operation, and click "End collection" after operation

Calculate and save: After running, click "Calculate and save", the reference voltage will be saved in the controller and displayed on the interface

Note: The reference voltage calculated by welding will not be displayed on the teach pendant, it can only be seen from the log

Control parameters



Proportional coefficient: It reflects the deviation of the system in proportion. Once there is a deviation in the system, the proportional adjustment will immediately produce an adjustment effect to reduce the deviation. Large proportional effect can speed up the adjustment and reduce the error, but the excessive ratio will reduce the stability of the system, and even cause system instability

Integral coefficient: It is to eliminate the steady-state error of the system and improve the indiscrimination degree. Because there is an error, the integral adjustment will be carried out until there is no difference, the integral adjustment will stop, and the integral adjustment will output a constant value. The strength of the integral action depends on the integral time constant Ti, the smaller Ti is, the stronger the integral action is. On the contrary, the larger Ti is, the weaker the integral action is. Adding integral adjustment can reduce the stability of the system and slow down the dynamic response. The integral action is often combined with the other two adjustment laws to form a PI regulator or PID regulator

Differential coefficient: The differential action reflects the rate of change of the system deviation signal. It is predictable and can predict the trend of deviation changes. Therefore, it can produce advanced control effects. Before the deviation is formed, it has been eliminated by the differential adjustment function. Therefore, it can improve The dynamic performance of the system. When the differential time is selected properly, the overshoot can be reduced and the adjustment time can be reduced. The differential action can amplify the noise interference, so if the differential adjustment is too strong, it will be detrimental to the system's anti-interference. In addition, the differential response is the rate of change, and when the input does not change, the output of the differential action is zero. The differential action cannot be used alone, it needs to be combined with the other two regulation laws to form a PD or PID controller

Deviation threshold: When the deviation of the controlled quantity is greater than this value, the proportional coefficient and integral coefficient will be reduced. We generally set a larger deviation

Integral limit: Prevent the error integral from being too large

Output limit: Prevent single adjustment from being too large

How to determine in-range and out-of-range

Set the voltage of the welder, and an average voltage will be calculated through the sampling period you set

For example: the reference voltage is 20V, the compensation threshold is 5V, the reference voltage plus or minus compensation threshold, and the calculated value will be compared with the calculated average voltage. If the calculated value is outside the average voltage range, it will be compensated, if the calculated value is within the average voltage range, it will not be compensated.

compensation value per time: The maximum compensation distance in each calculation cycle. If your sampling period is set to 10ms and the maximum compensation distance per time is 1mm, then the maximum compensation distance per time means to compensate 1mm at 100ms.

Compensation length (L) calculation formula

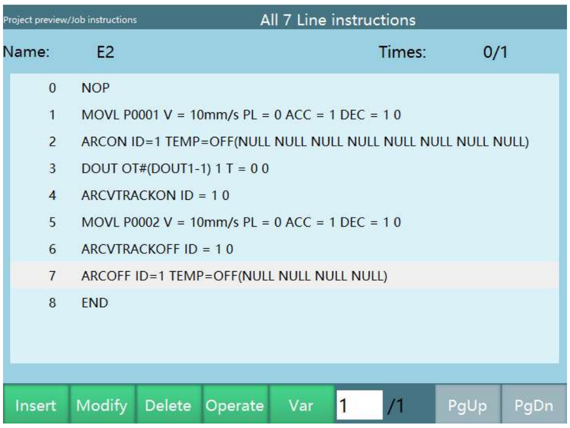
L=(average voltage-reference voltage)*correction factor

When L > maximum compensation distance per time, the compensation distance is determined by the period and the maximum compensation distance per time

When L < maximum compensation distance per time, the compensation distance is compensated according to the result calculated by the compensation length formula

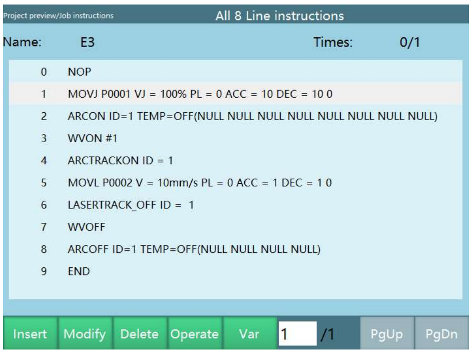
Program

Remember to set the ignition success signal in the "Process - Welding process - Welding IO" interface

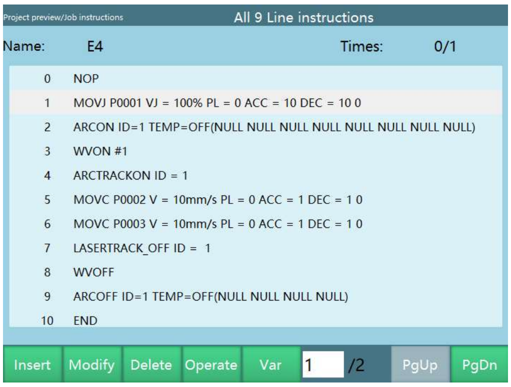


Use cases

Linear Weaving Tracking



Arc Weaving Tracking



P0001, P0002, P0003 are the 3 points on the arc respectively.

Palletizing Process

Catalogue

Palletizing Process

Usage scenarios

Robot palletizing is mainly used in food, beverage, logistics and other industries, and is also a typical example of industrial robot application, combined with different grippers, it can realize the crating and palletizing of various shapes of finished products in different industries. The value of palletizing is to palletize piles of goods in a certain pattern, so that the goods can be easily handled, unloaded and stored.

Traditionally, palletizing is done by hand, and in many cases this type of palletizing cannot be adapted to today's high-tech development. When the speed of the production line is too fast or the weight of the product is too large, manpower would be difficult to meet the requirements, and the use of manpower for palletizing requires a large number of people and high labor costs, yet it does not improve productivity.

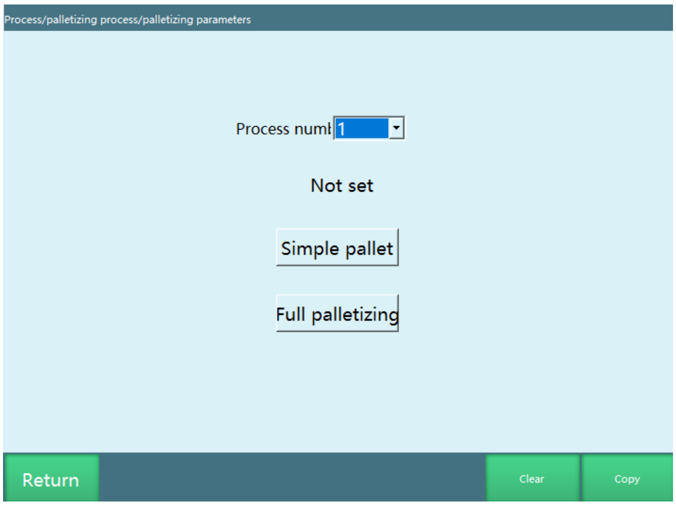
In order to improve the efficiency of handling and unloading, improve the quality of palletizing, save labor costs and ensure the personal safety of the employees, the application of palletizing robots will become more and more widespread.

Simple palletizing/Complete palletizing

Enter [Process/Palletizing process/Palletizing parameters], there are 99 process numbers in the "Process number" parameter. You can also select "Simple palletizing" or "Complete palletizing" here.

Clear parameters: Clear the parameters of complete palletizing/simple palletizing of the process number you currently selected.

Copy parameters: Copy the parameters of complete palletizing/simple palletizing of the currently selected process number to the process number you want.

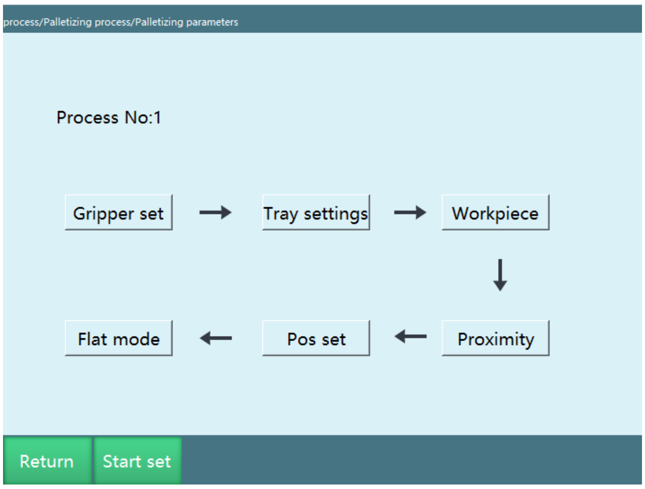


Note: It is better not to choose the same process number for simple palletizing and complete palletizing. If the process number for simple palletizing is 1, after setting the parameters, if you change the process number 1 to complete palletizing, then you have to reset the parameters for complete palletizing. Choosing different process numbers can save time and avoid errors when the program is running.

Complete palletizing

Parameter setting

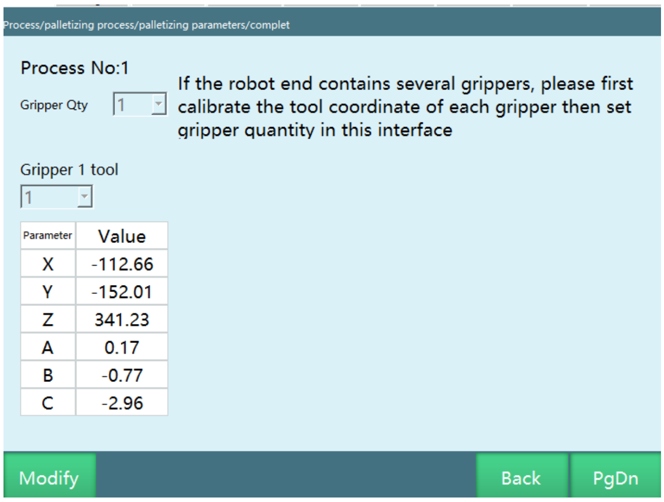
Complete palletizing process includes gripper setting, pallet setting, workpiece parameters, overlap mode, position setting and plane mode.



Gripper setting

The gripper setting is to set the tool hand used in the palletizing process.

1. If two suction cups pick up material separately (after one suction cup picked up material, switch to the other suction cup to pick up material), then set two grippers.
2. If two suction cups pick up material at the same time, then set one gripper.
3. If two suction cups discharge material separately (after one suction cup discharged material, switch to the other suction cup to discharge material), then set two grippers.
4. If two suction cups discharge material at the same time, then set one gripper.



Please go to the [Setting - Tool hand calibration] interface to calibrate the gripper (tool hand) in advance, and then set the gripper in this interface.

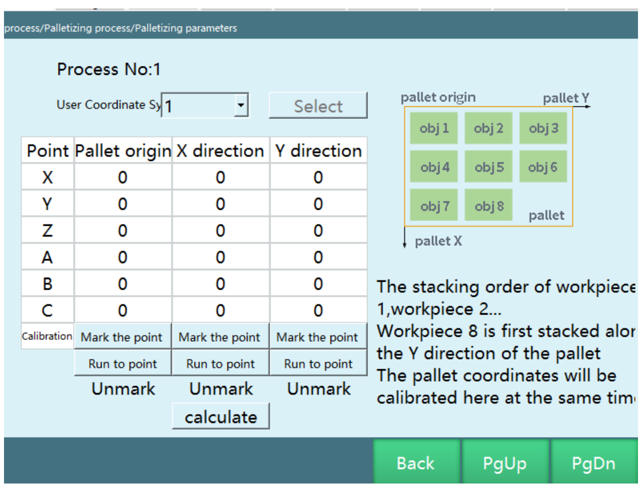
Number of grippers: The number of grippers, set it according to the actual situation, up to 4 grippers can be set.

Gripper X tool number: Set the tool hand number corresponding to the gripper, tool hand parameters need to be calibrated in advance.

Parameter value: The parameter value of each axis after selecting the calibrated tool hand number in this interface after the tool coordinates are calibrated in the tool hand calibration interface.

Pallet setting

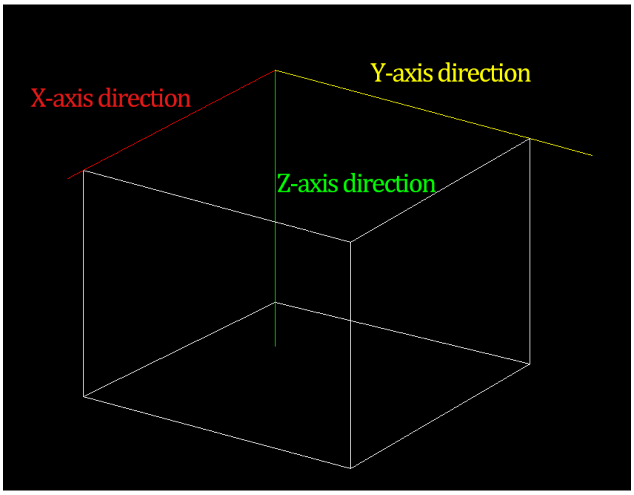
The pallet setting is to set the pallet user coordinates. The origin of the pallet, the Y direction of the pallet and the X direction of the pallet need to be calibrated in this interface.



User coordinate system: pallet coordinates, select the user coordinates to be calibrated as needed, calibrate the pallet coordinates (user coordinates), if you change the position of the coordinate system in the user coordinate calibration later, the coordinate system here will also change.

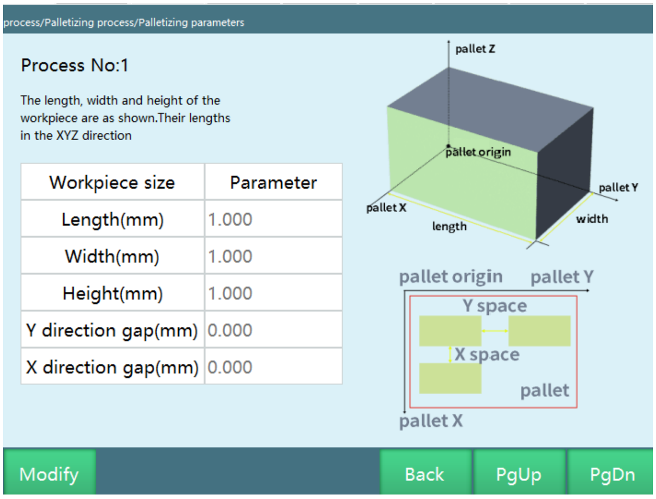
Note: Please use the tool hand selected by gripper 1 for calibration. When the user coordinates (pallet coordinates) are not calibrated, the parameter values of the user coordinates are 0, the user coordinate system will be consistent with the Cartesian coordinate system.

The X and Y directions must be marked based on the original X and Y directions of the robot, otherwise the marked Z direction of the pallet will be downwards and the second layer will be palletized downwards when palletizing.



Workpiece parameters

In the "Workpiece parameters" interface, you can set the length, width, height and clearance of the palletized workpiece under the user coordinate system.



The length, width and height are respectively the lengths in the YXZ direction under the pallet coordinate system (user coordinate system)

Workpiece size: parameter description

Length: the length of the workpiece in the Y direction under the pallet coordinate system

Width: the length of the workpiece in the X direction under the pallet coordinate system

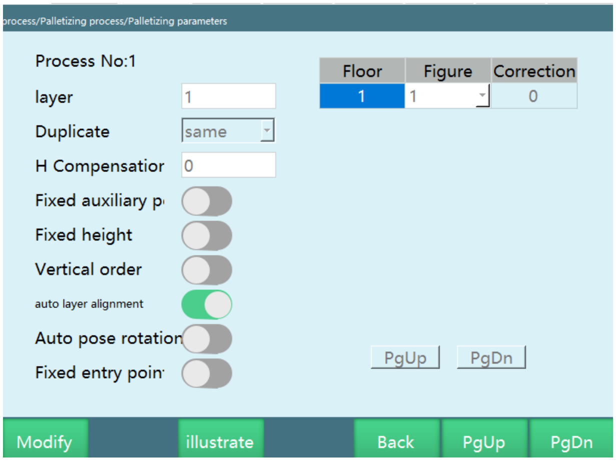
Height: the length of the workpiece in the Z direction under the pallet coordinate system

Pallet Y-direction clearance: the length of the clearance distance between two workpieces in the Y-axis direction under the pallet coordinate system

Pallet X-direction clearance: the length of the clearance distance between two workpieces in the X-axis direction under the pallet coordinate system

> Overlap mode

In the "Overlap mode" interface, you can set the palletizing layer and other related parameters, and select graphic template.



Number of layers: the total number of layers for palletizing, fill in according to actual needs

Duplicate relationship: duplicate relationship between each layer.

Select "Same": the same graphic template will be used for each layer;

Select "Alternate": alternate graphic templates for every two layers;

Select "Custom": user has to choose the graphic template used for each layer;

Same: Each layer has the same graphic template, and the same graphic template is used for palletizing. When this option is selected, only the first layer can be modified in the list on the right, and all the following layers are changed accordingly after the modification. In the following figure, the number of layers is 6, and the duplicate relationship is "same".

Floor	Figure	Correction
1	1	0
2	1	0
3	1	0
4	1	0
5	1	0
6	1	0

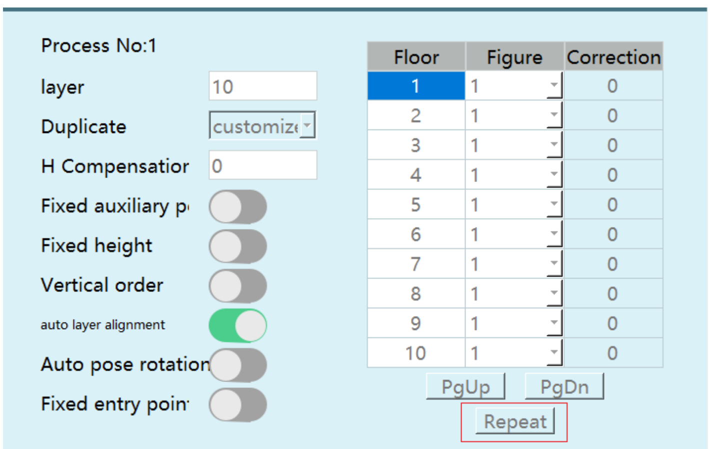
Alternate: Two graphic templates are used alternately. After selecting this option, only the first two layers in the list on the right can be modified, and all the following layers repeat the graphic number of the two layers after modification. In the following figure, the number of layers is 6, and the duplicate relationship is "alternate".

Floor	Figure	Correction
1	1	0
2	2	0
3	1	0
4	2	0
5	1	0
6	2	0

Custom: The graphic template can be set individually for each layer. In the following figure, the number of layers is 6, and the duplicate relationship is "custom".

Floor	Figure	Correction
1	1	0
2	2	0
3	3	0
4	4	0
5	5	0
6	6	0

Repeat: When the duplicate relationship is "custom" and the number of layers is large, if all layers repeat the graphic template of the previous N layers, then after filling the graphic template of the previous N layers, select the N+1th layer and click this button, the following layers will repeat the graphic template automatically.



Layer	Graphic number	Height correction
1	2	0
2	4	0
3	3	0
4	1	0
5	2	0
6	4	0
7	3	0
8	1	0
9	2	0

Layer	Graphic number	Height correction
1	7	0
2	4	0
3	3	0
4	5	0
5	1	0
6	1	0
7	1	0
8	1	0
9	1	0

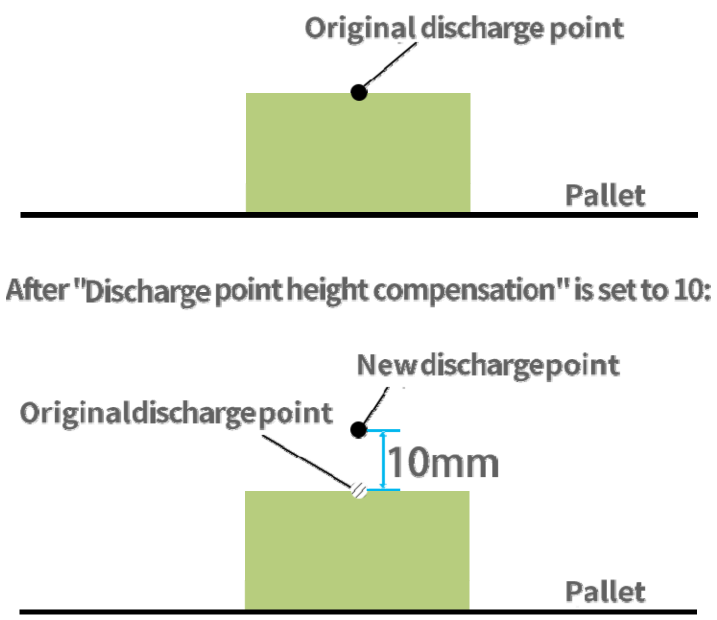
"Repeat" button →

Layer	Graphic number	Height correction
1	7	0
2	4	0
3	3	0
4	5	0
5	7	0
6	4	0
7	3	0
8	5	0
9	7	0

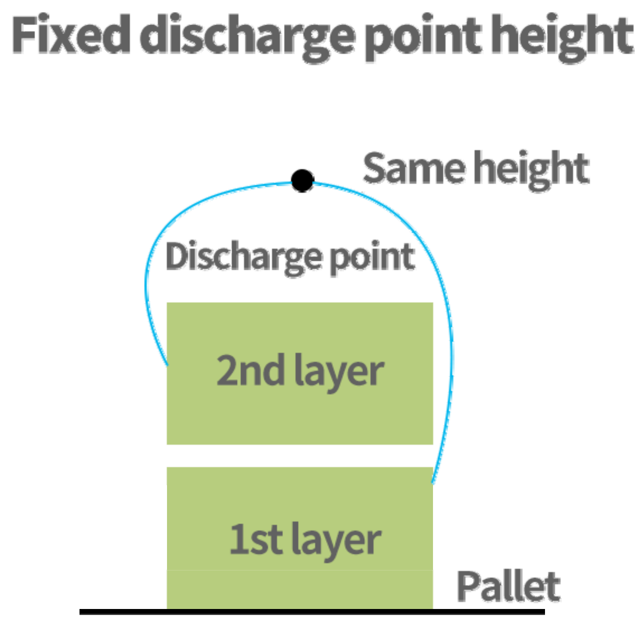
Fixed auxiliary point height: If the number of layers of palletizing is two, when this button is turned on, the auxiliary point of the first layer of workpieces and the second layer of workpieces is the same point, and the auxiliary point will not be offset in the Z+ direction when operating the second layer of workpieces.

For example, you need to palletize the workpieces in a closed box, if the number of workpieces to be palletized is large, the height of workpiece auxiliary point set when each workpiece is palletized will increase with the number of workpieces palletized, which will probably exceed the limits of the robot's joint parameters. In order to prevent such errors, we can turn on the "Fixed auxiliary point height" button and set the fixed auxiliary point height position, so that as the number of palletized workpieces increases, the auxiliary point will always be at the same height position, so that the limits of the joint parameters will not be exceeded, and the safety of the operator will be guaranteed.

Discharge point height compensation: After filling in, the height of the discharge point of all workpieces will be offset, and the height offset can be filled in according to actual needs. If the value is positive, it will be offset in the Z+ direction, and if the value is negative, it will be offset in the Z- direction (this parameter is invalid when depalletizing)

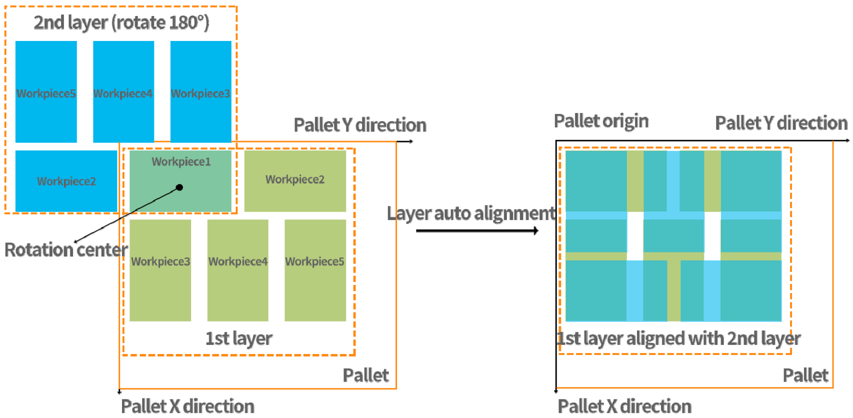


Fixed discharge point height: When selected, the height of the discharge point is the same for each layer of palletizing, and the height is the marked workpiece point height (only valid when palletizing).If the number of layers of palletizing is greater than one (for example, two layers), when this button is turned on, the workpiece point of each workpiece on each layer will be at the same height, and the height of the workpiece point will not be offset in the Z+ direction during palletizing the second layer.

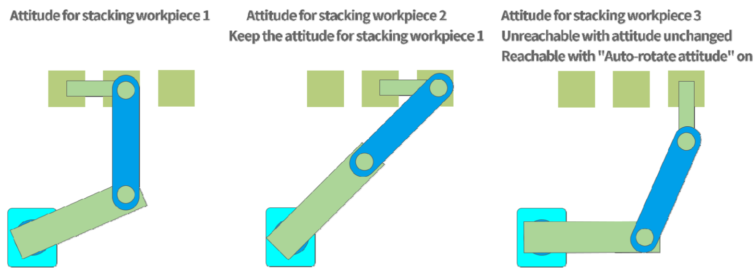


Vertical alignment: After selecting "Vertical alignment", you will first palletize a vertical column and then palletize the next vertical column.

Layer auto alignment: If "Layer auto alignment" is selected, the template of each layer will be automatically aligned, and the X-axis and Y-axis offsets will be calculated automatically.



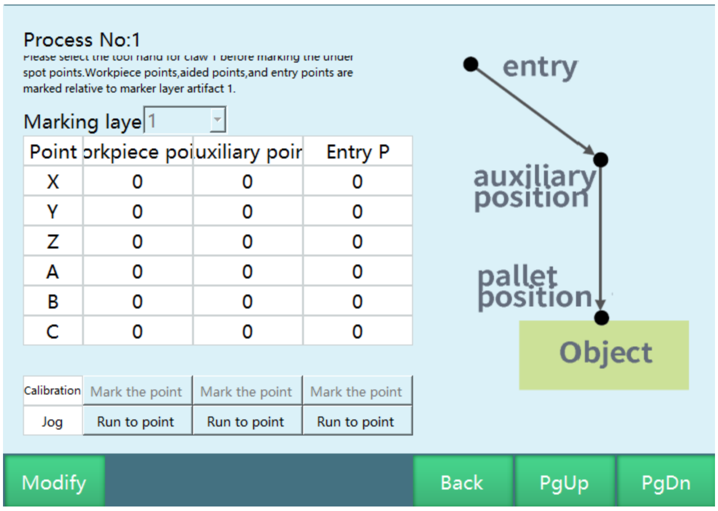
Auto-rotate attitude: When selected, if the tool hand cannot reach the auxiliary point and the discharge point with an inherent attitude during palletizing, but can reach by rotating the attitude of the tool hand, then it will automatically rotate. This function is only available when both auxiliary and workpiece points use joint interpolation.



Fixed entry point position: After picking, each workpiece will be palletized with the same entry point, and the Z axis will be optimized during entering the entry point

Position setting

In the "Position setting" interface, you can set the palletizing workpiece point, auxiliary point and entry point, please use the tool hand set on the "Gripper setting" interface to mark the position.



Marked layer number: which layer is the current calibrated workpiece point on; eliminating the need to clear the stacks and allowing you to directly choose to mark the current layer.

Workpiece point: the first pickup point or the last discharge point of the marked layer

Note: The order of the workpieces in the palletizing parameter setting interface is the palletizing order, while the depalletizing order is the opposite.

Auxiliary point: used in conjunction with the workpiece point, so that the workpiece can be placed to the workpiece point more safely, it is generally set above the workpiece point, if the workpiece needs to rotate some angle, it will rotate before reaching the auxiliary point, which will follow the placement position of the workpiece and make automatic offset.

Entry point: the entry point of the pallet. To prevent the robot from colliding with other objects, try to set the safe position of the robot as the entry point, which will follow the placement position of the workpiece and make automatic offset in the Z-axis direction. The PALENTER instruction turns on XYZ optimization, and the ABC axis coordinates change when running to the entry point

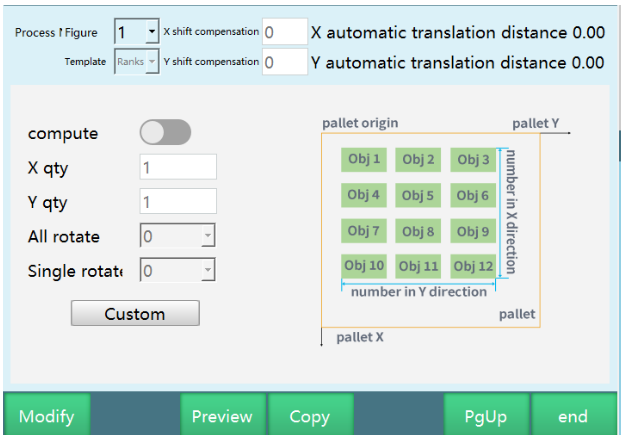
Mark this point: Click "Mark this point" after the robot has moved to the position.

Run to this point: The workpiece can only move to the marked point after clicking "Save" to save the marked values, if not saved, the workpiece will move to the previously marked point, and to move to the marked point, you need to click this button after pressing the DEADMAN button.

Note: Please use the tool hand set on the "Gripper setting" interface to mark the position.

Plane mode

In the "Plane mode" interface, you can set the graphic templates for palletizing.



Graphic number: the number of the graphic template

Template selection: There are 4 fixed graphic templates (row-column, criss-cross, hollow square, five-flower stack),you can also select "custom" to customize graphic template.

X translation compensation: offset of the overall graphic template relative to the original palletizing position on the X-axis of the pallet coordinate system

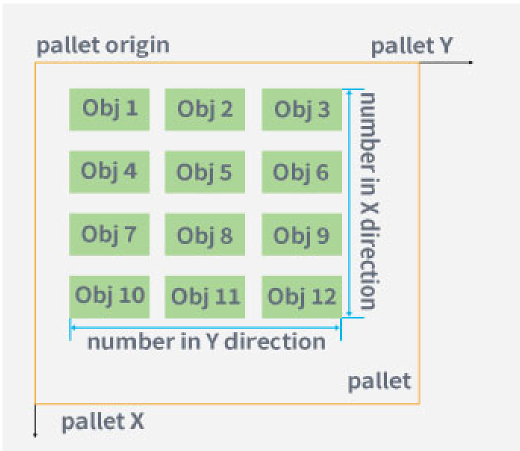
Y translation compensation: offset of the overall graphic template relative to the original palletizing position on the Y-axis of the pallet coordinate system

Automatic calculation: According to the pallet setting interface, the calibrated user coordinate system generates a rectangular pallet, and automatically calculates how many workpieces can be put in X direction and how many workpieces can be put in Y direction according to the workpiece size.

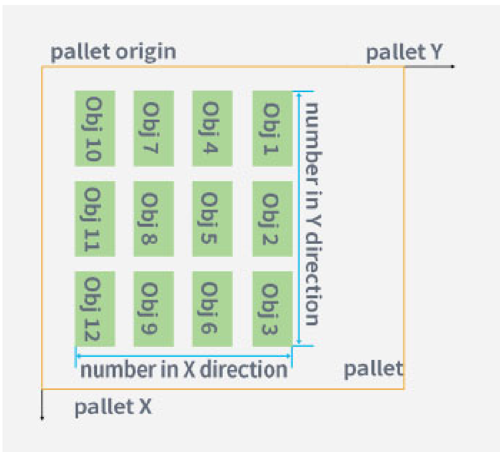
Number in X direction (row-column template, criss-cross template): the number of workpieces in X direction (criss-cross template: number of the workpieces with long side on X axis)

Number in Y direction (row-column template, criss-cross template**):** the number of workpieces in Y direction (criss-cross template: number of the workpieces with long side on Y axis)

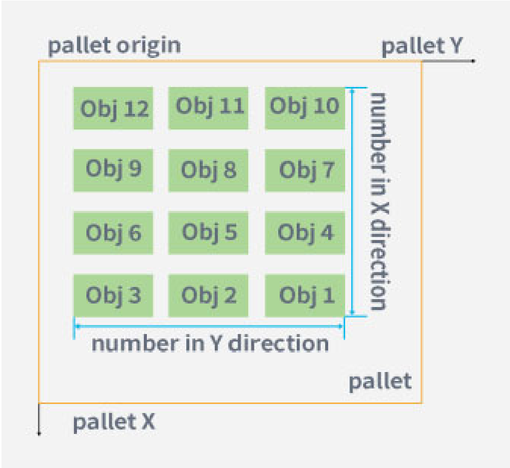
Overall rotation angle (row-column template, criss-cross template, hollow square template**):** the angle that the whole rotates clockwise around the first workpiece point: 0°, 90°, 180°or -90°



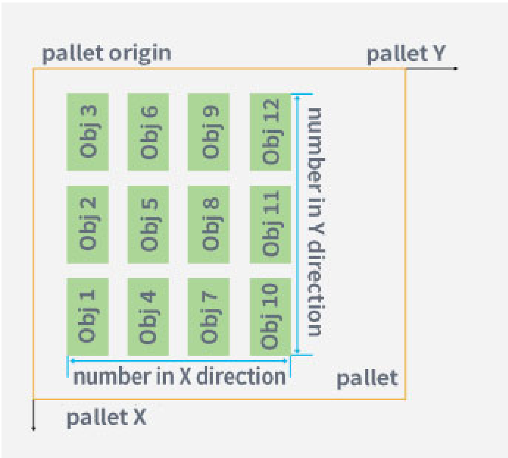
(overall rotation angle: 0°)



(overall rotation angle: 90°)



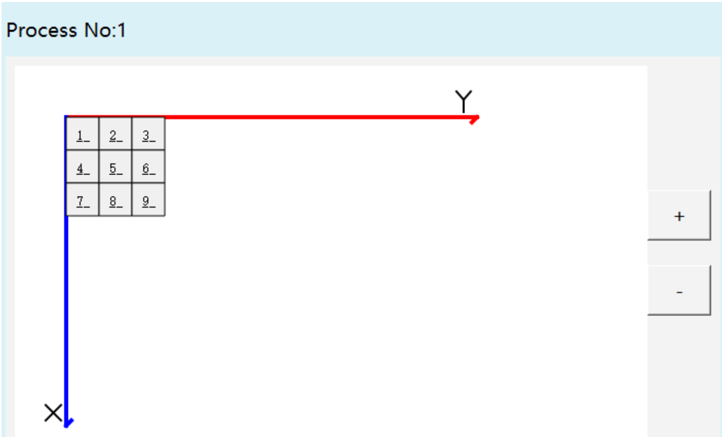
(overall rotation angle: 180°)



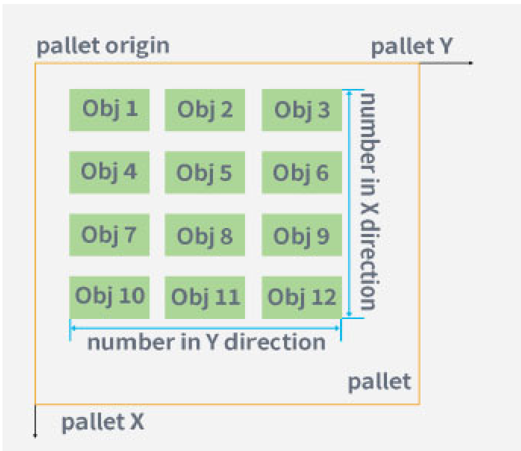
(overall rotation angle: -90°)

Workpiece rotation angle (row-column template, criss-cross template, hollow square template, five-flower template): the angle that all workpieces in the graphic template rotate clockwise: 0 °, 90 °, 180 ° or -90 °

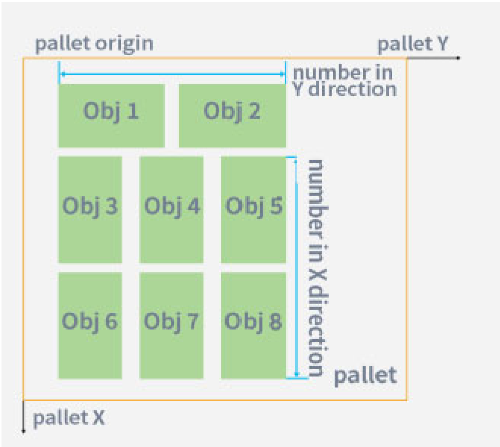
Preview: Preview the set graphic templates, which can be used to check whether the graphic template is set correctly, here we select the criss-cross template, the number of workpieces in X direction is 2, and the number of workpieces in Y direction is 3



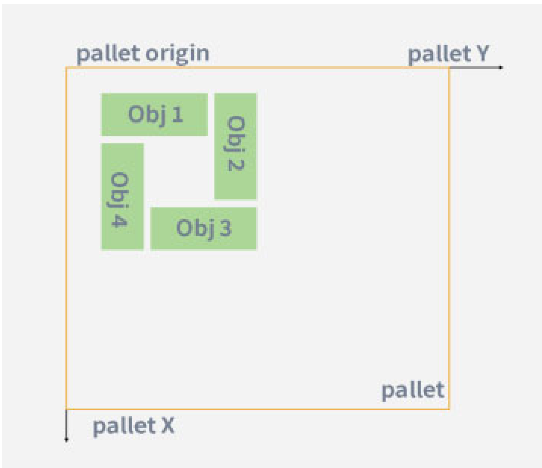
Row-column: the direction of the workpieces on the whole layer of the graphic template is the same, and the workpieces are palletized in sequence. As shown in the figure below, the number of workpieces in the X direction is 4, and the number of workpieces in the Y direction is 3



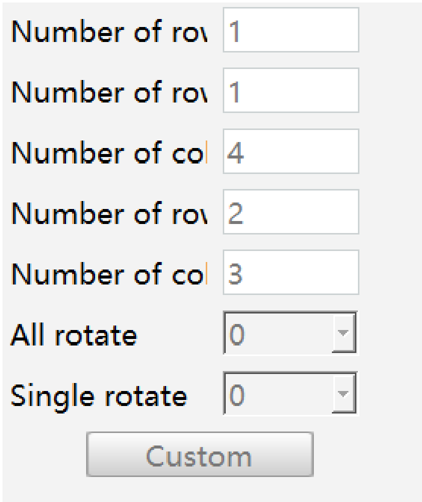
Criss-cross: the direction of the workpieces can be horizontal or vertical,the workpieces are arranged in a crisscross pattern (in this template, the number of workpieces in X direction is the number of the workpieces with long side on X axis, the number of workpieces in Y direction is the number of workpieces with long side on Y axis)

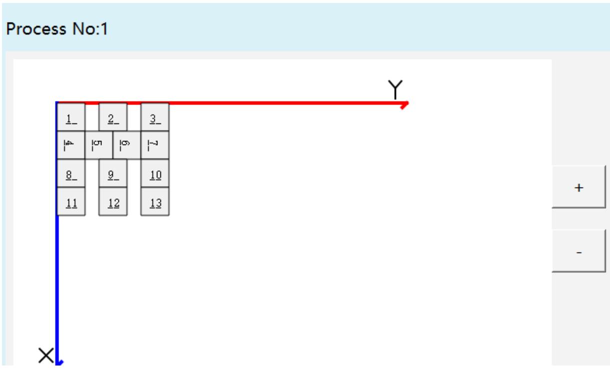


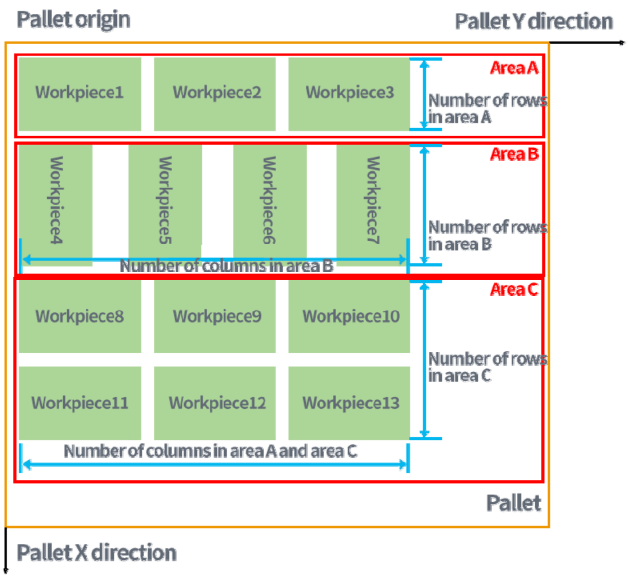
Hollow square: 4 workpieces on one layer, arranged in the form of a hollow square (the second workpiece rotates 90 degrees clockwise from the first workpiece, the third workpiece rotates 180 degrees clockwise from the first workpiece, and the fourth workpiece rotates 90 degrees counterclockwise from the first workpiece)



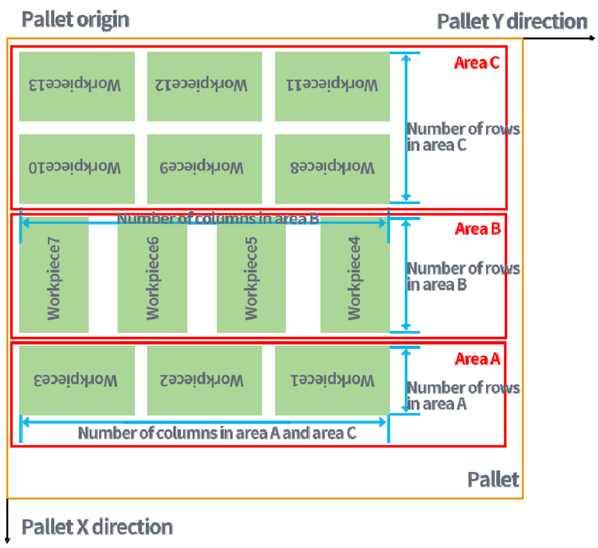
Five-flower stack: The workpieces are divided into three areas: area A, area B, area C; the number of columns in area A and area C can be set together, and the number of columns in area B can be set separately (as shown in the figure, workpieces 4-7 in area B are rotated clockwise by 90 degrees compared to workpieces 1-3 in area A and workpieces 8-13 in area C). The workpieces in area A and area C are left and right aligned with area B with the highest number of columns in the stack, as shown in Figure 1.



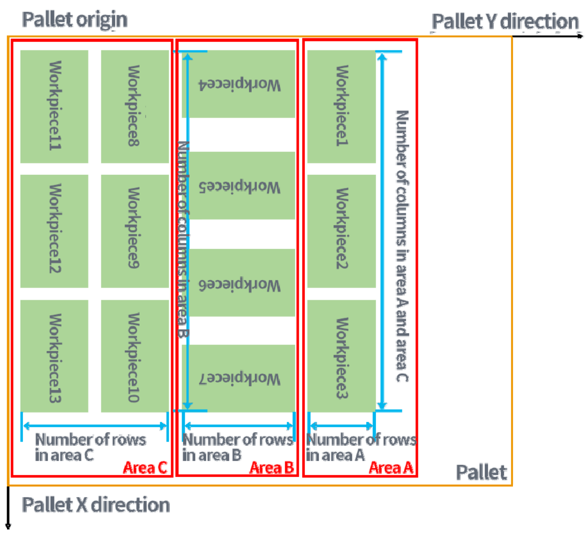




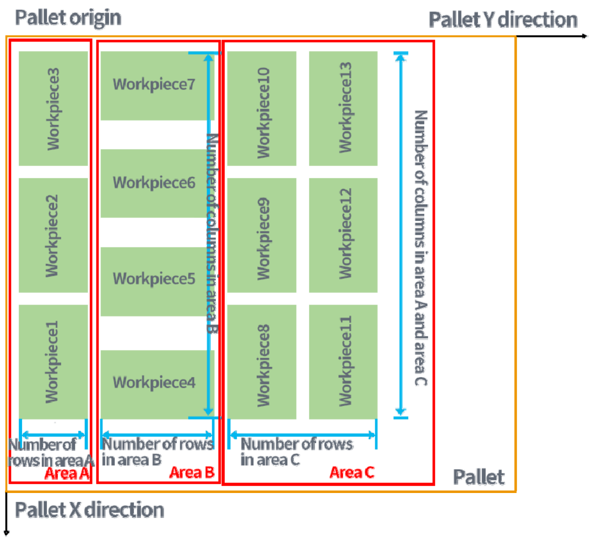
overall rotation angle: 0°



overall rotation angle: 180°

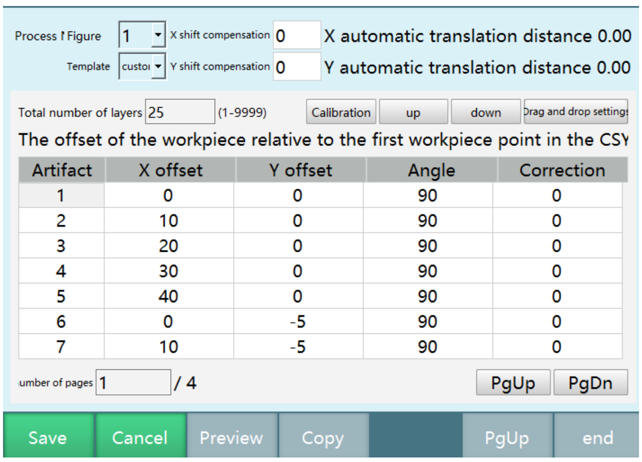


overall rotation angle: 90°



overall rotation angle: -90°

Custom: Customize the graphic template

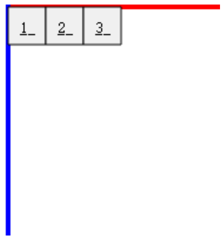


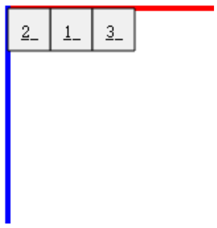
Artifact	X offset	Y offset	Angle	Correction
1	0	0	90	0
2	10	0	90	0
3	20	0	90	0
4	30	0	90	0
5	40	0	90	0
6	0	-5	90	0
7	10	-5	90	0

Total number of layer workpieces: total number of palletized workpieces, set according to actual needs. Note: Modifying the total number of workpieces will clear all workpiece parameters.

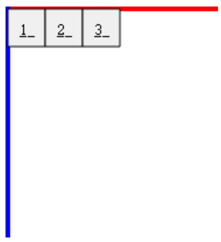
Calibration: You can set the workpiece point of palletizing by yourself, determine the point and click on the "Calibration" button. If you want to modify the workpiece point of palletizing set for the first time, you can click on the number of the workpiece, move the robot to the position you want and click on the "Calibration" button to complete the modification of workpiece point.

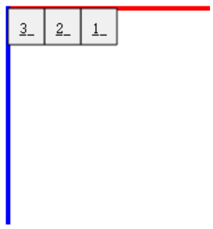
Up: After the workpiece point calibration is completed, if you want to set the position of palletized workpiece 2 to the position of palletized workpiece 1, you can click on the "Up" button, so that the position of workpiece 1 is changed. Here we take two workpieces as an example.





Down: After the workpiece point calibration is completed, if you want to set the position of palletized workpiece 1 to the position of palletized workpiece 3, you can click on the "Down" button, so that the position of workpiece 3 is changed.





X offset: the offset of the workpiece point on the X axis

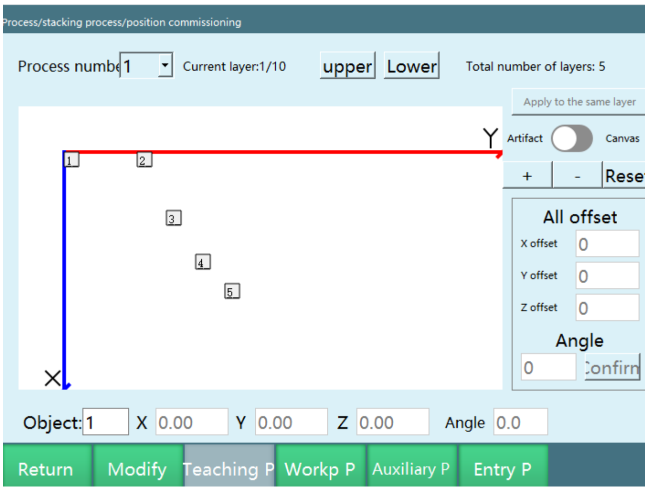
Y offset: the offset of the workpiece point on the Y axis

Rotation angle: the rotation angle of the workpiece relative to the angle of the first workpiece point

Height correction: After filling in, the height of the workpiece point, auxiliary point and entry point will be offset when the workpiece is palletized. If the value is positive, it will be offset in the Z+ direction, and if the value is negative, it will be offset in the Z- direction, we can correct the height of the workpiece point, auxiliary point and entry point

Dragging Setting: After setting the number of palletized workpieces, click the "Dragging Setting" button, as shown in the figure, you can drag the workpiece point to any position you want. In the custom template, in addition to filling in the XY offset, you can also drag the workpiece directly

Note: Before entering the "Dragging setting", set the total number of layer workpieces in the custom template, click "Save", and then click "Modify - Dragging setting"; after the dragging setting is finished, click the "Save" button in the "Dragging setting" first, return to the "Custom" interface, and click "Save" again.



Increase: increase the number of workpieces according to your needs

Reduce: reduce the number of workpieces

Workpiece/Canvas: drag the workpiece when the button is off, drag the canvas when the button is on

Reset: reset the canvas

Screen+: zoom in

Screen-: zoom out

Single/Whole: the offset of single workpiece/the whole in the X or Y direction; turn on the "Single/Whole" button to offset all workpieces in X or Y direction, and turn off the "Single/Whole" button to offset your currently selected workpiece in X or Y direction

X+/X-: offset step value in the positive or negative direction of X as a whole

Y+/Y-: offset step value in the positive or negative direction of Y as a whole

Step: the offset of the workpiece in the X or Y direction

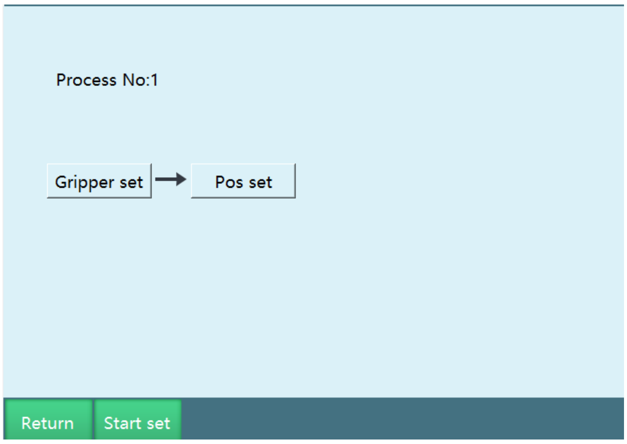
Angle: set the angle of each rotation; if the "Single/Whole" button is turned on, all workpieces will be rotated; if the "Single/Whole" button is turned off, you can rotate the workpiece you currently selected.

Forward/Reverse: the workpiece rotates by itself to set the angle value

Simple palletizing

Parameter setting

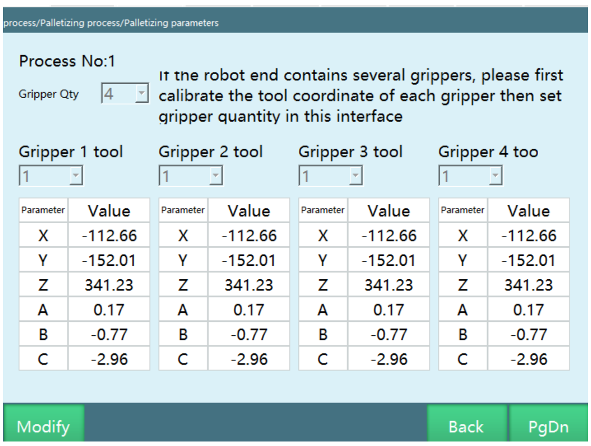
Simple palletizing parameter setting process includes gripper setting and position setting



Current type of use: when simple palletizing and complete palletizing share the same palletizing number, you need to set the correct type before use

Gripper setting

In the "Gripper setting" interface, you can choose the gripper (tool hand) for palletizing, please go to [Setting - Tool hand calibration] interface to calibrate the gripper (tool hand) first, and then set the gripper in this interface.



Number of grippers: the number of grippers, set according to the actual situation

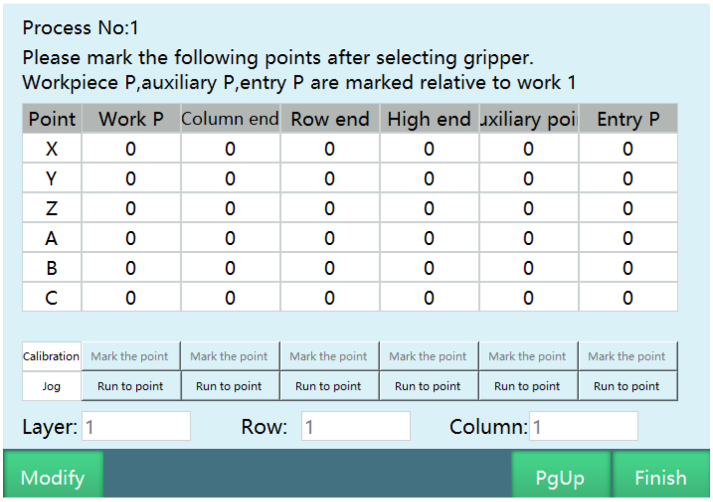
Gripper X tool number: set the tool hand number corresponding to the gripper, tool hand parameters need to be calibrated in advance

Parameter value: the parameter value is the offset of the end of the tool hand, which can only be selected here, but not calibrated

Position setting

Simple palletizing only supports row-column template. All palletizing directions and position points are marked, even if the marked pattern is not a rectangular palletizing pattern, it will also follow the marked directions during palletizing.

In the simple palletizing process, you only need to set the palletizing grippers and mark 6 position points. The settings of the grippers are the same as for complete palletizing. If the palletizing has more than one gripper for picking and palletizing separately, please mark the position points with the first gripper, the actions of the other grippers will be calculated automatically



Starting workpiece point: the position point of the first workpiece when palletizing.

End of column: the position point of the last workpiece in the column (user coordinate X axis) direction when palletizing.

End of row: the position point of the last workpiece in the row (user coordinate Y axis) direction when palletizing.

Height end: the position point of the first workpiece on the last layer when palletizing.

Auxiliary point: the auxiliary point of palletizing, it is recommended to set it above the starting workpiece point.

Entry point: the entry point of palletizing, it is recommended to set it to a safety point outside the pallet.

Number of layers: total number of layers of palletizing.

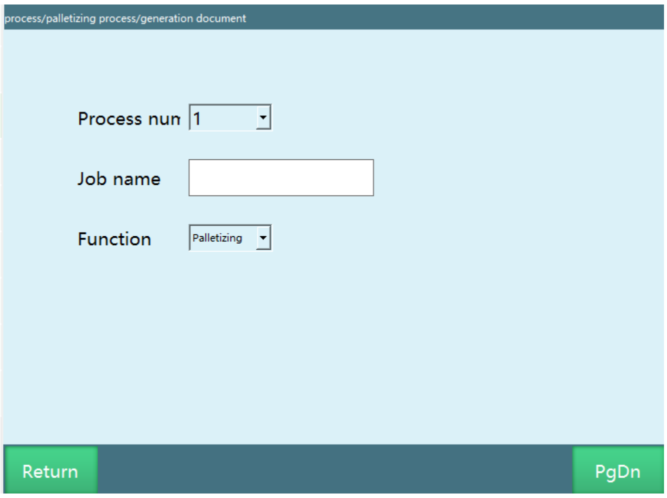
Number of rows: total number of rows of palletizing.

Number of columns: total number of columns of palletizing.

Generating file

Standard palletizing and depalletizing procedures can be generated using the "generating file" function, and the parameters of the process number need to be set in advance

Simple palletizing's "generating file" function is forbidden when there is no IO

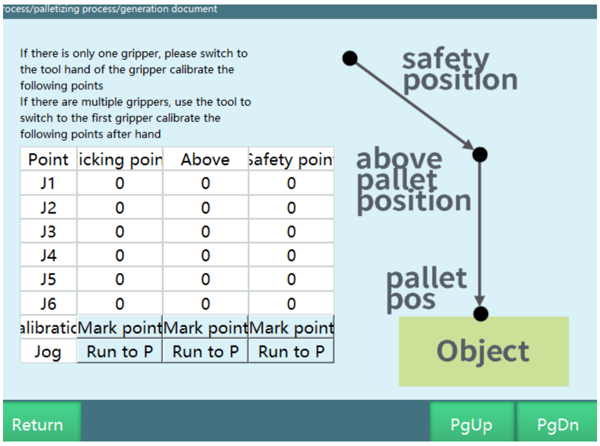


Current type of use: when simple palletizing and complete palletizing share the same palletizing number, you need to set the correct type before use

Process number: after selecting the process number, it is necessary to confirm whether the current palletizing type of use is the set one

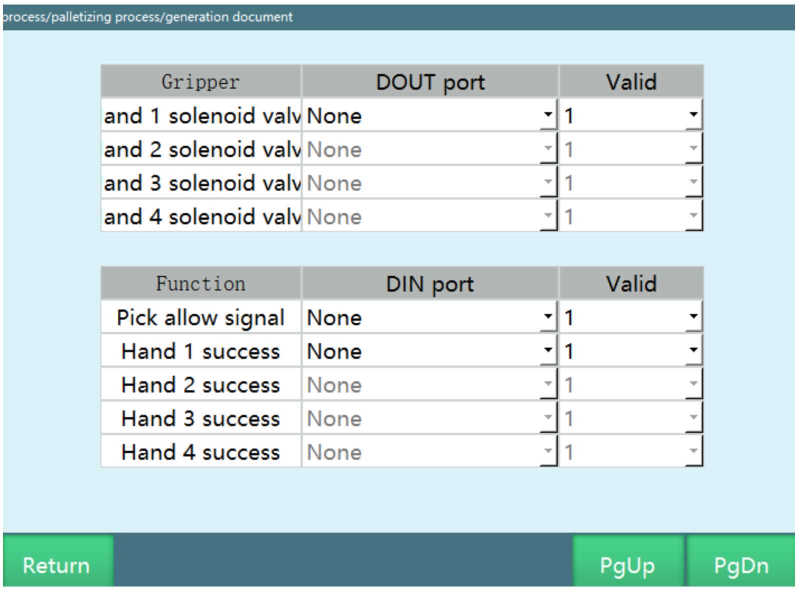
Program name: need to start with English letters (Pure Chinese is also OK)

Function: palletizing, depalletizing



Mark this point: teach to the corresponding point and click "Mark this point"

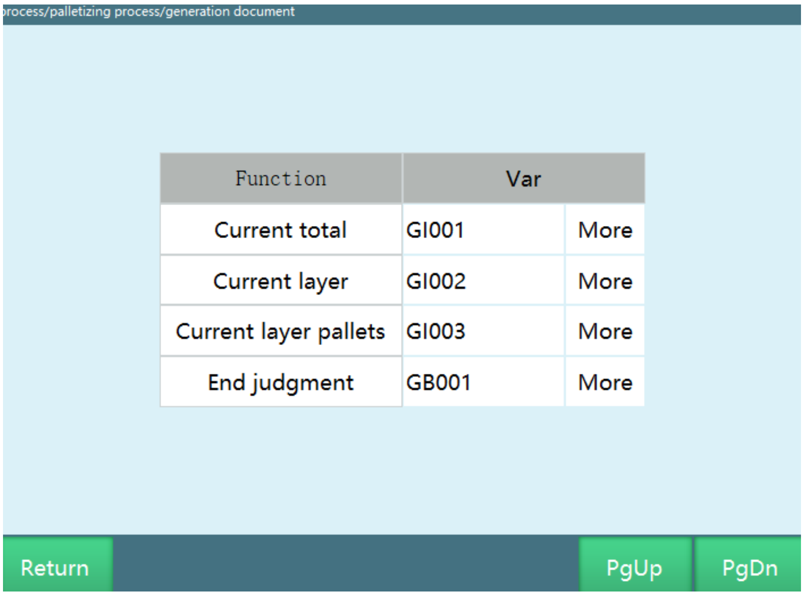
Run to this point: Click "Run to this point" to verify whether the point teaching is correct



Gripper solenoid valve: gripper IO output signal; the DOUT port of the gripper solenoid valve must be set, the "generating file" function only supports 1 output, if multiple outputs are required, you can return to the project after the job file is generated to modify the output signal instruction

Picking permission signal: wait for picking permission signal before palletizing picking, select according to your situation

Gripper picking successful signal: the signal to judge whether the gripper picking is successful



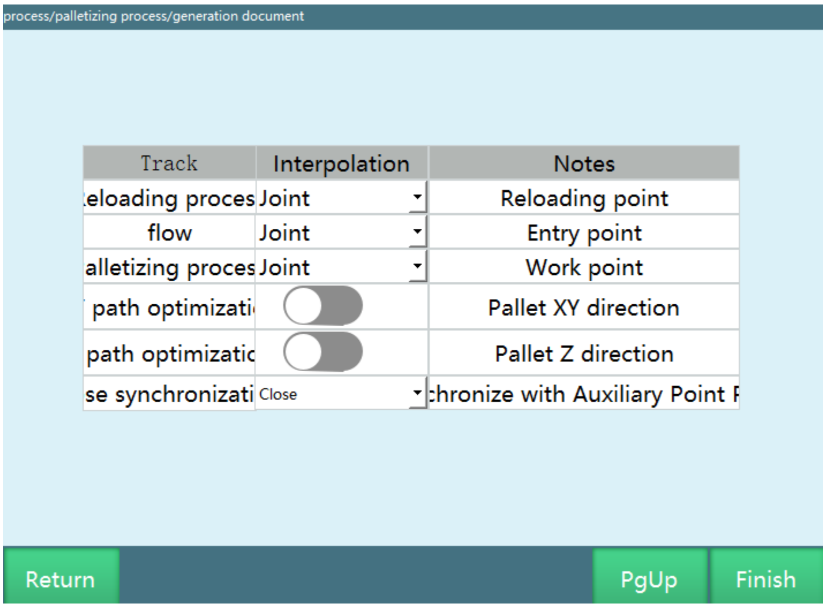
Note: For the "Current total number of palletized workpieces", "Current palletizing layer" and "Number of palletized workpieces on current layer" functions, if you do not select the variable type when generating the job file, we can check it in the [Process - Palletizing process] interface when executing the palletizing program.

Current total number of palletized workpieces: Cache the value of the "current total number of palletized workpieces" variable into the set variable

Current palletizing layer: Cache the value of the "Current palletizing layer" variable into the set variable

Number of palletized workpieces on current layer: Cache the value of the "Number of palletized workpieces on current layer" variable into the set variable

Palletizing end judgment: After palletizing is completed, change the variable value to jump out of the while loop



Pickup process: above the pickup point - pickup point; the interpolation method can be changed to joint interpolation or linear interpolation.

Intermediate process: above the pickup point - palletizing entry point; the interpolation method can be changed to joint interpolation or linear interpolation.

Palletizing process: palletizing entry point - palletizing auxiliary point - workpiece point; the interpolation method can be changed to joint interpolation or linear interpolation.

XY path optimization: Click to turn on to optimize the path of the pallet in XY direction

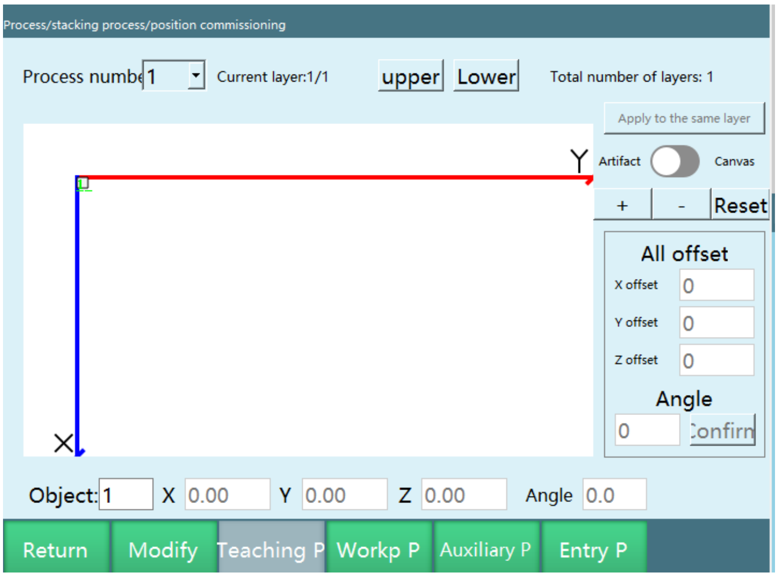
Z path optimization: Click to turn on to optimize the path of the pallet in Z direction

Attitude synchronization:

- Off: Attitude is executed according to the set point
- Auxiliary point attitude synchronization: Move to the entry point with the attitude of the auxiliary point
- Automatic attitude calculation: Calculate the trajectory attitude between the entry point, auxiliary point and the workpiece point according to the ratio of the distance from the entry point to the auxiliary point and the distance from the auxiliary point to the workpiece point automatically. The C attitude during the movement is always rotating. For example, the ratio of the

distance from the entry point to the auxiliary point and the auxiliary point to the workpiece point is 2:8, and the C attitude from the entry point to the workpiece point has rotated 100° in total, then 1 will rotate 20° from the workpiece point to the auxiliary point, and the remaining 80° will be rotated between the auxiliary point and the workpiece point.

Position debugging



Process number: the process number corresponding to the current parameter

Total number of layer workpieces: total number of workpieces on current layer

Upper layer: switch to the upper layer

Lower layer: switch to the lower layer

Workpiece/Canvas: drag the workpiece when the button is off, drag the canvas when the button is on

Reset: reset the canvas

Screen+: zoom in

Screen-: zoom out

Overall offset: workpiece overall offset

X offset: offset step value in the positive or negative direction of X as a whole

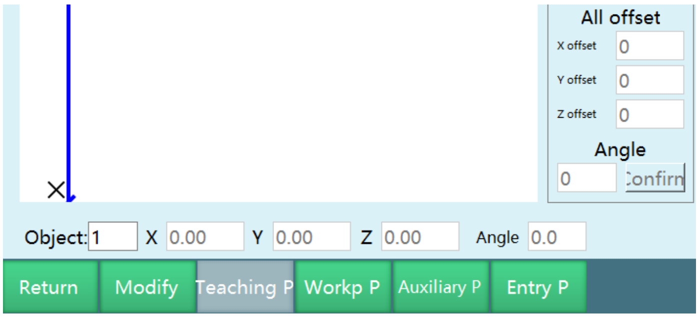
Y offset: offset step value in the positive or negative direction of Y as a whole

Z offset: offset step value in the positive or negative direction of Z as a whole

Angle: set the angle of each rotation

Each workpiece angle: overall offset angle

Apply to same layer: apply the parameters set for the current layer to the layers with the same graphic number



Current workpiece: 1 indicates the number of the workpiece

X indicates positive or negative offset of the workpiece to the X axis

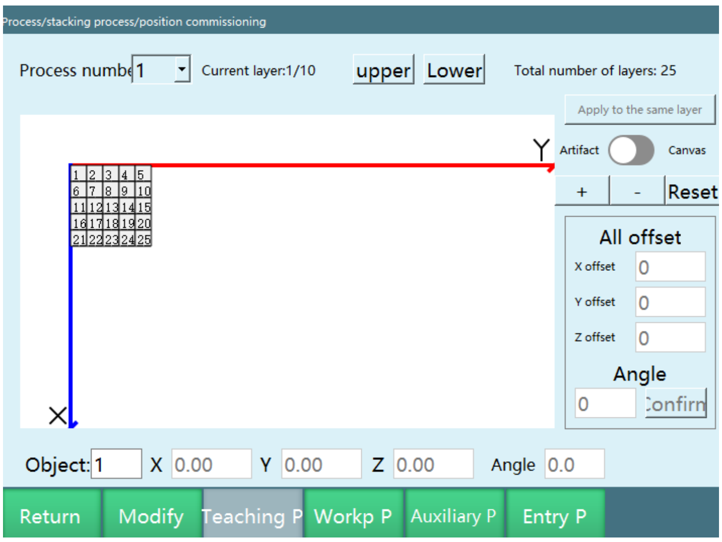
Y indicates positive or negative offset of the workpiece to the Y axis

Z indicates positive or negative offset of the workpiece to the Z axis

Angle indicates the degrees currently selected for the workpiece

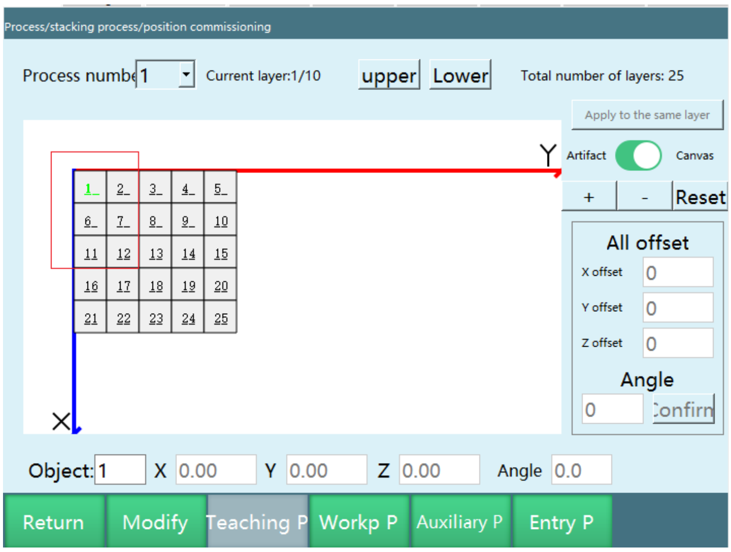
Workpiece/Canvas function demo

1. Turn on the button to drag the entire canvas, but this does not affect the coordinates of the workpiece.



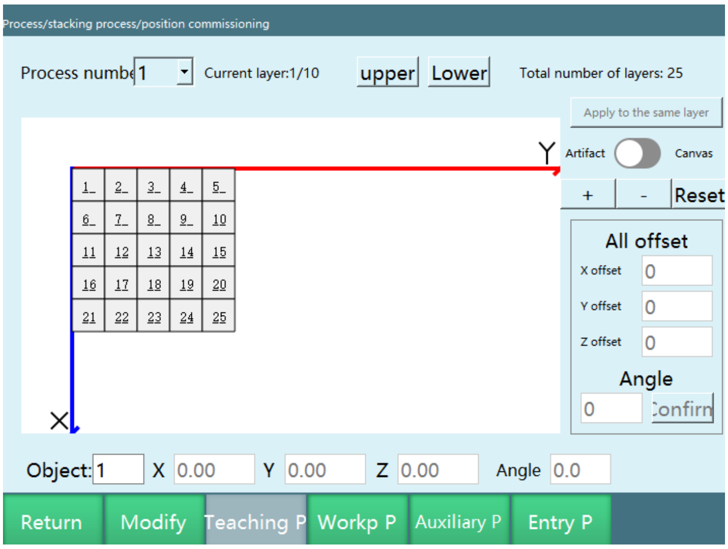
Reset function demo

2. After clicking the "Reset" button, the canvas returns to the initial position, but the canvas size is not reset

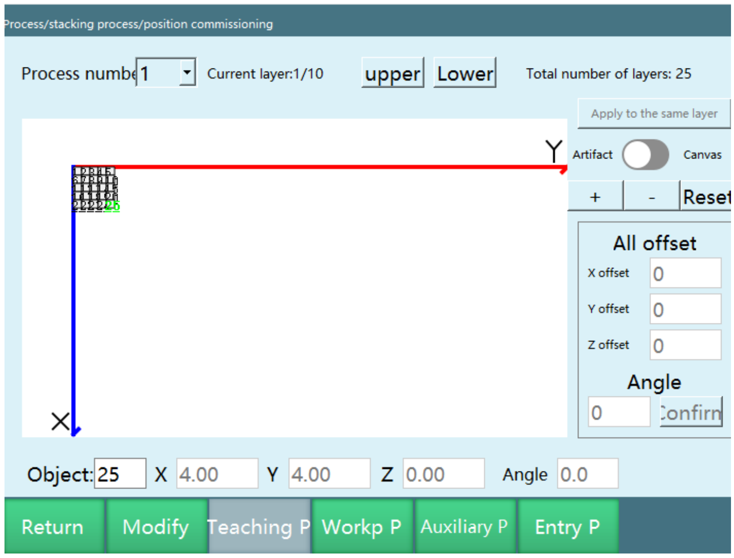


Screen+/Screen- function demo

3. Click "Screen+", the canvas zooms in

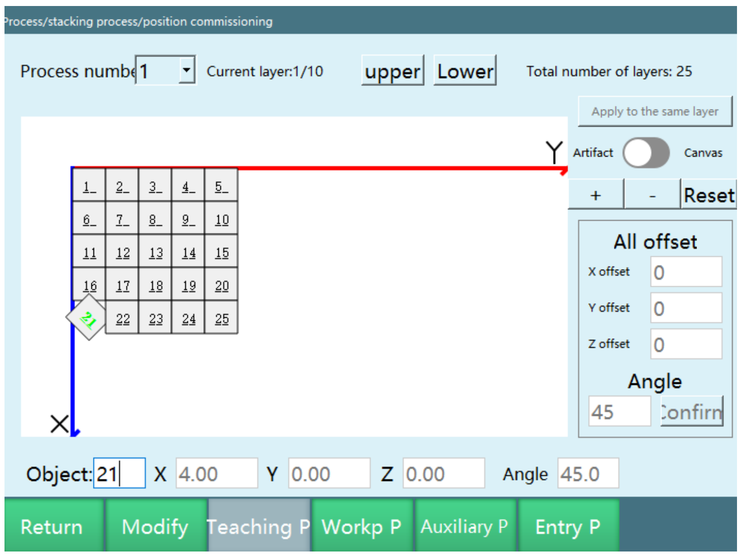


4. Click "Screen-", the canvas zooms out



Single/Whole offset function demo

5. Fill in the coordinate values, ie the value of X offset, Y offset, Z offset and angle (because this is a plane, so we can not see the effect on the Z axis), for example, fill in 26, 10, 1 and 45 respectively

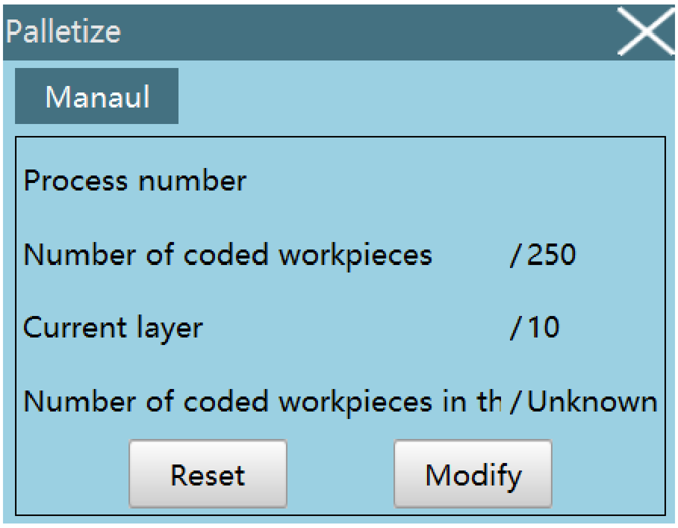


Palletizing status

The palletizing status can be used to check the current palletizing status, if the palletizing needs to start in the middle, it can be achieved by setting the number of layer and the number of workpiece to be palletized.

The number of palletized workpieces will be cleared after the controller restarts, but will not be cleared after re-running

You can view the palletizing status in [Status bar - Process - Palletizing]. There are two methods: you can select the palletizing process from the process selection bar on the "Operation parameters" interface (If palletizing is selected on the "Operation parameters", the palletizing process will always be displayed by default); you can also directly select palletizing from the process navigation bar on the teach pendant.



Process number: the process number of palletizing

Number of palletized workpieces: number of palletized workpieces/total number of workpieces

Number of current layer: the number of the layer currently being palletized/total number of layers (if the palletizing needs to start in the middle, set the number of layer to be palletized).Here we take two-layer palletizing as an example. When the robot is palletizing the first layer of workpieces, you can set the number of current layer to 2, then the robot will palletize the second layer of workpieces, and the display of the number of palletized workpieces will change.

Number of palletized workpieces on current layer: number of palletized workpieces on current layer/total number of workpieces on current layer (if the palletizing needs to start in the middle, set the number of workpiece to be palletized).For example, if you are palletizing the 3rd workpiece and want to start palletizing from the 7th workpiece, you can modify the number of palletized workpieces on current layer and start palletizing from the workpiece as you need.

Reset: clear the recorded palletizing data

Modify: when the robot is palletizing, click "Modify" to modify the current number of layer and the number of palletized workpieces on current layer.

Palletizing instructions

PALON (start palletizing)

Function: palletizing start judgement

Process number: 1-99

Type: palletizing, depalletizing

Current total number of palletized workpieces: Cache the value of the "Current total number of palletized workpieces" variable into the set variable

Note: You can control the number of layer and workpiece to be palletized by modifying the variables

Current palletizing layer: Cache the value of the "current palletizing layer" variable into the set variable

Note: You can control the number of layer and workpiece to be palletized by reading the variables

Number of palletized workpieces on current layer: Cache the value of the "Number of palletized workpieces on current layer" variable into the set variable

Note: You can control the number of layer and workpiece to be palletized by reading the variables

Example: PALON ID=1 TYPE=0 [variable name][variable name][variable name]

Note: The 3 count variables of the PALON (start palletizing) instruction will be written directly to the configuration, without the need to use the FORCESET (write file) instruction

PALGRIPPER (switch grippers)

Function: Select gripper

Process number: 1-99

Gripper: gripper 1, gripper 2, gripper 3, gripper 4

Example: PALGRIPPER ID=1 GRIPPERS=1

PALENTER (palletizing entry point)

Function: palletizing entry point

Process number: 1-99

Interpolation method: joint interpolation, linear interpolation, circular interpolation

Joint interpolation: The robot will move to the point by joint interpolation

Linear interpolation: The robot will move to the point by linear interpolation

Circular interpolation: The robot will form a circular path with two other points (previous point: MOVJ/MOVL; next point: MOVC)

VJ: speed range 2-9999

PL: position level 0-5

ACC: acceleration range 1-100

DEC: deceleration range 1-100

XY optimization: Optimize XY axis motion path

Z optimization: Optimize Z axis motion path, need to insert a fixed point before palletizing

When the entry point is lower than the fixed point, the entry point will be in the same line with the fixed point and the auxiliary point in height (the same line in side view, not the same line in top view, XY axis is unchanged)

When the entry point is between the fixed point and the auxiliary point, the height of entry point remains unchanged

When the entry point is above the fixed point and auxiliary point, the height of entry point will be optimized to the level of the fixed point

When the entry point and the auxiliary point are both higher than the fixed point, the height of entry point will be optimized to the level of the auxiliary point

Attitude: Off: Attitude is executed according to the set point

Auxiliary point attitude synchronization: Move to the entry point with the attitude of the auxiliary point

Automatic attitude calculation: There is an attitude before and after the entry point, and the entry point attitude will be calculated between these two attitudes

TIME: Time, the range is a non-negative integer, and the unit is ms. Early execution time of the next instruction.

Example: PALENTER ID=1 MOVJ VJ=10% PL=0 ACC=20 DEC=20 OFF OFF OFF 0

PALSHIFT (palletizing auxiliary point)

Function: palletizing auxiliary point

Process number: 1-99

Interpolation method: joint interpolation, linear interpolation, circular interpolation

Joint interpolation: The robot will move to the point by joint interpolation

Linear interpolation: The robot will move to the point by linear interpolation

Circular interpolation: The robot will form a circular path with two other points (previous point: MOVJ/MOVL; next point: MOVC)

VJ: speed

PL: position level 0-5

ACC: acceleration range 1-100

DEC: deceleration range 1-100

TIME: Time, the range is a non-negative integer, and the unit is ms. Early execution time of the next instruction.

Example: PALSHIFT ID=2 MOVJ VJ=30% PL=2 ACC=20 DEC=20

PALREAL (palletizing workpiece point)

Function: palletizing workpiece point

Process number: 1-99

Interpolation method: joint interpolation, linear interpolation, circular interpolation

Joint interpolation: The robot will move to the point by joint interpolation

Linear interpolation: The robot will move to the point by linear interpolation

Circular interpolation: The robot will form a circular path with two other points (previous point: MOVJ/MOVL; next point: MOVC)

VJ: speed

PL: position level 0-5

ACC: acceleration range 1-100

DEC: deceleration range 1-100

TIME: Time, the range is a non-negative integer, and the unit is ms. Early execution time of the next instruction.

Example: PALREAL ID=2 MOVJ VJ=30% PL=2 ACC=20 DEC=20

PALCLEAR (palletizing reset)

Function: palletizing reset, palletizing status clear

Process number: 1-99

Example: PALCLEAR ID=1

PALOFF (palletizing end judgment)

Function: palletizing end judgment

Process number: 1-99

End judgment variable: Conditions for judging the end of palletizing

Example: PALOFF ID=1[variable name]

Note: If the total number of workpieces of a process number is n, the variable value will be set to 0 for the first n-1 times of executing PALREAL instruction, and will be set to 1 for the nth time of executing PALREAL instruction. If PALCLEAR is performed in the middle, the variable will be reset to 0.

PALLET_POS (get the workpiece point)

Function: get the workpiece point

Process number: the process number where the palletizing parameters are stored

Layer number: the layer where the workpiece is on

Number: number of the workpiece

Get point type: the point type corresponding to the workpiece

Example: PALLET_POS ID=1 1 1   P0001

Read the workpiece point of xth workpiece on xth layer by variables

PAL_SET_EXAMPLE (simple palletizing instruction)

Function: simple palletizing instruction

Process number: the process number where the palletizing parameters are stored

Starting workpiece point: the position point of the first workpiece when palletizing

End of column: the position point of the last workpiece in the column direction when palletizing

End of row: the position point of the last workpiece in the row direction when palletizing

Height end: the position point of the first workpiece on the last layer when palletizing.

Auxiliary point: the auxiliary point of palletizing, it is recommended to set it above the starting workpiece point.

Entry point: the entry point of palletizing, it is recommended to set it to a safety point outside the pallet.

Number of rows: total number of rows of palletizing.

Number of columns: total number of columns of palletizing.

Number of layers: total number of layers of palletizing.

Example: PAL_SET_EXAMPLE ID=1(P0001 P0002 P0003 P0004 P0005 P0006)

The PAL_SET_EXAMPLE instruction will run after filling all parameters, which will be filled into the palletizing process/simple palletizing accordingly, and the local points need to be set in the variables (this instruction is the same as the simple palletizing position setting in the palletizing process)

Usage scenarios

Scenario 1 - pickup point fixed, palletizing layer-by-layer at discharge point

Parameter setting

1. Click [menu bar - Process - Palletizing process - Complete palletizing] on the right side
2. Select the process number according to the actual situation, here we select process number 1
3. Click "Gripper setting"
4. Select the gripper according to the actual situation, here we select "1" for number of grippers, and "1" for gripper tool number (gripper tool number is the tool hand number, we need to go to the [Settings - Tool hand calibration] interface to set the gripper first), here we can only choose, click "Save"
5. Click "PgDn" to enter the pallet setting (you can also click "Back" and then enter the pallet setting)
6. Calibrate the pallet coordinate system (user coordinate system) according to the actual pallet and click "Save"

Note: When calibrating the pallet, you need to calibrate it with tool hand, and the Z-axis of the calibrated coordinate system cannot face downward

7. Click "PgDn" to enter the position setting (you can also click "Back" and then enter the position setting)
8. Calibrate the workpiece point, auxiliary point and entry point according to the actual situation and click "Save"

Note: Calibration needs to be done with tool hand

9. Click "PgDn" to enter the workpiece parameter setting (you can also click "Back" and then enter the workpiece parameter setting)
10. Fill in the workpiece size parameters according to the actual situation, here we set length "50", width "30", height "15", clearance "0", then click "Save"

11. Click "PgDn" to enter the proximity parameter setting (you can also click "Back" and then enter the proximity parameter setting)
12. Set according to the actual situation, if not needed, you can skip it directly
13. Click "PgDn" to enter the overlap mode setting (you can also click "Back" and then enter the overlap mode)
14. Fill in the number of layers according to the actual situation, here we set the number of layers to "10", the duplicate relationship to "Same", and select "1" for the graphic number of the first layer, and leave other parameters blank, click "Save"
15. Click "PgDn" to enter the plane mode setting (you can also click "Back" and then enter the plane mode)
16. Select "1" for graphic number, select "Criss-cross" for the template selection, fill in "1" for the number in X direction, "3" for the number in Y direction, and leave other parameters blank by default, click "Save". Click "Preview" to view the set graphic template

Note: The overall rotation here refers to rotating 180 degrees as a whole with the center of the first workpiece as the center of rotation

17. Click "Finish" to complete the parameter settings

Programming

NOP
Start

BOOLEAN A001 = 0
Insert variable

PALCLEAR ID = 1
Clear the previous palletizing data

WHILE (A001 == 0)
Loop statement

MOVJ P001 VJ = 30 %   PL = 0 ACC = 20 DEC = 20
Pickup entry point

WAIT (DIN4 == 1)   T = 10
Pickup judgment

MOVJ P003 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Pickup auxiliary point

MOVJ P002 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Pickup workpiece point

DOUT OT#(5) 1
Pickup signal

TIMER T = 1
Delay

MOVJ P003 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Pickup auxiliary point

PALON ID = 1 TYPE = 0 [-] [-] [-] MULTI = 0
Palletizing start

PALGRIPPER ID = 1 GRIPPERS = 1
Gripper selection

PALENTER ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20 OFF OFF
Discharge entry point

PALSHIFT ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge auxiliary point

PALREAL ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge workpiece point

DOUT OT#(5) 0
Discharge signal

TIMER T = 1
Delay

PALSHIFT ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge auxiliary point

PALOFF ID = 1 A001
Palletizing end

ENDWHILE
Loop end

END
End

Scenario 2 - pickup point fixed, discharge point height compensation

Parameter setting

1. Open [Process - Palletizing process - Complete palletizing - Overlap mode], fill in "100" for the discharge point height compensation, and click "Save"
2. For other parameter setting steps, please refer to scenario 1

Programming

Note: Please fill in the relevant parameters according to the actual situation

NOP
Start

BOOLEAN A001 = 0
Insert variable

PALCLEAR ID = 1
Clear the previous palletizing data

WHILE (A001 == 0)
Loop statement

MOVJ P001 VJ = 30 %   PL = 0 ACC = 20 DEC = 20
Pickup entry point

WAIT (DIN4 == 1)   T = 10
Pickup judgment

MOVJ P003 VJ = 30 %   PL = 0 ACC = 20 DEC = 20
Pickup auxiliary point

MOVJ P002 VJ = 30 %   PL = 0 ACC = 20 DEC = 20
Pickup workpiece point

DOUT OT#(5) 1
Pickup signal

TIMER T = 1
Delay

MOVJ P003 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Pickup auxiliary point

PALON ID = 1 TYPE = 0 [-] [-] [-] MULTI = 0
Palletizing start

PALGRIPPER ID = 1 GRIPPERS = 1
Gripper selection

PALENTER ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20 OFF OFF
Discharge entry point

PALSHIFT ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge auxiliary point

PALREAL ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge workpiece point

DOUT OT#(5) 0
Discharge signal

TIMER T = 1
Delay

PALSHIFT ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge auxiliary point

PALOFF ID = 1 A001
Palletizing end

ENDWHILE
Loop end

END
End

Scenario 3 - pickup point fixed, layer height correction

Parameter setting

1. Open [Process - Palletizing process - Complete palletizing - Overlap mode], fill in "50" for height correction of each layer, and click "Save"

Programming

NOP
Start

BOOLEAN A001 = 0
Insert variable

PALCLEAR ID = 1
Clear the previous palletizing data

WHILE (A001 == 0)
Loop statement

MOVJ P001 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Pickup entry point

WAIT (DIN4 == 1) T = 10
Pickup judgment

MOVJ P003 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Pickup auxiliary point

MOVJ P002 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Pickup workpiece point

DOUT OT#(5) 1
Pickup signal

TIMER T = 1
Delay

MOVJ P003 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Pickup auxiliary point

PALON ID = 1 TYPE = 0 [-] [-] [-] MULTI = 0
Palletizing start

PALGRIPPER ID = 1 GRIPPERS = 1
Gripper selection

PALENTER ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20 OFF OFF
Discharge entry point

PALSHIFT ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge auxiliary point

PALREAL ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge workpiece point

DOUT OT#(5) 0
Discharge signal

TIMER T = 1
Delay

PALSHIFT ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge auxiliary point

PALOFF ID = 1 A001
Palletizing end

ENDWHILE
Loop end

END
End

Scenario 4 - pickup point fixed, discharge point height fixed, vertical alignment

Parameter setting

1. Open [Process - Palletizing process - Complete palletizing - Overlap mode], check the "Vertical alignment" and click "Save"
2. Note: When using vertical alignment, the duplicate relationship needs to be changed to "Same", click the button after "Vertical alignment", the duplicate relationship will be automatically changed to "Same"

Programming

NOP
Start

BOOLEAN A001 = 0
Insert variable

PALCLEAR ID = 1
Clear the previous palletizing data

WHILE (A001 == 0)
Loop statement

MOVJ P001 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Pickup entry point

WAIT (DIN4 == 1) T = 10
Pickup judgment

MOVJ P003 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Pickup auxiliary point

MOVJ P002 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Pickup workpiece point

DOUT OT#(5) 1
Pickup signal

TIMER T = 1
Delay

MOVJ P003 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Pickup auxiliary point

PALON ID = 1 TYPE = 0 [-] [-] [-] MULTI = 0
Palletizing start

PALGRIPPER ID = 1 GRIPPERS = 1
Gripper selection

PALENTER ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20 OFF OFF
Discharge entry point

PALSHIFT ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge auxiliary point

PALREAL ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge workpiece point

DOUT OT#(5) 0
Discharge signal

TIMER T = 1
Delay

PALSHIFT ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge auxiliary point

PALOFF ID = 1 A001
Palletizing end

ENDWHILE
Loop end

END
End

Scenario 5 - pickup point fixed, rotating 180 degrees as a whole at discharge point, XY translation compensation

Parameter setting

1. Open [Process - Palletizing process - Complete palletizing - Overlap mode].
2. Fill in the number of layers according to the actual situation, here we set the number of layers to "10" and the duplicate relationship to "Alternate", and select "1" for the graphic number of the first layer and "2" for the graphic number of the second layer, leave other parameters blank, click "Save"
3. Open [Process - Palletizing process - Complete palletizing - Plane mode]
4. Select "2" for graphic number, select "Criss-cross" for the template selection, fill in "1" for the number in X direction, "3" for the number in Y direction, "180" for overall rotation angle,"50" for X translation compensation and "100" for Y translation compensation, leave other parameters blank by default, click "Save". Click "Preview" to view the set graphic template

Programming

NOP
Start

BOOLEAN A001 = 0
Insert variable

PALCLEAR ID = 1
Clear the previous palletizing data

WHILE (A001 == 0)
Loop statement

MOVJ P001 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Pickup entry point

WAIT (DIN4 == 1) T = 10
Pickup judgment

MOVJ P003 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Pickup auxiliary point

MOVJ P002 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Pickup workpiece point

DOUT OT#(5) 1
Pickup signal

TIMER T = 1
Delay

MOVJ P003 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Pickup auxiliary point

PALON ID = 1 TYPE = 0 [-] [-] [-] MULTI = 0
Palletizing start

PALGRIPPER ID = 1 GRIPPERS = 1
Gripper selection

PALENTER ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20 OFF OFF
Discharge entry point

PALSHIFT ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge auxiliary point

PALREAL ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge workpiece point

DOUT OT#(5) 0
Discharge signal

TIMER T = 1
Delay

PALSHIFT ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge auxiliary point

PALOFF ID = 1 A001
Palletizing end

ENDWHILE
Loop end

END
End

Scenario 6 - pickup point fixed, workpieces rotate 90 degrees at discharge point

Parameter setting

1. Open [Process - Palletizing process - Complete palletizing - Overlap mode]
2. Fill in the number of layers according to the actual situation, here we set the number of layers to "10" and the duplicate relationship to "Same", select "3" for the graphic number of the first layer,leave other parameters blank, click "Save"
3. Open [Process - Palletizing Process - Complete Palletizing - Plane Mode]
4. Select "3" for graphic number, select "Row-column" for the template selection, fill in "2" for the number in X direction, fill in "3" for the number in Y direction, select "90" for the workpiece rotation angle, and leave other parameters blank by default, click "Save". Click "Preview" to view the set graphic template

Programming

NOP
Start

BOOLEAN A001 = 0
Insert variable

PALCLEAR ID = 1
Clear the previous palletizing data

WHILE (A001 == 0)
Loop statement

MOVJ P001 VJ = 30 %　PL = 0 ACC = 20 DEC = 20
Pickup entry point

WAIT (DIN4 == 1)　T = 10
Pickup judgment

MOVJ P003 VJ = 30 %　PL = 0 ACC = 20 DEC = 20
Pickup auxiliary point

MOVJ P002 VJ = 30 %　PL = 0 ACC = 20 DEC = 20
Pickup workpiece point

DOUT OT#(5) 1
Pickup signal

TIMER T = 1
Delay

MOVJ P003 VJ = 30 %　PL = 0 ACC = 20 DEC = 20
Pickup auxiliary point

PALON ID = 1 TYPE = 0 [-] [-] [-] MULTI = 0
Palletizing start

PALGRIPPER ID = 1 GRIPPERS = 1
Gripper selection

PALENTER ID = 1 MovJ VJ = 30 %　PL = 0 ACC = 20 DEC = 20 OFF OFF
Discharge entry point

PALSHIFT ID = 1 MovJ VJ = 30 %　PL = 0 ACC = 20 DEC = 20
Discharge auxiliary point

PALREAL ID = 1 MovJ VJ = 30 %　PL = 0 ACC = 20 DEC = 20
Discharge workpiece point

DOUT OT#(5) 0
Discharge signal

TIMER T = 1
Delay

PALSHIFT ID = 1 MovJ VJ = 30 %　PL = 0 ACC = 20 DEC = 20
Discharge auxiliary point

PALOFF ID = 1 A001
Palletizing end

ENDWHILE
Loop end

END
End

Scenario 7 - pickup point fixed, auxiliary point height fixed

Parameter setting

1. Open [Process - Palletizing process - Complete palletizing - Overlap mode]
2. Fill in the number of layers according to the actual situation, here we set the number of layers to "3" and the duplicate relationship to "Same", select "1" for the graphic number of the first layer, check "Fixed auxiliary point height", "Layer auto alignment" and "Auto-rotate attitude", click "Save"
3. Open [Process - Palletizing process - Complete palletizing - Plane mode]
4. Select "1" for graphic number, select "Row-column" for the template selection, fill in "3" for the number in X direction, fill in "4" for the number in Y direction, and leave other parameters blank by default, click "Save". Click "Preview" to view the set graphic template

Programming

NOP
Start

BOOLEAN A001 = 0
Insert variable

PALCLEAR ID = 1
Clear the previous palletizing data

WHILE (A001 == 0)
Loop statement

MOVJ P001 VJ = 30 %   PL = 0 ACC = 20 DEC = 20
Pickup entry point

WAIT (DIN4 == 1)   T = 10
Pickup judgment

MOVJ P003 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Pickup auxiliary point

MOVJ P002 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Pickup workpiece point

DOUT OT#(5) 1
Pickup signal

TIMER T = 1
Delay

MOVJ P003 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Pickup auxiliary point

PALON ID = 1 TYPE = 0 [-] [-] [-] MULTI = 0
Palletizing start

PALGRIPPER ID = 1 GRIPPERS = 1
Gripper selection

PALENTER ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20 OFF OFF
Discharge entry point

PALSHIFT ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge auxiliary point

PALREAL ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge workpiece point

DOUT OT#(5) 0
Discharge signal

TIMER T = 1
Delay

PALSHIFT ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge auxiliary point

PALOFF ID = 1 A001
Palletizing end

ENDWHILE
Loop end

END
End

Scenario 8 - depalletizing

Parameter setting

1. Click [menu bar - Process - Palletizing process - Complete palletizing] on the right side
2. Select the process number according to the actual situation, here we select process number 1
3. Click "Gripper setting"
4. Select the gripper according to the actual situation, here we select "1" for number of grippers, and "1" for gripper tool number (gripper tool number is the tool hand number, we need to go to the [Settings - Tool hand calibration] interface to set the gripper first), here we can only choose but not calibrate, click "Save"
5. Click "PgDn" to enter the pallet setting (you can also click "Back" and then enter the pallet setting)
6. Calibrate the pallet coordinate system (user coordinate system) according to the actual pallet and click "Save"

Note: When calibrating the pallet, you need to calibrate it with tool hand, and the Z-axis of the calibrated coordinate system cannot face downward

7. Click "PgDn" to enter the position setting (you can also click "Back" and then enter the position setting)
8. Calibrate the workpiece point, auxiliary point and entry point according to the actual situation and click "Save"

Note: Calibration needs to be done with tool hand. The depalletizing workpiece point is still set according to the palletizing process, and the depalletizing starts with the last workpiece on the highest layer

9. Click "PgDn" to enter the workpiece parameter setting (you can also click "Back" and then enter the workpiece parameter setting)
10. Fill in the workpiece size parameters according to the actual situation, here we set length "50", width "30", height "15", clearance "0", then click "Save"
11. Click "PgDn" to enter the proximity parameter setting (you can also click "Back" and then enter the proximity parameter setting)

12. Set according to the actual situation, if not needed, you can skip it directly
13. Click "PgDn" to enter the overlap mode setting (you can also click "Back" and then enter the overlap mode)
14. Fill in the number of layers according to the actual situation, here we set the number of layers to "10", the duplicate relationship to "Same", and select "1" for the graphic number of the first layer, and leave other parameters blank, click "Save"
15. Click "PgDn" to enter the plane mode setting (you can also click "Back" and then enter the plane mode)
16. Select "1" for graphic number, select "Criss-cross" for the template selection, fill in "1" for the number in X direction, "3" for the number in Y direction, and leave other parameters blank by default, click "Save". Click "Preview" to view the set graphic template

Note: The overall rotation here refers to rotating 180 degrees as a whole with the center of the first workpiece as the center of rotation

17. Click "Finish" to complete the parameter settings of process number 1

Programming

NOP
Start

BOOLEAN A001 = 0
Insert variable

PALCLEAR ID = 1
Clear the previous palletizing data

WHILE (A001 == 0)
Loop instruction

PALON ID = 1 TYPE = 1 [-] [-] [-] MULTI = 0
Depalletizing start

PALGRIPPER ID = 1 GRIPPERS = 1
Gripper selection

PALENTER ID = 1 MovJ VJ = 20 % PL = 0 ACC = 20 DEC = 20 OFF OFF
Pickup entry point

WAIT (DIN4 == 1) T = 10
Pickup judgment

PALSHIFT ID = 1 MovJ VJ = 20 % PL = 0 ACC = 20 DEC = 20
Pickup auxiliary point

PALREAL ID = 1 MovJ VJ = 20 % PL = 0 ACC = 20 DEC = 20
Pickup workpiece point

DOUT OT#(5) 1
Pickup signal

TIMER T = 1
Delay

PALSHIFT ID = 1 MovJ VJ = 20 % PL = 0 ACC = 20 DEC = 20
Pickup auxiliary point

PALOFF ID = 1 A001
Depalletizing end

MOVJ P001 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge entry point

MOVJ P003 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge auxiliary point

MOVJ P002 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge workpiece point

DOUT OT#(5) 0
Discharge signal

TIMER T = 1
Delay

MOVJ P003 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge auxiliary point

ENDWHILE
Loop end

END
End

Scenario 9 - palletizing after depalletizing

Parameter setting

Depalletizing parameters

1. Click [menu bar - Process - Palletizing process - Complete palletizing] on the right side
2. Select the process number according to the actual situation, here we select process number 1
3. Click "Gripper setting"
4. Select the gripper according to the actual situation, here we select "1" for number of grippers, and "1" for gripper tool number (gripper tool number is the tool hand number, we need to go to the [Settings - Tool hand calibration] interface to set the gripper first), here we can only choose, click "Save"
5. Click "PgDn" to enter the pallet setting (you can also click "Back" and then enter the pallet setting)
6. Calibrate the pallet coordinate system (user coordinate system) according to the actual pallet and click "Save"

Note: When calibrating the pallet, you need to calibrate it with tool hand, and the Z-axis of the calibrated coordinate system cannot face downward

7. Click "PgDn" to enter the position setting (you can also click "Back" and then enter the position setting)
8. Calibrate the workpiece point, auxiliary point and entry point according to the actual situation and click "Save"

Note: Calibration needs to be done with tool hand. The depalletizing workpiece point is still set according to the palletizing process, and the depalletizing starts with the last workpiece on the highest layer

9. Click "PgDn" to enter the workpiece parameter setting (you can also click "Back" and then enter the workpiece parameter setting)
10. Fill in the workpiece size parameters according to the actual situation, here we set length "50", width "30", height "15", clearance "0", then click "Save"

11. Click "PgDn" to enter the proximity parameter setting (you can also click "Back" and then enter the proximity parameter setting)
12. Set according to the actual situation, if not needed, you can skip it directly
13. Click "PgDn" to enter the overlap mode setting (you can also click "Back" and then enter the overlap mode)
14. Fill in the number of layers according to the actual situation, here we set the number of layers to "10", the duplicate relationship to "Same", and select "1" for the graphic number of the first layer, and leave other parameters blank, click "Save"
15. Click "PgDn" to enter the plane mode setting (you can also click "Back" and then enter the plane mode)
16. Select "1" for graphic number, select "Criss-cross" for the template selection, fill in "1" for the number in X direction, "3" for the number in Y direction, and leave other parameters blank by default, click "Save". Click "Preview" to view the set graphic template

Note: The overall rotation here refers to rotating 180 degrees as a whole with the center of the first workpiece as the center of rotation

17. Click "Finish" to complete the parameter settings of process number 1

Palletizing parameters

1. Click "Complete palletizing"
2. Select process number 2 and fill in the parameters of process number 2 according to the steps of process number 1

Note: Depalletizing parameters are consistent with palletizing parameters

Programming

NOP
Start

BOOLEAN A001 = 0
Insert variable

BOOLEAN A002 = 0
Insert variable

PALCLEAR ID = 1
Clear the previous depalletizing data

PALCLEAR ID = 2
Clear the previous palletizing data

WHILE (A001 == 0)
Loop instruction

PALON ID = 1 TYPE = 1 [-] [-] [-] MULTI = 0
Depalletizing start

PALGRIPPER ID = 1 GRIPPERS = 1
Gripper selection

PALENTER ID = 1 MovJ VJ = 20 % PL = 0 ACC = 20 DEC = 20 OFF OFF
Pickup entry point

WAIT (DIN4 == 1) T = 10
Pickup judgement

PALSHIFT ID = 1 MovJ VJ = 20 % PL = 0 ACC = 20 DEC = 20
Pickup auxiliary point

PALREAL ID = 1 MovJ VJ = 20 % PL = 0 ACC = 20 DEC = 20
Pickup workpiece point

DOUT OT#(5) 1
Pickup signal

TIMER T = 1
Delay

PALSHIFT ID = 1 MovJ VJ = 20 % PL = 0 ACC = 20 DEC = 20
Pickup auxiliary point

PALOFF ID = 1
Depalletizing end

PALON ID = 2 TYPE = 0 [-] [-] [-] MULTI = 0
Palletizing start

PALGRIPPER ID = 2 GRIPPERS = 1
Gripper selection

PALENTER ID = 2 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20 OFF OFF
Discharge entry point

PALSHIFT ID = 2 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge auxiliary point

PALREAL ID = 2 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge workpiece point

DOUT OT#(5) 0
Discharge signal

TIMER T = 1
Delay

PALSHIFT ID = 2 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge auxiliary point

PALOFF ID = 2 A001
Palletizing end

ENDWHILE
End

> Scenario 10 - palletizing interrupted, continue palletizing

Parameter setting

1. Complete process parameter setting before palletizing starts
2. Complete process parameter setting after interruption
3. Open [Status - Palletizing status]
4. For the process number, choose the one selected when setting the process parameter, here we select the process number 1 set before
5. If the palletizing position has been set to the 5th of the first layer, fill in "1" for the current number of layer, "5" for the number of palletized workpieces on current layer, and click "Save"

Programming

NOP
Start

BOOLEAN A001 = 0
Insert variable

PALCLEAR ID = 1
Clear the previous palletizing data

WHILE (A001 == 0)
Loop statement

MOVJ P001 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Pickup entry point

WAIT (DIN4 == 1) T = 10
Pickup judgment

MOVJ P003 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Pickup auxiliary point

MOVJ P002 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Pickup workpiece point

DOUT OT#(5) 1
Pickup signal

TIMER T = 1
Delay

MOVJ P003 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Pickup auxiliary point

PALON ID = 1 TYPE = 0 [-] [-] [-] MULTI = 0
Palletizing start

PALGRIPPER ID = 1 GRIPPERS = 1
Gripper selection

PALENTER ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20 OFF OFF
Discharge entry point

PALSHIFT ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge auxiliary point

PALREAL ID = 1 MovJ VJ = 30 %　PL = 0 ACC = 20 DEC = 20
Discharge workpiece point

DOUT OT#(5) 0
Discharge signal

TIMER T = 1
Delay

PALSHIFT ID = 1 MovJ VJ = 30 %　PL = 0 ACC = 20 DEC = 20
Discharge auxiliary point

PALOFF ID = 1 A001
Palletizing end

ENDWHILE
Loop end

END
End

Scenario 11 - palletizing with multi-gripper

Parameter setting

1. Click [menu bar - Process - Palletizing process - Complete palletizing] on the right side
2. Select the process number according to the actual situation, here we select process number 1
3. Click "Gripper setting"
4. Select the gripper according to the actual situation, here we select "4" for number of grippers, "2" for gripper 1 tool number, "4" for gripper 2 tool number, "5" for gripper 3 tool number and "1" for gripper 4 tool number (gripper tool number is the tool hand number, we need to go to the [Settings - Tool hand calibration] interface to set the gripper first), here we can only choose, click "Save"
5. Click "PgDn" to enter the pallet setting (you can also click "Back" and then enter the pallet setting)
6. Calibrate the pallet coordinate system (user coordinate system) according to the actual pallet and click "Save"

Note: When calibrating the pallet, you need to calibrate it with tool hand (can be calibrated with any one of the grippers), and the Z-axis of the calibrated coordinate system cannot face downward

7. Click "PgDn" to enter the position setting (you can also click "Back" and then enter the position setting)
8. Calibrate the workpiece point, auxiliary point and entry point according to the actual situation and click "Save"

Note: Calibration needs to be done with tool hand

9. Click "PgDn" to enter the workpiece parameter setting (you can also click "Back" and then enter the workpiece parameter setting)
10. Fill in the workpiece size parameters according to the actual situation, here we set length "50", width "30", height "15", clearance "0", then click "Save"
11. Click "PgDn" to enter the proximity parameter setting (you can also click "Back" and then enter the proximity parameter setting)
12. Set according to the actual situation, if not needed, you can skip it directly
13. Click "PgDn" to enter the overlap mode setting (you can also click "Back" and then enter the overlap mode)
14. Fill in the number of layers according to the actual situation, here we set the number of layers to "10", the duplicate relationship to "Same", and select "1" for the graphic number of the first layer, and leave other parameters blank, click "Save"
15. Click "PgDn" to enter the plane mode setting (you can also click "Back" and then enter the plane mode)
16. Select "1" for graphic number, select "Criss-cross" for the template selection, fill in "1" for the number in X direction, "3" for the number in Y direction, and leave other parameters blank by default, click "Save". Click "Preview" to view the set graphic template

Note: The overall rotation here refers to rotating 180 degrees as a whole with the center of the first workpiece as the center of rotation

17. Click "Finish" to complete the parameter settings

Programming

NOP
Start

BOOLEAN A001 = 0
Insert variable

PALCLEAR ID = 1
Clear the previous palletizing data

WHILE (A001 == 0)
Loop statement

MOVJ P001 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Pickup entry point

WAIT (DIN4 == 1) T = 10
Pickup judgement

MOVJ P003 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Pickup auxiliary point

MOVJ P002 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Pickup workpiece point

DOUT OT#(5) 1
Pickup signal

TIMER T = 1
Delay

MOVJ P003 VJ = 30 % PL = 0 ACC = 20 DEC = 20
Pickup auxiliary point

PALON ID = 1 TYPE = 0 [-] [-] [-] MULTI = 0
Palletizing start

PALGRIPPER ID = 1 GRIPPERS = 1
Gripper selection

PALENTER ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20 OFF OFF
Discharge entry point

PALSHIFT ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge auxiliary point

PALREAL ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge workpiece point

DOUT OT#(5) 0
Discharge signal

TIMER T = 1
Delay

PALSHIFT ID = 1 MovJ VJ = 30 % PL = 0 ACC = 20 DEC = 20
Discharge auxiliary point

PALOFF ID = 1 A001
Palletizing end

ENDWHILE
Loop end

END
End

Scenario 12 - Two stacks on one line (two stacks with same number of workpieces)

Parameter setting

1. Click [menu bar - Process - Palletizing Process - Palletizing parameters - Complete Palletizing]
2. Select the process number according to the actual situation, here we select process number 1 for the first stack
3. Click "Gripper Setting", select the gripper according to the actual situation, here we select "1" for number of grippers, and "1" for gripper tool number (gripper tool number is the tool hand number, we need to go to the [Settings - Tool hand calibration] interface to set the gripper first), click "Save" after modification
4. Click "PgDn" to enter the pallet setting (you can also click "Back" and then enter the pallet setting), calibrate the pallet coordinate system (user coordinate system) according to the actual pallet. Select the user coordinate system, first calibrate the origin of the pallet, the positive direction of the x-axis of the pallet, and the positive direction of the y-axis of the pallet. After the calibration is completed, you must click "Calculate". The positive direction of the z-axis without calibration is automatically calculated by the

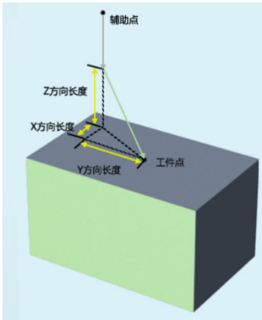
system according to the positive direction of the calibrated x-axis and y-axis. After all calibrations are complete, click "Save"

Note: When calibrating the pallet, you need to calibrate it with tool hand, and the Z-axis of the calibrated coordinate system cannot face downward

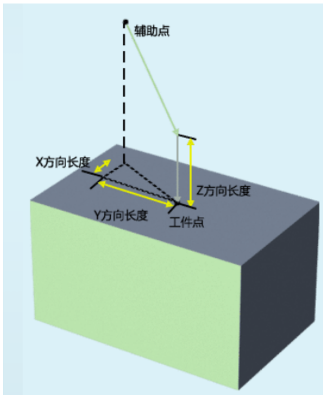
5. Click "PgDn" to enter the position setting (you can also click "Back" and then enter the position setting), calibrate the workpiece point, auxiliary point and entry point according to the actual situation and click "Save"
6. Click "PgDn" to enter the workpiece parameter setting, set the workpiece size according to the actual situation, here we set length "50", width "30", height "15", clearance "10", (The y-direction of the pallet is the workpiece length direction, the x-direction of the pallet is the workpiece width direction, the z-direction of the pallet is the workpiece height direction, and the clearance is the empty space between the workpiece and the workpiece), click "Save" after filling.
7. Click "PgDn" to enter the approach parameter setting (you can also click "Back" and then enter the approach parameter setting), and set the approach method and pallet detection. Choose whether to open or not according to the actual situation. Here, we turn on the approach enable switch and the pallet detection switch, set the length of pallet X direction "50", the length of pallet Y direction "50", the length of pallet Z direction "50", the thickness of pallet "10", and the total number of pallets "3".

Note: If the approach enable is turned off, then the program will not generate the palletizing approach point instruction (the opposite is true when the enable switch is turned on), and there are two approach methods: approach descent and descent approach.

Approach descent: For example, according to the approach parameters set above, the position of the approach point is 50 mm away from the position of the workpiece point in the X direction, 50 in the Y direction, and 50 in the Z direction. Use the set approach method to move from the approach point to the workpiece point



Descent approach: For example, according to the approach parameters set above, the position of the auxiliary point is 50 mm away from the position of the workpiece point in the X direction, 50 in the Y direction. Use the set approach method to move from the auxiliary point to the approach point (50 mm above the workpiece point), and then descent vertically from the approach point to the workpiece point



Pallet detection: The range of the total number of pallets is [1-5], and the fixed bound IO ports are 2-1~2-5 ports. The program will automatically detect whether the IO is open or closed to determine the number of pallets. For example, when the total number of pallets is "3", pallet thickness is "10" mm, if you start the program, it will automatically detect whether IO 2-1, 2-2, and 2-3 are open. When the number of pallets is reduced by one (2-1 or 2-2 or 2-3 is arbitrarily closed), the z-axis direction length of all workpiece points will be reduced by 10mm.

8. Click "PgDn" to enter the overlap mode setting (you can also click "Back" and then enter the overlap mode), fill in the number of layers according to the actual situation, here we set the number of layers to "2", the duplicate relationship to "Same", and select "1" for the graphic number of the first layer, and leave other parameters blank, (The number of layers is the total number of layers of the workpiece, and the duplicate relationship is the relationship between the placement of each layer), click "Save"

9. Click "PgDn" to enter the plane mode setting (you can also click "Back" and then enter the plane mode), select "1" for graphic number, select "Criss-cross" for the template selection, fill in "2" for the number in X direction, "1" for the number in Y direction, and leave other parameters blank by default, click "Save". Click "Preview" to view the set graphic template

Note: The overall rotation here refers to rotating 180 degrees as a whole with the center of the first workpiece as the center of rotation

10. Click "Finish" to complete the parameter settings

11. Use the above method to set the second stack according to the actual situation, select 2 for the process number. (Note: The user coordinate system in process number 2 needs to be recalibrated according to the actual situation or use the user coordinate system set in process number 1)

Programming

NOP
Start

PALCLEAR ID = 1
Palletizing reset, process number 1

PALCLEAR ID = 2
Palletizing reset, process number 2

WHILE (B001 == 0)
Loop statement

PALON ID = 1 TYPE = 0 [-] [-] [-] MULTI = 0
Stack 1 palletizing start

PALGRIPPER ID = 1 GRIPPERS = 1
Gripper selection

MOVJ P001 VJ = 50 % PL = 5 ACC = 10 DEC = 10 0
Pickup safety point

MOVJ P002 VJ = 50 % PL = 5 ACC = 10 DEC = 10 0
Point above the pickup point

MOVJ P003 VJ = 50 % PL = 5 ACC = 10 DEC = 10 0
Pickup point

DOUT OT#(1) 1 T = 0 0
Pickup signal

TIMER T = 1
Delay 1s

MOVJ P002 VJ = 50 % PL = 5 ACC = 10 DEC = 10 0
Point above the pickup point

PALENTER ID = 1 MovJ VJ = 50 % PL = 5 ACC = 10 DEC = 10 OFF OFF 0
Discharge entry point

PALSHIFT ID = 1 MovJ VJ = 50 % PL = 5 ACC = 10 DEC = 10 0
Discharge auxiliary point

PALAPPRO ID = 1 MovJ VJ = 50 % PL = 0 ACC = 20 DEC = 20 0
Discharge approach point

PALREAL ID = 1 MovJ VJ = 50 % PL = 5 ACC = 10 DEC = 10 0
Discharge workpiece point

DOUT OT#(1) 0 T = 0 0
Discharge signal

TIMER T = 1
Delay 1s

PALAPPRO ID = 1 MovJ VJ = 50 % PL = 0 ACC = 20 DEC = 20 0
Discharge approach point

PALSHIFT ID = 1 MovJ VJ = 50 % PL = 5 ACC = 10 DEC = 10 0
Discharge auxiliary point

PALOFF ID = 1
Palletizing end judgment

PALON ID = 2 TYPE = 0 [-] [-] [-] MULTI = 0
Stack 2 palletizing start

PALGRIPPER ID = 2 GRIPPERS = 1
Gripper selection

MOVJ P001 VJ = 50 % PL = 5 ACC = 10 DEC = 10 0
Pickup safety point

MOVJ P002 VJ = 50 % PL = 5 ACC = 10 DEC = 10 0
Point above the pickup point

MOVJ P003 VJ = 50 % PL = 5 ACC = 10 DEC = 10 0
Pickup point

DOUT OT#(1) 1 T = 0 0
Pickup signal

TIMER T = 1
Delay 1s

MOVJ P002 VJ = 50 % PL = 5 ACC = 10 DEC = 10 0
Point above the pickup point

PALENTER ID = 2 MovJ VJ = 50 % PL = 5 ACC = 10 DEC = 10 OFF OFF 0
Discharge entry point

PALSHIFT ID = 2 MovJ VJ = 50 % PL = 5 ACC = 10 DEC = 10 0
Discharge auxiliary point

PALAPPRO ID = 1 MovJ VJ = 50 % PL = 0 ACC = 20 DEC = 20 0
Discharge approach point

PALREAL ID = 2 MovJ VJ = 50 % PL = 5 ACC = 10 DEC = 10 0
Discharge workpiece point

DOUT OT#(1) 0 T = 0 0
Discharge signal

TIMER T = 1
Delay 1s

PALAPPRO ID = 1 MovJ VJ = 50 % PL = 0 ACC = 20 DEC = 20 0
Discharge approach point

PALSHIFT ID = 2 MovJ VJ = 50 % PL = 5 ACC = 10 DEC = 10 0
Discharge auxiliary point

PALOFF ID = 2 B001
Palletizing process number 2, cycle end judgment

ENDWHILE
Loop end

END
End

> Scenario 13 - Two stacks on one line (Two stacks with different number of workpieces)

Parameter setting

1. Click [menu bar - Process - Palletizing process - Palletizing parameters - Complete palletizing]
2. Select the process number according to the actual situation, here we select process number 1 for the first stack
3. Click "Gripper setting", select the gripper according to the actual situation, here we select "1" for number of grippers, and "1" for gripper tool number (gripper tool number is the tool hand number, we need to go to the [Settings - Tool hand calibration] interface to set the gripper first), click "Save" after modification
4. Click "PgDn" to enter the pallet setting (you can also click "Back" and then enter the pallet setting), calibrate the pallet coordinate system (user coordinate system) according to the actual pallet. Select the user coordinate system, first calibrate the origin of the pallet, the positive direction of the x-axis of the pallet, and the positive direction of the y-axis of the pallet. After the calibration is completed, you must click "Calculate". The positive direction of the z-axis without calibration is automatically calculated by the system according to the positive direction of the calibrated x-axis and y-axis. After all calibrations are complete, click "Save"

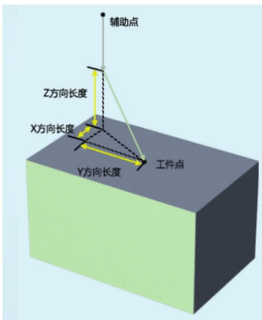
Note: When calibrating the pallet, you need to calibrate it with tool hand, and the Z-axis of the calibrated coordinate system cannot face downward

5. Click "PgDn" to enter the position setting (you can also click "Back" and then enter the position setting), calibrate the workpiece point, auxiliary point and entry point according to the actual situation and click "Save"
6. Click "PgDn" to enter the workpiece parameter setting, set the workpiece size according to the actual situation, here we set length "50", width "30", height "15", clearance "10", (The y-direction of the pallet is the workpiece length direction, the x-direction of the pallet is the workpiece width direction, the z-direction of the pallet is the workpiece height direction, and the clearance is the empty space between the workpiece and the workpiece), click "Save" after filling.

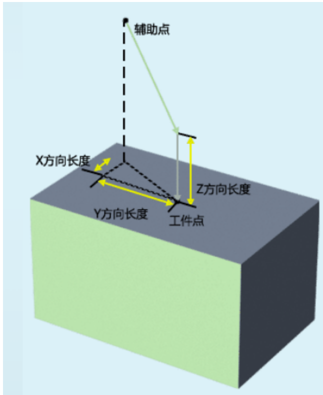
7. Click "PgDn" to enter the approach parameter setting (you can also click "Back" and then enter the approach parameter setting), and set the approach method and pallet detection. Choose whether to open or not according to the actual situation. Here, we turn on the approach enable switch and the pallet detection switch, set the length of pallet X direction "50", the length of pallet Y direction "50", the length of pallet Z direction "50", the thickness of pallet "10", and the total number of pallets "3".

Note: If the approach enable is turned off, then the program will not generate the palletizing approach point instruction (the opposite is true when the enable is turned on), and there are two approach methods: approach descent and descent approach.

Approach descent: For example, according to the approach parameters set above, the position of the approach point is 50 mm away from the position of the workpiece point in the X direction, 50 in the Y direction, and 50 in the Z direction. Use the set approach method to move from the approach point to the workpiece point



Descent approach: For example, according to the approach parameters set above, the position of the auxiliary point is 50 mm away from the position of the workpiece point in the X direction, 50 in the Y direction. Use the set approach method to move from the auxiliary point to the approach point (50 mm above the workpiece point), and then descent vertically from the approach point to the workpiece point



Pallet detection: The range of the total number of pallets is [1-5], and the fixed bound IO ports are 2-1~2-5 ports. The program will automatically detect whether the IO is open or closed to determine the number of pallets. For example, when the total number of pallets is "3", pallet thickness is "10" mm, if you start the program, it will automatically detect whether IO 2-1, 2-2, and 2-3 are open. When the number of pallets is reduced by one (2-1 or 2-2 or 2-3 is arbitrarily closed), the z-axis direction length of all workpiece points will be reduced by 10mm.

8. Click "PgDn" to enter the overlap mode setting (you can also click "Back" and then enter the overlap mode), fill in the number of layers according to the actual situation, here we set the number of layers to "2", the duplicate relationship to "Same", and select "1" for the graphic number of the first layer, and leave other parameters blank, (The number of layers is the total number of layers of the workpiece, and the duplicate relationship is the relationship between the placement of each layer), click "Save"

9. Click "PgDn" to enter the plane mode setting (you can also click "Back" and then enter the plane mode), select "1" for graphic number, select "Criss-cross" for the template selection, fill in "2" for the number in X direction, "1" for the number in Y direction, and leave other parameters blank by default, click "Save". Click "Preview" to view the set graphic template

Note: The overall rotation here refers to rotating 180 degrees as a whole with the center of the first workpiece as the center of rotation

10. Click "Finish" to complete the parameter settings

11. Use the above method to set the second stack according to the actual situation, select 2 for the process number. (Note: The user coordinate system in process number 2 needs to be recalibrated according to the actual situation or use the user coordinate system set in process number 1)

Programming

NOP
Start

PALCLEAR ID = 1
Palletizing reset, process number 1

PALCLEAR ID = 2
Palletizing reset, process number 2

WHILE {(B003 == 0)}
Loop statement

IF {(B001 == 0)}
Execute the if judgment statement of stack 1

PALON ID = 1 TYPE = 0 [-] [-] [-] MULTI = 0
Stack 1 palletizing start

PALGRIPPER ID = 1 GRIPPERS = 1
Gripper selection

MOVJ P001 VJ = 50 % PL = 0 ACC = 20 DEC = 20 0
Pickup safety point

MOVJ P002 VJ = 50 % PL = 0 ACC = 20 DEC = 20 0
Point above pickup point

MOVJ P003 VJ = 50 % PL = 0 ACC = 20 DEC = 20 0
Pickup point

DOUT OT#(1) 1 T = 0 0
Pickup signal

TIMER T = 1
Delay 1s

MOVJ P002 VJ = 50 % PL = 0 ACC = 20 DEC = 20 0
Point above pickup point

PALENTER ID = 1 MovJ VJ = 50 % PL = 0 ACC = 20 DEC = 20 OFF OFF 0
Discharge entry point

PALSHIFT ID = 1 MovJ VJ = 50 % PL = 0 ACC = 20 DEC = 20 0
Discharge auxiliary point

PALAPPRO ID = 1 MovJ VJ = 50 % PL = 0 ACC = 20 DEC = 20 0
Discharge approach point

PALREAL ID = 1 MovJ VJ = 50 % PL = 0 ACC = 20 DEC = 20 0
Discharge point

DOUT OT#(1) 0 T = 0 0
Discharge signal

TIMER T = 1
Delay 1s

PALAPPRO ID = 1 MovJ VJ = 50 % PL = 0 ACC = 20 DEC = 20 0
Discharge approach point

PALSHIFT ID = 1 MovJ VJ = 50 % PL = 0 ACC = 20 DEC = 20 0
Discharge auxiliary point

PALOFF ID = 1 B001
Stack 1 palletizing end judgment

ENDIF
End if

IF {(B002 == 0)}
Execute the if judgment statement of stack 2

PALON ID = 2 TYPE = 0 [-] [-] [-] MULTI = 0
Stack 2 palletizing start

PALGRIPPER ID = 2 GRIPPERS = 1
Gripper selection

MOVJ P001 VJ = 50 % PL = 0 ACC = 20 DEC = 20 0
Pickup safety point

MOVJ P002 VJ = 50 % PL = 0 ACC = 20 DEC = 20 0
Point above pickup point

MOVJ P003 VJ = 50 % PL = 0 ACC = 20 DEC = 20 0
Pickup point

DOUT OT#(1) 1 T = 0 0
Pickup signal

TIMER T = 1
Delay 1s

MOVJ P002 VJ = 50 % PL = 0 ACC = 20 DEC = 20 0
Point above pickup point

PALENTER ID = 2 MovJ VJ = 50 % PL = 0 ACC = 20 DEC = 20 OFF OFF 0
Discharge entry point

PALSHIFT ID = 2 MovJ VJ = 50 % PL = 0 ACC = 20 DEC = 20 0
Discharge auxiliary point

PALAPPRO ID = 2 MovJ VJ = 50 % PL = 0 ACC = 20 DEC = 20 0
Discharge approach point

PALREAL ID = 2 MovJ VJ = 50 % PL = 0 ACC = 20 DEC = 20 0
Discharge point

DOUT OT#(1) 0 T = 0 0
Discharge signal

TIMER T = 1
Delay 1s

PALAPPRO ID = 2 MovJ VJ = 50 % PL = 0 ACC = 20 DEC = 20 0
Discharge approach point

PALSHIFT ID = 2 MovJ VJ = 50 % PL = 0 ACC = 20 DEC = 20 0
Discharge auxiliary point

PALOFF ID = 2 B002
Stack 2 palletizing end judgment

ENDIF
End if

IF {(B001 == 1)} AND {(B002 == 1)}
Determine whether both stack 1 and stack 2 have finished palletizing

SETBOOL B003 = 1
Variable to jump out of loop

ENDIF
End if

ENDWHILE
Loop end

END
End

Polishing Process

Catalogue

Polishing Process

This chapter mainly describes the relevant information about the polishing process of this control system. iNexBot pioneered special instructions for edge solder joints polishing without complex programming.

It can realize automatic replacement of grinding wheels for various polishing, and the robot automatically polishes multiple times in different directions.

• Polishing of welding spatter

• Polishing of surface bumps and scratches

• Smoothing of weld reinforcement

• Smoothing of machining allowance

• Polishing of long and large welds

• Removal of edges and burrs

Combined with external axis equipment such as positioner, it can polish large sheet metal parts and ensure smooth and flat polishing effect

Combined with offline programming, it can achieve compliant polishing of complex curved workpieces

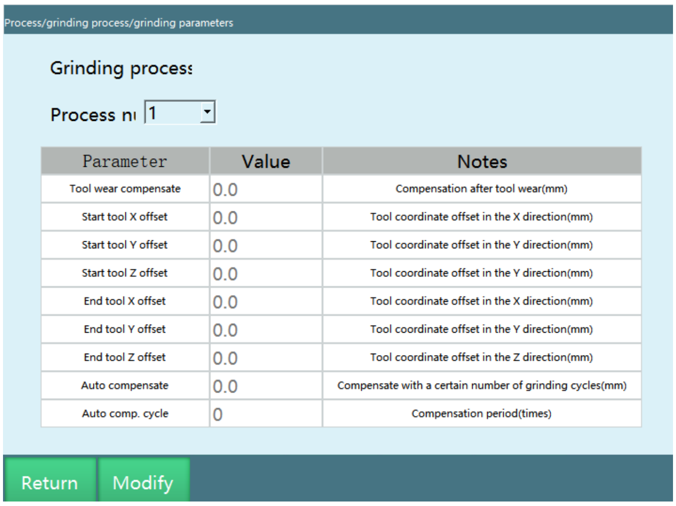
Combined with line scan laser tracking technology, it can achieve automatic programming of polishing

- 2-point positioning for straight line

- 3-point/4-point positioning for user coordinate system

Polishing parameters

Turn on the teach pendant, enter the "Process" interface, select "Polishing process", and enter the "Polishing parameters" interface. At this time, do not click "Modify", only the process number can be modified. After selecting one of the process numbers, click the "Modify" button to perform modifications.



Process number: 1-9 process numbers are provided, each of which stores all the parameters below that process number.

Tool wear compensation: The value of polishing tool wear, which will be automatically compensated after filling in

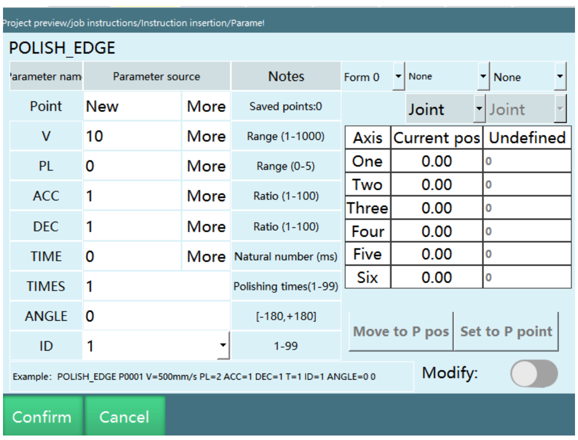
Tool offset at start point in X/Y/Z direction: Before polishing starts, the offset will be automatically performed at the start point

Tool offset at end point in X/Y/Z direction: After the polishing is completed, the offset will be automatically performed at the end point

Auto compensation period/auto compensation value: After every set number of polishing, all parameters will be automatically shifted by a certain distance

Polishing instructions

POLISH_EDGE (edge polishing) instruction



At present, the polishing process only supports polishing in the straight line direction. Compared with the MOVL instruction, the POLISH_EDGE in the polishing process adds the angle parameter (ANGLE) , the polishing times parameter (TIMES), and the process number parameter (ID).

V: linear motion speed, 2-1000 (mm/s)

PL: position level, 0-5

ACC: acceleration adjustment ratio, 1-100

DEC: deceleration adjustment ratio, 1-100

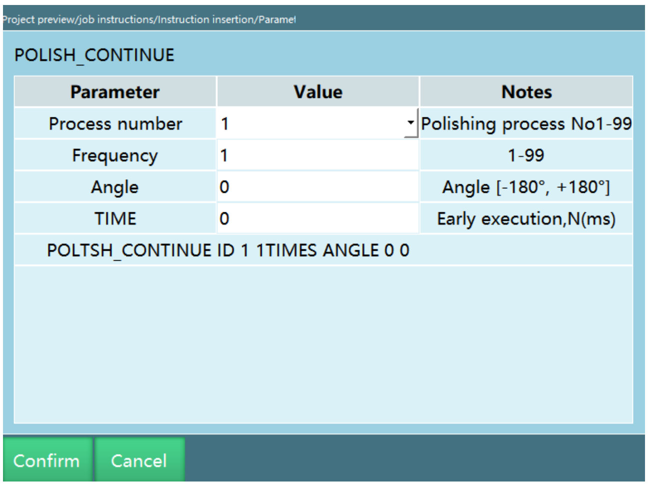
TIME: early execution time, natural number 1-999999

ANGLE: angle parameter, sets the polishing angle of the tool hand when polishing, -180° to +180°

TIMES: polishing times parameter, i.e. the number of times you need to polish, 1-99

ID: process number parameter, you can select the process number for which the polishing parameters have been set in the polishing process, 1-99

POLISH_CONTINUE (continue polishing) instruction



The main purpose of the POLISH_CONTINUE instruction is to facilitate the operator to check for leaks and fill in the gaps. During polishing, some parts may not be able to be polished well in the process, so this function is added to compensate for possible errors in some parts.

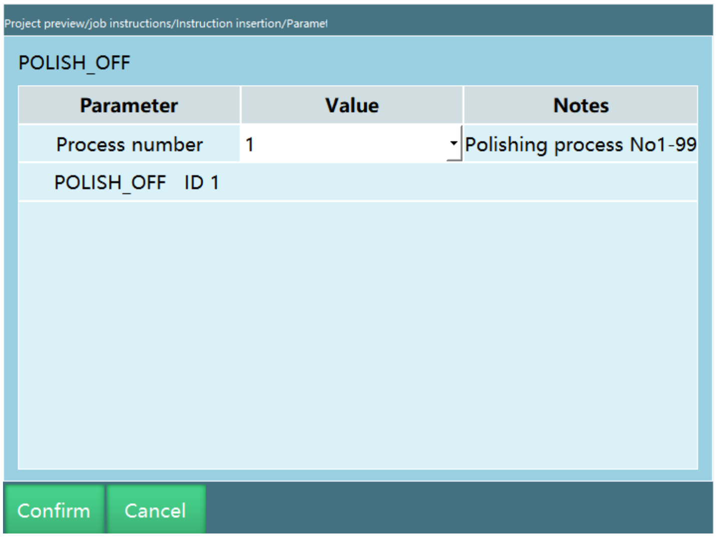
Process number: select the process number for which the polishing parameters have been set in the polishing process

Times: polishing times parameter, i.e. the number of times you need to polish, 1-99

Angle: set the polishing angle of the tool hand when polishing, -180° to +180°

TIME: early execution time, natural number 1-999999

POLISH_OFF (end polishing) instruction



End polishing instruction, run this instruction to end the polishing process.

Note: The whole process needs to be used together with the edge polishing. The polishing operator should do a good job of safety protection and examination data handover before the shift, prepare enough auxiliary materials such as abrasive sheets, steel wire wheels, sand paper and atomic ash, and check whether the operation of the abrasives is normal. The polishing operator must use the abrasives correctly when polishing to ensure the safety of use.

Usage scenarios

Scenario 1

Polish a straight line

Polishing times: 1, polishing angle: 0 degrees (the angle of the current teach point), start polishing

After polishing, wait for the signal to continue polishing

The template is as follows:

```
NOP
INT I001 = 0                                                                   For the subsequent WHILE loop
MOVJ P001 VJ = 10 %  PL = 0 ACC = 10 DEC = 10                                  Run to safety point
POLISH_EDGE P002 V = 10mm/s PL = 0 ACC = 10 DEC = 10 T = 1 ID = 1 ANGLE = 0    Edge polishing start
WHILE (I001 == 0)                                                              Inner loop
WAIT(DIN2 == 1) T = 10                                                         Wait, judge whether to continue polishing
IF (DIN1 == 1)                                                                 Judge, receive signal to continue polishing
POLISH_CONTINUE ID = 1 TIMES = 1 ANGLE = 0                                     Continue polishing
ELSEIF (DIN1 == 0)
JUMP *E                                                                        Insert label, jump out of loop
ENDIFDIN1==0时,                                                                Jump out of while loop when jumping out of if loop
ENDWHILEI001≠0
LABEL *E                                                                       Label, jump out of position
POLISH_OFF ID = 1                                                              End polishing
END
```

Scenario 2

Polish a straight line: polish 4 times at the teaching position, 2 times at a 15-degree angle in the positive direction, and 2 times at a 15-degree angle in the negative direction.

The template is as follows:

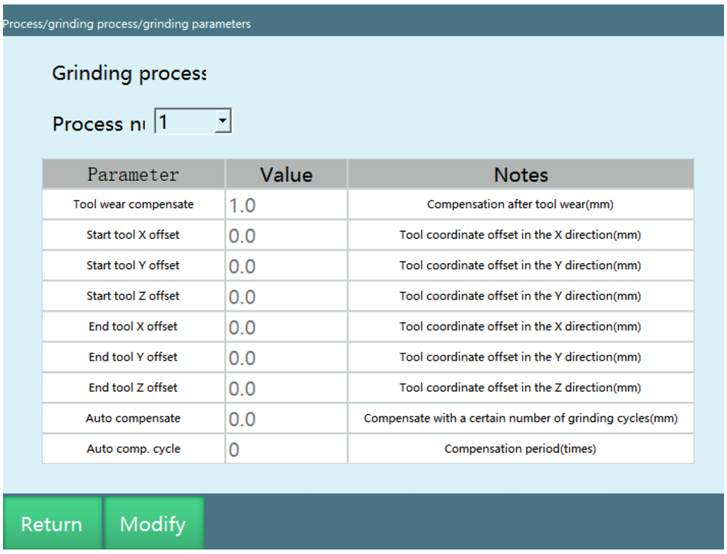
```
NOP
MOVJ P001 VJ = 10 %  PL = 0 ACC = 10 DEC = 10                                  Run to safety point
MOVL P001 V = 100mm/s  PL = 0 ACC = 10 DEC = 10                                Run to the start of the polishing line
POLISH_EDGE P002 V = 10mm/s PL = 0 ACC = 10 DEC = 10 T = 4 ID = 1 ANGLE = 0    Edge polishing start
POLISH_CONTINUE ID = 1 TIMES = 2 ANGLE = 15                                    Continue polishing
POLISH_CONTINUE ID = 1 TIMES = 2 ANGLE = -30                                   Continue polishing
POLISH_OFF ID = 1                                                              End polishing
END
```

Scenario 3

The polishing head is worn by 1mm, and the parameters need to be adjusted

Setup steps

1. Go to "Process/Polishing process/Polishing parameters", select the corresponding process number and click "Modify"
2. Fill in 1 for "Tool wear compensation" and click "Save"



Parameter	Value	Notes
Tool wear compensate	1.0	Compensation after tool wear(mm)
Start tool X offset	0.0	Tool coordinate offset in the X direction(mm)
Start tool Y offset	0.0	Tool coordinate offset in the Y direction(mm)
Start tool Z offset	0.0	Tool coordinate offset in the Y direction(mm)
End tool X offset	0.0	Tool coordinate offset in the X direction(mm)
End tool Y offset	0.0	Tool coordinate offset in the Y direction(mm)
End tool Z offset	0.0	Tool coordinate offset in the Z direction(mm)
Auto compensate	0.0	Compensate with a certain number of grinding cycles(mm)
Auto comp. cycle	0	Compensation period(times)

3. After the setup is complete, run the program

Scenario 4

Polish a straight line: polish 4 times at the teaching position, 2 times at a 15-degree angle in the positive direction by laser searching

The template is as follows:

```
NOP
MOVJ P001 VJ = 10 %  PL = 0 ACC = 10 DEC = 10                                    Run to polishing safety point
MOVL G001 V = 100mm/s  PL = 0 ACC = 10 DEC = 10                                  Run to the start of the polishing line
POLISH_EDGE G002 V = 10mm/s PL = 0 ACC = 10 DEC = 10 T = 4 ID = 1 ANGLE = 0      Edge polishing start
MOVJ P004 VJ = 10 % PL = 0 ACC = 10 DEC = 10                                     Run to searching safety point
SEARCH_START ID = 1 TYPE = 0                                                     Search start
MOVL P002 V = 10 mm/s PL = 0 ACC = 1 DEC = 1                                     Move to P002
SEARCH_STATIC ID = 1 1 GP001 0.1                                                 Store the static search results into GP001
MOVL P003 V = 10 mm/s PL = 0 ACC = 1 DEC = 1                                     Move to P003
SEARCH_STATIC ID = 1 1 GP002 0.1                                                 Store the static search results into GP002
SEARCH_END ID = 1                                                                Search end
POLISH_CONTINUE ID = 1 TIMES = 2 ANGLE = 15                                      Continue polishing
POLISH_OFF ID = 1                                                                End polishing
END
```

Vision Process

Catalogue

Vision Process

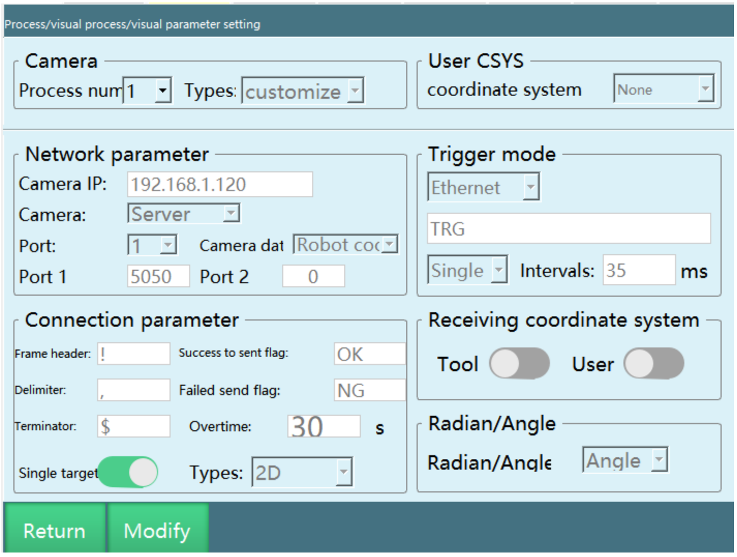
Vision process

It is not difficult for an industrial robot to perform repetitive motions, but if faced with a disordered environment, it can utilize the vision process to identify, analyze and judge the environment. The vision process has the advantages of precise positioning, convenient deployment, easy to use, flexible parameter configuration methods, and rich instructions. We often combine it with the conveyor belt process, which can effectively reduce the tracking error rate by using simple programming.

Vision parameter setting

Select "Vision process" and complete the settings of vision parameters/vision position/vision range/position debugging

The vision parameters are set as follows:



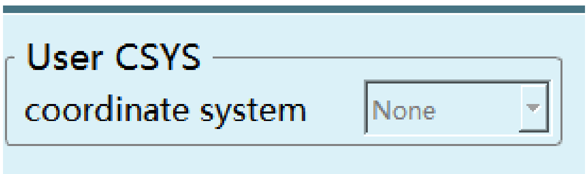
Camera selection

Process number: 1-99 process numbers are provided, each of which stores all the vision parameters and vision position parameters below that process number.

Type: Currently, only "Custom" type is supported, and users can set the parameters according to their needs.

User coordinate system

The vision system supports mapping the vision points to the user coordinate system, i.e. the points sent by the camera are the points in the vision coordinate system. Here you need to select a user coordinate system that has been matched with the camera.



If you select "None", the default is that the camera sends the points in a Cartesian coordinate system; the user can also select their own calibrated user coordinate system (the selected user coordinate system is calibrated in the "Settings - User coordinate calibration" page)

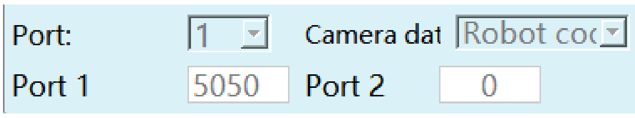
Network parameters

Camera IP: If the camera is used as the vision server, fill in the IP address of the camera here. The IP address of the camera and the IP address of the controller should be consistent with the first three bytes (counting from left to right), and the last one should be different, for example, both use the same network segment: 192.168.1.xxx.

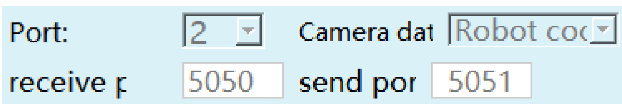
Camera: "Client" and "Server" can be selected here. If the camera is selected as the client, then the controller is the server, and the camera needs to be actively connected.

Camera data: Here you can choose "Robot coordinates" and "Pixel coordinates". If you choose "Robot coordinates", the data sent by the camera is the coordinates of the robot; if you choose "Pixel coordinates", the data sent by the camera is the pixel coordinates in the camera coordinate system

Number of ports: If the vision server uses the same port for data sending and receiving, the number of ports is 1



If different ports are used for data sending and receiving, then the number of ports is 2



Port 1 is to receive data; port 2 is to send data (port number cannot be set to the same value)

Port 1: Generally use 5050

Connection parameters

The frame header, separator and terminator cannot be set to the same character at the same time

The frame header and terminator can be set to blank except for the separator

Frame header: The beginning of the signal transmission. This must be the same as the parameter of the camera configuration.

Separator: Used to separate multiple signals when transferring them. This must be the same as the parameter of the camera configuration. **(This cannot be set to null)**

Terminator: The symbol that determines the end of signal transmission. This must be the same as the parameter of the camera configuration.

Successfully sent identifier: After the camera has taken the photo and successfully identified it, a success identifier will be sent after sending.

Failed to send identifier: If the camera has taken the photo but failed to identify it, a failure identifier will be sent.

Note: The above parameters are user-defined

For example, set the frame header to blank, set the separator to ",", set the terminator to "$", and turn on the "Single target" enable switch

The data format is: ,X,Y,Rz,$

Timeout: When this time is exceeded, it is determined that the connection has timed out and the connection is stopped. When it is filled as 0, there is no limit.

Single target : Turn on the enable switch, the camera will identify only one target point at a time.

Type: 2D, 2D+Height, 3D; e.g. the camera sends a string (frame header "**!**", separator "**,**", terminator "**$**"):

2D: The data format is: **!,X,Y,Rz,$**

2D+Height: The data format is: **!,X,Y,Rz,h,$**

3D: The data format is: **!,X,Y,Z,A,B,C,$**

Turn off the"Single target" enable switch: More than one target point can be identified, and the example N represents the number of identified target positions.

Type: 2D, 2D+Height, 3D; e.g. the camera sends a string (frame header "**!**", separator "**,**", terminator "**$**"): the N represents the number of identified target positions.

2D: The data format is: **!,N,X,Y,Rz,X,Y,Rz,$**

2D+Height: The data format is: **!,N,X,Y,Rz,h,X,Y,Rz,h,$**

3D: The data format is: **!,N,X,Y,Z,A,B,C,X,Y,Z,A,B,C,$**

Trigger method

I/O: Give the camera a trigger signal through the I/O board, here you need to set the DIN (IO input) signal port of the I/O.

Ethernet: Generally, the default trigger method is Ethernet. When the camera receives the "TRG" (or user-defined string) here, it should reply the coordinate value to the controller.

Trigger condition

Single trigger: When the condition is single trigger, the camera will be triggered once each time you run the VISION_TRG instruction in the program.

Continuous trigger: When the condition is continuous trigger, the camera will be triggered continuously each time you run the VISION_TRG instruction in the program.

Interval time: The time interval during continuous triggering (triggering cycle);

Receiving coordinate system

The received point information is the point information sent by the camera with a specific tool hand under a specific user coordinate system

Tool: When this enable switch is turned on, the point sent by the camera contains the tool hand used (used when working with multiple tool hands)

User: When this enable switch is turned on, the point sent by the camera contains the user coordinate system used (used when there are multiple workbenches)

Note: Before turning on the "Tool" and "User" enable switches, the "Hand-eye calibration user coordinate system" cannot be "None" (If both are off, it needs to be set to "None"), and the "User" and "Tool" enable switches can be turned on/off at the same time.

Angle/radian setting

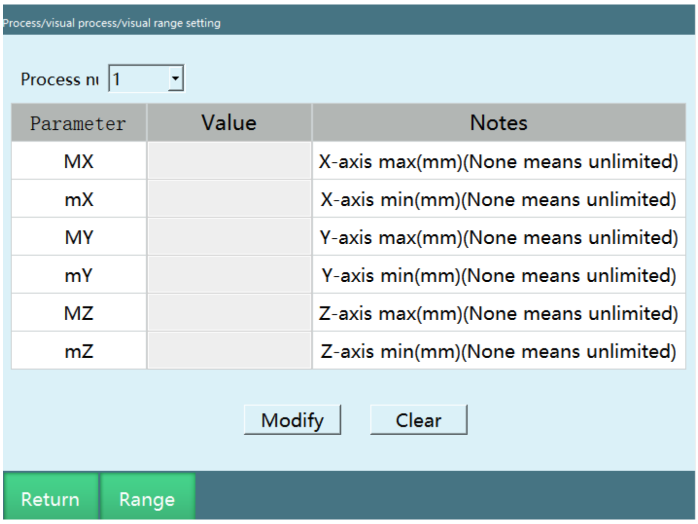
Select the unit type for the A/B/C axis in the "Vision position parameters", the unit of radian is rad, and the unit of angle is º (degree).

Note: The setting of the angle/radian affects the content of the analytical data and has nothing to do with the angle/radian switch in the operation

parameter. The angle/radian setting in the operation parameter only affects the display of the content about the angle/radian on the teach pendant

Vision range setting

Enter the "Vision range setting" interface from "Process" - "Vision process" - "Vision range setting".

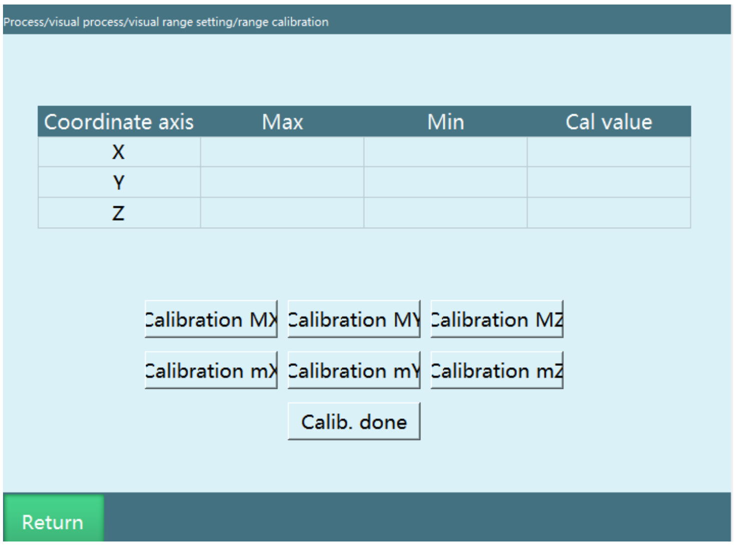


In order to avoid that the address parameters returned by the camera exceed the range that the robot can reach, the maximum range that the robot can reach is specified. If the parameters returned by the camera are out of range, the data will be automatically filtered, and the data will not take effect.

You can complete the calibration by manual teach or fill in the corresponding values directly.

Process number: 1-99 process numbers are provided, each of which stores all the vision range parameters below that process number.

Range calibration: Calibrate the maximum and minimum values of the XYZ axes in the Cartesian coordinate system



Calibrate MX: Calibrate the X-axis maximum value

Calibrate mX: Calibrate the X-axis minimum value

Calibrate MY: Calibrate the Y-axis maximum value

Calibrate mY: Calibrate the Y-axis minimum value

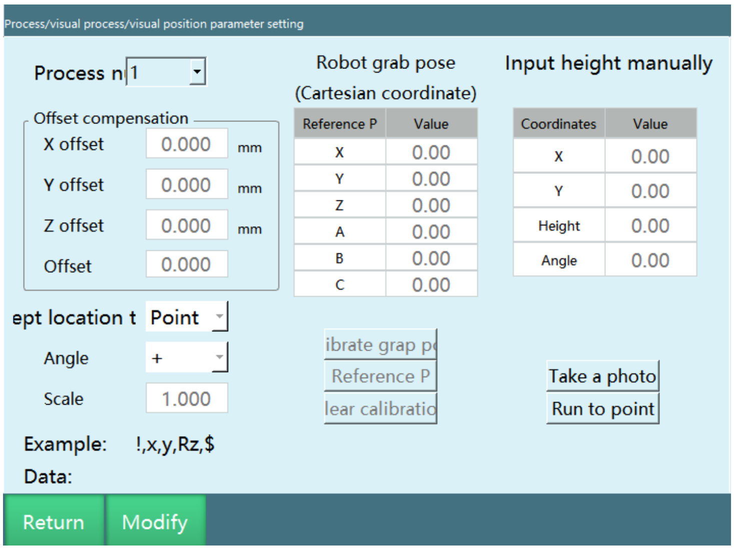
Calibrate MZ: Calibrate the Z-axis maximum value

Calibrate mZ: Calibrate the Z-axis minimum value

Calibration completed: Click to record all calibrated values in the maximum and minimum values.

Vision position parameters

Enter the "Vision position parameters" interface from "Process" - "Vision process" - "Vision position parameters".



Offset compensation: If the robot's gripping position is offset from its actual position in a fixed direction every time, please fill in the compensation here and it will be automatically compensated to the correct position.

Scale factor: If the position value sent by the camera is reduced by a specific ratio, you need to fill in the scale factor here. For example, if the camera sends a value of (300,200,100) and the actual position is (3,2,1), then 0.01 should be entered here.

Calculation formula: scale factor = actual position value/position value sent by camera

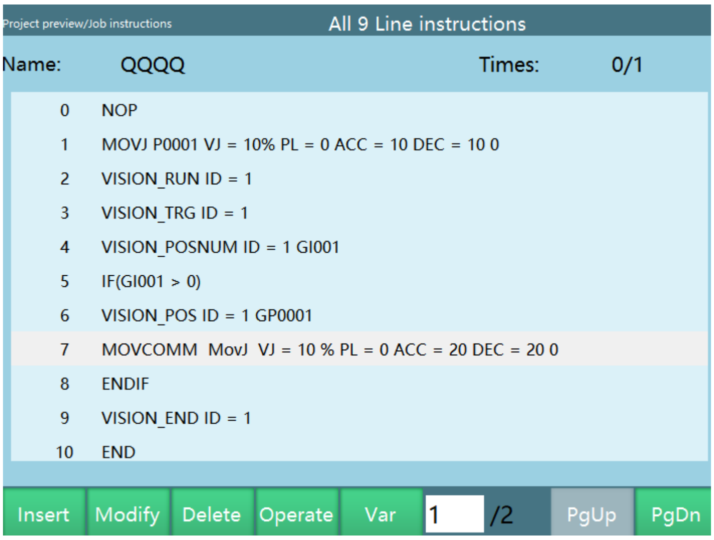
Angle direction: The direction of the points sent by the camera can be the same as or opposite to the direction of the rotation angle of the robot

Receiving position type: Point/trajectory

When selecting "Point", the camera takes photos and sends the points to the controller

When selecting "Trajectory", the camera will identify the trajectory and send a series of points during dispensing or spraying, **and the trajectory is run through an external point instruction**. If "Trajectory" is selected, **"Single target"** in the "Vision parameter setting" will only identify one segment of trajectory, if "Single target" is turned off, then the camera will identify multiple segments of trajectory

When "Trajectory" is selected for the "Receiving position type", the program job file is as follows:



Reference point and height

Calibrate gripping attitude: Here you need to mark the end attitude of the robot when gripping the object. After the calibration, every grip will be done with this attitude. (**The XYZ values here do not affect the position at the time of gripping**)

Run to reference point: Run to the point that was calibrated when the gripping attitude was calibrated.

Clear calibration: Clear the point data used to calibrate the gripping attitude.

Camera coordinates: If the camera cannot send the gripping height, you need to fill in the gripping height Z in the table on the right. If the camera can send the gripping height, the setting here has no effect. After setting, press and hold the DEADMAN button to power on, click the [Try photo] button to take a photo for test, the data sent by the camera will be displayed at the "Camera coordinates" and "Receiving data" parts. Press and hold the DEADMAN button to power on after taking a photo, click the [Run to this point] button, move the robot to the photo-taking position to verify whether it is accurate.

Try photo: Power on the servo, click "Try photo", open the network connection, and send the data according to the sample format

Sample format: Verify the arrangement according to the connection parameters already set in the "Vision parameter setting". For example, if the frame header is W, the separator is # and the terminator is $ in the connection parameters, and the height information is sent, then the format is W#x#y#angle#h#$

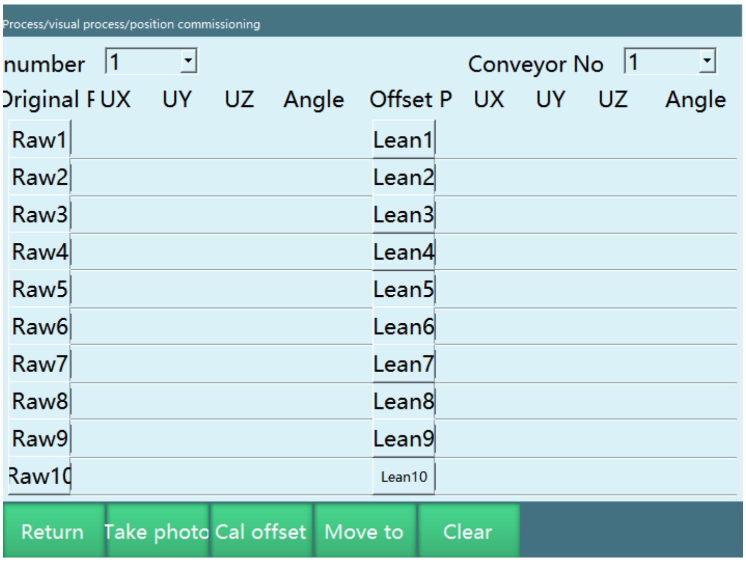
Receiving data: W#x#y#angle#h#$

Run to this point: The robot moves to the position sent by the camera

Position debugging

It is used in combination with the conveyor belt process for the debugging of the conveyor belt. After the camera takes a photo, it will send a point data to be stored in **[Original point]**, but the workpiece will be taken out by the conveyor for a distance, click "Calculate offset", the calculated position after offset will be stored in [**Point after offset**], click [Move here], and the robot will go directly to the calculated position after offset.

Enter "Process" - "Vision process" - "Position debugging", use vision process plus conveyor tracking process to debug the conveyor belt



Process number: The process number of the vision process.

Conveyor process number: The process number of the conveyor belt to be debugged.

Take photo: Press and hold the DEADMAN button to power on and click the [Take photo] button to take a photo for test, the position data sent by the camera will be displayed at [**Original point**] on the left side.

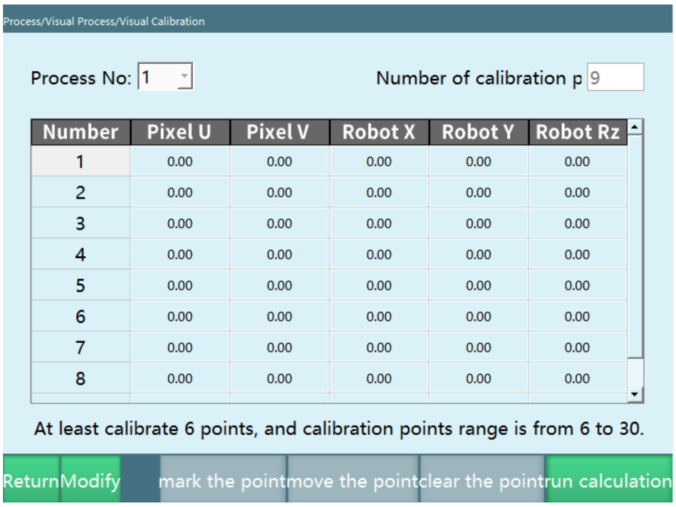
Move here: After taking a photo, press and hold the DEADMAN button to power on, select the corresponding point and click the [Move here] button, the robot will move to the position sent by the camera.

Calculate offset: After taking a photo, start the conveyor belt so that the workpiece is transported for a certain distance, and click "Calculate offset", the [**Point after offset**] on the right side will display the workpiece point after offset.

Clear: Clear all points.

Vision calibration

Vision calibration refers to installing the camera on the tool hand, marking the current point information of the robot, running the calculation, triggering the camera to go to the current point of the robot, and obtaining the corresponding pixel data. When all pixel data are obtained, the conversion relationship between camera data and robot points is calculated. The camera points sent subsequently can be converted into the actual motion points of the robot through the conversion relationship.



Process number: Process number of the vision process

Number of calibration points: The number of points to be taught; the range of the number of calibration points is 6-30

Mark this point: Record the current robot point data

Move here: Press and hold the DEADMAN button to power on the robot, select the serial number and click the [Move here] button, the robot will move to the point marked by the serial number

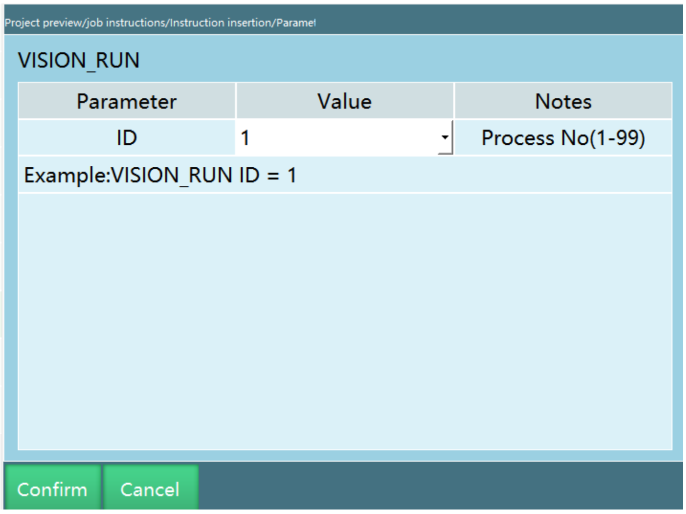
Clear this point: Clear the point data of the selected serial number, but do not clear the pixel data, the pixel data will be overwritten by the newly captured data after running the calculation

Run calculation: Click "Run calculation", the robot will move according to the point data taught before, and each time it moves to a point, it will trigger a photo taking and record the current pixel data

Vision instructions

VISION_RUN

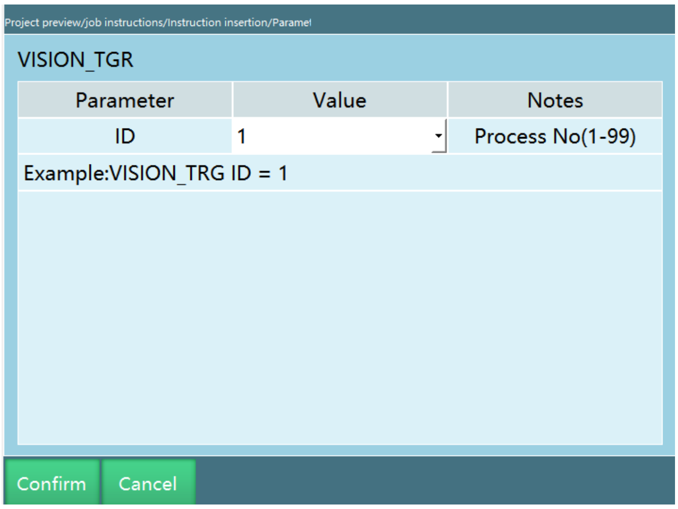
Vision start instruction



Run this instruction to connect the controller and the camera

VISION_TRG

Vision trigger instruction



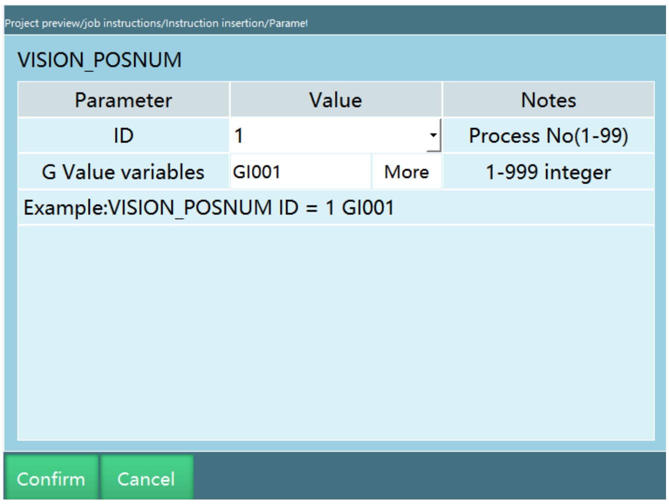
The controller issues a trigger signal after running this instruction

The specific trigger method is set in the "**Vision process-Vision parameter setting"** interface: 1. Select "IO": run this instruction to send out the corresponding IO signal; 2. Select "Ethernet": run this instruction to send a custom string to the camera.

After running this instruction, it will wait for the return value (position data sent) from the vision server, and continue to run the next instruction after obtaining the position.

VISION_POSNUM

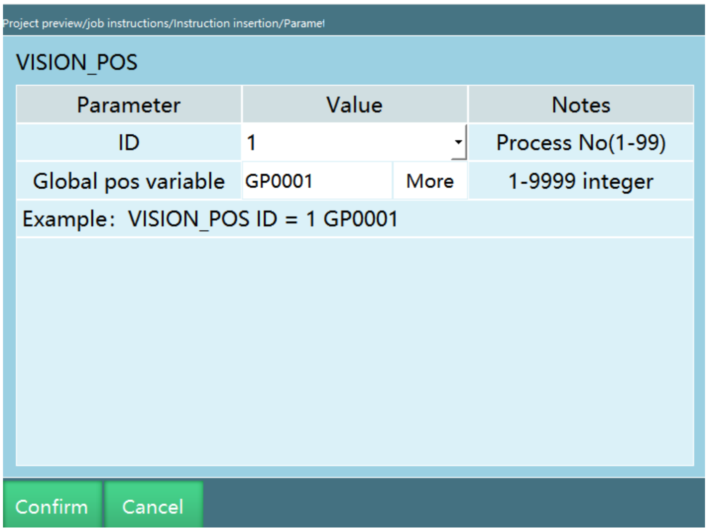
The instruction to get the number of vision positions



When the "Single target" enable switch is turned off in [Process-Vision process-Vision parameter setting], record the number of points sent by the camera. Every time this instruction is run, the number of points will be reduced by one

VISION_POS

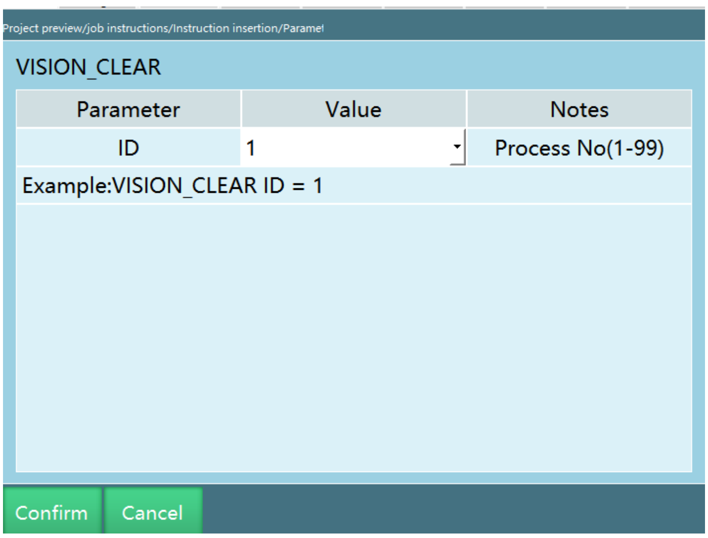
The instruction to get vision position



The point information sent by the camera is successively cached in the position of GP0001, for example, when two points are sent, the first time the instruction is run, the GP0001 stores the point 1, and the second time the instruction is run, the GP0001 stores the point 2

VISION_CLEAR

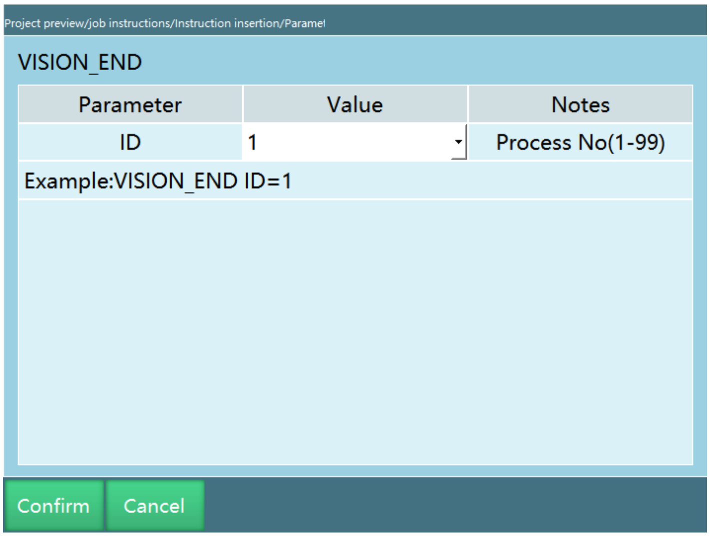
The instruction to clear vision position information



The cleared point information is the point sent by the camera in the **Vision position parameter**

VISION_END

Vision end instruction



End the vision process and the controller is disconnected from the camera.

Use cases

Gripping application

After the camera takes a photo of the material, it sends the data to the robot, then the robot goes to grip the material

Programming:

iNexBot

Spraying Process

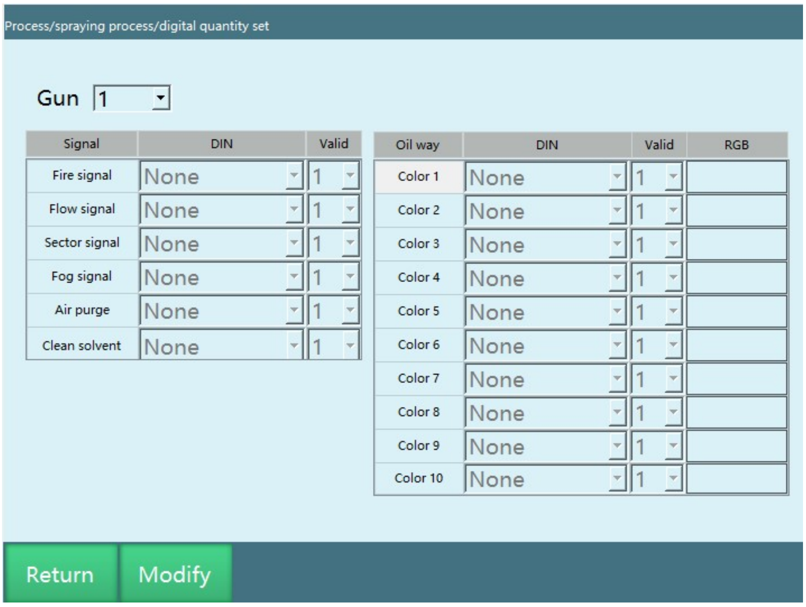
Catalogue

Spraying process

Robot Control System Operation Manual - Spraying Process

Digital setting

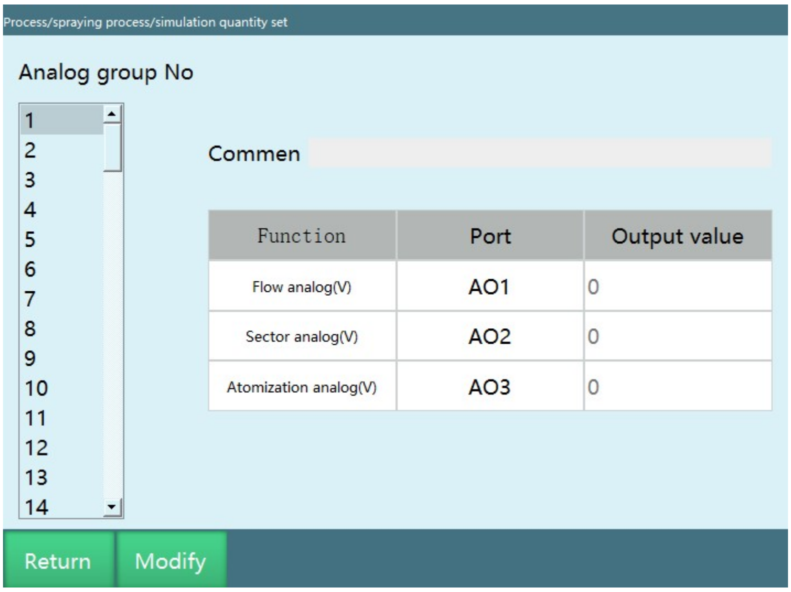
Turn on the controller, enter the "Process" interface, select "**Spraying process**" - "**Digital setting**" interface, click the "Modify" button to modify it, and click "Save" after modification. (As shown below)



After clicking "Modify", the "Modify" button changes to "Save", and the selection box turns white. At this time, you can select the gun number, and select the port, effective value and color number after the respective functions. Please use the 16-bit RGB format for the color number. After filling in the color number, the corresponding "Color oil circuit" box will change to the corresponding color.

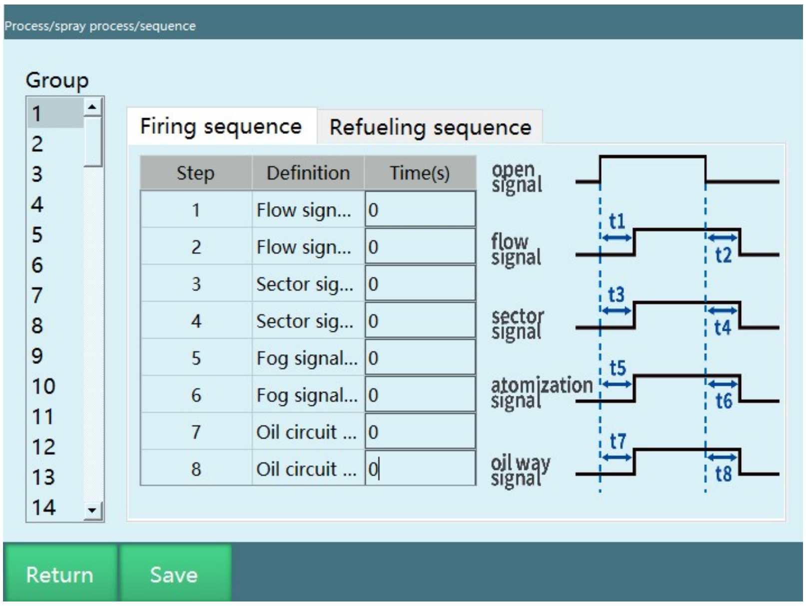
Analog setting

Select the group number you want to modify, and click the "Modify" button to modify the analog group number and fill in the notes. A total of 99 groups of sequence and their corresponding notes can be set, each group of sequence includes flow analog, sector analog and atomization analog, here is only for modification, you need to use the corresponding instruction to call the corresponding group number, and click "Save" after modification. (As shown below)



Sequence

Select the group number you want to modify, and click the "Modify" button to modify the sequence group number. A total of 99 groups of sequence can be set. Each group of sequence includes firing sequence and refueling sequence, here is only for modification, you need to use the corresponding instruction to call the corresponding group number, and click "Save" after modification. (As shown below)



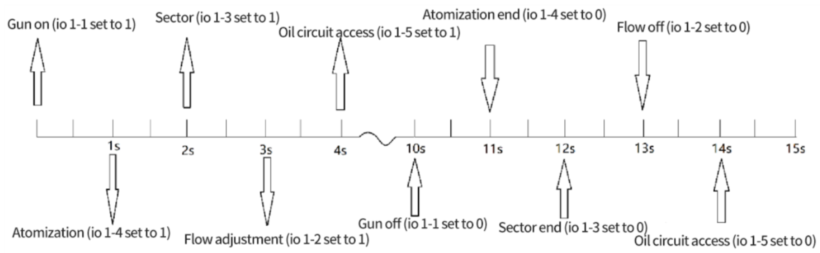
Firing sequence

This is the firing sequence, and the time to be set for each signal on the left corresponds to the sequence diagram on the right. The flow, sector, and atomization signals correspond to the ports set in the digital setting, and the oil circuit signals correspond to the ports set in the digital settings for the current color. (As shown below)

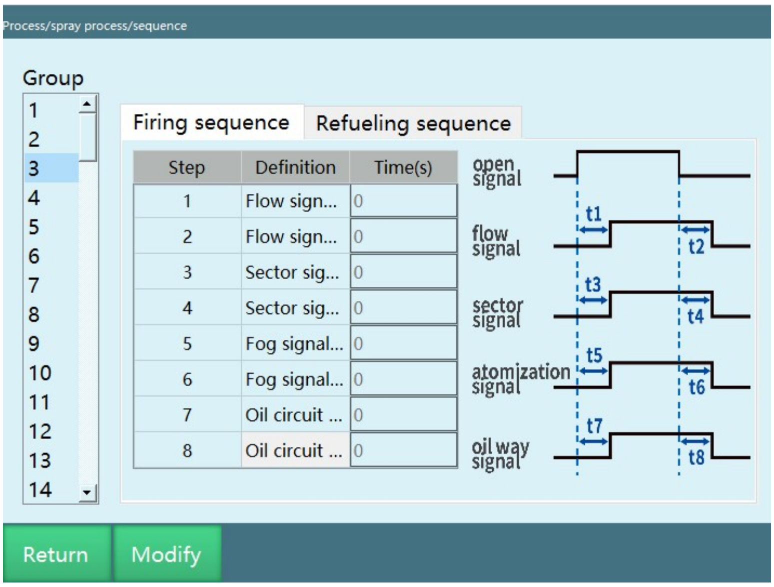
Example: (The example is only for illustration, please set according to the actual needs) Set the fire signal 1-1, sector signal 1-3, atomization signal 1-4, color oil circuit 1-5 on the IO board. The spraying time is 10s (that is, the firing time is 10s), t1=1, t2=1, t3=3, t4=3, t5=2, t6=2, t7=4, t8=4.

After spraying starts:

0s fire signal 1-1 output is 1	After 10s, fire signal 1-1 output is 0
After 1s, atomization signal 1-4 output is 1	After 11s, atomization signal 1-4 output is 0
After 2s, sector signal 1-3 output is 1	After 12s, sector signal 1-3 output is 0
After 3s, flow signal 1-2 output is 1	After 13s, flow signal 1-2 output is 0
After 4s, color oil circuit 1-5 output is 1	After 14s, color oil circuit 1-5 output is 0



Refueling sequence

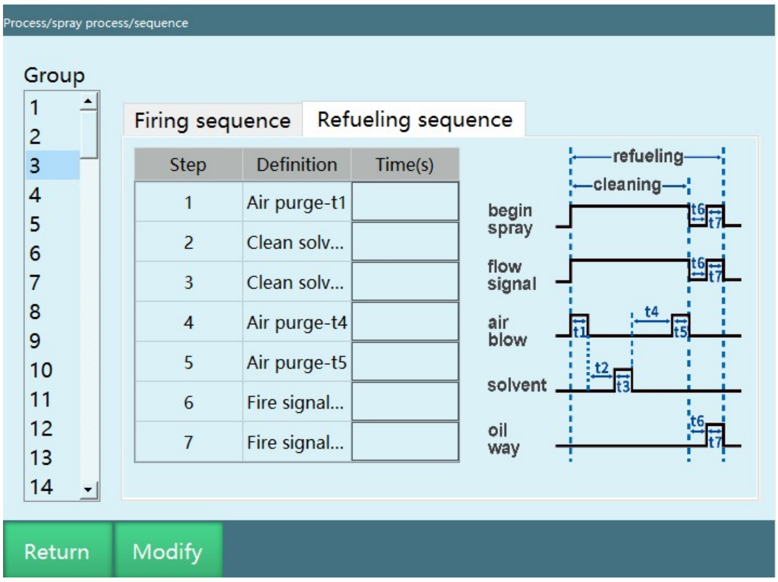


This is the refueling sequence, and the time to be set for each signal on the left corresponds to the sequence diagram on the right. The air purge, clean solvent, and fire signals correspond to the ports set in the digital setting. (As shown below)

Example: (The example is for illustration only, please set according to actual needs) Set gun signal 1-1, flow signal 1-2, air purge 1-3, clean solvent 1-4, color oil circuit 1-5 on the IO board. The firing time is 10s, t1=1, t2=3, t3=1, t4=4, t5=1, t6=3, t7=4.

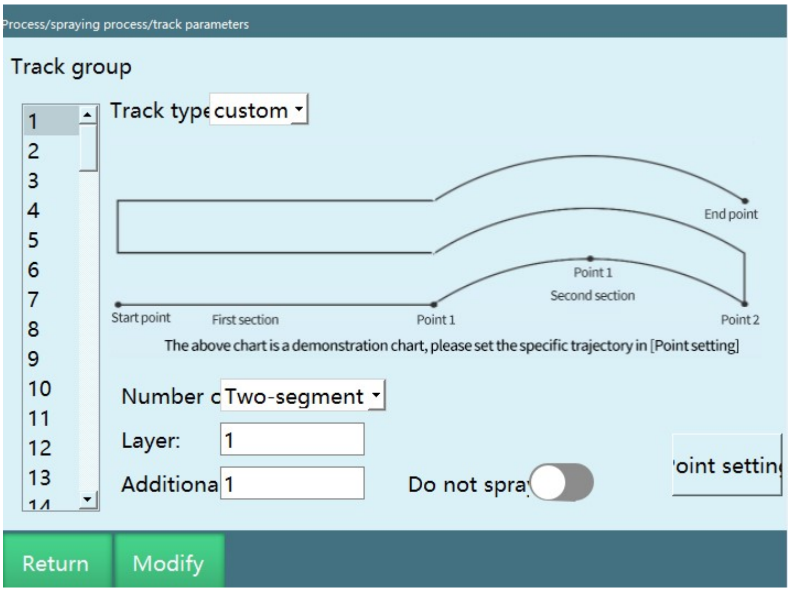
Refueling process:

0s fire signal 1-1 output is 1	9s air purge 1-3 output is 1	13s fire signal 1-1 output is 1
0s flow signal 1-2 output is 1	10s air purge 1-3 output is 0	13s flow signal 1-2 output is 1
0s air purge 1-3 output is 1	10s fire signal 1-1 output is 0	13s oil circuit signal 1-5 output is 1
1s air purge 1-3 output is 0	10s flow signal 1-2 output is 0	17s fire signal 1-1 output is 0
4s clean solvent 1-4 output is 1		17s flow signal 1-2 output is 0
5s clean solvent 1-4 output is 0		17s oil circuit signal 1-5 output is 0



Track parameters

A total of 99 track group numbers can be set. Each **track group number includes track type, track kind, number of spray layers, additional times, and mark points**. Click the "Modify" button to modify it, and click "Save" after modification. (As shown below)



- Spraying type: divided into three types: flat, solid and custom, you can set it according to your needs.

- Spraying kind: four kinds of flat spraying and two kinds of solid spraying, you can set it according to your needs.

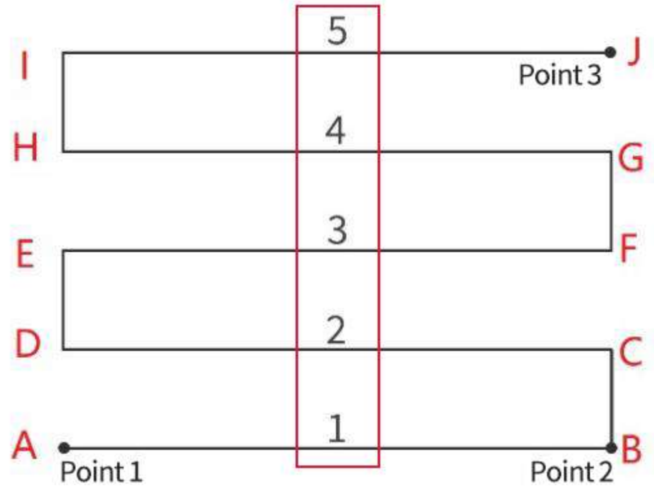
- Number of layers: The number enclosed in the red box in the picture below is the number of layers. Fill in the number and the corresponding number of layers will be sprayed.

- Additional times: Additional spraying times per layer, for example, if "Additional times" is set to 3, then spray each layer back and forth 4 times before entering the next layer.

- Mark points: The mark points correspond to the point on the right side of the graph. The first/second kind of flat type needs to mark three points, the third/fourth kind of flat type and the first/second kind of solid type need to mark four points.

Example: (The example is only for illustration, please set according to actual needs) Set the number of layers to 1, the additional times to 0, then the spray gun will spray from point A to point B;

Set the number of layers to 1, additional times to 1, then the spray gun will spray from point A to point B and then spray back to point A;

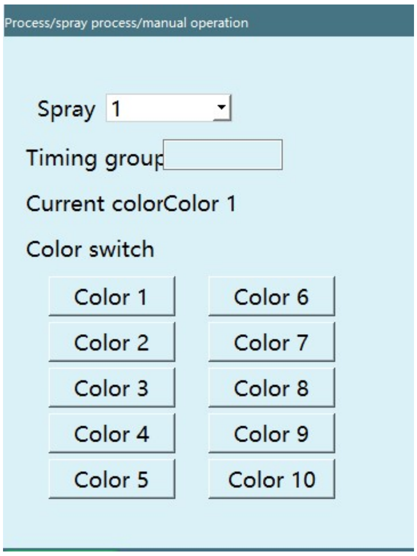


Set the number of layers to 2, additional times to 1, then the spray gun will operate in the sequence of points A → B → A → D → C → D;

Set the number of layers to 3, additional times to 2, then the spray gun will operate in the sequence of points A→B→A→B→C→D→C→D→E→F→E→F.

Manual operation

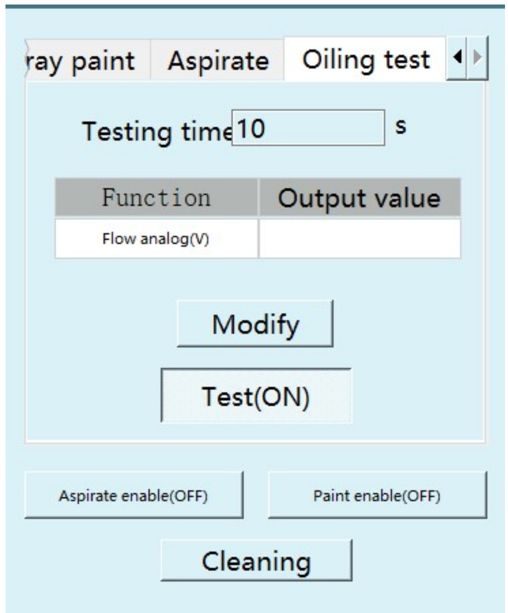
In the "Manual operation" interface, you can choose the spray gun number and sequence group number to be used, and click the corresponding color at the "Color switch" part to change the current color (the corresponding IO port will be set according to the sequence-refueling sequence). (As shown below)



When the value of the "Analog group number" input box is 0, the button "Modify analog" is valid and can be clicked and manually modified. When it is not 0, the button "Modify analog" in the spray paint, aspirate, and oil test is invalid and grayed out. The corresponding three values become the values in the input analog group number. The analogs in the spray paint, aspirate, and oil test all use the 3 analogs set in the spray paint.

(As shown below)

The "Test" button of the oil test is OFF by default. After setting the test time, press it to perform the oil test for the corresponding time. At this time, the io port of the oil circuit of the current color will become a valid value. (As shown below)



"**Aspirate enable**" and "**Paint enable**" are OFF by default, and they will switch to ON when pressed. (As shown below)

Press "**Aspirate enable"**, the IO port corresponding to "Digital setting - Air purge" will become a valid value; press "**Paint enable**", the corresponding IO port will be set according to the sequence-firing sequence; press "**Cleaning**"

*"Color switch", "Aspirate enable", "Paint enable", "Cleaning", and "Oil test" are interlocked, and only one function can be used at the same time. For example: if you press "**Cleaning**" when the "**Paint enable**" is ON, it will stop painting immediately for cleaning.

Spraying instructions

SPRAY_ON - Spraying start

The instruction to identify the start of spraying, run this instruction to start the spraying process.

Function: spraying process start

Gun: Gun 1-2

Sequence group number: fill in the sequence group number

Analog group number: fill in the analog group number

Flow analog, sector analog, atomization analog: modify when the analog group number is 0

Example: SPRAY_ON G=1 T=1 AO=1

SPRAY_OFF - Spraying end

The instruction to identify the end of spraying, run this instruction to end the spraying process

Function: spraying process end

Gun: Gun 1-2

Example: SPRAY_OFF G=1

SPRAY_CHANGE - Spray color change

The instruction to change the spray gun color, run this instruction to make the corresponding spray gun change the corresponding color according to the instruction parameters

Function: change color

Gun: Gun 1-2

Sequence group number: sequence group number 1-99

Color: gun color number 1-10

Example: SPRAY_CHANGE G=1 T=2 COLOR=1

SPRAY_MOVE - Spraying track

Spraying action instruction, spray according to the set track group number, speed, pl and acceleration

Function: Make the robot move according to the spraying track

Track group number: track group number 1-99

Spraying speed: 2-9999mm/s

Spraying PL: position level 0-5

Spraying acceleration: 1-100%

Spraying deceleration: 1-100%

Example: SPRAY_MOVE ID=1 V=10mm/s PL=0 ACC=1 DEC=1

SPRAY_POSE - Spray start position

Change the attitude at the start of spraying. If this instruction is not used, the spray gun will start spraying according to the attitude of the first point during calibration.

Function: switch robot attitude

Track group number: track group number 1-99

Point status: Absolute mark point/Attitude only

Speed: attitude changing speed

Acceleration: attitude changing acceleration

Deceleration: attitude changing deceleration

TIME: early execution time

Example: SPRAY_POSE ID=2 V=40mm/s ACC=4 DEC=4

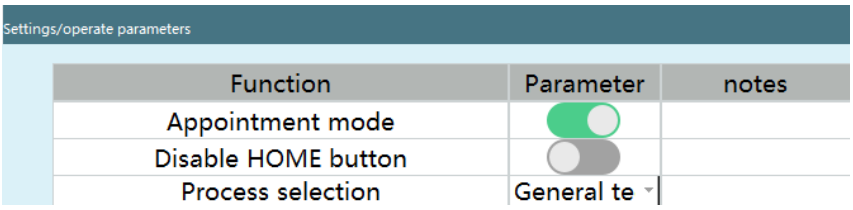
Special Process

Catalogue

Special Process

Interface

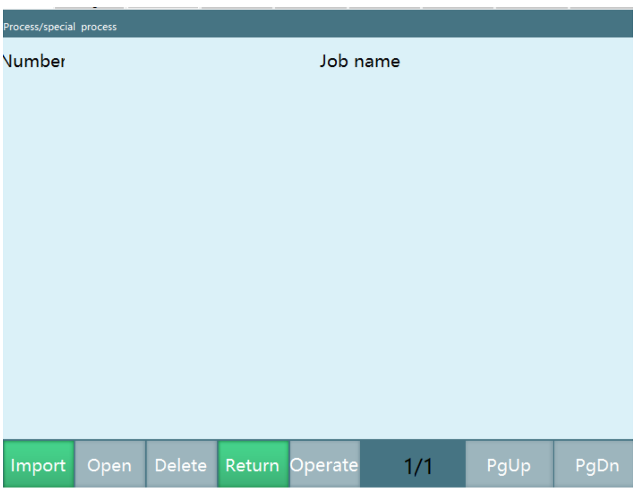
There are five options under "Settings" - "Operation parameters" - "Process selection": General process/Special process/Palletizing process/Welding process/Cutting process, select the "General process" from them

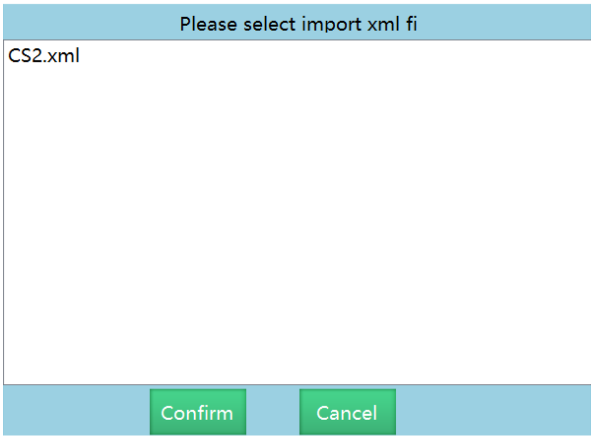


"General process" mode

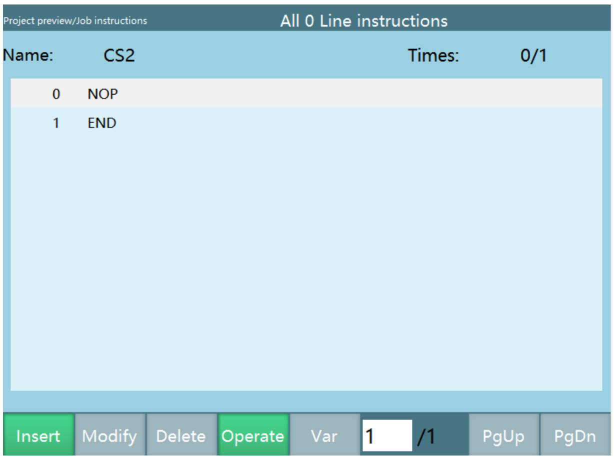
Click "Process" - "Special process"

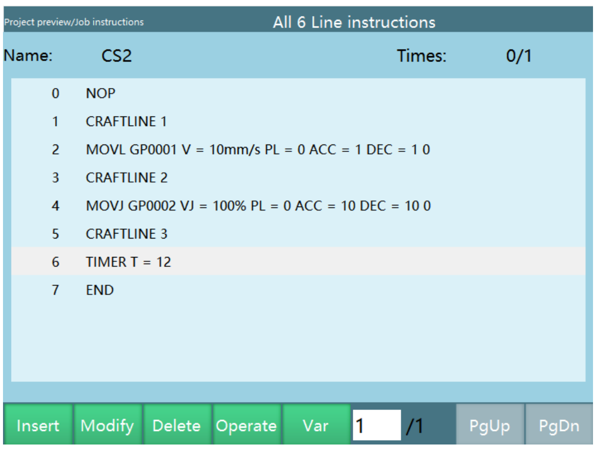
Click "Import" (the .xml file should be placed in the importxml folder of the U disk), select the file to be imported, and click "OK"





Select the file you want to open, click "Open", open it and pay attention to the number of procedures in the XXX.xml file, then switch to general mode, open the same named program and insert the CRAFTLINE instruction (the number of procedures in the XXX.xml file = the number of inserted CRAFTLINE instructions, corresponding to each other)





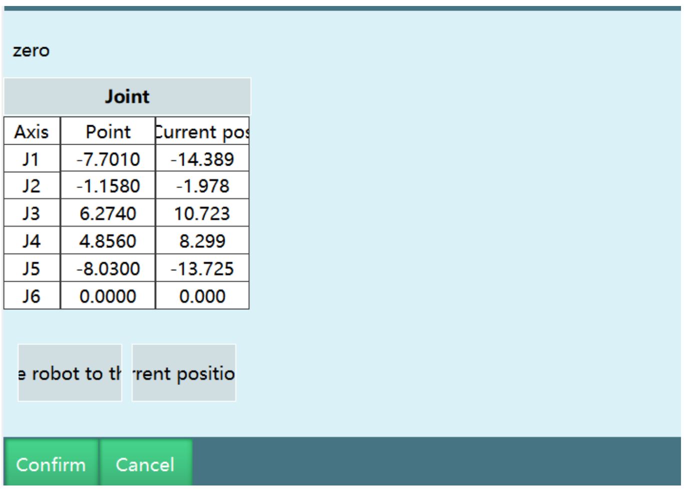
Notes

- Before importing the file, you should create a new program with the same prefix name as the imported XXX.xml in general mode

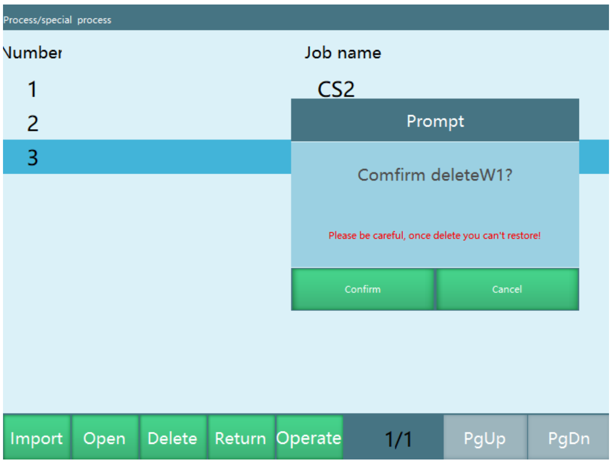
Point modification

Select the instruction that needs to be modified in the "Special process" interface and click "Modify" (if it is not modifiable, then clicking on "Modify" is invalid; if it contains modifiable items, click "Modify" to enter the modification interface)

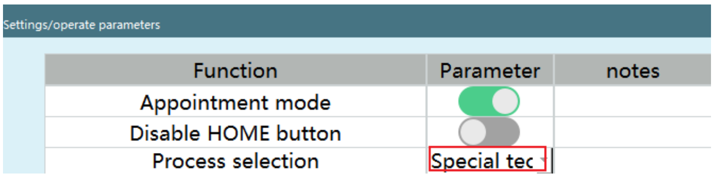
Click on "OK", all changes will be saved; click on "Cancel", all changes will not be saved.



If you want to delete a file, select the file you want to delete and click "Delete"



"Special process" mode



Click "Process" - "Special process", all the interfaces and functions are the same as in the general mode.

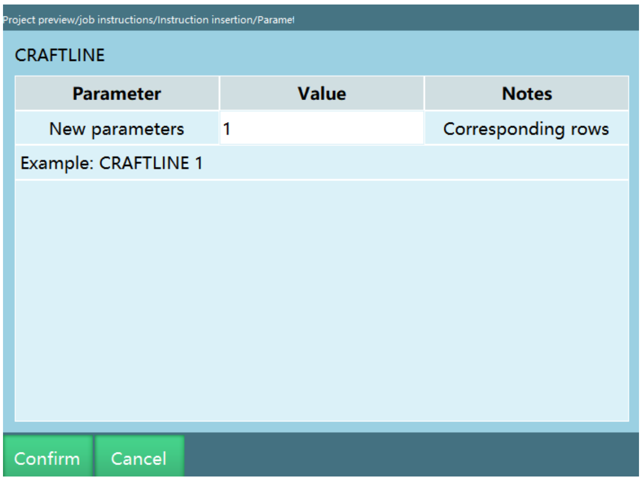
Click "Project", the interface is as follows, compared with the general process, it lacks the "Return" key

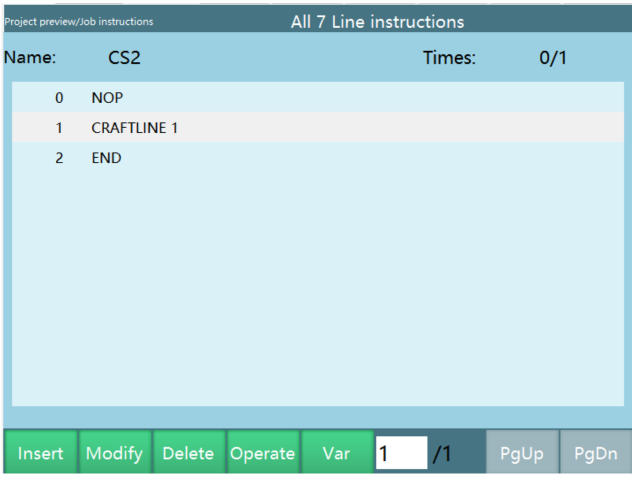
Click "Program", the interface is as follows, compared with the general process, it lacks the "Return" key

Programming

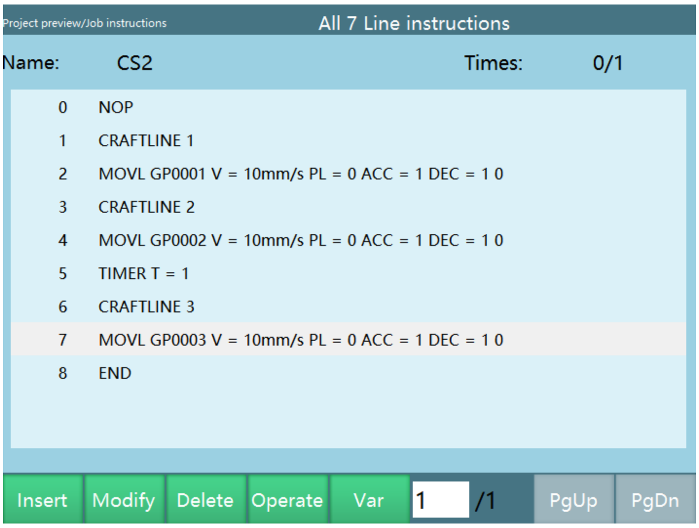
Write the program in general mode;

1. Create a new program (the program name of the new program must be consistent with the program name in the XML file code and the name of the job file; for example, if the program name is JOB1, then the XML file name is JOB1 and the RelationJobName is "JOB1" in the XML file)
2. CRAFTLINE instruction insertion: click "Insert", select the conditional control class, select "CRAFTLINE", click "OK", enter the corresponding number of lines, click "OK". (Be sure to enter the corresponding number of lines)





3. Insert a CRAFTLINE instruction, then insert other instructions in the same way as you insert a non-CRAFTLINE instruction (you can insert multiple non-CRAFTLINE instructions between two CRAFTLINE instructions)



4. When inserting the motion class instruction, select the local position variable/global position variable (for local position variable, select "New" to create new points; for global position variable, select"GP" for the parameter, and then select the corresponding position point, for example, GP0001, GP0002....., the selected point should correspond to the program name and point in the XML file)

MOVL			
'arameter nam	Parameter source		Notes
Point	New	More	Saved points:0
V	10	More	Range (1-1000)
PL	0	More	Range (0-5)

MOVL			
arameter nam	Parameter source		Notes
Point	GP0001	More	Saved points:0
V	10	More	Range (1-1000)
PL	0	More	Range (0-5)

Writing XML files

Red parts are modifiable;

You need to create and edit the corresponding XML file in Notepad++ software;

When writing, <operation> and </operation> count as one CRAFTLINE;

1. For example: <?xml version="1.0" encoding="utf-8"?>

<WramCup RelationJobName="W1" ProductName="保 1" ProcedureName="缩 1">

The name of the program written under the general process: "W1"; Product name: "保 1"; Procedure: "缩 1"

2. <operation>

<Context note="关节" name="到某一点去"/>

<Position note="关节直角" name="GP0001"/>

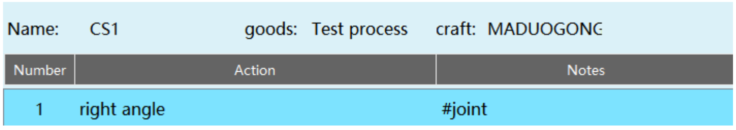
<Value note="变量 1" name="GP0001"/>

</operation>

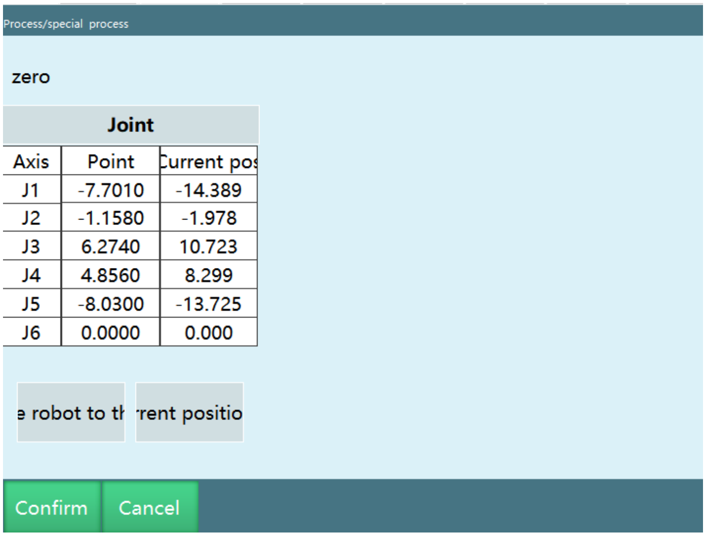
Action: "到某一点去"; Note: "直角"

Global position: "GP0001"; this means the robot will move to point GP0001

Motion class instruction corresponds to this type of code

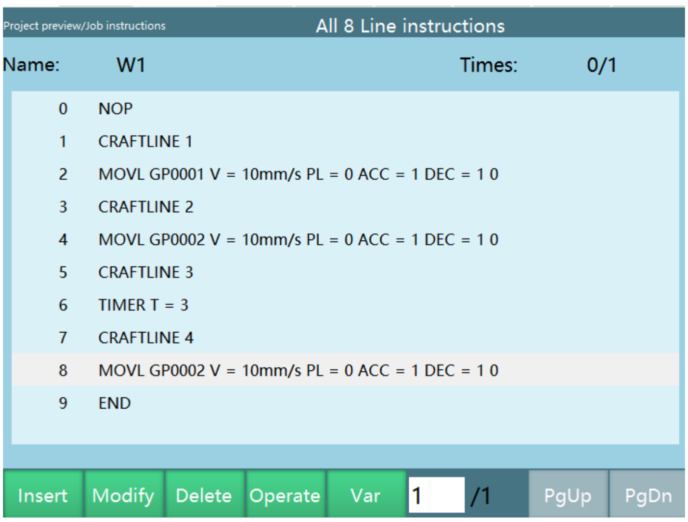


Select the instruction corresponding to this type of code and click "Modify", the modification interface is as follows



Select the instruction corresponding to this type of code and click "Modify", the interface does not change, this is because there are no modifiable items for this instruction.

Examples



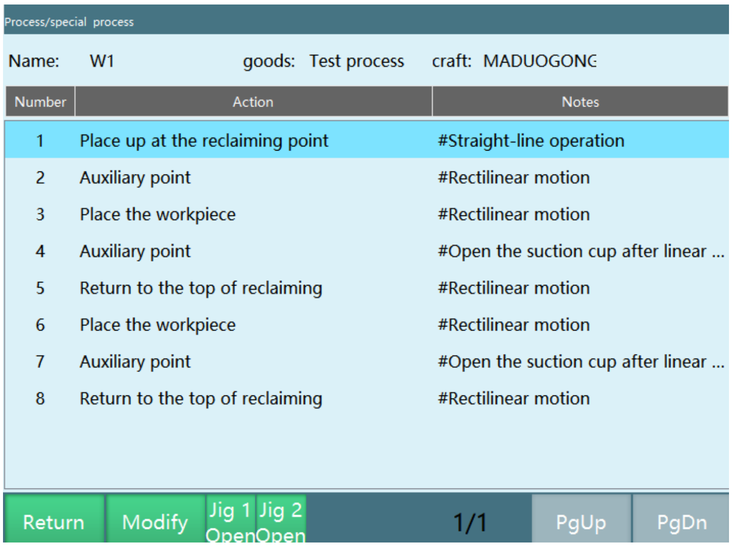
Write the program under the general process with the program name W1;

Then click "Process/Special process/Import", and import the corresponding XML file, the file content is as below:


```
<?xml version="1.0" encoding="utf-8"?>
<WramCup RelationJobName="W1" ProductName="保 1" ProcedureName="缩 1">
  <operation>
        <Context note="直线运行" name="取料点上方"/>
    <Position note="关节直角" name="P0001"/>
  </operation>
    <operation>
        <Context note="直线运动" name="辅助点"/>
    <Position note="关节直角" name="P0002"/>
  </operation>
  <operation>
        <Context note="直线运动" name="放工件点"/>
    <Position note="关节直角" name="P0003"/>
  </operation>
   <operation>
        <Context note="直线运动到位后打开吸盘" name="辅助点"/>
    <Position note="关节直角" name="P0004"/>
  </operation>
   <operation>
        <Context note="直线运动" name="返回取料上方"/>
    <Position note="关节直角" name="P0003"/>
  </operation>
   <operation>
        <Context note="直线运动" name="辅助点"/>
    <Position note="关节直角" name="P0002"/>
```

```xml
    </operation>
    <operation>
            <Context note="直线运动" name="放工件点"/>
      <Position note="关节直角" name="P0003"/>
    </operation>
     <operation>
            <Context note="直线运动到位后打开吸盘" name="辅助点"/>
      <Position note="关节直角" name="P0004"/>
    </operation>
     <operation>
            <Context note="直线运动" name="返回取料上方"/>
      <Position note="关节直角" name="P0003"/>
    </operation>
</WramCup>
```

After importing the XML file, switch the teach pendant from general process mode to special process mode, then click "Project", select the "W1" file, switch to run mode, and click "Start" to run



Process/special process

Name: W1 goods: Test process craft: MADUOGONG

Number	Action	Notes
1	Place up at the reclaiming point	#Straight-line operation
2	Auxiliary point	#Rectilinear motion
3	Place the workpiece	#Rectilinear motion
4	Auxiliary point	#Open the suction cup after linear ...
5	Return to the top of reclaiming	#Rectilinear motion
6	Place the workpiece	#Rectilinear motion
7	Auxiliary point	#Open the suction cup after linear ...
8	Return to the top of reclaiming	#Rectilinear motion

Return Modify Jig 1 Open Jig 2 Open 1/1 PgUp PgDn

The first line: displayed as serial number 1 in the special process program interface, as shown above

Corresponds to the first CRAFTLINE in the general process mode: run instruction 1,2, robot moves to point GP0001

The second line: displayed as serial number 2

Corresponds to the second CRAFTLINE in the general process mode: run instruction 3,4, robot moves to point GP0002

The third line: displayed as serial number 3

Corresponds to the third CRAFTLINE in the general process mode: run instruction 5,6, robot output is delayed by 3s

The fourth line: displayed as serial number 4

Corresponds to the fourth CRAFTLINE in the general process mode: run instruction 7, 8, robot moves to GP0002

Notes

- Programs can not be executed in special process mode, for example, you cannot insert LABEL and JUMP instructions; CRAFTLINE 1 cannot jump to CRAFTLINE 3

Robot Operation Manual

4-Axis Parallel

<<<

Catalogue

4-Axis Parallel Robot Operation Manual

Parallel Robot Introduction

A parallel robot, named Parallel Mechanism, or PM for short, can be defined as a closed-loop mechanism in which the moving and fixed platforms are connected by at least two independent kinematic chains, and it has two or more degrees of freedom and is driven in parallel.

Definition:

A closed-loop mechanism in which the moving and fixed platforms are connected by at least two independent kinematic chains, has two or more degrees of freedom and is driven in parallel.

Features:

1. No cumulative error, high accuracy;

2. The driving device can be placed on or close to the fixed platform, so that the moving part is light in weight, high in speed and good in dynamic response;

3. Compact structure, high stiffness and high load capacity;

4. A completely symmetrical parallel mechanism has better isotropy;

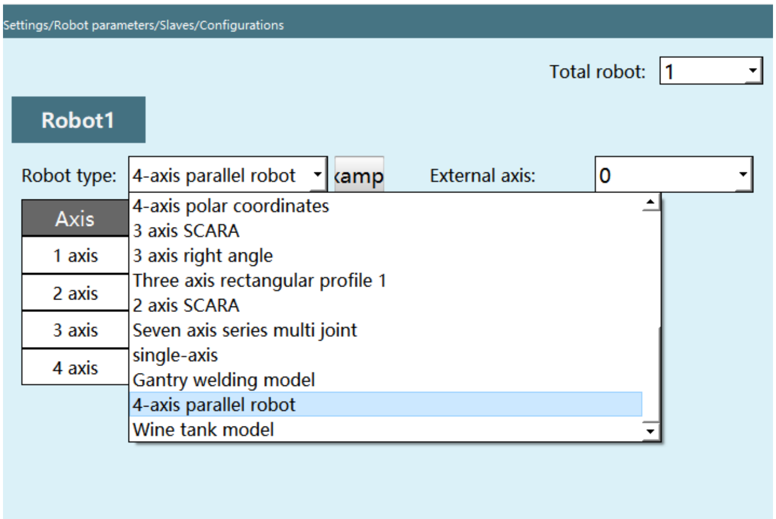
5. Small working space;

Based on these features, parallel robots are widely used in fields that require high stiffness, high accuracy or large loads without a large working space.

Parallel Robot Basic Operation

Slave configuration

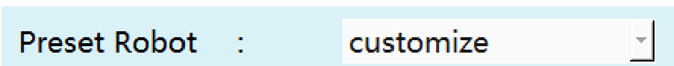
If you need to select 4-axis parallel robot, click [Settings-Robot parameters-Slave configuration-Robot] and select "4-axis parallel robot" in the "Robot type" drop-down menu, then click "Save".



Parameters presetting

When you click "Save" after selecting "4-axis parallel robot", you need to import the robot parameter profile, but in the "DH parameters" interface, we provide the function of presetting robot parameters. If this drop-down list contains the robot model you are using, you can set up all the robot parameters quickly and easily with this function. There is no need to import the controller configuration parameters separately.

1. Click [Preset robot] in the upper left corner of the "DH parameters" interface, you can select the robot model that has already been adapted, and the DH parameters and joint parameters of the robot will be filled in automatically after the selection.

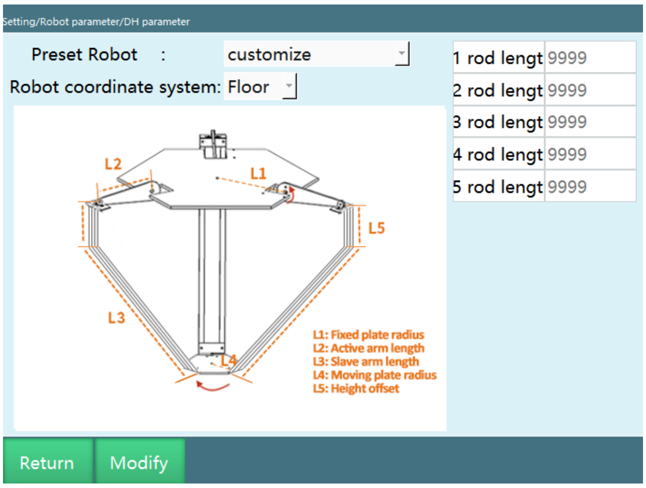


2.You need to calibrate the zero point manually after selecting the preset robot.

DH parameters setting

1. Fill in the rod length parameter of the robot; this parameter affects the linear motion and accuracy of the robot.

Note: Please do not power on and operate the robot until the DH parameters, joint parameters and zero point are set.



4-axis parallel robot

Parameter description

Preset robot

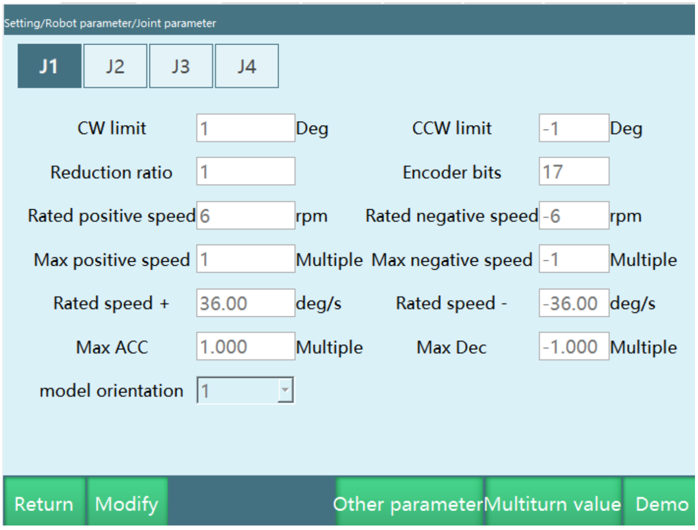
By importing the robot joint parameters and DH parameters into the controller in advance, you can eliminate the need to fill in the parameters repeatedly

Rod length

The rod length parameter should be filled in as shown in the model diagram on the "DH parameters" interface, if no value is given, we can only measure the length of each axis of the robot by using a ruler, and inaccurate filling will affect the robot motion accuracy

Joint parameters

Note: Please do not power on and jog the robot before DH parameters and joint parameters are set to prevent the robot from getting out of control and causing danger to the operator. If you need the robot to return to the zero position, click [Robot parameters - Zero position] to see if it is at the zero position, if not, please calibrate the zero point first.



Meaning of each parameter

CW limit

The maximum range of the robot joints in the positive direction. After importing the controller configuration, the values of each parameter in the joint parameter interface will be entered automatically and the values of the limits can be modified

CCW limit

The maximum position of the robot in the reverse direction during single-axis rotation (This value must be negative)

Reduction ratio

The ratio of instantaneous input speed to output speed in the reduction mechanism

Encoder bits

The number of bits of the encoder. Usually 17 or 23 bits

Rated positive RPM

The rated rotation speed of the motor in the positive direction

Rated reverse RPM

The rated rotation speed of the motor in the reverse direction (This value must be negative)

Maximum positive RPM

The maximum rotation speed of the motor in the positive direction; its value is a multiple of the rated positive RPM. If the rated positive RPM is 3000 rpm and the maximum positive RPM needs to be 6000 rpm, then fill in 2 times here.

Maximum reverse RPM

The maximum rotation speed of the motor in the reverse direction; its value is a multiple of the rated reverse RPM. If the rated reverse RPM is -4000 rpm and the maximum reverse RPM needs to be -6000 rpm, then fill in -1.5 times here. (This value must be negative)

Rated positive speed

The rated positive speed of the robot joint; it is automatically calculated from the rated positive RPM, encoder bits and the reduction ratio (the axis 3 of the 4-axis SCARA and axis 1 of the 4-axis SCARA special-shaped robot also need to add the pitch), no need to fill in.

Rated reverse speed

The rated reverse speed of the robot joint; it is automatically calculated from the rated reverse RPM, encoder bits and the reduction ratio, no need to fill in. (This value must be negative)

Maximum acceleration

The maximum acceleration of the robot joint movement; its value is a multiple of the rated positive (reverse) speed. If the rated positive speed is 300 degrees/s, the maximum acceleration needs to be 1500 degrees/s^2 , then fill in 5 times here.

Maximum deceleration

The maximum deceleration of the robot joint movement; its value is a multiple of the rated positive (reverse) speed. If the rated positive speed is 300 degrees/s, the maximum acceleration is 1200 degrees/s^2 , fill in -4 times here. It is recommended that the maximum acceleration and maximum deceleration values be the same. (This value must be negative)

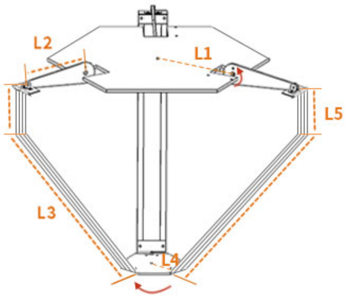
Model direction

The model direction should be set by referring to the joint positive direction diagram below, and the direction of the jogging "+" key of each axis should be the same as the joint positive direction diagram (choosing 1 for the same and -1 for the opposite)

Gear backlash

The angle to compensate for the filled value whenever the joint moves in the opposite direction; not filled by default.

Joint positive direction diagram

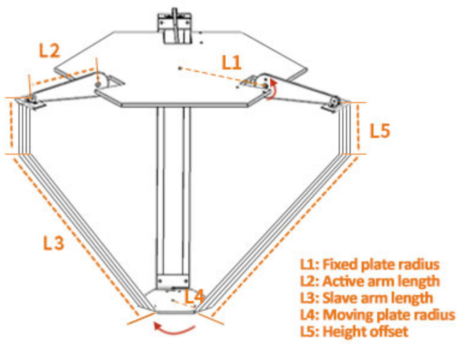


The direction shown in the figure is the positive direction of the robot joints

Note: Please do not power on and operate the robot until the positive direction of the joint has been set.

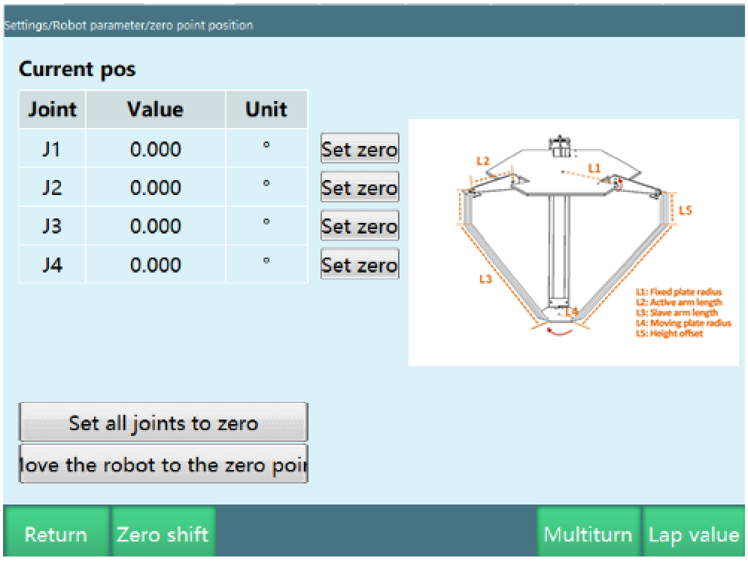
Zero calibration

If the robot zero position is a non-standard zero position, users can align the robot according to the robot's alignment hole, and then set the current robot position coordinates to the zero position on the robot zero position interface. The diagram of the zero position of the 4-axis parallel robot is as follows.



Parallel robot zero point commissioning: Align the driving shaft horizontally with the upper plate of the robot to perform zero calibration.

Make sure the robot is in this position and click "Set all joints to zero"



Note: Please do not power on and jog the robot before DH parameters and joint parameters are set to prevent the robot from getting out of control and causing danger to the operator.

If you need the robot to return to the zero position, click [Robot parameters - Zero position] to see if it is at the zero position, if not, please calibrate the zero point first.

Cautions

If the robot is not zero position calibrated, you can not return it to the zero point and jog it.

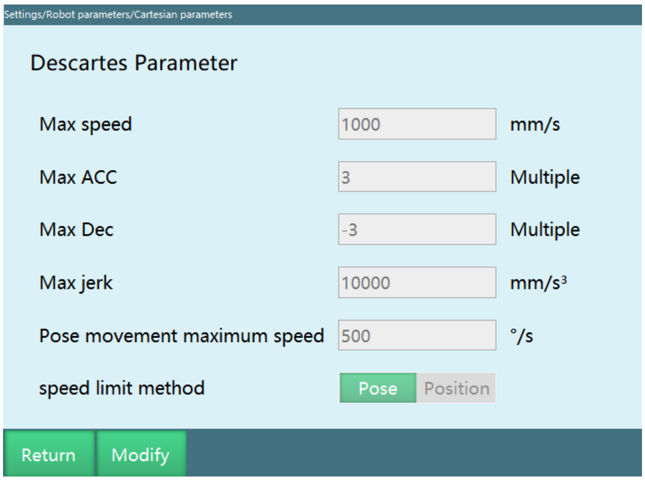
For systems that use multiple robots, each robot must perform origin position calibration.

When there is a coupling relationship between joint axes, such as the common coupling relationship between the fifth axis and the sixth axis of a robot, the fifth axis must be at the zero position, then the zero data recorded for the sixth axis will be valid, otherwise, the zero data recorded for the sixth axis will be invalid. So the zero data of the sixth axis must be recorded with the fifth axis at the zero position. If there is no coupling relationship, each axis can calibrate the zero position individually, and the respective zero position will not affect the zero position of other joints.

When all the used axes (body axes and auxiliary extension axes) have been zero-calibrated, the "All" indicator on the "Zero calibration" interface turns green, indicating that the robot has completed zero calibration, and the robot is ready to move in Cartesian space.

Cartesian parameters setting

Cartesian parameters can use default values directly.



Meaning of each parameter

Maximum speed

The maximum linear speed of the robot during operation.

Maximum acceleration

The maximum acceleration of the robot during operation; its value is a multiple of the maximum speed. If the maximum speed is 1000mm/s and the maximum acceleration needs to be 3000mm/s^2 , then fill in 3 times here.

Maximum deceleration

The maximum deceleration of the robot during operation; its value is a multiple of the maximum speed. If the maximum speed is 1000 mm/s and the maximum deceleration needs to be -3000 mm/s^2 , then fill in -3 times here. It is recommended that the maximum acceleration and maximum deceleration values be the same and the same as the maximum acceleration and maximum deceleration in the joint parameters. (This value must be negative)

Maximum jerk

This parameter is a reserved parameter and is currently invalid.

Pose movement maximum speed

The maximum speed of the robot during operation, if the instruction speed exceeds this value, it will be decelerated.

Speed limit method

Pose: The linear interpolation motion of the robot is limited by the maximum speed and the pose movement maximum speed.

Position: The linear interpolation motion of the robot is limited only by the maximum speed.

Application of Parallel Robot in Process

The parallel robot has great advantages in some processes as it can be seen from its advantages, such as: palletizing process, conveyor tracking process, vision process, searching and tracking, etc. (You can perform process-related tests of it through the manual)

iNexBot

Robot Operation Manual

4-Axis SCARA

Catalogue

4-Axis SCARA Robot Operation Manual

SCARA robot introduction

The SCARA robot has four joints, three of which are rotary joints with axes parallel to each other for positioning and orientation in the plane. The other joint is a movable joint, which is used to complete the movement of end pieces in perpendicular to the plane.

SCARA system has compliance in X-axis and Y-axis directions and good stiffness in Z-axis direction, this feature is especially suitable for assembly work.

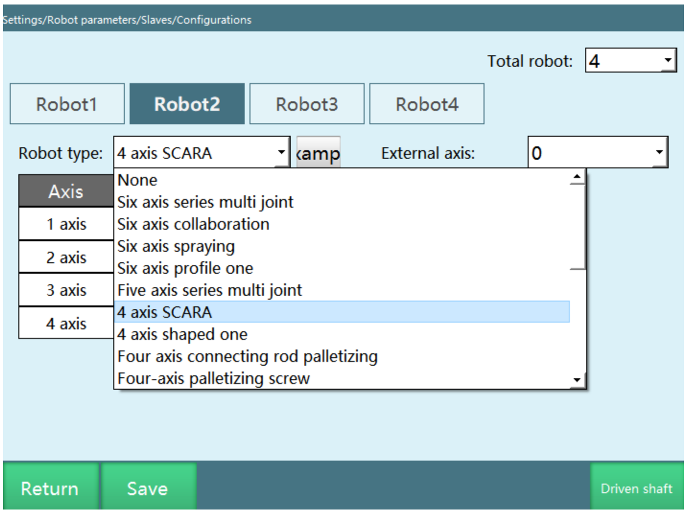
SCARA robot is also widely used in plastic industry, automotive industry, electronic product industry, pharmaceutical industry and food industry and other areas. Its main functions are parts handing and assembly work.

Its first and second axes have rotation characteristics, while the third and fourth axes can be manufactured in a variety of different forms depending on the needs of the job, with one having rotation and the other having linear movement characteristics. Due to its specific shape, its working area is similar to a sector.

4-axis SCARA robot

Slave configuration

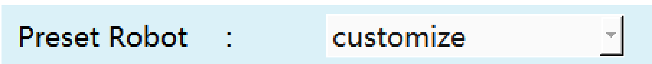
If you need to select 4-axis SCARA robot, click [Settings-Robot parameters-Slave configuration-Robot] and select "4-axis SCARA" in the "Robot type" drop-down menu, then click "Save".



Parameters presetting

When you click "Save" after selecting "4-axis SCARA", you need to import the robot parameter profile, but in the "DH parameters" interface, we provide the function of presetting robot parameters. If this drop-down list contains the robot model you are using, you can set up all the robot parameters quickly and easily with this function. There is no need to import the controller configuration parameters separately.

Click [Preset robot] in the upper left corner of the "DH parameters" interface, you can select the robot model that has already been adapted, and the DH parameters and joint parameters of the robot will be filled in automatically after the selection.



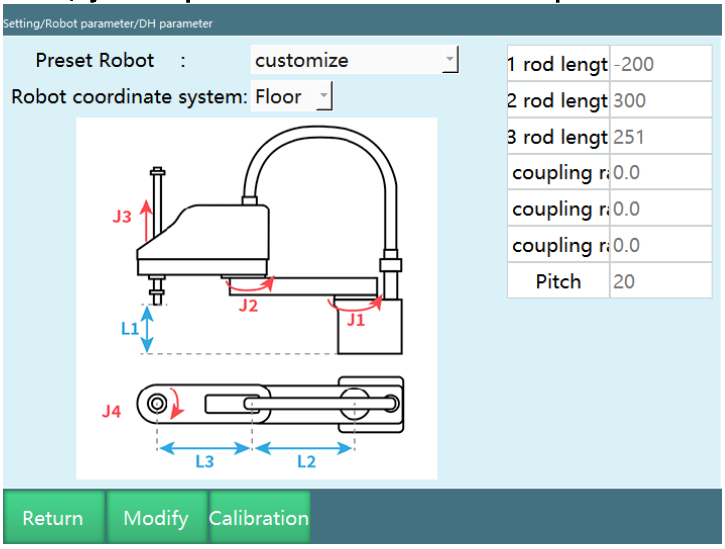
You need to calibrate the zero point manually after selecting the preset robot.

DH parameters setting

1. Fill in the parameters of the robot, such as rod length, coupling ratio and pitch; these parameters will affect the linear motion and accuracy of the robot.
2. If the robot accuracy is poor, you can return to this interface for 4-point calibration after the configuration is completed, and calibrate the rod length parameters. The following figure shows

the values of each parameter in the robot "DH parameters" interface after importing the controller configuration.

Note: Please do not power on and operate the robot until the DH parameters, joint parameters and zero point are set.



4-axis SCARA

Parameter description

Preset robot

By importing the robot joint parameters and DH parameters into the controller in advance, you can eliminate the need to fill in the parameters repeatedly

Rod length

The rod length parameter should be filled in as shown in the model diagram on the "DH parameters" interface, if no value is given, we can only measure the length of each axis of the robot by using a ruler, and inaccurate filling will affect the robot motion accuracy

Coupling ratio

How to tell if the robot is coupled?
We can run the robot, when jogging the axis 1, the axis 2 also moves, indicating that there is coupling.
How to calculate the coupling ratio?

For example, when axis A rotates by a degrees, it causes axis B to rotate by b degrees, so the coupling ratio of the two axes is

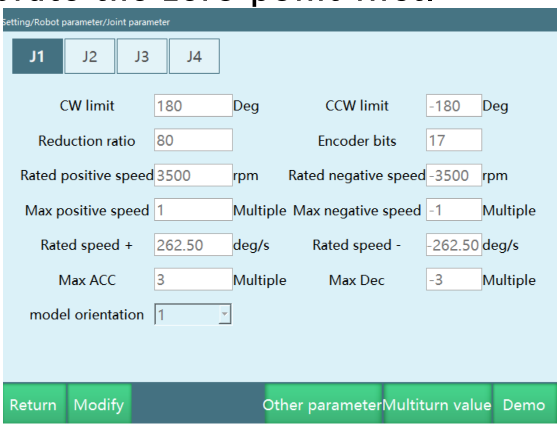
c=b÷a

Pitch

Pitch of the link responsible for up and down movement in 4-axis SCARA (axis 3 of 4-axis SCARA and axis 1 of 4-axis SCARA special-shaped robot)

Joint parameters

Note: Please do not power on and jog the robot before DH parameters and joint parameters are set to prevent the robot from getting out of control and causing danger to the operator. If you need the robot to return to the zero position, click [Robot parameters - Zero position] to see if it is at the zero position, if not, please calibrate the zero point first.



Meaning of each parameter

CW limit

The maximum range of the robot joints in the positive direction. After importing the controller configuration, the values of each parameter in the joint parameter interface will be entered automatically and the values of the limits can be modified

CCW limit

The maximum position of the robot in the reverse direction during single-axis rotation (This value must be negative)

Reduction ratio

The ratio of instantaneous input speed to output speed in the reduction mechanism

Encoder bits

The number of bits of the encoder. Usually 17 or 23 bits

Rated positive RPM

The rated rotation speed of the motor in the positive direction

Rated reverse RPM

The rated rotation speed of the motor in the reverse direction (This value must be negative)

Maximum positive RPM

The maximum rotation speed of the motor in the positive direction; its value is a multiple of the rated positive RPM. If the rated positive RPM is 3000 rpm and the maximum positive RPM needs to be 6000 rpm, then fill in 2 times here.

Maximum reverse RPM

The maximum rotation speed of the motor in the reverse direction; its value is a multiple of the rated reverse RPM. If the rated reverse RPM is -4000 rpm and the maximum reverse RPM needs to be -6000 rpm, then fill in -1.5 times here. (This value must be negative)

Rated positive speed

The rated positive speed of the robot joint; it is automatically calculated from the rated positive RPM, encoder bits and the reduction ratio (the axis 3 of the 4-axis SCARA and axis 1 of the 4-axis SCARA special-shaped robot also need to add the pitch), no need to fill in.

Rated reverse speed

The rated reverse speed of the robot joint; it is automatically calculated from the rated reverse RPM, encoder bits and the reduction ratio, no need to fill in. (This value must be negative)

Maximum acceleration

The maximum acceleration of the robot joint movement; its value is a multiple of the rated positive (reverse) speed. If the rated positive speed is 300 degrees/s, the maximum acceleration needs to be 1500 degrees/s^2, then fill in 5 times here.

Maximum deceleration

The maximum deceleration of the robot joint movement; its value is a multiple of the rated positive (reverse) speed. If the rated positive speed is 300 degrees/s, the maximum acceleration needs to be 1200 degrees/s^2, then fill in -4 times here. It is recommended that the maximum acceleration and maximum deceleration values be the same. (This value must be negative)

Model direction

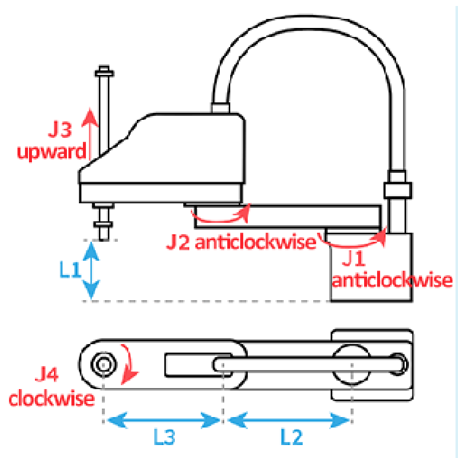
The model direction should be set by referring to the joint positive direction diagram below, and the direction of the jogging "+" key of each axis should be the same as the joint positive direction diagram (choosing 1 for the same and -1 for the opposite).

Gear backlash

The angle to compensate for the filled value whenever the joint moves in the opposite direction; not filled by default.

Robot type	Axis	Positive direction (top view or left view)
4-axis SCARA	J1	anticlockwise
	J2	anticlockwise
	J3	upward
	J4	clockwise

Joint positive direction diagram

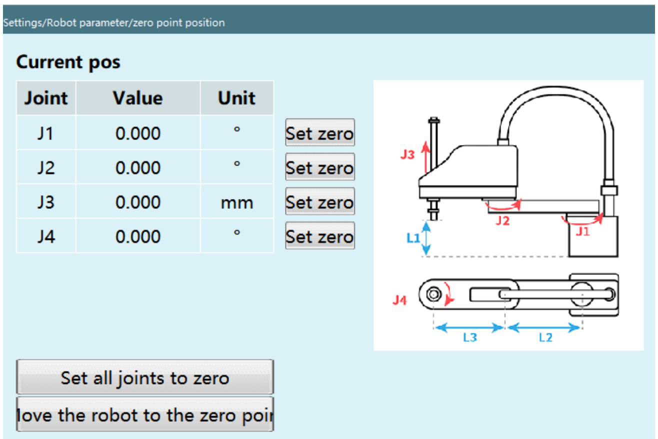


Note: Please do not power on and operate the robot until the positive direction of the joint has been set.

Zero calibration

If the robot zero position is a non-standard zero position, users can align the robot according to the robot's alignment hole, and then set the current robot position coordinates to the zero position on the robot zero position interface. The diagram of the SCARA zero position is as follows.

Make sure the robot is in this position and click "Set all joints to zero"

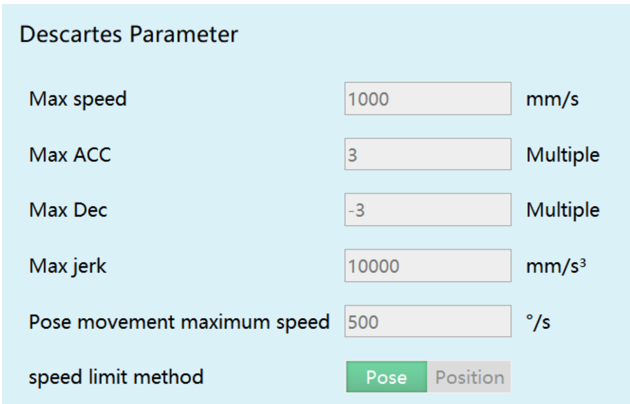


Note: Please do not power on and jog the robot before DH parameters and joint parameters are set to prevent the robot from getting out of control and causing danger to the operator. If you need the robot to return to the zero position, click [Robot parameters - Zero position] to see if it is at the zero position, if not, please calibrate the zero point first.

Cautions

If the robot is not zero position calibrated, you can not return it to the zero point and jog it.
For systems that use multiple robots, each robot must perform origin position calibration.
When there is a coupling relationship between joint axes, such as the common coupling relationship between the fifth axis and the sixth axis of a robot, the fifth axis must be at the zero position, then the zero data recorded for the sixth axis will be valid, otherwise, the zero data recorded for the sixth axis will be invalid. So the zero data of the sixth axis must be recorded with the fifth axis at the zero position. If there is no coupling relationship, each axis can calibrate the zero position individually, and the respective zero position will not affect the zero position of other joints.
When all the used axes (body axes and auxiliary extension axes) have been zero-calibrated, the "All" indicator on the "Zero calibration" interface turns green, indicating that the robot has completed zero calibration, and the robot is ready to move in Cartesian space.

Cartesian parameters setting
Cartesian parameters can use default values directly.



Meaning of each parameter

Maximum speed

The maximum linear speed of the robot during operation.

Maximum acceleration

The maximum acceleration of the robot during operation; its value is a multiple of the maximum speed. If the maximum speed is 1000mm/s and the maximum acceleration needs to be 3000mm/s², then fill in 3 times here.

Maximum deceleration

The maximum deceleration of the robot during operation; its value is a multiple of the maximum speed. If the maximum speed is 1000 mm/s and the maximum deceleration needs to be -3000 mm/s², then fill in -3 times here. It is recommended that the maximum acceleration and maximum deceleration values be the same and the same as the maximum acceleration and maximum deceleration in the joint parameters. (This value must be negative)

Maximum jerk

This parameter is a reserved parameter and is currently invalid.

Pose movement maximum speed

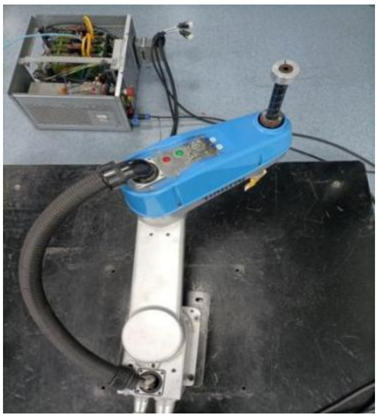
The maximum speed of the robot during operation, if the instruction speed exceeds this value, it will be decelerated.

Speed limit method

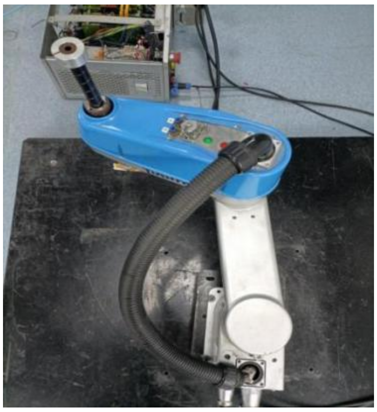
Pose: The linear interpolation motion of the robot is limited by the maximum speed and the pose movement maximum speed.
Position: The linear interpolation motion of the robot is limited only by the maximum speed.

4-axis SCARA robot left and right hand

Left and right hand (4-axis SCARA robots only)



left hand attitude



right hand attitude

For example:

1.Jog the robot to adjust it to position of the left hand attitude

2.Insert the movl instruction, modify the current position to the P001 point, and select "Left hand" and save

3.Jog the robot to adjust it to the position of the right hand attitude

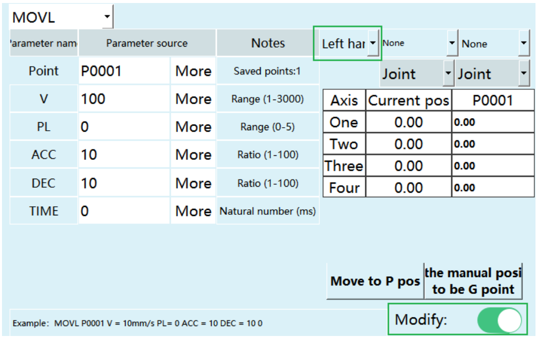
4.Step movl instruction

The running result will report an error: The left and right hand system of robot 1 is incorrectly used

Note: movj instruction does not distinguish between left and right hand

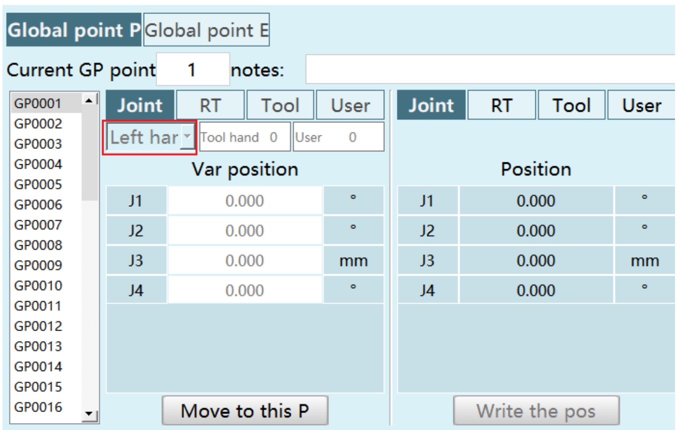
In summary,the right hand attitude cannot follow the position of the left hand

The left and right hand is generally used to compress the robot's movement space, and can also be used for obstacle avoidance. Generally, we only choose the cartesian coordinate system to set the left and right hand, and the decision procedure is based on the direction of axis 2. You can select the left and right hand in the instruction setting interface, and when the setting is completed, click the [Manual modify] button and then click the "OK" button, as shown in the figure

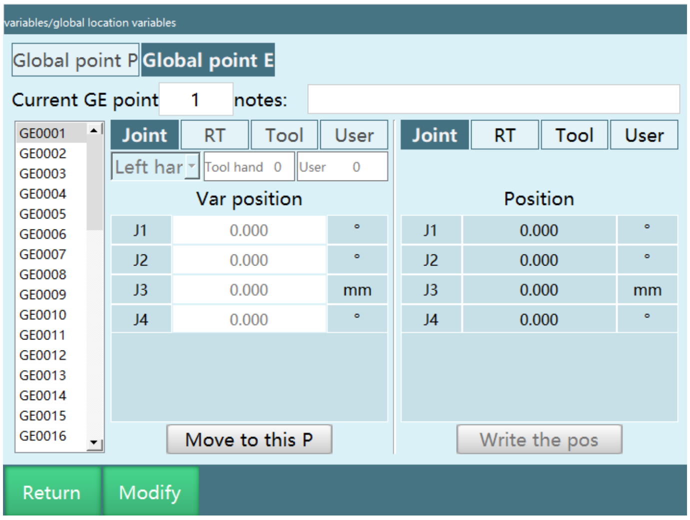


Global variables settings for left and right hand

Click [Variables] - [Global variables], then click the drop-down menu, you can set the left and right hand of the robot global point GP, the picture below shows the global variable interface to set the left hand.

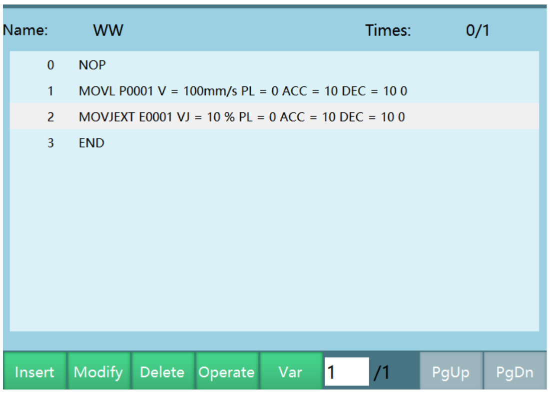


Set the left and right hand of GE points of 4-axis SCARA robots with external axes in the same way

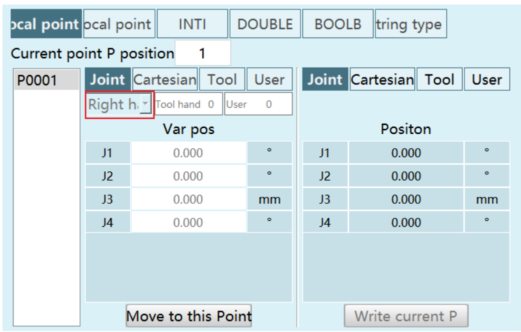


Local variables settings for left and right hand

Click [Program], select a program to open, and select [Variables] - [Local variables] at the bottom, as shown in the following figure



Click "Modify", click on the top drop-down arrow and select left or right hand, as shown in the figure

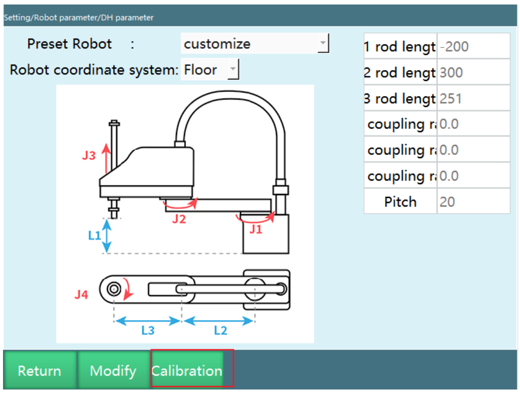


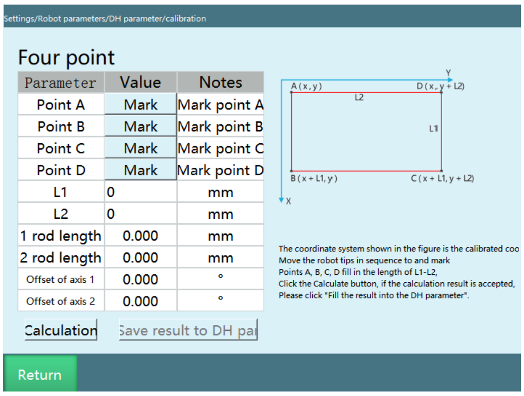
In the instruction parameter setting interface, you can select parameters to set the local position variable P point or E point

4-point calibration

4-point calibration can be used to correct rod length and zero point

Click the "Calibrate" button on the "DH parameters" interface to enter the 4-point calibration interface





Point A, point B, point C and point D form a rectangle; the values of L1 and L2 represent the width and length of the rectangle

The above figure shows the calibration of four points A,B,C,D.

For example: L1=50

L2=100

A (X,Y), B (X+50,Y), C (X + 50, Y+100), D (X, Y+100)

Click "Calculate" after calibrating the 4 points

Confirm the calculation result is correct and click "Fill the result into DH parameters"

2-point calibration

2-point calibration supports 4-axis SCARA.

Click the [2-point calibration] button at the bottom of the "Tool hand calibration" interface to enter the "2-point calibration" interface, as shown in the figure.

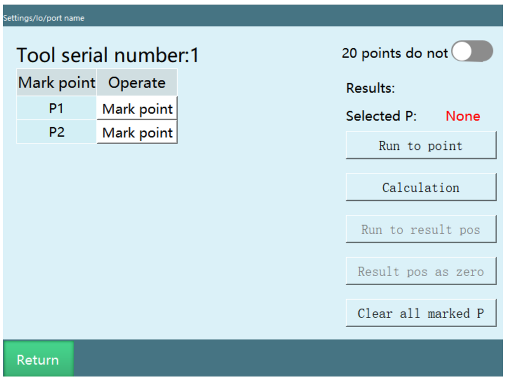


Figure. 2-point calibration

The specific calibration steps are as follows:

1.Find a reference point (pen tip) and make sure this reference point is fixed.

2.Calibrate a point under the left hand form (calibrated point and reference point tip to tip)

3.Calibrate a point under the right hand form (calibrated point and reference point tip to tip)

4.After calibrating 2 points, click [Calculate].

5.If you are not satisfied with the calibration of a point during the calibration process, you can click the [Cancel calibration] button corresponding to that line to cancel the calibration and then calibrate the point again.

6.You can click [Run to this point] after each point is calibrated, and the robot will run to that point.

7.Move the robot to another position, and then click [Run to calculation result position], then the robot will move to the original calibration position, which is equivalent to the zero position of the robot.

8.Mark the result position as zero point: After the 2-point calibration is finished, click [Calculate] and then click [Save results], a pop-up window will appear, displaying "Confirm whether to set the result to zero point"
9.[Clear all mark points]: The calibration points will be saved in the controller, and the calibration results will be cleared only after you click "Cancel calibration", "Clear all calibration points" and switch the tool hand into the calibration interface.

Cautions
Please try to take the pose of each point in any direction. If the pose taken is rotated in a certain direction, sometimes the accuracy is not accurate. Please keep the reference point fixed during the calibration process, otherwise the calibration error will increase.

Click the [Return] button at the bottom to return to the "Tool hand calibration" interface.

iNexBot

Robot Operation Manual

6-Axis Collaborative

<<<

Catalogue

6-Axis Collaborative Robot

Introduction

Collaborative robots, as the name implies, emphasize the concept of "collaboration" and focus on the collaboration between humans and robots. For this reason, usability, safety, and intelligence are particularly important.

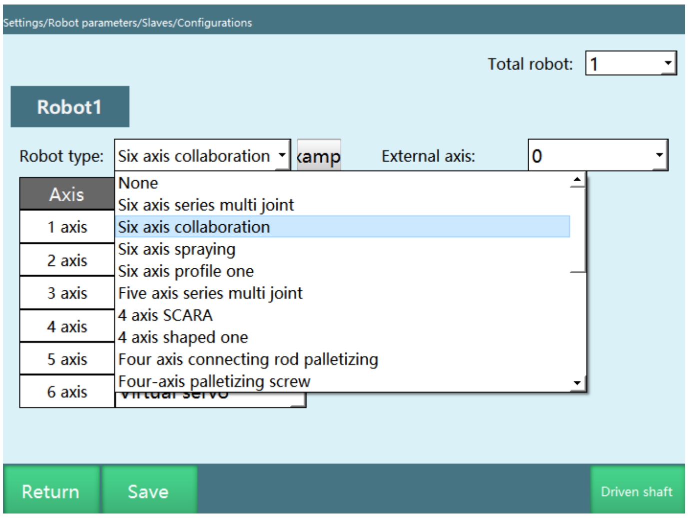
The original intention of collaborative robots is to realize human-robot collaboration, and achieve human-robot integration within a certain range without installing fences. Therefore, the emergence of collaborative robots has changed the production relationship and broken down the barriers between humans and robots.

First of all, from a business perspective, collaborative robots are the fastest growing category in the industrial robots market and have become a market pursuit with better performance than 6-axis robots and other traditional industrial robots. In addition, collaborative robots are more competitive than traditional industrial robots.

Besides, collaborative robots are humanoid robots whose purpose is to replace human hands. We can see that collaborative robots can achieve not only simple actions such as picking up cups, picking and placing materials, pressing buttons, but also pulling flowers and achieving a high degree of consistency in unattended retail areas such as coffee shops and milk tea stations. The existing collaborative robot product matrix is based on human design logic, including extreme motion and dynamics design, modular hardware structure design, flexible and reliable adaptation software, and multilingual combinations.

Slave configuration setting

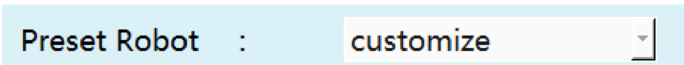
Enter the "Settings/Robot parameters/Slave configuration" interface, and select the "6-axis collaborative robot" from the "Robot type".



Parameters presetting

In the "DH parameters" interface, we provide the "Preset robot" function. If this drop-down list contains the robot model you are using, you can use this function to quickly and easily set up the parameters of your robot.

1.Click [Preset robot] in the upper left corner of the "DH parameters" interface, you can select the robot model that has already been adapted, and the DH parameters and joint parameters of the robot will be filled in automatically after the selection.

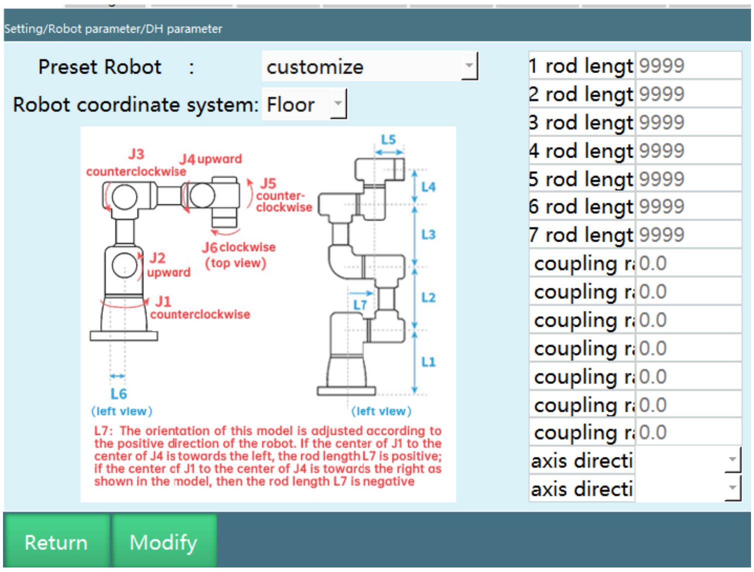


2.You need to modify the zero point manually after selecting the preset robot.

DH parameters setting

Enter the "Settings/Robot parameters/DH parameters" interface, and fill in the parameters of the robot, such as rod length, coupling ratio, axis 3 direction and axis 5 direction; these parameters will affect the linear motion and accuracy of the robot;

Note: After the DH parameters, joint parameters, etc. are set, power on and operate the robot to confirm whether the model orientation is correct.



6-axis collaborative robot

Parameter description

Preset robot

By importing the robot joint parameters and DH parameters into the controller in advance, you can eliminate the need to fill in the parameters repeatedly

Robot coordinate system

floor mounting

ceiling mounting

Note: The ceiling-mounted robots do not support recognition as well as collision detection.

Rod length

The control system needs to accurately model the robot in order to calculate the current coordinates of the end of the robot and the angle of rotation required for each joint axis as the robot moves from point A to point B.

Modeling the robot requires specifying the lengths of each part of the robot. These lengths are the rod length parameters, also known as DH parameters.

The rod length parameter should be filled in as shown in the model diagram on the "DH parameters" interface, inaccurate filling will affect the robot motion accuracy.

Coupling ratio

Some robot bodies are designed so that the motor spans many axes to drive a particular axis, which creates a coupling between the two axes. For example, if we rotate axis 2, axis 3 follows, which is axis coupling. To counteract this coupling effect, a coupling ratio is needed.

The calculation formula for the coupling ratio is:

$$\text{coupling ratio}=\frac{following\ axis\ rotation\ angle}{main\ axis\ rotation\ angle}$$

For example, if we rotate axis 2 by 10º and find that axis 3 follows the rotation by 15º, then the coupling ratio is

$$\frac{15}{10}=1.5$$

For detailed calculation method of coupling ratio, please refer to "NRC Debugging Manual"

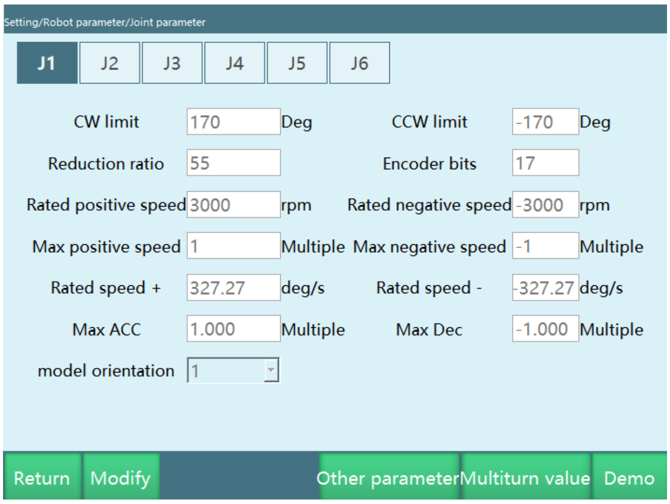
Axis 3 direction/Axis 5 direction

The axis 3 direction and axis 5 direction in 6-axis collaborative robot correspond to the two forms of collaborative robots.

> Joint parameters setting

The setup procedure is the same as "Robot Parameter Debugging"

Note: Please do not power on and operate the robot until the DH parameters and joint parameters are set.



Meaning of each parameter

CW limit

The maximum range of the robot joints in the positive direction.

CCW limit

The maximum range of the robot joints in the reverse direction (This value must be negative)

Reduction ratio

The reduction ratio of the reducer.

Encoder bits

The number of bits of the encoder.

Rated positive RPM

The rated rotation speed of the motor in the positive direction.

Rated reverse RPM

The rated rotation speed of the motor in the reverse direction (This value must be negative)

Maximum positive RPM

The maximum rotation speed of the motor in the positive direction; its value is a multiple of the rated positive RPM. If the rated positive RPM is 3000 rpm and the maximum positive RPM needs to be 6000 rpm, then fill in 2 times here.

Maximum reverse RPM

The maximum rotation speed of the motor in the reverse direction; its value is a multiple of the rated reverse RPM. If the rated reverse RPM is -4000 rpm and the maximum reverse RPM needs to be -6000 rpm, then fill in -1.5 times here. (This value must be negative)

Rated positive speed

The rated positive speed of the robot joint; it is automatically calculated from the rated positive RPM, encoder bits and the reduction ratio, no need to fill in.

Rated reverse speed

The rated reverse speed of the robot joint; it is automatically calculated from the rated reverse RPM, encoder bits and the reduction ratio, no need to fill in. (This value must be negative)

Maximum acceleration

The maximum acceleration of the robot joint movement; its value is a multiple of the rated positive (reverse) speed. If the rated positive speed is 300 degrees/s, the maximum acceleration needs to be 1500 degrees/s^2, then fill in 5 times here.

Maximum deceleration

The maximum deceleration of the robot joint movement; its value is a multiple of the rated positive (reverse) speed. If the rated positive speed is 300 degrees/s, the maximum acceleration needs to be 1200 degrees/s^2, then fill in -4 times here. It is recommended that the maximum acceleration and maximum deceleration values be the same. (This value must be negative)

Model direction

The model direction should be set by referring to the joint positive direction diagram below, and the direction of the jogging "+" key of each axis should be the same as the joint positive direction diagram (choosing 1 for the same and -1 for the opposite)

Actual joint direction

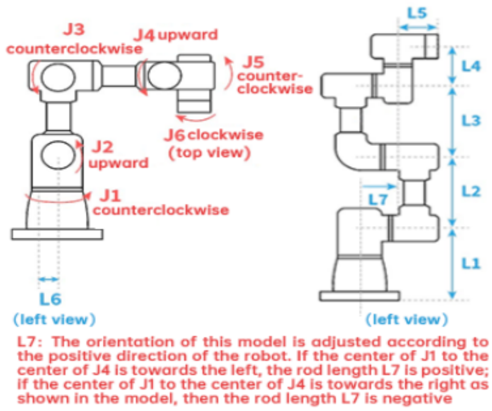
The default option is 1.

Gear backlash

The angle to compensate for the filled value whenever the joint moves in the opposite direction, not filled by default

Robot type	Axis	Positive direction (top view or left view)
6-axis collaborative robot	J1	anticlockwise
	J2	upward
	J3	anticlockwise
	J4	upward
	J5	anticlockwise
	J6	clockwise (top view)

Joint positive direction diagram

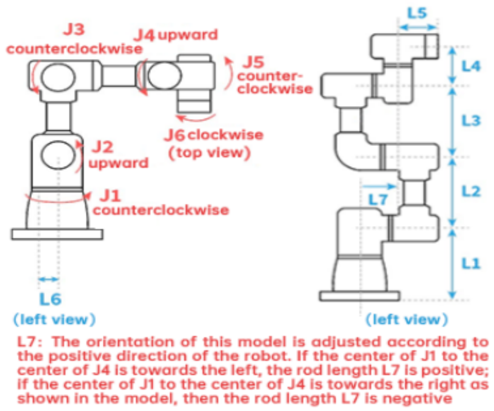


Note: Please do not power on and operate the robot until the positive direction of the joint has been set.

Zero calibration

If the robot zero position is a non-standard zero position, users can align the robot according to the robot's alignment hole, and then set the current robot position coordinates to the zero position on the robot zero position interface.

The schematic diagram of the zero point position of the 6-axis collaborative robot is as follows (here are the two forms of the zero point model respectively). The model direction of the left form is adjusted on the left side of the robot; the model direction of the right form is adjusted in the positive direction of the robot, if the center of J1 to the center of J4 is towards the left, the rod length L7 is positive; if the center of J1 to the center of J4 is towards the right as shown in the model, then the rod length L7 is negative:



Make sure the robot is in this position and click "Set all joints to zero".

Note: Please do not power on and operate the robot before DH parameters and joint parameters are set.

Notes

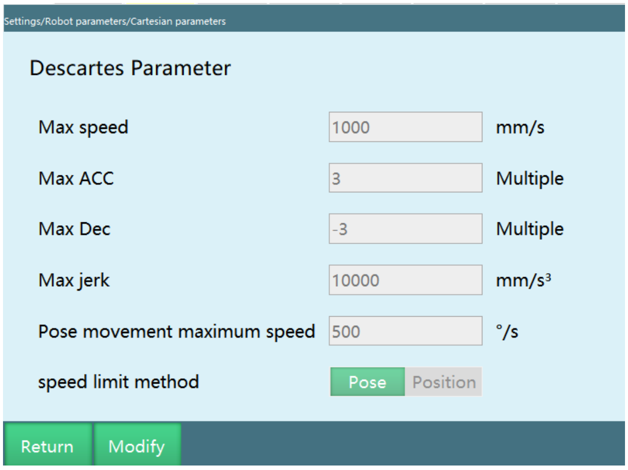
If the robot is not origin position calibrated, you can not return it to the zero point

For systems that use multiple robots, each robot must perform origin position calibration

When there is a coupling relationship between joint axes, such as the common coupling relationship between the fifth axis and the sixth axis of a robot, the fifth axis must be at the zero position, then the zero data recorded for the sixth axis will be valid, otherwise, the zero data recorded for the sixth axis will be invalid. So the zero data of the sixth axis must be recorded with the fifth axis at the zero position. If there is no coupling relationship, each axis can calibrate the zero position individually, and the respective zero position will not affect the zero position of other joints

Cartesian parameters setting

Cartesian parameters can use default values directly.



Meaning of each parameter

Maximum speed

The maximum linear speed of the robot during operation. (The inserted instructions that require V parameter are subject to Cartesian parameters)

Maximum acceleration

The maximum acceleration of the robot during operation; its value is a multiple of the maximum speed. If the maximum speed is 1000mm/s and the maximum acceleration needs to be 3000mm/s², then fill in 3 times here.

Maximum deceleration

The maximum deceleration of the robot during operation; its value is a multiple of the maximum speed. If the maximum speed is 1000 mm/s and the maximum deceleration needs to be -3000 mm/s², then fill in -3 times here. It is recommended that the maximum acceleration and maximum deceleration values be the same and the same as the maximum acceleration and maximum deceleration in the joint parameters. (This value must be negative)

Maximum jerk

This parameter is a reserved parameter and is currently invalid.

Pose movement maximum speed

The maximum speed of the robot during operation, if the instruction speed exceeds this value, it will be decelerated.

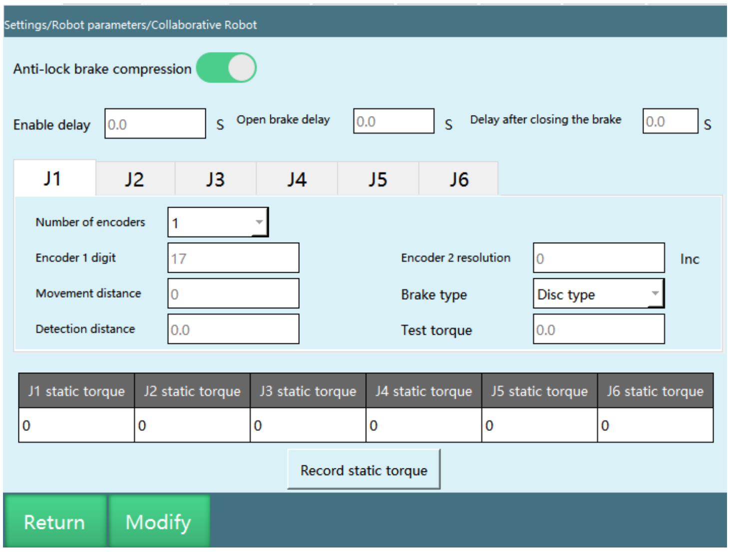
Speed limit method

Pose: The linear interpolation motion of the robot is limited by the maximum speed and the pose movement maximum speed.

Position: The linear interpolation motion of the robot is limited only by the maximum speed.

6-axis collaborative robot parameter setting

This interface is the parameter setting interface for 6-axis collaborative robots, other types of robots do not need to set.



Detailed usage of collaborative robot parameters

Enable delay

The delay time after pressing the enable key before issuing the enable instruction to the servo

Brake open delay

The delay time after issuing the enable instruction before issuing the brake open instruction to the servo

Delay after the brake is closed

The delay time elapsed from the closing of the brake until the servo responds to the next operation

Number of encoders

The number of encoders in single joint

Encoder 1 bits

The same as the encoder bits in the joint parameters

Encoder 2 resolution

The inc value of another encoder in single joint

Movement distance

The jogging distance of the joint before the brake is opened, generally 20

Brake type

Brake disc brake and pin-type brake

Detection distance

The joint movement distance used to detect whether the brake is open after opening the brake

Detection torque

After opening the brake, if the torque exceeds the detection torque when the joint runs detection distance, it is considered that the brake has failed to open

Appendix -Instruction Set

Catalogue

Appendix-Instruction Set

Bind variable parameter description

The following instructions add bind variables to the positional variable type:

P$INT: When the local integer variable (INT) is assigned a certain value, the local point P is the point represented by the value.

Usage example: I001 = 2 P$I001 is equivalent to P0002

P$GINT: When the global integer variable (GINT) is assigned a certain value, the local point P is the point represented by the value.

Usage example: GI001 =3 P$GI001 is equivalent to P0003

GP$INT: When the local integer variable (INT) is assigned a certain value, the global point GP is the point represented by the value.

Usage example: I001 = 4 GP$I001 is equivalent to GP0004

GP$GINT: When the global integer variable (GINT) is assigned a certain value, the global point GP is the point represented by the value.

Usage example: GI001 = 5 GP$GI001 is equivalent to GP0005

E$INT: When the local integer variable (INT) is assigned a certain value, the local point E is the point represented by the value.

Usage example: I001 = 6 E$I001 is equivalent to E0006

E$GINT: When the global integer variable (GINT) is assigned a certain value, the local point E is the point represented by the value.

Usage example: I001 = 7 E$I001 is equivalent to E0007

GE$INT: When the local integer variable (INT) is assigned a certain value, the global point GE is the point represented by the value.

Usage example: I001 = 8 GE$I001 is equivalent to GE0008

GE$GINT: When the global integer variable (GINT) is assigned a certain value, the global point GE is the point represented by the value.

Usage example: I001 = 9 GE$I001 is equivalent to GE0009

Motion control class

MOVJ-Point to point

Function

Move to the target point using joint interpolation. This instruction is used in the section where the robot is not constrained by trajectory in moving to the target point. The robot runs at the fastest speed in space.

Parameter description

P/GP: Use either a local position variable (P) or a global position variable (GP). When the value is "New", inserting this instruction will create a new P variable and record the robot's current position into that P variable.

VJ: The speed of the joint interpolation, the range is 1-100, and the unit is percentage. The actual movement speed is the maximum axis speed in the robot joint parameters multiplied by this percentage.

PL: Position level, range 0-5.

ACC: Acceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to the same value as VJ

DEC: Deceleration rate, the range is 1-100, the unit is percentage. It is recommended to set it to the same value as VJ.

TIME: Time, the range is a non-negative integer, and the unit is ms. Early execution time of the next instruction.

Note: When modifying the speed of MOVJ instruction, the acceleration & deceleration will be automatically displayed in a multiple of 1:1 with the speed. If you need to modify the acceleration or deceleration, you can do it manually.

Usage examples

MOVJ P0001 VJ = 10 % PL = 1 ACC = 10 DEC = 10 0

MOVJ GP0002 VJ = 10 % PL = 0 ACC = 7 DEC = 11 0

MOVL-Linear

Function

Move to the target point using linear interpolation. During the robot's movement to the target point, the movement trajectory of the robot end is a straight line.

Parameter description

P/GP: Use either a local position variable (P) or a global position variable (GP). When the value is "New", inserting this instruction will create a new P variable and record the robot's current position into that P variable.

V: Motion speed, the range is 1-1000 (The default Cartesian parameter maximum speed is 1000, the range varies according to the actual filled Cartesian parameter), the unit is mm/s.

PL: Position level, range 0-5.

ACC: Acceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to V*10%.

DEC: Deceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to V*10%.

TIME: Time, the range is a non-negative integer, and the unit is ms. Early execution time of the next instruction.

Note: When modifying the speed of MOVL instruction, the acceleration & deceleration will be automatically displayed in a multiple of 1:10 with the speed. If you need to modify the acceleration or deceleration, you can do it manually.

Usage examples

MOVL P0003 V = 200 mm/s PL = 2 ACC = 20 DEC = 20 0

MOVC-Circular

Note: MOVC, MOVCA, MOVCADOUBL, MOVS, MOVCEXT and other motion instructions that include circular, full circle and curve cannot be used alone, and their early execution speed, PL and other parameters are affected by the first (circular/full circle) instruction

Function

The robot moves in a circle through the 3 points taught by circular interpolation.

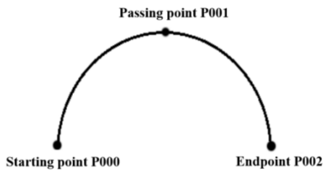
If the robot axis is taught by circular interpolation, the movement command is MOVC. The robot needs a MOVJ or MOVL instruction plus two MOVC instructions to walk a complete arc curve.

The starting point of single arc and the first arc of continuous arc can only be MOVJ or MOVL.

Single arc

When there is only one arc, as shown in the figure below, use circular interpolation to teach the three points of P000-P002.

If joint interpolation or linear interpolation is used to teach P000 before entering the arc, the trajectory of P000-P001 will automatically become a straight line.

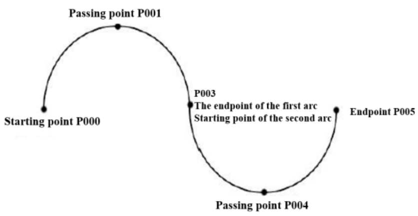


P000-Joint/Linear

P001-P002-Circular

Continuous arc

As shown in the figure below, when there are 2 or more consecutive circular arcs with curvature change, the arcs will eventually separate one by one. Therefore, please add joint or linear interpolation points at the connection point of the previous arc and the next arc.



P000-Joint/Linear

P001-P002-Circular

Parameter description

P/GP: Use either a local position variable (P) or a global position variable (GP). When the value is "New", inserting this instruction will create a new P variable and record the robot's current position into that P variable.

V: Motion speed, the range is 1-1000 (The default Cartesian parameter maximum speed is 1000, the range varies according to the actual filled Cartesian parameter), the unit is mm/s.

PL: Position level, range 0-5.

ACC: Acceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to V*10%.

DEC: Deceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to V*10%.

TIME: Time, the range is a non-negative integer, and the unit is ms. Early execution time of the next instruction.

Note: When modifying the speed of MOVC instruction, the acceleration & deceleration will be automatically displayed in a multiple of 1:10 with the speed. If you need to modify the acceleration or deceleration, you can do it manually.

Usage examples

MOVJ P0001 VJ = 10 % PL = 0 ACC = 1 DEC = 1 0

MOVC P0002 V = 100 mm/s PL = 0 ACC = 10 DEC = 10 0

MOVC P0003 V = 100mm/s PL = 0 ACC = 5 DEC = 8 0

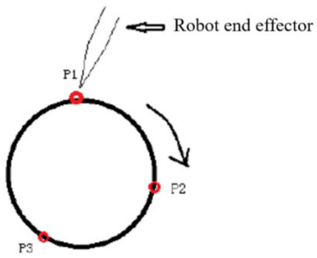
MOVCA-Full circle

Function

The robot walks a complete circle by teaching the starting point (MOVJ or MOVL) and two passing points (MOVCA) of the circle.

Instruction insertion prerequisites

Click on the "Tools" button in the upper status bar and select the previously calibrated tool hand.



Insertion steps for four instructions:

Click "Insert", click "Coordinate Switch Class", select "SWITCHTOOL" instruction, select the previously calibrated tool hand number.

Move to any point of the circle to be drawn such as P1 in the figure, click on "Insert", click on the "Motion Control" class and select MOVJ or MOVL.

Then move to any point of the circle to be drawn such as P2 in the figure (different from the point in step 2), click the "Coordinate System" button in the upper status bar, select the "Tool" coordinate system, click "Insert", click on the "Motion Control" class and select MOVCA

Move to any point of the circle to be drawn such as P3 in the figure (different from the points in steps 2 and 3), click the "Coordinate System" button in the upper status bar, select the "Tool" coordinate system, and click "Insert", click the "Motion Control" class, select MOVCA

Parameter description

P/GP: Use either a local position variable (P) or a global position variable (GP). When the value is "New", inserting this instruction will create a new P variable and record the robot's current position into that P variable.

V: Motion speed, the range is 1-1000 (The default Cartesian parameter maximum speed is 1000, the range varies according to the actual filled Cartesian parameter), the unit is mm/s.

PL: Position level, range 0-5.

ACC: Acceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to V*10%.

DEC: Deceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to V*10%.

TIME: Time, the range is a non-negative integer, and the unit is ms. Early execution time of the next instruction.

SPIN:

Attitude unchanged: Take a full circle trajectory with the same attitude as the P0001 calibrated attitude

6-axis non-rotating: Walk a full circle trajectory according to the calibrated attitude, while the 6-axis is fixed

6-axis rotating: Walk a full circle trajectory according to the calibrated attitude, and the 6-axis will rotate 360 degrees at the same time

Note: When modifying the speed of MOVCA instruction, the acceleration & deceleration will be automatically displayed in a multiple of 1:10 with the speed. If you need to modify the acceleration or deceleration, you can do it manually.

Usage examples

MOVJ P0001 VJ = 10 % PL = 0 ACC = 10 DEC = 10SPIN=1 0

MOVCA P0002 V = 100 mm/s PL = 0 ACC = 10 DEC = 10SPIN=1 0

MOVCA P0003 V = 100mm/s PL = 0 ACC = 10 DEC = 10SPIN=1 0

MOVS-Curve interpolation

Function

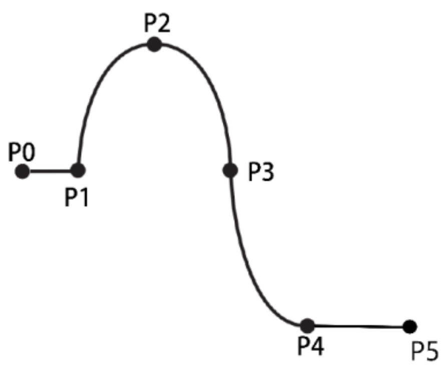
In welding, cutting, fusion welding, priming and other operations, if the free curve interpolation is used, the teaching operations for irregularly curved workpieces can become easier.

The trajectory is a spline curve passing through four points.

If free curve interpolation is used to teach the robot axis, the movement command is MOVS.

Single MOVS

Teach 4 points of P1-P4 as shown in the figure below. P1-P4 form a spline curve



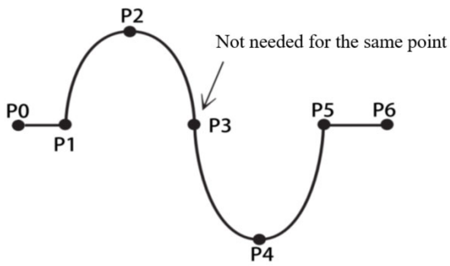
P0-Joint/Linear (The first motion instruction of the program cannot be MOVS)

P1-P4-Curve interpolation

P5-Joint/Linear

Continuous MOVS

A spline curve consisting of more than 4 points. P1-P5 form a spline curve



P0-Joint/Linear

P1-P5-Curve interpolation

P6-Joint/Linear

Note: The curve requires at least four curve points

Parameter description

P/GP: Use either a local position variable (P) or a global position variable (GP). When the value is "New", inserting this instruction will create a new P variable and record the robot's current position into that P variable.

V: Motion speed, the range is 1-1000 (The default Cartesian parameter maximum speed is 1000, the range varies according to the actual filled Cartesian parameter), the unit is mm/s.

PL: Position level, range 0-5.

ACC: Acceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to V*10%.

DEC: Deceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to V*10%.

TIME: Time, the range is a non-negative integer, and the unit is ms. Early execution time of the next instruction.

Note: When modifying the speed of MOVS instruction, the acceleration & deceleration will be automatically displayed in a multiple of 1:10 with the speed. If you need to modify the acceleration or deceleration, you can do it manually.

Usage examples

MOVJ P0001 VJ = 10 % PL = 0 ACC = 10 DEC = 10 0

MOVS P0002 V = 100 mm/s PL = 0 ACC = 10 DEC = 10 0

MOVS P0003 V = 100mm/s PL = 0 ACC = 10 DEC = 10 0

MOVS P0004 V = 100 mm/s PL = 0 ACC = 10 DEC = 10 0

MOVS P0005 V = 100mm/s PL = 0 ACC = 10 DEC = 10 0

IMOV-Increment

Function

Move from the current position according to the set incremental distance by means of joint or linear interpolation.

Parameter description

RP: Incremental variable, you can choose four kinds of coordinate systems: joint, Cartesian, tool and user, and fill in positive number for positive direction and negative number for negative direction for the corresponding axis. If not moving, fill in 0.

V/VJ:

When RP is the value in the joint coordinate system, here is VJ, i.e. the speed of joint interpolation, the range is 1-100, and the unit is percentage.

The actual movement speed is the maximum axis speed in the robot joint parameters multiplied by this percentage. When RP is the value in the Cartesian, tool, and user coordinate systems, here is V, i.e. the motion speed, the range is 1-1000 (The default Cartesian parameter maximum speed is 1000, the range varies according to the actual filled Cartesian parameter), and the unit is mm/s.

PL: Position level, range 0-5.

ACC: Acceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to V*10% or VJ.

DEC: Deceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to V*10% or VJ.

TIME: Time, the range is a non-negative integer, and the unit is ms. Early execution time of the next instruction

Note: When modifying the speed of IMOV instruction, the acceleration & deceleration will be automatically displayed in a multiple of 1:10 with the speed. If you need to modify the acceleration or deceleration, you can do it manually.

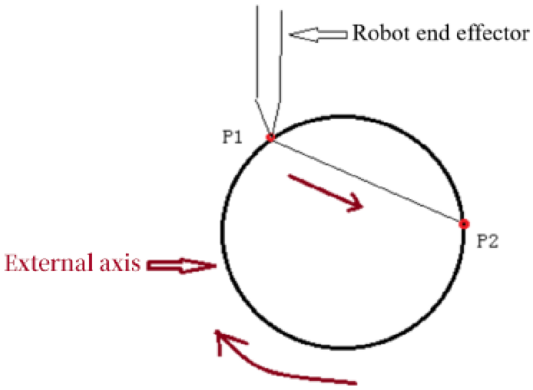
Usage examples

IMOV RP0001 V=10mm/s BF PL=0 ACC=1 DEC=1

MOVJEXT-External axis point to point

Function

The robot moves to the teaching position by means of joint interpolation, and the external axis moves by means of joint interpolation.



Parameter description

E: A variable that records robot and external axis position data simultaneously. When the value is "New", inserting this instruction will create a new E variable, and record the current position of the robot and the external axis to this E variable.

VJ: The speed of the joint interpolation, the range is 1-100, and the unit is percentage. The actual movement speed is the maximum axis speed in the robot joint parameters multiplied by this percentage. The external axis speed changes with the robot speed.

PL: Position level, range 0-5.

ACC: Acceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to the same value as VJ.

DEC: Deceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to the same value as VJ.

TIME: Time, the range is a non-negative integer, and the unit is ms. Early execution time of the next instruction

Note: When modifying the speed of MOVJEXT instruction, the acceleration & deceleration will be automatically displayed in a multiple of 1:1 with the speed. If you need to modify the acceleration or deceleration, you can do it manually.

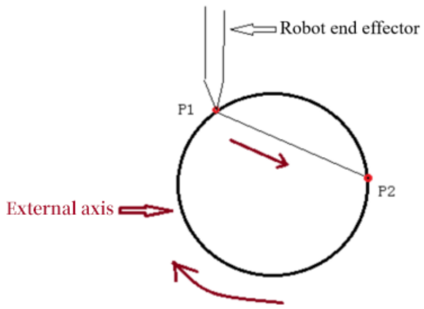
Usage examples

MOVJEXTE0001 VJ = 10 % PL = 0 ACC= 10 DEC = 10 0

MOVLEXT-External axis linear

Function

The robot moves to the teaching position by means of linear interpolation, and the external axis moves by means of joint interpolation.



Parameter description

E: A variable that records robot and external axis position data simultaneously. When the value is "New", inserting this instruction will create a new E variable, and record the current position of the robot and the external axis to this E variable.

V: Motion speed, the range is 1-1000 (The default Cartesian parameter maximum speed is 1000, the range varies according to the actual filled Cartesian parameter), the unit is mm/s. The external axis speed changes with the robot speed.

PL: Position level, range 0-5.

SYNC: Whether the robot moves synchronously with the external axis: when "Yes" is selected, the robot moves in a straight line in collaboration with the external axis; when "No" is selected, the robot moves in a straight line in space, and the external axis moves independently to the target angle.

ACC: Acceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to V*10%.

DEC: Deceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to V*10%.

Note: When modifying the speed of MOVLEXT instruction, the acceleration & deceleration will be automatically displayed in a multiple of 1:10 with the speed. If you need to modify the acceleration or deceleration, you can do it manually.

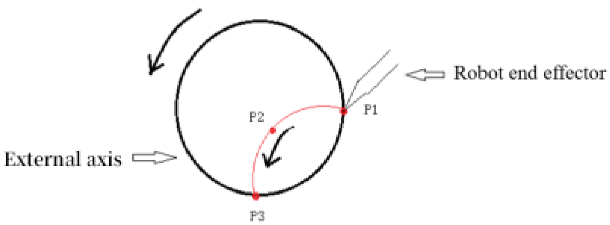
Usage examples

MOVLEXT E0002 V = 10 mm/s PL = 0 ACC = 1 DEC = 1 SYNC = 0 0

MOVCEXT-External axis circular

Function

The robot moves to the teaching position by means of circular interpolation, and the external axis moves by means of joint interpolation.



Parameter description

E: A variable that records robot and external axis position data simultaneously. When the value is "New", inserting this instruction will create a new E variable, and record the current position of the robot and the external axis to this E variable.

V: Robot motion speed, the range is 2-2000, the unit is mm/s. The external axis speed changes with the robot speed.

PL: Position level, range 0-5.

SYNC: Whether the robot moves synchronously with the external axis: when "Yes" is selected, the robot moves in an arc in collaboration with the external axis;

when "No" is selected, the robot moves in an arc in space, and the external axis moves independently to the target angle.

ACC: Acceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to V*10%.

DEC: Deceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to V*10%.

TIME: Time, the range is a non-negative integer, and the unit is ms. Early execution time of the next instruction.

Note: When modifying the speed of MOVCEXT instruction, the acceleration & deceleration will be automatically displayed in a multiple of 1:10 with the speed. If you need to modify the acceleration or deceleration, you can do it manually.

Usage examples

MOVLEXT E0002 V = 10 mm/s PL = 0 ACC = 1 DEC = 1 SYNC = 1 0

MOVCEXT E0003 V = 10 mm/s PL = 0 ACC = 1 DEC = 1 SYNC = 1 0

MOVCEXT E0004 V = 10 mm/s PL = 0 ACC = 1 DEC = 1 SYNC = 1 0

SPEED-Global speed

Function

The motion speed of all motion class instructions below the SPEED instruction is: instruction speed * speed in the upper status bar * SPEED (%).

Parameter description

Global speed (%): Speed percentage: 1-200.

Usage examples

SPEED= 9 %

SAMOV-Fixed-point movement

Function

The robot moves to a preset absolute position by joint interpolation.

If you do not want to move an axis, please leave the coordinate of the axis blank. (Do not fill in 0)

Parameter description

AP: Absolute position; four coordinate systems can be selected: joint, Cartesian, tool, and user; if the corresponding axis is not filled in, the corresponding axis will not move.

V/VJ:

When AP is the value in the joint coordinate system, here is VJ, i.e. the speed of joint interpolation, the range is 1-100, and the unit is percentage.

The actual movement speed is the maximum axis speed in the robot joint parameters multiplied by this percentage. When AP is the value in the Cartesian, tool, and user coordinate systems, here is V, i.e. the motion speed, the range is 1-1000 (The default Cartesian parameter maximum speed is 1000, the range varies according to the actual filled Cartesian parameter), and the unit is mm/s.

PL: Position level, range 0-5.

ACC: Acceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to V*10% or VJ.

DEC: Deceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to V*10% or VJ.

TIME: Time, the range is a non-negative integer, and the unit is ms. Early execution time of the next instruction.

Note: When modifying the speed of SAMOV instruction, the acceleration & deceleration will be automatically displayed in a multiple of 1:1 with the speed. If you need to modify the acceleration or deceleration, you can do it manually.

Usage examples

SAMOV AP0001 VJ= 10 %PL= 2 ACC= 10 DEC= 10

MOVJDOUBLE-Dual robot point to point

Function

When set to two robots, make two robots move to the target position by joint interpolation at the same time. Start and stop at the same time.

Parameter description

E: A variable that records the position data of two robots at the same time. When the value is "New", inserting this instruction will create a new E variable and record the current positions of the two robots to this E variable.

VJ: The speed of joint interpolation, the range is 1-100, and the unit is percentage. The actual movement speed is the maximum speed of the axis in the robot joint parameters multiplied by this percentage. The speeds of the two robots are synchronized.

ACC: Acceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to the same value as VJ.

DEC: Deceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to the same value as VJ.

TIME: Time, the range is a non-negative integer, and the unit is ms. Early execution time of the next instruction.

Note: When modifying the speed of MOVJDOUBLE instruction, the acceleration & deceleration will be automatically displayed in a multiple of 1:1 with the speed. If you need to modify the acceleration or deceleration, you can do it manually.

Usage examples

MOVJDOUBLE E0001 VJ = 10 % PL = 0 ACC= 10 DEC = 10 0

MOVLDOUBLE-Dual robot linear

Function

When set to two robots, make two robots move to the target position by linear interpolation at the same time. Start and stop at the same time.

Parameter description

E: A variable that records the position data of two robots at the same time. When the value is "New", inserting this instruction will create a new E variable and record the current positions of the two robots to this E variable.

V: Robot motion speed, the range is 1-1000 (The default Cartesian parameter maximum speed is 1000, the range varies according to the actual filled Cartesian parameter), and the unit is mm/s. The speeds of the two robots are synchronized.

ACC: Acceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to V*10%.

DEC: Deceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to V*10%.

TIME: Time, the range is a non-negative integer, and the unit is ms. Early execution time of the next instruction.

Note: When modifying the speed of MOVLDOUBLE instruction, the acceleration & deceleration will be automatically displayed in a multiple of 1:10 with the speed. If you need to modify the acceleration or deceleration, you can do it manually.

Usage examples

MOVLDOUBLE E0001 V = 100 mm/s PL = 0 ACC= 10 DEC = 10 0

MOVCDOUBLE-Dual robot circular

Function

When set to two robots, make two robots move to the target position by circular interpolation at the same time. Start and stop at the same time.

Parameter description

E: A variable that records the position data of two robots at the same time. When the value is "New", inserting this instruction will create a new E variable and record the current positions of the two robots to this E variable.

V: Robot motion speed, the range is 1-1000 (The default Cartesian parameter maximum speed is 1000, the range varies according to the actual filled Cartesian parameter), and the unit is mm/s. The speeds of the two robots are synchronized.

PL: Position level, range 0-5.

ACC: Acceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to V*10%.

DEC: Deceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to V*10%.

TIME: Time, the range is a non-negative integer, and the unit is ms. Early execution time of the next instruction.

Note: When modifying the speed of MOVCDOUBLE instruction, the acceleration & deceleration will be automatically displayed in a multiple of 1:10 with the speed. If you need to modify the acceleration or deceleration, you can do it manually.

Usage examples

MOVLDOUBLE E0001 VJ = 10 % PL = 0 ACC= 10 DEC = 10 0

MOVCDOUBLE E0002V = 100 mm/s PL = 0 ACC= 10 DEC = 10 0

MOVCDOUBLE E0003V = 100mm/s PL = 0 ACC= 10 DEC = 10 0

MOVCADOUBLE-Dual robot full circle

Function

When set to two robots, make the two robots move to the target position at the same time through full circle interpolation. Start and stop at the same time.

Parameter description

E: A variable that records the position data of two robots at the same time. When the value is "New", inserting this instruction will create a new E variable and record the current positions of the two robots to this E variable.

V: Robot motion speed, the range is 1-1000 (The default Cartesian parameter maximum speed is 1000, the range varies according to the actual filled Cartesian parameter), and the unit is mm/s. The speeds of the two robots are synchronized.

PL: Position level, range 0-5.

ACC: Acceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to V*10%.

DEC: Deceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to V*10%.

TIME: Time, the range is a non-negative integer, and the unit is ms. Early execution time of the next instruction.

Note: When modifying the speed of MOVCADOUBLE instruction, the acceleration & deceleration will be automatically displayed in a multiple of 1:10 with the speed. If you need to modify the acceleration or deceleration, you can do it manually.

Usage examples

MOVLDOUBLE E0001 VJ = 10 % PL = 0 ACC= 10 DEC = 10 0

MOVCADOUBLE E0002 V = 100 mm/s PL = 0 ACC= 10 DEC = 10 0

MOVCADOUBLE E0003 V = 100 mm/s PL = 0 ACC= 10 DEC = 10 0

MOVCOMM-External point

Function

Move to the point sent by the external device to the controller through Modbus or TCP in the specified interpolation mode.

Parameter description

Interpolation method: The interpolation method used when moving to the target point, including joint, linear, curve.

V/VJ:

When B is the value in the joint coordinate system, here is VJ, i.e. the speed of joint interpolation, the range is 1-100, and the unit is percentage.

The actual movement speed is the maximum axis speed in the robot joint parameters multiplied by this percentage. When B is the value in the Cartesian, tool, and user coordinate systems, here is V, i.e. the motion speed, the range is 1-1000 (The default Cartesian parameter maximum speed is 1000, the range varies according to the actual filled Cartesian parameter), and the unit is mm/s.

PL: Position level, range 0-5.

ACC: Acceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to V*10% or VJ.

DEC: Deceleration ratio, the range is 1-100, the unit is percentage. It is recommended to set it to V*10% or VJ.

TIME: Time, the range is a non-negative integer, and the unit is ms. Early execution time of the next instruction.

Usage examples

MOVCOMM MOVL VJ= 10 mm/s PL = 0 ACC = 1 DEC = 1 0

EXTMOV-External axis follow

Function

Instruction for external axes to follow the robot at a speed that is a multiple of the robot's linear speed or at a constant speed.

Parameter description

External axis: One axis of O1-O5 can be selected to follow

Type:

Follow type: Change speed with robot real time linear speed

K: External axis speed (°/s)=K*Linear speed (mm/s)

Constant speed type: Run at a constant speed

Speed value source: optional INT/DOUBLE/GINT/GDOUBLE/Hand-filled.

Variable name: Used to select variables when the speed value source is INT/DOUBLE/GINT/GDOUBLE

Hand-filled value: When the speed value source is hand-filled, it is used to input the speed value for constant operation

Usage examples

EXTMOV O1 FOLLOW 22.22

GEARIN-Electronic gear

Function

An instruction to make a certain axis of an external axis move along with a certain axis of the robot.

Parameter description

Spindle: J1~J6 axes of the robot can be selected

External axis: One axis of O1-O5 can be selected to follow (O3 is not supported at the moment)

Proportional relationship K: Follow axis speed (°/s) = K*Spindle speed (°/s)

Usage examples

GEARIN J1 O1 22.22

MRESET-Reset external axis multi-turn rotation amount

Function

The maximum and minimum limits set according to the external axis rotation axis. When the external axis rotates beyond the limit, use this instruction to reset the external peripheral coordinates and continue to rotate, so that the external axis will not report an error due to overrun.

Parameter description

MRESET: all axes or single axis

Usage examples

MRESET 0

DRAG_TRAJECTORY-Drag teaching

Function

The robot runs according to the previously recorded trajectory.

Parameter description

Track name: The name of the robot track

Playback rate: Motion speed (0~500%)

Usage examples

DRAG_TRAJECTORY Track1 20%

SWITHCPAYLOAD-Switch load parameters

Function

In actual operation, the actual load and load parameter match

SWITHCPAYLOAD instruction is used to switch load parameters

It affects collision detection and torque feedforward

Parameter description

Load number: You can fill in the tool number or use the bind variable function

Usage examples

SWITHCPAYLOAD 1

MOVARCH-Arch motion

Function

Allows the robot to move along an arch-type trajectory

Parameter description

P/GP: Use either a local position variable (P) or a global position variable (GP). When the value is "New", inserting this instruction will create a new P variable and record the robot's current position into that P variable.

V: Motion speed, the range is 1-1000 (The default Cartesian parameter maximum speed is 1000, the range varies according to the actual filled Cartesian parameter), and the unit is mm/s.

PL: Position level, range 0-5.

ACC: Acceleration ratio, the range is 1-100, the unit is percentage.

DEC: Deceleration rate, the range is 1-100, the unit is percentage.

Displacement axis: (X, Y, Z) The axis that is displaced during arch-type movement, and the standard arch-type movement is displaced in the Z-axis direction.

Displacement distance: the distance that needs to be displaced on the displacement axis. The standard arch-type movement is 25mm displacement on the Z axis.

TIME: Time, the range is a non-negative integer, and the unit is ms. Early execution time of the next instruction.

View trajectory diagram: You can view the arch-type motion trajectory diagram

Note: When modifying the speed of MOVARCH instruction, the acceleration & deceleration will be automatically displayed in a multiple of 1:1 with the speed. If you need to modify the acceleration or deceleration, you can do it manually.

Usage examples

MOVARCH P0001 V=10 PL=0 ACC=10 DEC=10 X 10 0

MOVARCH GP0001 V=10 PL=0 ACC=10 DEC=10 X 10 0

Input and output class

DIN-IO input

Function

Read the digital input value of the IO board and store it into an integer or Boolean variable.

Parameter description

Port value storage: Store the input value in the variable name and variable type of the target variable.

Input IO board: If there are multiple EtherCAT IOs, you can choose the corresponding IO board.

Input group number (Number of input channels): The input is read according to the group, which is a group of 1 channel, 4 channels and 8 channels. For a group of 1 channel, 16 DIN ports are 16 groups; for a group of 4 channels, 1-4, 5-8, 9-12, 13-16 as a group; for a group of 8 channels, 1-8, 9-16 as a group. The data read into the variable is to convert the input port value from binary to decimal and store it in the variable.

For example: a group of 8 channels, the value of port 1-8 is 10110101, then starting from port 8 is 10101101. Convert it to decimal as 173, then store it in variable as 173.

Usage examples

DIN I001 IN#(5)

DOUT-IO output

Function

Set the corresponding IO port on the IO board to high or low.

Parameter description

Output IO board: If there are multiple EtherCAT IOs, you can choose the corresponding IO board.

Output group number (Number of output channels): The output is output according to the group, which is a group of 1 channel, 4 channels and 8 channels. For a group of 1 channel, 16 DOUT ports are 16 groups; for a group of 4 channels, 1-4, 5-8, 9-12, 13-16 as a group; for a group of 8 channels, 1-8, 9-16 as a group.

Output value (Variable source): Divided into manual selection and variable type. Manual selection is to check the box below, the selected output is 1, and the unselected output is 0. Example: When the output group number is 4-way output, the second group, port 1 and port 3 are selected in the selection box below, and the other two are left blank, then when this command is run, the output ports of the IO board are numbered 5-8 The output value of the port is 1010. When INT, GINT, BOOL, GBOOL is selected as the variable source, the corresponding variable value will be converted into binary and output to the IO board.

Example: If the variable value is 173, it will be 10101101 when converted to binary. If 8 channels are a group, and the binary value is output from port 8, then the value of port 8-1 is 10101101, and the value of port 1-8 is 10110101.

Variable name: When the variable source is INT, GINT, BOOL, GBOOL, select the variable name to be output here.

Time: The time to reverse the output, the output is reversed after the specified time. For example, DOUT1=1, time is 2, then DOUT1 outputs high level for 2 seconds and then turns to low level. If the time is 0, it will output high level continuously.

Error stop processing: Output value hold, that is, when an error is reported, the io continues to output according to the parameters set by the instruction; Time-out stop, that is, stop at the end of the timing

Usage examples

DOUTOT#(1) I001 0

AIN-Analog input

Function

Read the input value of the corresponding analog input port into the target variable.

Parameter description

Analog input port: The analog input port to be read.

Variable value source: The variable type of the target variable.

Variable name: The variable name of the target variable.

Usage examples

AIN D001 B001

AOUT-Analog output

Function

Set the output value of the corresponding analog output port to the defined value.

Parameter description

Analog output port: The port to be output.

Variable value source: The variable type of the value to be output.

New parameter: When the variable value is customized, enter the hand-filled data here, the range is 0-10V, and the corresponding port will output the value.

Variable name: The variable name of the variable whose value is to be output.

Usage examples

AOUTAOUT1 1.1

PULSEOUT-Pulse output

Function

Output on pin 4 (PWM+) of the DB9 terminal on the R1 PWM IO board according to the set pulse frequency and number.

Parameter description

Number: Number of pulses.

Frequency: Pulse frequency.

Usage examples

PULSEOUT RATE = 100 SUM = 100

READ_DOUT-Read output

Function

Read the output status of the digital output port and store it in the target variable.

Parameter description

Output IO board: If there are multiple EtherCAT IOs, you can choose the corresponding IO board.

Variable type: The variable type of the target variable to be stored.

Variable name: The variable name of the target variable to be stored.

Output group number (Number of output channels): The value of the output port is read according to the group, which is a group of 1 channel, 4 channels and 8 channels. For a group of 1 channel, 16 DOUT ports are 16 groups; for a group of 4 channels, 1-4, 5-8, 9-12, 13-16 as a group; for a group of 8 channels, 1-8, 9-16 as a group.

For example: a group of 8 channels, the value of port 1-8 is 10110101, then starting from port 8 is 10101101. If it is converted to decimal, it is 173, then the variable is 173.

Usage examples

READ_DOUT I001 OT#(1)

Timer class

TIMER-Delay

Function

Delay for the set value, and then continue to run.

Parameter description

Variable value source: You can manually fill in the value in the new parameter. You can also select the bind variable in "More" option, which will delay the length of time corresponding to the variable value.

Usage examples

TIMER T= 10

Operation class

ADD-Add

Function

Addition operation (+), A=A+B.

Parameter description

Variable: The variable type of the summand A, you can fill in by hand or select the variable type in "More" option.

Variable value: The variable type of the addend B, you can fill in by hand or select the variable type in "More" option.

Usage examples

ADD GI001 22; Meaning: GI001=GI001+22

ADD GI002 I003; Meaning: GI002=GI002+I003

SUB-Subtract

Function

Subtraction operation (-), A = A - B.

Parameter description

Variable: The variable type of the minuend A, you can fill in by hand or select the variable type in "More" option.

Variable value: The variable type of the subtrahend B, you can fill in by hand or select the variable type in "More" option.

Usage examples

SUB GI001 22; Meaning: GI001=GI001-22

SUB GI002 I003; Meaning: GI002=GI002-I003

MUL-Multiply

Function

Multiplication (*), A=A*B.

Parameter description

Variable: The variable type of the multiplicand A, you can fill in by hand or select the variable type in "More" option.

Variable value: The variable type of the multiplier B, you can fill in by hand or select the variable type in "More" option.

New parameter: When the variable value source selects "Custom", this input box is valid, and the filled value is the value of B.

Source parameter: When the variable value source selects "Variable", here is the variable name of B.

Usage examples

MUL GI001 22; Meaning: GI001=GI001*22

MUL GI002 I003; Meaning: GI002=GI002*I003

DIV-Divide

Function

Division operation (÷), A = A÷B.

Parameter description

Variable: The variable type of the dividend A, you can fill in by hand or select the variable type in "More" option.

Variable value: The variable type of the divisor B, you can fill in by hand or select the variable type in "More" option.

Usage examples

DIV GI001 22; Meaning: GI001=GI001÷22

DIV GI002 I003; Meaning: GI002=GI002÷I003

MOD-Modulo

Function

Modulo operation (Mod), A=A Mod B.

Parameter description

Variable: The variable type of the dividend A, you can fill in by hand or select the variable type in "More" option.

Variable value: The variable type of the divisor B, you can fill in by hand or select the variable type in "More" option.

Usage examples

MOD GI001 22; Meaning: GI001=GI001 Mod 22

MOD GI002 I003; Meaning: GI002=GI002 Mod I003

SIN-Sine

Function

Sine operation (sin), A=sin(B), the unit of B is radians (rad).

Parameter description

Variable: The variable type of the result value A, you can fill in by hand or select the variable type in "More" option.

Variable value: The variable type of sine radian value B, you can fill in by hand or select the variable type in "More" option.

Usage examples

SIN GI001 22; Meaning: GI001=sin(22)

SIN GI002 I003; Meaning: GI002=sin(I003)

COS-Cosine

Function

Cosine operation (cos), A=cos(B), the unit of B is radians (rad).

Parameter description

Variable: The variable type of the result value A, you can fill in by hand or select the variable type in "More" option.

Variable value: The variable type of the cosine radian value B, you can fill in by hand or select the variable type in "More" option.

Usage examples

COS GI001 22; Meaning: GI001=cos(22)

COS GI002 I003; Meaning: GI002=cos(I003)

ATAN-Arctangent

Function

Arctangent operation (cos), A=arctan(B), the unit of B is radians (rad).

Parameter description

Variable: The variable type of the result value A, you can fill in by hand or select the variable type in "More" option.

Variable value: The variable type of the arctangent radian value B, you can fill in by hand or select the variable type in "More" option.

Usage examples

ATAN GI001 22; Meaning: GI001=arctan(22)

ATAN GI002 I003; Meaning: GI002=arctan(I003)

LOGICAL_OP-Logic operation

Function

Logic operation (and/or/not), B001 = I001andI002.

Parameter description

Parameter 1 type: The variable type of parameter 1 involved in the operation.

Parameter 1 name: The variable name of parameter 1 involved in the operation.

Operation type: Logical AND (&&), logical OR (||), logical NOT (!).

Parameter 2 type: The variable type of parameter 2 involved in the operation.

Parameter 2 name: The variable name of parameter 2 involved in the operation.

Result storage variable type: The type of variable where the result of the operation is stored.

Result storage variable name: The name of the variable where the result of the operation is stored.

Usage examples

LOGICAL_OPB001 = I001AND 10; Meaning: Variable I001, constant 10 logic and operation results are stored in B001

Condition control class

Note: When conditional judgment needs to use strings for comparison, the actual comparison is the ASCII code value corresponding to the character

CALL-Call subprogram

Function

Call another program, after the called program has finished running, the program will return to the next line of the original program below the CALL instruction to continue running.

Parameter description

CALL: The name of the called program.

Usage examples

CALL [Program]; Meaning: Call the program Program

CALL_LUAFILE-Call LUA file

Function

Call the Lua file uploaded from upgrade.

Parameter description

CALL_LUAFILE: Call Lua file

Number of incoming parameters: The number of incoming parameters in the Lua file

Incoming parameter selection: Select the number and value of the required incoming parameters (the number should be the same as the actual Lua file)

Number of output parameters: The number of output parameters in the Lua file

Output parameter selection: Select the number and value of the required output parameters (the number can be less than the actual Lua file)

Usage examples

CALL_LUAFILE [$demo.lua$] IN (1.0,2.0,3.0,) OUT (2.0,2.0)

That is, call the Lua file demo.lua, pass three values into the demo, namely 1, 2, 3, and the demo will send out two values, both of which are 2.

IF-If

Function

If the condition of IF instruction is satisfied, the instruction between IF and ENDIF will be executed, if the condition of IF instruction is not satisfied, the program will jump to ENDIF instruction and continue to run the instruction below ENDIF without running the instruction between IF and ENDIF.

The judgment condition of IF is (comparand 1 comparison mode comparand 2), for example, comparand 1 is 2, comparand 2 is 1, comparison mode is ">", then

2>1, the judgment condition is established; if comparison mode is "<" or "==", the judgment condition is not established.

The IF instruction can be used alone or in combination with the ELSEIF and ELSE instructions. Note that the ELSEIF and ELSE instructions cannot be used independently of the IF instruction!

Note: When the beginning of the program is IF and the last line is ENDIF instruction, please insert a TIMER (delay, 0.1s) instruction above the IF instruction or below the ENDIF instruction, otherwise the program will crash when the conditions of the IF instruction are not met.

When inserting an IF instruction, an ENDIF instruction will be inserted at the same time. When deleting an IF instruction, please be careful to delete the corresponding ENDIF instruction, otherwise the program will not run.

IF instruction can nest another IF instruction or other conditional judgment instructions such as WHILE and JUMP.

Now the IF instruction supports multi-condition judgment and judges in order. If there are parentheses, first judge the ones inside the parentheses, and then judge the ones outside the parentheses. Up to 5 judgment conditions are supported.

Parameter description

Parameter type: The type of comparand 1: variable or digital/analog input value.

Parameter name:

If the selected parameter type is "Variable" (INT, DOUBLE, BOOL, GINT, GDOUBLE, GBOOL), then this is the variable name of comparand 1.

If the selected parameter type is input value (DIN, AIN), then here is the port number of the digital input or analog input

Comparison mode:

== equal to

< less than

> greater than

<= less than or equal to

>= greater than or equal to

!= not equal to

Variable value source: The type of comparand 2: custom, variable or digital/analog input value.

New parameter:

If the variable value source is "Custom", then this is not selectable.

If the variable value source is "Variable" (INT, DOUBLE, BOOL, GINT, GDOUBLE, GBOOL), then this is the variable name of the comparand 1.

If the variable value source is input value (DIN, AIN), then here is the port number of the digital input or analog input

Source parameter: If the variable value source is "Custom", fill in the value of the comparand 2 directly here.

Usage examples

IF(GI001>=D001)

Other instructions, such as MOVJ, etc.

ENDIF

ELSEIF-Else if

Function

The ELSEIF instruction must be inserted between IF and ENDIF. An ELSE instruction or multiple ELSEIF instructions can also be inserted between ELSEIF and ENDIF.

When the condition of IF is satisfied, the ELSEIF instruction and the instructions between ELSEIF and ENDIF will be ignored, only the instructions between IF and ELSEIF will be run, and then jump to the line of instruction below ENDIF to continue running.

When the condition of IF is not satisfied, the program will jump to the ELSEIF instruction to judge the judgment condition of ELSEIF. If it is met, run the instructions between ELSEIF and ENDIF, and then continue to run the instruction below ENDIF; if it is not met, jump directly to the line of instruction below ENDIF to continue running.

If multiple ELSEIFs are nested in IF and ENDIF, when the judgment condition of IF is not established, the judgment condition of the first ELSEIF is first judged, if it is established, the instruction between the first ELSEIF and the second ELSEIF is executed; if it is not established, then judge the judgment condition of the second ELSEIF, and so on.

Note: When deleting the IF instruction, the corresponding ELSE and ENDIF instructions must be deleted, otherwise the program will not run.

Now the ELSEIF instruction supports multi-condition judgment and judges in order. If there are parentheses, first judge the ones inside the parentheses, and then judge the ones outside the parentheses. Up to 5 judgment conditions are supported.

Parameter description

Parameter type: The type of comparand 1: variable or digital/analog input value.

Parameter name:

If the selected parameter type is "Variable" (INT, DOUBLE, BOOL, GINT, GDOUBLE, GBOOL), then this is the variable name of comparand 1.

If the selected parameter type is input value (DIN, AIN), then here is the port number of the digital input or analog input

Comparison mode:

== equal to

< less than

> greater than

<= less than or equal to

>= greater than or equal to

!= not equal to

Variable value source: The type of comparand 2: custom, variable or digital/analog input value.

New parameter:

If the variable value source is "Custom", then this is not selectable.

If the variable value source is "Variable" (INT, DOUBLE, BOOL, GINT, GDOUBLE, GBOOL), then this is the variable name of the comparand 1.

If the variable value source is input value (DIN, AIN), then here is the port number of the digital input or analog input

Source parameter: If the variable value source is "Custom", fill in the value of the comparand 2 directly here.

Usage examples

IF(GI001>=D001)

Other instruction 1, such as MOVJ, etc.

ELSEIF(D001<9)

Other instruction 2, such as MOVJ, etc.

ENDIF

ELSE-Else

Function

The ELSE instruction must be inserted between IF and ENDIF, but only one ELSE instruction can be embedded in an IF instruction.

When the judgment condition of IF is established, the instruction between IF and ELSE will be executed and then jump to the next line of instruction of ENDIF to continue running, without running the instruction between ELSE and ENDIF.

When the judgment condition of IF is not established, the program will jump to the instruction between ELSE and ENDIF to continue running, without running the instruction between IF and ELSE.

Note: When deleting the IF instruction, the corresponding ELSE and ENDIF instructions must be deleted, otherwise the program will not run.

Parameter description

None

Usage examples

IF(GI001<9)

Other instruction 1, such as MOVJ, etc.

ELSE

Other instruction 2, such as MOVJ, etc.

ENDIF

WAIT-Wait

Function

WAIT means wait, you can choose whether to have a wait time or not. If the "TIME" option is not checked, the program will stay in the WAIT instruction and wait until the judgment condition is met. If the "TIME" option is checked, the program will wait for the length of the parameter and then continue to run the next instruction. If the condition becomes true while waiting, the next instruction will be run immediately.

Now the WAIT instruction supports multi-condition judgment and judges in order. If there are parentheses, first judge the ones inside the parentheses, and then judge the ones outside the parentheses. Up to 5 judgment conditions are supported.

Parameter description

Parameter type: The type of comparand 1: variable or digital/analog input value.

Parameter name:

If the selected parameter type is "Variable" (INT, DOUBLE, BOOL, GINT, GDOUBLE, GBOOL), then this is the variable name of comparand 1.

If the selected parameter type is input value (DIN, AIN), then here is the port number of the digital input or analog input

Comparison mode:

== equal to

< less than

> greater than

<= less than or equal to

>= greater than or equal to

!= not equal to

Variable value source: The type of comparand 2: custom, variable or digital/analog input value.

New parameter:

If the variable value source is "Custom", then this is not selectable.

If the variable value source is "Variable" (INT, DOUBLE, BOOL, GINT, GDOUBLE, GBOOL), then this is the variable name of the comparand 1.

If the variable value source is input value (DIN, AIN), then here is the port number of the digital input or analog input

Source parameter: If the variable value source is "Custom", fill in the value of the comparand 2 directly here.

TIME:

Optional, if you do not select this option, you will wait forever until the condition is met.

If selected, you can fill in the waiting time (seconds), and after that time, even if the condition is still not met, the program will still jump to the next line and continue to run.

Is PL continuous: refers to whether the smoothness of the trajectory curve of the robot will be interrupted

Continuous: The robot running curve is relatively smooth after the conditions are met

Not continuous: The smoothness of the robot's trajectory is interrupted after the conditions are met

Filter time:

Optional, no effect if not selected.

If selected, it refers to the required input signal duration. When the input signal duration meets the filter time (without waiting for TIME), jump to the next line to continue running. If the filter time is not met, wait for the TIME time and then jump to the next line to continue running.

Usage examples

WAIT (GI001==2) T = 2 F = 1

WHILE-Loop

Function

When the condition of WHILE instruction is satisfied, the instruction between WHILE and ENDWHILE will be run cyclically. If the judgment condition is not satisfied before running to WHILE instruction, the program will jump to ENDWHILE instruction directly when running to the WHILE instruction without running the instruction between WHILE and ENDWHILE; if the judgment condition becomes not satisfied during running the instruction between WHILE and ENDWHILE, the program will continue to run until it reaches ENDWHILE line, it will not loop but continue to run the instruction below ENDWHILE.

The judgment condition of WHILE is (comparand 1 comparison mode comparand 2), for example, comparand 1 is 2, comparand 2 is 1, comparison mode is ">", then 2>1, the judgment condition is established; if comparison mode is "<" or "==", the judgment condition is not established.

Note: When inserting an WHILE instruction, an ENDWHILE instruction will be inserted at the same time. When deleting an WHILE instruction, please be careful to delete the corresponding ENDWHILE instruction, otherwise the program will not run.

When the beginning of the program is WHILE and the last line is ENDWHILE instruction, please insert a TIMER (delay, 0.3s) instruction at the beginning or end of the program, otherwise the program will crash when the conditions of the WHILE instruction are not met.

The WHILE instruction can be nested with multiple WHILE, IF or JUMP and other judgment class instructions.

Now the WHILE instruction supports multi-condition judgment and judges in order. If there are parentheses, first judge the ones inside the parentheses, and then judge the ones outside the parentheses. Up to 5 judgment conditions are supported.

Parameter description

Parameter type: The type of comparand 1: variable or digital/analog input value.

Parameter name:

If the selected parameter type is "Variable" (INT, DOUBLE, BOOL, GINT, GDOUBLE, GBOOL), then this is the variable name of comparand 1.

If the selected parameter type is input value (DIN, AIN), then here is the port number of the digital input or analog input

Comparison mode:

== equal to

< less than

> greater than

<= less than or equal to

>= greater than or equal to

!= not equal to

Variable value source: The type of comparand 2: custom, variable or digital/analog input value.

New parameter:

If the variable value source is "Custom", then this is not selectable.

If the variable value source is "Variable" (INT, DOUBLE, BOOL, GINT, GDOUBLE, GBOOL), then this is the variable name of the comparand 1.

If the variable value source is input value (DIN, AIN), then here is the port number of the digital input or analog input

Source parameter: If the variable value source is "Custom", fill in the value of the comparand 2 directly here.

Usage examples

WHILE(GI001<2)

Other instruction 1, such as MOVJ, etc.

WHILE(D001<10)

Other instruction 2, such as MOVJ, etc.

ADD D001 1

ENDWHILE

Other instruction 3

ADD GI001 1

ENDWHILE

LABEL-Label

Function

The target label of JUMP instruction jump.

Parameter description

Label name: The name of the label, which needs to be a string starting with a letter.

Usage examples

LABEL *A1

JUMP-Jump

Function

JUMP is used to jump and must be used in conjunction with the LABEL (label) instruction.

JUMP can be set with or without judgment conditions. When the JUMP instruction is set to have no judgment condition, the program will jump directly to the corresponding LABEL instruction when running to the instruction and continue to run the next instruction below LABEL.

When the JUMP instruction is set to have a judgment condition, if the condition is satisfied, jump to the LABEL instruction line; if the condition is not satisfied, ignore the JUMP instruction and continue to run the next instruction below the JUMP instruction.

LABEL label can be inserted above or below JUMP, but cannot jump across programs.

LABEL label name must consist of two or more characters starting with a letter.

Inserting the LABEL label has no effect on the operation of the program, but it must comply with the rules of program operation, for example, it cannot be inserted above the MOVC instruction or the local variable definition instruction.

Now the JUMP instruction supports multi-condition judgment and judges in order. If there are parentheses, first judge the ones inside the parentheses, and then judge the ones outside the parentheses. Up to 5 judgment conditions are supported.

Parameter description

Label name: The label name of the inserted LABEL instruction, optional.

Judgment condition:

Optional, if selected, the judgment condition can be set. If not selected, the program will jump directly after running to JUMP.

Parameter type: The type of comparand 1: variable or digital/analog input value.

Parameter name:

If the selected parameter type is "Variable" (INT, DOUBLE, BOOL, GINT, GDOUBLE, GBOOL), then this is the variable name of comparand 1.

If the selected parameter type is input value (DIN, AIN), then here is the port number of the digital input or analog input

Comparison mode:

== equal to

< less than

> greater than

<= less than or equal to

>= greater than or equal to

!= not equal to

Variable value source: The type of comparand 2: custom, variable or digital/analog input value.

New parameter:

If the variable value source is "Custom", then this is not selectable.

If the variable value source is "Variable" (INT, DOUBLE, BOOL, GINT, GDOUBLE, GBOOL), then this is the variable name of the comparand 1.

If the variable value source is input value (DIN, AIN), then here is the port number of the digital input or analog input

Source parameter: If the variable value source is "Custom", fill in the value of the comparand 2 directly here.

Usage examples

MOVJ

LABEL *C1

Other instruction 1, such as MOVJ, etc.

JUMP *C1 WHEN (I001==0)

Other instruction 2

UNTIL-Until

Function

The UNTIL instruction is used to jump out of a motion process. This means that the program pauses during one motion and starts the next one. When the condition is met, the program immediately pauses and starts an instruction below the ENDUNTIL instruction, regardless of whether the robot is currently running or not.

The judgment condition of UNTIL is (comparand 1 comparison mode comparand 2), for example, comparand 1 is 2, comparand 2 is 1, comparison mode is ">", then 2>1, the judgment condition is established; if comparison mode is "<" or "==", the judgment condition is not established.

Note: When inserting an UNTIL instruction, an ENDUNTIL instruction will be inserted at the same time. When deleting an UNTIL instruction, please be careful to delete the corresponding ENDUNTIL instruction, otherwise the program will not run.

Now the UNTIL instruction supports multi-condition judgment and judges in order. If there are parentheses, first judge the ones inside the parentheses, and then judge the ones outside the parentheses. Up to 5 judgment conditions are supported.

Parameter description

Parameter type: The type of comparand 1: variable or digital/analog input value.

Parameter name:

If the selected parameter type is "Variable" (INT, DOUBLE, BOOL, GINT, GDOUBLE, GBOOL), then this is the variable name of comparand 1.

If the selected parameter type is input value (DIN, AIN), then here is the port number of the digital input or analog input

Comparison mode:

== equal to

< less than

> greater than

<= less than or equal to

>= greater than or equal to

!= not equal to

Variable value source: The type of comparand 2: custom, variable or digital/analog input value.

New parameter:

If the variable value source is "Custom", then this is not selectable.

If the variable value source is "Variable" (INT, DOUBLE, BOOL, GINT, GDOUBLE, GBOOL), then this is the variable name of the comparand 1.

If the variable value source is input value (DIN, AIN), then here is the port number of the digital input or analog input

Source parameter: If the variable value source is "Custom", fill in the value of the comparand 2 directly here.

Usage examples

UNTIL(GI001<2)

Other instructions

ENDUNTIL

MOVJ P0003

CRAFTLINE-Process jump

Function

Special process instruction, after running this instruction in the program, the program will jump to the corresponding line in the special process interface.

Parameter description

New parameter: The corresponding line number in the special process interface.

Usage examples

CRAFTLINE 22

CMDNOTE-Instruction note

Function

Instruction note, you can use this instruction to add note in the appropriate position of the program for debugging.

If a note instruction is inserted, when single-stepping this instruction, the program will jump to the next line of instruction to run, and there will be no error prompt.

Parameter description

Note content: The note content supports Chinese and English, case, number input and symbol input.

Usage examples

##iINEXBOT$$; Meaning: The content of the note is "INEXBOT".

POS_REACHABLE-Judgment of reachability

Function

Reachability judgment instruction, used to judge whether the target point can be reached, if the point can be reached, set the variable to 1, if not, set it to 0.

Parameter description

Position variable name: P point, G point.

Motion type: MOVJ, MOVL.

Status storage variable type: BOOL, GBOOL.

Status storage variable name: BOOL,GBOOL variable name.

Usage examples

POS_REACHABLE MOVJ P001 B001; Meaning: Calculate if it is possible to run to position P001 using MOVJ interpolation, if it is possible to reach, the value of B001 is 1, if not, the value of B001 is 0.

CLKSTART-Timing start

Function

The CLKSTART instruction is used for timing. Run this instruction to start the timing and record the time in a local or global DOUBLE variable.

Accuracy of timing instruction is two decimal places (i.e. 10ms, error ±2ms)

Parameter description

Serial number: The serial number of the timer, you can use 32 timers to time separately at the same time.

Storage variable type: Store the timed time into a local DOUBLE variable or a global GDOUBLE variable.

Storage variable name: The variable name of the variable that stores the time.

Usage examples

CLKSTARTID= 1 D001; Meaning: Process number 1 starts timing, and the timing result is stored in D001.

CLKSTOP-Timing stop

Function

The CLKSTOP instruction is used to stop the timing of the timer of the corresponding serial number. The value stored in the variable will not reset to zero after stopping.

Parameter description

Serial number: The serial number of the timer to stop timing.

Usage examples

CLKSTOPID=1; Meaning: Process number 1 stops timing

CLKRESET-Timing reset

Function

The CLKRESET instruction is used to reset the timer to zero for the corresponding serial number. If this instruction is not used, the next time the CLKSTART instruction is run, the timing will be accumulated.

Parameter description

Serial number: The serial number of the timer to be reset to zero.

Usage examples

CLKPESETID=1; Meaning: Reset the timing result of process number 1.

READLINEAR-Read linear speed

Function

Read the linear speed of the robot into the variable in real time

Parameter description

Variable type: The type of the storage variable, optional GINT/GDOUBLE.

Variable name: The name of the storage variable.

Usage examples

READLINEAR GDOO1

CALL_LUASTRING-Call LUA statement

Function

Realize corresponding function or operation by calling Lua statement

Parameter description

Statement: the Lua statement to be entered

More: hand-filled and variables;

Hand-filled: enter the corresponding and correct Lua statement by yourself, and you can directly step or run

Variables: Write Lua statements into string variables, and realize their functions by calling the corresponding string variables

Usage examples

CALL_LUASTRING [statement]

or CALL_LUASTRING string variable

Variable class

SET-Assignment

(Define the variable and go directly to the local variable interface to cancel the instruction)

Function

Define local integer, float, and Boolean variables, and assign values at the same time.

Parameter description

Variable: Click "More" to select the desired variable type.

Variable value source: Assign value to the above variable, you can fill in by hand or select the variable type in "More" option.

Usage examples

INTI001 = 11

INT I002 = GI003

FORCESET-Write to file

Function

During the running of the program, all calculations and assignment operations are to change the values in the cache, and the values will not be stored in the system file, that is, the values of all global variables will be restored when the program stops running.

To force the global value variables in the content to be written to the file, you can use the FORCESET instruction.

Parameter description

Variable name: Click "More" to select the variable name you want to force writing to the file.

Usage examples

FORCESETGI001

String class

STRING-SPELL-String append

Function

Add the characters you need to the variable of the original string or the variable of the empty string to form a new string variable

Parameter description

Variable: Variable type and name

Variable value: Constant or bind other variable

Usage examples

STRING_SPELL [S001 + S002]

STRING-SLICE-String index interception

Function

Intercept a part of a string in the string variable and store it in the specified variable

Parameter description

Variable: The variable name of the extracted string.

Start index: The start position of the index.

End index: The end position of the index.

Variable value: The location where the intercepted data is stored.

Usage examples

STRING_SLICE S001 (I001, I001) S001 I001

STRING-SPLIT-String separator splitting

Function

Split the string in the variable with one of the characters in the string variable, and store the splitted characters in the specified variable in order

Parameter description

Variable: The parameter where the search string is located.

Separator: The type of separator.

The first variable in which the data is stored: The first position where the queried data are stored in sequence.

Data storage number: Record the amount of extracted data.

Usage examples

STRING_SPLIT S001 (I001, I001) S001 I001

STRING-LOCATE-String positioning query

Function

Query the position of a character in a string variable, and store the position and quantity in the specified variable in turn

Parameter description

Variable: The parameter where the search string is located.

Variable to be indexed: The character to be searched.

The first variable in which the data is stored: The first position where the queried data are stored in sequence.

Data storage number: Record the amount of extracted data.

Usage examples

STRING_LOCATE S001 S002 I001 0

STRING-LENGTH-String length

Function

Calculate the length of a string in a string variable, and stores the calculated length in the specified variable.

Parameter description

Variable: The variable whose length is to be calculated.

Data storage variable: Record the amount of extracted data.

Usage examples

STRING_LENGTH S001 I001

STRING-TO-String to non-string

Function

Convert a string in a string variable to a non-string

Parameter description

String variable: The string to be translated.

Non-string variable: The target variable of translation.

Usage examples

STRING_TO S001 I001

TO-STRING-Non-string to string

Function

Convert a variable in a non-string variable to a string

Parameter description

Non-string variable: The variable to be translated.

String variable: The target variable of the translation.

Usage examples

TO_STRING I001 S001

Coordinate switch class

SWITCHTOOL-Switch tool hand

Function

Switch the currently used tool hand coordinate system during program running.

Parameter description

Tool coordinate: The tool number of the tool hand coordinate system to switch to.

Usage examples

SWITCHTOOL (3)

SWITCHUSER-Switch user coordinate

Function

Switch the currently used user coordinate system during program running.

Parameter description

User coordinate: The serial number of the user coordinate system to switch to.

Usage examples

SWITCHUSER (3)

USERCOORD_TRANS-User coordinate transformation

Function

Superimpose the B and C user coordinate systems (×), and place the result in the A user coordinate system.

Parameter description

User coordinate A: The result is stored in this user coordinate system, here is the user coordinate system serial number.

User coordinate B: User coordinate system serial number.

User coordinate C: User coordinate system serial number.

Usage examples

USERCOORD_TRANS (1) (2) (3)

SWITCHSYNC-Switch external axis

Function

Switch the currently used external axis during program running.

Parameter description

External axis group number: The group number of the external axis to switch to.

Usage examples

SWITCHSYNC 1

Network communication class

SENDMSG-Send data

Function

Send a string message to another network device.

Parameter description

ID: Process number in the "Settings-Network Settings" interface.

Characters sent: The string to be sent. If you want to send variables, add $ before the variable. If you want to send $ character, two $ are needed. Escape characters and formatted output are supported.

Usage examples

SENDMSG ID = 1 #$D001#

PARSEMSG-Parse data

Function

Parse the data sent by another network device over TCP and store the data in multiple variables.

When a TCP receives multiple values, it will store the values into multiple variables respectively, and the variables used are the first variable, the second variable and so on. That is, if 3 values are sent, i.e. A, B, C, and the first variable set is named GI006, then A will be stored in GI006, B in GI007, and C in GI008.

Parameter description

ID: Process number in the "Settings-Network settings" interface.

The first variable in which the data is stored (First variable type): The type of the first storage variable. Click "More" to select variable type

Clear cache after parsing: Clear the cached data after parsing it

Data storage quantity: Record the number of extracted data by variable

First variable name: The name of the first storage variable.

Usage examples

PARSEMSG ID = 1 GI006CLEARCAHE= 0; Meaning: Store the received data in the variable GI001, and clear the cached data after parsing is completed

READCOMM-Read data

Function

Read the points sent by Ethernet or Modbus, store the points in the position variable, and store the number in the numeric variable.

Parameter description

Process number: The process number of the network communication to open the communication.

Communication method: Use Ethernet communication or Modbus communication.

Position variable type: Global position variable and local position variable can be selected.

Position variable name: The name of the position variable; store the received points, if there are multiple points, the position variables will be extended. For example, the instruction position variable is filled with GP003, and when three points are received, they are stored in GP003, GP004, and GP005 respectively.

Variable type: Global integer and local integer can be selected.

Variable name: Name of variable; store the number of received points.

Note: Currently available only for Modbus.

Usage examples

READCOOMID=1 EHTERNETTOG001I001

OPENMSG-Open data

Function

Open the network communication.

Parameter description

ID: Process number in the "Settings-Network settings" interface.

Usage examples

OPENMSGID= 1

CLOSEMSG-Close data

Function

Close the network communication.

Parameter description

ID: Process number in the "Settings-Network settings" interface.

Usage examples

CLOSEMSG ID = 2

PRINTMSG-Output information

Function

Print the string by means of prompt message.

Parameter description

Output character: The string to be printed. If you want to print variables, add $ before the variable. If you want to print $ character, two $ are needed. Escape characters and formatted output are supported.

Usage examples

PRINTMSG #this is $D001#

MSG_CONNECTION_STATUS-Get information connection status

Function

Get the connection status of a process number in the network settings

Parameter description

Process number: The process number of the network setting that needs to be known

Status storage variable name: Click "More" to select BOOL/GBOOL storage type

Usage examples

MSG_CONN_ST 1 B001

Position variable class

Note: For the newly added variables of the position variable type in the following instructions, please refer to the description of the bind variables in the motion control class section.

USERFRAME_SET-User coordinate modification

Function

Change the value of an axis of the user coordinate system.

Parameter description

User coordinate number: The user coordinate number whose value is to be changed.

User coordinate parameter: The user coordinate axis whose value is to be changed.

Variable type: You can choose hand-filled value or other variable.

Variable name: When selecting other variable, selecting the variable name here will assign the value of that variable to the axis corresponding to the user coordinate.

Hand-filled value: When the variable type selects hand-filled value, directly fill in the target value to be changed here.

Usage examples

USERFRAME_SET ID = 1 UX GI001

USERFRAME_SET ID = 2 UY 99

TOOLFRAME_SET-Tool coordinate modification

Function

hange the value of an axis of the tool coordinate system.

Parameter description

Tool coordinate number: The tool coordinate number whose value is to be changed.

Tool coordinate parameter: The tool coordinate axis whose value is to be changed.

Variable type: You can choose hand-filled value or other variable.

Variable name: When selecting other variable, selecting the variable name here will assign the value of that variable to the axis corresponding to the tool coordinate.

Hand-filled value: When the variable type selects hand-filled value, directly fill in the target value to be changed here.

Usage examples

TOOLFRAME_SET ID = 1 TX GI001; Meaning: Change the X-axis offset parameter of tool hand 1 to the variable value of GI001

TOOLFRAME_SET ID = 2 TY 99; Meaning: Change the X-axis offset parameter of tool hand 2 to 99

READPOS-Read points

Function

Read the value of an axis of a position variable into a float variable.

Parameter description

Variable type: The type of float variable to read, local or global.

Variable name: The variable name of the float variable to read.

Position variable type: The type of position variable to read, current position, local position variable or global position variable.

Position variable name: When the position variable type selects local position variable or global position variable, select the corresponding variable name here. If you choose PINT, PGINT, G$INT or G$GINT, select the corresponding integer variable name here. For example, if you select P$INT (variable name I001, I001=33), then the obtained position variable is P033.

Position variable coordinate system: The coordinate system where the position variable value to be read is located.

Position variable axis: The axis of the position value to be read in the corresponding coordinate system.

Usage examples

READPOS GD004 P$GI003 RF 1

POSADD-Point add

Function

Position variable addition operation (+). This instruction can add the value of a single axis of the position variable (global, local), and then assign it to the axis.

Parameter description

Position variable type: The type of the position variable to be changed, local or global.

Position variable name: The variable name of the position variable to be changed.

Position variable coordinate system: The corresponding coordinate system where the position variable axis to be changed.

Position variable axis: The axis of the position variable to be changed in the corresponding coordinate system.

Variable type: You can choose hand-filled value or other variable.

Numeric variable name: When selecting other variable, selecting the variable name here will add the value of that variable to the value of the corresponding axis of the position variable, and then assign the value to that position variable.

Hand-filled value: When variable type selects hand-filled value, filling in the target value directly here will add that value to the value of the corresponding axis of the position variable and assign it to that position variable.

Usage examples

POSADD P0001 RF 1 788

POSSUB-Point subtract

Function

Position variable subtraction operation (-). This instruction can subtract the value of a single axis of a position variable (global, local), and then assign it to the axis.

Parameter description

Position variable type: The type of the position variable to be changed, local or global.

Position variable name: The variable name of the position variable to be changed.

Position variable coordinate system: The corresponding coordinate system where the position variable axis to be changed.

Position variable axis: The axis of the position variable to be changed in the corresponding coordinate system.

Variable type: You can choose hand-filled value or other variable.

Numeric variable name: When selecting other variable, selecting the variable name here will subtract the value of the variable from the value of the corresponding axis of the position variable, and then assign the value to that position variable.

Hand-filled value: When variable type selects hand-filled value, filling in the target value directly here will subtract that value from the value of the corresponding axis of the position variable and assign it to that position variable.

Usage examples

POSSUB P0001 RF 1 88

POSSET-Point change

Function

This instruction can modify the value of a single axis of a position variable (global, local).

Parameter description

Position variable type: The type of the position variable to be changed, local or global.

Position variable name: The variable name of the position variable to be changed.

Position variable coordinate system: The corresponding coordinate system where the position variable axis to be changed.

Position variable axis: The axis of the position variable to be changed in the corresponding coordinate system.

Variable type: You can choose hand-filled value or other variable.

Numeric variable name: When selecting other variable, selecting the variable name here will assign the value of the corresponding axis of the position variable to that position variable.

Hand-filled value: When variable type selects hand-filled value, filling in the target value directly here will assign the value of the axis corresponding to the position variable to the position variable.

Usage examples

POSSUB P0001 RF 1 88

COPYPOS-Copy point

Function

Copy the values of all axes of a position variable to another position variable.

Parameter description

Source position variable type: The type of the position variable to be read. You can select the current position, i.e. assign the current robot position to another position variable.

Source position variable name: The variable name of the position variable to be read.

Target position variable type: The variable type of the position variable being assigned.

Target position variable name: The variable name of the position variable being assigned.

Usage examples

COPYPOSG003 TOP001

COPYPOSCURPOSTOP002

POSADDALL-Point add all

Function

Position variable addition operation (+). This instruction can perform addition operation on the value of several axes of position variable (global, local), and then assign it to the axis.

Parameter description

Position variable type: The type of the position variable to be changed, local or global.

Position variable name: The variable name of the position variable to be changed.

Position variable coordinate system: The corresponding coordinate system where the position variable axis to be changed.

More: You can choose hand-filled value or other variable.

Numeric variable name: When selecting other variable, selecting the variable name here will add the value of the corresponding axis of the position variable to the value of that variable, and then assign it to the position variable.

Hand-filled value: When variable type selects hand-filled value, filling in the target value directly here will add that value to the value of the corresponding axis of the position variable and assign it to that position variable.

Usage examples

POSADDALL GP0001 RF I001 GI001 D001 GD001 10.1 10

POSSUBALL-Point subtract all

Function

Position variable subtraction operation (-). This instruction can perform subtraction operation on the value of several axes of position variable (global, local), and then assign it to the axis.

Parameter description

Position variable type: The type of the position variable to be changed, local or global.

Position variable name: The variable name of the position variable to be changed.

Position variable coordinate system: The corresponding coordinate system where the position variable axis to be changed.

More: You can choose hand-filled value or other variable.

Numeric variable name: When selecting other variable, selecting the variable name here will subtract the value of the variable from the value of the corresponding axis of the position variable, and then assign the value to that position variable.

Hand-filled value: When variable type selects hand-filled value, filling in the target value directly here will subtract that value from the value of the corresponding axis of the position variable and assign it to that position variable.

Usage examples

POSSUBALL GP0001 RF I001 GI001 D001 GD001 10.1 10

POSSETALL-Point change all

Function

This instruction can modify the values of several axes of position variable (global, local).

Parameter description

Position variable type: The type of the position variable to be changed, local or global.

Position variable name: The variable name of the position variable to be changed.

Position variable coordinate system: The corresponding coordinate system where the position variable axis to be changed.

More: You can choose hand-filled value or other variable.

Numeric variable name: When selecting other variable, selecting the variable name here will assign the value of the corresponding axis of the position variable to that position variable.

Hand-filled value: When variable type selects hand-filled value, filling in the target value directly here will assign the value of the axis corresponding to the position variable to the position variable.

Usage examples

POSSETALL GP0001 RF I001 GI001 D001 GD001 10.1 10

TOFFSETON-Trajectory offset start

Function

This instruction provides a real-time offset of the robot's trajectory.

Parameter description

Offset coordinate system: The coordinate system corresponding to the trajectory to be changed.

Offset type: You can choose hand-filled value or other variable type.

Offset: When the variable type is hand-filled value, filling in the target value directly here will add the robot's trajectory coordinates to this hand-filled value.

More: You can choose hand-filled value or other variable.

Numeric variable name: When selecting other variable, selecting the variable name here will assign the value of the corresponding axis of the position variable to that position variable.

Hand-filled value: When variable type selects hand-filled value, filling in the target value directly here will assign the value of the axis corresponding to the position variable to the position variable.

Usage examples

TOFFSETON RF GI001 I002 2 3 4 5

TOFFSETOFF-Trajectory offset end

Function

The trajectory offset ends, and the subsequent motion trajectories will no longer offset.

Usage examples

TOFFSETOFF

READPOSMSG-Read point information

Function

Read the value of point tool number, user coordinate number, coordinate system, attitude angle/radian and form information into an integer variable.

Parameter description

Variable type: You can choose between global position variable and local position variable.

Variable name: The name of the position variable.

Information: Tool number/user coordinate number/coordinate system/angle/radian/form.

More (Target variable type): The variable type of the position variable being read.

Target variable name: The variable name of the position variable being read.

Usage examples

READPOSMSG P0001 TOOL I001

POS_STRETCH-Point stretch

Function

Shorten or lengthen the length of the line and the ends of the arc, and change the middle point of the arc to change the trajectory of the arc.

Parameter description

Stretch type: Support line or arc instruction stretch.

Start point: The start point of a line or arc instruction.

Arc midpoint: The midpoint of the arc instruction.

End point: The end point of a line or arc instruction.

Start point offset: The distance by which the start point is shortened or stretched.

End point offset: The distance by which the end point is shortened or stretched.

Output start point position: Save the stretched start point position in the local point position or global point position.

Output end point position: Save the stretched end point position in the local point position or global point position.

Usage examples

POS_STRETCH LINE P0001 P0002 10 10 P0004 P0005

SETPOSMSG-Set point information

Function

Set the coordinate system, angle/radius, form, tool number, and user coordinate number of the point

Parameter description

Variable type: You can choose between global position variable and local position variable.

Coordinate system: Set the coordinate system number by local integer variable, global integer variable and "unchanged" option.

Angle/radian: Set angle/radian by local integer variable, global integer variable and "unchanged" option.

Form: Set the form by local integer variable, global integer variable and "unchanged" option.

Tool number: Set the tool number by local integer variable, global integer variable and "unchanged" option.

User coordinate number: Set user coordinate number by local integer variable, global integer variable or "unchanged" option.

Usage examples

SETPOSMSG P0001 1 1 1 1 1 1

Program control class

PTHREAD_START-Start thread

Function

Start the background task. The background task ends when executed once. To edit the background task, please go to "Settings - Background Tasks" interface for programming. Local background tasks will synchronize the stop and run of the main program, global background tasks will not do this.

Parameter description

Type: Select local background or global background

Background task: The name of the background task

Usage examples

PTHREAD_START [TTT]

PTHREAD_END-Exit thread

Function

Close the started background tasks.

Parameter description

Type: Select local background or global background

Background task: The name of the background task

Usage examples

PTHREAD_END [TTT]

PAUSERUN-Pause running

Function

Pause the program.

Parameter description

Type: The type of program to be paused, including all, main program, background program.

Program: Name of the program to be paused.

Usage examples

PAUSERUN [TTT]

PAUSERUN MAIN

PAUSERUN ALL

CONTINUERUN-Continue running

Function

Continue running the paused program (the stopped program cannot be continued).

Parameter description

Type: The type of program to continue running, including main program, local background program.

Program: The name of the program to continue running.

Usage examples

CONTINUERUN [TTT]

CONTINUERUN MAIN

STOPRUN-Stop running

Function

Stop all programs.

Parameter description

None

Usage examples

STOPRUN

RESTARTRUN-Rerun

Function

Rerun the stopped program.

Parameter description

None

Usage examples

RESTARTRUN

WINDOW-Popup instruction

Function

Pop up the filled content prompt window, the number of displayed buttons is the number of options, and save the value of the clicked button (option) into a local integer variable.

Parameter description

Prompt: The content displayed in the pop-up window.

Bind variable: Local integer variables

Number of options: 1-3 buttons

Option 1 content: Button 1 content

Option 1 value: Button 1 value

Option 2 content: Button 2 content

Option 2 value: Button 2 value

Option 3 content: Button 3 content

Option 3 value: Button 3 value

Usage examples

WINDOW #Prompt# I001 3 #Button 1# 1 #Button 2# 2 #Button 3# 3

PTHREAD_STATE-Thread state

Function

Insert thread instruction to view the status of the currently executed thread program, stop equals 1, pause equals 2, run equals 3

Parameter description

Type: You can choose local background, global background or main program

Background task: The name of the background task

Storage variable type: For example, the selected variable name is GI001, after opening the thread, the value of GI001 will change with the state. Stop: GI001=1; Pause: GI001=2; Run: GI001=3

Usage examples

PTHREAD_STATE[program file name] GINT GI001=0

Conveyor Tracking Process

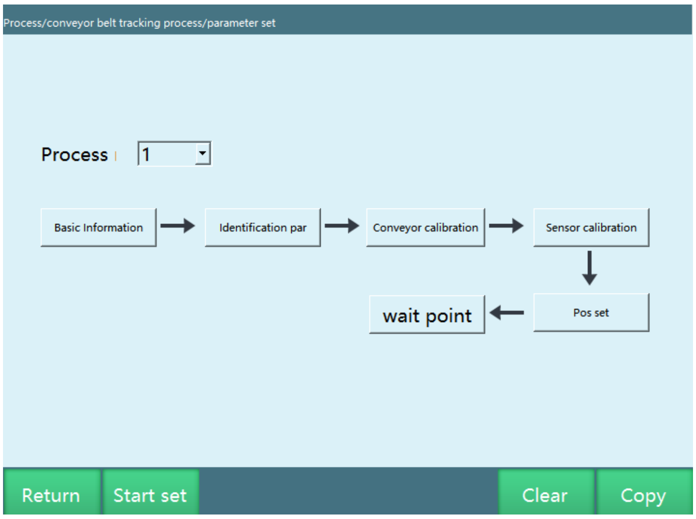
<<<

Catalogue

Conveyor Tracking Process

Click [Process], select [Conveyor tracking process], and click [Parameter settings] to enter the conveyor tracking process parameter setting interface.

Conveyor tracking means that the robot uses the material point position entered by the user and the corresponding encoder value when the material is in this position, to calculate the material point position in real time and track the material through movement.

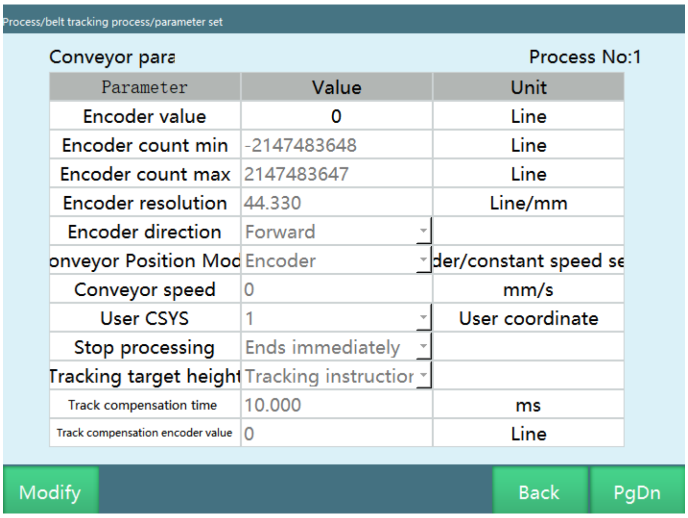


Clear parameters: Clear all parameters of this process number

Copy parameters: Copy all parameters of this process number to another process number

Basic information

Before setting all parameters, please select a process number in the "Parameter settings" interface, each process number holds all parameters. The basic information is the basic setting for the parameters of the conveyor belt.



The "Basic information" interface contains the following parameters:

Encoder value

After the encoder is successfully connected, the robot will automatically identify the readings of the currently connected encoder, which are read-only;

If the encoder is not connected successfully, there will be no encoder value. Generally, there are two cases: 1: The wiring of the encoder port is incorrect; 2: The encoder is connected to the wrong port on the IO board.

1: The position of the connection port on the IO board can refer to the definition diagram of the corresponding IO board

2: The wiring method of the encoder port can refer to the definition diagram of the corresponding encoder

Encoder count max/Encoder count min

The maximum value that can be counted by the encoder data processing module is based on the IO board used as the encoder data processing module, there are currently two value ranges: (0,6000) or (-2^31, 2^31-1)

Encoder resolution

The unit pulse emitted by the encoder when the conveyor belt moves 1mm; this value is the calibration result of the encoder resolution

Encoder direction

Start the conveyor belt and observe whether the offset and speed increase as the belt moves. If there is no change, then the encoder type or resolution setting does not match the actual situation. If the offset and speed decrease with the movement of the conveyor belt, then check [Reverse] here

Conveyor position mode

Select "Encoder": normal sensor calibration

Select "Constant speed setting": when "Constant speed setting" is selected, there is nothing to do with the encoder, and the conveyor speed can be set manually. (Note: After manually modifying the speed, you need to re-calibrate the sensor)

Note: When setting constant speed, there is error in sensor position calibration calculation. Error factor: The movement time interval of the conveyor belt calculated at the time of calibration is too large.

[Solution]: Stop the robot tool hand on the follow-up path of the workpiece, and calibrate the workpiece directly when it passes the tool hand, which can reduce the error.

Conveyor speed

Current conveyor belt speed, read-only

User coordinate system

The user coordinate system can be calibrated according to the actual movement direction of the conveyor belt, and the motion tracking and calculation are carried out under this user coordinate system

Conveyor stop processing

Robot stops immediately: When the conveyor belt stops unexpectedly during the tracking process, the robot will stop this tracking and return to the safety point to wait for the next tracking signal, with a waiting timeout of 2min. Robot continues running: When the conveyor belt stops unexpectedly during the tracking process, the robot will not stop running but will continue to complete the previously planned trajectory.

Tracking target height

Sensor sensing: Determine the maximum height of the target workpiece according to the height captured by the vision and the height triggered by the sensor

Tracking instruction teach: the start height when teaching the trajectory is the tracking height

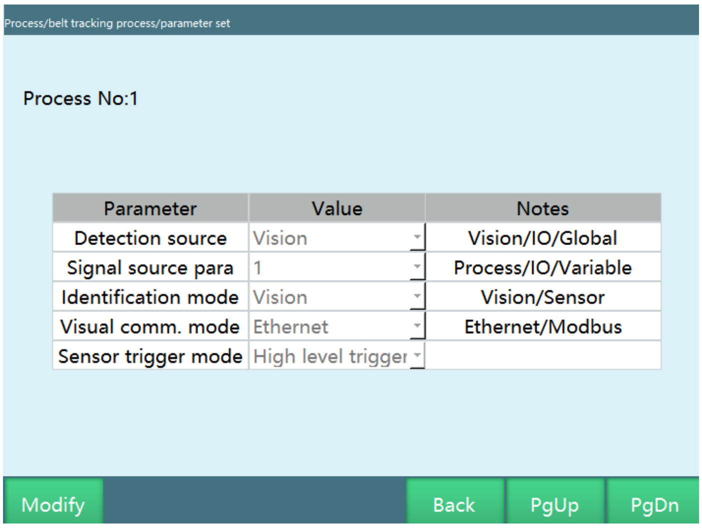
Tracking compensation time

Used to solve the tracking lag problem; calculated from time and conveyor belt speed. The tracking lag is mainly caused by the filtering of the encoder data and the execution of the planned motion of the robot.

Tracking compensation encoder value

Used to solve the tracking lag problem; calculated from encoder value and resolution

Parameter identification



Parameter	Value	Notes
Detection source	Vision	Vision/IO/Global
Signal source para	1	Process/IO/Variable
Identification mode	Vision	Vision/Sensor
Visual comm. mode	Ethernet	Ethernet/Modbus
Sensor trigger mode	High level trigger	

Workpiece detection signal source

Workpieces on conveyor can be detected by three methods: vision, IO, and global variables

Signal source parameters

If the workpieces on conveyor are detected by IO, the signal source parameter can select the IO port number;

If the workpieces on conveyor are detected by vision, the signal source parameter can select the corresponding visual process number;

If the workpieces on conveyor are detected through the global variables, the signal source parameter can select the global Boolean variable.

Workpiece identification method

The workpieces on conveyor can be identified through vision and sensors; when selecting sensors, the visual communication method is not required

Visual communication method

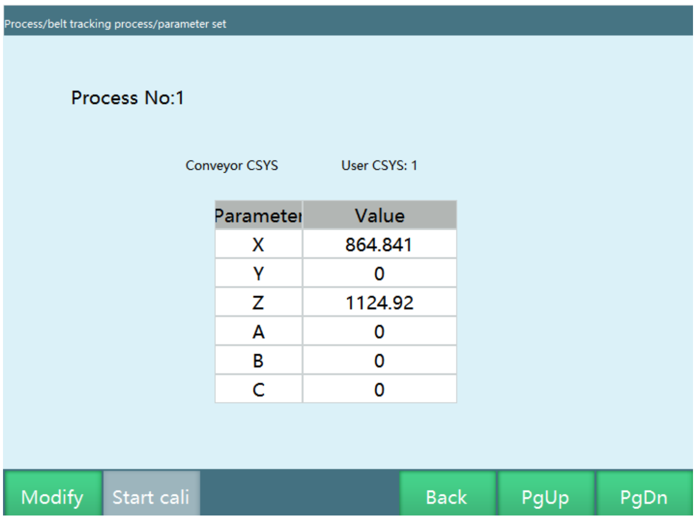
If the workpieces on conveyor are identified by vision, two communication methods can be selected: Ethernet and Modbus

Sensor trigger method

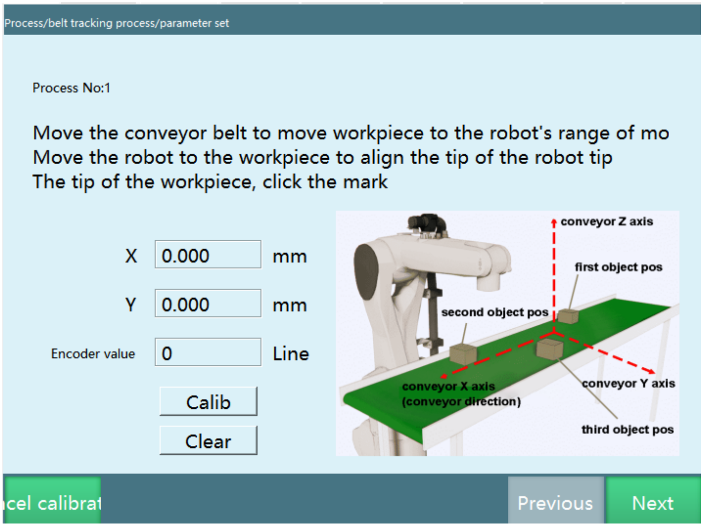
Only when the workpiece detection signal source is set to digital IO, the signal source parameters select the corresponding IO port, and the workpiece identification method is sensor, then the sensor trigger method can take effect. The trigger method is divided into two types: high level trigger (triggering when io signal is 1), low level trigger (triggering when io signal is 0)

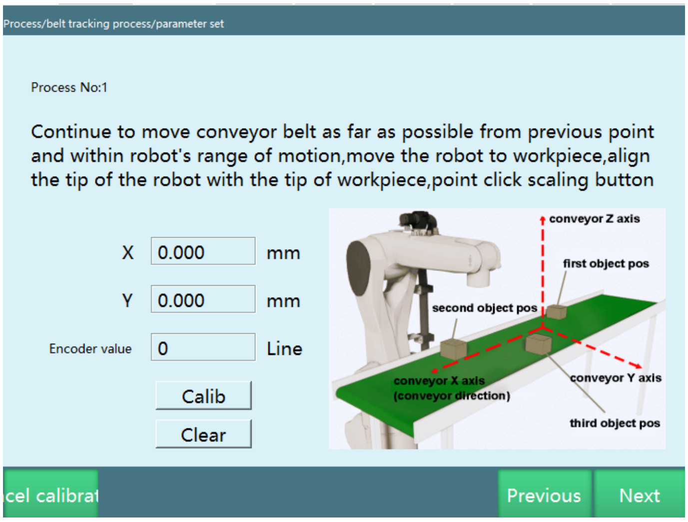
Conveyor calibration

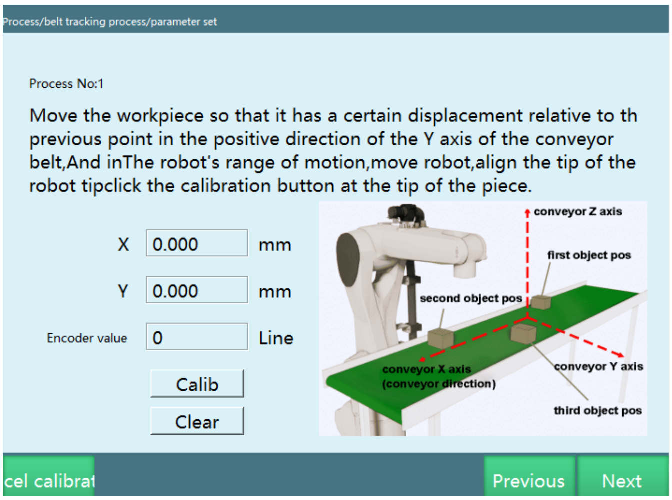
The user coordinate system is selected in the "Basic information", and it needs to be calibrated by the user in advance



Conveyor belt coordinate system calibration: calibrate 3 points, calculate the user coordinate system of the conveyor belt; click [Modify], and then click [Start calibration] button to enter the calibration interface





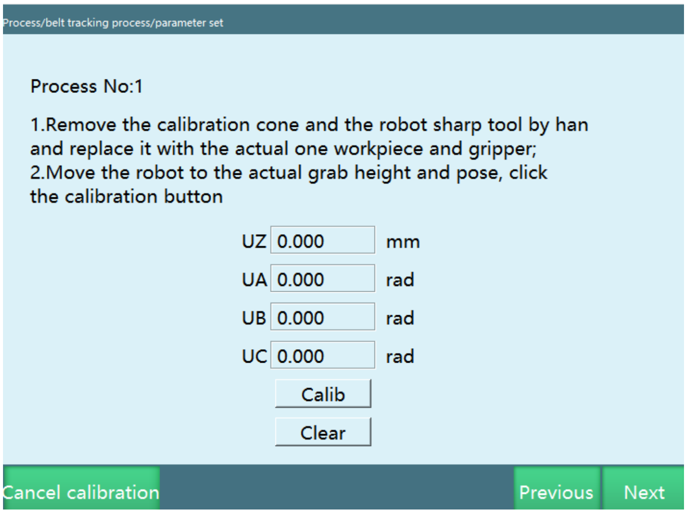


Step 1: Place a pointed calibration cone on the conveyor belt, move the conveyor belt so that the calibration cone on the conveyor belt moves within the motion

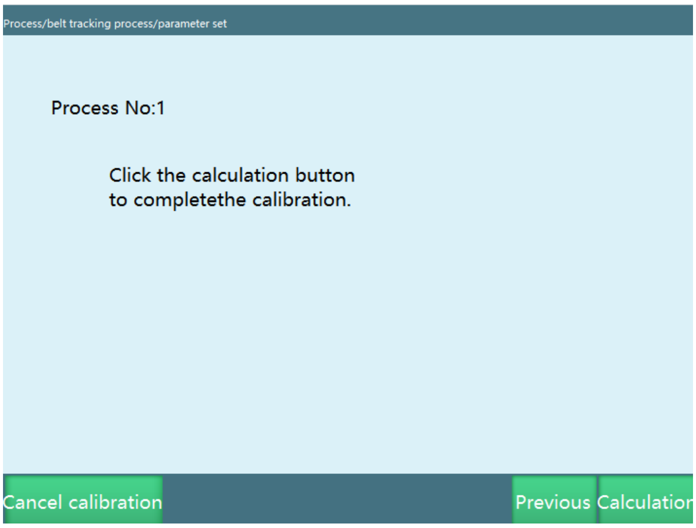
range of the robot, move the robot to the workpiece so that the robot tool end tip is aligned with the tip of the calibration cone, click [Calibrate].

Step 2: Teach the robot to raise slightly , continue to move the conveyor belt so that the calibration cone on the conveyor belt is as far away as possible from the previous point and within the robot's motion range, move the robot to the calibration cone so that the robot tool end tip is aligned with the tip of the calibration cone, click the [Calibrate] button.

Step 3: Move the calibration cone so that it has a certain displacement in the positive direction of the Y-axis of the conveyor belt relative to the previous point, and within the motion range of the robot, move the robot so that the robot tool end tip is aligned with the tip of the calibration cone, click the [Calibrate] button.

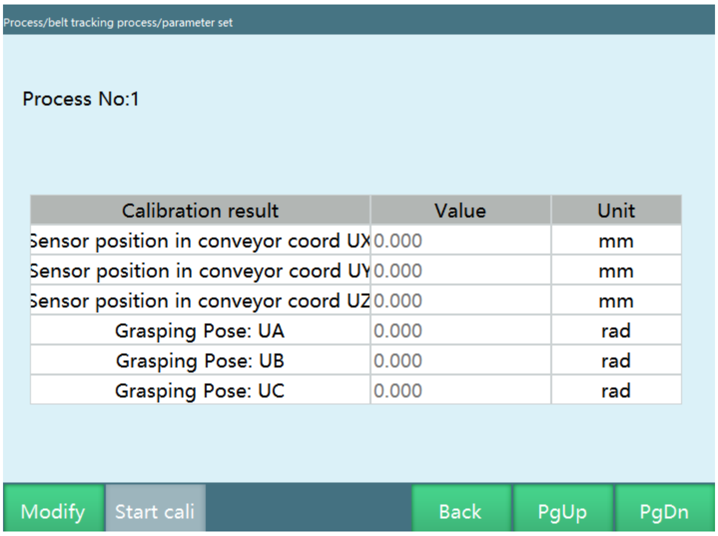


Step 4: Raise the robot for a certain distance, and click the [Calculate] button to complete the calibration.



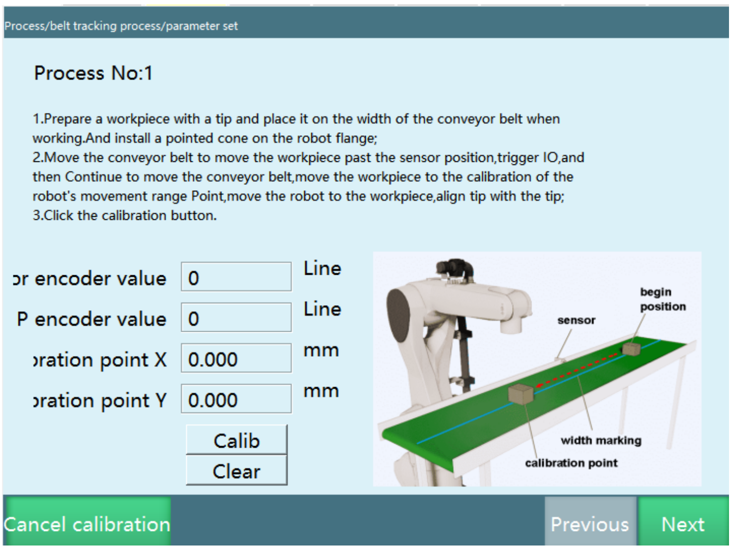
Note: The direction of the selected user coordinate system should be consistent with the calibration direction of the conveyor belt

Sensor calibration



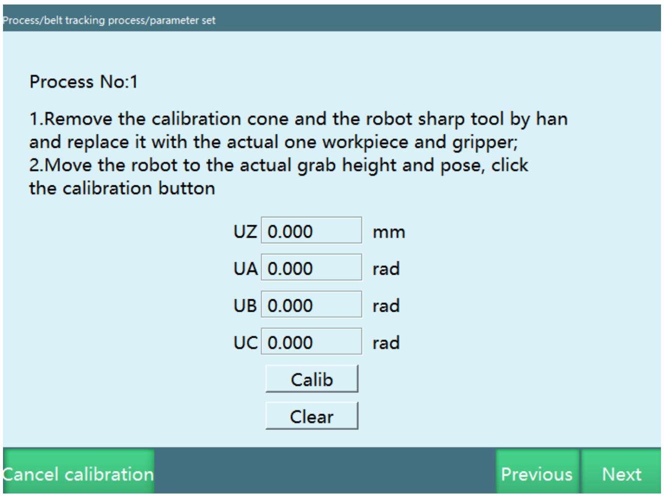
If you use the sensor to identify the workpiece, you need to calibrate the sensor on this interface; click [Modify], and then click [Start Calibration] button to enter the calibration interface.

Note: If you use vision to identify the workpiece, then calibration is not required, just skip it.

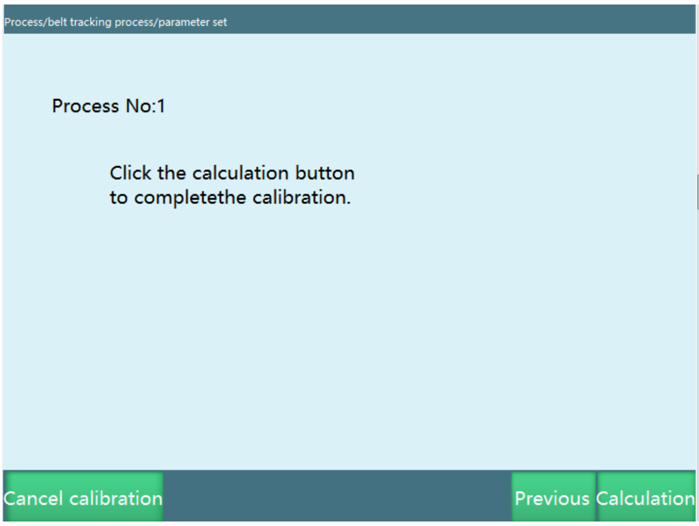


Step 1: Prepare a workpiece with a tip, place it at the width of the conveyor belt at work, and install a pointed cone on the robot flange; move the conveyor belt so that the workpiece moves past the sensor position, trigger IO, then continue

to move the conveyor belt to move the workpiece to the calibration point within the robot's range of motion, stop the conveyor belt, move the robot to the workpiece so that the tip aligns with the tip; click the [Calibrate] button.

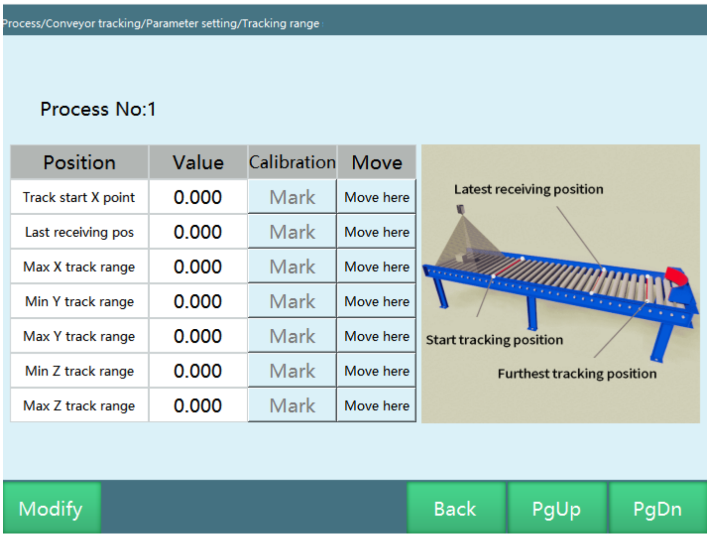


Step 2: Remove the calibration cone and the robot pointed tool hand and replace it with the actual workpiece and gripper; run the robot to the actual grasping height and attitude, and click the [Calibrate] button.



Step 3: Click the [Calculate] button, the calibration parameters are stored.

> Tracking range setting



This interface is used to set some key positions and tracking range of the robot during the tracking process

Tracking start X point

This parameter only records the value of the X-axis of the conveyor belt coordinate (the running direction of the conveyor belt), and the robot will track only when the workpiece exceeds this position during each tracking.

When the robot is in the previous tracking process, and the next workpiece has exceeded the tracking start X point, the robot will directly perform the tracking process for the workpiece after completing the previous tracking process.

If the robot does not perform the tracking process at this time, and the workpiece has not reached the position of the tracking start X point, the robot will wait at this position.

Tracking range X max

The maximum position of the tracking range on the X-axis of the conveyor belt (the running direction of the conveyor belt); the robot abandons tracking as soon as the position is exceeded, regardless of whether the workpiece is being tracked or not.

Tracking range Y min

The minimum position of the tracking range on the Y-axis of the conveyor belt (perpendicular to the running direction of the conveyor belt); if the workpiece does not reach this position, the robot does not track.

Tracking range Y max

The maximum position of the tracking range on the Y-axis of the conveyor belt (perpendicular to the running direction of the conveyor belt); if the workpiece exceeds this position, the robot does not track.

Tracking range Z min

The minimum height of the robot during tracking.

Tracking range Z max

The maximum height of the robot during tracking.

Latest receiving position

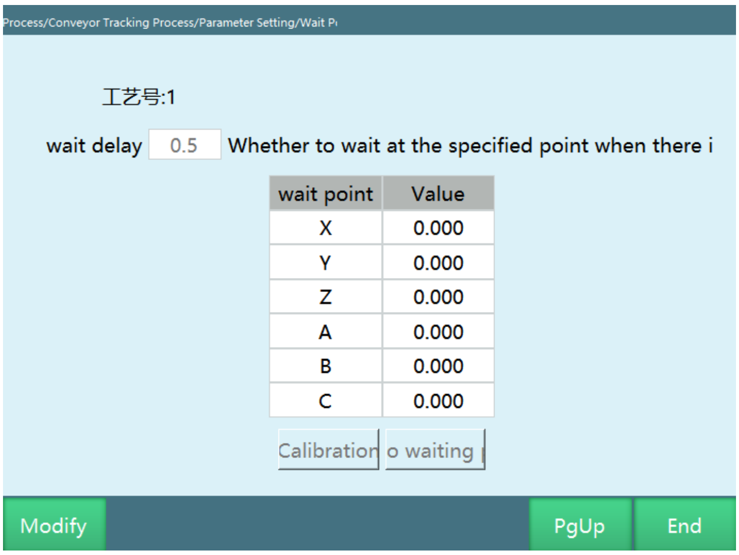
The latest receiving position of the workpiece on the conveyor X-axis (the running direction of the conveyor belt).

If the workpiece exceeds this position before being tracked, the robot will not track the workpiece.

Note: If you find that the calibration range is not reasonable, please reconfirm the calibration of the user coordinate system and check whether the direction of the selected user coordinate system is reasonable.

Waiting point

During the tracking process, the robot will stay at the waiting point when there is no workpiece, and wait until the signal of workpiece is detected, then continue to track. If there is a workpiece, the robot will continue to track without going to the waiting point



Waiting delay: determine whether there is a workpiece within 0.5s, and continue to track if there is a workpiece. If there is no workpiece, the robot will go to the waiting point

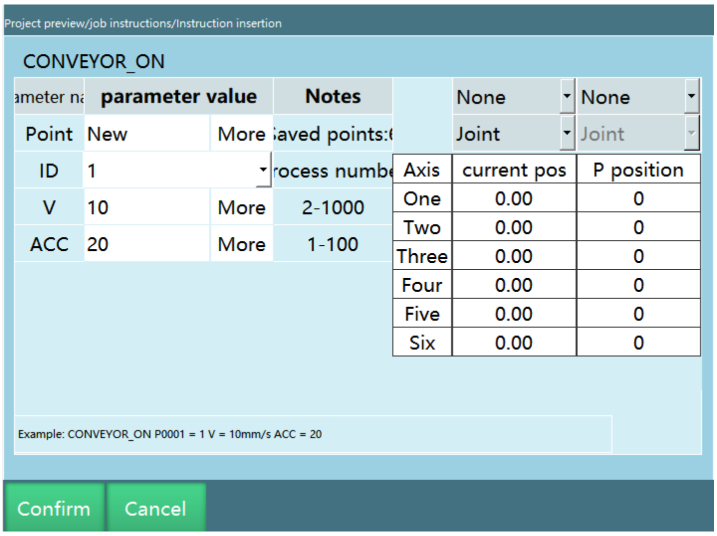
Calibrate this point: no matter what coordinate system the points are marked in, the saved points are still the points in the user coordinate system

Run to waiting point: run to the marked waiting point

Conveyor instructions

CONVEYOR_ON instruction

Conveyor tracking start instruction, which is used in combination with the CONVEYOR_OFF instruction

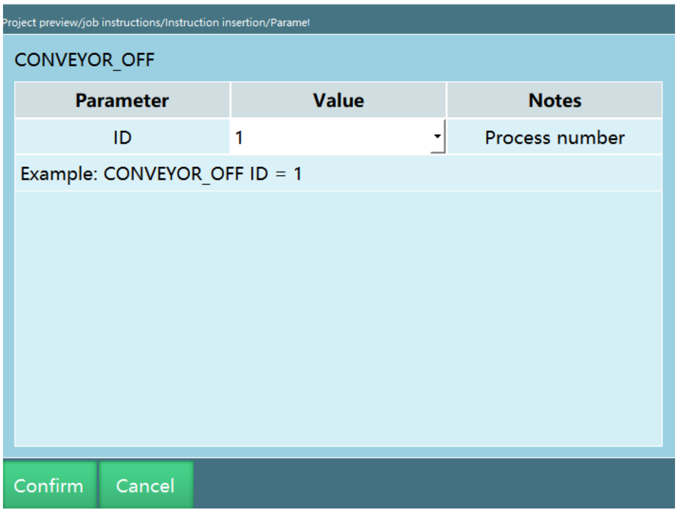


- Reference point position data
- You can select an existing position variable or create a new one.This point is the reference point in the conveyor tracking process and also determines the tracking height. It is recommended to set this point to the middle point of the workpiece to be tracked, or the first point of the trajectory if a trajectory needs to be taken on the workpiece.
- P point, GP point, workpiece point can be selected
- ID
- The process number of the conveyor tracking process.

- V
- Maximum speed during conveyor tracking, range 1-9999.
- ACC
- Acceleration during conveyor tracking, range 1-100.

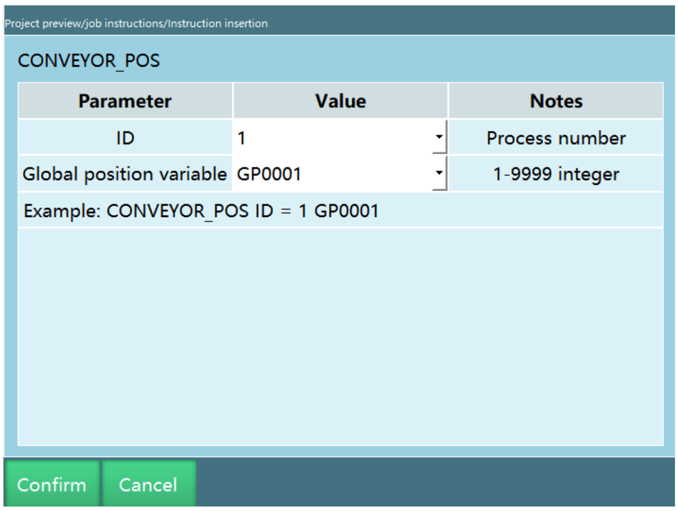
CONVEYOR_OFF instruction

Conveyor tracking end instruction



CONVEYOR_POS instruction

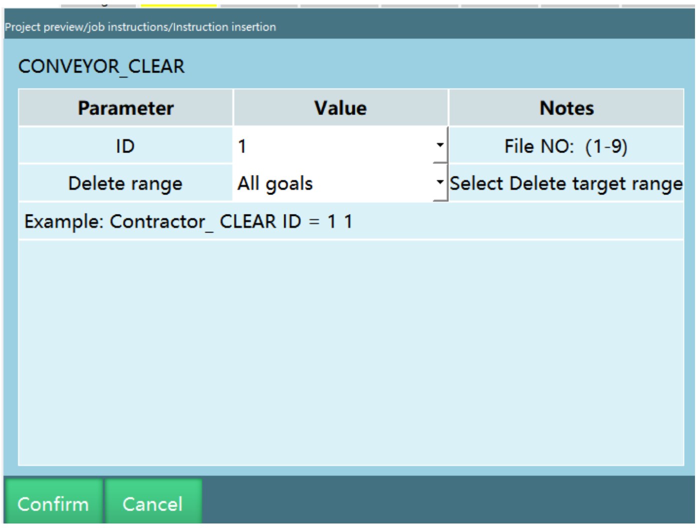
Instruction to get conveyor tracking position



When this instruction is executed, the sensor calibration result of the conveyor belt process number 1 is stored in the global point GP001.

CONVEYOR_REMOVE instruction

Delete conveyor tracking target



Deletion range: all targets

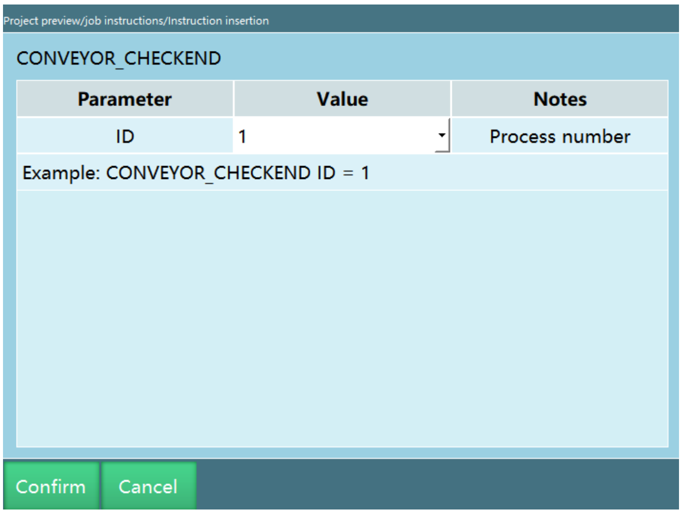
When the running program gives the conveyor tracking start signal multiple times, all but the first signal are deleted

Deletion range: this target

When the running program gives the conveyor tracking start signal multiple times, the first signal will be deleted during each loop

CONVEYOR_CHECKEND instruction

Conveyor workpiece detection end instruction



CONVEYOR_CHECKPOS instruction

Conveyor workpiece detection start instruction, which is used in combination with the CONVEYOR_CHECKEND instruction

Run the instruction and wait for the conveyor workpiece detection signal

Programming

Use the sensor, MOVJ to walk the track

```
NOP
MOVJ P001 VJ = 20 %   PL = 0 ACC = 20 DEC = 20          Move to starting safe position
CONVEYOR_CHECKPOS ID = 1                                Start detecting external data
TIMER T = 1                                             Delay 1s
WHILE (GI001 == 1)                                      Inner loop, cycle tracking
CONVEYOR_ON G001 ID = 1 V = 100 mm/s   ACC = 20         Start conveyor tracking
TIMER T = 1                                             Stay over the workpiece for 1s
MOVJ G002 VJ = 20 %   PL = 0 ACC = 20 DEC = 20          Walk the track on workpiece
MOVJ G003 VJ = 20 %   PL = 0 ACC = 20 DEC = 20          Walk the track on workpiece
MOVJ G004 VJ = 20 %   PL = 0 ACC = 20 DEC = 20          Walk the track on workpiece
CONVEYOR_OFF ID = 1                                     End tracking
ENDWHILE                                                Cycle tracking
CONVEYOR_CHECKEND ID = 1                                Stop detecting data
END
```

Use the sensor, external point function to walk the track

When using this function, simply select "External point" at the position where P-point and G-point are selected when inserting the CONVEYOR_ON instruction, and insert the MOVCOMM instruction under CONVEYOR_ON.

```
NOP
MOVJ P001 VJ = 20 %   PL = 0 ACC = 20 DEC = 20          Move to starting safe position
CONVEYOR_CHECKPOS ID = 1                                Start detecting external data
TIMER T = 1                                             Delay 1s
WHILE (GI001 == 1)                                      Inner loop, cycle tracking
CONVEYOR_ON OUTP ID = 1 V = 100 mm/s   ACC = 20         Start conveyor tracking
TIMER T = 1                                             Stay over the workpiece for 1s
MOVCOMM                                                 Use external point function to walk the track
CONVEYOR_OFF ID = 1                                     End tracking
ENDWHILE                                                Cycle tracking
CONVEYOR_CHECKEND ID = 1                                Stop detecting data
END
```

Vision conveyor tracking

When using this function, the workpiece is tracked by vision which is selected for workpiece detection signal source.

NOP	Start
INT I001 = 0	Define variable
MOVJ P008 VJ = 60 % PL = 0 ACC = 60 DEC = 60	Safety point
VISION_RUN ID = 1	Vision process 1 open
CONVEYOR_CHECKPOS ID = 1	Conveyor workpiece detection start
VISION_TRG ID = 1	Vision trigger
WHILE (I001 == 0)	Cycle grab
CONVEYOR_ON P005 ID = 1 V = 500 mm/s ACC = 50	Conveyor tracking start (trajectory point 1)
MOVL P003 V = 500 mm/s PL = 0 ACC = 50 DEC = 50	Trajectory point 2
MOVL P005 V = 500 mm/s PL = 0 ACC = 50 DEC = 50	Trajectory point 3
MOVL P004 V = 500 mm/s PL = 0 ACC = 50 DEC = 50	Trajectory point 4
MOVL P006 V = 500 mm/s PL = 0 ACC = 50 DEC = 50	Trajectory point 5
MOVL P007 V = 500 mm/s PL = 0 ACC = 50 DEC = 50	Trajectory point 6
MOVL P003 V = 500 mm/s PL = 0 ACC = 50 DEC = 50	Trajectory point 7
CONVEYOR_OFF ID = 1	Conveyor tracking end
ENDWHILE	Loop end
CONVEYOR_CHECKEND ID = 1	Conveyor workpiece detection end
VISION_END ID = 1	Vision end
END	Program end

Human-Robot Collaboration

<<<

Catalogue

Human-Robot Collaboration

This chapter mainly introduces the role of dynamics and how to use it.

Because of the complex nonlinearity, time-varying uncertainty, and strong coupling of the robot (especially during high speed motion), to enable the robot to move at the desired speed and acceleration, the servo motor of each joint of the robot must have sufficient force and torque to drive the linkages and joints of the robot, otherwise the linkages will affect the robot's positioning and trajectory tracking accuracy due to slow movement, so it is necessary to establish feedforward torque control based on a dynamics model, so that the feedforward compensation torque can be quickly calculated in real time.

Human-Robot Collaboration (HRC) refers to work scenarios in which humans and automated robots share a workspace and work simultaneously.

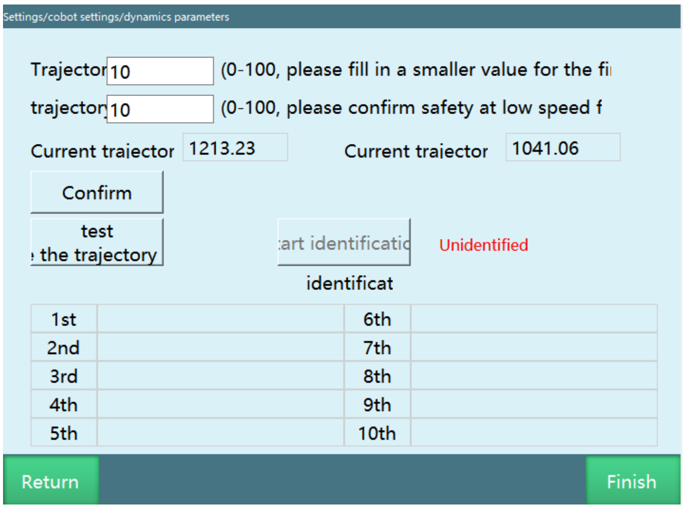
Dynamics parameters

Before using the mechanical functions, you first need to set up the dynamics parameters so that the controller builds a dynamics model of the robot.

Enter **"Settings/Human-robot collaboration/Dynamics parameters"** to set the dynamics parameters.

Identification

Before entering the identification interface, you need to read the notes related to identification carefully. When the robot performs identification, it is best to increase the range and speed from small to large. If there are external factors that prevent the robot from reaching the trajectory range of 100, you can properly adjust the positive and negative limits in the joint parameters. By running a test to make sure that there are no obstacles around the robot and that it can run at a speed of 100, you can start identifying. During the identification process, it is best not to operate the teach pendant, and people need to stay away from the robot. If you need to pause the operation, you can stop the robot by clicking on the stop button on the teach pendant or by pressing the emergency stop button.



Parameter description

Trajectory range: The maximum and minimum motion range of the robot is calculated based on the trajectory range.

Trajectory speed: The speed of the robot while it is running, independent of the global speed.

Current trajectory Z maximum value/Current trajectory Z minimum value: The range of the current trajectory in Z-axis direction.

Identification error: After identification, six parameters will appear representing the error of each of the six axes (The smaller the value, the smaller the error, but the value cannot be zero).

Read before use

Warnings

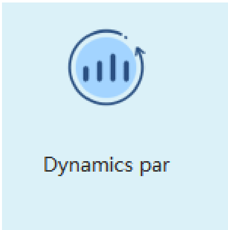
- At present, this identification method is only applicable to the identification of the robot body dynamics parameters of a 6-axis robot under no load.
- The dynamics parameters identified by this identification method are

independent of the hand-filled dynamics parameters.

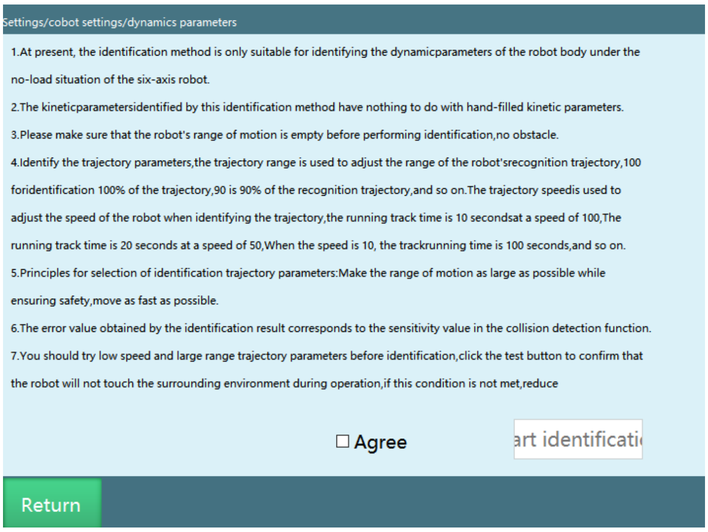
- Make sure that the robot's range of motion is clear and free of obstacles before performing identification.
- In the trajectory identification parameters, the trajectory range is used to adjust the trajectory identification range of the robot, 100 is to identify 100% of the trajectory, 90 is to identify 90% of the trajectory, and so on.The trajectory speed is used to adjust the speed of the robot when it performs the trajectory identification. The trajectory running time is 10 seconds when the speed is 100, 20 seconds when the speed is 50, 100 seconds when the speed is 10, and so on.
- The selection principle of trajectory identification parameters: Make the range of motion as large as possible and the speed as fast as possible while ensuring safety.
- The error value obtained from the identification result corresponds to the sensitivity value in the collision detection function.
- Before identification, you should try the low speed and wide range of trajectory parameters, and click the "Test" button to confirm that the robot will not touch the surrounding environment during operation. If the condition is not met, reduce the trajectory range parameter, run at low speed again to ensure that it will not touch the surrounding environment, and then set the trajectory speed to 100, and click the "Start identification" button to perform robot parameter identification.
- When testing trajectory safety, the robot will run two trajectories. Please do not approach the robot until the test is completed.
- The robot will run two trajectories when identifying the trajectory, please run 10 times, you must not approach the robot during this time, because the robot may start at any time.
- The identification work is executed ten times, including the process of running trajectory, obtaining data, analyzing data, calculating dynamics parameters, etc. The error value is displayed on the interface after each execution, and the whole process will last about 30 minutes, please do not do any operation during this period to avoid affecting the identification work.

Operation steps

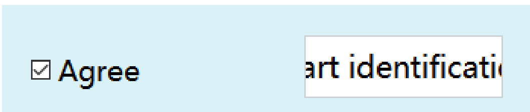
1. Adjust the robot joint parameters-joint limits to ensure that all robot movements are within safe limits and that all of the following trajectories will move within the limits.
2. **Move the robot to the zero position.**
3. Click [Settings-Human-robot collaboration-Dynamics parameters] to enter the "Dynamics parameters" interface.



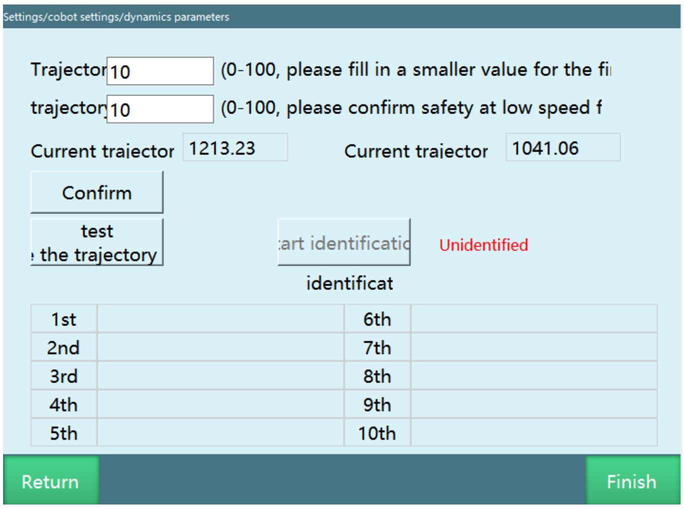
4. Read the instructions carefully.



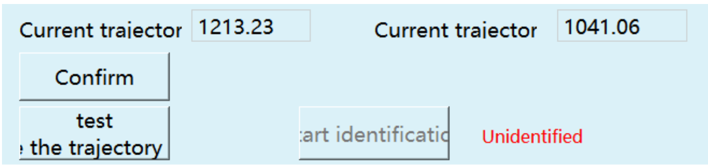
5. After reading all the instructions, click "Read and agree" and click "Start identification".



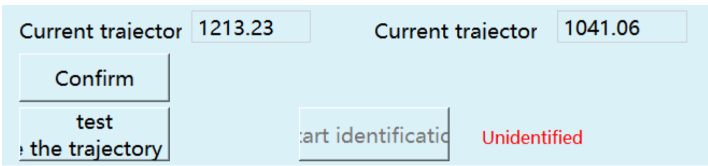
6. After entering the operation interface of identification, fill in 10 for "Trajectory range" and 10 for "Trajectory speed".



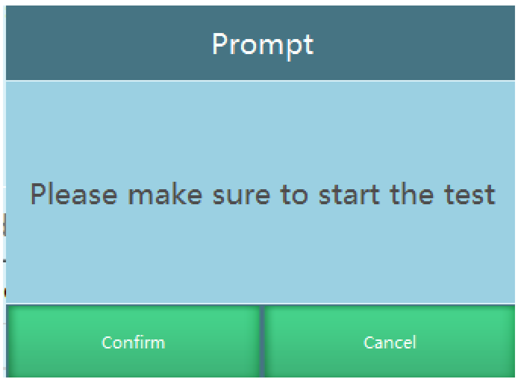
7. Click "OK", check the current trajectory Z maximum value and current trajectory Z minimum value, check whether the range is reasonable, confirm that the trajectory can be reached before entering the next step.



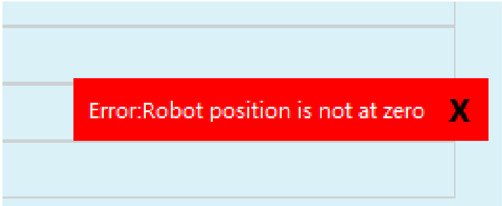
8. Click "Test (make sure the trajectory is safe)".



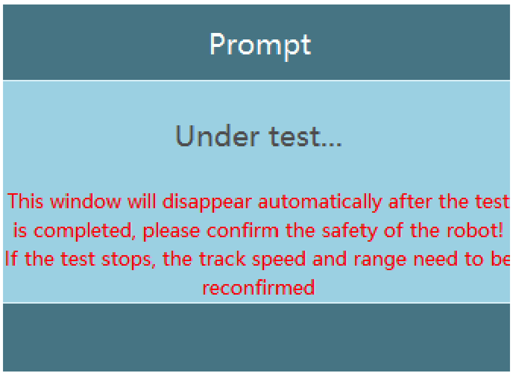
9. A test prompt window will pop up, click "OK".



10. If an error is reported, please follow the prompt to return to zero point first.

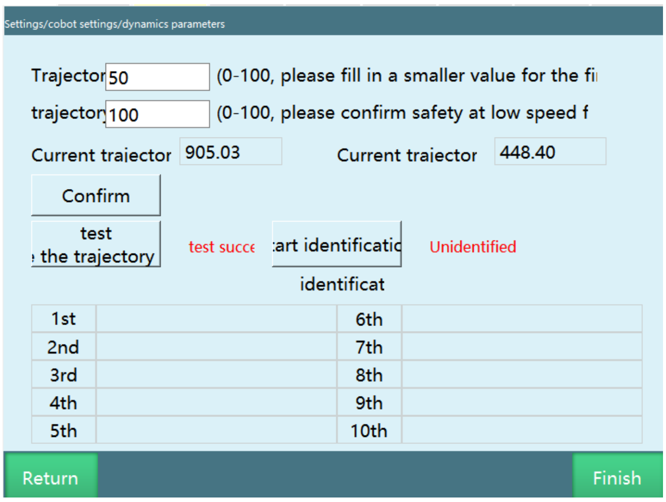


11. If no error is reported, a pop-up will indicate "Testing in progress..."

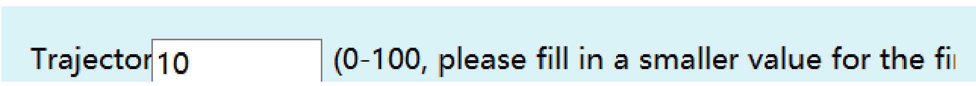


12. During robot movement, you can stop the robot by pressing "Stop" button, switching modes, or pressing the emergency stop button at the top right of the teach pendant.

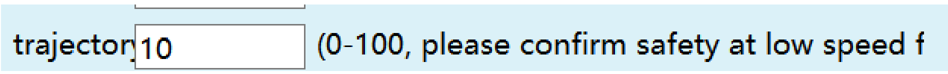
13. When the test is completed, the interface will prompt "Test successful".



14. If the trajectory range is small, the trajectory range can be enlarged. In principle, the larger the trajectory range, the higher the accuracy of identification.

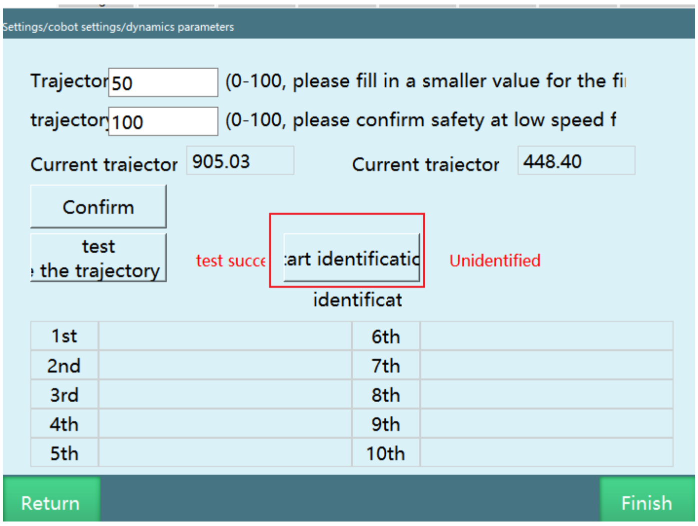


15. The trajectory speed can be slow when testing, but the trajectory speed must be set to 100 when identifying.

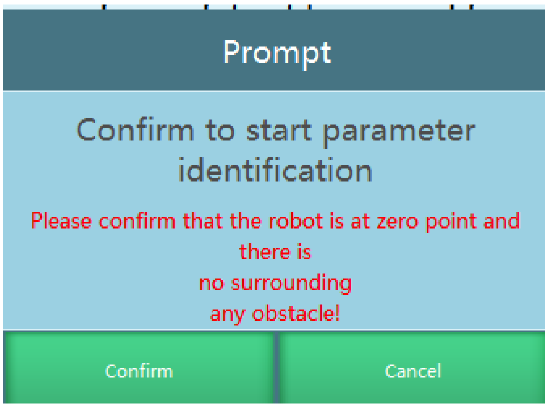


16. Maximize the trajectory range on the basis of ensuring safety, once the speed is set to 100, you can start the identification.

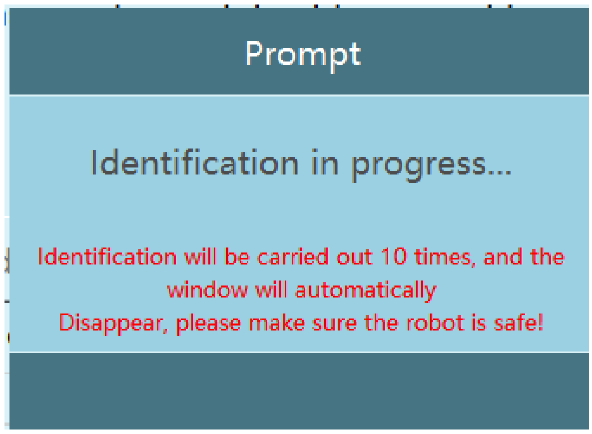
17. Click "Start identification".



18. Re-confirm that the trajectory is safe and people are away from the robot, then click "OK".

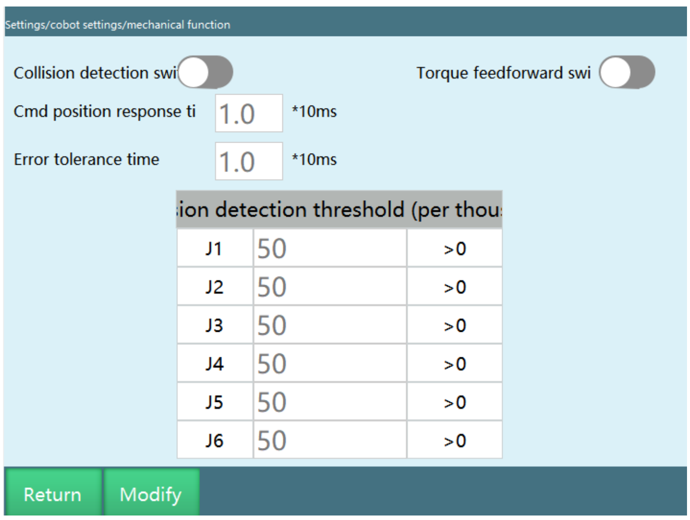


19. Then a pop-up window will appear to indicate that identification is in progress, please do not approach the robot before the end of identification. The robot may run the next trajectory at any time.



Mechanical functions

The mechanical functions include collision detection, torque feedforward and drag teaching, you need to enter **"Settings/Human-robot collaboration/Mechanical functions"** to set the collision detection and torque feedforward functions.



Collision detection switch: When turned on, the robot will perform collision detection based on sensitivity. Usually, you need to find the value that will not determine a collision when the robot is running, and then the robot will run normally

Instruction position response time: The robot body has touched something during operation, but the system will delay reporting an error because of the set time; when the time is up, the error appears and the robot is powered down

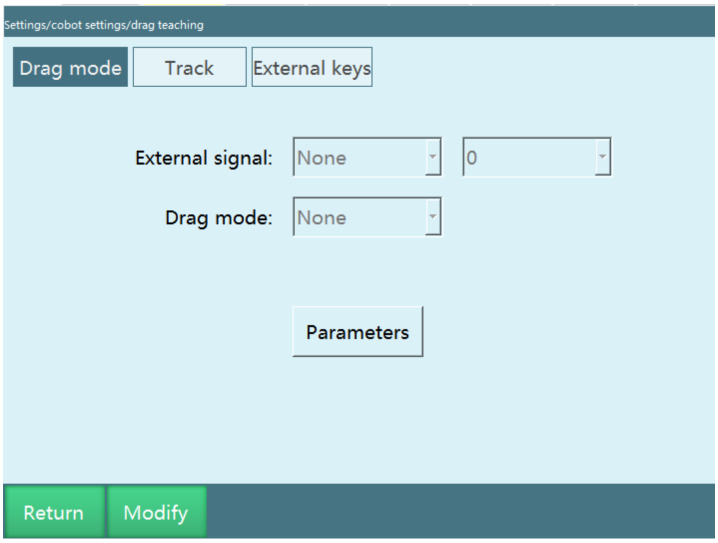
Error allowance time: PID adjustment causes torque fluctuations that mistakenly trigger a collision warning, this function is to prevent this phenomenon from occurring, and the alarm will not appear if the torque returns to the normal range within the set time

Drag teaching

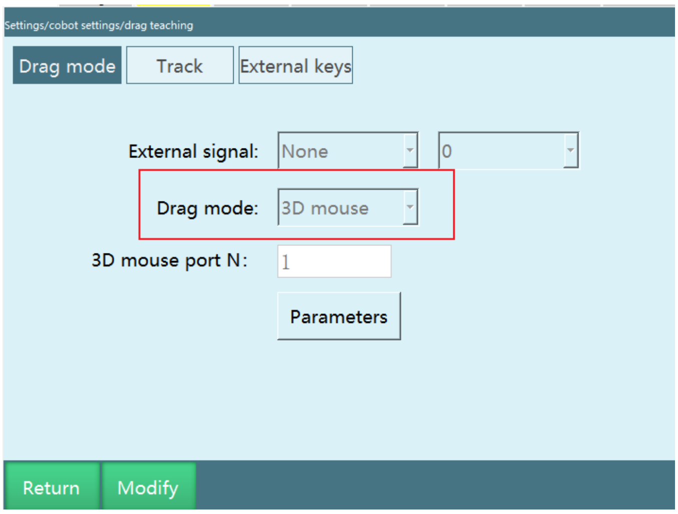
For "Drag mode", you can choose between "Torque" and "3D mouse".

You can set IO signals to switch between drag mode and jog mode, you can also switch between them using the ○ button on the teach pendant or the "Teach mode" button in the "Monitor" window.

You can switch to drag mode by triggering the external trigger signal (for example, the signal trigger mode is 0, then it must switch from 1 to 0 to take effect, if it is based on IO signals, then the ○ button will not take effect after the IO trigger).



Select "3D mouse" for "Drag method" after clicking on the "Modify" button.

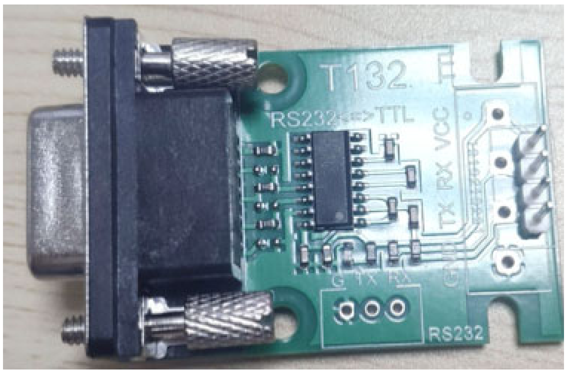


3D mouse

Accessory description

3D mouse-related accessories:

1. TTL to RS232 adapter



2. 5V power supply
3. 3D mouse body
4. Cable storage box
5. 3D mouse fixing plate

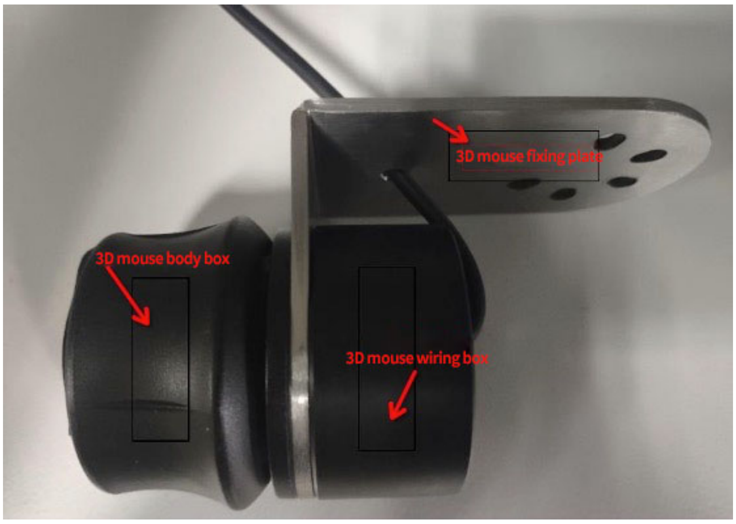
Wiring definition:

4	RxD (input)	orange

Wiring diagram:

Installation:

The installation parts of the 3D mouse are divided into 3D mouse body, 3D mouse wiring box and fixing plate.

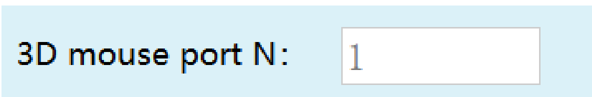


The 3D mouse cable box is used to store some of the 3D mouse connection cables and the fixing plate is used to mount the 3D mouse on the end of the robot. The 3D mouse can be mounted on the end of the robot after the components are assembled as shown above. It is also possible to use the 3D mouse without mounting it on the end of the robot, but the sense of direction when dragging is not as intuitive as when it is mounted on the end of the robot.

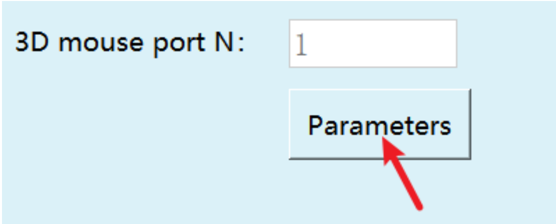
Power supply: External 5V power supply.

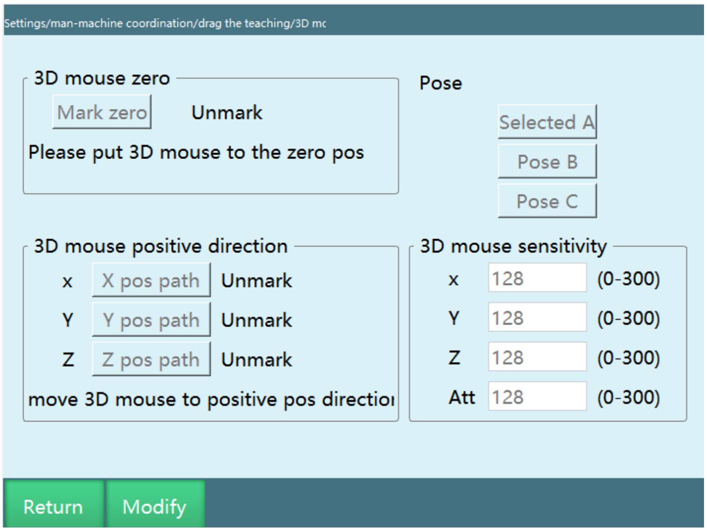
Wiring setup: The mouse conversion cable is plugged into the Com1 serial port of the controller and the Com1 serial port needs to support RS232 communication to be used directly.

Instructions for use and precautions:



3D mouse port number: equivalent to COM port on the controller; select the COM port with the number you filled in.





If you install the 3D mouse on the robot body, make sure the robot is safe to use before you use it！！！！！

Mark zero: Mark the 3D mouse zero position, "Unmarked" means no zero point has been marked, and "Marked" will be displayed when it is marked.

Usage: Click "Modify", then click "Mark zero" to finish marking, no need to move the mouse.

Mark positive direction: Mark X, Y, Z positive directions, "Unmarked" means no direction has been marked, and "Marked" will be displayed when it is marked. If the communication fails after pressing, the system will show communication failure, and the direction will follow the last marked direction in this case.

Usage: Click "Modify", then click the "Mark direction" button, then press the corresponding direction of the mouse, when the indication of successful direction marking appears, it means the marking of that direction is completed.

Pose control: Select the pose controlled by the mouse rotation, you can choose to control pose A, B, C.

Usage: Click "Modify" and click the corresponding pose button to complete the selection.

3D mouse sensitivity: Used to control the sensitivity of the 3D mouse to control the corresponding direction and posture.

Usage: Click "Modify", enter the value, the value range is 0-300, the larger the number, the higher the sensitivity.

Key sequence for first use:

1. Click "Modify"
2. Mark the zero point
3. Mark XYZ directions
4. Set the sensitivity value
5. Click "Save"

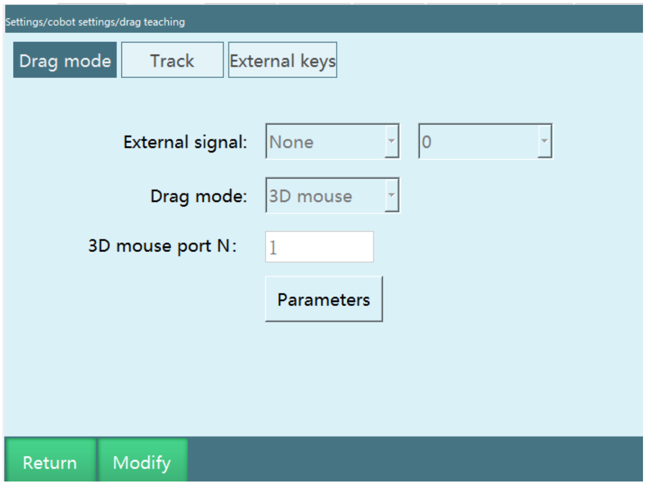
3D mouse control of robot movement

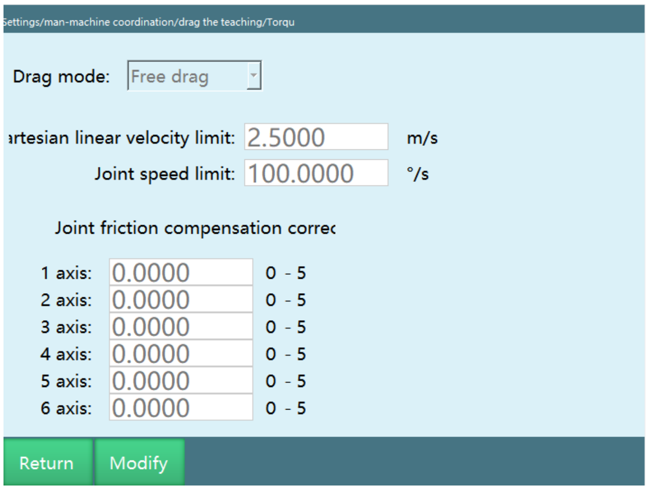
1. Complete zero point setting and direction marking
2. Enable the servo via the teach pendant
3. By pressing the 3D mouse in the corresponding direction, the robot can be controlled to move in that direction
4. 3D mouse supports robot motion in all coordinate systems, but the direction correspondence only applies to the Cartesian coordinate system, and the other coordinate systems are used to control individual joint movements, which is different from the motion in the Cartesian coordinate system

Torque dragging

Parameter description

Parameter setting interface





Drag mode: You can choose from three modes, such as free dragging, position dragging and posture dragging

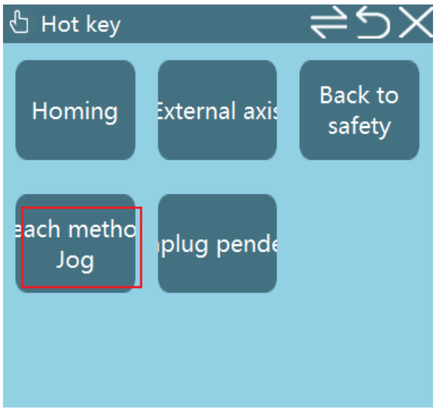
Cartesian space line speed limit: temporarily invalid

Joint space speed limit: the maximum speed when dragging, the robot will power down and stop when the limit is exceeded

Joint friction compensation correction factor: range 0-5, the closer the parameter is to 5, the more flexible the joint is; it is recommended that the parameter be tested from 0.

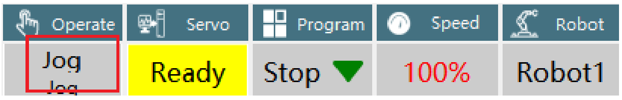
Drag mode switching

1. Switch using the Teach pendant - Monitor - Shortcut - Jog method button



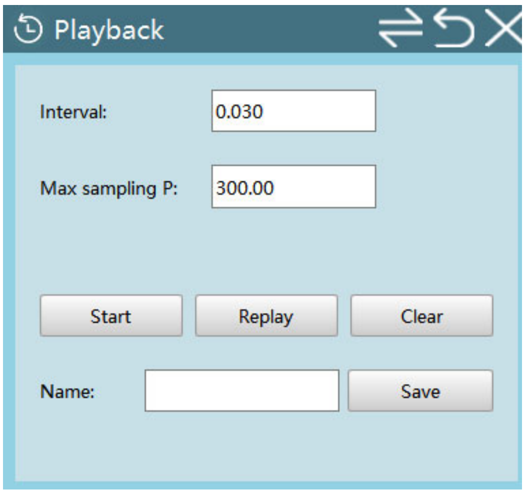
2. Switch using the ○ button (the bottom left button) on the teach pendant
3. Switch using external signal (DIN input signal)

Check if the status bar of the teach pedant is in drag mode



After entering drag mode, power up and drag the robot

Drag teaching trajectory playback

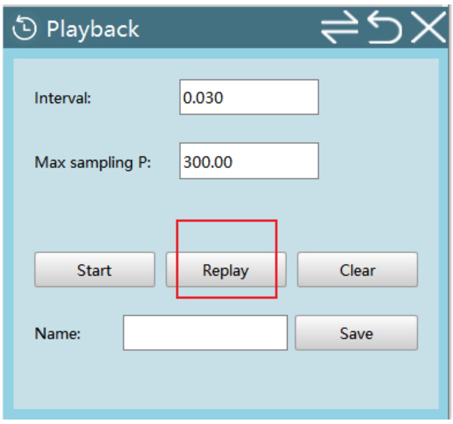


Sampling interval: unit s, acquiring points at every sampling interval.

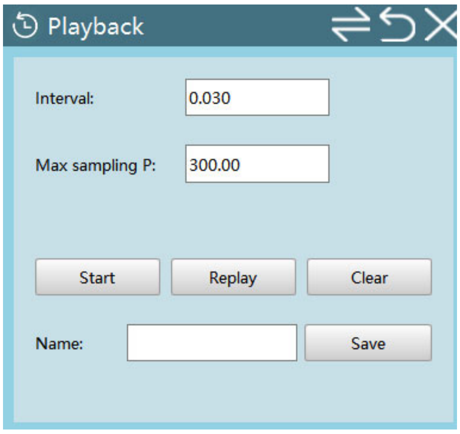
Maximum number of sampling points: range 200 to 12000, the maximum number of points of a recorded trajectory.

Operation steps:

1. Enter Monitor - Shortcut - Trajectory playback interface

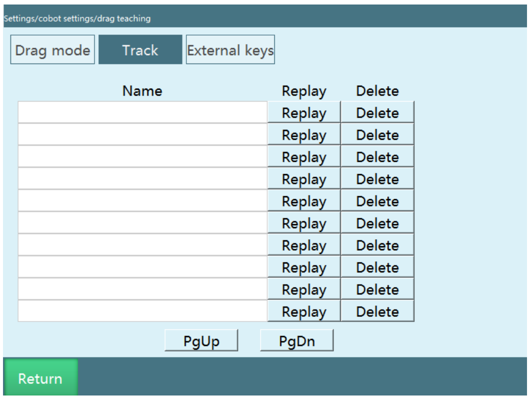


2. Switch to drag mode, set sampling interval and maximum number of sampling points



3. Power up, click the "Start" button in the "Monitor" pop-up and start dragging the robot
4. Click "Stop" or wait until the point recording is completed, the interface shows that the trajectory has been recorded
5. At this point you can power down, switch to jog mode, click the "Playback" button to play back the trajectory you just dragged
6. Enter the trajectory name and click "Save" to save the trajectory you just recorded
7. Clear: Clear the recorded trajectories

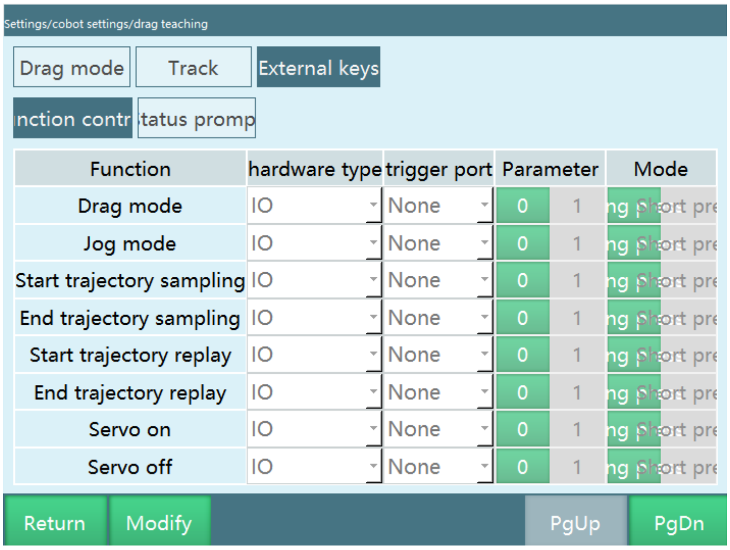
Trajectory management



Enter "Settings-Human-robot collaboration-Drag teaching-Trajectory management" interface

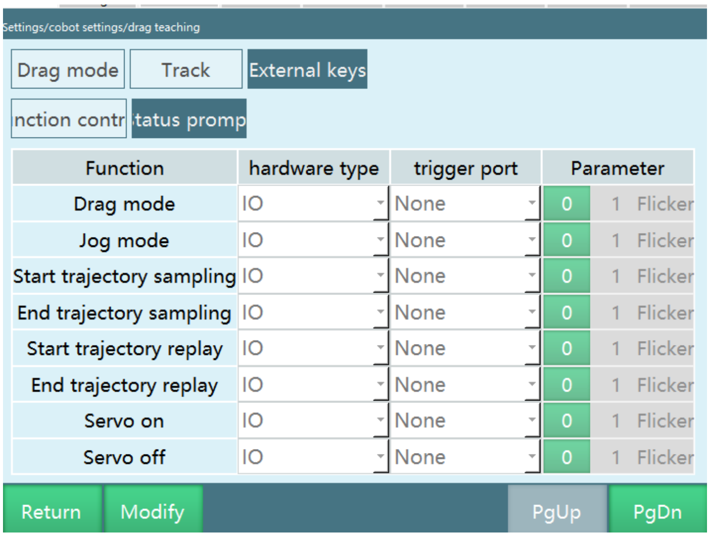
Here you can play back and delete the saved trajectories

External keys



In the "External keys" interface, you can control the robot’s drag/jog mode, start/end track acquisition, start/stop track playback, up/down enable and other functions through the set trigger port, parameters and methods

(Note: The same type can use the same trigger port, the trigger signal must be rising edge or falling edge to be valid, and the long press needs to meet the continuous input of 3-10)



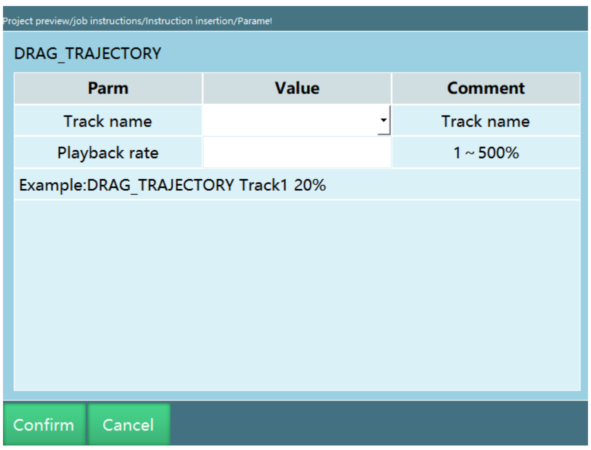
When the status prompt interface satisfies the corresponding function, IO will make a corresponding response according to the set trigger port and parameter type

Drag teaching instructions

DRAG_TRAJECTORY instruction

This instruction is used to call the trajectory playback record. When the playback rate is filled in 100%, it means the current dragging speed, 500% means five times the current dragging speed, and so on.

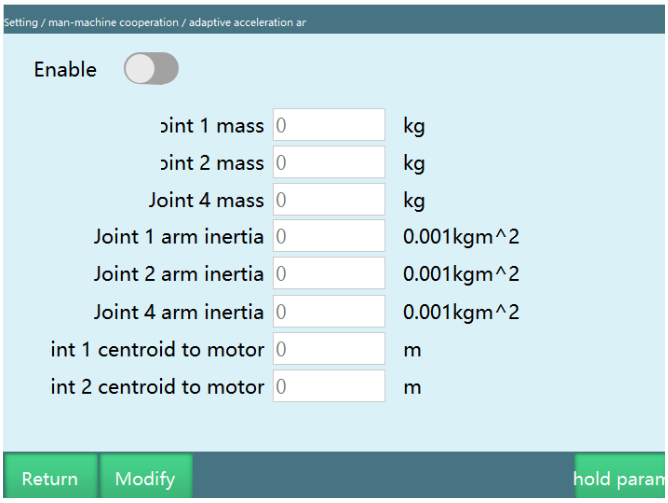
Note: The running speed of this instruction is the dragging speed x playback rate, and the status bar speed does not affect the speed of this instruction.



Adaptive acceleration/deceleration

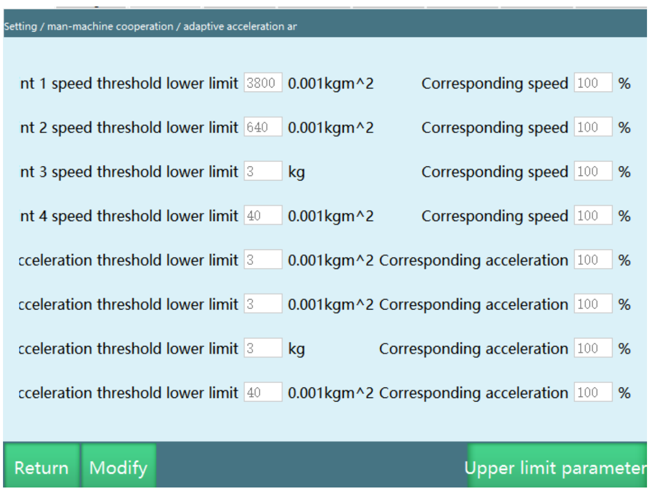
When the adaptive acceleration/deceleration enable switch is on, it protects the motor from excessive torque during motor movement (only supports 4-axis scara robots)

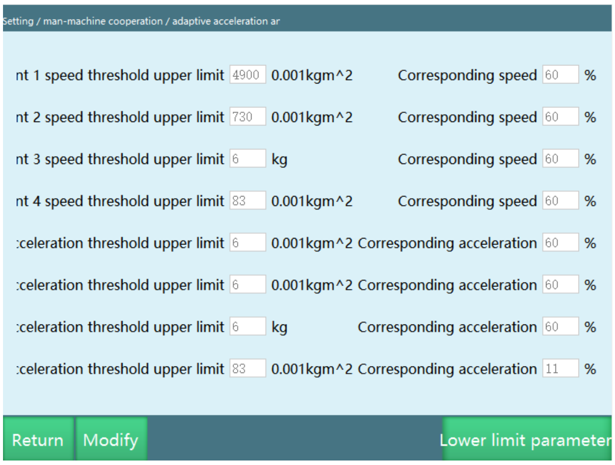
You can enter **"Settings/Human-robot collaboration/Adaptive acceleration/deceleration"** to set the adaptive acceleration/deceleration. The relevant steps are as follows:



Fill in the appropriate parameters according to your needs and turn on the enable switch to take effect.

Threshold parameters: The interface to fill in the upper and lower speed and acceleration limits, as shown below

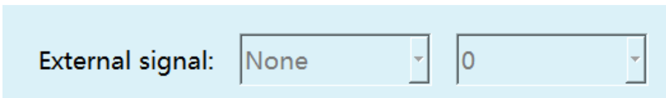




Load drag teaching process

1. Perform identification according to the dynamics parameter process
2. Install the load after identifying successfully
3. Then set up the parameters in the drag teaching interface, and the drag mode can be selected from torque and 3D mouse.

Switching between drag mode and jog mode can be done using the ◯ button on the teach pendant, "Teach mode" button in "Monitor" interface, and external trigger IO signals.



Notes

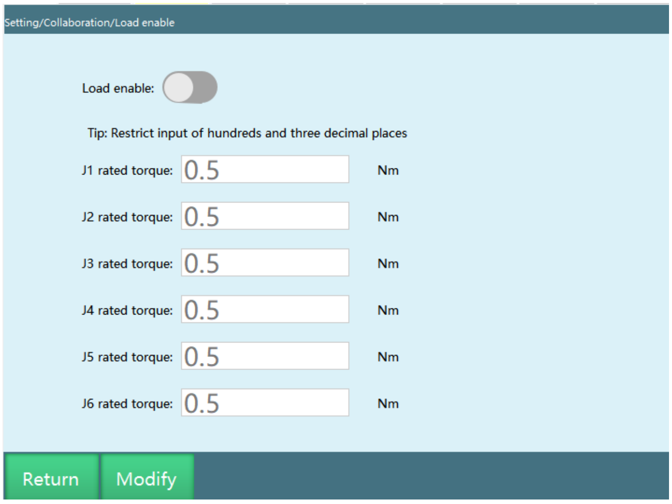
- The ◯ button and the "Teach mode" button are not available after switching to drag mode using the IO signals.

4. Finally, set the load enable parameters (load parameters are set in the tool hand interface and load enable interface respectively). Turn on the load enable switch, save the settings and then switch the teach pendant from jog mode to drag mode. You can drag after powering up.

- Load enable interface settings are as follows: Settings - Human-robot collaboration - Load enable

Load Enable: Determine whether to enable the load function or not. When the load enable switch is turned on, the system will calculate the load torque when the robot arm is running according to the load parameters under the selected load number.

Rated torque: the rated torque of each joint motor (refer to the rated torque in the servo parameters)

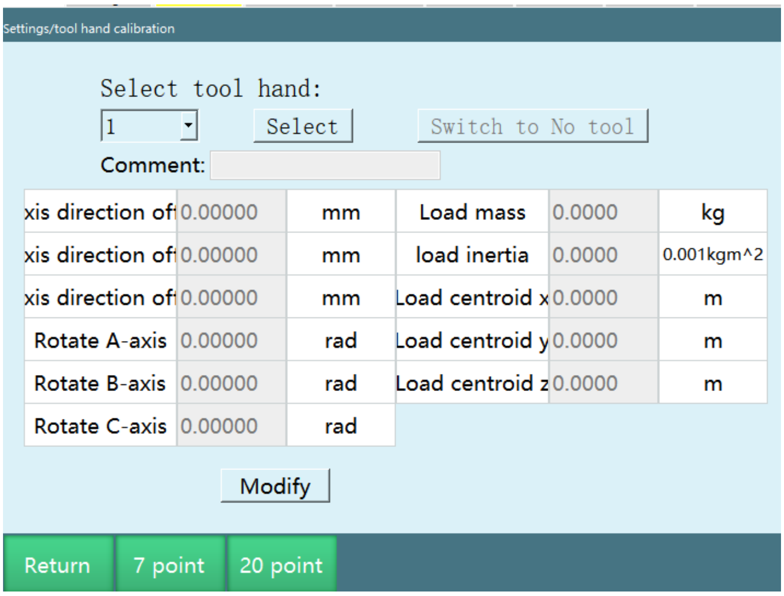


- Tool hand interface settings are as follows: Settings - Tool hand calibration

Load number (i.e. tool hand number): The tool hand number is the load number.

Load mass: total mass of the robot end load

Load inertia: the rotational inertia of the load



The following XYZ are based on end coordinate system. (End coordinate system axis confirmation method: in the case of no tool hand, go TX, TY, TZ under the tool coordinate system to confirm the direction of XYZ)

X: Offset (distance) of load center of mass along X direction, starting from the center of flange

Y: Offset (distance) of load center of mass along Y direction, starting from the center of flange

Z: Offset (distance) of load center of mass along Z direction, starting from the center of flange

XYZ supplementary note: It is recommended that after installing the load, adjust the axis 6 zero point at the zero position of the robot, so that the load center of mass is directly in front of the robot, at this time, X is the horizontal distance between the load center of mass and the axis 6 center, Z is the vertical distance between the load center of mass and the axis 6 center, and Y is 0

Notes

- The above specific parameters can be consulted with the manufacturer

iNexBot

Multi-Robot Mode and Dual-Robot Collaboration

<<<

Catalogue

Multi-Robot Mode and Dual-Robot Collaboration

Multi-robot mode

Multi-robot mode refers to controlling multiple robots by debugging one teach pendant.

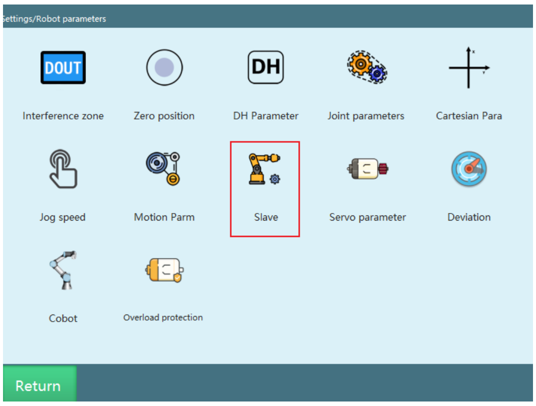
This product supports controlling up to 4 robots at the same time. This chapter will introduce the methods and steps for setting the number of robots to be controlled at the same time, switching robots, dual-robot collaboration, and multiple robots running programs at the same time.

Setting robot

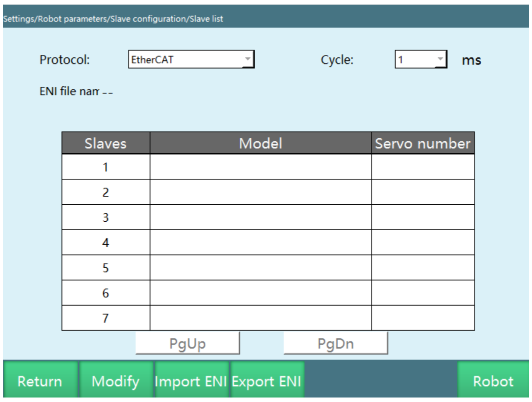
In the robot selection interface under the "Settings" interface, you can select the number and type of robots.

The steps are as follows:

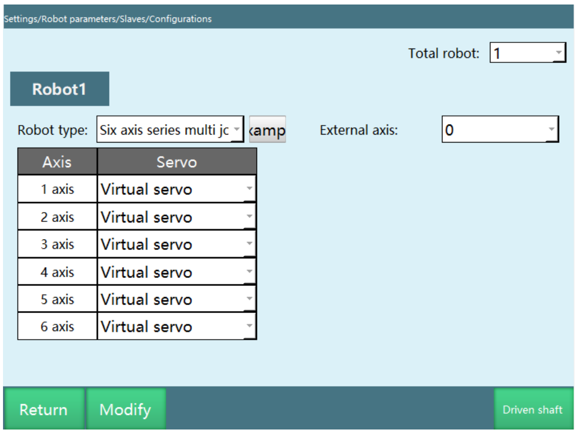
1. Switch the permission to "Admin";
2. Go to "Settings/Robot parameters/Slave configuration";

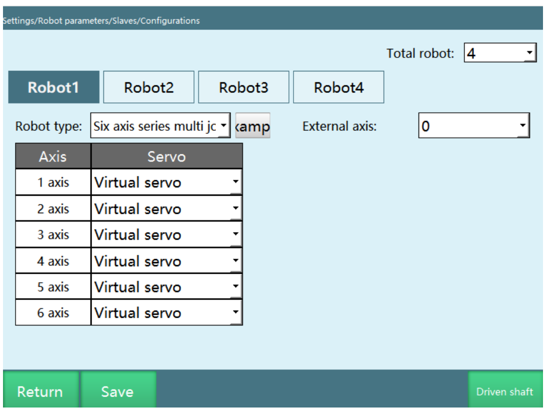


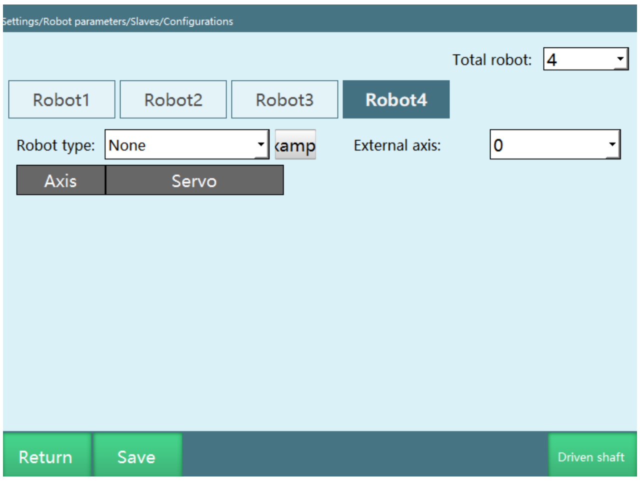
3. Go to "Settings/Robot parameters/Slave configuration/Slave list";

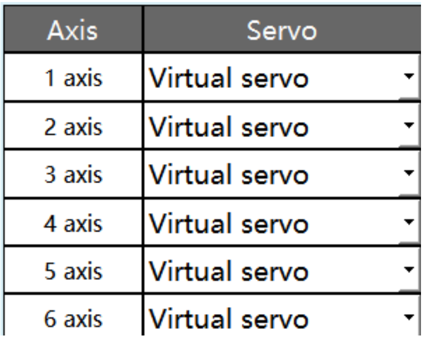


4. In the "Number of robots" drop-down box, you can select the number of robots to be controlled at the same time, as shown in the figure below. When the number is 1, it means single robot mode. In this case, if the model of robot 1 is changed, the interface of the teach pendant will also change accordingly. At present, the maximum number of controlled robots is 4, so the number of robots can be selected is in the range (1~4);





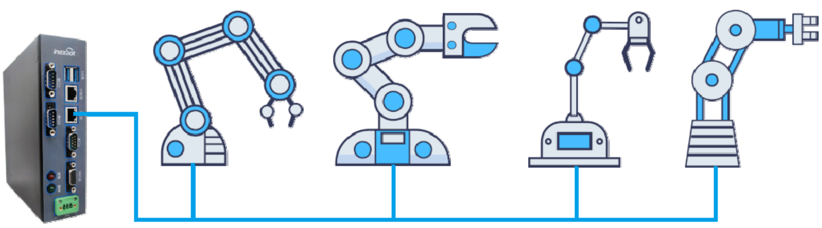




Axis	Servo
1 axis	Virtual servo
2 axis	Virtual servo
3 axis	Virtual servo
4 axis	Virtual servo
5 axis	Virtual servo
6 axis	Virtual servo

5. After selecting the number, you need to set the model of each robot and the corresponding servo model. The order of the robot is determined by the order in which the controller is connected in series with the robot;
6. After all robot models and servo models are set, press the [OK] button to save;
7. Restart.

The robot sequence is as follows:



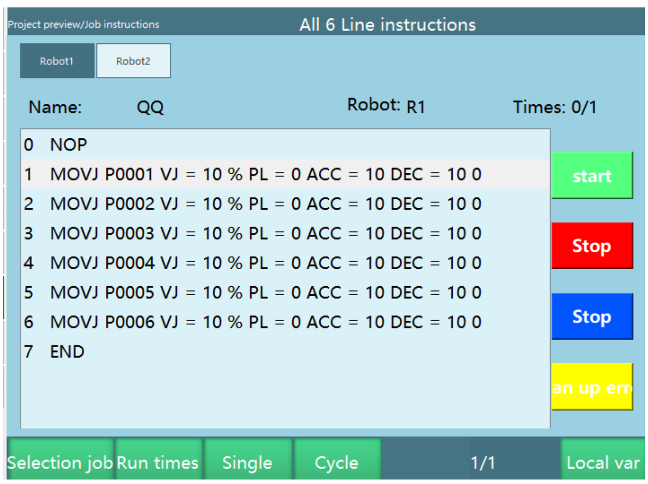
Switching robot

- When the mode selection key is at "Teach mode", press the [Robot] button to switch between the robots and teach them separately. At this time, the "Robot" column in the upper status bar will display the serial number of the currently operating robot.
- The job files are not common between robots, and the job files are also switched when the robot is switched.
- If the robot is switched to a different type, the related interfaces will also change. When the switched robot type is a 4-axis SCARA robot, "DH parameter setting", "User coordinate system setting", "Joint parameter setting", "Robot zero position", "Servo status", "IMOV instruction insertion" and other interfaces will switch to the corresponding interfaces according to the number of the axis of the current robot.
- The coordinate system on the right side of the interface will also change, how many axes the current robot has, how many axes will be displayed there.

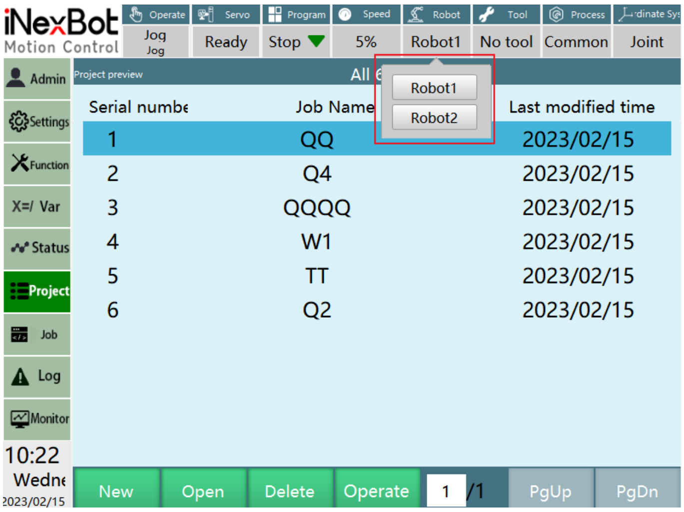
Multi-robot mode

- Main interface

When the mode selection key is at "Run mode", press the [Robot] button to switch between each robot and enter the multi-robot mode. The interface is as follows:



Press [Select program] to choose among various programs, and the interface is as follows:



After selecting the program, click the [Open] button in the operation area at the bottom of the interface, set the current program as the program running by the current robot.

- Operation area

In this mode, you can only start and stop the running programs.

Click the [Robot 1] button, [Robot 2] button, [Robot 3] button, and [Robot 4] button in the operation area at the top of the interface to switch the display interface of each robot.

Click the [Start] button in the operation area on the right side of the interface to run the selected program for the current robot.

Click the [Stop] button in the operation area on the right side of the interface to stop the operation of the current robot.

Click the [Servo ready] button in the operation area on the right side of the interface to enter the servo ready state for the current robot.

Click the [Clear error] button in the operation area on the right side of the interface to clear up any servo errors occured for the current robot.

Click the [Set times] button in the operation area at the bottom of the interface to set the running times after which the current robot will stop.

Click the [Cycle mode] button in the operation area at the bottom of the interface to set the current robot to run for infinite times.

Click the [Select program] button in the operation area at the bottom of the interface to set the programs that the current robot runs.

The physical buttons [Start] and [Stop] on the teach pendant are for all robots, when pressed, all robots will start or stop running.

Dual-robot collaboration

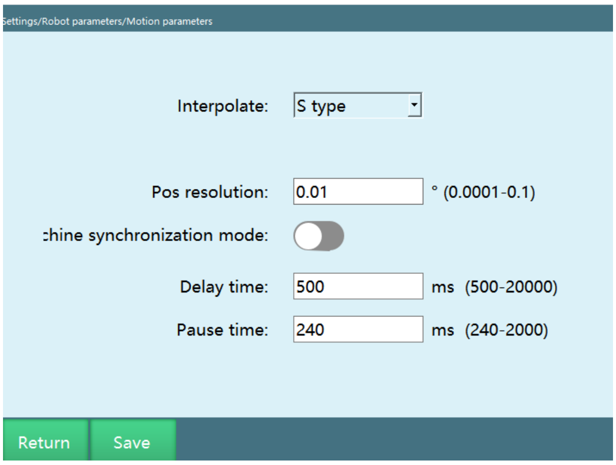
As to dual-robot collaboration, please use **two identical 6-axis robots**, and configure them according to multi-robot mode settings.

Please fill in the same values for the joint parameters and DH parameters of the two robots.

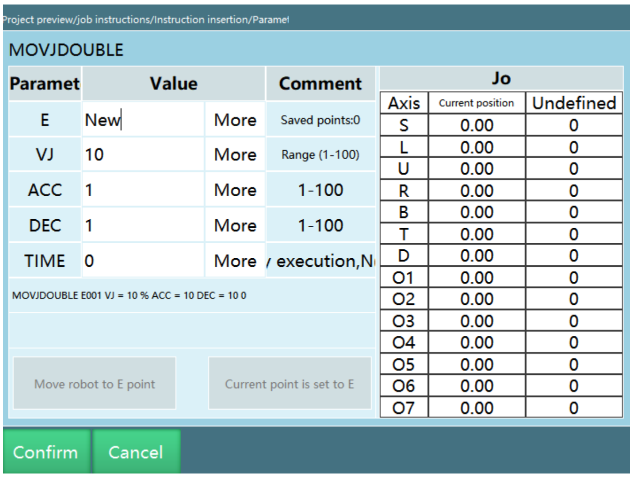
To enable dual-robot collaboration, please enable dual-robot collaboration in **"Settings-Robot parameters-Motion parameters"**.

Notes

- Turning off the dual-robot collaboration button requires a restart of the controller system; but turning it on does not require a restart.
- If the number of robots is greater than 2, the dual-robot collaboration function will be automatically turned off when restarting.
- Dual-robot collaboration cannot be used in conjunction with multi-robot mode.
- Dual-robot mode and external axes cannot be used at the same time.



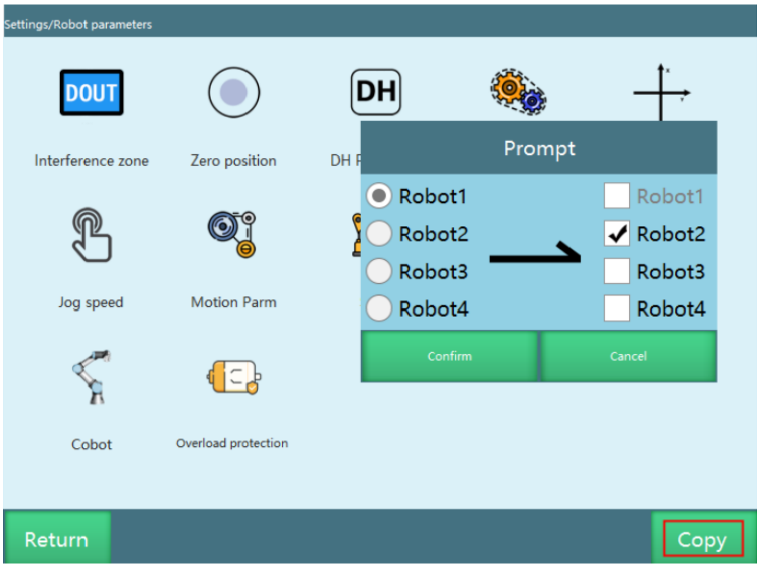
After enabling the dual-robot collaboration, the first robot is the master robot and the second robot is the slave robot. Please use the [Robot] button on the left side of the teach pendant to switch the master and slave robots for teaching. After switching to the slave robot, "Robot2" will be displayed at the current operating robot in the status bar above the teach pendant. Please do not use the [External axis] button to switch to robot 2 for teaching.



The instructions to control the simultaneous movement of the two robots are MOVJDOUBLE, MOVLDOUBLE, MOVCDOUBLE and MOVCADOUBLE. For example, both robots can move to the position point with joint interpolation or linear interpolation.

"Copy parameter" function

Function: Copy the parameters of the current robot to other robots



Note: 1. The "Copy parameter" button is only displayed when the number of robots is greater than 1

2. The copied parameters do not include: zero position, slave configuration, NP parameters, servo parameters, collaborative robot

3. The target robot copied to does not contain the original robot

Instructions

Dual-robot point-to-point MOVJDOUBLE

When set to two robots, make the two robots move to the target position with joint interpolation at the same time; start and stop at the same time

Dual-robot linear MOVLDOUBLE

When set to two robots, make the two robots move to the target position with linear interpolation at the same time; start and stop at the same time

Dual-robot circular MOVCDOUBLE

When set to two robots, make the two robots move to the target position with circular interpolation at the same time; start and stop at the same time

Dual-robot full circle MOVCADOUBLE

When set to two robots, make the two robots move to the target position with full circle interpolation at the same time; start and stop at the same time