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

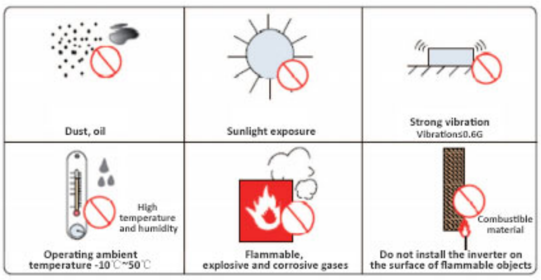
3.Please install it in a place that is not easy to vibrate. Vibration should not be greater than 0.6G. Pay special attention to keep away from equipments such as punches

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



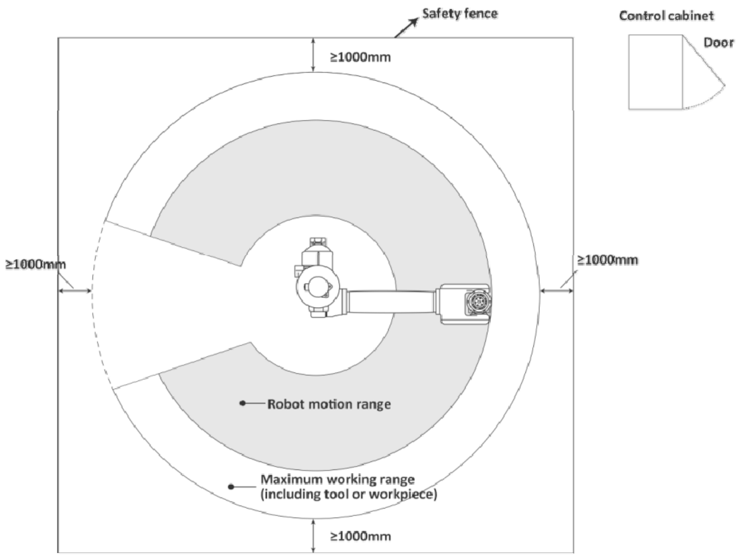
Installation location

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:

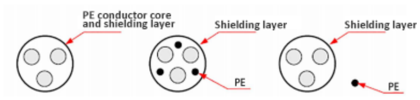
Level 1: sensitive signals (low-voltage analog signals, high-speed encoder signals, high-speed communication signals, ±10V analog signals, low-speed 422&485 signals, digital input and output signals)

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

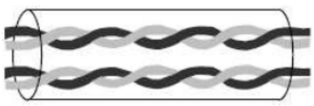
1.In the process of cable selection, it is recommended to use symmetrical shielded cables for input and output main circuit cables. Compared with four-core cables, the use of symmetrical shielded cables can reduce the electromagnetic radiation of the entire conduction system

2.Recommended power cable type - symmetrical shielded cable

Recommended signal cable type - shielded twisted pair cable



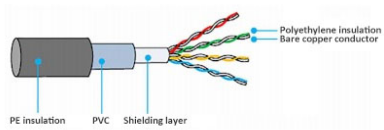
Schematic diagram of symmetrical shielded cable



Schematic diagram of shielded twisted pair cable

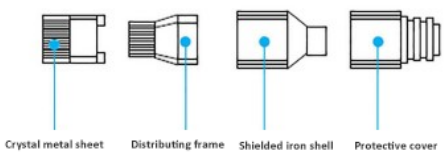
Note: Shielded twisted pair cable is recommended for digital signal lines

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

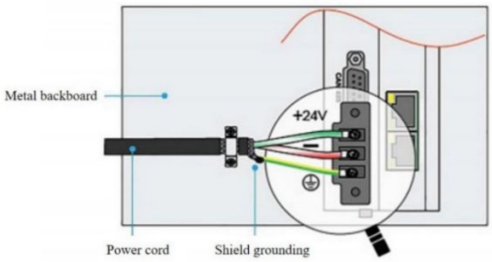
3.Keep the cables separated according to the grade, and if cables of different grades must be crossed, they should be kept 90° crossed

Grounding requirements

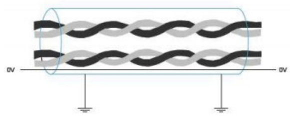
Warnings

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.

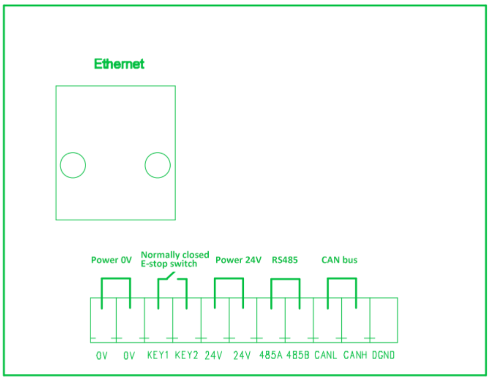
Note:

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

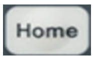
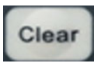


Teach Pendant Buttons and Interface Introduction

T30 teach pendant physical buttons

Left side

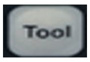
	Switch current servo status

	Switch current robot (only available in multi-robot mode)
	Switch between current robot and external axes (only available when there are external axes)
	Click the button to return to zero point
	Click the button to return to reset point
	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

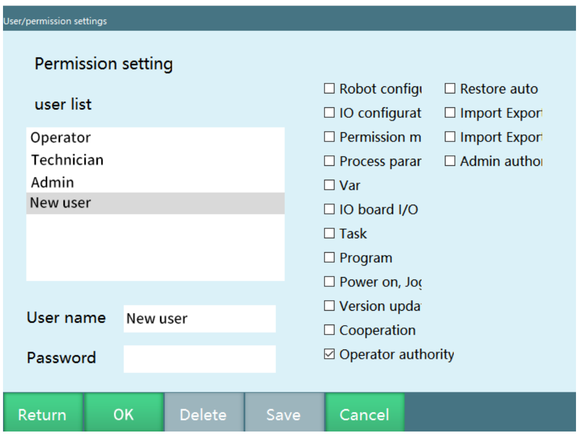
Basic instructions

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

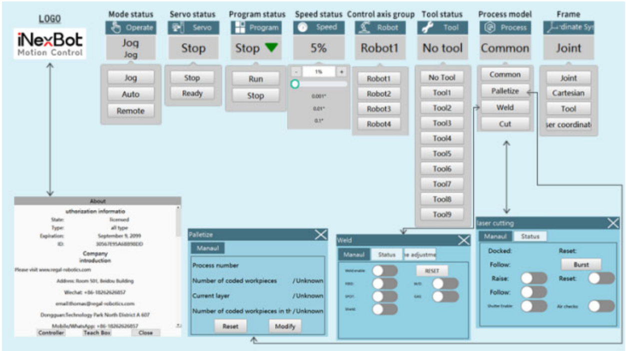
Permission settings:

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:

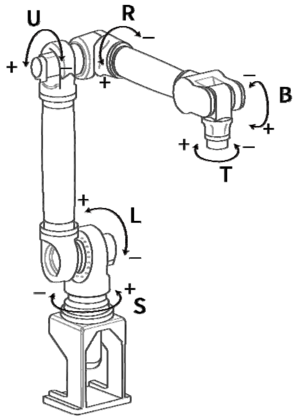
The front end of the robot moves in parallel along the X, Y and Z axes of the base. A, B and C rotate around the X, Y and Z axes respectively. The Euler angle rotational sequence used in this system is X'Y'Z' and the fixed angle rotational sequence is ZYX.

Tool coordinate system:

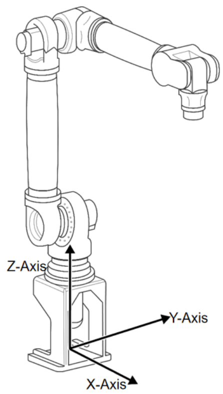
The tool coordinate system takes the effective direction of the robot's wrist tool as the Z-axis, defines the origin of the coordinate system at the tip point of the tool, and the tip point of the body moves in parallel according to the coordinates. TA, TB and TC rotate around the TX, TY, TZ axes respectively.

User coordinate system:

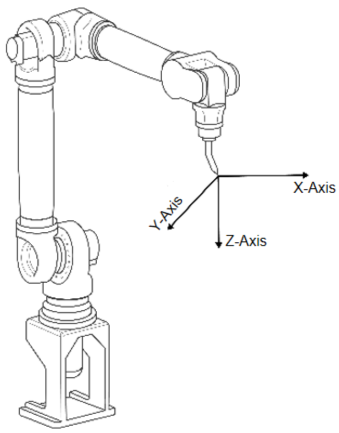
XYZ Cartesian coordinates are defined anywhere. The body tip point moves in parallel according to the coordinates.



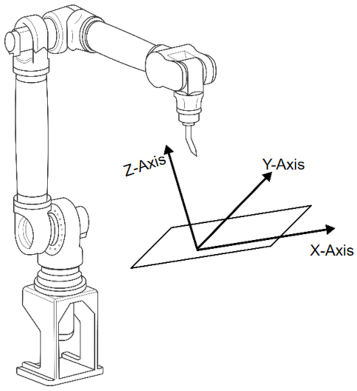
Joint coordinate system



Cartesian coordinate system



Tool coordinate system

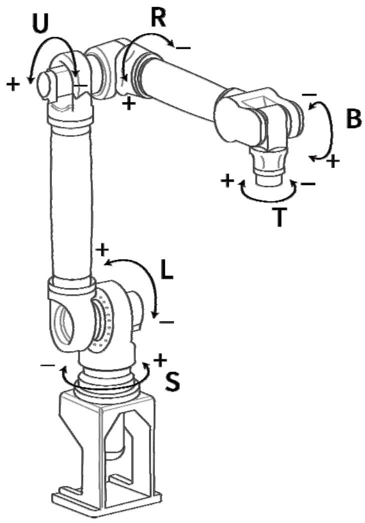


User coordinate system

Coordinate systems and axis operations

Joint coordinate system

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

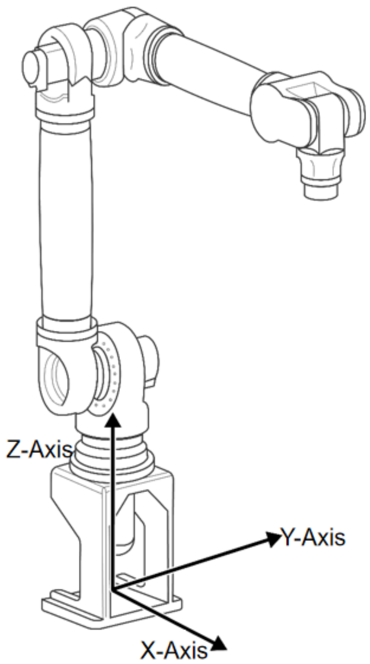


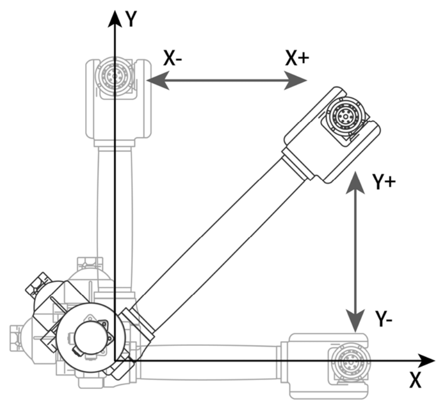
Axis operations in joint coordinate system

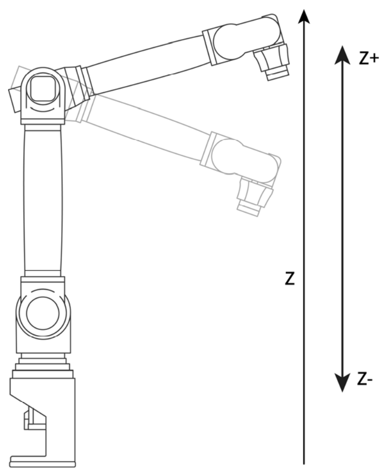
Axis name		Axis operation	Action
Basic axis	S axis	S+/S-	Body rotates left and right
	L axis	L+/L-	Lower arm moves forward and backward
	U axis	U+/U-	Upper arm moves up and down
Wrist axis	R axis	R+/R-	Wrist rotates
	B axis	B+/B-	Wrist moves up and down
	T axis	T+/T-	Wrist rotates

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-

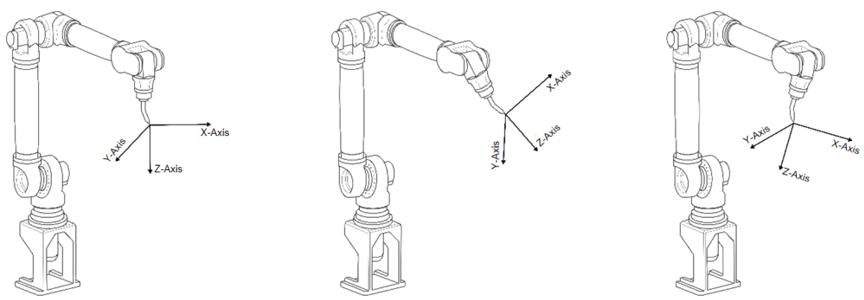
Axis operations in Cartesian coordinate system

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

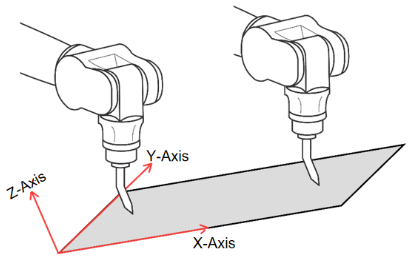
In the tool coordinate system, the robot moves in parallel along the X, Z and Y axes defined at the the tool tip point.

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.

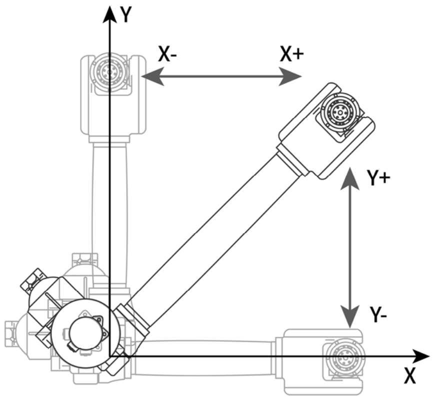


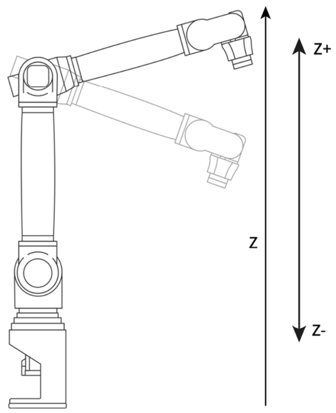
Axis operations in tool coordinate system

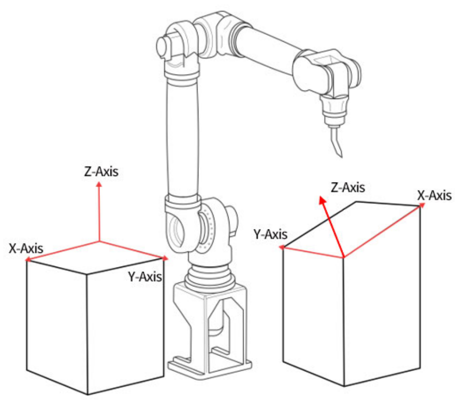
Axis name		Axis operation	Action
Basic axis	TX axis	TX+/TX-	Move in parallel along the TX axis
	TY axis	TY+/TY-	Move in parallel along the TY axis
	TZ axis	TZ+/TZ-	Move in parallel along the TZ axis
Attitude axis	TA axis	TA+/TA-	Rotate around TX axis
	TB axis	TB+/TB-	Rotate around TY axis
	TC axis	TC+/TC-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

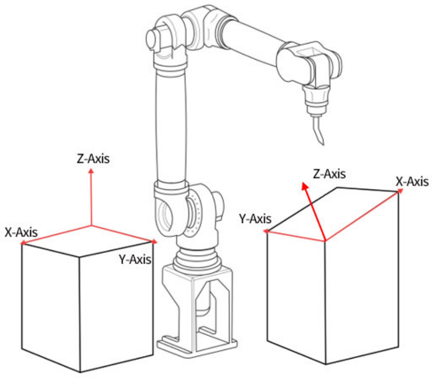
Use case of user coordinate system

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

Here, we will illustrate this with a few examples.

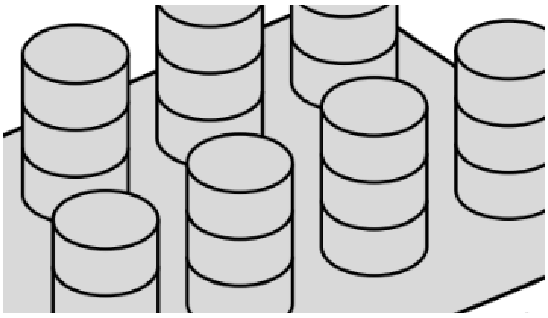
1.When there are multiple fixture tables:

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



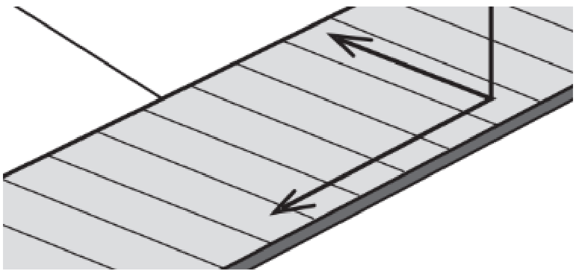
2.When engaged in arranging and stacking operations:

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

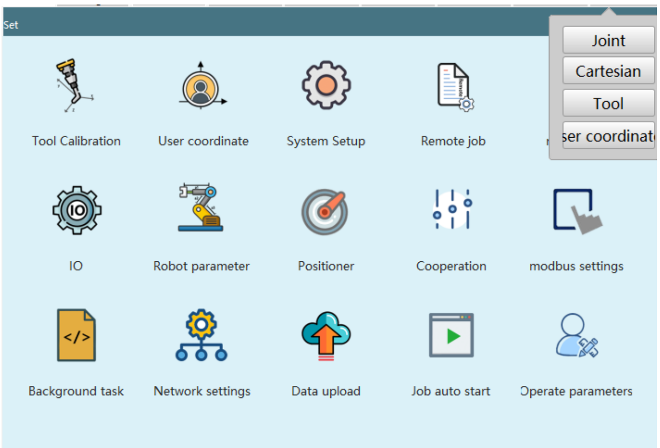
Coordinate system description and switching

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

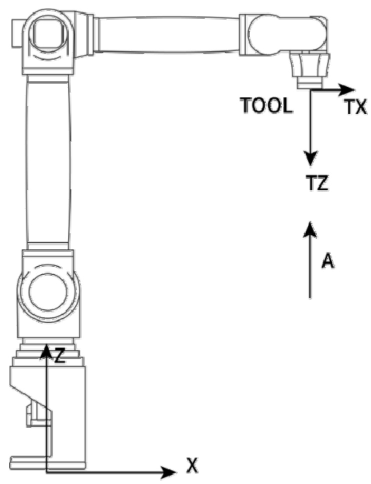


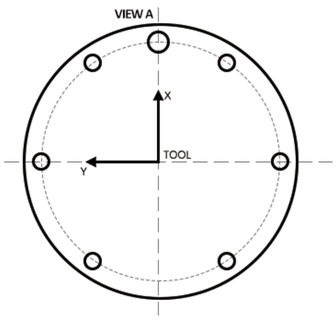
Tool Hand and User Coordinates

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.





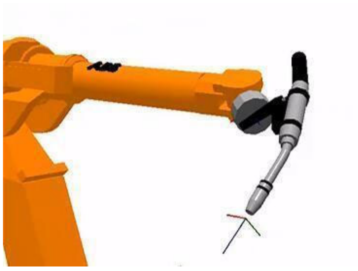
TCP: TOOL CENTER POINT

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.

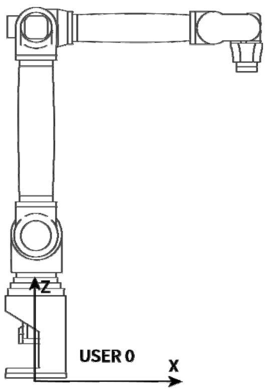
In order to describe the position of an object in space, it is necessary to fix a coordinate system on the object, and then determine the pose of the coordinate system (origin position and three coordinate axis attitudes), i.e., seven DOFs (degrees of freedom) are needed to completely describe the pose 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?



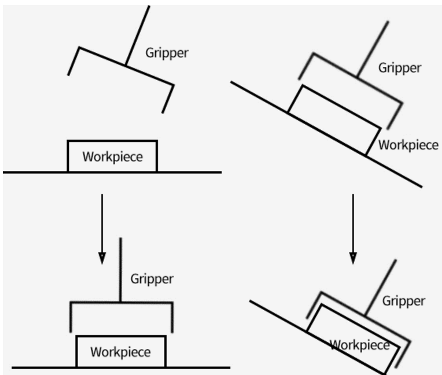


Fig.1 Fig.2

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.

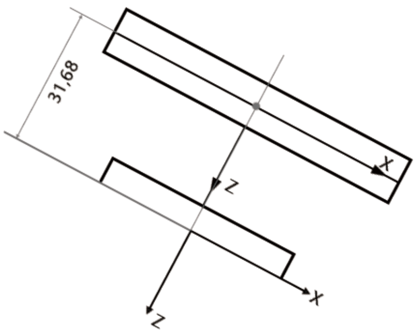
Conclusion: The role of establishing the tool coordinate system:

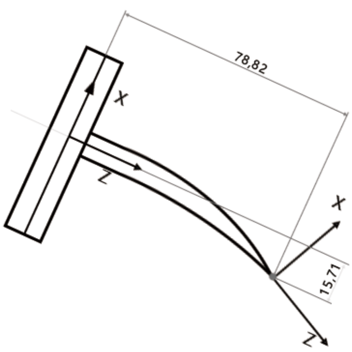
1.Determine the TCP point (i.e. tool center point) of the tool to facilitate the adjustment of the tool state.

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

Tool coordinate system characteristics

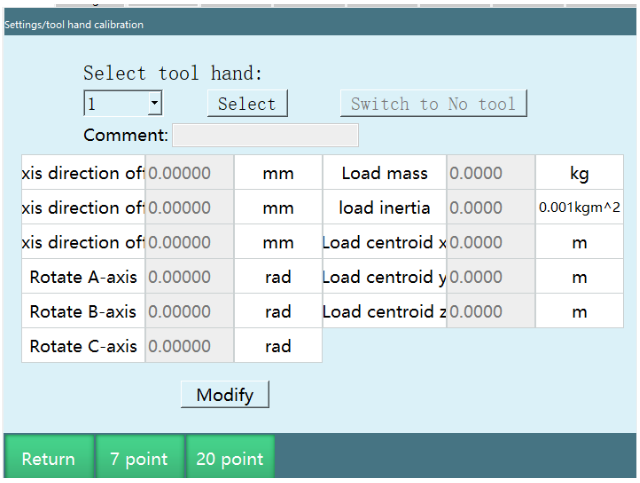
The new tool coordinate system is a change from the default tool coordinate system. The position and orientation of the new tool coordinate system always maintain the absolute position and attitude relationship with the flange, but it is always changing in space.





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

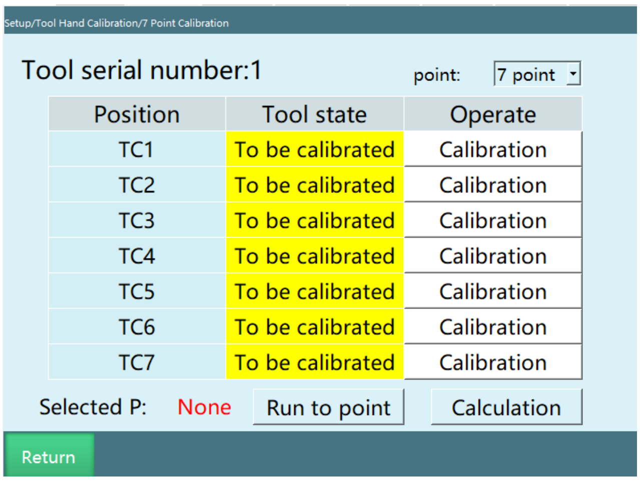
Keep the flange parallel to the horizontal plane before measuring the data

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

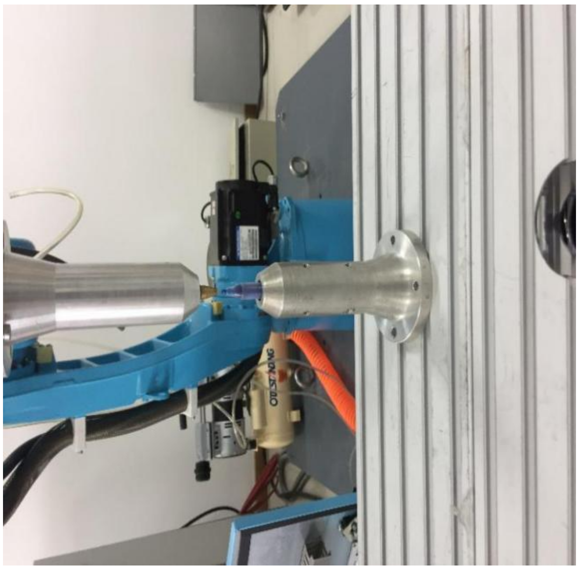
Click the [7-point calibration] button at the bottom to enter the "7-point calibration" interface, as shown in the figure



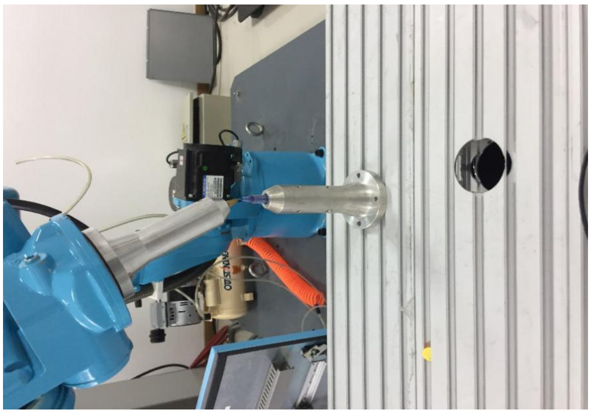
If the detailed parameters of the tool are not available, TCP calibration can be performed to automatically calculate each dimensional parameter of the tool. The specific calibration steps are as follows:

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

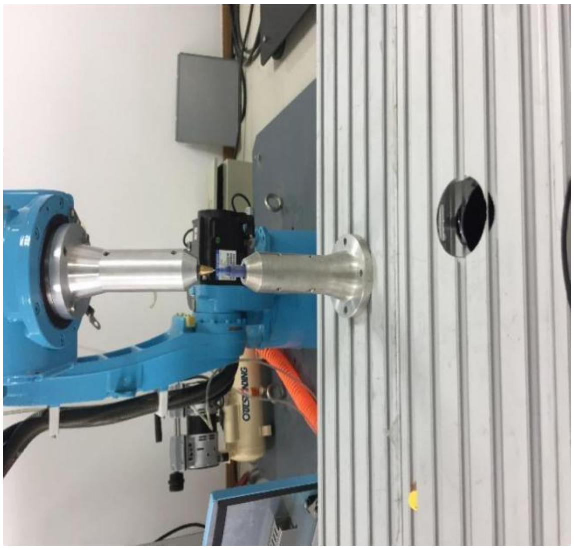
2.With the tool end vertical and facing the reference point, click the [Calibrate] button corresponding to "TC1" in the interface, as shown in the figure below.



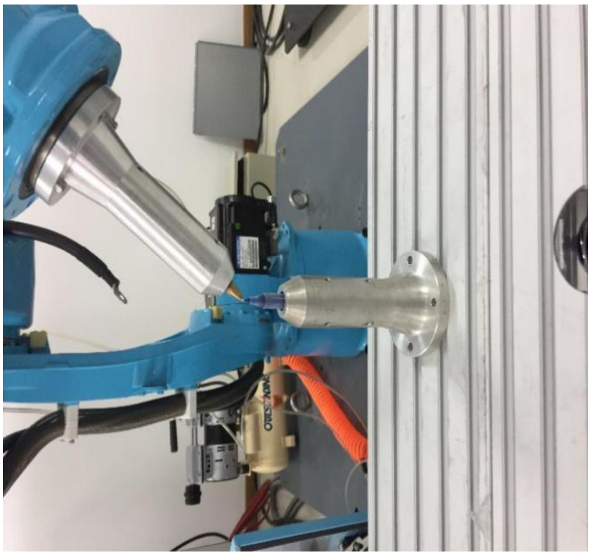
3.TC2 calibration: Switch the robot to a posture with the tool end facing the reference point, and click the [Calibrate] button corresponding to that line, as shown in the figure below.



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.



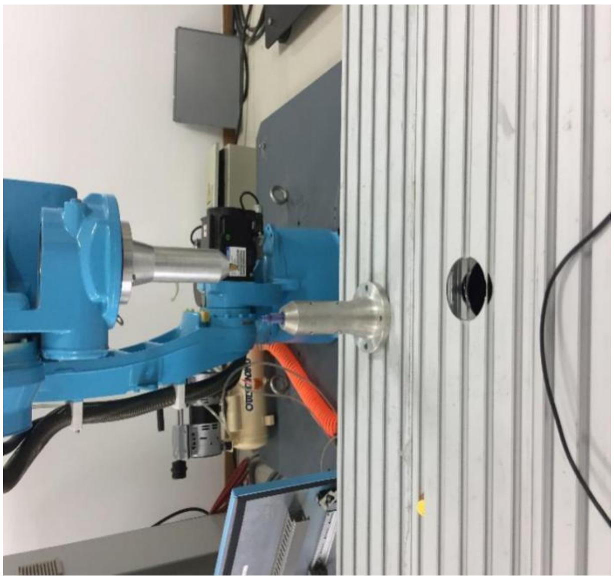
5.TC4 calibration: Switch the robot to a posture with the tool end facing the reference point, and click the [Calibrate] button corresponding to that line, as shown in the figure below.



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.

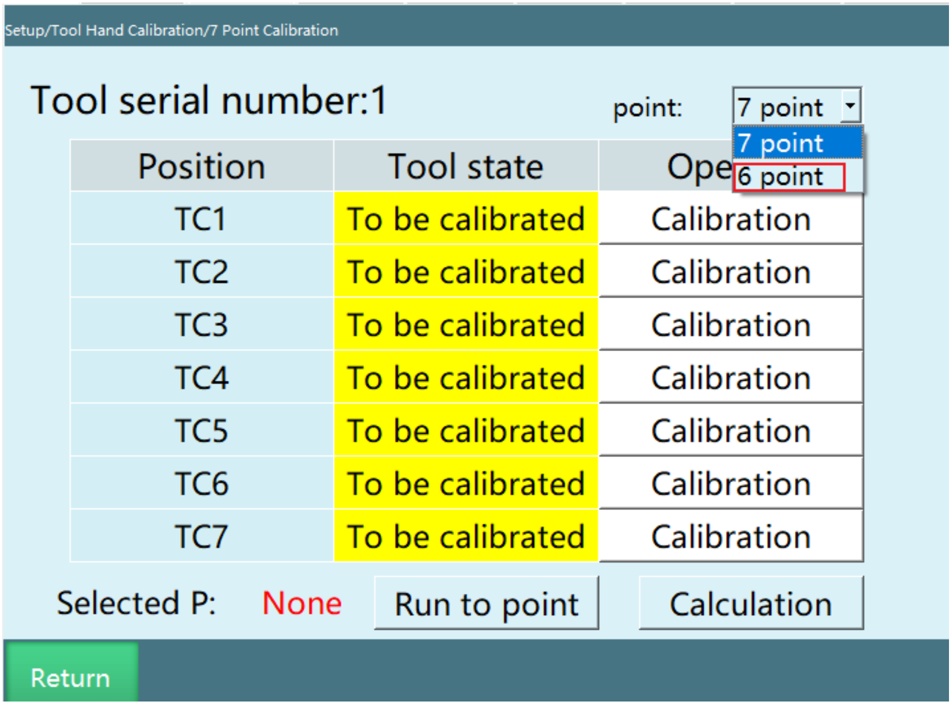
If you are not satisfied with a point that has been calibrated during the calibration process, you can click the [Cancel calibration] button corresponding to that line to cancel the calibration and then calibrate the point again.

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.

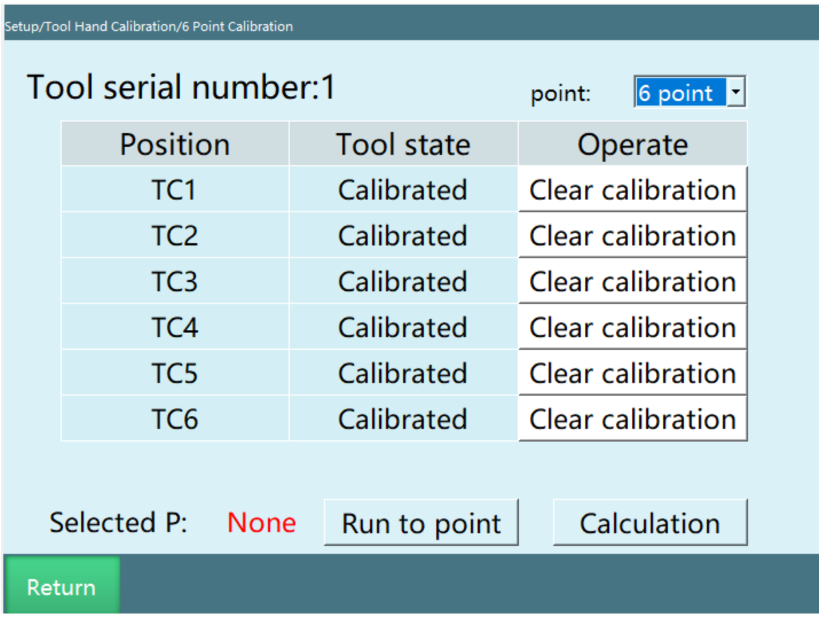
Point 2: The robot rotates 180° around the C-axis on the basis of the first point

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.

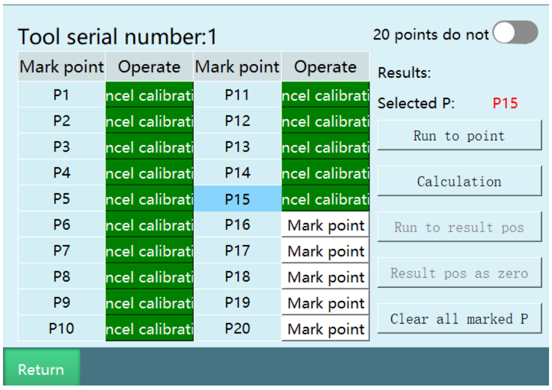
4.Click the [Return] button at the bottom to return to the "Tool hand calibration" interface.

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.

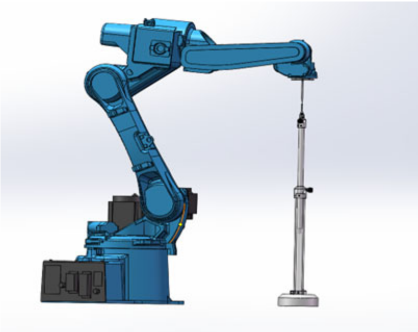


1.Find a reference point (the tip of the calibration cone is the reference point) and make sure this reference point is fixed.

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

The specific steps are as follows:

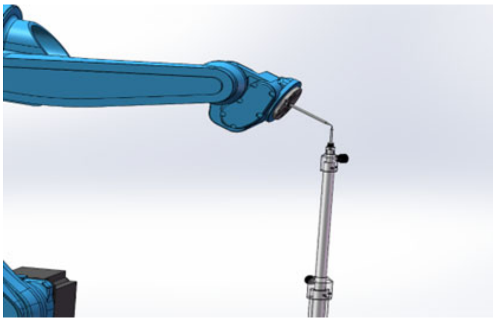
1.Point 1: The robot returns to the zero point, align the tip of the robot with the tip of the calibration cone through Cartesian coordinate system, and calibrate the first point;



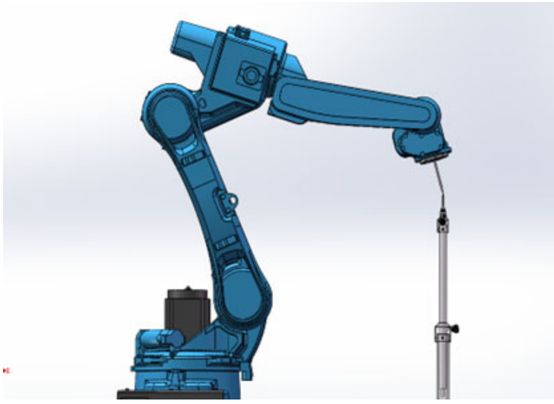
2.Point 2: On the basis of the first point, the robot rotates 180 degrees around the C-axis through the Cartesian coordinate system, align the tip of the robot with the tip of the calibration cone, and calibrate the second point;

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;

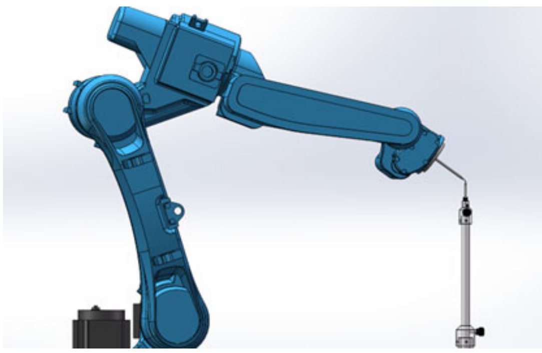


5.Point 5: On the basis of the fourth point, perform B+ movement through the Cartesian coordinate system, make J5>-90°, align the tip of the robot with the tip of the calibration cone, and calibrate the fifth point;

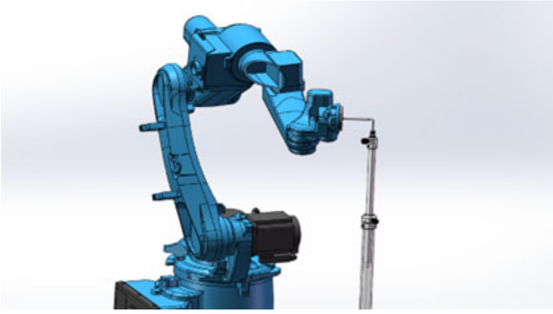


6.Point 6: Select the first point and move the robot to the first point, and on the basis of the first point, perform B+ movement through the Cartesian coordinate system, make J5>-90°, align the tip of the robot with the tip of the calibration cone, and calibrate the sixth point;

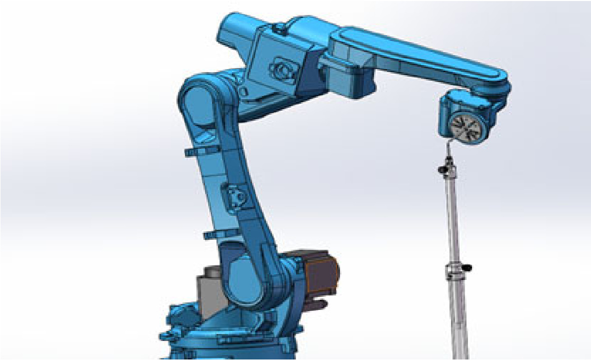
7.Point 7: On the basis of the first point, perform B- movement through the Cartesian coordinate system, make J5>-90°, align the tip of the robot with the tip of the calibration cone, and calibrate the seventh point;



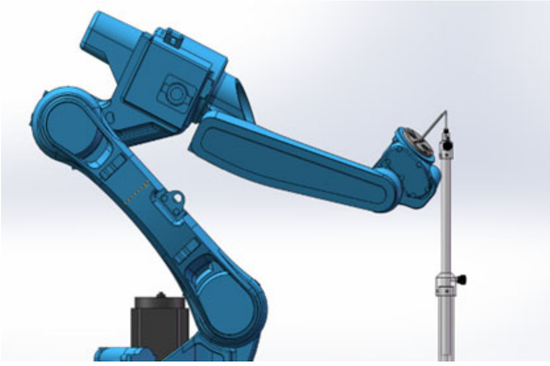
8.Point 8: On the basis of the seventh point, perform A+ movement through the Cartesian coordinate system, rotate by 90°and make J5>-90°, align the tip of the robot with the tip of the calibration cone, and calibrate the eighth point;



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;

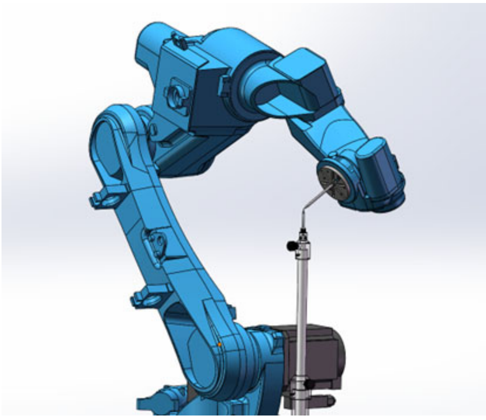


10.Point 10: The robot returns to the first point, jog axis 5 through the joint coordinate system to make axis 5 up and J5<-90°, align the tip of the robot with the tip of the calibration cone, and calibrate the tenth point;



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.

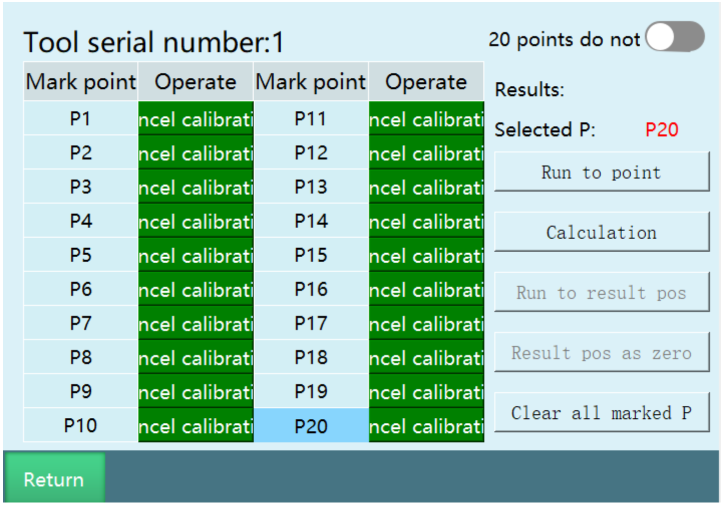
Please keep the reference point fixed during the calibration process, otherwise the calibration error will increase.

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.

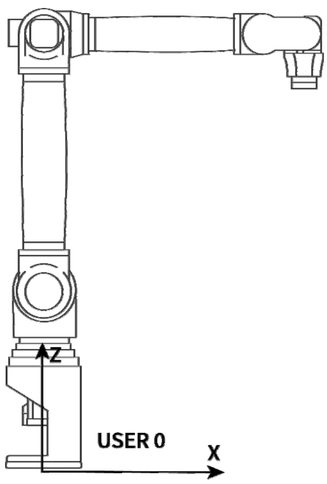
Please keep the reference point fixed during the calibration process, otherwise the calibration error will increase.

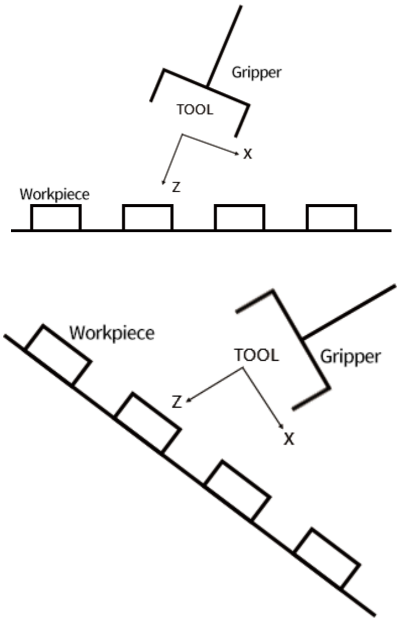
User coordinate system

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.

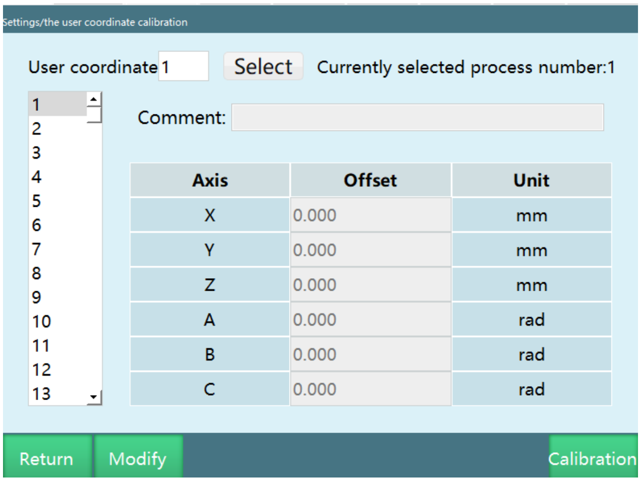
2.Determine the movement direction on the workbench for easy debugging.

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.



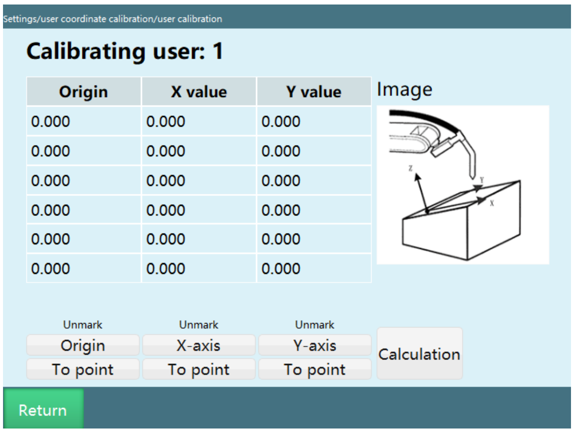
The parameters of the user coordinates are as follows

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

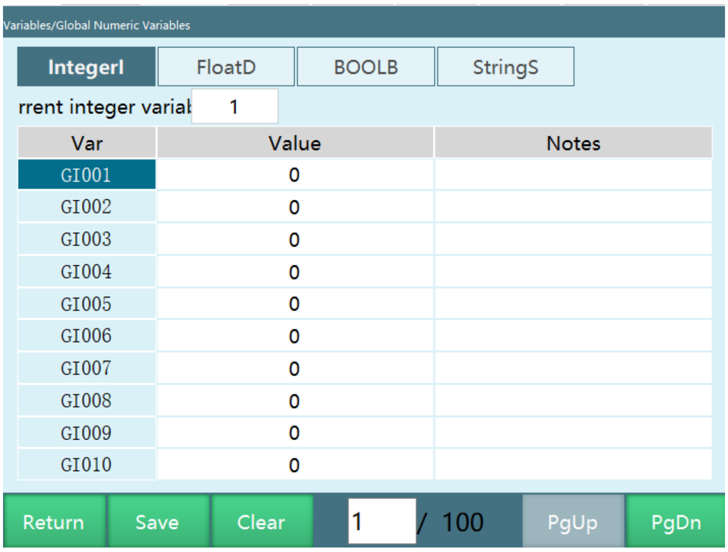
Global numerical variables are variables that can act on all programs of all robots. For example, program AA of robot 1 and program BB of robot 2 can use the same global numerical variable at the same time. This section will mainly explain the use of the global variable interface, as well as the use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

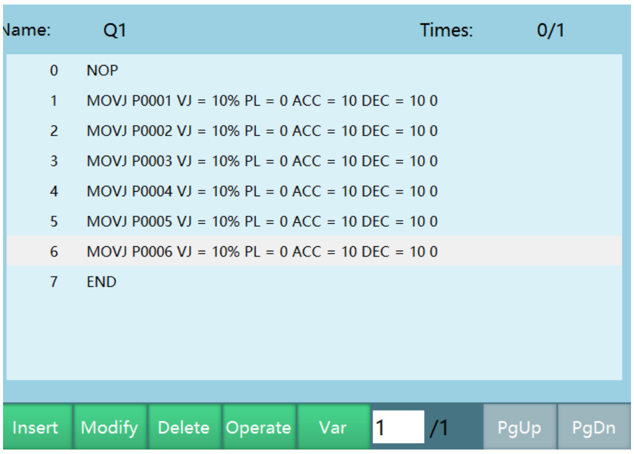
Select "Variable class" - "SET" and click "OK";

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.

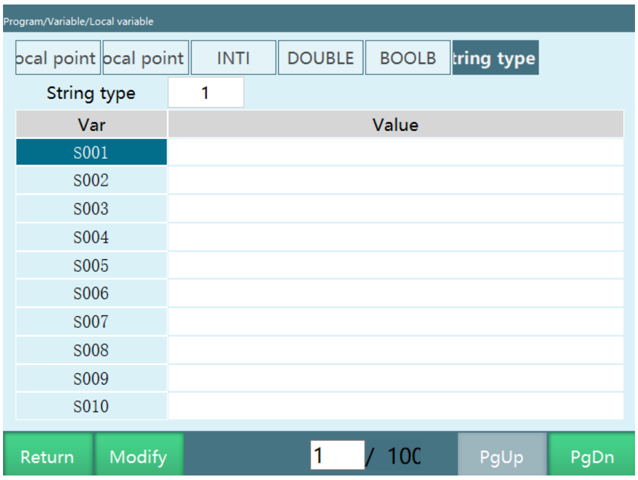
Local numerical variables

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.

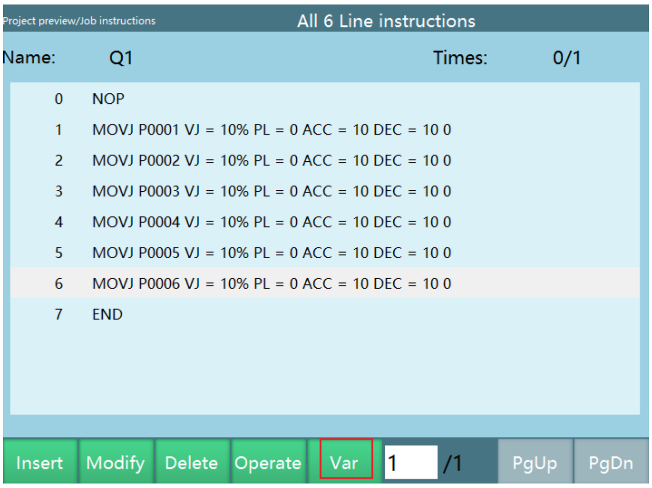
All the local numerical variables defined can only be used in the current program and cannot be used by other programs or background programs.

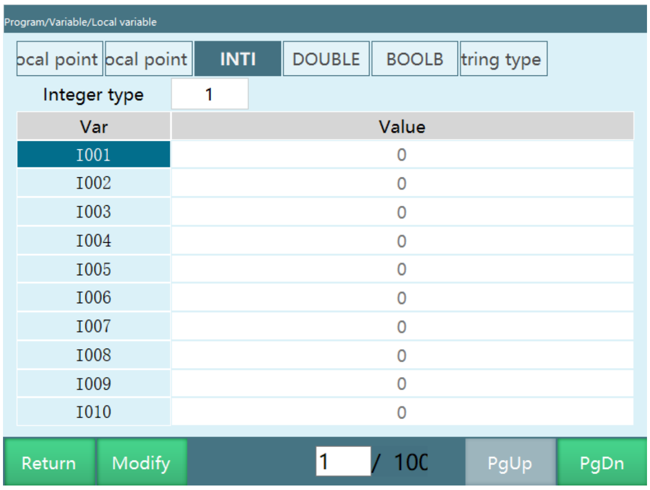


Use of local variables

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

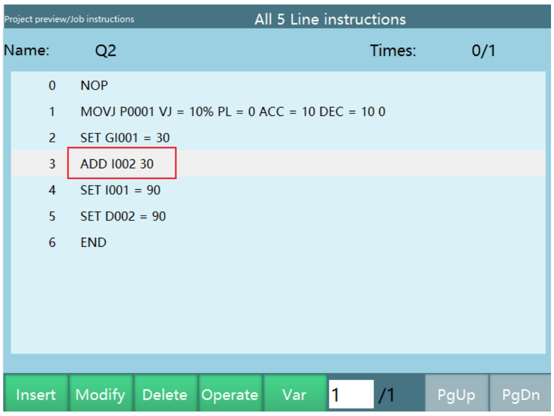
Local string variable can store all variable types and non-variable types, **such as: numbers, symbols, letters (including case), Chinese characters**

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.

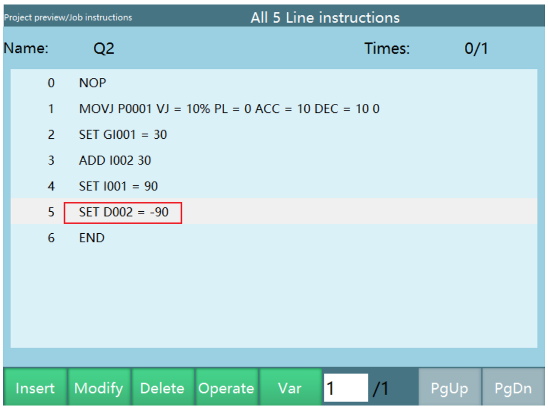
Assignment of values to local variables by calculation instructions

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



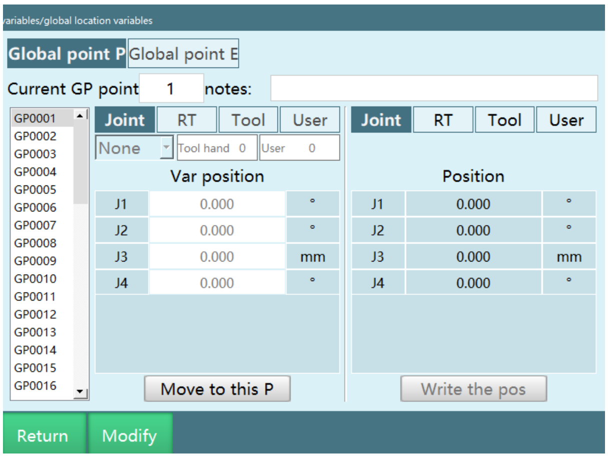
Position variables

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.

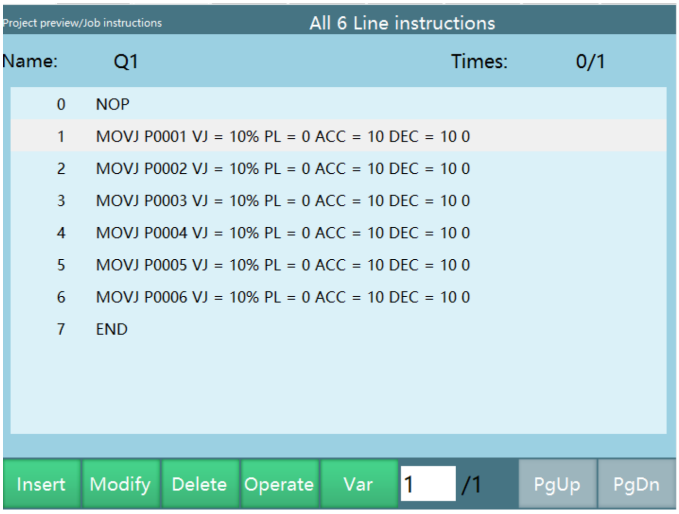
Local position variables

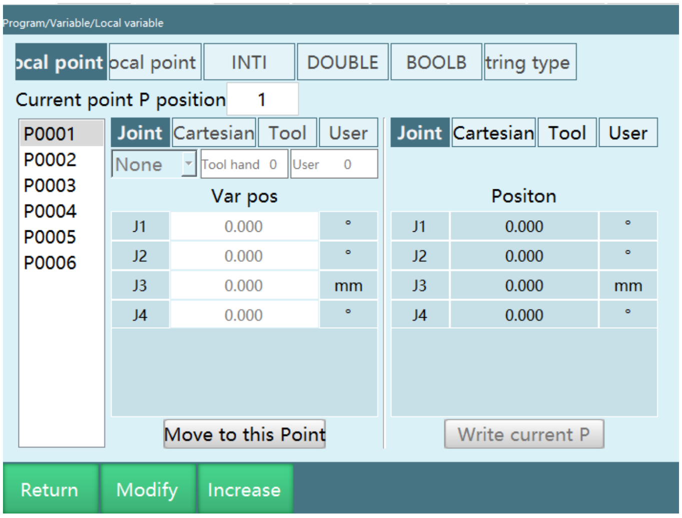
The local position variable (P000X) can only be used for a single job file and cannot be used across all job files.

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

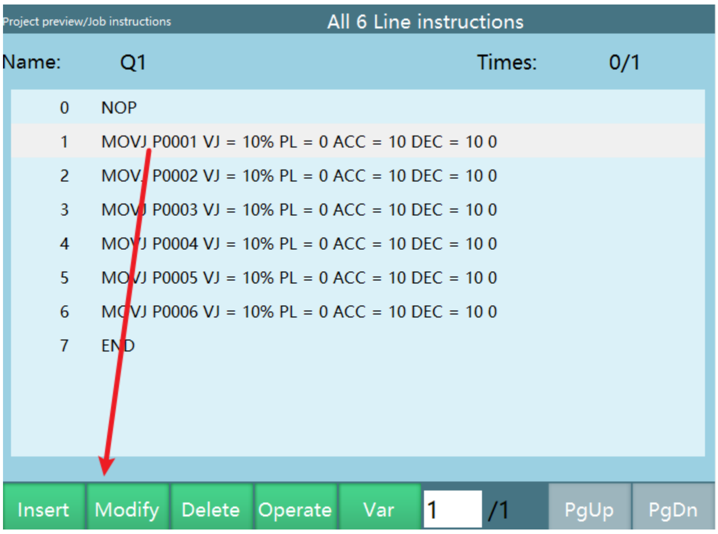


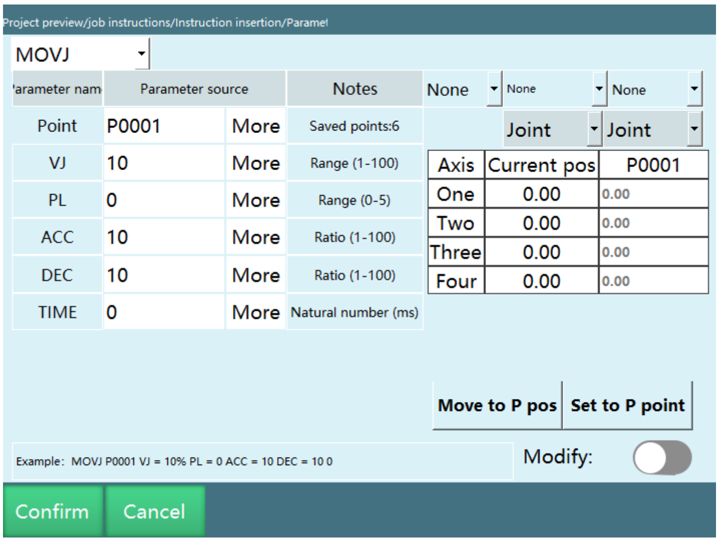


2.You can perform functions such as "Modify points", "Add points", "Run to this point", and "Write to current position" for local position variables

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.

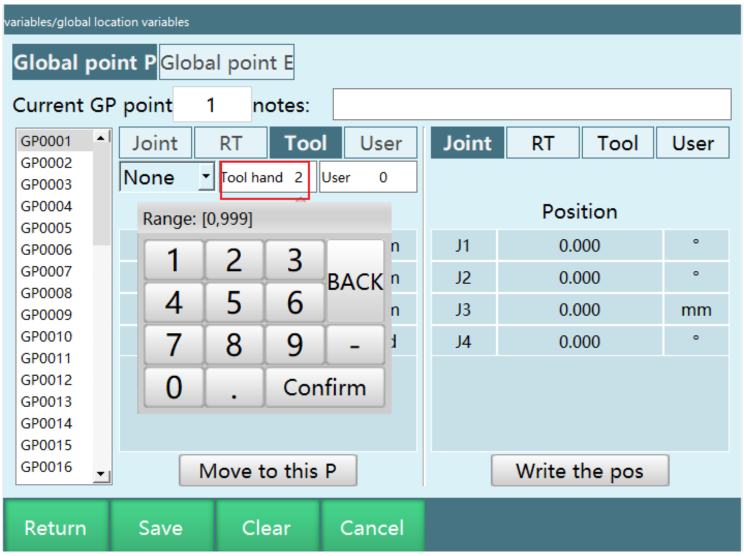
When the current point is selected, the robot will automatically calculate the form of the current point, and the form value corresponds to the interval in which the robot's 135 axes are located. For example: Form 3=010 (axis 1, axis 3, axis 5)+1=011, axis 1 is not within -90°~90°, axis 3 is within the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

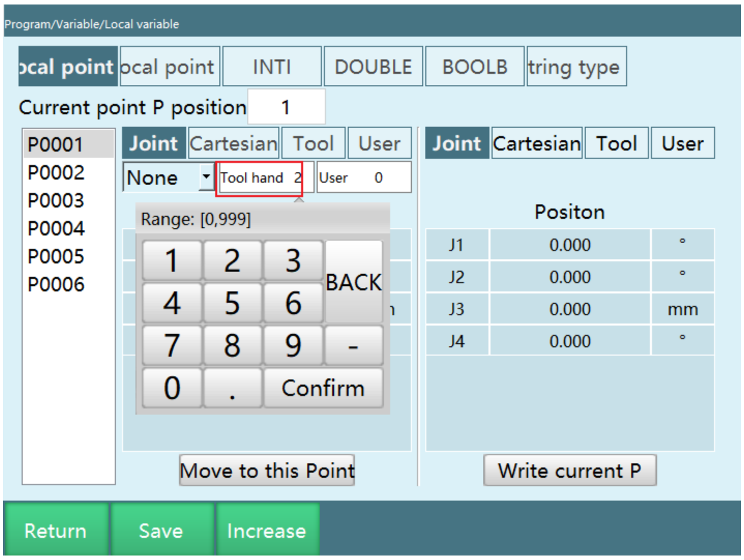
Tool hand parameters

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)



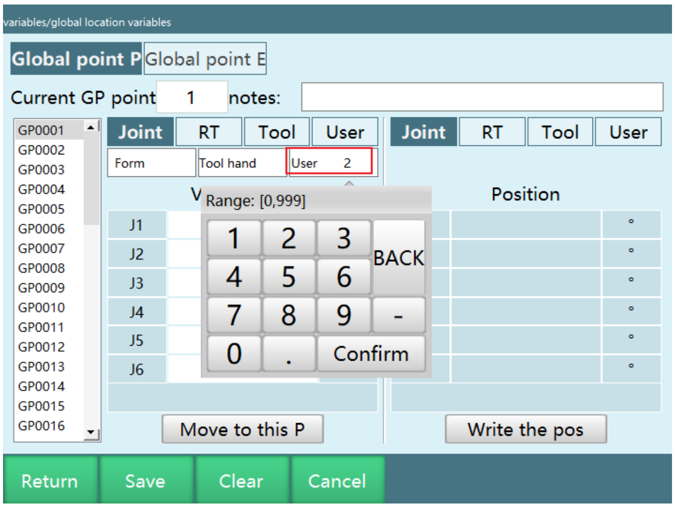


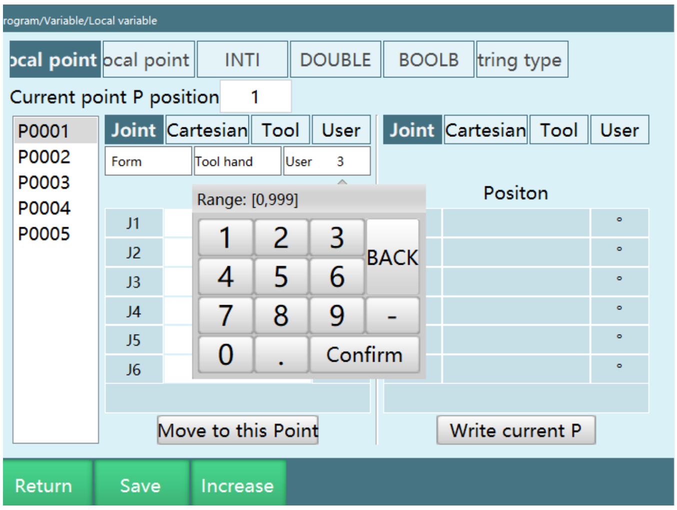
User coordinate parameters

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

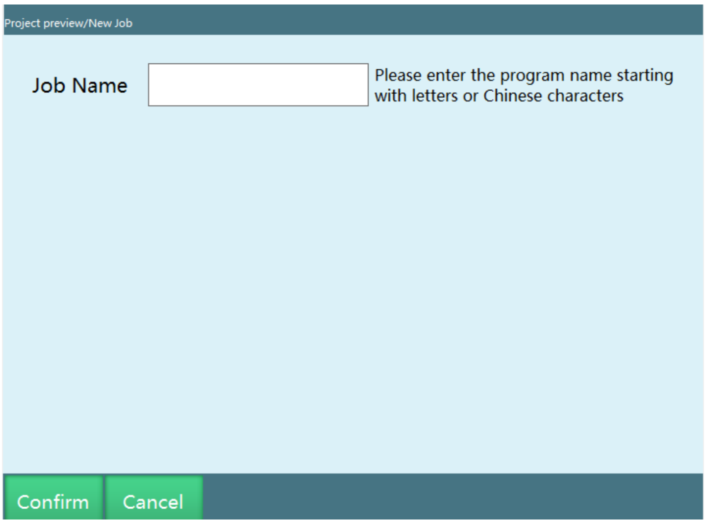
2.Click [Project] on the left

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

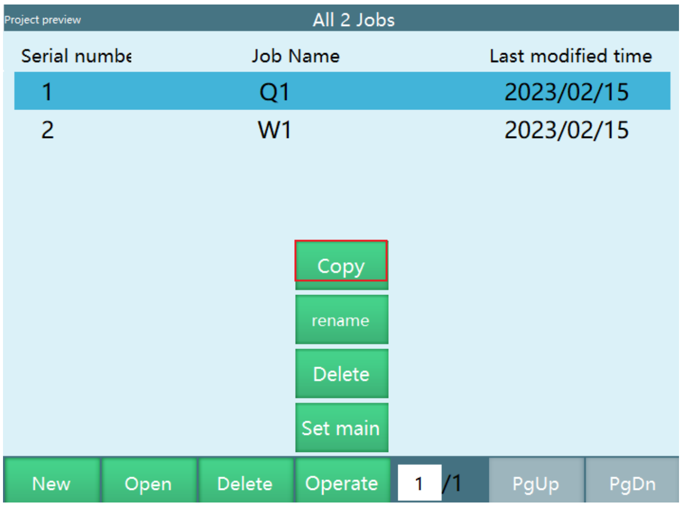
3.Click the [Open] button at the bottom, the program opens successfully

Copy program

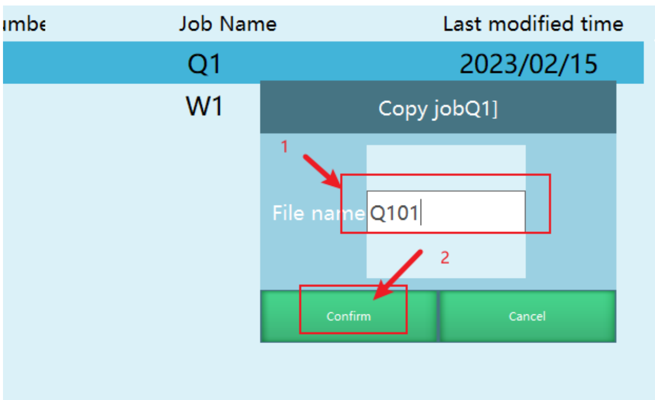
To copy an existing program, the user needs to perform the following steps:

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

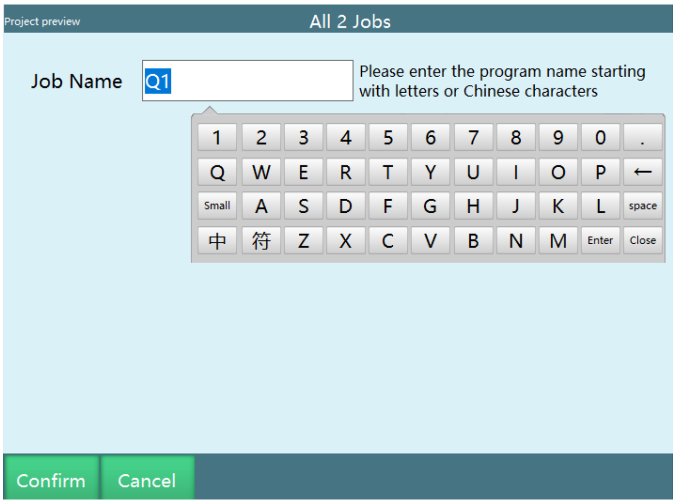
The rename operation can change the name of the selected program

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

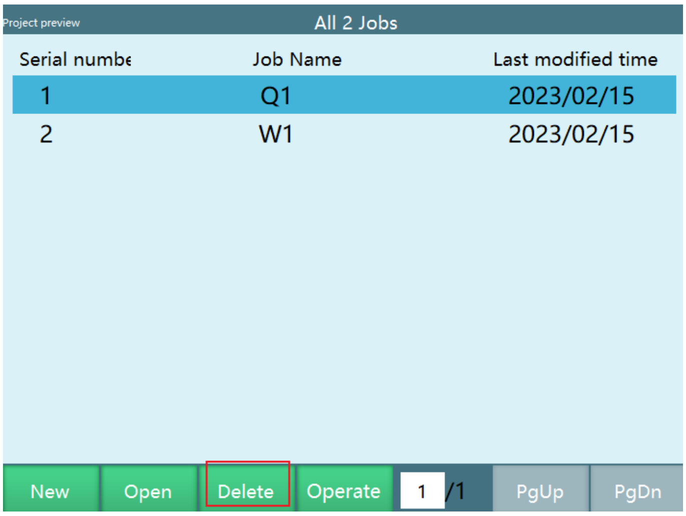
Delete program

The delete operation can delete the selected program

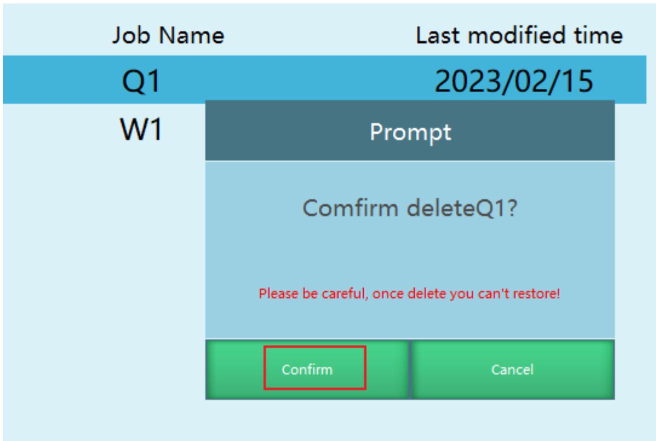
The relevant operation steps are as follows:

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

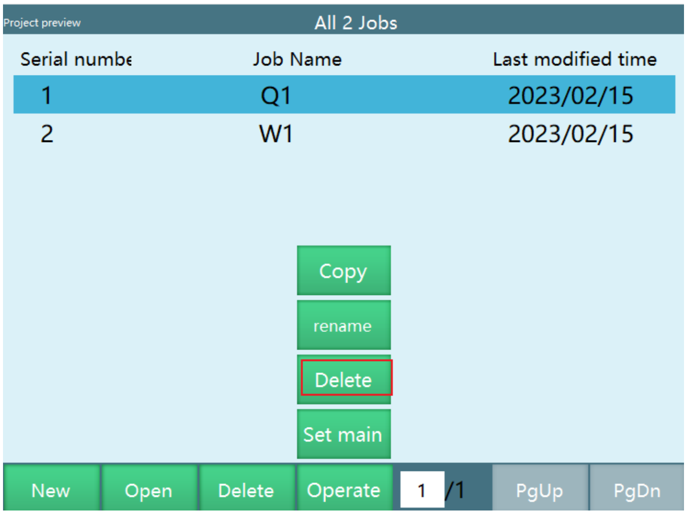


Batch delete

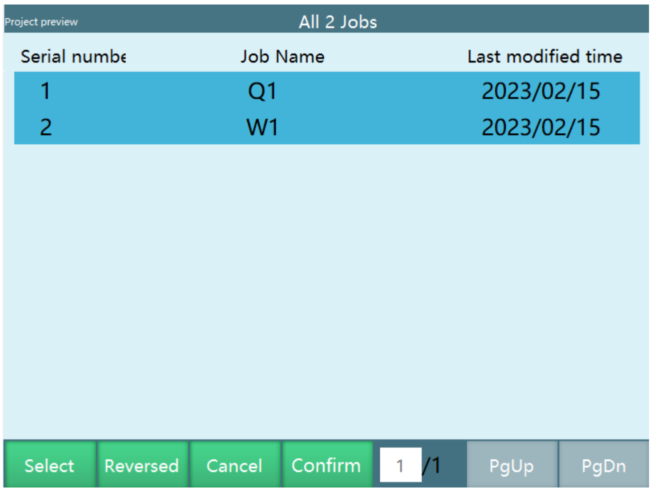
The batch delete function can delete multiple programs at one time. The method of use is as follows:

1.Click [Project]

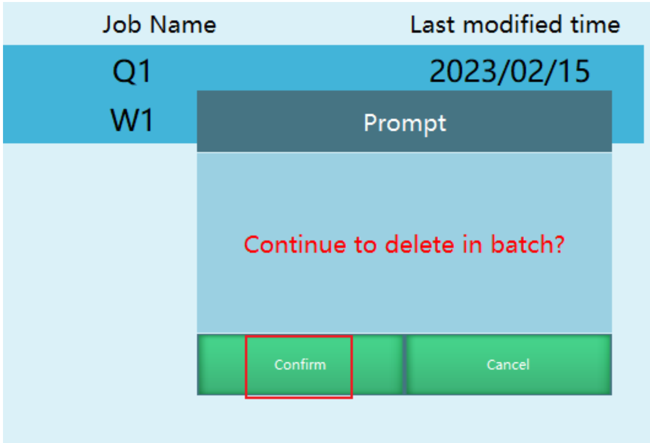
2.Click [Operation] on the bottom menu bar and select the [Batch delete] button



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

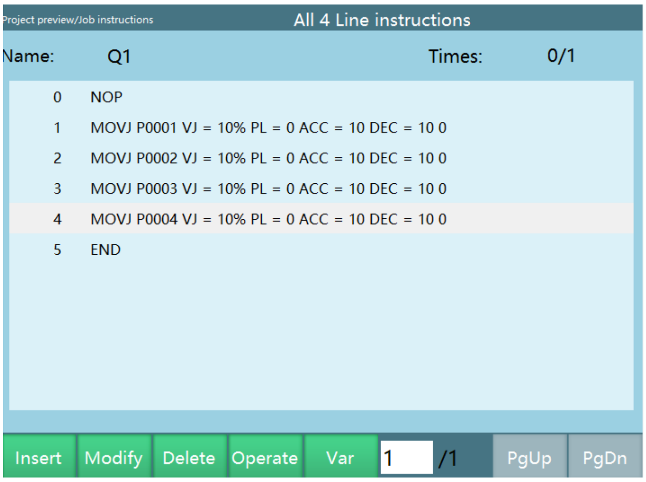
The relevant steps are as follows:

1.Switch to "Admin" account

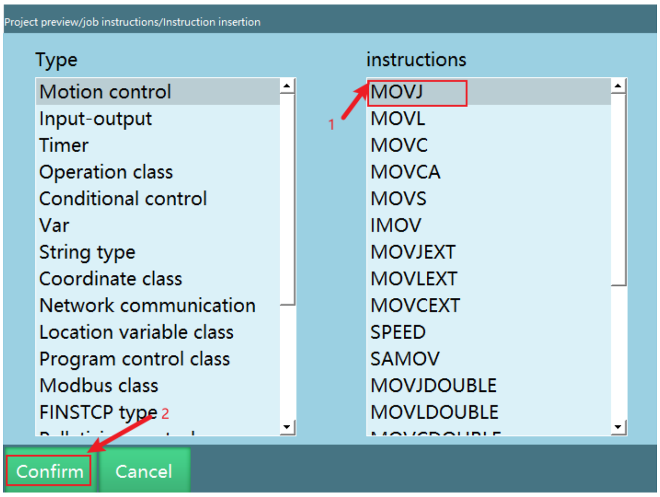
2.Click [Project] on the left

3.Click [New]

4.Enter the program instruction interface

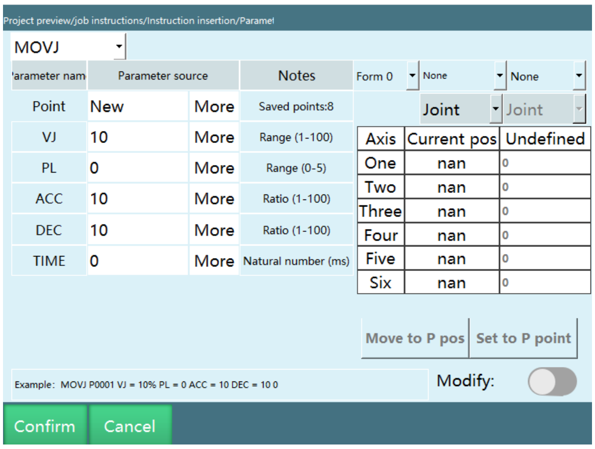


5.Click the [Insert] button, the "Instruction type" menu will pop up



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

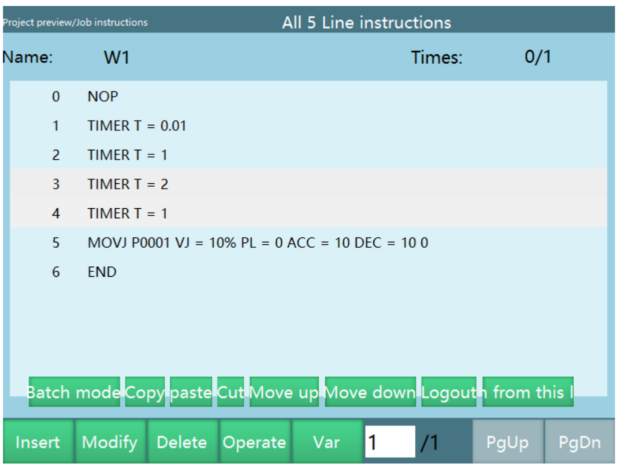
Modify instructions in batch mode or single-line mode

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

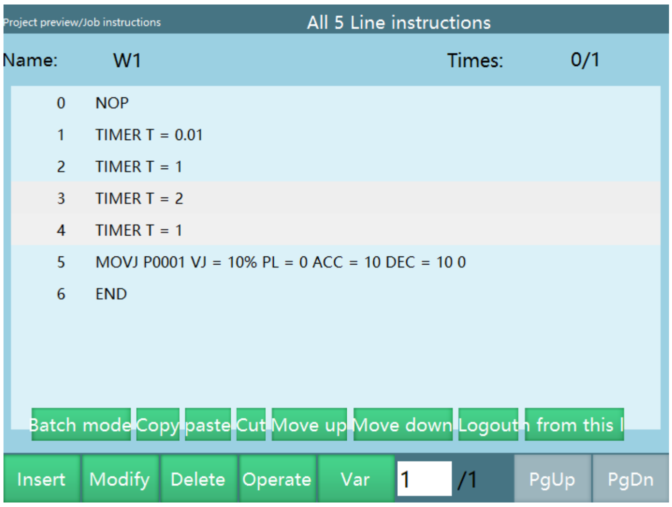
Ⅱ. If the user wants to batch copy, paste, cut, delete, modify, log out, move up, and move down the instructions across job files, take batch copy as an example, the steps are as follows:

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

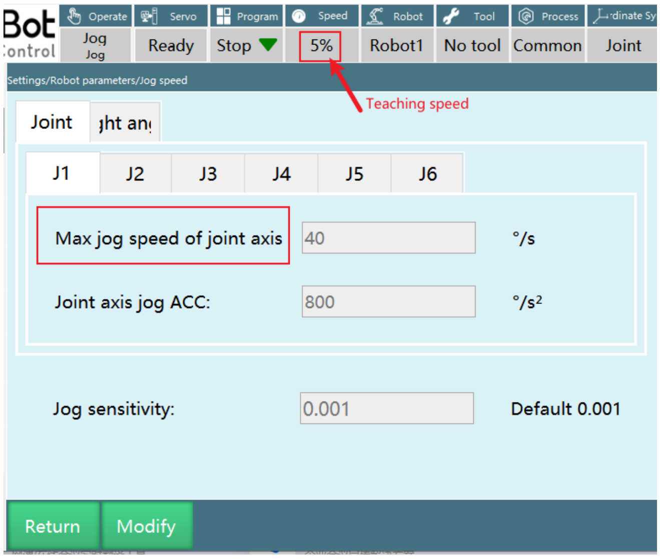
4.The selected line will move down automatically. If you want to step the next line of instruction, press the [Step] button again

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]**

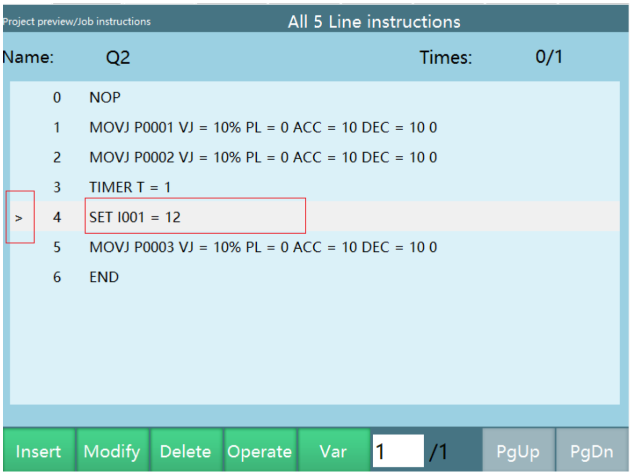
Notes

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

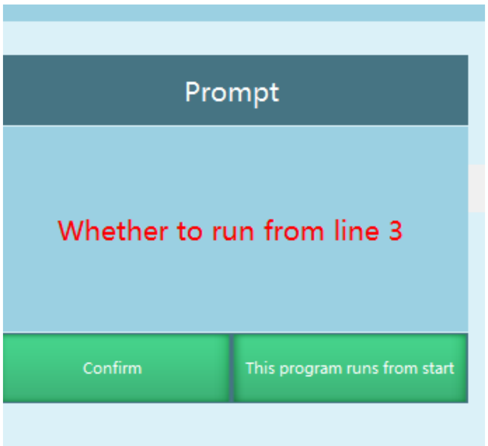
The running speed after using the global speed is: teach speed*instruction speed*global speed

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



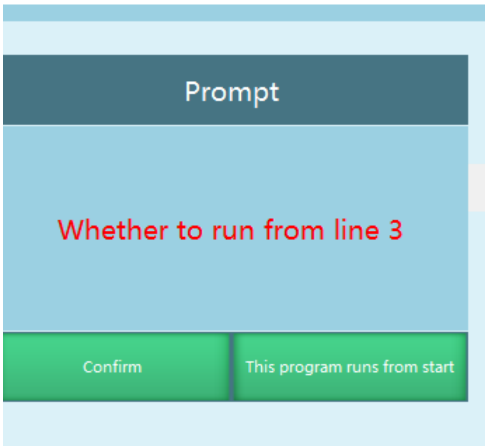
1.Switch to running mode, click [Start], there will be a prompt pop-up window when running



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

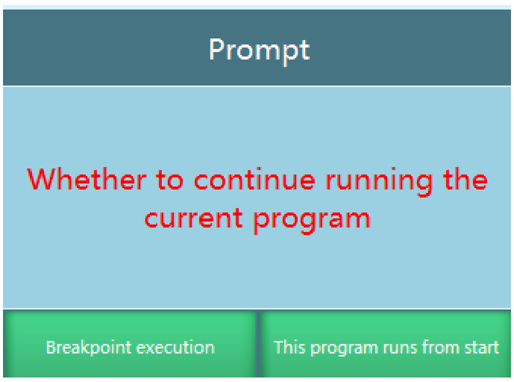
Jog the robot

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

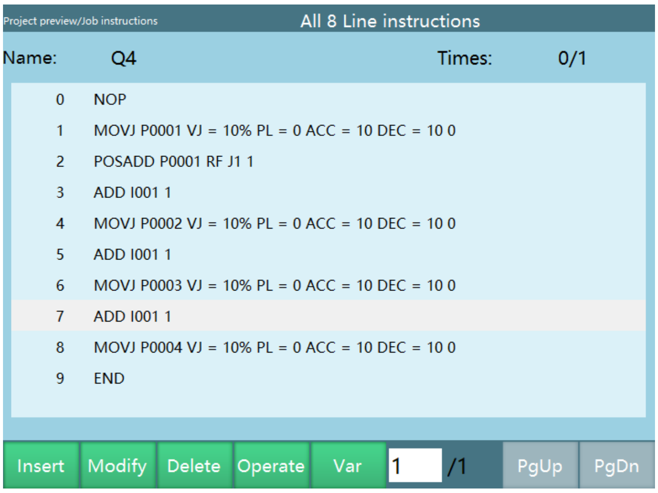
4.Error when running program instruction and power off

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.

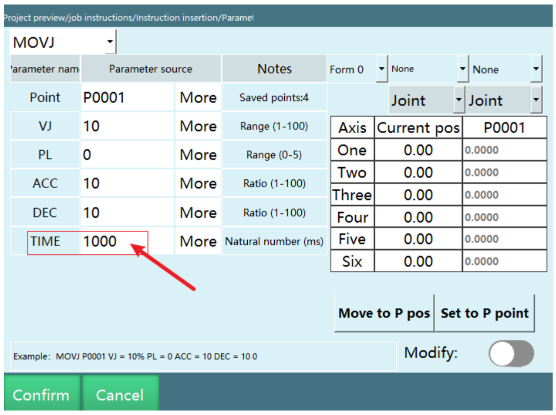
Example: The initial state of P0001 and I001 is shown in the figure, and it changes as follows during the operation: P0001 J1+1, I001+1.

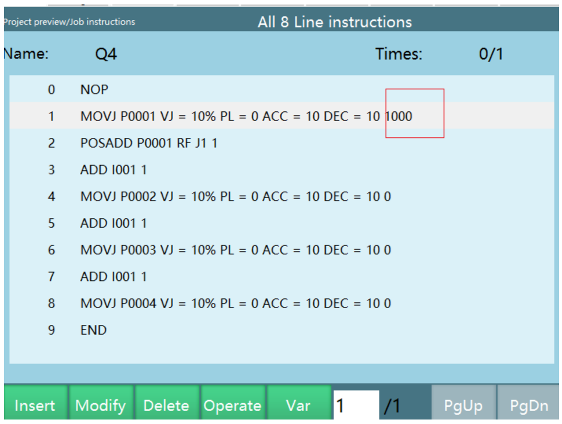


When running to the 6th line, P0001 J1=1, I001=2, a breakpoint occurs when switching to teach mode, after switching to teach mode, P0001, I001 is displayed as the initial value, at this time press [DEADMAN] to power on, it will be displayed as P0001 J1=1, I001=2, the initial value is restored after power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

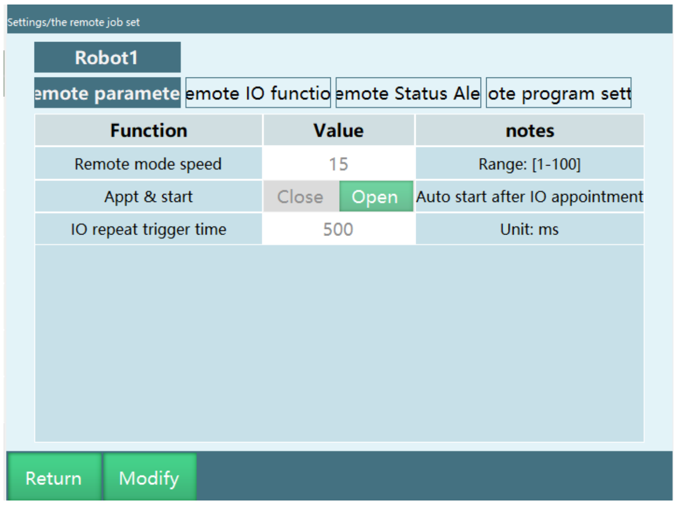
Remote mode speed

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

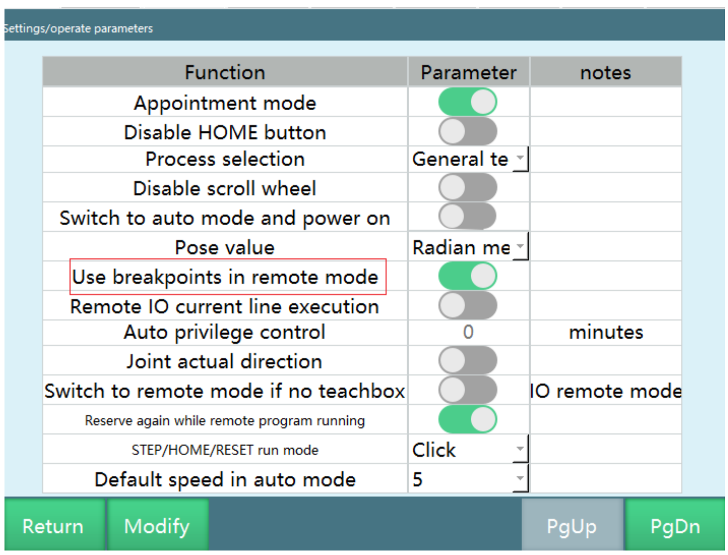


2..Click [Modify] to modify the remote mode speed

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