BETRUN

Operation Manual

V2.2022.01



www.Troysupply.com

Warnings

★ Before operating the robot, press the emergency stop key on the teach pendant, and confirm that the main servo power supply is cut off, and the motor is in the state of power loss and holding brake.

In case of emergency, if the robot cannot be braked in time, personal injury or equipment damage may be caused.

★ When the servo power supply is switched on after the emergency stop is removed, the servo power supply shall be switched on after the accident causing the emergency stop is removed.

The robot action caused by misoperation may cause personal injury accidents.

★ When teaching within the action range of the robot, please observe the following:

Keep looking at the robot from the front.

Follow the operating procedures.

The strain scheme when the robot suddenly moves to its own orientation is considered.

Be sure to set up a shelter in case.

The robot action caused by misoperation may cause personal injury accidents.

★ When carrying out the following operations, please confirm that there is no one within the action range of the robot and the operator is in a safe position:

When the robot control cabinet is powered on

When operating the robot with the teach pendant

During commissioning

During automatic reproduction

Inadvertently entering the action range of the robot or contacting with the robot may cause personal injury accidents.

Notes

★ To operate the robot, you must confirm:

Whether operators have received relevant training on robot operation

Have enough understanding of the motion characteristics of the robot

Have enough knowledge of the dangers of robots

Work without drinking

Did not take drugs that affect the nervous system and slow response

★ Before robot teaching, the following items shall be checked, and any abnormality shall be repaired in time or other necessary measures shall be taken.

Whether the robot moves abnormally

Whether the external auxiliary equipment associated with the robot is normal

★ After use, the teach pendant must be put back to its original place and ensure that it is firmly placed.

★ Prevent the accidental drop of the teach pendant from causing the misoperation of the robot, resulting in personal injury or equipment damage.

HUNAN Betrun Robot Co., Ltd.

Content

Part I Basic Programming and Operation 1

1 Welcome to the Betrun control system 2

1.1 System introduction 2

1.2 Main controller 3

1.3 Teach pendant 4

1.3.1 Scroll wheel functions 4

1.3.2 Key functions 5

1.3.3 System login permission 8

1.3.4 Operation modes 9

1.4 Control modes of robot 10

1.5 Setting of safety zone and interference zone 10

1.5.1 Safety zone 10

1.5.2 Interference zone 11

2 Coordinate systems 13

2.1 Five common coordinate systems of robot 13

2.1.1 Base coordinate system 13

2.1.2 Joint coordinate system 13

2.1.3 Tool coordinate system and flange coordinate system 14

2.1.4 User coordinate system 15

2.2 Establishment of tool coordinate system 15

2.3 Establishment of user coordinate system 18

3 Robot manual teaching 20

3.1 Enable button 20

3.2 Manual Teaching in Different Coordinate Systems 21

3.3 Manual teaching speed 23

3.4 Operational feedback 23

4 Online teaching programming 25

4.1 Contents of teaching programming 25

4.1.1 Motion trajectory 25

4.1.2 Operating conditions 25

HUNAN Betrun Robot Co., Ltd.

4.1.3 Operation sequence 25
4.2 "Point" in teaching programming 26
4.2.1 "Point" management 26
4.2.2 Creation of "point" 27
4.2.3 Display of current "point" 28
4.2.4 Home position 28
4.3 Online teaching a basic program 29
4.3.1 Task description 29
4.3.2 Create a basic program 30
4.3.3 Manual teaching for the program design 31
4.3.4 Program running 34
5 Betrun programming 35
5.1 Program instruction 35
5.2 Global variable definition 35
5.3 Programming and testing 36
5.3.1 New program 36
5.3.2 Manual teaching programming 36
5.3.3 Program notes 36
5.3.4 Program editing 37
5.3.5 Trajectory debugging 38
5.3.6 Modification of track points 40
5.3.7 Program saving 41
5.4 Programming interface menu function 42
5.5 Modular program 43
5.5.1 Use SUB_ CALL instruction 43
5.5.2 Create a "modular program" 44
5.6 Program execution in automatic mode 45
5.6.1 Program automatic positioning 45
5.6.2 Parameter setting 45
5.6.3 Use of keys 46
5.7 Run the program by using external keys 47
5.8 Execution of the reservation programs 48

HUNAN Betrun Robot Co., Ltd.

5.8.1 Reservation Settings 48
5.8.2 Execution of The Reservation Programs 49
5.9 Program management 50
6 PLC programming 52
6.1 PLC programming environment 52
6.1.1 Ladder component symbols 53
6.1.2 Basic logic instruction in instruction table 53
6.1.3 Run PLC program 54
6.1.4 Load PLC program 55
6.1.5 PLC program encryption 55
6.2 PLC component library 56
7 I/O communication of Betrun system 57
7.1 Interface signal definition 57
7.2 Standard I/O circuit 57
7.3 Basic port operation 58
7.3.1 Monitor relay port 59
7.3.2 Set output port 59
7.3.3 Set analog quantity 60
7.3.4 Monitor input port 60
8 Software configuration and use 62
8.1 Port configuration 62
8.1.1 Function port 62
8.1.2 Indication port 63
8.2 Custom alarm prompt 63
8.3 Other configurations 63
8.3.1 Emergency stop trigger output configuration 64
8.3.2 Calibration cycle configuration 64
8.3.3 Off screen time configuration 65
8.3.4 Abnormal braking detection 65
8.3.5 Network configuration 65
8.4 Language configuration 66
9 System backup and update 67

HUNAN Betrun Robot Co., Ltd.

9.1 System backup 67
9.1.1 Backup configuration 67
9.1.2 System program backup 68
9.1.3 User program backup 69
9.1.4 PLC program backup 70
9.2 System update 71
9.2.1 View system version 71
9.2.2 Update configuration 72
9.2.3 System program upgrade 73

Part II Function Package Instructions 74
1 Position searching and tracking in welding 75
1.1 Preparation 75
1.1.1 Preparation for electric position searching 75
1.1.2 Preparation for laser position searching 75
1.1.3 Direct and indirect position searching 77
1.2 Laser direct position searching 78
1.2.1 Get the searching position point 78
1.2.2 Point replacement 79
1.2.3 Intersection point calculation 80
1.2.4 Searching-position point operation 81
1.2.5 Position stretch 81
1.3 Indirect electric position searching 82
1.3.1 Parameter setting 83
1.3.2 Electric position searching process 85
1.3.3 Program case 87
1.4 Indirect laser position searching 88
1.4.1 Parameter setting 88
1.4.2 Laser position searching process 88
2 Interval Welding 90
2.1 Interval welding instructions 90
2.2 Function modes 90

HUNAN Betrun Robot Co., Ltd.

2.2.1 Space mode 91
2.2.2 Time mode 91
2.2.3 Reciprocating mode 92
2.2.4 Stay mode 93
3 Weaving 94
3.1 Create weaving function package 94
3.2 Program case 95
3.3 Parameter test 95
3.4 Swing in place 96
4 Cooperative movement 99
4.1 Create collaboration function package 99
4.1.1 Preparation 99
4.1.2 Establish tool coordinate system 100
4.1.3 Create the calibration points 100
4.1.4 Create the cooperation package 101
4.2 Use of the cooperation function package 101
5 Tracking 103
5.1 Hardware preparation 103
5.2 Create ontology tracking function package 103
5.2.1 No encoder 104
5.2.2 Pulse encoder 105
5.2.3 No trigger port 106
5.3 Create slide tracking package 108
5.3.1 No encoder 108
5.3.2 Pulse encoder 108
5.4 Use of tracking function package 108
6 Collision detection 109
6.1 Program running parameter sampling 109
6.2 Create collision detection function package 110
6.3 Use of collision detection function package 110
6.3.1 Use collision detection instruction 110
6.3.2 Start collision detection in automatic mode 111

HUNAN Betrun Robot Co., Ltd.

Part III Process Package Instructions **112**
1 Palletizing **113**
1.1 Preparation 115
1.1.1 Create user coordinate system 115
1.1.2 Workpiece fixture control 115
1.2 Create palletizing process package 116
1.2.1 Coordinate system selection 116
1.2.2 Reference point setting 116
1.2.3 Layout method setting 117
1.2.4 Layer information setting 118
1.3 Generate palletizing program based on "template" 119
1.4 Modification of palletizing program 121
1.4.1 Global variable definition 121
1.4.2 Modify reference points 122
1.5 Offset setting 123
1.5.1 Pallet offset setting 123
1.5.2 Transition point offset settings 123
2 Welding **124**
2.1 System composition 124
2.2 Create welding process package 124
2.2.1 Welding machine configuration 125
2.2.2 Time setting 125
2.2.3 IO port configuration 127
2.2.4 Basic parameter setting 127
2.2.5 Calibration 128
2.3 Welding program design 128
3 Spraying **130**
3.1 Create spraying process package 130
3.1.1 Motion parameter setting 130
3.1.2 Control parameter setting 135
3.2 Generate spraying basic program 136

HUNAN Betrun Robot Co., Ltd.

Part IV Modbus Communication **137**
1 Modbus communication hardware connection **138**
2 Modbus communication protocol **138**
2.1 Frame format 139
2.2 Communication process 139
2.3 Data unit (PDU) description 140
3 Modbus function code and data type **141**
3.1 Function code definition 141
3.2 Data type mapping for Betrun system as a slave 141
3.3 Custom function code for Betrun system as a slave 143
4 Modbus Communication for Betrun system as a slave **144**
4.1 RS232 communication settings 145
4.2 TCP/IP communication settings 148
4.3 Write global variable 149
4.4 Read global variables 151
5 Modbus Communication for Betrun system as a master **152**
5.1 Communication settings 152
5.2 Read slave data 153
5.3 Write slave data 155
6 Robot motion control based on Modbus **156**
6.1 Betrun program control special register 156
6.2 Motion control method 158

Part V Application of Coordinate Systems **161**
1 Coordinate system transformation instruction **162**
2 Program reuse among multiple robots **164**
3 Quick correction of tool coordinate system **167**

Appendix 1 Instruction description **169**
Appendix 2 Definition of global variables **175**
Appendix 3 Definition of PLC components **178**

HUNAN Betrun Robot Co., Ltd.

Part I Basic Programming and Operation

1 Welcome to the Betrun control system

This chapter introduces the characteristics and application scenarios of Betrun control system, explains the composition of system hardware and software, and the way of using the system to control robot motion.

1.1 System introduction

Betrun control system is independently developed by Hunan Betrun robot Co., Ltd. and has completely independent intellectual property rights. It can customize intelligent factory solutions according to customer needs.



Figure 1.1 Appearance of Betrun control system

The appearance of the system is shown in Figure 1.1, which mainly includes two parts: main controller and teach pendant. The main controller is installed in the robot control cabinet, and the teach pendant is placed on the robot control cabinet. The basic technical parameters are shown in Tables 1.1 and 1.2.

The system currently supports:

- Robot: 4-axis / 6-axis robot
- Maximum number of supported axes: 11
- Driver: pulse, Ethercat bus, mixed

System features:

- Hardware: the embedded system design scheme is adopted, which has the advantages of small volume, low power consumption and fast response to system maintenance and upgrading.
- Motion performance: the control cycle is 2ms, the deviation between the actual motion trajectory and the ideal motion trajectory of the manipulator is small, and the motion process is stable and flexible.
- Operation: it adopts graphical man-machine operation interface and programming mode, which is simple to use and easier for customers to accept.

Table 1.1 technical parameters of Betrun controller

Number	Item	Specification description
1	Control hardware	Multiprocessor system 8-way pulse drive interface, MODBUS bus, dual network port, USB Expansion board Flash memory for mass storage 22 optocoupler isolated inputs, 10 collector open circuit outputs and 8 brake outputs
2	Control software	Object oriented design Betrun software products Pre installed software
3	Input voltage	220V~50Hz、24V

Table 1.2 technical parameters of teach pendant

Number	Item	Specification description
1	Material	Reinforced plastic shell
2	Quality	1kg
3	Operation	8-inch touch screen + Physical keys + Scroll wheel + Touch pen
4	Operating habits	The left hand passes through the hand rope and gently presses the enable button.
5	Language	At present, in addition to Chinese, it also supports English, Russian, Korean and Vietnamese.

1.2 Main controller

The main controller is composed of CPU, action control circuit, storage circuit, bus circuit, input / output (I / O) circuit, power supply circuit, etc., which is loaded with core function software package to complete the functions of robot motion control, auxiliary equipment control, input / output response, external communication and so on.

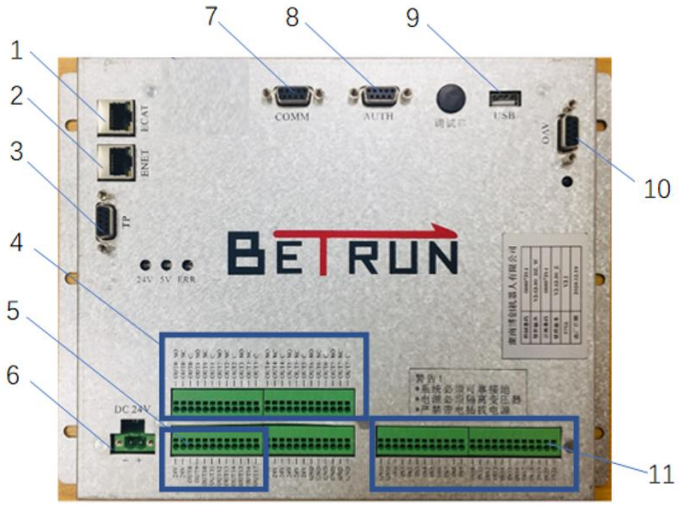


Figure 1.2 Appearance of the main controller

1——The printing is "ECAT", which is the EtherCAT bus interface used to connect the robot

Betrun Robot Co., Ltd.

servo driver

2——The printing is "ENET", which is the network port used for Modbus Communication in TCP / IP mode.

3——The screen printing is "TP", which relates to the teach pendant.

4——8-way relay output port

5——10 open collector output ports

6——24V DC input

7——The printing is "COMM", which is the RS485 and RS232 communication interfaces.

8——Reserved interface

9——USB 2.0 interface used to update and back up the system program

10——The printing is "AVO", which is the analog output interface. (Additional options are required.)

11——22 optocoupler isolation input ports, of which the last two are used as phase A and phase B ports of external encoder (used in pipeline tracking function)

1.3 Teach pendant

Teach pendant is a device to facilitate operators to complete the control of robot system online. The subsequent contents of this manual are based on the teach pendant.

Betrun teach pendant has the following specific features:

- It adopts 8-inch touch screen and is equipped with a touch pen. When wearing gloves, it can operate the interface through the touch pen;
- More abundant keys are provided for convenient operation;
- Equipped with a scroll wheel, which greatly improves the efficiency of manual teaching programming.

1.3.1 Scroll wheel functions

The scroll wheel functions are different in two working modes: focus mode and scroll mode.

The Default mode is focus mode. In focus mode, you can open a basic program with the scroll wheel.

- You can quickly switch the highlight of the control on the software interface by rolling the scroll wheel;
- Pressing the scroll wheel is equivalent to the Enter key;
- When a control with a list is highlighted, press the scroll wheel to enter the control list and quickly switch by rolling the scroll wheel. When you press the scroll wheel again, you will return to the control switch.

Press and hold the scroll wheel for 2 seconds to switch to the scroll mode. For all units with scroll bars, you can scroll through the content.

- In the main interface of teach pendant, you can use the scroll wheel to find the function icon;

HUNAN Betrun Robot Co., Ltd.

- In the programming interface, you can use the scroll wheel to browse the program line.

Press and hold the scroll wheel for 2 seconds again to switch to the focus mode.

1.3.2 Key functions

Teach pendant is a hand-held programmer for robot online operation. It can realize various operation and display functions required by robot programming and control without mouse and keyboard.

The main key functions on the front and back of the teaching pendant are described as follows.



Figure 1.3 Front appearance of the teach pendant

The physical keys on the front of the teach pendant are divided into three types, and the function definitions are shown in Table 1.3.

Table 1.3 Functions of physical keys on the front of the teach pendant

Type	Number	Function description
Program execution keys	1	Mode switch It is used to switch between manual mode and automatic mode.
	2	"Program stop" key in automatic mode It can stop the running program and the robot stop at the current position. Then press the "program run" key, and the program will return to the first instruction line and start running.
	3	"Program pause" key in automatic mode It can make the running program pause in the current position line and the robot stop at the current position. Then press the "program run" key, and the program will continue to run from the current

		instruction line.
	4	"Program run" key in automatic mode
	5	"Step forward" key in automatic mode Under "single motion" mode: select a curve below the current position as the target line, and then click "step forward", it will go and pause on the target line. Click the "Program stop" key to exit. Under "single step motion" mode: click "step forward" to execute one line of instructions downward. The motion mode cannot be switched under the s paused state.
	6	"Step back" key in automatic mode Under "single motion" mode: select a curve above the current position as the target line, and then click "step back", it will go reversely and pause on the target line. Click the "Program stop" key to exit. Under "single step motion" mode: click "step back" to execute one line of instructions upward. Non-curves are not executed during a step back. The motion mode cannot be switched under the s paused state. Be careful: (1) It has a high security level. After logging in the system each time, you need to manually turn on the function according to the configuration interface. (2) It is unavailable in the following scenarios: no reverse movement in subprogram; no reverse movement during offset; no reverse movement during stacking; no reverse movement during collaborative process; ……
	22	"Emergency stop" key It can be used to stop all operations of the system and servo in case of "emergency".
Software editing keys	7	"HOME" key It is mainly used to return to the home page from the subpage, including the exit of the modal dialog box.
	8	"Shift" combination key You can customize the function through system - user defined key.
	9	F1: the default is "move focus forward" in the roller focus mode. F2: The default is "move focus backward" in the wheel focus mode. You can customize the function through system - user defined key.
	10	I / O key: basic port debugging. Press once to pop up and again to hide.
	11	"Menu" key You can pop up a box on the touch screen for corresponding query and setting operations.
	12	"Tab" key You can switch the operation history box/ configuration and

N Betrun Robot Co., Ltd.

		monitoring box. "Tab" combination key You can customize the function through system - user defined key.
Manual teaching Keys	13	"-V, +V" keys in manual mode They are used to increase or decrease the teaching speed according to a certain step. The step is divided into four gears according to the current speed: 0.01, 0.1, 1 and 10.
	14	"J6 -" and "J6 +" keys in manual mode After pressing "enable", J6 joint motion can be controlled in the joint coordinate system (C attitude adjustment in the base coordinate system).
	15	"J5 -" and "J5 +" keys in manual mode After pressing "enable", J5 joint motion can be controlled in the joint coordinate system (B attitude adjustment in the base coordinate system).
	16	"J4 -" and "J4 +" keys in manual mode After pressing "enable", J4 joint motion can be controlled in the joint coordinate system (A attitude adjustment in the base coordinate system).
	17	"J3 -" and "J3 +" keys in manual mode After pressing "enable", J3 joint motion can be controlled in the joint coordinate system (z-axis direction movement in the base coordinate system).
	18	"J2 -" and "J2 +" keys in manual mode After pressing "enable", J2 joint motion can be controlled in the joint coordinate system (y-axis direction movement in the base coordinate system).
	19	"J1 -" and "J1 +" keys in manual mode After pressing "enable", J1 joint motion can be controlled in the joint coordinate system (x-axis direction movement in the base coordinate system).
	20	"EXT" key in manual mode Switch to additional axis teaching mode.
	21	Coordinate system switching key in manual mode It is used to switch the currently used teaching coordinate system (Joint Coordinate System → Base Coordinate System → User Coordinate System →Joint Coordinate System). Tool Coordinate System needs to be selected at the same time sometimes.

N Betrun Robot Co., Ltd.

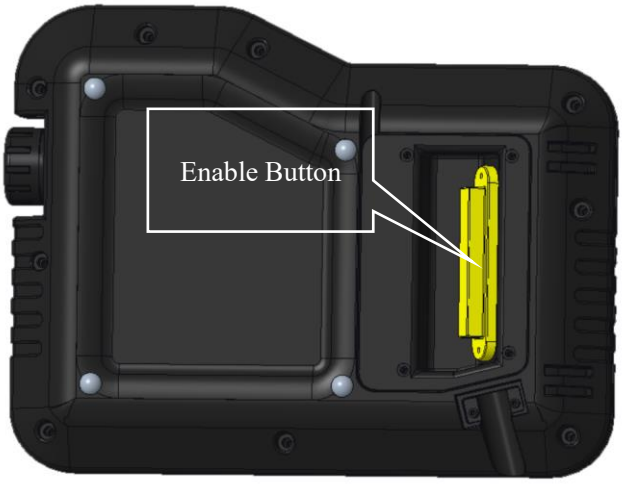


Figure 1.4 Rear appearance of teach pendant

An enable button is used on the back of the teach pendant. During manual teaching, the left hand passes through the hand rope and gently presses the button.

This enable button has three states:

State 1: when fully released, the driving device of the robot is cut off;

State 2: when gently pressed, the driving device of the robot is connected;

State 3: when pressed hard, the driving device of the robot is cut off.

1.3.3 System login permission

Turn on the power switch of the robot control cabinet and the human-computer interaction interface of the teach pendant starts.

The login interface will appear on the teach pendant screen, which is divided into three login permissions: administrator, operator and viewer, as shown in Figure 1.5.

Different types of operators have different permissions. The specific permissions are shown in Table 1.4.

Three different types of users need to check the corresponding type and log in through user name and password.



Figure 1.5 Login interface

HUNAN Betrun Robot Co., Ltd.

Table 1.4 User permissions

Number	Type	User permissions
1	Administrator	● Robot control system programming teaching ● Debugging of peripheral equipment ● Manage the permissions of different operators in the system ● Maintenance of robot control system ● Robot system power on / off operation ● Start and stop of robot program ● Recovery of system alarm state (part of the work needs to be carried out in the robot work area and must receive professional training for robots.)
2	Operator	After logging in the "Administrator" mode, set the "user name" and "password" of the operator mode in [user] - [user management]. ● Robot system power on / off operation ● Start and stop of robot program ● Recovery of system alarm state (only work outside the robot working area.)
3	Viewer	You do not need to set "user name" and "password" when logging in. You can "log in" directly. You can only "Browse" the system and can't do anything else.

1.3.4 Operation modes

There are two operation modes of using the teach pendant: manual mode and automatic mode. The switching between the two modes is realized by the "mode switch" button on the teach pendant.

Manual mode

Carry out "manual teaching" for the robot, and carry out parameter setting, program design, configuration of various functions and process packages, system management, etc., which are realized interactively in the manual mode interface.

In manual mode, the teach pendant can be used to control the movement of the robot by using 6 groups of manual control physical keys on the right side of the front while pressing the "enable" button on the back, to realize the manual operation of single joint movement and linear movement of the robot at different movement speeds.

Automatic mode

It is mainly used for program execution.

The robot performs single step, single or cyclic motion according to the program setting and the set motion speed.

1.4 Control modes of robot

Based on Betrun control system, the robot has several different control modes, as shown in Table 1.5.

Table 1.5 Robot control modes

Control mode	Method	Speed
Manual operation of teach pendant	In the manual mode of the teach pendant, 6 groups of control physical keys are used to control the single axis or linear motion of the robot	● The maximum speed of joint movement is 600r/min ● The maximum speed of linear motion is 200 mm/s ● The maximum rotation speed in the tool coordinate system is 50 degrees/second When manually controlling the physical key operation, the default is 10% slow operation on this basis
Manual fast operation	In the manual mode of the teach pendant, use the interface software keys to test the program	● Run at the speed set by the program 6 groups of physical keys cannot be controlled manually
Run in automatic mode	In the automatic mode of the teach pendant, use the interface software keys to run the program	● Run at the speed set by the program 6 groups of physical keys cannot be controlled manually
External control	When there is no teach pendant, the external key / upper computer communication mode controls the program to run	● Run at the programmed / set speed

1.5 Setting of safety zone and interference zone

1.5.1 Safety zone

In the process of industrial robot running, it is often necessary for the robot to return to the starting point of program curve or zero point from a certain position. This motion process does not consider the surrounding environment of the robot, but the control system calculates the motion trajectory by point motion control. In some cases, if the operator does not consider the position of the workpiece, the machine collision accident will occur due to mis operation.

In order to ensure that the robot does not touch other objects and is in a safe state when starting the robot from the current position and posture to the program starting position and posture, it is necessary to set the safe starting area of the robot through the teaching pendant after the installation of the robot is completed. When the robot is not in the safe area of a job, the job program cannot be

HUNAN Betrun Robot Co., Ltd.

started and run, and the corresponding prompt box will pop up. The program cannot be started until the robot is manually taught to the safe area. Similarly, when the robot exits the operation, it also needs to be zeroed in the safety zone, and the zeroing safety zone needs to be set in advance.

Click [tools] - [regional setting] - [safety zone] in the manual interface of the teach pendant, as shown in Figure 1.6, and the setting interfaces of "Starting-safety zone" and "Home-safety zone" appear. Multiple safety zones can be set, which can be completed by creating the starting point and ending point of the opposite corner of the safety zone (as shown in Figure 1.7).

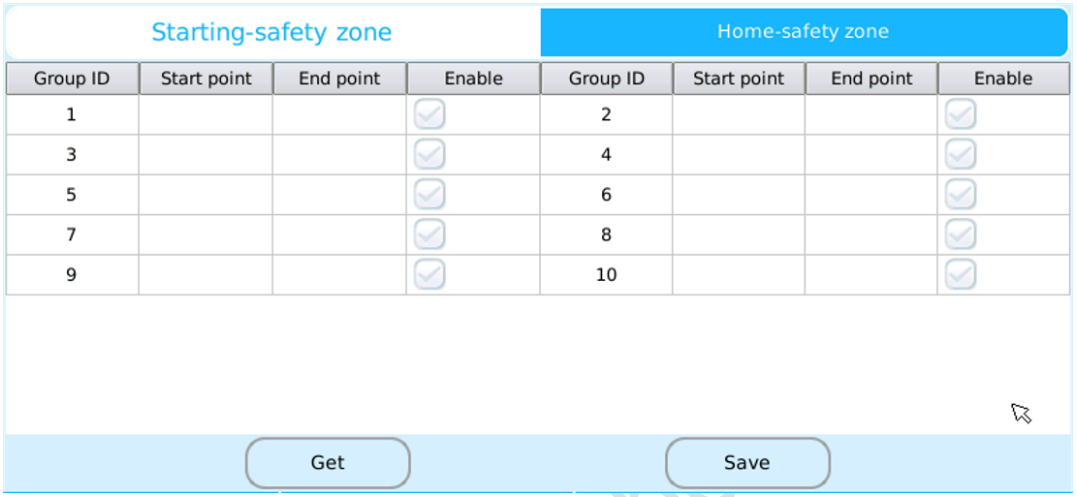


Figure 1.6 Safety zone setting interface

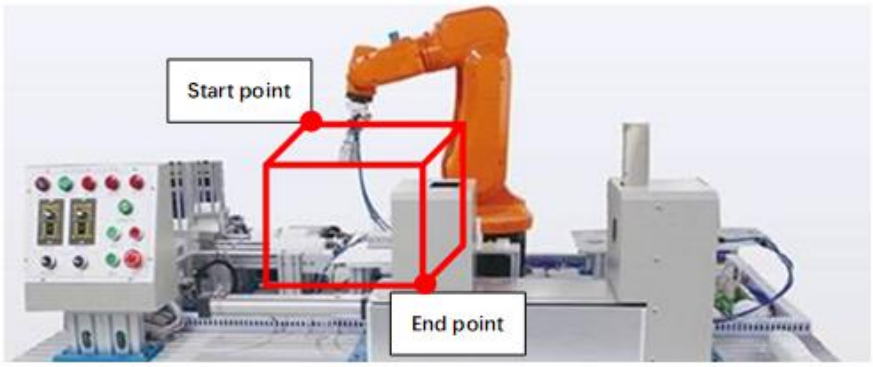


Figure 1.7 Start point and end point of safety zone

1.5.2 Interference zone

Setting the "interference area" has no effect on the running track of the system curve, but it can interact with the program through PLC programming or with other robots through the output port to work together. For example, in the stamping process, the robot must set the picking and discharging interference area in order to ensure the safety when working with the front and rear robots. The interference function of the system is divided into three types: point interference, space interference and axis interference. After setting the "interference section", whether it is manual teaching or program operation, the system will detect if it enters the "interference section".

Click [tools] - [area setting] - [interference area] in the manual interface of the teach pendant, as shown in Figure 1.8, and three types of interference area setting interfaces of "Space interference", "Point interference" and "Axis interference" appear.

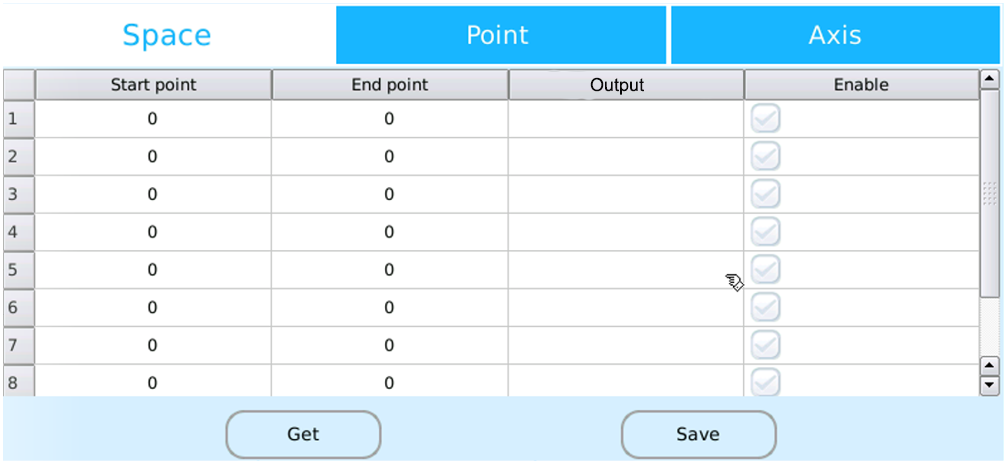


Figure 1.8 Interference area setting interface

The interference function is represented by "I" type in PLC editing, and different numbers represent different interference intervals:

- Point interference - when setting "point" interference, when running to this point, enter the interference interval, which is represented by i0-i19 in PLC editing.
- Space interference - when setting "spatial" interference, the interference interval refers to a "spatial body" generated diagonally with "starting point" and "end point" (excluding the "face" generated by "starting point", "end point" and "base coordinate system", so the "interference interval" can be set larger as needed). It is represented by i20-i39 in PLC editing.
- Axis interference - it is divided into 8 groups, which are respectively used to set the interference interval of 8 axes (j1-j8). It is represented by i40-i47 in PLC editing.

Robot Co., Ltd.

2 Coordinate systems

This chapter introduces five coordinate systems commonly used by industrial robots, and how to use teach pendant to establish tool coordinate system and user coordinate system.

2.1 Five common coordinate systems of robot

Coordinate system is a position index system defined on the robot or space in order to determine the position and attitude of the robot, including: base coordinate system, joint coordinate system, tool coordinate system, flange coordinate system and user coordinate system. In the process of "manual teaching", the motion of the robot is carried out in several different coordinate systems.

2.1.1 Base coordinate system

The base coordinate system is composed of the origin and coordinate orientation of the robot base, which is the basis of other coordinate systems of the robot. The direction of the base coordinate system is specified: the x-axis is forward, the z-axis is upward, and the y-axis is determined according to the right-hand rule (as shown in Figure 2.1).

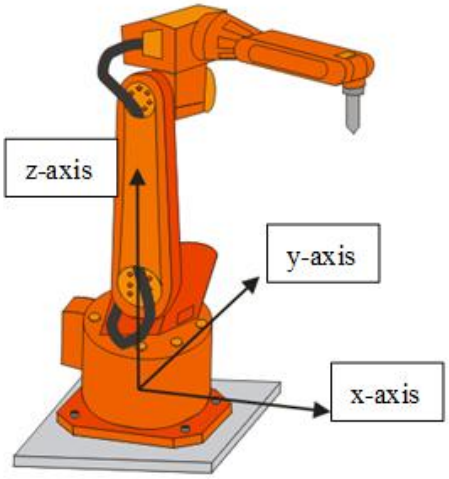


Figure 2.1 Robot base coordinate system

2.1.2 Joint coordinate system

Joint coordinate system the joint coordinate system of robot is used to describe the motion of each independent joint of robot. For six axis robots, the joint type is rotating joint. In the joint coordinate system, each joint can be driven in turn, so that the end of the robot can reach the specified position.

First, specify the "zero position" of the robot in the joint coordinate system, as shown in Figure 2.2. Then, the coding direction is set by the controller to specify the rotation direction of each joint. In the joint coordinate system, the rotation direction of each joint of the robot must be consistent

with Figure 2.2, that is, standing directly behind the robot body and using the "right-hand screw rule".

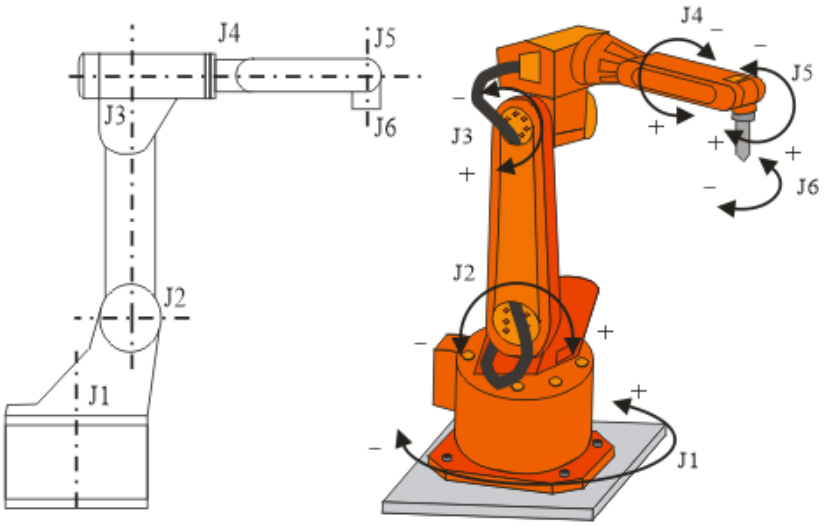


Figure 2.2 Robot joint coordinate system

2.1.3 Tool coordinate system and flange coordinate system

Tool coordinate system and flange coordinate system tool coordinate system is used to determine the position and attitude of the tool. It is composed of tool center position (TCP) and coordinate orientation. The tool coordinate system must be set in advance. The default coordinate axis direction is to rotate around the axis under the base coordinate system, or you can customize the coordinate axis direction. When there is no definition, the origin of the robot's default tool coordinate system is in the center of the robot's wrist flange. At this time, the coordinate system established with the center of the flange as the origin is called the flange coordinate system. When the robot is in the zero position, the flange coordinate system is parallel to the base coordinate system, as shown in Figure 2.3.

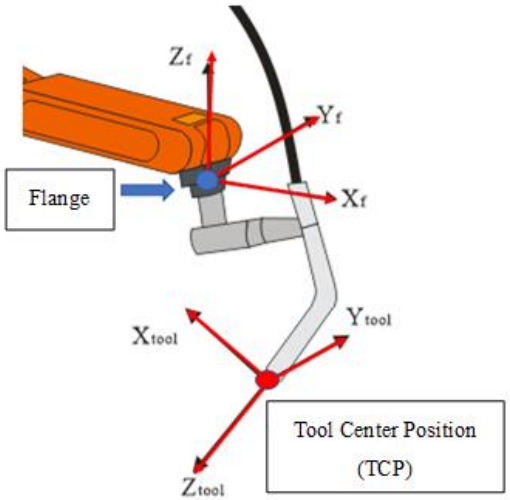


Figure 2.3 Robot tool coordinate system

2.1.4 User coordinate system

User coordinate system is a rectangular coordinate system defined by the user within the allowable range of robot action, which can be defined by the user according to their own needs and habits. As shown in Figure 2.4, the user coordinate system can be defined on the workbench.

With a user coordinate system, we can:

(1) Quantify the position of the point on the workbench.

(2) When teaching, the robot can move strictly according to the edge path of the workpiece in the user coordinate system.

(3) When the workpiece position or the workbench changes, transformation instructions of user coordinate system can be used without teaching the robot again.

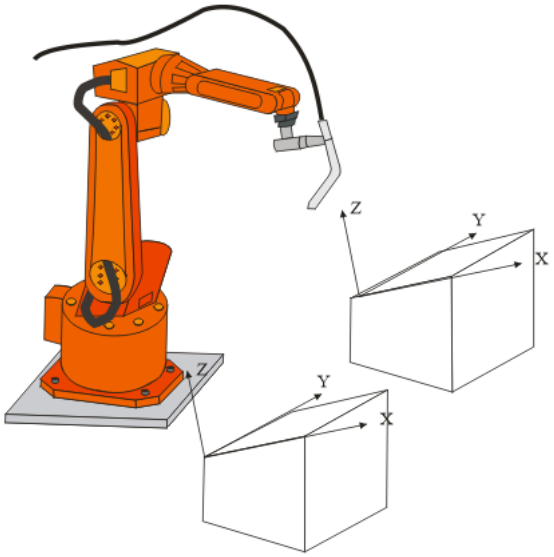


Figure 2.4 Robot user coordinate system

2.2 Establishment of tool coordinate system

The establishment of a tool coordinate system is also called tool coordinate system calibration.

There are two main approaches：

- **Multi-point calibration method**

We usually use four-point method. Let the tool TCP arrive at the same point with four different attitudes, and obtain the data of these four points, then the system can calculate the position of TCP relative to the center of flange.

When we need change the Axis direction, we use six-point method. Based on the four-point method, the data of points are obtained in x-axis and z-axis respectively, and the y-axis is determined by the right-hand rule.

Click [tool coordinate system] in the "manual interface" of the teach pendant, and the new tool

coordinate system interface as shown in Figure 2.5 will appear.

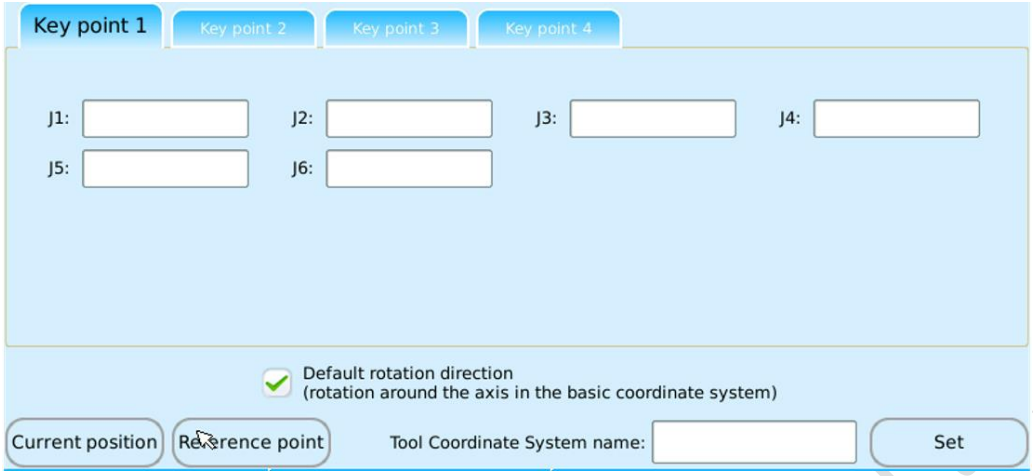


Figure 2.5 Tool coordinate system interface 1

When checked, the attitude of the default tool coordinate system is to rotate around the axis under the base coordinate system. The tool coordinate system can be established only by using four-point method to realize TCP measurement.

Principle of four-point method:

(1) Find a very accurate fixed point (the sharp cone end of the workpiece) within the working range of the robot as the reference point.

(2) Manually teach the robot, and use different tool postures to just touch the TCP of the robot tool with the reference point as far as possible, as shown in Figure 2.6.

(3) Through the 6-joint data of the robot flange position in four different postures of key point 1- key point 4, the system can automatically calculate the TCP data.

(4) The system records TCP data and constructs the conversion relationship between tool coordinate system, flange coordinate system and base coordinate system.

Note: If the TCP setting is accurate, when the tool is checked in the manual teaching, the TCP position will not change when the tool attitude is changed.

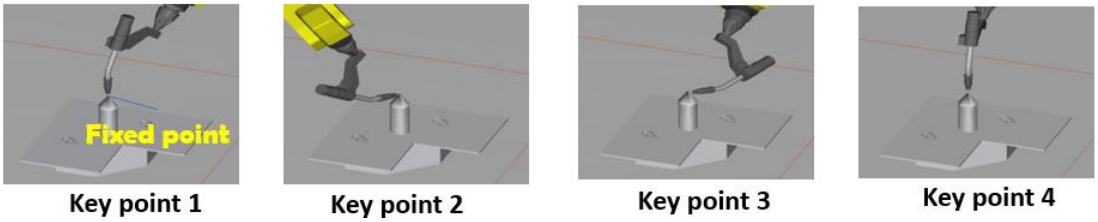


图 2.6 The sharp cone end of the workpiece is used as the reference point to establish the tool coordinate system

When the tool coordinate system attitude to be established is not the default rotation direction, it is necessary to additionally determine the direction point x and the direction point z, as shown in Fig. 2.7 and Fig. 2.8. Determine key point 1- key point 4 points according to the previous method, and then direction point z point shall move vertically upward on the basis of key point 4, and

direction point x point shall move forward on the basis of key point 4 to determine the direction points of Z axis and X axis.

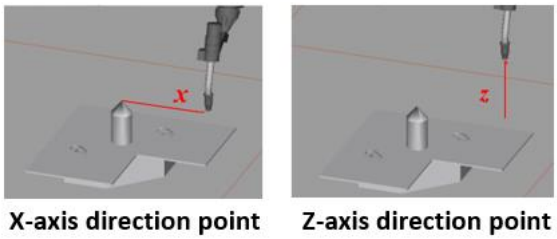


Figure 2.7 Additionally determine the direction point x and the direction point z

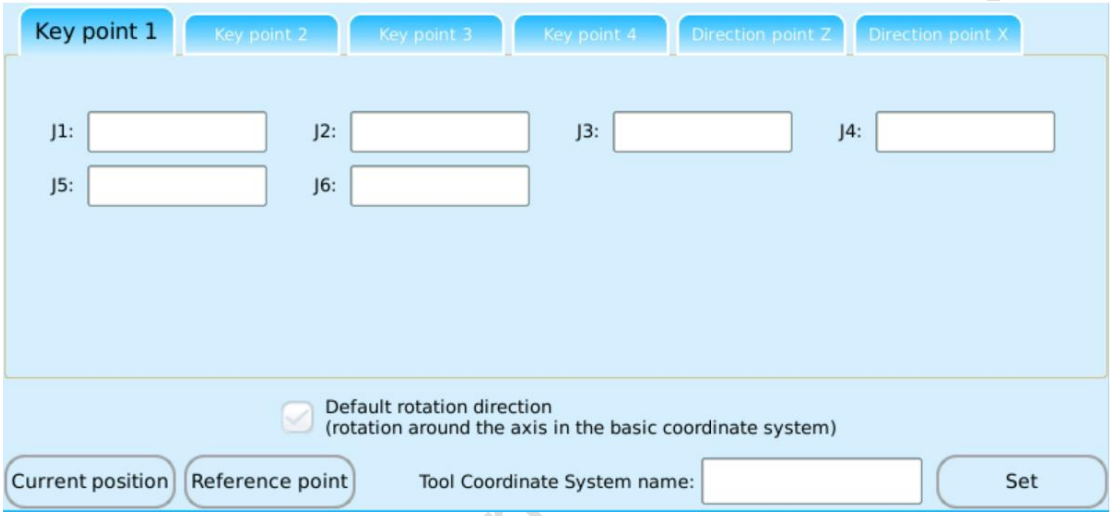
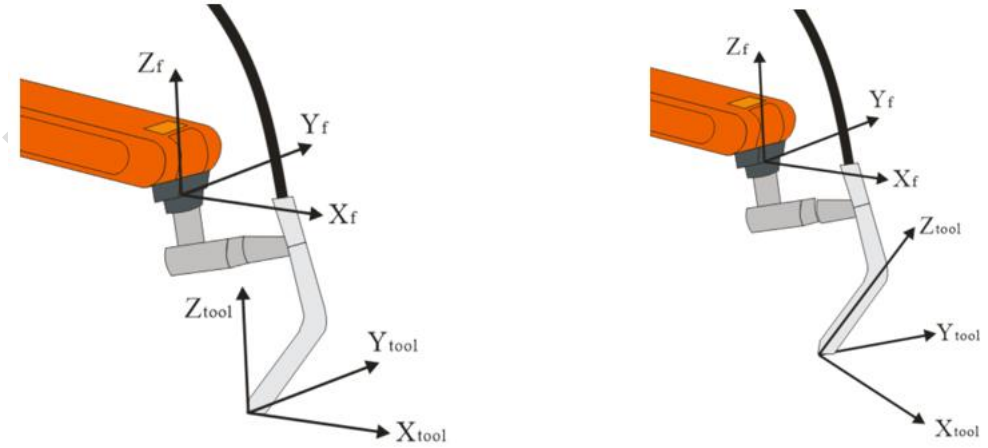


Figure 2.8 Tool coordinate system interface 2

Comparison:

Four-point calibration method: The tool coordinate system is parallel to the flange coordinate system.

Six-point calibration method: Define the X and Z directions by yourself.



(a) Four-point calibration method (b) Six-point calibration method

Figure 2.9 Comparison of two calibration methods

- **External reference calibration method**

let the tool TCP reach a measured external reference point (mark point), as shown in Fig. 2.10. If the tool needs to be changed frequently, this method can be used for rapid calibration.

All the program points associated with the original tool coordinate system will be associated with the revised new tool coordinate system, and the six joint data will be automatically corrected to keep the pose of TCP unchanged.

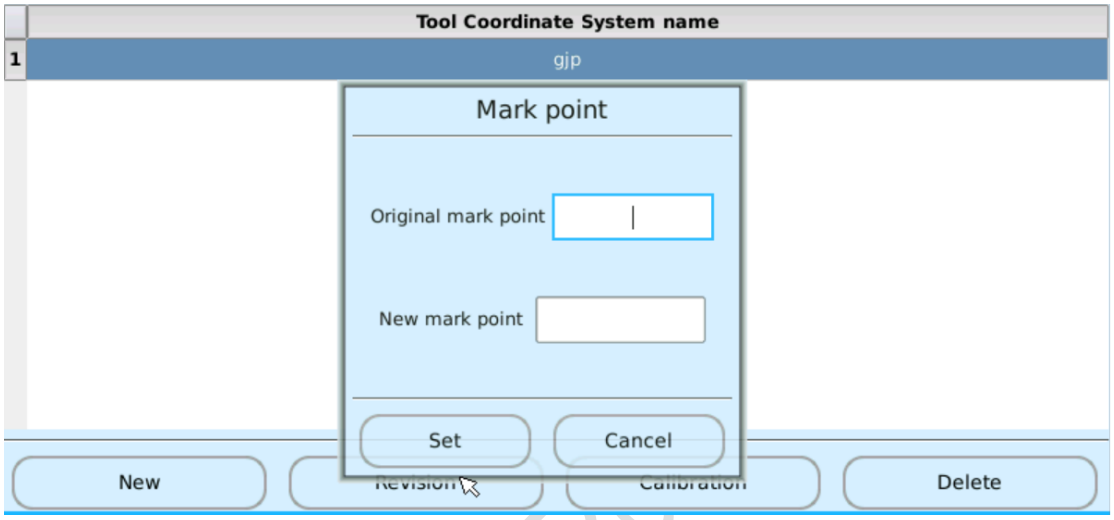


Figure 2.10 Using mark point for rapid calibration

2.3 Establishment of user coordinate system

The establishment of user coordinate system generally uses the three-point method:

(1) We let the tool TCP reaches the origin of the coordinate system, a point on the x axis and a point on the y axis.

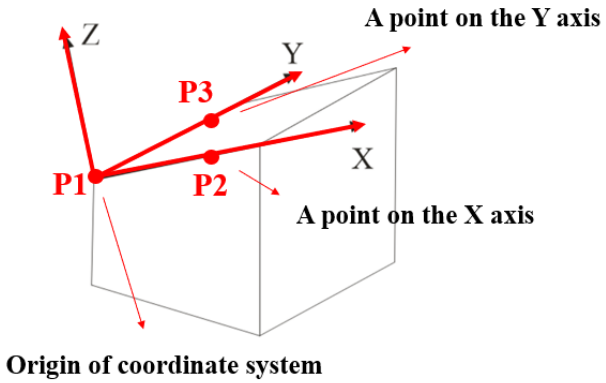


Figure 2.11 Three-point method to establish user coordinate system

As shown in t Figure 2.12, we obtain the data and create three points. Please pay attention here, the created "point" needs to check the tool coordinate system.

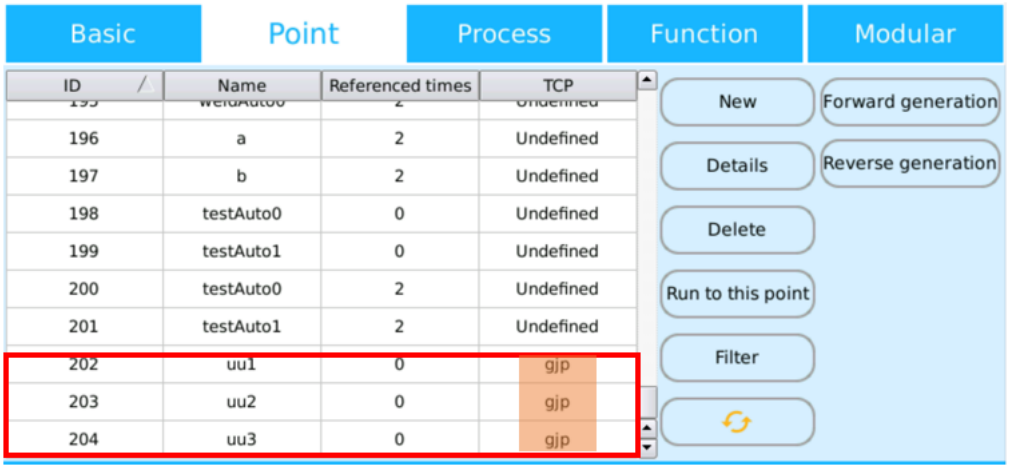


Figure 2 12 Create three points for user coordinate system

(2) Click [user coordinate system] in the "manual interface" of the teach pendant, and the user coordinate system interface as shown in Figure 2.13 will appear. Then the user coordinate system can be established by referencing the three points in turn.

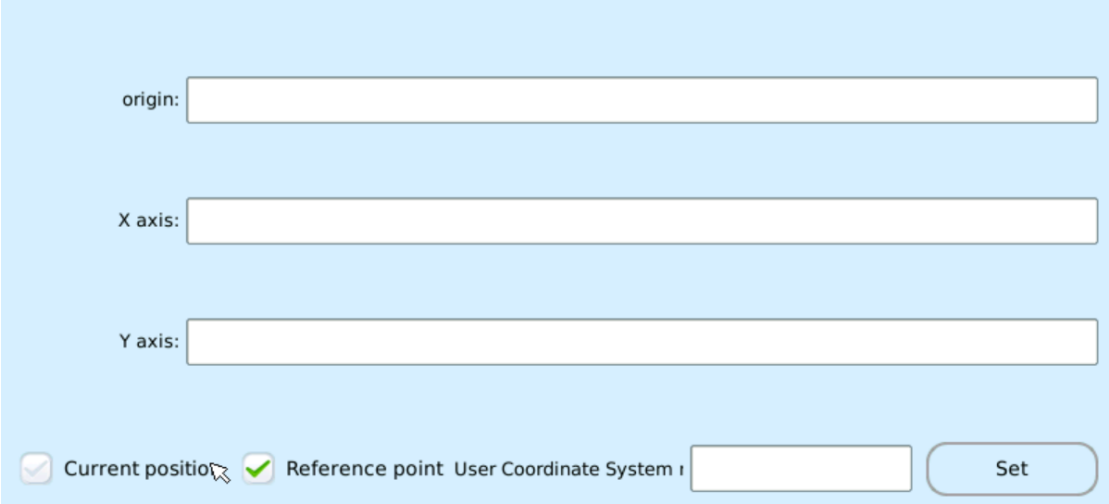


Figure 2.13 User coordinate system interface

3 Robot manual teaching

Using teach pendant to operate robot manually is the basis of industrial robot programming.

You can switch the teach pendant to "manual mode"; Press the "enable" button on the back of the teach pendant, select the coordinate system and set the teaching speed; then operate the button on the right side of the teach pendant to control the movement of the robot, as shown in Figure 3.1.

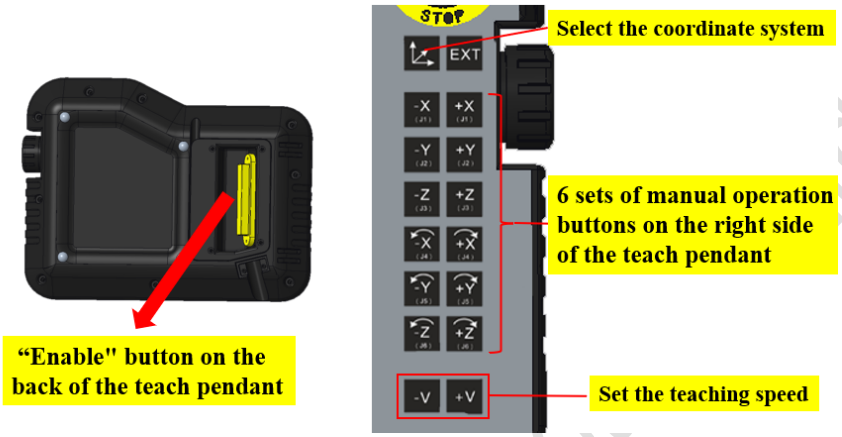


Figure 3.1 Buttons for manual teaching

3.1 Enable button

Carry out "manual teaching" in the "manual mode" of the teach pendant. The way to start is to press the "enable" button on the back of the teaching pendant. Release the "enable" button, and the teach pendant cannot control the motion of the robot.

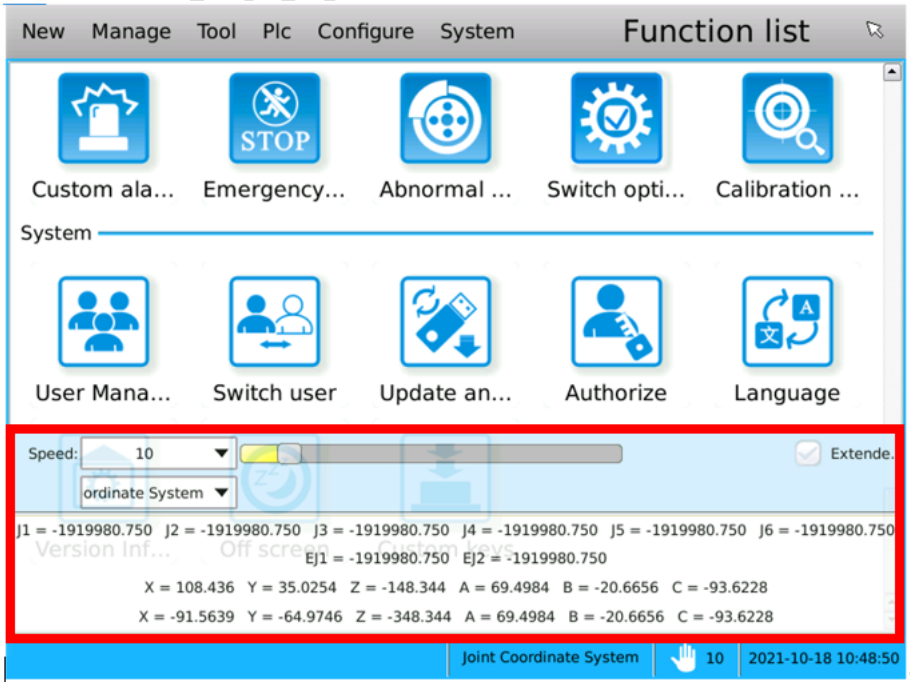


Figure 3.2 Manual teaching configuration and motion monitoring page

After pressing the "enable" button, the manual teaching configuration and motion monitoring page will be displayed at the same time, as shown in Figure 3.2. The upper part is the teaching configuration: including teaching speed, additional axis radio box, teaching coordinate configuration, precise position function button and so on. The lower part is the display of the current joint angles and the coordinate values under the corresponding coordinate system.

This page will pop up automatically when the "enable" button is pressed. It will be hidden after the focus is lost. In manual mode, when the "enable" button is pressed, slide from left to right or from right to left anywhere on the interface to call up the interface again.

3.2 Manual Teaching in Different Coordinate Systems

When selecting different coordinate systems and manually controlling the physical keys according to the six groups on the right side of the teach pendant, the motion mode of the robot is different.

Operation type: joint coordinate system

The robot has six movable joints, which are defined as j1-j6. In the joint coordinate system, six groups of buttons enable each joint of the robot to rotate independently, as shown in Figure 3.3.

The joint coordinate system can be selected for large range motion without TCP attitude.

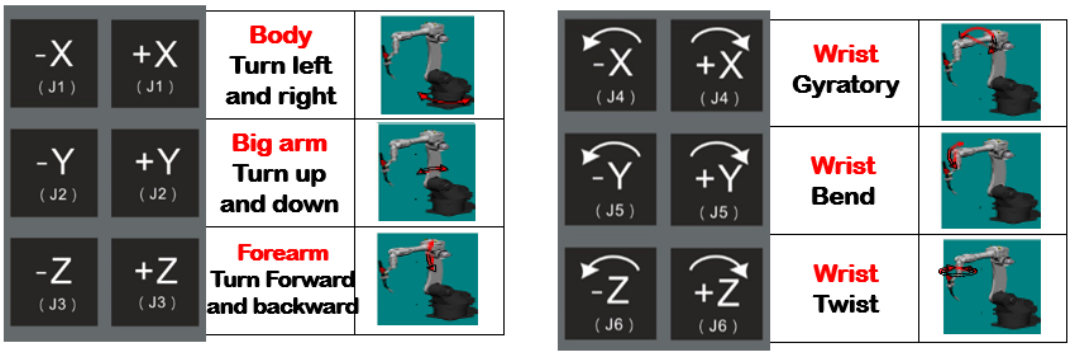


Figure 3.3 Keys function when the operation type is " joint coordinate system"

Operation type: base coordinate system + TCP

The base coordinate system is a common coordinate system in teaching. In this operation type, the robot is multi-joint linkage.

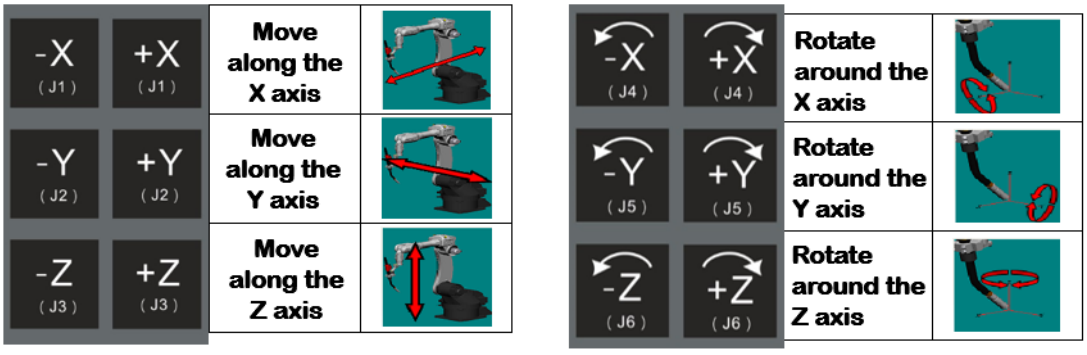


Figure 3.4 Keys function when the operation type is " base coordinate system"

Among the six groups of keys: the first three groups of keys respectively control TCP to move horizontally along the X, Y and Z axes of the base coordinate system, and the last three groups of keys make the tool rotate around the X, Y and Z axes of the base coordinate system on the premise that TCP position remains unchanged.

Note: if the corresponding tool coordinate system is not selected before manual operation, the point at the same position during rotation is the center of the connecting flange.

Operation type: user coordinate system + TCP

The robot is multi-joint linkage.

Among the six groups of keys: the first three groups of keys respectively control TCP to move horizontally along the X, Y and Z axes of the user coordinate system. The last three groups of keys make the tool rotate around the X, Y and Z axes of the user coordinate system on the premise that TCP position remains unchanged.

Note: if the corresponding tool coordinate system is not selected before manual operation, the point at the same position during rotation is the center of the connecting flange.

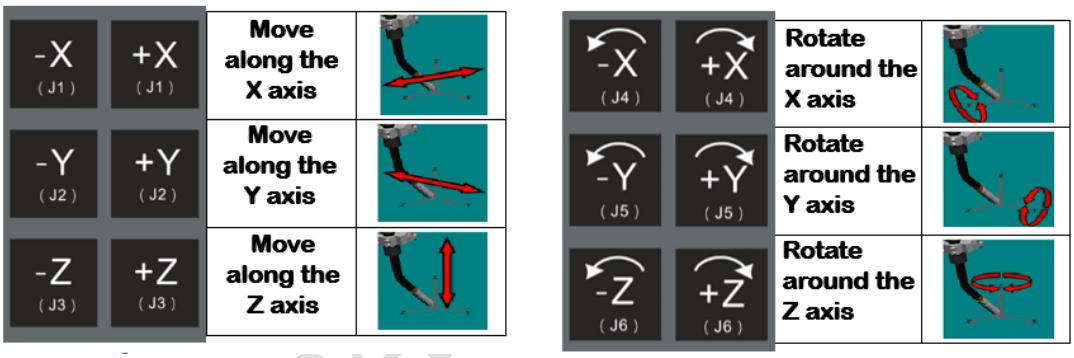


Figure 3.5 Keys function when the operation type is " user coordinate system"

Operation type: tool coordinate system

The robot is multi-joint linkage.

Among the six groups of keys: the first three groups of keys respectively control TCP to move horizontally along the X, Y and Z axes of the tool coordinate system. The last three groups of keys make the tool rotate around the X, Y and Z axes of the tool coordinate system on the premise that TCP position remains unchanged.

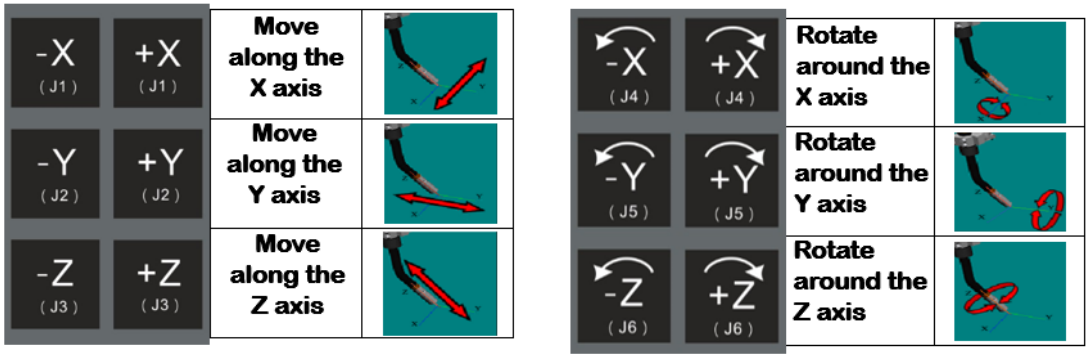


Figure 3.6 Keys function when the operation type is " tool coordinate system"

In the "tool coordinate system" operation mode, it is relatively easier to control the motion track of the tool TCP, and the posture of the tool can be adjusted when the TCP position is not moved. If the six point method is used to establish the tool coordinate system and the effective direction of the tool is set as the Z axis, as shown in Figure 3.7, in some cases, the tool coordinate system is the most suitable for the translation operation without changing the tool attitude relative to the workpiece.

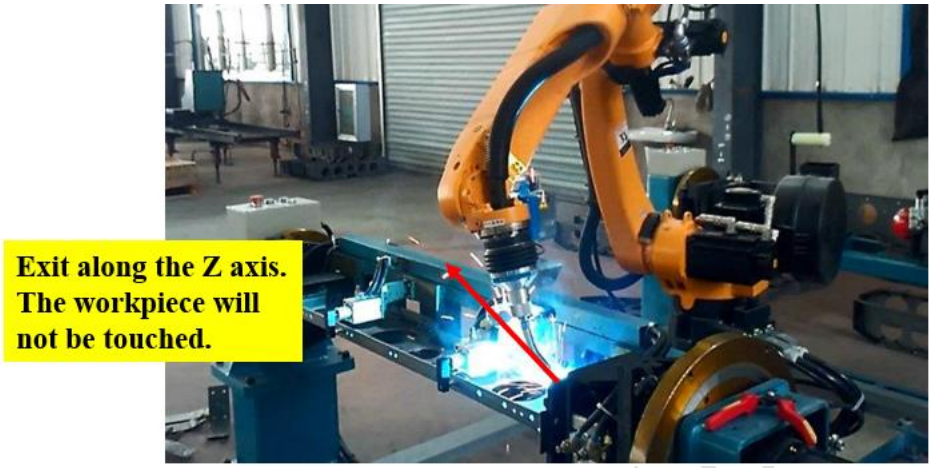


Figure 3.7 Manual teaching in tool coordinate system

3.3 Manual teaching speed

During manual teaching, it is better for the robot to move slowly in the working area. At present, the system settings during manual teaching:

- The maximum speed of joint movement is 600 r/min;
- In the base coordinate system, tool coordinate system and user coordinate system, the maximum speed of linear movement in X, Y and Z directions is 200 mm/s, and the maximum speed of rotation around X, Y and Z axes is 50 degrees/s.

On this basis, the "motion configuration floating box" adopts the percentage method to control the motion speed of the robot. The system provides three speed regulation methods: stepless speed regulation, specified input and gear selection.

- Stepless speed regulation: realized by dragging the adjustment button.
- Specify input: click the speed text box and a soft keyboard will appear for direct input.
- Gear selection: click the drop-down triangle button on the right side of the speed text box to display the set gear value for selection.

3.4 Operational feedback

For each operation on the teaching pendant, the history can be seen in the operation feedback display box.

For each operation on the teach pendant, the history can be seen in the operation feedback display box. The operation history display box of the teach pendant is a pop-up page, which is displayed at the bottom of the screen.

HUNAN Betrun Robot

In manual mode: slide the interface from left to right or from right to left, you can call up the operation history pop-up box, and it will be hidden automatically when the focus is lost.

Manual mode + press the enable key: slide the interface from left to right or from right to left. By default, the motion configuration page will be called out first. At this time, switch with the "tab" key to switch to the operation history page and automatically hide when the focus is lost.

In automatic mode: slide the interface from left to right or from right to left. You can call up the operation history pop-up box and hide it automatically when the focus is lost.

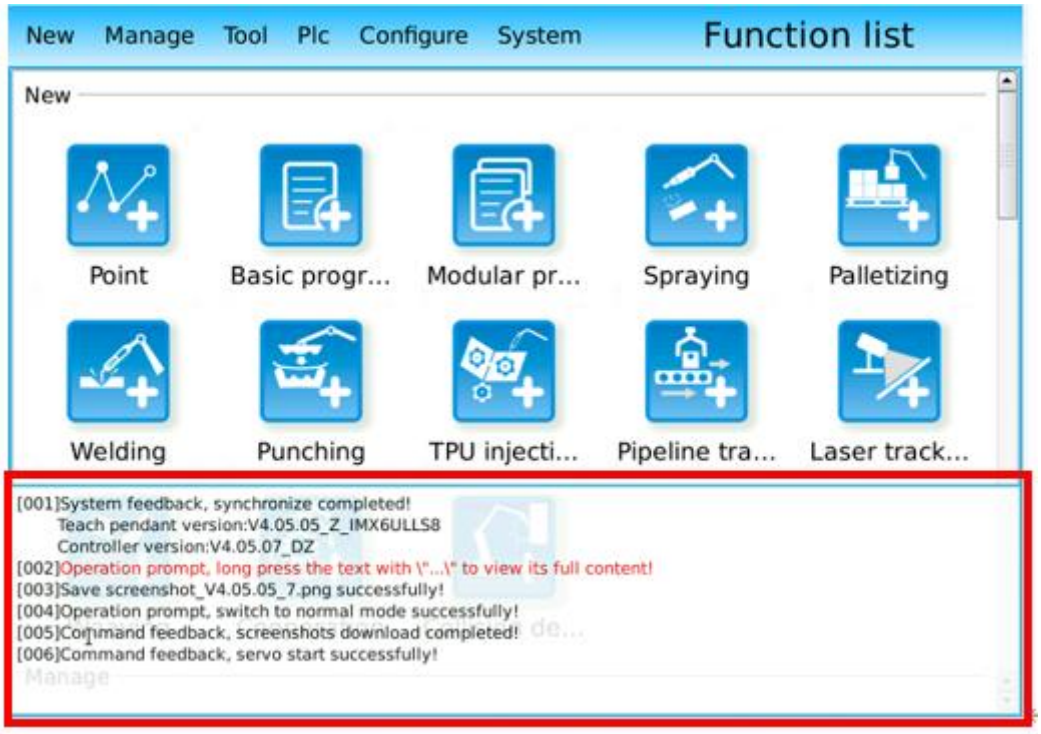


Figure 3.8 Operation feedback display box

4 Online teaching programming

This chapter explains how to quickly generate and execute a simple robot motion control program through teaching programming, to help operators master the basic programming method of teach pendant.

4.1 Contents of teaching programming

The information required for robot operation includes three parts: motion trajectory, operation conditions and operation sequence.

4.1.1 Motion trajectory

Motion trajectory is the key point for the robot to complete an operation. When teaching, it is impossible to teach all the points on the operation trajectory once, but segments the trajectory, manually teach the starting point, and end point of each segment of the trajectory, then confirm the attributes of each segment of the trajectory in the programmed curve motion command, and then the path planning module of the control system performs interpolation operation.

4.1.2 Operating conditions

In teaching programming, in addition to telling the robot motion trajectory, it is also necessary to reasonably configure the process conditions of robot operation in advance. For example, in order to obtain good welding quality, robot welding operation needs to set welding conditions, including welder model, welding sequence, welding control signal, welding current and voltage control and other parameters.

There are two types of input methods for operating conditions on Betrun teach pendant:

(1) Create function package or process package

(2) Set by command in the program

4.1.3 Operation sequence

The movement and process implementation of robot need to be carried out according to a certain operation sequence. The setting of operation sequence mainly involves two aspects.

(1) Process sequence of operation object

When writing the robot trajectory program by manual teaching, it is necessary to integrate the operation sequence of the process.

(2) Action sequence of robot and peripheral equipment

In the complete industrial robot system, in addition to the robot itself, it also includes some peripheral equipment. The discovery device provides functions such as collaboration, pipeline

HUNAN Betrun Robot Co., Ltd.

tracking and so on, which can be set up before programming and invoked in the program.

4.2 "Point" in teaching programming

"Point" is a description of the position and attitude of the robot. Specifically, it is the position and attitude of the center of the end flange in space when the tool TCP reaches the target position when the robot is taught manually. This information can be uniquely determined by the joint angle data of the robot.

4.2.1 "Point" management

Click [manage] - [point], as shown in Figure 4.1, to open the "point" management interface, it shows the relevant information of the points, including ID number, name, reference times and the name of the corresponding TCP (tool center point).

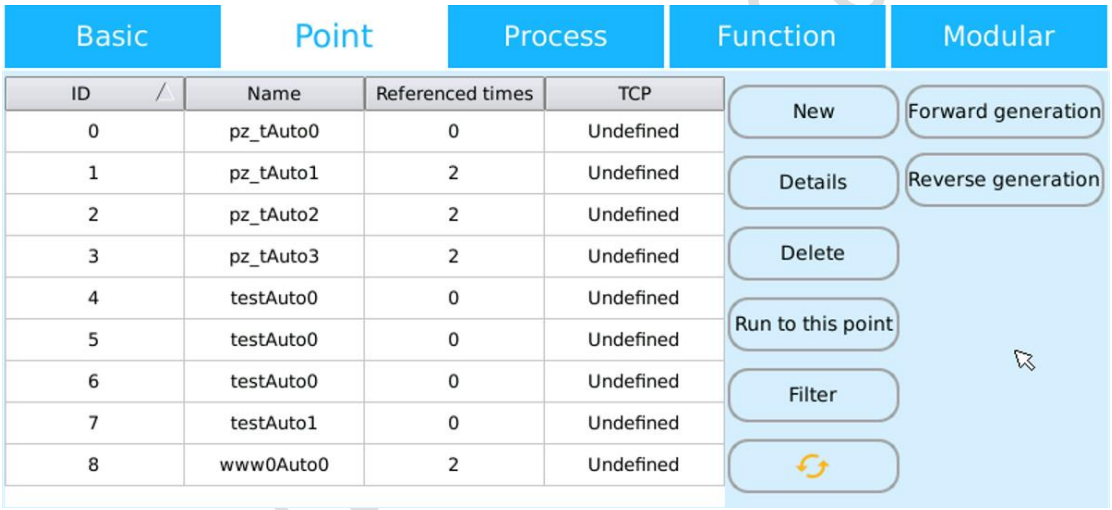


Figure 4.1 "Point" management interface

(1) ID: the ID number is the unique identification of the point, which is automatically generated when it is created. The points referenced in the curve motion instruction in the basic program are associated and uniquely confirmed by the ID number.

(2) Name: the point name can be automatically generated during teaching or input by operators, which is convenient for point reference and management in the program.

(3) Reference times: refers to the number of times the point is referenced in all basic programs in the controller.

Note: when the reference times is greater than 1, modifying the data of this point may affect some motion curves in a basic program or the motion curves in many basic programs.

(4) TCP: the corresponding tool coordinate system needs to be checked. When it is undefined, this point can only determine the position and attitude of the robot flange center, but cannot determine the position and attitude of the TCP.

4.2.2 Creation of "point"

When the robot reaches the specified position, click [new] - [point], as shown in Figure 4.2, to open the "point" information interface. It can be named according to the purpose of the point. Enter the "point" name and click [current position], then the system will read the six joint data of the current robot and display the X, Y and Z coordinate values at the same time.

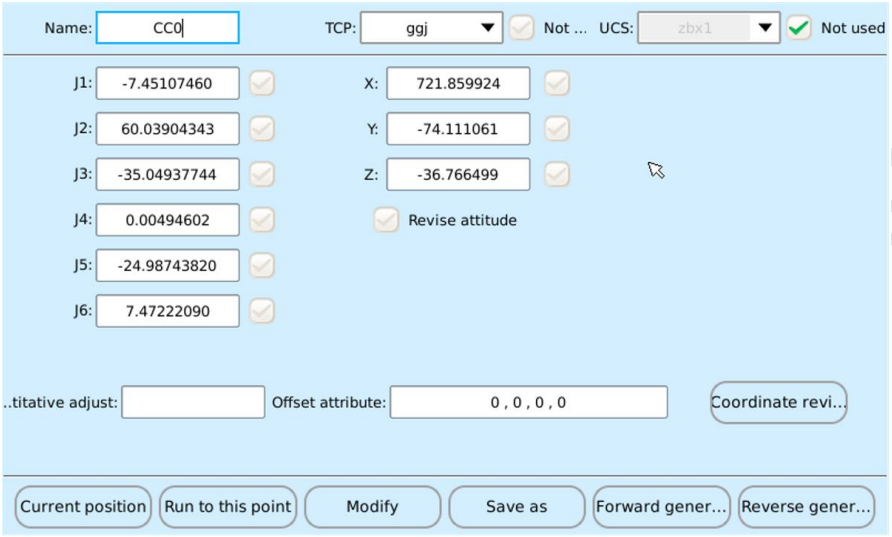


Figure 4.2 "Point" information interface

In the "Point" information interface, five basic functions are provided:

(1) . Click [run to this point] to control the robot to directly reach the position and attitude of this point.

(2) Select “user coordinate system”, we can view the position data of the point in the user coordinate system.

(3) Check “Revise attitude", then enter the revised value, and click [coordinate revision], then the 6 joint values will be revised. Click Modify or save as a new point.

(4) If checking the "check boxes", you can modify multiple coordinate value data at the same time by entering the value in the quantitative adjust text box.

(5) The "offset attribute" function is provided. You can enter the maximum absolute value of the point offset to limit the operator's modification of the point.

Note:

(1) The six joint coordinate value can only determine the position of the center of the robot flange. However, the operator pays attention to the posture of TCP when teaching, so the tool coordinate system must be selected when creating a new point, and the controller can carry out interpolation operation based on TCP to generate straight-line or arc continuous path trajectory.

(2) When saving point data, only six joint data and associated tool coordinate system ID are stored, and user coordinate system ID is not saved. Therefore, when you need to copy the program written on one robot to other robots, you need to click the [forward generation] button to bind the user coordinate system.

(3) The displayed X, Y and Z coordinate values have the following meanings:

- When "do not use" tool coordinate system is checked, it is the X, Y and Z coordinate values of the flange center under the base coordinate system;

HUNAN Betrun Robot Co., Ltd.

- When selecting the corresponding tool coordinate system, the X, Y and Z coordinate values of the tool TCP in the base coordinate system;
- When the tool coordinate system and user coordinate system are selected at the same time, it is the X, Y and Z coordinate values of TCP in the user coordinate system.

4.2.3 Display of current "point"

When entering the manual or automatic mode main interface of the teaching pendant, you can see the real-time display of the current "point" information of the robot as shown in Figure 4.3 at the bottom of the interface.

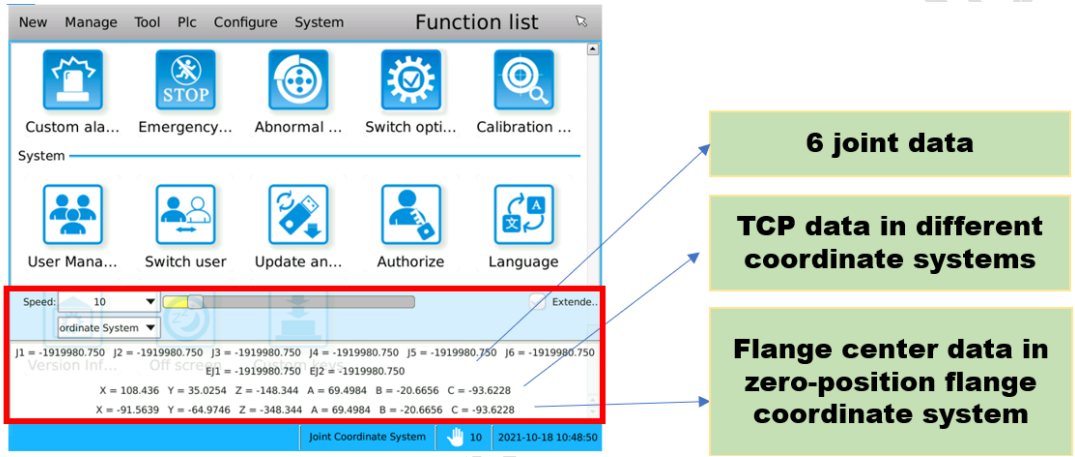


Figure 4.3 Real time display of current "point" information

4.2.4 Home position

As shown in Figure 4.4, after the installation of the robot system, each joint of the robot body will move to each position as shown in the figure before leaving the factory, and then set the home point in the "machine parameter setting" interface. When the robot manually teaches and reads the joint angle of the "point", it is positioned with the set "home position" as the reference point.

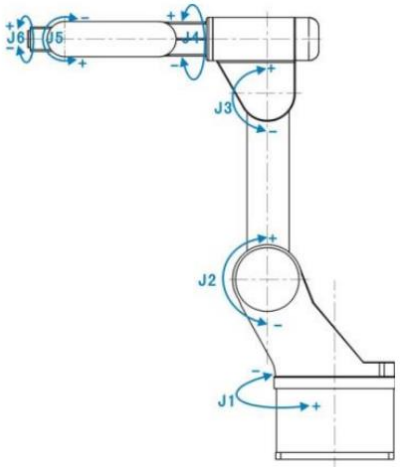


Figure 4.4 Robot home position attitude

When the teach pendant is in "automatic mode", click the "run to home position" button on the interface (as shown in Figure 4.5) to control the robot to run to the home position. At this time, read the angle value of each joint of the current "point", and the angle value is displayed as "0".

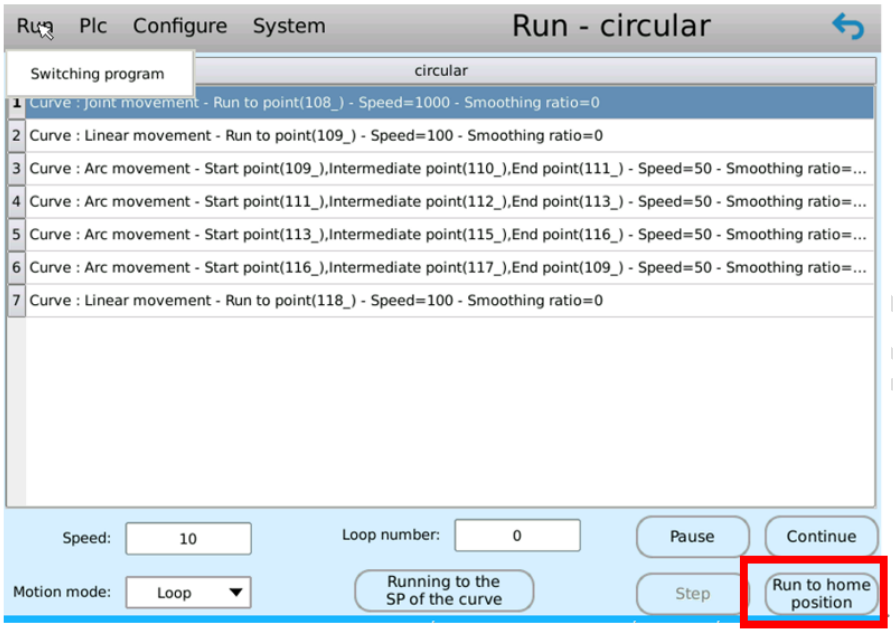


Figure 4.5 "run to home position" button in "automatic mode"

4.3 Online teaching a basic program

4.3.1 Task description

Write a "test1" motion curve control program for robot welding through manual teaching, let the robot move from T1 (operation start point) to T2 (operation adjacent point) at the speed of 1000mm/s, and then move in a straight line at the speed of 100mm/ from T2 to T3 (operation start point) to start welding at the speed of 10mm/s, After welding to T4 (operation end point), exit to T5 (operation avoidance point) in a straight line at the speed of 100mm/s. The trajectory of the robot is shown in Figure 4.6.

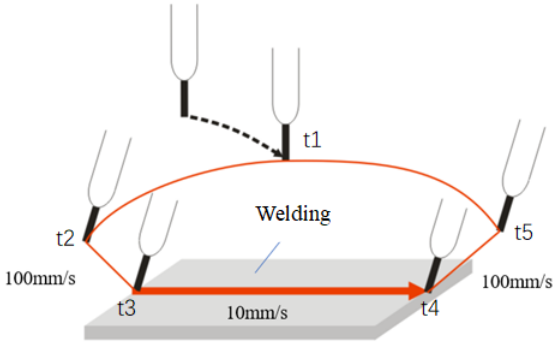


Figure 4.6 Robot motion trajectory for welding

Before creating the basic program, there are four things to do:

- Set startup safety zone and home safety zone

- Create the tool coordinate system and user coordinate system
- Create the welding process package
- Carry out trajectory path planning

After completing the above steps, you can start teaching and write a basic program.

4.3.2 Create a basic program

The programs in Betrun software system are divided into four types: basic program, process package, function package and modular program. Among them, the other three types will eventually be converted into basic programs for interpretation and execution by the main controller.

This section describes how to create a new basic program.

Log in to the software with administrator privileges

Turn on the power supply of the robot control cabinet, wait for the human-computer interaction interface of the teaching pendant to start, and then log in to the system as an administrator. The other two types of operators (operators and viewers) do not have programming permission when logging in.

Enter the basic program programming interface

Switch the mode change-over switch of the teach pendant to the "manual" sign. On the manual main interface: click [new] - [basic program] to enter the basic program programming interface, as shown in Figure 4.7. Here you can create a "basic program".

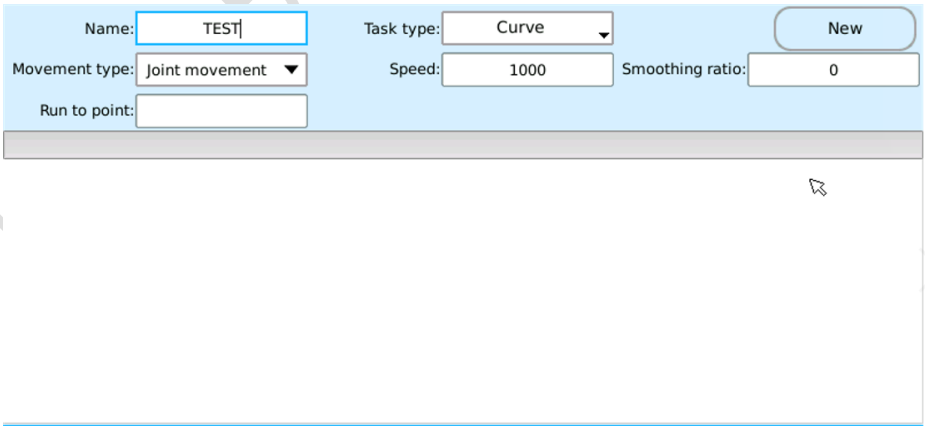


Figure 4.7 Basic program programming interface

Program naming and creation

After entering the "basic program" creation interface, the first step is to establish the program name. As shown in Figure 4.7, click the blank box after "program name" on the touch screen by hand, and a soft keyboard will appear. Enter the English letter "XXX", click [Enter] on the soft

Betrun Robot Co., Ltd.

keyboard, and the blank box after "program name" will appear "XXX".

After naming the new program, click the [new] button in the upper right corner of the interface, and the program name "program: XXX" will appear on the blank programming area in the middle, as shown in Figure 4.8. At this time, the [new] button will change to the [Cancel] button, indicating that the new program is created; Click the [Cancel] button to cancel the whole current program operation.

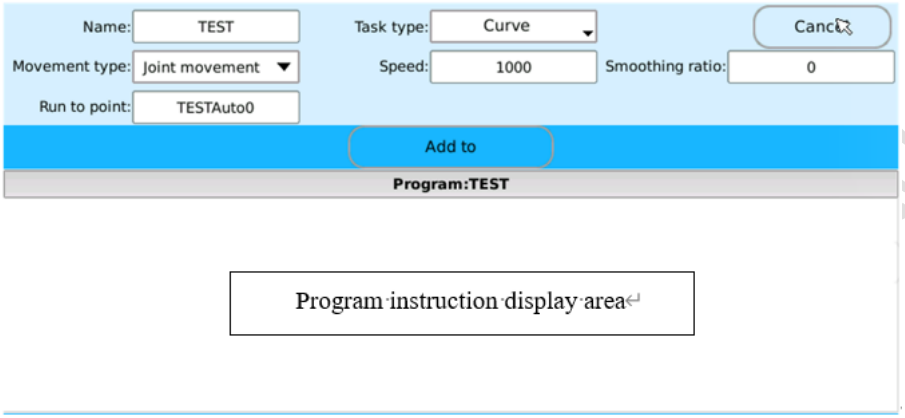


Figure 4.8 Creating the basic program named "TEST"

4.3.3 Manual teaching for the program design

After the basic program is created, it is necessary to conduct manual teaching and add corresponding instructions one by one in the program instruction display area.

4.3.3.1 Select task type

In the "task type" drop-down menu, we can see various instructions provided by the software, as shown in Figure 4.9. Among them, the command to control the precise motion of the robot is the default "curve" of "task type", or select "movement statement" - "curve motion" in the drop-down menu (both are the same command).

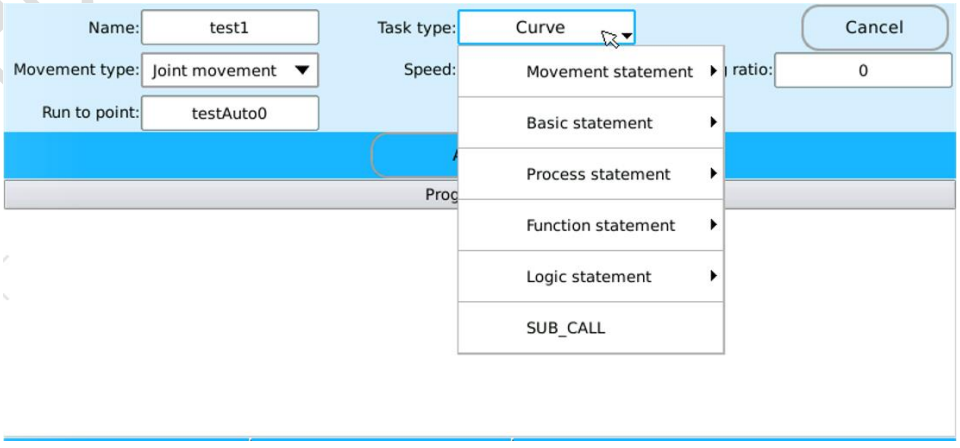


Figure 4.9 Various instructions included in the task type

4.3.3.2 Curve instruction description

A curve instruction runs in the robot system, which is expressed as a moving track that accurately controls the robot from "start point" to "end point". In this instruction, the following contents need to be specified:

- Movement type: Specifies the motion control method of the moving track.
- Run to point: teach the pose of the end point that the curve segment robot TCP will move.
- Speed: Specifies the movement speed of the robot.
- Smoothing ratio: the smoothness of robot motion.

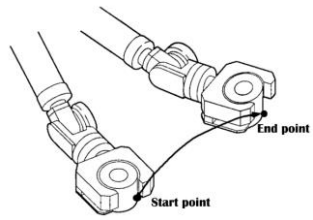
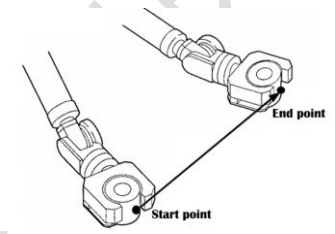
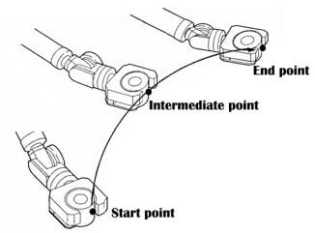
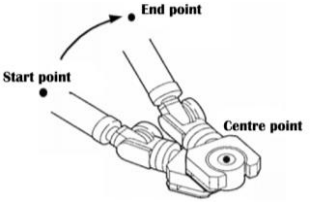
The format of curve instruction is as follows:

Curve - Movement type: Run to point (ID_point name); Speed=XXXX; Smoothing ratio =XXX

Movement type

The movement type specifies how the motion of the movement track is controlled. The movement types and functions currently provided are shown in Table 4.1.

Table 4.1 Description of curve instruction movement type

Type	Figure	Description
Joint movement		The basic movement method of moving the robot TCP to the specified position. All axes of the robot accelerate at the same time. After moving at the teaching speed, it decelerates at the same time and stops.
Linear movement		A moving method for controlling the robot TCP moving trajectory in a linear motion mode from the start point to the end point.
Arc movement		A moving method to control the TCP moving track of robot in the form of arc from the start point through the intermediate point to the end point.
Circular movement		A movement method for controlling the TCP movement trajectory of the robot in a circular motion from the start point to the end point. The Centre point needs to be taught.

Hunan Betrun Robot Co., Ltd.

Run to point

The running track of the program is connected by the "start point" and "end point" of each curve in the program. The "end point" of each curve is the "start point" of the next curve. Therefore, when using the curve instruction, you only need to set the "end point" of the curve through manual teaching, that is, the "motion to point".

After the program is named and created successfully, the system will automatically set a "point" name for the "run to point" for the first curve instruction. As shown in Figure 4.9, the name "testauto0" is set for the new "TEST" program. You can click the "run to point" text box to change the name of the point, or select an existing "point" for reference.

Speed

In the process of robot movement, the speed of movement needs to be set. The speed unit in the "joint movement" type is r/min, and the maximum should not exceed the setting parameters in "machine parameters"; The speed units of "linear movement", "arc movement" and "circular movement" types are mm/s, and the maximum speed should not exceed 2000 mm/s.

Smoothing ratio

By setting the "smoothing ratio", the robot can run more smoothly. The range of "smoothing ratio" parameter is "0-100", "0" means no "smoothing", and "100" means the maximum "smoothing".

4.3.3.3 Teaching programming

The trajectory of the robot is composed of several curve instructions. According to the task, there are two methods to teach manually and add curve instructions: one is to teach t1-t5 points manually and add curve instructions synchronously in the basic program programming interface; The other is to manually teach new t1-t5 points in the "point" management interface, and then add curve instructions by referencing points in the basic program programming interface.

After adding curve instruction by manual teaching method for many times, the program instructions display area is shown in Figure 4.10. Click one of the instructions, you can modify, replace and delete the instruction.

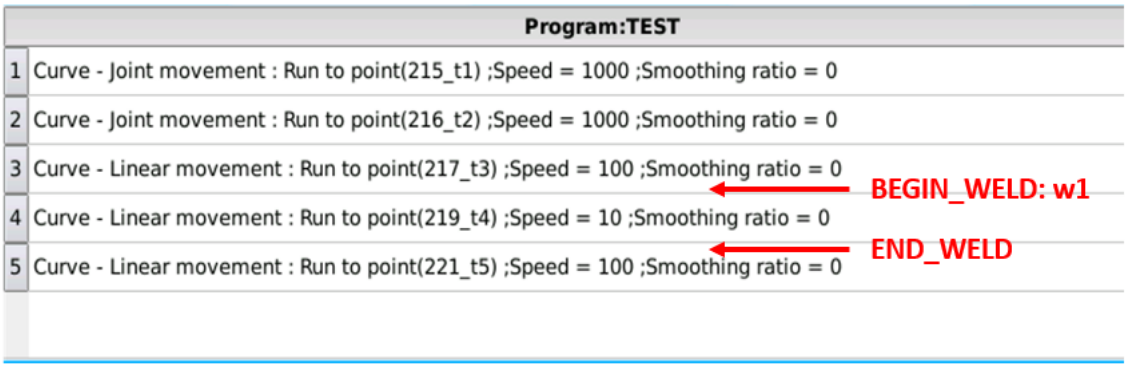


Figure 4.10 TEST program

HUNAN Betrun Robot Co., Ltd.

In the "TEST" program, if the program is run for a single time, the robot will eventually stop at T5. If the motion program is cycled, after the last curve trajectory is completed, the robot takes T5 as the start point and continues to run the first curve to the "run to point" T1 to form a closed loop of motion trajectory.

4.3.3.4 Save program

After the program design is completed, click the [save] button to save the program. When the program is saved in "program management", the creation of "basic program" is completed.

4.3.4 Program running

After the program is created, switch the mode change-over switch of the teach pendant to the "automatic" sign, and the system will switch to enter the "automatic interface". In automatic mode, the program can be run without pressing the "Enable" button.

t Co., Ltd.

5 Betrun programming

This chapter describes the type of instructions constituting the application program in Betrun control system, the steps of writing, debugging, modifying, and executing the application program, the composition of modular program, and how the system starts up and automatically runs the program.

5.1 Program instruction

Program instruction is the basic unit of robot program and the execution method to control the specified actions of robot and peripheral equipment. In the Betrun control system, writing the program only needs to select the corresponding instructions, input parameters, and click the "add" button. There is no need to manually input the command line code, which is very convenient and easy to use.

The instructions provided in Betrun control system are divided into the following types, as shown in Figure 5.1.

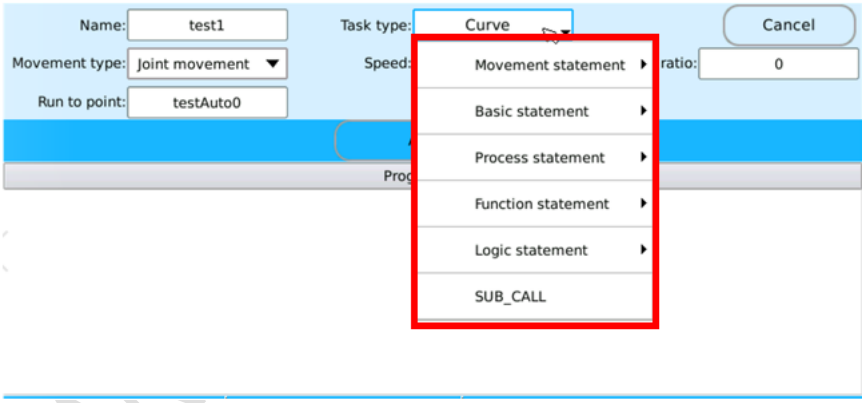


Figure 5.1 Instructions provided in Betrun control system

See Appendix 1 for specific instructions.

5.2 Global variable definition

In order to cooperate with program instruction function, logic judgment and external communication interaction, the system defines global variables with specific meaning in the storage space.

See Appendix 2 for details.

5.3 Programming and testing

5.3.1 New program

Switch the mode change-over switch of the teach pendant to the "manual" sign, and enter the new basic program interface in the main interface of manual mode.

See Section 4.3, Chapter 4, part I of the manual for specific operation.

5.3.2 Manual teaching programming

In the task type, select the corresponding instruction through the drop-down menu (as shown in Figure 5.2), and the interface of the instruction parameter setting area will switch to the parameter setting interface corresponding to the instruction. Set the parameters of the instruction and click the "add" button to complete the programming.

Among them, see Chapter 3 for the procedure manual teaching steps of the robot motion part.

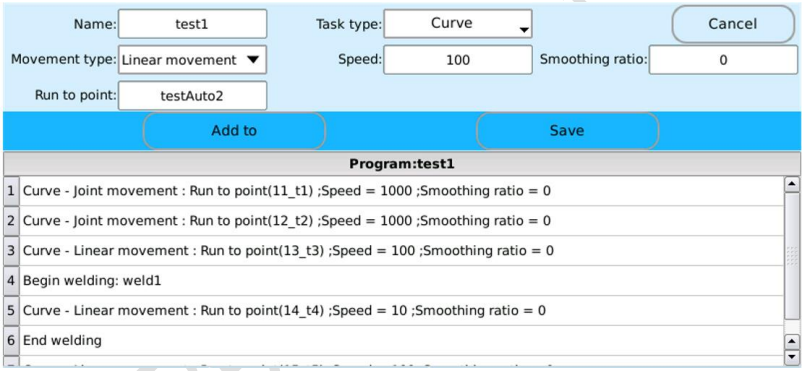


Figure 5.2 Basic program editing interface

5.3.3 Program notes

The program instruction display is divided into left and right parts, as shown in Figure 5.3. The program content is displayed on the left. Click the blank area on the right of the instruction to pop up the soft keyboard, and you can enter the program notes in Chinese or English.



Figure 5.3 Program content and annotation display

5.3.4 Program editing

After the program is created, modify the program as needed. You can change, insert, delete, copy, cut and other operations of instructions through the toolbar and clicking the program name.

Deletion, modification and insertion of instructions

Select an instruction in the program instruction display area on the touch screen, the instruction will turn gray, and the task type and parameter setting area above the interface will be changed to be consistent with the instruction. The toolbar in the programming area will be switched to figure 5.4.



Figure 5.4 Toolbar in the programming area

Clip and copy of program segments

Click the program name bar, and it will switch to the full screen interface of the program. Click the command on the touch screen, the command will change color, the "selected task" appears on the right, and the [Delete], [Copy] and [Cut] buttons appear on the lower bar at the same time, as shown in Figure 5.5. Multiple instructions can be selected at one time for copying and cutting.

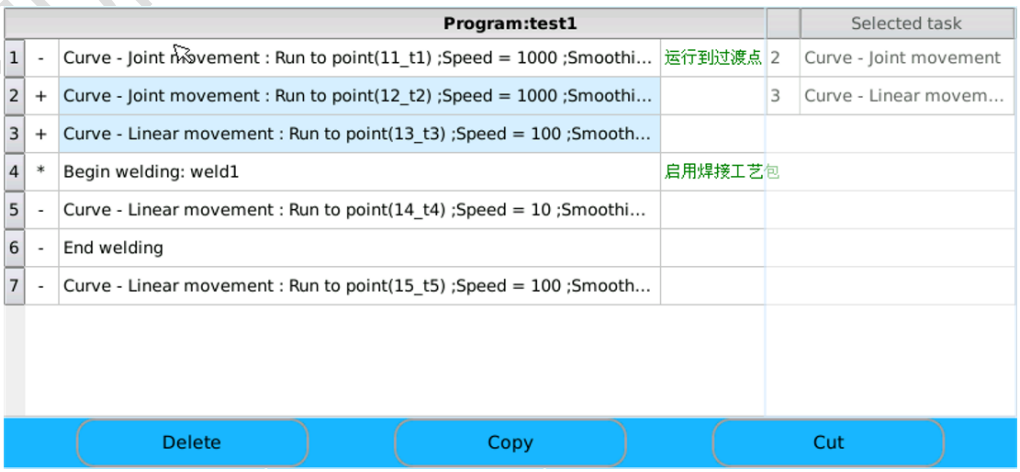


Figure 5.5 Copy and cut program instructions

After clicking [copy] or [cut], the lower bar will switch to another toolbar, as shown in Figure 5.6. At this time, select the position to be inserted, and the corresponding instruction will turn gray. You can realize the "selected task" instruction in front of the "front" or "rear" of the gray instruction [forward paste] or [reverse paste].



Figure 5.6 Forward or reverse pasting instructions at the specified position

5.3.5 Trajectory debugging

In the manual mode, there are two methods to debug whether the robot motion trajectory instructions in the program is correct: line by line debugging in the program command display area and entering the special motion trajectory debugging interface through the menu.

Line by line debugging of program instruction display area

elect a curve command in the program command display area on the touch screen, the curve command will become gray, and the task type and parameter setting area above the interface will be changed to be consistent with the command. The toolbar will be switched to Figure 5.7.

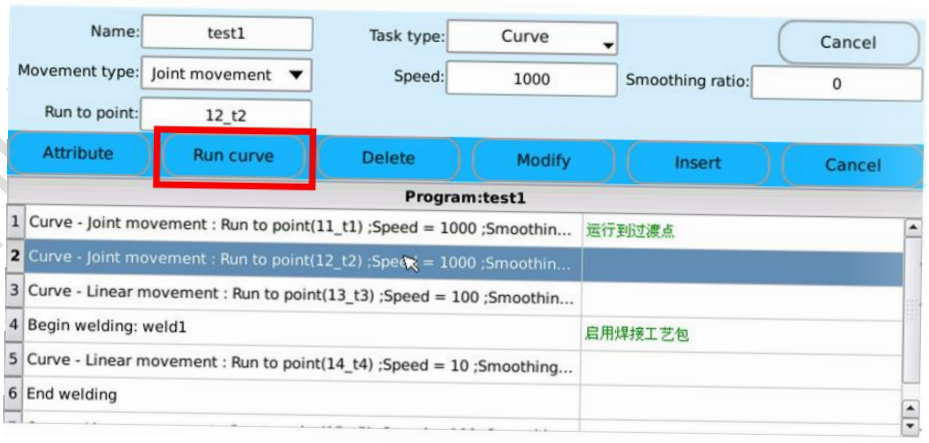


Figure 5.7 [Run curve] button will appear when selecting curve command

The [operation curve] button on the toolbar is a one-step debugging button only available for curve instructions. Under the condition of ensuring safety, press the "enable" button on the back of the teach pendant and click the [Run curve], and the robot will execute the curve instruction and run

to the "run to point" set by the curve.

Select the next curve instruction, continue to press any "enable" button on the back of the teach pendant, and click [run curve] for manual teaching to check whether the motion trajectory of the robot meets the design requirements.

Motion track debugging interface

The program includes not only a series of robot trajectory instructions executed in sequence, but also many basic instructions, process instructions, function instructions and logic instructions. In the program instruction display area, look for the motion instruction line by line for debugging, which is inefficient and easy to miss. Therefore, the system provides a special motion trajectory debugging interface.

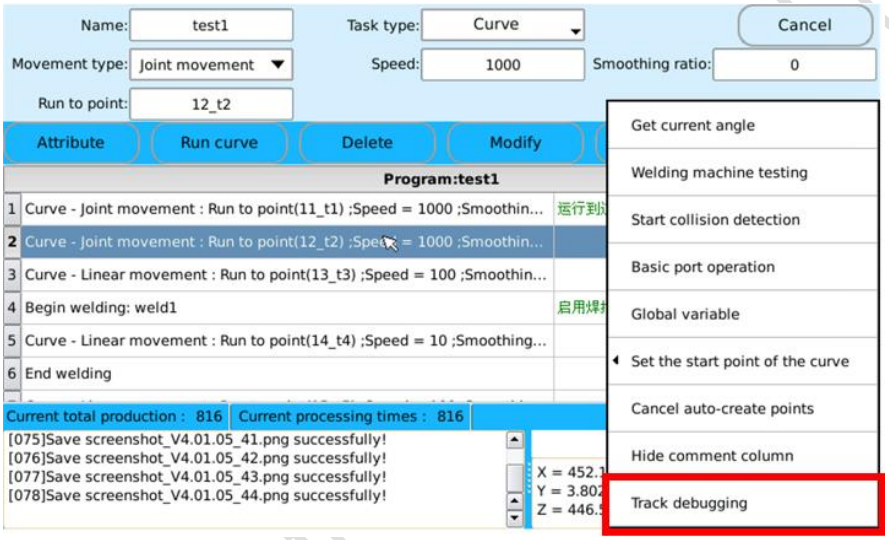


Figure 5.8 Menu [track debugging]

Click the [menu] key on the teach pendant, and click the [track debugging] key in the pop-up menu (as shown in Figure 5.8), and the motion track debugging interface will appear.

As shown in Figure 5.9, on the motion track debugging interface, other instructions in the program are shielded, and all curve instructions are presented visually in turn, so as to enable the user to verify the consistency between the running track of the program and the teaching track.

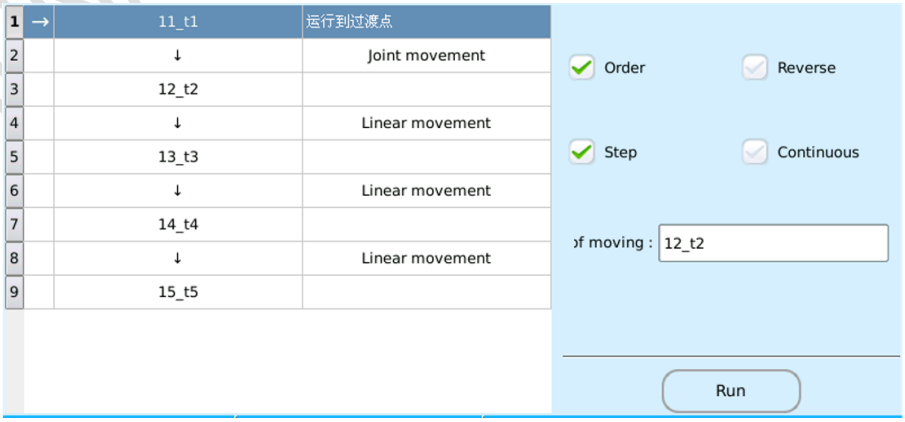


Figure 5.9 Motion trajectory debugging interface

In this interface, you can check "single step" or "continuous" operation by "sequence" or "reverse order". You can select the corresponding instruction to stop the program when it runs to the

instruction. Note that click the [run] button here to debug the motion track. It is a manual teaching process, and you need to press the "enable" button on the back of the teaching pendant at the same time.

Note:

(1) "Track debugging" is to extract curve tasks in the program and run them one by one to preview the running track of the program. Therefore, only curve instructions are executed and non-curve instructions are skipped. When the program contains logical statements "if", "while" and "Jmp", special attention should be paid: if there is an "if", "while" or "Jmp" that is not completely paired between the two points to be run, the result of judgment and jump is uncertain because the non-curve does not run. At this time, the track between these two points cannot complete automatic operation, which needs to be judged by the user, manually reset the starting point to skip such statements.

(2) During track operation, the interface will be set to read-only to prevent users from modifying configuration parameters during operation.

5.3.6 Modification of track points

There are two ways to modify the position and attitude of the track point in the program during the motion track debugging or due to subsequent changes in the environment: at the instruction of the program editing interface and at the point management interface.

Modify at the instruction of program editing interface

Enter the basic program editing interface, select the motion instruction line to be modified, and the instruction will turn gray. Click this instruction again, and the program editing toolbar will switch to Figure 5.10. The border of the text box after "run to point" will be highlighted.

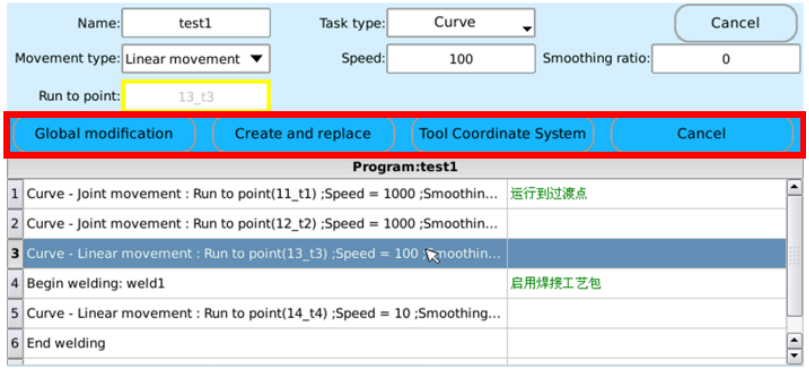


Figure 5.10 "point" modification in curve instruction

At this time, press the "enable" button to manually teach to a new position and attitude, and click:

- [global modification] - a prompt box appears, "are you sure to modify the joint angle of curve point * * to the joint angle of the current posture?", Click "yes" and the point data will be modified and saved. If other motion instructions in the program use the point data, it will also be modified synchronously.
- [Create and replace] - a prompt box appears, "do you want to create and replace curve points * * with the current joint angle?", Click "yes", a new "point" will be created and saved, and

the "motion to point" of the instruction will be replaced with the new point. At this time, the original * * point data will not be modified.

Modify in the point management interface

In the system, some key track points will be referenced by multiple programs for many times. At this time, if you need to modify the point data, you can implement it in the point management interface.

Click [manage] - [point] to enter the "point" management interface, as shown in Figure 5.11. Select the point to be modified, click the [details] button, the "edit box" of the selected "point" will appear, press the "enable" switch, manually teach to the new position and attitude, click "read the current point", and click "save" to complete the modification of "point".

Note: if the modified "point" is referenced, be sure to switch to the "program" management interface and [check] the program that "references" the modified "point" (as shown in Figure 5.12), otherwise the motion track in the program will make mistakes and cause accidents.

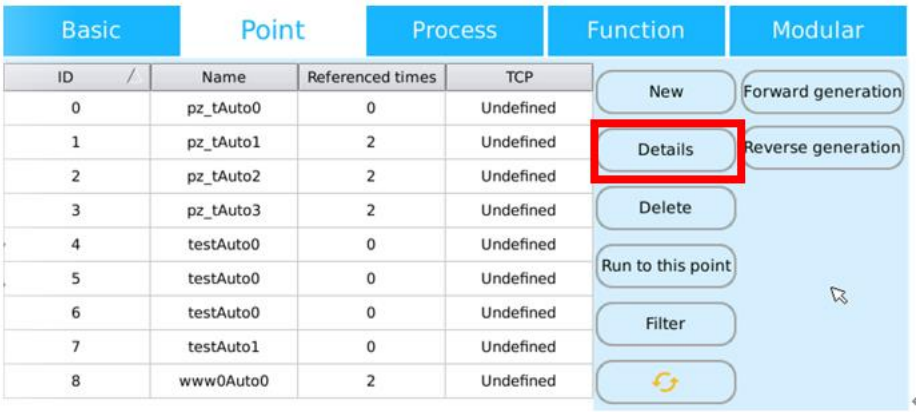


Figure 5.11 Click the [details] button to modify the data of the point

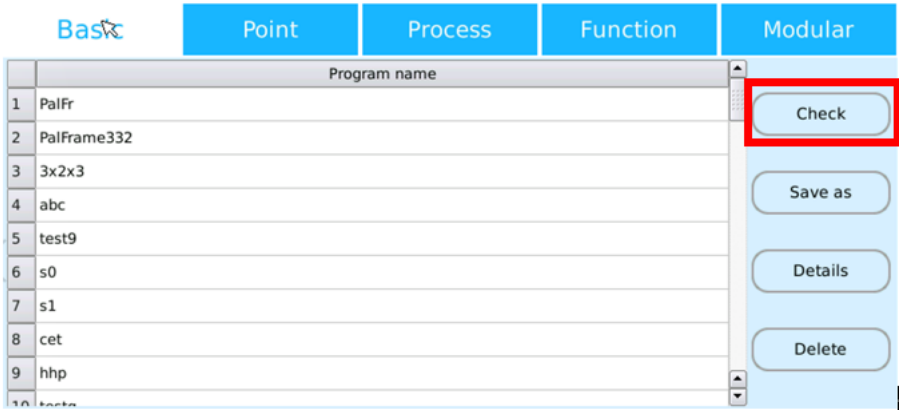


Figure 5.12 Program check

5.3.7 Program saving

After program editing and debugging, you must click the [Temporary save] or [Formal save] button to complete the establishment of "basic program", as shown in Figure 5.13.

[Temporary save]: Continue to stay in the program editing interface.

[Formal save]: Automatically jump to the basic program management interface.

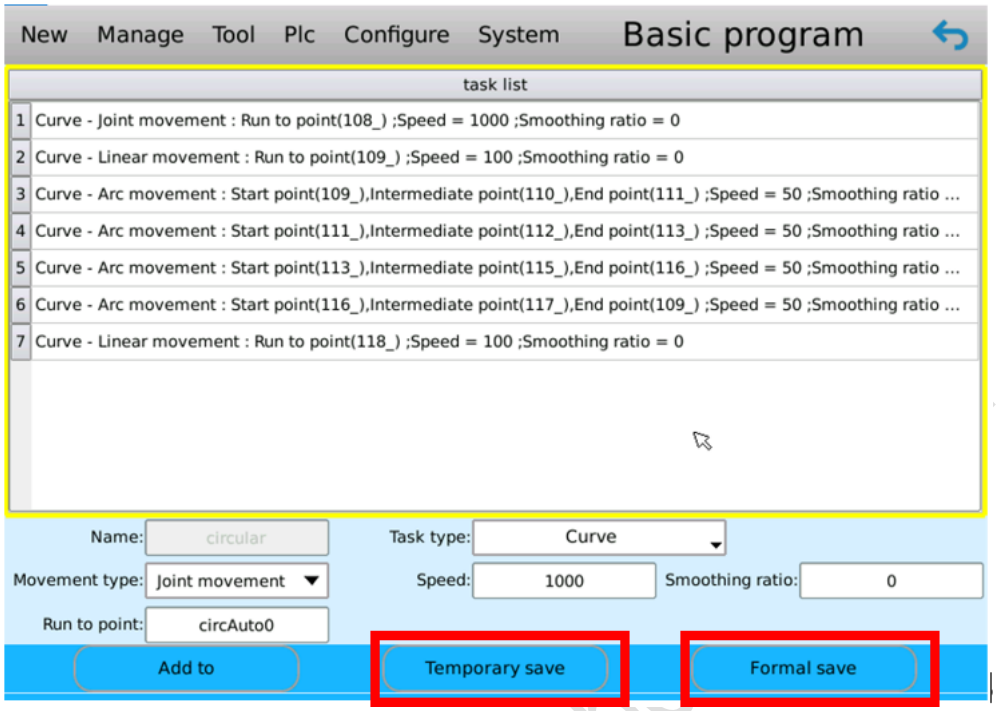


Figure 5.13 Program saving

5.4 Programming interface menu function

Click the [menu]key in the lower right corner of the teaching pendant, and a menu will appear in the lower right corner of the interface, as shown in Figure 5.14.

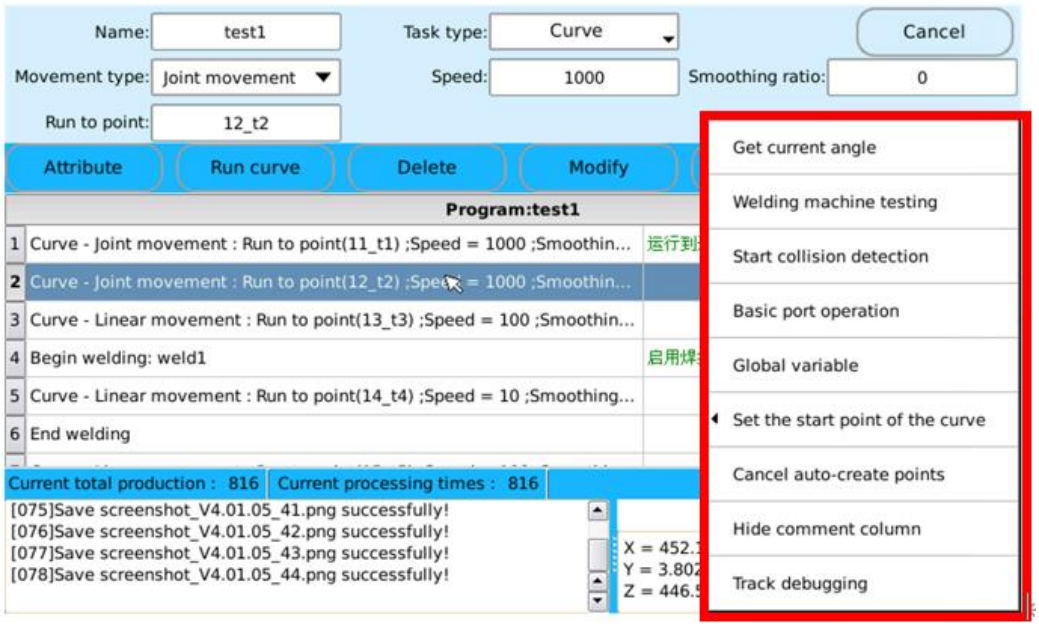


Figure 5.14 Menu of programming interface

The function description of the menu is shown in Table 5.1.

Table 5.1 Menu function description

Number	Item	Function description
1	Basic port operation	Monitor the status of input port and relay port, and check the output effect by setting output port and analog quantity.
2	Get current angle	The operation record display area displays the joint angle data of the current point.
3	Global variable	Open the global variable interface to view the global variable data.
4	Welding machine resting	Adjust the welding machine according to the welding process.
5	Set the start point of the curve	Set the starting point of the program running curve.
6	Auto-create point	Auto name the "run to point" in curve instruction.
7	Hide comment column	Hide / show the comment column in the program instruction display area.
8	Track debugging	Enter the program track debugging interface.
9	Start collision detection	Select the created collision detection function package and alarm output port to make the robot in collision detection state and stop moving immediately in case of collision.

5.5 Modular program

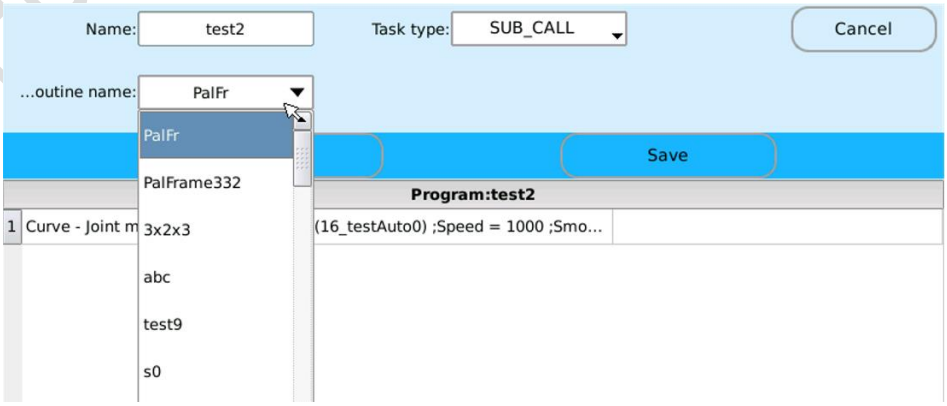
Modular programming refers to dividing a large program according to its functions, writing subroutines to complete the determined functions, and then calling it in the main program. To this end, betrun system provides two modular programming methods:

(1) Using SUB_CALL instruction in basic program

(2) Enter [modular program] to create "modular program"

5.5.1 Use SUB_ CALL instruction

As shown in Figure 5.15, you can use sub_ The CALL instruction calls another basic program in a basic program. Click "save" and the subroutine will be saved into the program as an instruction.



HUNAN BETRUN Robot Co., Ltd.

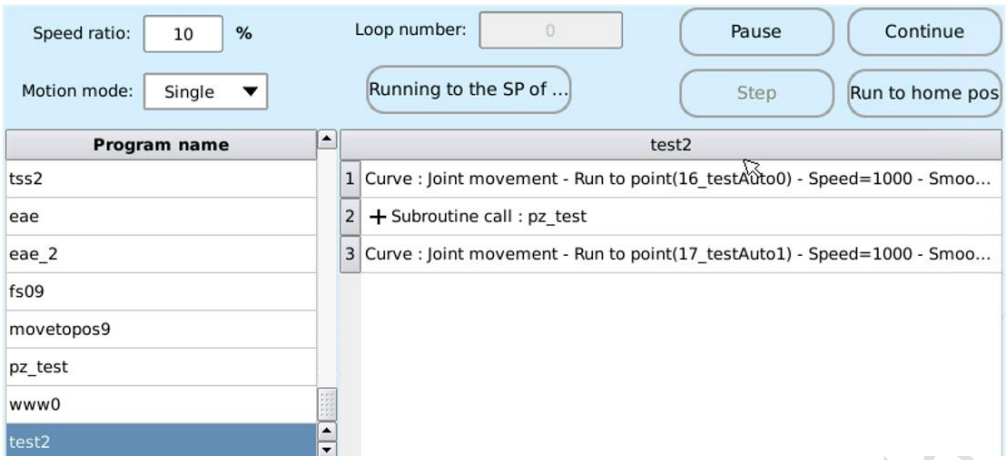


Figure 5.15 Using SUB_CALL in basic program

Switch the teaching pendant to the "automatic mode" interface, and a plus sign will appear in front of the Subroutine instruction. As shown in Figure 5.16, click the plus sign and the instruction line of the called subroutine will appear.

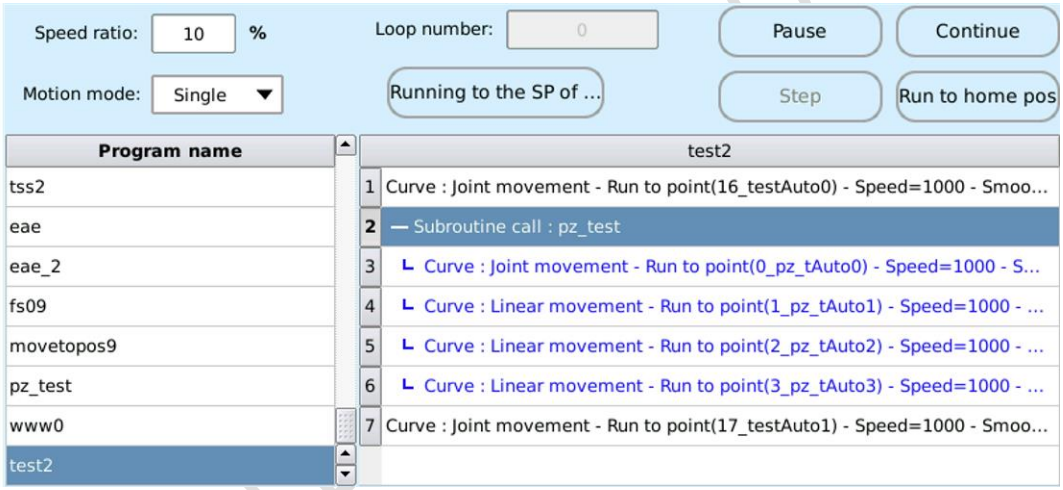


Figure 5.16 Presentation of subprogram in automatic mode

In this way, the program design is simpler and more intuitive, and the readability and maintainability of the program are improved. The disadvantage is that when running the main program, the preprocessing of the subroutine is running to SUB_CALL instruction. If there are many curve instructions in the subroutine and they need to be smoothed, the subroutine preprocessing during operation will affect the operation efficiency.

5.5.2 Create a "modular program"

Enter the modular program programming interface. The programming process of the modular program is the same as that of the basic program, but there are multiple "pcall" task types in the "modular program". In "modular program", you can call "subroutine" through "pcall" task type to generate "modular program". The "subroutines" called by the "pcall" task type are all completed "basic programs".

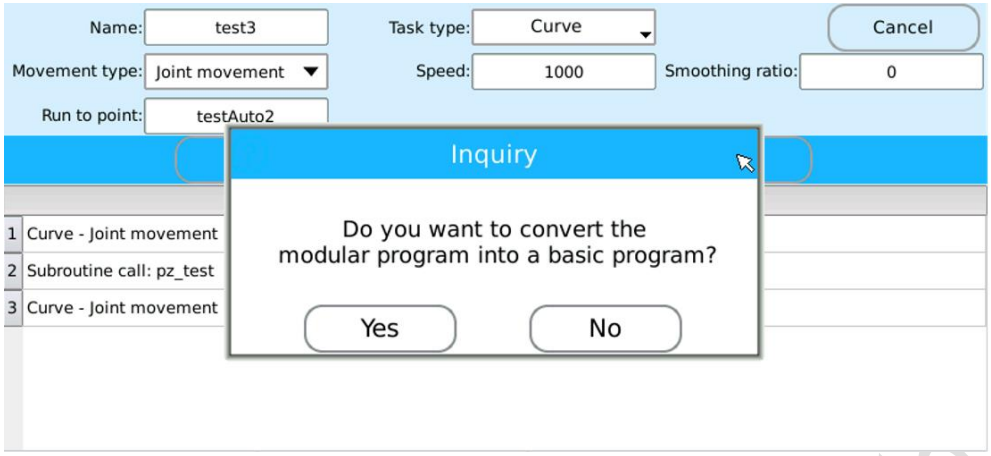


Figure 5.17 "PCALL" task type and usage

After the modular program is created, click [Save] to save it into a "modular program" in [management] - [modular program], and then you will be prompted: "Do you want to convert the modular program into a basic program?", as shown in Figure 5.17. After selecting [Yes], the system will convert the "modular program" into a "basic program" and save it in the "basic program".

In this way, when the teach pendant is switched to "automatic mode", only the generated basic program will appear in the list, and the program instruction will not be preprocessed in the process when the program is running, so as to ensure the operation efficiency. The disadvantage is that when the program is long, its readability is poor, which is not convenient for program testing.

5.6 Program execution in automatic mode

After the program is created, switch the teach pendant to "automatic mode" to run the program.

5.6.1 Program automatic positioning

For the last newly created program or the last modified and saved program, switch to "automatic mode", and the system will automatically locate the name of the program and automatically query its details, and locate the task line of the last operation of the program in the program (operations include creating, modifying, inserting, and deleting the task line).

5.6.2 Parameter setting

As shown in Figure 5.18 below, various parameters need to be set on the configuration page before "operation", including movement speed, movement times and movement mode.

The description of parameters is shown in Table 5.2.

For safety, it's best to slow down the " speed" before running the program, select the "single" motion mode to test the running track, and then set the "speed" and "motion mode" according to the

HUNAN Betrun Robot Co.,Ltd.

operation requirements after there is no problem

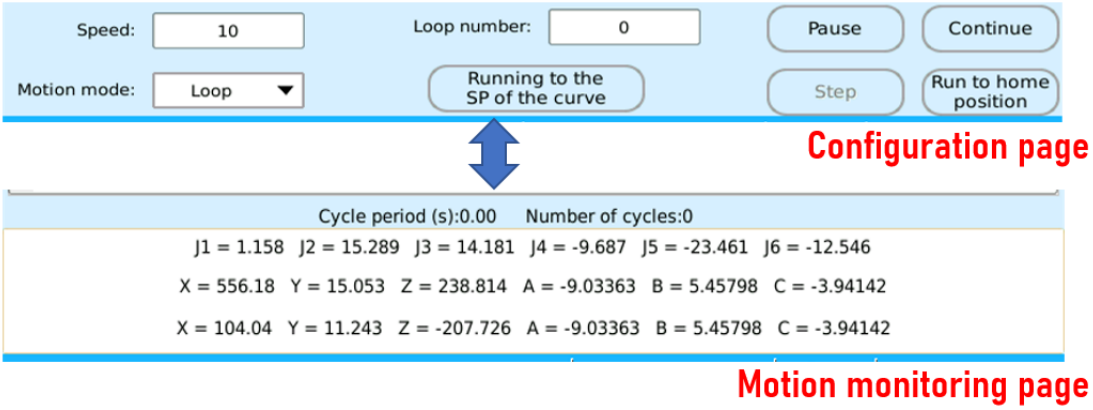


Figure 5.18 pages in automatic mode interface

Table 5.2 Description of program operation parameters

Number	Parameters	Description
1	Speed	Set the overall speed of "program", with a range of 1-100 (percentage).
2	Loop number	Used in "Loop" motion mode to determine the number of cycles. Number of movements = 0: the program will cycle indefinitely.
3	Motion mode	Three motion modes are available: a. "Single" motion: let the program perform a complete operation, that is, the motion track runs from the "start point" along the track set by the "program" to the "end point". b. "Step" motion: let the program start from the "start point" and execute "step by step" along the track set by the "program". Specific format: select the "program name" to be executed, press the physical key [program start] on the teach pendant, wait until the first step of the robot body is completed, then click the [step] key to proceed to the next step, and then click the [single] key repeatedly to complete the single step operation. c. "Loop" motion: let the program "cycle" run, which should be used in conjunction with the setting of "motion times". When the parameter in "motion times" is set to "0", the circular motion will become the "infinite cycle" motion mode.

In automatic mode: the program is displayed as the configuration page by default before startup. After the program starts running, the interface will automatically switch to the motion monitoring page.

Click the program task list to switch the motion monitoring page back to the configuration page. When the interface is idle for 5 seconds, it will automatically switch back to the motion monitoring page.

5.6.3 Use of keys

In the "automatic mode" mode of the teach pendant, see Table 1.3 in Chapter 1 for the description of key functions.

HUNAN Betrun Robot Co., Ltd.

5.7 Run the program by using external keys

After programming and testing with the teach pendant, by setting the port configuration, you can run a job program using the external keys instead of entering the teach pendant interface.

Step 1: install physical keys on the control box and connect the input port of the main controller.

Step 2: in [user management], select the check box of [automatic login] and complete the "automatic login" function setting according to the prompt, as shown in Figure 5.19.

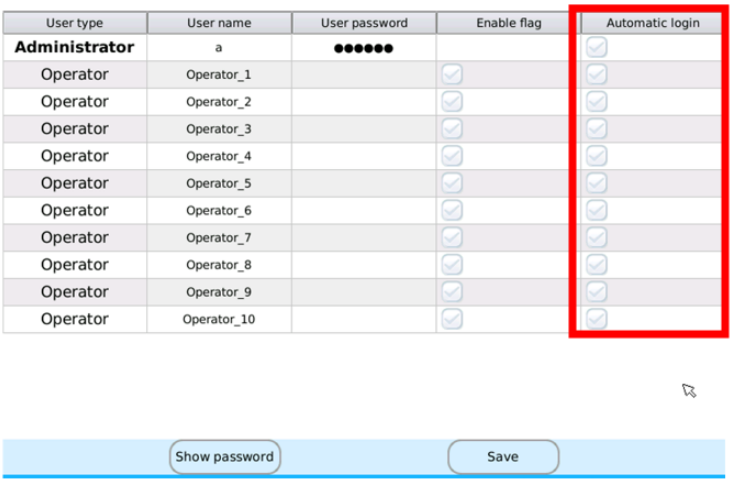


Figure 5.19 Select the check box of [automatic login]

Step 3: Select [configuration] - [environment configuration], as shown in Figure 5.20. Check "Load running parameters automatically". The parameters set in "automatic mode" (such as "speed ratio", "motion mode" and "loop number") will be saved, and the parameters will be loaded automatically after the system is restarted.

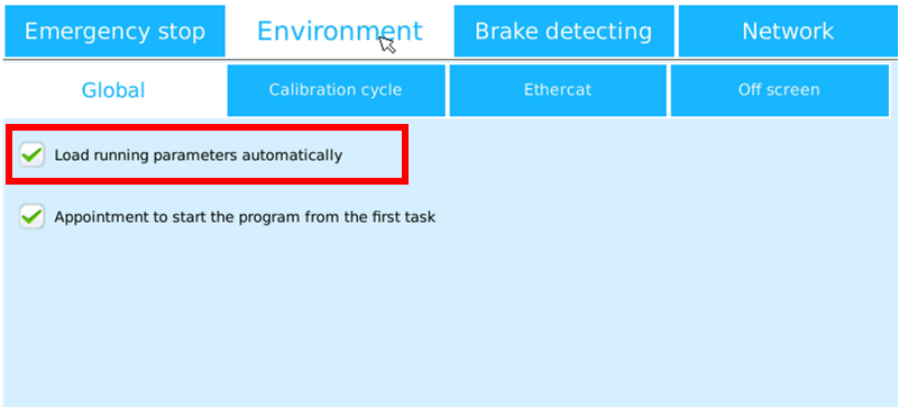


Figure 5.20 Check "Load running parameters automatically"

Step 4: select [port configuration], the system provides 8 special functions. Enter the port according to the physical keys installed in the control box, and configure the special function, as shown in Figure 5.21.

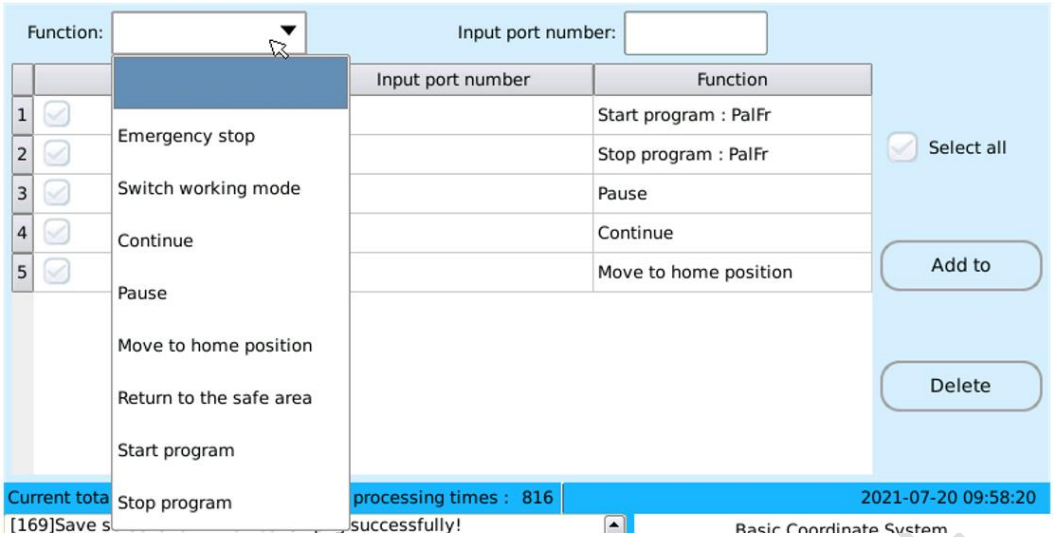


Figure 5.21 Port configuration

Note:

- The same special function can be configured many times with different I/O input ports. If the I/O input port is configured as a special function, it can no longer be used as a common I/O input port.
- There are some differences between "emergency stop" function and other "special functions": when configuring "emergency stop" function, it is "valid" when I/O input port level is low; When configuring other "special functions", it is "valid" when I/O input port level is high.

Step 5: set the operation parameters in the “automatic mode” interface of the teaching pendant, execute the selected program, and the parameters during program operation will be “saved”. Restart the system, you can control the program by using the external keys.

5.8 Execution of the reservation programs

When a robot needs to cooperate with many operators to complete multiple operations, the operators can click the [reservation] button to make a reservation after preparing the workpiece. The robot will complete the corresponding program work in turn according to the reservation order of different operators.

5.8.1 Reservation Settings

In the manual interface of the teach pendant, click [reservation] to enter the "program reservation" setting interface, as shown in Figure 5.22. The description of each parameter is shown in the Table 5.3.

After setting the parameters, click the [add to] button to complete the reservation setting of one program, repeat the operation, select “enable” to take effect. The number of “enable” reservation procedures cannot exceed 10.

HUNAN Betrun Robot Co., L

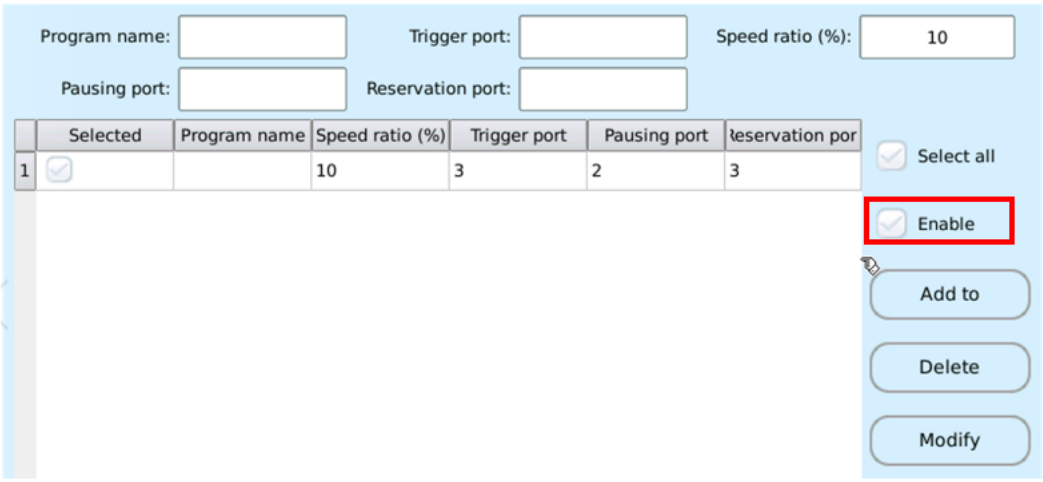


Figure 5.22 Program reservation execution settings

Table 5.3 Parameter description of program reservation setting interface

Number	Parameter	Description
1	Program name	Select a basic program for program reservation.
2	Trigger port	Set the "IO input" port number that triggers the "reservation program", and the "IO input" port number cannot coincide with the "function configuration" port number.
3	Speed ratio	Set the running speed ratio of "reservation program", the unit is "%", and the value is an integer of "1-100".
4	Pausing port (indication)	Set the "IO output" indication port number when the "reservation program" pauses or stops. When the "reservation program" runs, the output is "low level". Otherwise, the output is "high level".
5	Reservation port (indication)	Set the "IO output" port number after the program reservation is successful. If an external “LED light” is connected, the "LED light" will start to "flash" when the program reservation is successful and has not been "executed"; When the program starts "running", the "LED light" will change from "flashing" to "on".
6	Enable	Select "enable" to take effect.

5.8.2 Execution of The Reservation Programs

When the system uses the "reservation program", it must be "triggered" in the "automatic mode" state to run. The "sequence" of the running program runs according to the time sequence of successful triggering. When the "reservation program" runs successfully and the program is "triggered" again, it will not be counted.

When multiple programs are reserved successful, and the "program stop" or "emergency stop" button is pressed through the teach pendant, all "reservation programs" will be stopped and the "count" will be cleared, including "reservation programs" of "being executed" and "not executed". When executing the reservation program, if it is in the "pause" state, press the "trigger" button corresponding to the program again, and the reservation program will "continue" from the current

HUNAN Betrun Robot Co., Ltd.

command line; If you need to start from the first task, you need to select [configuration] - [other configuration] - [environment configuration], as shown in Figure 5.23, and check "Appointment to start the program from the first task".

Note: the [stop] external key can only stop the program of the current operator and will not affect the program execution of other operators.

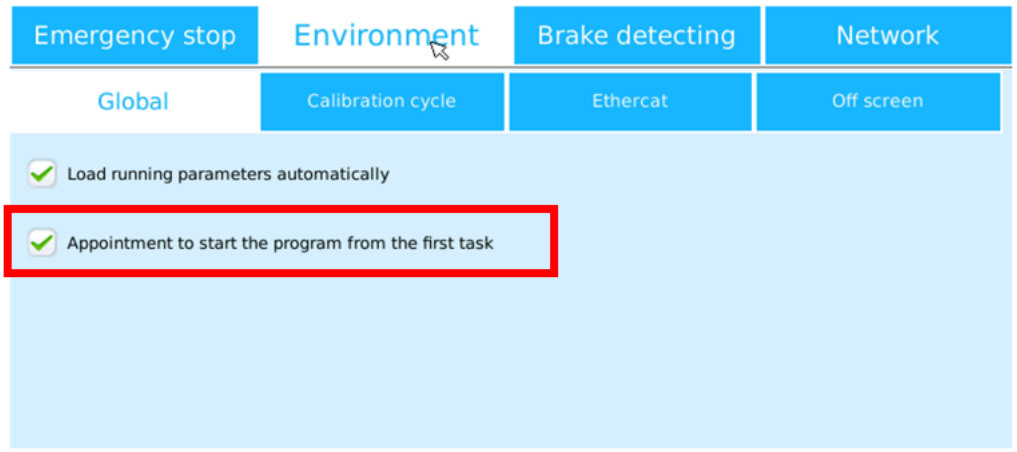


Figure 5.23 Program reservation related configuration

5.9 Program management

After the program is established, sometimes the set "point" position changes, or the program needs to be modified according to the robot operation requirements. At this time, it should be realized through "program management".

Click [management] - [basic program], as shown in Figure 5.24, and the program name will be displayed in the list.

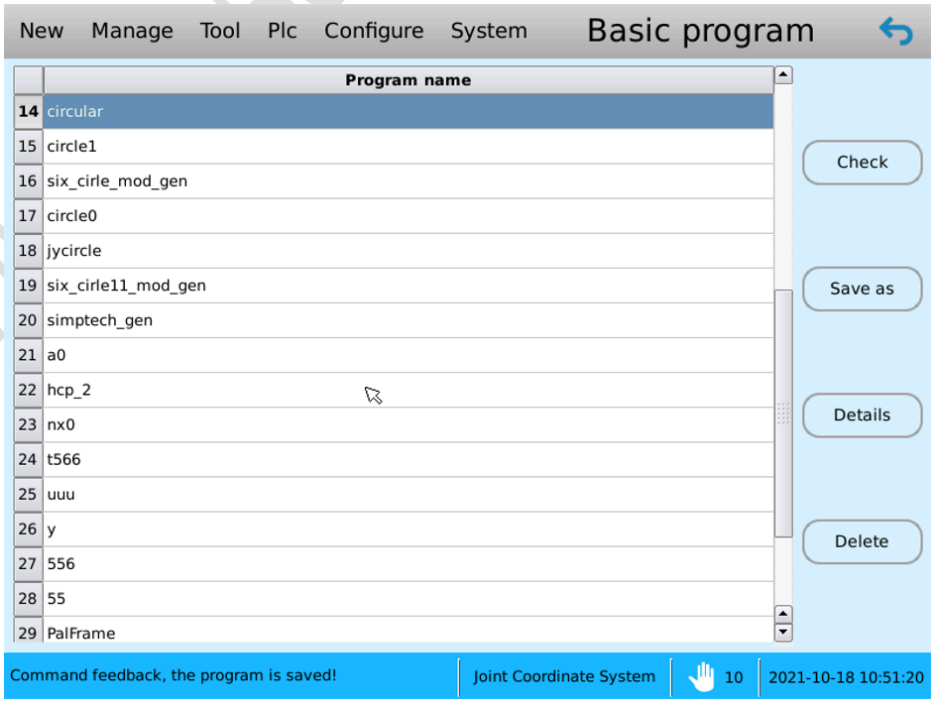


Figure 5.24 Basic program management interface

The management of "basic program" is divided into "check", "save as", "details" and "delete" of the program. The functions of each key are shown in Table 5.4.

Table 5.6 Function description of "basic program" management keys

Number	Key	Description
1	Check	When it is used in conjunction with the management of "points", when a "reference point" in the program changes, it must "check" the program (all other programs that have referenced this "change point" need to "check"), otherwise the motion trajectory in the program will make mistakes and cause accidents.
2	Save as	Select the "program name", click the "save as" button to enter the basic program creation interface of the selected "program", modify and save the "program name", and then save the "program" as a program with a new name under the condition that the original program remains unchanged.
3	Details	You can return to the basic program creation interface to "delete", "modify" and "insert" the selected program.
4	Delete	Delete the completed program.

Co., Ltd.

6 PLC programming

In order to facilitate the programming habits of operators, the system provides PLC programming mode, so that users can use the standard PLC programming language to write specific control applications.

6.1 PLC programming environment

The system provides two PLC programming environments: ladder diagram and instruction table. Click [PLC edit] to enter the PLC ladder diagram programming interface shown in Figure 6.1. Click [instruction list] on the interface to switch to PLC instruction list programming interface, as shown in Figure 6.2.

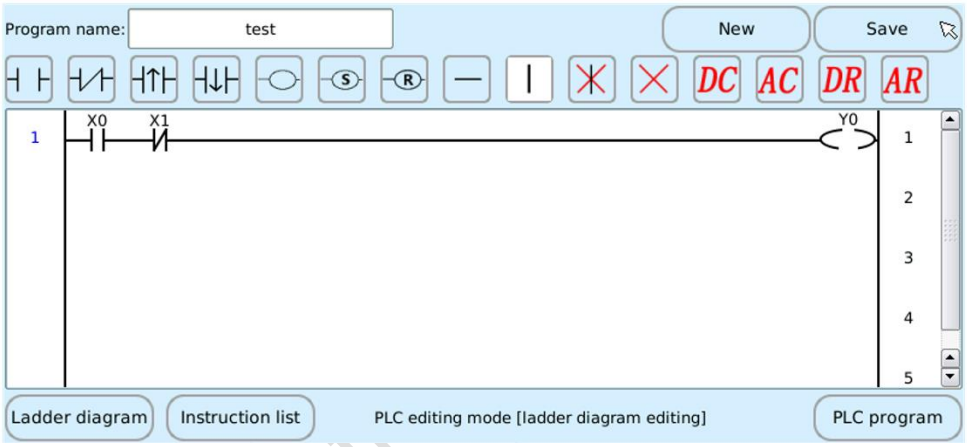


Figure 6.1 PLC programming ladder diagram

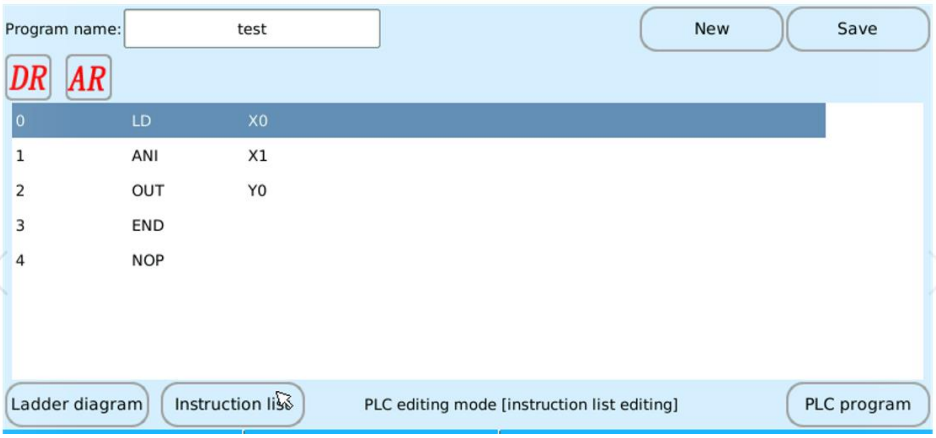


Figure 6.2 PLC instruction list programming interface

The capacity of PLC program is 800 component symbols, and the basic instructions can realize the logic control of internal relay, timer, counter, input relay X and output relay y. The ladder column range is 0 ~ 100, and the command line range is 0 ~ 1000.

6.1.1 Ladder component symbols

The basic symbols of components provided by PLC ladder diagram programming interface are shown in Table 6.1.

Table 6.1 Component symbol description

Number	Symbol name	Symbolic graphics	Description
1	Normally open contact		Normally open contact of each element (relay)
2	Normally closed contact		Normally closed contact of each element (relay)
3	Direct output coil		Coil of each element (relay)
4	Set output coil		Coil of each element (relay) (set hold)
5	Reset output coil		Coil of each element (relay) (reset hold)
6	Horizontal connection		Used to connect the contacts and coils of each relay horizontally
7	Vertical connection		Used to vertically connect the contacts and coils of each relay (usually for branch)

6.1.2 Basic logic instruction in instruction table

There are 15 basic logic instructions in the PLC instruction table, which are used to realize the basic logic control, as shown in Table 6.2.

Table 6.2 Basic logic instruction in instruction table

Number	Instruction name	Description
1	LD, LDI instructions	LD: normally open contact and bus connection command. LDI: normally closed contact and bus connection command. LD and LDI instructions can be used for X, Y, S, A, M, C and t (including TGT, TGH, AGT and AGH). They can also be used for the start point of branch circuit in combination with AND and ORB instructions. Out: relay coil output command. The out command can be used for Y, M, C and T (including TGT, TGH, AGT and AGH), but cannot be used for input relay X, status relay S and alarm relay A.
2	AND, ANI instructions	AND: normally open contact series connection command. ANI: normally closed contact series connection command. AND and ANI instructions can be used for X, Y, M, C and T (including TGT, TGH, AGT and AGH).

HUNAN Betrun Robot Co., Ltd.

3	OR, ORI instructions	OR: normally open contact parallel connection command. ORI: normally closed contact parallel connection command. OR and ORI instructions can be used for X, Y, M, C and T (including TGT, TGH, AGT and AGH).
4	ORB instruction	ORB: parallel connection command of series circuit block. The circuit formed by connecting more than two contacts in series is called "series circuit block". When the series circuit blocks are connected in parallel, ORB command is used.
5	ANB instruction	ANB: serial connection command of parallel circuit block. The ANB command connects the parallel circuit block in series with the previous circuit. Before using the ANB command, the internal connection of the parallel circuit block should be completed.
6	MPS, MRD, MPP Stack memory and multiple output instructions	MPS, MRD and MPP instructions are stack in, stack read and stack out instructions respectively, which are used in multiple output circuits.
7	SET, RST instructions	SET: set instruction, the instruction to hold the operation. RET: reset command, the command to keep the operation reset. SET instruction can be used for Y and M, RST instruction can be used for Y, M, C and T (including TGT, TGH, AGT and AGH).
8	NOP, END instructions	NOP: empty operation instruction. END: end instruction, indicating the end of the program.

6.1.3 Run PLC program

Click [PLC] - [Run PLC] to enter the PLC program execution interface shown in Figure 6.3. In this interface, you can run, stop and monitor the PLC program, and view the program details.

In the interface shown in Figure 6.3, the automatic operation of PLC program can also be realized through setting. Click the [PLC program] button in the lower right corner, then select the program to be run automatically, click the [menu] key in the lower right corner, and select "Setup program self-starting" in the pop-up menu item, as shown in Figure 6.4. Then power off and restart, and the PLC program will run automatically.

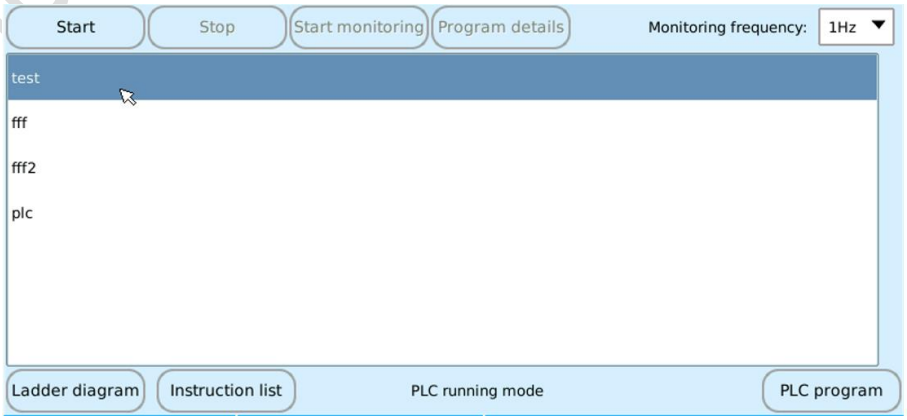


Figure 6.3 PLC program execution interface

HUNAN Betrun Robot Co., Ltd.

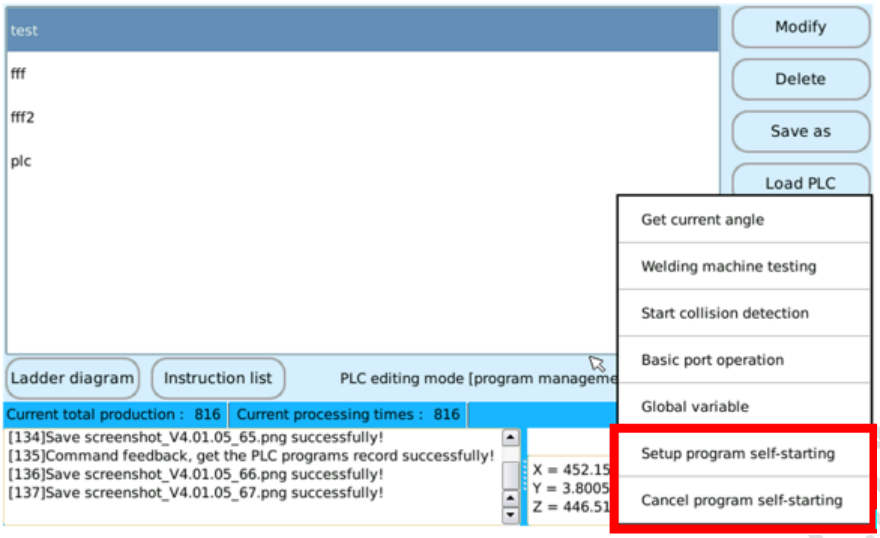


Figure 6.4 Setting the automatic operation of PLC program

6.1.4 Load PLC program

The "PLC Editor" software can be used on the PC to complete PLC programming, and finally generate the PLC program required by the control system.

Copy the PLC program file to the folder named PLC under the root directory of USB flash disk (create a new folder named PLC under the root directory). The teach pendant enters the [PLC-edit] interface, click [PLC program] - [load], and select the PLC program to be loaded in the pop-up list (the loaded PLC program name cannot be the same as the program name already in the control system).

6.1.5 PLC program encryption

The system provides PLC program encryption function. As shown in Figure 6.5, the password can be set in this interface. If you need to check the PLC program details later, you need to enter the password, as shown in Figure 6.6.

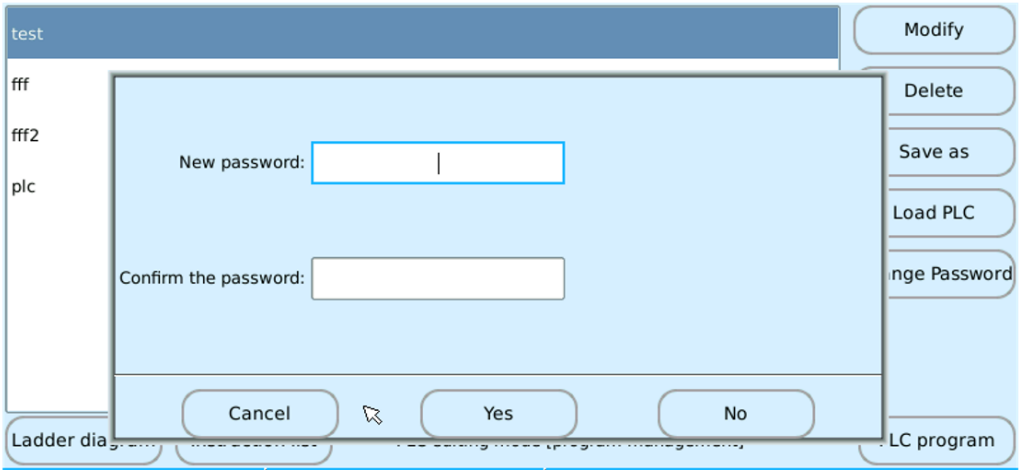


Figure 6.5 Setting password for PLC program

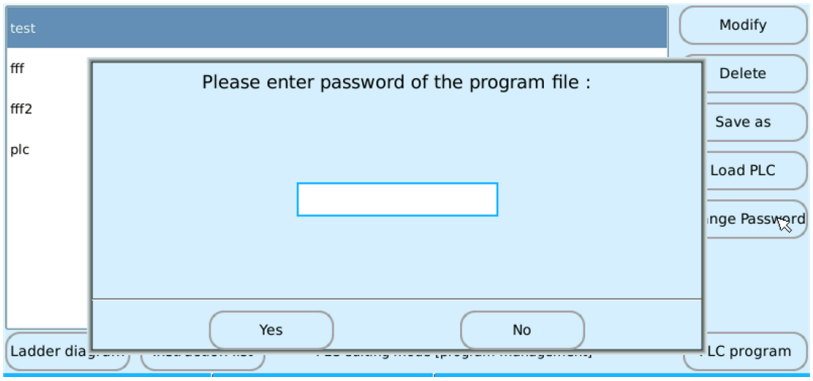


Figure 6.6 Enter password to view PLC program details

6.2 PLC component library

Components are address symbols used to identify signals processed in PLC. The component number is the serial number assigned to the component. Components in the PLC system are represented by X, Y, M, etc.

See Appendix 3 of the manual for the definition of PLC elements.

7 I/O communication of Betrun system

Betrun system provides rich I / O communication interfaces, including standard I / O circuit, serial port, network port, USB interface, etc., which can easily realize the communication with peripheral devices.

7.1 Interface signal definition

The definitions of comm and AVO interfaces of the main controller are shown in Table 7.1 and table 7.2. By default, there is no analog output module in the controller and additional configuration is required.

Table 7.1 Definition of RS485 and RS232

Pin	Signal	Definition	Pin	Signal	Definition
1	CAN1H	CAN1_H	6	/	
2	CAN1L	CAN1_L	7	U5TX	232_TX
3	U4B	485_B	8	U5RX	232_RX
4	U4A	485_A	9	GND	232_GND/
5	/				

Table 7.2 Definition of analog output interface

Pin	Signal	Definition	Pin	Signal	Definition
1	AOUT1	Analog output 1	6	/	Empty, no signal
2	AOUT2	Analog output 2	7	/	Empty, no signal
3	AOUT3	Analog output 3 (not open)	8		
4	AOUT4	Analog output 4 (not open)	9		
5	GND	Analog output ground			

7.2 Standard I/O circuit

The standard I/O port circuit of the system is divided into three types:

(1) 20 optocoupler isolation inputs

The circuit diagram of optocoupler isolation input is shown in Figure 7.1.

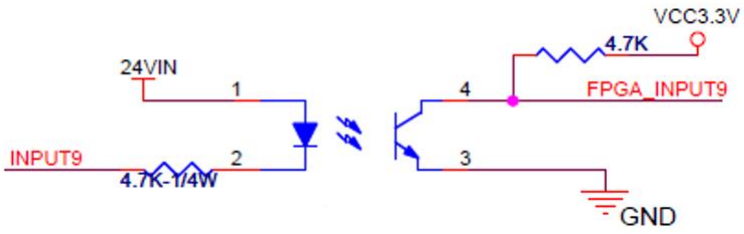


Figure 7.1 Optocoupler isolation input circuit

HUNAN BETRUN Robot Co., Ltd.

(2) 10 open collector outputs

The open collector output circuit is shown in Figure 7.2. The default system output is low level and the triode is off; The system output is at high level, and the triode is on. When the collector is open and the output port is connected externally, pay attention to current limiting, and do not exceed the maximum collector current of the triode.

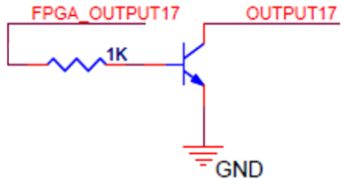


Figure 7.2 Open collector output circuit

(3) 8 relay outputs

The relay output circuit is shown in Figure 7.3, and the ports are normally closed NC, common terminal COM and normally open NO. By default, the system outputs low level, the two contacts of common terminal COM and normally closed NC are closed, and they are disconnected from normally open NO. When the system output is high level, the common terminal COM is disconnected from the normally closed NC and switched to the normally open NO contact.

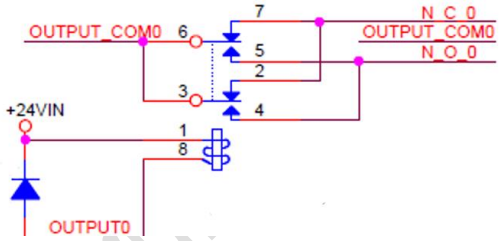


Figure 7.3 Relay output circuit

7.3 Basic port operation

The system provides input and output related program instructions for programming. In addition, click the [menu] key in the lower right corner of the teach pendant, and a menu will appear in the lower right corner of the interface.

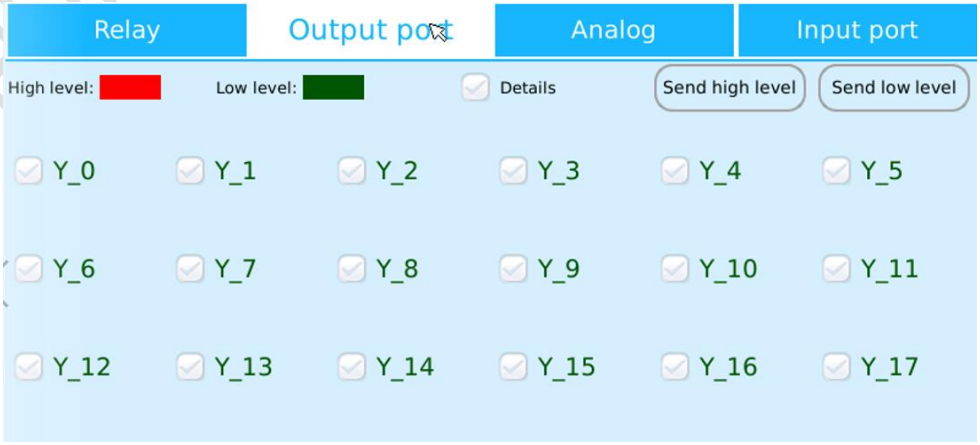


Figure 7.4 Basic port operation interface

Click [Basic port operation] to enter the basic port operation interface, as shown in Figure 7.4,

including "monitoring relay port", "setting output port", "setting analog quantity" and "monitoring input port".

7.3.1 Monitor relay port

As shown in Figure 7.5, after selecting "Relay", click the [add to] key to pop up a prompt box of "add single or multiple relays", enter the "relay port" number to be monitored, and click the [set] key to complete the "add" of "relays". When multiple relays are "added" at the same time, the serial numbers are connected by "underline".

After completing the "add" of "relays", you can carry out "status monitoring", "status setting" and "delete" operations on these "relays".

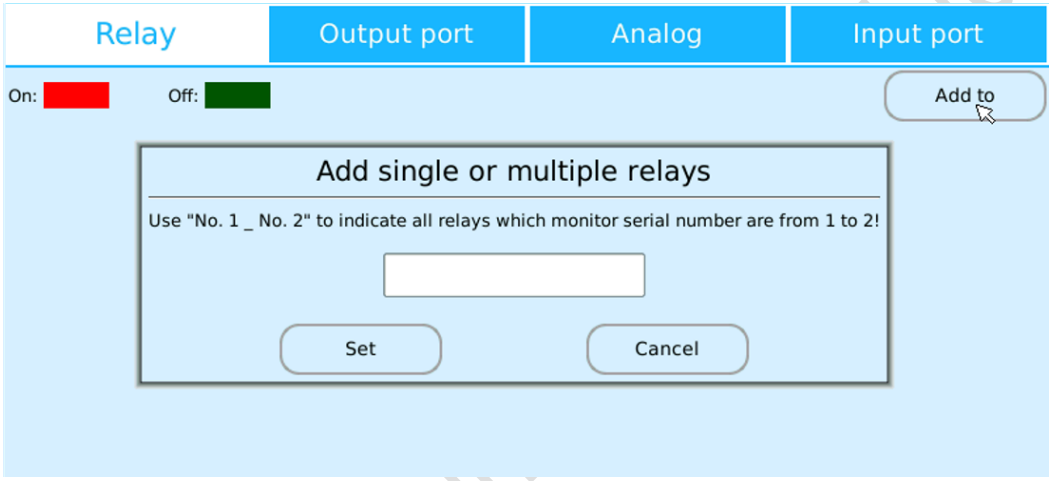


Figure 7.5 Monitor relay port

7.3.2 Set output port

As shown in Figure 7.6, after selecting "Output port", you can perform "status monitoring" and "set status" operations on the "output port". The "set status" operation means that the "output port" can be manually "send high level" and "send low level".

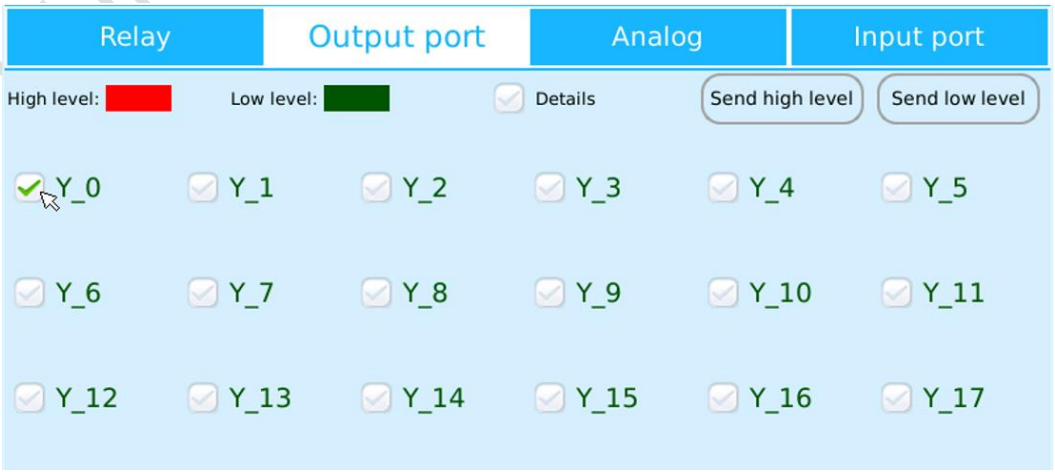


Figure 7.6 Set output port

7.3.3 Set analog quantity

As shown in Figure 7.7, after selecting "Analog", the "analog value" output can be set for the "analog quantity output channel" of the system. Among them, the number of analog channels depends on the actual situation of the system, and 02/4 channels are possible.

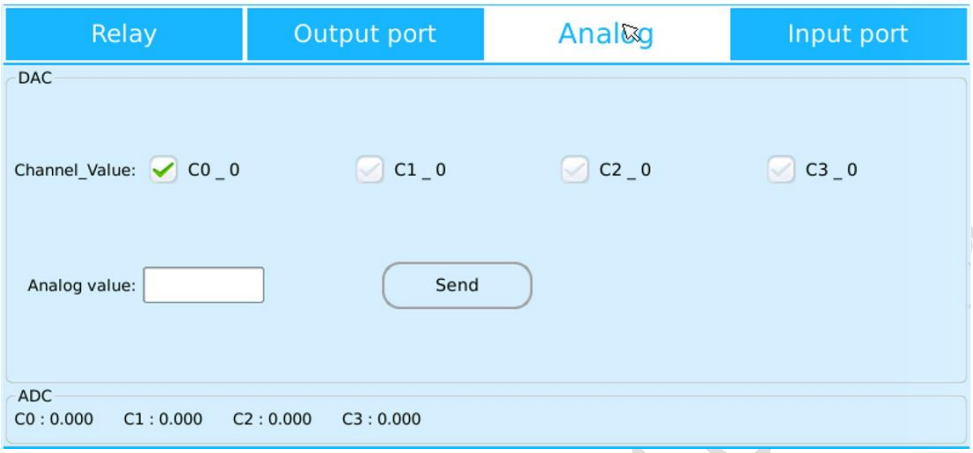


Figure 7.7 Set analog quantity

After inserting AVO module, the main controller provides 2 channels of "analog" output channels, which are "0" and "1" respectively, and the output range of "analog value" is [0, 10V]. The specific setting steps are as follows: first select the "analog" output channel, set the "analog value", and click the [send] button to complete the "analog output" setting.

7.3.4 Monitor input port

As shown in Figure 7.8, after selecting "input port", the system "input port" can be "status monitored".

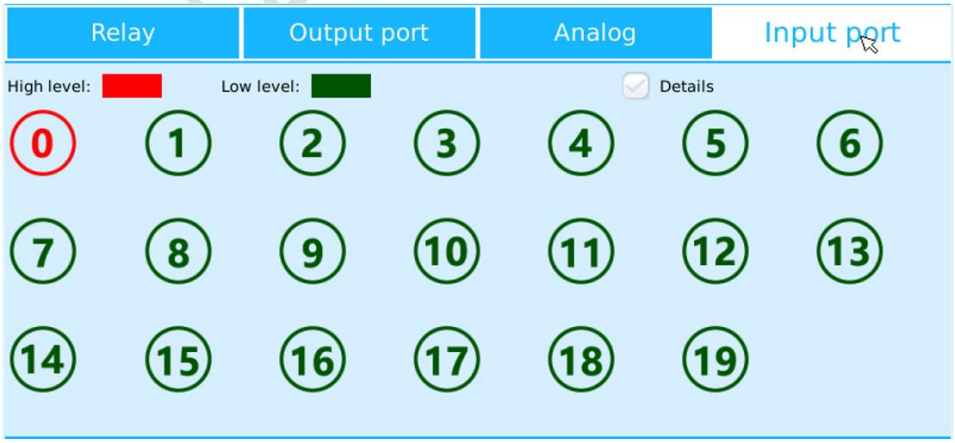


Figure 7.8 Monitor input port

In the interface, after "checking" the "details" function, you can add "remarks" information to each "input port".

When the system is externally connected with "IO expansion module", you can press the "menu" key and select "Show expansion port", and then you can perform "status monitoring" and "setting

status" operations on the "input port", "output port" and "analog channel" in the "IO expansion module".

8 Software configuration and use

Betrun system provides port configuration, automatic positioning of programs and tasks, custom alarm and other functions, which is convenient for operators to quickly complete program programming and debugging.

8.1 Port configuration

In the manual interface, click [tool]-[port], including function port, indication port and program reservation. The procedure reservation method is detailed in Section 5.8 of Chapter 5 of the manual.

8.1.1 Function port

Function port can map the input port to a function related to the control system, such as start program, stop program, etc., as shown in Figure 8.1.

At present, the special function keys that can be set directly include: emergency stop, switching working mode, continue, pause, return to zero, return to safety zone, start program, and stop program. The operator does not need programming, just set the function corresponding to the corresponding input key, and then directly control the program operation through the external key.

I/O input port number "0-19" can configure these functions arbitrarily. Different I/O input ports can be configured for the same special function, but only one special function can be configured for the same I/O input port, and the configuration will take effect only after the "add" button is clicked; If the I/O input port is configured with special functions, it can no longer be used as an ordinary I/O input port. Only after the special function of "I/O input port" configuration is "deleted", the "I/O input port" can be used as a normal I/O input port again. See Chapter 5, section 5.7 of the manual for more instructions.

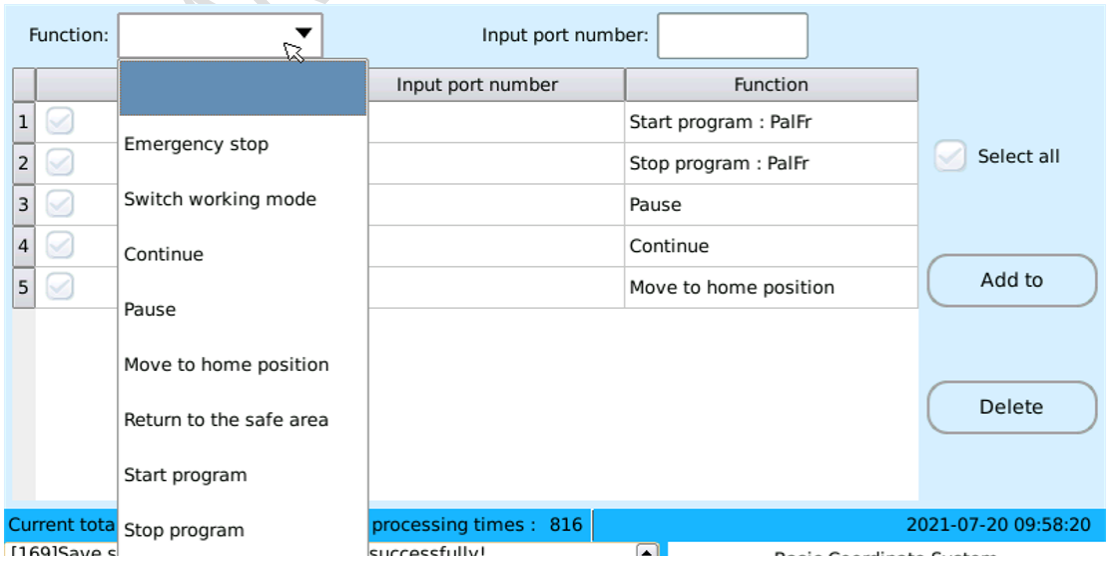


Figure 8.1 Function port setting

8.1.2 Indication port

The indicating port can map the output port to a certain state of the system, such as running state, servo state, alarm state, etc., so that the user can judge the state of the current program through output, as shown in Figure 8.2.

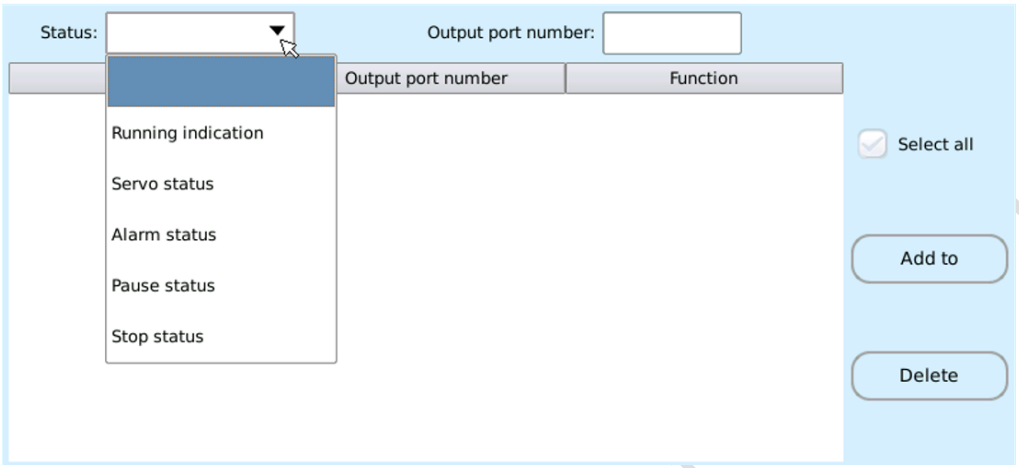


Figure 8.2 Indication port setting

8.2 Custom alarm prompt

The operator can define the alarm prompt information and map it to the output element w of PLC in turn, as shown in Figure 8.3.

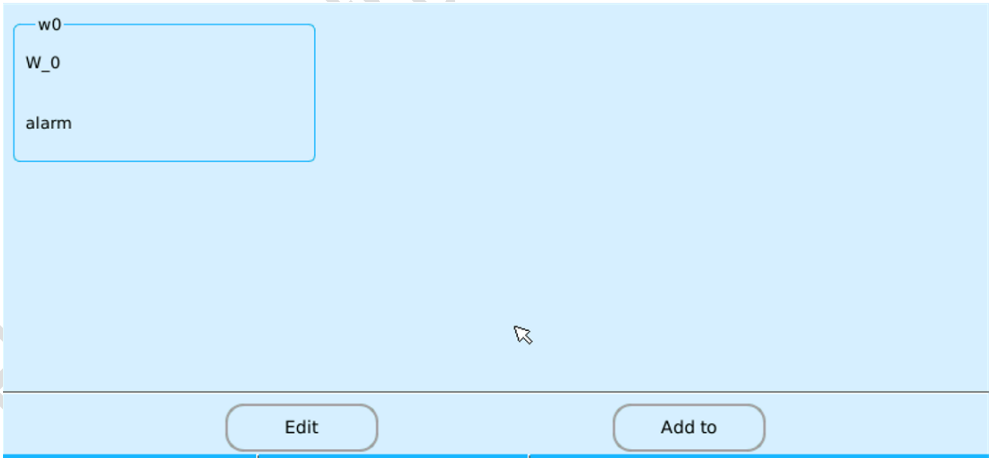


Figure 8.3 Custom alarm setting

8.3 Other configurations

Click [configure] in the manual interface to configure emergency stop, operation environment, abnormal braking and network.

8.3.1 Emergency stop trigger output configuration

When pressing the [emergency stop] button of the teach pendant or the external [emergency stop] button, in order to stop the work of other cooperative systems outside the robot at the same time, it is necessary to configure the corresponding output for the "emergency stop" trigger. As shown in Figure 8.4, the emergency stop output configuration can be customized to output high level or low level.

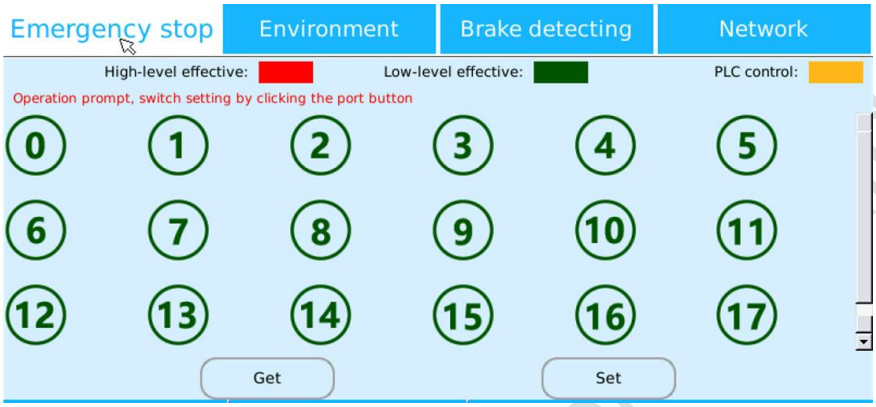


Figure 8.4 Emergency stop output configuration

The system provides three setting modes for the output port, which are distinguished by different colors: "high level effective" red, "low level effective" green and "PLC control" yellow. Different states can be switched by clicking the port button.

Operation steps of "emergency stop configuration" of the system: first select the serial number of "IO output port" according to the actual situation and click the port button to switch the state, and finally click the [set] button to complete "emergency stop configuration". The initial state of the system for "emergency stop configuration" is "active at low level" by default, that is, when the controller receives the "emergency stop" command, the IO output port outputs "low level".

8.3.2 Calibration cycle configuration

When performing programmed work, robots often deviate from their predetermined path over time. Therefore, in order to ensure the accuracy and reliability of the robot on the production line, recalibration is necessary. As shown in Figure 8.5, you can configure how many times the robot runs the program and then calibrate it once.

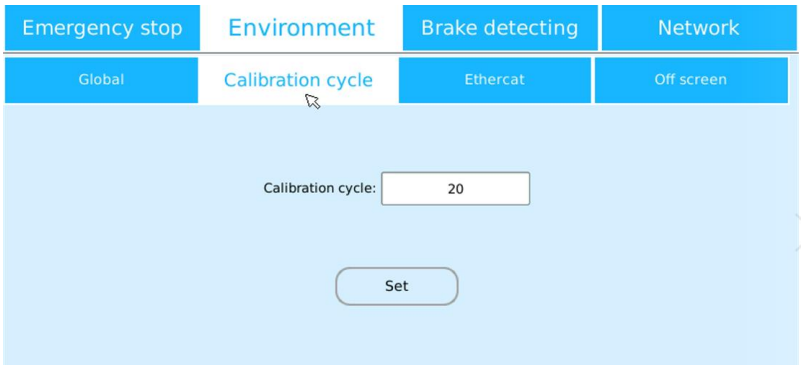


Figure 8.5 Robot calibration cycle configuration

HUNAN Betun Robot Co., Ltd.

8.3.3 Off screen time configuration

The teach pendant can enter the sleep state during the working process of the robot. As shown in Figure 8.6, you can set how long it takes to turn off screen display. Click the teach pendant screen to wake up.

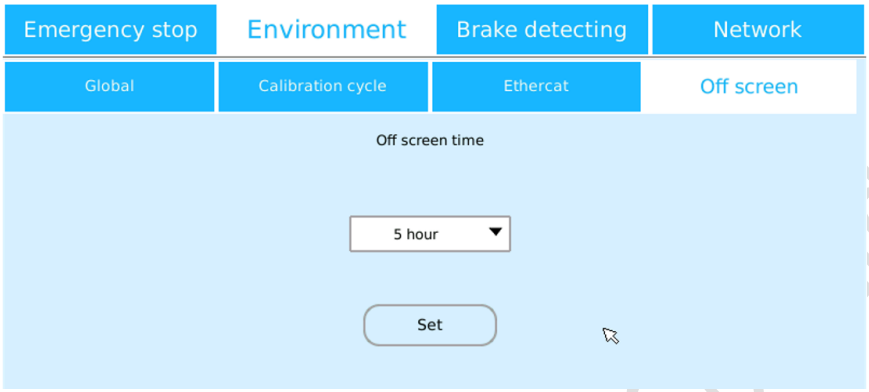


Figure 8.6 Off screen time configuration

8.3.4 Abnormal braking detection

The robot body needs six degrees of freedom, and the power provided comes from six three-phase AC servo motors. When the robot is not working, the brake circuit of the motor is not powered on, and the motor is fixed by friction of the brake pad. After passing through the reducer, the whole mechanical mechanism is deadlocked. When working, the motor is powered on, the brake circuit is powered on, the brake pads are released, and the motor is fixed by the magnetic field. Therefore, it is necessary to detect abnormal braking conditions, as shown in Figure 8.7. The braking detection attribute of each pulse driver port can be set.

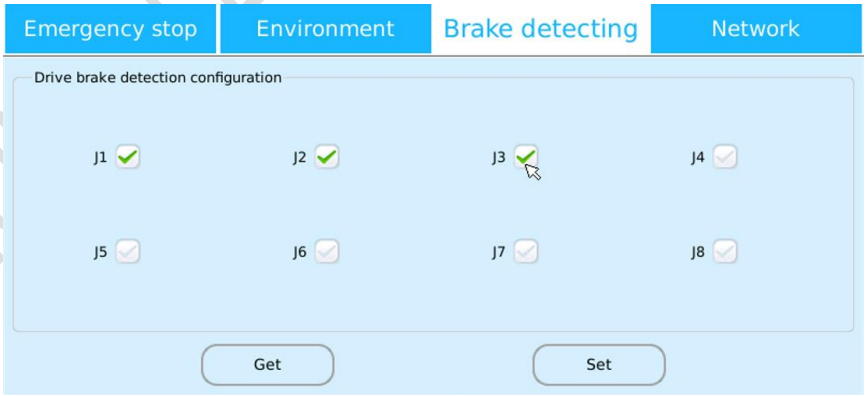


Figure 8.7 Brake abnormality detection attribute setting

8.3.5 Network configuration

As shown in Figure 8.8, before using the network, it is necessary to configure and detect the network IP address, port number, etc.

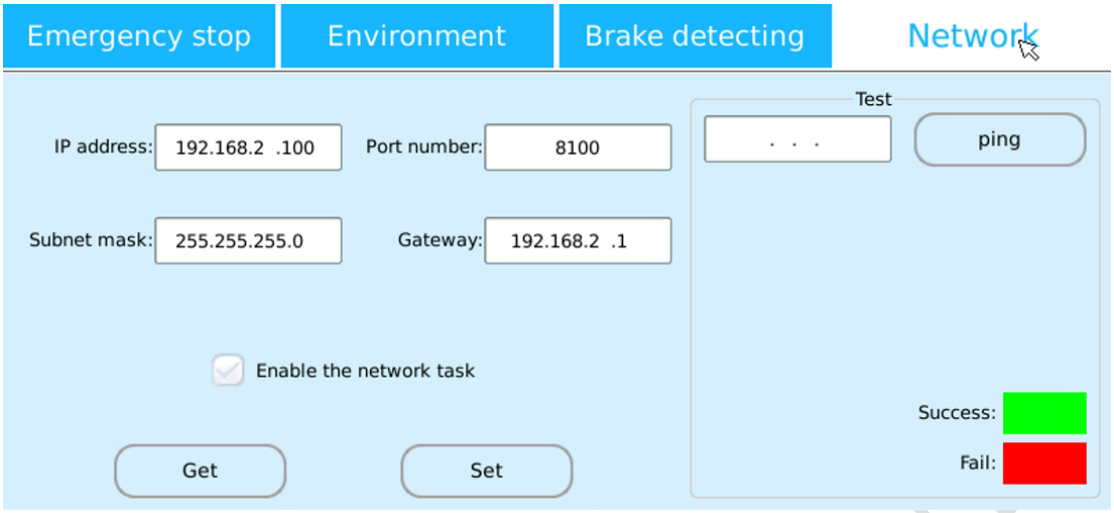


Figure 8.8 Network configuration

8.4 Language configuration

Click [system] - [language] in the manual interface to switch the language of the teaching pendant system. At present, in addition to Chinese, English, Russian, Korean and Vietnamese are also supported.

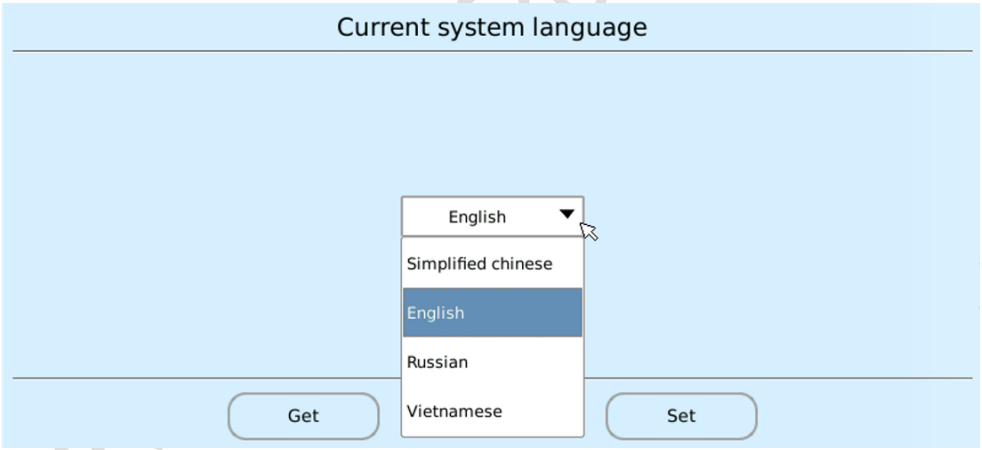


Figure 8.9 Language configuration

9 System backup and update

9.1 System backup

Betrun system software provides direct configuration and USB flash disk configuration backup functions. The backup programs and file types are shown in Table 9.1.

Table 9.1 Backup programs and file types

Number	Storage location	Program type
1	Controller	Exe system program "robcon", FPGA program and user program
2	Teach pendant	Exe system program "robteach", resource document, manual

9.1.1 Backup configuration

Enter the system [update and backup], as shown in Figure 9.1. There are two backup configuration methods: direct configuration and USB flash disk configuration.

- Select "direct configuration" to check the backup of "user program", "PLC program" and "exe system program".

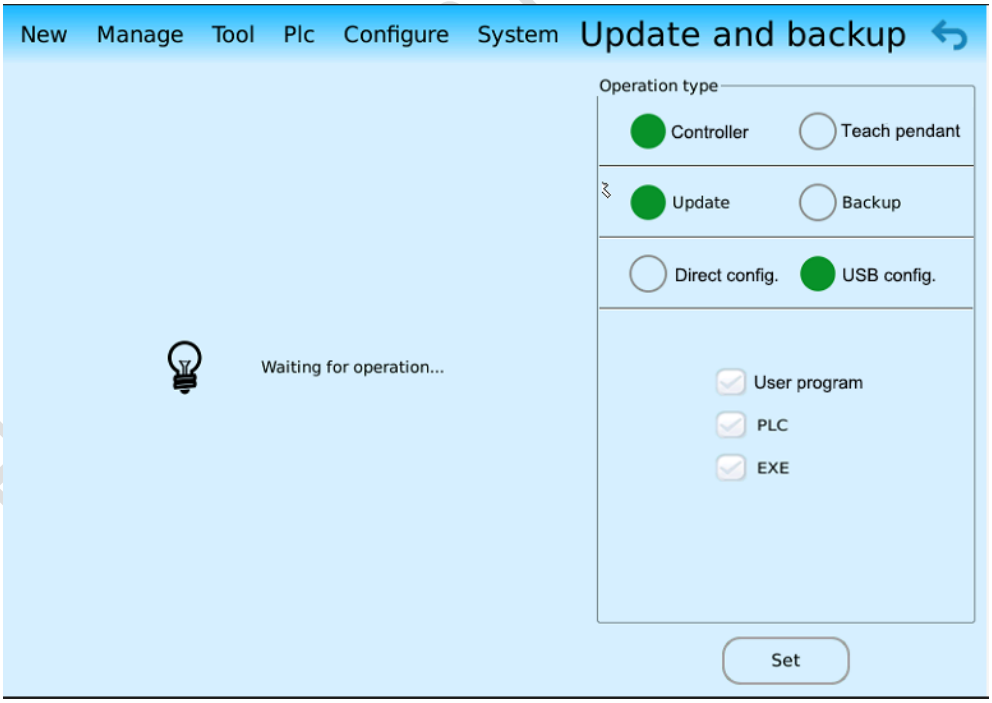


Figure 9.1 update and backup interface

- Select "USB flash disk configuration" and prepare a FAT32 format USB2 0 and create a new folder and configuration file in it. This method can backup more file types.

The steps are as follows:

1) Create a new folder under the root directory of the USB flash disk. The name must be "robcon".

2) Create a new text document in the "robcon" folder and rename it "config.ini".

3) Open "config.ini" with notepad and edit the text content as shown in Figure 9.2. The text contents related to backup are described in table 9.2.

If you want to back up, set the corresponding parameters. If you don't want to back up, set the parameters to "0".

Table 9.2 Description of configuration file content

Text content	Description	Text content	Description
[control_backup]	**Controller backup**	**[teach_backup]**	**Teach pendant backup**
app	1：Exe system program backup	pic	2：Screenshot backup
fpga	1：FPGA program backup	doc	1：Document backup
record	1：User program backup 2：PLC program backup 3：The user program and PLC program are backed up at the same time 64：User program reuse	app	1：Exe system program backup

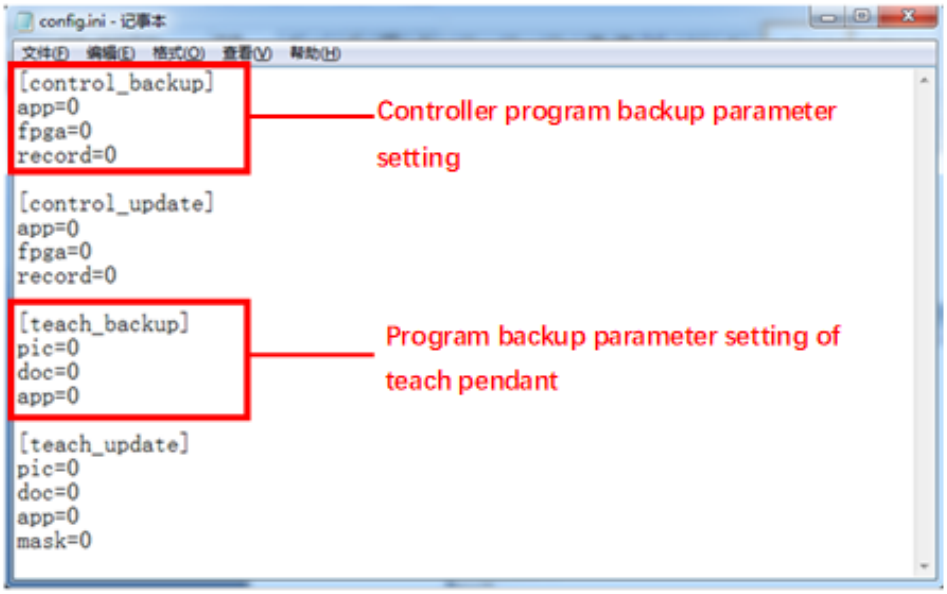


Figure 9.2 Related contents of "config.ini" configuration file backup

9.1.2 System program backup

Before backup, you need to select "direct configuration", check "exe" (controller and teach pendant system files can only be backed up separately), or select "USB flash disk configuration", and then set parameters in the "config.ini" configuration file in the "robcon" folder of USB flash disk (controller and teaching pendant system files can be backed up at the same time).

As shown in Figure 9.3, when the USB flash disk is configured to back up the EXE system program of the main controller, modify the parameter app = 1 in [control_backup] in the "config.ini" configuration file to save the configuration file. When backing up the teaching pendant exe system

program, modify the parameter app = 1 in [teach_backup] in the "config.ini" configuration file and save the configuration file.

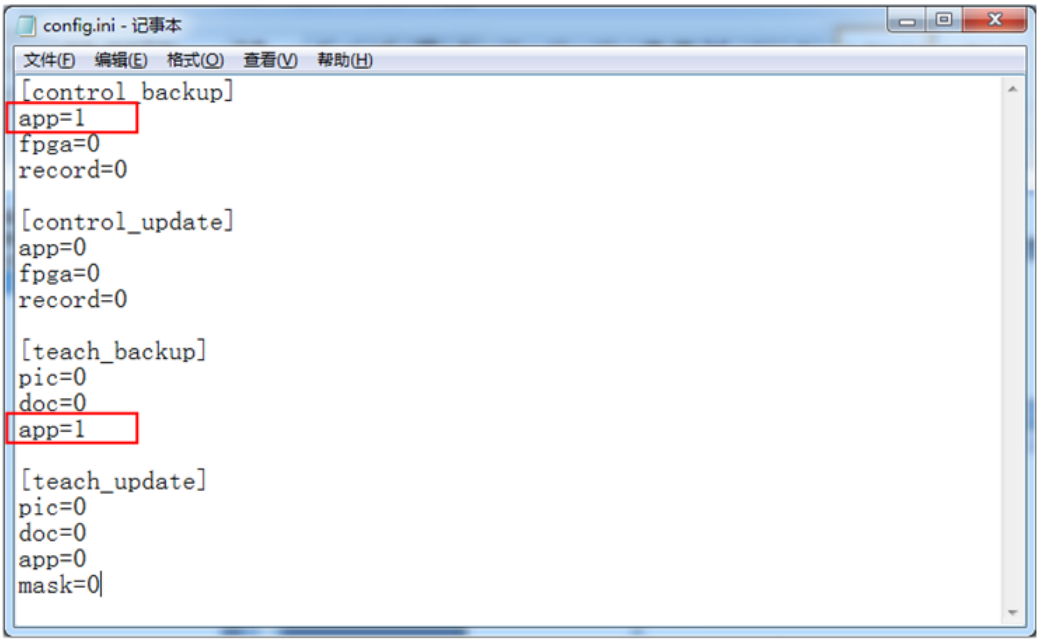


Figure 9.3 Configuration file setting parameter backup system program

After setting the backup configuration, insert the USB flash disk into the controller, click [system] - [update and backup] in the manual interface of the teach pendant, select "backup" and click [OK]. After the backup is successful, the "robcon" folder of USB flash disk will contain three files, as shown in Figure 9.4.

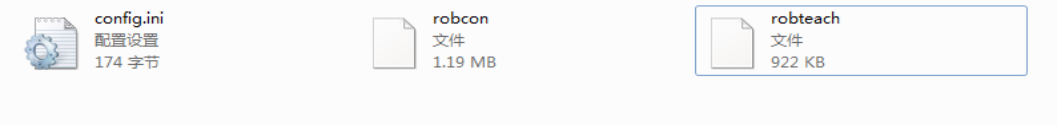


Figure 9.4 Files of USB flash disk backup controller and teach pendant program

9.1.3 User program backup

Before backup, you need to select "controller" - "direct configuration", check "user program", or select "USB flash disk configuration", then set the "[control_backup]" parameter record = 1 in the USB flash disk configuration file "config.ini", and save the configuration file, as shown in Figure 9.5. After setting the configuration file, insert the USB flash disk into the controller, and click [system] - [update and backup] in the manual interface of the teach pendant to backup.

After the user program is backed up successfully, the folder "robcon" will contain "program" folder, "teachRecord.dat" file and "Point.dat" file, in which the "program" folder contains the program files established in the system and is stored in "*.raw" format.

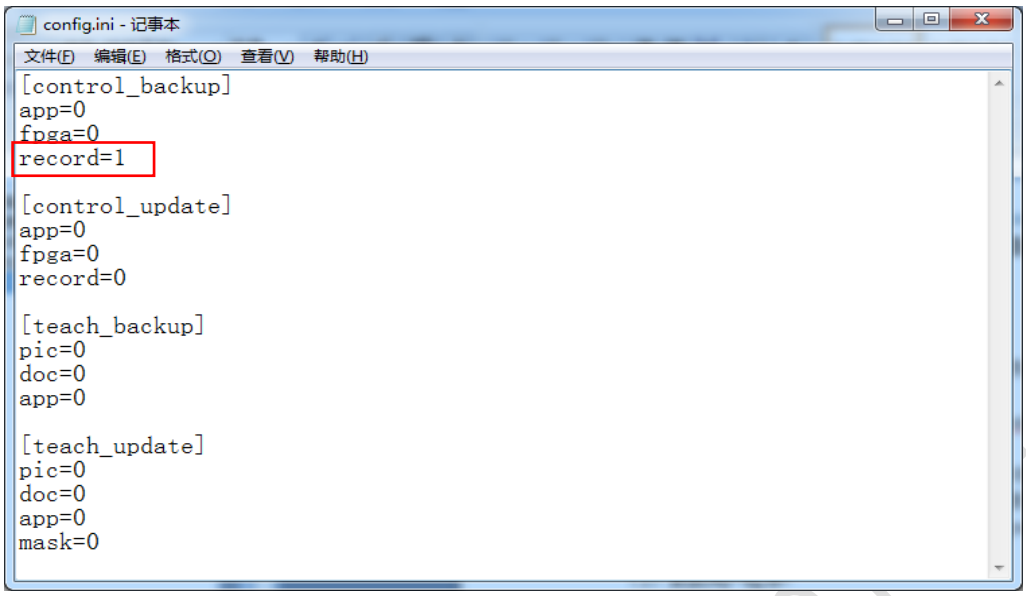


Figure 9.5 Configuration file backup controller user program

9.1.4 PLC program backup

Before backup, you need to select "controller" - "direct configuration", check "PLC", or select "USB flash disk configuration", and then save the configuration file in the parameter "record = 2" of "[control_backup]" in the USB flash disk configuration file "config.ini". After setting the configuration file, insert the USB flash disk into the controller and click [system] - [update and backup] in the manual interface of the teach pendant to backup.

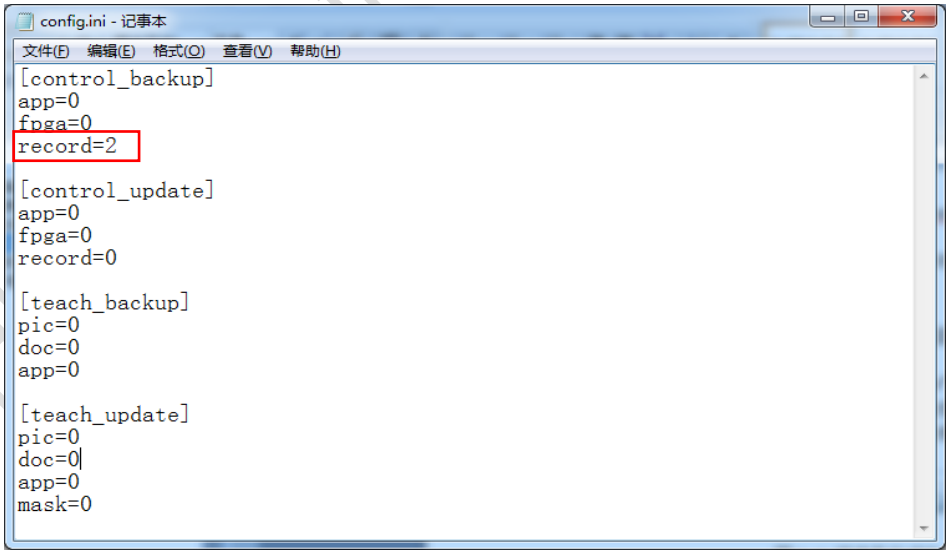


Figure 9.6 Configuration file backup PLC program

After the PLC program is backed up successfully, the "robcon" folder will contain two files: the configuration file "config.ini" and the "PLC" folder. Among them, the PLC program in the system is backed up and saved in the "PLC" folder and stored in the format of "*.PLC", as shown in Figure 9.7.

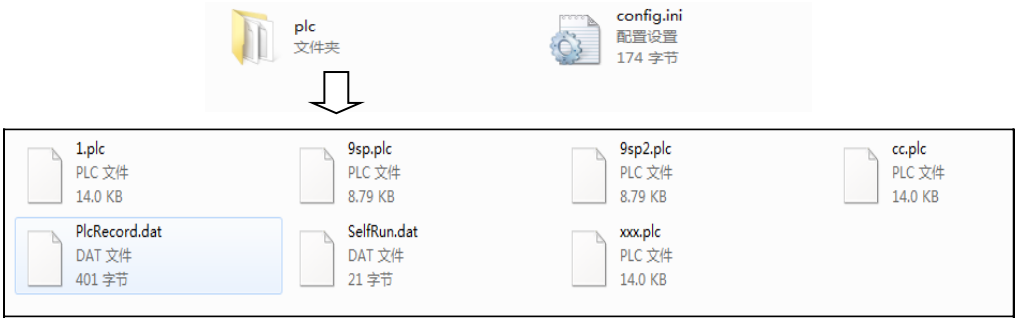


Figure 9.7 PLC program backup files

9.2 System update

Betrun system software provides the functions of direct configuration update and USB flash disk configuration update. The updatable programs and file types are shown in table 9.3.

Table 9.3 Update programs and file types

Number	Storage location	Program type
1	Controller	Exe system program "robcon", FPGA program and user program
2	Teach pendant	Exe system program "robteach", resource document, manual, mask file

9.2.1 View system version

Select [system] - [version] information to view the system version information. As shown in Figure 9.8 below, you can view the specific information of "machine code", "controller system version number", "teach pendant system version number", "authorization status" and "copyright" of the control system.

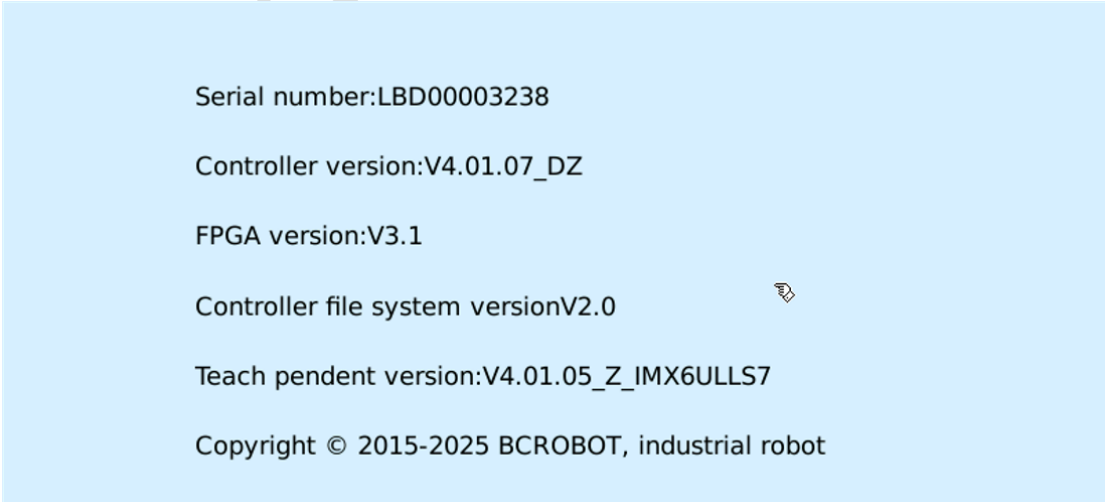


Figure 9.8 View system program version information

Betrun Robot Co., Ltd.

9.2.2 Update configuration

Before system update, prepare a FAT32 format USB2 0, copy the "robcon" folder with updated files into the root directory. Then you need to "configure directly" in [update and backup], or configure in "config.ini" on USB flash disk.

1) In the root directory of the USB flash disk, the name of the update folder must be "robcon". The "robcon" folder contains the update application files robcon and robteach, "config.ini" configuration file and "check.dat" verification file.

2) Open "config. Ini" with Notepad, as shown in Figure 9.9, and the text content description related to update is shown in Table 9.4.

For updates, the corresponding parameters will be set in the configuration file. If not updated, the parameters will be set to "0".

Table 9.4 Description of configuration file content

Text content	Description	Text content	Description
[control_update]	Controller program update	[teach_ update]	Teach pendant program update
app	1：Exe system program update	pic	1：Resource document update
fpga	1：FPGA program update	doc	1：Manual update
record	1：Local user program update 2：Local PLC program update 3：The user program and PLC program are updated at the same time 64：Reuse and update of other machine user programs	app	1：Exe system program update
		mask	1：Mask file update

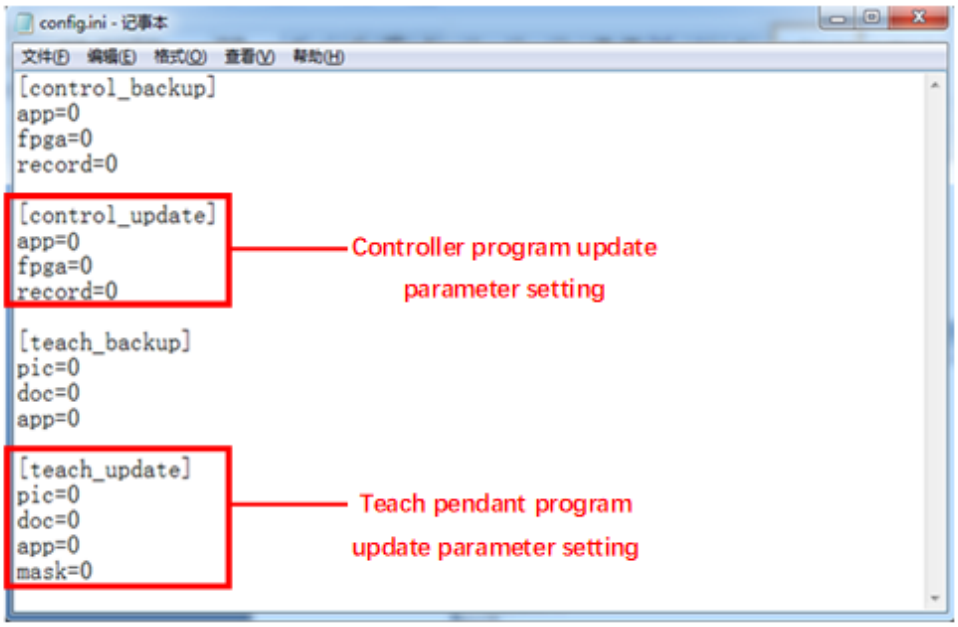


Figure 9.9 Update related contents of "config.ini" configuration file

9.2.3 System program upgrade

The system upgrade includes the upgrade and update of "controller program" and "teach pendant program":

1) If you use USB flash disk configuration, you need to set the parameters in the configuration file "config.ini", mainly in the two "types of" [control_update] "and" [teach_update] ": app = 1, as shown in Figure 9.10.

2) Log in as administrator and click [System] – [update and backup].

3) **First update the teach pendant. After the update is completed, the system will remind "whether to restart?". Please click "NO" and then update the controller. After the update is completed, click restart to confirm.**

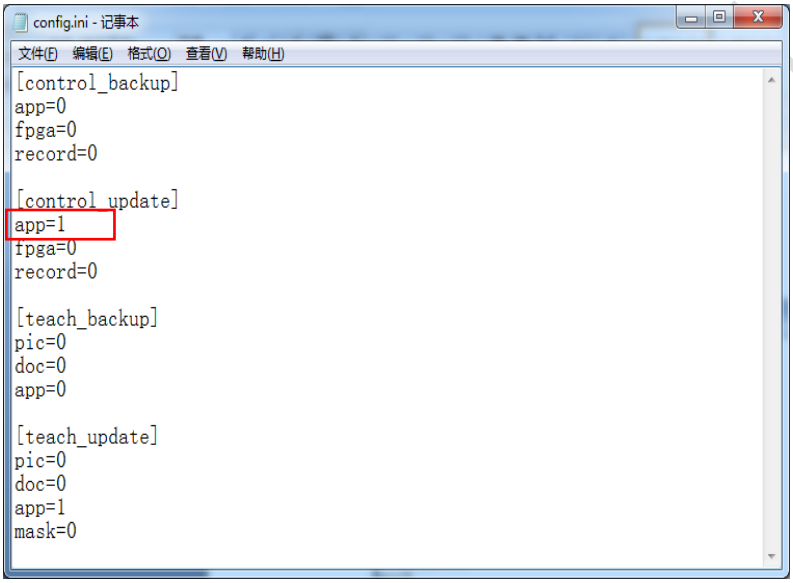


Figure 9.10 Configuration file setting parameter update exe program

Part II Function Package Instructions

1 Position searching and tracking in welding

In the robot welding operation, some products have high welding accuracy requirements and tooling assembly requirements. The welding robot alone cannot solve the actual needs of the factory. At this time, it is necessary to carry out position searching and tracking. The position searching and tracking function can solve the deviation caused by welding thermal deformation, workpiece blanking deviation and clamping.

Weld tracking is divided into electric tracking and laser tracking. Electric tracking is generally used for thick plates, mostly swing welding, such as excavator bucket arm, crane welding, etc. The approximate position of the weld is calculated by the welding wire contacting the edge of the base metal. Laser tracking is to feed back the weld information to the robot through the laser tracking sensor. In the welding process, the robot will automatically modify the trajectory according to the obtained data to complete the welding.

1.1 Preparation

1.1.1 Preparation for electric position searching

For electric position searching, the following preparations need to be made:

(1) The workpiece surface must be free of rust, oxide layer, paint, or other insulating coating.

(2) Gun cleaning and wire cutting must be carried out before positioning.

(3) When using water-cooled welding gun, it is recommended to use distilled water or other non-conductive coolant. Impure water (such as salty mineral water) will reduce the sensitivity or voltage of position searching.

(4) Establish the tool coordinate system.

1.1.2 Preparation for laser position searching

For laser position searching, the system needs to connect the laser vision sensor on one side of the welding gun through the LAN network port. Then, click [new] - [function] - [laser tracking] to create a "laser tracking" function package, which is used to realize the laser position searching of welds in the program.

The creation steps of "laser tracking" function package are as follows:

Step 1: set the basic parameters, as shown in Figure 1.1, and the parameter description is shown in Table 1.1.

Step 2: calibrate the sensor. As shown in Figure 1.2, the calibration needs to select a reference point p1 on the weld, move the robot tool TCP to the reference point, obtain the robot six joint data at this point, and then click [start calibration]. For this reference point, change the detection height

HUNAN Betrun Robot Co., Ltd.

(Z-axis) and detection offset (X-axis), detect from 6 different position points (p2-p7), and read the robot six joint data and sensor value of p2-p7 respectively, to complete the calibration process, as shown in Figure 1.3.

Step 3: click the [save] button to save the laser tracking function package, which is used for laser direct position searching and indirect position searching.

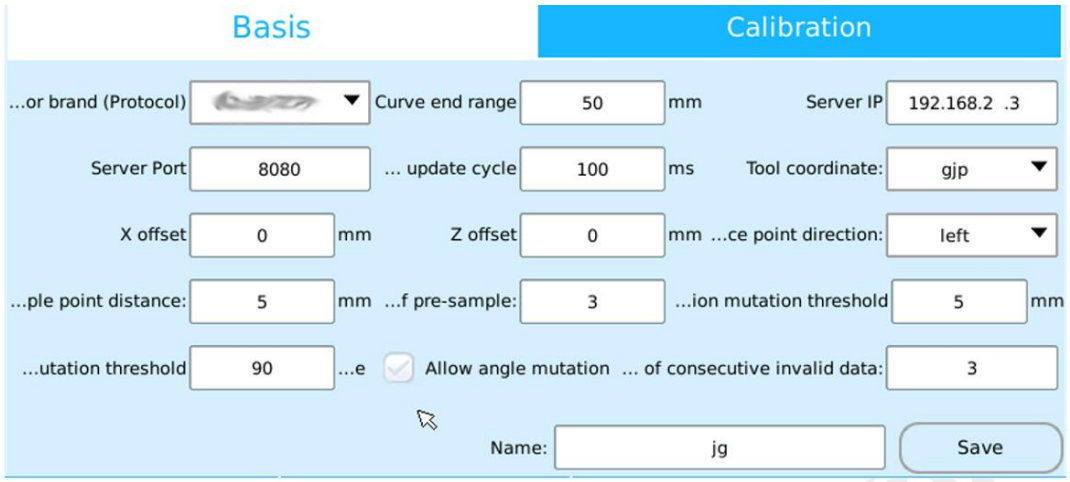


Figure 1.1 Basic parameters setting of laser tracking function package

Table 1.1 Description of basic parameters of laser tracking function package

Number	Parameter	Description
1	Curve end range	Set the position relationship between the end point of laser tracking during welding and the end point of teaching curve, that is, stop tracking when entering the end range of curve end point set during teaching to avoid laser tracking error near the end of welding. (Unit: mm.)
2	Server IP, Server Port	LAN network port communication settings.
3	Tool coordinate	Select the tool coordinate system created by the welding gun.
4	X offset, Z offset	Set the offset between the laser search point and the actual welding point. Unit: mm.
5	Reference point direction	Direction of the reference point
6	sample point distance, the number of pre-samples	The distance between sampling points and the number of pre-samples in the tracking process.
7	Position mutation threshold	The length threshold of no weld feature in the process of weld tracking. If the threshold is exceeded, the laser tracking data is in error.
8	Angle mutation threshold	Set according to the weld to be tracked. If the threshold is exceeded, it is considered that the laser tracking data is wrong.
9	Maximum number of consecutive invalid data samples	Monitor whether the laser seam tracking sensor system works normally.

HUNAN Betrun Robot

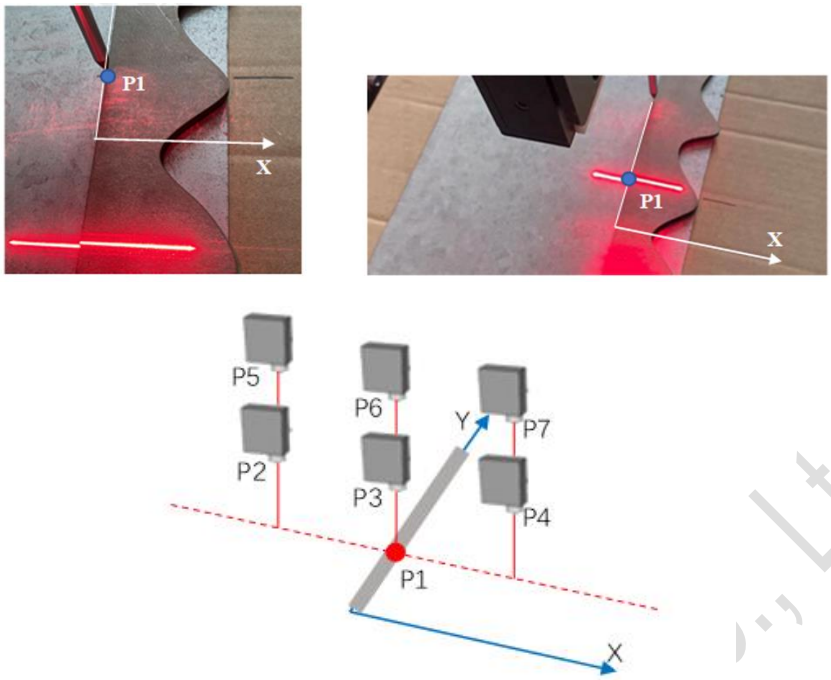


Figure 1.2 Schematic diagram of calibration process

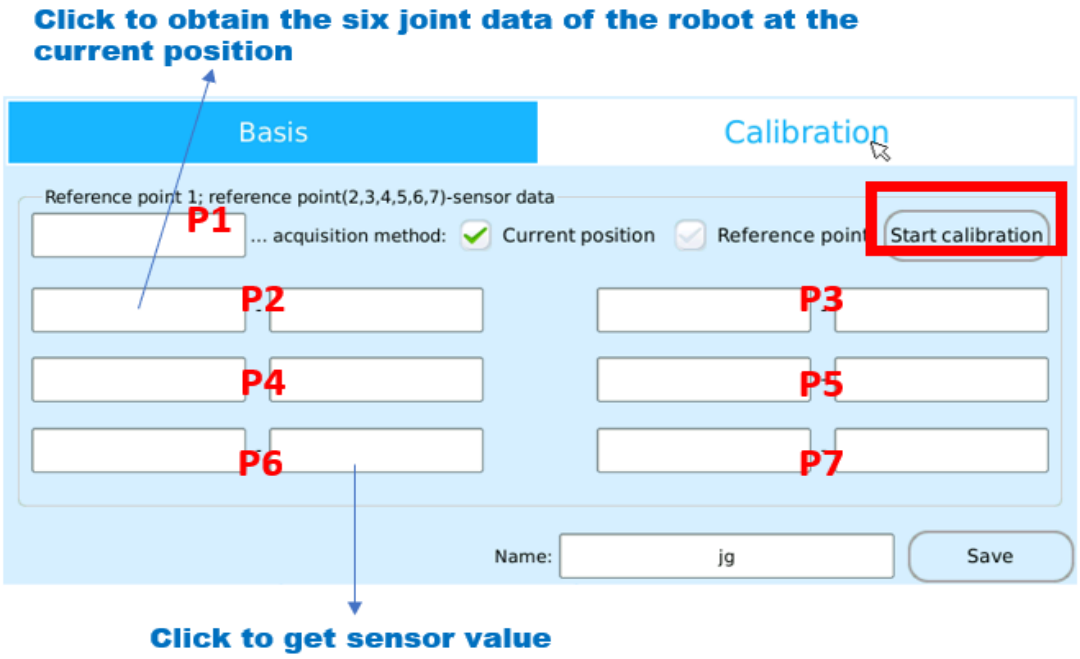


Figure 1.3 Calibration acquisition point information interface

1.1.3 Direct and indirect position searching

- Direct position searching: directly obtain the "run to point" on the weld.
- Indirect position searching: the offset value is obtained through position searching to correct the "run to point" of manual teaching.

1.2 Laser direct position searching

Click [new] - [basic program] on the touch screen in the main manual interface, and select "task type" - "function statement" - "direct searching position".

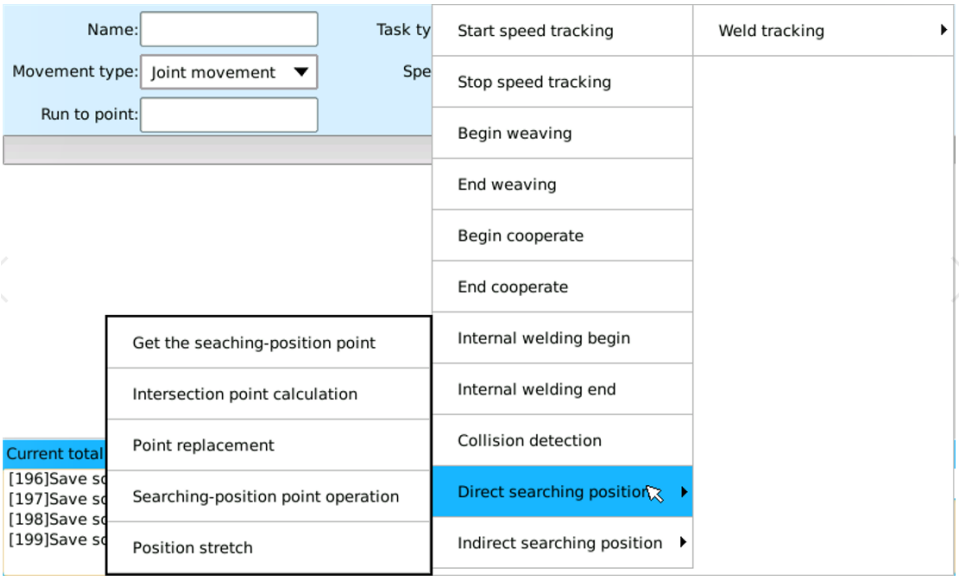


Figure 1.4 "Direct searching position" function instructions

The "direct searching position" function of the system includes five instructions: "Get the searching position point", "Intersection point calculation", "Point replacement", "Searching position point operation" and "Position stretch", as shown in Figure 1.4. After creating the "tool coordinate system" and "laser tracking" function package, you can use the "direct searching position" function instructions.

1.2.1 Get the searching position point

Select the task type of "Get the searching position point", as shown in Figure 1.5, which includes four parts: "search site ID", "function package", "weld type" and "weld serial number".

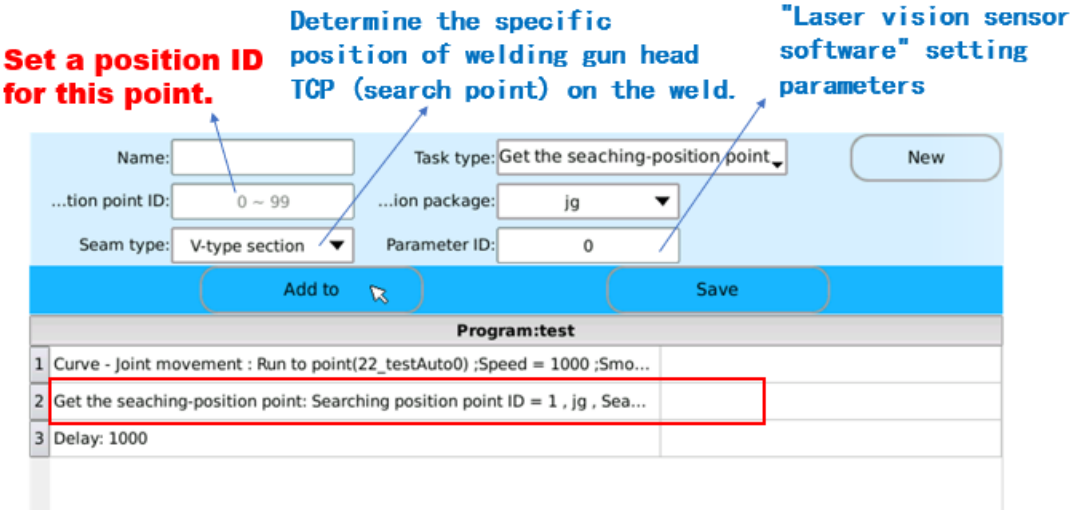


Fig. 1.5 Function instruction of "Get the searching position point"

See Table 1.2 for specific parameter settings.

Table 1.2 Description of function parameters of "Get the searching position point"

Number	Parameter	Description
1	Searching position point ID	Configure the "ID" number of the searching position point, with a range of "0 - 99". This ID is "unique" in a single program. (Note: this ID is different from the point ID definition in point management.)
2	Function package	Select the name of the "laser tracking" function package to be used.
3	Seam type	Set the "weld" type, including "V-shaped section", "left lap", "right lap", "fillet weld", "butt joint", "left edge", "right edge", "external corner", "left circumferential seam" and "right circumferential seam".
4	Parameter ID	Setting "weld serial number" which is related to the setting parameters of "laser vision sensor display and control software".

1.2.2 Point replacement

In the robot welding procedure, the "run to point" in the motion curve instruction is obtained by manual teaching, which is deviated from the actual weld position. At this time, the actual weld position point data can be obtained through the "point replacement" instruction to replace the "run to point" in the curve.

As shown in Figure 1.6, after selecting the task type of "point replacement", two parameters of "curve type" and "end point" need to be set.

Curve types include "Linear" and "arc". The end point is the "ID" number of the "replacement point", and the "run to point" of the curve followed by the replacement instruction. For arc type welds, three points need to be replaced. Firstly, you should select the linear type to replace the starting point, and then select the arc type to replace the middle point and the end point.

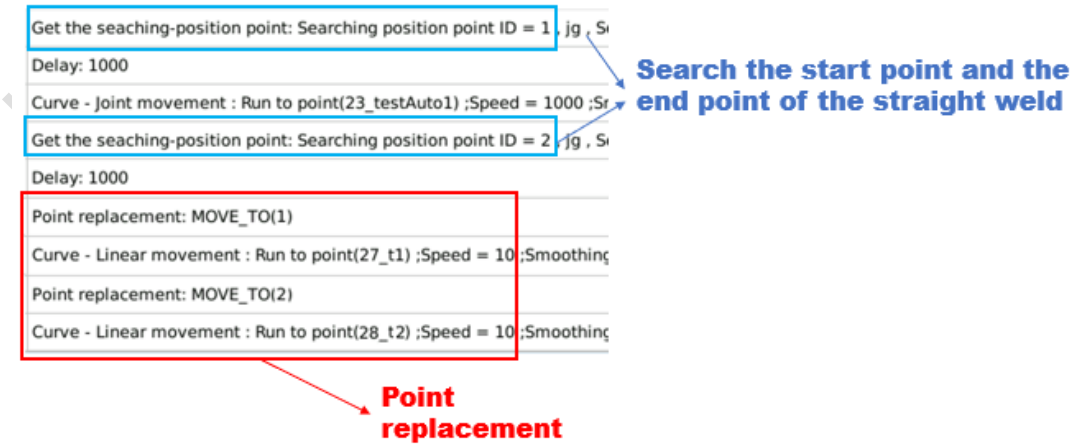


Figure 1.6 Point replacement

HAN Betrun Robot Co., Ltd.

1.2.3 Intersection point calculation

The system can obtain the data of four "reference points" by using the method of "get the searching position point" on the two curves, and then calculate the coordinates of the intersection points of the two curves based on the four "reference points".

As shown in Figure 1.7, the straight line L1 and the straight line L2 intersect at point P, where A1 and B1 are the "reference points" obtained by "get the searching position point" on the straight line L1 respectively, and A2 and B2 are the "reference points" obtained by "get the searching position point" on the straight line L2 respectively.

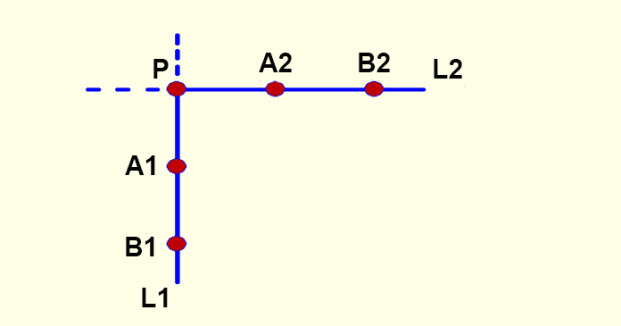


Figure 1.7 Schematic diagram of intersection point calculation

Select the "intersection point calculation" task type, as shown in Figure 1.8. You need to set the parameters of "distance threshold", "intersection ID", "intersection curve 1" and "intersection curve 2". The specific description of parameters is shown in Table 1.3.

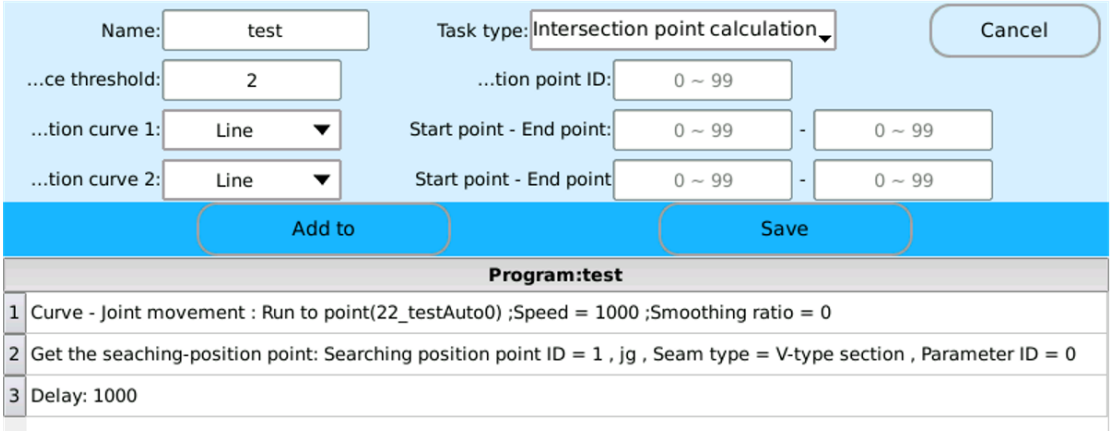


Figure 1.8 "Intersection point calculation" instruction setting

Table 1.3 Description of functional parameters of "intersection point calculation"

Number	Parameter	Description
1	Distance threshold	Threshold value of intersection distance between two curves, unit: mm.
2	Intersection point ID	Set the "ID" number of the "intersection point", represented by "0 - 99". (Note: this ID is different from the point ID definition in point management.)
3	Intersection curve 1, Intersection curve 2	Set the type of "intersecting curve", including "linear" and "arc".

4	Start point – End point	Set the "reference point" of the curve, that is, the "ID" number obtained by "get searching position point" in the program.

The program case using the "intersection point calculation" instruction is shown in Figure 1.9.

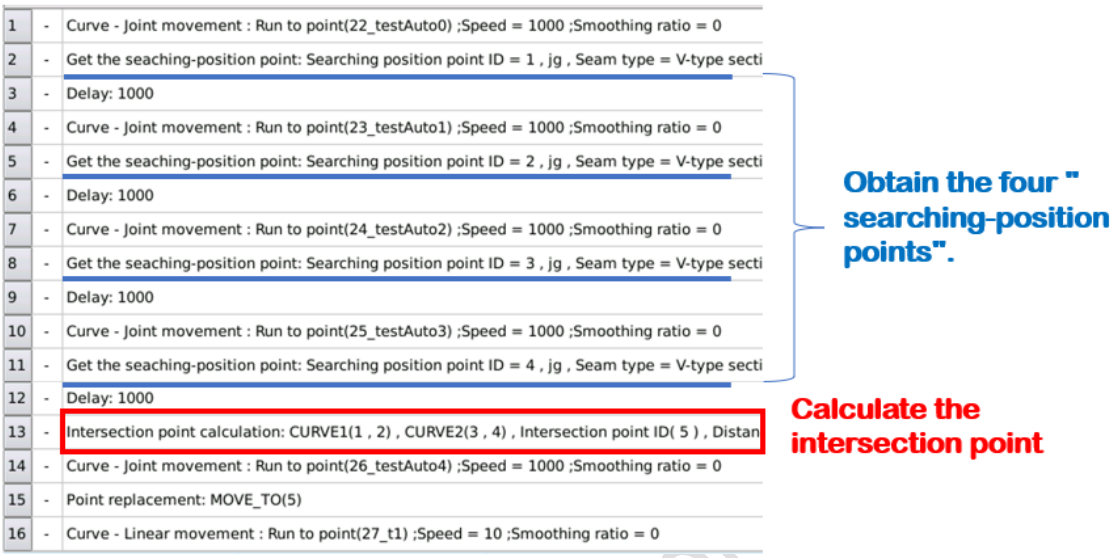


Figure 1.9 Program case

1.2.4 Searching-position point operation

This instruction can get the coordinate value of "searching-position point" and assign it to "global variable", or set the value of global variable to the coordinate value of " searching-position point".

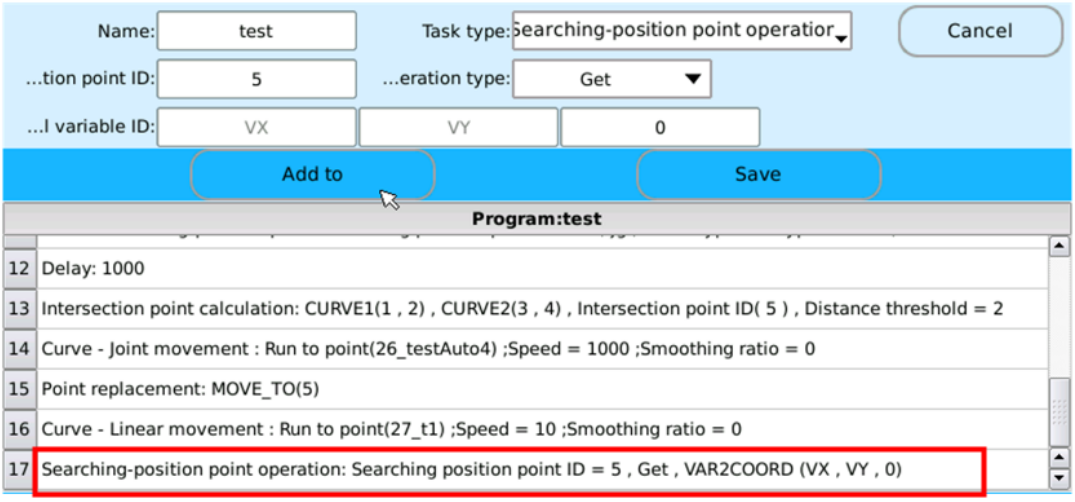


Figure 1.10 Program case

1.2.5 Position stretch

As shown in Figure 1.11, if the starting point and end point of the weld are not suitable for direct

laser position searching, it can be programmed quickly through "point stretch".

As shown in Figure 1.12, for linear welds, it is necessary to set the offset of start point and end point and the IDs. The program is shown in Figure 1.13. The direction from the start point to the end point is positive. Pay attention to the offset direction when setting the offset. For arc welds, three searching position points and offsets of start point, middle point and end point need to be set.

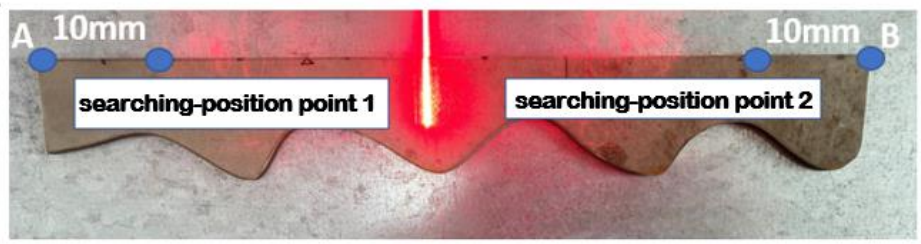


Figure 1.11 Start point A and end point B of weld are not suitable for direct laser positioning

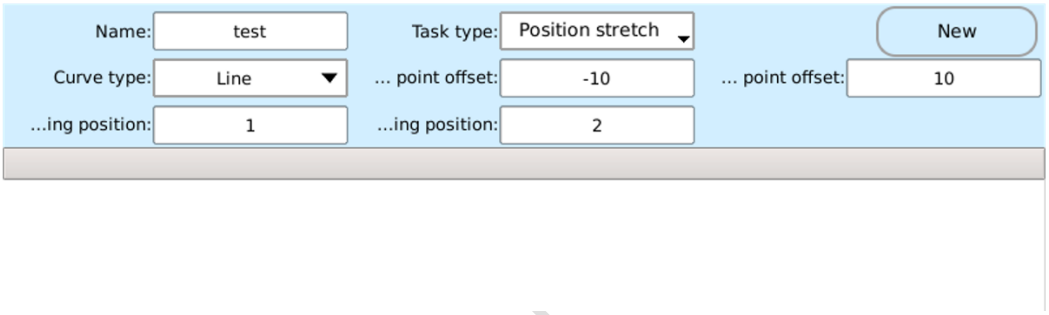


Figure 1.12 "Point stretch" instruction setting

1	-	Curve - Joint movement : Run to point(28_testAuto0) ;Speed = 1000 ;Smoothing ratio = 0
2	-	Delay: 100
3	-	Get the seaching-position point: Searching position point ID = 1 , jg , Seam type = V-type sect
4	-	Delay: 100
5	-	Curve - Joint movement : Run to point(29_testAuto1) ;Speed = 1000 ;Smoothing ratio = 0
6	-	Delay: 100
7	-	Get the seaching-position point: Searching position point ID = 2 , jg , Seam type = V-type sect
8	-	Delay: 100
9	-	Position stretch: LINE(1 , 2) , Start point offset = -10 , End point offset = 10

Figure 1.13 Application case of "point stretch" instruction

1.3 Indirect electric position searching

As shown in Figure 1.14, indirect searching position instructions include: searching begin, searching next, searching end, searching-offset begin and searching-offset end. These five instructions constitute the process of indirect electric position searching.

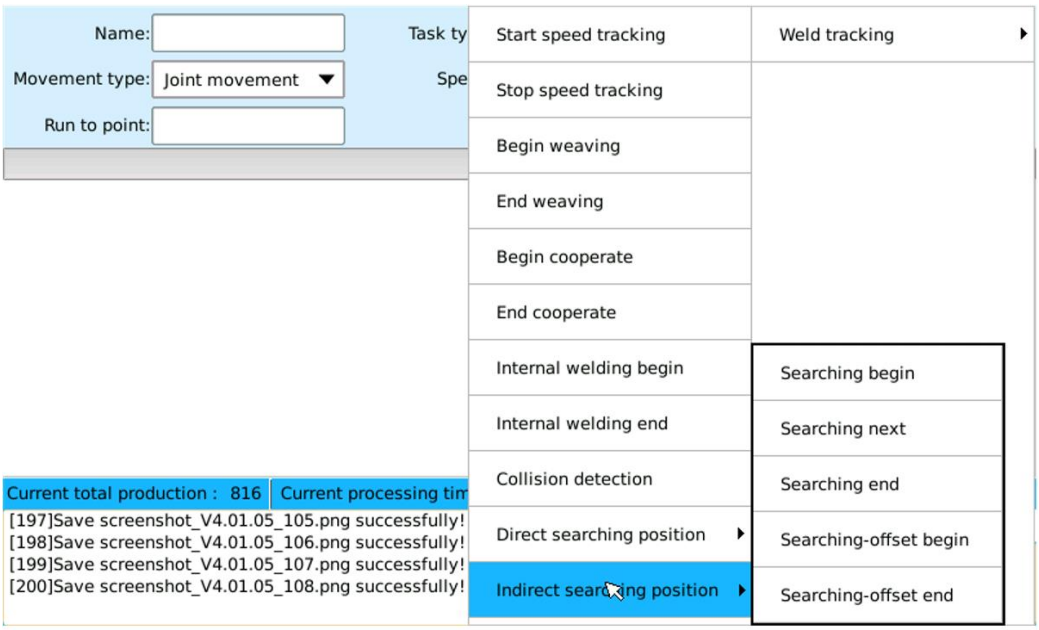


Figure 1.14 Indirect electric position searching instructions

Take one-dimensional (1D) indirect electric position searching as an example, as shown in Figure 1.15, the principle of indirect electric position searching is below:

(1) When manual teaching, set the reference point for the current position of the workpiece and write the positioning track program.

(2) During the actual welding operation, the robot touches the workpiece from a → b → touch the workpiece, and the robot stops moving. Then, the system will use the deviation between the current position and the program set reference point to correct the welding path of the current workpiece.

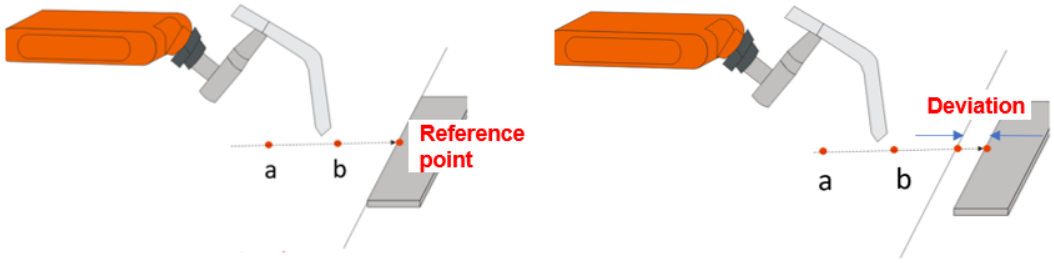


Figure 1.15 Principle of one-dimensional (1D) indirect electric position searching

1.3.1 Parameter setting

Click [indirect searching position] – searching begin], and select the searching position mode as "searching position by electric signal", as shown in Figure 1.16, which is the parameter setting interface. The specific parameters are shown in Table 1.4.

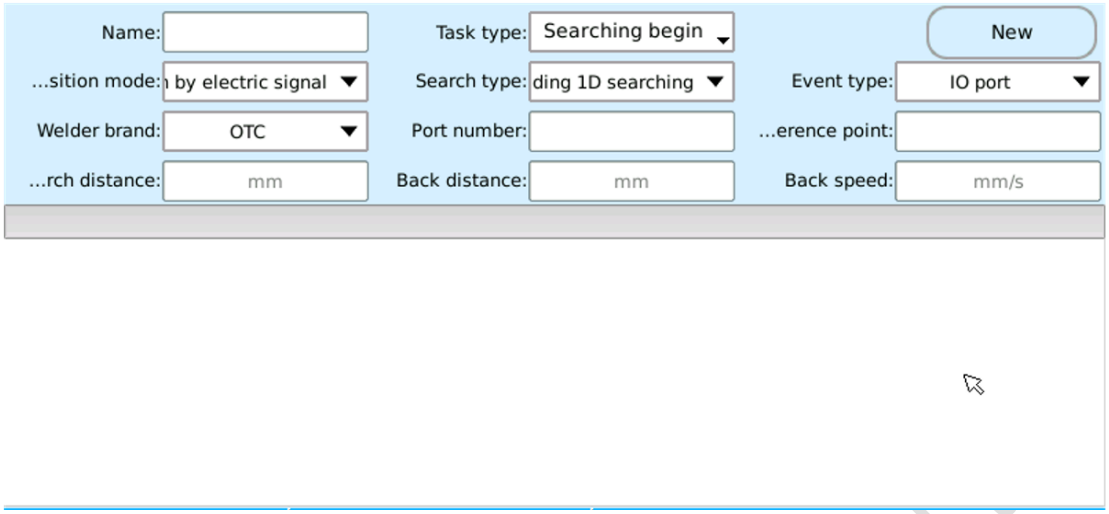


Fig. 1.16 Parameter setting of electric position searching begin

Table 1.4 Description of electric position searching begin instruction parameters

Number	Parameter	Description
1	Search type	Including: angle welding 1D searching, angle welding 1D searching, angle welding 1D searching, V-Section searching, ARC-inside searching, ARC-outside searching, angle welding 2D+R searching, angle welding 3D+R searching, etc. 1D: position finding on one side of XYZ. 2D: locate any two of the three XYZ faces. 3D: position finding on three XYZ faces. 2D + rotation: refers to the rotation position finding of adding one side to any two sides of XYZ. 3D + rotation: refers to the rotation position finding of adding one side to the three XYZ faces.
2	Event type	Trigger includes: communication port and I / O port Communication port: CAN, LAN, RS485, RS232
3	Welder brand	The system provides: OTC, LBBBD-ME, LBBBD-MA, JIN_RUI, DMZN and general analog welding machine.
4	Port number	The port number of the trigger.
5	Reference point	Enter the ID of c, f and i reference points according to the search type.
6	Search distance	Along the direction a to b, the search distance from point a. Unit: mm.
7	Back distance	The return distance to c', f', i' after the contact signal is generated by the contact between the welding wire and the workpiece. Unit: mm.
8	Back speed	The moving speed of the welding wire when it returns after the contact signal is generated by the contact between the welding wire and the workpiece. Unit: mm/s.

HUNAN Betrun Robot Co., Ltd.

1.3.2 Electric position searching process

As shown in Figure 1.17, the start point and end point of the weld curve command obtained through manual teaching in the program are point a and point B respectively. In the process of robot operation, to accurately determine the actual position of weld a and B points, it is necessary to search the position.

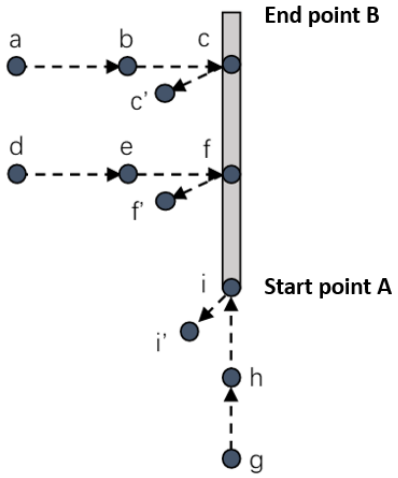


Figure 1.17 Schematic diagram of indirect position searching process

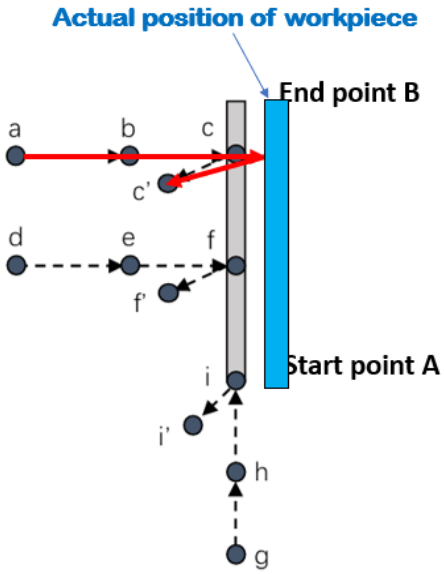
The positioning curve is: a → b, d → e, g → h, that is, the moving direction of robot positioning is determined by these six points. The locating reference points are c, f and i respectively. Once the contact signal is generated by the contact between the welding wire and the workpiece, it will return to c', f' and i' (these three return points in the figure are actually on the positioning curve). Different types of position searching processes are shown in Table 1.5.

Table 1.5 Description of indirect electric position searching process

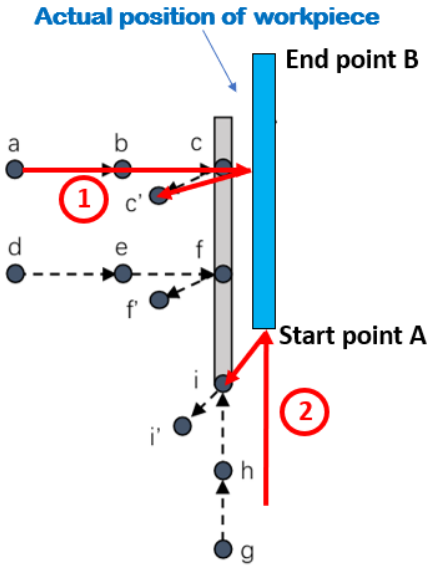
Number	Type	Description
1	1D	Searching track: a→b→c→c' →P→A→B Description: Reach point a and start searching → set searching direction of point b → search to point c and return to point c'→ calculate deviation value (searching end) → run to buffer point P → start searching offset → start welding from a to b →searching offset end
2	2D	Searching track: a→b→c→c' →P1→g→h→i→i' →P2→A→B Description: Reach point a and start searching → set searching direction of point b → search to point c and return to point c '→ run to buffer point P1 → start next searching at point g → set searching direction of point h → search to point i and return to point i' → calculate deviation value (searching end) → run to buffer point P2 → start searching offset → start welding from a to b →searching offset end
3	2D+ Rotation	Searching track: d→e→f→f' →P1→a→b→c→c' →P2→g→h→i→i'

HUNAN Betrun Robot Co., Ltd.

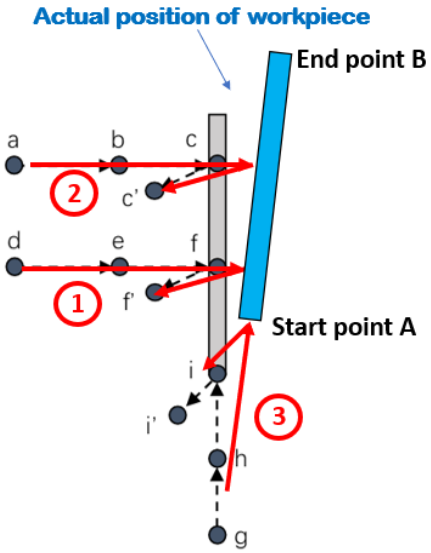
		→P3→A→B Description: (omitted) Note: the first and second search points determine the rotation angle, and its direction is the movement direction of the third search, that is, it is necessary to find position f first and then position c, and the direction of f→ c is consistent with that of g→h.



1D position searching process



2D position searching process



2D+Rotation position searching process

Figure 1.18 Indirect electric position searching process

1.3.3 Program case

The case of 1-dimensional (1D) indirect electric position searching program is shown in Figure 1.19.

Program:test111		
1	-	Curve - Joint movement : Run to point(1347_a) ;Speed = 1000 ;Smoothing ratio = 0
2	-	Searching begin: Searching position by electric signal , OTC , X_5 , Angle welding 1D searching , Referer
3	-	Curve - Linear movement : Run to point(1348_b) ;Speed = 10 ;Smoothing ratio = 0
4	-	Searching end
5	-	Curve - Joint movement : Run to point(1349_GDD) ;Speed = 100 ;Smoothing ratio = 0
6	-	Searching-offset begin
7	-	Curve - Linear movement : Run to point(1350_A) ;Speed = 100 ;Smoothing ratio = 0
8	-	Begin welding: ytyy6
9	-	Curve - Linear movement : Run to point(1351_B) ;Speed = 5 ;Smoothing ratio = 0
10	-	End welding
11	-	Searching-offset end

Figure 1.19 1D position searching program

The case of 2D+rotation indirect electric position searching program is shown in Figure 1.20.

Program:test2		
1	-	Curve - Joint movement : Run to point(1352_d) ;Speed = 1000 ;Smoothing ratio = 0
2	-	Searching begin: Searching position by electric signal , OTC , X_5 , Angle welding 2D+R searching , Refer
3	-	Curve - Linear movement : Run to point(1353_e) ;Speed = 10 ;Smoothing ratio = 0
4	-	Curve - Joint movement : Run to point(1354_GDD) ;Speed = 10 ;Smoothing ratio = 0
5	-	Curve - Joint movement : Run to point(1355_a1) ;Speed = 10 ;Smoothing ratio = 0
6	-	Searching next
7	-	Curve - Linear movement : Run to point(1356_b1) ;Speed = 10 ;Smoothing ratio = 0
8	-	Curve - Joint movement : Run to point(1354_GDD) ;Speed = 1000 ;Smoothing ratio = 0
9	-	Curve - Joint movement : Run to point(1357_gdd3) ;Speed = 1000 ;Smoothing ratio = 0
10	-	Curve - Joint movement : Run to point(1358_g) ;Speed = 1000 ;Smoothing ratio = 0
11	-	Searching next
12	-	Curve - Linear movement : Run to point(1359_h) ;Speed = 10 ;Smoothing ratio = 0
13	-	Searching end
14	-	Curve - Linear movement : Run to point(1357_gdd3) ;Speed = 1000 ;Smoothing ratio = 0
15	-	Searching-offset begin
16	-	Curve - Linear movement : Run to point(1360_A) ;Speed = 100 ;Smoothing ratio = 0
17	-	Curve - Linear movement : Run to point(1361_B) ;Speed = 10 ;Smoothing ratio = 0
18	-	Searching-offset end

Figure 1.20 2D+rotation position searching program

1.4 Indirect laser position searching

Laser indirect position searching principle: in the process of position searching along the weld track, the laser sensor scans, uses the deviation value between the scanned actual position and the manual teaching position, and corrects the path through the offset instruction, so as to obtain the real track value and complete the welding.

1.4.1 Parameter setting

Click [indirect searching position] – searching begin], and select the searching position mode as "searching position by laser signal". The specific parameters are shown in Table 1.6.

Table 1.6 Parameter description of start indirect laser position searching

Number	Parameter	Description
1	Function package	Select the "laser tracking" function package.
2	Seam type	Set the "weld" type, including "V-shaped section", "left lap", "right lap", "fillet weld", "butt joint", "left edge", "right edge", "external corner", "left circumferential seam" and "right circumferential seam".
3	Parameter ID	Setting "weld serial number" which is related to the setting parameters of "laser vision sensor display and control software".
4	Curve type	Line, arc, circle. 2 points for straight line positioning and 3 points for arc and circle positioning.
5	Search type	Including "static searching" and "dynamic searching".

1.4.2 Laser position searching process

In the process of laser position searching, there is no need for the welding wire to contact the workpiece, that is, given the start point A and end point B of the weld, after starting the position searching, the laser tracking sensor will scan the workpiece and calculate the deviation value. The specific positioning process is shown in table 1.7.

Table 1.7 Description of indirect laser position searching process

Number	Instruction	Description
1	Joint movement to A	Curve instruction
2	Searching begin	Find the start point A of the previous curve
3	Linear movement to B	Curve instruction
4	Searching next	Find the line end point B
5	Searching end	Searching instruction
6	Joint movement to C	Move to transition point
7	Searching-offset begin	Searching offset begin instruction
8	Joint movement to D	Run to the welding start point D during manual teaching

HUNAN Betrun Robot Co., Ltd

9	Begin welding	Begin welding instruction
10	Linear movement to E	Weld to the weld end point E based on the offset
11	End welding	End welding instruction
12	Searching-offset end	Searching offset end instruction

Note:

(1) Run the curve to point A, and then use the "searching begin" instruction to successfully locate the point A on the weld. Run the curve to point B, and then use "searching next" to locate the point B on the weld successfully, so as to obtain the offset value.

(2) When using the "searching-offset begin" and "searching-offset end" function, the curve instructions in the interval will move to the point based on the offset.

Ltd.

2 Interval Welding

Interval welding is a function designed for realizing regular welding track in welding operation. Through simple parameter setting, this function can be used in the basic program editing interface in the form of instructions.

2.1 Interval welding instructions

In the basic programming interface, select "Task type" - "Function statement", we can see two function instructions for interval welding, as shown in Figure 2.1

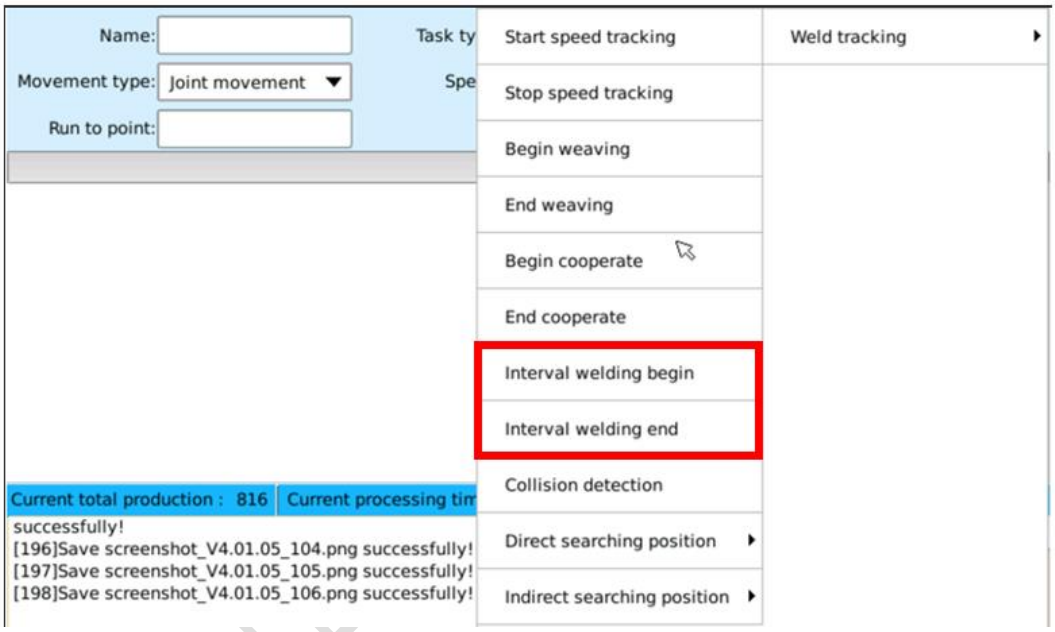


Figure 2.1 "Interval welding begin" and "interval welding end" function instruction pair

2.2 Function modes

As shown in Figure 2.2, four modes are provided in the "interval welding begin" function instruction interface, namely "space mode", "time mode", "reciprocating mode" and "stay mode".

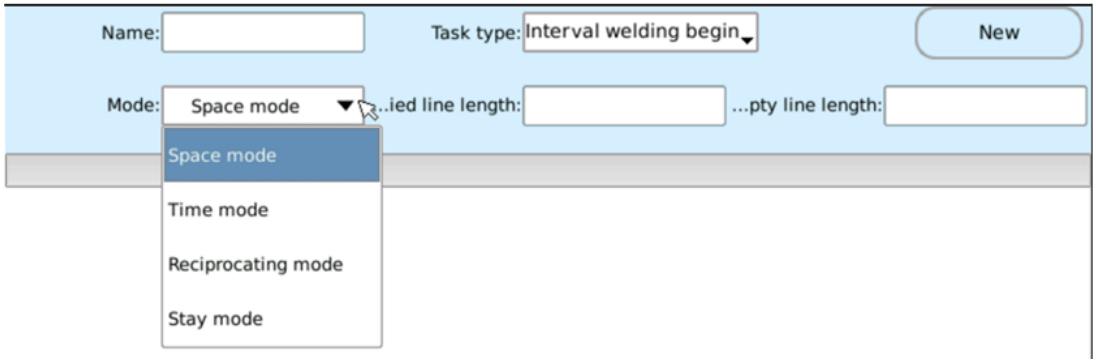


Figure 2.2 Four modes of interval welding function

2.2.1 Space mode

In this mode, the robot welds intermittently. As shown in Figure 2.3, it is necessary to set the occupied line length (unit: mm) and empty line length (unit: mm). Where: the occupied line length is the welding operation length, and the empty line length is the non-welding operation length.

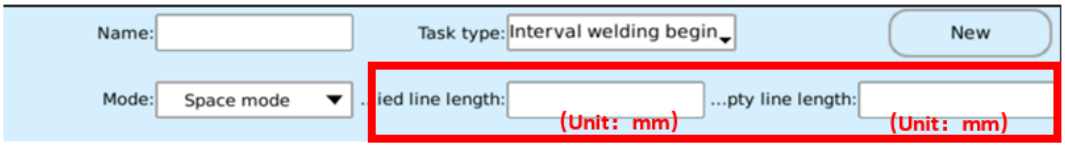


Figure 2.3 Parameter setting of "space mode" interval welding

The example and trajectory of welding procedure with space mode interval welding function are shown in Figure 2.4. The welding track will start from point A, weld first, then empty line, then weld, and then empty line, and complete the whole section to point B according to this law.

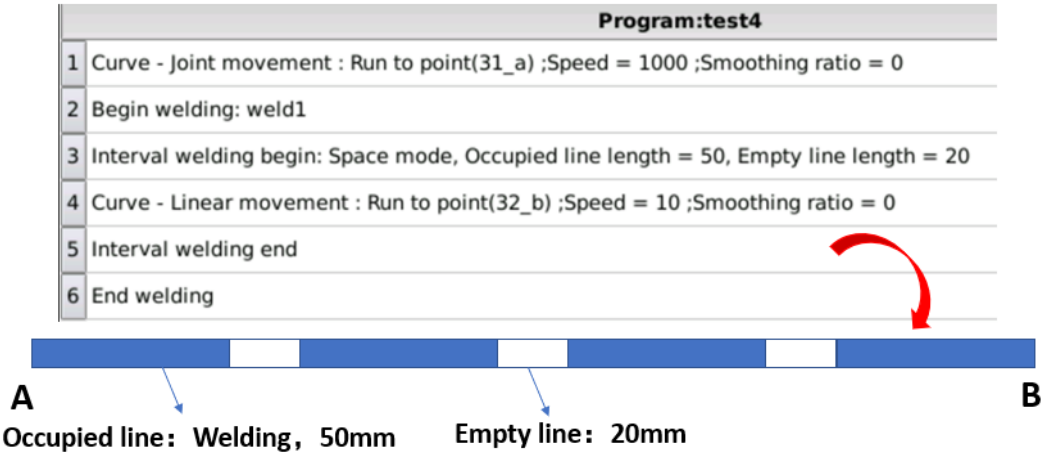


Figure 2.4 Example of "space mode" welding procedure and welding track

2.2.2 Time mode

In this mode, the robot welds intermittently. As shown in Figure 2.5, it is necessary to set the empty line length (unit: mm) and residence time (unit: ms). Where: the length of empty line is the length of non-welding operation, and the residence time is the length of welding at a stationary point.

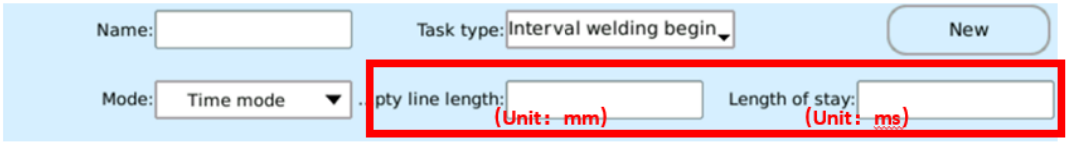


Figure 2.5 Parameter setting of "time mode" interval welding

The example and track of welding procedure with time mode interval welding function are shown in Figure 2.6. The welding track will start from point A, first stop at point A and weld at this point for the length of stay time, then move without welding for the empty line, then stop and weld for

the length of stay time, and then move without welding for the empty line, and complete the whole section to point B according to this law.

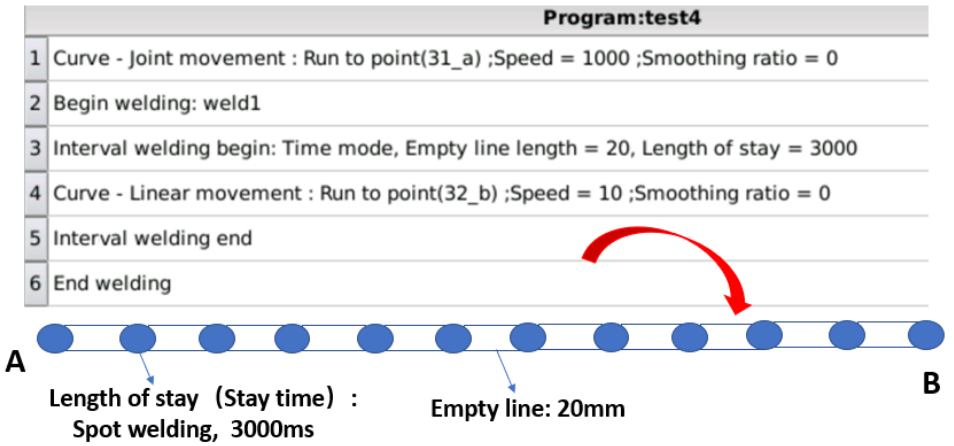


Figure 2.6 Example of "time mode" welding procedure and welding track

2.2.3 Reciprocating mode

In this mode, the robot continuously welds. As shown in Figure 2.7, the occupied line length (unit: mm) and backward length (unit: mm) need to be set. Where: the occupied line length is the single-layer welding operation length, and the backward length is the fallback welding operation length.

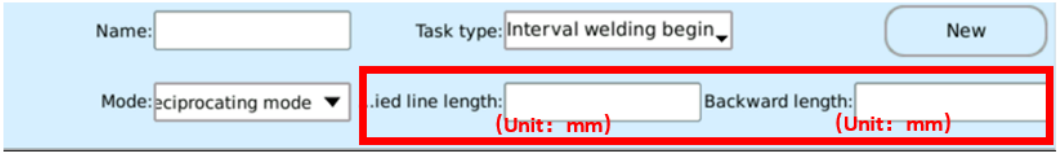


Figure 2.7 Parameter setting of " reciprocating mode" interval welding

Figure 2.8 shows an example and trajectory of welding procedure with reciprocating mode interval welding function. The welding track will start from point A, first weld " occupied length + backward length", then back weld " backward length", then continue to weld " occupied length + backward length", and then back weld "backward length", and complete the whole section to point B according to this law.

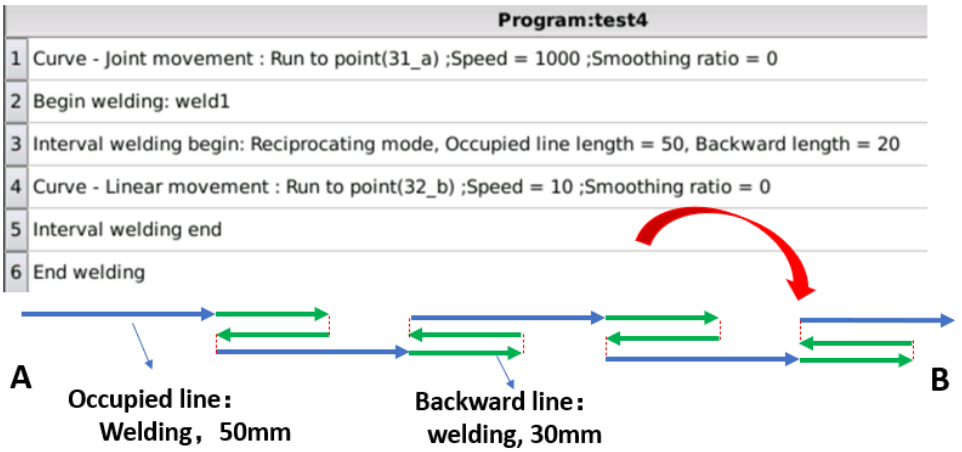


Figure 2.8 Example of "reciprocating mode" welding procedure and welding track

2.2.4 Stay mode

In this mode, the robot continuously welds. As shown in Figure 2.9, it is necessary to set the occupied line length (unit: mm) and stay time (unit: ms). Where: the occupied line length is the welding operation length, and the stay time is the welding time at a certain point.

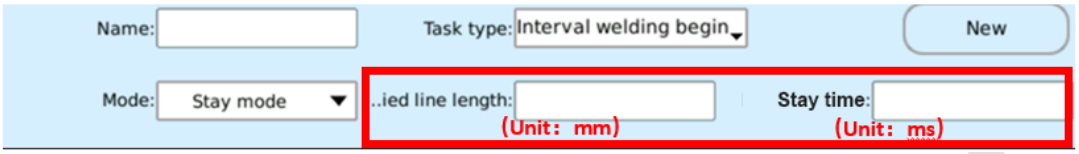


Figure 2.9 Parameter setting of " reciprocating mode" interval welding

An example of a welding procedure using the "reciprocating mode" interval welding function and its trajectory are shown in Figure 2.10. The welding track will start from point A, weld for the stay time at point A first, then weld the occupied line length, then stop and weld for the stay time, and then weld the occupied line length. Complete the whole section to point B according to this law.

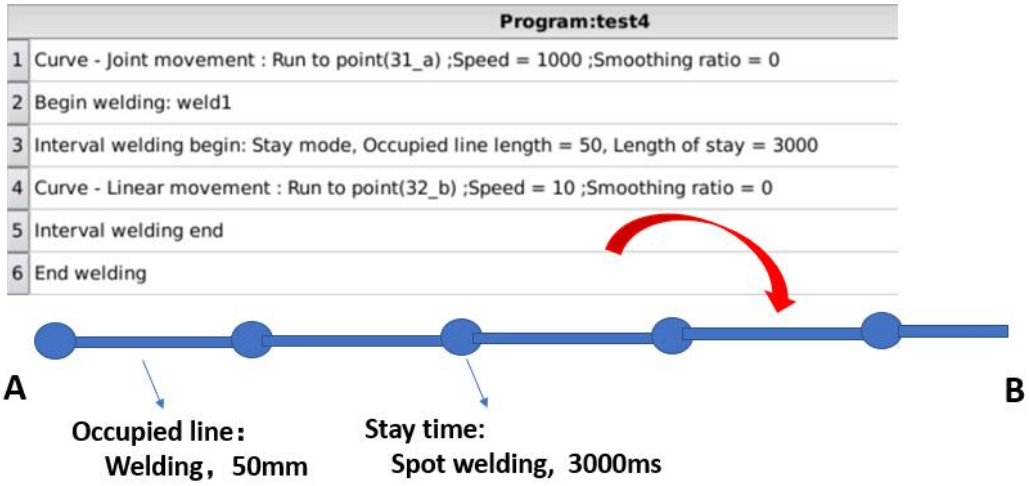


Figure 2.10 Example of "stay mode" welding procedure and welding track

3 Weaving

During the automatic welding, the welding with weaving has been widely adopted as a method to improve the welding efficiency. In order to reduce the robot teaching time, weaving function for welding is provided in the Betrun control system.

3.1 Create weaving function package

Click [new] - [function] - [swing arc] to create a "swing arc" function package through setting, which is used to realize robot swing arc welding in the program.

To create a Z letter form weaving function package, we need choose Z type for the weaving mode. The parameters to be set include: weaving frequency (Hz), weaving amplitude (mm), left stay time (ms) and right stay time (ms).

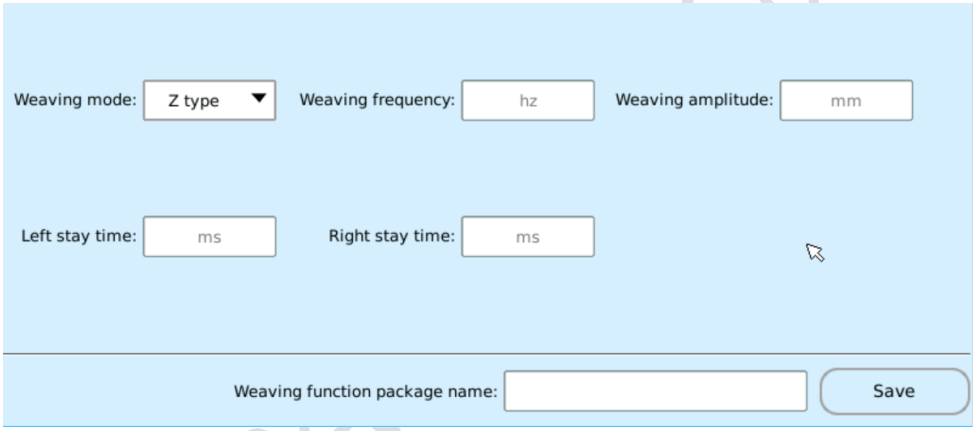


Figure 3.1 setting of Z-type weaving function package

To create a arc form weaving function package, we need choose Arc for the weaving mode. The parameters to be set include: weaving frequency (Hz), weaving amplitude (mm), left stay time (ms), right stay time (ms) and circular pendulum radius (mm).

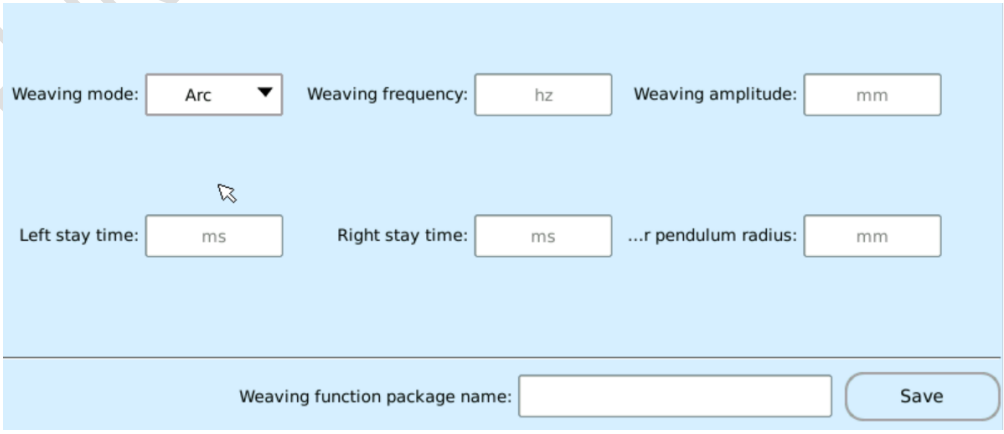


Figure 3.2 setting of arc weaving function package

3.2 Program case

When writing the program, the weaving function and the welding process need to be used simultaneously. The order of use is: "begin welding", then "begin weaving", the linear motion command of the welding direction, "end weaving", and finally "end welding". The program case is shown in Figure 3.3.

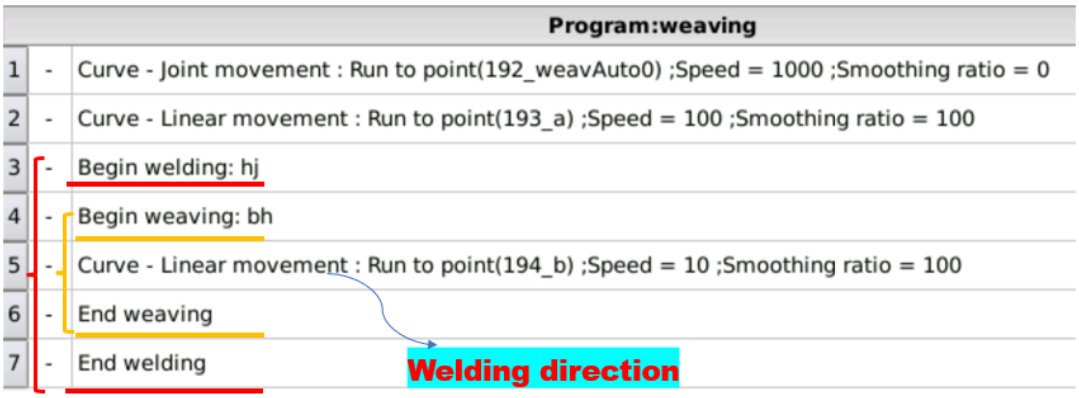


Fig. 3.3 Use of the weaving function in program

3.3 Parameter test

According to the parameter setting in Figure 3.1, create a new Z-type weaving function package, and then write the program, as shown in Figure 3.4, and the schematic diagram of running track is shown in Figure 3.5. Where, the unit distance is equal to the welding linear motion speed divided by the weaving frequency.

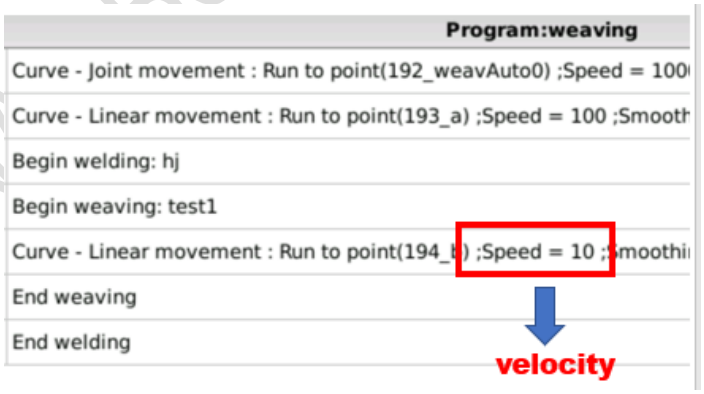


Figure 3.4 Program case of Z type weaving function package

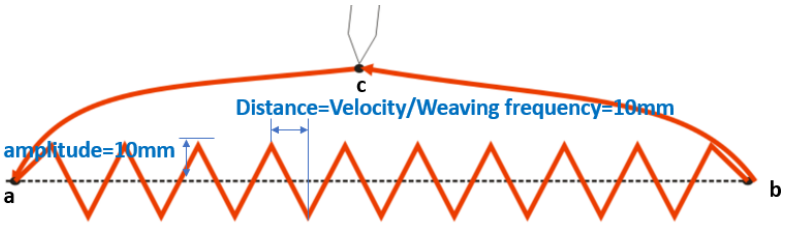


Figure 3.5 Schematic diagram of Z type weaving track

According to the parameter setting in Figure 3.2, create a new arc weaving function package, and then write the program, as shown in Figure 3.6, and the schematic diagram of running track is shown in Figure 3.7.

		Program:weaving
1	-	Curve - Joint movement : Run to point(192_weavAuto0) ;Speed = 1(
2	-	Curve - Linear movement : Run to point(193_a) ;Speed = 100 ;Smoc
3	-	Begin welding: hj
4	-	Begin weaving: test2
5	-	Curve - Linear movement : Run to point(194_b) ;Speed = 10 ;Smoot
6	-	End weaving
7	-	End welding

Figure 3.6 Program case of arc weaving function package

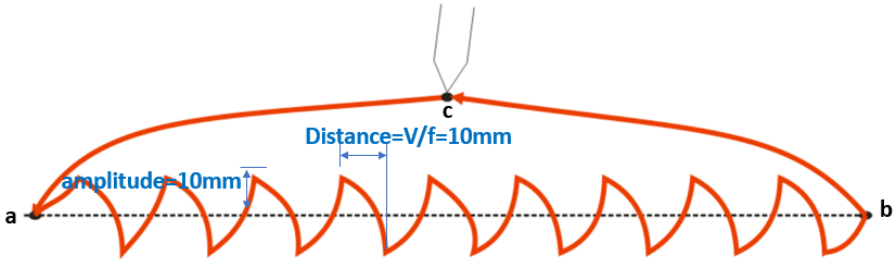


Figure 3.7 Schematic diagram of arc weaving track

3.4 Swing in place

For welding operations that require additional joint cooperation, such as long linear welds and large arc welds, Betrun system provides the function of "swing in place", that is, the additional joint drives the weld to move, and the welding gun performs swing welding in place.

The steps are as follows:

Step 1: create the weaving function package

To realize swing in place, first create a weaving function package, and then, as shown in Figure 3.8, when inserting the weaving function instruction, you need to check "swing in place" in the programming interface.

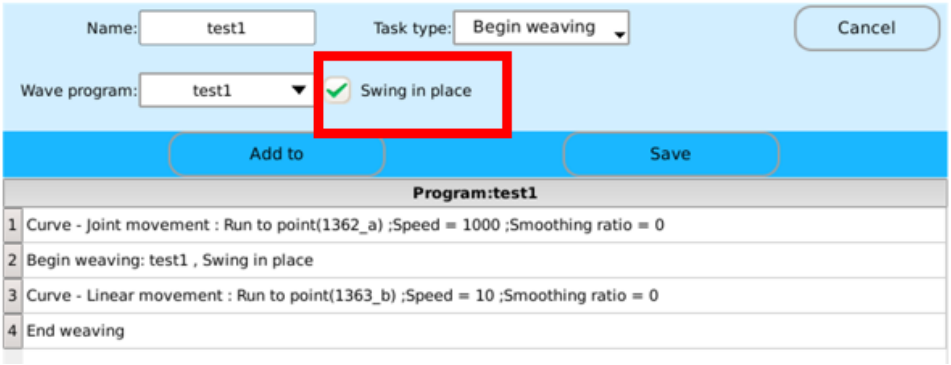


Figure 3.8 Check "swing in place" in the programming interface

Step2: manual teaching

First, as shown in Figure 3.9, first teach the center point of the swing in place, and manually teach the additional axis to the start point. As shown in Figure 3.10, the point j1-j6 joint data and the tool coordinate system determine the center point of the swing of the welding gun, and this point contains the joint data of the start point of the additional axis. This point is referenced as the start point of linear movement instruction between the swing in place function instruction pair.

Note: the additional joint needs to be set to joint mode

Then, teach the motion end point of the additional axis and the swing arc direction point. Figure 3.11 shows the direction of the swing and the end point information. This point is referenced as the end point of linear motion between the swing in place function instruction pair.

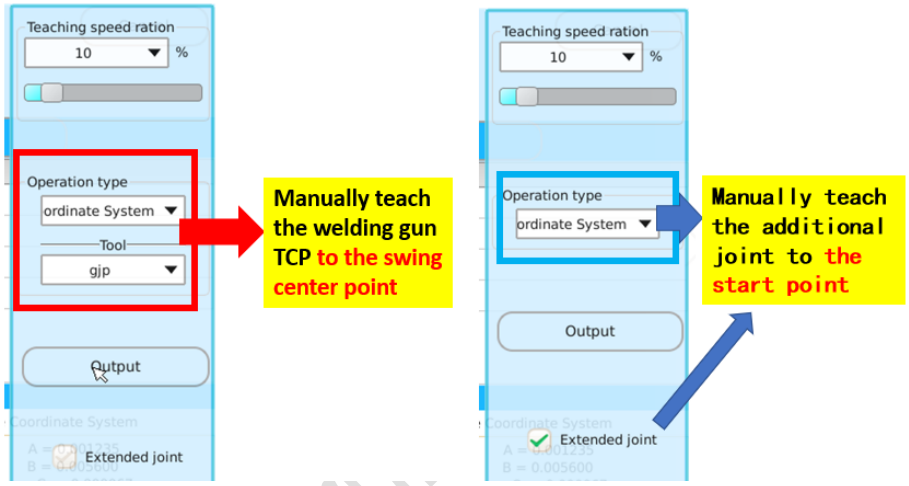


Figure 3.9 Schematic diagram of manual teaching configuration

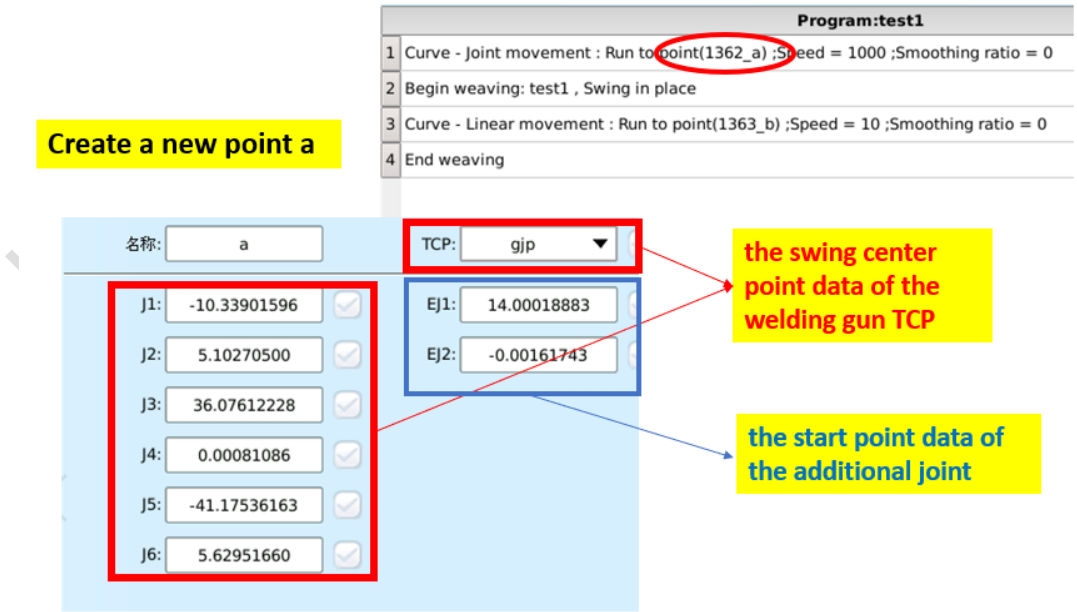


Figure 3.10 Start point information and reference

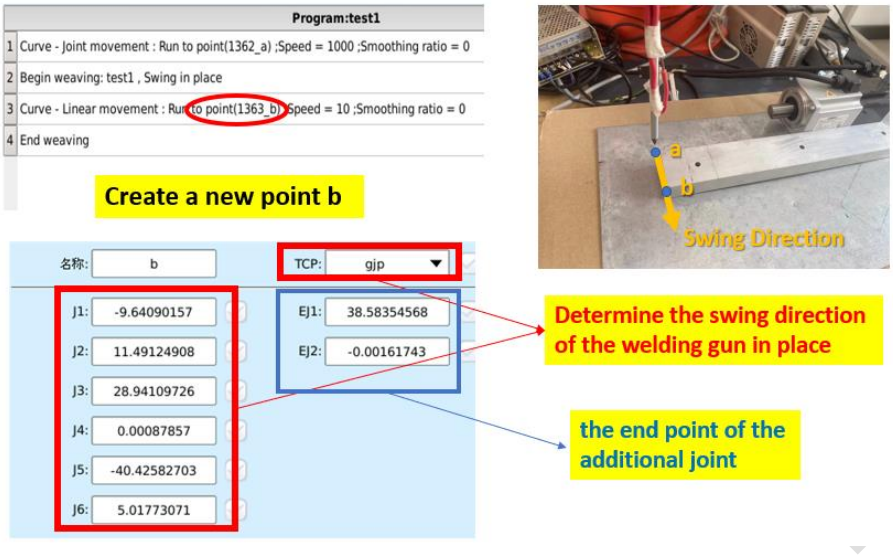


Figure 3.11 Direction of the swing and the end point information

Run the program, the welding gun of the robot will take point a as the center and swing the arc in place in the direction of ab. at the same time, the additional axis starts from the starting point and runs to the end point according to the linear motion speed.

4 Cooperative movement

In actual production, more workpieces need to be displaced during welding to make the weld in a better position (posture). In this case, the positioner and robot can move separately, that is, after the positioner is displaced, the robot is welded again; It can also move at the same time, that is, the positioner moves while the robot welds, that is, the positioner and the robot move together. At this time, the motion of the positioner and the motion of the robot are combined, so that the motion of the welding gun relative to the workpiece can not only meet the requirements of the weld trajectory, but also meet the requirements of the welding speed and the posture of the welding gun. This composite motion is called the cooperative movement of the additional axis and the robot body.

In order to realize cooperative movement, this system encapsulates this part of work into "cooperation" function package.

4.1 Create collaboration function package

Click [new] - [function] - [collaboration], as shown in Figure 4.1, you can create a "collaboration" function package by setting up the program to invoke the program to realize the additional axis (turntable 1 (pitch), turntable 2 (roll) and slide rail) and work with the robot.

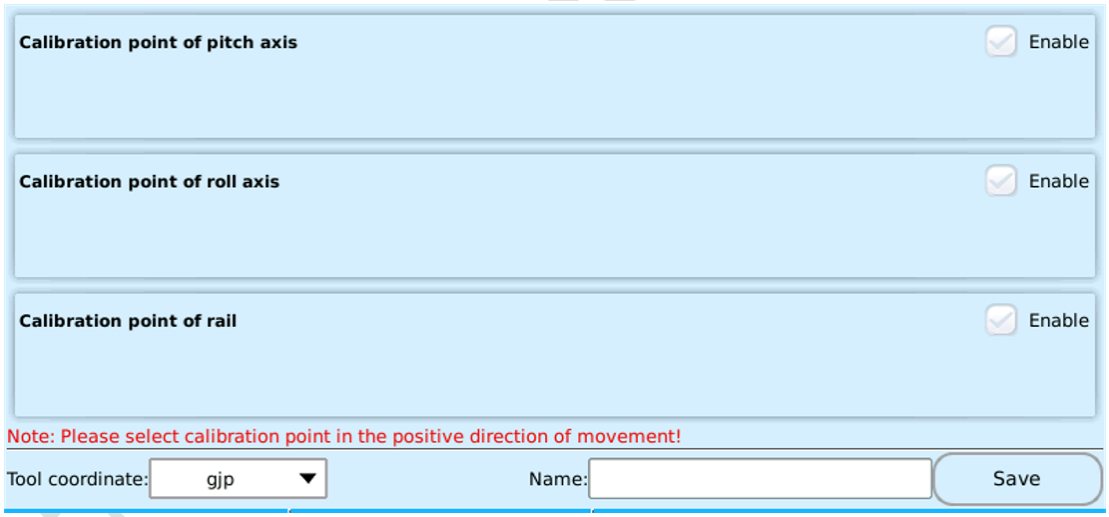


Figure 4.1 Collaboration function package settings

4.1.1 Preparation

Before applying the cooperative function between the additional axis and the robot body, it is necessary to check whether the motion direction of each axis (including the additional axis) of the robot is normal, whether the straight-line motion of the robot is a straight line, and detect the "zero position" of the robot.

Note: the use of additional axes for collaboration needs to be set to "joint mode".

4.1.2 Establish tool coordinate system

In order to make the robot carry out correct linear interpolation, arc interpolation and other interpolation actions, it is necessary to correctly input the size information of the welding gun tool and define the position of the control point. Therefore, it is necessary to establish the tool coordinate system.

4.1.3 Create the calibration points

In order to achieve cooperation, three calibration points need to be established in the tool coordinate system.

Take slide rail cooperation as an example to illustrate the establishment method of calibration point:

(1) TCP touches the calibration rod along the Z axis of the tool coordinate system to create a new point.

(2) Move the slide rail in the positive direction for a certain distance. TCP touches the calibration rod to create a second point.

(3) Continue to move the slide rail in the positive direction for a certain distance. TCP touches the calibration rod to create a third point.

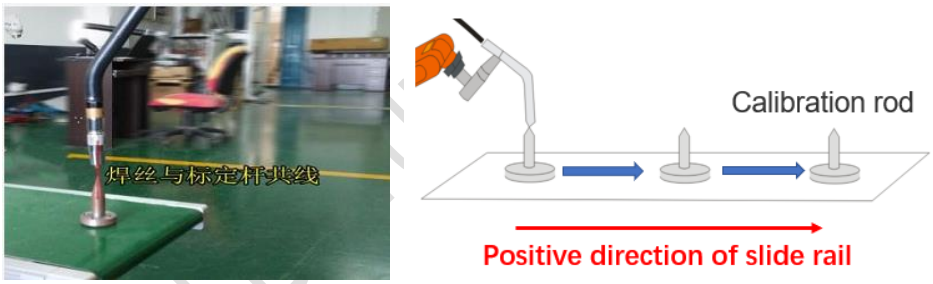


Figure 4.2 New calibration point

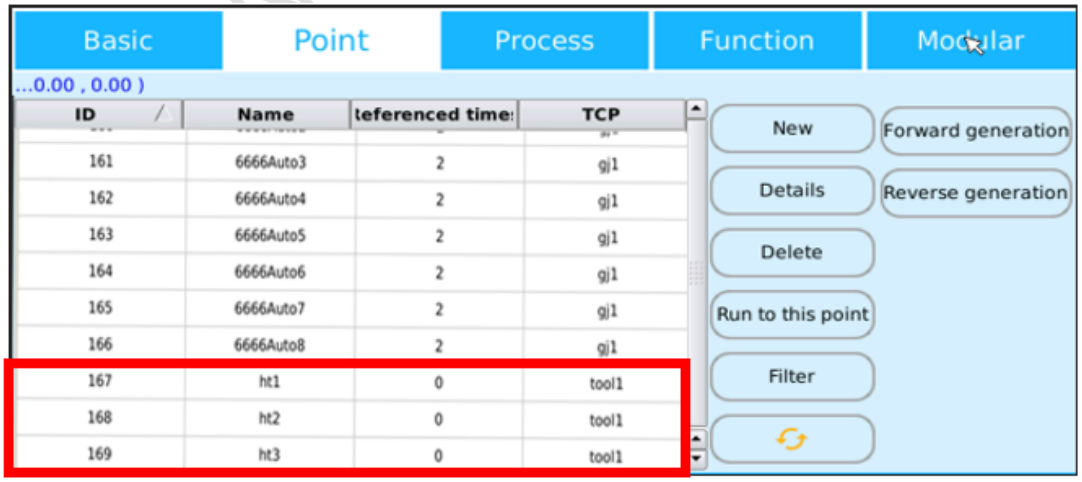


Figure 4.3 Create calibration points

Note:

(1) The tool coordinate system needs to be checked for the calibration points.

(2) When there are multiple additional axes, other additional axes need to be in the home position

when determining the calibration points of the slide rail.

4.1.4 Create the cooperation package

After the three calibration points are determined, click [New] – [function] – [cooperation] to create a cooperation function package:

(1) Create a cooperation function name, such as "slide cooperation".

(2) Enables the external axis that requires collaboration.

(3) Determine that the axis number is 0 or 1.

(4) Determine the name of the tool coordinate system to use, here is "gjp"

(5) Reference the created points (id167, id168, id169) to P1, P2 and P3. Long press point P to reference the created point.

(6) Saving the cooperation function package will automatically calibrate the reduction ratio of the additional shaft and import it into the system. As shown in Figure 4.4, the calibrated deceleration ratio is 128.24.

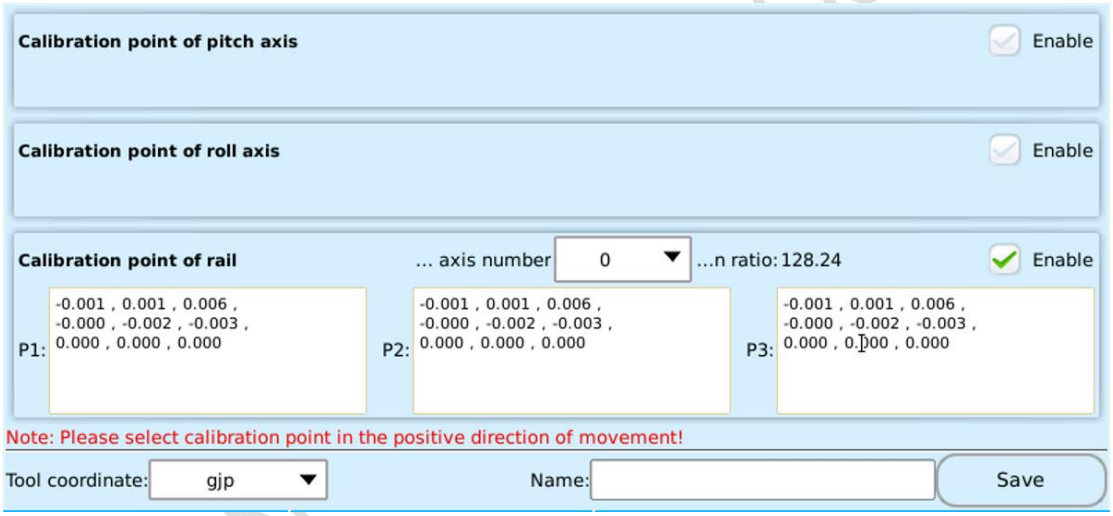


Figure 4.4 Create the cooperation function package

Note: multiple additional axes can be “enabled” to cooperate at the same time, but the calibration process for each additional axis must be carried out on the premise that other axes are in the home position.

4.2 Use of the cooperation function package

By adding function instructions, the cooperation package can be invoked in the program.

Note: according to the "enable" configuration of the function package, set the corresponding additional axis as disabled or enabled.

Instructions need to appear in pairs. When the curve motion command between the two instructions is taught manually, move the robot joint and the additional axis to the specified position respectively, and then determine the "run to point".

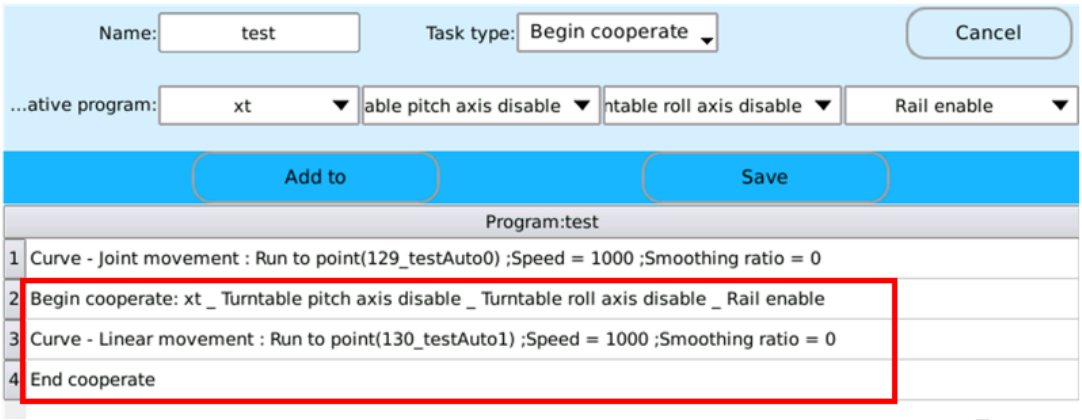


Figure 4.5 Add cooperation function instruction

Note:

(1) The teaching point of the motion curve must be brought with the tool coordinate system.

(2) If slide rail cooperation is not used, do not move the slide rail at the motion curve. The same is true for other cooperation axes.

(3) You can choose to enable one or all additional axis cooperation according to the actual needs.

(4) The internal curve of cooperation function must be "linear movement" or " arc movement".

(5) If the current position is in the middle of the arc curve when pausing, you can't run directly from the current curve command when running again. You need to return to the previous curve to start running.

Co.,Ltd.

5 Tracking

In some cases, the working object of the robot is the workpiece running on the assembly line at a uniform speed. For this application, it is relatively complex and low precision to carry out manual teaching during assembly line operation. Therefore, the system provides the "tracking" function package, so that the workpiece can be in a static state during manual teaching, and then the operation program can be written. By calling the "tracking" function, the robot can automatically calculate the motion trajectory and complete the assembly line operation.

There are two ways of assembly line tracking: ontology tracking and slide tracking. "Ontology tracking" is the use of manipulator body to realize assembly line motion tracking, which is suitable for small-scale tracking applications; "Slide tracking" uses an additional external slide to realize pipeline motion tracking, which is suitable for a wide range of tracking applications.

5.1 Hardware preparation

To realize assembly line tracking, the following hardware conditions are required at least:

- Assembly line: install the encoder or know its running speed. When installing the encoder, the last two channels of the 22 channel optocoupler isolation input port are used as the phase A and B ports of the external encoder, as shown in Figure 5.1.
- Slide: required for slide tracking, not required for small-scale body tracking.
- Sensor or limit switch: used for tracking trigger.

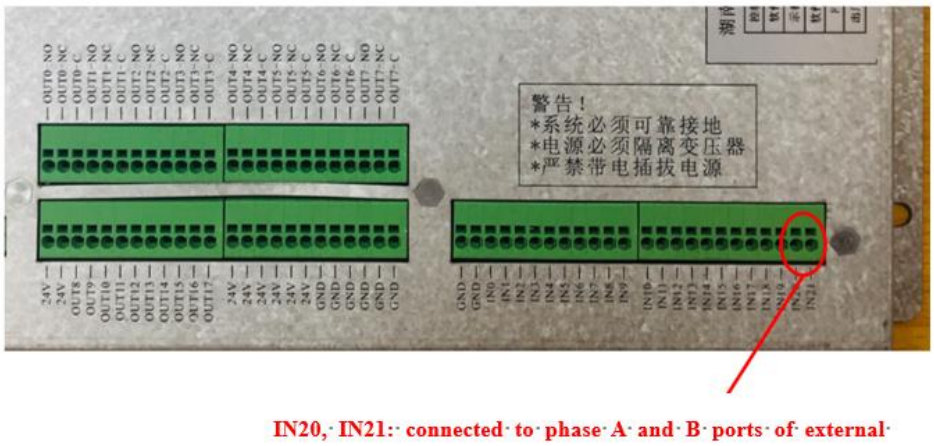


Figure 5.1 System connected to phase A and B ports of external encoder

5.2 Create ontology tracking function package

Click [New] – [function] – [tracking] to create a "tracking" function package through settings.

According to different encoder types, "ontology tracking" can be divided into "without encoder" and "pulse type".

5.2.1 No encoder

Select the "no encoder" function, and the parameters to be set are shown in Figure 5.2.

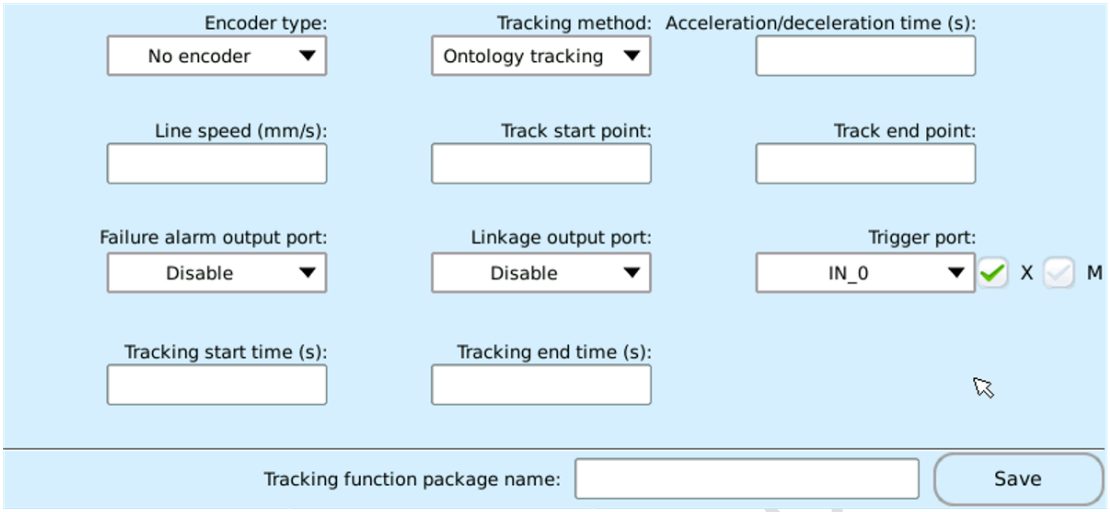


Figure 5.2 Ontology tracking parameter setting without encoder

The description is shown in Table 5.1, and the parameter illustration is shown in Figure 5.3.

Table 5.1 Description of ontology tracking function parameters without encoder

Parameter	Description
Acceleration/deceleration time (s)	The acceleration/deceleration time determines the "acceleration/deceleration speed". The smaller the time, the greater the "acceleration/deceleration speed". The unit is "s".
Line speed	Ontology tracking speed, unit: mm/s.
Track start point, Track end point	Determine the direction of ontology tracking and move in a straight line from "start point" to "end point".
Failure alarm output port	It is connected with remote monitoring through system I/O output port. When the tracking function program is "triggered" and has not been "tracked" after the "tracking end time", it indicates that the "tracking" failed and will be displayed on the remote monitoring end. It is unnecessary to set when this function is not used.
Linkage output port	By associating the system I/O output port with the assembly line, the "pause", "continue" and "start" of the system can also control the operation of the assembly line. It is unnecessary to set when this function is not used.
Trigger port	Determine an I/O input X or relay M port, and provide a high level "1" when triggered by the sensor or limit switch.
Tracking start time	After the workpiece triggers the sensor, the "tracking" time starts, and the unit is "s".
Tracking end time	The time to end "tracking" after the workpiece triggers the sensor, and the unit is "s". If you haven't "tracked" after this setting time, it means that the "tracking" failed this time.

HUNAN Betrun Robot

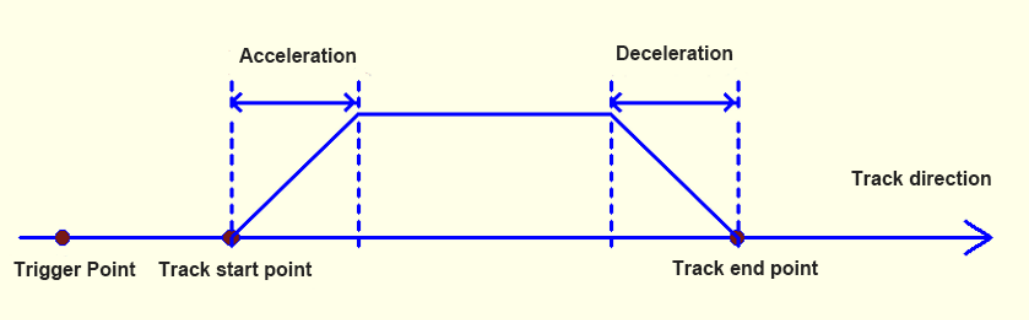


Figure 5.3 Parameter illustration

The method of creating a function package is as follows:

(1) Create tracking function package name.

(2) Open the assembly line and start timing when the workpiece triggers the sensor or limit switch. When it reaches the "track start point" position, record the time and set it to "tracking start time". Manually teach the robot tool TCP to this position, and click the input box of "track start point" to create a new "track start point".

(3) When the workpiece reaches the "track end point" with the assembly line, set the time as "tracking end time". Manually teach the robot tool TCP to this position, and click the input box of "track end point" to create a new "track end point".

(4) After parameter setting, click Save to generate tracking function package.

5.2.2 Pulse encoder

After selecting the tracking function of "pulse type ontology tracking", its parameter setting is shown in Figure 5.4 and the description is shown in Table 5.2.

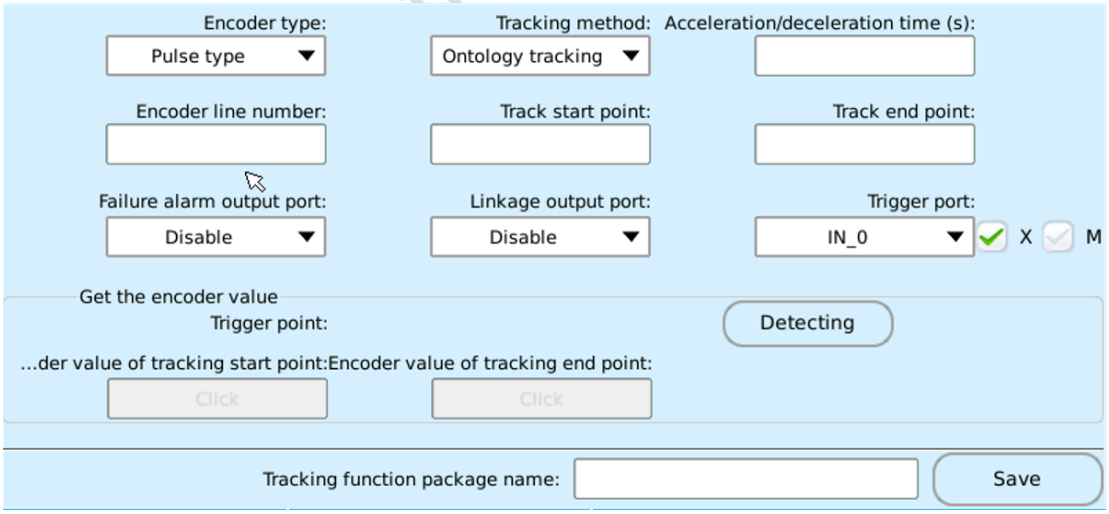


Figure 5.4 Parameter setting of pulse type ontology tracking

Table 5.2 Description of pulse type ontology tracking function parameters

Parameter	Description
Acceleration/deceleration time (s)	The acceleration/deceleration time determines the "acceleration/deceleration speed". The smaller the time, the greater the "acceleration/deceleration speed". The

	unit is "s".
Encoder line number	The tracking speed of the body is determined by the specification of the pulse encoder.
Track start point, Track end point	Determine the direction of ontology tracking and move in a straight line from "start point" to "end point".
Failure alarm output port	It is connected with remote monitoring through system I/O output port. When the tracking function program is "triggered" and has not been "tracked" after the "tracking end time", it indicates that the "tracking" failed and will be displayed on the remote monitoring end. It is unnecessary to set when this function is not used.
Linkage output port	By associating the system I/O output port with the assembly line, the "pause", "continue" and "start" of the system can also control the operation of the assembly line. It is unnecessary to set when this function is not used.
Trigger port	Determine an I/O input X or relay M port, and provide a high level "1" when triggered by the sensor or limit switch.
Encoder value of tracking start point	Position detection: the coding value when the encoder runs to the "tracking start point" from the trigger of I/O input port.
Encoder value of tracking end point	Position detection: the coding value when the encoder runs to the "tracking end point" from the trigger of I/O input port. If the encoder runs these "coded values" and has not started "tracking", it means that the "tracking" failed this time.

The method of creating a function package is as follows:

(1) Create tracking function package name.

(2) Open the assembly line, click [Detecting], when the workpiece trigger sensor reaches the tracking start point, click the input box under " Encoder value of tracking start point " to obtain the tracking start point encoder value, manually teach the robot tool TCP to this position, and click the input box of "track start point" to create a "track start point".

(3) When the workpiece arrives at the tracking end point along with the assembly line, click the input box under " Encoder value of tracking end point " to obtain the end point encoder value. At the same time, manually teach the robot tool TCP to this position, and click the input box of "track end point" to create a "track end point".

(4) After parameter setting, click Save to generate tracking function package.

5.2.3 No trigger port

When the robot carries out the pipeline tracking operation of the encoder body, it does not need to pass the external trigger signal, but calculates the pipeline tracking conditions based on the trigger spacing for motion control, and ensures the connection between products, or gives the corresponding spacing of products according to the actual situation.

As shown in Figure 5.5, products with a length of 200mm need to be tracked and fed by the assembly line, and there is no gap between products. It can be seen that the trigger spacing is 200mm

HUNAN Betrun Robot Co., Ltd.

long. In the pipeline function package setting interface in Figure 5.6, the trigger type can be selected as encoder, and the trigger spacing is set to 200mm.



Figure 5.5 Schematic diagram of product spacing

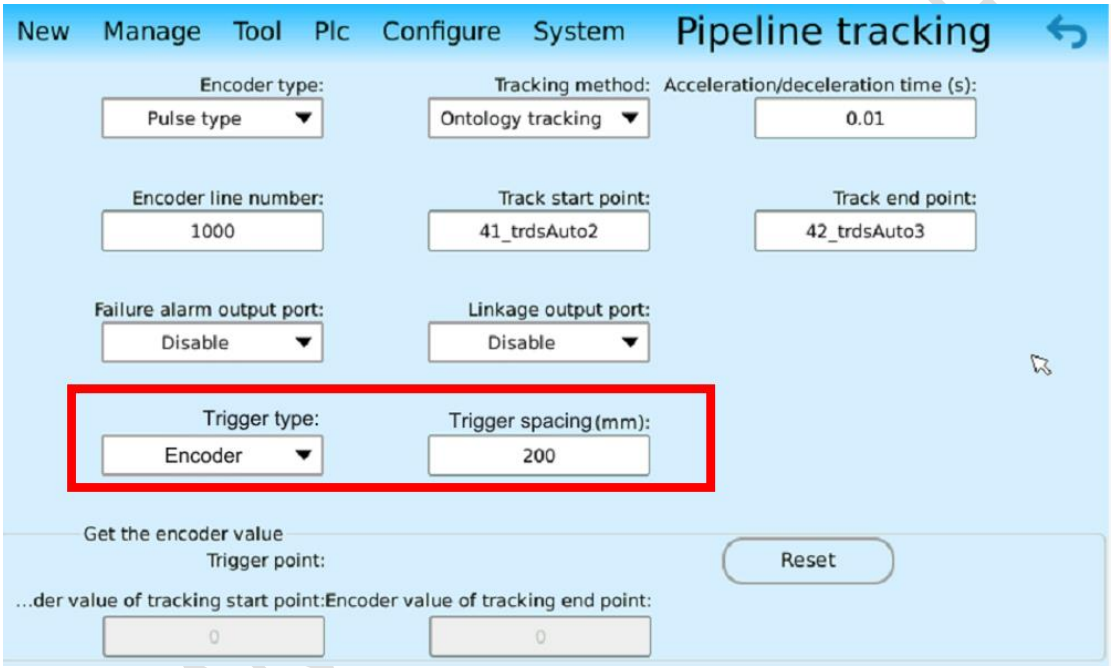


Figure 5.6 Encoder pipeline tracking through trigger spacing

Note:

In this way, the value of the encoder obtained from the tracking start point and tracking end point must be accurate. It is recommended to adopt the method of tip alignment to ensure that the actual trigger spacing is consistent with the set trigger spacing.

As shown in Figure 5.7, open the pipeline, click [start detection trigger point], open the pipeline to run for a short distance, stop the pipeline, click the input box under "tracking starting point encoder value" to obtain the tracking starting point encoder value, manually teach the robot tool TCP to this position, and click the input box of "tracking starting point" to create a "tracking starting point"; When the workpiece arrives at the tracking end point along with the pipeline, click the input box under "tracking end point encoder value" to obtain the end point encoder value. At the same time, manually teach the robot tool TCP to this position, and click the input box of "tracking end point" to create a "tracking end point".

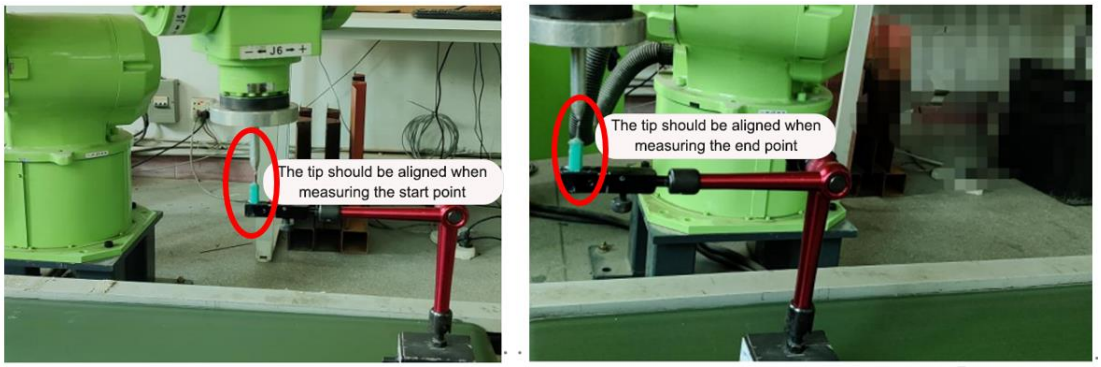


Figure 5.7 Track the start point and the end point by means of tip alignment

5.3 Create slide tracking package

Like "ontology tracking", according to different encoder types, "slide tracking" mode is also divided into "no encoder" and "pulse type" encoder types.

5.3.1 No encoder

If the tracking function of "no encoder" is selected, the "additional shaft number" parameter needs to be configured. Before setting the "additional axis number" parameter, the additional axis of the connected slide rail must be set to "joint mode" in "machine parameters".

5.3.2 Pulse encoder

The "additional shaft number" parameter needs to be configured. Before setting the "additional axis number" parameter, the additional axis of the connected slide rail must be set to "joint mode" in "machine parameters".

5.4 Use of tracking function package

By adding function instructions, the tracing function package can be invoked in the program. The start and stop instructions need to appear in pairs. The trajectory between the instruction pairs is obtained by manually teaching the robot to operate the workpiece on the assembly line when the assembly line is stationary.

The program case is shown in Figure 5.8.

		task list	
1	-	Curve - Joint movement : Run to point(212_testAuto0) ;Speed = 100...	
2	-	Start speed tracking: lsxgz	
3	-	Curve - Linear movement : Run to point(213_testAuto1) ;Speed = 10...	
4	-	Curve - Linear movement : Run to point(214_testAuto2) ;Speed = 10...	
5	-	Stop speed tracking	

Figure 5.8 Program case of tracking function package

HUNAN Betrun Robot Co., Ltd.

6 Collision detection

In order to enhance the safety of the operation process of industrial robots, Betrun system provides the function of robot collision detection, so that the robot can stop moving immediately in case of accidental collision with surrounding objects during operation, so as to avoid major damage.

The realization of collision detection function includes three steps: (1) program running parameter sampling; (2) Create collision detection function package; (3) Use of collision detection function.

6.1 Program running parameter sampling

As shown in Figure 6.1, in the automatic mode of the teach pendant, set the speed ratio to 100% and the operation mode to circular motion, so that the robot can cycle the program according to the operation speed and reproduce the whole operation process.

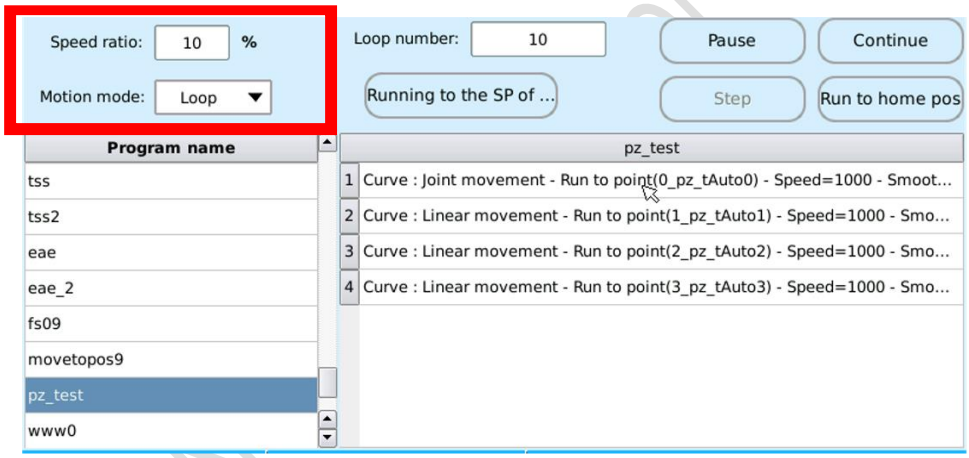


Figure 6.1 Automatic mode interface setting of teach pendant

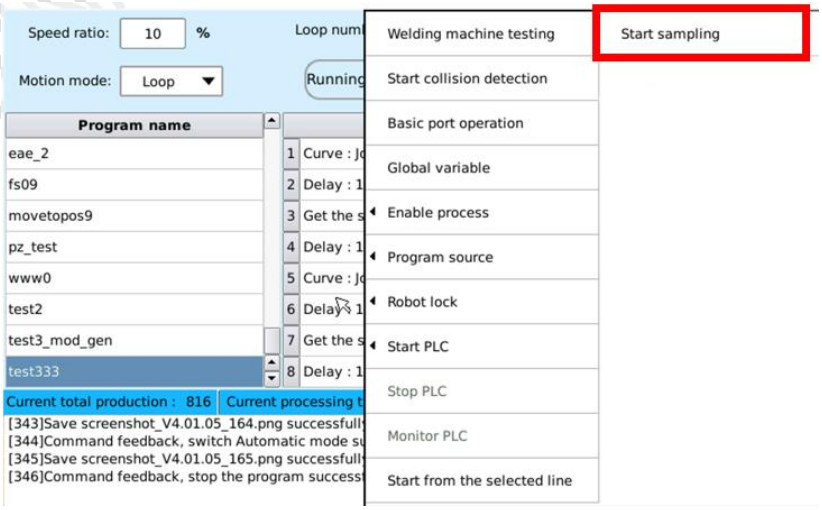


Figure 6.2 Menu bar start sampling button

After the robot starts the operation, as shown in Figure 6.2, click the [menu] button and select "start sampling". The sampling process needs to run the operation procedure for 5-7 times, then click the [menu] button and select "Stop Sampling". The collected parameters appear in the operation record display area, as shown in Figure 6.3, indicating that the collection is successful.

Note: in order to ensure the accuracy of sampling parameters, the load when the robot runs the program needs to be exactly the same as that in the actual operation.

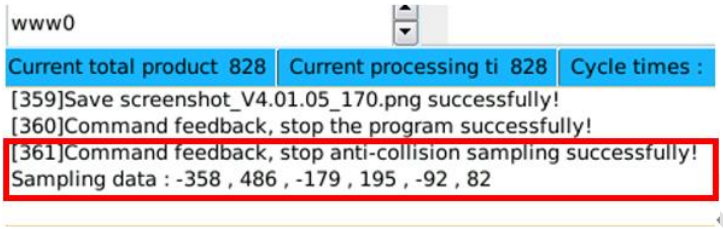


Figure 6.3 Collected parameters appear in the operation record display area

6.2 Create collision detection function package

Switch to the manual mode of the teaching pendant, and click [new] - [function] - [collision detection] to enter the collision detection function package creation page. As shown in Figure 6.4, the parameters collected in the automatic mode have been automatically filled in the relevant area. Just enter a name and click save.

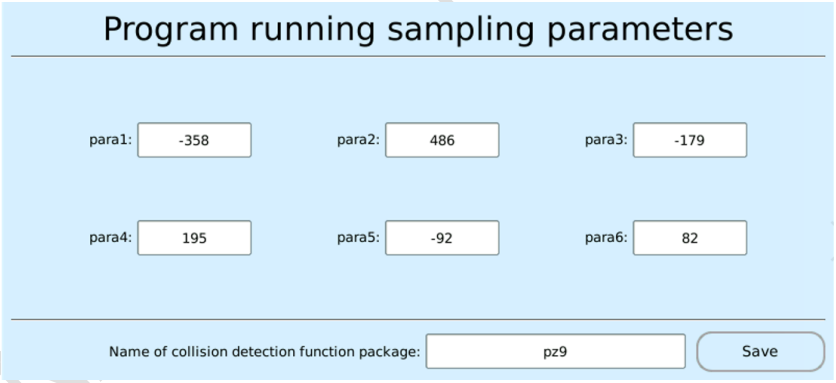


Figure 6.4 "Collision detection" function package creation interface

6.3 Use of collision detection function package

There are two ways to use the collision detection function package.

6.3.1 Use collision detection instruction

In the "program basic instruction" interface, open the "program basic instruction" function. As shown in Figure 6.5, the instruction needs to select the collision detection function package, set the sensitivity level, and alarm output port. Among them: the sensitivity level is 1-5. The lower the level,

the higher the anti-collision sensitivity.

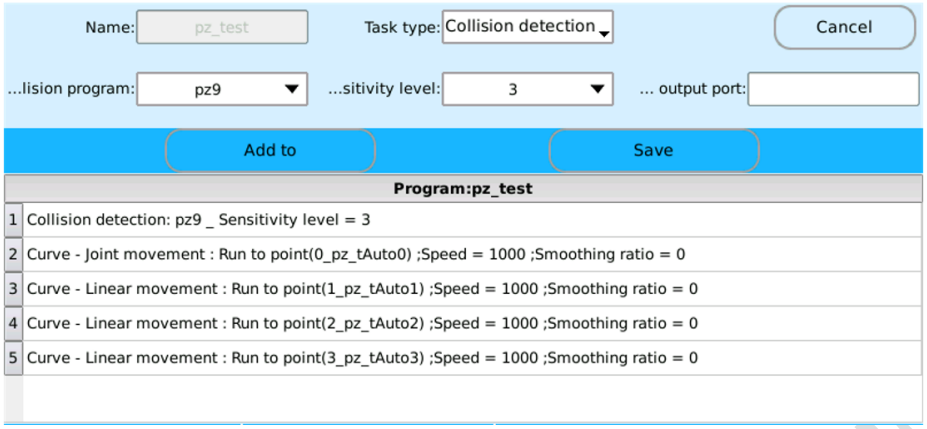


Figure 6.5 "Collision detection" instruction

6.3.2 Start collision detection in automatic mode

As shown in Figure 6.6, in the automatic mode of the teach pendant, click [menu] and select "start collision detection". In the pop-up window (Figure 6.7), select the created collision detection function package, set the sensitivity level, and alarm output port. No matter whether the program is running or not, the collision detection will start immediately and remain effective until you click "stop collision detection" in [menu].

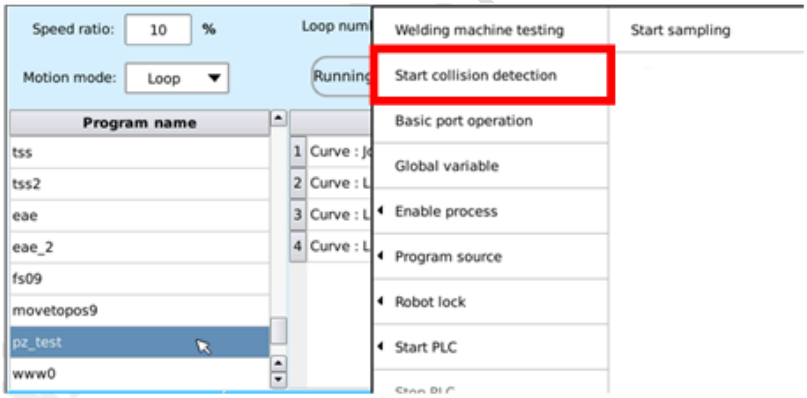


Figure 6.6 Start collision detection in menu bar

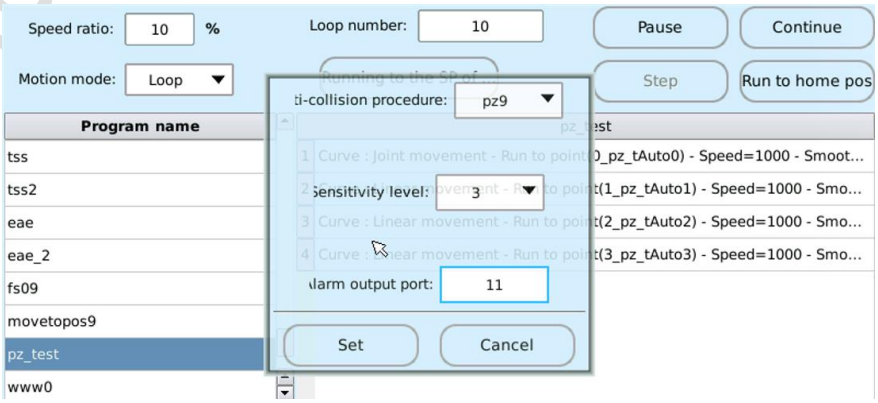


Figure 6.7 Collision detection configuration window

Part III Process Package Instructions

1 Palletizing

Palletizing is to stack the stacks on the pallet according to a certain arrangement law.

As shown in Figure 1.1, it is a typical application of palletizing robot. The robot takes the stack from the assembly line, then moves to the specified position and angle of the pallet, and releases the stack to place the stack.

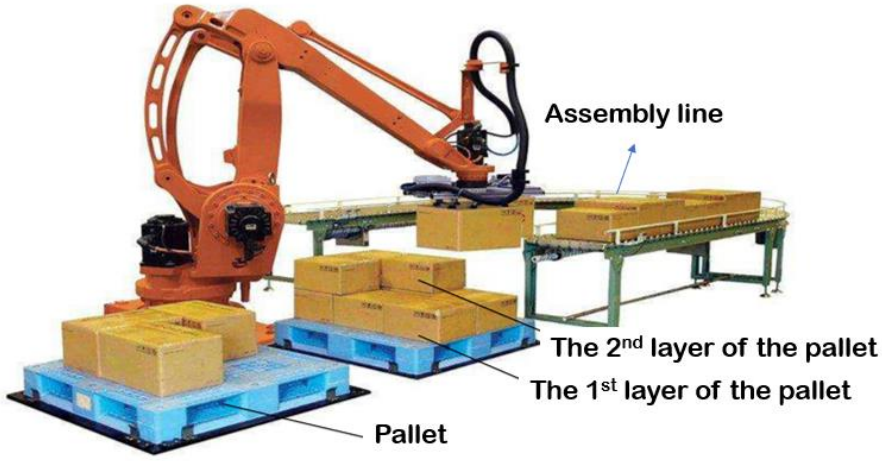


Figure 1.1 Robot palletizing

For palletizing process, manual teaching robot trajectory usually includes two parts:

- Assembly line: transition point (near), prepare to take point, taking point and leaving point after taking
- Pallet stacking: transition point (enter), prepare to release point, release point, leaving point after releasing, transition point (away)

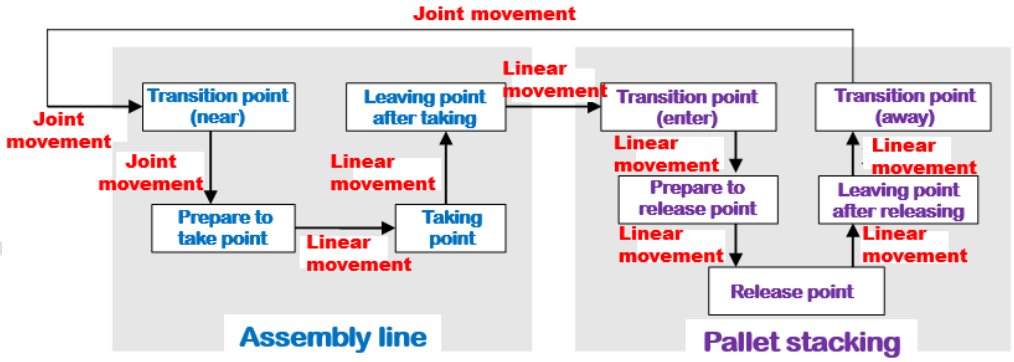


Figure 1.2 Flow chart of robot palletizing trajectory

As shown in Figure 1.2, the characteristics of robot trajectory are:

(1) The taking and releasing process of each stack is the same, except for the track points.

(2) For a certain assembly line, the moving track points of each stack are the same.

(3) For a pallet, after the placement point of the first stack is determined, the placement point of each stack can be calculated in turn when the layer arrangement on the pallet and the spacing between stacks are known.

Therefore, Betrun system encapsulates the palletizing process of a single pallet into a process

package by simply setting the pallet user coordinate system, the placement position of the pallet, the pallet frame points, layout methods and floor height.

Then, for four different palletizing templates:

- Single assembly line single pallet
- Single assembly line double pallet
- Double assembly line single pallet
- Double assembly line double pallet

Betrun system creates the palletizing template by setting the assembly line frame points corresponding to the palletizing process package, and quickly generates the palletizing program to realize the automatic palletizing operation of the robot.

The terms involved in palletizing are shown in Table 1.1.

Table 1.1 Description of palletizing operation

Number	Term	Description
1	Stack	Items to be placed, such as workpieces, cartons, and products.
2	Pallet	Items (areas) used to place stacks.
3	Number of palletizing process packages	100 (pallet0-99). That is, it can support up to 100 pallets.
4	Layout method	Up to 99 different layout methods can be realized, usually one layout per layer.
5	First stack	The first stack of a layer on a pallet.
6	Transition point (enter)	The teaching point on the track for the robot moving to the pallet movement after taking the stack from the assembly line.
7	Transition point (away)	The teaching point on the track for the robot moving to the assembly line after releasing the stack on the pallet.
8	**Release point**	**The pose of the robot when the stack is placed at the origin of the user coordinate system. In the future, the pose of each stack is offset and calculated based on it.**
9	Prepare to release point	Generally, it is located directly above the release point, and it can also have a certain offset angle.
10	Leaving point after releasing	Generally, it is located directly above the release point, and it can also have a certain offset angle.
11	Transition point (near)	The teaching point on the track for the robot moving to the assembly line before taking the stack.
12	Taking point	The teaching point for taking stack on the assembly line.
13	Prepare to take point	Generally, it is located directly above the taking point, and it can also have a certain offset angle.
14	Leaving point after taking	Generally, it is located directly above the taking point, and it can also have a certain offset angle.

HUNAN Betrun Robot Co., Ltd.

1.1 Preparation

1.1.1 Create user coordinate system

In the process of palletizing, in order to stack accurately, the user coordinate system needs to be established based on the pallet frame.

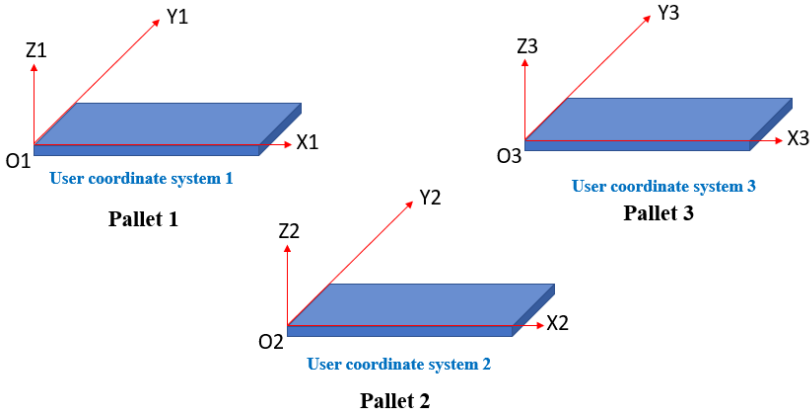


Figure 1.3 Create the user coordinate system for each pallet

Click [user coordinate system] in the "manual interface" of the teach pendant. Let the tool TCP reaches the origin of the coordinate system, a point on the x axis and a point on the y axis. The user coordinate system can be established by referencing the three points in turn. Please pay attention here, the created "point" needs to check the tool coordinate system. See Chapter 2, Section 2.3 of the manual for the detail.

Note:

- **The origin O of the user coordinate system is the "release point" of the pallet, the pose of each stack is offset and calculated based on it.**
- The establishment of user coordinate system refers to the right-hand spiral rule. When establishing pallet coordinates, the positive direction of Z is usually far away from the pallet. Therefore, it is necessary to consider which side is in the X and Y directions when establishing the user coordinate system.

1.1.2 Workpiece fixture control

In the process of palletizing, the clamping action of workpiece fixture can be controlled directly through output, or M relay controlled by PLC programming.

PLC ladder diagram programming is shown in Figure 1.4. Yxx represents the output port of the control fixture solenoid valve or cylinder.

```
---| |-------------------------------O
M100                                Yxx
```

Figure 1.4 PLC ladder diagram of fixture control

HUNAN Betrun Robot Co.,Ltd.

1.2 Create palletizing process package

Click [new] - [palletizing] to create a palletizing process package. There are four steps to set, including selecting the user coordinate system, setting the reference points, determining layout methods and setting layer information.

1.2.1 Coordinate system selection

The first step is to select the user coordinate system, as shown in Figure 1.5. The user coordinate system is established based on a pallet in palletizing operation.

Note: if there are multiple pallets in the palletizing operation, you need to create multiple user coordinate systems, and then configure the corresponding palletizing process package for each pallet.

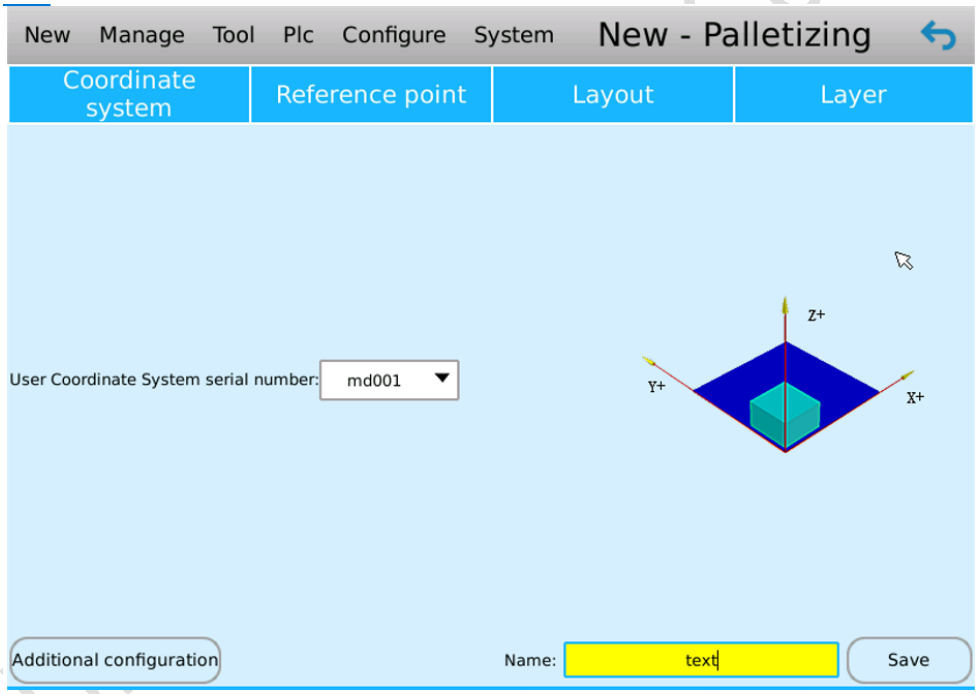


Figure 1.5 Select user coordinate system

1.2.2 Reference point setting

The second step is to set the motion track point when the robot places the stack at the origin O of the pallet user coordinate system through manual teaching, as shown in Figure 1.6. The "transition point" can be used according to the actual situation. If it is not "enabled", the stack grabbed from the assembly line will move directly to the "prepare to release point".

Note: the manual teaching is to place the stack at the origin O of the pallet user coordinate system.

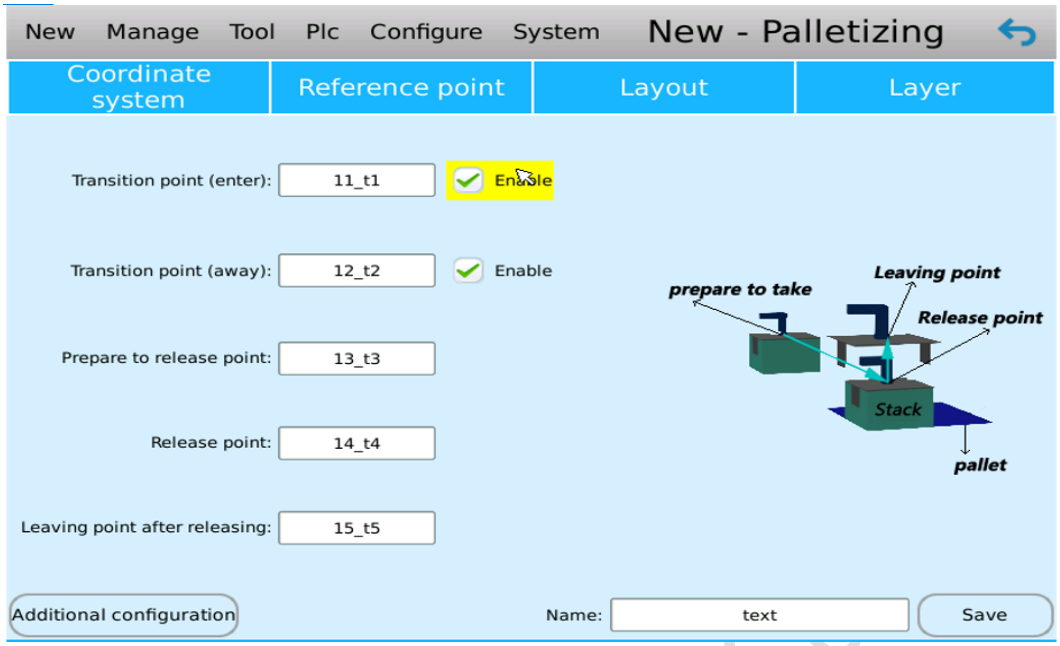


Figure 1.6 Reference point setting

1.2.3 Layout method setting

The third step is to set the layout method to be adopted for stacking on the pallet, as shown in Figure 1.7.

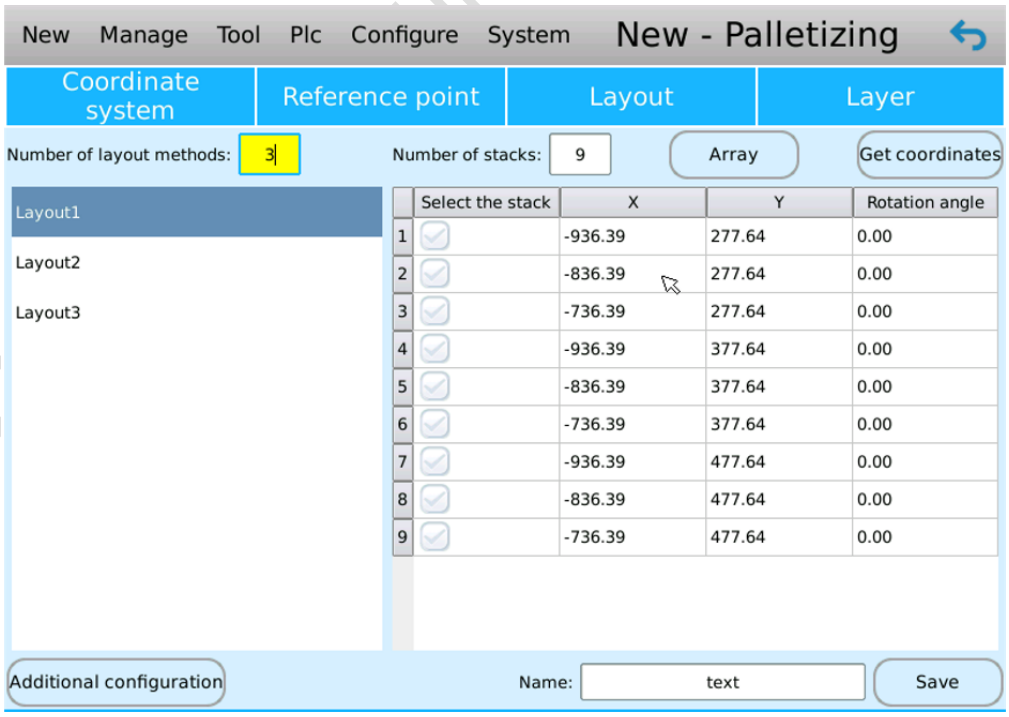


Figure 1.7 Set layout methods

First, enter the number of layout methods that will be used for stacking on the pallet. If there is only one type of layout, enter "1", and enter n for n types. After input, "layout 1", "layout 2" ...,

"layout n" will be automatically generated in the box.

Then, select a layout method and enter the number of stacks, that is, the number of stacks contained in the layer with the current layout method, and then you can teach manually to obtain the position coordinates of each stack.

If it is array layout, click the [array] button to enter the array setting interface, as shown in Figure 1.8. Select the running direction, determine the offset data of the first stack relative to the "release point", enter the row spacing, column spacing, etc., and finally click [Set].

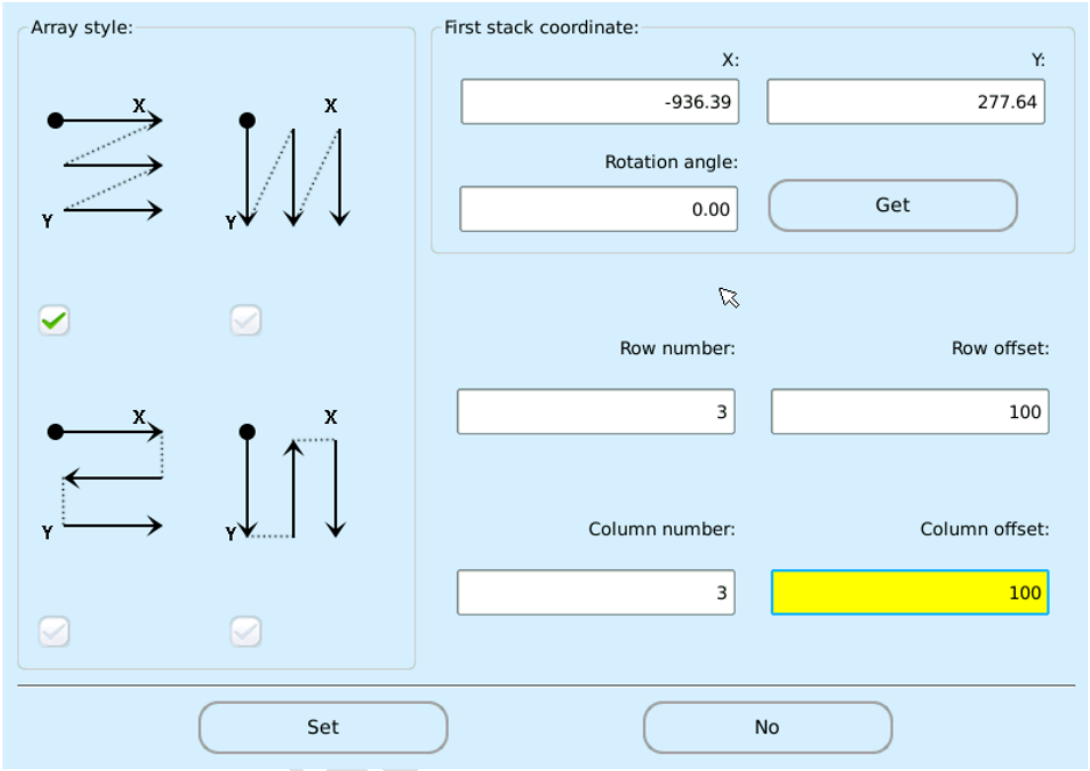


Figure 1.8 Array layout setting

1.2.4 Layer information setting

The fourth step is to set layer information, as shown in Figure 1.9.

First, enter the number of layers on the pallet. If only one layer of stacks is placed on the pallet, enter "1" and N for layer n. After input, "layer 1", "layer 2" ..., "layer n" will be automatically generated in the box.

Then, starting from the first layer at the bottom of the pallet, select "layout method" layer by layer, and set the layer height and transition point offset height.

Finally, click the [save] button to complete the creation of "palletizing process". If there are multiple pallets, you need to create multiple palletizing packages.

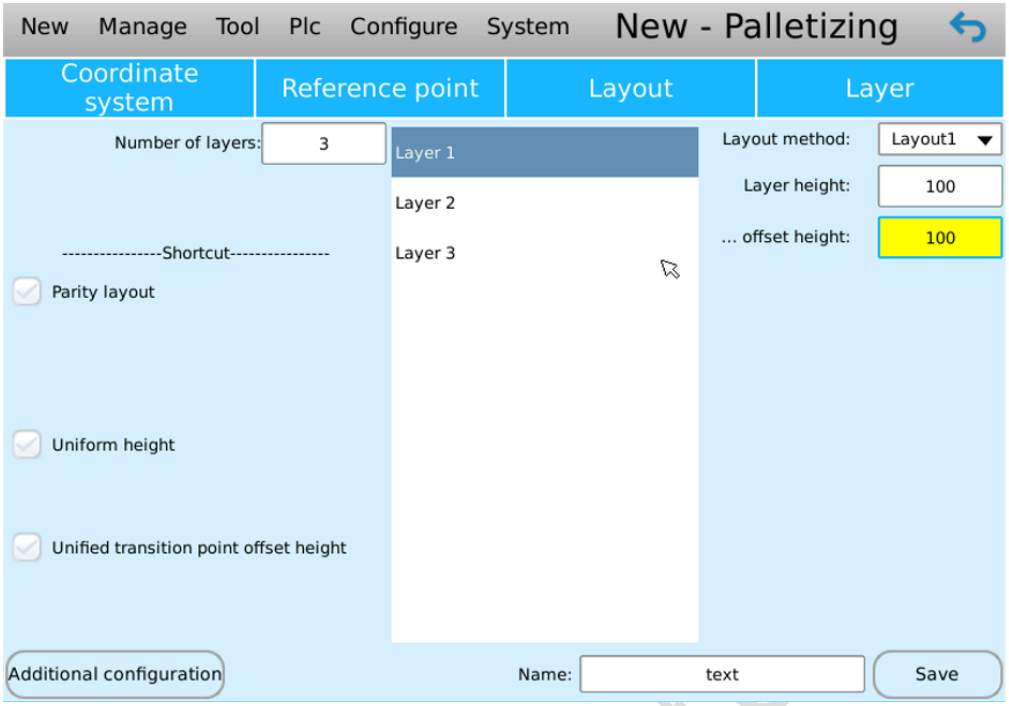


Figure 1.9 Set layer information

1.3 Generate palletizing program based on "template"

On the screen shown in Figure 1.9, click the [menu] button of the teach pendant and select [template], and the interface shown in Figure 1.10 appears.

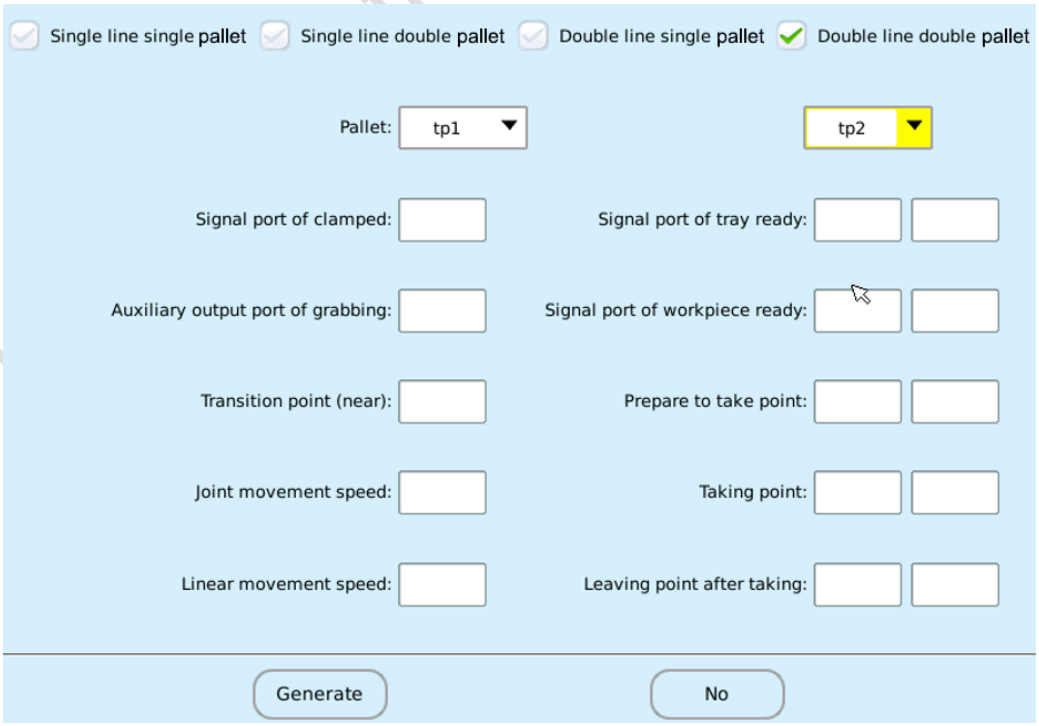


Figure 1.10 Palletizing program generation template

There are four types of palletizing operation provided by the system: single line single pallet, single line double pallet, double line single pallet and double line double pallet. The basic palletizing program can be automatically generated by template setting.

Take "double line and double pallet" as an example to explain the template setting steps.

As shown in Figure 1.10, the template is generated for the "double line and double pallet" palletizing program. First select the "palletizing process" packages of the created two pallets, and then set the tool signal ports, transition point, robot motion speed and the corresponding two assembly line taking points. The relevant instructions are shown in Table 1.2.

Table 1.2 Description of template parameters

Number	Term	Description
1	Pallet	Select the palletizing package created for this pallet.
2	Signal port of clamped	A sensor is installed on the tool to detect the clamping of the tool. Set the corresponding input port of the sensor.
3	Auxiliary output port of grabbing	Output port to control the tool clamping
4	transition point (near)	The teaching point on the track for the robot moving to the assembly line before taking the stack. Multiple assembly lines take this point as the transition point.
5	Joint movement speed	Running speed of corresponding motion curve in figure 1.2.
6	Linear movement speed	Running speed of corresponding motion curve in figure 1.2.
7	Signal port of tray ready	The input port used to know whether the pallets is ready. The order needs to correspond to the palletizing process package.
8	Signal port of workpiece ready	The input port used to know whether the workpiece is ready. The order needs to correspond to the assembly line.
9	Prepare to take point	Generally, it is located directly above the taking point, and it can also have a certain offset angle. The order needs to correspond to the assembly line.
10	Taking point	The teaching point for taking stack on the assembly line. The order needs to correspond to the assembly line.
11	Leaving point after taking	Generally, it is located directly above the taking point, and it can also have a certain offset angle. The order needs to correspond to the assembly line.

HUNAN Betrun Robot Co., Ltd

Click [generate] to automatically generate the basic program file "PalFrame". Click [manage] - [basic program], select the "PalFrame" program, click [save as] to modify the program name and save it.

Note: the name of the palletizing basic program generated through the template each time is "PalFrame". If it is not saved as a program with other names, each newly generated basic program will automatically overwrite the basic program generated by the previous template.

1.4 Modification of palletizing program

When creating the "palletizing process" program, enable the "transition point (enter)" and "transition point (away)", and then the basic program generated through the template is shown in Figure 1.11, which will list the instruction segment of the robot's running track curve between the start palletizing process instruction and the end palletizing process instruction.

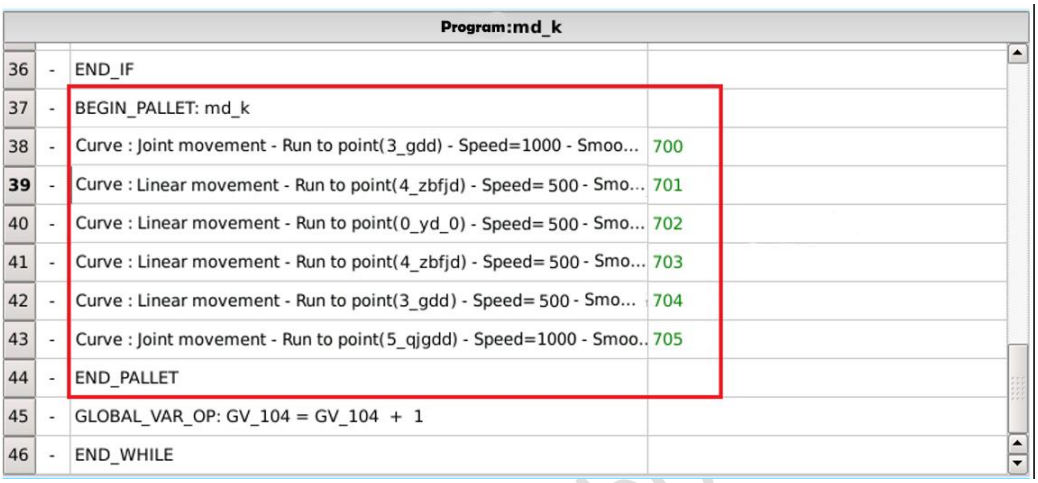


Figure 1.11 Palletizing basic program

1.4.1 Global variable definition

On the basis of the palletizing basic program generated through the template, in order to quickly realize the dynamic modification of the motion track according to the operation requirements, the system uses a group of global variables as the flag of whether the six curves in the palletizing process are skipped, and provides global variables "gv218" and "gv219", which respectively represent "current palletizing layers" and " current serial number to be palletized", which can be used as the logic judgment of the program.

Table 1.3 Description of global variables of palletizing program (1)

Number	Global variable	Description
1	GV700-GV705	Palletizing motion curve skip flag（0-don't skip, 1-skip）
7	GV218	Current number of layers in palletizing process
8	GV219	Serial number of the position to be stacked in the current layer in palletizing process
9	GV100-GV109	Number of stacks already placed in 10 pallets in palletizing process

The corresponding global variable number is given on the right side of the instruction line in Figure 1.11. See table 1.3 for details. According to the flag definition, for example, in the global variable "gv700-gv705", if "gv700 = gv702 = gv703 = gv704 = gv705 = 0, gv701 = 1" is set, the curve running track will change: when the curve runs to the "transition point (enter)" position, it will no longer run to the "ready to release point" position, but directly run from the "transition point (enter)" position to the "release point" position.

1.4.2 Modify reference points

The palletizing basic program can also modify the reference points according to the five groups of global variables provided by the system. The description of global variables is shown in table 1.4.

Table 1.4 Description of global variables of palletizing program (2)

Number	Global variable	Description
1	GV706	Special, palletizing "transition point (enter)" modification mark（0 - do not modify, 1 - modify）
	GV707	Special, modified palletizing "transition point (enter)" ID
2	GV708	Special, palletizing "prepare to release point" modification mark（0 - do not modify, 1 - modify）
	GV709	Special, modified palletizing "prepare to release point" ID
3	GV710	Special, palletizing "release point" modification mark（0 - do not modify, 1 - modify）
	GV711	Special, modified palletizing "release point" ID
4	GV712	Special, palletizing "leaving point after releasing" modification mark（0 - do not modify, 1 - modify）
	GV713	Dedicated, modified palletizing "leaving point after releasing" ID
5	GV714	Special, palletizing "transition point (away)" modification mark（0 - do not modify, 1 - modify）
	GV715	Special, modified palletizing "transition point (away)" ID

Application example: in the palletizing program shown in Figure 1.11, reference points are: "transition point (enter)" and "transition point (away)" (the ID number is 3); "prepare to release point" and "leaving point after releasing" (the ID number is 4); "release point" (the ID number is 0).

When the global variables gv706 = gv708 = gv710 = gv712 = gv714 = 1, gv707 = gv715 = 26, gv709 = gv713 = 27, gv711 = 28 are set, as shown in Figure 1.12, it means to modify the five " reference points": "transition point (enter)" and "transition point (away)" (the ID number is 26); "prepare to take point" and "leaving point after releasing" (the ID number is 27); "release point" (the ID number is 28).

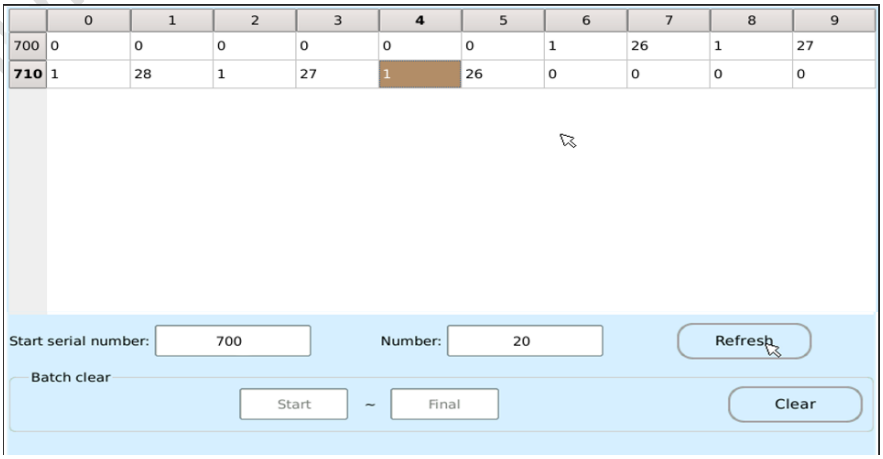


Figure 1.12 Set global variable modification reference points

HUNAN Betrun Robot Co., Ltd.

1.5 Offset setting

The system provides corresponding offset settings for the pallet offset in the process of palletizing and the need to offset the transition point according to the stacking position.

1.5.1 Pallet offset setting

As shown in Figure 1.13, click [additional configuration] in the process package interface, and the system provides the "enable external offset" function of the user coordinate system. Check "enable external offset". The system uses three global variables gv200, gv201 and gv202 to correspond to the X-axis offset, Y-axis offset and rotation of the pallet user coordinate system respectively. The offset of the pallet user coordinate system can be realized by assigning values to these three global variables in the program.

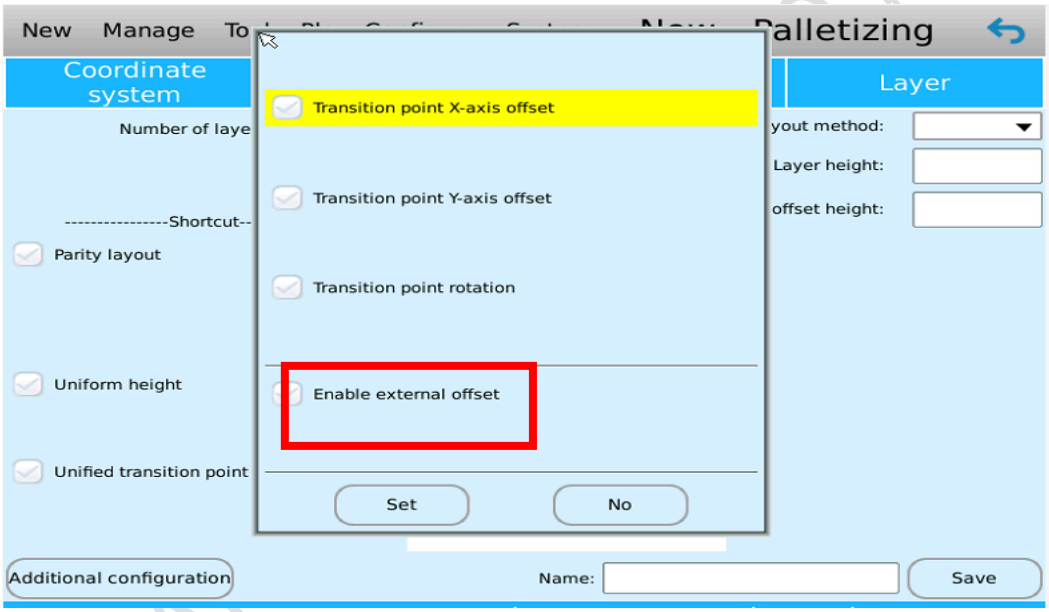


Figure 1.13 Pallet user coordinate system enable external offset setting

1.5.2 Transition point offset settings

In the palletizing process package, by default, the transition point is a fixed point when the robot puts stacks on the pallet in turn. In some special applications, such as placing knifes, the "transition point" needs to be offset with the position of the "prepare to release point". At this time, check the corresponding transition point X-axis offset, transition point Y-axis offset and transition point rotation in the interface of figure 1.13.

2 Welding

Welding, also known as fusion welding and fusion welding, is a manufacturing process and technology of joining metals or other thermoplastic materials by heating, high temperature or high pressure.

Using industrial robot to realize welding needs to complete two parts of teaching and programming design, one is the teaching programming of robot motion trajectory, and the other is the setting of robot welding operation. In the second part, the setting of robot welding operation is to obtain good welding quality, and set the welding conditions, including welder model, welding sequence, welding control signal, wire feeding speed, welding current and voltage control and other parameters.

In betrun system, the work of the second part is encapsulated into a "welding" process package, which can be quickly generated by setting it in the corresponding interface and called in the basic program.

2.1 System composition

Welding robot system is generally composed of industrial robot, welding gun, wire feeder, welding machine and isolator.

The connection between the control system and the welding machine includes: Io input, IO output, and analog output. The details are shown in Figure 2.1.

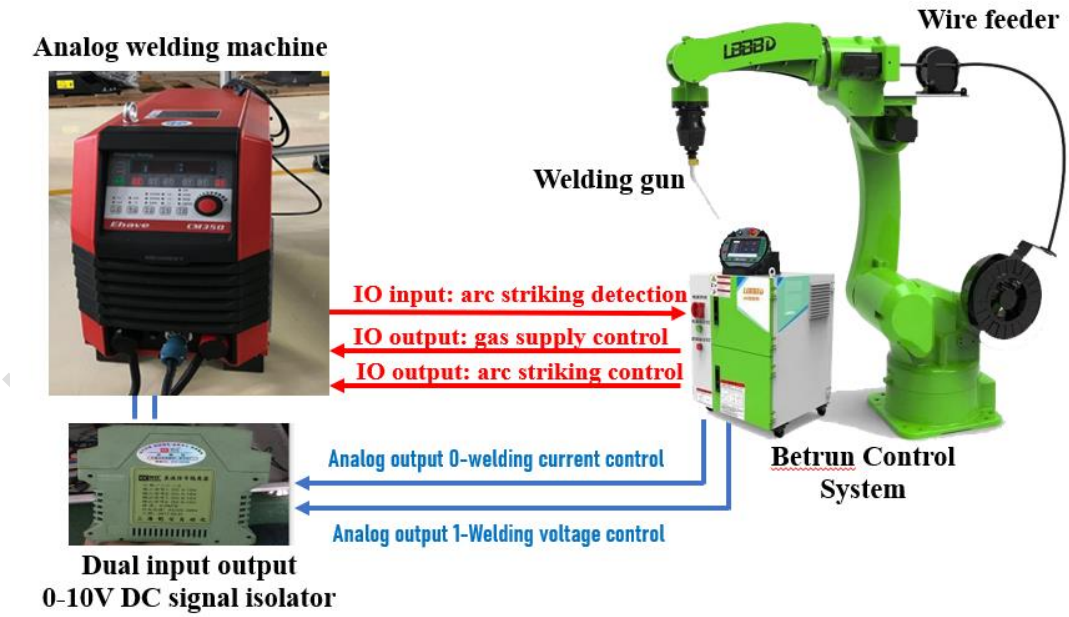


Figure 2.1 Connection between the control system and the welding machine

2.2 Create welding process package

Click [new] - [process] - [welding] to create a welding process package. There are five steps to

set, including welding machine configuration, time, IO port, basic parameters, and calibration.

2.2.1 Welding machine configuration

Step 1: select the welding machine. The system provides several welding machines brands. You can choose one and set the welding mode, as shown in Figure 2.2.

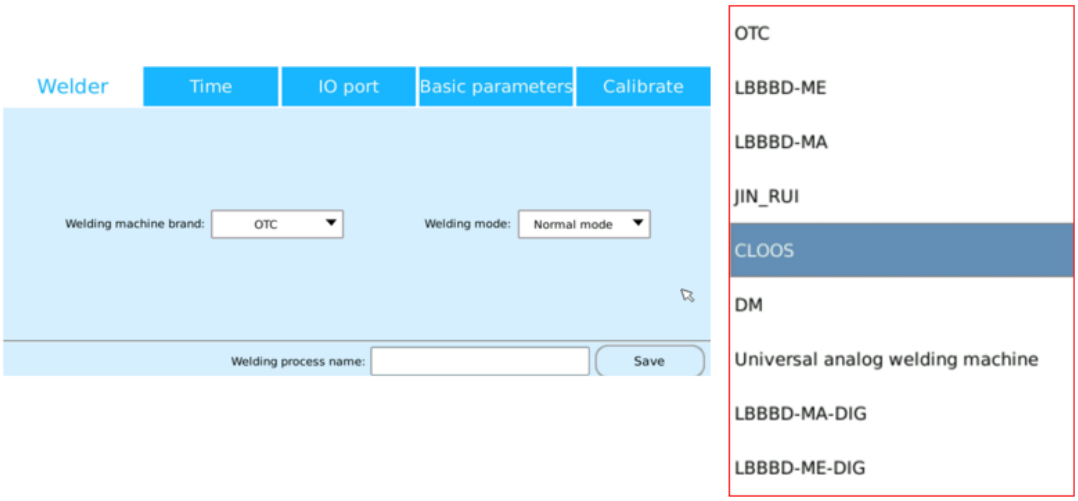


图 2.2 Select the welding machine

2.2.2 Time setting

Step 2: set the time parameters of voltage control in each operation step in the welding process, as shown in Figure 2.3. The time parameter description and recommended values are shown in Table 2.1. Figure 2.4 shows the sequence diagram of welding voltage control.

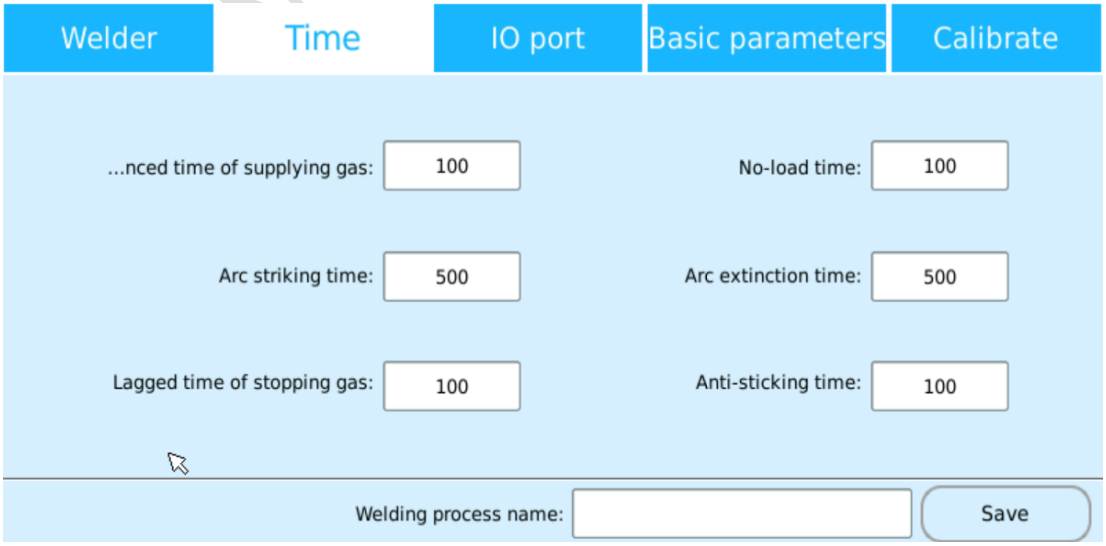


Figure 2.3 Set welding time parameters

Table 2.1 Description of welding process time parameters

Number	Time parameter	Description	Recommended value (ms)
1	Advanced time of supplying gas	Advanced time of supplying gas before welding arcing.	100
2	No-load time	The time between the end of air supply in advance and the beginning of arcing.	100
3	Arc striking time	The time between sending the arcing signal and receiving the successful arcing.	Preliminary definition: 500 If the arc striking is too fast and unsuccessful, the time can be appropriately increased. If there is accumulation in the welding, the time can be appropriately reduced.
4	Arc extinction time	The time between the end of welding and the actual stop of welding.	Preliminary definition: 500 If the arc suppression is too fast and unsuccessful, the time can be appropriately increased. If there is accumulation in the welding, the time can be appropriately reduced.
5	Anti-sticking time	The time between the welding end signal and the stop of arc stopping voltage.	100
6	Lagged time of stopping gas	Delay time of closing air supply when all welding work is completed.	100

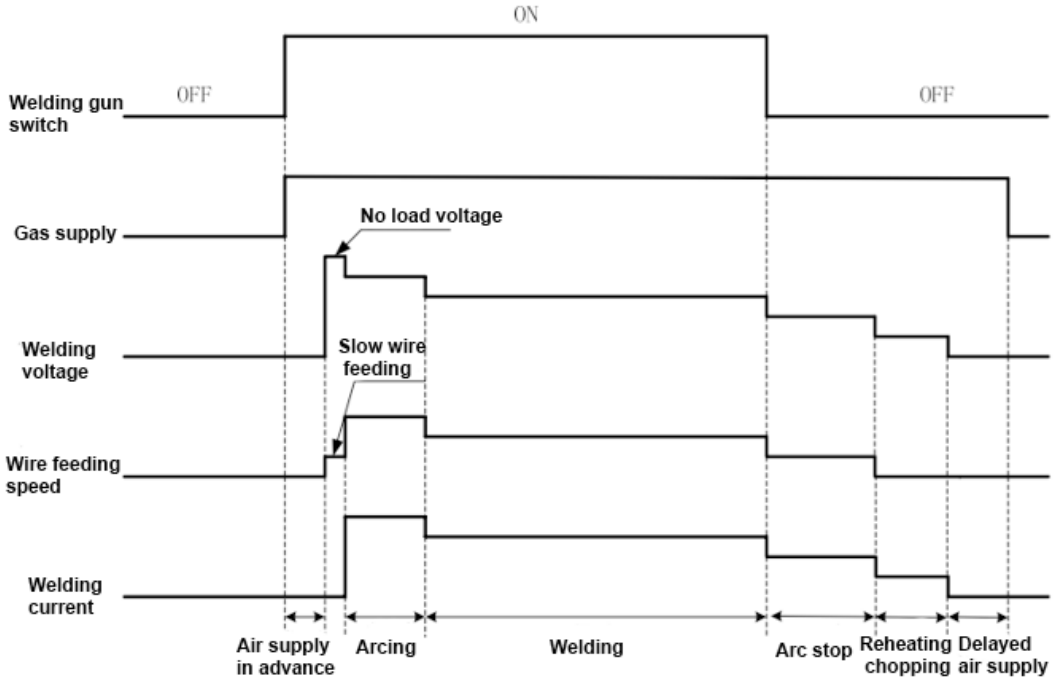


Figure 2.4 Sequence diagram of welding voltage control

Betrun Robot Co., Ltd.

2.2.3 IO port configuration

Step 3: find the output port of are striking control signal, output port of gas supply control signal and input port of arc striking detection signal according to the welding machine manual, and allocate them respectively according to their respective I/O allocation table, as shown in Figure 2.5.

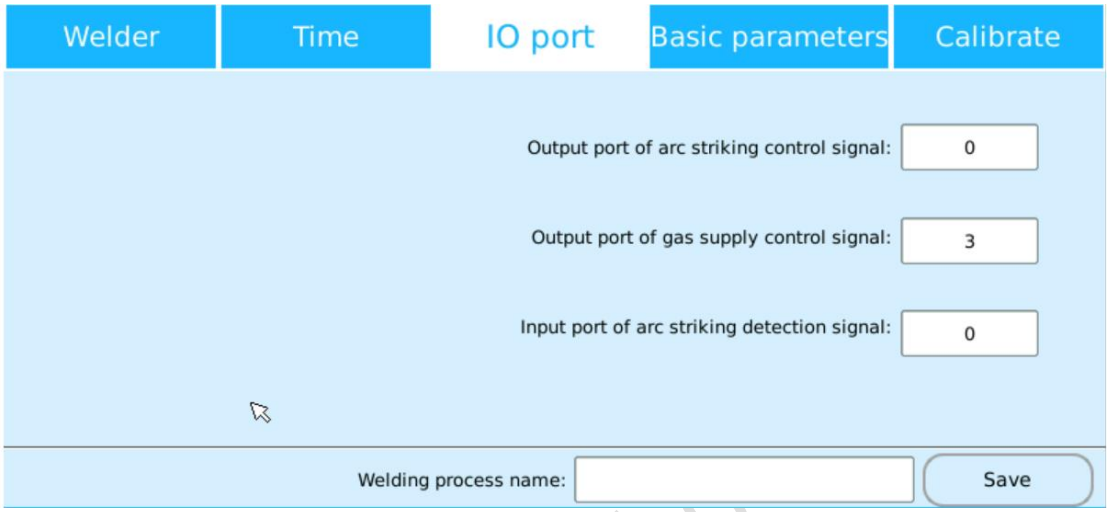


Figure 2.5 Welding IO port setting

2.2.4 Basic parameter setting

Fourth step: set the control voltage (unit: V) and current (unit: A) required for each operation step in the welding process according to the process requirements and the welding conditions (plate thickness, welding position, welding speed, material and other parameters). The reference values are shown in Figure 2.6.

- Welding voltage: provide welding wire melting energy. The higher the voltage, the faster the welding wire melting speed;
- Welding current: adjust the wire feeding speed to balance with the melting speed.

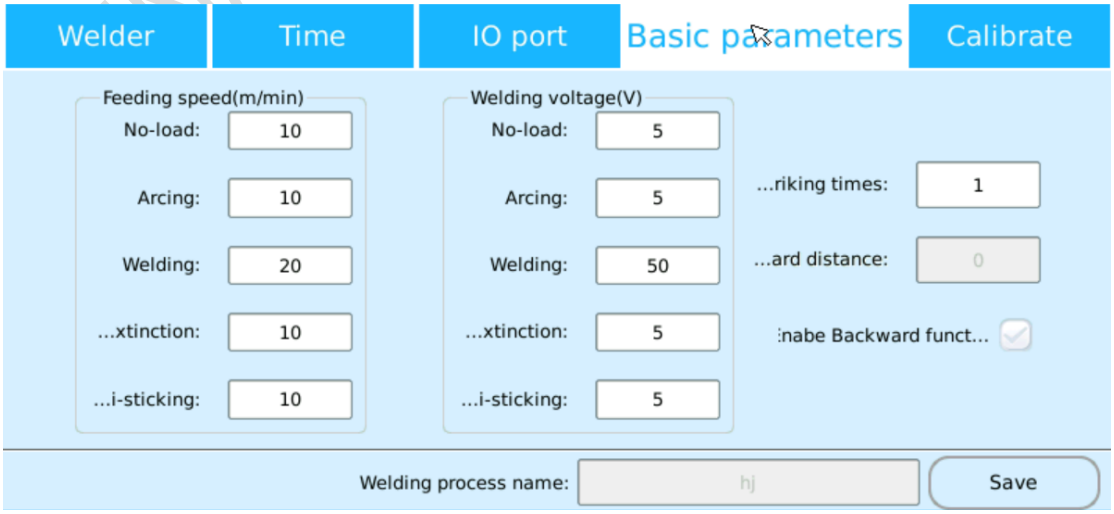


Figure 2.6 Set welding voltage and current parameters

- If selecting “Enable Backward function”, you need set the Backward distance. When the welding process is stopped and started again, it will be back for a certain distance before continuing welding.

2.2.5 Calibration

For the analog welding machine, in order to realize the current and voltage analog quantity control, as shown in Figure 2.7, the value of the analog quantity output port needs to be corresponding to the actual required current and voltage value, that is, calibration.

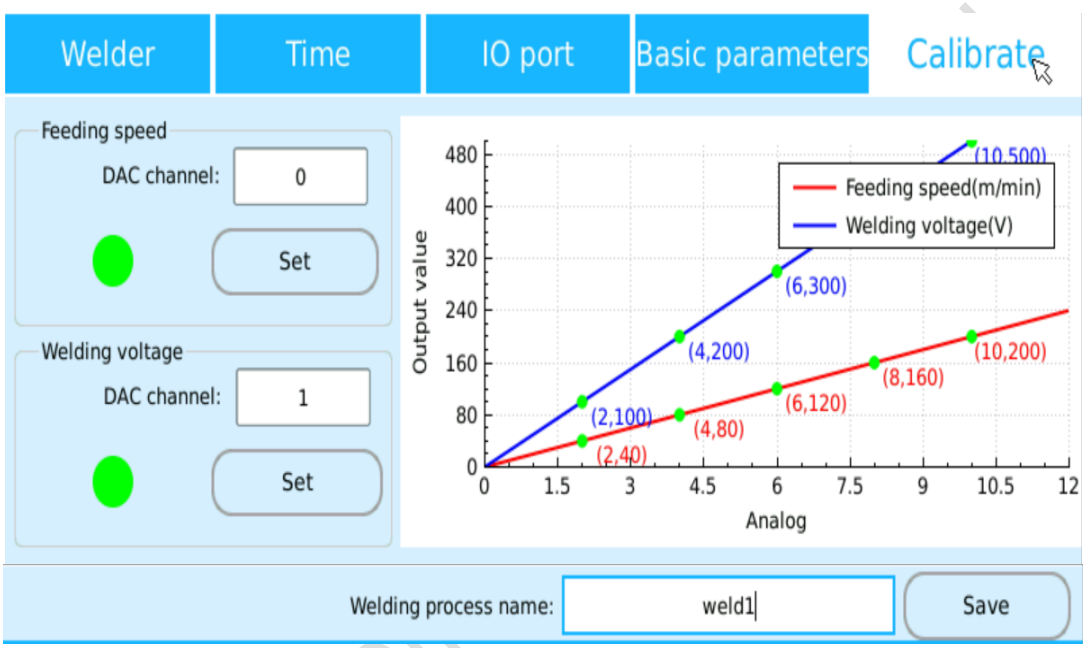


Figure 2.7 Current and voltage calibration

First, input the current output port "0" (if the display is not current but voltage, it can be adjusted with channel 1), click "calibration" to enter the "current verification" screen. Input the corresponding current value when the current is "1" and the corresponding current value when the current is "6". Click "OK" to finish.

Then input the voltage output port "1" (if the display is not current but voltage, it can be adjusted with channel 0), click "calibration" to enter the "voltage verification" screen. Input the corresponding voltage value when the voltage is "1" and the corresponding voltage value when the voltage is "6". Click "OK" to finish.

Note: for some analog welding machines, such as DM welding machine, the calibration of current and voltage is not completely linear and needs to be divided into multiple sections.

2.3 Welding program design

When writing the program, write different welding points, lines, or arcs in turn. In the process of manual teaching, the angle adjustment of the welding gun shall rotate each axis in the way of joint movement to adjust the angle. After the angle is appropriate, it shall be moved to the required position in the way of base coordinates. For each welding, "start welding" and "stop welding" shall

appear in pairs, and different welding processes can be used in different curve segments.

	Program:welding
1	Curve - Joint movement : Run to point(195_weldAuto0) ;Speed = 1000 ;Smoothing ratio = 0
2	Curve - Linear movement : Run to point(196_a) ;Speed = 100 ;Smoothing ratio = 0
3	Begin welding: weld1
4	Curve - Linear movement : Run to point(197_b) ;Speed = 5 ;Smoothing ratio = 0
5	End welding

Figure 2.8 Welding process program

Note:

(1) "Begin welding" and "End welding" should appear in pairs.

(2) The motion type of welding section program must be linear, circular, circular motion or delay of spot welding, and cannot be joint motion.

(3) For different welding materials, corresponding gas shall be selected; Welding wires of different specifications shall be selected according to the process requirements; According to the selected welding wire, check whether the wire feeding wheel and electric nozzle match.

(4) Factors affecting the welding effect: gas type and flow, material of welding wire and conductive nozzle, welding angle, welding speed, welding control current and voltage.

Ltd.

3 Spraying

Using industrial robot to realize spraying needs to complete two parts of teaching and programming design, one is the teaching programming of robot motion trajectory, and the other is the setting of robot spraying control parameters. In this system, the spraying process package can be quickly created and the basic program can be directly generated only by setting in the corresponding two interfaces.

3.1 Create spraying process package

Click [new] - [spraying] to create a spraying process package. There are two steps: motion parameter setting and control parameter setting.

3.1.1 Motion parameter setting

The system provides six "spraying modes": "straight line", "plane", "plane inch", "arc line", "arc surface" and "arc surface inching". According to the "spraying mode", the motion parameters will be different.

"Straight line" spraying mode

As shown in Figure 3.1, it is the motion parameter setting interface of "straight line" spraying mode, and the parameter description is shown in Table 3.1.

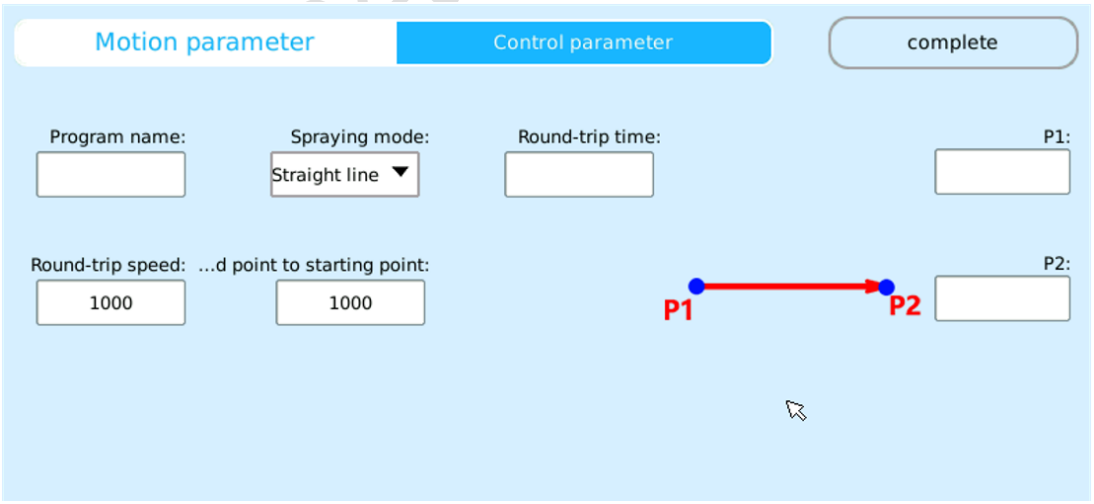


Figure 3.1 "Straight line" spraying mode motion parameter setting interface

Table 3.1 Description of motion parameters of "straight line" spraying mode

Number	Parameter	Description
1	Round-trip time	The number of repeated spraying times. If there is no repeated spraying, enter "0".

Betrun Robot Co., Ltd.

2	P1、P2	The starting point and ending point of linear spraying in the figure. (You need to "check" the corresponding tool coordinate system when creating points)
3	Round-trip speed	The spraying speed from the starting point to the end point belongs to the "linear" movement type, and the value range is "1-2000" (mm/s).
4	End point to starting point	The running speed from the end point (P2) to the starting point (P1) of the spraying straight line belongs to the "straight line" motion type, and the value range is "1-2000" (mm/s).

"Plane" spraying mode

As shown in Figure 3.2, it is the motion parameter setting interface of "plane" spraying mode, and the parameter description is shown in Table 3.2.

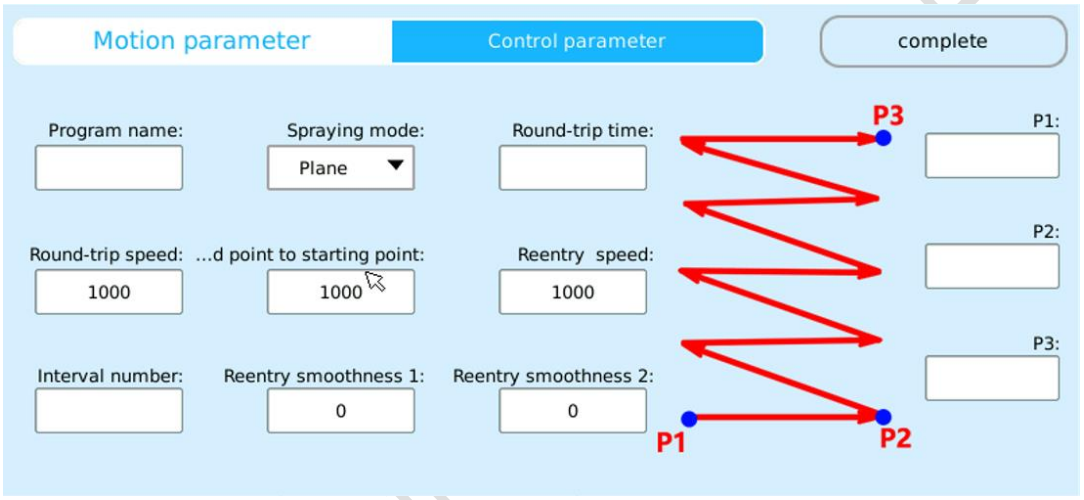


Figure 3.2 "Plane" spraying mode motion parameter setting interface

Table 3.2 Description of motion parameters of "plane" spraying mode

Number	Parameter	Description
1	Round-trip time	The number of repeated spraying times. The value range is "1-10". If there is no repeated spraying, enter "0".
2	Interval number	The turn back times between "first straight line" and "last straight line", with the value range of "1-80".
3	P1、P2、P3	Three points for "plane" spraying which are shown in the figure. (The created point needs to "check" the corresponding tool coordinate system)
4	Round-trip speed	The spraying speed of P1 moving towards P2, with the value range of "1-2000" (mm/s).
5	Reentry speed	The spraying speed of turning back from P2, with the value range of "1-2000" (mm/s).
6	End point to starting point	The running speed from the end point P3 to the starting point P1, with the value range of "1-2000" (mm/s).
7	Reentry smoothness - 1, - 2	Smoothness of "reentry" curve, with the value range of "0-100".

"Plane inch" spraying mode

As shown in Figure 3.3, it is the motion parameter setting interface of "plane inch" spraying mode, and the parameter description is shown in Table 3.3.

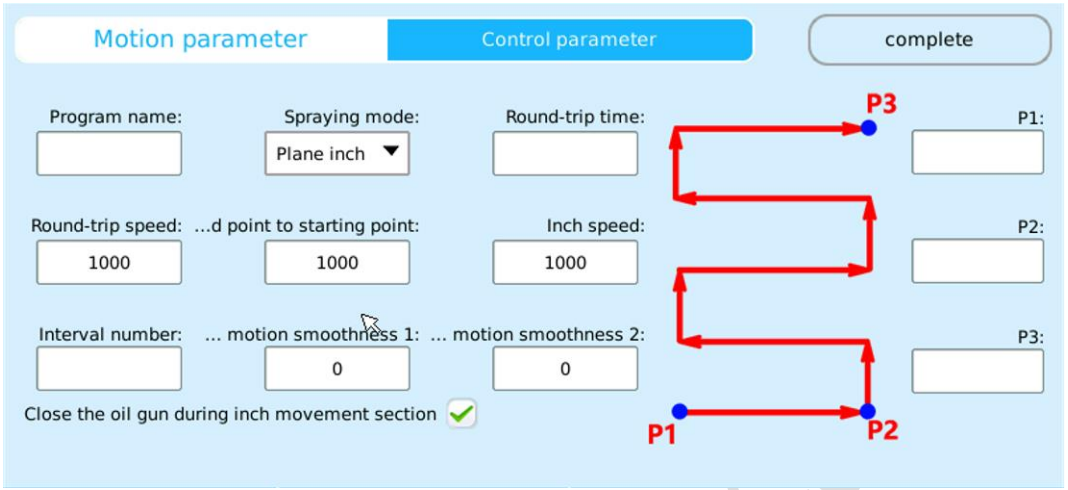


Figure 3.3 "Plane inch" spraying mode motion parameter setting interface

Table 3.3 Description of motion parameters of "plane inch" spraying mode

Number	Parameter	Description
1	Round-trip time	The number of repeated spraying times. The value range is "1-10". If there is no repeated spraying, enter "0".
2	Interval number	The turn back times between "first straight line" and "last straight line", with the value range of "1-80".
3	P1、P2、P3	Three points for "plane inch" spraying which are shown in the figure. (The created point needs to "check" the corresponding tool coordinate system)
4	Round-trip speed	The back and forth spraying speed in P1-P2 direction, with the value range of "1-2000" (mm/s).
5	Inch speed	The operating speed of inch movement section perpendicular to P1-P2, with the value range of "1-2000" (mm/s).
6	End point to starting point	The running speed from the end point P3 to the starting point P1, with the value range of "1-2000" (mm/s).
7	Inch motion smoothness -1, -2	Smoothness of inch motion curve, value range "0-100".
8	Close the oil gun during inch movement section	If "checked", the oil gun will be closed when the inching section is running and will be automatically opened after passing through the inching section.

"Arc line" spraying mode

As shown in Figure 3.4, it is the motion parameter setting interface of "arc line" spraying mode, and the parameter description is shown in Table 3.4.

HUNAN BAIJIN Robot Co

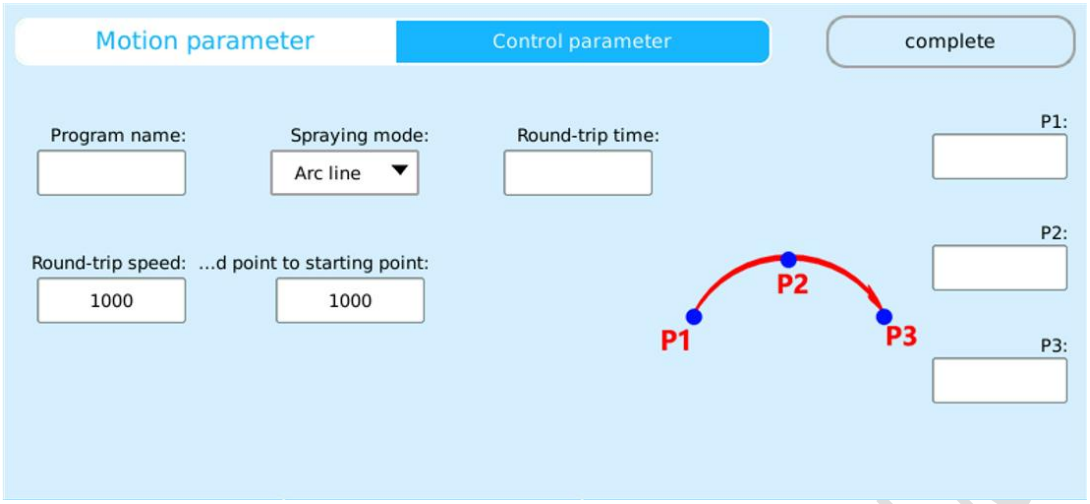


Figure 3.4 "Arc line" spraying mode motion parameter setting interface

Table 3.4 Description of motion parameters of "arc line" spraying mode

Number	Parameter	Description
1	Round-trip time	The number of repeated spraying times. The value range is "1-10". If there is no repeated spraying, enter "0".
2	P1、P2、P3	Three points for "arc line" spraying which are shown in the figure. (The created point needs to "check" the corresponding tool coordinate system)
3	Round-trip speed	The spraying speed from the starting point to the end point, with the value range of "1-2000" (mm / s).
4	End point to starting point	The running speed from the end point P3 to the starting point P1, with the value range of "1-2000" (mm/s).

"Arc surface" spraying mode

As shown in Figure 3.5, it is the motion parameter setting interface of "arc surface" spraying mode, and the parameter description is shown in Table 3.5.

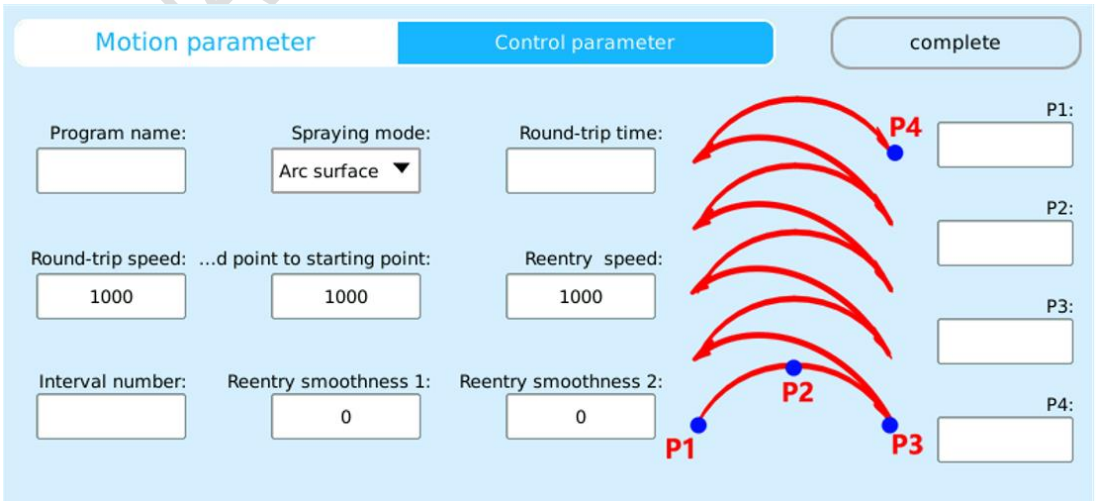


Figure 3.5 "Arc surface" spraying mode motion parameter setting interface

Table 3.5 Description of motion parameters of "arc surface" spraying mode

Number	Parameter	Description
1	Round-trip time	The number of repeated spraying times. The value range is "1-10". If there is no repeated spraying, enter "0".
2	Interval number	The turn back times between "first arc" and "last arc", with the value range of "1-80".
3	P1、P2、P3、P4	Four points for "arc surface" spraying which are shown in the figure. (The created point needs to "check" the corresponding tool coordinate system)
4	Round-trip speed	The spraying speed of P1 moving towards P3, with the value range of "1-2000" (mm/s).
5	Reentry speed	The spraying speed of turning back from P3, with the value range of "1-2000" (mm/s).
6	End point to starting point	The running speed from the end point P4 to the starting point P1, with the value range of "1-2000" (mm/s).
7	Reentry smoothness - 1, - 2	Smoothness of "reentry" curve, with the value range of "0-100".

"Arc inch" spraying mode

As shown in Figure 3.6, it is the motion parameter setting interface of "arc inch" spraying mode, and the parameter description is shown in Table 3.6.

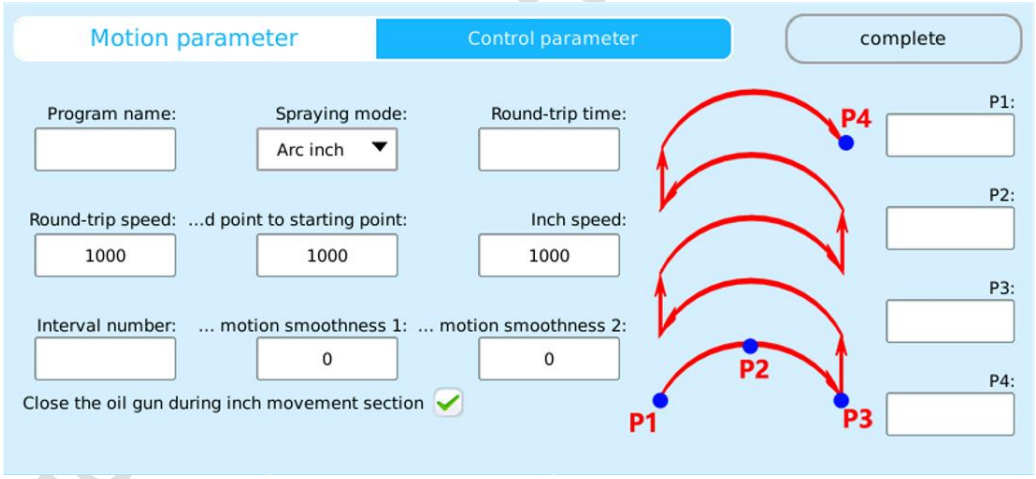


Figure 3.6 "Arc inch" spraying mode motion parameter setting interface

Table 3.6 Description of motion parameters of "arc inch" spraying mode

Number	Parameter	Description
1	Round-trip time	The number of repeated spraying times. The value range is "1-10". If there is no repeated spraying, enter "0".
2	Interval number	The turn back times between "first arc" and "last arc", with the value range of "1-80".
3	P1、P2、P3、P4	Four points for "arc inch" spraying which are shown in the figure. (The created point needs to "check" the corresponding tool coordinate system)
4	Round-trip speed	The back and forth spraying speed in P1-P3 direction, with the value range of

		"1-2000" (mm/s).
5	Inch speed	The operating speed of inch movement section in P3 to P4 direction, with the value range of "1-2000" (mm/s).
6	End point to starting point	The running speed from the end point P4 to the starting point P1, with the value range of "1-2000" (mm/s).
7	Inch motion smoothness -1, -2	Smoothness of inch motion curve, value range "0-100".
8	Close the oil gun during inch movement section	If "checked", the oil gun will be closed when the inching section is running and will be automatically opened after passing through the inching section.

3.1.2 Control parameter setting

The "control parameters" of the six spraying modes in the "spraying process" of the system are the same, as shown in Figure 3.7.

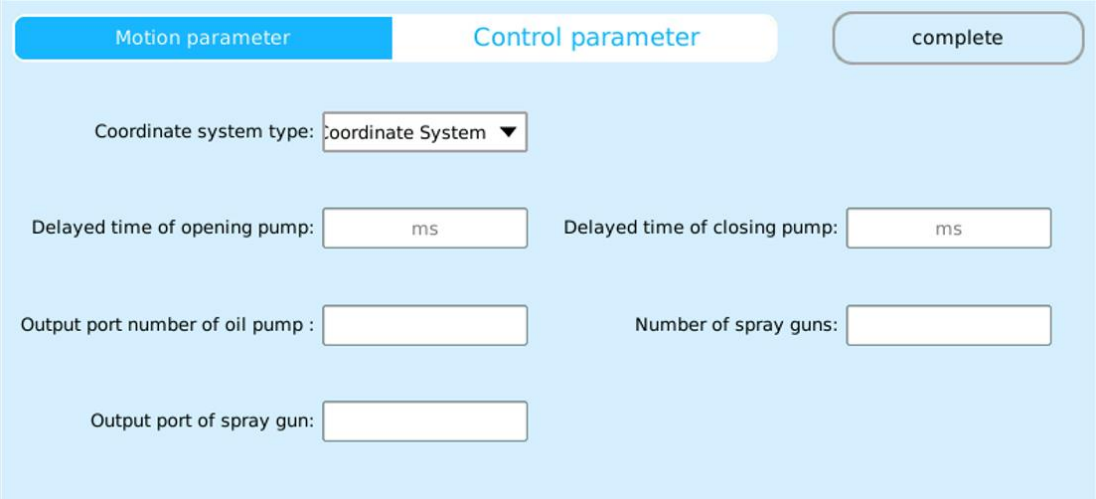


Figure 3.7 Control parameter setting interface of spraying process package

Table 3.7 Description of control parameters of spraying process package

Number	Parameter	Description
1	Coordinate system type	If a user coordinate system is created for the spray plane, select the corresponding user coordinate system, which is "base coordinate system" by default.
2	Delayed time of opening pump	The unit is "milliseconds" (ms).
3	Delayed time of closing pump	The unit is "milliseconds" (ms).
4	Output port number of oil pump	Use the I/O port to output "high/low level" to set the "on/off" of "oil pump".
5	Number of spray guns	The range is "1-4", that is, up to "4" oil guns can be controlled at the same time.

6	Output port of spray guns	Set with "number of spray guns". When there are multiple "spray guns", the serial numbers of "spray gun output port" are connected with "-".

3.2 Generate spraying basic program

After setting the "motion parameters" and "control parameters", enter the name of the process package and click the [finish] button. A [generate] button will appear. Click the [generate] button - [OK] generate - automatically generate and save the "spraying" basic program.

Part IV Modbus Communication

1 Modbus communication hardware connection

Betrun system provides a communication mode based on Modbus protocol, which can make PC, PLC, single chip microcomputer system and visual sensor system as the master station to remotely operate the global variables, input / output and other resources in Betrun control system (slave station), so as to realize the remote control of the robot; Betrun control system can also be used as the master station to access slave stations such as sensors through Modbus.

Modbus is a bus standard used for serial link communication (RS232/RS485) and TCP/IP. As shown in Figure 1, the Betrun main controller provides RS232/RS485 port specially used for Modbus communication and network port for TCP/IP mode.



Figure 1 Modbus communication connection port

The definition of RS232 / RS485 port signal is shown in Table 1.

Table 1 Definition of RS485 and RS232 communication interface signals

PIN	Signal	Definition	PIN	Signal	Definition
1	+5V	+5V Output	6	I2C2_SDA	Bidirectional data line
2	I2C2_SCL	Clock line	7	RS232_TX	RS232 Signal transmission port
3	RS485B	RS485 Signal B	8	RS232_RX	RS232 Signal receiving port
4	RS485A	RS485 Signal A	9	GND	GND 0V
5	GND	GND 0V			

2 Modbus communication protocol

Modbus is the application layer message transmission protocol on the 7th layer of OSI model. By providing function code provisions, it realizes remote read-write access to various resources of the slave station in the form of master station request and slave station response.

2.1 Frame format

The frame format of Modbus communication is shown in Figure 2 according to different transmission modes. The function code plus data part is called data unit (PDU).

Note:

(1) Modbus protocol RTU mode requires that the beginning and end of each frame are separated by at least 3.5 characters. Each character interval is the number of character bits/baud rate, with 1 start bit, 8 data bits, 1 stop bit and 10 data bits without parity bit. Taking the baud rate of 9600b/s as an example, the character interval is 1.14ms, that is, the beginning and end of each frame are accompanied by a time interval of at least 3.5*1.14=4.01ms. In the byte stream, take 1.5 character time as the inter frame interval. At the above communication rate, 1.5*1.14=1.71ms, that is, when an interval of 1.71ms appears in the character stream, it may mean the end of a frame of data. However, it must be satisfied that no more characters will be received within a subsequent period of time (2 characters), otherwise the end of the frame is considered as an error frame.

(2) The contents of Modbus RTU and ASCII are exactly the same. Different ASCII methods use ":" to identify the beginning of the frame and "CR IF" to identify the end of the frame. LRC is used for verification, and the content of one byte in RTU frame is replaced with two ASCII characters. For example, in the RTU mode, the device address 01 has only one byte, which is converted to the string "01" (hexadecimal 30 31) in the ASCII mode

(3) The device identification and function code in Modbus TCP are the same as those in Modbus RTU. It can be considered that a header is added to the front of Modbus RTU message and the tail of CRC verification is removed.

Modbus RTU message format

Beginning	Device address	Function code	Data	CRC verification	End
At least 3.5 characters	8 bits	8 bits	n*8bits	16 bits	At least 3.5 characters

Modbus ASCII message format

Beginning	Device address	Function code	Data	LRC verification	End
1 character	2 characters	2 characters	n characters	2 characters	2 characters

Modbus TCP message format

Interactive id	Protocol id	Message length	Device address	Function code	Data
2 bytes	2 bytes	2 bytes	1 byte	1 byte	n bytes

Figure 2 MODBUS frame format

2.2 Communication process

The Modbus communication process is shown in Figure 3 and Figure 4. The master station

HENAN BeTrun Robot Co., Ltd.

requests and the slave station returns the response. When there is an error, the slave station returns the abnormal response.

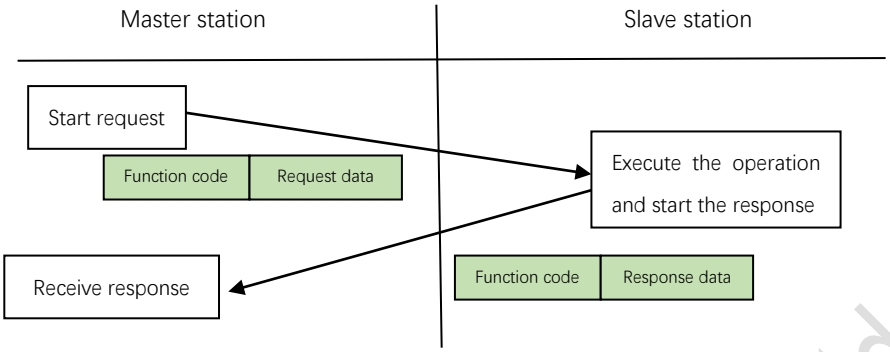


Figure 3 Modbus communication process (no error)

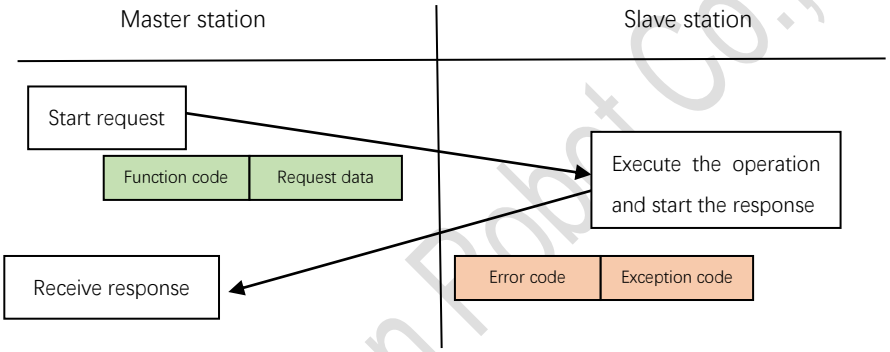


Figure 4 Modbus communication process (abnormal response)

2.3 Data unit (PDU) description

Modbus protocol defines three data units (PDUs). They are:

1) Modbus request PDU
2) MODBUS response PDU
3) MODBUS abnormal response PDU

Define the Modbus request PDU as:

mb_req_pdu = {function_code, request_data}

Among them:

function_code - [1 byte] MODBUS function code.

request_data - [n bytes]. This field is related to the function code and usually includes information such as variable reference, variable, data offset, sub function code, etc.

Define MODBUS response PDU as:

mb_rsp_pdu = { function_code, response_ data}

Among them:

function_code - [1 byte] MODBUS function code.

HUNAN Betrun Robot Co., Ltd.

response_data - [n bytes]. This field is related to the function code and usually includes information such as variable reference, variable, data offset, sub function code, etc.

Define the Modbus abnormal response PDU as:
mb_excep_rsp_pdu = { function_code, request_data}
Among them:
function_code - [1 byte] MODBUS function code + 0x80.
exception_code - [1 byte].

3 Modbus function code and data type

3.1 Function code definition

In the definition of Modbus communication protocol, for different data types of "slave station", MODBUS adopts different function codes to realize remote read-write access, as shown in Table 2.

Table 2 MODBUS data types and access function codes

Number	Data type	Object type	Access type	Data access	Function code	(HEX)
1	Input discrete quantity	Single bit	read-only	Read discrete input	02	02
2	Coil	Single bit	Read and write	Read coil	01	01
				Write single coil	05	05
				Write multiple coils	15	0F
3	Input register	16 bits	read-only	Read input register	04	04
4	Hold register	16 bits	Read and write	Read multiple registers	03	03
				Write single register	06	06
				Write multiple registers	16	10
				Read / write multiple registers	23	17
				Mask write register	22	16

3.2 Data type mapping for Betrun system as a slave

When the Betrun control system is used as a "slave", the definition of mapping the data type to the Betrun control system is shown in Table 3. Based on the basic data type, the custom read-write access of general register M is added.

Hunan Betrun Robot Co., Ltd.

Table 3 Betrun module corresponding to MODBUS data type

Number	Modbus data type	Address and description of corresponding Betrun module
1	Coil	0-(17+36) Output signal, switching value. (The first 18 are the on-board quantity, and the last 36 are the expansion board quantity.)
2	Input discrete quantity	0-(19+40) Input signal, switching value. (20 are the number of onboard boards, and the last 40 are the number of expansion boards.)
3	Input register	0-99 Status register (16 bits short type, switching value)
		100-149 Alarm register (16 bits short type, switching value)
4	Hold register	Global variable (32-bit floating point type) Analog quantity (32-bit floating point type)
5	General register M	0-499 Custom 0x41 read and 0x42 write commands. Refer to 0x02 and 0x0f commands. Corresponding to M relay in PLC.

Note:

Comparing Table 2 and table 3, it can be seen that the "function code" can only be 16 bit words for reading and writing the "hold register", while the global variables and analog quantities defined in the Betrun control system are 32-bit floating-point data.

Table 4 shows the mapping relationship between the two. One global variable corresponds to two MODBUS register addresses.

Table 4 Address mapping relationship

Modbus hold register address	Global variable address in Betrun controller
0	0_L
1	0_H
2	1_L
3	1_H
……	……
2i	i_L
2i+1	i_H

The program modc.exe is used to further explain the storage form of data in Betrun controller.

As shown in Figure 5, for example:

Floating point number 5.5 in the global variable n of Betrun control system, the hexadecimal order by byte is 00 00 B0 40. Then, when using MODBUS function code 06 for write operation, the address and data of PDU holding register in the data frame are respectively:

- The Modbus data of low address (2n) is 0000
- The Modbus data of high address (2n + 1) is 40B0

That is, the 32-bit floating-point number is divided into two 16-bit words, with the lower 16 bits

HUNAN Betrun Robot Co., Ltd.

in the front and the higher 16 bits in the back. In the 16-bit hold register, the higher byte is in the front and the lower byte is in the back.

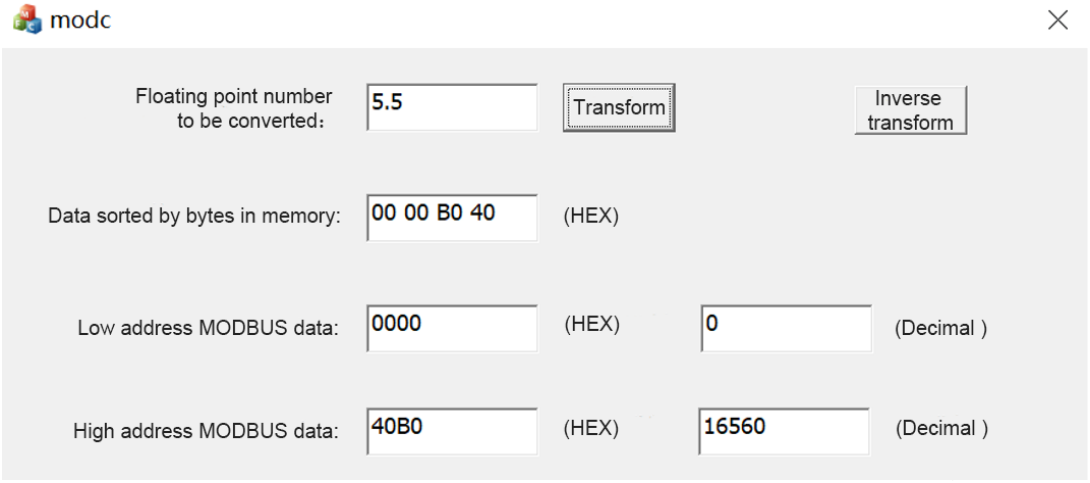


Figure 5 Using modc.exe to convert

3.3 Custom function code for Betrun system as a slave

Betrun system has customized the function code of reading and writing general relay M, which is described as follows.

Read general relay M

Request PDU

Function code	1 byte	0x41
Beginning address	2 bytes	0x0000-0x01F3（0-499）
Number of relays	2 bytes	1-500（0x01F4）

Response PDU

Function code	1 byte	0x41
Number of bytes	1 byte	N*
Relay status	N bytes	n＝N or N+1

Error

Function code	1 byte	0xC1
Exception code	1 个字节	0x01 或 0x02 或 0x03

Example: request to read relay M20-M38 status

Request		Response	
Domain name	HEX	Domain name	HEX
Function code	41	Function code	41
Beginning address Hi	00	Number of bytes	03
Beginning address Lo	14	M27-M20 status	FF

Henan Betrun Robot Co.,

Number of relays Hi	00	M35-M28 status	55
Number of relays Lo	13	M38-M36 status	AA

Write general relay M

Request PDU

Function code	1 byte	0x42
Beginning address	2 bytes	0x0000-0x01F3（0-499）
Number of relays	2 bytes	1-500（0x01F4）
Number of bytes	1 byte	N*
Relay value	N*×1 byte	

Response PDU

Function code	1 byte	0x42
Beginning address	1 byte	0x0000-0x01F3（0-499）
Output quantity	N bytes	1-500（0x01F4）

Error

Function code	1 byte	0xC2
Exception code	1 byte	0x01 or 0x02 or 0x03

Example: write 10 M relays from M20

Request		Response	
Domain name	HEX	Domain name	HEX
Function code	42	Function code	42
Beginning address Hi	00	Beginning address Hi	00
Beginning address Lo	14	Beginning address Lo	14
Number of relays Hi	00	Number of relays Hi	00
Number of relays Lo	0A	Number of relays Lo	0A
Number of dytes	2		
Output quantity Hi	CD		
Output quantity Lo	01		

4 Modbus Communication for Betrun system as a slave

Before using PLC, single chip microcomputer system and visual sensor system as master station and Betrun control system as slave station for data interaction and program control through Modbus communication, PC can be used for Modbus communication test based on RS232.

During communication, Betrun control system acts as a slave station with address 1.

Hunan Betrun Robot Co., Ltd.

4.1 RS232 communication settings

As shown in Figure 6, the PC is physically connected with the RS232 port of the Betrun controller.

According to the definition of DB-9 connector signal in Table 1, "TXD send data", "RXD receive data" and "GND ground" of Betrun RS232 are 7, 8 and 9 pins. The RXD, TXD and GND defined by standard RS232 in PC are 2, 3 and 5 pins, so it is necessary to make a serial port adapter line (as shown in Figure 7) and connect the pins on both sides (7-2, 8-3 and 9-5).

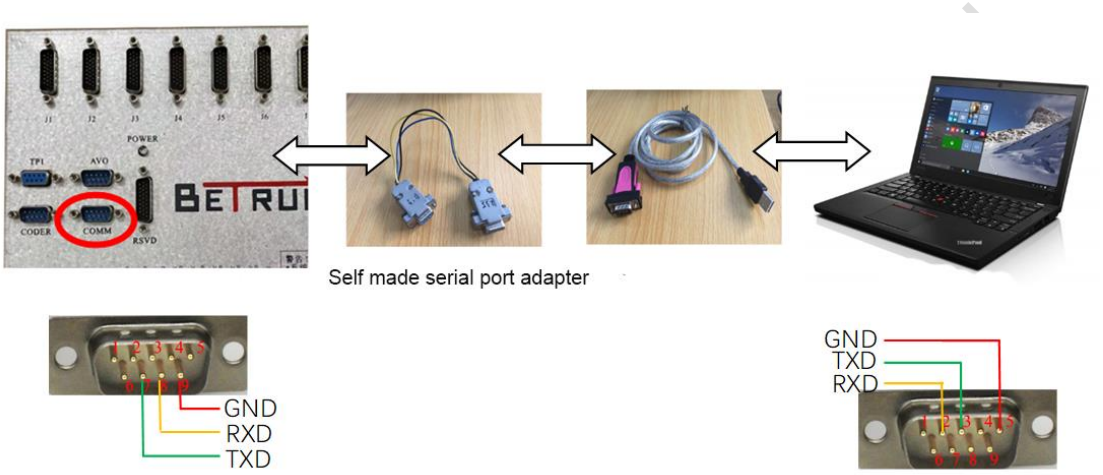


Figure 6 RS232 hardware connection

Note:

(1) The serial port adapter line is connected to one end of the Betrun controller, and the DB-9 needs a female head. The DB-9 connector at the other end needs to be paired according to the serial port to USB cable or the male and female of RS232 port on PC.

(2) The self-made serial port adapter shall be connected as shown in Figure 7, and it cannot be connected reversely.



Figure 7 Corresponding connection mode of self-made serial port adapter DB-9

The communication protocol parameters are configured as follows:

(1) Click [configuration] - [communication protocol] - [MODBUS] in the manual interface of Betrun system teach pendant, as shown in Figure 8, set the parameters and click [save].

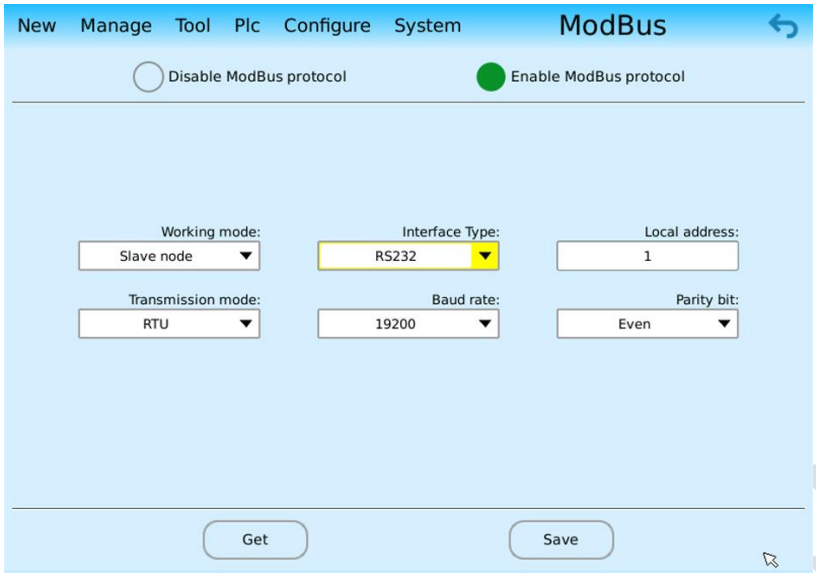


Figure 8 RS232 connection setting of Modbus of Betrun control system

(2) Download and install the Modbus host emulator Modbus Poll on the PC.

After installation and registration, you can see the shortcut of modbuspoll on the desktop. Double click the shortcut to open the software. The interface is shown in Figure 9.

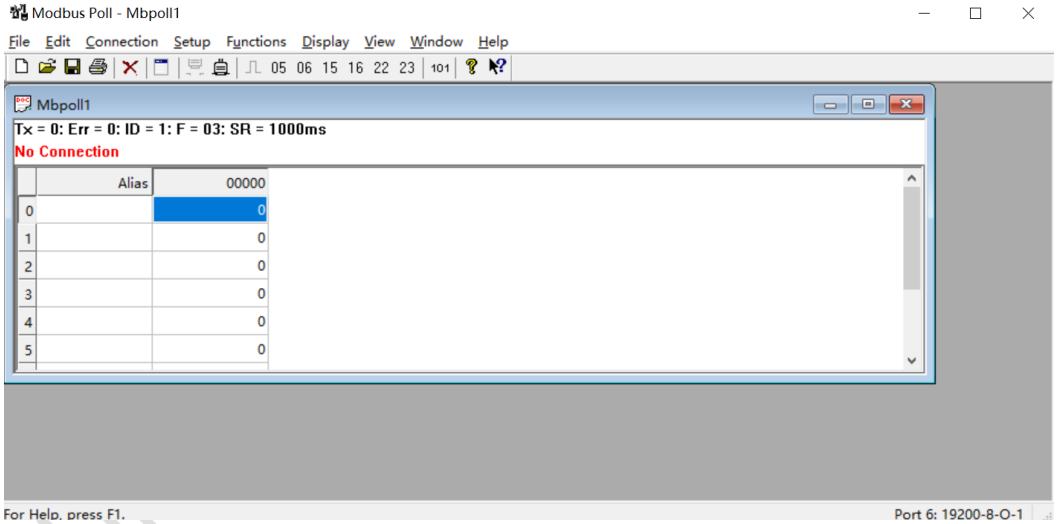


Figure 9 Modbus Poll interface

Click [connection] - [connection setup], and click connection - > connect to open the registration window; Open the readme file extracted from the compressed package, copy the serial number of Modbus poll, paste it into the registration column of the registration window, click OK, the cracking is completed, and enter the PC connection setting interface, as shown in Figure 10.

According to the RS232 parameter setting of Modbus in Betrun system, the same parameter configuration is carried out in MODBUS poll software.

Note: the serial port number of PC port 6 indicates the serial port (com6) used. You need to right-click "my computer" - "management" - "device manager" to view it (as shown in Figure 11). After connecting the adapter, a new port number will appear, that is, the serial port used for Modbus communication.

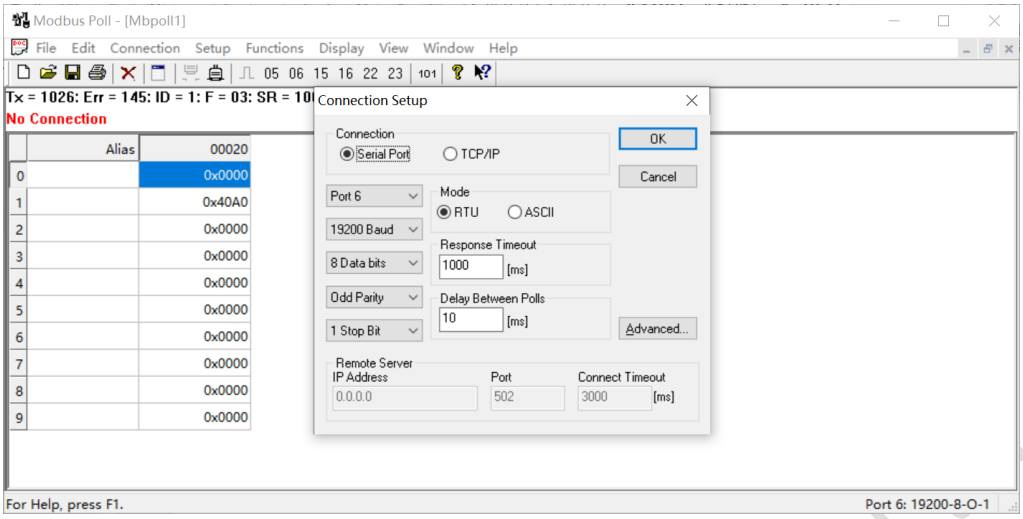


Figure 10 RS232 connection setting of PC MODBUS

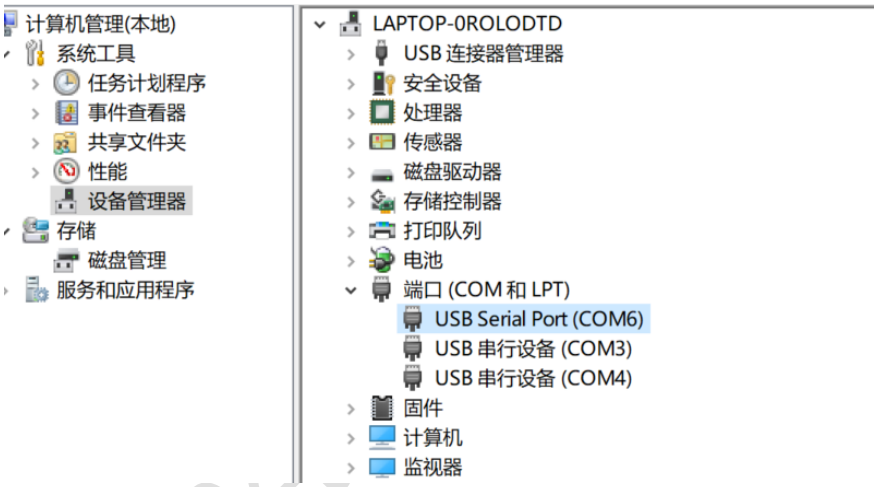


Figure 11 Viewing USB serial port number

Click OK after confirming the setting. If the connection and reading are correct, the main window will display the relevant information of the read register, and the red "no connection" disappears, as shown in Figure 12.

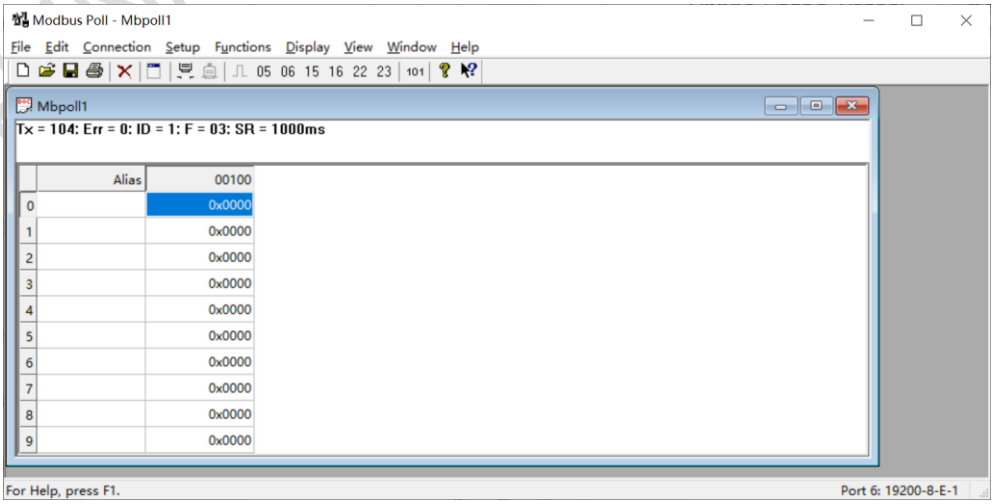


Figure 12 Successful Modbus communication connection

4.2 TCP/IP communication settings

When using TCP/IP mode for communication connection, you only need to connect the network port in Figure 1 with the PC network port, and then set the parameters.

(1) As shown in Figure 13, in the teaching pendant interface of Betrun control system, click [configuration] - [network configuration], enter parameters, and click [setting] to complete network setting.

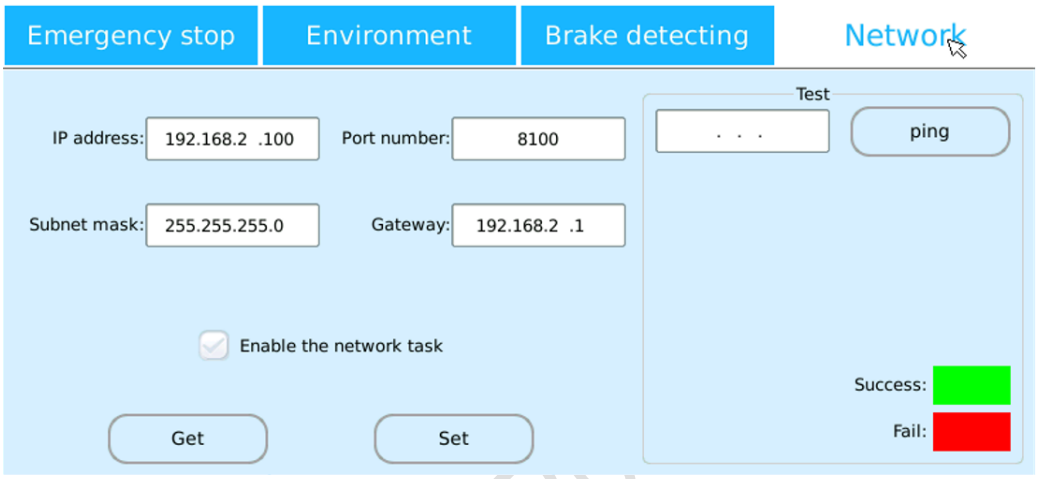


Figure 13 Network configuration of betrun system

(2) Click [configuration] - [communication protocol] - [MODBUS], as shown in Figure 14, select the interface type as "Lan", set the TCP communication port as "502", and click [save].

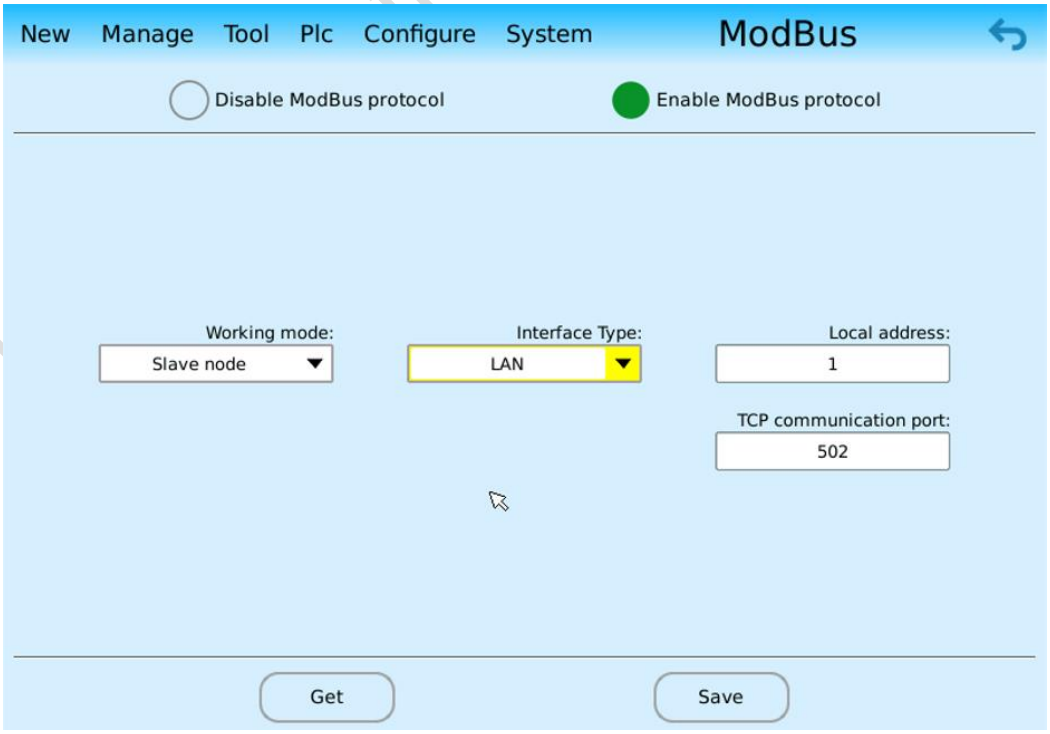


Figure 14 LAN connection setting of Betrun control system MODBUS

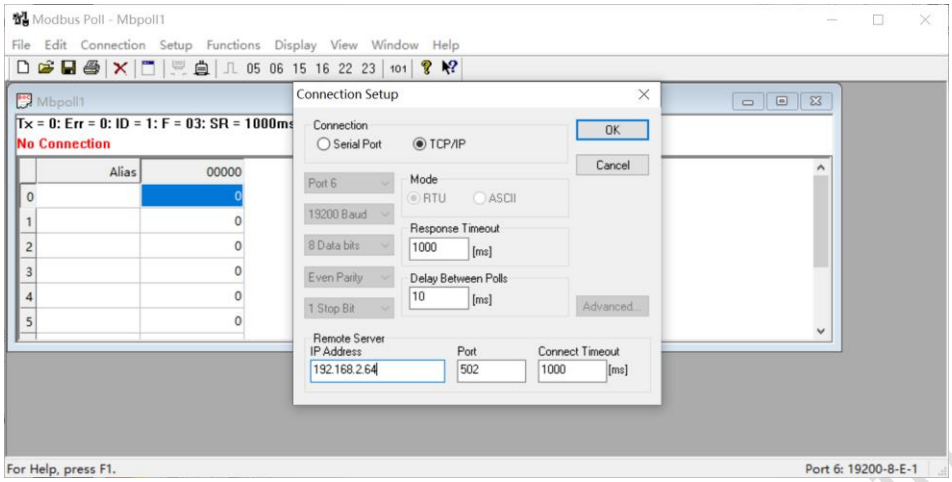


Figure 15 LAN connection setting of PC MODBUS

(3) Enter the connection setting interface in the Modbus poll software of PC, as shown in Figure 15. Set the network IP address of the betrun system according to the configuration in Figure 13; Set the port number according to the configuration in Figure 14.

Click OK after confirming the setting. If the connection and reading are correct, the main window will display the relevant information of the read register, and the red "no connection" disappears.

4.3 Write global variable

The remote control of the robot through MODBUS is realized by reading and writing the special register for program control of betrun system (global variables defined in Section 6.1), so the process of writing global variables needs to be tested.

Example: write 5.5 to the Betrun system global variable GV50.

(1) Click the physical button [menu] - [global variable] on the teach pendant of Betrun system, and set the starting sequence number of the global variable to 50, as shown in Figure 16. As you can see, the value in GV50 is now 0.

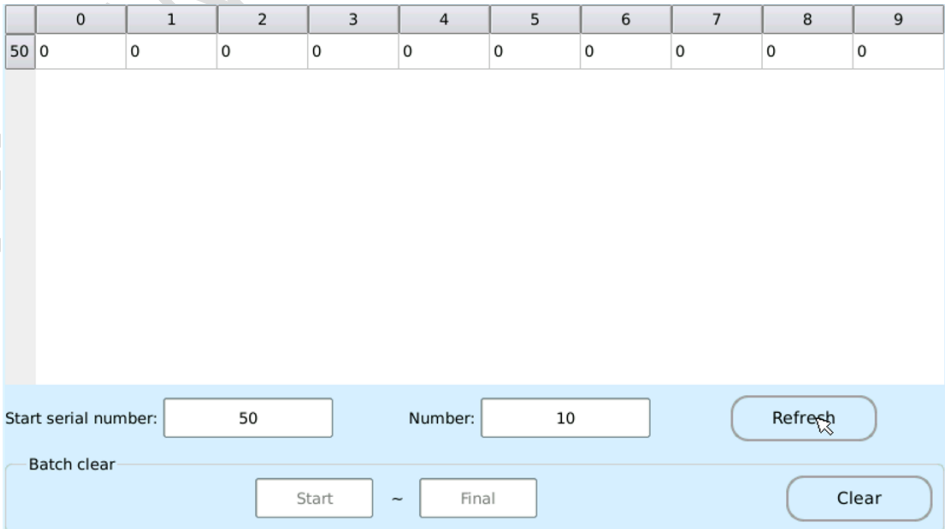


Figure 16 Monitoring global variables through teach pendant

(2) Set the data display mode of Modbus poll to hexadecimal, as shown in Figure 17.

Hunan Betrun Robot Co., Ltd.

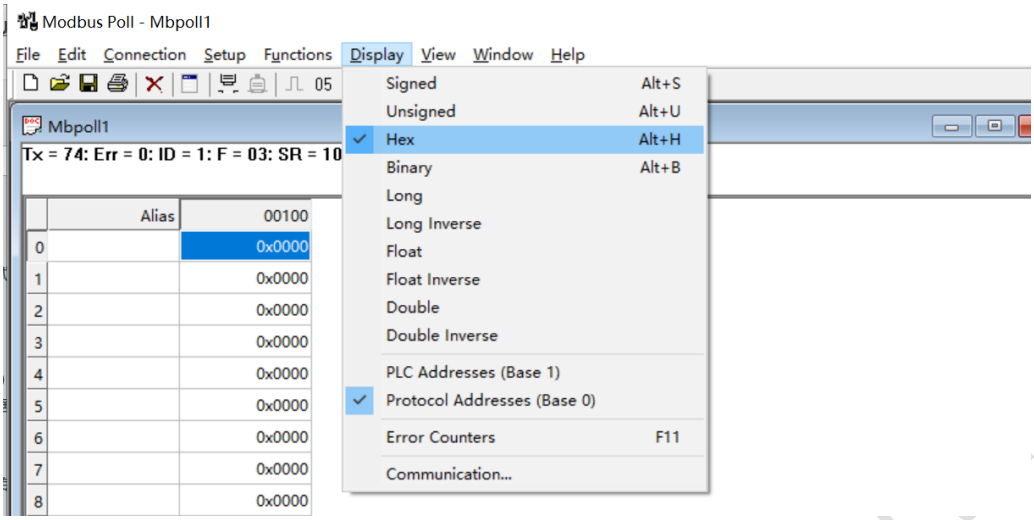


Figure 17 Setting MODBUS poll data display mode

(3) Click [functions] - [06: write single register] of Modbus poll to write hexadecimal 0000 and 40b0 into the holding register with address 100 and address 101 respectively, as shown in Figure 18. The result of this operation is to "write" 5.5 to the global variable gv50 of the Betrun system.

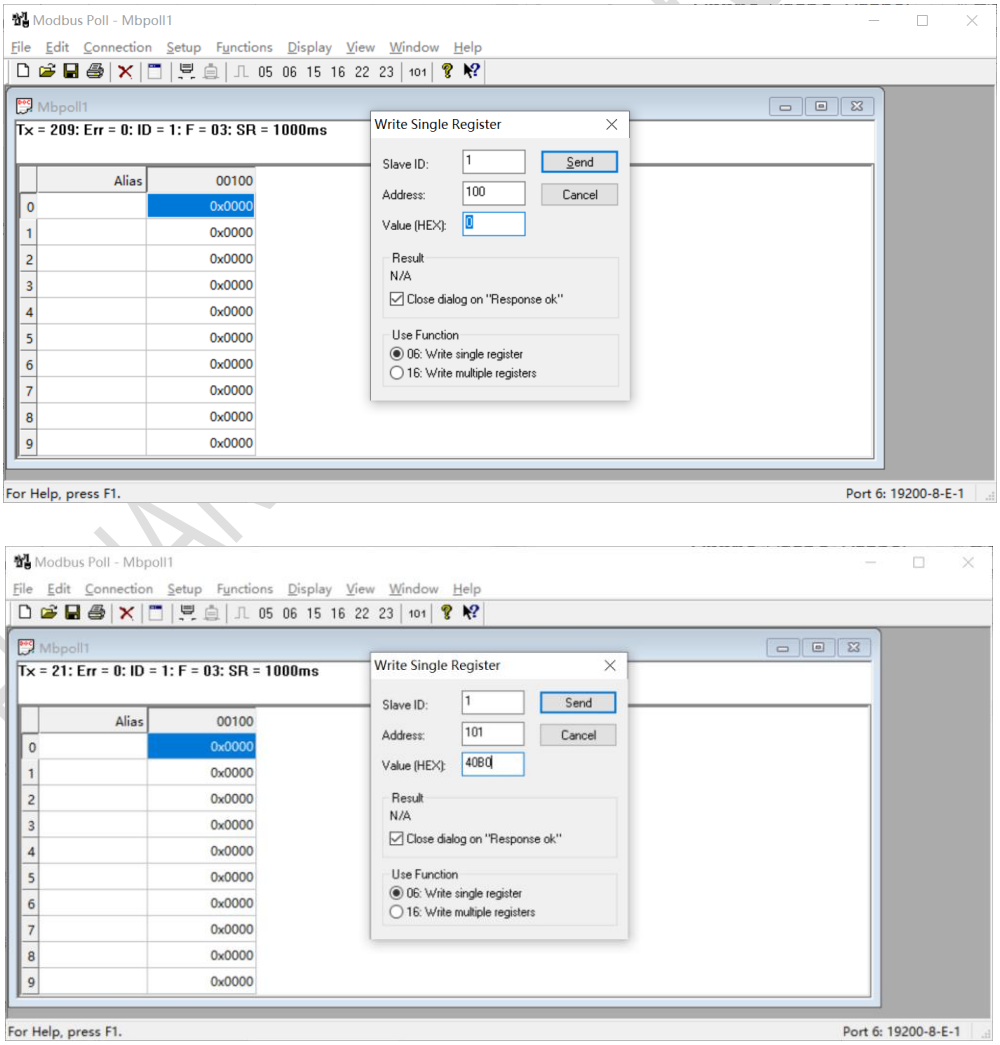


Figure 18 Write values to the system global variable GV50

(4) Click [refresh] on the teaching pendant interface, as shown in Figure 19. The value in GV50 changes to 5.5, indicating that the writing is successful.

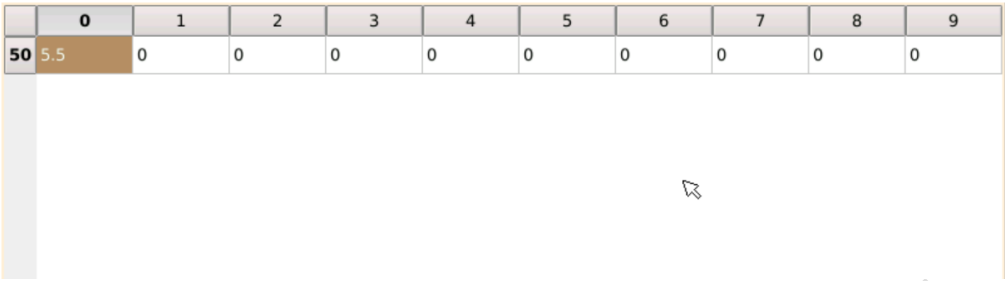


Figure 19 Global variables written successfully

4.4 Read global variables

As shown in Figure 20, the global variables GV206-GV211 of the Betrun system are the X-Y-Z coordinate value and a-b-c attitude value of the robot in the base coordinate system. In the process of interaction, the current position and posture of the robot end can be obtained by reading the global variables.

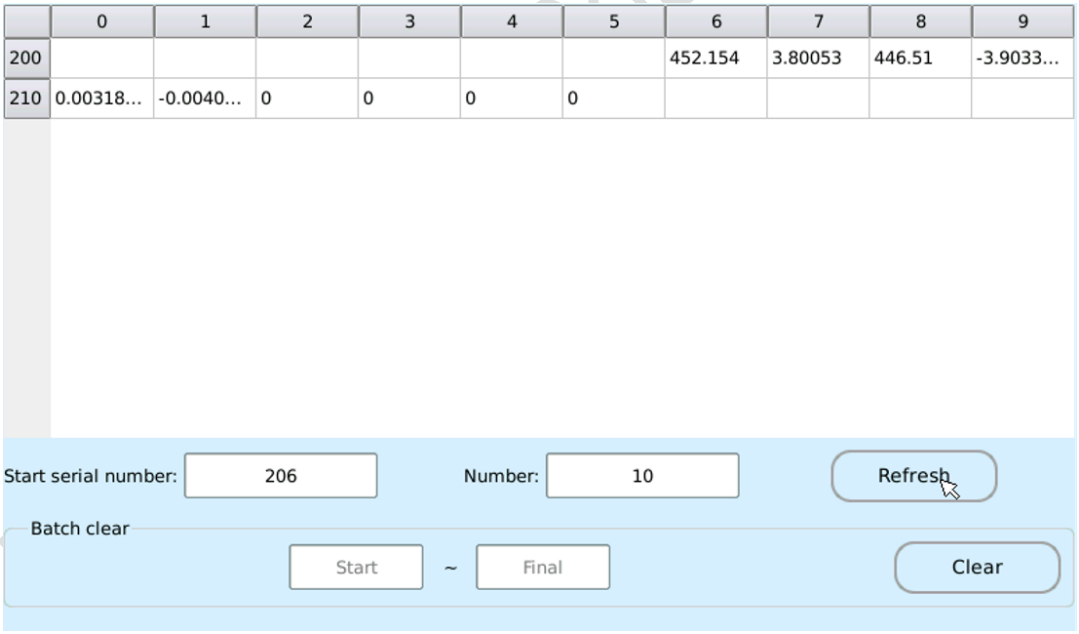


Figure 20 Global variables GV206-GV211 of system

Click [setup] - [read/write definition] of Modbus poll, set the function code to "03", the start address of read holding register is 412, and the quantity of continuous reading is 12. As shown in Figure 21.

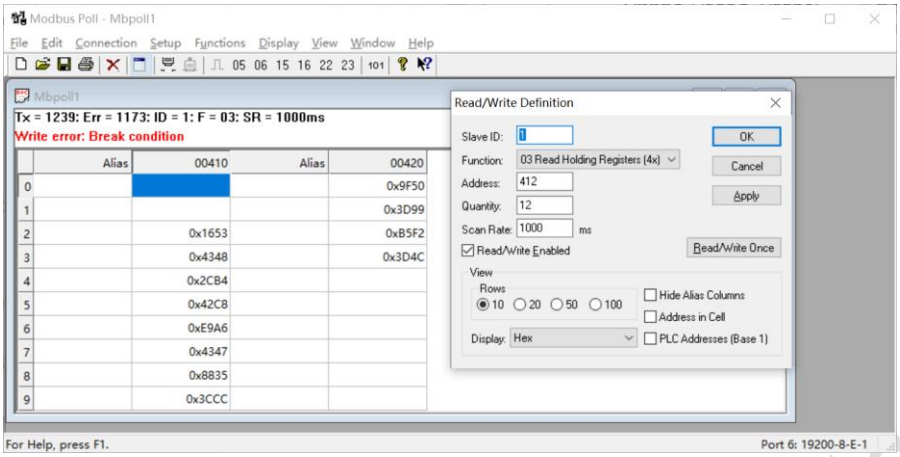


Figure 21 Reading multiple system global variables

5 Modbus Communication for Betrun system as a master

In some applications, Betrun control system needs to be used as the master station to read and write to the slave station based on Modbus communication protocol. For example, the sensor data is read in real time, and the motion trajectory is accurately controlled through the offset command in the program.

5.1 Communication settings

Click [configuration] - [communication protocol] - [MODBUS] in the manual interface of betrun system teaching pendant, and select the working mode as "master node". As shown in Figure 22, the TCP communication port can be set by selecting "interface type", configuring local address, remote address, and server IP address. Refer to sections 4.1 and 4.2 for the specific process.

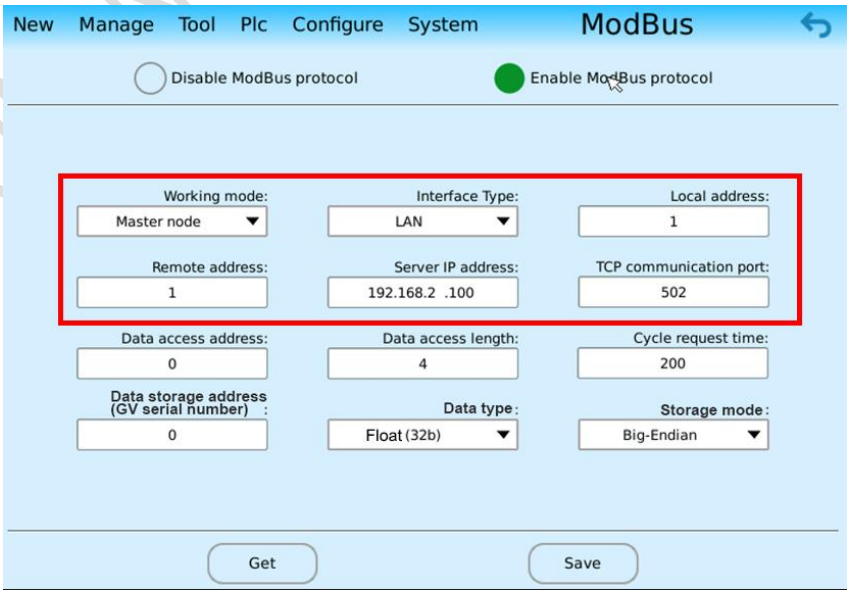


Figure 22 Modbus communication configuration interface for "master node"

HUNAN Betrun Robot Co., Ltd.

5.2 Read slave data

At present, Betrun control system provides the function of reading slave station data regularly. As shown in Figure 23, set relevant parameters in the red box in the Modbus communication configuration interface. Table 5 gives a detailed description of the parameters.

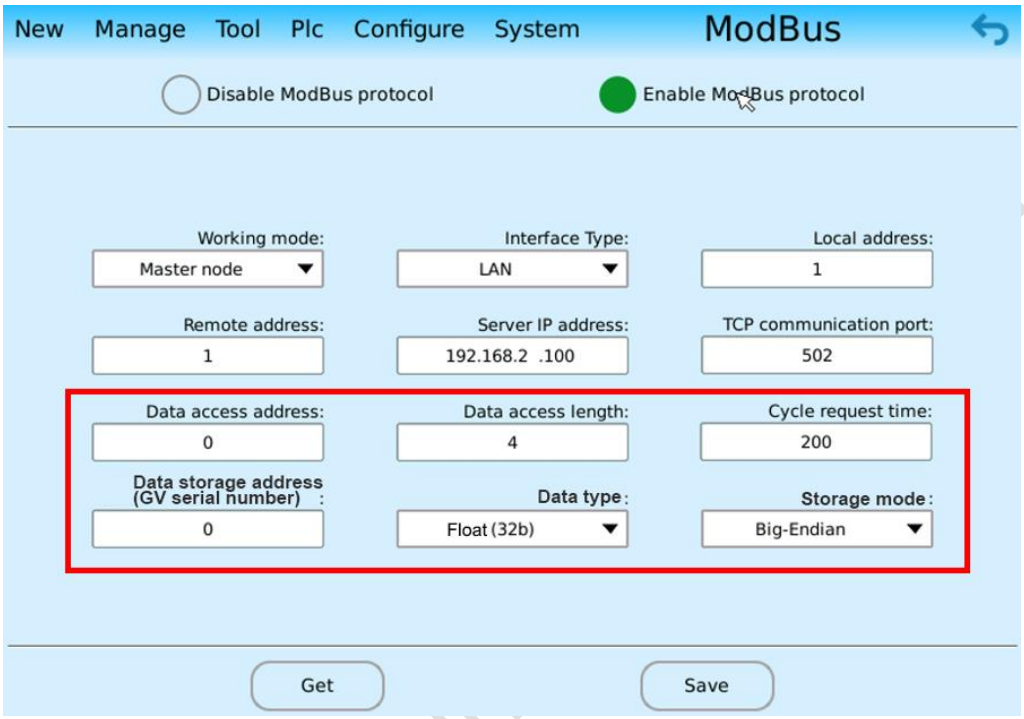


Figure 23 Configuration of reading slave data

Table 5 Description of configuration parameters for reading slave data

Number	Parameters	Description
1	Data access address	The first address of the holding register (16 bit word) in which data needs to be read from the slave station. Betrun control system accesses the slave data and adopts function code 03 (reading multiple registers).
2	Data access length	The number of hold registers (16 bit words) that need to read data from the slave station
3	Cycle request time (mm)	Interval between timed data reading
4	Data storage address (global variable serial number)	The first address of the global variable (32-bit floating-point type) used to store the read data in the Betrun control system
5	Data type	The data types in the slave station include 16 bit integer and 32-bit floating point
	Storage mode	When the data in the slave station is of 32-bit floating-point type, it is also necessary to clarify its storage mode, which is divided into Big-Endian and Little-Endian.

Example: as shown in Figure 24, the slave data to be read is 7.9, and the holding register of the

slave adopts the big endian mode to save the data.

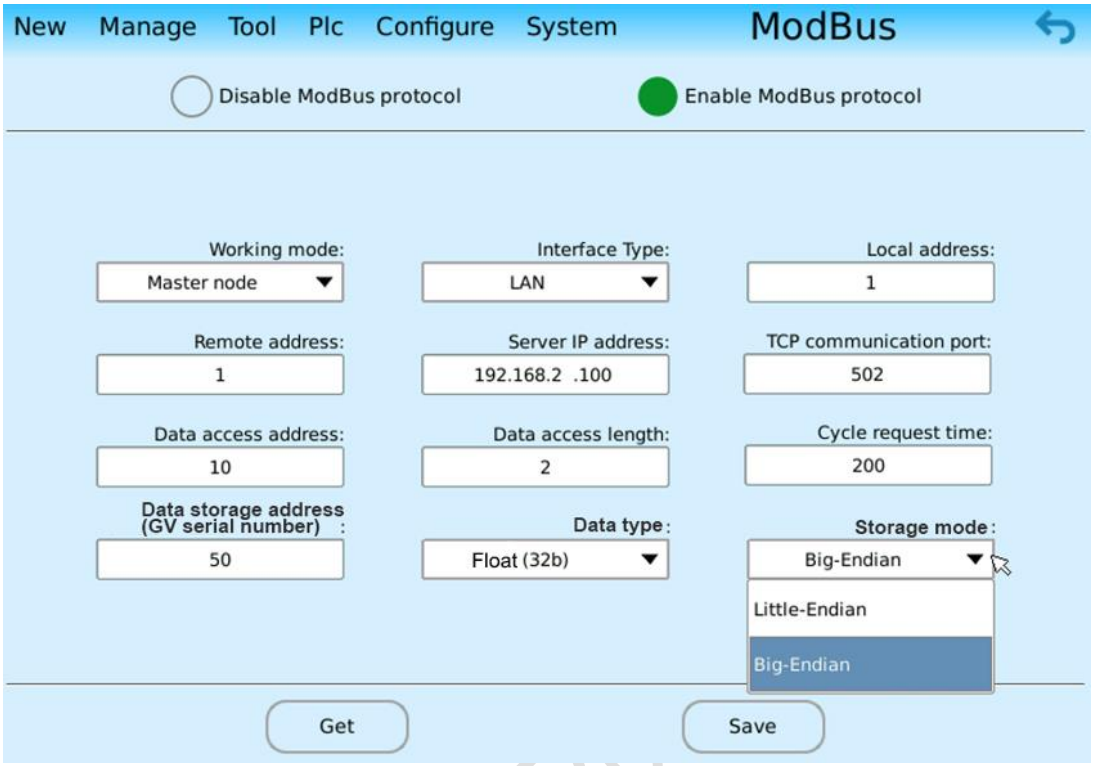


Figure 24 Example of reading slave floating point data

The slave station divides the 32-bit floating-point number into two 16 bit words, with the lower 16 bits first, and saves it in the hold register with address 10; It is stored in the address from high to 16 bits in the register. The Big-Endian mode is used to save data, that is, in the 16 bit hold register, the high byte is in the front and the low byte is in the back. (Little-Endian mode means that the low byte in the 16 bit hold register comes first and the high byte comes last.)

When reading with MODBUS function code 03, the address and data of PDU hold register in the data frame are respectively:

- The data of low address 10 is CCCD
- The data of high address 11 is 40FC

After reading the data, it is stored in the global variable 50 of Betrun control system. The hexadecimal order is CD CC FC 40 by byte. Click the physical button [menu] - [global variable] in the teach pendant of Betrun system, and set the starting sequence number of the global variable as 50, then the value in GV50 is 7.9.

If the slave data type is 16 bit integer and the setting is as shown in Figure 25, the data of the hold register with the slave address of 10 (Modbus defaults to Big-Endian mode) is read and stored in GV50, and the data of the hold register with the slave address of 11 (Modbus defaults to Big-Endian mode) is read and stored in GV51.

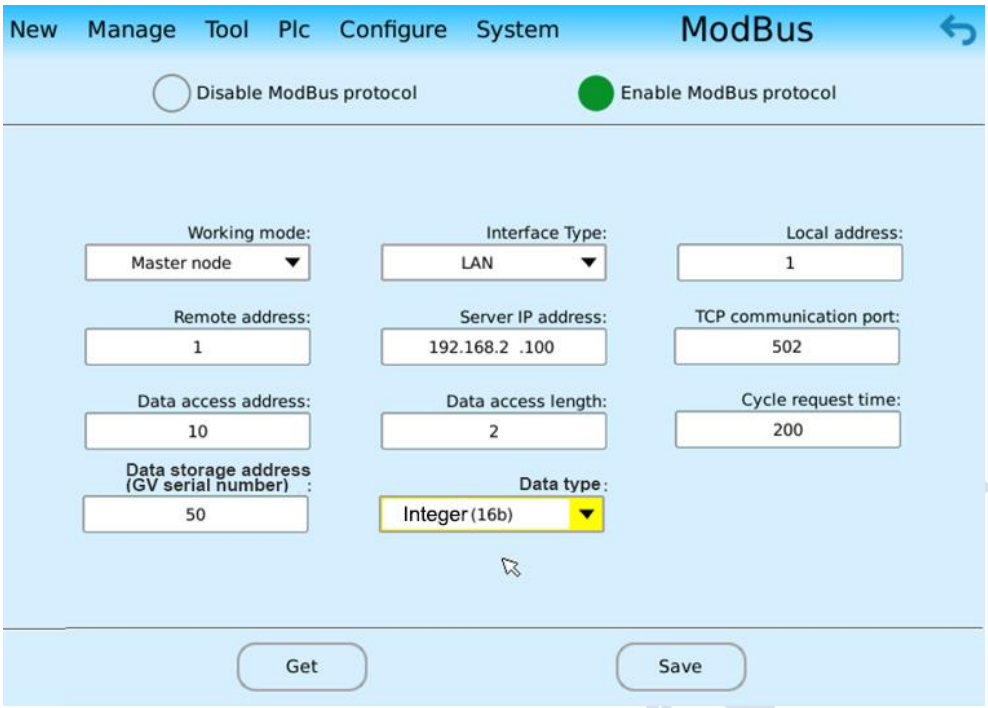


Figure 25 Example of reading slave integer data

5.3 Write slave data

Betrun control system writes slave data through instructions in the program. As shown in Figure 26, click "write MODBUS register" in the "basic instruction". With this instruction, the background will write multiple constants to the hold register of the slave station through the 16 function code or write the values of multiple global variables to the hold register of the slave station.

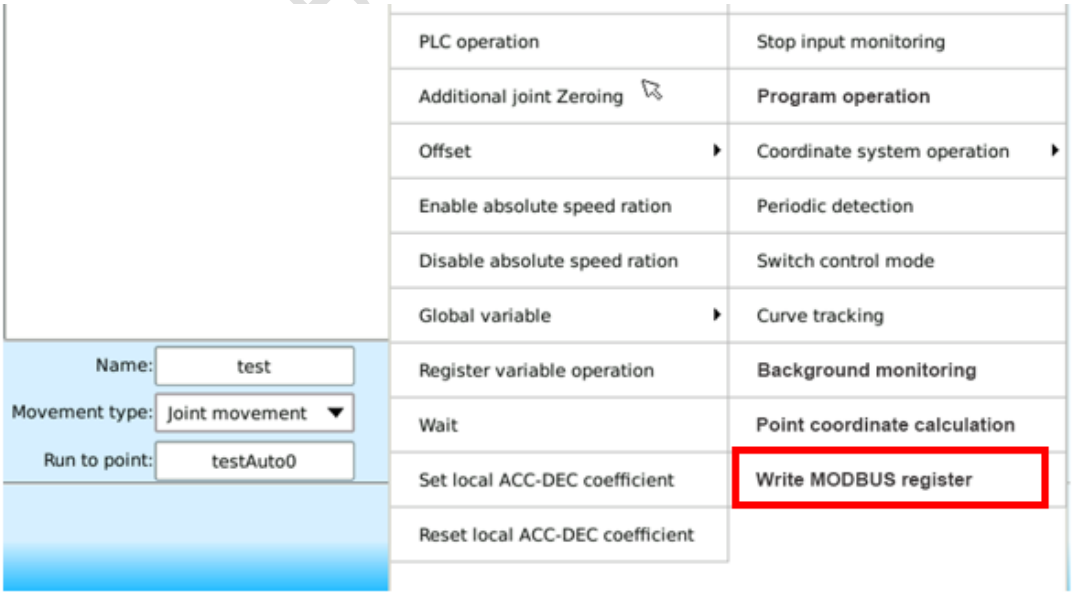


Figure 26 "Write MODBUS register" instruction

Note:

(1) The data source is a constant or global variable, and its written quantity refers to the number

of 32-bit floating-point numbers. The storage mode of the written floating-point numbers is the mode configured according to the Modbus master station.

(2) When writing constants, the number of written constants cannot exceed 3; When writing global variables, the number of writes cannot exceed 10.

6 Robot motion control based on Modbus

When Betrun control system is used as a "slave", the external can realize the motion control of the robot through Modbus protocol.

6.1 Betrun program control special register

When Betrun is a slave station, the external accesses the Betrun control system through Modbus, and its operation objects are global variables, input/output and other resources in the Betrun control system. Therefore, in the global variable definition of Betrun control system, the global variables GV220-GV225 are defined as the special register for program control. Program control can be realized by "writing" the global variables, which has the same effect as the key operation of the teach pendant.

Table 6 shows the functions mapped to Modbus register address [440-451] for program control. Among them, the way to control the robot motion is "incremental motion". Before operation, the "incremental motion" needs to be assigned with a global variable (joint offset from joint 1 to joint 6 or x-y-z-a-b-c offset in the Base Coordinate System).

Table 6 Description of Modbus register program control function

Address	Corresponding global variable	Description	Parameter	Note
440	220_L	Run program	Command instruction, 1 - run the program, 0x40 - move according to the joint coordinate increment, 0x41 - move according to the base coordinate increment	Valid only for 0x06 command
441	220_H	Stop program	Command instruction, 1 - stop	Valid only for 0x06 command
442	221_L	Pause program	Command indication, 1-pause	Valid only for 0x06 command
443	221_H	Continue procedure	Command instruction, 1-continue	Valid only for 0x06 command
444	222_L	Switch working mode	Working mode, 0-manual, 1-automatic	Valid only for 0x06 command
445	222_H	Run to zero	Zero position type, 0-7-corresponding single axis zeroing,	Valid only for 0x06 command

HUNAN Betrun Robot Co., Ltd.

			11-full axis zeroing	
446	223_L	Set the running program ID	ID of the program to be run	Valid for 0x03, 0x06 and 0x10 commands
447	223_H	Set motion mode	Motion mode, 0-step motion, 1-single motion, 2-cycle motion	Valid for 0x03, 0x06 and 0x10 commands
448	224_L	Set cycle number	Number of cycles, 0-infinite cycle, others - number of cycles	Valid for 0x03, 0x06 and 0x10 commands
449	224_H	Set running speed	Operating speed, 1-100	Valid for 0x03, 0x06 and 0x10 commands
450-451	225	Sets the desired speed for incremental motion	Floating point type, speed unit Joint coordinates: RPM Base coordinate: mm/S	Valid for 0x03, 0x06 and 0x10 commands

In the global variable definition of Betrun control system, global variables GV280, GV281 and GV291 are also defined as program operation status display registers. The robot operation can be obtained remotely by "reading" the global variables.

Map the global variable to the address of Modbus register, and table 7 gives the description of status display function parameters.

Table 7 Description of Modbus register status display function

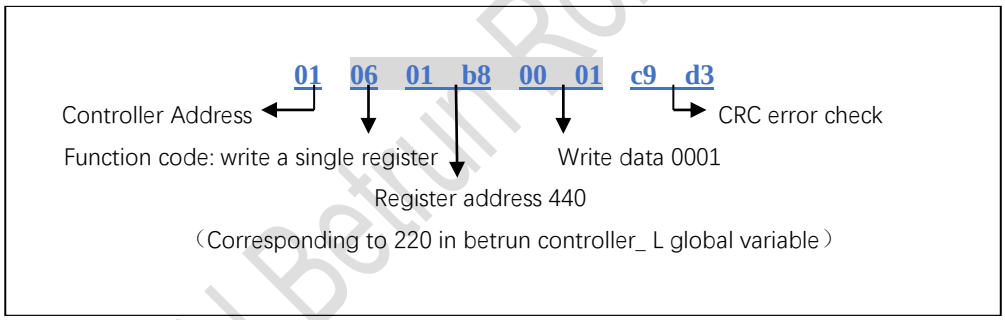
Address	Corresponding global variable	Description	Parameter	Note
560	280_L	Current working mode of controller	0-manual, 1-automatic	Valid only for 0x03 command
561	280_H	Current running state of the program	0-stop, 1-running, 2-pause	Valid only for 0x03 command
562	281_L	Alarm type	B0 - emergency stop alarm, B1 - servo alarm, B2 - abnormal brake alarm, B3 - algorithm alarm, B4 - obtain encoder angle alarm	Valid only for 0x03 command
582	291_L	Command return status	0 - initial value 0xff command execution completed 1 - working mode error 2 - emergency stop alarm 3 - servo alarm 4 - abnormal brake alarm	Valid only for 0x03 command

HUNAN Betrun Robot Co., Ltd.

			5 - bus servo initialization error 6 - during exercise 7 - data length error 8 - get encoder angle error 9 - update encoder angle error 10 - calculation error 11 - algorithm error during operation 12 - movement timeout	
576,577	288	System time	System date: year (2b) + Month (1b) + day (1b)	Valid only for 0x03 command
578,579	289	System time	System time: hour (2b) + minute (1b) + second (1b)	Valid only for 0x03 command
580,581	290		System millisecond counter	Valid only for 0x03 command

Example of Modbus control command:

Function: write global variable 220 in Betrun controller through MODBUS RS232 communication mode_ L (mapped to MODBUS holding register, address 440) to control the robot to run the specified program.



Note: before using this instruction, you need to set global variables such as operation mode, speed, operation times and program ID.

6.2 Motion control method

When Betrun control system is used as a "slave", there are three methods to realize robot motion control through Modbus protocol.

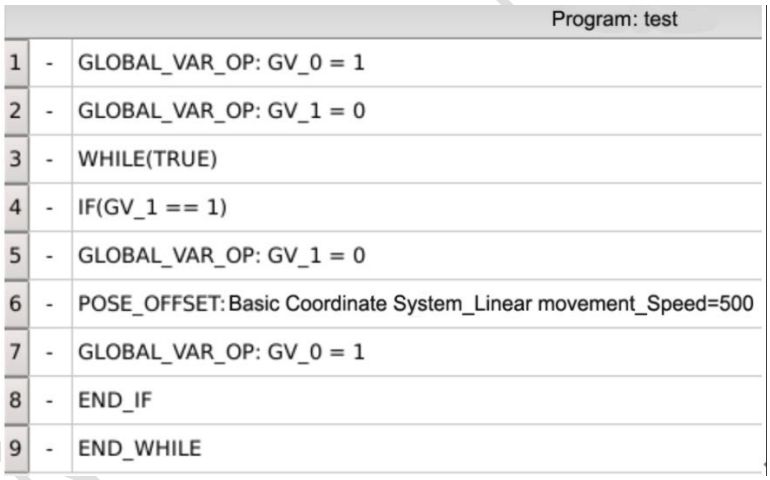
(1) Through the write operation of the special register for program control, the single step incremental motion control of the robot is directly carried out.

(2) Use the Pose_ Offset instruction to control the incremental motion of the robot.

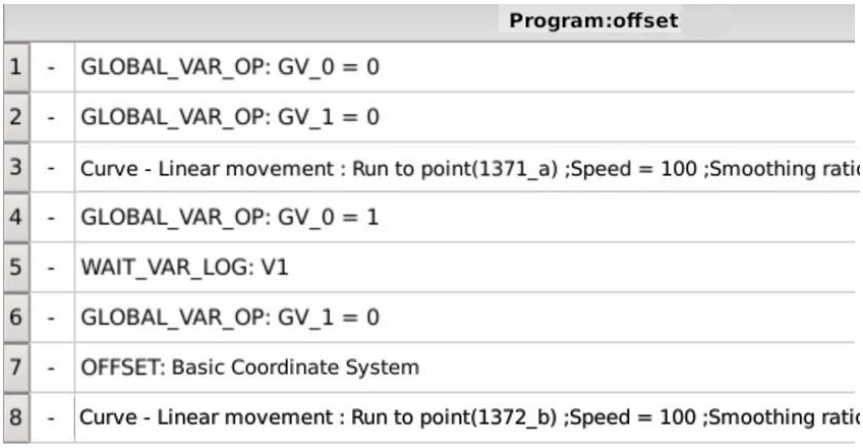
(3) Use the visual sensor and the Offset instruction.

HUNAN Betrun Robot Co., Ltd.

Operation: directly control the single step incremental motion of the robot	
Premise	Modbus communication parameters, baud rate, data bit, stop bit, etc. have been set
Steps	1 Switch to automatic mode (222_L) 2 Set incremental parameters (212-217) 3 Set the desired movement speed (225) 4 Start movement (220_L, set 0x40/00x41) 5 Query command return status (291) 6 Query whether there is an alarm during movement (281) 7 During the movement, pause (221_L), continue (221_H), or stop (220_H) the movement as required

Operation: Use the Pose_Offset instruction to realize incremental motion	
Premise	Modbus communication parameters, baud rate, data bit, stop bit, etc. have been set
Steps	1 Write basic programs in the control system (through teaching pendant or off-line programming)  (GV_0 and GV_1 are global variables of the controller) 2 Switch to automatic mode (222_L) 3 Set the ID of the running program (223_L) 4 Set motion mode (223_H, 1-single motion) 5 Set the desired movement speed (225) 6 Start movement (220_L, 1-run program) 7 After calculating the new offset data, the external vision sensor program judges GV_0. If it is 1, the offset data will be written to the global variable corresponding to the Pose_Offset instruction, and GV_0 set 0 and GV_1 set 1. 8 Query command return status (291) 9 Query whether there is an alarm during movement (281) 10 During the movement, pause (221_L), continue (221_H), or stop (220_H) the movement as required

Operation: Using vision sensor and the Offset instruction	
Premise	Modbus communication parameters, baud rate, data bit, stop bit, etc. have been set
Steps	1 Write basic programs in the control system (through teaching pendant or off-line programming)



(GV_1 is the global variable of the controller, Offset is the offset instruction, and its function is to modify the "run to point" of the first curve behind it.)

2 Switch to automatic mode (222_L)
3 Set the ID of the running program (223_L)
4 Set motion mode (223_H, 1-single motion)
5 Set the desired movement speed (225)
6 Start movement (220_L, 1-run program)
7 After calculating the new offset position, the external vision sensor program writes the data into the global variable corresponding to the offset instruction and GV_ 1 set 1.
8 Query command return status (291)
9 Query whether there is an alarm during movement (281)
10 During the movement, pause (221_L), continue (221_H), or stop (220_H) the movement as required

Part V Application of Coordinate Systems

Coordinate system is a position index system defined on the robot or space in order to determine the position and attitude of the robot. Commonly used include: joint coordinate system, base coordinate system, tool coordinate system and user coordinate system. The joint coordinate system and base coordinate system are defined by default. The tool coordinate system and user coordinate system need to install the connecting flange installation tool (end effector) and place the workpiece table before the teaching and programming of industrial robot.

In order to facilitate operators to flexibly apply the coordinate system in the process of teaching and programming, Betrun control system provides three "coordinate system switching" functions..

1 Coordinate system transformation instruction

Purpose: to expand the teaching program of robot based on a tool on one worktable into a pipeline operation program with multiple worktables and replaceable tools.

As shown in Figure 1, Betrun control system provides a pair of basic instructions for "coordinate system transformation", which need to appear in pairs when used. This pair of instructions can realize the transformation of three coordinate systems, and the functional expression is shown in Table 1.

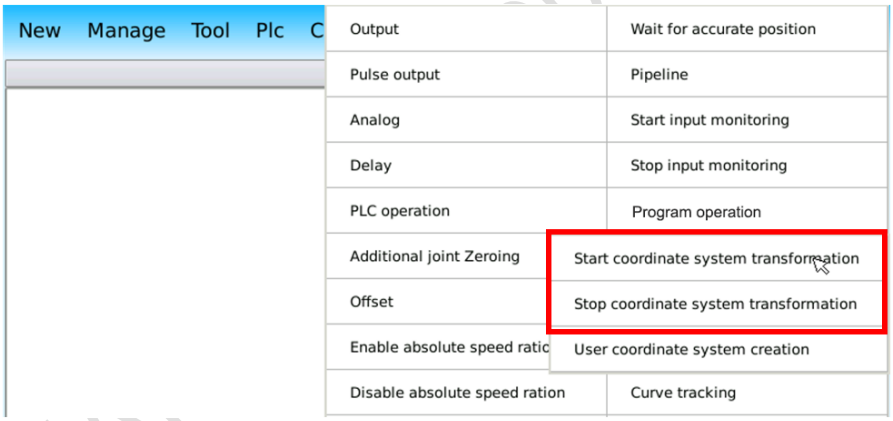


Figure 1 "coordinate system transformation" command pair

Table 1 function description of "coordinate system transformation" command

Number	Coordinate system type	Function description
1	User	When multiple worktables need assembly line operation, multiple user coordinate systems can be established. It is only necessary to carry out teaching programming on one of the worktables, and then use the user coordinate system transformation instructions many times, and call the teaching program segment in it to realize the pipeline action of multiple worktables.
2	Tool	For multiple process operation tasks with the same trajectory, such as welding + grinding, the tool coordinate system can be established for the welding gun and grinding tool respectively. It only needs to use the welding gun to complete the teaching programming, and then use the tool coordinate system transformation command in the program to transform the welding gun

		tool coordinate system into the grinding tool coordinate system, in which the teaching program segment is called, and the grinding operation can be completed according to the same motion track.
3	User + tool	For a teaching program, the user coordinate system and tool coordinate system can be transformed at the same time to realize the combination of the above two functions. For example: teach the welding track program on worktable 1, and then call the program through coordinate system transformation to complete the welding and grinding on worktable 1, welding and grinding on worktable 2.

Case: as shown in Figure 2, the user coordinate system test is established on two worktables respectively test_u1 and test_u2. During the teaching, only P1-P3 points need to be taught on the worktable 1, and then the user coordinate system is changed by using the coordinate system transformation instruction in the program according to Fig. 4 and calling the movement track program segment in the instruction pair. The robot will continue to complete the same movement at the corresponding position of the worktable 2 after moving on the worktable 1. When there are many items on the worktable, it is recommended to add transition points before the coordinate system transformation instruction to prevent unpredictable collision of motion trajectory during worktable switching.

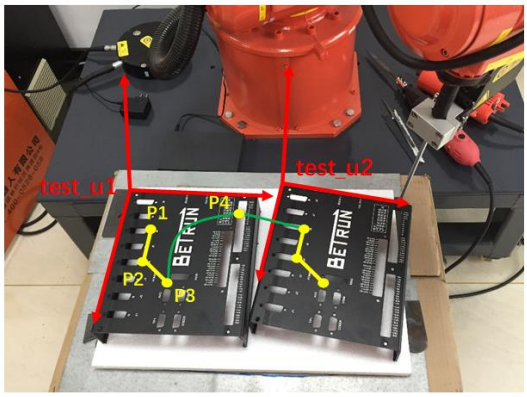


Figure 2 User coordinate system transformation instruction use case diagram

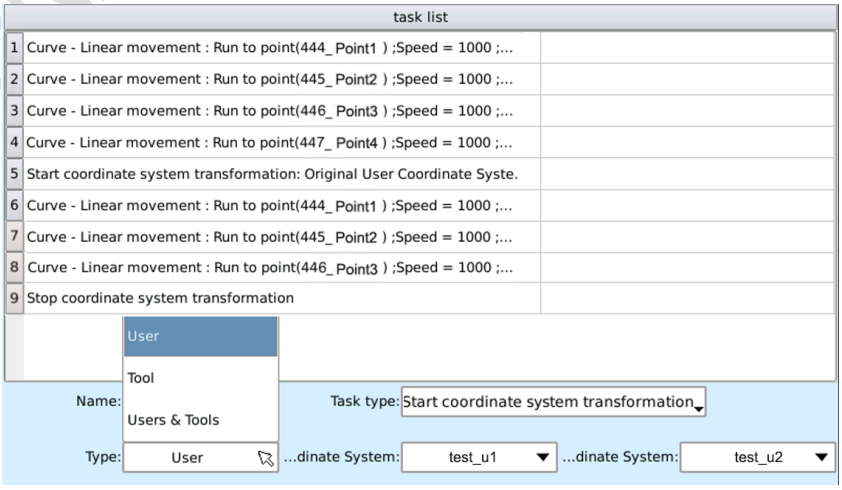


Fig. 3 application case of "user coordinate system transformation" instruction

Note:

(1) When establishing the user coordinate system, the reference point needs to be associated with the corresponding tool coordinate system. The teaching points P1-P3 in the program also need to be associated with the corresponding tool coordinate system. As shown in Figure 4, it is necessary to remove the "√" in front of "not used" and select the currently used tool coordinate system.



Figure 4 Associated tool coordinate system when creating a new point

(2) To copy and paste the program segment, click the program name bar to switch to the full screen interface of the program. Select the program to be copied (the "selected task" will appear on the right), click the [copy] button, and then the paste button and options will appear. Select the "enable coordinate system transformation" command line, check "post paste", and click the [forward paste] button to complete the copy and paste of the program segment.

(3) When the operation on the worktable is complex and the program section is long, modular program can be used for programming.

2 Program reuse among multiple robots

Purpose: to convert the program teaching point on a robot into a program package with parameters through "forward generation", and then copy it to multiple robots for "reverse generation" of the program teaching point, so as to realize the sharing and use of the program.

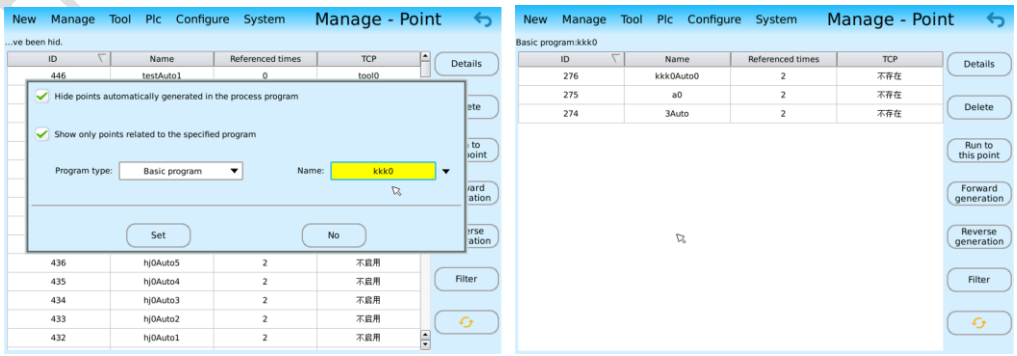


Figure 5 Show the points in a program

Step 1: complete the association of program points with user coordinate system and robot parameters on the current robot teaching pendant

(1) As shown in Figure 5, in the "point management" interface of betrun teaching pendant, click the [filter] button and select the name of the basic program to be copied to other robots, then only the points in the program can be presented.

(2) As shown in Figure 6, select all points, then click the [forward generation] button, select the name of the user coordinate system used in the current robot teaching point, and click [confirm], then all program points will include not only the six joint data and the tool coordinate system data during teaching, but also their relative pose data in the user coordinate system.

Note: when establishing the user coordinate system, the reference point needs to be associated with the corresponding tool coordinate system. The teaching points in the program also need to be associated with the corresponding tool coordinate system.

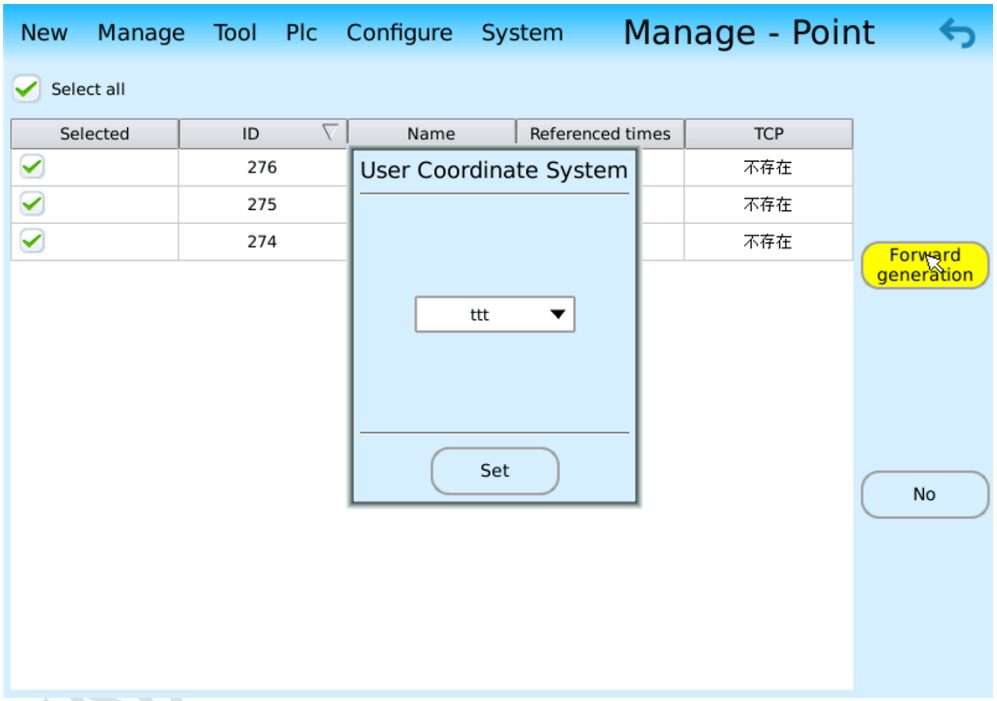


Figure 6 Data association between points and user coordinate system

Step 2: back up the user program with USB flash disk and update it to another robot controller

Prepare a FAT32 format USB2 0 and create a new folder and configuration file in it.

The steps are as follows:

(1) Create a new folder under the root directory of the USB flash disk. The name must be "robcon".

(2) Create a new text document in the "robcon" folder and rename it "config.ini".

(3) Open "config.ini" with notepad and edit the text content as shown in Figure 7. Here, the user program is not a simple copy and backup, but needs to bring the robot parameters and set record = 64.

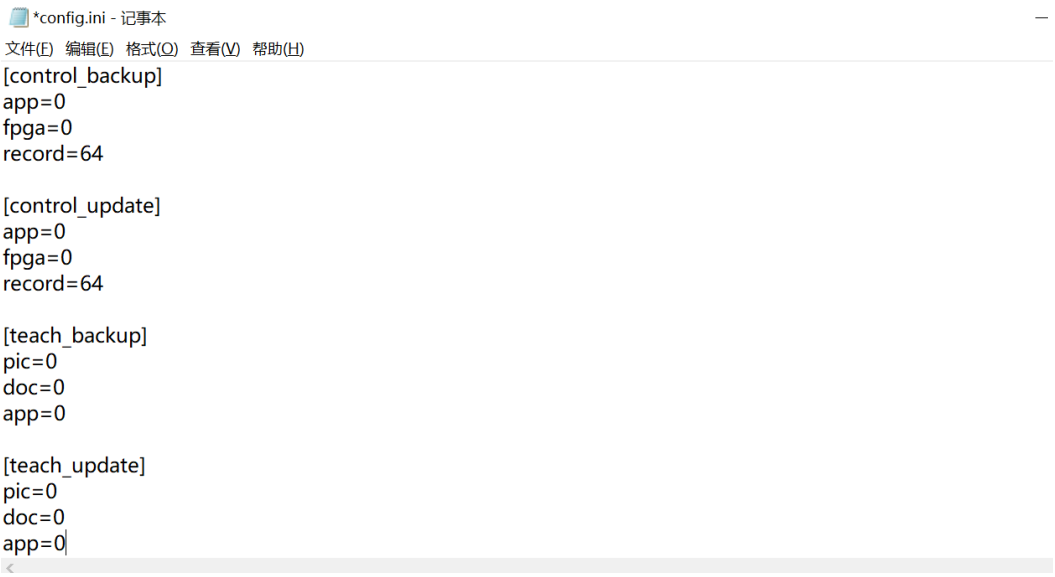


Figure 7 "config.ini" configuration

Insert the USB flash disk into the controller, enter [tools] - [update and backup], and select the controller for backup.

Then insert the USB flash disk into the target robot controller (the software version needs to be the same), enter [tools] - [update and backup], select the controller and update it.

Step 3: establish the tool coordinate system and user coordinate system in the target robot teaching pendant.

There are differences in machine parameters between the target robot and the original robot, and the data of tool coordinate system and user coordinate system can not be the same. Therefore, before the target robot uses the application, it is necessary to re-establish tool coordinate system and user coordinate system.

Note: because all program points are associated with the tool coordinate system ID, after the new tool coordinate system is established, the copied original tool coordinate system needs to be deleted to ensure that the new tool coordinate system uses the same ID number as the original tool coordinate system, and the copied program points will be associated with the new tool coordinate system.

When establishing the user coordinate system, the reference point needs to be associated with the tool coordinate system.

Step 4: complete the reverse generation of program points on the target robot teaching pendant.

In the "point management" interface of the target robot betrun teaching pendant, click the [filter] button and select the name of the copied basic program to present only the points in the program.

Click the [reverse generation] button, select all points, then click the [reverse generation] button again, select the name of the user coordinate system newly created by the target robot, and click [confirm], then all program points will be converted into six joint data matching the user coordinate system and robot parameters of the target robot.

HUNAN Betrun Robot Co.,

After completing the above steps, the target robot can complete the same task as the original robot without manual teaching programming.

3 Quick correction of tool coordinate system

Purpose: if the tool is replaced due to wear or damage, the tool coordinate system can be corrected by using the external reference method (the calibration can be completed only by aligning the tool with a measured external reference point).

In the tool coordinate system interface of Betrun teaching pendant, as shown in Figure 8, when creating a new tool coordinate system, you can create a mark point.

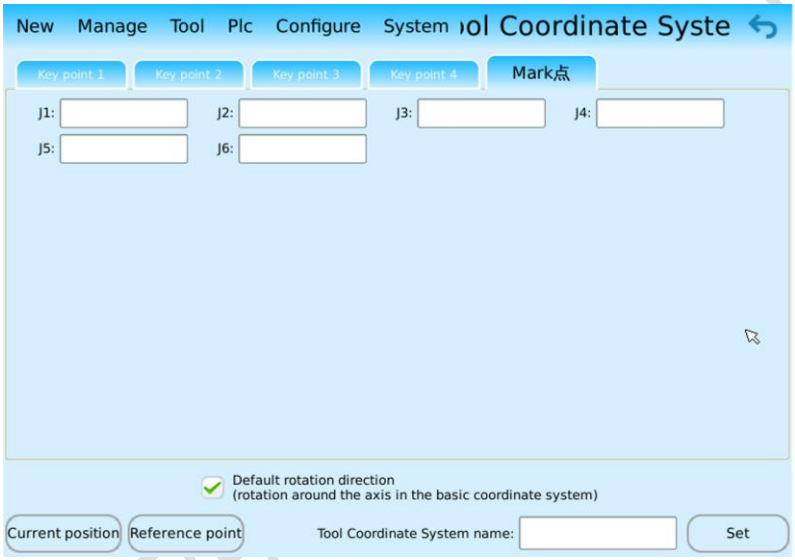


Figure 8 Creating mark points when creating a tool coordinate system

After replacing the tool, in the tool coordinate system interface of Betrun teach pendant, as shown in Figure 9, select the tool coordinate system and click the [revision] button to pop up the new mark point input box.

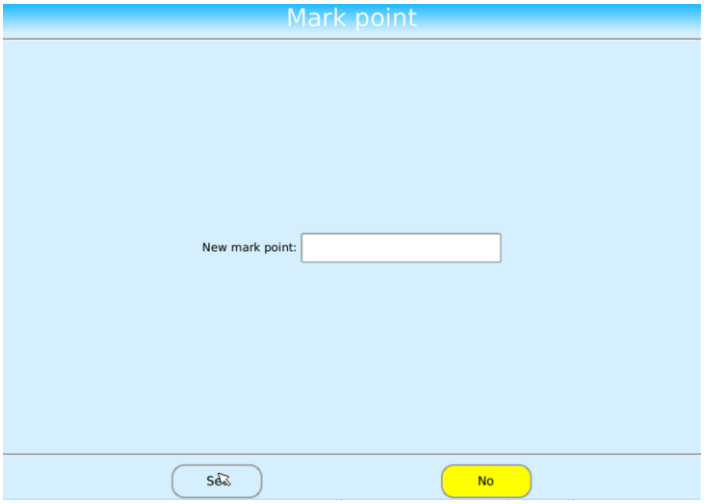


Figure Set new mark point

"New mark point" refers to the "point" read when the new tool reaches the same position on the worktable with the same attitude after tool replacement. Note that when creating this point, you also need to associate the tool coordinate system.

Click [Set] and the original tool coordinate system data will be quickly corrected to the new tool coordinate system data.

At the same time, all program points associated with the original tool coordinate system will be associated with the corrected new tool coordinate system, and its six joint data will be automatically corrected to keep the pose of TCP unchanged.

Appendix 1 Instruction description

(1) Movement instructions

Please refer to the motion instructions provided in Table 1 when programming the robot. When the tool attitude needs to be changed greatly, it is suggested to divide the motion trajectory into several curves for teaching.

Table 1 motion instructions

Instruction name	Description		
Curve movement	Function	Robot motion control.	
	Items	Movement type	Joint motion / linear motion / arc motion / circular motion
		Run to point	The end point of the curve movement Arc movement: it is also necessary to set the intermediate point (the center of the arc is on the vertical line connecting the start point and the end point, and the intermediate point is any point on the arc) Circular motion: it is also necessary to set the centre point and circular surface point (a sphere can be determined by the start point and centre point, and a circle can be determined by adding any point on the circular surface.)
		Speed	Range(0, 2000], unit: mm/s
		Smoothing ratio	The range is "0-100", "0" means there is no "smooth", and "smooth" is the largest when it is "100".
	Note	For arc movement, during debugging, if the current position is in the middle of the arc curve when pausing, you can't run directly from the current curve instruction when running again, but you need to return to the previous curve instruction to start running.	
Additional joint movement	Function	Additional axis motion control.	
	Items	Axis	EJ1/EJ2/……
		Control	Start/Stop
		Direction	Positive/Negative
		Speed	Unit: r/min
Forbidden additional joint	Function	Disable and enable additional axes.	
	Items	ENABLE EJ1/DISABLE EJ1；ENABLE EJ2/DISABLE EJ2	
Additional joint Zeroing parameters	Function	Return the additional axis to zero.	
	Items	Number	EJ1/EJ2/……

(2) Basic instructions

Basic instructions include: I / O instruction, register instruction, global variable instruction, speed control instruction, wait event instruction, other control instructions, etc. Please refer to Table 2.

HUNAN Betrun Robot Co., Ltd.

Table 2 Basic instructions

Instruction name	Description		
Output	Function	Change the state of the output port by assignment.	
	Items	Output port number	Y0-Y17
		Output level value	0 indicates low level and 1 indicates high level.
Pulse output	Function	Make the output port output in pulse form by assignment.	
	Items	Output port number	Y0 -Y17
		Polarity	Positive/Negative
		Pulse width	Unit: ms
Analog quantity	Function	Sets the analog value of the channel.	
	Items	Channel number	0-3, 4 channels in total.
		Data source	Variable (take value from global variable) / constant (input analog value directly)
		Analog value	[0, 12), (required when "constant" is selected as the data source)
Start input monitoring Stop input monitoring (Used in pairs)	Function	Monitor the level value of the input port, and assign the output port if there is any change.	
	Items	Input port number	X0-X19
		Port value	0 indicates low level and 1 indicates high level.
		Event relationship	And/
		Output port number	Y0-Y17
Global variable assignment and operation	Function	Global variable assignment operation.	
	Items	Number	General global variable 0-99
		Operation symbol	=, +=, -=, /=
		Assignment	Constant/Variable
Global variable self-increment	Function	The value of the global variable increases by 1.	
	Items	Number	General global variable 0-99
Global variable self-subtraction	Function	The value of the global variable is reduced by 1.	
	Items	Number	General global variable 0-99
Global variable reset	Function	The value of an interval global variable is reset to 0.	
	Items	Starting number	General global variable 0-99
		End number	General global variable 0-99
Register variable	Function	Assignment of register variables.	
	Items	Number	Register 0-499
		Operation symbol	=, +=, -=, /=
		Assignment	Constant/Variable
Digital welding machine operation	Function	Digital welding machine operation.	
	Items	Brand	Jinrui
		Function	Wire feeding / wire withdrawing / gas inspection
		Enable or not	Enable / Non enable
Wait for accurate position	Function	Wait for the robot to reach the specified target point within the distance deviation in the setting time.	
	Items	Target point ID	"Point" ID number

HUNAN Betrun Robot Co., Ltd.

		Distance deviation	Unit: mm
		Time	Unit: ms
Pipeline	Function	The curve instruction and non-curve instruction behind the pipeline instruction are executed in parallel for a period of time. For example, while the robot is moving, preheat the welding gun through the pipeline time. The pipeline duration setting cannot exceed the running duration of non-curve instructions executed in parallel.	
	Items	Pipeline duration	Unit: ms
PLC operation	Function	Operate the PLC relay.	
	Items	M number	0-499
		State	Valid/Invalid
Wait (event)	Function	Wait for the event.	
	Items	Event type	Input port / PLC / register variable / global variable / signal edge / signal edge hold
		logical relationship	And / Or
		Related parameters	Enter relevant information according to the event type.
Delay	Function	Delay.	
	Items	Parameter type	Variable / Constant
		Time	Unit: ms, (required when "constant" is selected)
		Variable number	0-99

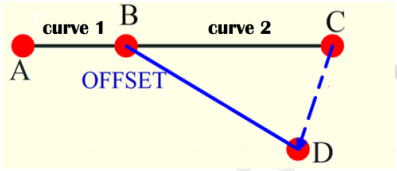
(3) Offset instructions

Aiming at the problem of low precision of teaching points in the process of manual teaching programming and the deviation of workpiece placement position in the actual operation process, the system provides a variety of offset instructions (Table 3) to realize the offset of robot motion trajectory during the operation of the program.

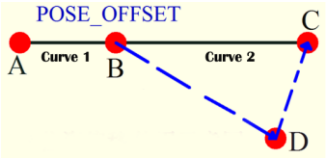
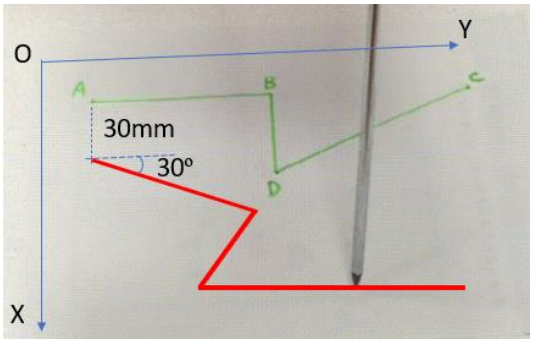
Table 3 Offset command

Instruction name	Description		
Offset	Function	In a coordinate system, set the offset to offset the "run to point" of the next curve instruction.	
	Items	Coordinate system	Base coordinate system / User coordinate system / Tool coordinate system
		Data source	Global variables / teaching parameters (input) (Set the offset of X, Y and Z axes and the offset of A, B and C attitude angles)
	Example	Procedure: Curve- Linear movement: Run to point(1-B); Speed=1000; Smoothing ratio=0 **OFFSET：Base coordinate system** Curve- Linear movement: Run to point(2-C); Speed=1000; Smoothing	

HUNAN Betrun Robot Co., Ltd.

		ratio=0 Explain: Curve 1: A-B; Curve 2: B-C; When the OFFSET instruction is not used, the complete trajectory is A-B-C; When the OFFSET instruction is used after curve 1, the movement of original curve 2 to point "C" will shift to point "D". Therefore, after using the OFFSET instruction, the running track of the program will change to A-B-D. The offset "amplitude" and "offset direction" of point d relative to point C are given by the global variables in the base coordinate system.  Global variable: "value" represents the offset amplitude, and "+" and "-" before "value" represent the offset direction.
Pose Offset	Function	In a coordinate system, by setting the offset, add a new curve from the current position, and the "run to point" is the offset point after the offset of the current position, and then run from the offset point to the "run to point" of the next curve instruction.
	Items	Coordinate system: Base coordinate system / User coordinate system / Tool coordinate system / Joint coordinate system Data source: Global variables / teaching parameters (input) (Set the offset of X, Y and Z axes and the offset of A, B and C attitude angles) Curve type: Joint movement /Linear movement Speed: Range(0, 2000], unit: mm/s
	Example	Procedure: Curve- Linear movement: Run to point(1-B); Speed=1000; Smoothing ratio=0 **POSE_OFFSET ： Base coordinate system;** Linear movement; Speed=1000 Curve- Linear movement: Run to point(2-C); Speed=1000; Smoothing ratio=0 Explain: Curve 1: A-B; Curve 2: B-C; When the POSE_OFFSET instruction is not used, the program running track is A-B-C; When the POSE_OFFSET instruction is used after curve 1 (i.e., the position of point B), the motion track of the point will first offset from the current point "B" to the position of point "D" and then run to the position of point "C". Therefore, after

HUNAN Betrun Robot Co., Ltd.

		using the POSE_OFFSET instruction, the operation track of the program will become A-B-D-C. The offset "amplitude" and "offset direction" of point D relative to point B are given by the global variables in the base coordinate system.  Global variable: "value" represents the offset amplitude, and "+" and "-" before "value" represent the offset direction.
Begin outline offset End outline offset	Function	Offset the outline formed by the curve between begin outline offset instruction and end outline offset instruction by setting the offset.
	Item	User coordinate system
	Example	Procedure: BEGIN_OUTLINE_OFFSET：h0 Curve- Linear movement: Run to point(1-A); Speed=100; Smoothing ratio=0 Curve- Linear movement: Run to point(1-B); Speed=100; Smoothing ratio=0 Curve- Linear movement: Run to point(1-C); Speed=100; Smoothing ratio=0 Curve- Linear movement: Run to point(1-D); Speed=100; Smoothing ratio=0 END_OUTLINE_OFFSET Explain: Before using this pair of instructions, you need to establish a corresponding "user coordinate system" for the "outline" and select it. The "direction" and "amplitude" of the contour offset are set by "assignment" to the corresponding "global variable". 
Begin range-offset End range-offset	Function	By setting the offset, all the run to points of the curve between " Begin range-offset" and " End range-offset" can be operated with the "offset" instruction in a certain coordinate system.

HUNAN BE... Ltd.

	Items	Coordinate system	Base coordinate system / User coordinate system
		Data source	Global variables / teaching parameters (input)

(4) Process instructions

In order to facilitate users to better use the Betrun control system, the system fixes some operation processes, and manual teaching programming is particularly cumbersome. Through "collection" to form a "process package", users can easily complete the original complex and cumbersome work. To invoke the process package in the basic program, we need to use the process instruction, and use the method in detail in the part III of the instruction.

(5) Function instructions

For some complex operation processes, the Betrun control system is packaged in advance, and the packaging is successful. Users only need to set parameters and can call them in the basic program. See Part II of the manual for the call of function instructions.

(6) Logic instructions

Many logic judgment, loop and jump operations are often involved in the application. The logic instructions provided by Betrun control system are shown in Table 4.

Table 4 Logic instructions

Instruction name	Description
IF	Conditional judgment
ELSE_IF	Conditional branch judgment
END_IF	End of condition judgment
WHILE	Loop
CONTINUE	Exit this cycle and judge the conditions of the next cycle
BREAK	Exit loop
END_WHILE	End loop
JMP	Jump
LABEL	Label
COND_UNTIL_BEGIN	Interval interrupt start
CON_UNTIL_END	Interval interrupt end

HUNAN Betrun Robot Co., Ltd.

Appendix 2 Definition of global variables

Address	Data meaning
GV0-GV99	**General**
GV100-GV109	Number of stacks already placed in 10 pallets in palletizing process
GV110-GV199	Exclusive reservation
GV200	**X offset (for "offset", "section offset", "outline offset" instructions)**
GV201	**Y offset (for "offset", "section offset", "outline offset" instructions)**
GV202	**Z offset (for "offset", "section offset" instructions)**
GV203	**Attitude angle A offset (for "offset", "section offset" instructions)**
GV204	**Attitude angle B offset (for "offset", "section offset" instructions)**
GV205	**Attitude angle C offset (for "offset")/Rotation (for "outline offset")**
GV206-GV208	**XYZ data in Base Coordinate System**
GV209-GV211	**ABC data in Base Coordinate System**
GV212-GV217	**6 joints offset / X-Y-Z-A-B-C offsets in base coordinate system (for "position offset" instruction)**
GV218	Current number of layers in palletizing process
GV219	Serial number of the position to be stacked in the current layer in palletizing process
GV220-GV225	Special register for program control
GV226-GV229	Exclusive reservation
GV230	**X offset (for "offset", "section offset", "- Intermediate point)**
GV231	**Y offset (for "offset", "section offset", "- Intermediate point)**
GV232	**Z offset (for "offset", "section offset", "- Intermediate point)**
GV233-GV235	Rotation angle (for "offset", "section offset", "- Intermediate point)
GV236-GV240	**Rotation angle of the additional joints 0-4 (Additional joint offset instruction)**
GV241	User coordinate system ID
GV242-GV244	XYZ coordinate value in user coordinate system
GV245	The tool coordinate system ID used to display the robot end coordinates and attitude, and the pose offset function. The default value is 255, indicating no tools.
GV246-GV256	Current joint angle of each joint
GV257_L	Completion flag of arc swing process parameter setting
GV257_H	Completion flag of welding machine process parameter setting
GV258-GV263	Arc swing process parameters (mode < 0-z, 1-arc, > - frequency – amplitude - left stay time -right stay time - arc swing radius)
GV264-GV265	Welding process parameters (welding current / wire feeding speed - welding voltage)

HUNAN Betrun Robot Co., Ltd.

GV266-GV279	Exclusive reservation
GV280_L	Mode (0-manual, 1-automatic)
GV280_H	Running status (0-stop, 1-running, 2-pause)
GV281	Alarm category (represented by bits, 0-no such alarm, 1-have such alarm) (B0 emergency stop alarm, B1 servo alarm, B2 brake abnormal alarm, B3 algorithm alarm, B4 encoder angle acquisition alarm)
GV282	Running times
GV283	External encoder task enable flag (read / write)
GV284	Zero crossing trigger port of external encoder (read / write)
GV285	External mechanism reduction ratio (read only)
GV286	Number of external encoder lines (read / write)
GV287	Current value of external encoder (read only)
GV288	System date: Year (2B) + Month (1B) + Day (1B)
GV289	System time: Hour (2B) + Minute (1B) + Second (1B)
GV290	System millisecond counter
GV291	Incremental motion command return status (0 - initial value, 0xff - command execution completed, others: error status)
GV292-GV299	Exclusive reservation
GV300-GV499	General, Data loss after power failure
GV500-GV699	General, No data loss after power failure
GV700-GV705	Special, palletizing frame curve skip flag（0-don't skip, 1-skip）
GV706	Special, palletizing transition point (entry) modification mark（0 - do not modify, 1 - modify）
GV707	Special, modified palletizing transition point (entry) ID
GV708	Special, palletizing prepare to release point modification mark（0 - do not modify, 1 - modify）
GV709	Special, modified palletizing prepare to release point ID
GV710	Special, palletizing release point modification mark（0 - do not modify, 1 - modify）
GV711	Special, modified palletizing release point ID
GV712	Special, palletizing leaving point modification mark（0 - do not modify, 1 - modify）
GV713	Dedicated, modified palletizing leaving point ID
GV714	Special, palletizing transition point (away) modification mark（0 - do not modify, 1 - modify）
GV715	Special, modified palletizing transition point (away) ID
GV716	Special, Hongtu packaging process, current quantity

HUNAN Betrun Robot Co., Ltd.

GV717	Special, Hongtu packaging process, total production
GV718-GV799	Exclusive reservation

Note: gv700-gv799 is special global variables with no data loss after power failure.

Appendix 3 Definition of PLC components

Number	Component type	Component number	Component range
1	Input relay (X)	20 + 40 (expansion board)	X0-X19; X20-X59 (expansion board)
2	Output relay (Y)	18+36 (expansion board)	Y0-Y17; Y18-Y53 (expansion board)
3	Status relay (S)	100	S0-S99
4	Alarm relay (A)	50	A0-A49
5	Universal relay (M)	500	M0-M499
6	Counter (C)	50	C0-C49
7	Hotkey relay (F)	30	F0-F16; F17-F29 (reserved)
8	Interference relay (I)	48	I0-I47
9	User defined alarm relay (W)	60	W0-W59
10	Universal 10ms timer (TGT)	10	TGT0-TGT9
11	Universal 100ms timer (TGH)	10	TGH0-TGH9
12	Universal 1s timer (TGS)	10	TGS0-TGS9
13	Cumulative 10ms timer (TAT)	10	TAT0-TAT9
14	Cumulative 100ms timer (TAH)	10	TAT0-TAT9
15	Cumulative 1s timer (TAS)	10	TAS0-TAS9

The specific description of PLC elements is as follows:

1. Input relay

Number	Meaning	Note
X0-X19	System onboard input INPUT0-INPUT19	
X20-X59	Expansion board input INPUT0-INPUT39	

2. Output relay

Number	Meaning	Note
Y0-Y7	System onboard output OUTPUT0-OUTPUT7	Relay output
Y8-Y17	System onboard output OUTPUT8-OUTPUT17	OC output
Y18-Y33	Expansion board output OUTPUT0-OUTPUT15	Relay output
Y34-Y53	Expansion board output OUTPUT16-OUTPUT35	OC output

3. Status relay

HUNAN Betrun Robot Co., Ltd.

Program status and interaction for interaction with PLC.

Number	Meaning	Note
S0	Run	
S1	Stop	
S2	Pause	
S3	Manual mode	
S4	Automatic mode	
S5	Single step motion	
S6	Single motion	
S7	Loop motion	
S8	Servo on	
S9	Servo off	
S10	Additional axis 0 position control	
S11	Additional axis 0 speed control	
S12	Additional axis 0 tracking control	
S13	Additional axis 1 position control	
S14	Additional axis 1 speed control	
S15	Additional axis 1 tracking control	
S16-S69	Y0-53 output status during pause	
S70	There are continuing events	
S71	Program running completed	
S72-S99	Reserve	

4. Alarm relay

The state when the controller gives an alarm.

Number	Meaning	Note
A0	There's an alarm.	
A1	Emergency stop alarm	
A2	Servo alarm	
A3	Joint overrun alarm	
A4	Algorithm operation error alarm	
A5	Encoder angle update error alarm	
A6	Failed to get external encoder angle	

HUNAN Borun Robot Co., Ltd.

A7	Tracking failed	
A8	Failed to read stacking process file	
A9-A49	Reserve	

5. Universal relay

Input / output interaction area.

Number	Meaning	Note
M0-499	User defined	

6. Counter

Number	Meaning	Note
C0-C49	User defined	

7. Hotkey relay

Start the function corresponding to the hotkey.

Number	Meaning	Note
F0	Emergency stop	
F1	Continue to run	
F2	Pause the program	
F3	Stop the program	
F4	Switch to manual mode	
F5	Switch to automatic mode	
F6	Run to zero – all the joints	
F7	Run to zero – joint 1	
F8	Run to zero – joint 2	
F9	Run to zero – joint 3	
F10	Run to zero – joint 4	
F11	Run to zero – joint 5	
F12	Run to zero – joint 16	
F13	Run to zero – joint 7	
F14	Run to zero – joint 8	
F15	Run to zero – joint 9	
F16	Run to zero – joint 10	
F17	Run to zero – joint 11	

HUNAN Betrun Robot Co., Ltd.

F18	Run to safety zone	
F19	Run the program	The parameter is the program ID
F20-F29	Reserve	

8. Interference relay

Number	Meaning	Note
I0-I19	Point interference output	Interference point 0 - interference point 19
I20-I39	Spatial interference output	Space interference 0 - Space interference 19
I40-I47	Axis interference output	Axis interference 0 - axis interference 7

9. User defined alarm

Number	Meaning	Note
W0-W59	User defined alarm input	The corresponding alarm prompt box pops up on the teach pendant interface

10. Timer

Number	Meaning	Note
TGT0-TGT9	Universal 10ms timer	
TGH0-TGH9	Universal 100ms timer	
TGS0-TGS9	Universal 1s timer	
TAT0-TAT9	Cumulative 10ms timer	
TAH0-TAH9	Cumulative 100ms timer	
TAS0-TAS9	Cumulative 1s timer	

Betrun Robot Co., Ltd.