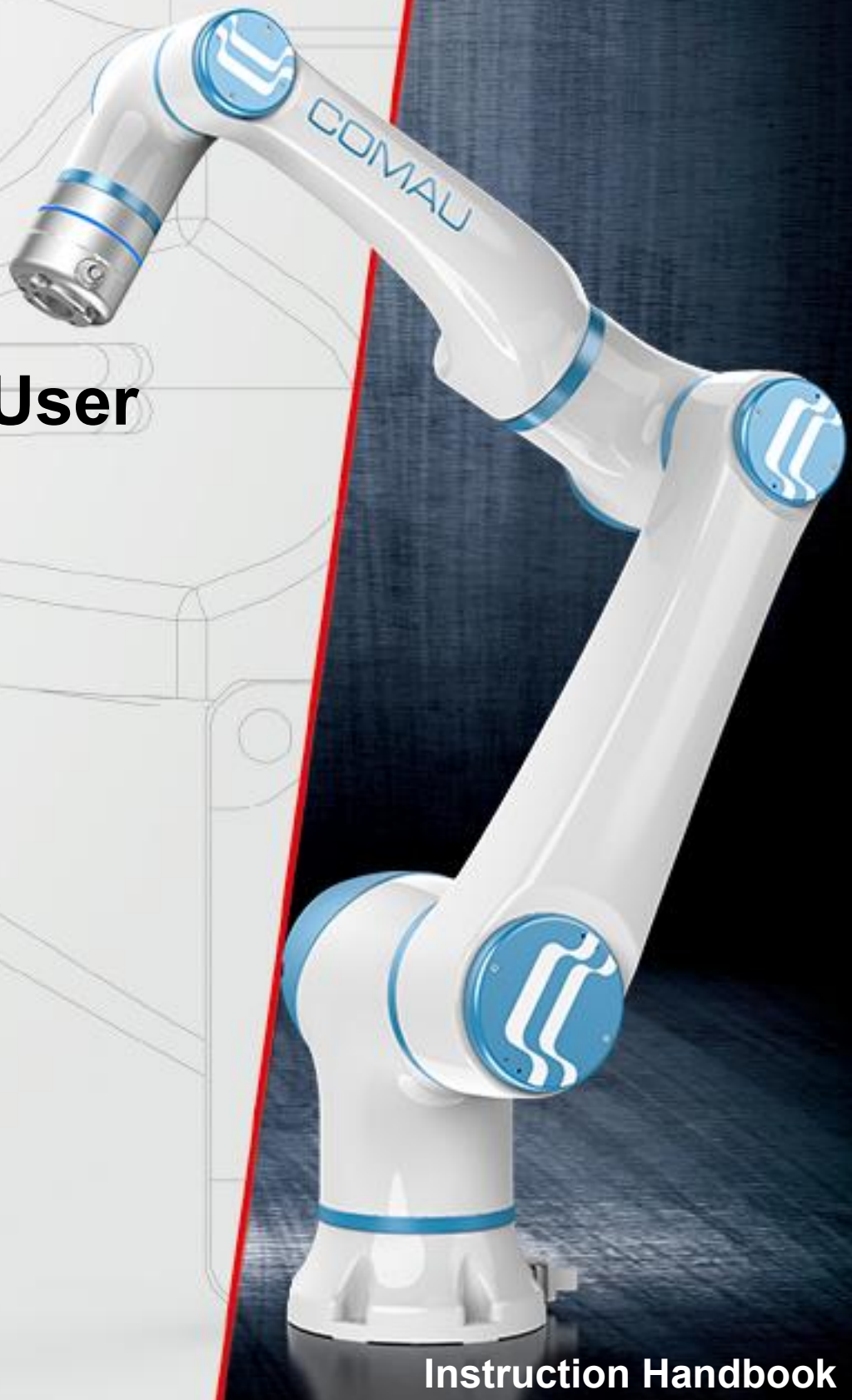
MyCo Robot Series



Software User Manual

CR00758383-en_00/2025.12

Instruction Handbook

The contents of this handbook are the property of COMAU S.p.A.

Any reproduction, even partial, is prohibited without prior written authorization from COMAU S.p.A.

COMAU reserves the right to modify, without notice, the functions of the product presented in this handbook.

Copyright © 2020-2025 by COMAU - Published 12/2025

Contents

Preface 5
How to Use This Manual 6
Modification History 7
Software Introduction 8
1.1 Login Method 9
1.2 User Login 9
Operation Interface 10
2.1 Overview of the Operation Interface 11
2.2 Status Indicator 12
2.3 Stack Menu 17
Operating Modes and Three-Position Button 19
3.1 Operating Modes 20
3.2 The Three-Position Button 21
Jog 24
4.1 Understanding the Interface 25
4.2 Point List 31
4.3 Move to Home 34
4.4 Z-axis Align 35
Configuration 37
5.1 TCP Setting 38
5.2 Payload Setting 45
5.3 Base Mounting Angle 52
5.4 I/O Setting 53
5.5 Safety Limits 58
5.6 User Coordinate System 64
5.7 Safety Range 69
5.8 Communication (Modbus) 74
5.9 Tracking Conveyor 78
5.10 Force Sensor 81
5.11 Brake Operation 87
5.12 Boot Configuration 88

5.13 Home 97
5.14 Basic Free Drive Adjustment 99
Program 101
6.1 Introductory Program 102
6.2 Understanding the Interface 102
6.3 Motion Command 108
6.4 Logic Command 120
6.5 Control Command 130
6.6 Template 133
6.7 Thread 140
6.8 Variable 141
6.9 Debugging and Running 143
6.10 Program Example 145
Running 151
7.1 Initializing Robot 152
7.2 Selecting the Program 153
7.3 Running Status 153
7.4 Page Introduction 154
Path 156
8.1 Page Introduction 157
8.2 New Path 158
8.3 Path List 160
Monitoring 161
9.1 Electrical Control Box I/O 162
9.2 End I/O 163
9.3 Extended Modbus 164
9.4 Local Modbus Slave 164
9.5 Robot Status 167
9.6 Log 169
Management 170
10.1 Authorization Management 171
10.2 Network Configuration 172

10.3 System Update 173
10.4 User Management 174
CMApp 180
11.1 App 181
Exception Handling 183
12.1 Protective Stop 184
12.2 Emergency Stop 184
12.3 Safe Collision Stop 185
12.4 Trigger Safeguard Monitoring 185
12.5 Beyond Safety Range 187
12.6 Singularity Point States 187
End-effector Connection 194
Virtual Robot 196
14.1 VMWare Virtual Machine 197
14.2 Simulation Control Box 209

Preface

This chapter contains the following topics:

– How to Use This Manual

How to Use This Manual

This manual is prepared for operators of MyCo series robots. If the developer intends to carry out large-scale programming, certain Python programming knowledge is required. The manual will guide the operator in the use of the robot from the following aspects:

- Software control: Guide operators to install and run the software.
- Parameter configuration: Introduce the basic configuration and related functions of the robot.
- Robot programming: Introduce the robot programming commands in detail.

Glossary of Terms:

Terms	Explanations
Robot Arm	Usually refers to the joints, links, and electrical control boxes of the robot.
Robot	A complete equipment consisting of robot arm and electrical control box.
Controller Software Operation Interface	The graphical interfaced for operating the robot.
System Software	A collection of operation interfaces and back-end services.
Program	A collection of commands written by a programming module.
Program Files	A file that stores program commands.
User Coordinate System (UCS)	User-defined reference coordinate system.
Tool Center Point (TCP)	The coordinate system centered on the tool at the robot arm's end-effector.
I/O	Input/output, including CI/CO, DI/DO, AI/AO, EndDI/EndDO, and EndAI/EndAO.
Script	The Script module inserted in programming.
Waypoint	The robot arm target position used in the program.
Point	The specific position information of the robot arm recorded and stored.

Modification History

Date	Edition of the Handbook	Contents
2025-12	00/2025.12	First release of the handbook
2026-3	00/2025.12	**Add** Programming - Control Command - ForceControl **Modified** I/O Setting - Configurable Input I/O I/O Setting - Configurable Output I/O Programming - Motion Command - Move Programming - Template Command - Tracking Monitoring - EndI/O - Universal Input, Output and Analog Input

Software Introduction

This chapter contains the following topics:

– Login Method
– User Login

1.1 Login Method

Comau S.p.A Teach Pendent has an operating software that enables manual operation, programming, parameter configuration and user monitoring of robots. The user can control the robot through the function buttons on the operation interface, and enable the robot to move along the user-specified path and complete the expected actions. the teach pendant serves as the robot’ s display and operation terminal.

There are two ways to access the controller software interface and operate the robot:

- By using the HuayanRobot teach pendant.

  Users can directly operate the robot through the teach pendant. In this case, the teach pendant serves as the robot’s display and operation terminal.

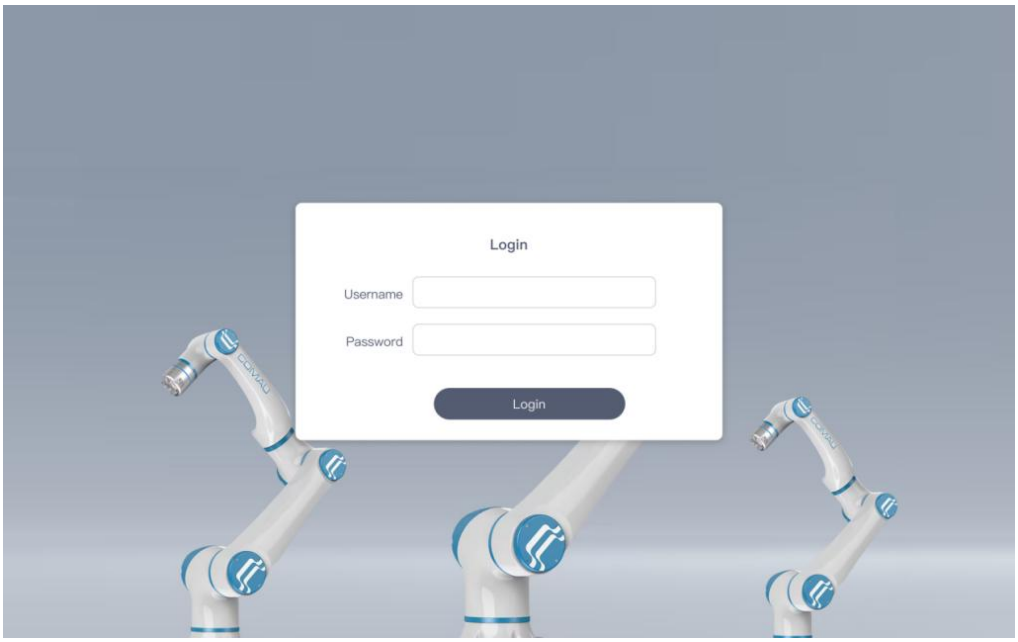
- By connecting through external terminals

  External terminals include personal computers and tablets. PCs, tablets, and other terminal devices can connect to the robot via TCP using a browser/server (B/S) architecture. For connection details, see Chapter 13 Terminal Connection.

1.2 User Login

After starting the software and accessing the login page, users with login permission can enter their usernames and passwords to log in.

Fig. 1.2.1 User Login



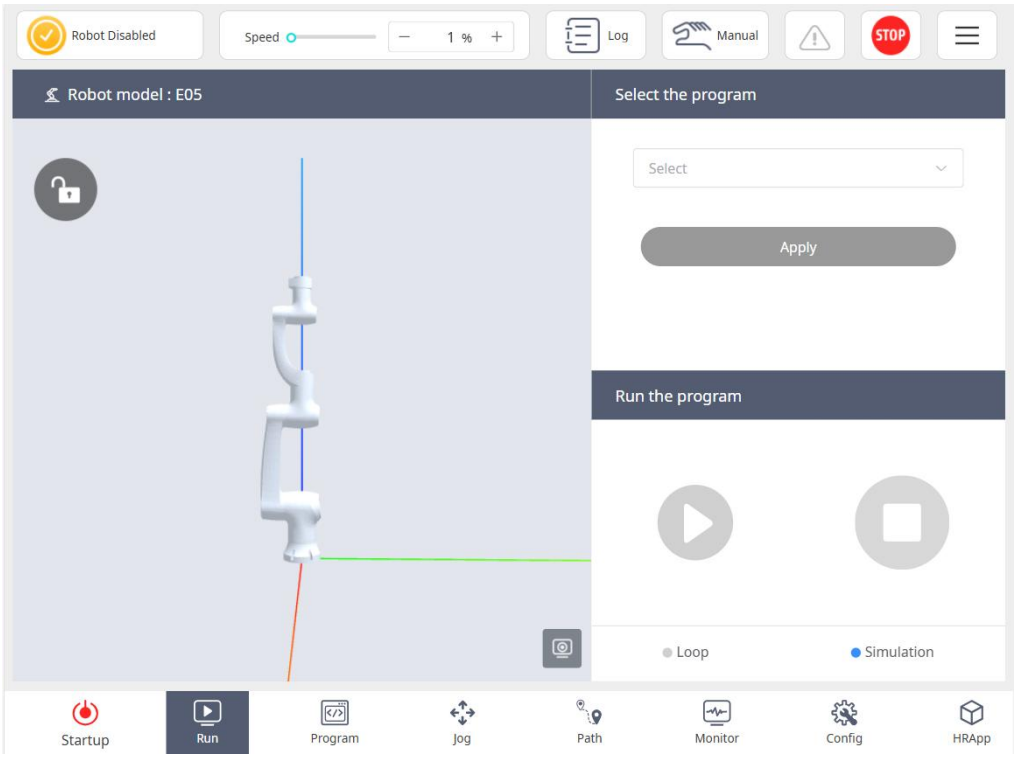
Operation Interface

This chapter contains the following topics:

– Overview of the Operation Interface
– Status Indicator
– Stack Menu

2.1 Overview of the Operation Interface

Fig. 2.1.1 Robot Operation Interface



Main components on the robot control interface:

1. Menu Bar

2. Robot ON/OFF

3. Action Bar

2.1.1 Menu Bar



Table 2.1.1.1 Operation Function Interface

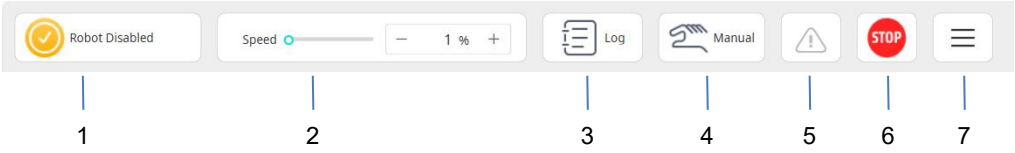
SN	Name	Function	Reference
1	Run	Controller software operation homepage	7 Run
2	Program	Control of robot arm motion through programming	6 Program
3	Jog	Controlling the pose and motion of the robot arm	4 Jog

4	Path	Establishing the continuous path motion of robot arm	8 Path
5	Monitor	Monitoring the robot status	9 Monitor
6	Config	Configuring the robot	5 Config
7	APP	Comau plug-in platform	11 APP

2.1.2 Status Indicator

SN	Name	Function	Reference
1	Status Indicator	Viewing the startup status of the robot Turning the robotic system ON or OFF	2.2 Status Indicator

2.1.3 Action Bar



SN	Name	Function	Reference
1	Robot Status	Viewing the robot running status	
2	Speed	Debugging current robot operating speed	
3	Log	Clicking to access the log and view the robot log	
4	Mode	Automatic/manual mode switching	3 Manual/Automatic Modes
5	Alarm	Prompt of current error	
6	Stop	Emergency stop	12.2 Emergency Stop
7	Stack Menu	Management Language About Log Out Restart Shutdown	2.3 Stack Menu

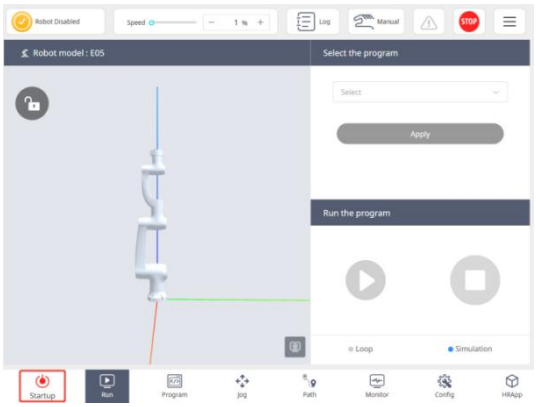
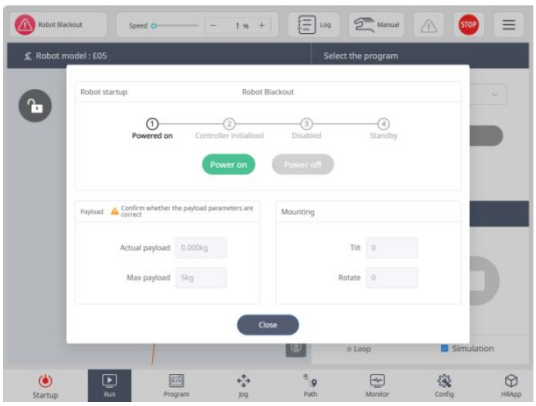
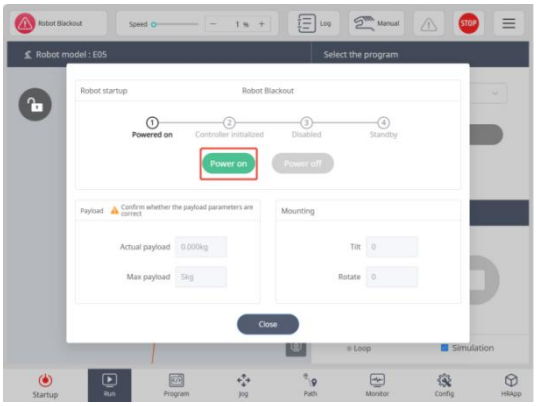
2.2 Status Indicator

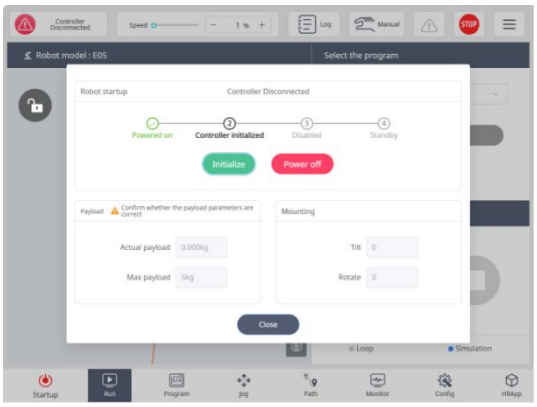
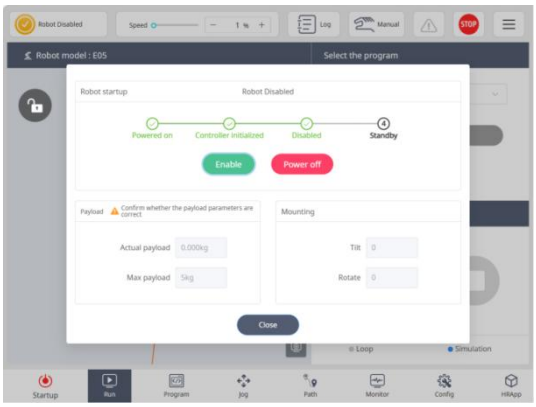
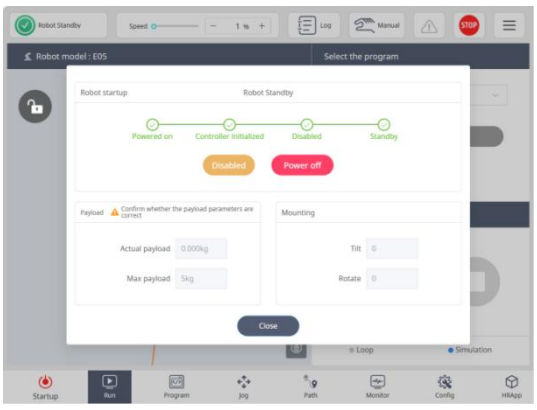
The status indicator indicates the current status of the robot arm: the color yellow

indicates the disabled state, green indicates the enabled state, purple indicates the running state, blue indicates the free drive state, and red indicates the uninitialized state or abnormal state.

2.2.1 Robotic System ON/OFF

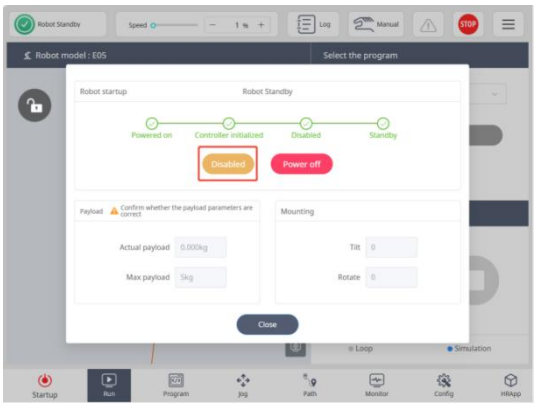
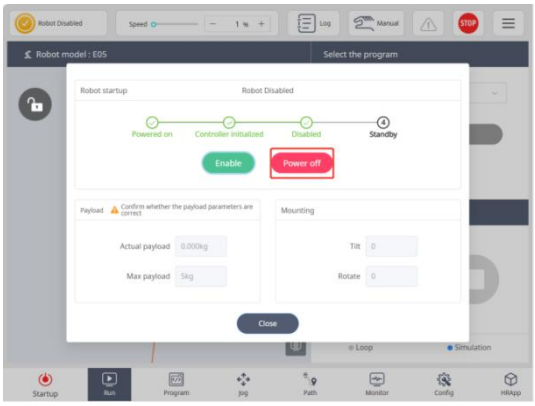
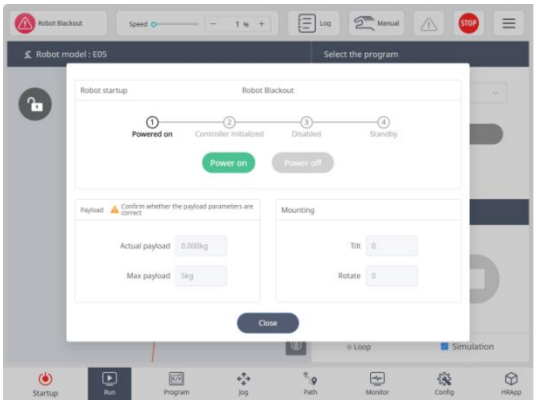
The steps to turn on the robot are as follows:

SN	Operation	Examples
1	Enter the controller software operation interface, connect the robot, and log in to the controller;	
2	After logging in, the robot is still not electrified; power on and enable the robot motion;	
3	Click the "Robot Blackout" indicator button to access the Power on popup;	
4	After clicking "Power on", the robot is electrified, but the joint brakes are not released, and the robot arm motion is not enabled;	

5	Click "Initialize" to initialize the controller;	
6	After successful initialization, the robot will be stably in the disabled state. Before clicking the "Enable" button, be sure to confirm whether the robot settings information (current payload, base mounting angle position, TCP and other parameters) is correct. The current payload and mounting angle are displayed below the "Enable" button for verification;	
7	Click the "Enable" button, the robot arm is powered on, and the status indicator turns green, showing "Standby", and the indicating enabling successful;	

Note: Before enabling the robot, please carefully check the robot configurations. If configured incorrectly, the robot arm or control box may not work properly, and personnel or equipment hazards may even occur. Also, please confirm whether the pose of the robot arm is consistent with the actual pose; otherwise, it is necessary to check the match between the robot arm and the electrical control box.

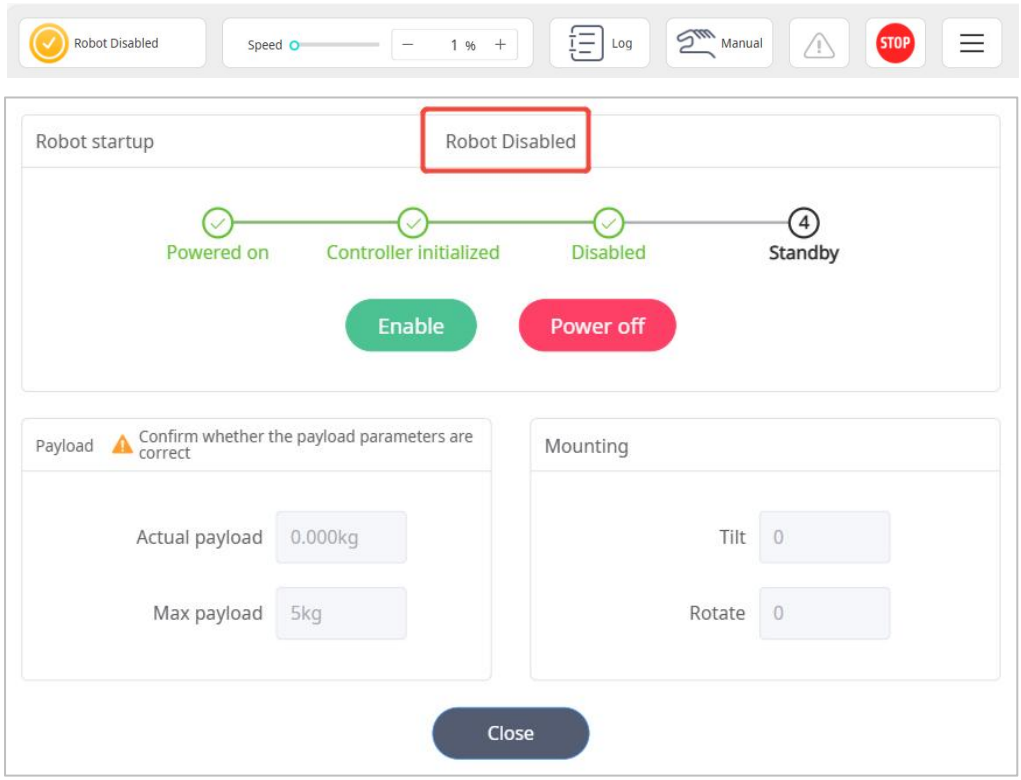
The steps to turn off the robot are as follows:

SN	Operation	Examples
1	Click the "Disabled" button; and when the status indicator turns yellow or gray, it indicates that the robot is disabled;	
2	Click "Power off" until the status indicator turns red, and the robot is powered off;	
3	The current status indicates successful blackout;	

2.2.1.1 Robot State Description

During the entire process of the robot from startup to normal operation, the page will display the status prompts in real time:

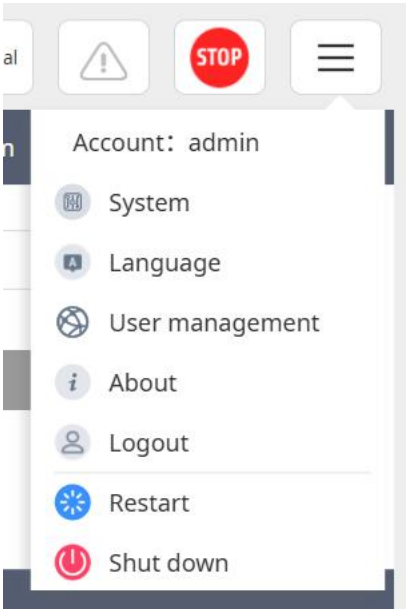
Fig. 2.2.1.1.1 Page Prompt



- After Successfully Starting the Teach Pendant: it prompts "The robotic arm is powered off". If the robot fails to power on, the corresponding error and cause will be prompted. Please check whether there are hardware failures according to the prompts;
- If Initialization Fails After Power-on: it prompts "The controller is not initialized", indicating that the current robot has been powered on but not initialized;
- If Not Enabled After Initialization: it prompts "Disabled", indicating that it is currently initialized;
- After Enabling: it prompts "Standby", indicating that the robot arm is ready for the next operation;
- Free Drive: "Free drive" means that the robot arm is in free drive mode, and within a safe range, the robot arm can be freely dragged and moved to the target position;
- Running: "Program Running" indicates that the robot arm is executing the program. Meanwhile, no settings or programming is allowed. Click "Pause" or "Stop", and the robot will pause or stop the running program;

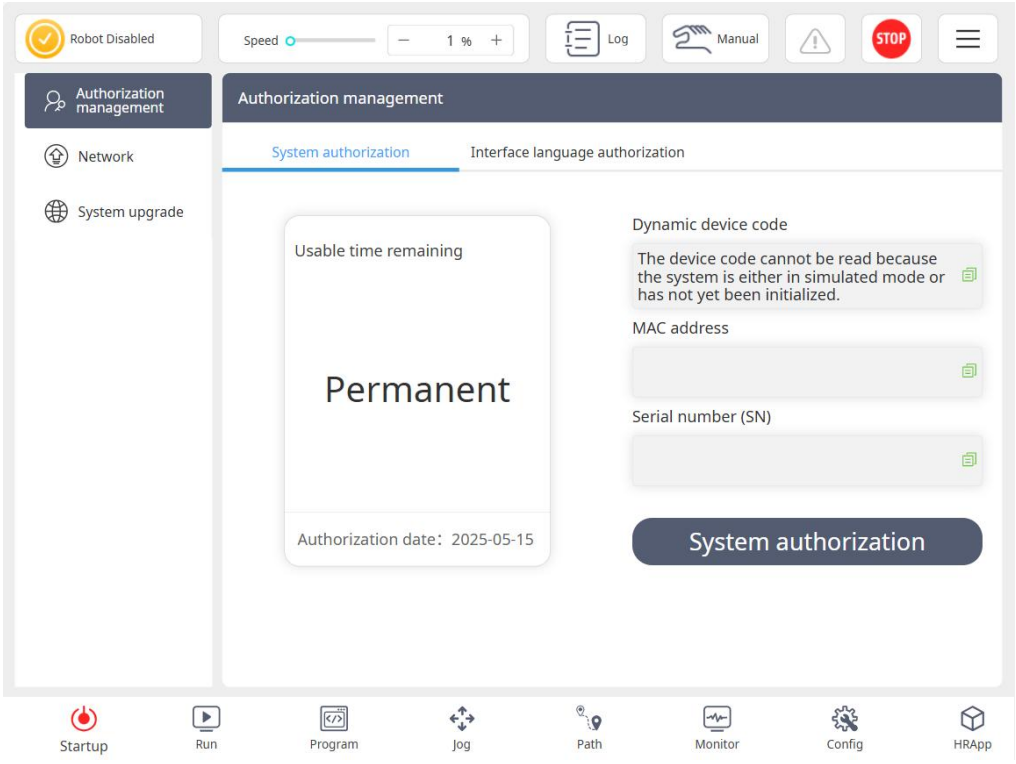
2.3 Stack Menu

Fig. 2.3.1 Stack Menu



– **Management: Click "Admin" in the Stack Menu to access the System page,** where authorization management, network configuration, system update and user management operations are available. Refer to Chapter 9 Management for details;

Fig. 2.3.2 Administration



– **Language:** Users can switch the language of the software system. The following languages are available: Chinese, English, Korean, German, Japanese, Spanish, Russian, Polish, French, Italian and Czech;

- **About:** Viewing the version information of the controller software system;
- **User Logout:** If the account is currently logged in, the user can click "Log out" to log out and further log in in the menu;
- **Restart:** The software system will perform a restart operation. If there are currently unsaved files, a prompt will appear to remind the user to save the files. Users can choose to restart after saving the files, or restart directly without saving the files, or cancel the restart operation;
- **Shutdown:** To turn off the power supply of the teach pendant, electrical control box and robot arm. If there are currently unsaved files, a prompt will appear to remind the user to save the files. Users can choose to restart after saving the files, or restart directly without saving the files, or cancel the restart operation;

Operating Modes and Three-Position Button

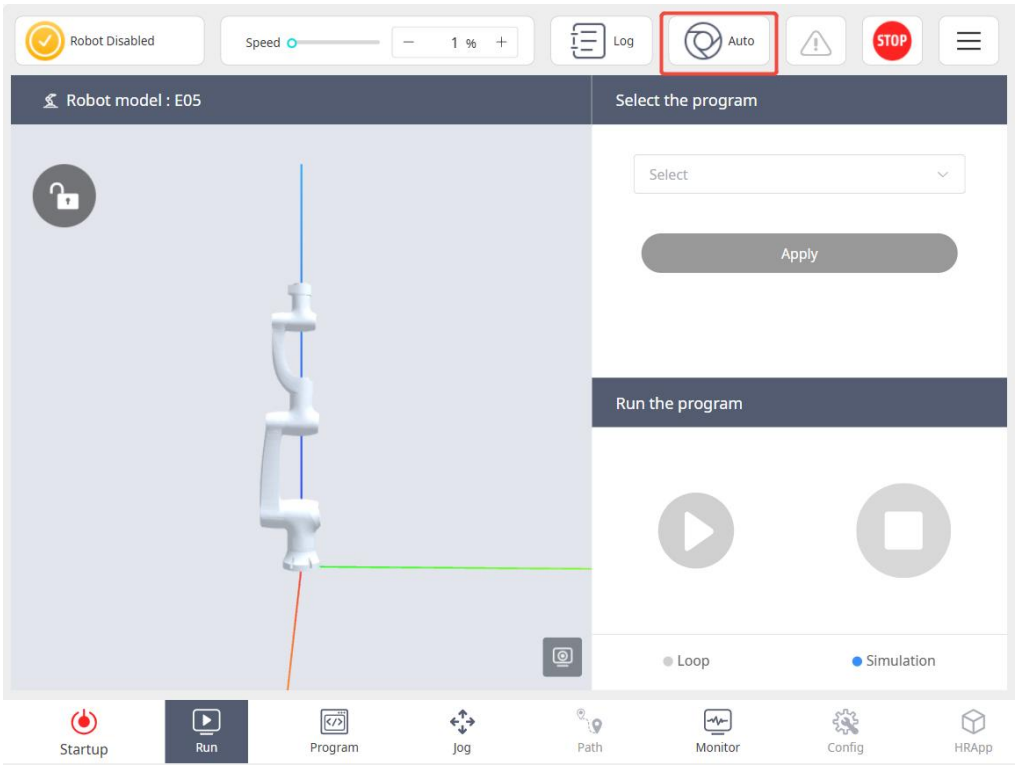
This chapter contains the following topics:

– Operating Modes
– The Three-Position Button

3.1 Operating Modes

3.1.1 Auto Mode

Fig. 3.1.1.1 Auto Mode Interface



The auto mode refers to an user operation mode. When enabled, only the saved run-able program, robot arm jog functions and robot monitoring functions can be executed. Editing the configuration and the run-able programs of the robot is not allowed.

3.1.2 Manual Mode

Fig. 3.1.2.1 Manual Mode Interface



When the system is in Manual mode and the robot is enabled, the user can perform all operations available in the system. To improve operational safety, it is recommended to use the robot together with the three-position button.

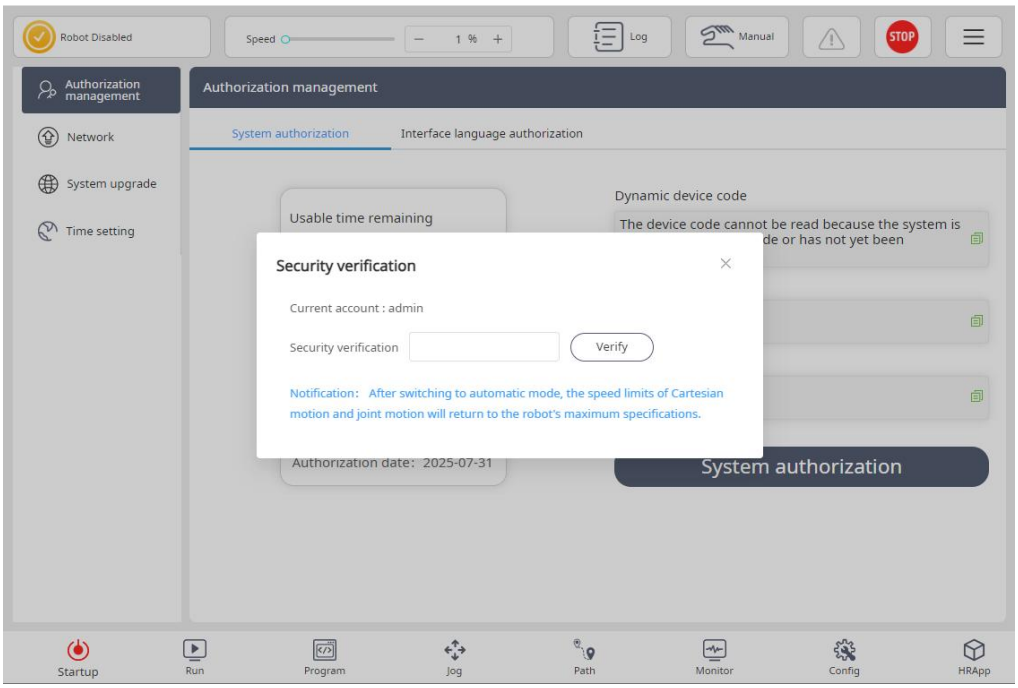
In manual mode, the central speed limit of the currently used TCP is as follows:

1. For MoveL, MoveC, MoveJ, MovePathL, SpeedJ, and SpeedL, the TCP center speed is limited to 250 mm/s.
2. ServoP, ServoJ, MovePathJ, and MovePathJOL are not subject to this limit.
3. For conveyor tracking, position tracking, and force-controlled search motions, only the speed of the primary motion is limited; the resultant

composite speed is not limited.

3.1.3 Mode Switching

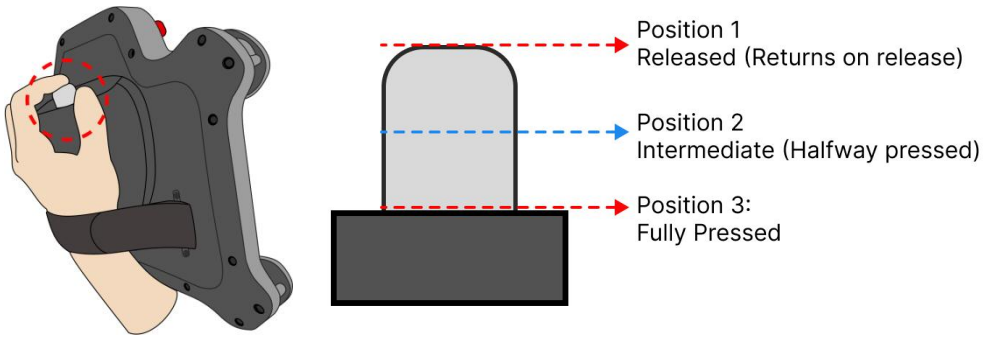
Fig. 3.1.3.1 Mode Switching



Note: When switching between Auto mode and Manual mode, password verification is required.

3.2 The Three-Position Button

Fig. 3.2.1 The Three-Position Button



The three-position button is located on the back of the teach pendant. It is a human – machine interaction safety device with three positions, designed to prevent injury caused by unintended operations during manual teaching or debugging.

The button corresponds to different working states depending on the pressing depth, thereby ensuring safe control of robot motion.

– Second Position (Intermediate Position)

When the button is pressed to and held in the intermediate position, the robot

can enter the enabled state or higher states, such as motion or program execution, allowing safe teaching or debugging operations.

– **First / Third Position (Released / Fully Pressed)**

When the button is released or fully pressed to the bottom position, the robot can remain only in the disabled state at most.

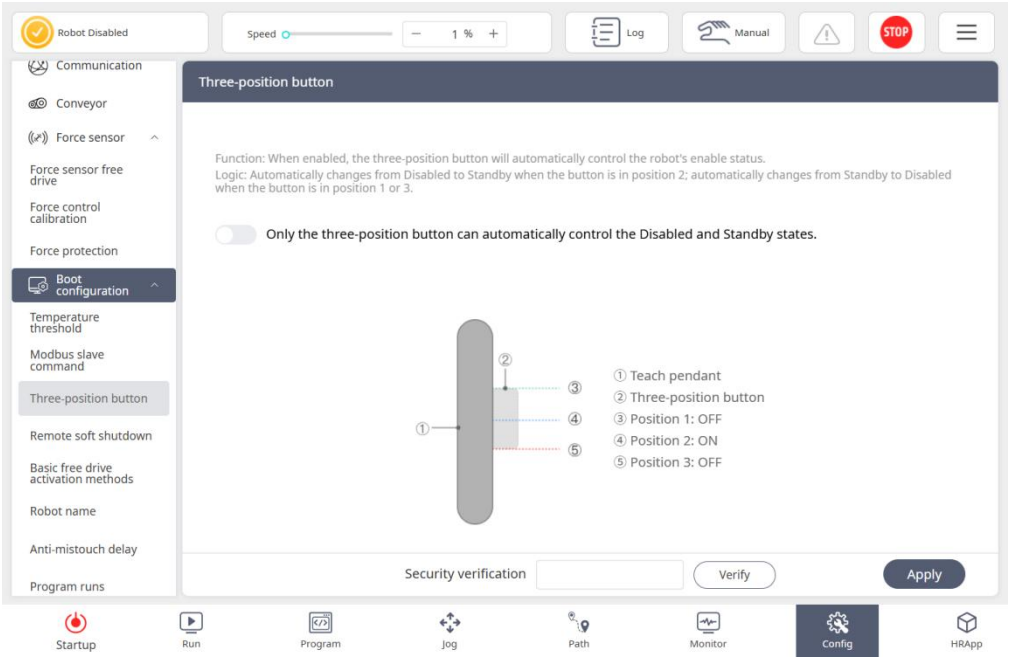
If the button is switched from the second position to the first or third position while the robot is enabled or in a higher state, the system immediately stops robot motion or program execution and transitions the robot to the disabled state.

– The three-position button is effective only in Manual mode and is used to ensure the safety of personnel and equipment during teaching and debugging. In Auto mode, this button is inactive.

– In the current software version, the functionality of the three-position button is enabled by default to provide out-of-the-box safety protection and to avoid risks caused by configuration omissions.

3.2.1 Automatic Enable/Disable Function of the Three-Position Button

Fig. 3.2.1.1 The Three-Position Button



– To balance operational safety and convenience, the system provides an automatic enable/disable function controlled by the three-position button.

– When this function is enabled, the system automatically manages the robot’s enabled state according to the physical position of the three-position button. In this case, the enable/disable button on the software interface becomes unavailable.

– When the three-position button is in the intermediate position, the robot automatically changes from the disabled state to the enabled state, without

requiring any additional operation on the software enable switch.

Note: The three-position button is a safety protection device used in the robot system. During teaching, debugging, and equipment maintenance, the user should ensure that the button remains in the intermediate position so that the safety control logic remains effective.

Jog

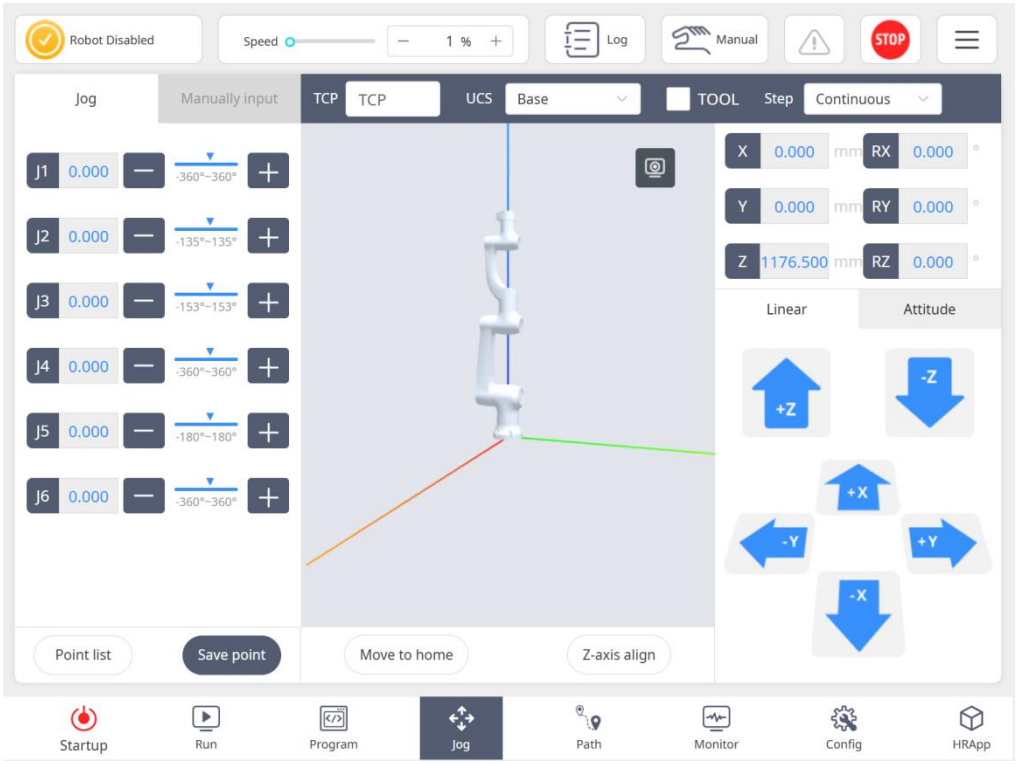
This chapter contains the following topics:

– Understanding the Interface
– Point List
– Move to Home
– Z-axis Aligne

4.1 Understanding the Interface

This section mainly introduces how to control the robot arm motion to the specified target position. This chapter describes how to move the robot arm to a specified target position using the Jog interface. The jog teaching interface is intended for operating the basic motions of the robot arm, The Jog interface is used for basic robot motion operations, including joint motion, Cartesian motion, and zero-force teaching.

Fig. 4.1 Jog Interface



4.1.1 TOOL Mode

Fig. 4.1.2.1 TOOL Mode



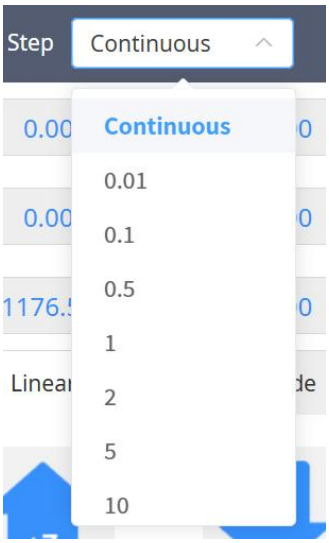
When this checkbox is ticked, it enters the TOOL mode. In TOOL mode, the robot arm moves based on the currently selected TOOL coordinate system.

4.1.2 Step

Fig. 4.1.2.1 Step Button



Fig. 4.1.2.2 drop-down menu Display



The step value is used to control incremental robot motion. If any option other than “Continuous” is selected, the robot pose can only be adjusted step by step. For example, if the step is 0.1, click J1- once, the value change is -0.100; and click twice, the value changes by -0.200.

The step range is 0.01 ~ 50.00.

Note:
- In joint motion mode, the step unit is degrees (°).
- In Cartesian motion mode, the position step unit is millimeters (mm), and the orientation step unit is degrees (°)

4.1.3 Coordinate System

Fig. 4.1.3.1 Coordinate System Settings



The user coordinate system currently used to control the robot arm can be defined, and different user coordinate systems are optional in the drop-down menu. The robot moves based on the selected coordinate system. If the coordinate system is selected incorrectly, the robot may perform unexpected actions, resulting in collision.

4.1.4 Speed Bar

Fig. 4.1.4.1 Speed Bar



The speed bar displays the current speed ratio of the robot. This ratio is limited by the active safety level.

Users can adjust the speed ratio by dragging the slider, clicking the plus/minus buttons, or entering a value directly. The adjustment range is 1% to 100%.

The configured motion speed multiplied by the speed ratio gives the maximum

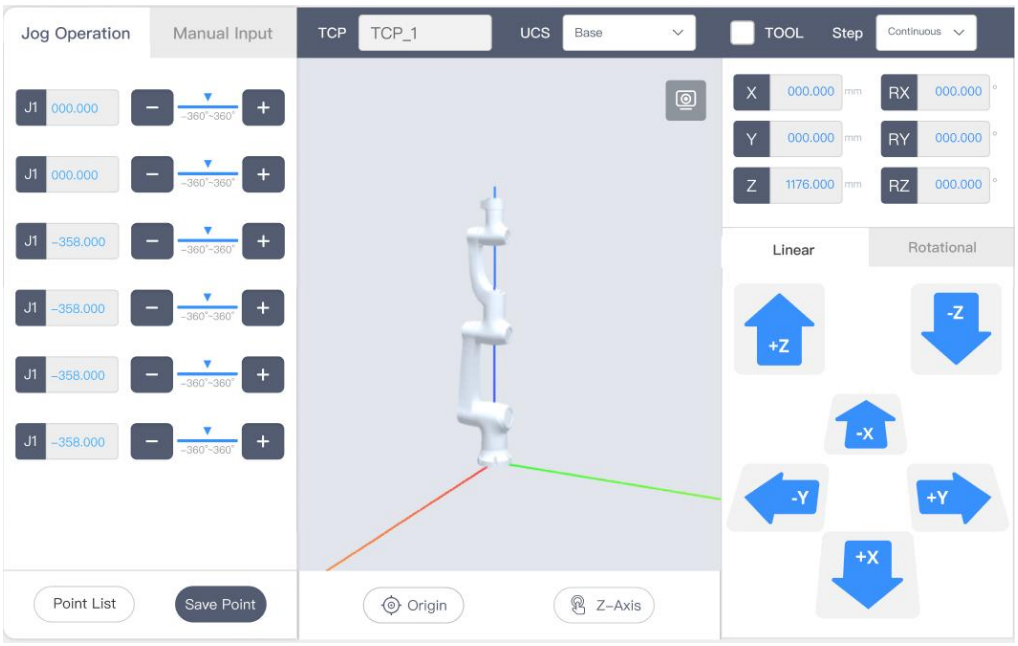
motion speed under the current setting.

In reduced mode under safety-level limitation, the maximum reduced speed ratio is 30% by default, and the speed bar is displayed in green.

4.1.5 Cartesian Space Motion

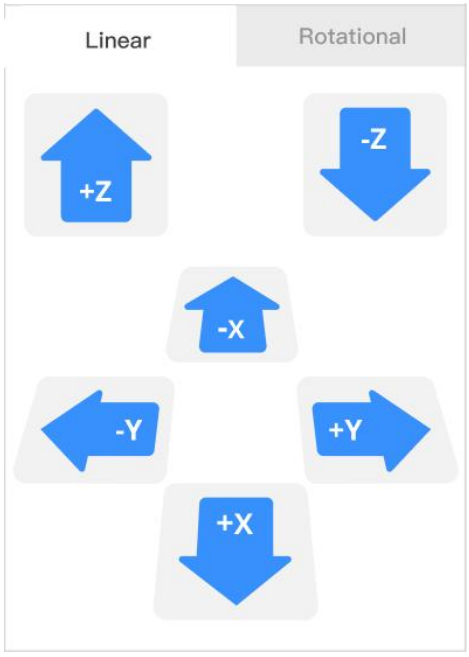
The rotation of the current robot arm pose is based on XYZ fixed-angle rotation mode.

Fig. 4.1.5.1 Cartesian Space Interface



— Definition of +/- Direction Motion Along X, Y, and Z Axes

Fig. 4.1.5.2 Linear Motion Control

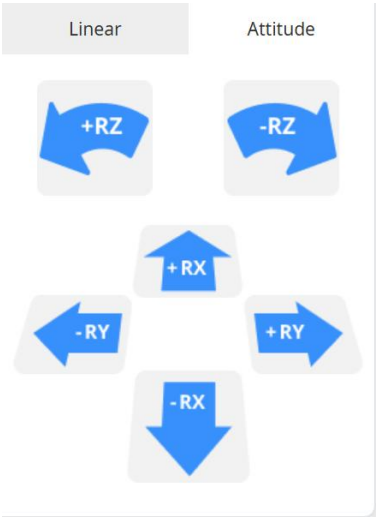


Within the safe space, the robot end TCP can move forward or reverse in the corresponding axis direction of the currently selected coordinate system:

- Press and Hold the Button: the robot TCP at the end-effector will continue to move in the +/- direction along the corresponding axis, and stop moving after releasing the button;
- Single Click on Button (Step Mode): If the step mode is not "Continuous", click the button once, and the robot TCP at the end-effector will move by a step value along the +/- direction of the corresponding axial direction;

— Definition of +/- Direction Rotation Along RX, RY and RZ Axes

Fig. 4.1.5.3 Rotary Motion Control



RX, RY, and RZ here do not have the same meaning as the RX, RY, and RZ values shown for the TCP at the upper left of the page.

The RX, RY, and RZ shown at the upper left describe the TCP orientation relative to its own tool coordinate system, whereas the RX, RY, and RZ used here describe physical rotation of the TCP about the axes of the selected user coordinate system. Due to rotational coupling, multiple angle values may change simultaneously.

RX indicates that the robot TCP at the end-effector rotates around the corresponding direction of the X axis of the current coordinate system; while RY and RZ represent rotation about the corresponding directions of the Y axis and Z axis separately.

- Press and Hold the Button: the robot TCP at the end-effector will continue to rotate along the corresponding direction of the corresponding axis, and stop moving after releasing the button;
- Single Click on Button (Step Mode): If the step mode is not "Continuous", click the button once, and the robot TCP at the end-effector will rotate by a step value along the corresponding direction of the corresponding axis;

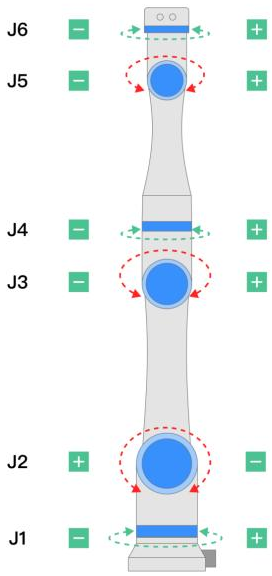
4.1.6 Joint Motion

Fig. 4.1.6.1 Robot Arm Joint Motion Display



J1-J6 sequentially represent the 1st axis to the 6th axis of the robot arm. The text box displays the angular coordinate values of each corresponding axis position of the current robot arm, in ° (angle).

Fig. 4.1.6.2 Motion Directions of Each Joint



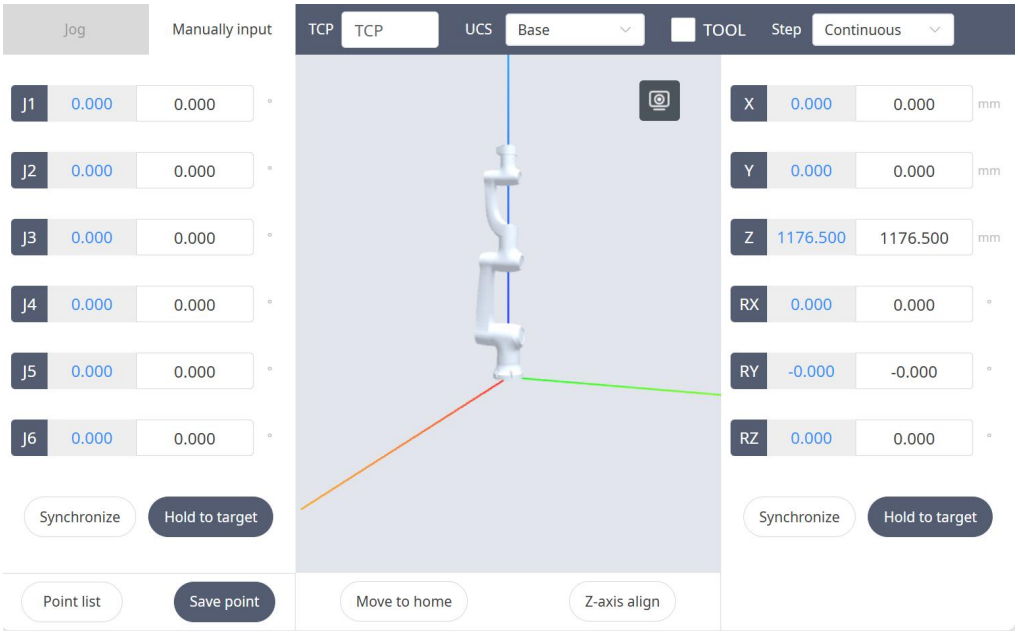
— **+/- Direction of Joints**

Within the safe space, the corresponding axis of the robot arm can move in the +/- direction, and the text box will display the current position of the axis in real time.

- **Single Click on Button**: A single click on the button, the corresponding axis moves by 2° in the +/- direction;
- **Press and Hold the Button:** A long press on the button, the corresponding axis continuously moves in the +/- direction, and the motion stops after releasing the button;
- **Fine-Tuning Function**: If "Fine-tuning" is selected, the corresponding axis will move in the +/- direction according to the fine-tuning value input by the user;

4.1.7 Manually Input

Fig. 4.1.7.1 Manually Input Interface



Click the "Manually input" button, and the user can manually enter the coordinate value of known spatial coordinate values of the point or joint angle.

4.1.8 Hold to Target

After values are entered, press and hold “Hold to Target” to move the robot toward the target pose. When the target pose is reached, the system displays a prompt indicating that the target point has been reached, and the robot stops moving.

Joint Angle Input: If a joint angle value exceeds the valid range, the system displays the allowable input range for that axis.

Cartesian Coordinate Input: When “Hold to Target” is pressed and held, the system starts calculating the target point. If the robot cannot reach the target point, the system displays the error message “Target point unreachable”, and the user

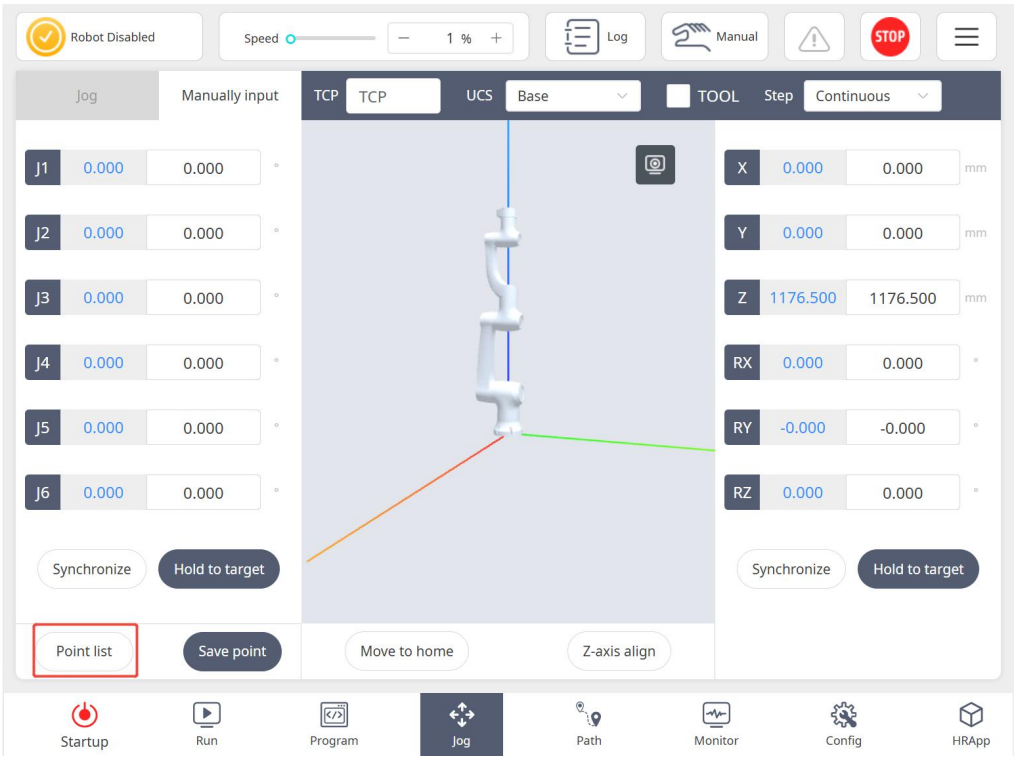
must enter a new value.

If the current input is valid but the motion exceeds the safety boundary during the hold-to-target process, the system displays the prompt “Target pose exceeds Cartesian safety space” and pops up a reset prompt.

- **TOOL Mode:** After entering TOOL mode, press and hold “Hold to Target”; the robot moves to the target point in MoveL mode under the tool coordinate system.
- **Joint Angle:** Enter the values in the joint-angle text boxes, then press and hold “Hold to Target”; the robot reaches the specified target point in MoveJ mode.
- **Cartesian Coordinates:** Enter the values in the Cartesian-coordinate text boxes, then press and hold “Hold to Target”; the robot reaches the specified target point in MoveL mode.
- **Hold to Target:** Press and hold this button to move the robot toward the target point. If the target point is reached, the robot stops and the system displays “Target point reached”. If the button is released during motion, the robot stops moving.
- **Synchronize:** Synchronize the values in the text boxes with the robot’s current actual position.

4.2 Point List

Fig. 4.2.1 Jog Interface



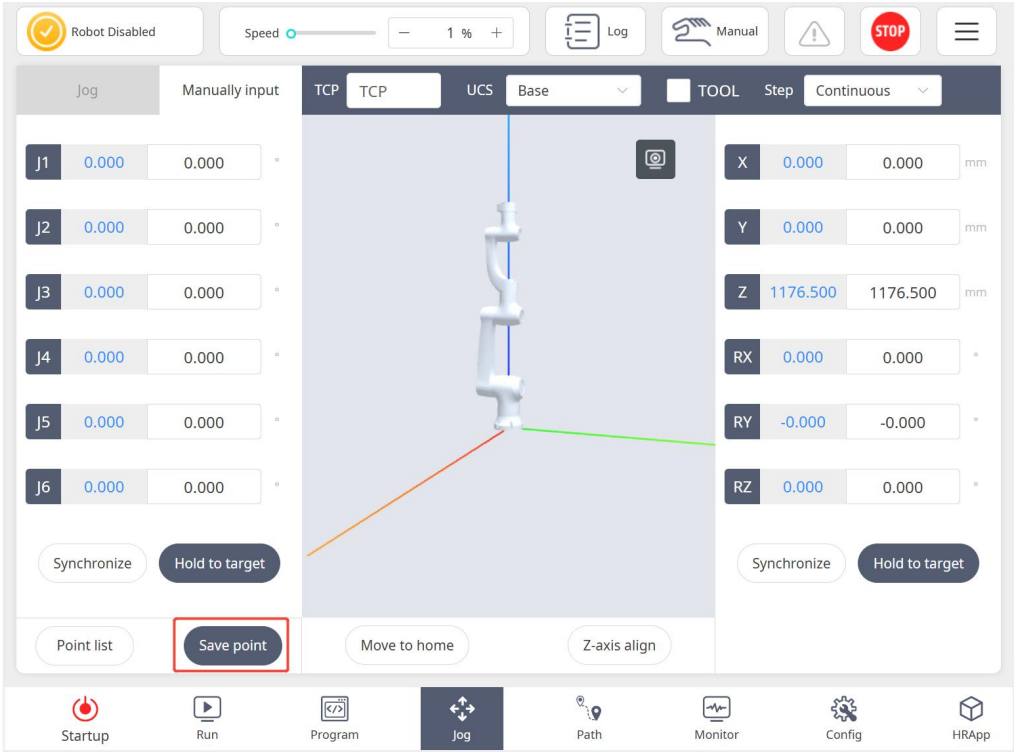
4.2.1 Point List Application

There may be multiple waypoints in the program that use the same point. The point can be saved to the point list in advance to facilitate subsequent direct call. To modify a point, just update the corresponding point in the point list, and the point data in the script will be updated synchronously.

4.2.2 Save Point

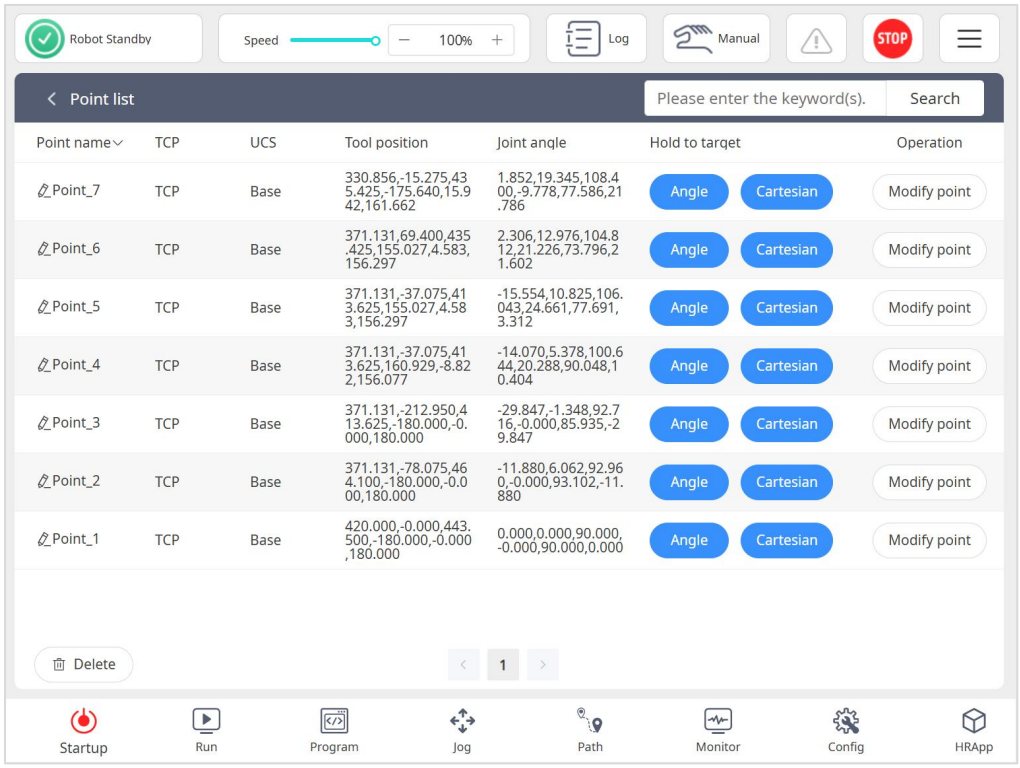
After clicking "Save Point", the current point data of the robot arm will be saved in the point list, and the point name will be saved according to the system preset. Any further modification can be done in the point list.

Fig. 4.2.2.2 Save Point



4.2.3 Point List

Fig. 4.2.3.1 Point List



The stored points are included in the point list;

- **Point Name:** The name of the currently saved point in the list;
  - **Modify Point:** Click before the point name to modify the point name;
  - **Sort:** Click the drop-down menu of point name to select the name sorting method;
  - **Default Sort:** sort according to the order of point generation time, from the latest to previous;
  - **Names in Order:** sort according to the order of characters in the ASCII code table;
  - **Names in Reverse Order:** sort according to the reverse order of characters in the ASCII code table;
- **TCP:** The TCP name used by the currently saved point in the list;
- **UCS (Coordinate System):** The name of the user coordinate system used for the saved points in the list;
- **Tool Position:** The saved point of the robot arm and the tool position of the robot arm's end-effector;
- **Joint Angle:** The joint angle parameters of the saved points in the list;
- **Hold to Target:**
  - **Angle:** Press and hold “Angle”; the robot arm moves to the point’s

joint-angle position in MoveJ mode.

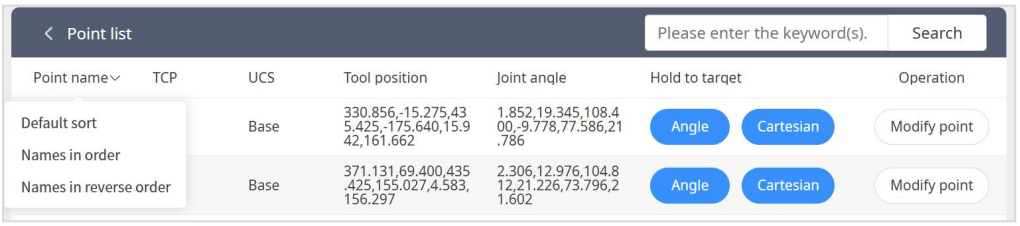
- **Space:** Press and hold “Space”; the robot arm moves to the point’s Cartesian position in MoveL mode.

– **Modify Point:** Jump to the Jog interface to modify the point. After saving, the modification takes effect. If the point is used by waypoints, the related waypoints are updated accordingly.

– **Delete:** Tick the box in front of the point and click to delete the currently selected point;

Click the point name, and sort by default order, name order or name reverse order are available in the drop-down menu;

Fig. 4.2.3.2 Point Sort



Enter keywords in the upper search bar to conduct fuzzy search for point names. When there are a large number of points, the point search function can be of great help.

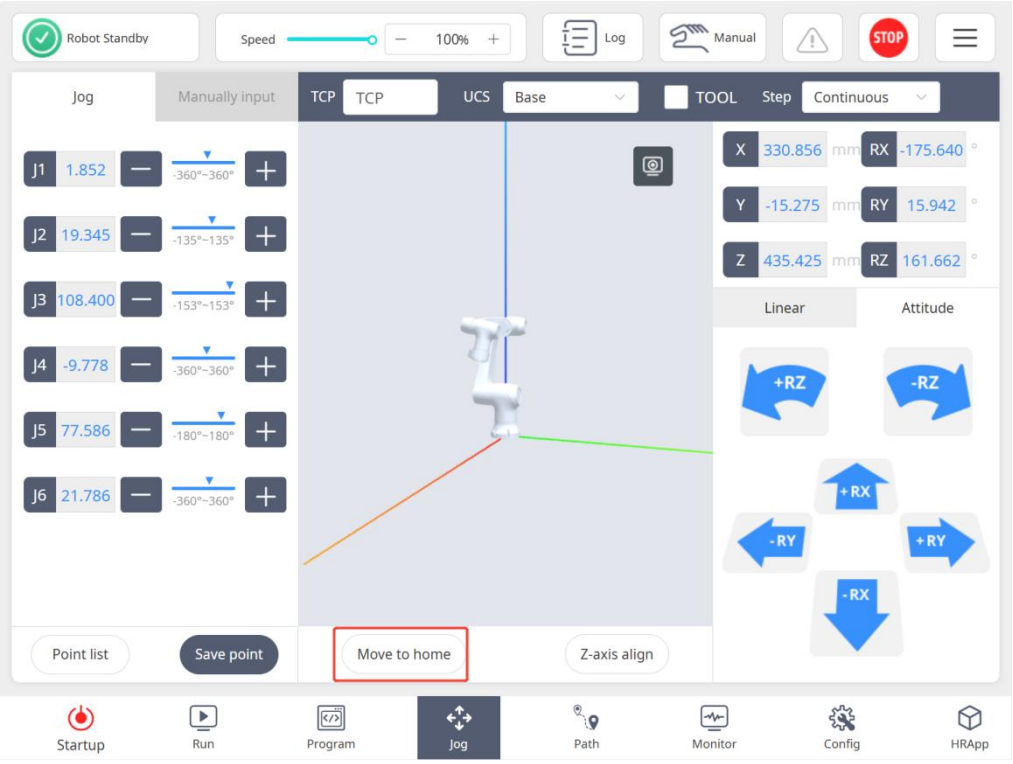
Fig. 4.2.3.3 Point Search



Note: If a point used by a waypoint is deleted from the point list, an error will be reported for the relevant waypoint. Therefore, before deleting a point, firstly check if it is used by any waypoint.

4.3 Move to Home

Fig. 4.3.1 Move to Home Button



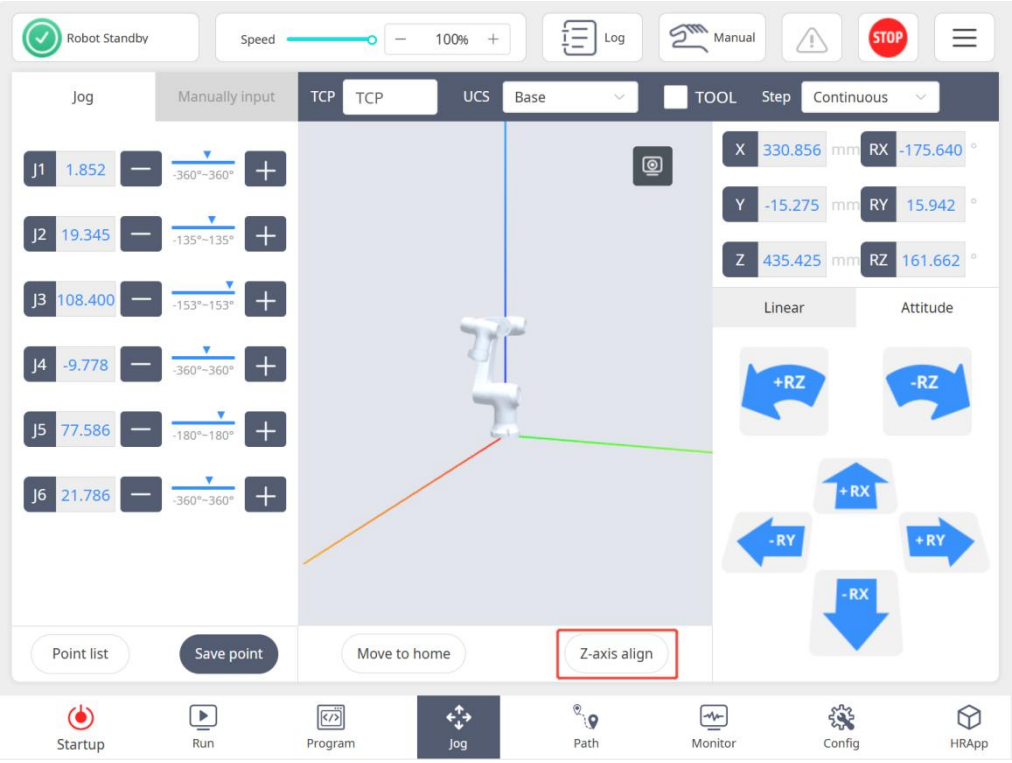
Press and hold the "Move to home" button, the robot arm moves from the current pose to the home pose in MoveJ motion mode.

The default home pose is the mechanical zero point. For E-series robots, it is an angle of 0 for all joints; for S-Series robots, it is an angle of -90 for the 2nd and 4th joints, and 0 for the remaining joints. The home position can be modified by Configuration-> Home.

Note: Please pay close attention to any obstacles around the robot arm, as the robot arm may collide with obstacles during the auto homing. It is recommended to move the robot arm to a safe position before moving to home.

4.4 Z-axis Align

Fig. 4.4.1 Z-axis Align Button



In any pose of the robot arm (except when it is at a singularity point), press and hold the "Z-axis align" button, the TCP Z direction of the end-effector tool of the robot arm will move to an pose parallel to the Z direction of the user coordinates in the MoveJ motion mode. During the motion, the spatial position parameters (X, Y, Z) and orientation parameters (RX, RY, RZ) of the end-effector tool of the robot arm will change. After the motion, the spatial position parameters (X, Y, Z) of the robot arm's end-effector will be recovered to that before the motion, while the parameters of RX, RY, and RZ will change due to the pose adjustment of the robot arm.

There are three cases of Z-axis alignment directions.

- When the included angle between the Z-axis positive direction of the tool coordinate system of the robot arm's end-effector and that of the current user coordinate system is an acute angle, after the Z-axis alignment of the robot arm, the Z-axis positive direction of the tool coordinate system and that of the user coordinate system will be in the same directions;
- When the included angle between the Z-axis positive direction of the tool coordinate system of the robot arm's end-effector and that of the current user coordinate system is an obtuse angle, after the robot arm aligning to the Z-axis, the Z-axis positive direction of the tool coordinate system and that of the user coordinate system will be in the reverse directions;
- When the Z-axis positive direction of the tool coordinate system of the robot arm's end-effector is perpendicular to that of the current user coordinate system, after the robot arm aligning to the Z-axis, the Z-axis direction is uncertain, and the motion is carried out according to the principle of shortest movement distance;

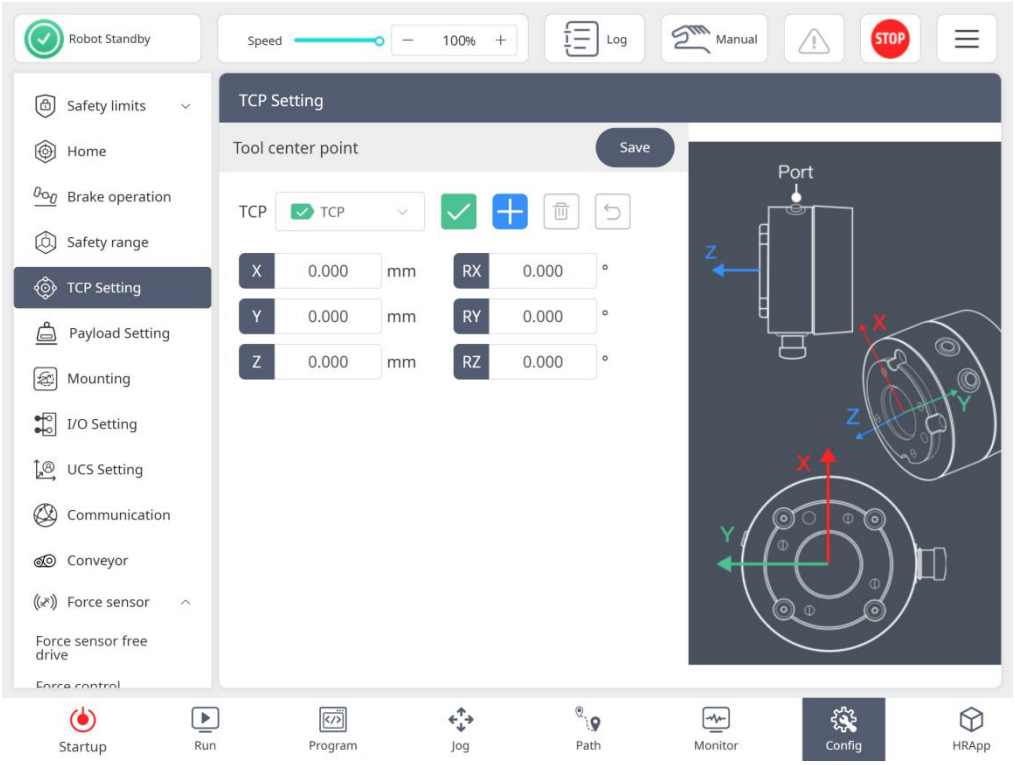
Configuration

This chapter contains the following topics:

– TCP Setting
– Payload Setting
– Base Mounting Angle
– I/O Setting
– Safety Limits
– User Coordinate System
– Safety Range
– Communication (Modbus)
– Tracking Conveyor
– Force Sensor
– Brake Operation
– Boot Configuration
– Home
– Basic Free Drive Adjustment

5.1 TCP Setting

Fig. 5.1.1 TCP Setting



The tool coordinate point (TCP) is the center point of the tool at the robot arm's end-effector. Each TCP contains a set position translation and pose rotation relative to the center of the tool output flange.

The X, Y, and Z coordinates determine the position of the TCP, while the RX, RY, and RZ coordinates determine its orientation. When all values are zero, the TCP coincides with the center point of the tool output flange, as shown below:

Fig. 5.1.2 TCP Demonstration

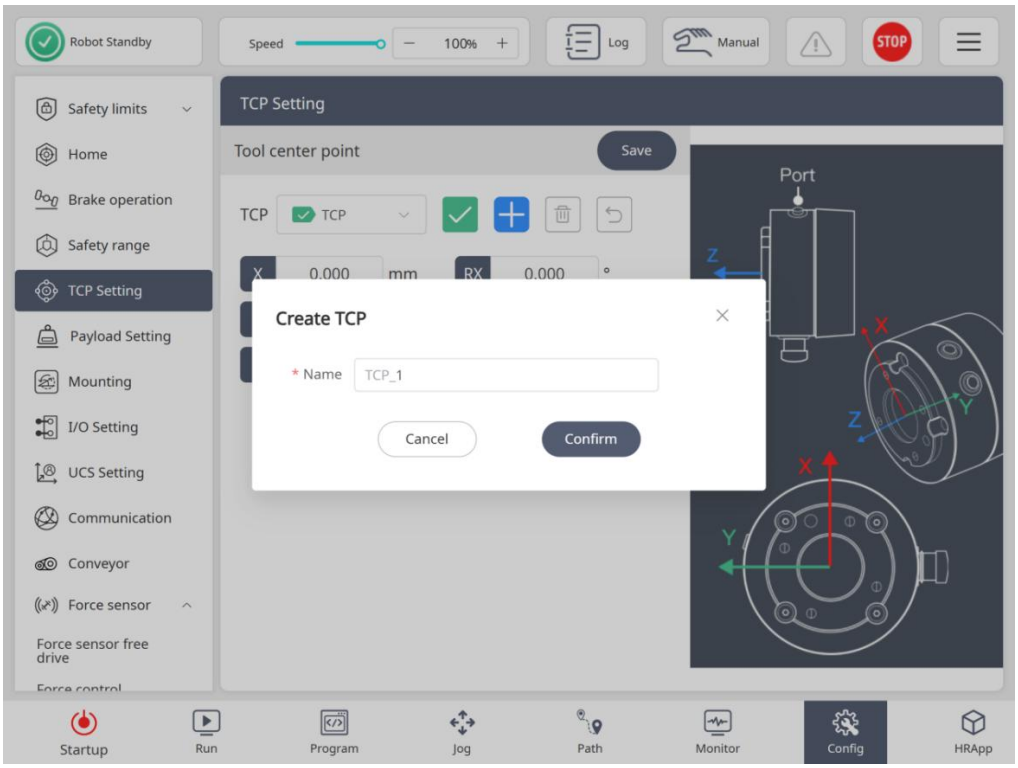


5.1.1 Create TCP

Click , the Create TCP popup appears. There is a system preset name, while users can also customize the TCP name. It will take effect after clicking "Confirm". The newly created TCP will be displayed in the drop-down menu option of the current TCP.

The TCP name can be entered in the input box. The name should be between 1 to 24 characters in length and can contain letters, numbers and underscores.

Fig. 5.1.1.1 Create TCP Interface



5.1.2 TCP Position Setting

The user can manually enter the value to set the TCP position, if the user already knows the offset value of the new TCP relative to the center of the flange (by measuring or viewing the size specifications). After entering the value, click "Apply" to take effect.

Fig. 5.1.2.1 TCP Offset

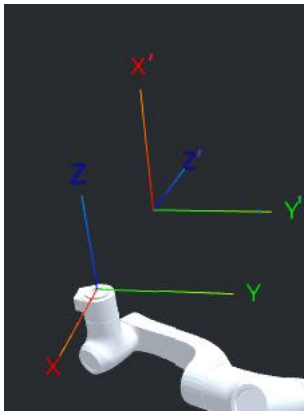
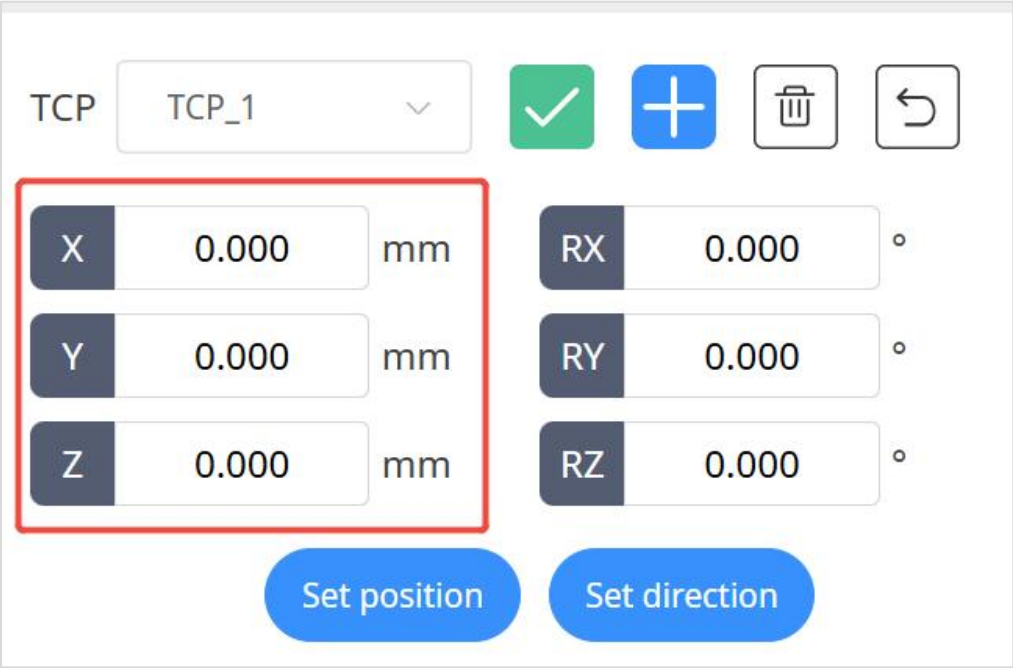


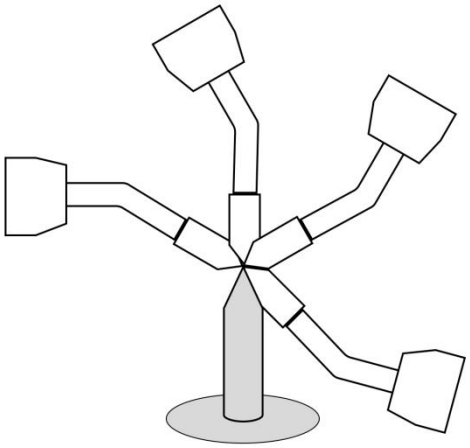
Fig. 5.1.2.2 TCP Position Setting



If the user does not know the offset value of the new TCP relative to the center of the flange, the origin position of the newly created tool coordinate system can be determined by the four teaching points.

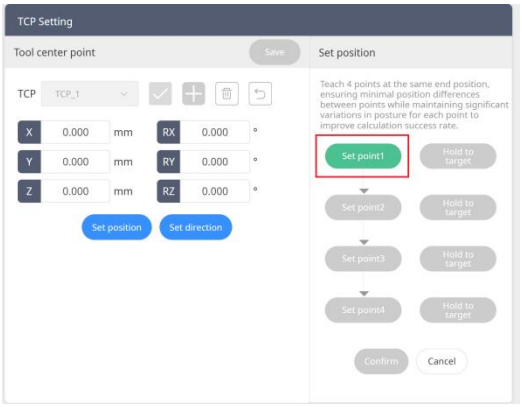
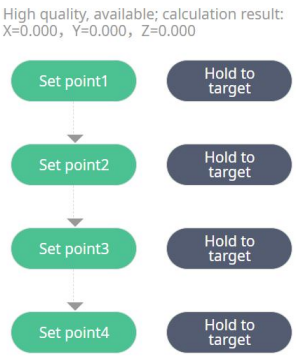
The teaching principle of the four-point method: Teaching by moving the end point of the tool from four different angle directions to the same position.

Fig. 5.1.2.3 4-point Teaching



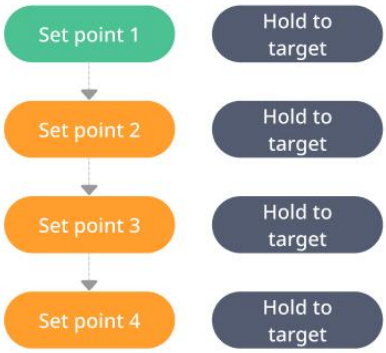
Example:

SN	Operation	Example
1	Click the " + " button to generate TCP_1;	Create TCP * Name TCP_1

2	Click the "Set position" button to enter the four-point teaching interface. According to the prompt, move the tool center point at the robot arm's end-effector to the same position from different angles. The upper right corner of the teaching box displays prompts during setting;	
3	When the first four points are taught, the green point button indicates that the points meet the standards; meanwhile, "High quality, available" is displayed directly above and the calculation result is displayed, indicating that it is usable;	

– If after completing the teaching of the first three points, the Set point button becomes yellow, it indicates average point quality; meanwhile, the upper right corner prompts the calculation result, indicating that the point is available. Users can decide whether to re-teach the point of the yellow button based on the actual accuracy needs. After re-teaching, the green button color indicates improved point quality and higher accuracy of TCP;

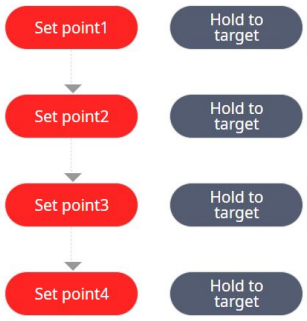
Fig. 5.1.2.4 Low Point Quality



– If after completing the teaching of four points, the Set point button becomes red, it indicates unqualified point quality; meanwhile, the upper right corner prompts "Unqualified and position cannot be calculated". The user needs to re-modify the point corresponding to the red button, so as to obtain green buttons for all or partially yellow buttons before successful setting. It is suggested that all points

should be taught to become green in buttons, which means better point quality and higher TCP accuracy;

Fig. 5.1.2.5 Unqualified Point Quality



– Users will continue to teach the 4th point to verify the TCP accuracy. When all points buttons are in green or partially yellow, click "Apply" to confirm the teaching positions;

Fig. 5.1.2.6 Teaching Completed



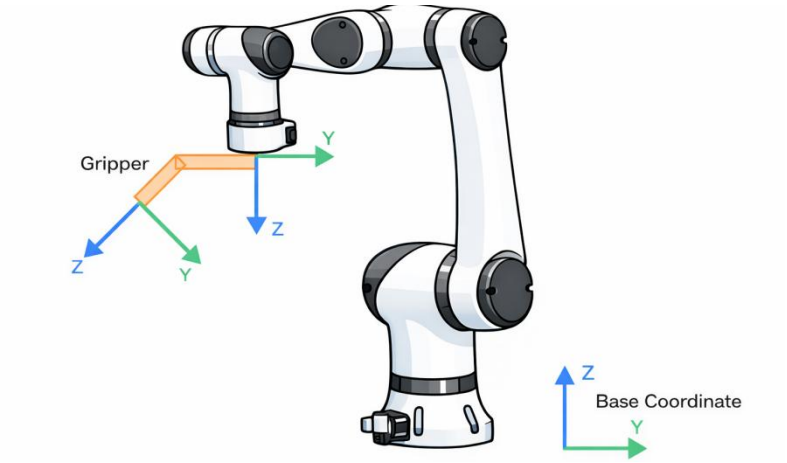
– To exit the teaching process, click "Cancel";

Note: The teaching points must be sufficiently diversified to ensure the accurate calculation results.

5.1.3 TCP Direction Setting

When the XYZ direction of the tool is inconsistent with the XYZ direction of the flange at the end-effector, the most obvious one is the Z direction, then the direction of the tool TCP needs to be re-taught; The orientation is set for the reference based on a coordinate system.

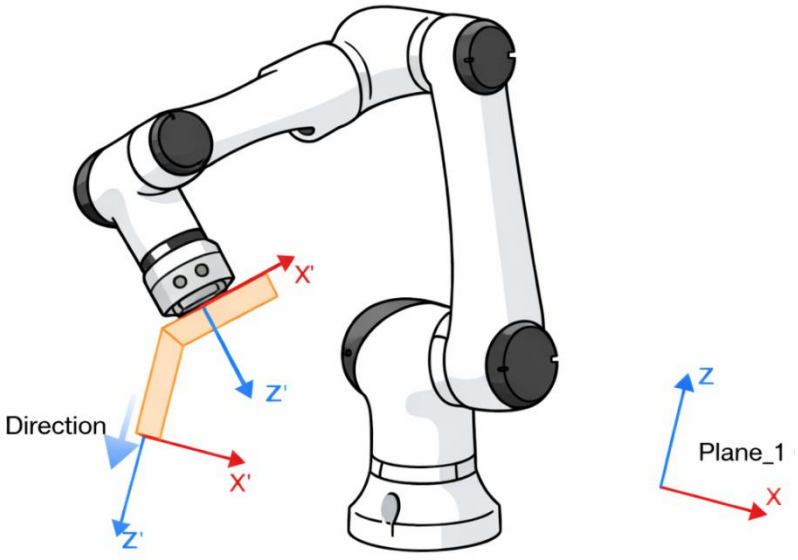
Fig. 5.1.3.1 TCP Direction Change



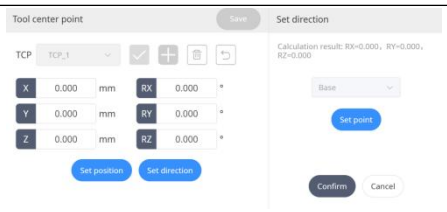
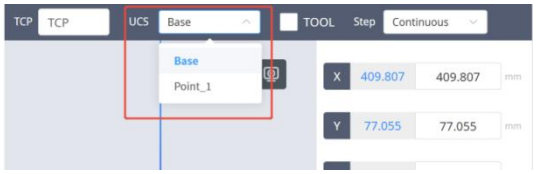
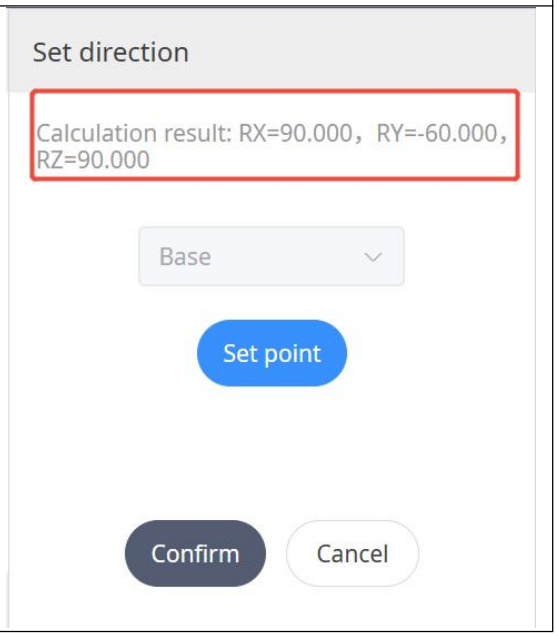
There are two ways to obtain the values of TCP position, one is by measurement, and the other is by the teaching calculation of teach pendant. The teaching calculation method is contained herein.

Teaching Calculation Method

Fig. 5.1.3.4 Teaching Calculation Method



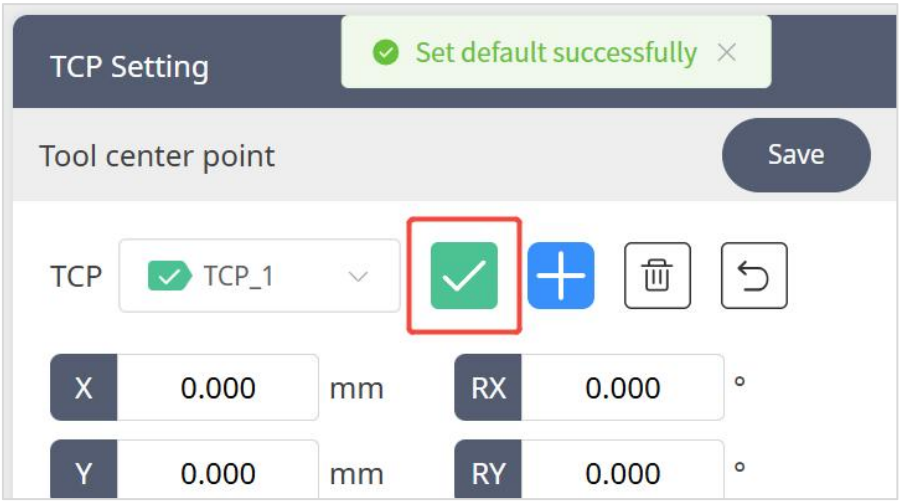
SN	Operation	Example
1	Click the "Set direction" button to enter the teaching page;	

2	Click the "Set point" button;	
4	In the point setting page, select a coordinate system from the drop-down list. For example, the "Plane_1" coordinate system means that the direction of the newly created TCP _2 is parallel to that of the "Plane_1" coordinate system, and the directions are completely consistent;	
5	Click "Confirm" to return to the teaching page to complete the teaching in TCP direction, and the calculation result will be prompted directly above;	

5.1.4 Default TCP

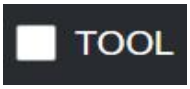
Only one of the configured TCPs can serve as the default TCP. In the TCP drop-down menu, there is a green icon to the left of TCP name set as default. To set the currently selected tool coordinates to the default TCP, click icon.

Fig. 5.1.4.1 set as default TCP



The TOOL mode in the jog interface uses the default TCP.

Fig. 5.1.4.2 the jog interface TOOL mode



5.1.5 Others

- Modify: After selecting the current TCP, users can modify the position and direction offset of the TCP;
- Delete: Delete the tool coordinates selected in the current TCP;
- Cancel: Cancel the currently unsaved teaching process or entered value;
- Apply: Click "Apply" to effect the edited data;
- Set As Default: To set a named TCP as the default TCP;

Note: If the TCP used by a waypoint is deleted, the data in the waypoint will be deleted synchronously. The waypoint becomes unusable, and the user needs to re-edit the waypoint and the corresponding TCP.

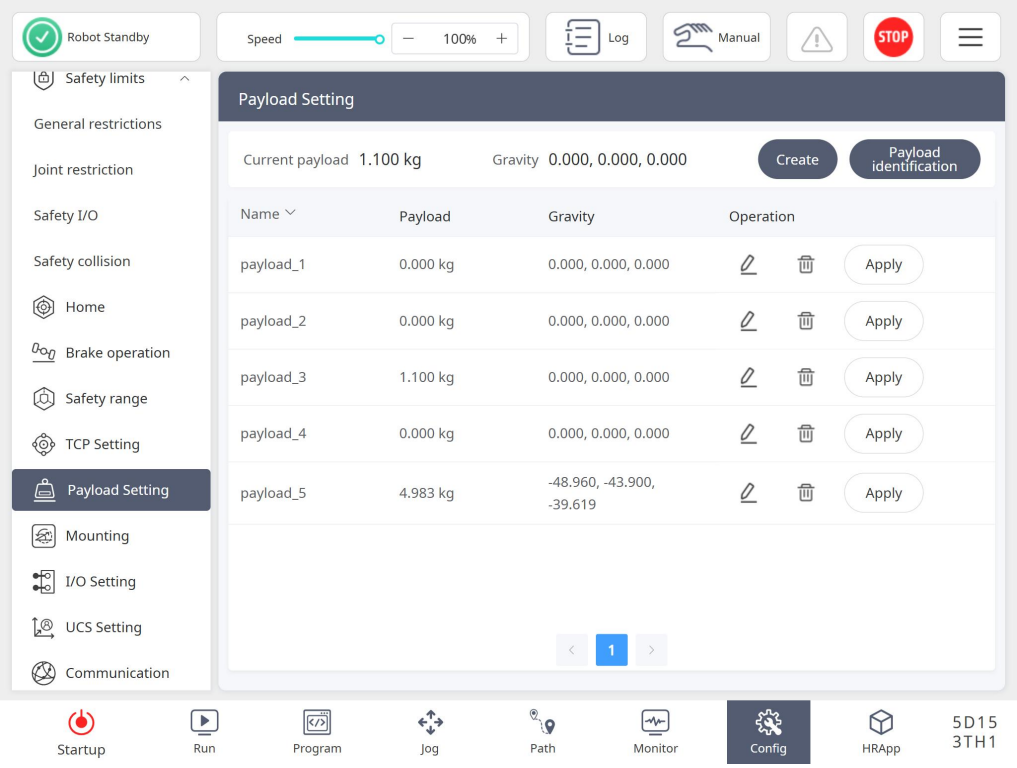
5.2 Payload Setting

Payload refers to the mass and center of gravity parameters of the tool at the robot's end. Correct payload configuration is crucial for the robot's motion accuracy, basic free drive, and advanced free drive , as well as force protection.

The system supports creating and managing multiple payload configurations. Users can save corresponding payload parameters for different end payload and switch between them quickly when needed.

Note: Incorrect payload settings will seriously affect the robot's operation. Functions including basic free drive, advanced free drive , and force protection will no longer be accurate, and potential hazards may occur!

Fig.5.2.1 Payload Setting Interface



5.2.1 Current Payload

At the top of the Payload Setting page, the current payload value and its center of gravity information are displayed. The displayed current payload refers to the payload currently used by the robot, which is directly related to the robot's operation.

Fig.5.2.1.1 Current Payload

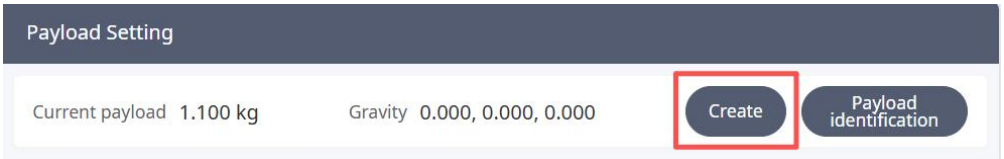


Each time the payload setting page is accessed or refreshed, the system automatically displays the payload parameters currently in use.

5.2.2 Create

The "Create" function is used to add a new payload item, save the payload information by name, and optionally apply it as the current payload. After saving, the new payload will be recorded in the payload list. If "Apply as current payload" is not selected, the information will only be saved without affecting the robot's operation.

Fig.5.2.2.1 Create Payload Button



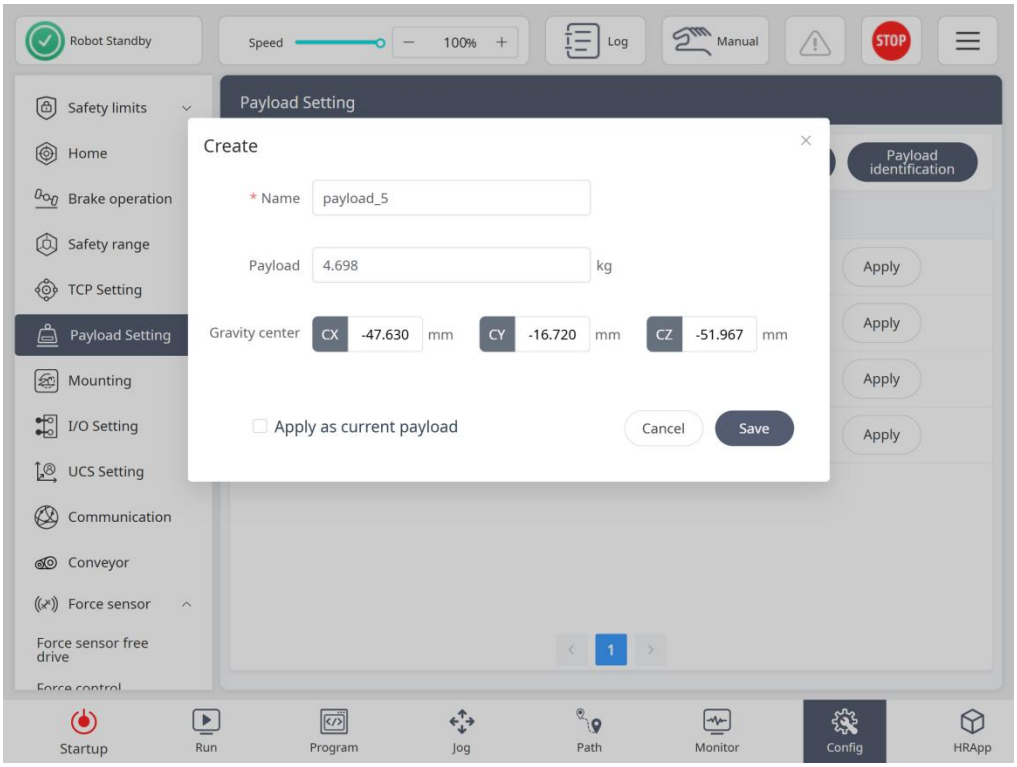
Click [Create] to open the Add Payload window. The system provides a default name, which can be customized by the user. Users can enter the payload value and center of gravity coordinates (CX, CY, CZ), and check the box to set it as the current payload. The settings take effect after clicking [Save], and the new payload will be added to the end of the payload list.

Naming Rules: The name must be 1-24 characters long, can contain letters, numbers, and underscores, and must not duplicate existing payload names.

Payload Setting: The payload value must be a positive number within the maximum payload capacity of the robot (e.g., 0~5 kg for a 5 kg robot).

Center of Gravity Setting: CX, CY, and CZ can be set to values between -1000 and 1000.

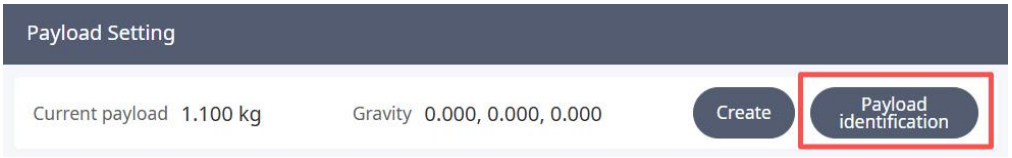
Fig.5.2.2.2 Create Payload Interface



5.2.3 Payload Identification

The payload identification automatically calculates the mass and center of gravity of the current end payload. During identification, the robot will move continuously. Please ensure sufficient space is reserved.

Fig.5.2.3.1 Payload Identification Button

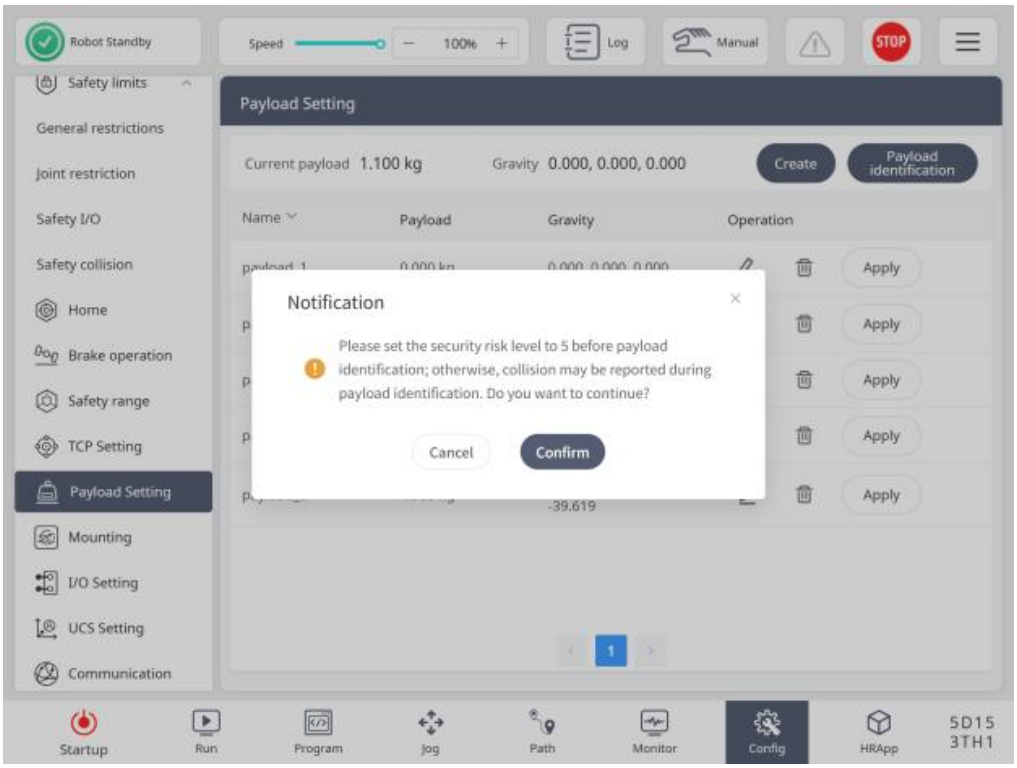


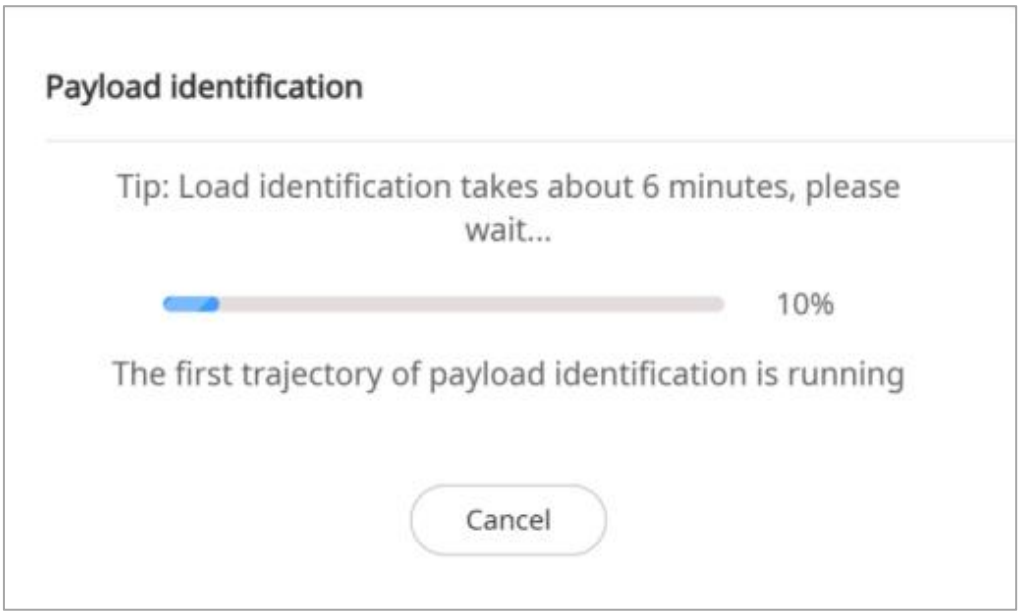
Click [Payload identification] to open a safety confirmation popup. After clicking [Confirm], the robot starts moving and initiates payload identification. During the process, the robot moves along three predefined trajectories to analyze the end

payload. The identification takes approximately 6 minutes.

Warning: Continuously monitor the robot during payload identification. In case of collision risk, press the emergency stop button immediately.

Fig.5.2.3.2 Payload Identification in Progress





During payload identification, users can manually click [Cancel] to terminate the process. If an error occurs, the robot will stop moving, the payload identification will end, and an error message will pop up. Restarting payload identification is required. Terminating identification will not save any data related to the current process.

Fig.5.2.3.3 Payload Identification Wrong

The first trajectory is wrong, the identification cannot be completed, and the payload identification ended

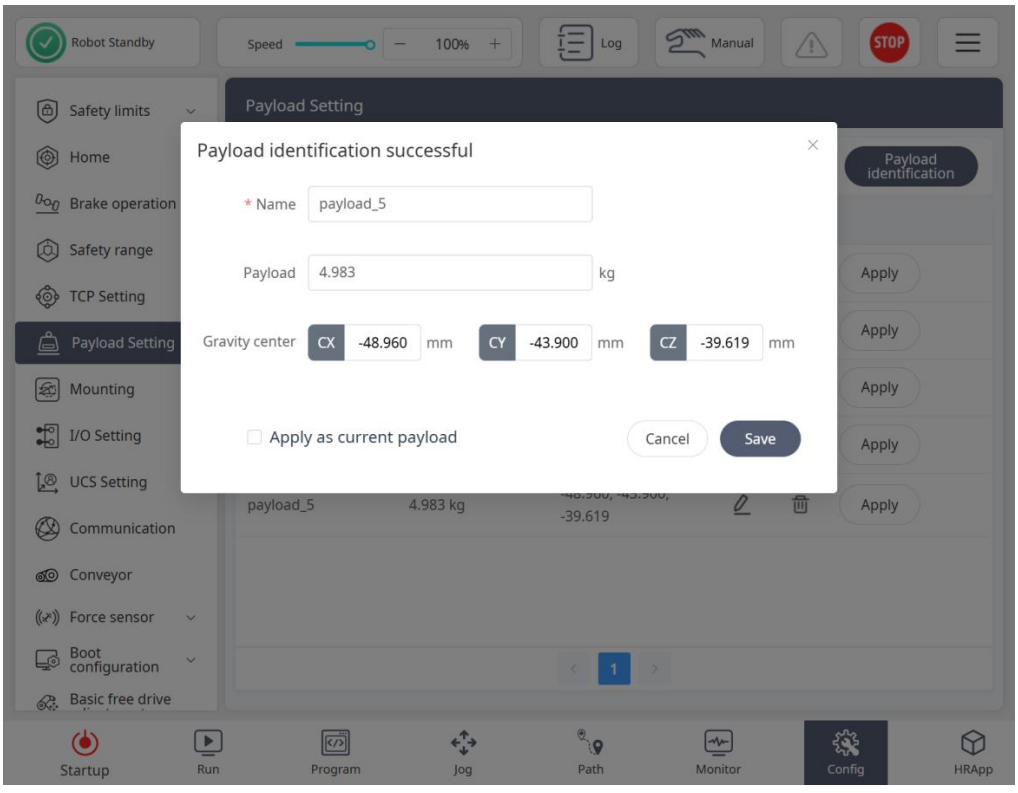
After successful payload identification, the robot stops moving, and the system displays a popup with the calculated payload mass and center of gravity parameters. Users can choose:

Apply as current payload:Check this box to apply the identified parameters as the current payload after clicking [Save].

Save:Save the identified parameters as a new payload configuration item, add it to the payload list.

Cancel:Discard the identified results.Even if [Apply as current payload] is checked, it will not take effect.

Fig.5.2.3.4 Successful Payload Identification



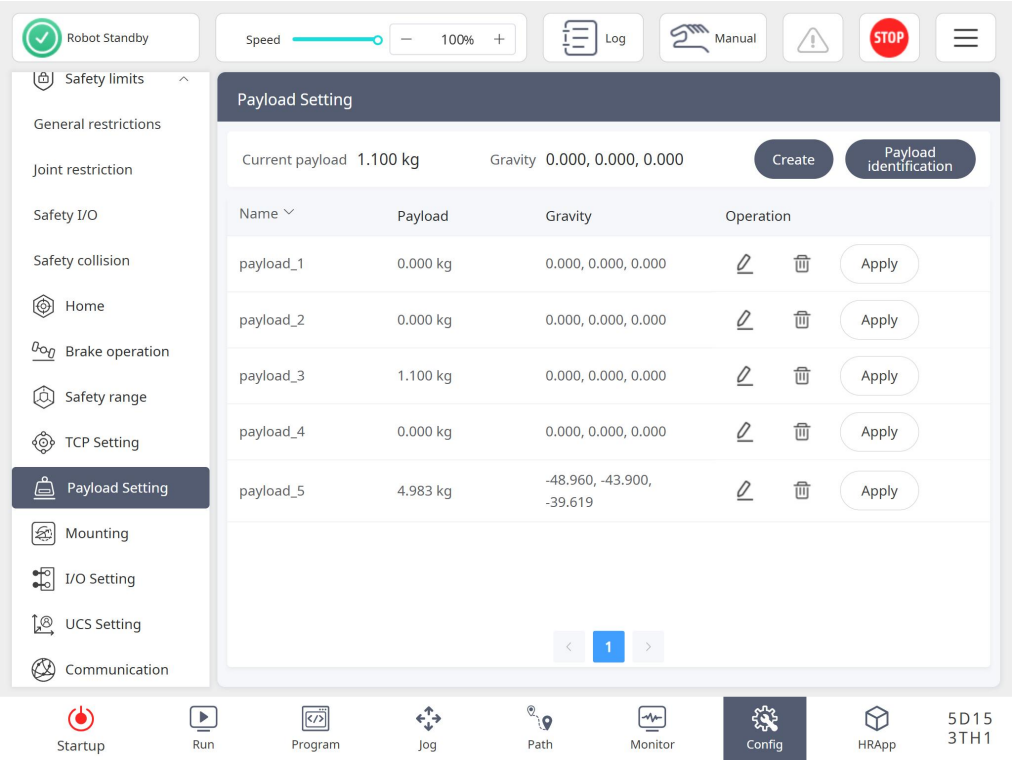
Note: When the robot is unloaded, the parameters obtained through payload identification may be inaccurate, but the torque result (product of mass and center of gravity) remains reliable.

Note: When the base tilt angle of the robot's base mounting angle is non-zero, the result of payload identification will be inaccurate.

5.2.4 Payload List

The payload list displays each payload's name, mass, center of gravity information, and operation options (Apply, Modify, Delete).

Fig.5.2.4.1 Payload List



The payload list records all saved payloads:

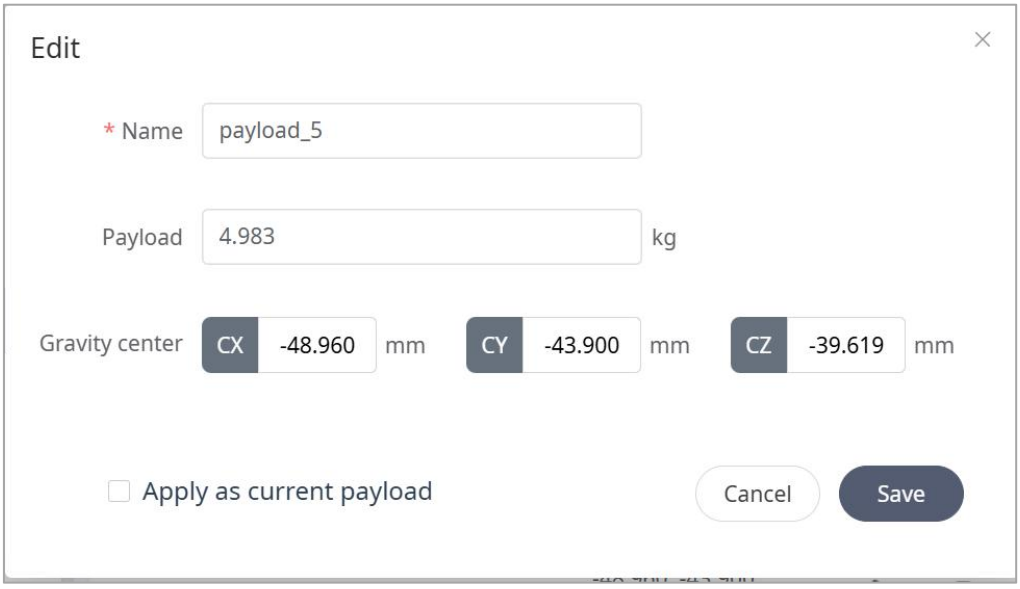
- **Name:** Displays the name of each saved payload configuration item.
  - **Sorting:** Click the drop-down menu next to "Name" to select a sorting method:
  - **Default Sort:** By creation time (oldest first).
  - **Name Ascending:** By ASCII character order.
  - **Name Descending:** By reverse ASCII character order.
- **Payload:** The mass of the payload configuration item.
- **Gravity:** The center of gravity coordinates (CX, CY, CZ) of the payload configuration item.
- **Operation:**
  - **Apply:** Set the payload's mass and center of gravity as the current payload, updating the current payload value.
  - **Modify:** Open the edit popup to modify the payload's name, mass, or center of gravity.
  - **Delete:** Open a confirmation popup to delete the payload configuration item.

5.2.5 Modify Payload

Click [Modify] ,for a payload configuration item to open the edit popup. Users can modify the payload's name, mass, or center of gravity and click [Save] to apply changes. Check [Apply as current payload] in the popup to synchronize the

modified parameters as the current payload after saving.

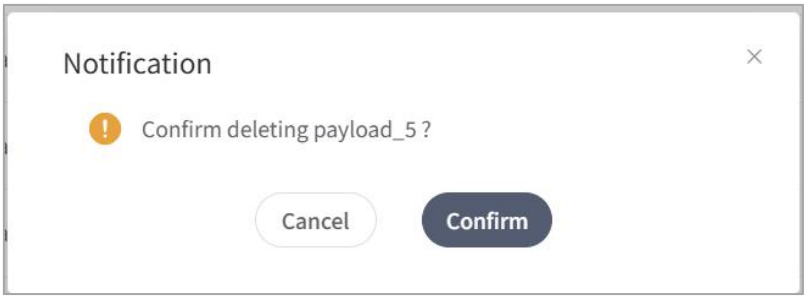
Fig.5.2.5.1 Modify Payload Interface



5.2.6 Delete Payload

Click [Delete] for a payload configuration item to open a confirmation popup. The item will be deleted after clicking [Confirm].

Fig.5.2.6.1 Deleting Payload Confirmation



5.2.7 Others

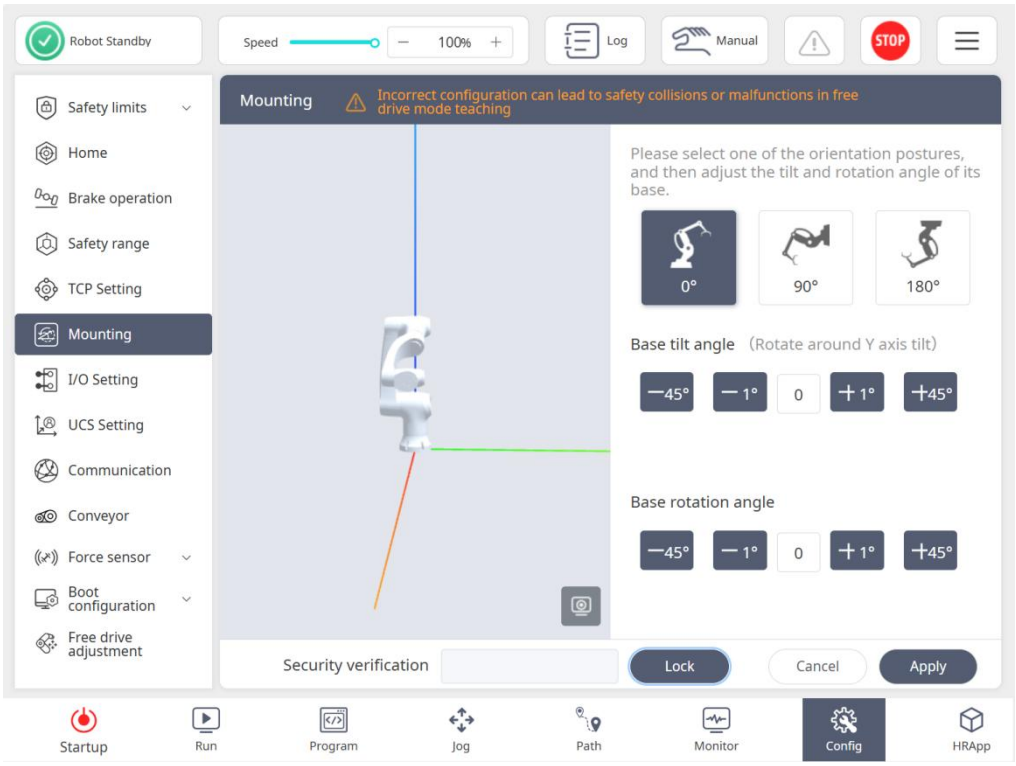
Save: Payload parameters and the payload list are saved together with the program file. Therefore, after updating the current payload or payload list, you must save the program in the programming interface to ensure the changes are actually persisted.

Payload Identification: If you log out during payload identification, the result popup will not be displayed when you log back in and access the Payload Setting page. Therefore, to obtain and process the identification result upon completion, it is recommended to remain logged in and stay on the Payload Setting page during the process.

Warning: Payload settings directly affect the robotic arm's motion performance and safety. Ensure that payload parameters are correctly configured based on the actual end effector.

5.3 Base Mounting Angle

Fig. 5.3.1 Security Verification



This page configures the installation position and angle of the robot arm. The security password (consistent with the currently logged in administrator password) must be verified before operation.

The world coordinate system is a global three-dimensional rectangular coordinate system, usually referenced by the ground or installation platform, which is used to describe the absolute position and orientation of objects in space. Its origin is usually defined at the center of the robot arm base, with the X-axis point to the body cable directly, the Y-axis on the left side of the body cable, and the Z-axis vertically upward.

When the robot arm is mounted on the ground or on a table parallel to the ground, and the robot is in an initial pose, the base coordinate system of the robot is parallel to the Z axis of the world coordinate system.

If the robot arm is ceiling-mounted, wall-mounted or installed at a certain angle, the base coordinate system of the robot will no longer coincide with the world coordinate system. It is necessary to determine a new base coordinate system according to the actual installation mode. Through functions in the interface, users can redefine the user coordinate system of the robot to match the installation method, thus ensuring that the robot's motion direction and operating logic are as expected.

The controller uses advanced dynamics models to ensure the smooth and accurate movement of the robot arm, and enables the robot arm to support itself when it is in the basic free drive mode. Therefore, the installation angle and position of the robot arm must be set correctly.

Warning: Always apply the correct installation settings; otherwise, it may cause accidents, or the robot arm may lose control at the moment of pressing the free drive button!

5.3.1 Base Tilt Angle

The base tilt angle is relative to the horizontal plane.

Commonly used angles are that for ceiling (180°), wall (90°) and ground (0°). If other tilt angles are required, they can be adjusted by ±1° (hold and press/single click) and ±45° (single click). The angle for ground is by default (0°).

Tilt Angle: Rotate around the Y-axis of the world coordinate system, + corresponds to counterclockwise, the effective range is ±360°.

5.3.2 Base Rotation Angle

The base rotation angle refers to the angle at which the base rotates around its vertical line.

It can be adjusted by ±1° (hold and press/single click), ±45° (single click). The default value is 0°.

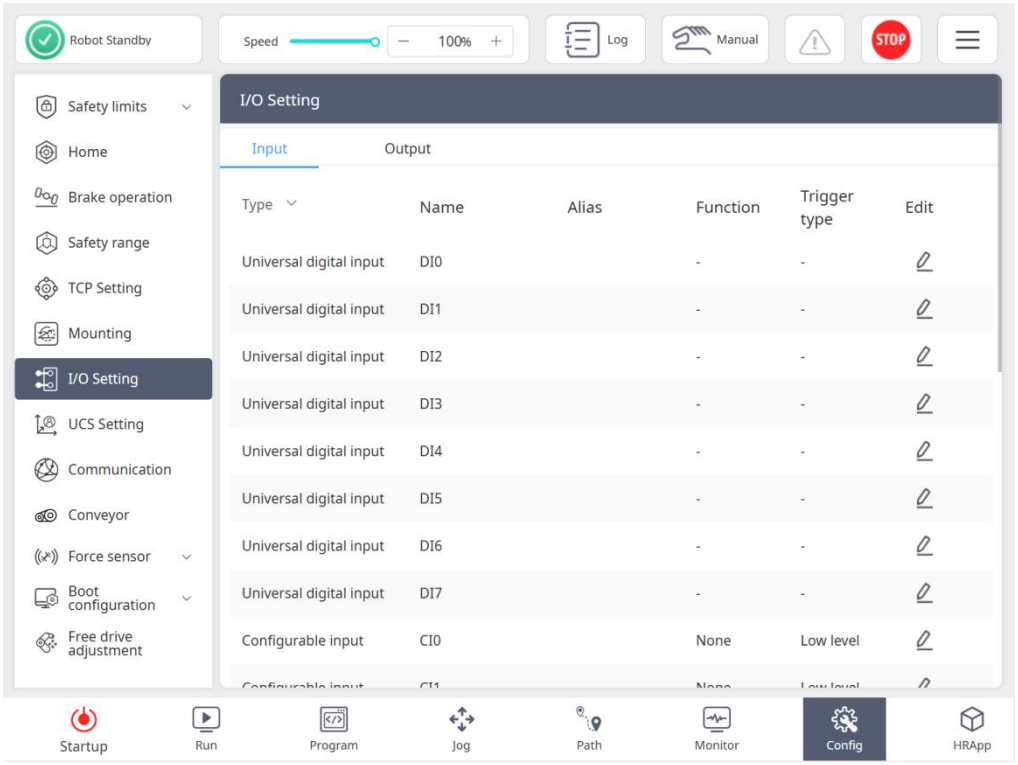
Rotation Angle: Rotate around the Z axis of the robot arm base coordinate system, + corresponds to counterclockwise, the effective range is ±360°.

5.4 I/O Setting

Configure the Input and Output I/O interface.
The input I/O categories include General Input, Configurable Input, Analog Input, Tool Input, and Tool Analog Input.
The output I/O categories include General Output, Configurable Output, Tool Output, and Analog Output.

Among them, “Configurable Input” and “Configurable Output” I/O allow the alias, function, and trigger type to be configured, while other I/O types only allow the alias to be configured.

Fig. 5.4.1 I/O Setting Interface

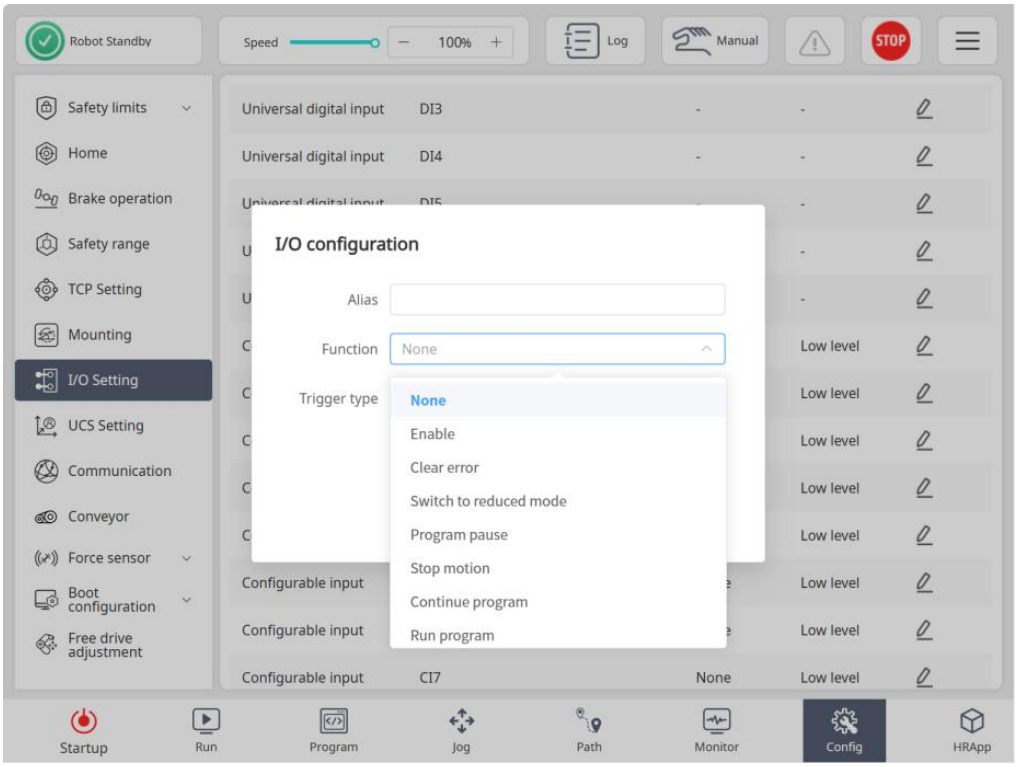


5.4.1 Configurable Input I/O

The I/O named CI0 to CI7 are Configurable input I/O. These input I/O signals can be configured to trigger corresponding actions. The available actions include Enable, Clear error, Switch to reduced mode, Program pause, Stop motion, Continue program, Run program, Pauses movement, Continuous movement, Electrify robot, and Initialize controller.

Note: Each function can only be set with one I/O, and the same function is not available to multiple I/Os;

Fig. 5.4.1.1 Input I/O Configuration



The controller supports triggering a variety of operations through I/O signals, and the trigger type can be selected from"Rising edge" or "Falling edge". When the configuration function is enabled, the controller will continuously monitor the I/O status and execute the corresponding command when the selected trigger condition is met. The specific functions are as follows:

- **Enable**: After being triggered, the system sends an Enable command.
- **Clear error**: After being triggered, the system sends a Clear error command.
- **Switch to reduced mode:** After being triggered, the system switches to reduced mode.
- **Program pause:** After being triggered, the system sends a Program pause command.
- **Stop motion:** After being triggered, the system sends a Stop motion command.
- **Continue program:** After being triggered, the system sends a Continue program command.
- **Run program:** After being triggered, the system sends a command to run the Func_main function and start the timer.

  Note: To use this function, the following conditions must be met: a function script named Func_main has been created; the script has been compiled and the system has been switched to script mode.
- **Pauses movement:** After being triggered, the system sends a Pauses movement command.
- **Continuous movement:** After being triggered, the system sends a Continuous movement command.

- **Electrify robot:** After being triggered, the system sends an Electrify robot command.
- **Initialize controller:** After being triggered, the system sends an Initialize controller command.

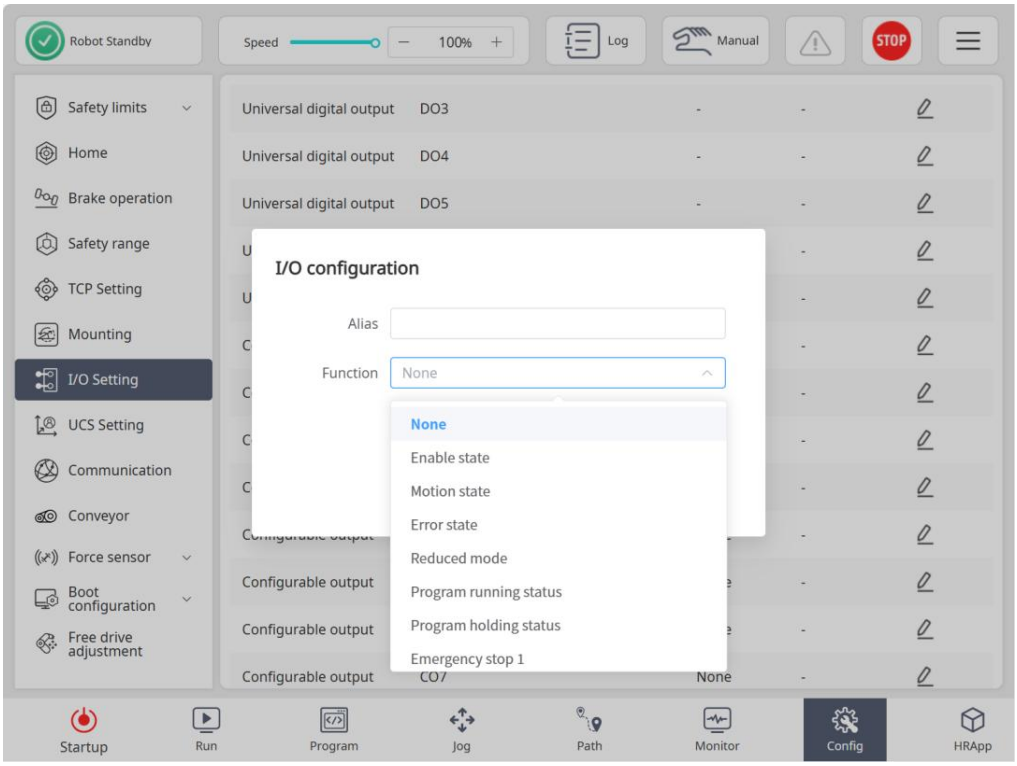
For example, if CI0 is configured with the Enable function and the trigger type is set to High level, the robot is enabled when CI0 receives a high-level signal; when CI0 receives a low-level signal, the robot is disabled.

Note: Each function can be assigned to only one I/O signal. For example, if the Enable function has already been assigned to CI0, other CI channels cannot be assigned the Enable function.

5.4.2 Configurable Output I/O

The I/O named CO0 ~ CO7 is a configurable output I/O, and the output I/O signal can be set to reflect the current state of the robot. When enabled, the robot default output is set to high.

Fig. 5.4.2.1 Output I/O Configuration



The controller supports real-time monitoring of the robot status and outputs a high-level signal when the specified condition is met. The available functions are as follows:

- **Enable state:** Monitors whether the robot is in the enabled state. When the condition is met, a high-level signal is output.
- **Motion state:** Monitors whether the robot is moving. When the condition is met, a high-level signal is output.
- **Error state:** Monitors whether the robot is in an error state. When the condition

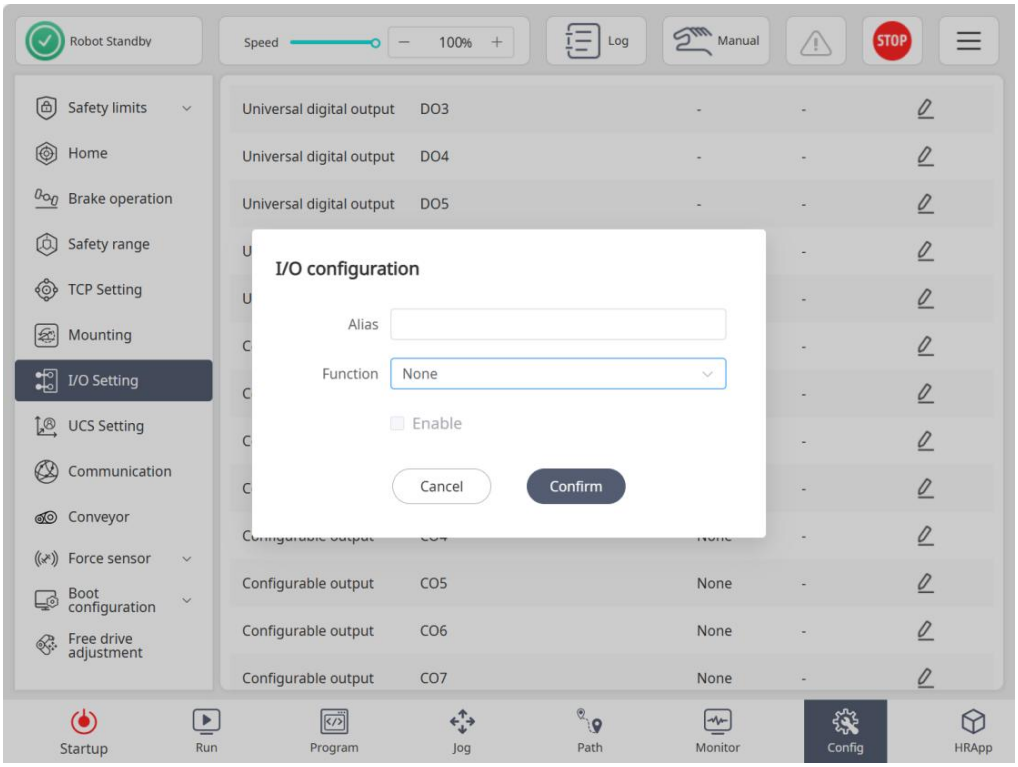
is met, a high-level signal is output.

- **Reduced mode:** Monitors whether the robot is in reduced mode. When the condition is met, a high-level signal is output.
- **Program running status:** Monitors whether the robot is in the program running state. When the condition is met, a high-level signal is output.
- **Program holding status**: Monitors whether the robot is in the program holding state. When the condition is met, a high-level signal is output.
- **Emergency stop 1**: Monitors whether the robot is in the emergency stop state. When the condition is met, a high-level signal is output.
- **Emergency stop 2**: Monitors whether the robot is in the partial emergency stop state. When the condition is met, a high-level signal is output.
- **Movement pause state**: Monitors whether the robot is in the movement pause state. When the condition is met, a high-level signal is output.

For example, if CO4 is configured with the Motion state function, CO4 outputs a high-level signal when the robot arm is moving; otherwise, CO4 outputs a low-level signal.

5.4.3 I/O Configuration Editing

Fig. 5.4.3.1 I/O Configuration Editing



- **Alias:** Available for input and output signals, to facilitate memorizing the function of each signal. Click the edit button to modify;
- **Function:** Only for configurable I/O (input CI0 ~ CI7, output CO0 ~ CO7), used to configure trigger actions;

- **Trigger Type:** Determined only by the trigger action of configurable input I/O settings, matching "Rising edge" or "Falling edge", or matching "High level" or "Low level" according to the corresponding action;
- **Enable:** function for configurable I/Os, unselected by default; when the "Enable" option is selected, the corresponding function is activated, and I/O information is input and output according to the selected function. If not selected, there is no difference between a configurable I/O and a general I/O. The name of the enabled I/O will be indicated by a small blue circle superscript;

Example:

Configurable Input I/O: If the CI0 configuration activates the "Enable" function, "High level", when CI0 inputs a high level signal, the robot is enabled; while when CI0 inputs a low level signal, the robot is disabled. If Enable is not selected, the function is not activated.

Configurable Output I/O: If CO4 activates the "Motion state" function, when the robot is in motion state, the corresponding CO4 is at a high level; otherwise, CO4 is at a low level. If Enable is not selected, the function is not activated.

5.5 Safety Limits

To set the general restrictions, joint restrictions, safety I/O, and safety collision of the robot arm, it requires verification of the security password (consistent with the administrator password) before operation.

The user can set the safety limit levels for normal and reduced modes in the general restrictions interface. Under normal conditions, the system will operate in one of the following safety modes, each with a safety limit level setting:

Normal Mode: The defaulted safety mode;

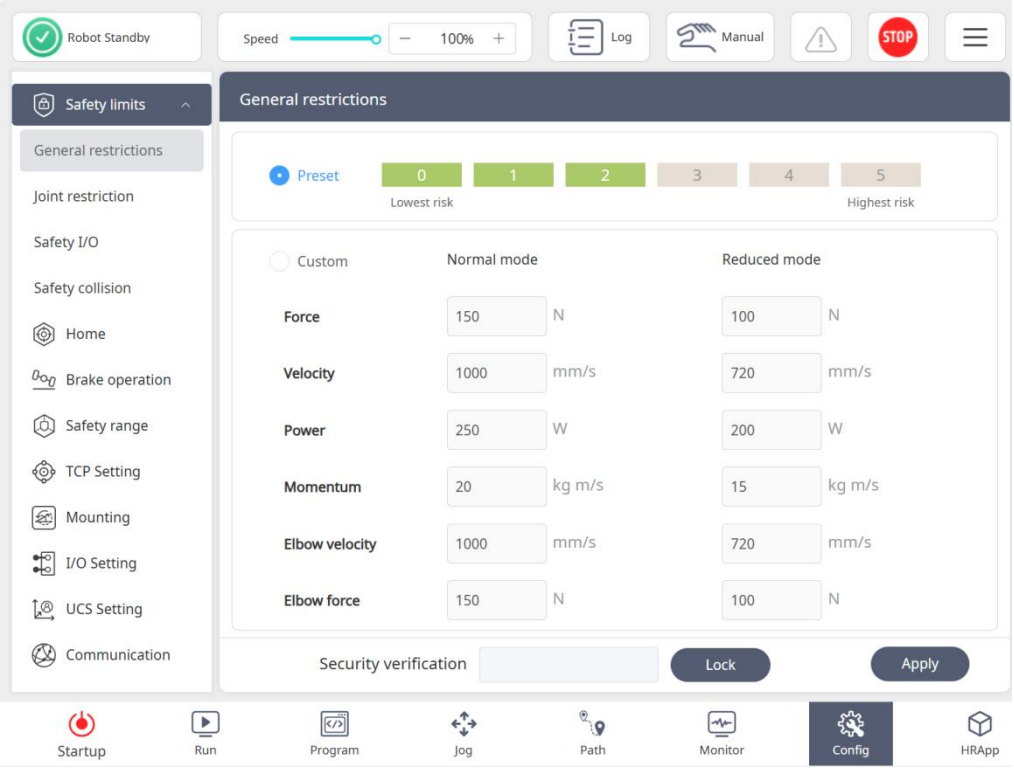
Reduced Mode: A mode with stricter safety restrictions within the range of normal mode. In this mode, the restrictions on speed and momentum are stricter for both the robot arm's end-effector and joints. When activated, the robot arm's movement in joint space will be restricted;

Way to trigger the reduced mode: Through the configurable input I/O signals.

5.5.1 General Restrictions

Different safety levels correspond to different parameters of force, velocity, power, momentum, elbow velocity, and elbow force in the robot normal mode and reduced mode. Each parameter limits for each safety level is displayed in the text box below.

Fig. 5.5.1.1 Safety Level Interface



As shown in the figure above, after verifying the password, select the preset parameters to select the risk level. The parameters corresponding to each level are displayed below, including: force, velocity, power, momentum, elbow velocity and elbow force. Users need to perform a risk assessment and select the appropriate risk level according to the production environment.

Select Custom to enter the custom mode. In custom mode, the parameters can be manually modified to accommodate the actual scenarios.

Note: Parameters must not be altered arbitrarily, as this may result in significant safety risks.

- **Force:** To limit the maximum force exerted by the robot arm in the event of a collision;
- **Velocity:** To limit the maximum velocity of the robot arm;
- **Power:** To limit the maximum mechanical work done by the robot arm to the environment;
- **Momentum:** To limit the maximum momentum of the robot arm;
- **Elbow Velocity:** To limit the maximum velocity of the elbow (J3) of the robot arm;
- **Elbow Force:** To limit the maximum force exerted by the elbow (J3) of the robot arm to the environment;

"High Risk" indicates that a smaller parameter limit is set,; for example, level 5. At this time, the robot arm has lower sensitivity to external collisions. It may require a greater external force to stop it, resulting in a lower safety factor;

"Low Risk" indicates that a large parameter limit is set; for example, level 0. At

this time, the robot arm has higher sensitivity to external collisions. It may require a lower external force to stop it, resulting in a higher safety factor;

There are a total of 6 levels on the safety preset axle to choose from, and the required limit gradually increases from left to right.

After the setting is completed, click "Apply" to take effect. Please "Lock" before exiting.

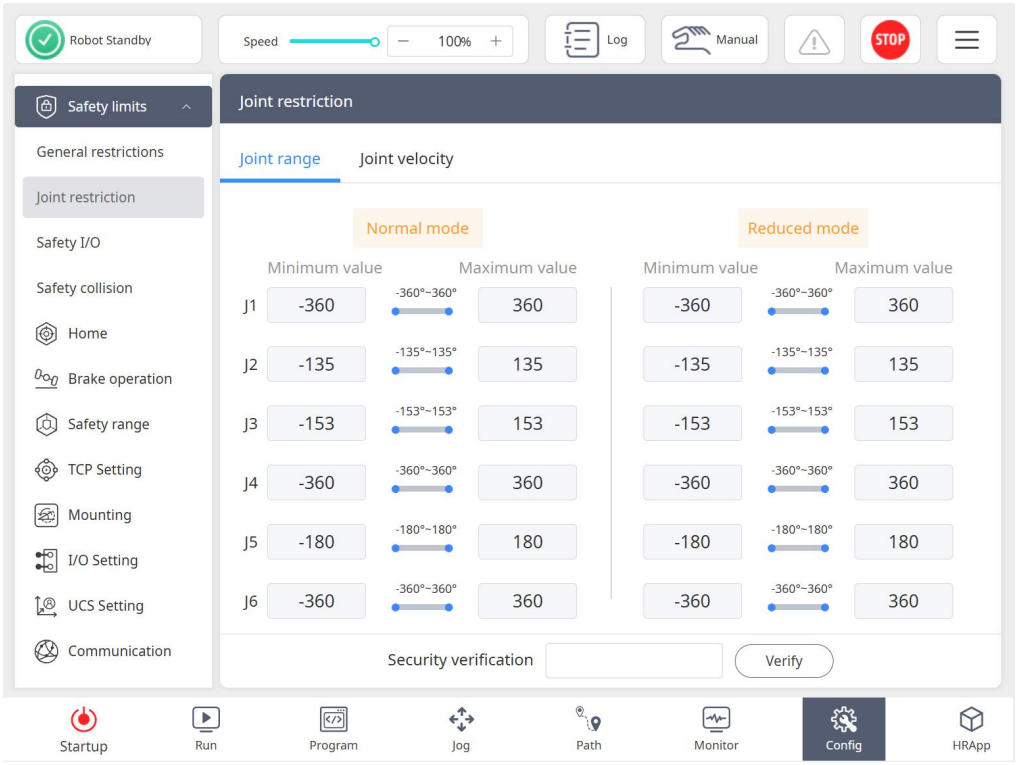
Note: Do not select the stage with the least restriction unless necessary, as which guarantees low safety for the operator and robot.

5.5.2 Joint Restriction

Joint restriction is used to limit the motion of the robot arm joints in the joint space, including the joint range and joint velocity. There are also normal mode and reduced mode restrictions. The safety level limit does not affect the settings of the joint restriction.

Joint range defines the range of positions for each joint.

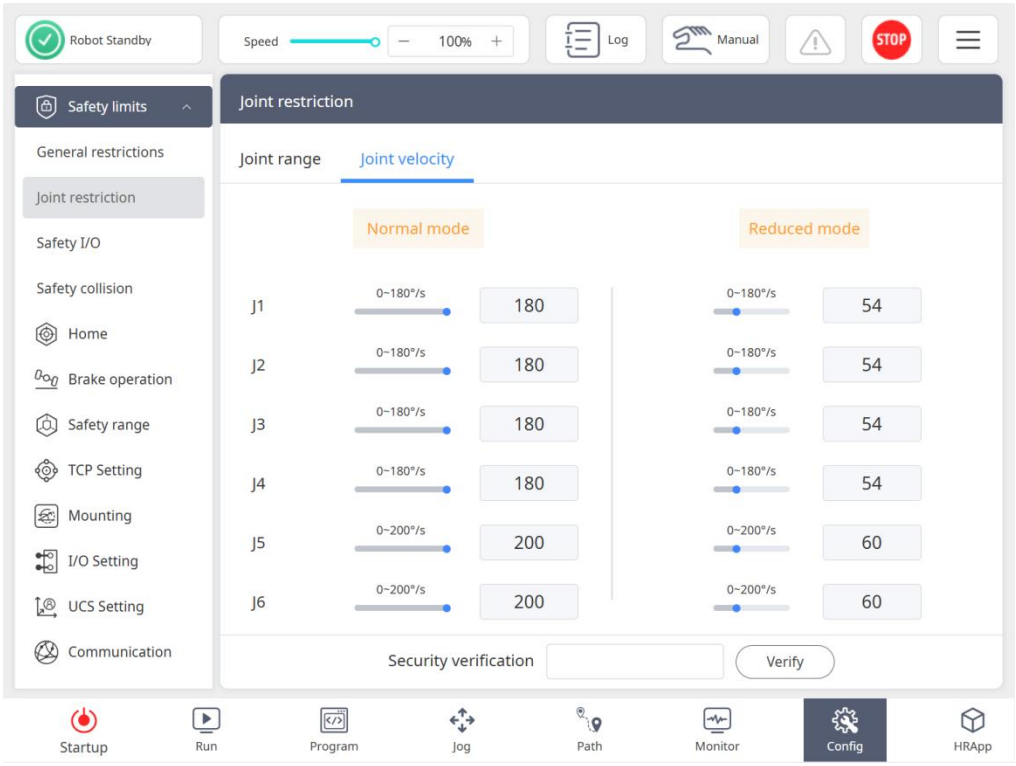
Fig. 5.5.2.1 Joint Range Restriction Interface



– The default joint angle range of the robot arm is shown in the interface.

Joint velocity defines the maximum angular velocity of each joint.

Fig. 5.5.2.2 Joint Velocity Limits Interface

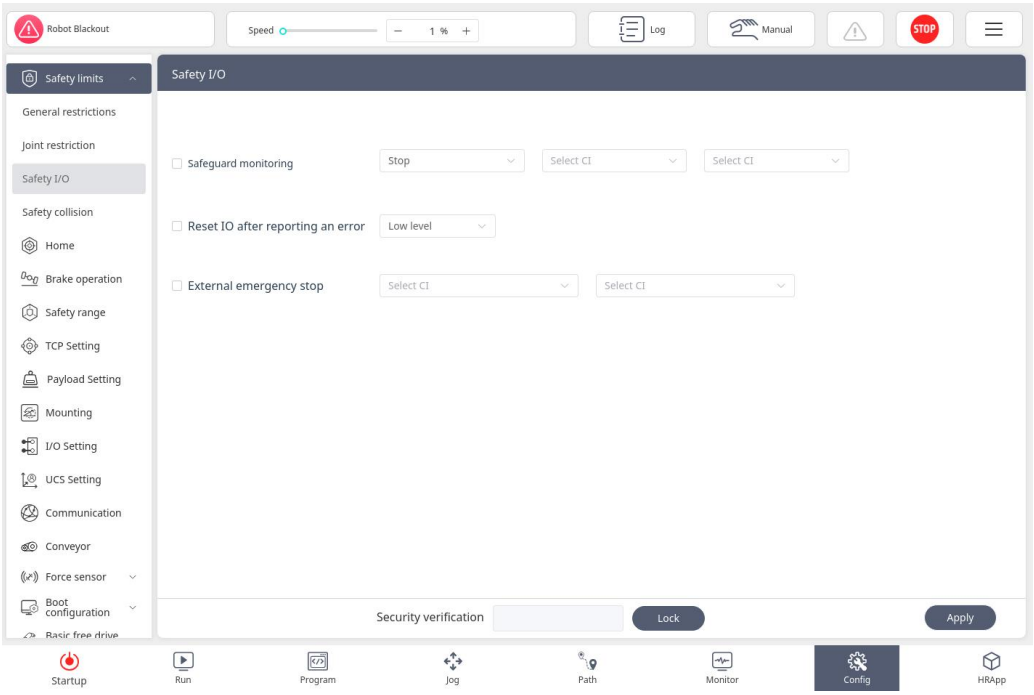


– The default maximum joint angular velocity of the robot arm is shown in the interface.

After the setting is completed, click "Apply" to take effect. Please "Lock" before exiting.

5.5.3 Safety I/O

Fig. 5.5.3.1 Safety I/O Interface



Note: This function supports adaptation to various types of electrical control boxes, and the configuration rules for dedicated I/Os of safeguard monitoring for different

models of electrical control boxes are as follows: the 4th-generation electrical control boxes can flexibly designate two CIs as dedicated I/Os for safeguard monitoring (refer to Figure 5.5.4 for the safety I/O interface); the 3rd-generation and unified electrical control boxes have fixed-configured dedicated I/Os for safeguard monitoring with no need for CI configuration (refer to Figure 5.5.5 for the safety I/O interface), among which the two dedicated I/Os for safeguard monitoring of 3rd-generation electrical control box are SDI1+ and SDI2+, and those of unified electrical control box are SD01 and SD21.

- **Setup Steps:**
  - Connect the corresponding I/O ports in the electrical control box to external devices using cables.
  - Enter the “Safety I/O” interface to complete the settings for safeguard monitoring, error reset I/O, and local emergency stop;
  - If it is a 4th-generation electrical control box, additionally configure the selected CI ports in the “Safety I/O” interface when setting up safeguard monitoring.
- **Reset I/O After Reporting an Error:** After checking, if the robot reports an error, the output I/O of the electrical control box will be uniformly set to high or low;
- **Safeguard Monitoring:** Check to start safeguard monitoring, and configured to stop, pause program running or stop jogging when triggered;
- **"Enable Safeguard Monitoring Stop":** When the safety I/O signal is monitored, the motion stops and a popup appears to indicate the safety protection mode. The user needs to release the protective switch or turn off the signal, and then click "Reset" to clear the error before operating the robot arm again. After reset, the robot arm remains powered;
- **"Enable Safeguard Monitoring with Jogging Allowed"**: When the safety I/O signal is monitored, the robot arm stops moving and the safeguard monitoring state is displayed, and jogging and configuration editing are allowed. The user needs to release the protective switch or turn off the signal, and then click "Reset" in the running interface;
- **"Enable Safeguard Monitoring Pause"**: When the safety I/O signal is monitored, the program will be paused. The user needs to release the protective switch or turn off the signal, and then click "Reset" to continue running the current program;
  - **External Emergency Stop:** When selected, configured to stop or pause program running if triggered;

When the safety I/O signal is monitored, the motion is stopped and a popup appears to indicate the safety protection mode. Only when both preset detection ports are restored to the high levels can the emergency stop be reset, and then click the "Reset" button to operate the robot arm again.

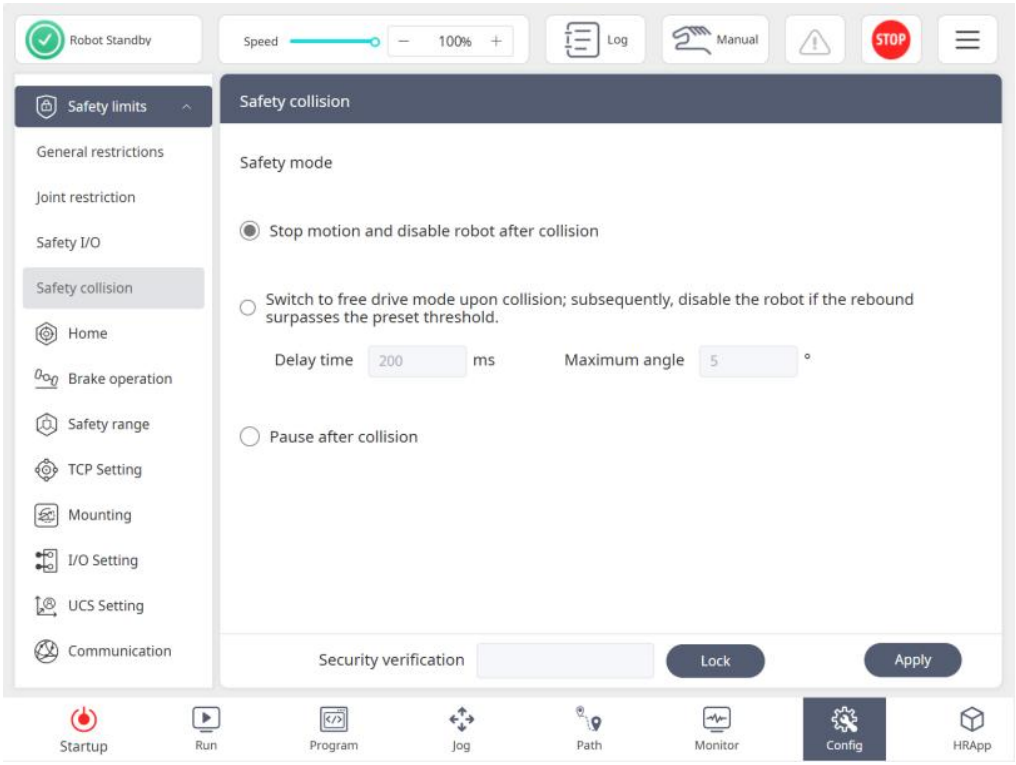
After the setting is completed, click "Apply" to take effect.

5.5.4 Safety Collision

When the robot arm encounters an obstacle, it will select different motion states according to the force fed back by the obstacle. The threshold value of the force is determined by the level configured in the safety limits, and the user can configure the safety risk according to the actual needs and refer to "5.4.1. General Restrictions". Meanwhile, the user can also select the motion state of the robot arm after the collision. The optional modes are as follows:

- Stop Motion and Disable Robot After Collision: After collision, the robot arm stops moving and pop up to remind the user click "Reset" to clear the error before re-operating the robot arm. The robot is disabled after reset.
- Switch to Free Drive Mode Upon Collision; Subsequently Disable the Robot If the Rebound Surpasses the Preset Threshold: In this mode, the robot arm encounters and collides with an obstacle during operation and enters the basic free drive mode. However, when the rebound caused by the collision exceeds the preset value (the default value is a delay of 200ms, a maximum angle of 5°), it will stop moving and pop up to remind the user to click "Reset" to clear the error before re-operating the robot arm. After reset, the robot is disabled.
- Pause After Collision: After collision, the robot arm stops moving and a popup appears to notify the user to click "Reset" to clear the error and continue running the current program after clearing the obstacles.

Fig. 5.5.4.1 Safety Collision Interface



The Mode 1 is the User Default Mode: "Stop motion and disable robot after collision".

5.6 User Coordinate System

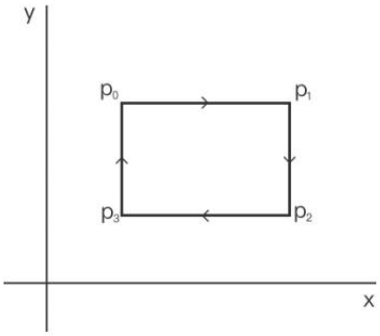
The user coordinate system can be set by points, lines and planes. The user-defined coordinate system enables the robot to work in collaboration with different workbenches or fixtures.

Application Significance: In practical applications, the robot may need to work in different locations, and the ground in the new scenario may be tilted, or even the robot may be in a flipped state. Due to the tilt or rotary scenarios, the original motion path may no longer work.

In new scenarios, if it is necessary to continue using the preset script program while ensuring that the waypoints in the script remain unchanged, it can be achieved simply by modifying the corresponding user coordinate system. The following illustration shows the user coordinate system impact on the path:

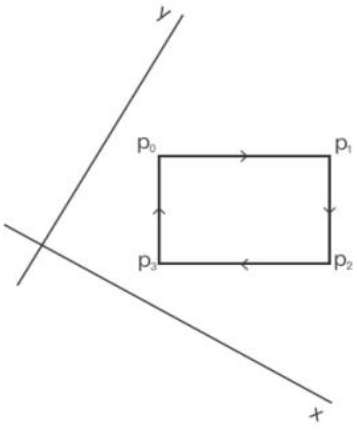
- The original path of the robot teaching;

Fig. 5.6.1 Path in Default Scenario



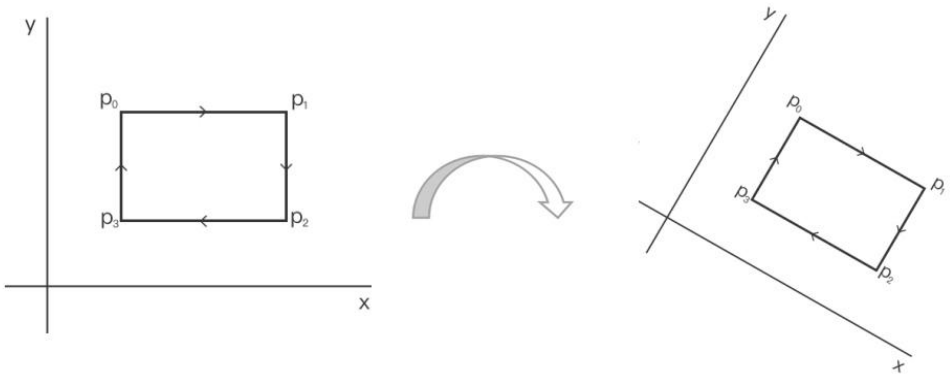
- After changing the robot scenario, if the current coordinate system is continuously used, the motion path of the script is as shown below, which is not desired by the user;

Fig. 5.6.2 Path after Changing Scenario



- The significance of custom coordinate system: the robot shifts relatively as a whole with the change of coordinate system, and continues the original path operation;

Fig. 5.6.3 Custom Coordinate System Effects

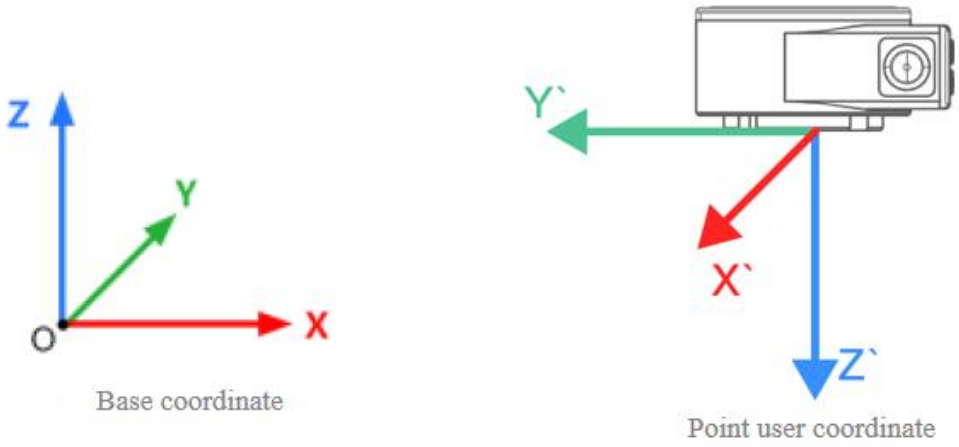


5.6.1 Point User Coordinate System

Click "+ point", and the "Point _N" point coordinate system for editing will be displayed in the right area, as "Point_1" as shown in the figure below: Click "Set point" to set the point coordinate system.

The point user coordinate system is a user coordinate system established based on a point, which is the origin of the user coordinate system; and the position and direction of the origin of the user coordinate system at this point are the values of the current TCP in the Base coordinate system.

Fig. 5.6.1.1 Point User Coordinate System



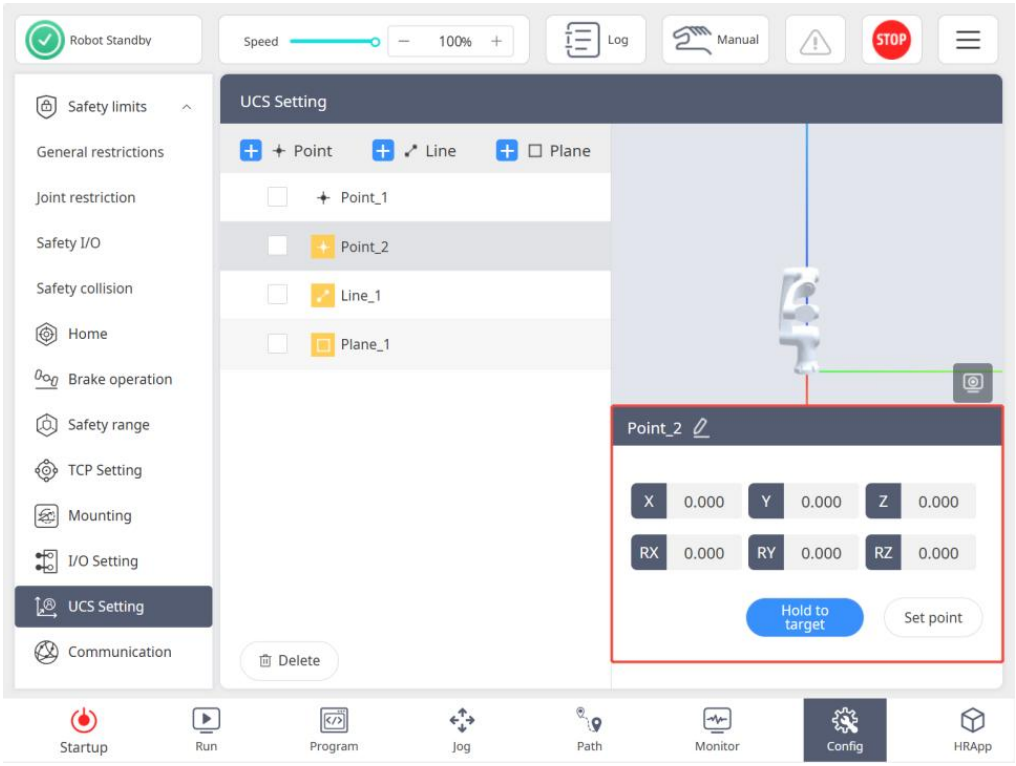
The specific calculation steps are as follows:

- Point1 is taught under the base coordinate;
- Determining the position and direction of the taught point at the current

end-effector;

- The position and orientation of the point user coordinate system are the values of the current TCP under the Base coordinates.

Fig. 5.6.1.2 Point User Coordinate System Design



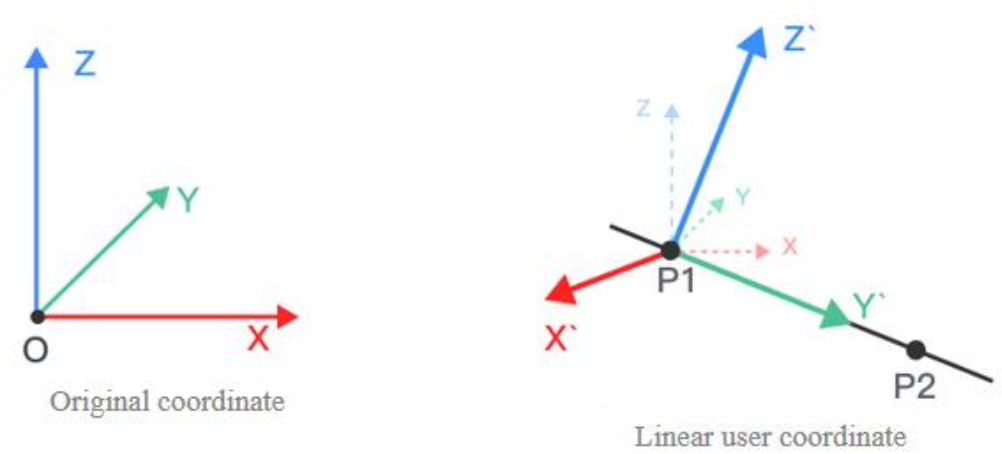
- **Rename:** Click the edit icon next to Point_N, and enter a new name and confirm;
- **Set Point:** Click "Set point" to jump to the jog interface and set the point;
- **Hold to Target:** Press and hold this button to move the robot arm to the taught target point under the selected function. Once the movement is complete, the coordinate system of the function will align with the TCP (Tool Center Point) coordinate system;
- **Delete:** Check the coordinate system in the list and delete it; after deleting the coordinate system, if the coordinate system is used in the programs, the related parameters in the programs will be deleted synchronously;

Note: If the user coordinate system used by a waypoint is deleted, the data in the waypoint will be deleted synchronously. The waypoint cannot be called again, and the user needs to reedit the waypoint.

5.6.2 Line User Coordinate System

Click "+ Line", as shown in the figure below, to add the line function Line_1. The characteristics of the straight line can be calculated by two taught points, and the origin position of the line user coordinate system of the straight line is the same as that of the first taught point.

Fig. 5.6.2.1 Linear User Coordinate System

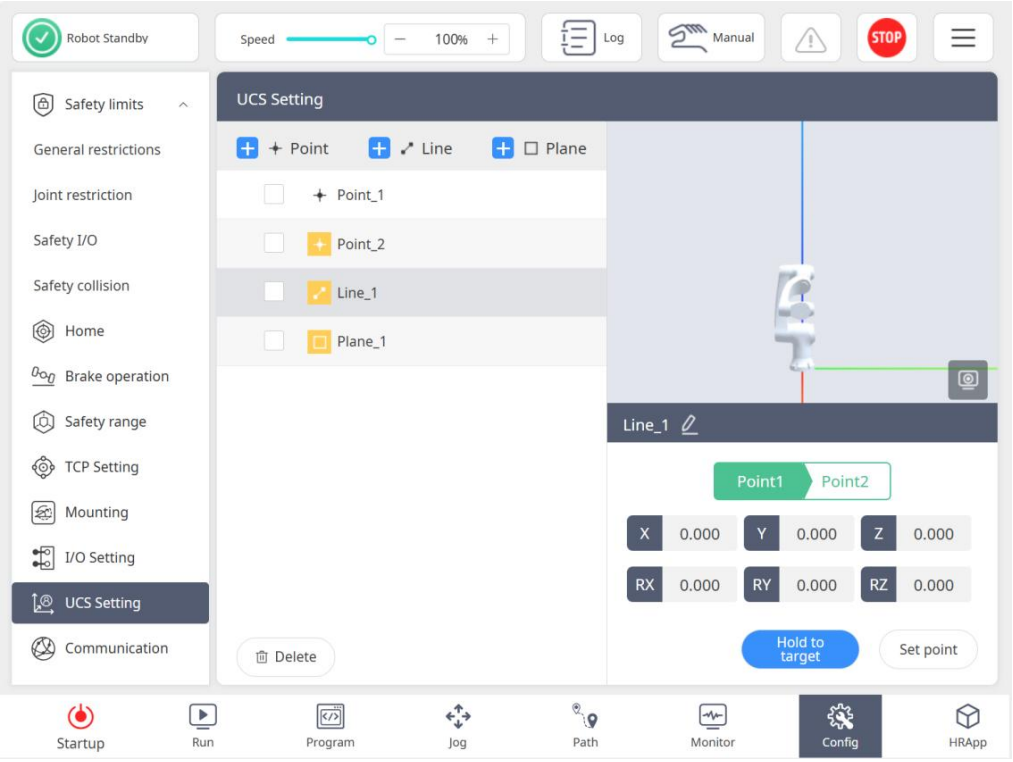


The specific calculation steps are as follows:

The first point P1 points to the direction of the second point P2, that is, the Y+ axis of the linear user coordinate system; the Z+ axis of the linear user coordinate system is related to the Z+ axis of the original TCP coordinate.

According to the right-hand rule, the X+ axis of the line User Coordinate System of the straight line can be calculated.

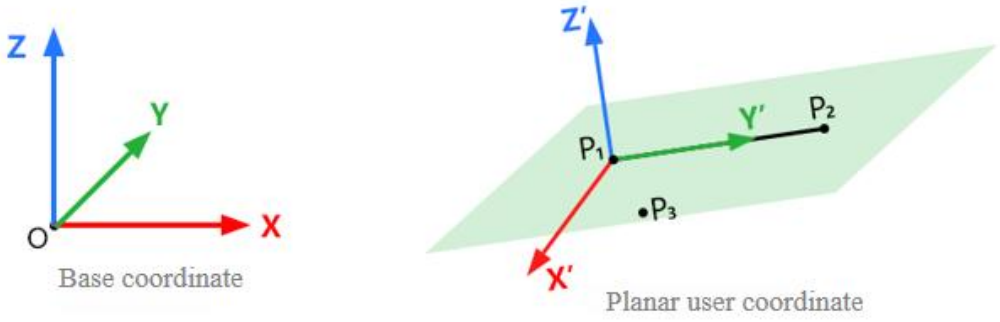
Fig. 5.6.2.2 Linear User Coordinate System Interface



5.6.3 Plane User Coordinate System

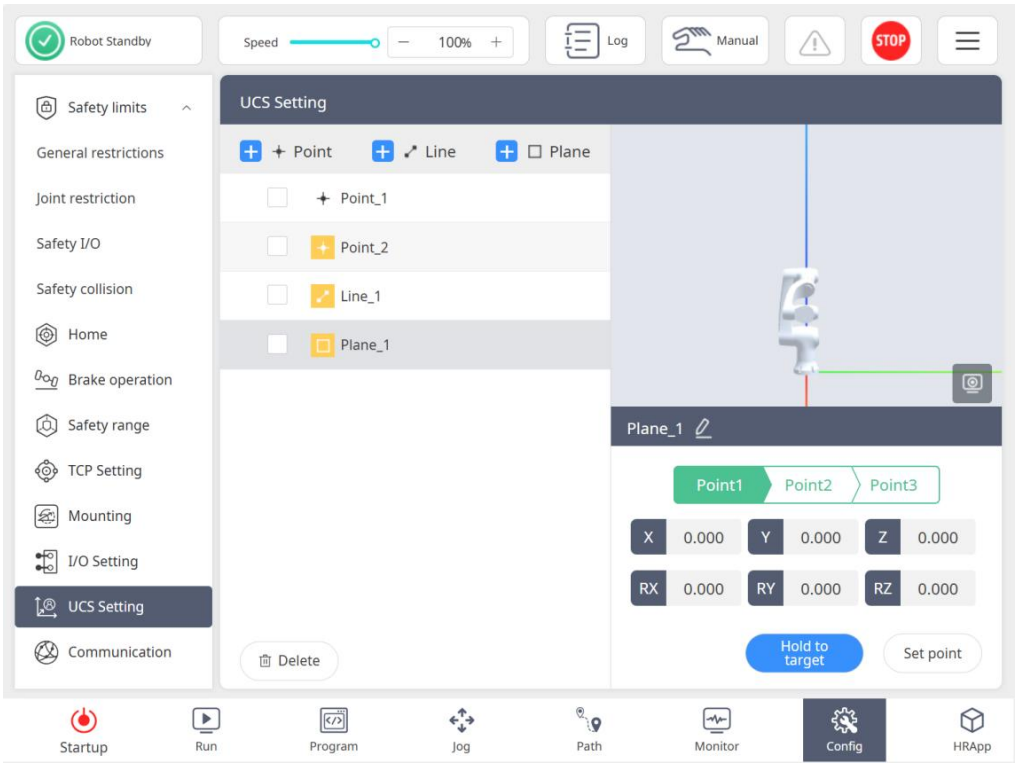
Click "+ Plane", as shown in the figure below, to add the plane function Plane_1. The plane is defined by the functions of three taught points. The position of the origin of the coordinate system is the same as that of the first taught point.

Fig. 5.6.3.1 Planar User Coordinate System



- Moving the robot end-effector to point P1;
- Moving the robot end-effector to point P2, the direction of P1 pointing to P2 is the +Y axis of the user coordinate system;
- Continue to teach point P3. At this time, the three points form a plane called Plane_1, and point P3 determines the +X axis (which is perpendicular to P1 and P2, and pointing to P3);
- The Z axis of the newly built user Plane_1 is perpendicular to the plane formed by the three point, and the +Z axis can be determined according to the right-hand rule.

Fig. 5.6.3.2 Plane User Coordinate System Interface



5.7 Safety Range

The safety range can be set as a restricted area through plane mode or space mode, and the robot arm is prohibited from entering. This range further constrains the maximum space in which the robot arm's end-effector can operate.

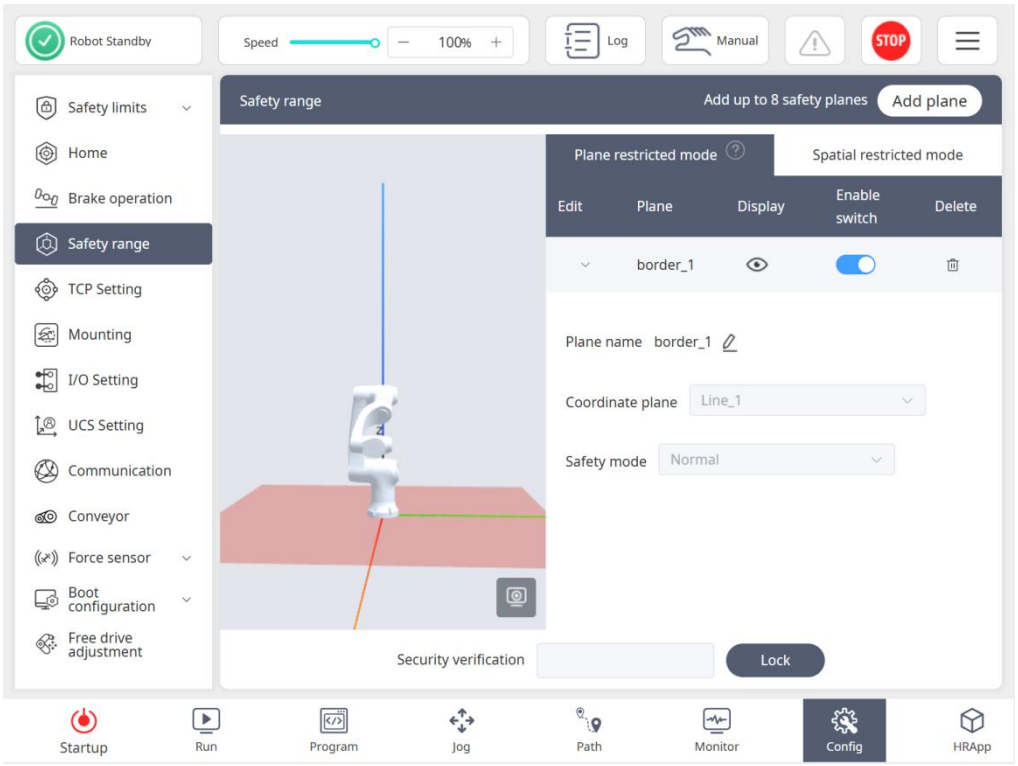
When the robot arm's end-effector moves within the safe motion range, if the end-effector touches the restricted area, the robot arm will stop moving and report an error. Upon the occurrence of an error, click "Reset" to retract the robot arm's end-effector and move it back to the safety zone before normal operation can resume.

When the robot arm's end-effector is within the restricted area, the system will monitor the motion trend in real time:

- Away From Restricted Area: If the end-effector motion trend is far away from the restricted area, the robot arm is allowed to continue moving until it enters the safety range;
- Deep into Restricted Area: If the end-effector motion trend is deep into the restricted area, it will be regarded as triggering the safety space restriction, and the robot arm will immediately stop moving and report an error.

Note: Security verification is required before setting.

Fig. 5.7.1 Safety Range Interface



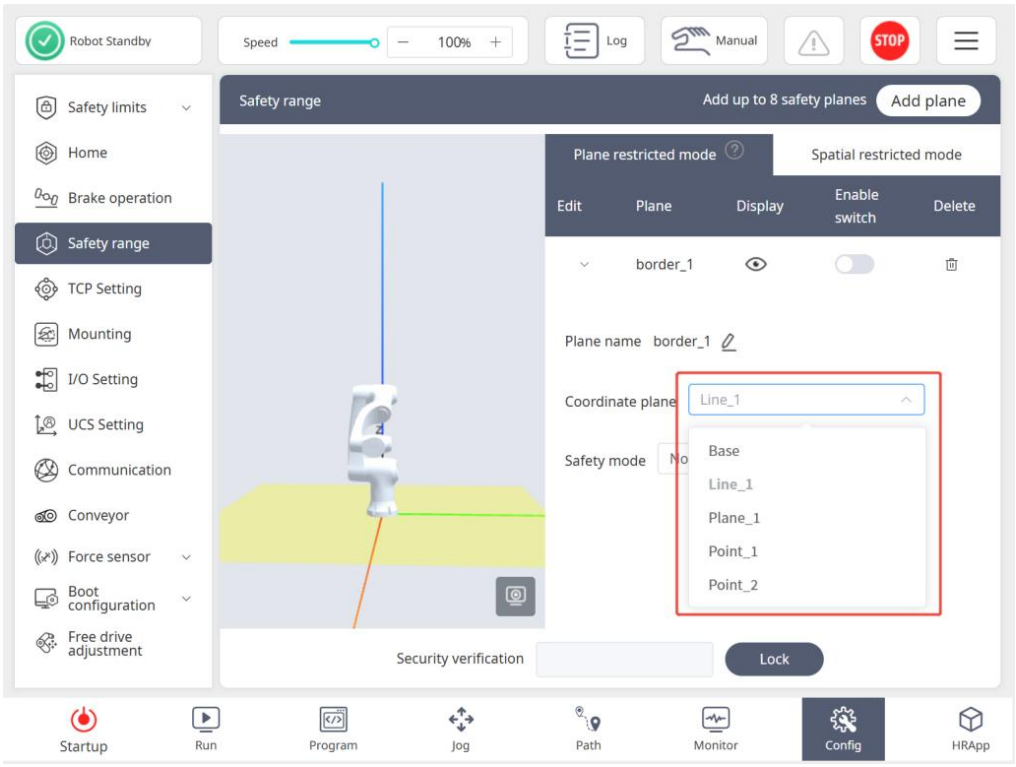
5.7.1 Plane Restricted Mode

Safety plane: A plane is defined by the user coordinate system and coincides with

the XY plane of the selected UCS. The -Z direction is the safety area, and the +Z direction is the restricted area.

When enabled, if the robot arm's end-effector is close to or in the restricted area, the safety plane becomes red; while when the robot arm's end-effector is not close to or is not in the restricted area, or the safety plane is not enabled, the safety plane turns yellow.

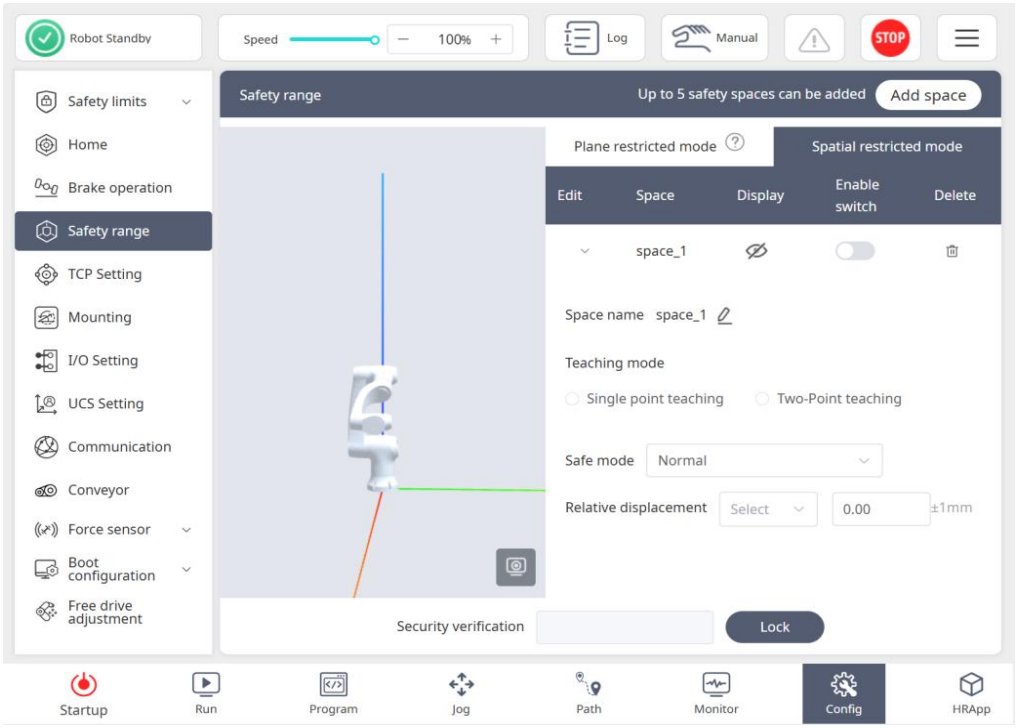
Fig. 5.7.1.1 Plane Mode



5.7.2 Space Restricted Mode

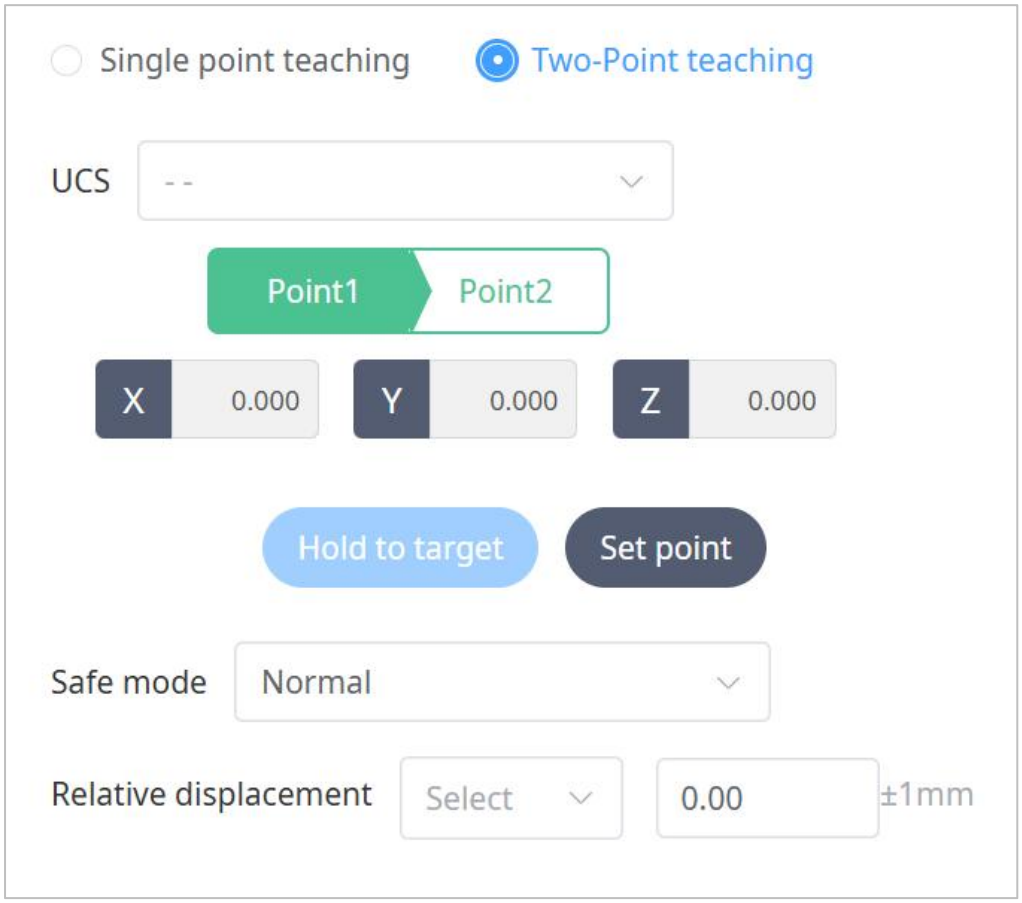
Safety Volume: Eithin the robot's range of motion, the user can define a cuboid volume inside of which the robot cannot operate.

Fig. 5.7.2.1 Set Restricted Space



The Restricted Space can be defined in two ways: single-point teaching and two-point teaching.

Fig. 5.7.2.2 Two-point Teaching in Space Mode

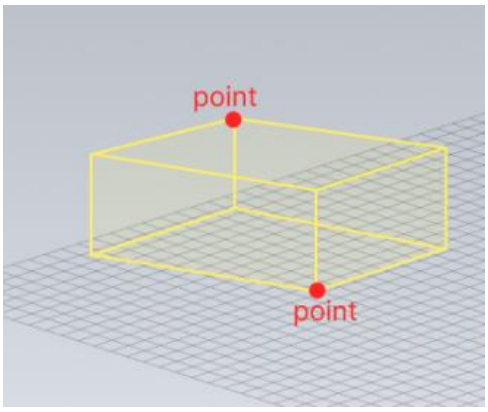


Two-Point Teaching: A rectangular restricted space is established by teaching two points. The two taught points must be relative to the same User Coordinate

System,which is displayed in the interface and can be selected from the UCS list.

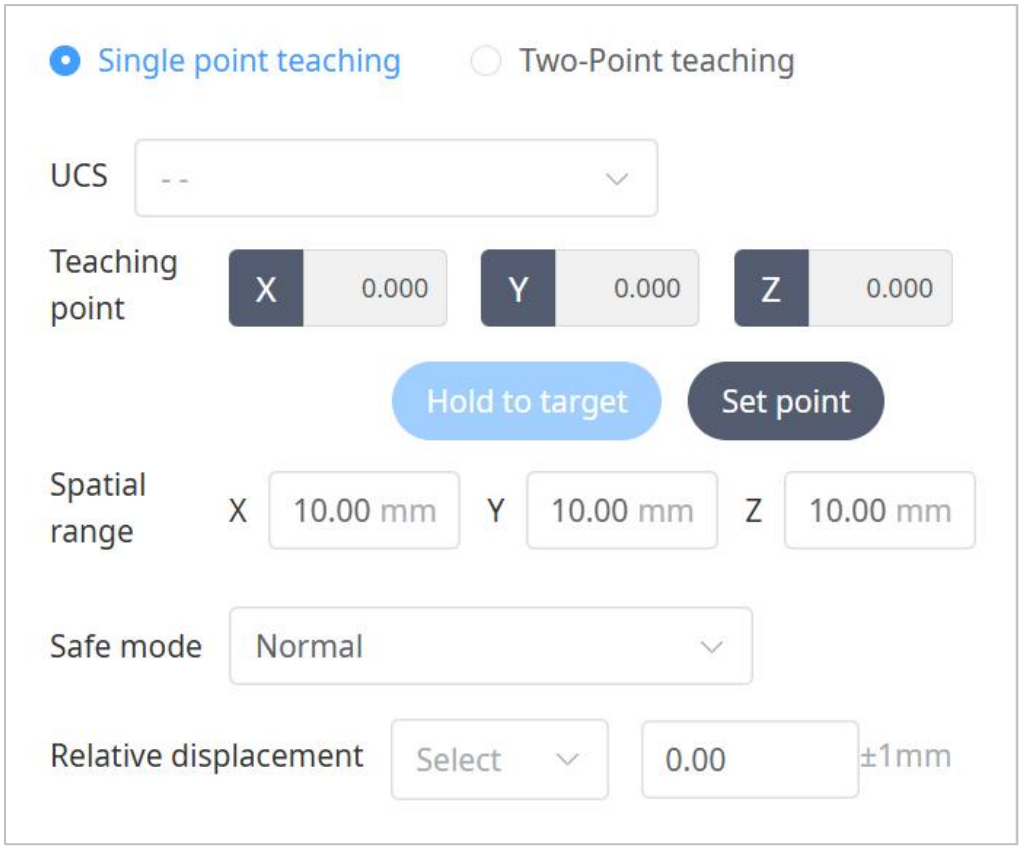
After teaching two points located in the same coordinate system, the system typically constructs a cuboid with these two points as diagonal points to define a restricted space range.

Fig. 5.7.2.3 Two-Point Teaching Space



In the process of teaching, the user must complete the teaching of point 1 before teaching point 2.

Fig. 5.7.2.4 Space Mode Single-Point Teaching

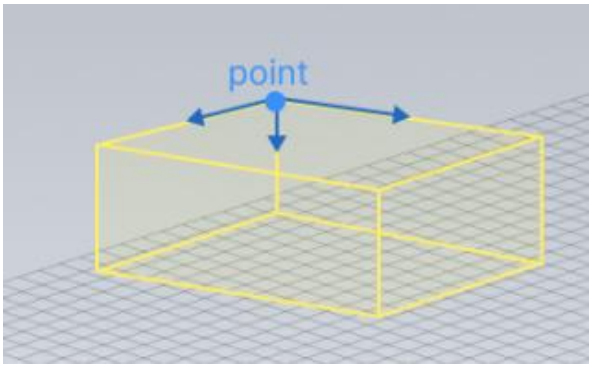


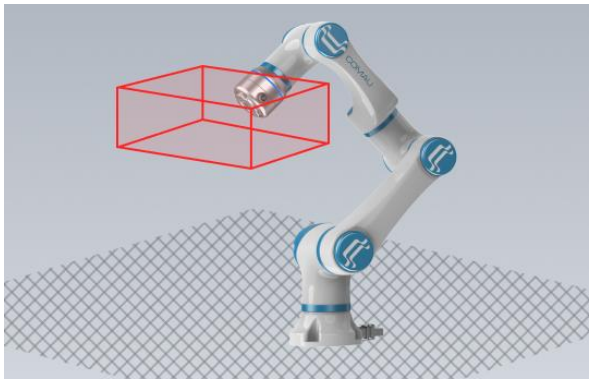
Single-Point Teaching: After teaching a point, the extension ranges of the XYZ axes of the current point is set, and the teaching of the restricted space is completed.

In the teaching process, the points are taught based on the user coordinate system. The system typically constructs a cuboid by using these two points, the

taught point and the origin of the coordinate system, as diagonal points to define a restricted space range. This cuboid space corresponds to the current teaching - defined restricted space.

Fig. 5.7.2.5 Single-Point Teaching Space

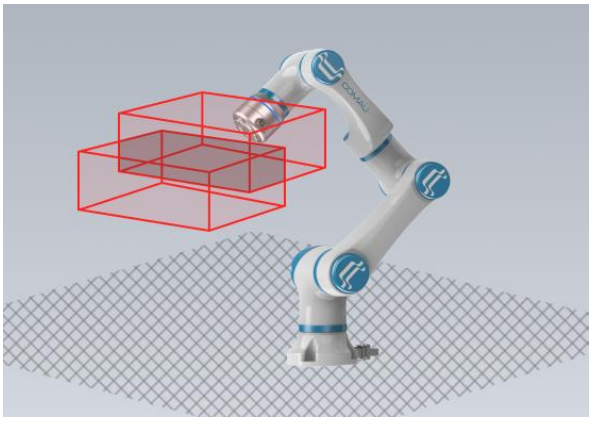




End-Connector Contact Space

- When the robot arm's end-effector position is in a cuboid restricted space, the system will mark the center point of the cuboid space. If the motion trend of the robot arm's end-effector is moving away from the center point of the cuboid space, it will be regarded as moving away from the cuboid and can move normally to a safe range.
- If the motion trend of the robot arm's end-effector moves toward to the center point of the cuboid space, it will be regarded as triggering the safety space restriction, the robot arm will be stopped and an error will be reported.

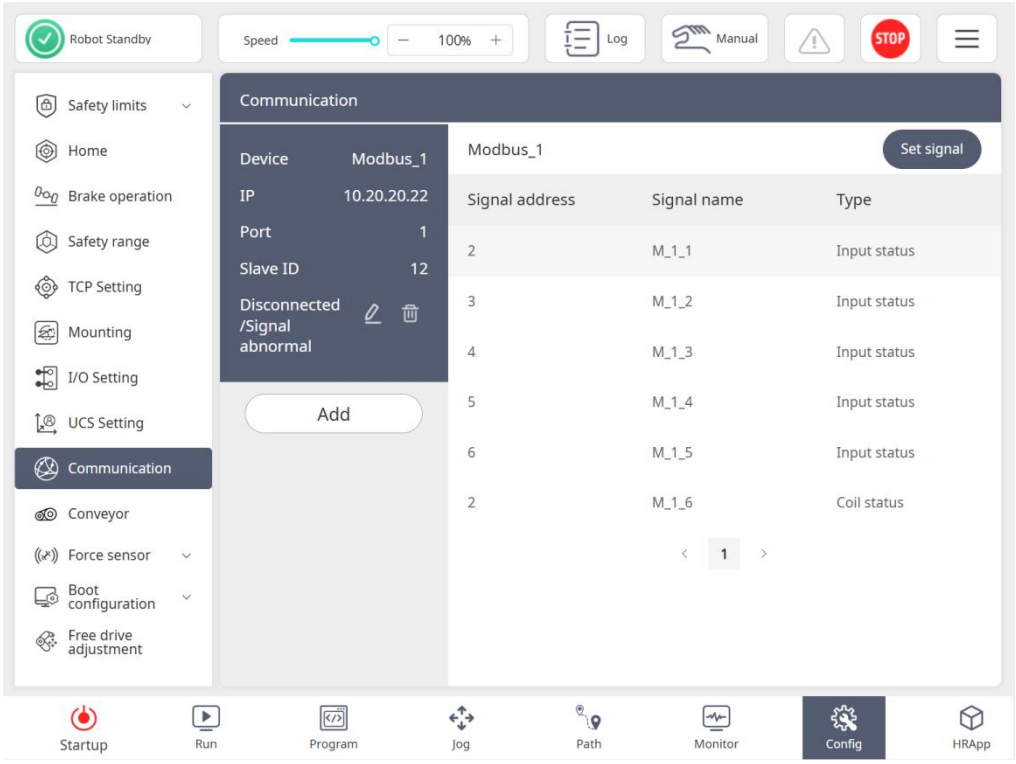
Fig.5.7.2.6 Illustration of Multiple Overlapping Spaces



– When the robot arm's end-effector is within a compound space formed by overlapping multiple cuboid spaces, it is regarded as a safety space error. The compound space's restrictive will be disabled and the robot arm can move freely. To re-enable the restrictive function of the compound space, move the robot arm back into the safety range.

5.8 Communication (Modbus)

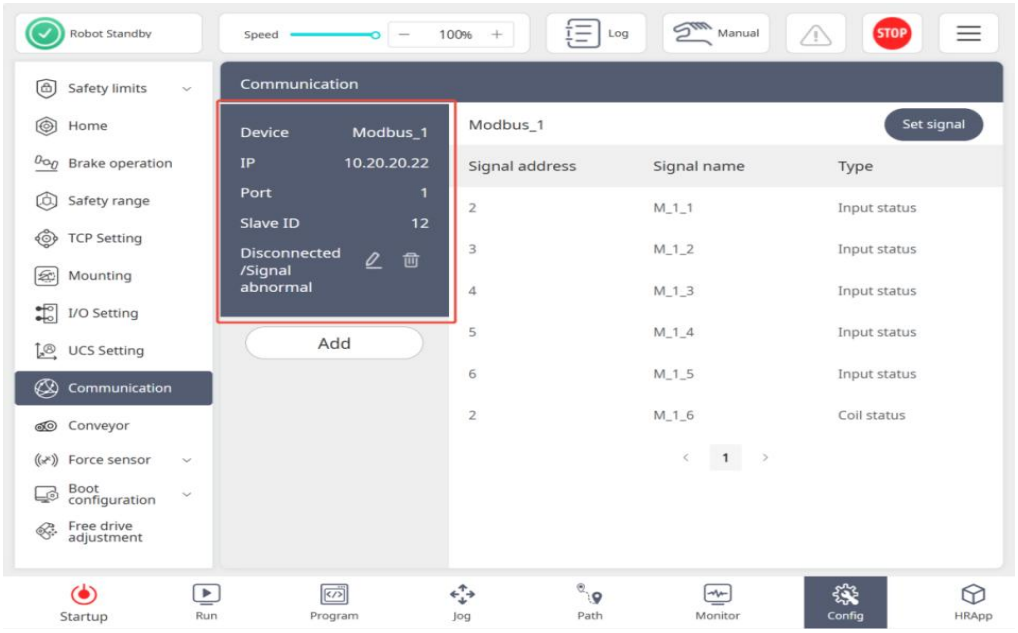
Fig. 5.8.1 Communication nterface



To set the Modbus external slave device signal. A connection to a Modbus server (or slave) of a specified IP address can be created by input/output signals (registers or numbers). Each signal address is unique.

5.8.1 Device List

Fig. 5.8.1.1 Device List

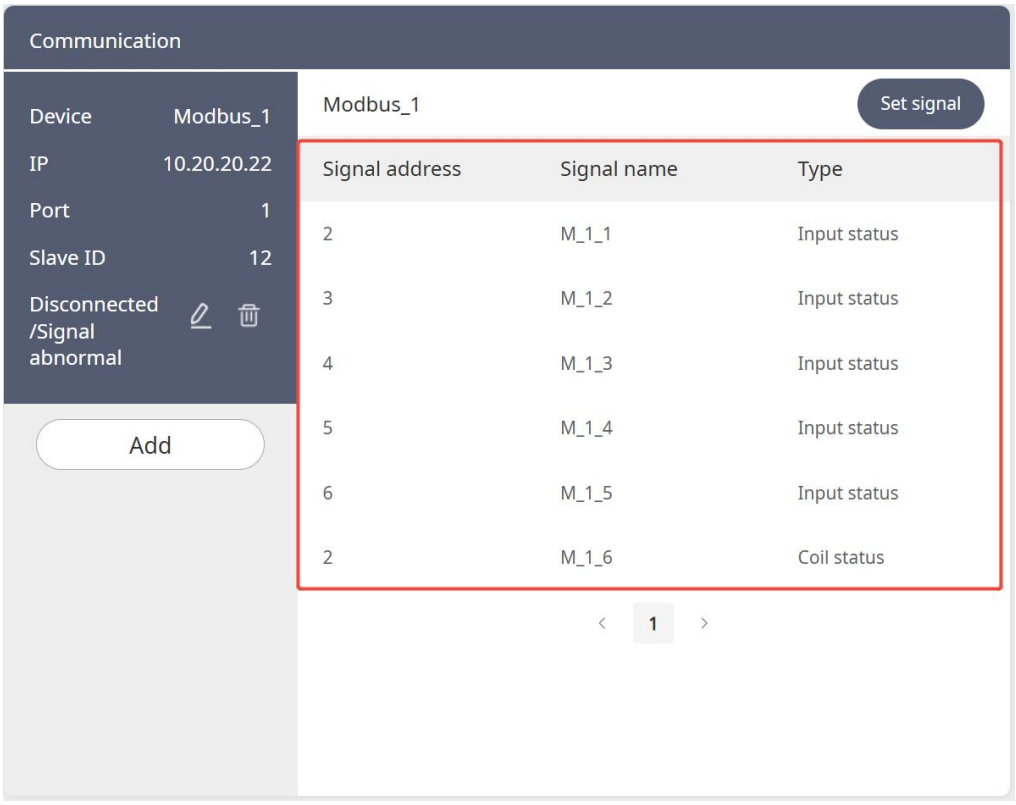


- **Add:** Click Add device, and a popup appears for data entry, including the IP address, port, and slave ID; the Modbus name is not modifiable;
- **Edit:** To modify the Modbus device parameters;
- **Delete:** To delete the Modbus device;
- **Select:** Dark color means selected, and the signal list of the selected device is displayed on the right;

5.8.2 Signal List

The signals in the list are arranged by the signal group intervals added;

Fig. 5.8.2.1 Signal List



Set Signal: Click "Set signal" to edit signal group parameters, including signal group, signal type, address, read quantity and signal name;

Fig. 5.8.2.2 Signal Setting



- **Add a Group:** Add a group of signals of the same type;
- **Type:** Use this drop-down menu to select the signal type.

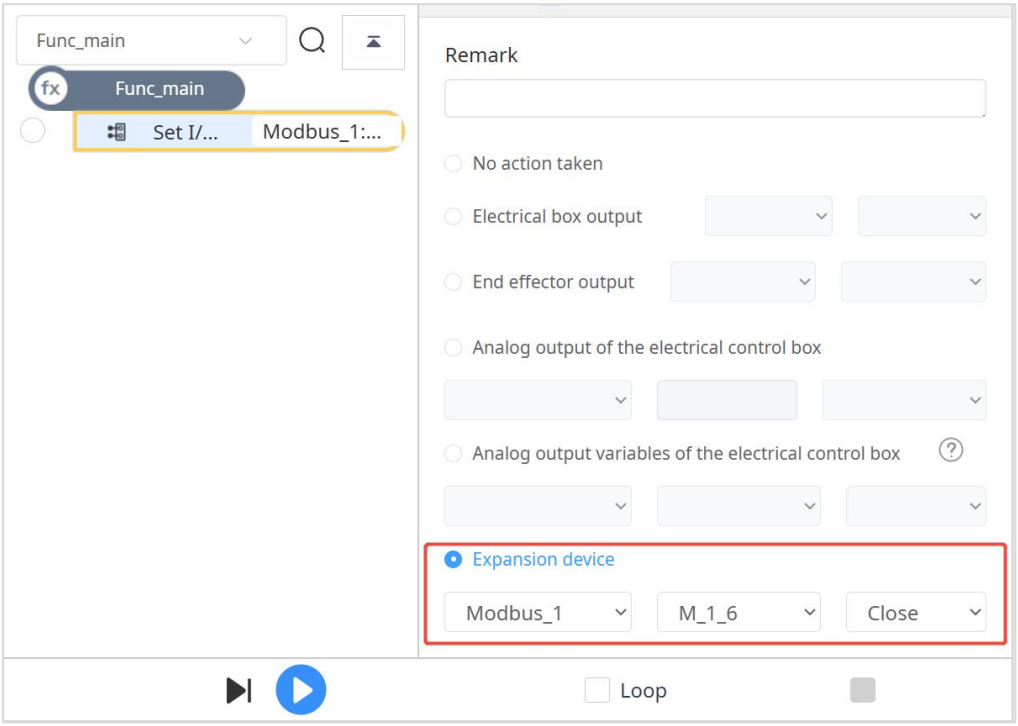
Table 5.8.1 List of Signal Types

Signal Type	Target	Access Type	Application Scenario
Coil Status	Single bit	Read and write	For controlling/alarming switch status
Input Status	Single bit	Read only	For input diagnostic functions
Holding Register	16-bit	Read and write	For transmitting user-set parameters
Input Register	16-bit	Read only	For storing input signal values

- **Signal Address:** To enter the signal address to be read, which is the starting value of the signal group;
- **Read Quantity:** To input the quantity to be read from the beginning to the end of the signal address; for example, if the signal address is 00100, and the number of reads is 3, then the signal within value range 00100 ~ 00102 is read;
- **Signal Name:** To facilitate user search, duplicate naming is not permitted;
- **Unfold:** Fold and collapse multiple signals in a group; click again to unfold;
- **Delete:** Check the box in front of the signal group name to delete the entire signal group;
- **Apply, Cancel:** Click apply, the configuration modifications are implemented immediately upon confirmation. Click cancel, it will revert pending adjustments and delete unsaved input.

Note: After the Modbus signal is added, the SetI/O command will be directly available; in addition, the monitoring interface will also be added a Modbus monitoring module accordingly.

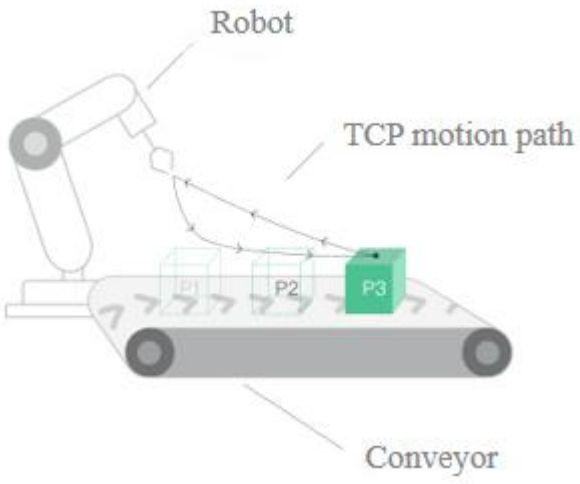
Fig. 5.8.2.3 Application Demonstration within the Program



5.9 Tracking Conveyor

The purpose of the tracking conveyor is to adjust the movement of the robot and perform a synchronous tracking of the conveyor by the robot.

Fig. 5.9.1 Tracking Conveyor



5.9.1 Conveyor Encoder Connection

This document provides detailed specifications and connection methods for the conveyor encoder in relation to the control cabinet. Other hardware interfaces or functionalities are beyond the scope of this documentation.

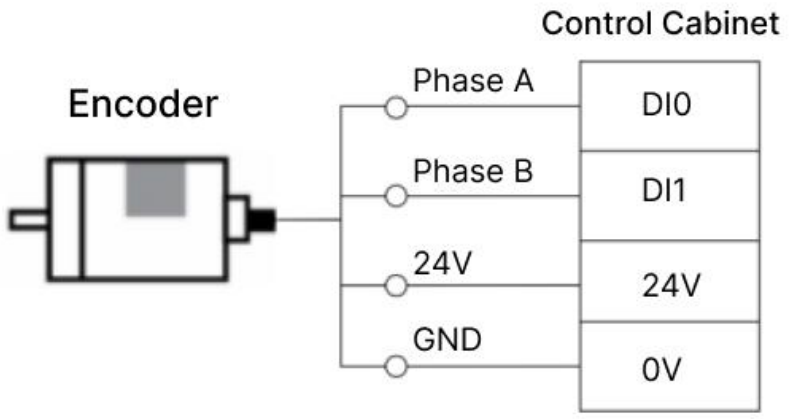
1. Encoder Technical Parameters

Parameter	Specification
Type	Incremental Encoder (PNP Output Type)
Recommended Resolution	2000 pulses per revolution (2000 P/R)

Note: It is recommended to operate within the rated voltage range and adopt the recommended resolution to ensure measurement accuracy and system compatibility.

2. Electrical Connection Definition

Fig. 5.9.1.1 Connection Diagram of Encoder and Control Cabinet



Connection Diagram of
Encoder and Control Cabinet

Encoder Signal Terminal	Control Cabinet IO Terminal	Remarks
Phase A	DI0	-
Phase B	DI1	-
24V	24V	An external power supply may be used, but both circuits must share the same source.
GND	0V	

Note:Ensure wiring strictly follows the correspondence in the table above. Maintain correct polarity and reliable contact.

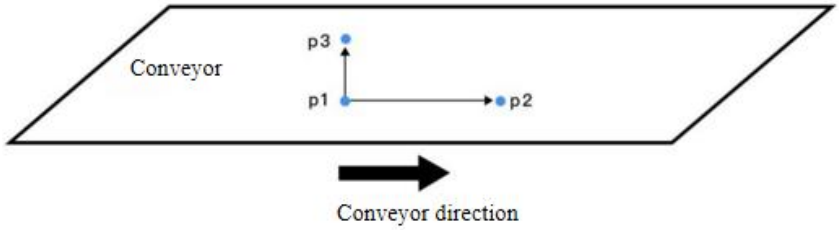
3. Connection Steps (Perform with Power Off)
(1) Confirm the encoder output type is PNP voltage output.
(2) Connect the encoder Phase A to the control cabinet DI0 terminal.
(3) Connect the encoder Phase B to the control cabinet DI1 terminal.
(4) Connect the encoder power supply 24V to the control cabinet 24V terminal.
(5) Connect the encoder power supply GND to the control cabinet 0V terminal.

4. Connection Verification
After powering on, rotate the encoder and observe whether the DI0/DI1 indicator lights on the controller software's front-end monitoring interface flicker accordingly.

5.9.2 User Coordinate System Setting

When the hardware conveyor is connected (please refer to the manual for how to connect the hardware conveyor), the conveyor needs to be calibrated in order to ensure that the direction and movement of the robot arm are synchronized with the conveyor. To calibrate the conveyor, it is necessary to ensure that the Y-axis motion direction of the robot arm is consistent with the motion direction of the conveyor, so it is necessary to teach a user coordinate first.

Fig. 5.9.2.1 Conveyor User Coordinate System



With the conveyor stopped, the user coordinate system can be defined by setting three points (P1, P2, P3):

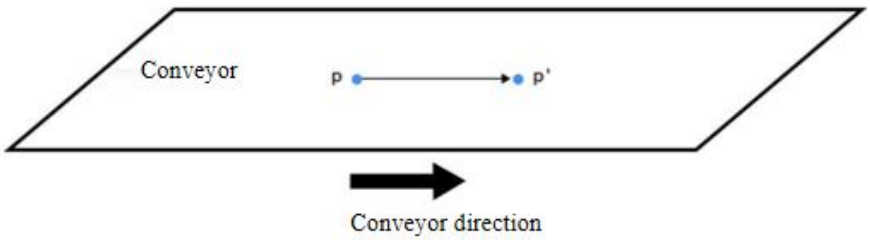
- Move the robot arm's end-effector to point P1 and set it as Point1;
- To ensure that the +Y axis of the user coordinate system is consistent with the moving direction of the conveyor, the conveyor can be started;
- Move the robot arm's end-effector to point P2 and set it as Point2;
- Determine point P3 on the conveyor plane (P3 cannot be collinear with P1 and P2), and move the robot arm's end-effector to point P3 and set it as Point3.

After the above steps are completed, the user coordinate system teaching is completed.

Note: The positive direction of the Y axis of the user coordinate system is the same as the running direction of the conveyor, and the plane formed by the X and Y axes is parallel to the conveyor plane.

5.9.3 Conveyor Calibration

Fig. 5.9.3.1 Conveyor Calibration



Once the user coordinate system is set, it needs to be set to the current user coordinates. Before calibrating the conveyor, configure the conveyor type and

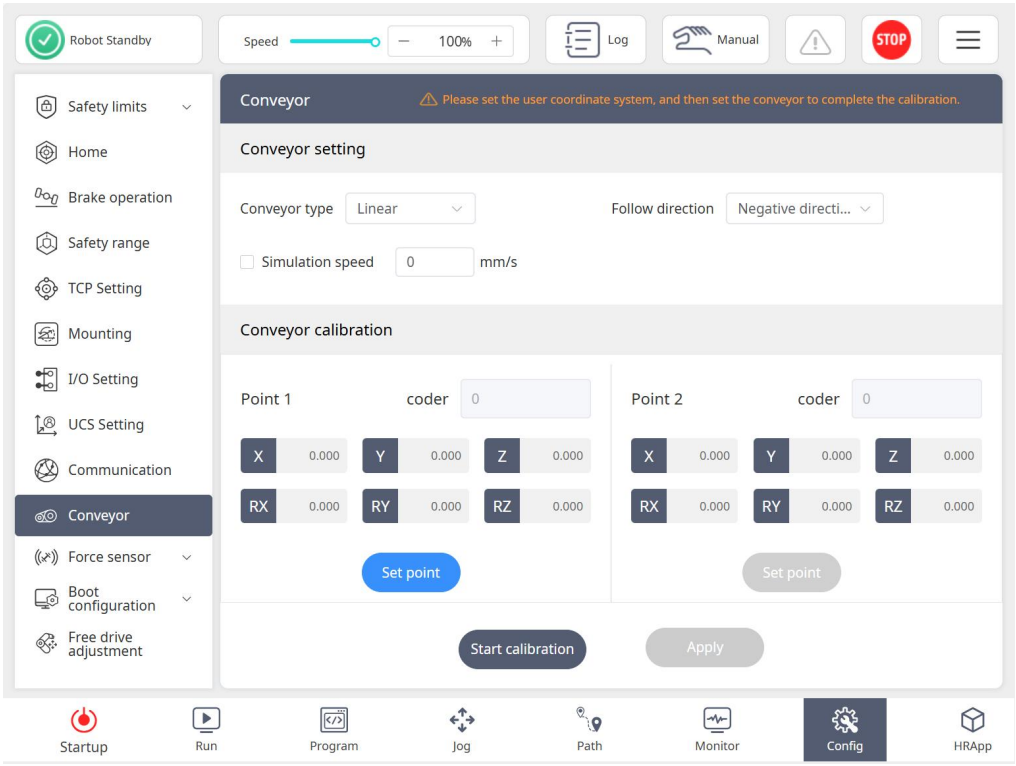
follow direction.

When the conveyor stops, make a sign P on the conveyor, click Set point for the calibration point 1, move the robot arm's end-effector to the point P of the conveyor, and complete the setting of calibration point 1;

Continue to run the conveyor, allow its point P move to point P'; then stop the conveyor, click Set point for the calibration point 2, move the robot arm's end-effector to the point P' of the conveyor, and complete the setting of calibration point 2;

Finally, click "Start calibration", and the conveyor calibration is completed. After the calibration is completed, a completion prompt will pop up. If the calibration error exceeds the allowed threshold, an error window will pop up.

Fig. 5.9.3.2 Conveyor Calibration Interface



— **Conveyor Type:** The conveyor type only supports linear models.

- **Follow Direction:** The directions are positive and negative. When set to positive, the Y coordinate of the calibration point is positive; when the set to negative, the Y coordinate of the calibration point is negative.

— **Simulation Speed:** This is the operational speed used for debugging conveyor belts and is applied when selected.

— **Coder:** Indicating the value read by the encoder.

Note: This feature is only supported on control cabinets of the 4th generation and above.

5.10 Force Sensor

Traditional industrial robots are only capable of position control, limiting them to simple contact tasks like welding and painting. However, many tasks require interaction with the environment and specific contact forces. This need gave rise to force control function, which adjusts robot trajectories using contact force feedback signals.

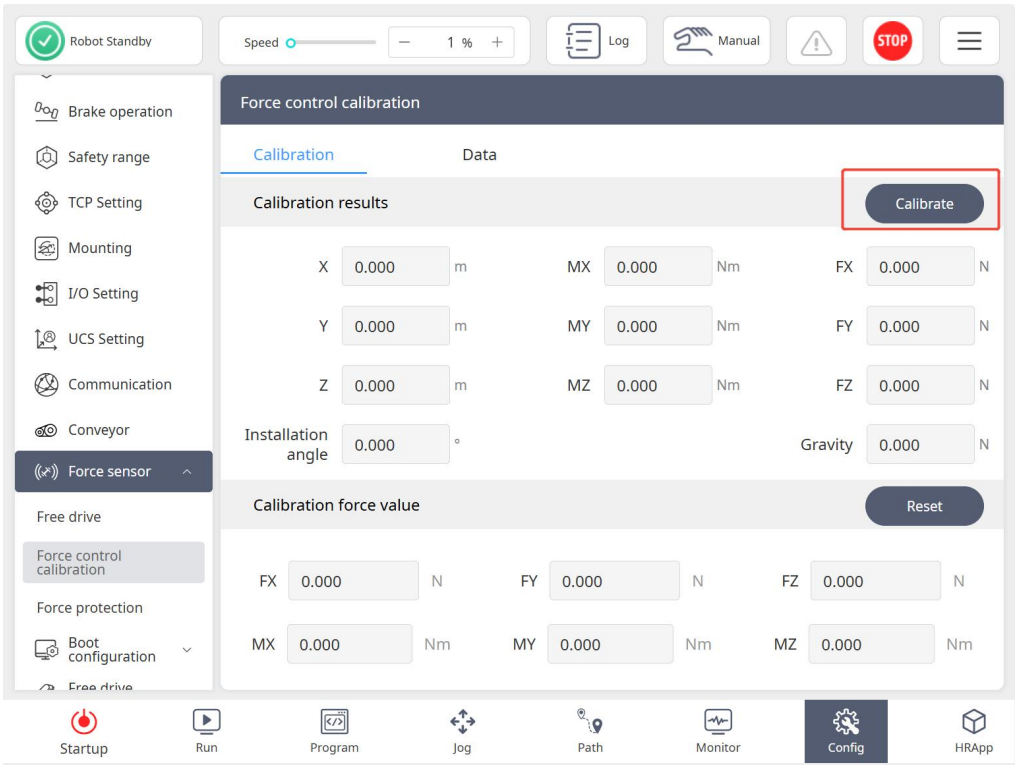
The Advanced Free Drive Mode is a control mode that allows an industrial six-axis robot arm to move freely under external forces. Typically used with torque sensors, it enables human operators or other external forces to directly push or guide the robot arm without traditional programming or controller input. This mode is mainly used in programming, teaching and debugging phases, making robot arm operation more intuitive and convenient.

In Advanced Free Drive Mode, each joint of the robot arm can move freely when external forces are applied. The control system does not actively apply torque. Instead, it uses force/torque sensors to detect external forces, calculates the needed drive current to counteract gravity and inherent resistance. This allows the arm to be easily moved by hand or other external forces when external forces are applied.

Note: This function requires external forces to be applied directly to the robot arm's end-effector. As the robot arm only has a force sensor at its end-effector and none on other parts, the robot arm will not respond and will remain stationary in Advanced Free Drive Mode if external forces are not applied there.

5.10.1 Calibration

Fig. 5.10.1.1 Calibration Result Interface



Click the "Calibrate" button on the page to enter the calibration result interface.

Select the calibration accuracy. The user accuracy can be selected from eight to sixteen bits. After selection, the corresponding calibration value will be generated in the calibration list.

Next, click "Calibration Point 1" to navigate to the "Jog" interface, move the robot arm to the target position, confirm and complete the calibration for point 1. Repeat this process to complete the calibration for the remaining points.

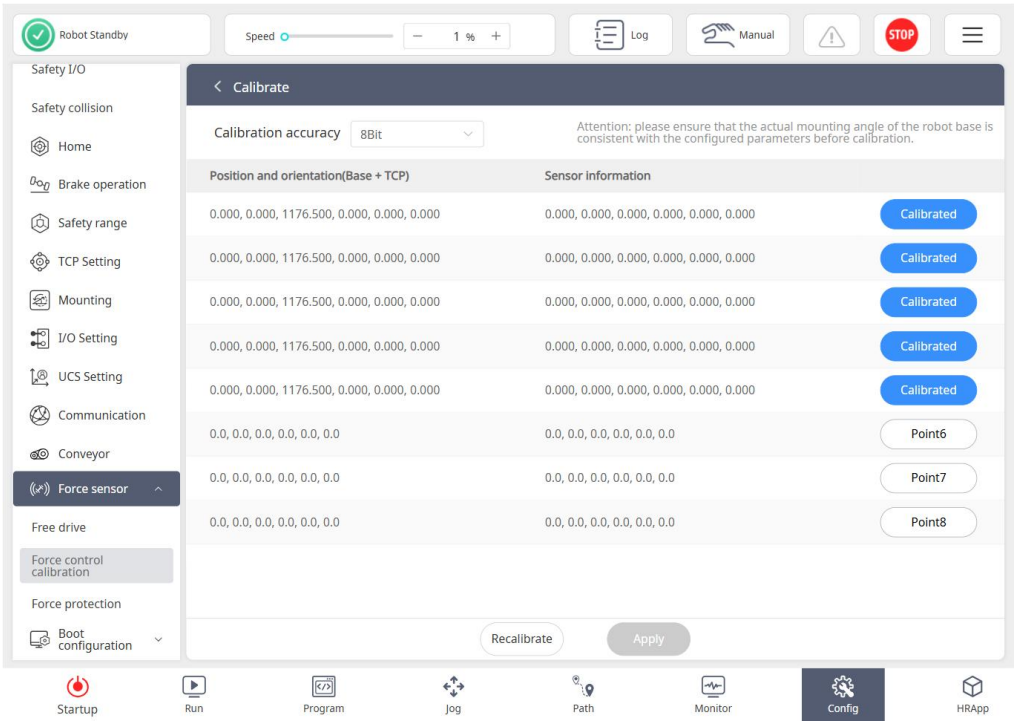
Finally, click "Apply" to complete the calibration for the force sensor.

Click the "Reset" button to reset all currently calibrated points and clear all calibration parameters, returning them to zero, so that calibration can be performed again.

- The calibration accuracy can be selected from eight bits to sixteen bits. The higher the number of bits, the better the calibration accuracy;
- Click the "Calibration point n" button to navigate to the "Jog" interface, where users can use the jog buttons to move each joint of the robot arm;
- In the "Jog" interface, calibration points must cover multiple directions of RX, RY, and RZ;

Note: When a payload is attached at the robot arm's end-effector during calibration process, self-collision must be avoided;

Fig. 5.10.1.2 Reset and Apply Results



Enter the calibration interface and select the calibration accuracy. The accuracy range is from 8 to 16 bits. A higher number of bits results in better calibration performance. After selection, the system will generate the corresponding number of calibration points in the calibration list (e.g., selecting 8-bit accuracy will generate 8 calibration points).

- **Begin Calibration**: Start with [Calibration Point 1]. After clicking, the system will

jump to the "Jog" page. On this page, move the robotic arm to the target position and confirm. This completes the calibration for this point. Then, proceed to complete the calibration for all remaining points in the list sequentially.

- **Calibration Requirement**: The postures of the robotic arm at all calibration points must cover multiple directions, specifically +/- RX, +/- RY, and +/- RZ. For specific point posture planning, please refer to Table 5.9.1 below (using 8-bit accuracy as an example).

After calibrating all points, click the [Apply] button. The system will save and use all the calibration data, finalizing the calibration process.

Clicking the [Reset] button will clear all currently calibrated point information and data, restoring the system to its initial state, allowing you to start the calibration over.

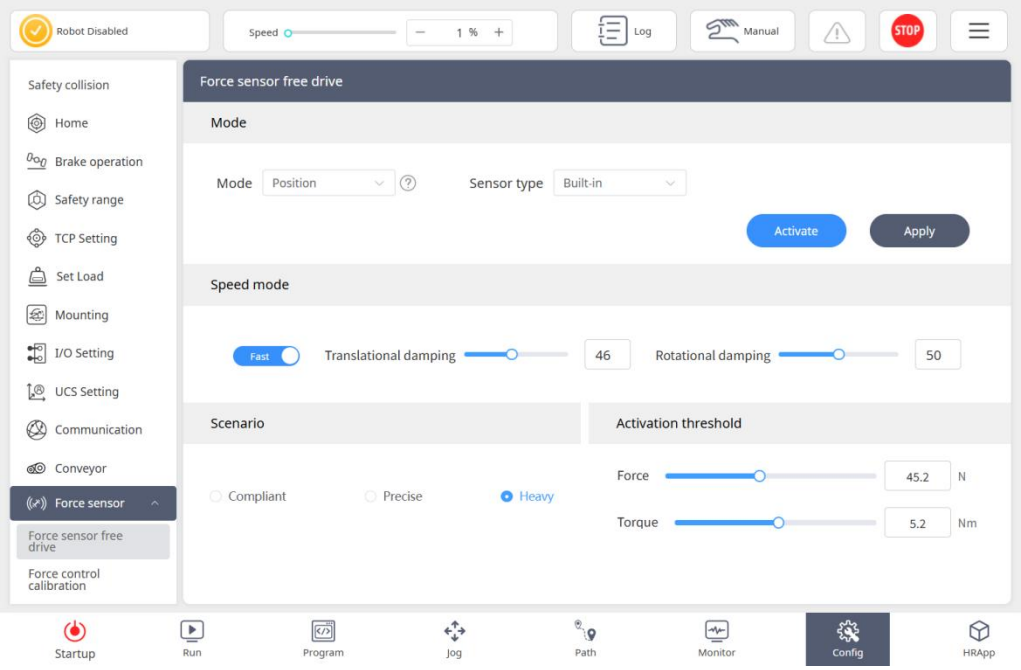
Example Using 8-bit Calibration

No	Joint Coordinates (J1, J2, J3, J4, J5, J6)
1	(0,0,90,-30,90,0)
2	(0,0,90,0,90,0)
3	(0,0,90,30,90,0)
4	(0,0,90,-30,60,0)
5	(0,0,90,0,60,0)
6	(0,0,90,30,60,0)
7	(0,0,90,-30,120,0)
8	(0,0,90,0,120,0)

5.10.2 Advanced Free Drive

The Advanced Free Drive Mode includes position mode, attitude angle mode, free mode and custom mode, and users can choose according to their needs. The default displays are location mode and last set mode.

Fig. 5.10.2.1 Free Drive and Compliance



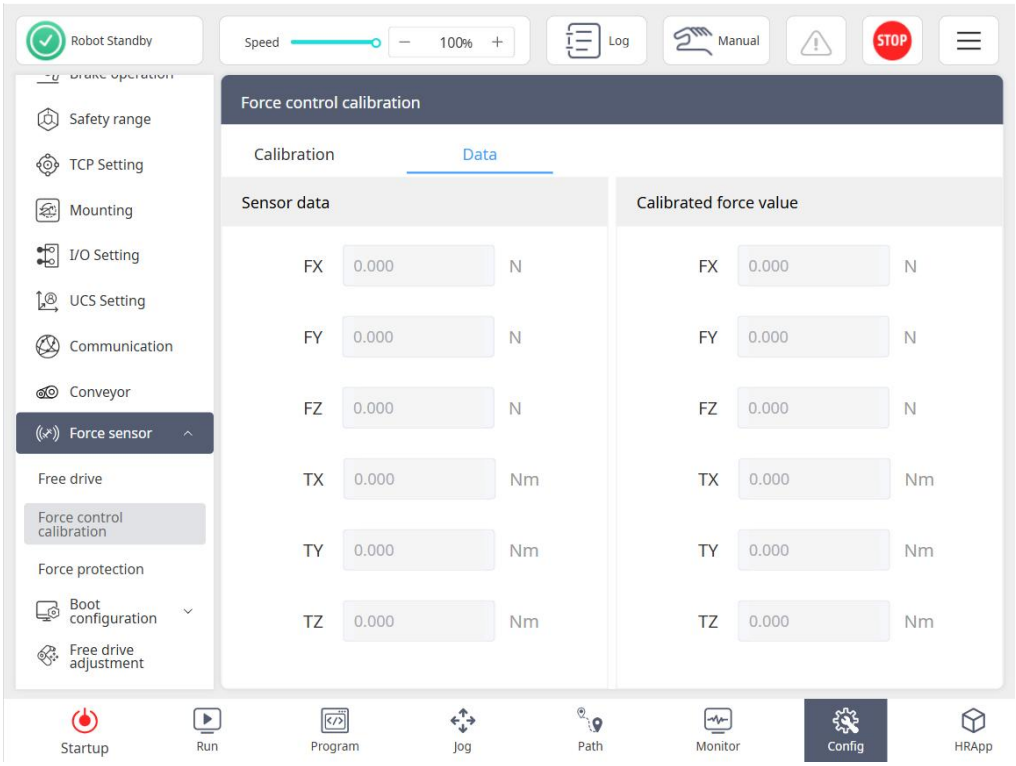
- **Position Mode**: Only movable along the X, Y, Z axes;
- **Attitude Angle Mode**: Only movable in RX, RY, RZ directions;
- **Free Mode:** Movable in X, Y, Z, RX, RY, RZ directions;
- **Custom**: The user can select one or more positions or rotation angles to move the robot arm;
- **Compliance**: By adjusting the compliance parameters, users can customize the damping of force-controlled dragging when enabling the Advanced Free Drive mode, thereby optimizing the feel of the operation hand;
- **Speed Mode**: Includes Fast Mode and Slow Mode.
- **Scene**:
  - **Compliance Scene**: Suitable for medical, rehabilitation environments.
  - **Precision Scene**: Suitable for welding, high-interference environments.
  - **Heavy-Duty Scene**: Suitable for screw driving environments.
  - **Activation Threshold**: The minimum external force/torque required to initiate the robot arm's movement.
- **Fast Mode**: In this mode, the damping adjustment function is available. Users can customize the haptic feel of force-controlled dragging by adjusting the parameters for translational damping and rotational damping.
- **Slow Mode**: In this mode, the damping parameters are locked and cannot be adjusted.
- **Damping**: Damping is categorized into Translational Damping (affecting movement along the X, Y, Z axes) and Rotational Damping (affecting rotation about the RX, RY, RZ axes). Users can adjust these two types of damping independently to meet different control requirements.

Click the "Apply" button to save the selected mode, implement it and obtain feedback prompts. After clicking the "Activate" button, the user can freely move the robot arm under the applied mode.

5.10.3 Data

On the left is the data feedback from the force sensor, and on the right is the data calculated based on the calibrated accuracy.

Fig. 5.10.3.1 Data Interface



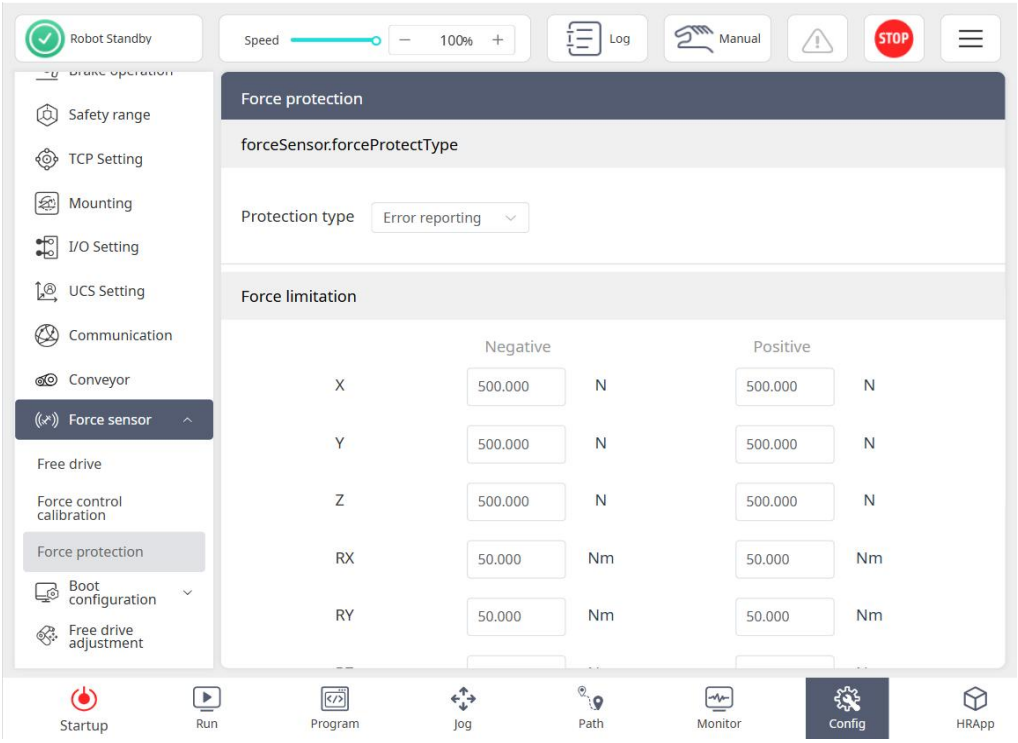
5.10.4 Force protection

When the force sensor is subjected to an excessive impact, the force protection function will be activated.

Force protection types include two modes: power off and error reporting.

Users can adjust the force limit parameters in all directions, including the force magnitude, positive/negative values in the X, Y, and Z directions, and torque magnitude in the RX, RY, and RZ directions. After confirming the adjustments, the users must click "Apply" to implement them.

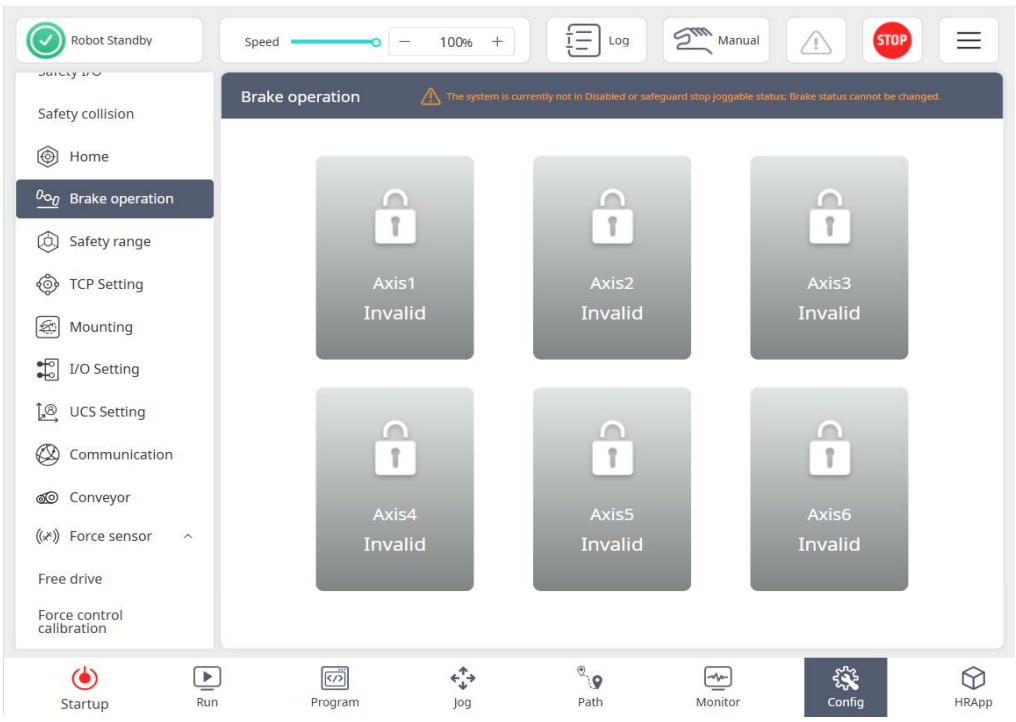
Fig. 5.10.4.1 Force Protection



5.11 Brake Operation

When the robot malfunctions or the joints of the robot arm need to be moved in a zero-gravity manner (for example, the robot arm is subjected to external forces, resulting in an error report and automatic disabling) it is not possible to activate the basic free drive function to change the robot arm's pose. Therefore, brake releasing operation of the joints of the robot arm is required. This function must be operated in the disabled state and is not available when the robot is enabled.

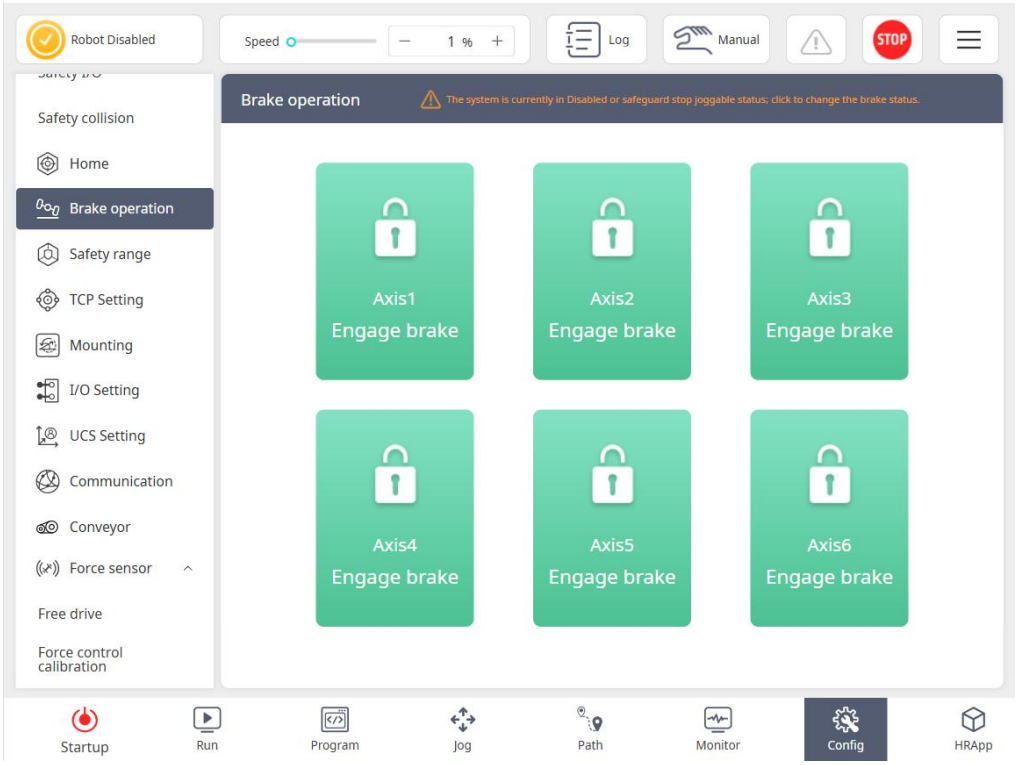
Fig. 5.11.1 Brake Release Prohibited When Enabled



Before using this function, the robot needs to be disabled. Once disabled, this function is available. Clicking on the corresponding joint will release the brake and unlock the relevant joint. The page displays the brake engagement or release status of the current individual joint of the robot arm.

Axis1 ~ Axis6 correspond to joints J1-J6 of the robot arm.

Fig. 5.11.2 Braking/Releasing Available in Disabled State



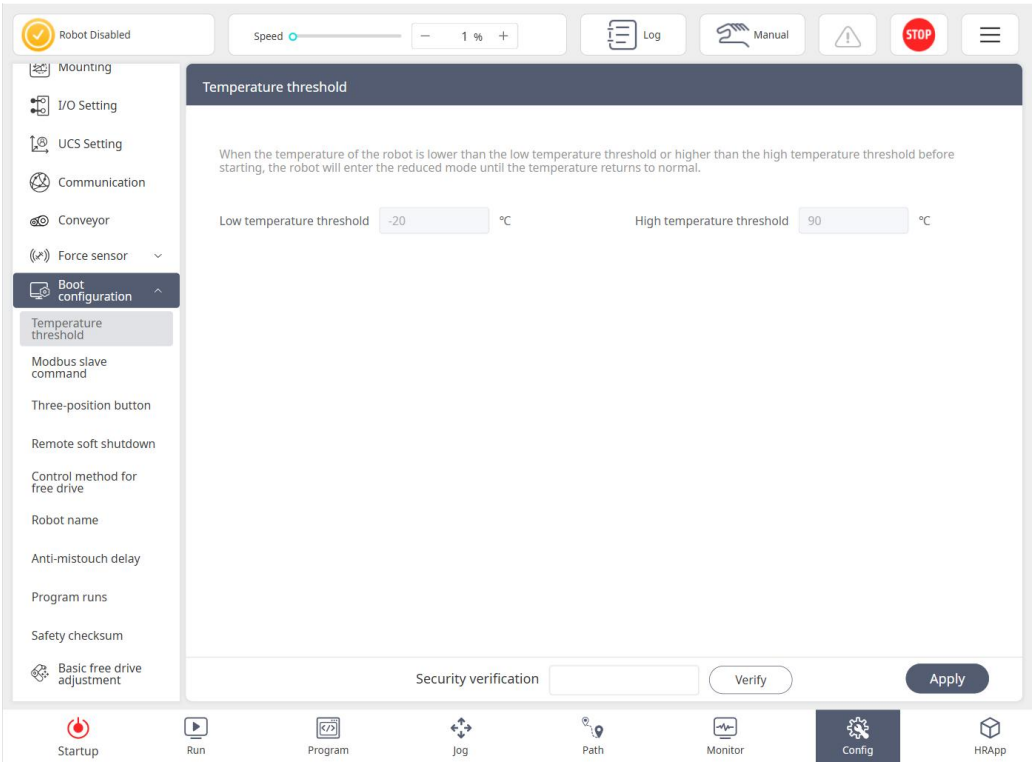
5.12 Boot Configuration

To configure the robot boot configuration, including temperature threshold, 3PE (3-Position Enabling) button, Modbus slave station command, and remote shutdown. The user needs to verify the security password (consistent with the administrator password) before operating.

5.12.1 Temperature Threshold

This page is designed to set the low-temperature and high-temperature thresholds. If the joint temperature of the robot arm exceeds the low-temperature threshold or surpasses the high-temperature threshold, the robot will enter reduced mode.

Fig. 5.12.1.1 Temperature Threshold



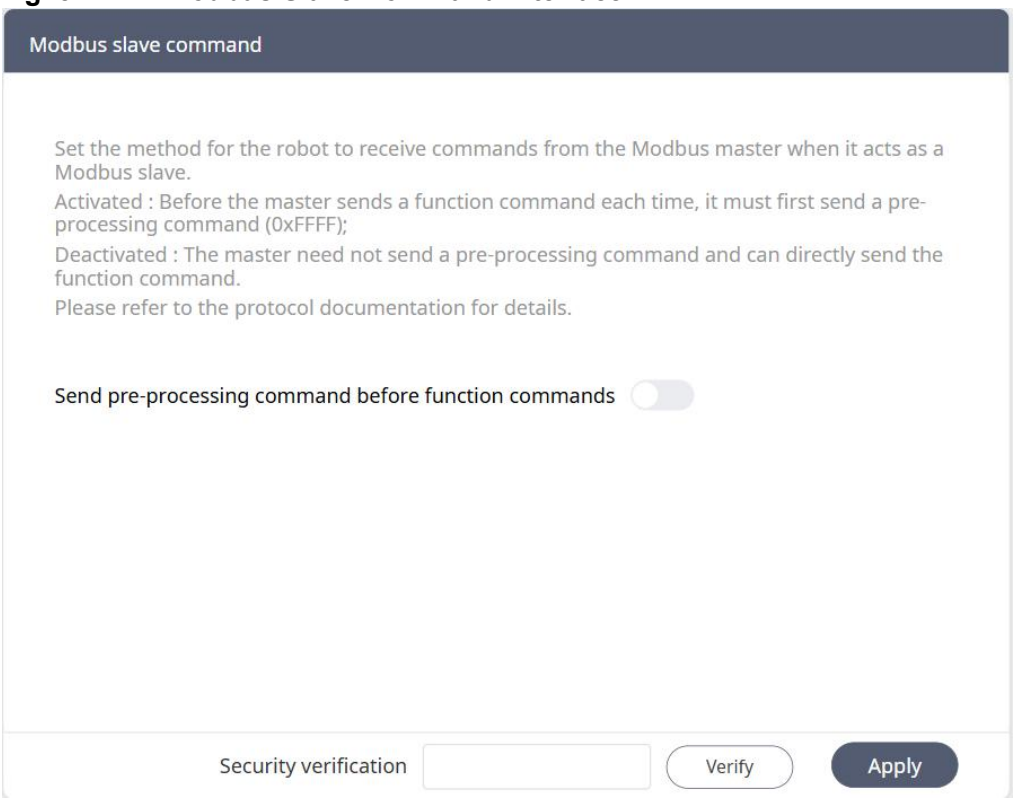
Note:

- The low-temperature threshold is set within the range of -20 to 80 and must be an integer.
- The high-temperature threshold is set within the range of -10 to 90 and must be an integer.
- The high-temperature threshold must be higher than the low-temperature threshold, with a temperature difference of more than 10℃.

5.12.2 Modbus Slave Command

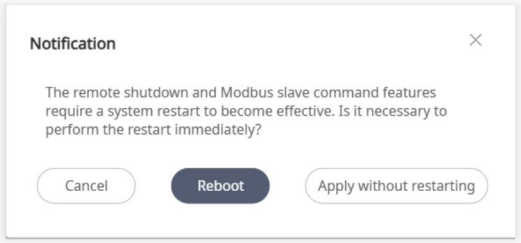
For detailed information, refer to section 5.7 Communication (Modbus). To set the commands sent by the Modbus slave station. This function is disabled by default.

Fig. 5.12.2.1 Modbus Slave Command Interface



After applying the configuration, a system restart is required for the changes to take effect.

Fig. 5.12.2.2 Configure and Apply



5.12.3 Three-position button

The three-position button is located on the back of the teach pendant box and is a button with three positions. When the user releases the button, it is in the first position, and the button function is disabled; when the user lightly presses it to the middle (second position), the button function is activated; when the user presses the button firmly to the bottom (third position), the button function is disabled. During equipment debugging, the three-position button can effectively reduce personal injury caused by the equipment.

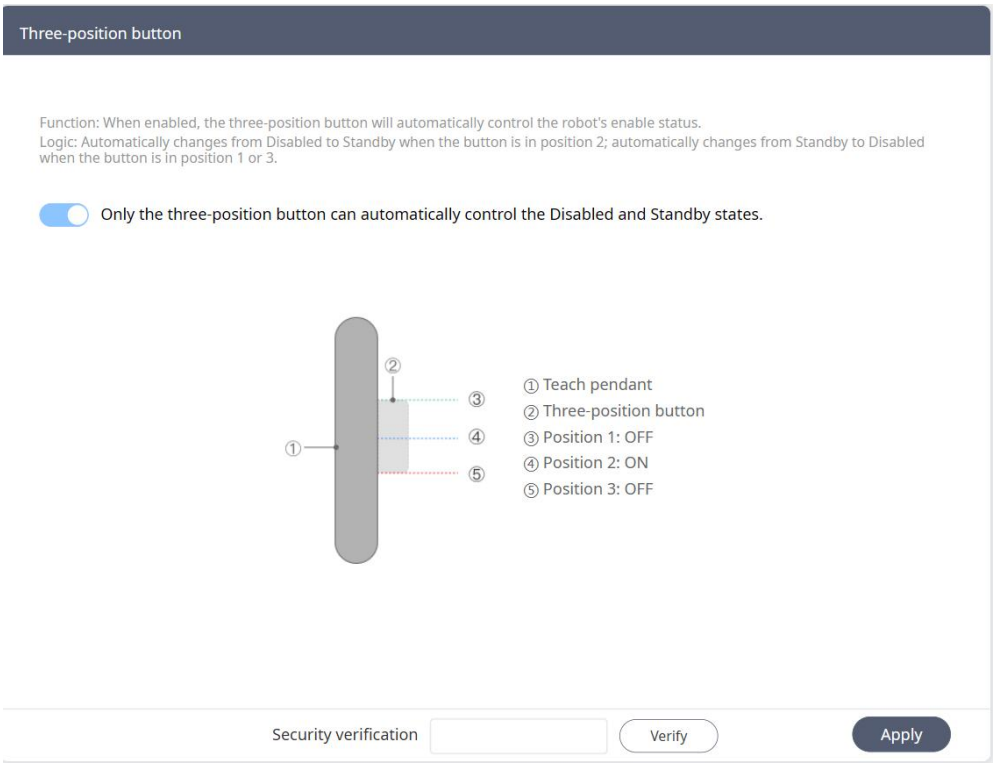
Whether the three-position button is enabled can be controlled by operating the corresponding switch on the "Boot configuration" page;

After the three-position button is enabled, some associated restricted functions can only be used when the three-position button is in the second position. If the three-position button is no longer in the second position during the normal use of these functions, the system will immediately stop the relevant movements or

operations and return to the disabled state.

The three-position button only functions in manual mode and does not work in auto mode.

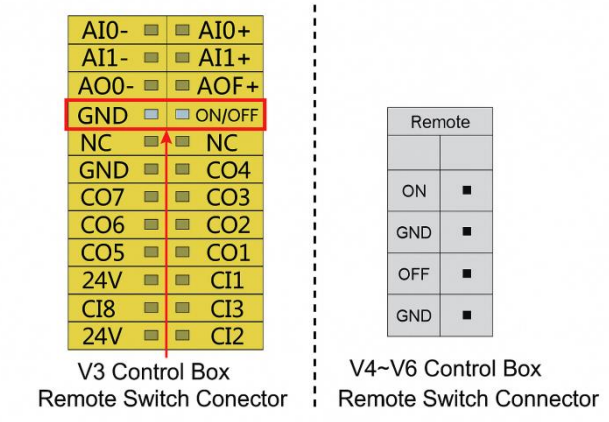
Fig. 5.12.3.1 Three-position button Interface



Note: This function is a standard safety function, designed to ensure safety during robot operation and prevent collision hazards caused by incorrect operations.

5.12.4 Remote Soft Shutdown

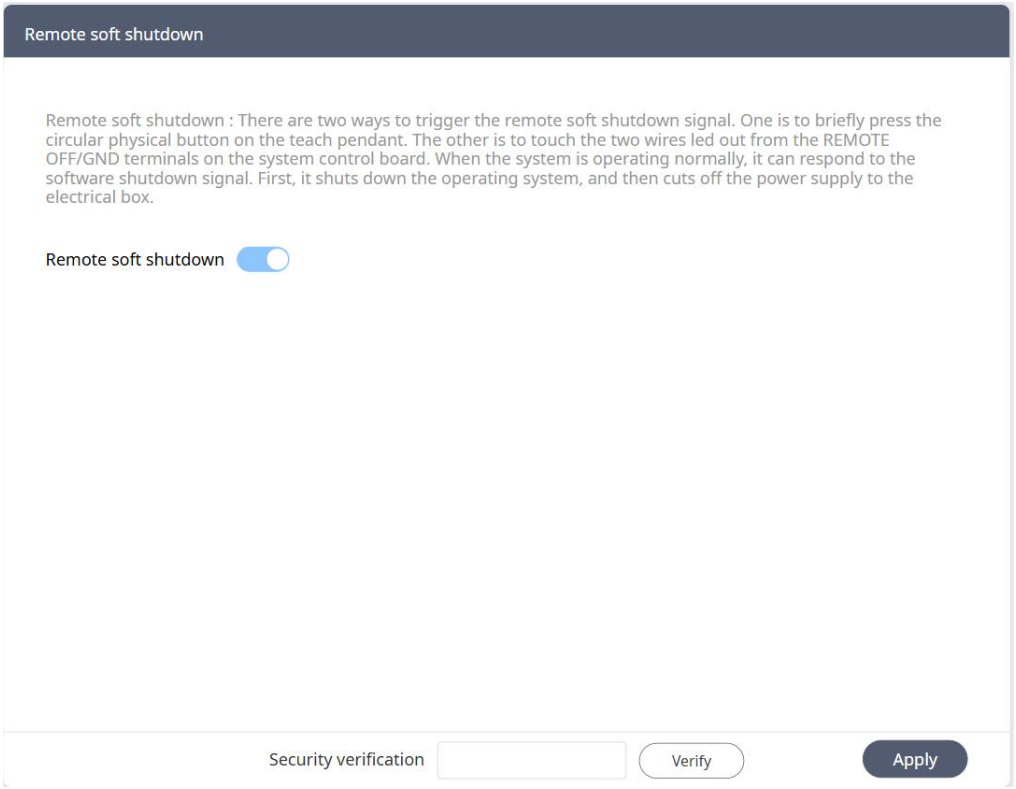
The remote soft shutdown signal can be triggered in two ways: one is by briefly pressing the circular physical button on the teach pendant, and the other is by shorting the two wires extending from the REMOTE OFF/GND ports on the system control board.



When the system is running normally, it can respond to the software shutdown signal by first shutting down the operating system and then cutting off the power

supply to the electrical control box. Activating this function requires a reboot to take effect.

Fig.5.12.4.1 Remote Soft Shutdown



5.12.5 Control method for free drive

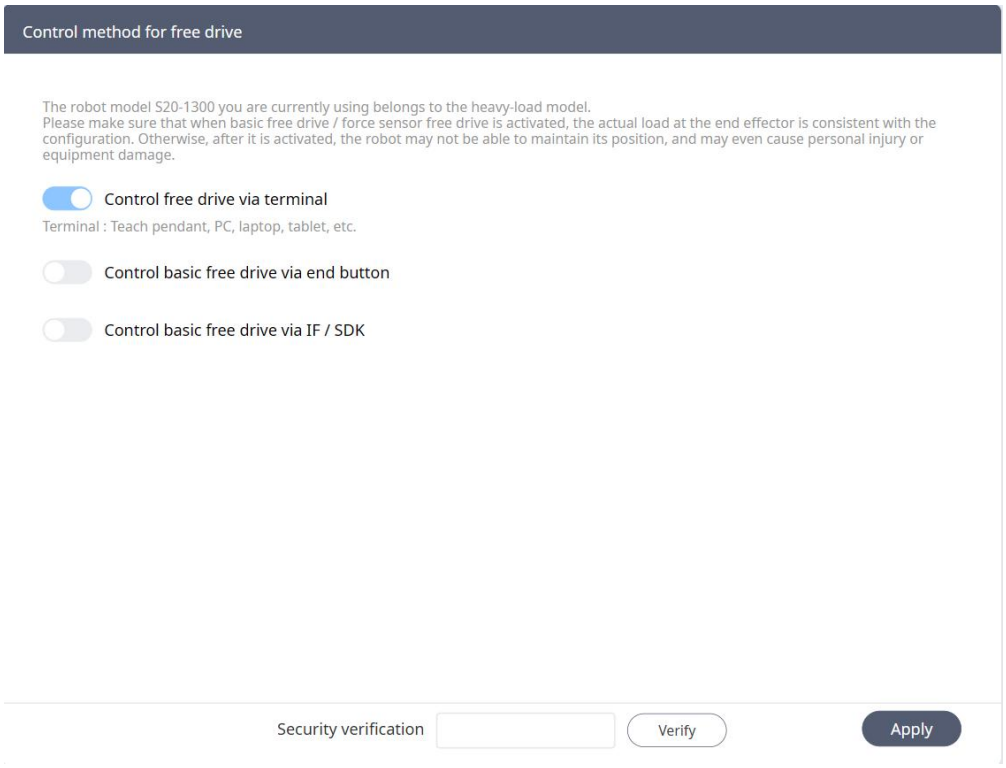
There are several ways to activate the basic free drive mode:

Terminal Configuration: Users can access the "jog" interface on the terminal (teach pendant, PC, laptop, pad) and activate the basic free drive.

End-effector Button: After enabling the end-effector I/O function, press the end-effector Free button to activate the basic free drive.

IF/SDK: Using IF/SDK to activate the basic free drive. IF/SDK are software interfaces for the 3rd party application to connect to and control the robot.

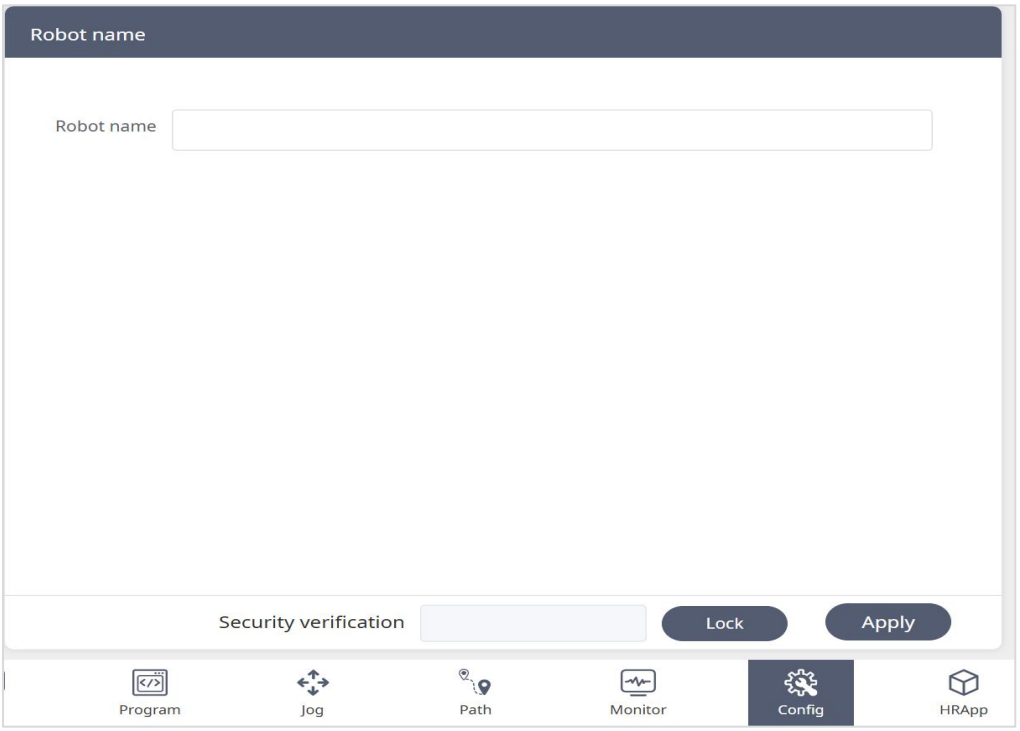
Fig.5.12.5.1 Advanced Free Drive and Basic Free Drive Activation Methods



5.12.6 Robot Name

The robot name can be customized to distinguish different robots.

Fig.5.12.6.1 Robot Name Page

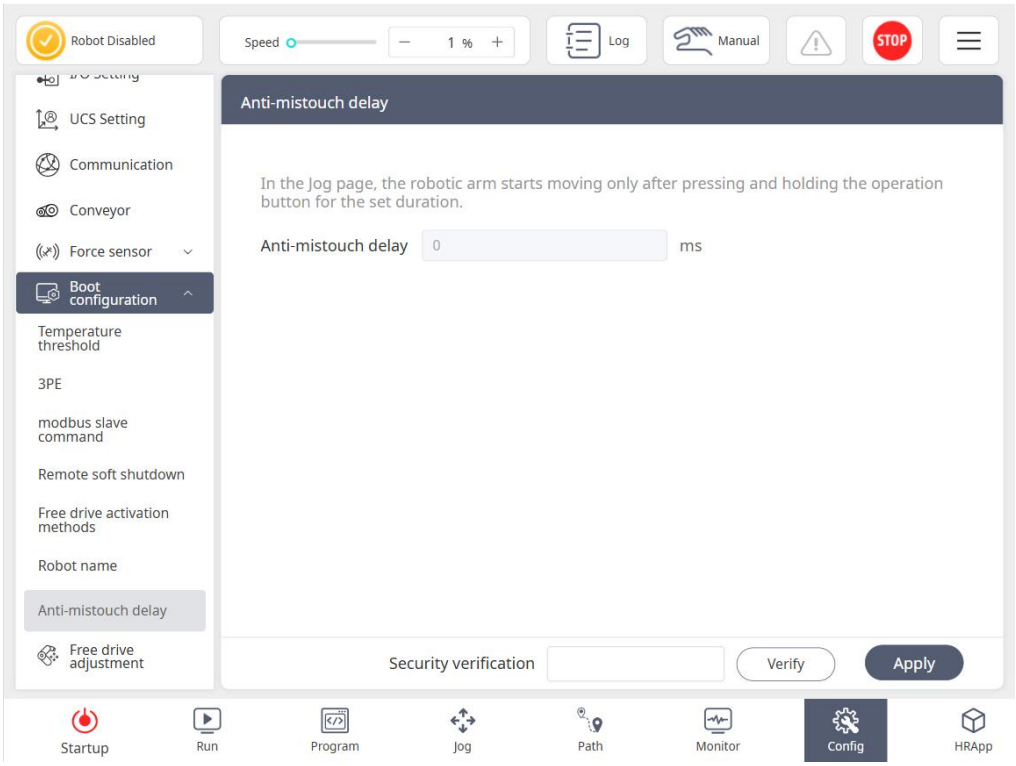


5.12.7 Anti-false Touch Delay

On the jog interface, the robot arm starts moving only after pressing and holding the button for the set time.

Security verification is required for setup.

Fig.5.12.7.1 Debounce Time



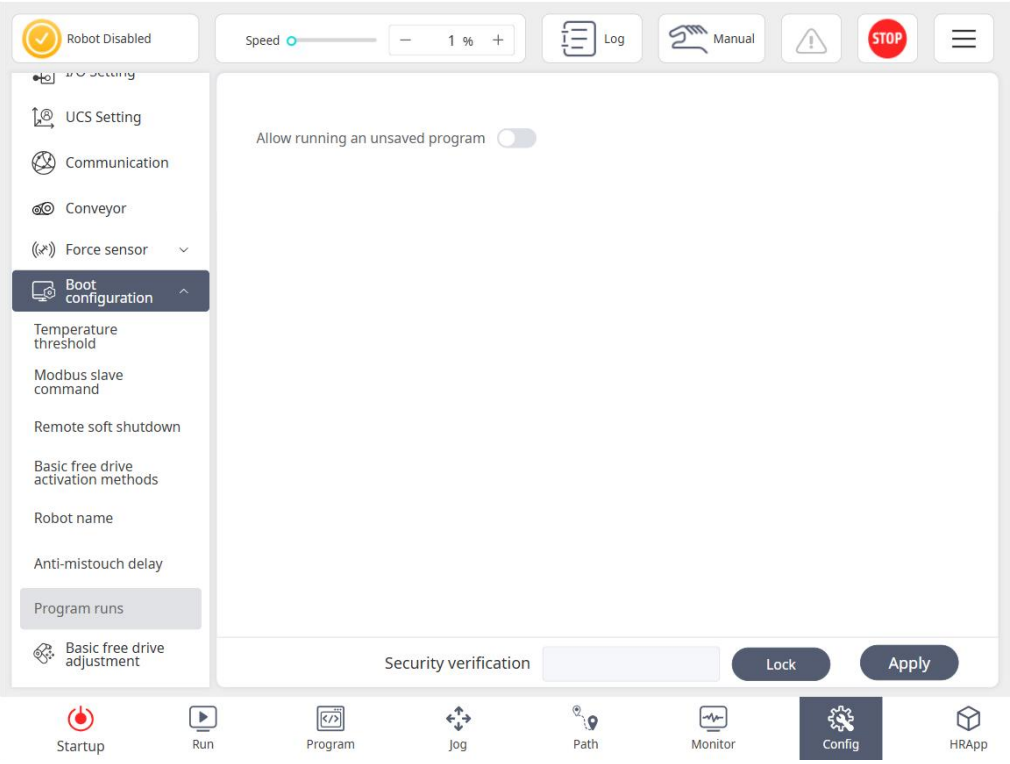
5.12.8 Program runs

"Allow running an unsaved program" is an important setting that balances safety and convenience. It allows you to choose whether the system requires a secondary confirmation when running a program that has been modified but not saved. By default, the system will intercept and prompt to ensure operational safety. Enabling this function allows the system to skip all prompts and directly execute the current version of the program in memory, which is designed for scenarios involving automation integration via external interfaces.

Off : When running an unsaved program, the system will intercept the operation and display prompts, waiting for user confirmation.

On: When running an unsaved program, the system will execute it directly without any prompts.

Fig.5.12.8.1 Debounce Time



5.12.9 Safety Checksum

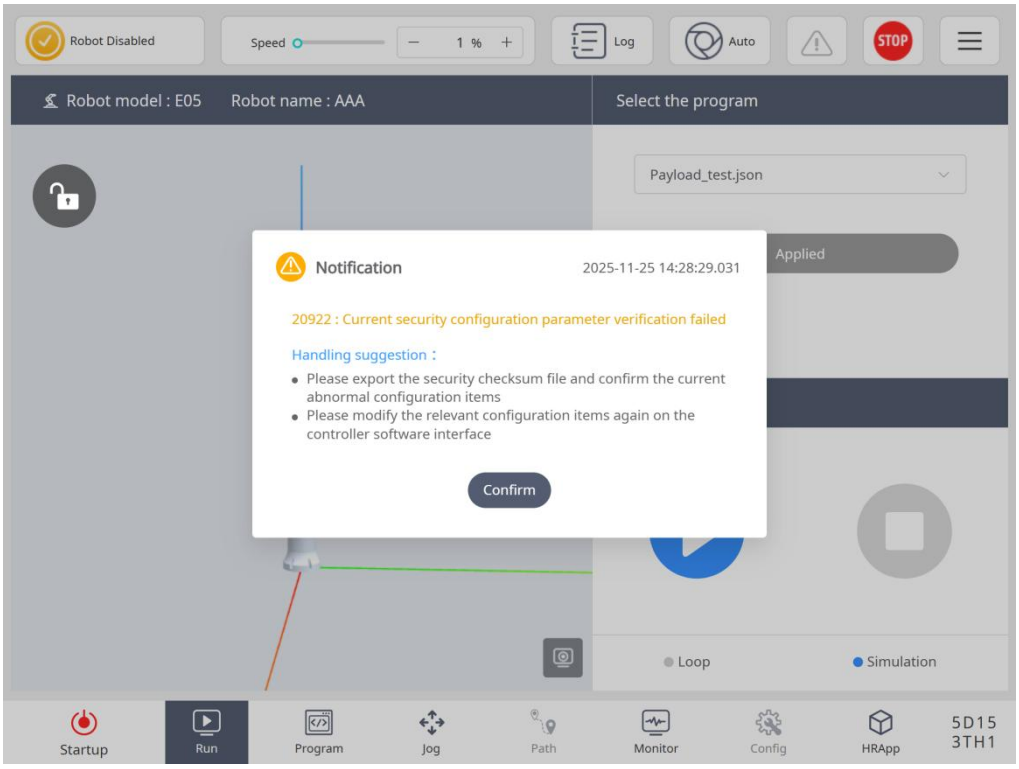
The safety checksum function is used to verify critical security parameters when switching to auto mode, preventing unverified modifications bypassing the front-end interface and ensuring the robot's safe operation.

Automatic Verification: When switching to automatic mode, the system automatically verifies the consistency of security parameters between the current configuration and the validated saved configuration.

Exception Handling: If the verification fails , the robot system will prohibit enabling.

Configurable: This function can be enabled or disabled, with the default setting being disabled.

Fig. 5.12.9.1 Safety Checksum Prompt When Switching to Auto Mode



The safety checksum result is presented as an 8-bit hexadecimal check code.

Fig. 5.12.9.2 Safety Checksum Code

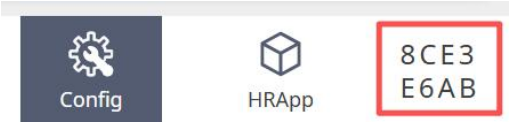
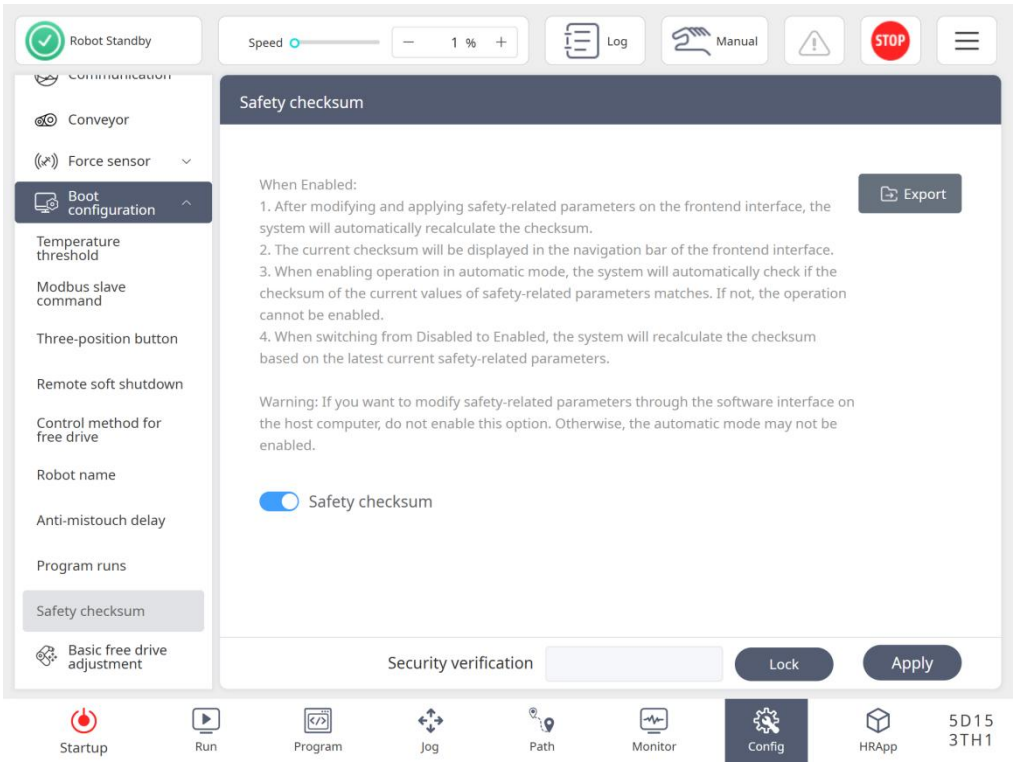


Fig. 5.12.9.3 Safety Checksum Configuration Page



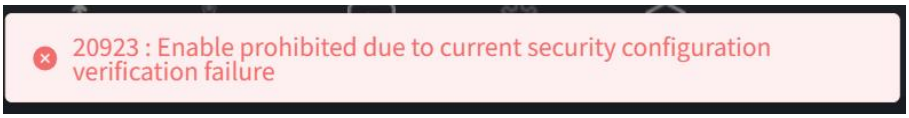
Parameters involved in verification include: Safety collision, Safety home, Safety

plane, Safety space, Temperature threshold, Remote soft shutdown, Basic free drive activation methods and Basic free drive adjustment.

All parameters involved in verification can be exported as a file via the front-end interface.

If verification fails, the message "Current safety configuration parameter verification failed" is displayed.Clicking [Enable] under verification alarm will trigger the error message "Enable prohibited due to current security configuration verification failure".If the robot is already enabled when verification fails, it will automatically switch to the Disabled state.

Fig. 5.12.9.4 Enable Prompt in Auto Mode When Verification Fails



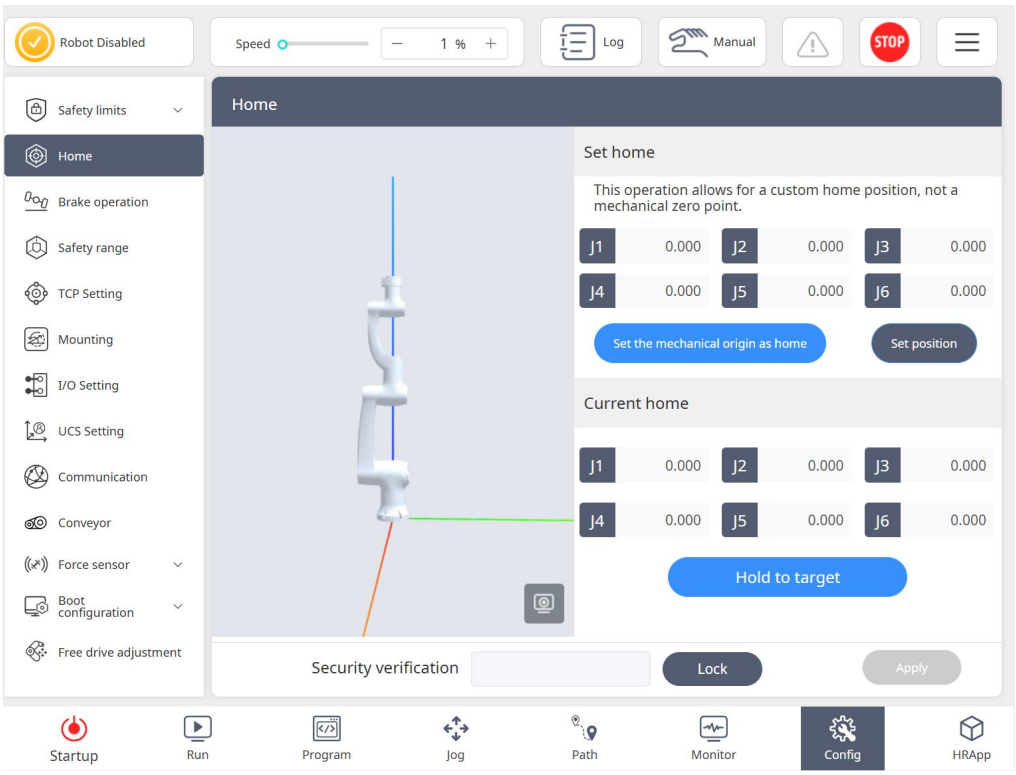
Note:

- Security parameters can only be modified in manual mode.
- The checksum will be recalculated automatically when the relevant parameters involved in the verification are modified.
- No parameter consistency verification will be performed when the function is disabled..

5.13 Home

A customizable origin position can be set. Press and hold the "Move to Home" button in the "Jog" interface to reach it. In programming, a Home command can be added to move the robot arm to reach this safe position during back-and-forth motion.

Fig. 5.13.1 Home Function Diagram



As shown in the figure above, after passing the security verification, click "Set Position"to set home position. After the setting is successful:

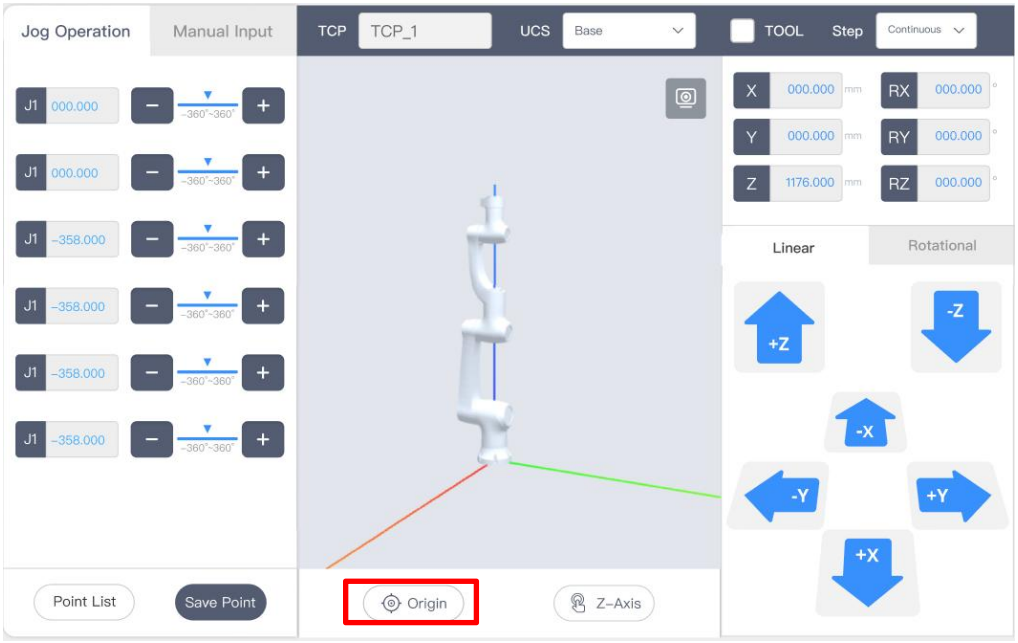
- In the "Jog" interface, click "Move to home" to set home position;
- In the programming interface, executing the Home command will also return to home position;

In addition, click "Set the mechanical origin as home" to set the mechanical origin (factory-set origin) as the current home. For detailed information on the default mechanical origin, please refer to 4.10 Move to Home.

In the "Current home" module, the joint angles of the current origin will be displayed. Press and hold the "Hold to target" button, and the robot arm will move to the currently set home position.

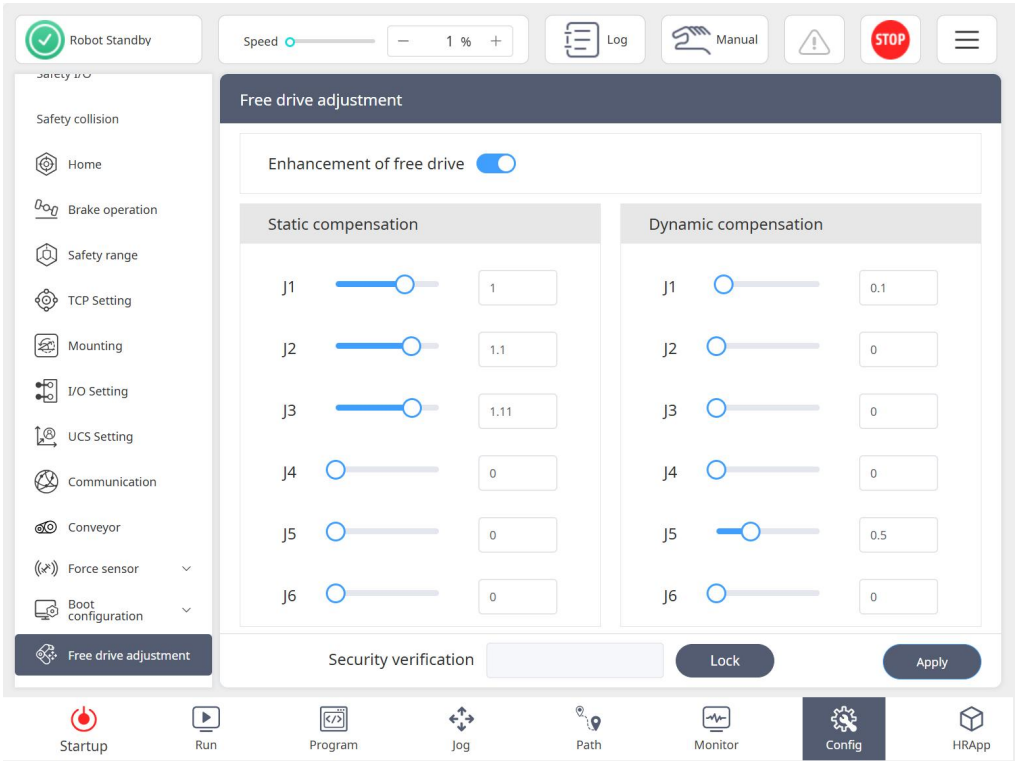
Press and hold "Move to home" on the "Jog" interface, and the robot arm will also return to the currently set home position.

Fig. 5.13.2 Move to Home on the Jog Interface



5.14 Basic Free Drive Adjustment

Fig.5.14.1 Basic Free Drive Adjustment



Adjust the compensation parameters so that the robot arm can be guided easily and smoothly by the operator in zero-force teaching mode while maintaining stability and accuracy.

Static Compensation: used to compensate static friction when the robot arm is stationary, improving positioning accuracy and stability in static or low-speed conditions. This parameter directly affects how easily the robot arm can be guided

when stationary or moving slowly.

Dynamic Compensation: used to compensate dynamic friction when the robot arm is moving, improving trajectory tracking accuracy and stability during motion. This parameter directly affects motion smoothness at higher speeds.

Program

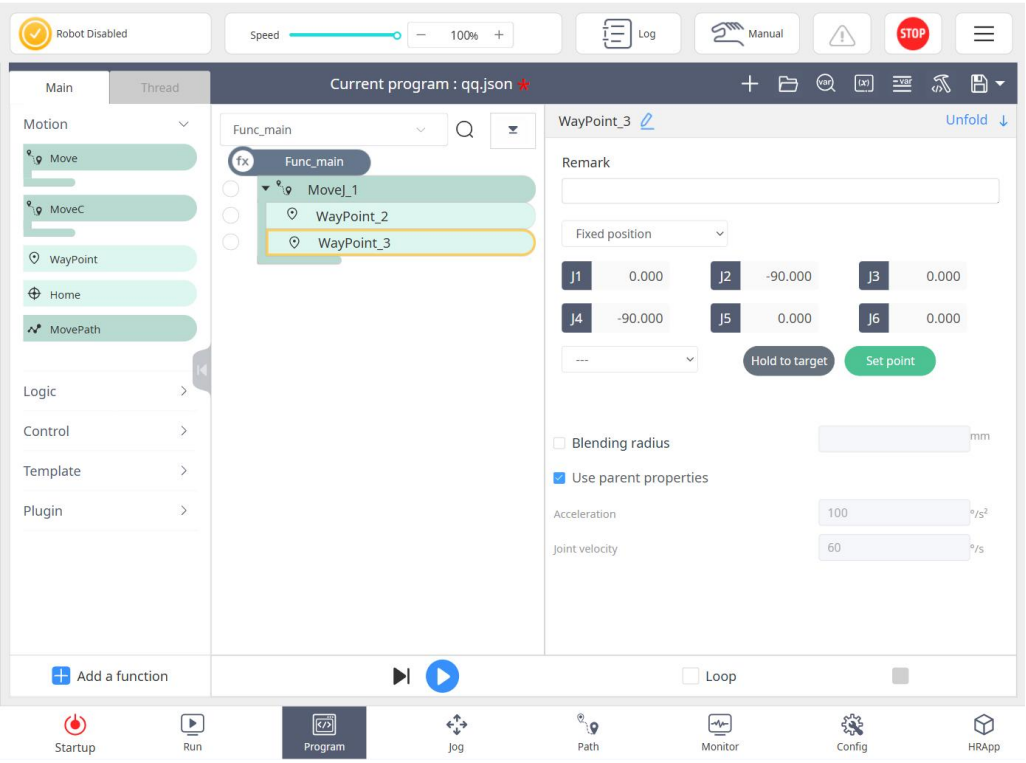
This chapter contains the following topics:

– Introductory Program
– Understanding the Interface
– Motion Command
– Logic Command
– Control Command
– Template
– Tracking
– Thread
– Variable
– Debugging and Running
– Program Example

6.1 Introductory Program

Motion commands are the most basic and frequently used commands in robot programming. The following example shows how to create a simple introductory program.

Fig. 6.1.1 Programming Interface



1. Under Func_main, drag a Move command from the command menu to the function node. A waypoint is automatically created under the Move node and named WayPoint_2. The default motion type of Move is MoveJ. Then drag another waypoint under WayPoint_2 to create WayPoint_3.

2. In WayPoint_2, click Set Point to switch to the Jog interface and teach the point position. Then click Confirm to return. Switch to WayPoint_3 and repeat the same steps to teach the position of WayPoint_3. WayPoint_2 serves as the starting point of robot motion, and WayPoint_3 serves as the end point.

3. After the point positions are set, the waypoint information is displayed on the right side of the interface. Users can then adjust parameters such as speed and acceleration as required.

4. Click “Run” to execute the motion.

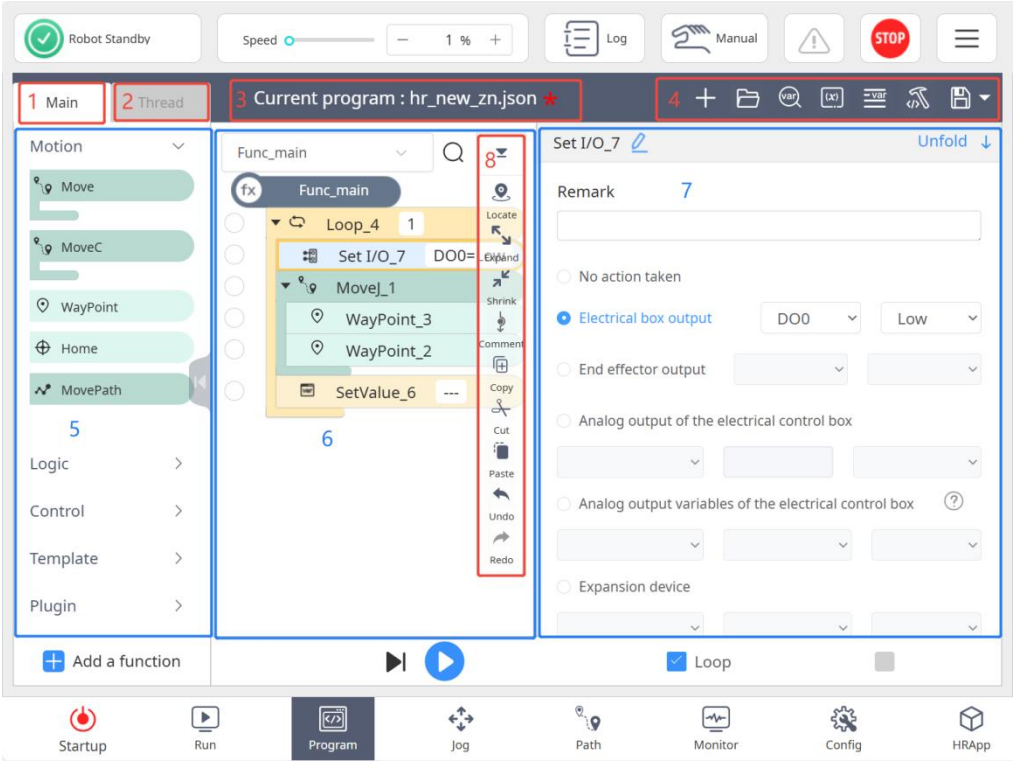
6.2 Understanding the Interface

The Programming interface is used for creating, editing, debugging, and managing robot programs. Users can drag commands from the command menu into functions, configure command parameters, and run or debug the program

after editing is completed.

6.2.1 Definition of Components

Fig. 6.2.1.1 Definition of Components



- **Main Program:** The main thread running by the robot, including functions such as motion control, logic processing, template calling and third-party plugin reference.
- **Thread:** Run in parallel with the main program, usually used to control external devices (such as signal status monitoring) and communicate with the main program through variables and output signals.
- **File Name:** Display the name of the current program file.
- **Functional Operations:** To look up, debug variables, create, load, variables, compile and save
- **Command Menu:** Display available commands in categories, and support drag and drop to the program list for use; the menu is collapsible to hide.
- **Command List:** The core command sequence of the program. Users can drag and drop commands from the command menu to the list and freely adjust the order of commands; the list must contain a command to run the program.
- **Parameter Setting:** Display and set the attributes related to the selected command, and support adding remark information.
- **Operation Area:** Support copying, pasting, skipping, etc. of commands in the command list.
- **Bottom Running Area:** Provide debugging functions (such as single-step

debugging and loop debugging).

6.2.2 Interface Introduction

Fig. 6.2.2.1 Programming Interface Operation Buttons

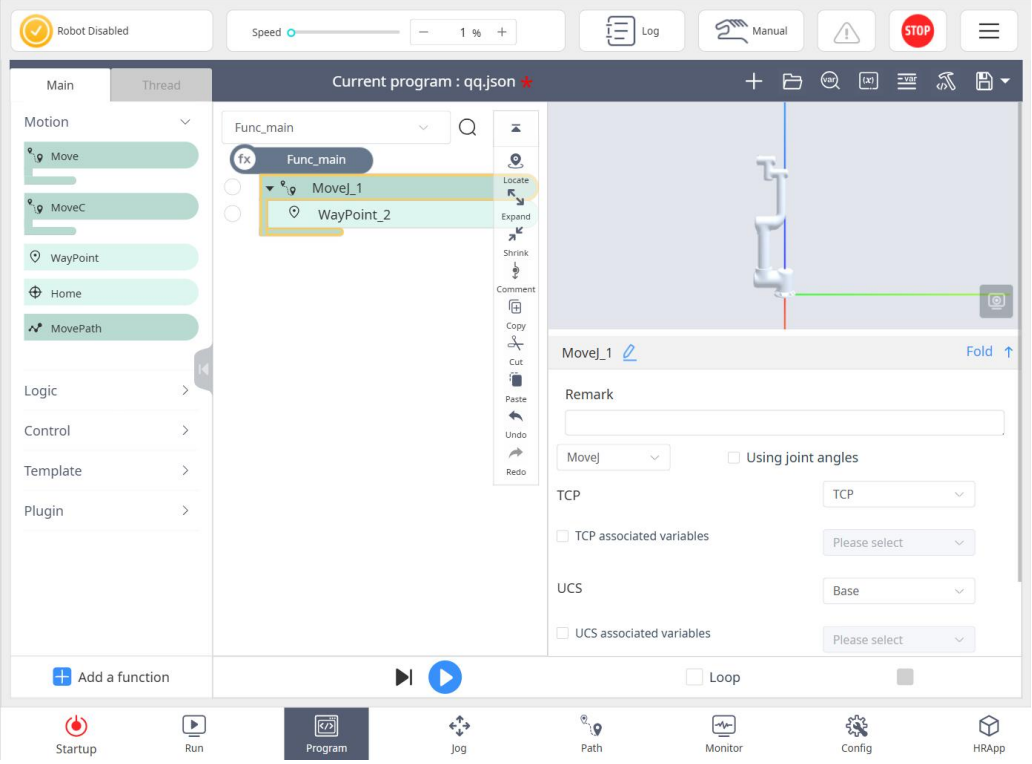


Table 6.2.1 Overview of Button Functions in Programming Interface

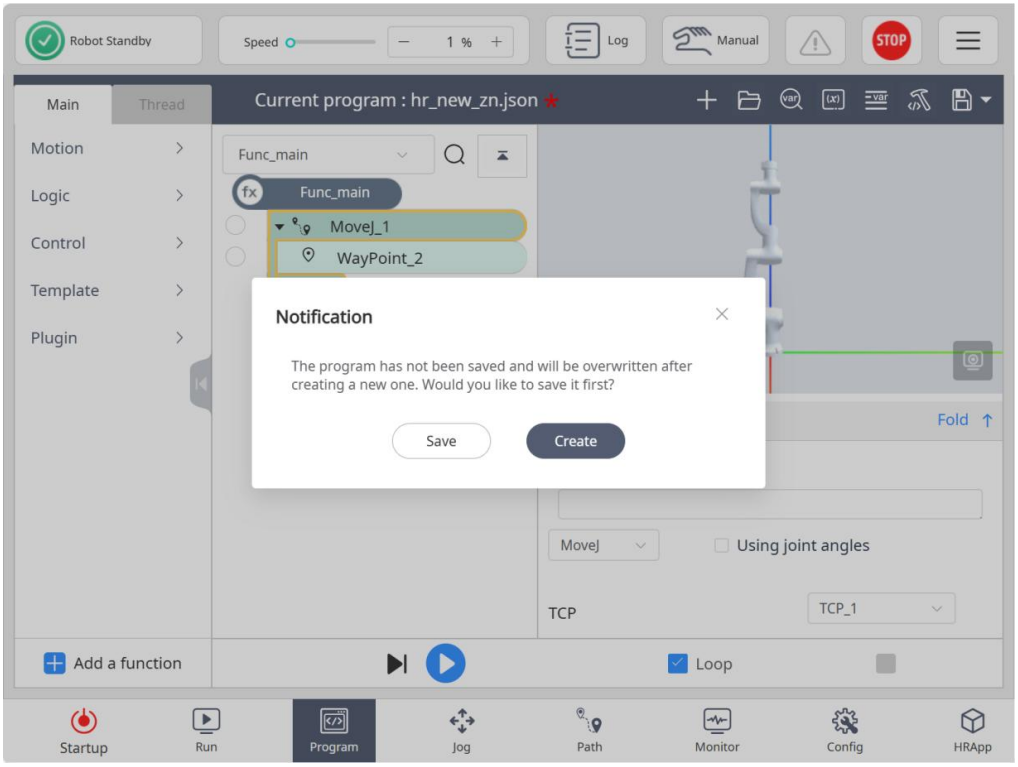
	Lookup	Command Lookup;.
	Create	Create a new program;
	Debug Variables	Monitor variables in scripts, changes in debugging state is only displayed when the program is running;
	Open	Load programs, import and export files, delete programs;
	Variable	Manage variables, view variable values;
	Compile	Compile the current program;
	Save	After editing the program, click the "Save" button and select Save/Save As;
	Unfold	When editing command parameters, more areas can be expanded upward while the 3D model is hidden; click again to fold;

	Remark	Remarks for functions or commands to facilitate the interpretation of functions;
Locate	Locate	Locate the currently executing command in the program;
Expand	Expand	Expand all statements of the currently selected function;
Shrink	Shrink	Shrink all statements of the currently selected function;
Comment	Comment	Select a command and click "Comment" to gray it out. The program will then skip it during execution.
Copy	Copy	Click "Copy", select one or more commands to be copied from the program list, and click "OK"; used in conjunction with pasting;
Cut	Cut	Click "Cut", select one or more commands to be cut from the program list, and click "OK"; used in conjunction with pasting;
Paste	Paste	Select a command and click "Paste", then the copied command will be copied below the selected command;
Undo	Undo	Click "Undo" to withdraw the last modification;
Redo	Redo	Click "Redo" to restore the previous modification;
	Delete	Drag and drop the command back to the menu area, and the red trash can will appear. Release your finger to delete the command;
	Search	Click the "Search" button to turn the function drop-down menu into a search input box. Enter the command you want to find. To exit the search mode, click "Cancel";

6.2.3 Create

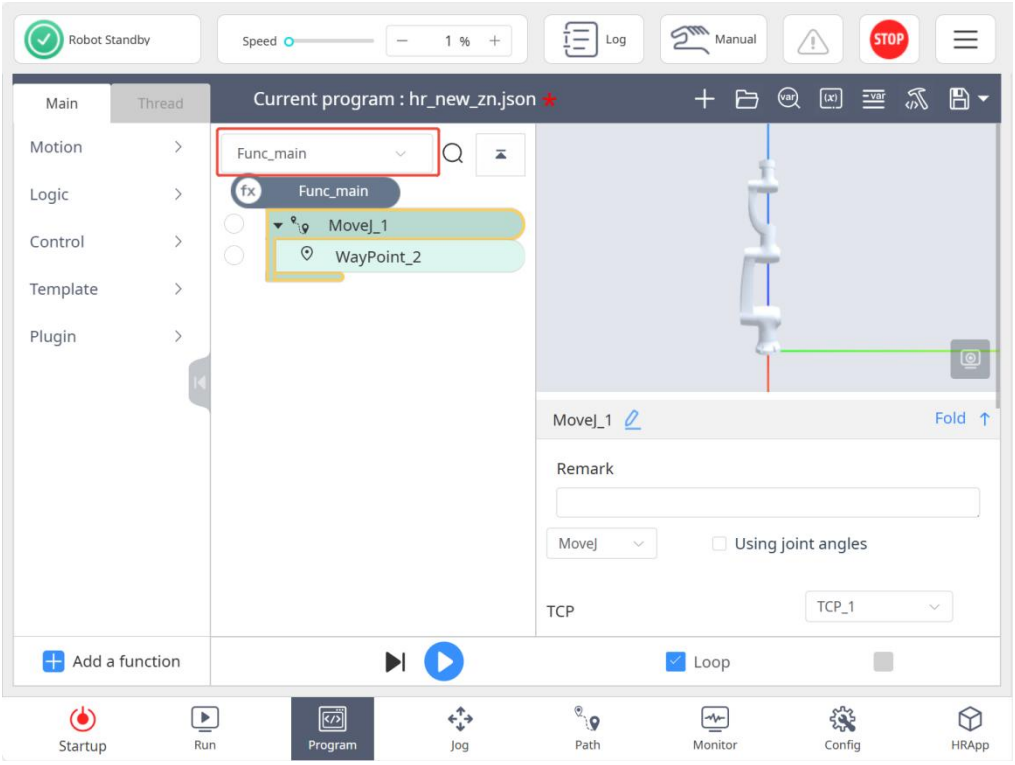
Click "Create", enter the file name in the pop-up box, and click "OK" to create a new program file, as shown in the figure below. If there are other current files unsaved, a popup will appear to ask whether to save the edited files or not;

Fig. 6.2.3.1 Creating a Program File



6.2.4 Function

Fig. 6.2.4.1 Main Function Call



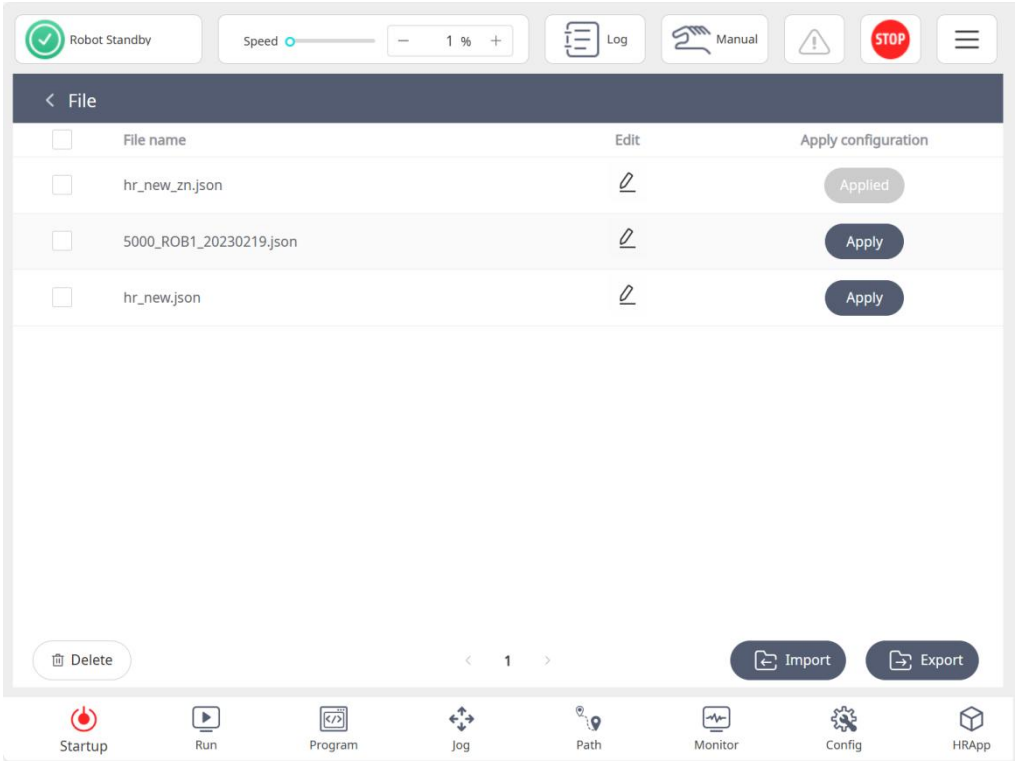
Func_main:Func_main is the main function. It is automatically created when a new program is generated and cannot be deleted. On the Run page, program execution always starts from Func_main.Users can also create additional ordinary functions. Ordinary functions can be called repeatedly by using the Call command.

Func_N: Add a function Click "Add a function", and the function drop-down menu will add a function accordingly, named Func_1, 2, 3...; common functions can be deleted by dragging it to the left menu. The user can create a repeatable function for repetitive operations and repetitively call it using the Call logic command.

In order to facilitate debugging and reading scripts, different processes can also be divided into separate function modules, which can be called and executed through the Call command.

6.2.5 Program List

Fig. 6.2.5.1 Program List



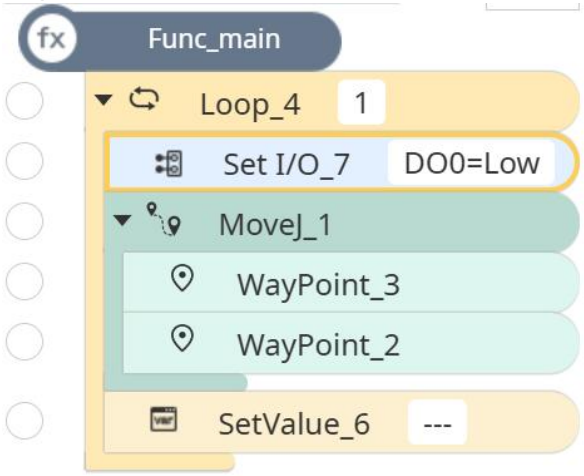
In the Program List, users can create, open, rename, delete, import, and export program files.

- **Rename:** Click "Edit", the file name setting popup appears, and click "OK"to take effect;
- **Delete:** Check the box in front of the file and click "Delete"; the system will prompt: Delete the program or not?
- **Import:** Import files from the default path in the system or from the USB flash drive to the list. After importing, the imported files will be displayed in the list;
- **Export:** Check the box in front of the file and click "Export" to export to the system default path or a USB flash drive;
- **Apply:** To open the program file; Click "Apply", and note that the currently running program remains in a temporary state until saved. If the currently edited program is unsaved, a prompt box will pop up to remind saving the current file.

Users can choose to save the current file or apply the selected file directly, without saving the previous one;

6.2.6 Command

Fig. 6.2.6.1 Program



- **Drag-and-Drop and Move:** Drag-and-drop the command from the command menu to the program list node;
- **Command Color Block:** Classified according to menu category. Green is the motion commands, yellow is the logic commands, blue is the control commands, gray is the template commands, and red is the third-party plugin commands;
- **Command Style:** The command being edited is in the selected style with yellow border, and the length of the command color block changes with that of the parameters and titles;
- **Command Title:** In the case of If command, it displays "if_remarks" after the remarking, with a maximum of 24 characters.
- **Parameter Box:** Some commands have parameter box prompts, indicating the current parameter settings of the command. Up to 24 characters are displayed, which facilitates browsing the command parameters set in the list.

6.3 Motion Command

6.3.1 Move

The motion modes include joint motion (MoveJ), spatial linear motion (MoveL), and circular motion (MoveC). After the motion category is expanded in the command menu, Move and MoveC are displayed. When the user drags a Move command under Func_main, a MoveJ motion node is added by default. Users can switch the motion mode in the drop-down list between MoveJ and MoveL.

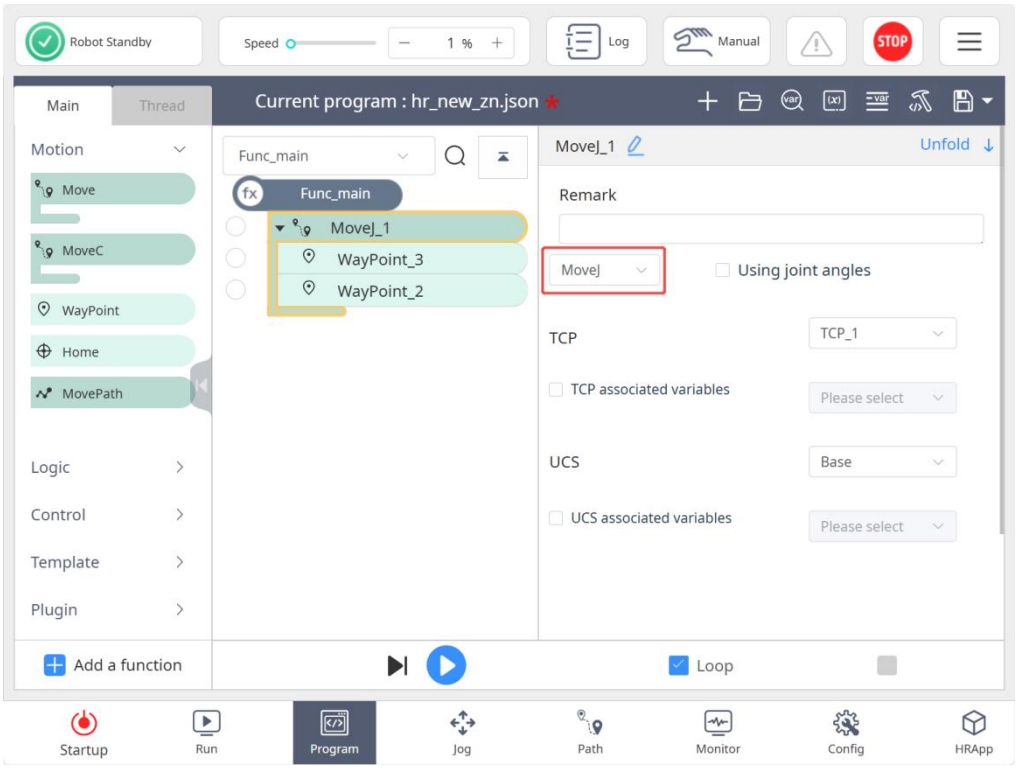
When the user selects Move or other related commands that can add waypoint commands, click the "Point" button at the robot arm's end-effector, and a new

waypoint command containing the current robot arm position will be added under the selected command.

Note: Switching between MoveJ and MoveL does not affect the waypoints under their nodes. However, switching to other motion types, such as from MoveJ to MoveC, may result in the deletion of all waypoints under the node, leaving only the start, intermediate, and end point.

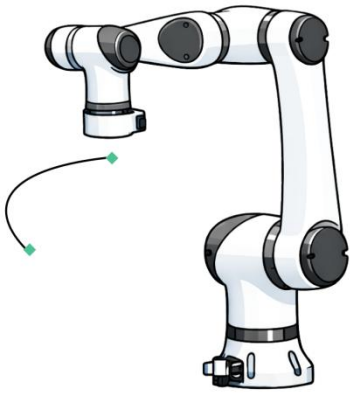
6.3.1.1 MoveJ Joint Motion

Fig. 6.3.1.1.1 MoveJ Motion



The joint axes of the robot arm move synchronously and quickly from one point to another. It is suitable for scenarios where there are no requirements for the motion path of the robot arm's end-effector. The robot arm's end-effector moves along an irregular curve. Please pay attention to whether there is a risk of potential collision.

Fig. 6.3.1.1.2 MoveJ Motion



Parameters include enabling "Using joint angles", Set TCP, Acceleration, and Joint velocity;

– Using Joint Angles

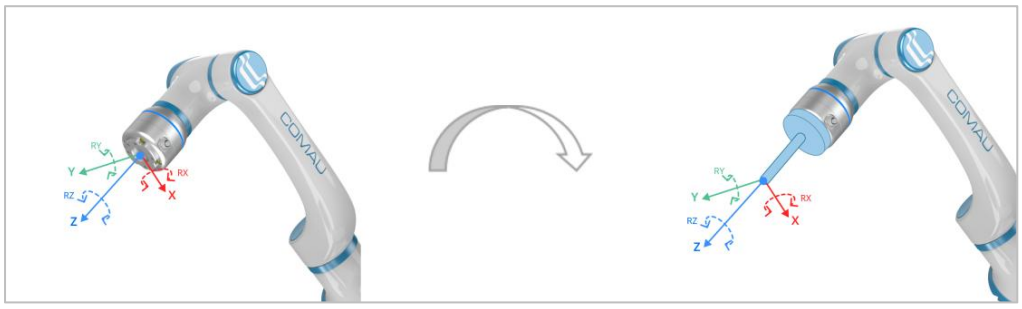
If the "Using joint angles" function is turned on, the "Set TCP" option is not available. In this mode, the motion of the robot arm is determined only by the joint angle of the waypoint, independent of the selected TCP (tool center point).

– Set TCP

The user can switch to the appropriate TCP (tool center point) according to the tool at the end-effector. Select the TCP that matches the current tool at the end-effector from the drop-down menu. When setting a waypoint, the program records the spatial coordinate value of the tool function corresponding to that point. The motion path of the robot arm is determined only by the spatial position of the tool at the end-effector of the waypoint.

By default, the TCP option uses "Factory TCP".

Fig. 6.3.1.1.3 Comparison of Tool Coordinates After Switching TCP



– Acceleration

The user can set the joint acceleration of the robot arm when moving from the current waypoint to the next waypoint.

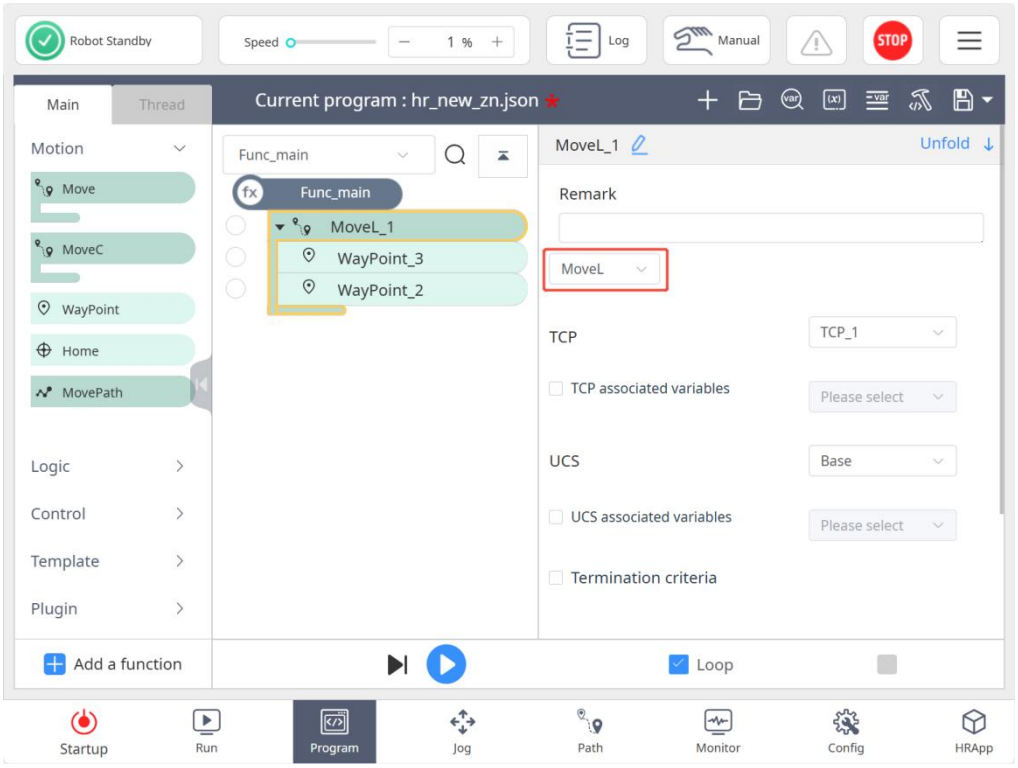
– Joint Speed

The user can set the joint speed of the robot arm when moving from the current waypoint to the next waypoint.

Note: These parameters will be used as the parent properties of the waypoints under the MoveJ motion path, and the same applies to other motion modes.

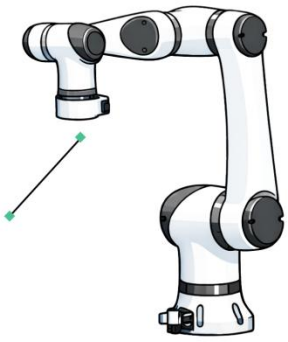
6.3.1.2 MoveL Spatial Linear Motion

Fig. 6.3.1.2.1 MoveL Motion



If the robot arm is desired to move in a straight line, a spatial linear motion command can be utilized. This command enables the tool center point (TCP) to move along a straight-line path from the start point to the end point. When the orientations of the starting point and the end point are different, the TCP orientation will synchronously rotate with the position movement, eventually aligning with the orientation at the end point.

Fig. 6.3.1.2.2 MoveL Motion



Parameters include: Set TCP, Features of the Coordinate System, Acceleration, and Tool speed;

– Set TCP

The user can switch the appropriate TCP according to the tool at the end-effector. Select the TCP that matches the current tool at the end-effector from the drop-down menu. When a waypoint is set, the program records the spatial coordinate value of that point in the selected tool coordinate system. The TCP option is set to "TCP" by default. This parameter is a factory initial value, its spatial coordinates are (X = 0, Y = 0, Z = 0), and its orientation parameters are (RX = 0,

RY = 0, RZ = 0).

When editing, the TCP associated variables can be selected, and the called variable is of string type.

– **Coordinate System**

Select the suitable coordinate system from the drop-down menu. After switching the coordinate system, the path formed by the waypoint will be relatively translated as a whole based on the coordinate system switched to.

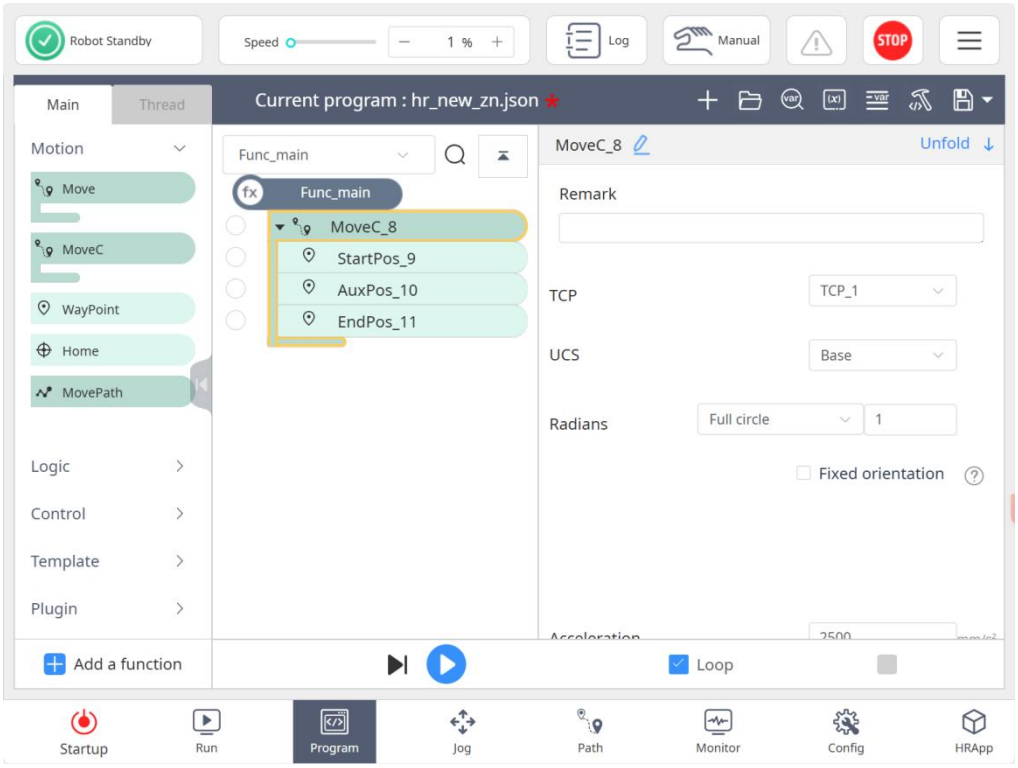
When editing, the coordinate system associated variables can be selected, and **the called variable is of string type.**

– **Acceleration:** The joint acceleration of the robot arm as it moves from the current waypoint to the next waypoint.

– **Tool Speed: The joint speed of the robot arm as it moves from the current waypoint to the next waypoint.**

6.3.1.3 MoveC Arc Motion Command

Fig. 6.3.1.3.1 MoveC Motion



The MoveC command is used to generate a circular or arcuate path. It is based on the geometric principle of defining a circle or arc using three distinct points. The three points must not be colinear and any two points must not coincide. If two points are at the same position or if all three points lie on a straight line, a circular path cannot be determined.

– **Set TCP**

Please refer to the TCP setting description of "6.3.1.1 MoveJ Joint Motion" for details;

— Coordinate System Function

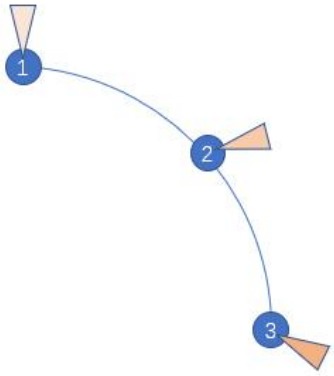
Please refer to the coordinate system function description of "6.3.1.2 MoveL Spatial Linear Motion" for details;

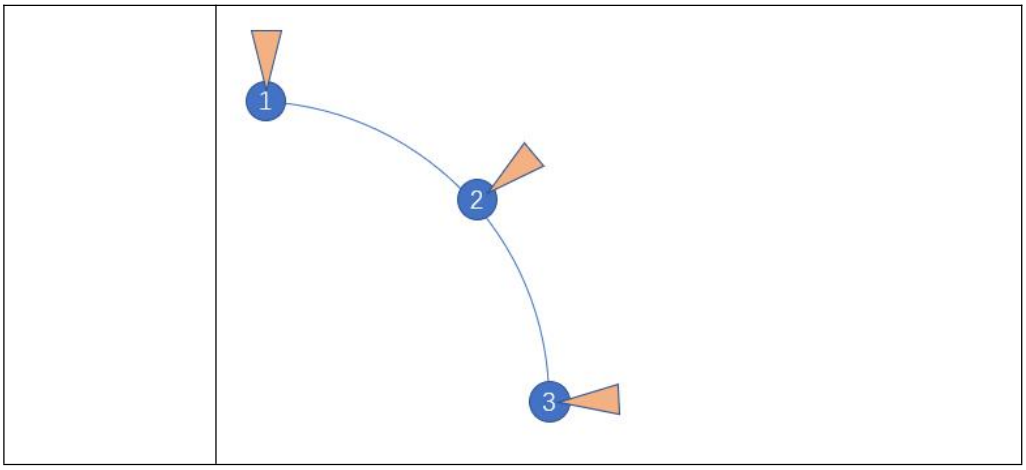
— Blending Radius

Please refer to the blending radius description of "6.3.2 Waypoint" for details;

— Function Description

Table 6.3.1 MoveC Motion function Description

Feature	Description
Circular Arc	There are two types: arc, full circle
StartPos	The current position of the robot arm, that is, the starting point of the arc
AuxPos	A point on the circular path can be set by either "manual teaching" of the PCS coordinate values or by "selecting a variable" to input the coordinate values.
EndPos	The endpoint of the circular path, which is also considered a waypoint in the case of a full circle, can be either manually taught as a PCS coordinate value or input via a variable.
Circular Number	When the arc type is set to full circle, this specifies how many times the robot arm repeats the circular path.
Fixed Direction	There are two types: When the "Fixed direction" option is not selected, the orientation transitions from the start point to the end point, independently from the second waypoint's orientation. As shown in the figure below, points 1, 2, and 3 represent the start point, intermediate point, and end point of the arc, respectively.  When the "Fixed direction" option is selected, the orientation remains consistent with the initial TCP orientation throughout the motion. Points 1, 2, and 3 represent the start point, intermediate point, and end point of the circular arc, respectively. The tool at the end-effector maintains the initial orientation as it moves from the start to the end point along the path, as shown in the figure below.



6.3.2 WayPoint

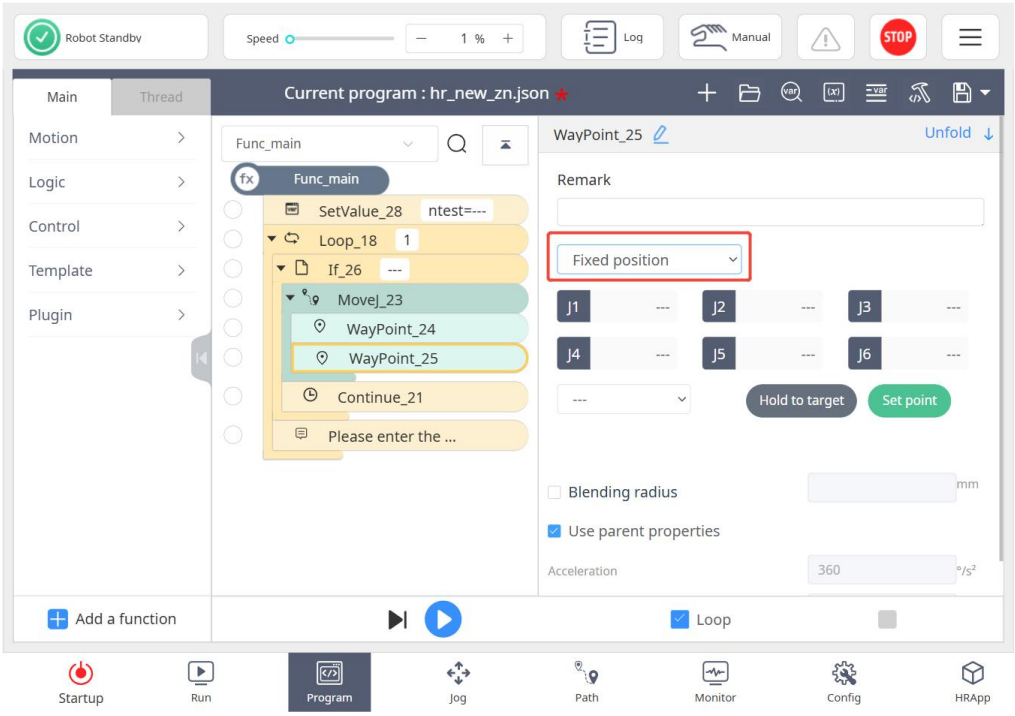
WayPoint is a point on the motion path of the robot arm. A waypoint is a core element of a robot program and is used to indicate a specified position to which the robot arm is moving towards. The fixed position waypoints are taught by physically moving the robot arm to the desired position.

Waypoints are only used within Move nodes and cannot become first-level nodes alone (except within templates). Affected by the parameters in the Move function, waypoints should be taught according to the relationship between the functions applied in the Move function and TCP to achieve the desired TCP position.

Three waypoint position modes are available: fixed position, translation, and variable. Users can switch between these modes according to practical needs. The default mode is "fixed position".

6.3.2.1 Fixed Position

Fig. 6.3.2.1.1 Fixed Position



— Point Information

If the waypoint is within the MoveL function, the position information of the current point is displayed as P [X, Y, Z, RX, RY, RZ]; if the waypoint is within the MoveL function, P [J1, J2, J3, J4, J5, J6] is displayed;

— Set Point

To switch to the "Jog" interface to teach the target position of the robot arm, and set the target motion position of the waypoint called WayPoint_1 waypoint.

— Hold to Target

Press and hold the "Hold to Target" button to move the robot arm to the current point position. A "Target point reached" prompt will appear when it arrives.

— Point List

Click the drop-down menu to select the point, and the point information will display the position information of the current point. To view the specific location of this point, please press and hold "Hold to Target". You can edit the point list on the jog interface.

Note: If a point that's currently being used for a waypoint is deleted from the waypoint list, the corresponding waypoint information will be synchronously deleted and an error reported. The waypoint can be restored by modifying its information.

— Use Parent Properties

By default, the "Use parent properties" is selected, the acceleration and joint velocity (tool velocity) in Move function will be used; if it is not selected, the currently specified acceleration and joint velocity (tool velocity) will be used instead.

— Blending Radius

When the blending radius is configured between two adjacent linear segments, the robot arm will transition smoothly via a circular arc, enhancing operational efficiency. This ensures continuous motion without pausing at waypoints. A smaller blending radius creates a sharper corner, while a larger radius results in smoother cornering.

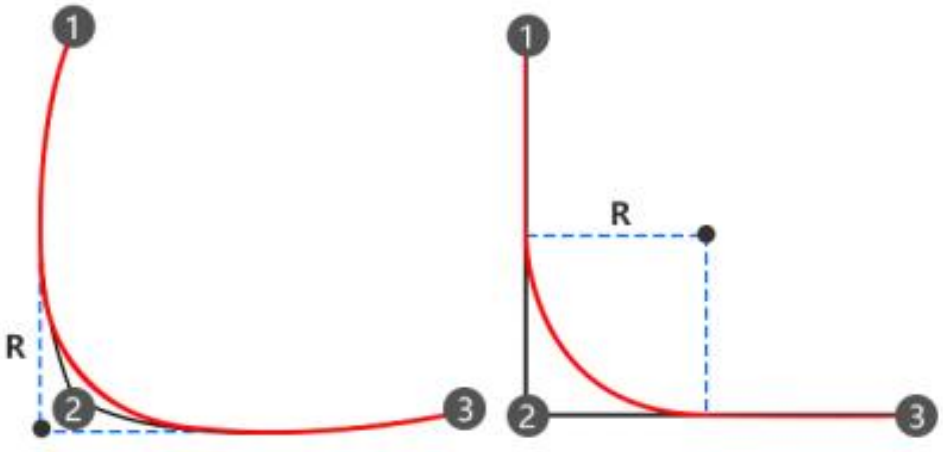
The blending radius enables smooth transitions via arcs. The set radius must not exceed the straight line distance, otherwise, a calculation error will be reported. If two straight lines can be connected into one straight line, the set radius is invalid.

Note: In MoveL and MoveJ functions, selecting the checkbox enables blending radius; deselecting it makes the robot stop at the point before moving to the next. In MoveC function, selecting the checkbox uses the current point's blending radius, while deselecting it applies the blending radius set in MoveC.

Example:

Blending Radius in MoveJ and MoveL Commands

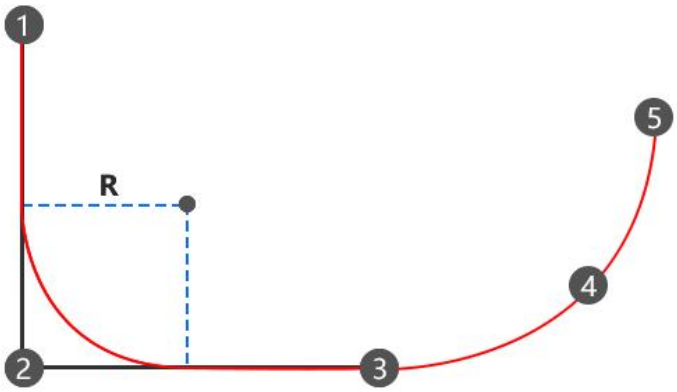
Fig. 6.3.2.1.2 Blending Radius in MoveJ and MoveL Commands



As shown in the figure, P1 is the current position of the robot arm, P2 is the target point of the first command, and P3 is the target point of the second command. The red path is the calculated path. The left and right figures are the MoveJ and MoveL commands respectively. The setting of the blending radius should be done at the next target point after the skipped point.

Blending Radius in MoveC Command

Fig. 6.3.2.1.3 Blending Radius in MoveC Command

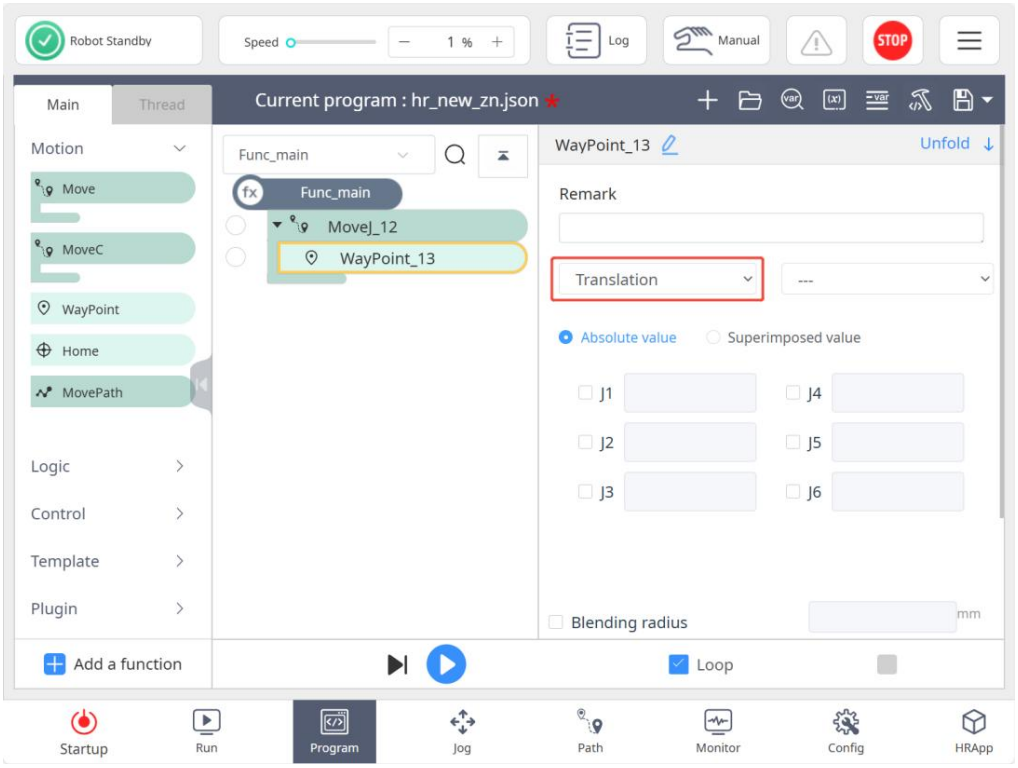


P1 is the current position of the robot arm, P2 is the target point of the first MoveL command, and P3 is the target point of the second MoveL command, which is also the starting point of the arc. P4 is the intermediate point on the arc path, and P5 is the target position of the arc.

As shown in the figure, the red path represents the calculated motion trajectory. A blending radius is applied between the two linear segments to achieve a smooth arc transition. The combination of MoveL + MoveC ensures continuous motion between the linear path and the arc.

6.3.2.2 Translation

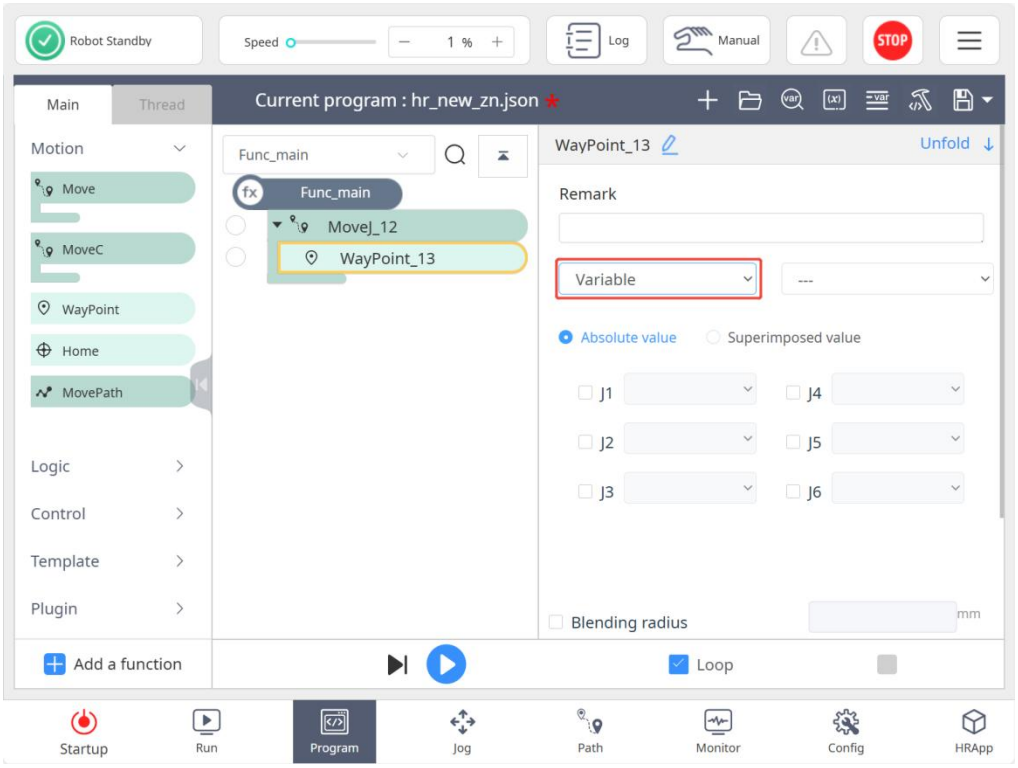
Fig. 6.3.2.2.1 Translation



- **Point List:** Select the specified point in the drop-down menu of point list in the upper right corner; for example, Point _1 is the waypoint 1;
- **Absolute Value:** Move to the absolute coordinate position in the currently selected user coordinate system (such as the Base coordinate system).
- **Superimposed Value:** The last point serves as the reference point, and the movement distance is based on the superimposed value;
- **Select Axes:** Specify certain robot axes for absolute or superimposed translation (J1, J2, J3, J4, J5, J6; X, Y, Z, RX, RY, RZ)
- **Blending Radius:** For details, please refer to the blending radius description of 6.3.2 Waypoint;
- **Use Parent Properties:** By default, the "Use parent properties" is selected, the acceleration and joint velocity (tool velocity) in Move function will be used; if it is not selected, the currently specified acceleration and joint velocity (tool velocity) will be used instead.

6.3.2.3 Variable

Fig. 6.3.2.3.1 Variable



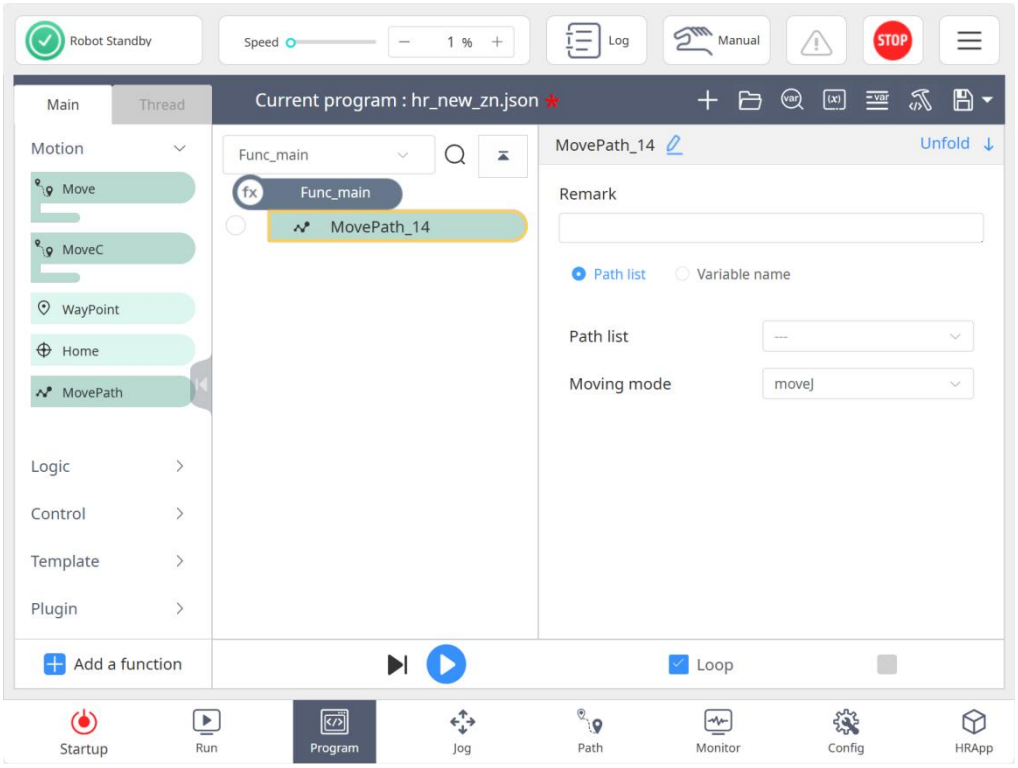
Variables can be added to the variable list. Feedback point data can be received via the script command and assigned to the variable.When the robot arm moves to the point, it moves based on the obtained variable position;

6.3.3 MovePath

MovePath is the command used for continuous path motion of the robot arm. A valid path must be created in advance before this command can be used.

Drag MovePath under Func_main, then select the corresponding path name and motion mode from the path list. Currently, only MoveJ is supported. On the Path page, users can modify the speed ratio of path execution through the path setting function.

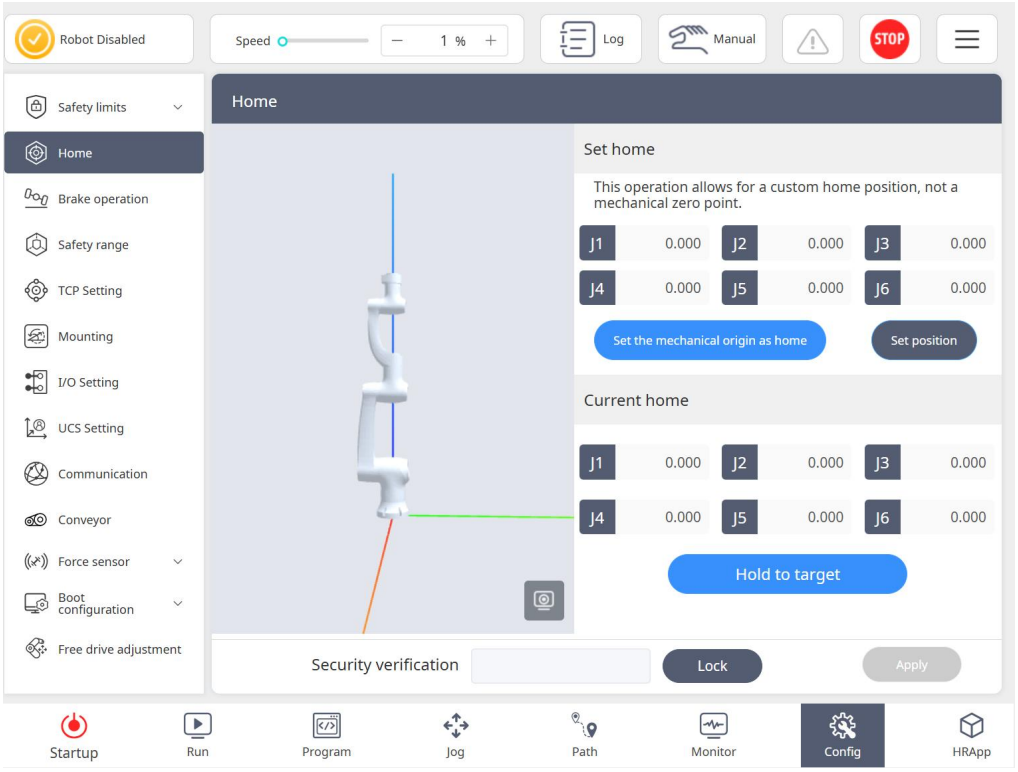
Fig. 6.3.3.1 MovePath Feature



6.3.4 Home

Home is the motion command that moves the robot arm back to the configured safety home position.

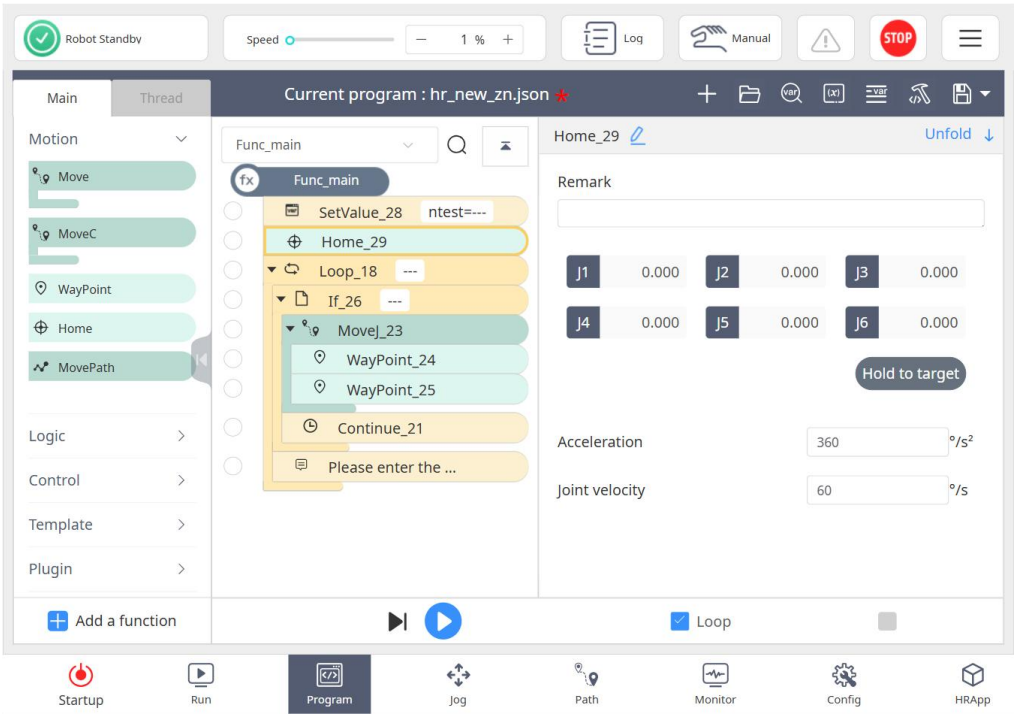
Fig. 6.3.4.1 Configuration Interface—Home



The safety home position can be modified by the user in Configuration > Home. Drag the Home command under Func_main. The current home position is

displayed on the right side, and the speed and acceleration can be configured. When this command is executed, the robot arm returns to the configured safety home position.

Fig. 6.3.4.1 Program Interface—Home



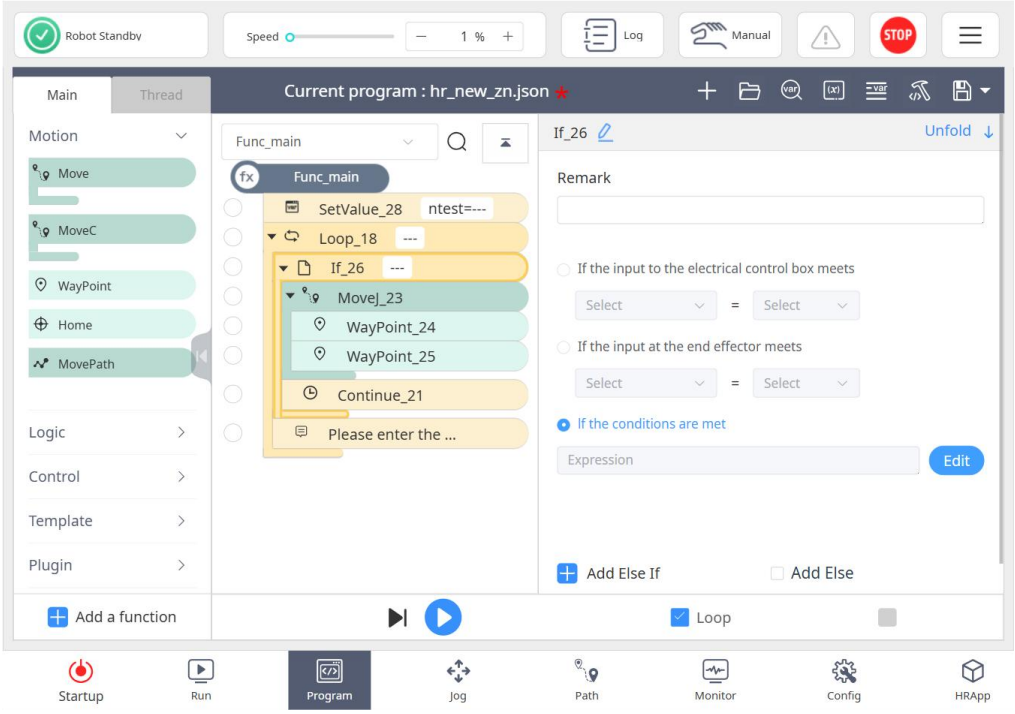
6.4 Logic Command

6.4.1 If

The conditional command guides the robot arm to perform corresponding motions based on different conditions. When the If condition is met, the robot arm executes the motion sequence under the If module. Once execution begins, the robot arm completes all commands within the If module, regardless of whether the condition remains true during execution.

Each If command can nest multiple Else If branches and one Else branch. When the If condition is true, the robot arm will execute the command in the If module; When the If condition is not met, it checks each Else If condition in order and executes the first matching branch. If no conditions match, it executes the Else branch commands.

Fig. 6.4.1.1 If Condition Command



Three judgment conditions are provided:

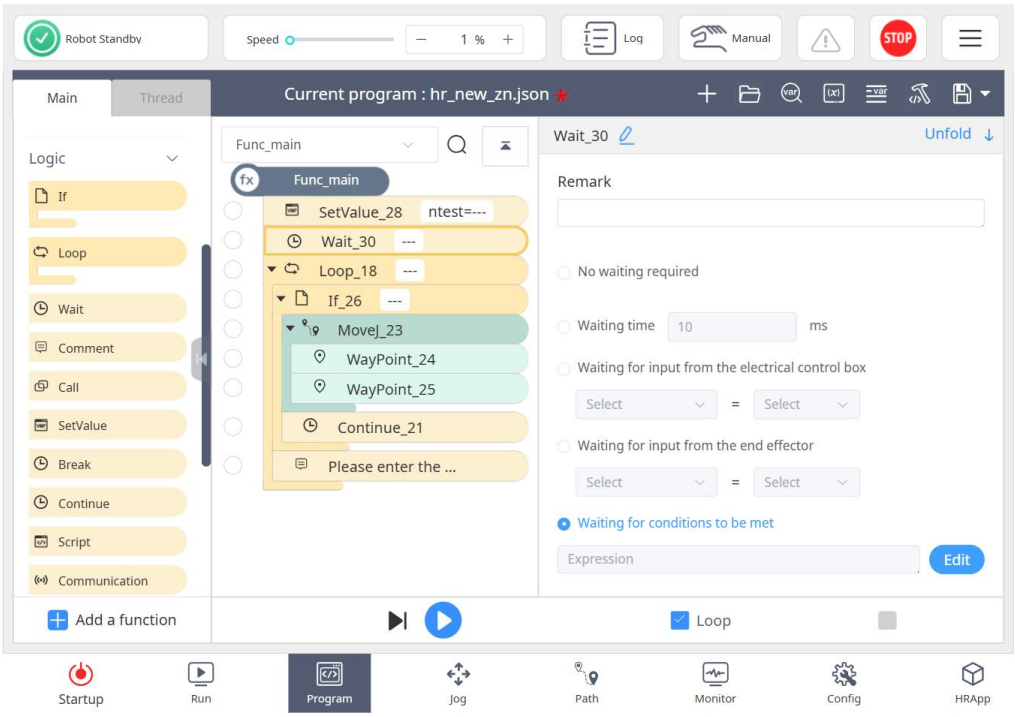
- Judgement of the input I/O status of the electrical control box;
- Judgement of the input I/O status at the robot arm's end-effector;
- Judgment via custom condition expressions.

6.4.2 Wait

Wait Command: During program execution, the program will wait for a set time or until a condition is met before continuing.

Waiting for the specified time, I/O signals, or expressions.

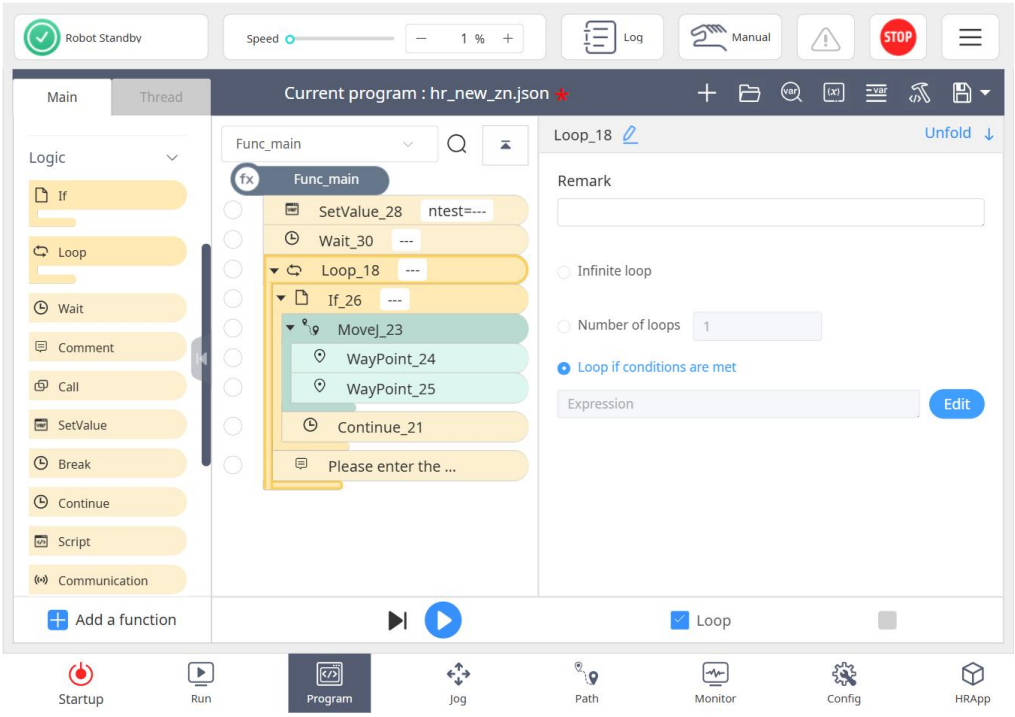
Fig. 6.4.2.1 Wait Command



6.4.3 Loop

Loop Command: This logic repeats a program sequence until the set number of iterations is exceeded or a preset condition is no longer met. The loop conditions can be set as infinite, a specific number of iterations, or a custom condition expression.

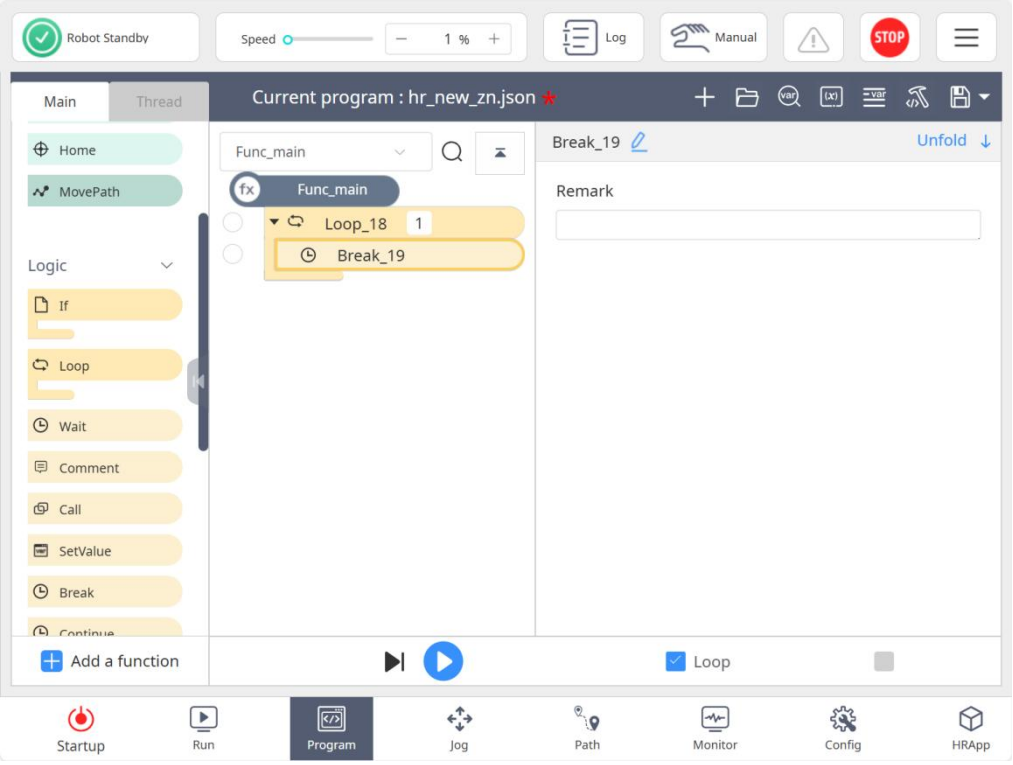
Fig. 6.4.3.1 Loop Command



6.4.4 Break

When a loop encounters a Break statement, it stops immediately and resumes execution with the next statement outside the loop. Any remaining statements in the loop body and the loop condition are ignored.

Fig. 6.4.4.1 Break Terminates the Loop Command



Note:

- Break command exits only the innermost loop. For nested loops, use Break command in each loop layer to exit.
- Break command is only valid within a Loop node and can not be used elsewhere.

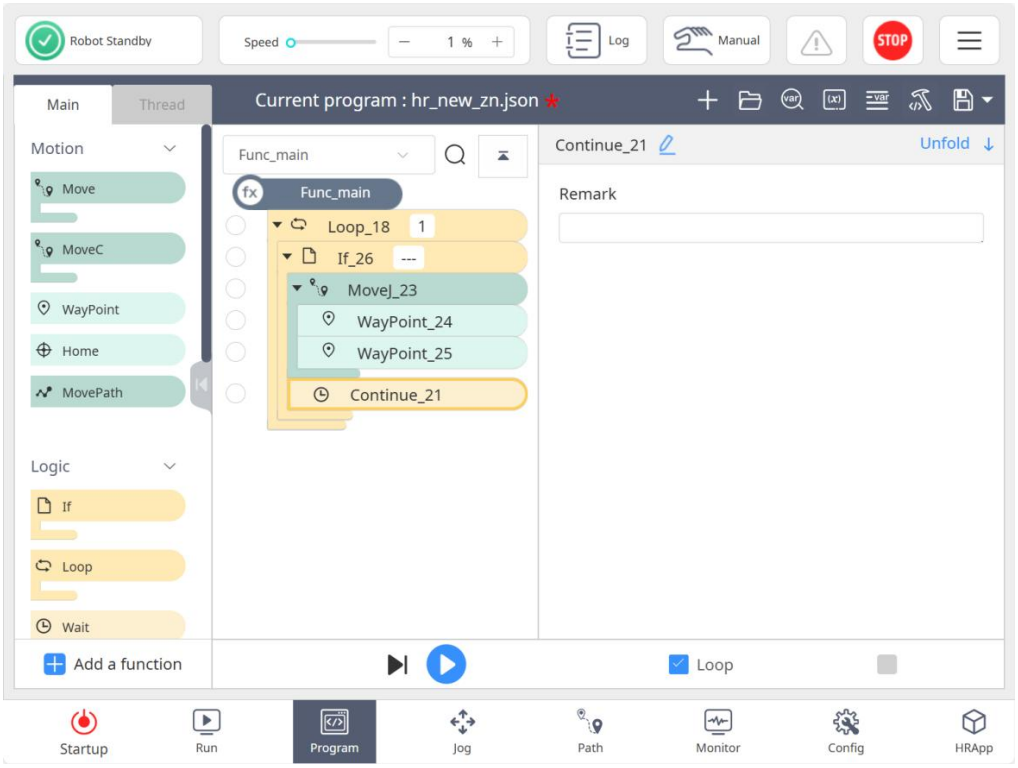
6.4.5 Continue

When a loop encounters a Continue statement, it skips the remaining codes in the current iteration and proceeds to the next loop iteration's condition check. If this occurs in the last loop iteration, Continue command behaves like the Break command, exiting the loop.

Note:

Continue terminates the current loop iteration but doesn't exit the entire loop. Instead, it proceeds to the next iteration's condition check.
It only ends a single loop cycle and does not stop the loop's continuation.
Continue is only used within a Loop node.

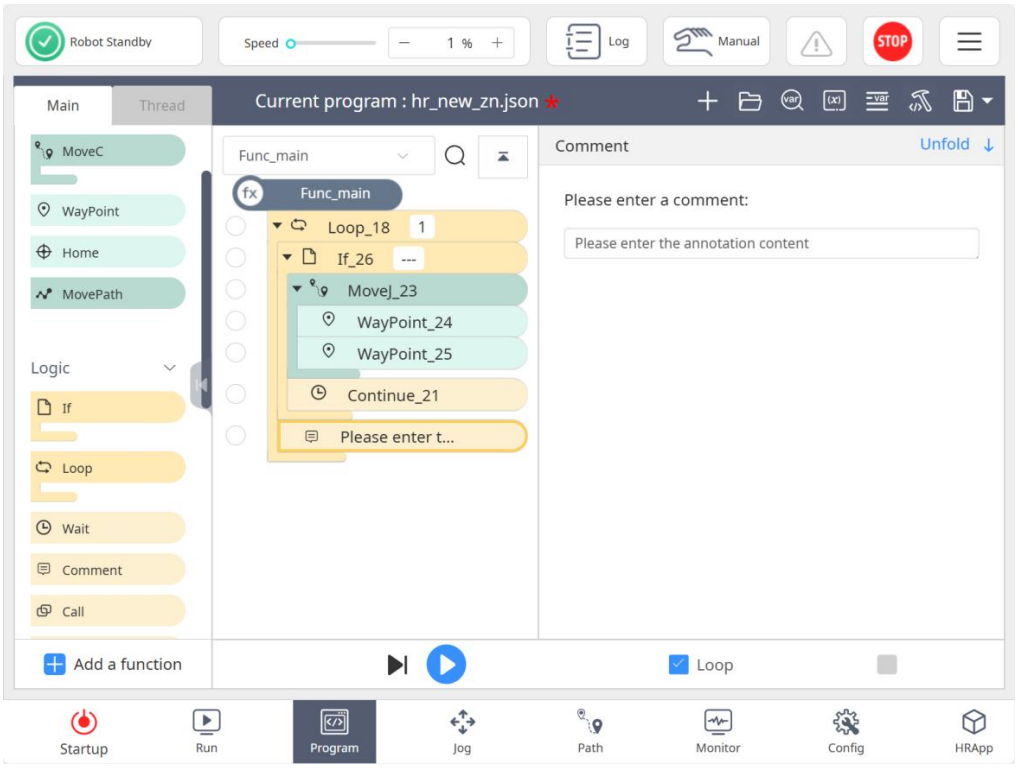
Fig. 6.4.5.1 Continue Skips the Current Iteration of the Loop Command



6.4.6 Comment

Comment is used to add explanatory notes and can be interspersed anywhere to explain and clarify function statements

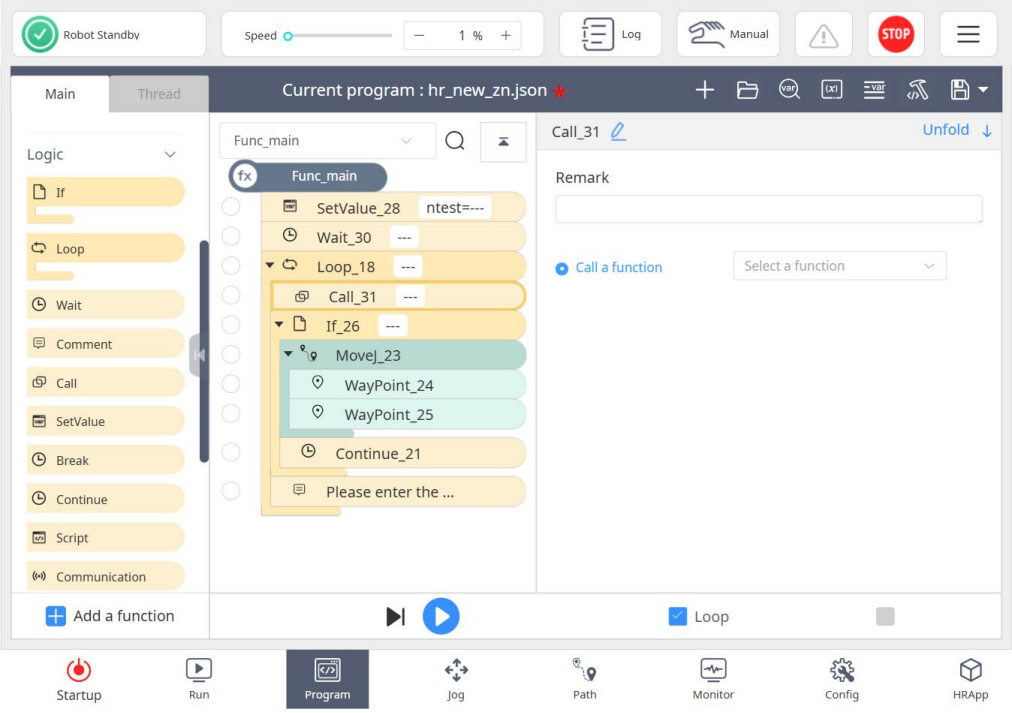
Fig. 6.4.6.1 Comment Statement



6.4.7 Call

Call is the function call command, used to invoke functions defined in the Function section. When the program encounters a Call command, it jumps to the specified function and executes its logic. After the function completes, the program returns to the statement immediately following the Call command and continues execution.

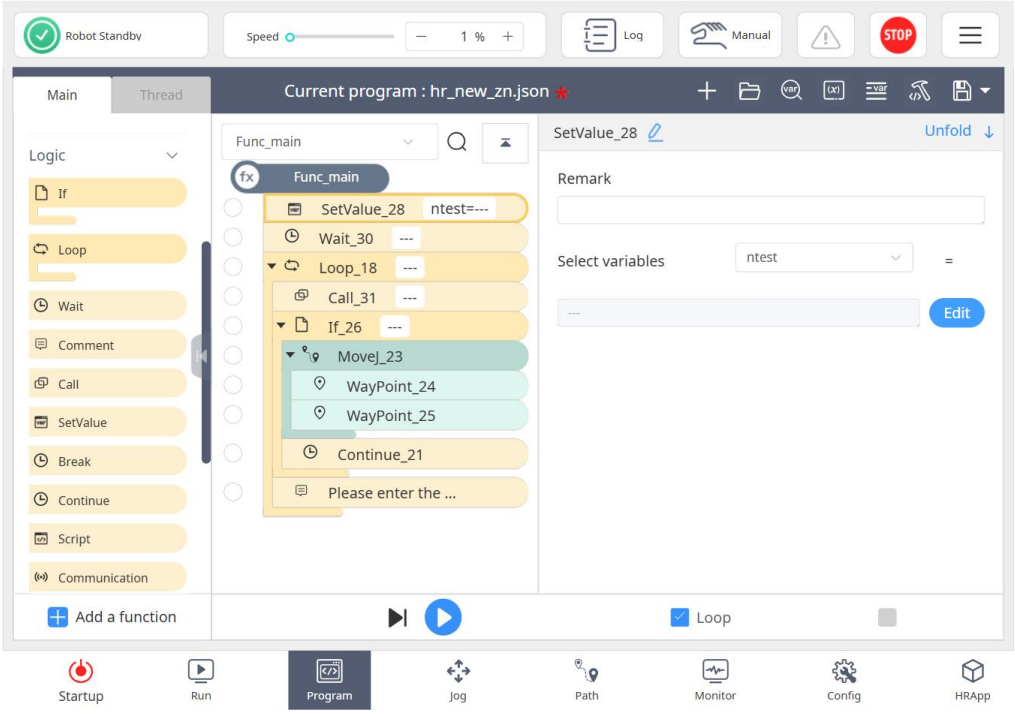
Fig. 6.4.7.1 Call Function Command



The function to be called can be directly selected from the drop-down menu of this instruction. When the program reaches the Call instruction, it will jump to the selected function and execute its content.

6.4.8 Set Value

Fig. 6.4.8.1 Set Value for Variable Command

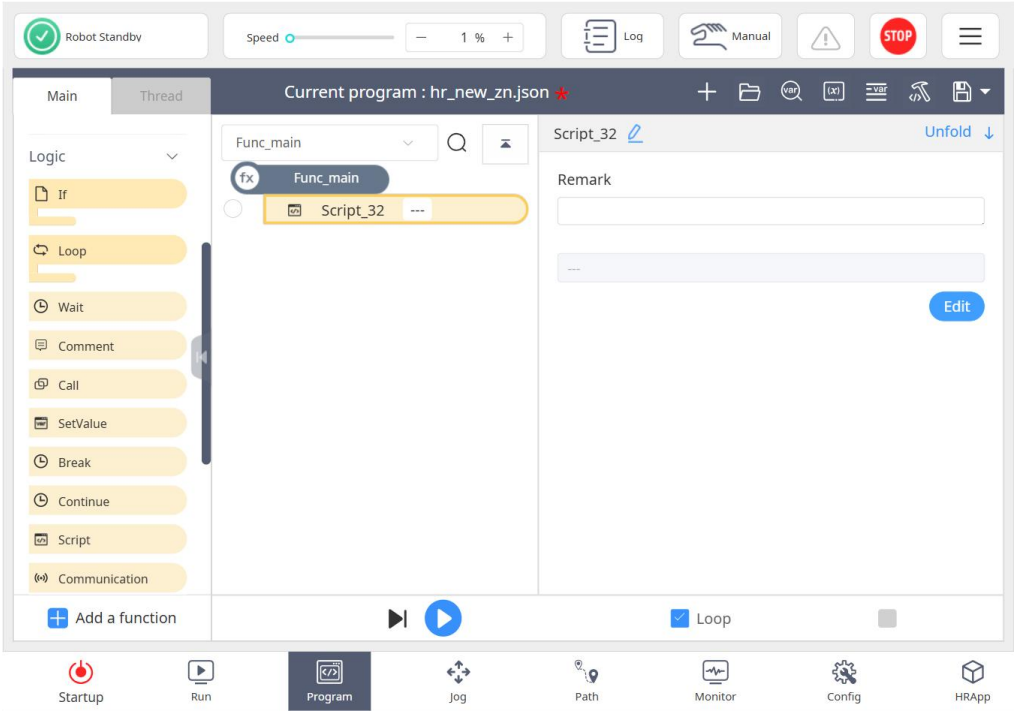


When assigning values to variables, users can edit expressions to calculate new values and assign the results to the selected variables. This method allows for flexible variable updates based on dynamic calculations or logical operations.

6.4.9 Script

Via script, robot position data acquisition and communication can be carried out;

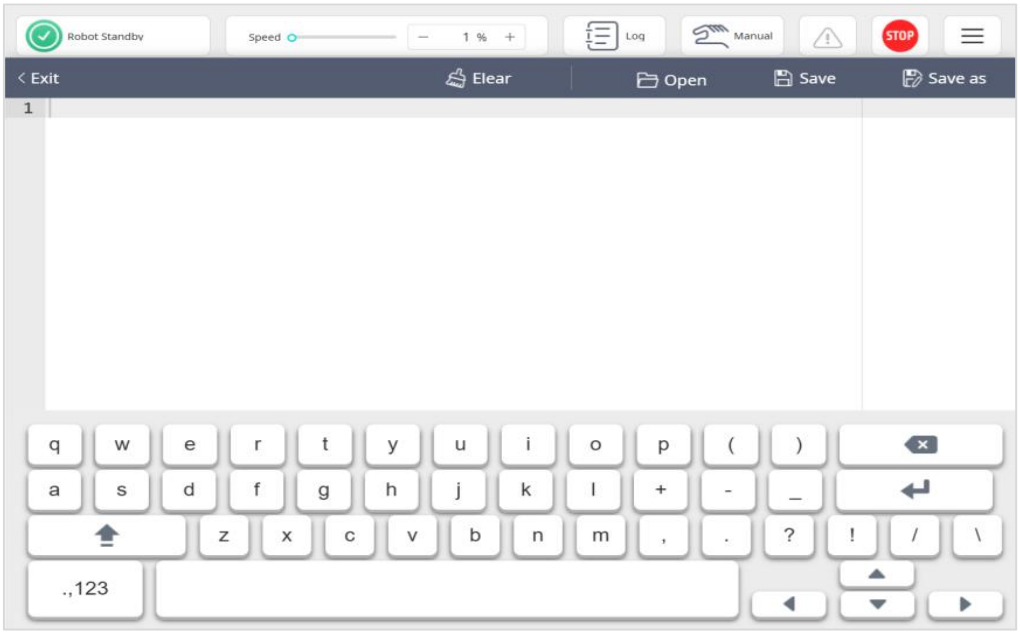
Fig. 6.4.9.1 Script



Users can import and export Python files from other sources into the script for editing, or create scripts directly. Editing must be done in the file editing interface.

Edited files must be saved before they can be referenced in the program, and their names will appear in the Script property information. For more script-related instructions, please refer to the documentation titled "Comau Script Library Manual".

Fig. 6.4.9.2 Script File

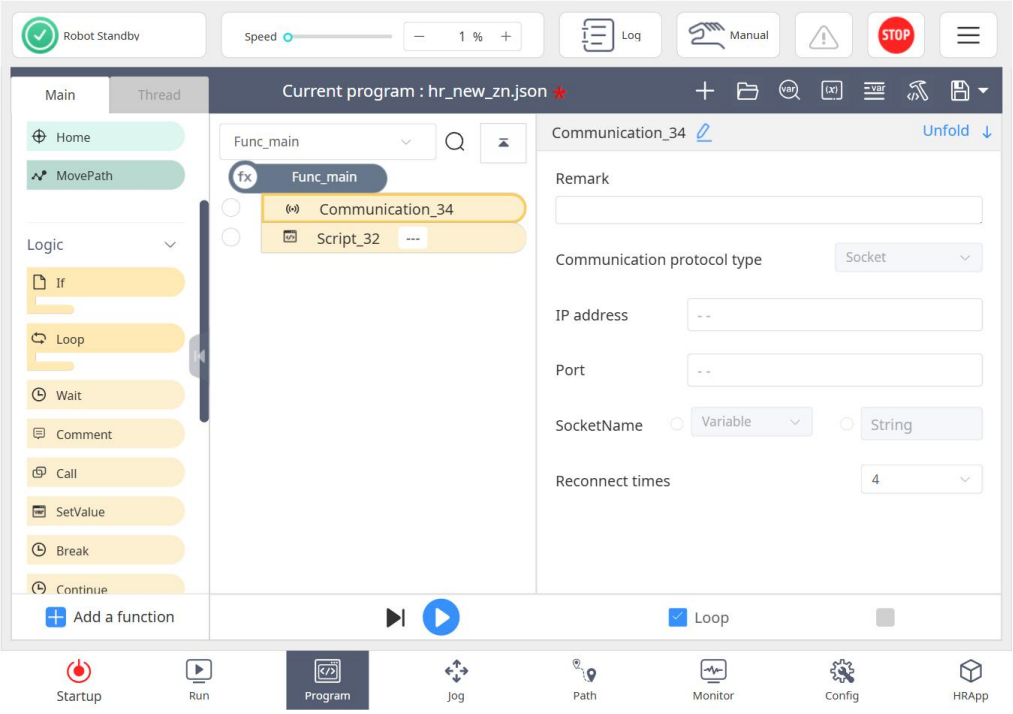


- **Exit:** Quits the editing interface. When exiting, the system checks if the file is saved. If not, it prompts the user to save before exiting. After exiting, the Script properties display the file name, indicating successful program reference;
- **Clear:** Clears all codes after user confirmation; a pop-up window will appear to prompt whether to clear and then perform clearing;
- **Open:** Opens files from the default system path or other sources. If a script is being edited, it prompts the user to save before importing;
- **Save:** Saves files to the default path or exports via USB, with a prompt to name the file;
- **Save As:** Save as a file to the system default directory or exports via USB, with a prompt to name the file;

6.4.10 Communication

Used for TCP communication with external devices when the controller acts as a client.

Fig. 6.4.10.1 Communication

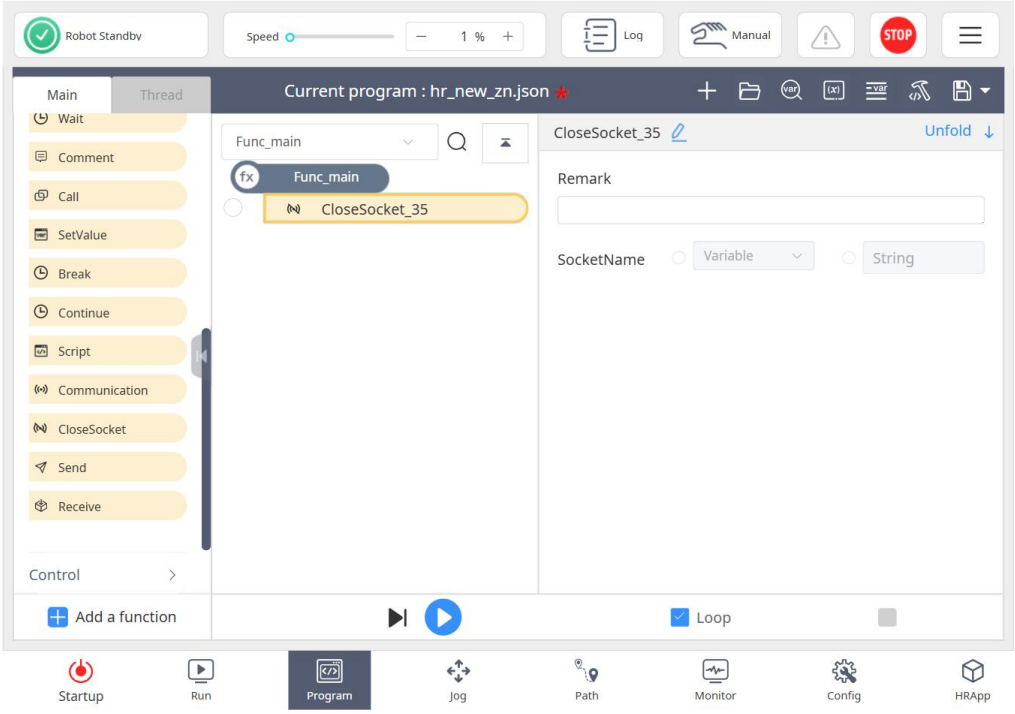


Connect to the TCP server using its IP and port number, and assign a connection name (SocketName) via variables or an input box. If the Communication connection succeeds, Send and Receive commands use this SocketName to interact. If reconnection attempts exceed the specified limit, an error pop-up window will appear and the error is logged. Multiple Communication commands can establish several connections to the same or different servers.

6.4.11 CloseSocket

Used to close the established Socket connection;

Fig. 6.4.11.1 CloseSocket

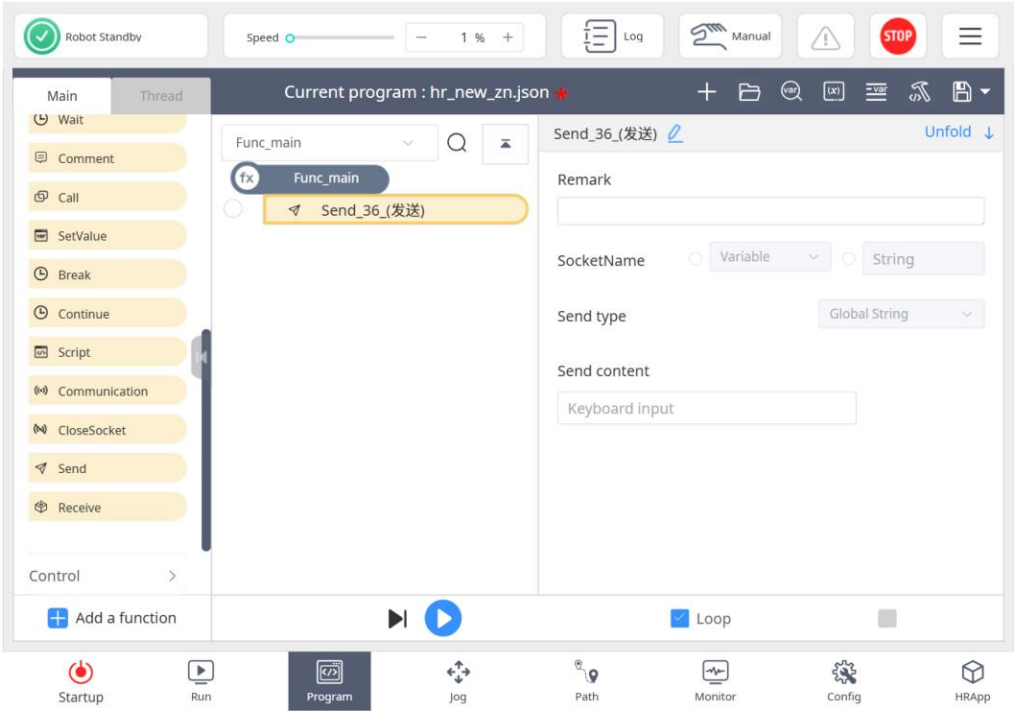


Specify the connection name (SocketName) of the connection to be closed. Set SocketName via a string variable or input box. Ensure the connection is established with the Communication command before calling this command.

6.4.12 Send

Sends string data to a server via an established Socket connection.

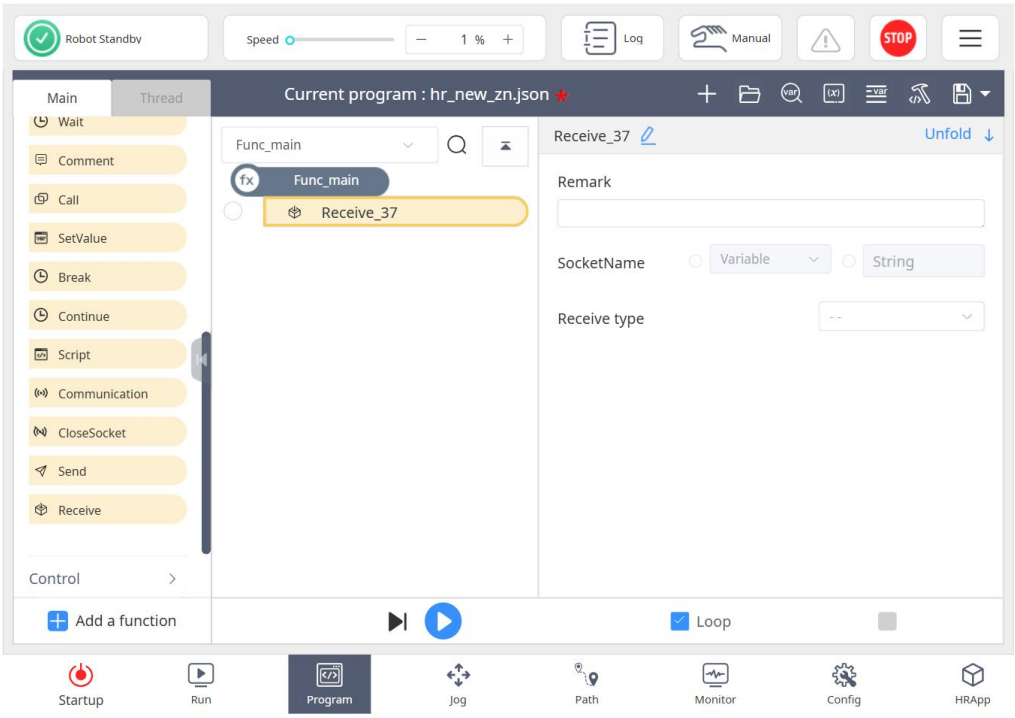
Fig. 6.4.12.1 Send Data



6.4.13 Receive

Receives string data from an established Socket connection and can split/process it with delimiters.

Fig. 6.4.13.1 Receive Data



- **Global String:** Receives and stores messages directly into the receiving variable without any processing. The stored string exactly matches the source string from the server.
- **Split String:** After receiving the message, it is first truncated according to the start and end fields, then split into segments based on delimiters. These segments are then sequentially stored in multiple receiving variables (e.g., Receiving Variable 1, 2, 3, etc.).

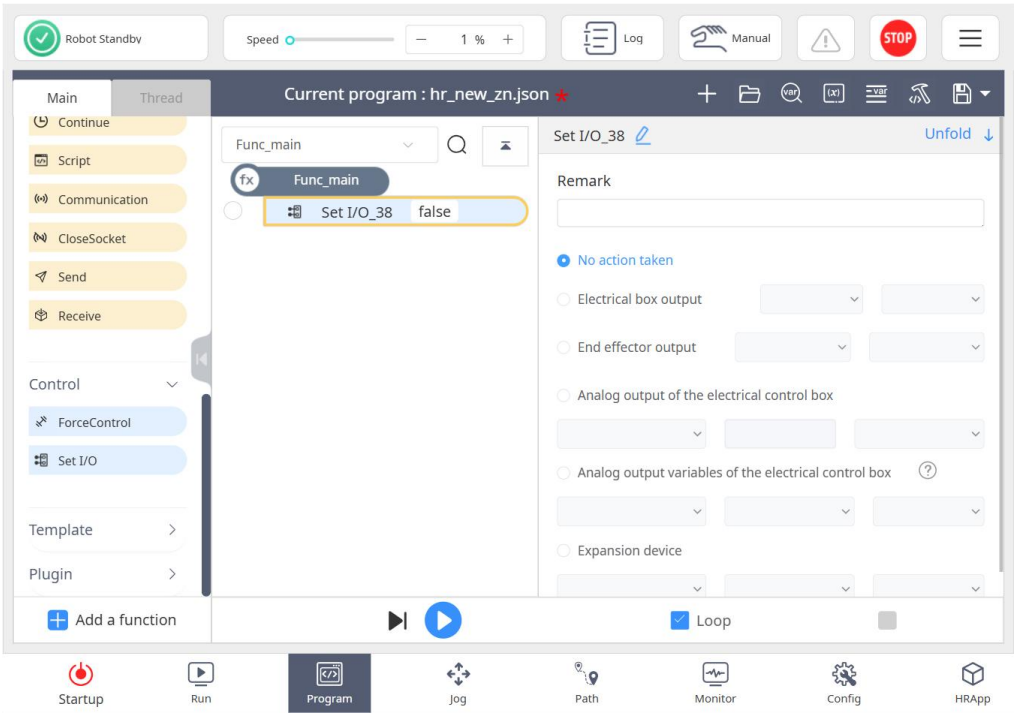
6.5 Control Command

6.5.1 Set I/O

Set the input/output (I/O) status for the electrical control box and the end-effector, as well as the current or voltage values of the electrical control box's analog output channels. Also, set the variable values of current or voltage for these channels. If Modbus is configured, set its output status to high or low.

Set the electrical control box's output I/O status, which involves setting the corresponding I/O signals to either high or low. When the robot executes the Set I/O command, the I/O status changes, after which subsequent commands are carried out.

Fig. 6.5.1.1 Set I/O Command



6.5.2 Set Payload

The Set Payload command is used to switch between different payload configurations during program execution. By specifying a saved payload name, the robot arm can dynamically adjust payload parameters to adapt to the mass and center-of-gravity characteristics of different end-effectors.

Motion Characteristics

If there are waypoint motion-related commands before or after this command, blending (motion blending) is not supported.

When this command is executed, the robot arm will immediately update the payload parameters, which will affect the torque control of subsequent motions.

After payload switching, the dynamic characteristics of the robot arm will be adjusted accordingly to ensure motion accuracy.

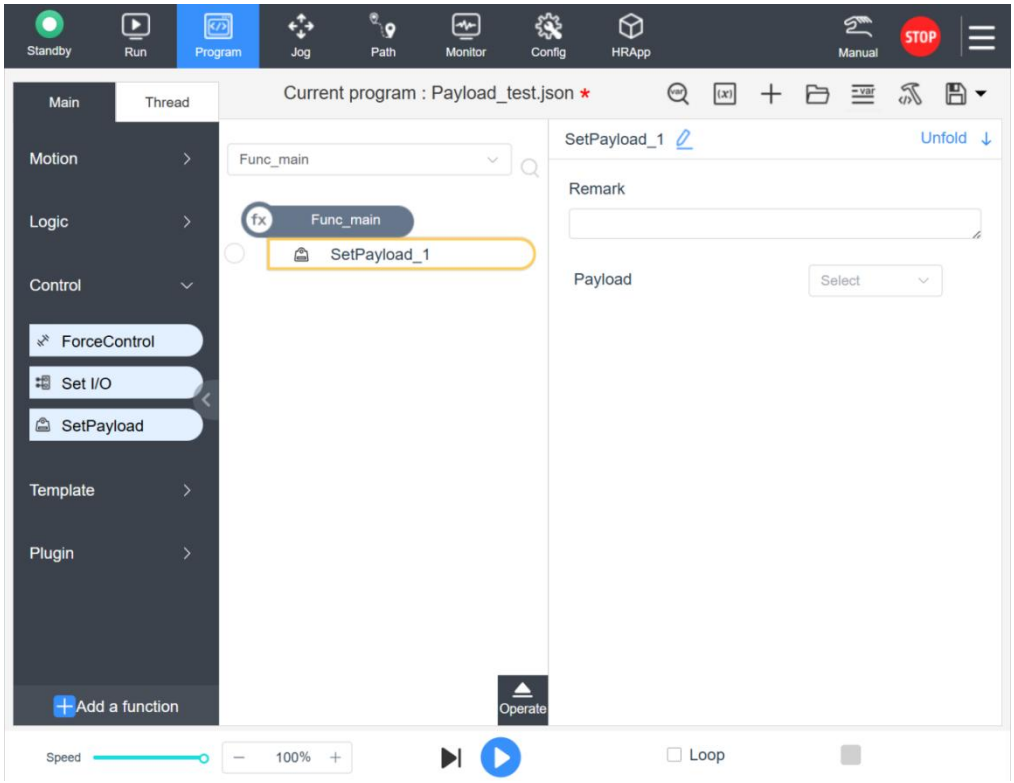
Typical Application Scenarios

After the robot replaces the end-effector, automatically switch to the corresponding payload parameters.

Dynamically adjust the payload when processing workpieces of different weights in the same program.

Switch payload configurations according to the currently used tool in multi-tool collaboration tasks.

Fig. 6.5.2.1 Set I/O Command



6.5.3 ForceControl

The ForceControl command is used to enable force control mode during robot motion, so that the robot can interact compliantly with the external environment according to the configured force or torque parameters. By configuring the control mode, reference coordinate system, and force-control-related parameters, users can implement application scenarios such as constant-force contact.

When program execution reaches the ForceControl command, the system enters the corresponding force control state according to the current settings, and the motion is executed subject to the configured speed limits.

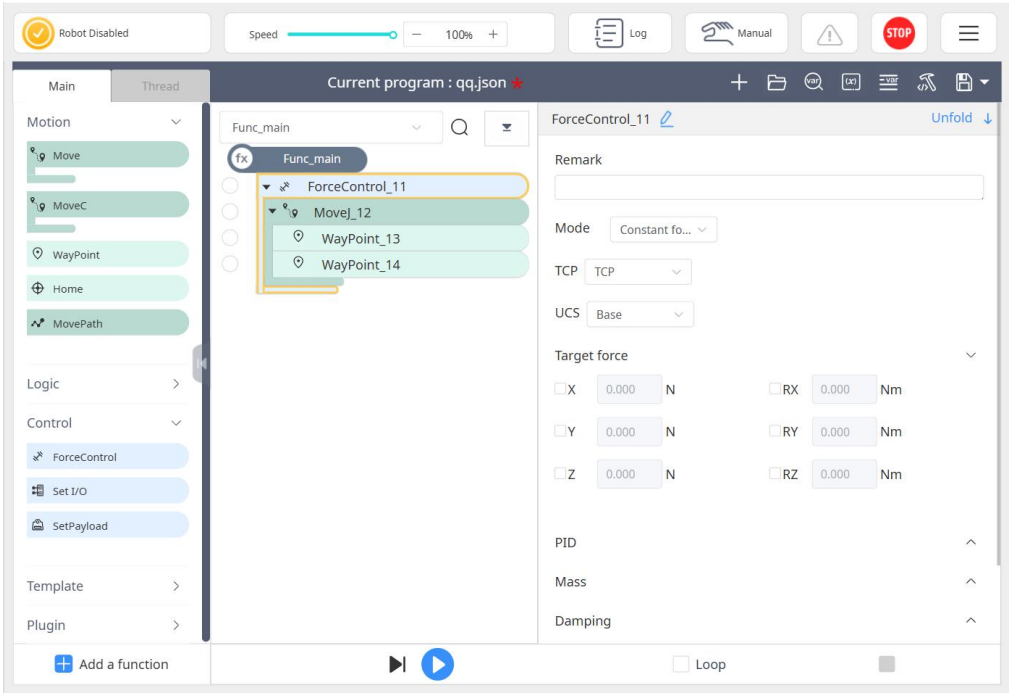
The commonly used parameters are described below:

- **Mode:**Used to select the type of force control, such as constant force control. Different modes correspond to different control strategies.
- **TCP:**Specifies the tool coordinate system currently used for force control calculation. The system performs force and pose calculation based on this TCP.
- **UCS:**Sets the reference coordinate system for force control (such as Base), which is used to define the force direction and control reference.
- **Target Force:**Sets the target force or torque in each direction, and is used to control the contact state between the robot and the workpiece.
- **PID:**Force control tuning parameters used to optimize response speed and stability.
- **Mass / Damping / Stiffness:**Used to adjust the compliance characteristics and dynamic response of the system, thereby improving stability and safety during contact.

- **Linear Speed:**Sets the maximum linear motion speed in force control mode (mm/s).
- **Angular Speed:**Sets the maximum angular speed in force control mode (°/s).

Note:
The ForceControl command is usually used together with motion commands or process steps. Before entering force control mode, make sure that the robot TCP and coordinate system have been configured correctly, and confirm that the external environment is safe, so as to avoid abnormal contact or impact caused by improper parameter settings.

Fig. 6.5.3.1 ForceControl Command

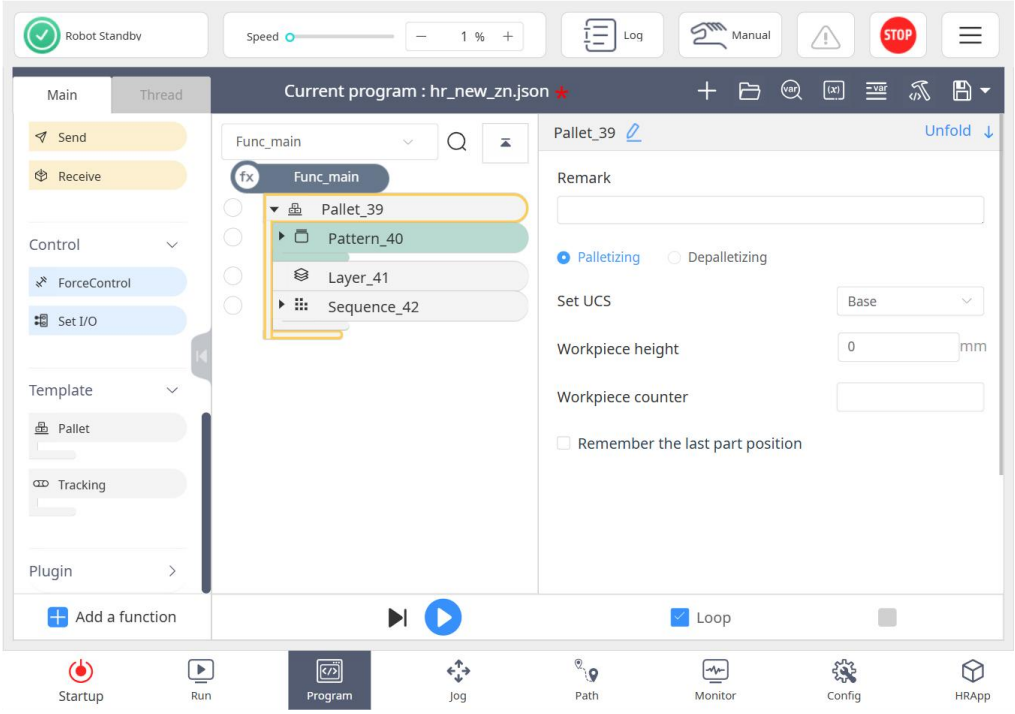


6.6 Template

6.6.1 Pallet

The Pallet command enables continuous object (workpiece) picking, palletizing, and depalletizing on a fixed pallet. The user can set parameters including: coordinate systems feature of palletizing and depalletizing, pallet coordinate system, workpiece height, a workpiece counter, and options to remember the last workpiece position, run pre-start or post-end programs. The workpiece counter can be viewed in "Current Variables" during program execution. If "Remember Last Workpiece Position" is selected, the program resumes from where it stopped previously. Selecting 'pret-start procedure' or 'post-process procedure' allows adding extra logic or control commands before or after the Pallet command.

Fig. 6.6.1.1 Pallet Template



In addition, the pallet command includes three "sub-commands": "Pattern", "Layer", and "Sequence".

6.6.1.1 Pattern

Select a pallet pattern from three options: "Straight Line", "Block", and "Irregular". Take the straight - line pattern for example. After selecting it, define a start point (P1) and an end point (P2). Also, set the "Interval Count," which determines the total number of items for palletizing, as shown in the figure below:

Fig. 6.6.1.1.1 Mode Selection

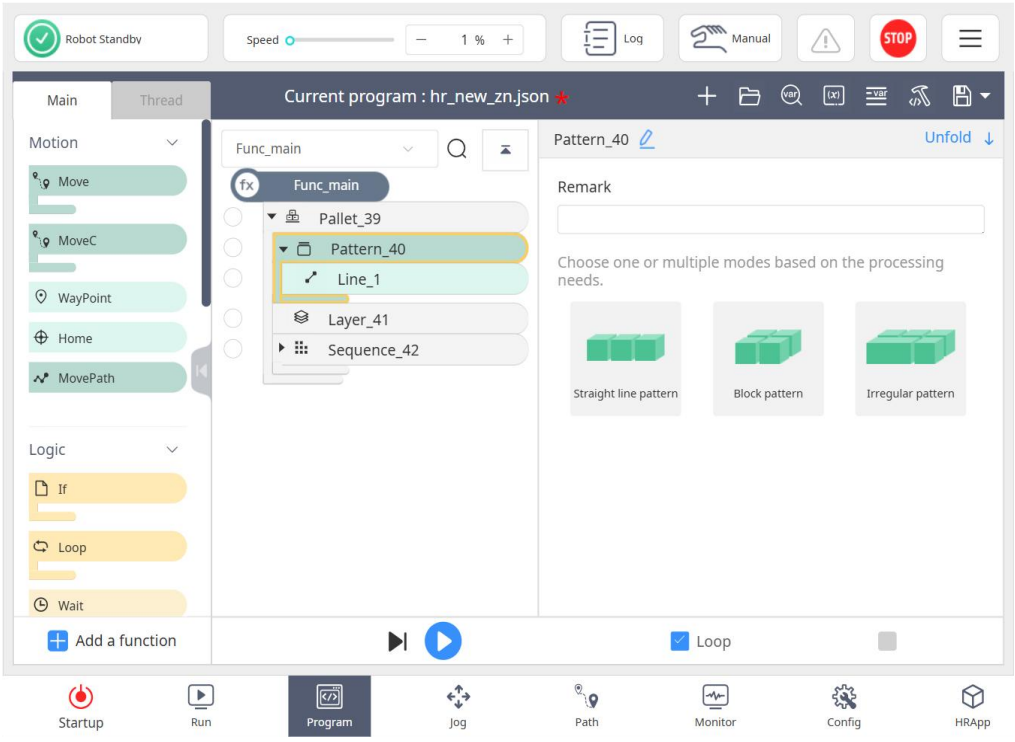
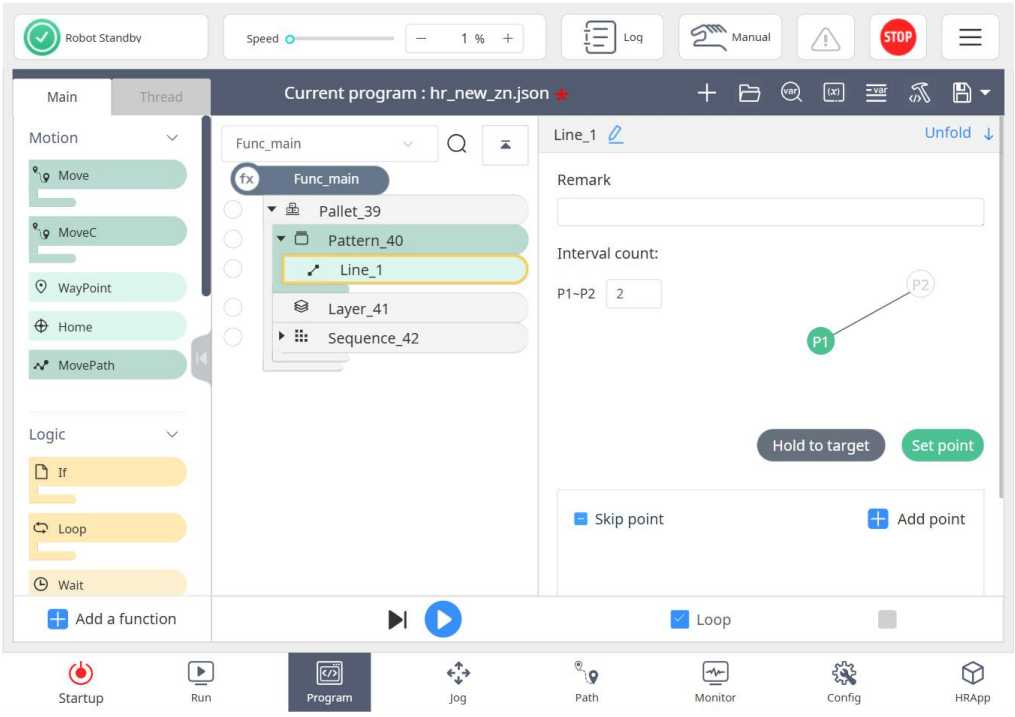


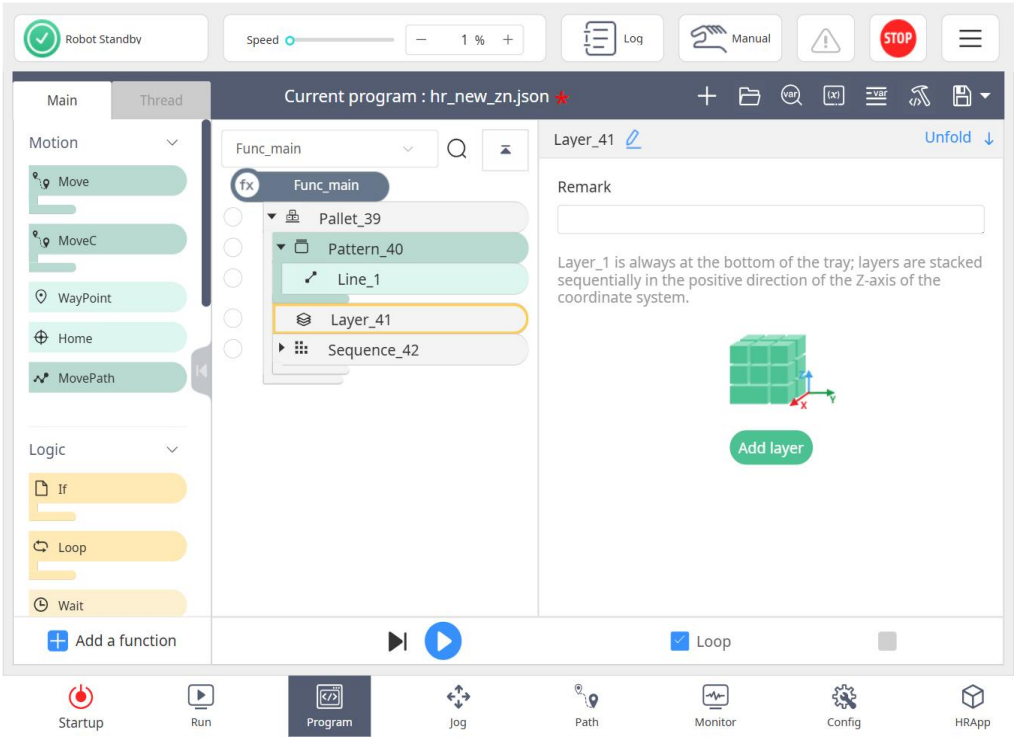
Fig. 6.6.1.1.2 Interval Count



6.6.1.2 Layer

In the "Layer" settings, you can specify the number of pallet layers and define the shape for each individual layer.

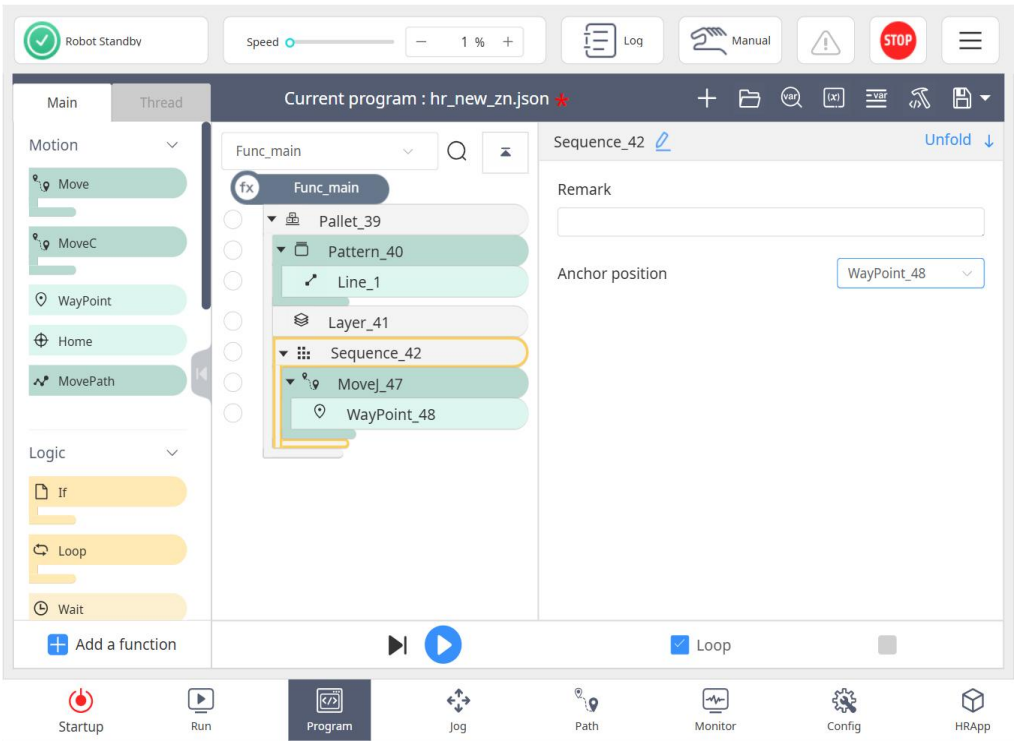
Fig. 6.6.1.2.1 Number of Pallet Layer



6.6.1.3 Sequence

To implement palletizing/de-palletizing, place 'WayPoint' waypoints under 'Sequence'. Set three waypoints as follows: WayPoint_1 right above the first object to palletize, WayPoint_2 as the palletizing anchor point, and WayPoint_3 as the lifting distance after palletizing completion. To set the pallet anchor point, simply select 'WayPoint2' from the drop-down menu.

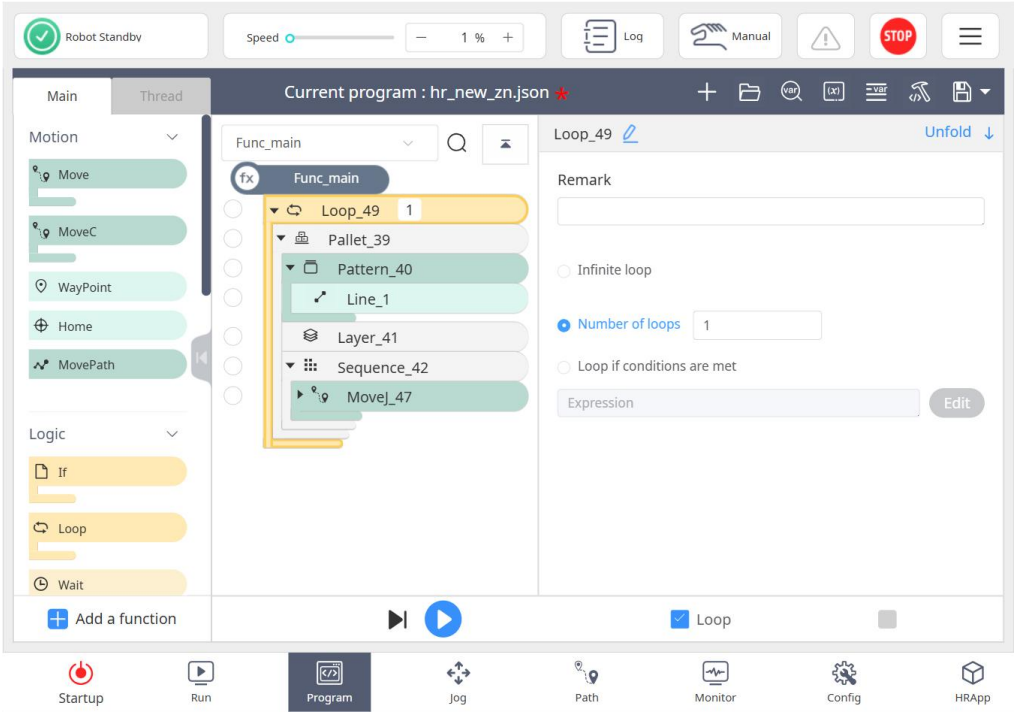
Fig. 6.6.1.3.1 Anchor Position Selection



To implement palletizing/de-palletizing, add a Loop command in the sequence

based on the total number, as shown in the figure below.

Fig. 6.6.1.3.2 Loop Selection



Note: The anchor point is a special "WayPoint". A "WayPoint" under the "Sequence" command is required to select the anchor point. During execution, the robot will move through these WayPoints at each pallet interval, forming a short path. The function of the anchor point is as follows: when a "WayPoint" is designated as an anchor point, its position will automatically align to each palletizing/de-palletizing point on the pallet. Since the path is fixed, the other WayPoints under the "Sequence" command will automatically adjust accordingly. They will undergo relative translation, and the entire path will also shift relatively.

6.6.2 Tracking

The Tracking function is used to adjust robot motion so that it runs synchronously with the conveyor, enabling accurate tracking.

Conveyor tracking consists of two parts: Basic Function and Advanced Function.

Basic Function: basic conveyor tracking is implemented based on the Tracking command together with MoveL waypoint commands.

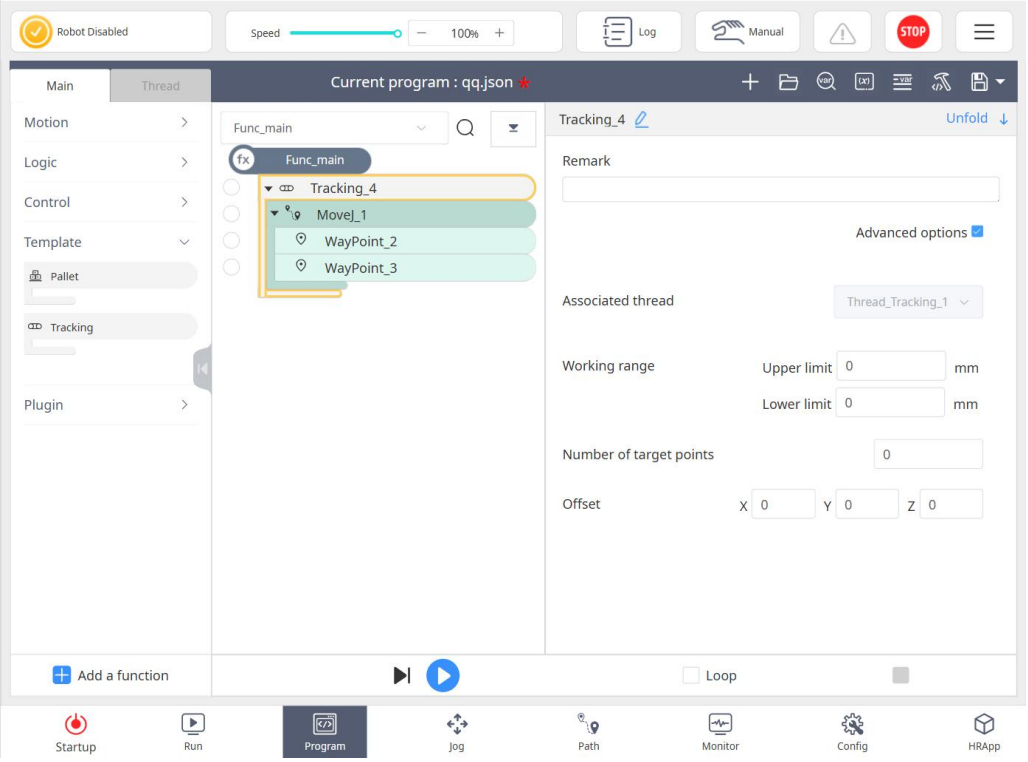
Advanced Function: on top of the basic function, external devices such as a vision system are used to provide target positions, and other external devices can be used to trigger the tracking signal, thereby achieving high-accuracy conveyor tracking.

6.6.2.1 Basic Conveyor Tracking Function

When the Advanced option of the Tracking command is not selected, the

command operates in Basic Function mode. In this mode, all MoveL waypoint motions within the Tracking block are executed as conveyor tracking motions.

Fig. 6.6.2.1.1 Basic Tracking Function



6.6.2.2 Advanced Conveyor Tracking Function

The Advanced Function consists of two parts:

- the main Tracking program with the Advanced option selected
- the automatically generated associated thread

Within the defined working range, the main Tracking program performs the tracking motion. The specific target position to be tracked does not need to be taught manually; instead, it is passed in by the associated thread.

The commonly used parameters are described below:

Associated Thread:
This item cannot be selected manually. When the Advanced option is enabled, the bound thread is automatically selected. This thread is used to receive tracking target positions through TCP communication.

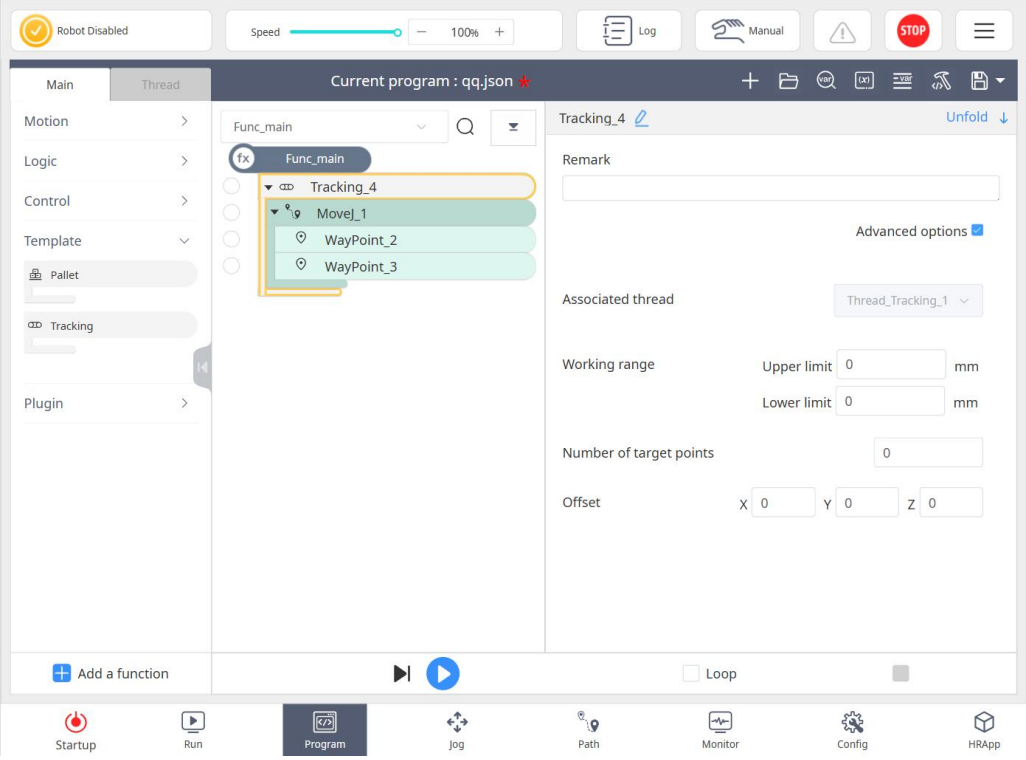
Working Range:
The upper and lower limits at which tracking is allowed, defined relative to the origin of the user coordinate system.
The configured upper and lower limits must not exceed the robot motion range.
The upper limit must not be greater than the lower limit.
In addition, a sufficient margin must be reserved between the lower limit and the robot motion limit. This margin must be large enough to ensure that, even if the workpiece is close to the lower limit when tracking starts, the robot can still

complete the entire process before reaching its motion limit.

Number of Target Points:
The number of target points to be tracked for a single signal trigger. After this value is set, the corresponding number of TargetPoint items is generated.

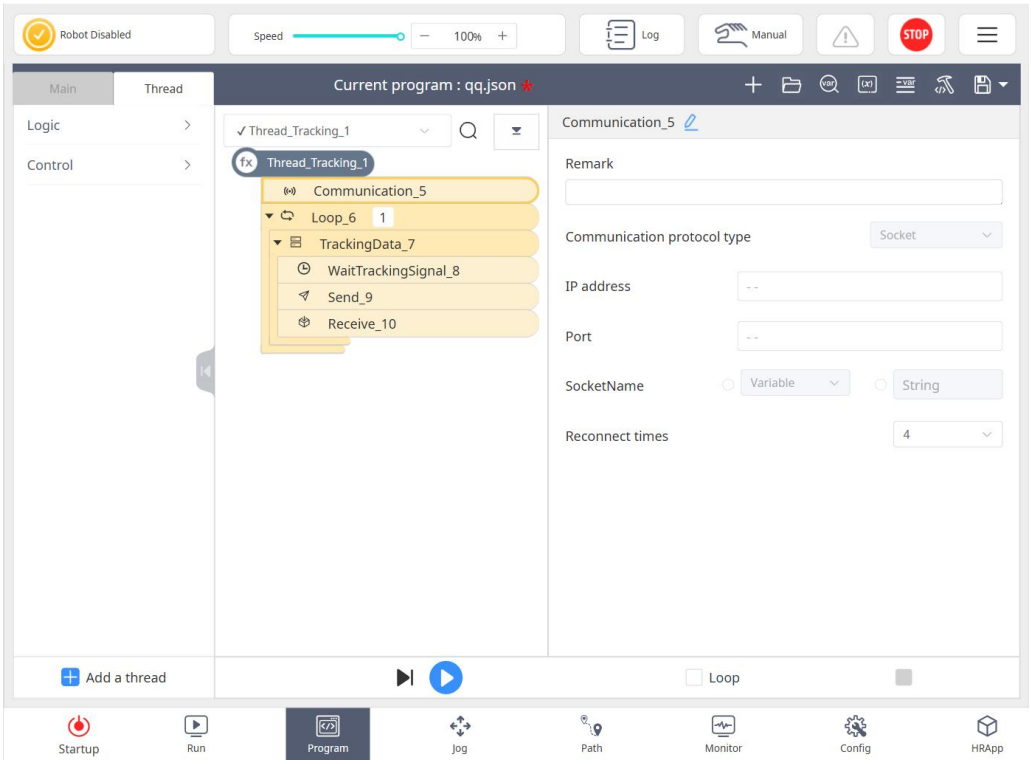
Offset:
The final tracking target position is affected by the offset. Offset values Δ are added in the X, Y, and Z directions.

Fig. 6.6.2.2.1 Tracking Advanced Options



The associated thread used by the advanced function is a combination of multiple commands. Its purpose is to communicate with an external vision device and obtain the target positions for tracking.

Fig. 6.6.2.2.2 Tracking Thread



6.6.2.3 Set the User Coordinate System and Track the Conveyor

When the hardware conveyor has been connected properly (for specific hardware connection methods, refer to the relevant manual), the conveyor must be calibrated to ensure that the robot direction and motion remain synchronized with the conveyor.

For details on setting the user coordinate system and conveyor tracking, refer to 5.6 User Coordinate System and 5.9 Conveyor Tracking.

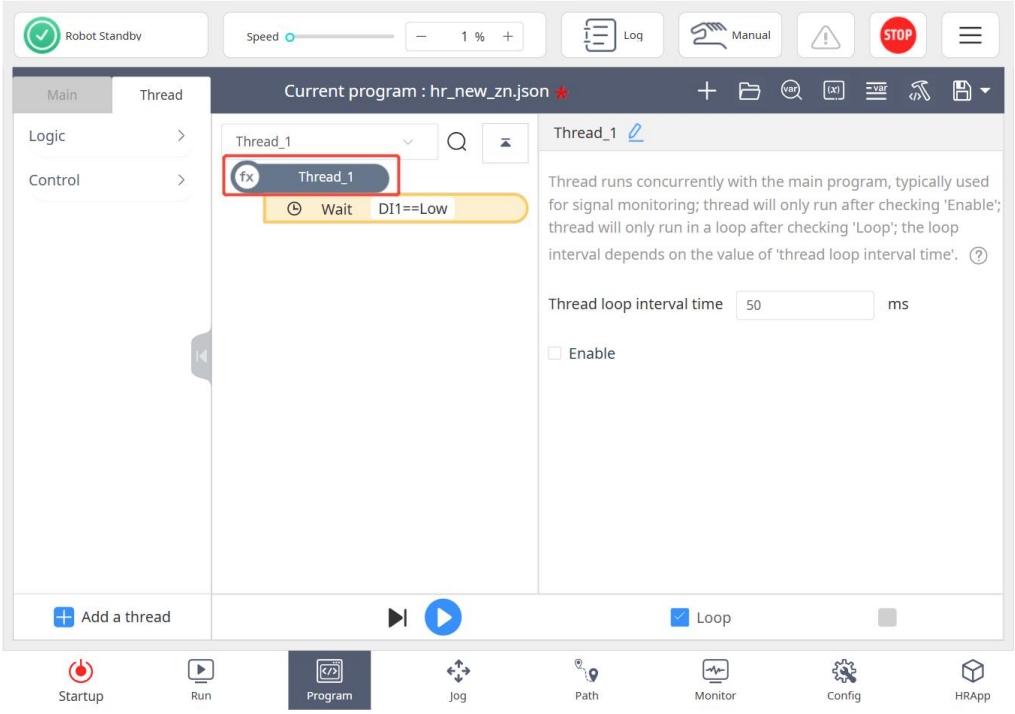
6.6.2.4 Determining the Motion Mode

After the user coordinate system has been established and conveyor tracking has been configured successfully, use the MoveL function to perform motion. Under the Tracking function, only this motion type is allowed.

After the corresponding user coordinate system is selected, the robot arm moves synchronously with the conveyor within the defined range.

6.7 Thread

Fig. 6.7.1 Thread Interface



A thread is a parallel execution unit of the robot main program and can be used for real-time monitoring of signal logic. By clicking Add a thread, users can create a new subthread.

A subthread can run cyclically at a specified interval to monitor logical signals in real time, such as I/O and Modbus states, and communicate with the main thread in real time. The system supports up to five subthreads running simultaneously to enable parallel multitask processing.

The commonly used parameters are described below:

Thread Loop Interval:
If the value is set to 0 ms, the thread runs continuously in parallel with the main thread in a loop.
The default value is 10 ms, which means one loop cycle is triggered every 10 ms.

Enable:
When selected, the thread is enabled and takes effect.
When not selected, the thread is inactive.
Enabled threads are marked with a "√" symbol in the thread drop-down list.

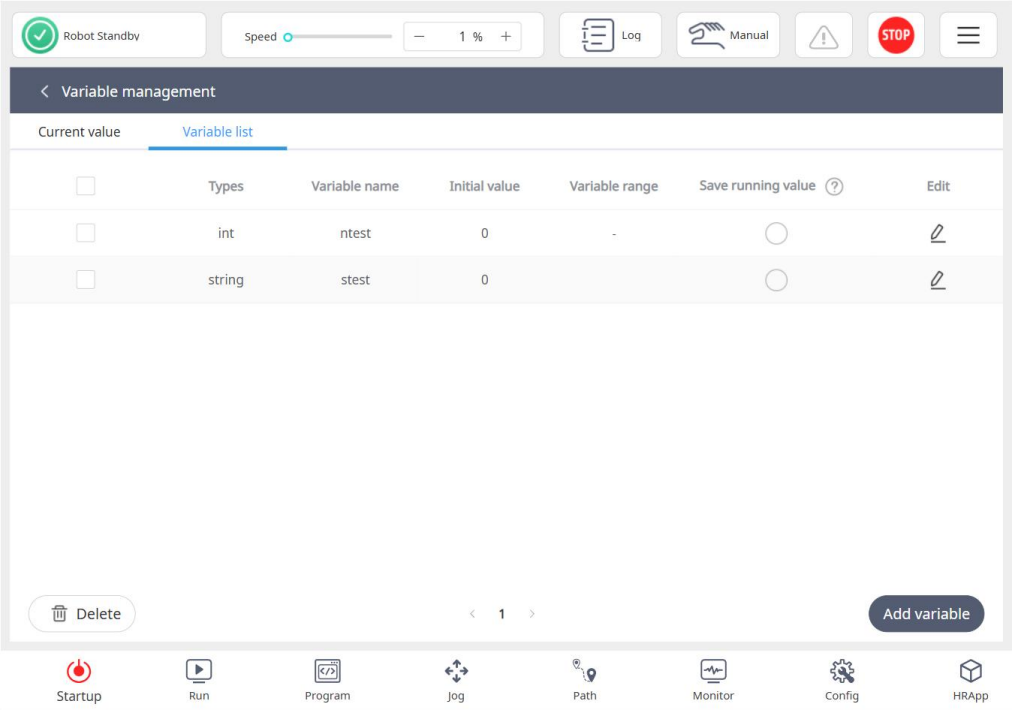
Note:
- Motion commands cannot be added to a subthread.
- After the main thread is stopped, all subthreads are terminated.

6.8 Variable

6.8.1 Variable List

Declare the variables as global variables in the variable list.

Fig. 6.8.1.1 Variable List



The variables can be edited, and there are no restrictions on the number of variables that can be configured.

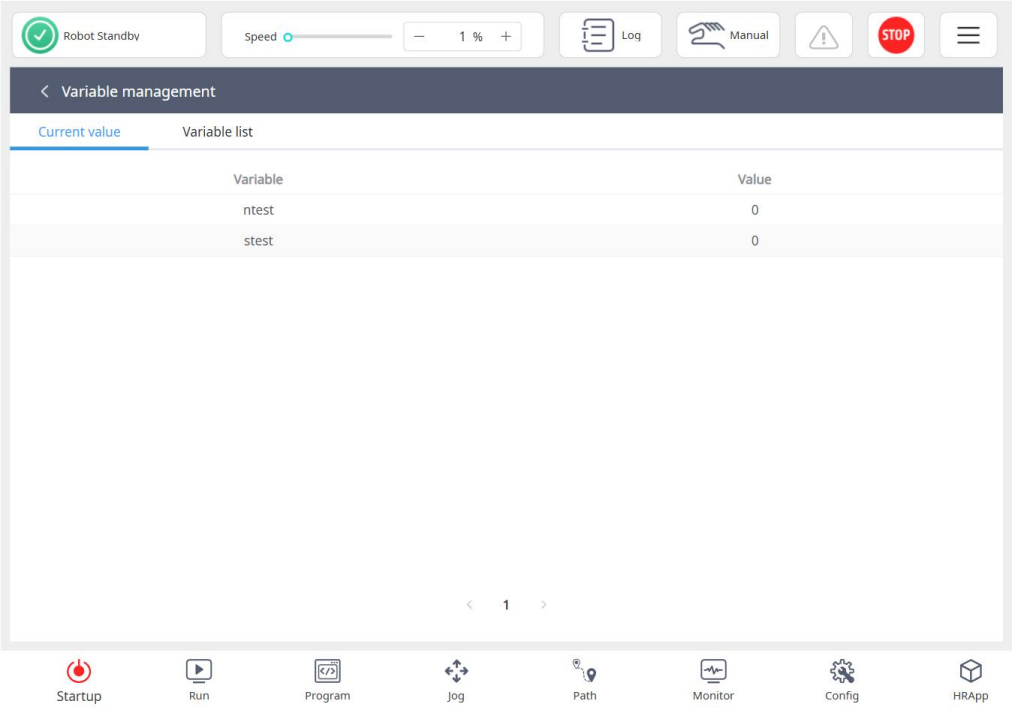
- **Add Variable:** Used to create a new variable;
- **Edit:"** **"** is used to modify the selected variables, including renaming, changing type, assigning an initial value, and configuring retention of the value from the previous execution cycle;
- **Delete:** Used to delete the selected variables. The user can update variable values during program execution by assigning values to variables through the SetValue command;
- Variable Range: Defines additional constraints on variable values by manually setting upper and lower bounds.
- Variable type and description:
  - **int integer variable:** The range is unlimited, and custom range is supported.
  - **double double-precision floating-point value:** Floating-point number (decimal).
  - **string:** Store a sequence of characters.
  - **joint_position:** Consists of six independent floating-point values, separated by commas.
  - **tool_position:** Consists of six independent floating-point values, separated by commas.
- **Variable Range:** Within the scope of the data type to which a variable belongs, its range can be further limited to meet specific needs.

– **Whether to Save the Last Runtime Value:**If the Save option is selected, the variable value is recorded in real time and stored in the configuration file. When the program runs again, the saved value is used instead of the initial value.

6.8.2 Current Variable Value

Displays variables used by the user during program editing. During script debugging, the current values of these variables are updated in real-time based on program execution, reflecting the latest assignments. Users can concurrently validate in real-time whether variables are correctly assigned while debugging.

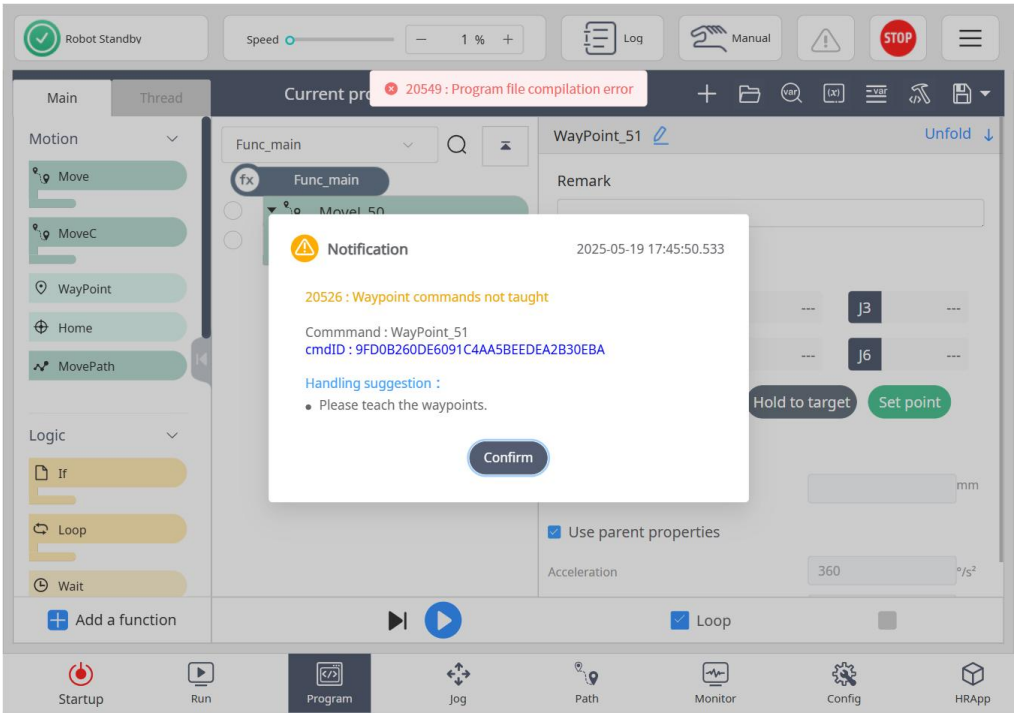
Fig. 6.8.2.1 Current Variable Value



6.9 Debugging and Running

After programming, debugging is required to verify if the program runs successfully. If it fails, a corresponding pop-up window will appear to prompt the user.

Fig. 6.9.1 Program Debugging and Running



6.9.1 Running

After completing program editing, select the function to debug from the drop-down menu. Debugging starts from Func_main.

If robot's end-effector is at the program's starting position when you click the "Run  "button, the program will start running directly. If you click the "Run  "button, step-by-step debugging of the program will start.

Note 1: The program cannot be executed successfully if there are commands with unset parameters.

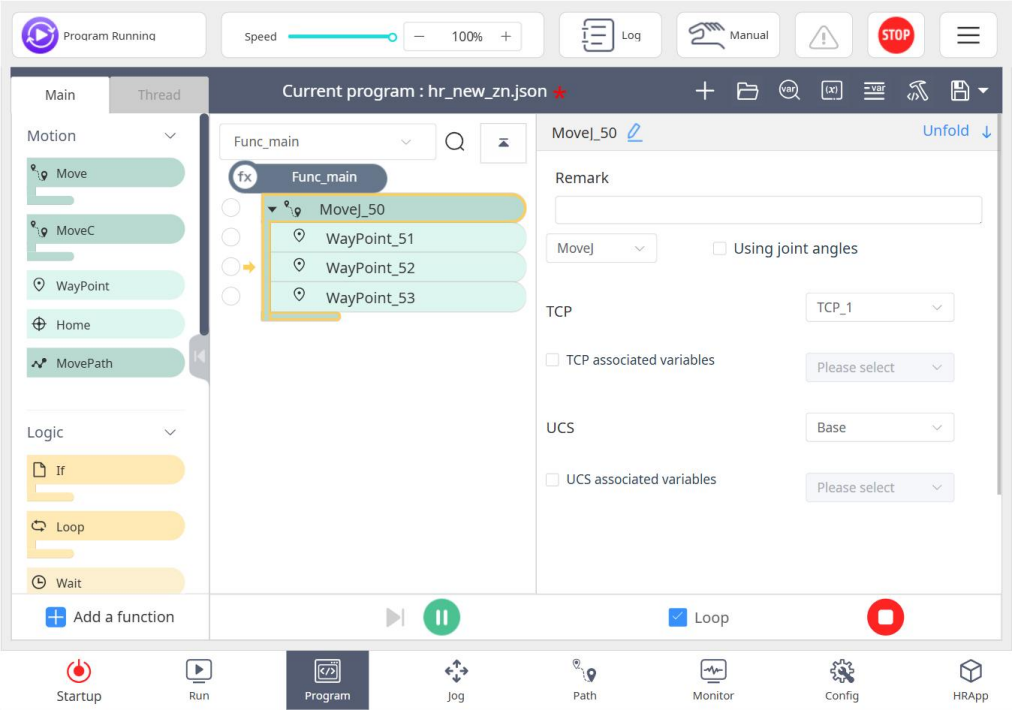
When the user clicks"Run", if there are unset commands in the program, a pop-up window will appear and warn, which prompts: "Program file compilation error"; meanwhile, the system popup will remind the specific contents and solutions of the command in issue, so that users can quickly identify and correct errors.

Note 2: The program may fail to execute due to parameter errors.

6.9.2 Executing

A yellow arrow will appear before the current command to indicate its progress.

Fig. 6.9.2.1 Program Execution



- **Pause:** After the program starts, click the "Pause" button to interrupt execution. The button then switches to "Run"; clicking it again continues the program.

- **Stop:** Once the program is running, click the "Stop" button to terminate program execution.

Note: When a program encounters an error during execution, a system pop-up window will show the error location and specific cause. Users can adjust parameter settings according to these prompts and re-execute the program for debugging.

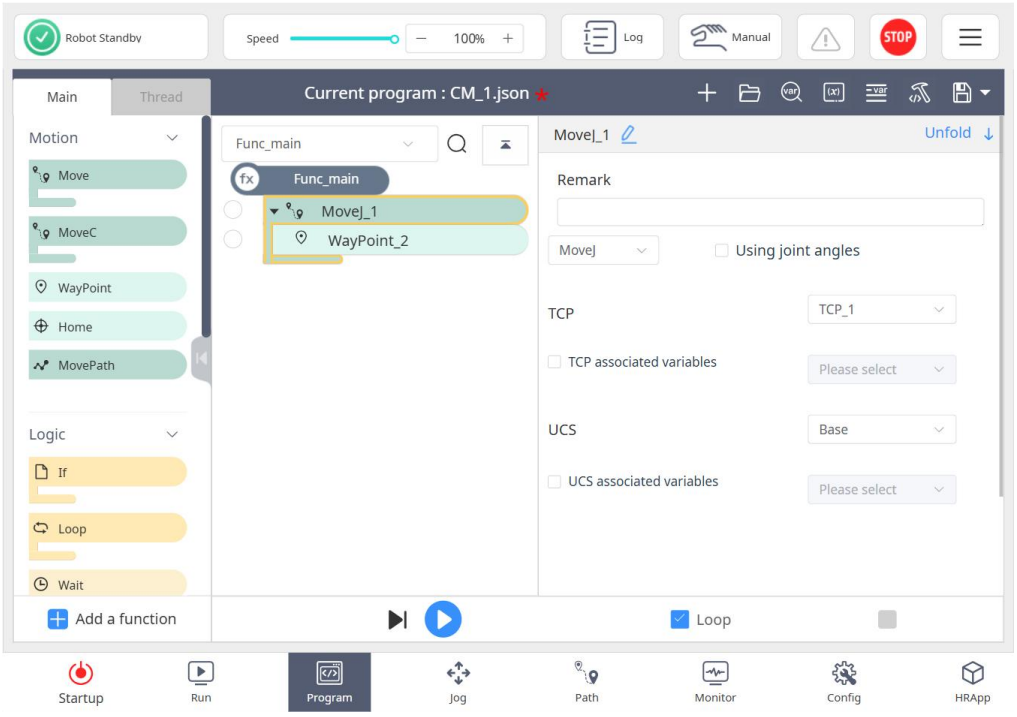
6.10 Program Example

During robot operation, motion commands are the most basic and frequently-used commands. Below is a simple pick-and-place program. (The current end-effector is a gripper.)

- When the robot is enabled, create a program in the programming interface: File → New → Name the Program → Confirm Creation.

- Under the Func_main function, drag a Move command from the command menu to the function node. A WayPoint_2 is automatically added under the Move node. Change the motion type of Move to MoveL. Select a suitable TCP for the gripper (end-effector) (pre-configured in settings). The coordinate system defaults to Base. Set appropriate acceleration and tool speed for the waypoint.

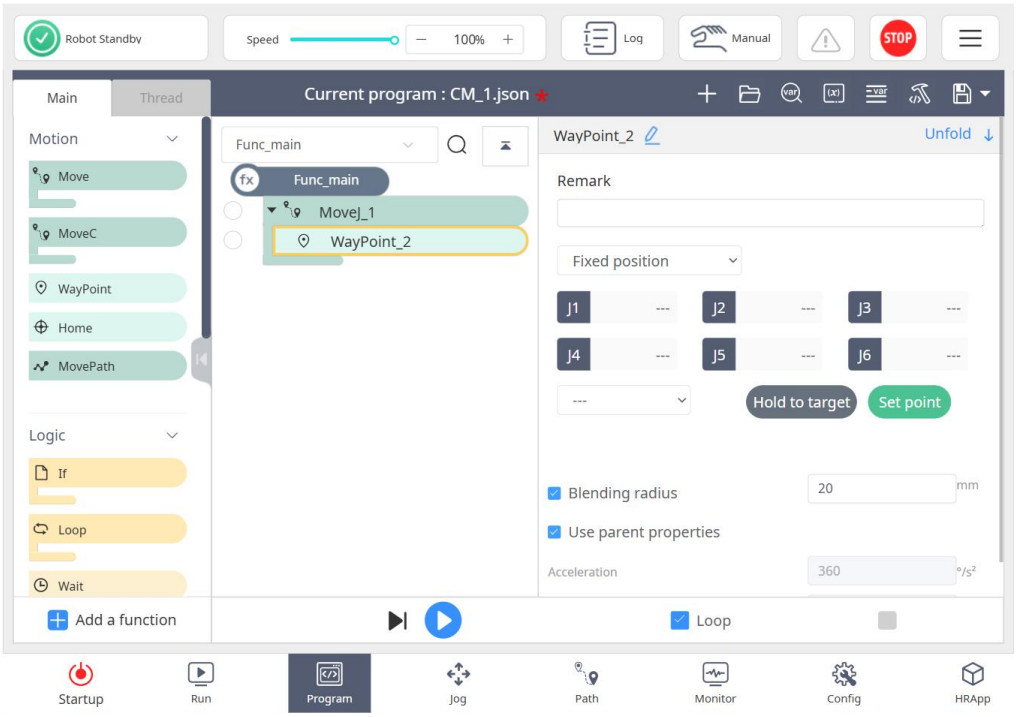
Fig. 6.10.1 Setting Status



– Teach pendant WayPoint_2. If the desired point is already in the point list, select it directly from the drop-down menu in the current interface. WayPoint_2 serves as the robot's starting point.

– If you do not want the robot to stop between WayPoint_2 and WayPoint_3, set and check the blending radius, ensuring it does not exceed the distance between the two points.

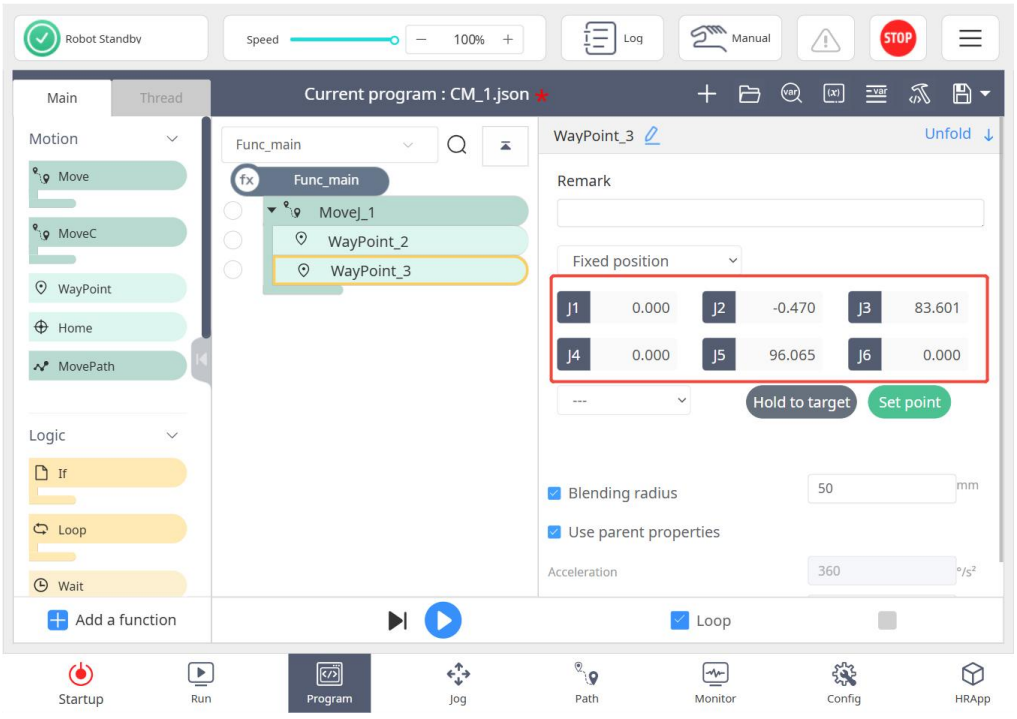
Fig. 6.10.2 Blending Radius Setting



– Drag and drop a waypoint from the command menu to under WayPoint_2 to

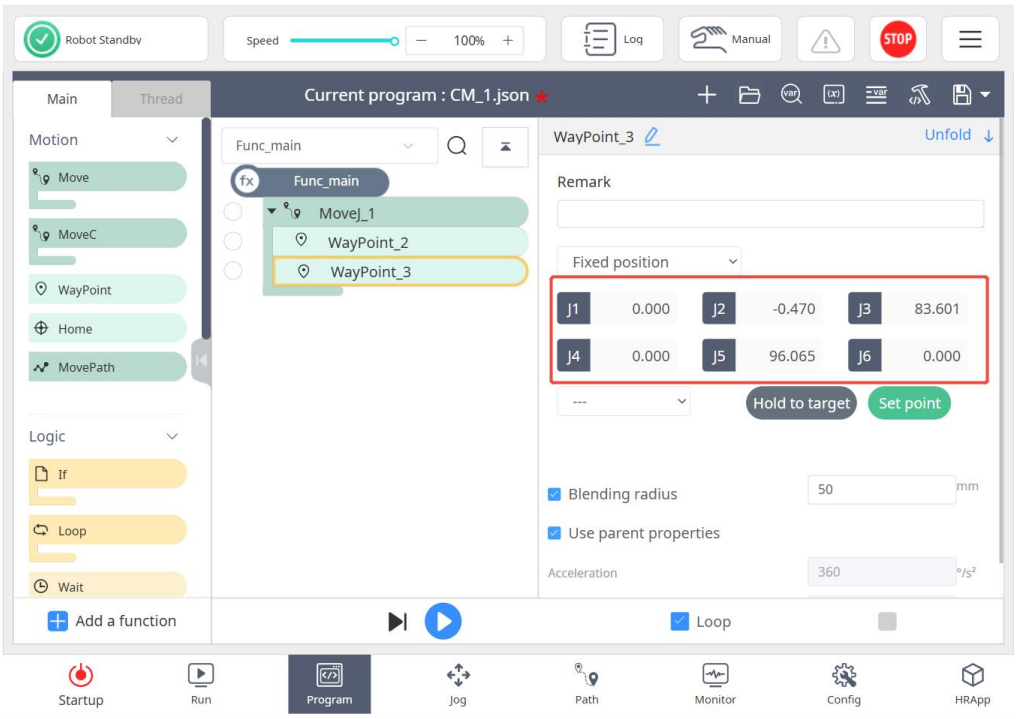
generate WayPoint_3. Teach WayPoint_3, which is above the material's position. In the MoveL mode, the robot arm moves from WayPoint_2 to WayPoint_3 to prepare for material grasping.

Fig. 6.10.3 WayPoint3 Setting



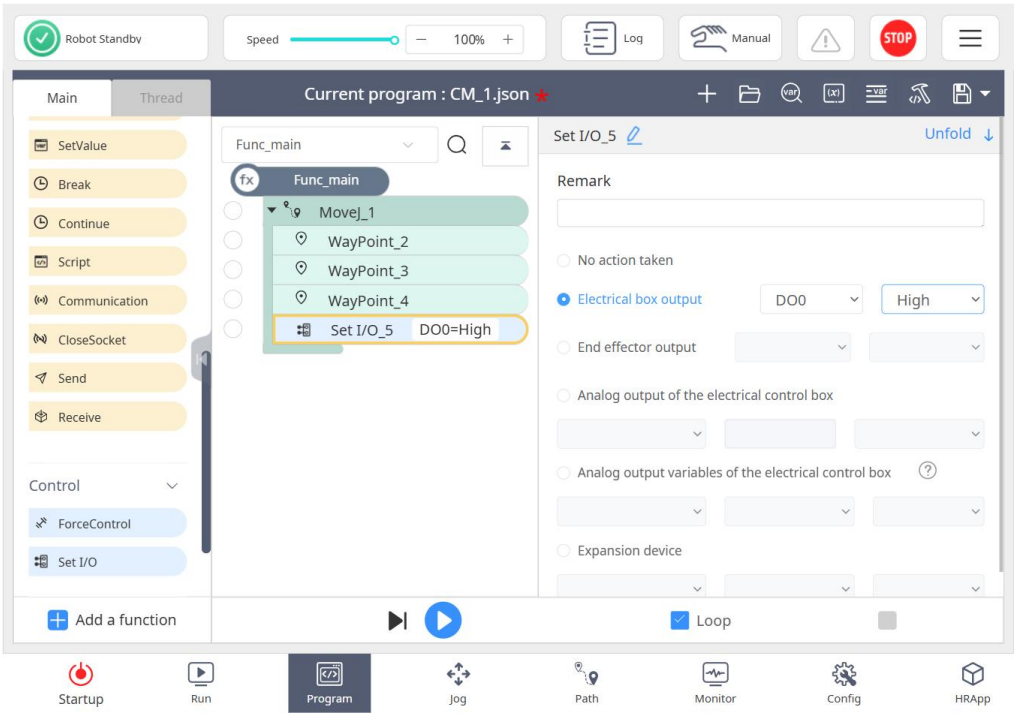
– Drag and drop a waypoint from the command menu to under WayPoint_3 to automatically create WayPoint_4. Teach WayPoint_4, which is the target point above the material, positioning the gripper above the material in preparation for grasping.

Fig. 6.10.4 WayPoint4 Setting



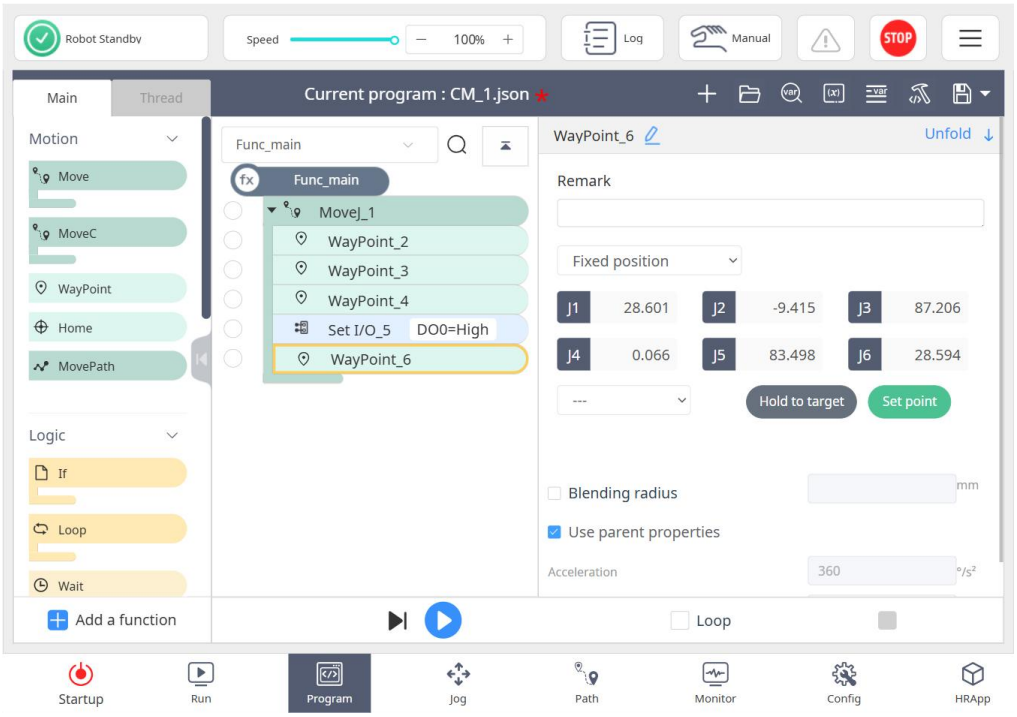
– Drag and drop Set I/O from the command menu to under WayPoint_4 and set the electrical control box output as DO0 = High, and close the gripper to pick up the material.

Fig. 6.10.5 Set I/O



– Drag and drop a waypoint from the command menu to under SetI/O to automatically generate WayPoint_6. Teach Waypoint_6, which is a transitional safe position. The robot arm picks up the material and lifts it to the safe position.

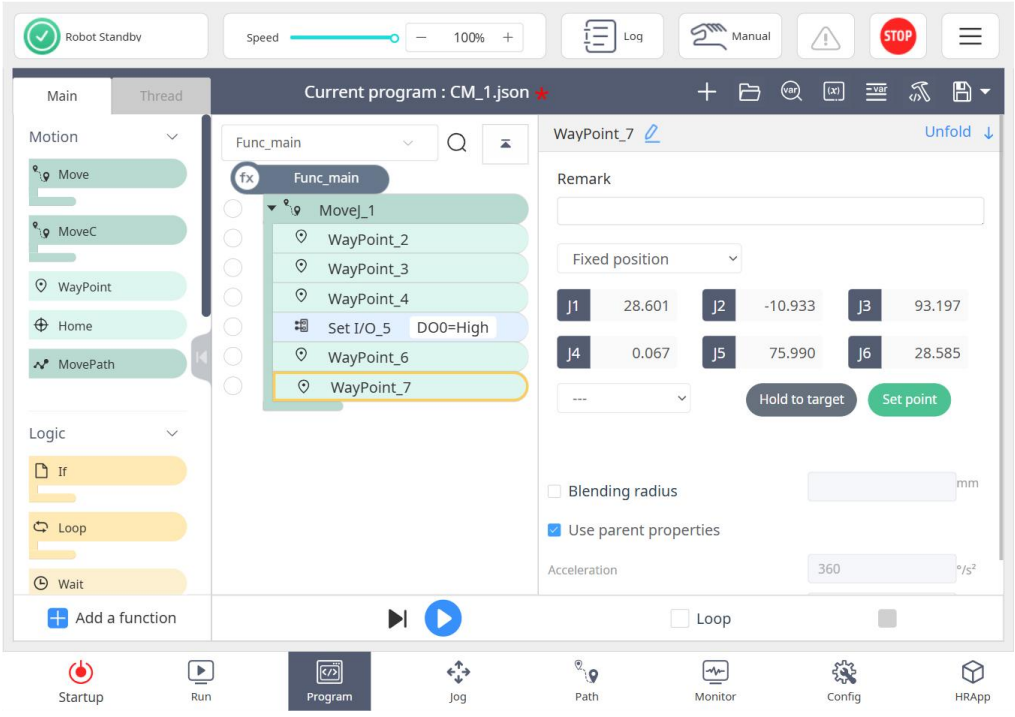
Fig. 6.10.6 Safe Position of Transition



– Drag and drop a waypoint from the command menu to under WayPoint_6 to

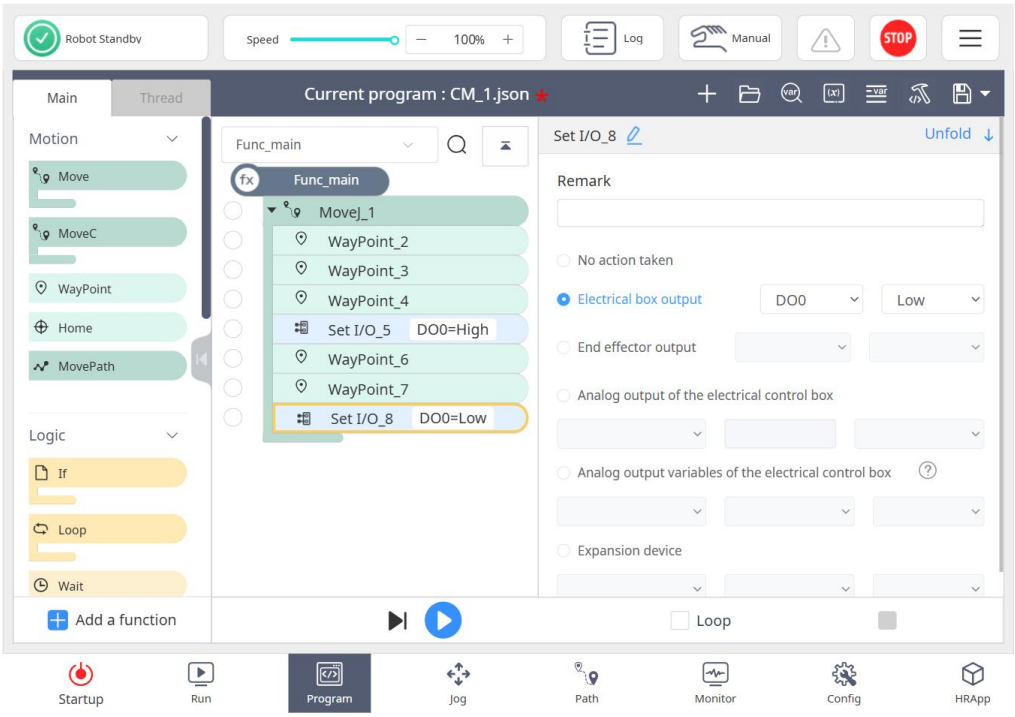
automatically generate WayPoint_7. Teach Waypoint_7, which is the final workpiece placement position. In MoveL mode, the robot moves from WayPoint_6 to WayPoint_7 to reach the final placement position.

Fig. 6.10.7 Set Final Position



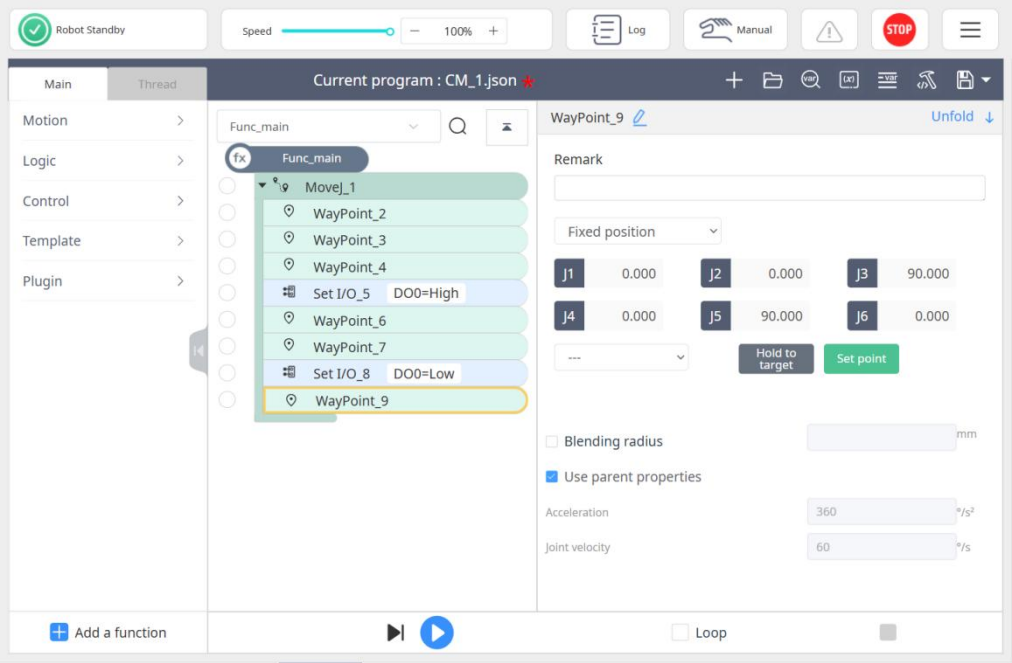
– Drag and drop a SetI/O from the control command in the command menu to under WayPoint_7. Set the electrical control box output DO0 = Low (since the gripper is connected to I/O DO0). The gripper opens to place the material.

Fig. 6.10.8 Re-Set I/O



– The robot arm returns to its safe home position.

Fig. 6.10.9 Return to Safe Points



This program performs a pick-and-place operation with intermediate safe points to avoid obstacles during MoveL linear motion. After editing, click "Run" to execute it. Selecting the loop option enables continuous program execution.

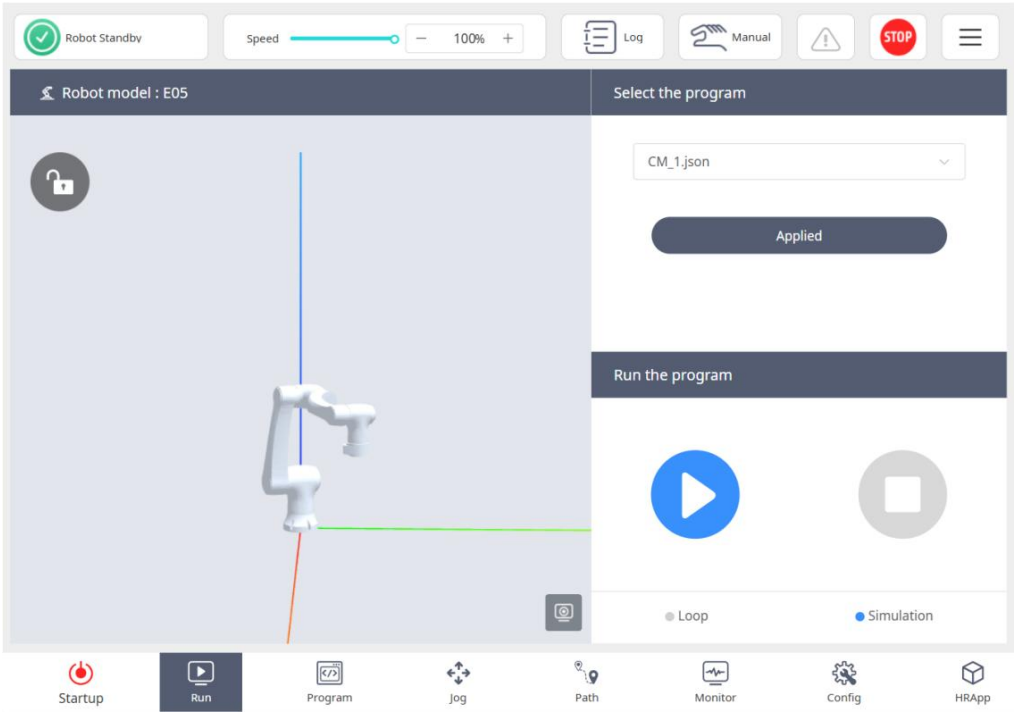
Running

This chapter contains the following topics:

– Initializing Robot
– Selecting the Program
– Running Status
– Page Introduction

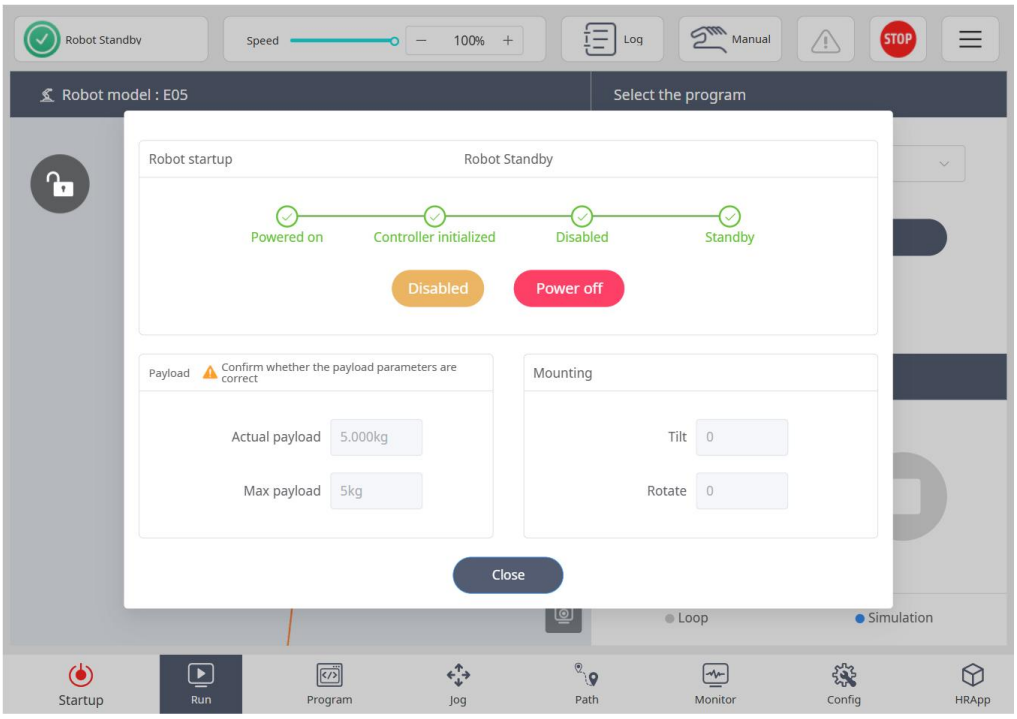
The run interface allows users to directly execute pre-saved programs, enabling quick robot operation for general operators.

Fig. 7.1 Running Interface



7.1 Initializing Robot

Fig. 7.1.1 Initializing Robot

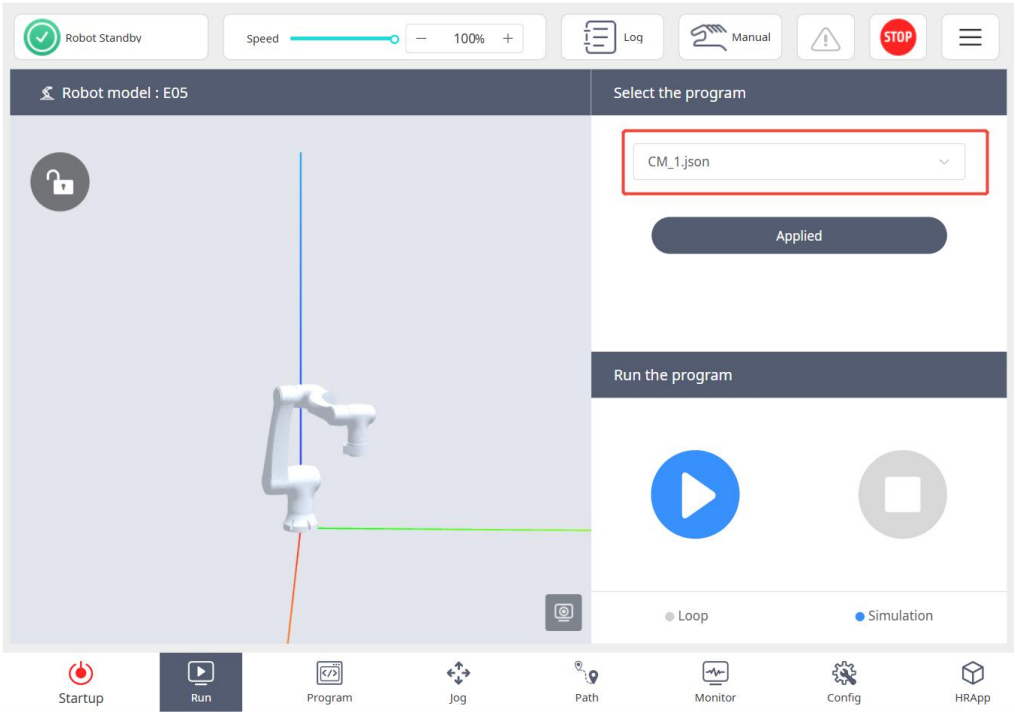


Robot Power Up and Enabling Operations

Please refer to "2.2.1 Initializing Robot" for details.

7.2 Selecting the Program

Fig. 7.2.1 Program Switch Interface



The drop-down menu lists all saved programs in the software. Users can select a program from it and apply it to the robot.

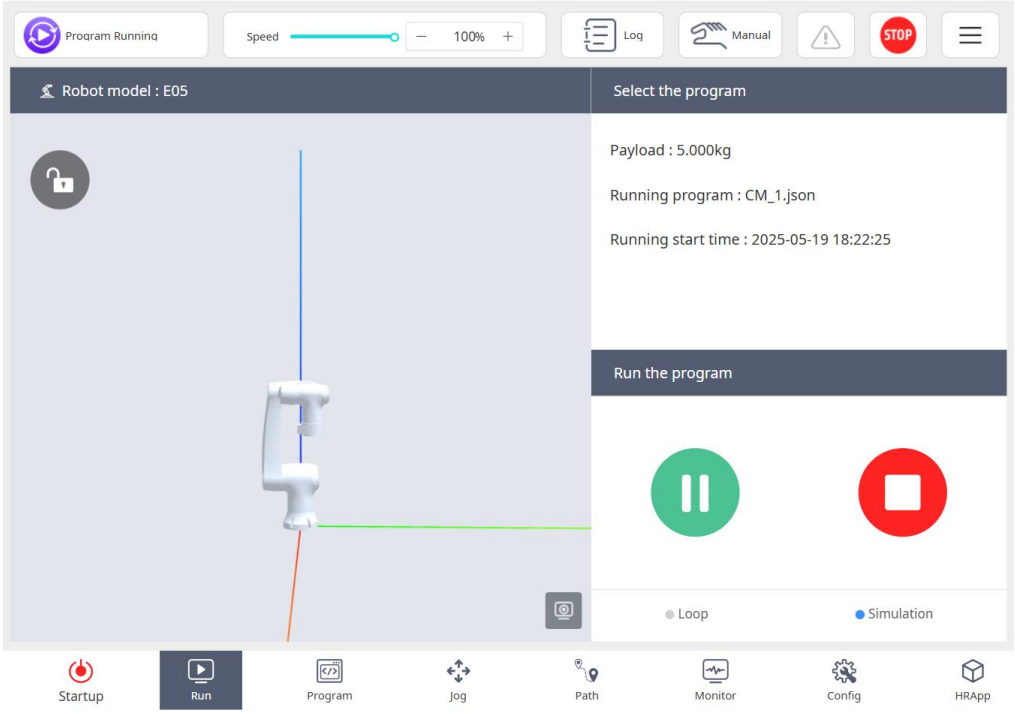
- **Run:** After the program is loaded, clicking the "Run " button will start the program directly if the robot's current position matches the program's starting point. Otherwise, the robot will first move to the program's starting position and then start execution.
- **Pause:** After the program starts, clicking the "Pause " button to interrupt the program execution.
- **Stop:** Once the program is running, clicking the "Stop " button to terminate program execution.

Note: If the program includes enable commands, the users cannot run it when the robot is disabled. Manually enable the robot and wait for the robot arm to return to the start position before running the program.

7.3 Running Status

The Running Status section displays the robot status during operation, allowing users to view information such as the name of the script file currently being executed, the payload, the start time, and the robot operating duration.

Fig. 7.3.1 Run the Program

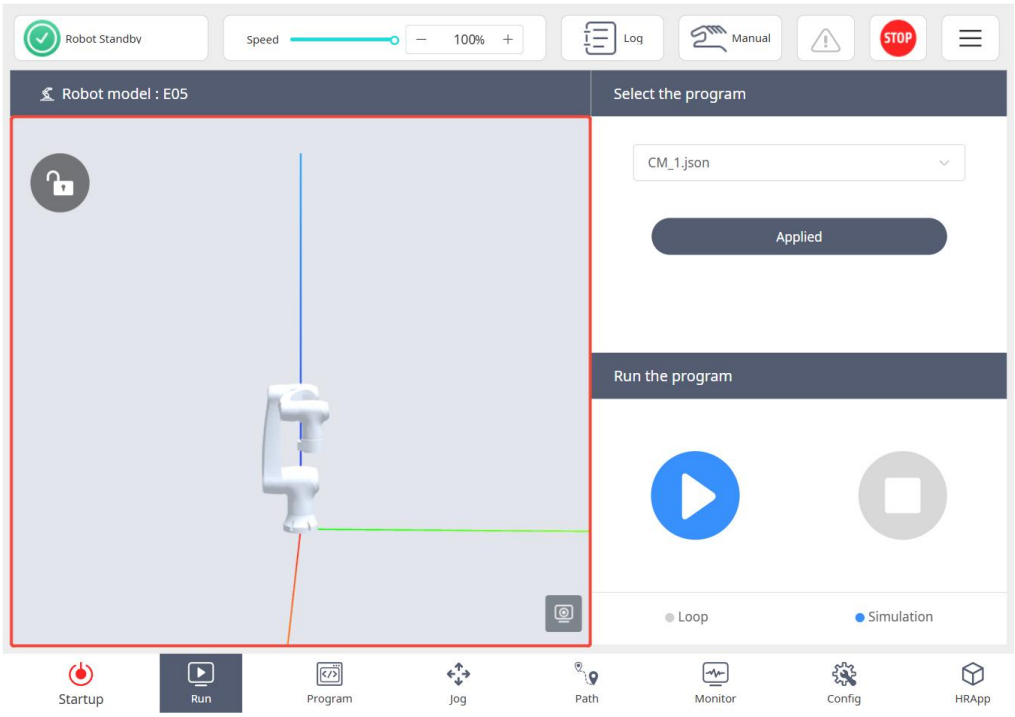


7.4 Page Introduction

7.4.1Robot 3D Model

Synchronized with the robot arm.

Fig. 7.4.1 Robot 3D Model



7.4.2Recovery

After clicking , the view will reset to the default perspective of the 3D graphic.

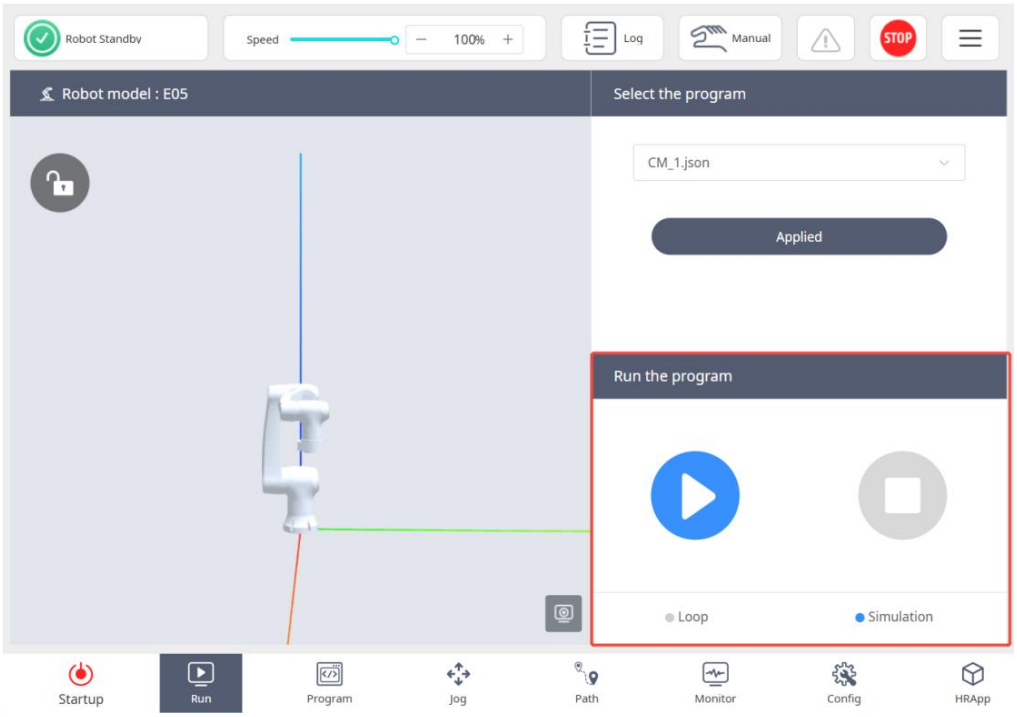
7.4.3 Locking

Once safety-locked, the interface is inoperable. Re-entry requires a password, which effectively prevents unintended operations during robot runtime.

7.4.4 Operation Area

The users can adjust the robot arm's speed, control script execution (run, pause, stop), and switch between simulated and real robot arm.

Fig. 7.4.3.1 Bottom Operation Area



- **Simulation led on:** Indicates connection to actual robot hardware. During teaching, ensure the robot and its surroundings are positioned safely;
- **Simulation led off:** Indicates a simulated robot connection. It mirrors the real robot in functionality but without hardware-related safety concerns, enabling faster and safer teaching;
- **Switching Real/Simulated Robot Arm:** Users can only switch between these modes by selecting the desired option when the robot is powered off. Switching is disabled in other states.

Path

This chapter contains the following topics:

– Page Introduction
– New Path
– Path List

On the Path page, users can create, record, and save robot arm motion paths for repeated use in programming.

8.1 Page Introduction

Fig. 8.1.1 Path Recording Interface

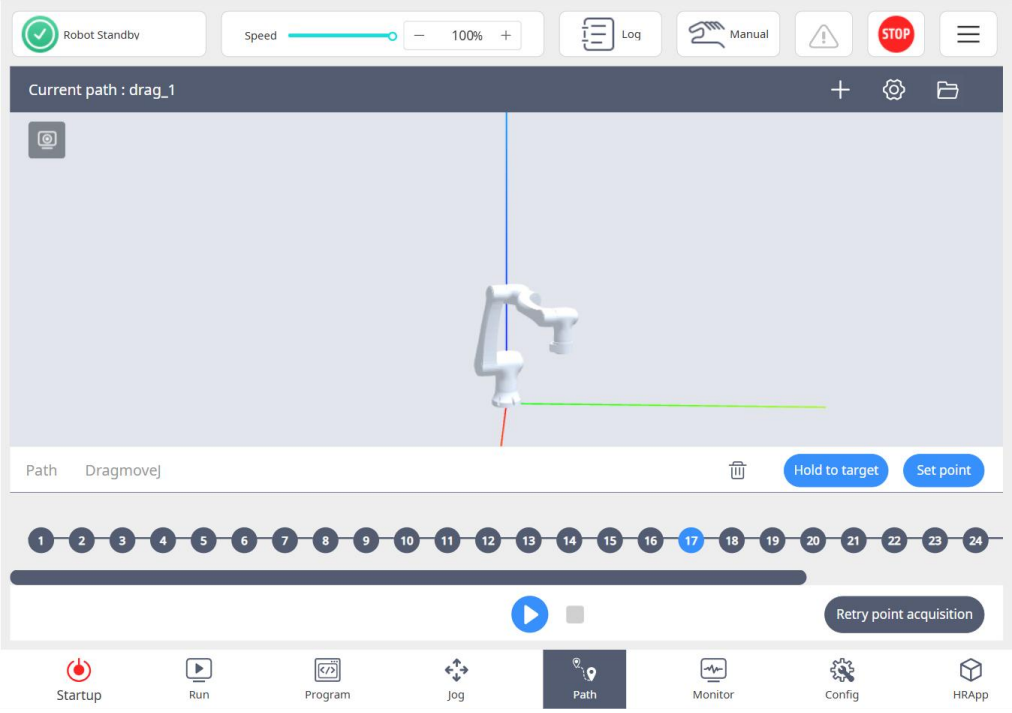
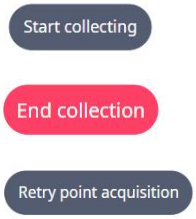


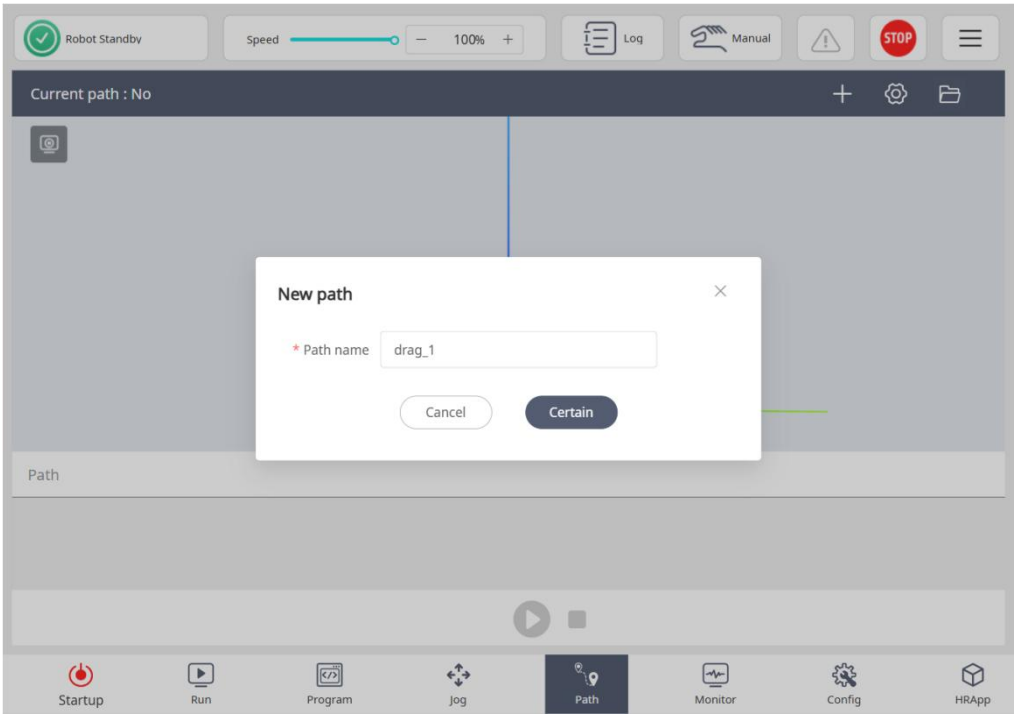
Table 8.1.1 Path Operation Interface Description

Current path : drag_1	Custom Name for the Current Path: When creating a new path, users can customize its name for easy identification and management;
	Restore: Restore the default perspective of the 3D view;
	Create Path: Click to initiate the creation of a new path.
	Path Collecting Settings: Configure the acquisition mode, blending radius, and speed ratio here;
	Path Files: Displays a list of created path files;
	Delete: Remove the currently selected path waypoint;
Hold to target	Hold to Target: Press and hold to move the robot arm to the currently selected path waypoint;
Set point	Set Point: Re-teach the currently selected path waypoint. Clicking this option navigates to the Jog interface.
	Path Waypoint: These are the waypoints collected for the current path, where the blue color indicates the currently selected waypoint. Operations can be performed on the

	selected waypoint;
	Start Collecting: After creating a new path, click this button to begin collecting waypoints. End Collection: During the acquisition process, click this button to stop collecting waypoints. Retry Point Acquisition: After completing the acquisition, this button allows you to recollect all waypoints for the current path.
	Run: Run the path after acquisition;
	Stop: Stop running the current path;
	The Current Path Operation Mode: The trajectory will be executed using the MoveJ motion command. This mode cannot be altered;

8.2 New Path

Fig.8.2.1 New Path

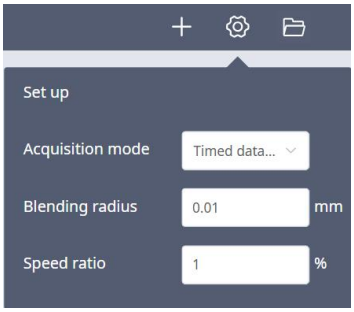


– After clicking the button , a pop-up window for creating a new path will appear on the page. The default path name is "drag_N", and users can modify "N". After creating a new path, the page will display the name of the current path.



- After creation is complete, users can configure the path acquisition. Clicking on the button  will bring up the settings window.

Fig.8.2.2 Trajectory Capture Settings



- In the window, users can configure settings such as the acquisition mode, blending radius and the speed ratio.

  - **Acquisition Mode:** There are two modes, "Timed Data Acquisition" and "Specific Point Data Acquisition". In Timed Data Acquisition, the system collects robot waypoints every 200 ms by default. In Specific Point Data Acquisition, it collects waypoints based on the robot arm's real-time position and orientation.

  - **Blending Radius:** Used to define the smooth transition area between points. For detailed information, please refer to the Blending Radius section of 6.3.2 Waypoint;

  - **Speed Ratio:** The operating speed of the current path and is independent of the speed bar at the upper left corner;

- After completing the settings, click the "Start collection" button to begin collecting path waypoints.

  Acquisition Mode 1: Timed Data Acquisition. The system collects the current position every 200 ms. If there is a certain distance from the last waypoint, the current position will be saved as the path waypoint.

  Acquisition Mode 2: Specific Point Data Acquisition. When the robot arm moves to the point to be collected by any method, click the "Add point" button or press the "Point" button on the end effector to collect the current position as a path waypoint.

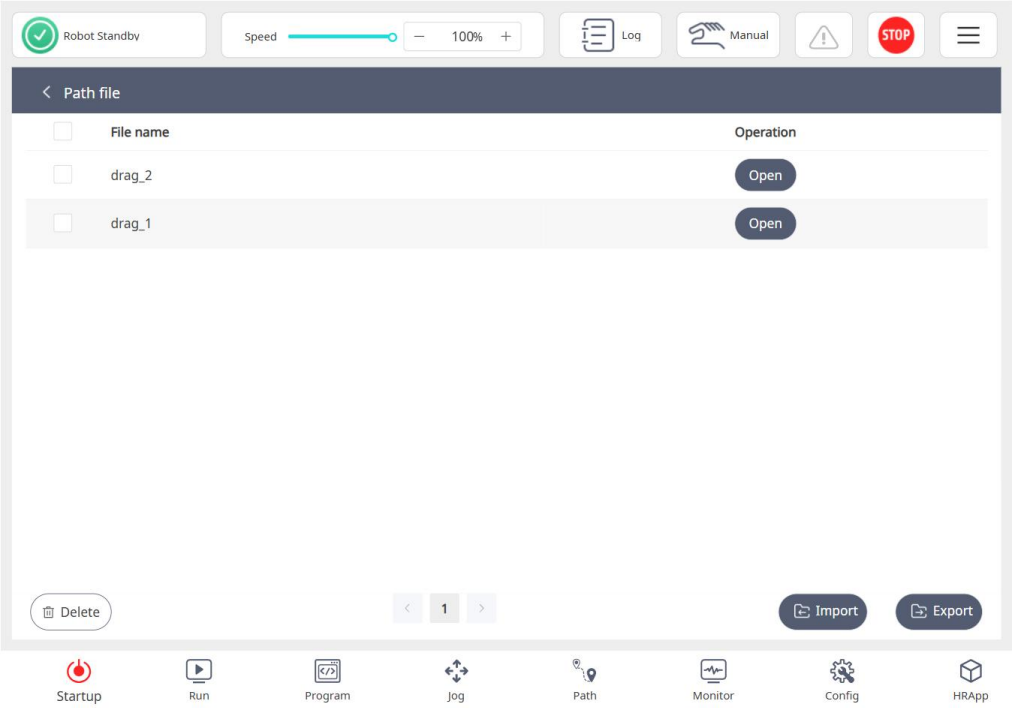
  After completing the waypoint collection, click the "End collection"button. A prompt "Collection completed" will appear. The system will then calculate and verify whether the path can operate normally. If it cannot, users can click the "Set points" button or the "Retry point acquisition" button to reset and collect the waypoints until the path is correct and it can operate properly..

- After the collection is completed, the path is automatically saved to the path list.

- Once the path calculation is confirmed to be normal, the users can click the "Run"button to start or stop the robot arm's motion along the current path.

8.3 Path List

Fig.8.3.1 Path List



- **Path Name:** Name of the saved robot path file;
- **Open:** Click to open the current path file in the path page;
- **Delete:** Tick to select and delete the current path file;
- **Import:** Import path files from an external source. The imported files must match the system file format;
- **Export:** Export the currently selected path externally

.

Monitoring

This chapter contains the following topics:

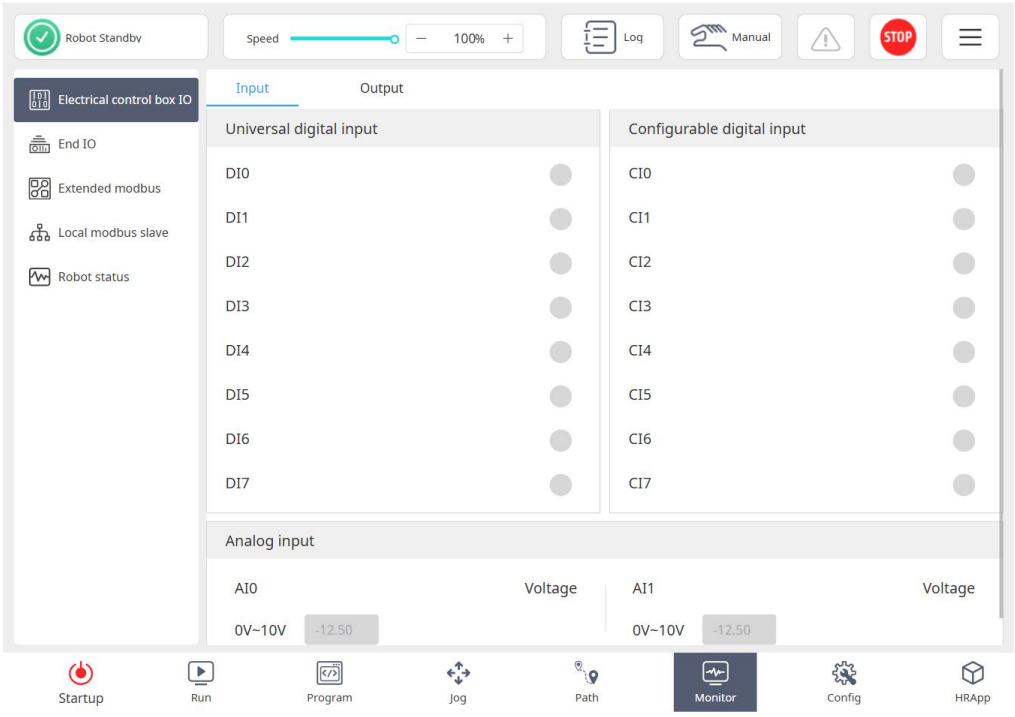
– Electrical Control Box I/O
– End I/O
– Extended Modbus
– Local Modbus Slave
– Robot Status
– Log

On this interface, users can monitor and configure the real-time I/O signals of the electrical control box. It displays the current status of the I/O signals, including the I/O status during program execution. When the program stops, all output signals will remain as they are. Note that signals changing extremely fast might not be accurately displayed due to the interface's refresh rate limits.

9.1 Electrical Control Box I/O

9.1.1 General Input and Output

Fig. 9.1.1.1 Electrical Control Box I/O Interface



- **General Input:** DI0 ~ DI7, 24V power supply. If the input is at a high level, the corresponding circle turns blue; a gray circle indicates a low level;
- **General Output:** DO0 ~ DO7, 24V power supply. Buttons correspond to the output switches ○, the output level can be set as high and the switch turns blue ◉.

9.1.2 Configurable Input and Output

- **Configurable Input:** CI0 ~ CI7, 24V power supply. If the input is at a high level, the corresponding circle turns blue, and the gray circle indicates low level;
- **Configurable Output:** CO0 ~ CO7, 24V power supply. Buttons correspond to the output switches ○, the output level can be set as high and the switch turns

blue .

- Configurable Input and Output Feature
- Please refer to the "5.3 I/O Setting" configuration instructions for more details.

9.1.3 Analog Input and Output

- **Analog Input:** 2 channels, AI0 and AI1, with a voltage range of [0-10V];
- **Analog Output:** 2 channels, AO0 and AO1, with a current range of [4-20mA] and a voltage range of [0-10V].

9.2 End I/O

The user needs to perform security verification in advance, and then click the "End Effector IO Capabilities" button to enable the functions of the buttons at the end-effector.

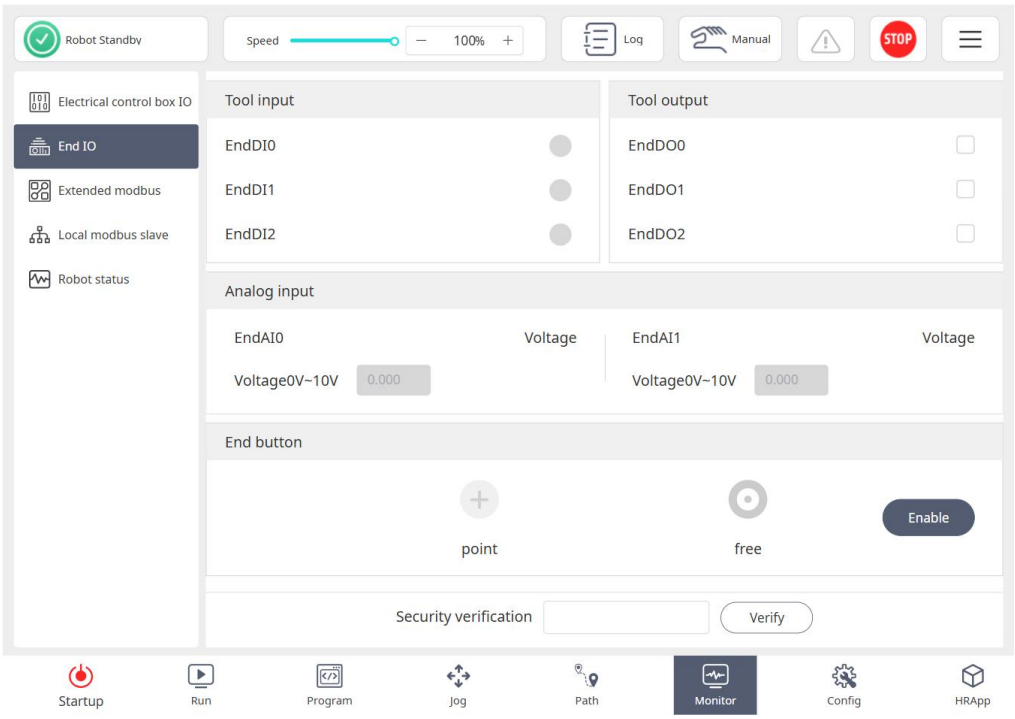
Free: Activates the basic free drive mode;

Point: For saving positions.

Note: For safety consideration, once the "Free" button at the end-effector is activated, the basic free drive mode on the "Jog" interface will not be available.

The page defaults to displaying 2 buttons at the end-effector, but the actual number varies with the end-effector type.

Fig. 9.2.1 End I/O Interface

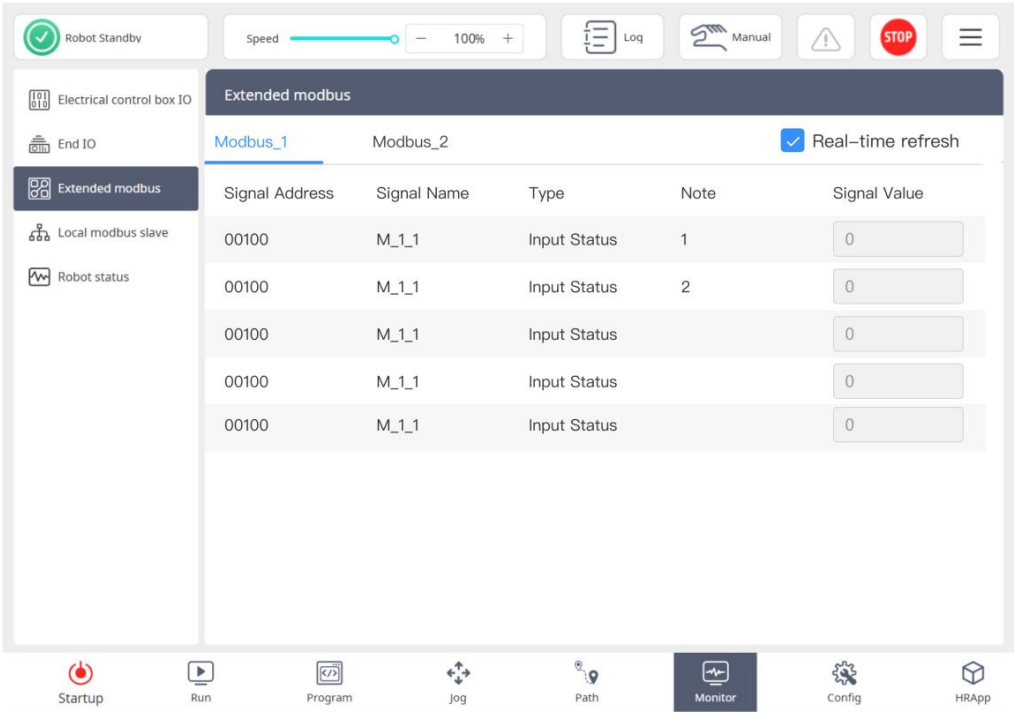


9.2.1 Universal Input, Output and Analog Input

- **Tool Input:** EndDI0 ~ EndDI2, 24V power supply. If the input is at a high level, the corresponding circle turns blue; a gray circle indicates a low level;
- **Tool Output:** EndDO0 ~ EndDO2, 24V power supply. Buttons correspond to the output switches , the output level can be set as high and the switch turns blue ;
- **Analog Input:** 2 channels, EndAI0 and EndAI1. Voltage input only, within the range of [0-10V].

9.3 Extended Modbus

Fig.9.3.1 Extended Modbus



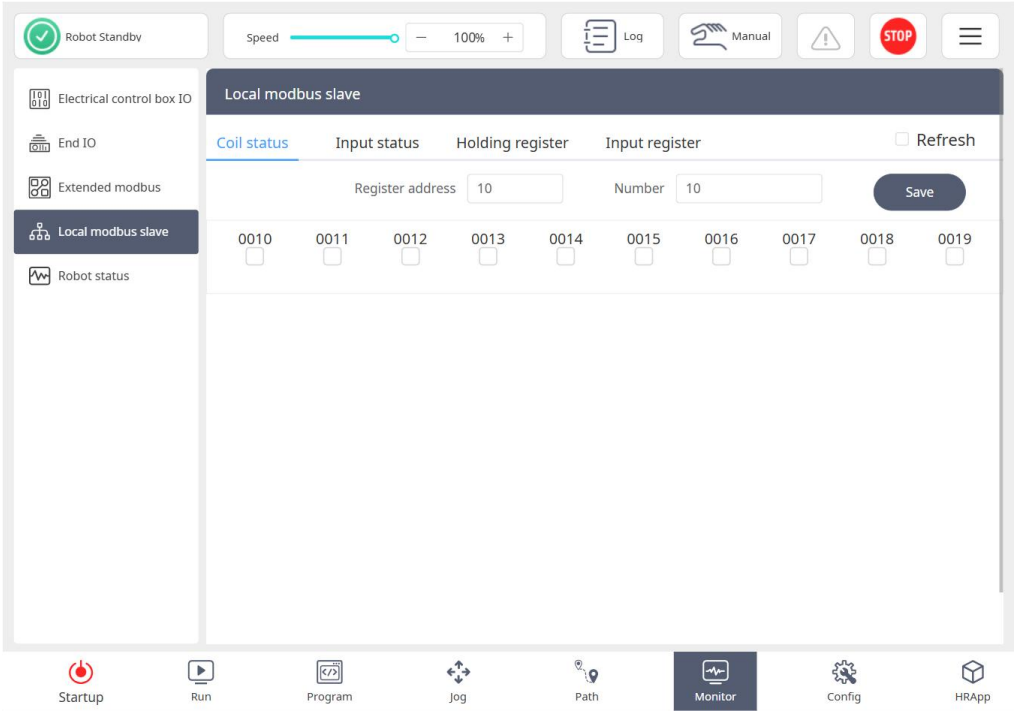
When a Modbus communication device is added to the configuration and successfully connected to the slave device, the signals of the added device will be displayed here for users to monitor the status. If data fails to be shown, please check whether the Modbus device parameters are properly configured in the settings.

Select"Real-Time Refresh"to view the latest data. To conserve system resources, the refresh function will automatically terminate after a certain period of time. If the interface is exited and then re-entered, the Modbus data will stop updating. The user needs to re-select"Real-Time Refresh"to reactivate the data update function.

9.4 Local Modbus Slave

This page is designated for performing read and write operations on Coil Status, Input Status, Holding Register, and Input Register. Additionally, it offers two input boxes to set the register address and quantity, enabling flexible monitoring or operation of specific register ranges.

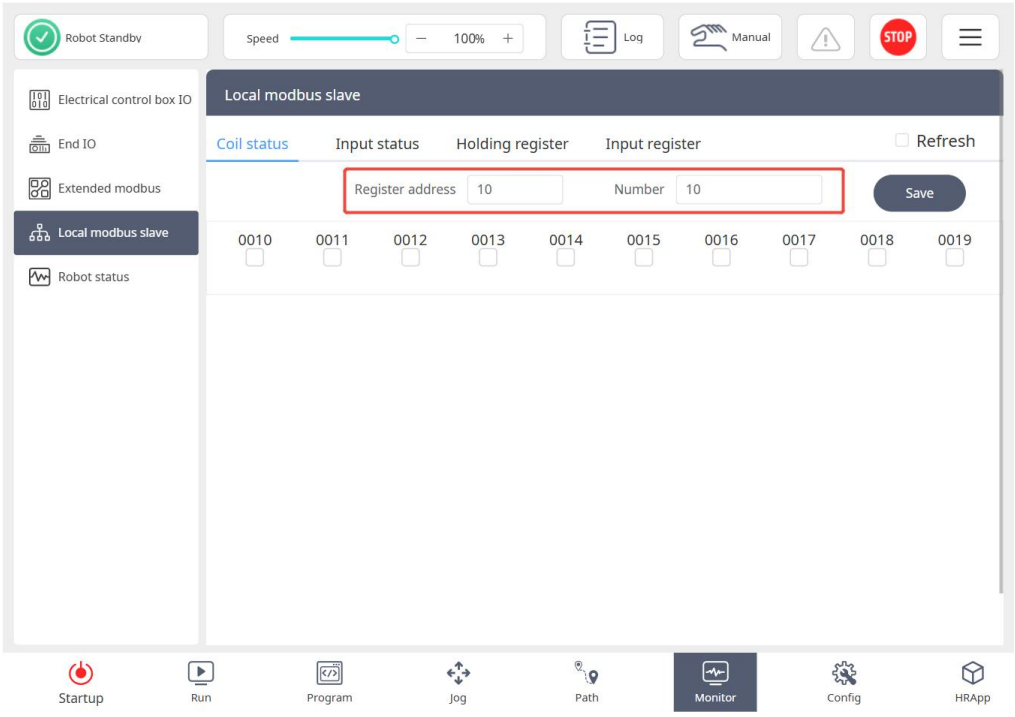
Fig. 9.4.1 Local Modbus Slave



9.4.1 Coil status

Indicates the output of digital signals, which are readable and writable with values of 0 or 1.

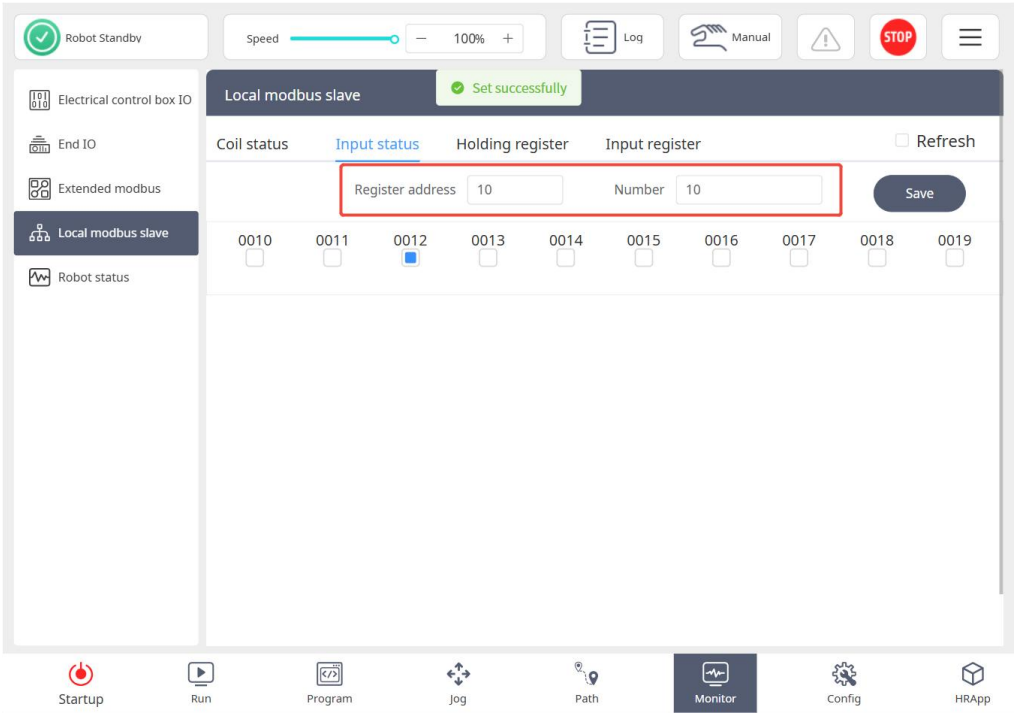
Fig. 9.4.1.1 Coil Status



9.4.2 Input Status

Indicates the input of digital signals, which are readable only with values of 0 or 1.

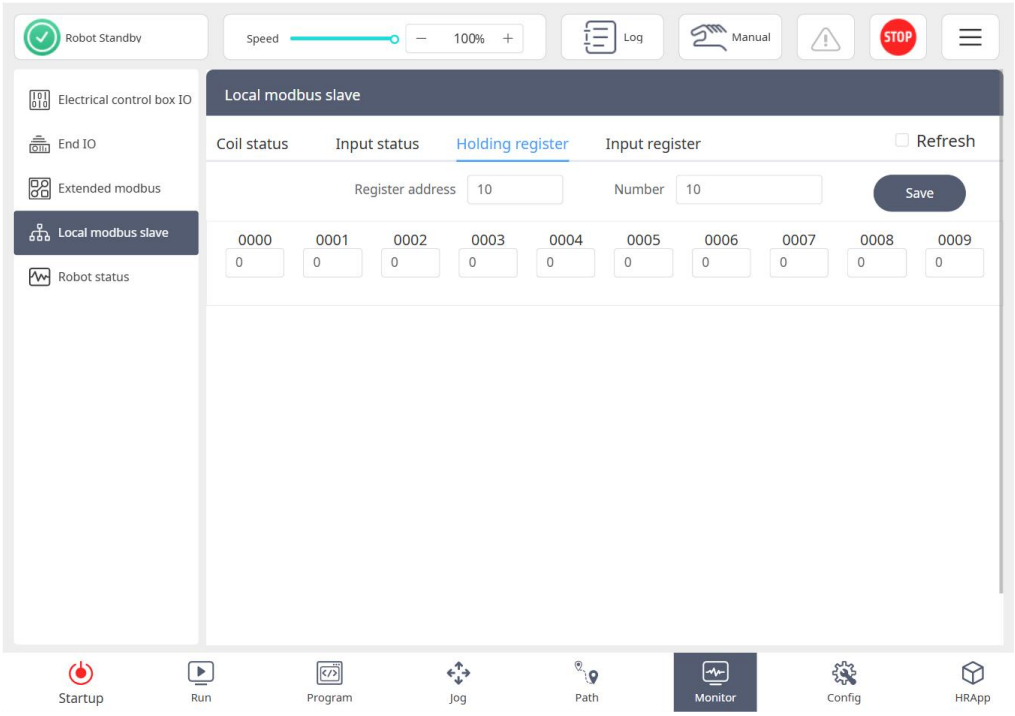
Fig. 9.4.2.1 Input Status



9.4.3 Holding Register

Stores numerical data, which is readable and writable in 16-bit integer format.

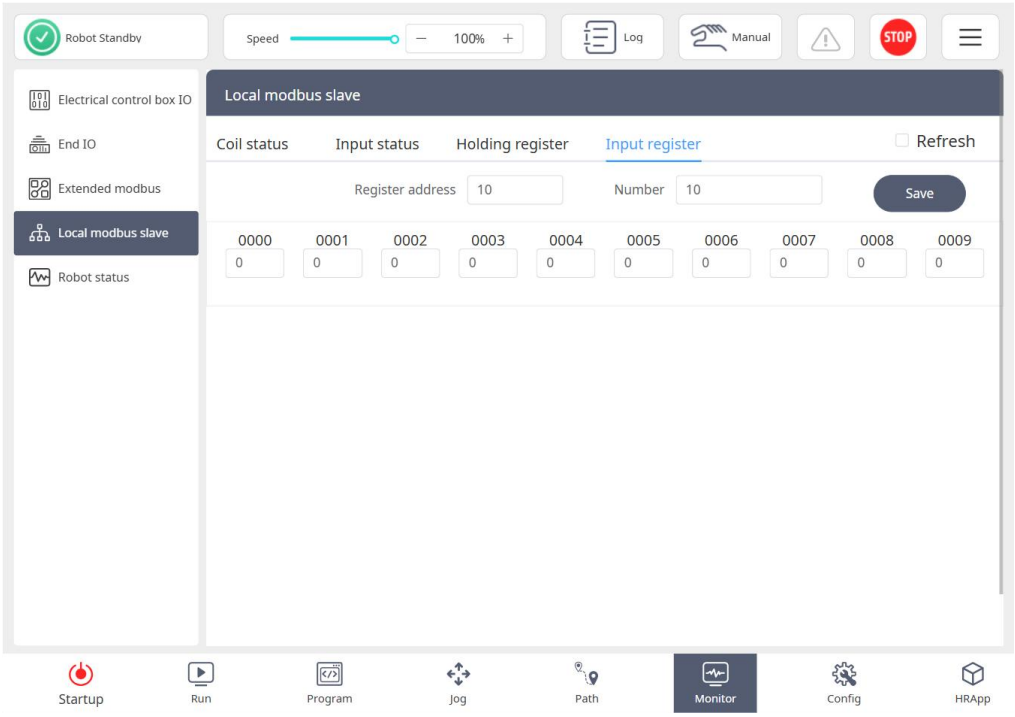
Fig. 9.4.3.1 Holding Register



9.4.4 Input Register

Stores numerical data, which is readable only in 16-bit integer format.

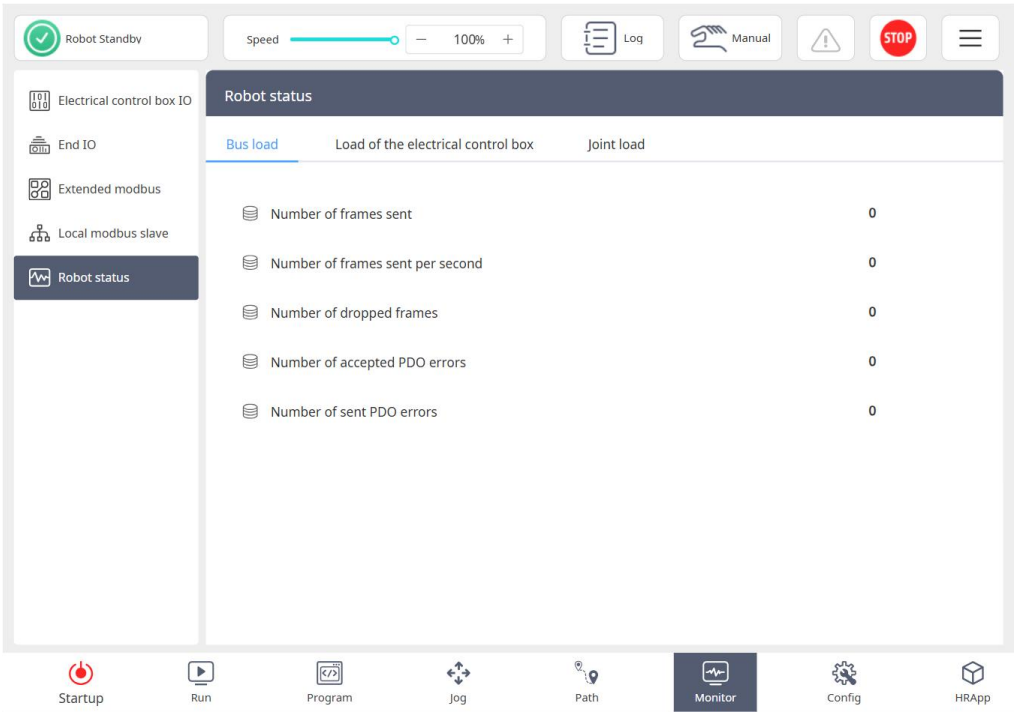
Fig. 9.4.4.1 Input Register



9.5 Robot Status

9.5.1 Bus Load

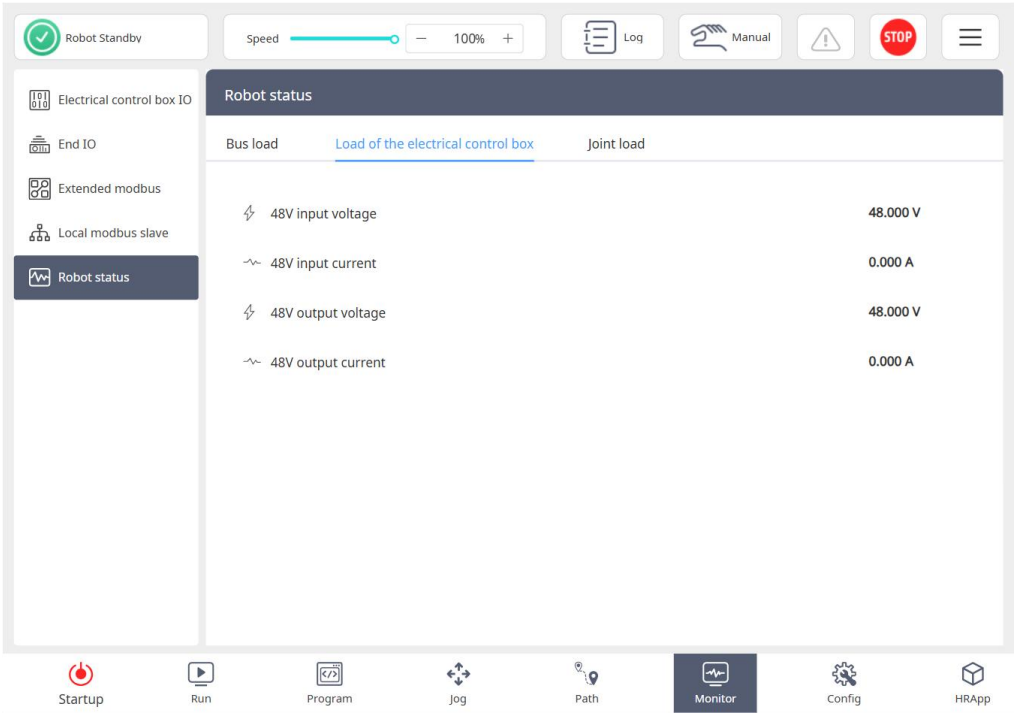
Fig. 9.5.1.1 Bus Load



Displays real-time data frame transmission and reception status.

9.5.2 Load of the Electrical Control Box

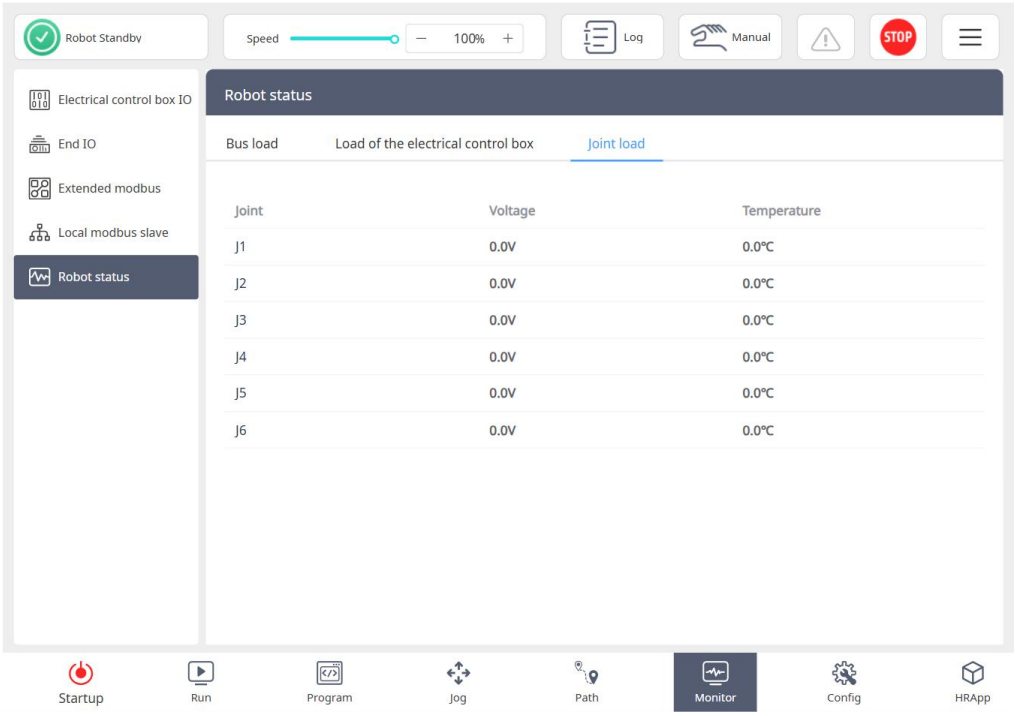
Fig. 9.5.2.1 Load of the Electrical Control Box



Displays real-time power supply status of the electrical control box.

9.5.3 Joint Load

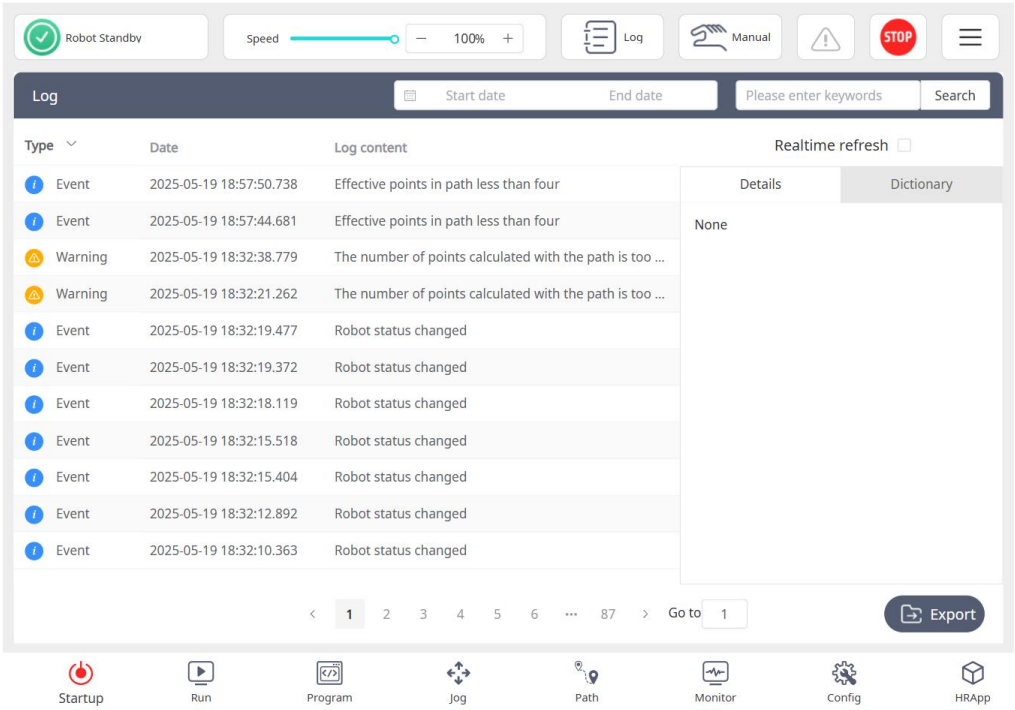
Fig. 9.5.3.1 Joint Load



Displays the joint-related information about the robot arm, including the voltage and temperature status of the joint load.

9.6 Log

Fig. 9.6.1 Logs



Displays message severity classification, log message generation time and specific message content.

- **Type:** Log types include error, warning, event, operation record, parameter record, programming record, etc. Log lists can be filtered by querying types.
- **Log Display** Content: Log display includes user name, operation, operation parameters, command jump operations, etc. The operation logs cover user behaviors, such as login, script running, and robot jogging operation; the parameter logs record parameter modifications, including previous and new values of parameters such as TCP, command, and variable parameters, as well as command jump chain links for parameter changes.
- **Export:** Export all logs to external devices. When users have questions, logs can be exported and sent to the technical support team for navigating and problem-solving.
- **Details:** Clicking on a specific log entry will display its detailed information.
- **Dictionary:** Enter the alarm and return code ID to query relevant errors.

Management

This chapter contains the following topics:

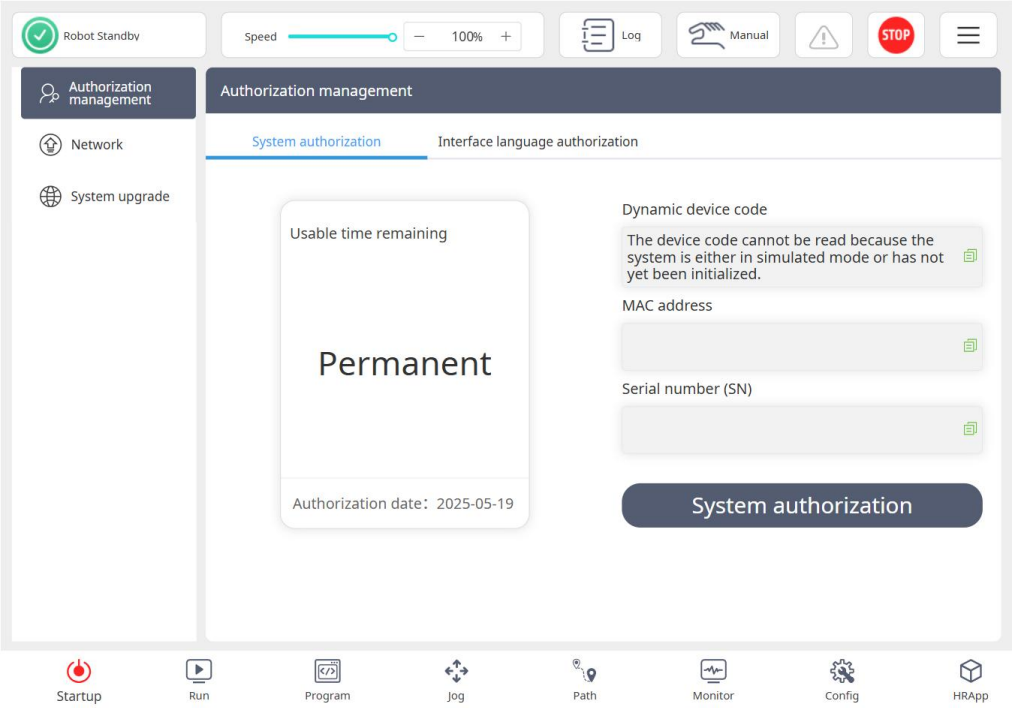
– Authorization Management
– Network Configuration
– System Update
– User Management

10.1 Authorization Management

10.1.1 System Authorization

If users need to operate the physical robot body using the programming functions, device authorization is required. To activate it, please contact the relevant supplier and the technical support team. Please provide the dynamic device code, MAC address, and serial number (SN code). After authorization, the page will display the authorization duration in real-time.

Fig. 10.1.1.1 Authorization Management



The custom permission level management allows users to comprehensively manage login permissions during usage.

SN Code: The unique device identifier distinguishing different units of the same type.

MAC Address: The hardware identifier for network devices, ensuring uniqueness in a network.

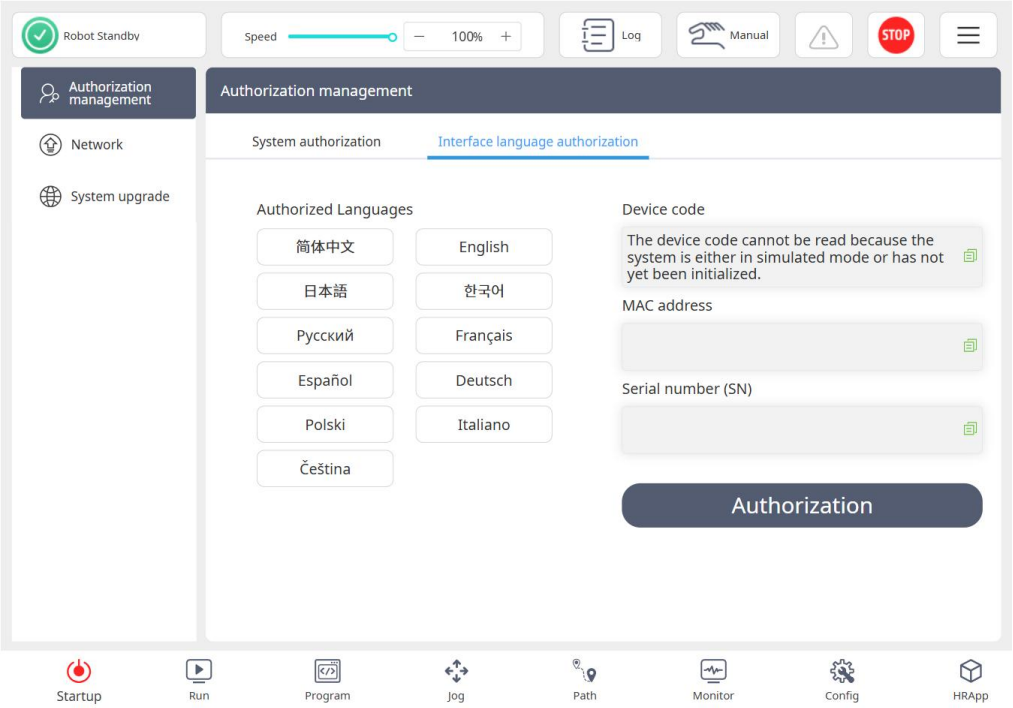
After three consecutive incorrect entries, system restart or waiting until the next day is required to re-enter. Authorization codes are valid for the day they are generated.

10.1.2 International Authorization

After two consecutive incorrect entries of the international language authorization code, the system must be restarted to input it again.

Once the authorization code is generated, it is permanently valid.

Fig.10.1.2.1 International Authorization Interface

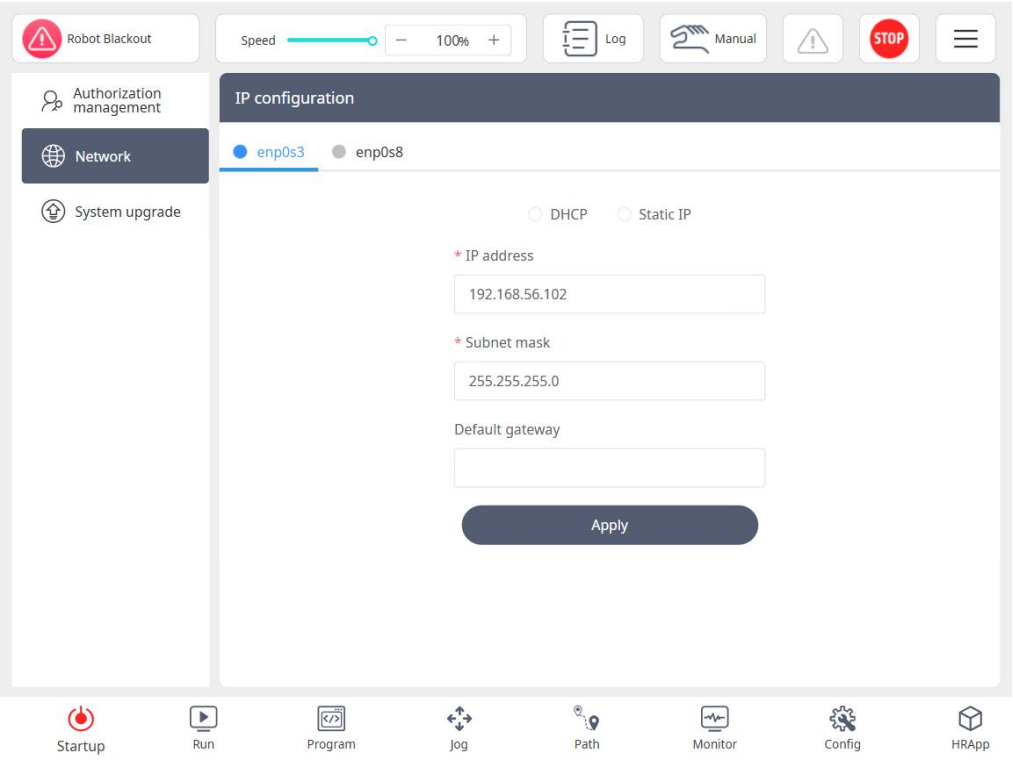


10.2 Network Configuration

On the robot teach pendant, click Management-> Network to set a static IP address, as shown in the figure below:

Note: The static IP address must not be 192.168.0.0, 192.168.0.1 or 192.168.0.255.

Fig. 10.2.1 Network Configuration Interface

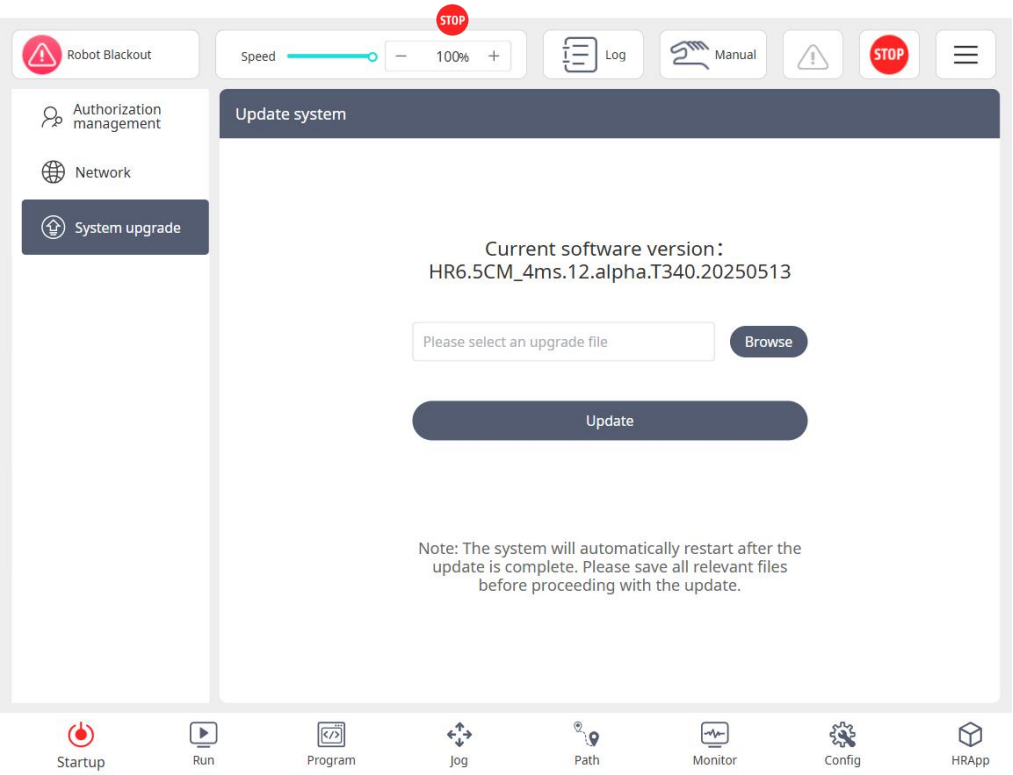


The blue prompts show the current network IP in use. After clicking Apply, the system will automatically reboot. Click OK to confirm. Users can also opt for DHCP dynamic allocation. Once the robot network is successfully established, it will automatically assign IP addresses.

To use DHCP for IP address allocation, ensure the robot network and DHCP server are on the same physical network; otherwise, IP acquisition will fail. For DHCP-related queries, consult the relevant IT department of the client.

10.3 System Update

Fig. 10.3.1 System Update Interface



This function allows for the controller software system upgrades. The current software version is displayed, and an upgrade package can be imported to update the robot arm's controller software system.

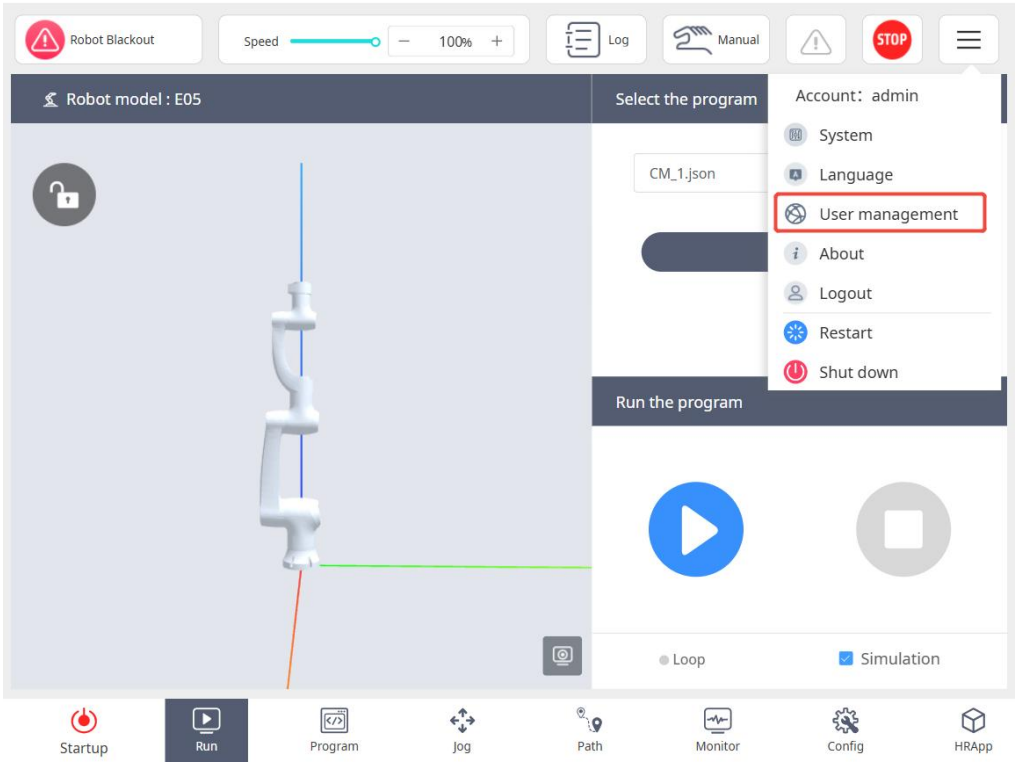
- **Browse:** Open the folder to select the upgrade package;
- **Update Package:** Provided by technical support team or corresponding suppliers;
- **Update:** By clicking Update, the software system will be automatically upgraded, and it will take 2-3 minutes to finish upgrading;

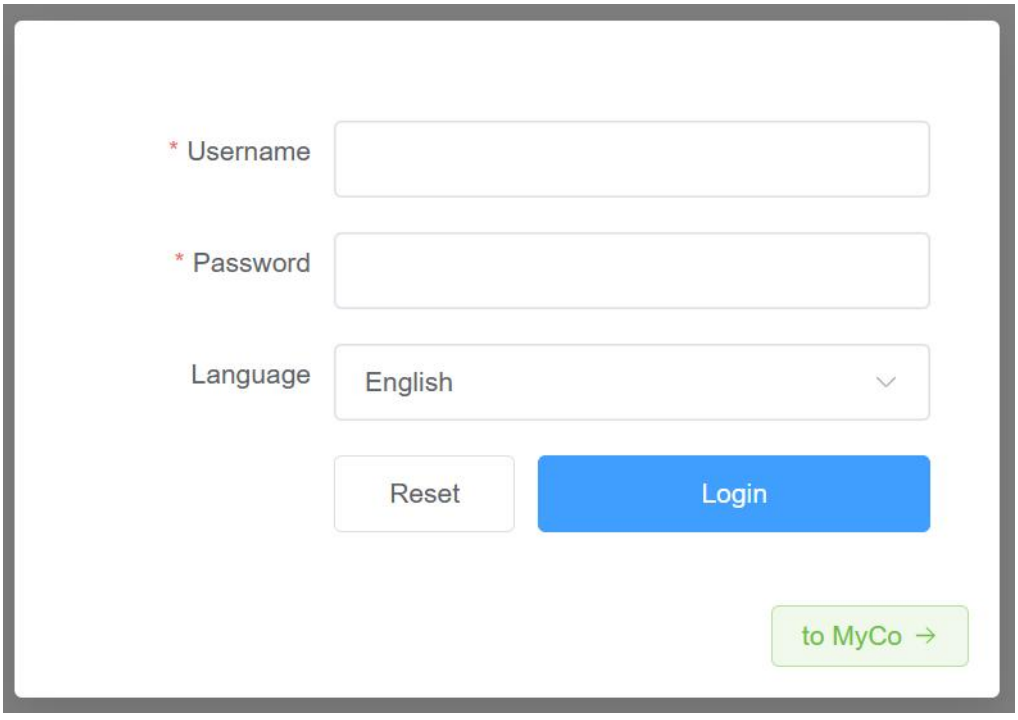
Note: Before upgrading, please contact the technical support team for guidance. More importantly, please ensure the robot arm is powered-off and all the data is backed up. The system will reboot after the upgrade. If it fails, check the upgrade package or contact the technical support team for help. Upgrading from version 5.0 to 6.0 is irreversible.

10.4 User Management

10.4.1 Accessing the User Management Backend

Fig. 10.4.1.1 Accessing User Management Backend Interface





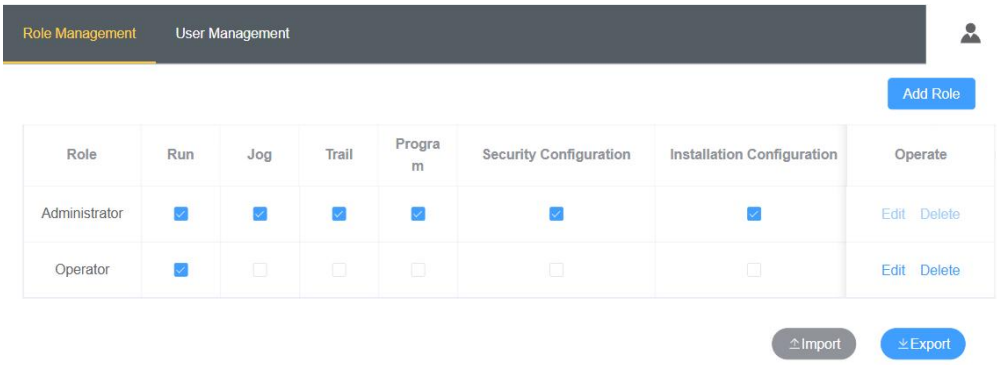
Click "User management" from the stack menu to enter the user management backend. Log in using an account with administrator privileges.

Back to Main System: Click to return to the login page of the robot operating

system, log in again and enter the system operation. Clicking "To MyCo" to return to the login page of the robot arm controller software system. Re-log in to enter the system and continue operations.

10.4.2 Role Management

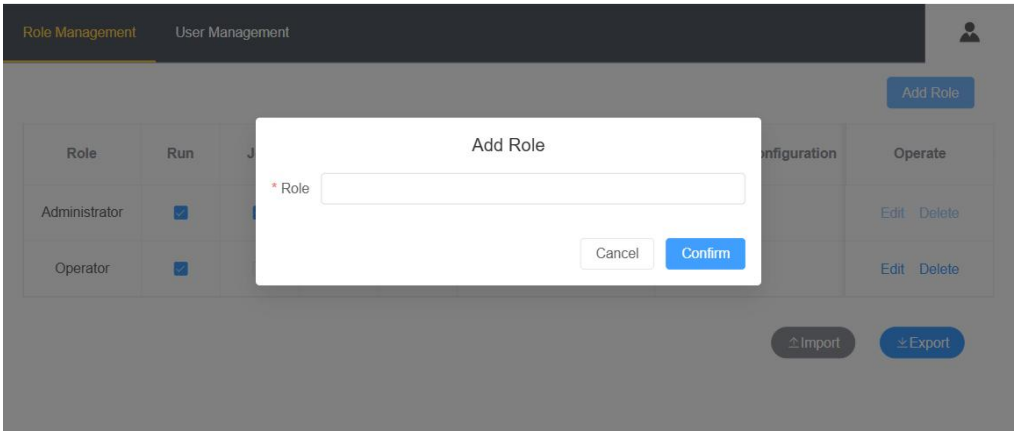
Fig. 10.4.2.1 Role Management Interface



Users can manage role permission groups on the page, including adding role, modifying role functions, deleting role and editing role permission names.

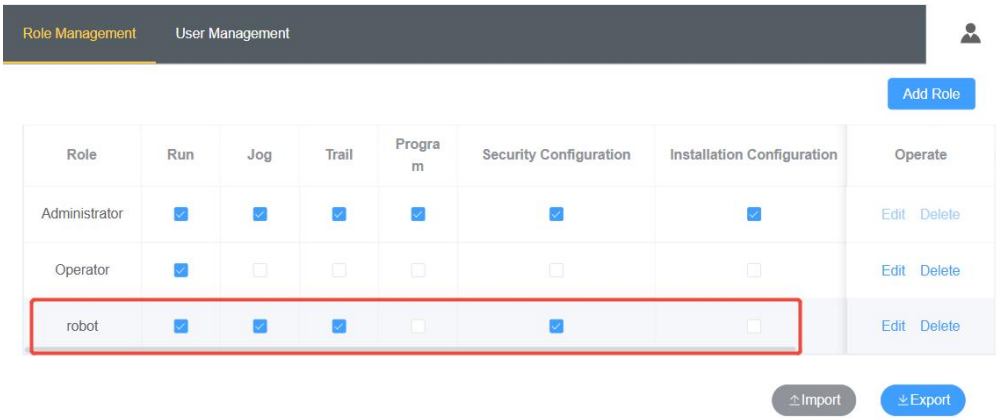
Add role permission: Click "Add Role", enter a role name in the pop-up window, confirm to activate it, and the new role will appear in the role permission list.

Fig. 10.4.2.2 Add Role



10.4.3 Role Management

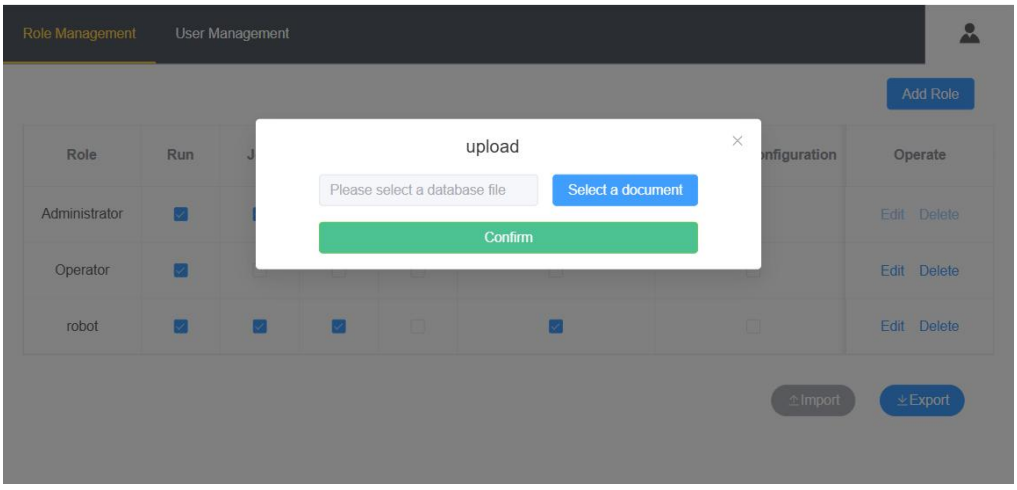
Fig. 10.4.3.1 Role Management



In the role permission list, users can customize the specific functions for each role permission group. Once the functions are selected, members of the role permission group will gain access to those functions.

- **Delete:** Click "Delete" to remove a role permission group. If users are assigned to the role permission group, the system will warn that they will lose their operational permissions. These users will not be able to operate the system until an administrator reassigns their permissions.
- **Edit:** Click "Edit"to change the name of a role permission group.
- **Export:** Click "Export"to save the current permission groups and their settings as a file of a fixed role.db type.
- **Import:** Click "Import" to upload files and restore previously exported role permission groups and settings.

Fig.10.4.3.2 upload



The role management table outlines specific permission groups for different user roles, such as admins and operators.

Table 10.4.3 Role Management

Role Type	Role Description	Remark
Admin	Access to full permissions, able to manage all accounts and secure passwords.	Manage all account passwords, including security passwords

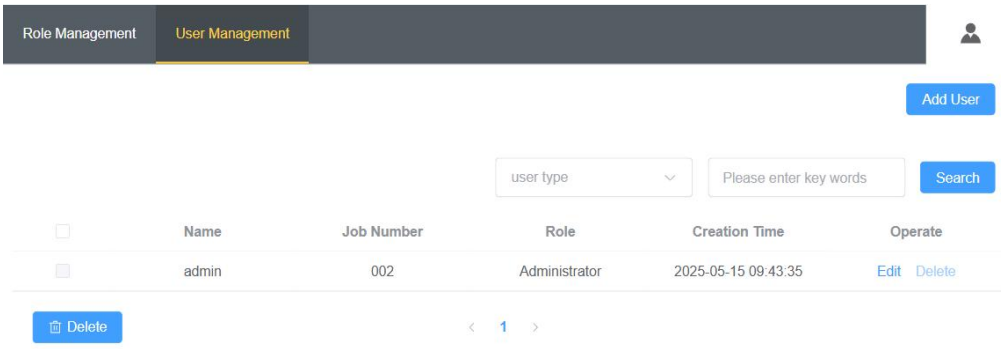
Operator	Can only access to t he "Run" interface. Other operations require additional administrator authorization, and the administrator will provide the login account and password.	The administrator provides the login credentials (account and password).

- The default admin account and password are set before delivery. If forgotten or lost, please consult the corresponding supplier and the technical support team;
- Users can access robot functions based on their account permissions.;
- If the operator forgets the credentials (account and password), please request assistance from the administrator;

The administrator can assign specific permissions to particular users as needed.

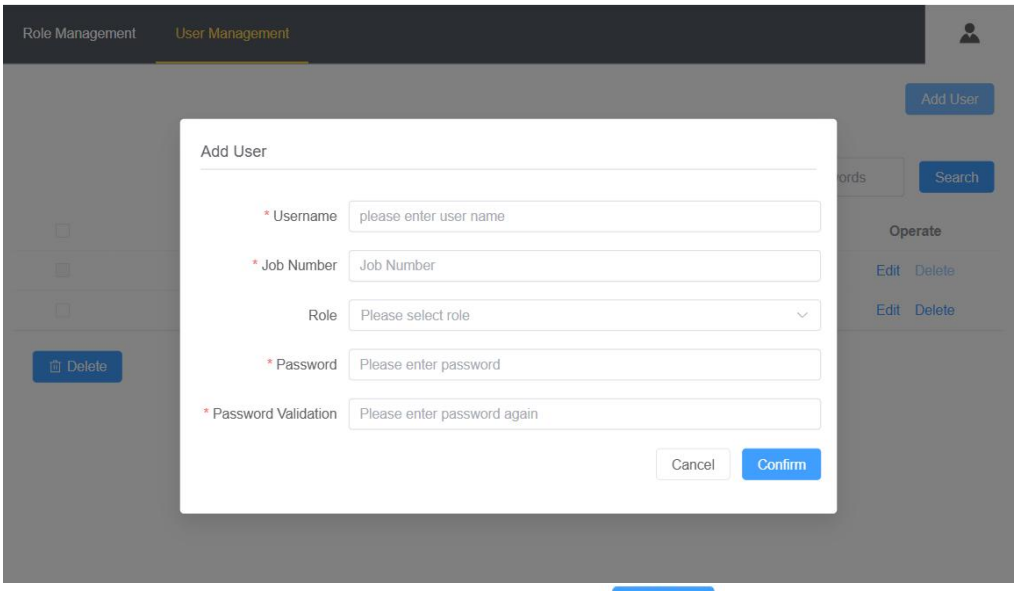
10.4.4 User Management

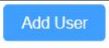
Fig.10.4.4.1 User List



On this page, users can manage system users, including adding new accounts, deleting accounts, and editing user details.

Fig.10.4.4.2 Add User



Add user: After administrator account login, click 

"Add user". A popup will

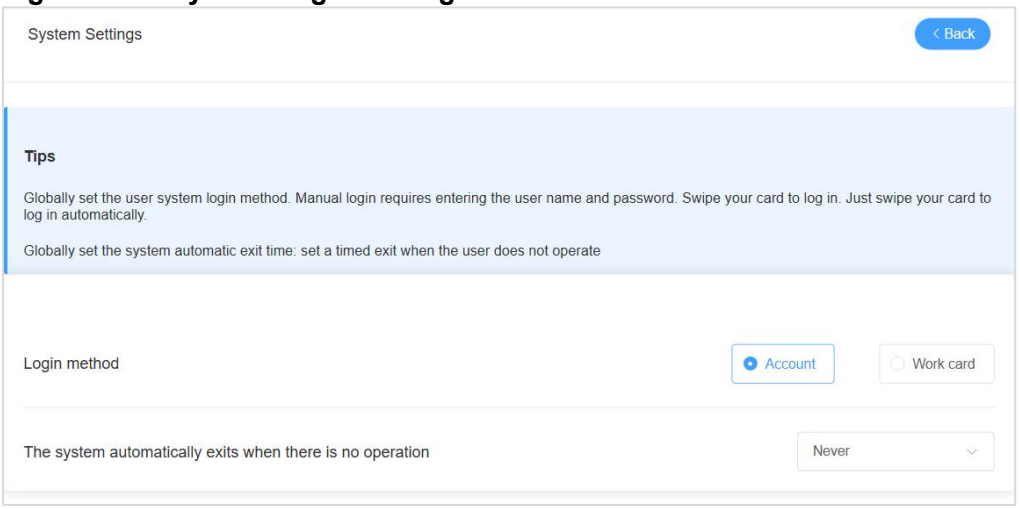
appear for entering user name, job number, role permission group, password, and password verification. After confirmation, the addition will be successful.

Edit user: Click Edit and the user editing popup will appear for editing the user name, job number, permissions, password and other contents of the currently selected user. Click Confirm to activate it.

Delete user: Click Delete to delete the currently selected user. The system will prompt for confirmation before deletion; the deletion will be finalized upon confirmation.

10.4.5 Login Setting

Fig. 10.4.5.1 System Login Settings



The system offers multiple login methods: credentials-based and card-based. Credentials-based login is the default. For card-based login, configure the card reader for one-tap access.

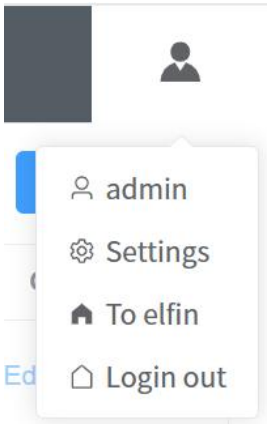
Fig. 10.4.5.2 Login and Auto Logout



In the settings, the users can configure the system's idle auto-logout time. If the system is unused for the set period, it will automatically log out the current user to prevent accidental robot operation.

The default is "Never" (no auto-logout). Users can adjust it based on their needs.

Fig. 10.4.5.3 Return to Main System



Click "Personal Center"on the upper-right-corner navigation bar, then select "Main System"to exit the user management platform and return to the controller software operation interface. Re-login is required to operate the robot.

CMApp

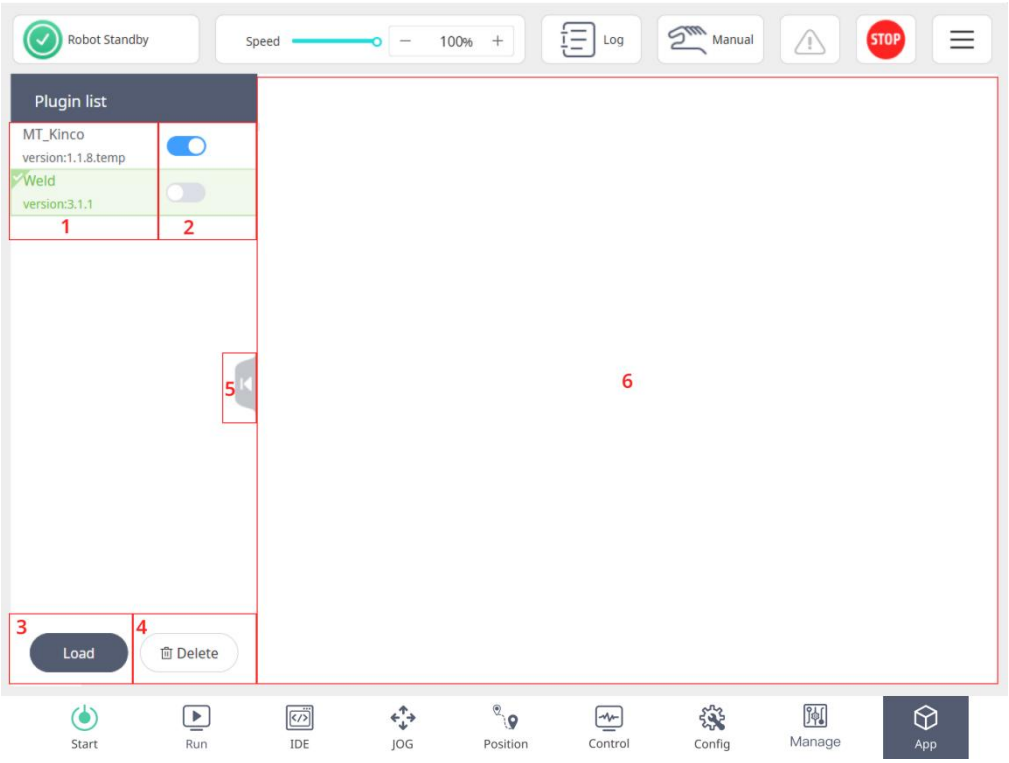
This chapter contains the following topics:

– App

11.1 App

Based on the MyCo software architecture, this is a plugin framework designed to enhance the multifunctional capabilities of robotic systems. The platform enables users to develop and integrate custom functionalities, facilitating seamless interoperability between robotic units and external intelligent systems (e.g., IoT devices, automated workflows, or legacy automation infrastructure). By standardizing API connectivity and protocol adaptation, the platform streamlines system integration, improves operational throughput, and ensures backward/forward compatibility across heterogeneous robotic networks.

Fig. 11.1.1 Plugin Platform



Plugin Management Interface:

- **Plugin List:** Displays all currently imported plugins, showing their names and status of the programming switches;
- **Switch:** When enabled, the plugin's related instructions are visible in the Program interface for use. When disabled, the plugins are hidden;
- **Load:** Import relevant plugin installation packages from external files. For obtaining specific installation packages, please consult the technical support team;
- **Delete:** Removes the selected plugin. Upon confirmation, the plugin is permanently deleted;
- **Expand/Collapse:** Controls the expansion or collapse of the plugin list for easier viewing;
- **Plugin Details:** Displays configuration settings and details of the selected

plugin;

Note: The system currently supports only plugins developed by the company. For any additional robotic functionality requirements, please contact the technical support team.

Exception Handling

This chapter contains the following topics:

– Protective Stop
– Emergency Stop
– Safe Collision Stop
– Trigger Safeguard Monitoring
– Beyond Safety Range
– Singularity Point States

12.1 Protective Stop

During robot operation, errors may occur, leading to abnormal performance or other exceptions. To avoid such abnormal conditions, a protective stop is required. A Protective Stop is a "safe stop" issued by the robot controller when an error is detected. This warning message appears on the teach pendant or external monitor to indicate the error state. In robotics, protective stops are categorized as follows:

- **Class 0 stop,** also known as Stop 0: Immediately cut off the power supply of the motor.

  Since the motor loses power during the deceleration process and cannot be controlled, the robot arm will deviate from the predefined running path, Class 0 stop is an uncontrolled stop mode. The emergency stop button is implemented by performing Stop 1 first; if Stop 1 does not take effect within the specified time, Stop 0 will be automatically started.

- **Class 1 stop**, also known as Stop 1: The controller controls the motor to slow down to stop, and then cuts off the power supply.

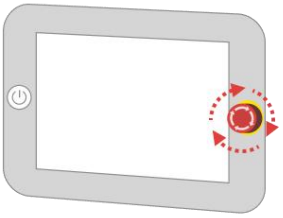
  This is a controlled stop method, and the robot arm basically completes the deceleration according to a predefined path. In the latest robotic systems, due to the use of safety controller technology, after the emergency stop button is triggered, the system prioritizes the execution of Stop 1 stops (except in manual mode) to protect the braking system. If Stop 1 does not take effect within the specified time limit, the system will automatically upgrade to Stop 0 stop.

- **Class 2 stop**, also known as Stop 2: It is completely decelerated through the servo system. After stopping, the power supply is not cut off, and the motor is still electrified. It is a completely controlled stop mode. Nowadays, the program stops of robots all fall into this category.

  The current controller software realizes protective stop operations through various functions, including emergency stop, safety range and collision.

12.2 Emergency Stop

For the "Class 0 stop" in protective stop, if the user predicts that a sudden hidden danger will occur during the robot arm motion, please press the emergency stop button immediately. After which, the mechanical arm and electrical control box are powered off and stop working, and reset can be done according to the prompt. After resetting, the robot is in a power-off state, and the user needs to power it on again to get the robot back to normal working state.



Reset Steps:

Unscrew the red emergency stop button on the teach pendant clockwise first.

Click the "Reset" button in the popup to clear the error.

Return to the running homepage and click the "Power on" button to return to the normal state.

Note: If the emergency stop button is not unscrewed and the user clicks "Reset" directly, a "Reset failed" message will be prompted. The user must follow the reset process to return to normal operation. If repeated resets fail, check for hardware issues.

12.3 Safe Collision Stop

When the robot arm hits an obstacle, it will judge whether to stop moving according to the force fed back by the obstacle. The magnitude of the force is judged according to the level configured in the safety limits. If the safety risk level is higher, the greater feedback force is needed to stop the robot; if the safety risk level is lower, the smaller feedback force can also stop the robot.

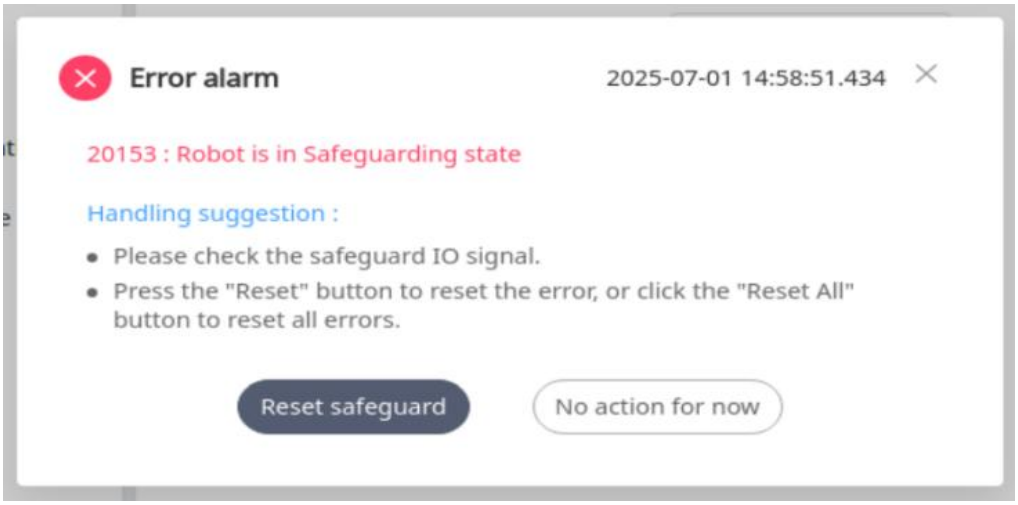
After stopping, it can be restored to normal by resetting. To be on the safe side, make sure the robot doesn't hit an obstacle when resetting, so as not to trigger a safe collision stop again.

12.4 Trigger Safeguard Monitoring

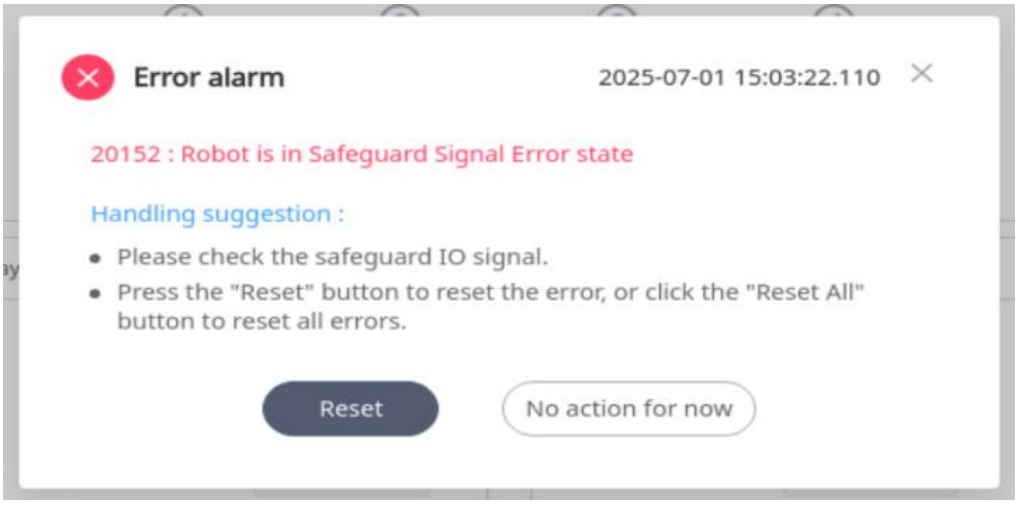
Enable the safeguard monitoring in the safety limits, configure the safety I/O signal to stop or pause, and when the I/O signal is monitored, the system will trigger a protective stop and pop up a prompt.

The two input channels of the safeguard monitoring cannot be the same. If the user select the configurable inputs CI0 and CI1, there are the following three possibilities:

- If both CI are high, the robot can move normally;
- If both CI signals are at low level, the interface displays the prompt “Robot is in Safeguarding state”. At this time, the robot remains in the disabled state. After both CI signals are restored to high level, click “Reset safeguard” and then enable the robot again. The robot can then move normally.



— If the two CI are one high and one low, the interface prompts "The robot is in a state of inconsistent safeguard monitoring signals, please reset". At this time, the robot is in a power-off state. Restore both CI to high levels, click Reset, re-power on and enable, the robot will be able to move normally;



— If there is an error in one of the two CI, such as one of the CI timing out during use, resulting in the two CI being unable to be set high or low at the same time, the interface prompts "The robot is in the safeguard monitoring error state, please reset". At this time, the robot is in a power-off state. Restore both CIs to normal, click Reset, re-power-on and enable, the robot will be able to move normally.

- **Safeguard Monitoring Stops:** When the safety I/O signal is monitored, it stops moving and a popup appears to indicate the safety protection state. The robot can only be operated again after resetting and clearing the error. After reset, the robot is not de-electrified.
- **Safeguard Monitoring Pause:** When the safety I/O signal is monitored, the script will be paused. After resetting and clearing errors, the current script can continue to run.
- **Safeguard Monitoring Stop and Jog:** When the safety I/O signal is monitored, the script running will be terminated. If the safety light curtain is currently triggered, jog operation is available to adjust the robot

position (while the script running is disabled). After resetting and error clearing, the current script can be run again.

Reset Steps:

- Release the protective switch first.
- Click the "Reset" button in the popup to clear the error.

Note: If the switch is not released, click "Reset" directly, and it will prompt "Reset failed". Users still need to follow the reset process to return to normal.

12.5 Beyond Safety Range

During the movement of the robot arm, if the movement exceeds the restricted safe range (joint range), a prompt will pop up. The user needs to press and hold to restore. Operate the robot to move away from the restricted area, and the state return to normal when the robot exits the restricted area. Refer to "Safety Range" for details.

12.6 Singularity Point States

"Singularity" is a common robot phenomenon, which is not caused by robot design, processing, or system defects. It is an inherent characteristic of six-degree-of-freedom non-redundant robots. Customarily, the configuration where strange phenomena occur is usually called "singular configuration", also called "singularity point".

Singularity point usually refers to some positions of the robot arm in its workspace, which will cause some performance parameters of the robot arm (such as velocity, acceleration, force, etc.) to be discontinuous or infinite. These positions may be due to the structural design of the robot arm, joint configuration, or characteristics of the kinematics equation. At the singular position, the robot arm may encounter the following situations:

- **Infinite Joint Velocity:** At some singularity points, in order to keep the velocity of the end effector constant, it may be necessary to have an infinite angular velocity of a certain joint, which is impossible in practical applications.
- **Infinite Joint Acceleration:** Similar to velocity, some singularity points may require infinite acceleration of the joint to achieve the desired end acceleration.
- **Infinite Moment:** In some positions, in order to maintain the balance of the robot arm or generate a certain force, infinite driving torque may be required.
- **Discontinuity of Kinematic Solutions:** At some singularity points, multiple kinematic solutions may exist for the robot arm, i.e., multiple different combinations of joint angles can lead to the same end position and pose. This will make it difficult for the control system to choose a

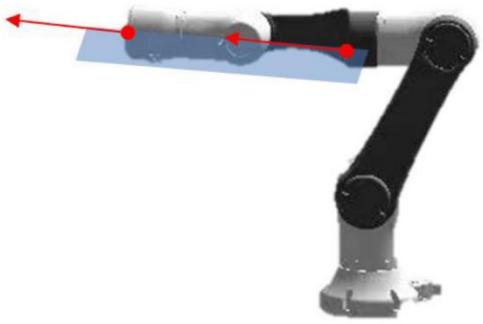
stable solution.

- **Range of Workspace:** At the edge of the workspace of the robot arm, singularity points may also appear, because some joints of the robot arm may have been extended to the limit and cannot be moved further. A proper understanding of the concept of singularity points is helpful to avoid these unfavorable positions when designing and controlling the robot arm, and ensure the stable operation and efficient work of the robot arm. In practical applications, engineers will reduce the impact of singularity points by optimizing the structural design of the robot arm and improving the control algorithm.

12.6.1 Wrist Joint Singularity

When the MyCo robot "joint 4" and "joint 6" are parallel, the robot is in wrist singularity. In this singular pose, "joint 5" of the robot is usually near 0°, as shown in the following figure:

Fig. 12.6.1.1 Wrist Joint Singularity

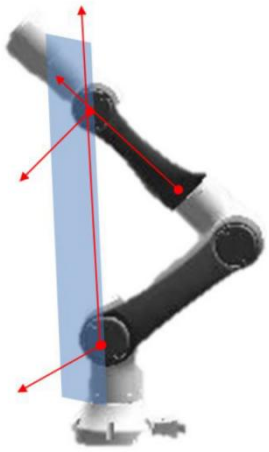


When the user uses Cartesian space commands to control the robot to translate and rotate, and the robot reaches the position of the wrist joint singularity and its adjacent area, the controller will trigger the software emergency stop, and the user will be informed that "The robot is approaching "Singularity point"". When wrist singularity occurs, the motion velocity of "joint 4" and "joint 6" may approach the joint-limiting velocity.

12.6.2 Shoulder Joint Singularity

When the end of the robot arm "ball joint center point" (the ball joint center point refers to the intersection of the three axes of "joint 4", "joint 5" and "joint 6" of the robot arm) is above the rotation center of "joint 1", the robot arm is in the shoulder joint singularity configuration. Under conventional installation, the shoulder joint is usually located at the top of the robot arm.

Fig. 12.6.2.1 Shoulder Joint Singularity



When the user uses Cartesian space command to control the robot arm to translate and rotate, and the robot arm reaches the singularity position of the shoulder joint and its adjacent area, the controller will trigger the software emergency stop. When shoulder singularity occurs, the motion of "joint 1" and "joint 4" may rotate rapidly by 180°

12.6.3 Elbow Joint Singularity

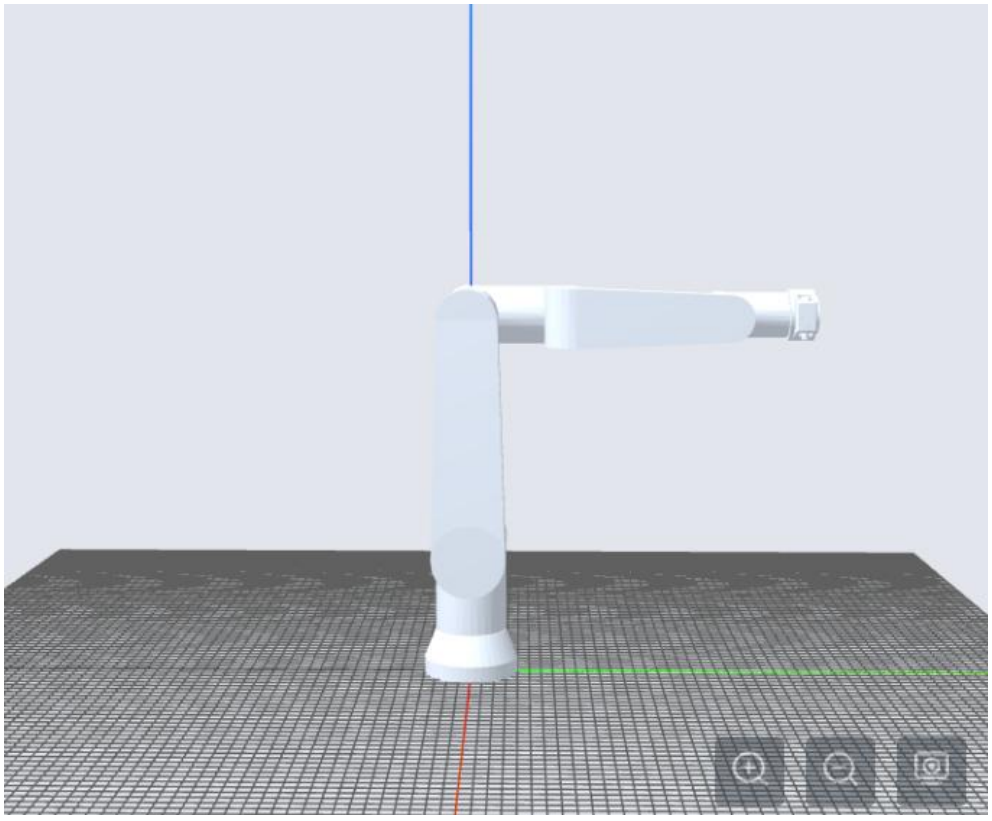
When the robot arm "joint 2", "joint 3", "joint 4" are coplanar (as shown in Fig. 3), the robot arm is in the shoulder singularity configuration. The elbow joint configuration only appears when the robot arm is fully deployed, which is relatively intuitive. In this singularity pose, the robot arm "joint 3" is usually near the 0° position.

Fig. 12.6.3.1 Elbow Joint Singularity



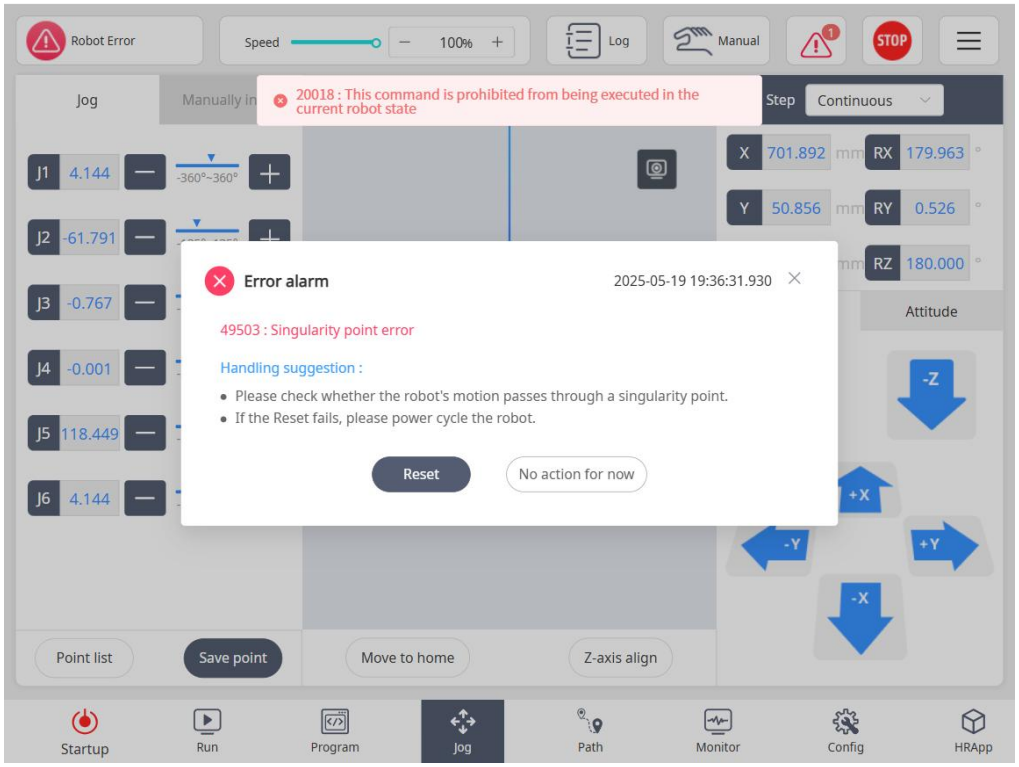
When the user uses Cartesian space commands to control the robot arm to translate and rotate, and the robot reaches the shoulder joint singularity position and its adjacent area, the controller will trigger the software emergency stop, and the user will be informed that "The robot is approaching "Singularity point"". When shoulder singularity occurs, the motion velocity of "joint 2" and "joint 4" may approach the joint-limiting velocity.

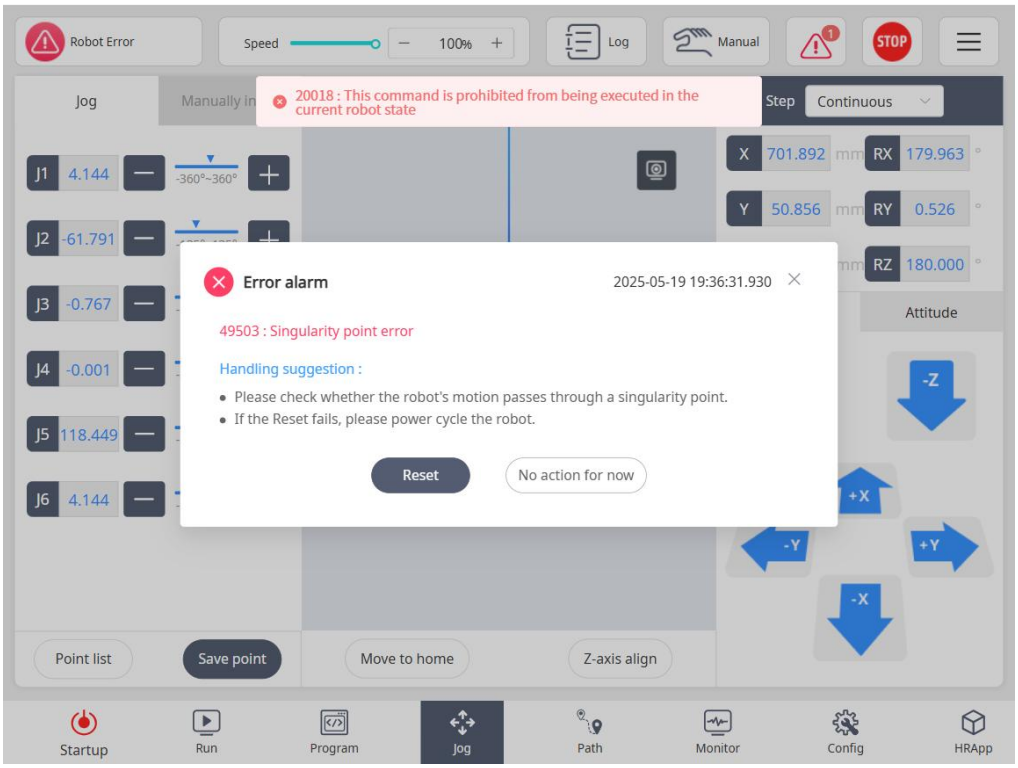
12.6.4 Singularity Warning and Recovery



When the robot arm approaches the wrist joint singularity point, such as (90, 0, 90, 0, 0, 0), when the user uses Cartesian space commands to control the robot arm to perform position translation and pose rotation, the robot software emergency stop will be triggered, prompting "Singularity error".

When a singularity error occurs, the user can dismiss the singularity alarm as follows





– Click "Reset" to clear the error and the popup disappears;

– Adjust any axis from J1 to J6 so that the robot pose is not at the singularity position.

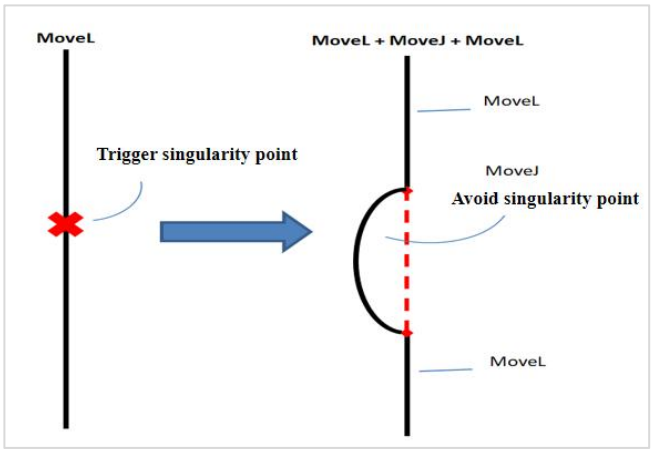
Note: When encountering a singularity point, the pose of the robot arm can only be moved away from the singularity point by adjusting the joint angle after reset. At this time, adjusting the tool position direction (X, Y, Z, RX, RY, RZ) is ineffective.

12.6.5 Singularity Avoidance

– Try to keep the Cartesian space path away from the singularity point. Taking the wrist singularity as an example, keep "joint 5" as far away from the 0° position as possible. It is recommended that all points in a path meet the requirements of angle of "joint 5" less than 3° or greater than 3°.

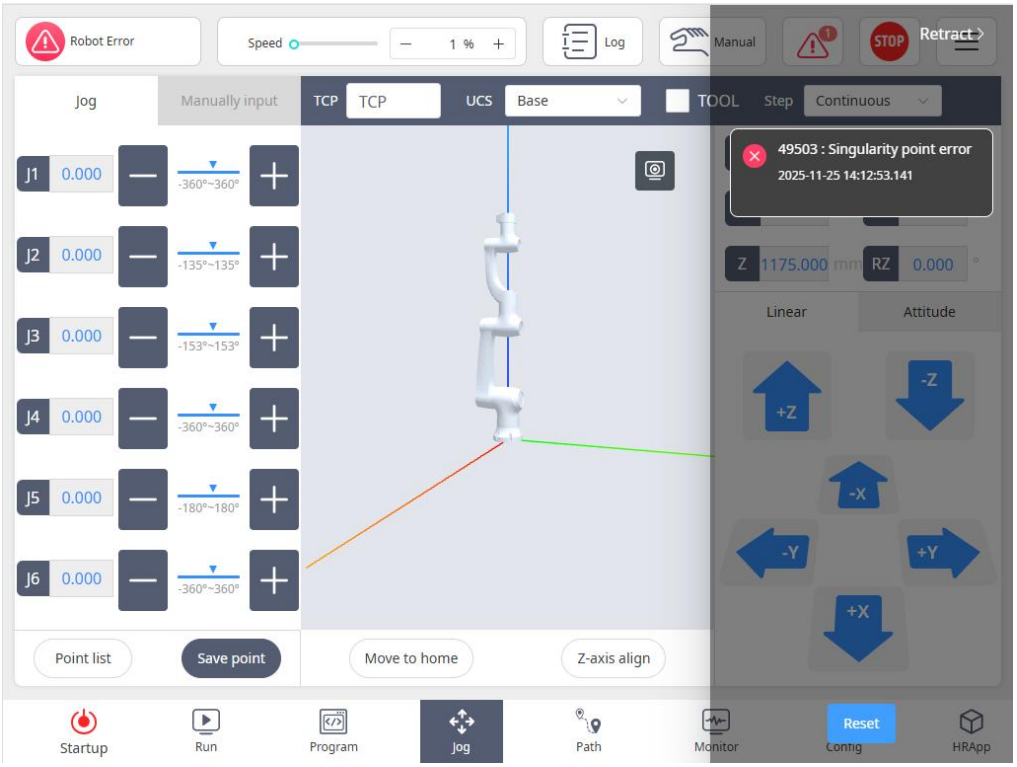
– Use joint motion "MoveJ" to transit the Cartesian space, and pass through the singularity points.

Fig.12.6.5.1



12.6.6 Warning Popup

Fig.12.6.6.1 Warning Popup

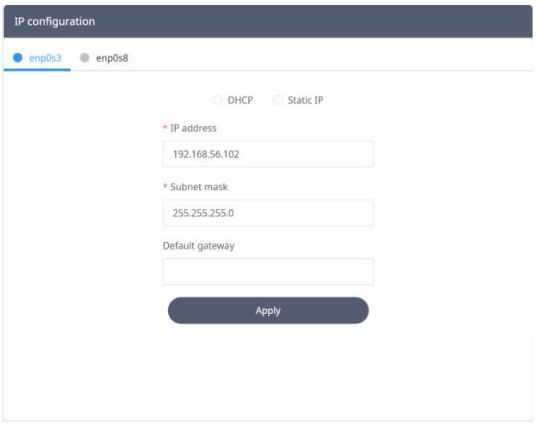
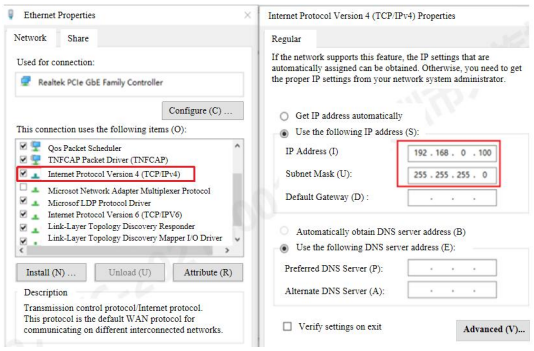


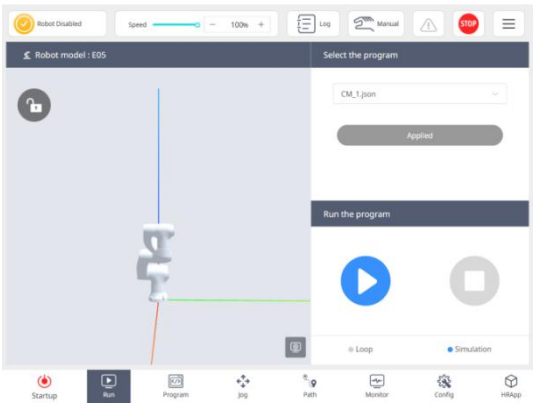
When an error prompt appears in the system, selecting "Do not process for the time being" closes the alarm popup. This suspends processing of the error, and an alert

icon  will appear at the top right. Clicking this icon displays all active error prompts. Users may then click the "Reset" button to reset errors that are no longer valid in the robot's current state, causing the popup to disappear.

End-effector Connection

The controller software operation interface can be remotely displayed on the personal computer through the network. But the computer and the robot need to be in the same subnet. In order to realize communication between the computer and the robot, the network can be configured in the following ways:

SN	Operation	Example
1	On the robot teach pendant, click Menu-> Management-> Network to set a fixed IP address. Note: IP addresses cannot be 192.168. 0.0, 192.168. 0.1 or 192.168. 0.255;	
2	Configure the laptop TCP/IPv4 network as a static IP address;	
3	After configuration, check whether the laptop and the robot are in the same network segment. If so, the robot can be controlled through the web page;	

4	Open the browser, type the address in the following format in the URL bar: Robot IP +/dist (such as 192.168. 0.10/dist), and enter the web page. After successful login, users can control Comau S.p.A through Comau S.p.A software;	

Virtual Robot

This chapter contains the following topics:

– VMWare Virtual Machine
– Simulation Control Box

14.1 VMWare Virtual Machine

Comau Robotics provides a Virtual Robot feature that allows users to familiarize themselves with and operate Comau robot functions through the controller software without connecting to a physical robot.

Users must install VMWare Workstation Player or a compatible version beforehand. The following section describes the configuration and usage of the virtual machine in this environment.

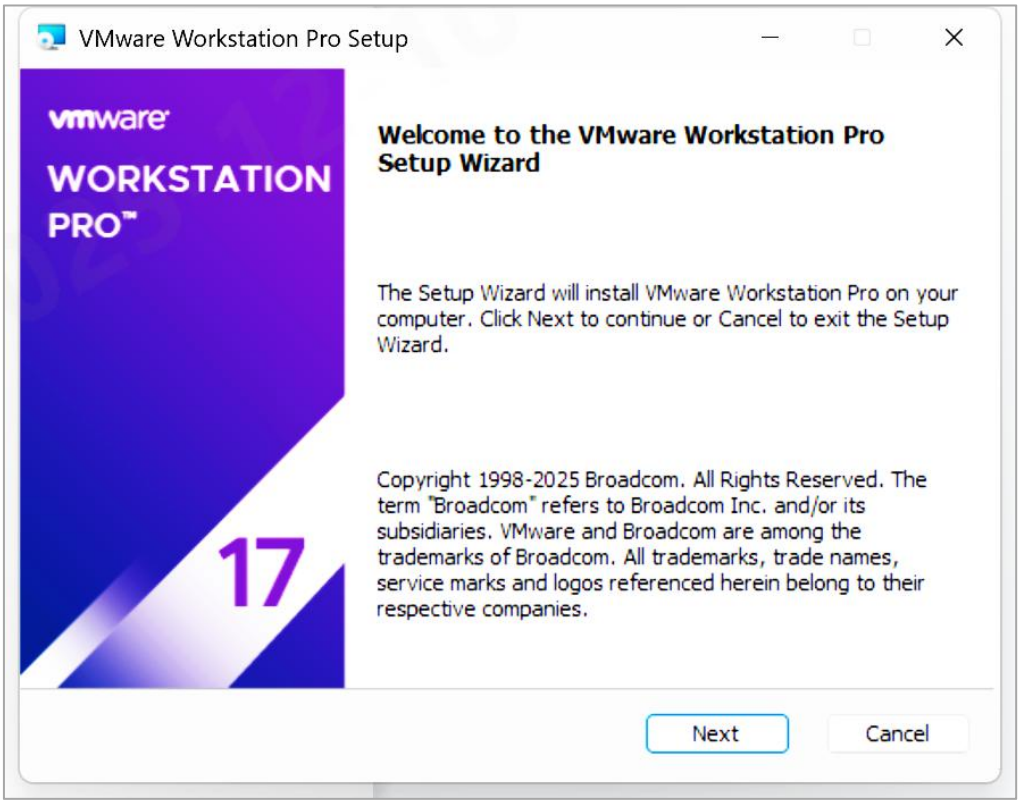
The virtual robot must run on a Windows 11 host system using VMWare. The instructions below detail the setup and operation process.

14.1.1 Installation

Please download VMWare from the official website. The installation procedure is as follows:

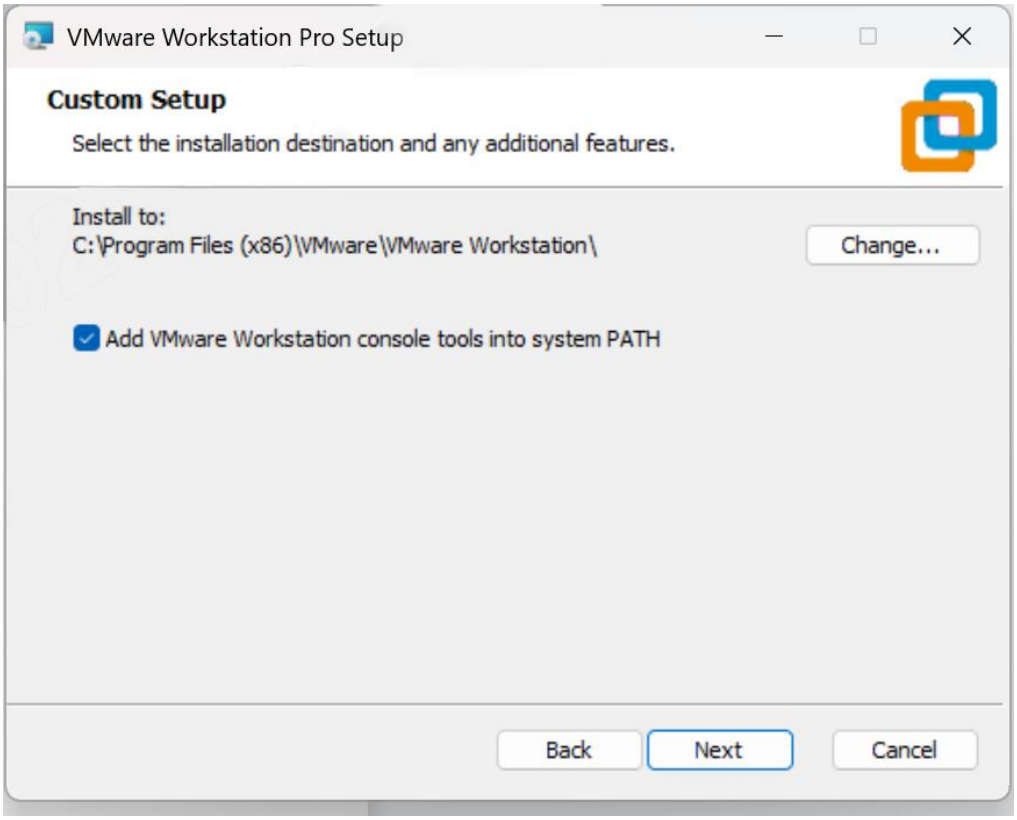
1.Double-click the installation package to begin. Click Next.

Fig.14.1.1.1 Install VMWare



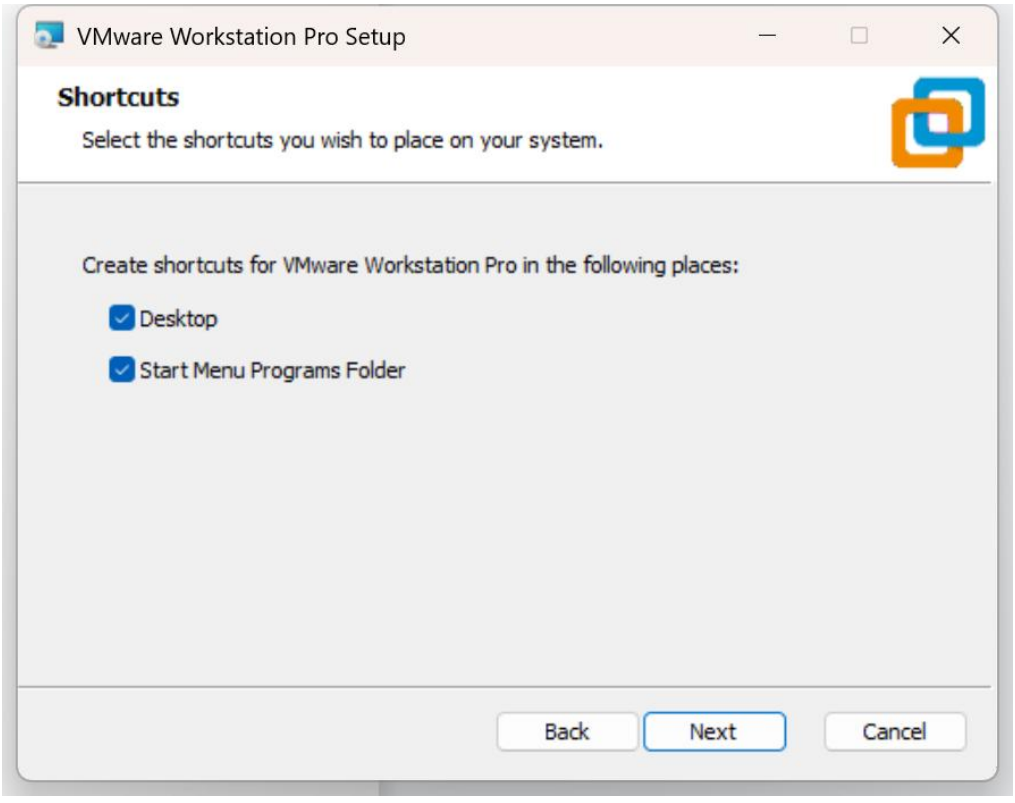
3.Select the installation directory and check “Add VMware Workstation console tools into the system PATH”, then click Next.

Fig.14.1.1.2 Custom Setup



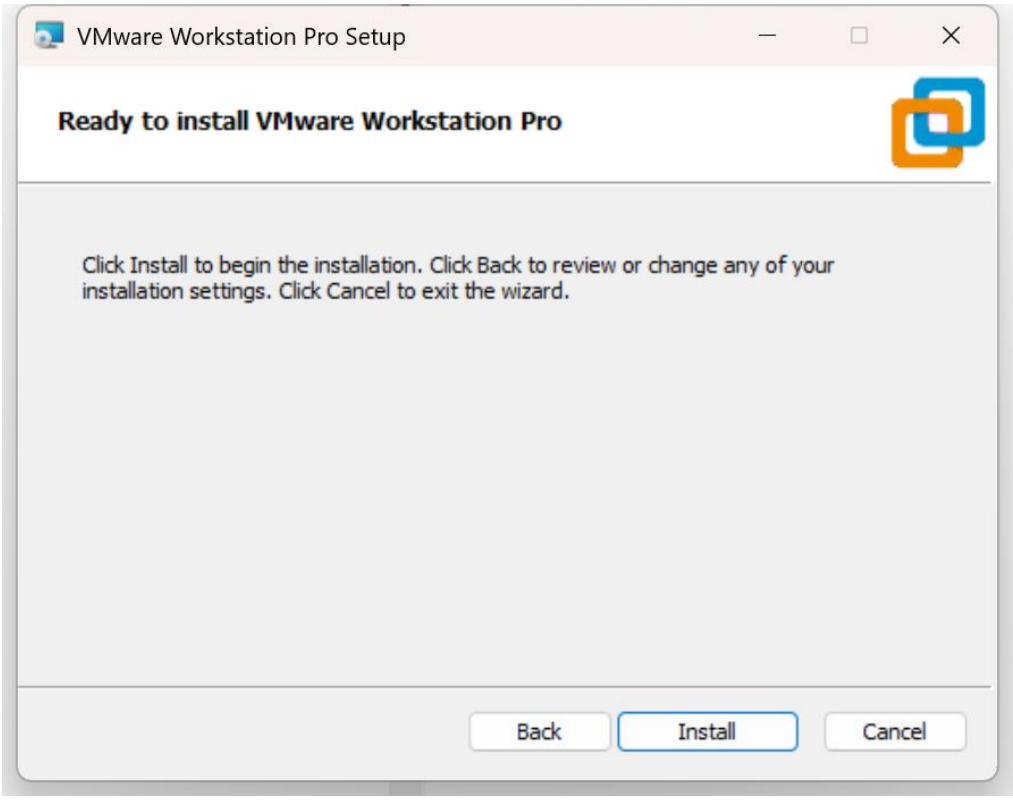
4.Based on user preference, select whether to create desktop or start menu shortcuts.

Fig.14.1.1.3 Shortcuts



5.Click Install.

Fig.14.1.1.4 Ready to Install VMware Workstation Pro



6.Wait for the installation to complete.

7.After installation, two virtual network adapters will be automatically added to the system. They must not be deleted or their properties modified.

Fig.14.1.1.5 virtual network



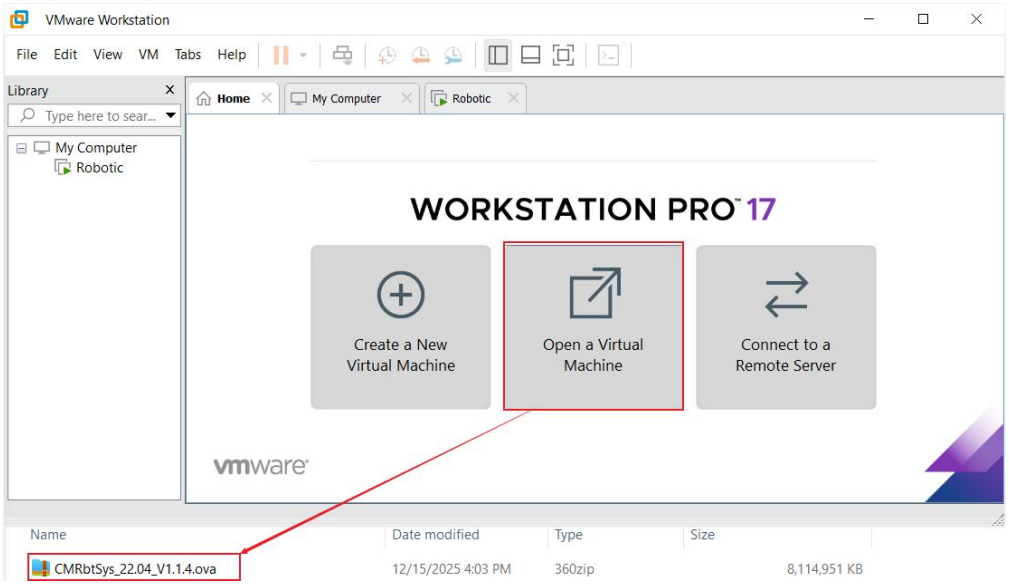
14.1.2 Loading the Virtual Machine

Place the virtual machine file provided by our company in a suitable directory on the Windows system. The virtual machine file must be version V1.1.4 or higher. Please contact our technical support to obtain the latest available virtual machine file. This document uses CM*RbtSys_22.04_V1.1.4.ova* as an example for description. The virtual machine file is approximately 7 GB in size. After being loaded into VMware, it will automatically generate files totaling around 23 GB.



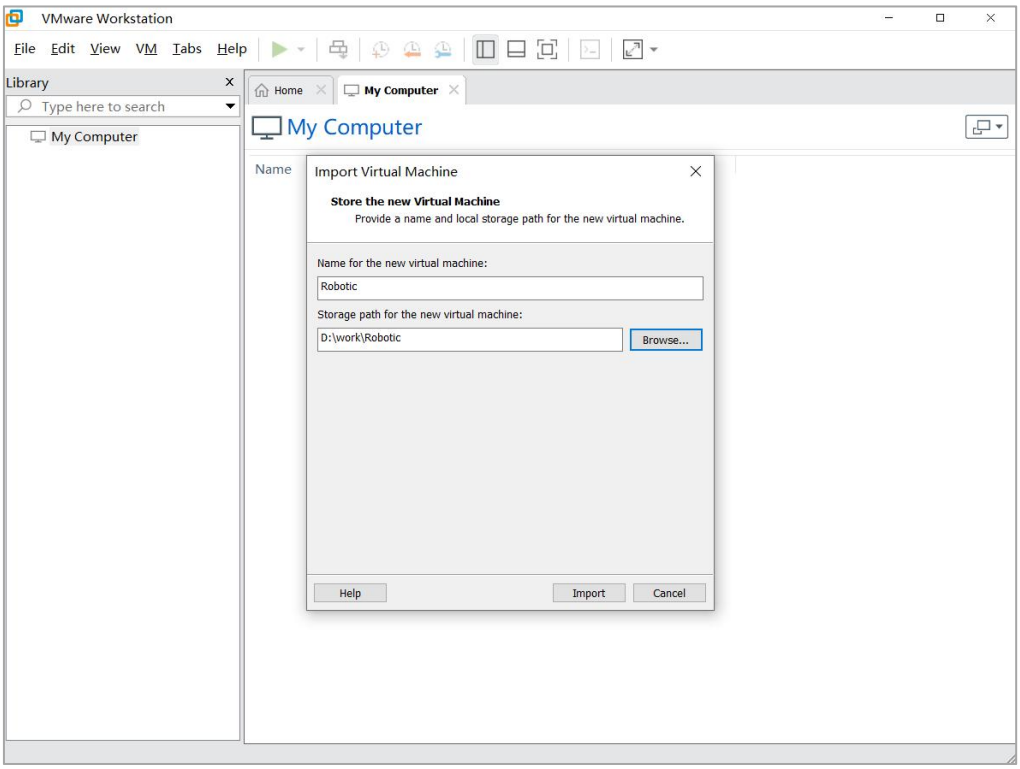
1. Open VMWare and click Open a Virtual Machine, then select the virtual machine package file and click Open

Fig.14.1.2.1 Import Virtual Machine Package



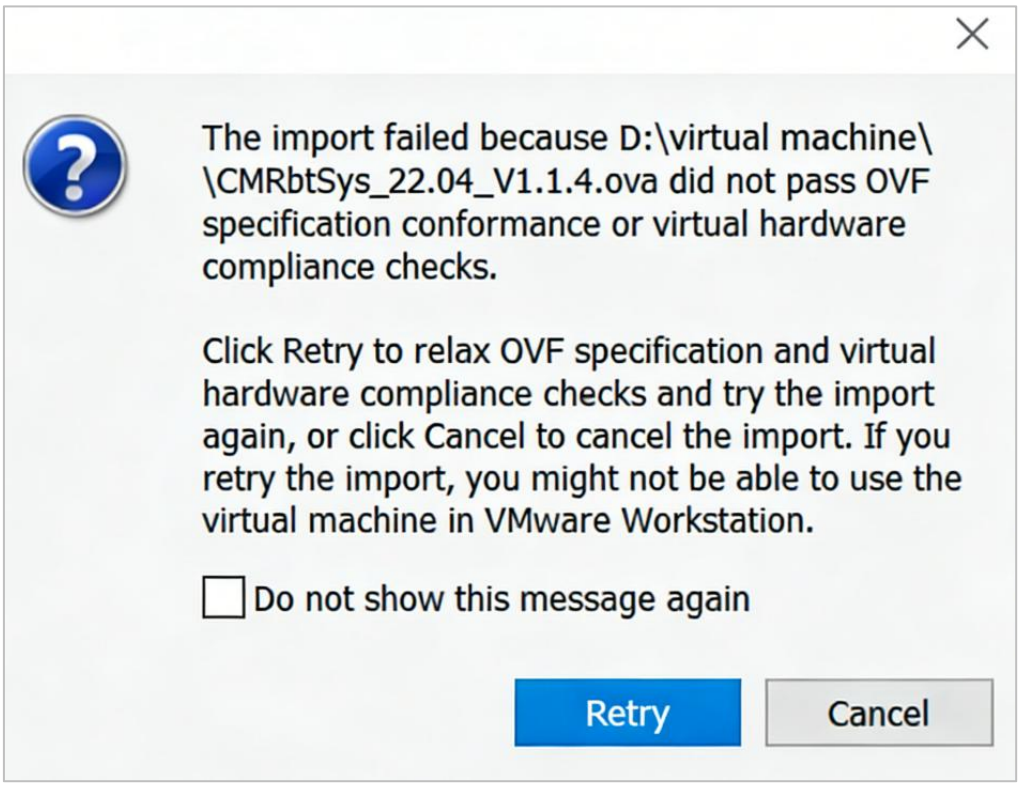
2. Assign a name and storage path for the new virtual machine, then click Import to begin the import process.

Fig.14.1.2.3 Set VM Name and Path during Import



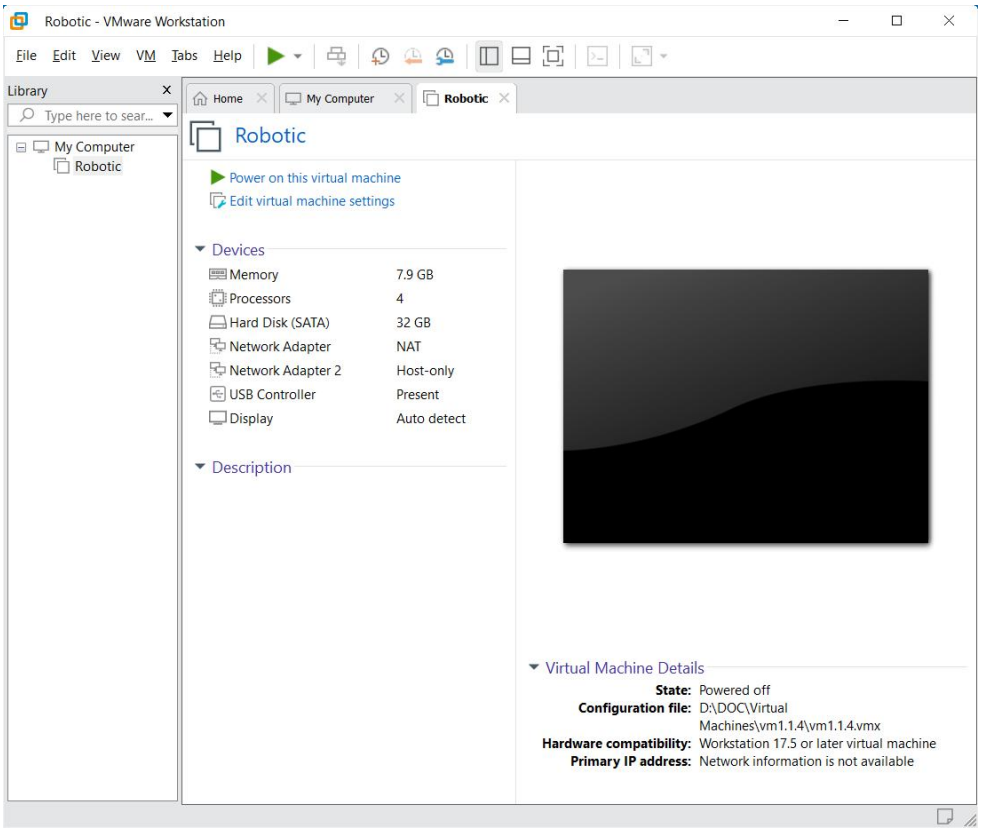
3. During the import process, a prompt similar to the following may appear. Click "Retry" to proceed.

Fig.14.1.2.4 OVF Compliance Error Dialog



After a successful import, the newly imported virtual machine will be visible under "My Computer".

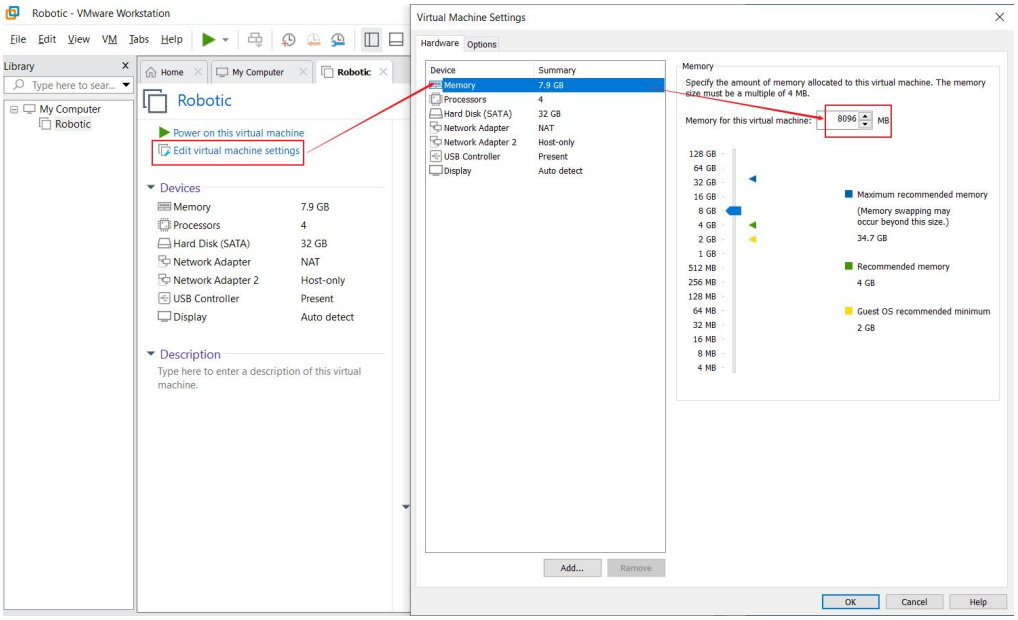
Fig.14.1.2.5 My Computer



4. After the import is complete, select the robot virtual machine and click "Edit Virtual Machine Settings" to configure its runtime environment:

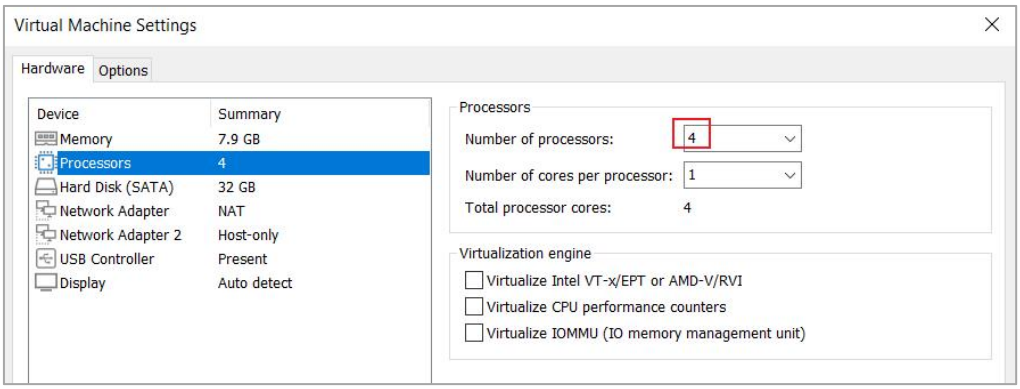
– Set the virtual machine's memory (RAM). A minimum of 8 GB is recommended.

Fig.14.1.2.6 Edit Virtual Machine Settings



– Set the number of CPUs. 4 cores are recommended.

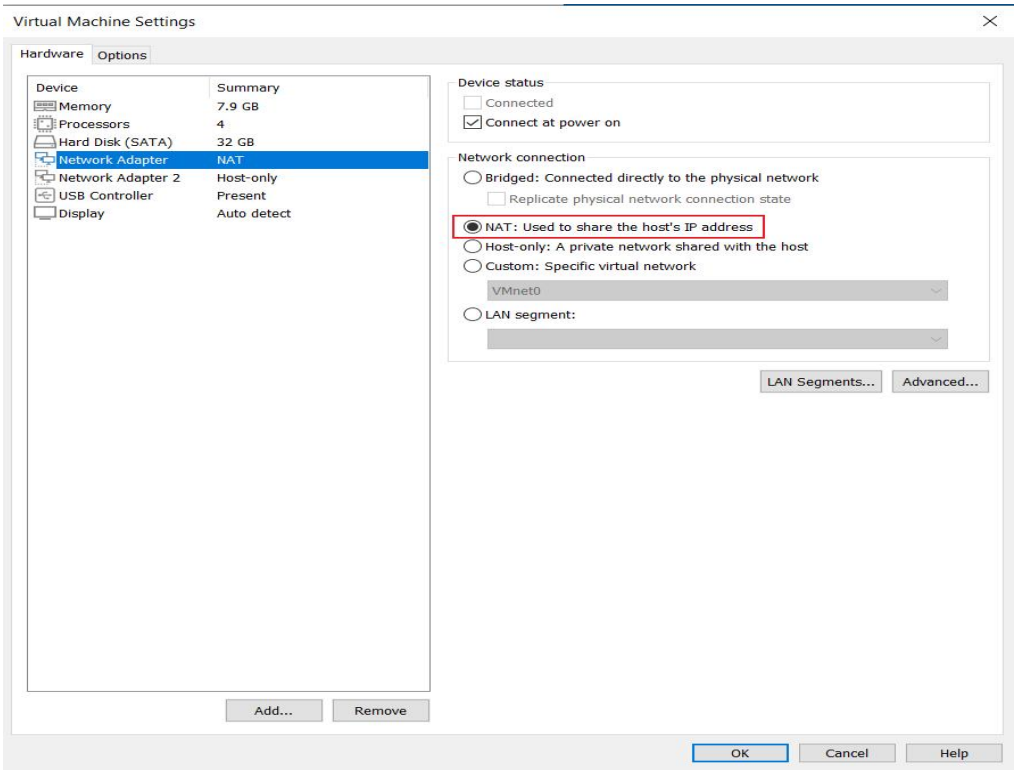
Fig.14.1.2.7 Set the number of CPUs

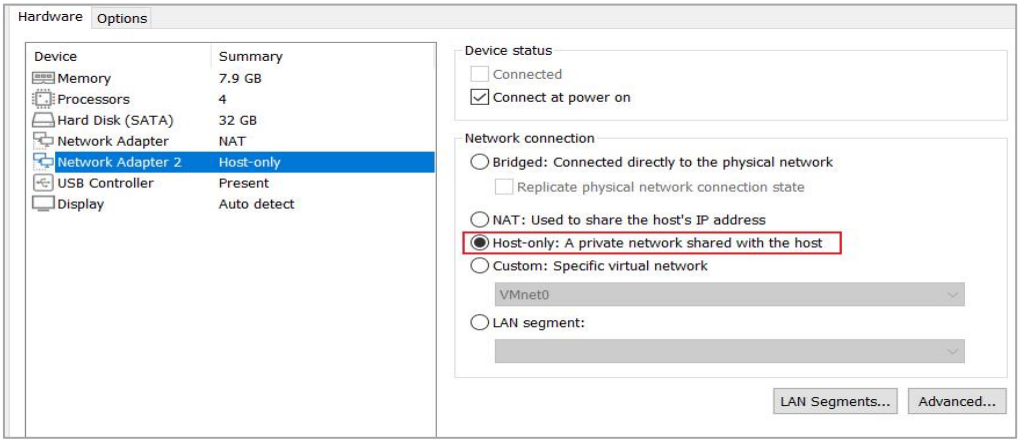


– Configure the virtual machine's network adapters.

For "Network Adapter", select "NAT" mode.
For "Network Adapter 2", select "Host-Only" mode.

Fig.14.1.2.8 Configure network adapters





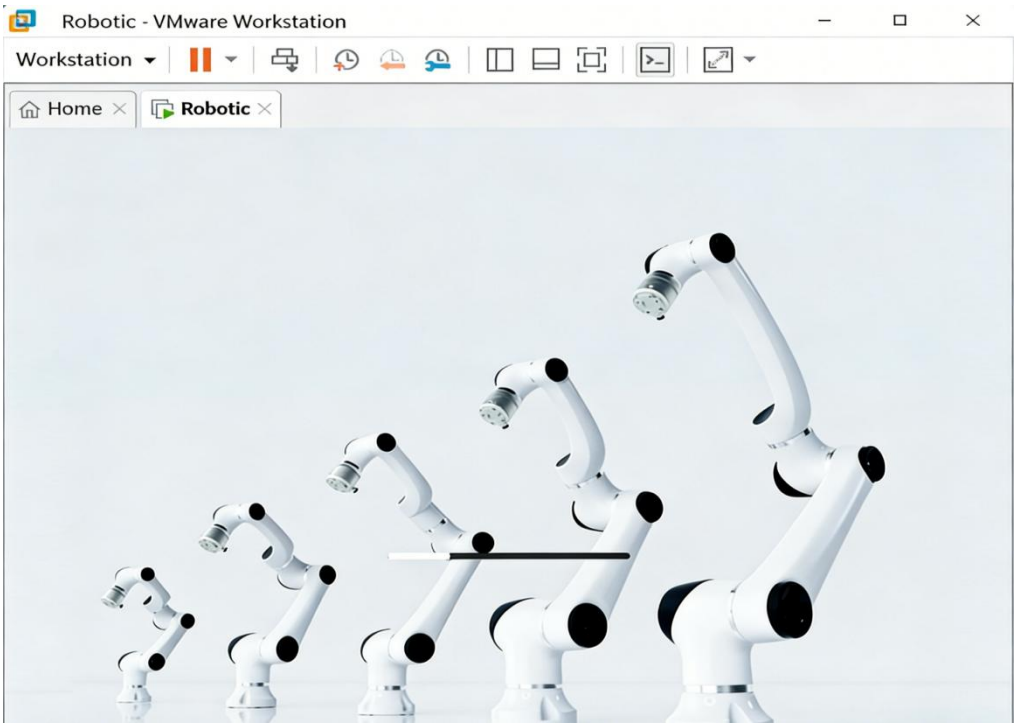
5. Click "Power on this virtual machine".

Fig.14.1.2.9 Power on this virtual machine



6. Wait for the virtual machine to finish booting.

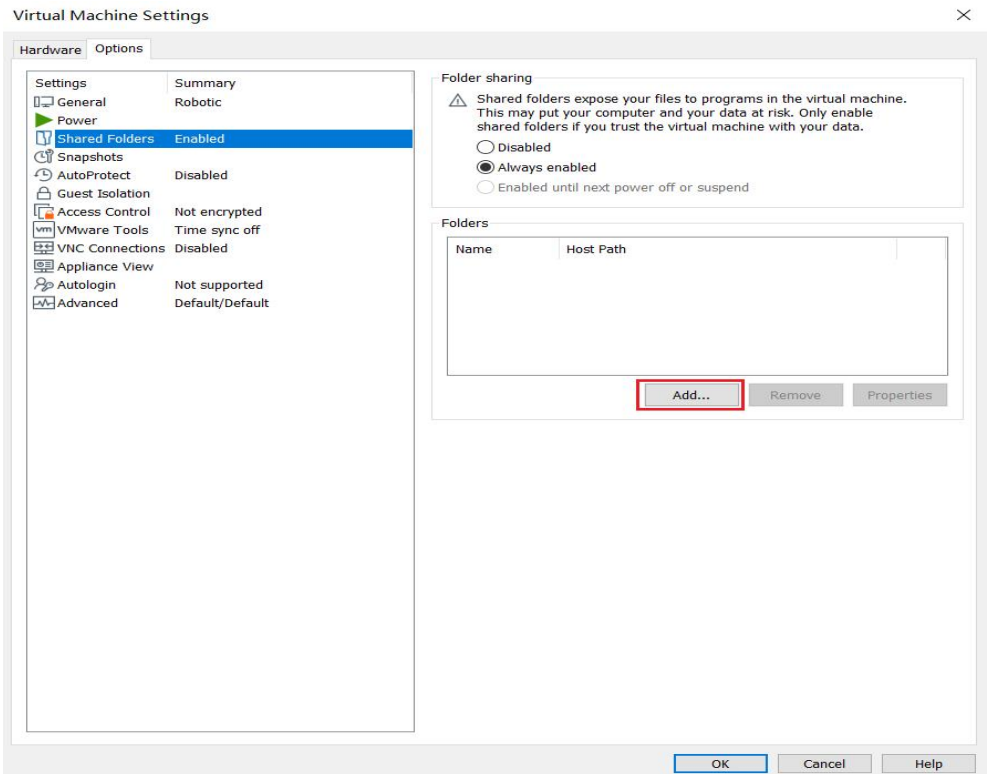
Fig.14.1.2.10 Start Virtual Machine



7. Configuring Shared Folders

Shared folders are directories accessible from both within the virtual machine system and the host system. After starting the virtual machine, navigate to the VMWare menu bar:VM(Virtual Machine)→ Settings → Options → Shared Folders. Click"Add...".

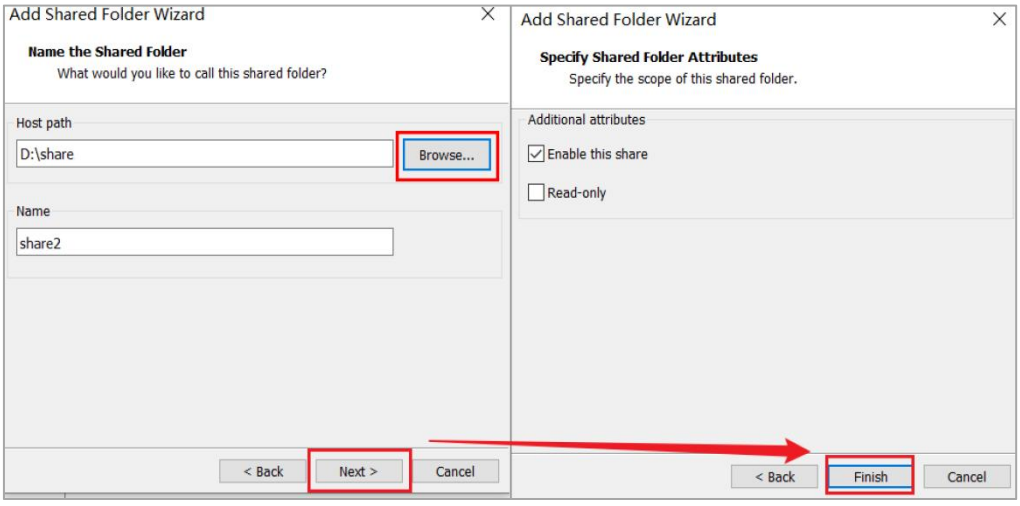
Fig.14.1.2.11 Configuring Shared Folders



Follow the steps below to add a shared folder:

Fig.14.1.2.12 Add Shared Folders Wizard

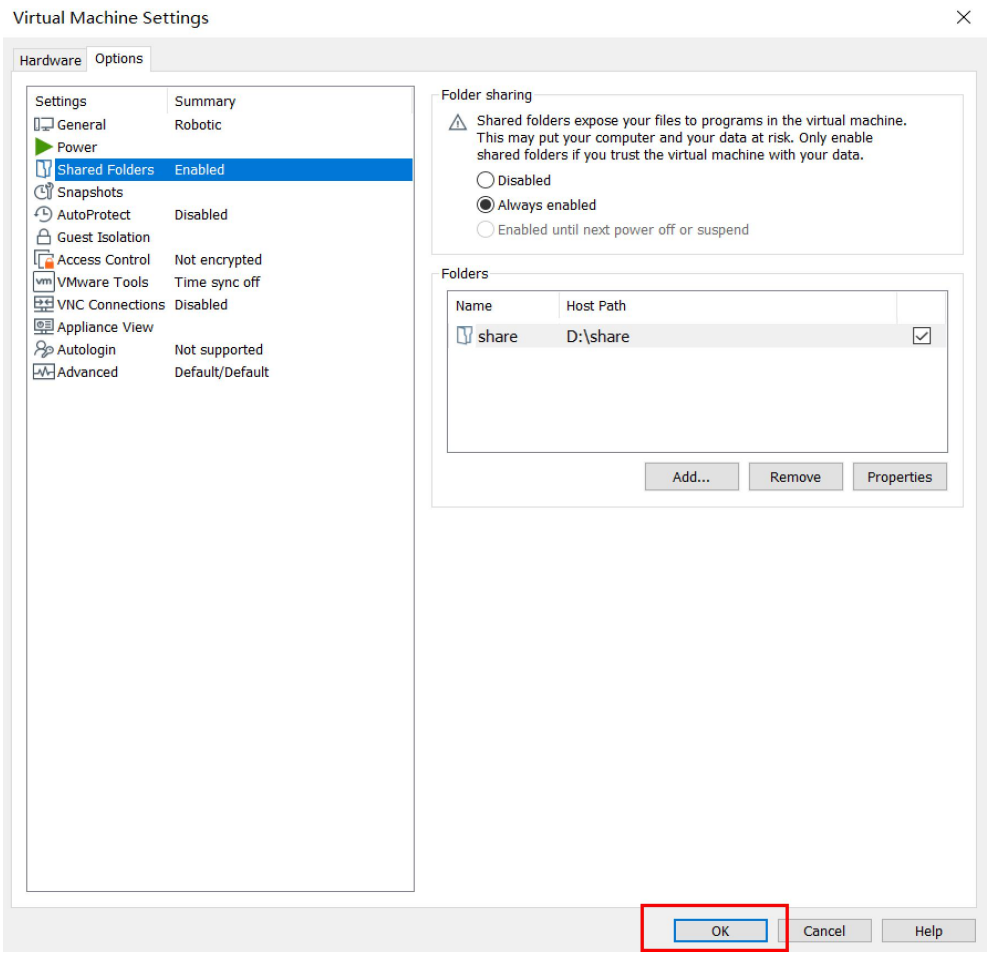




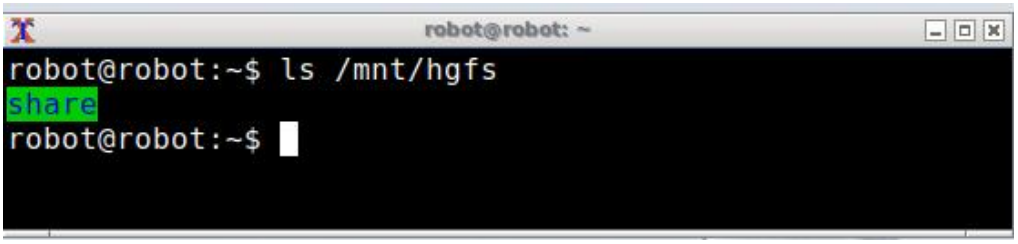
You can specify a custom Name for the folder.

Finally, click "OK" to confirm.

Fig.14.1.2.13 Configuring Shared Folders



Inside the virtual machine, you can access the shared folder under the /mnt/hgfs directory.



14.1.3 Logging into the Virtual Robot Control System

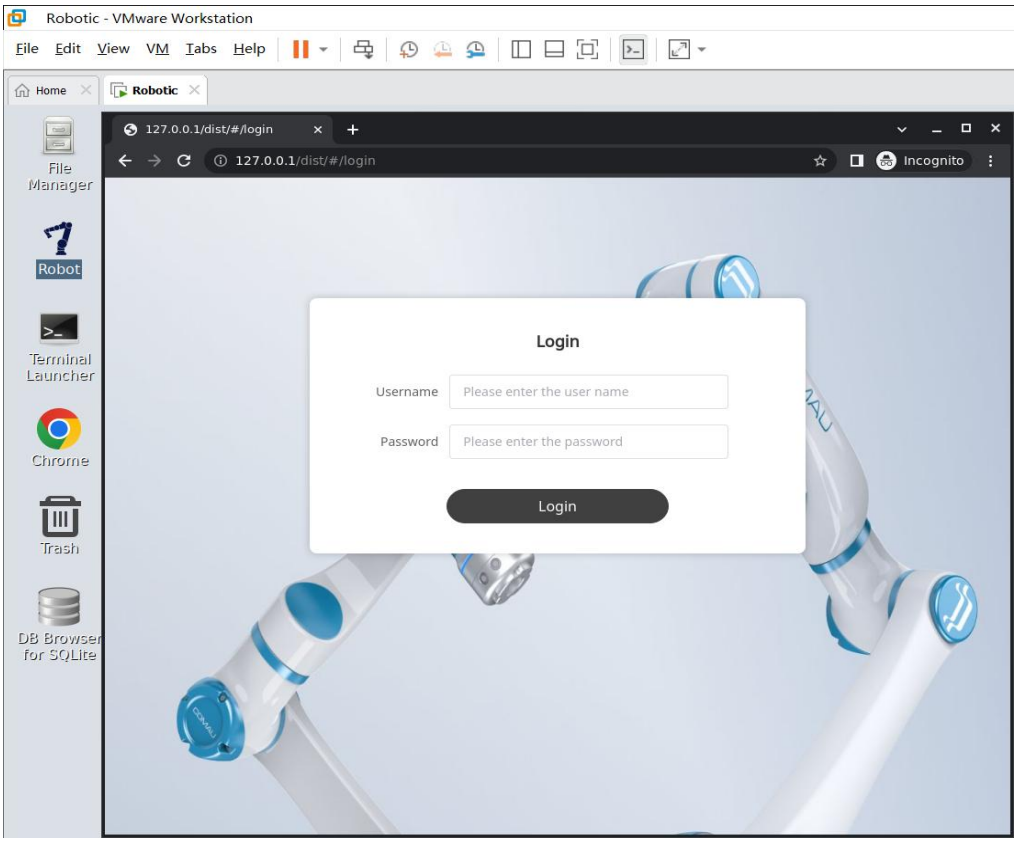
1. Inside the virtual machine, select Robot and run it:

Fig.14.1.3.1 Run Robot Controller



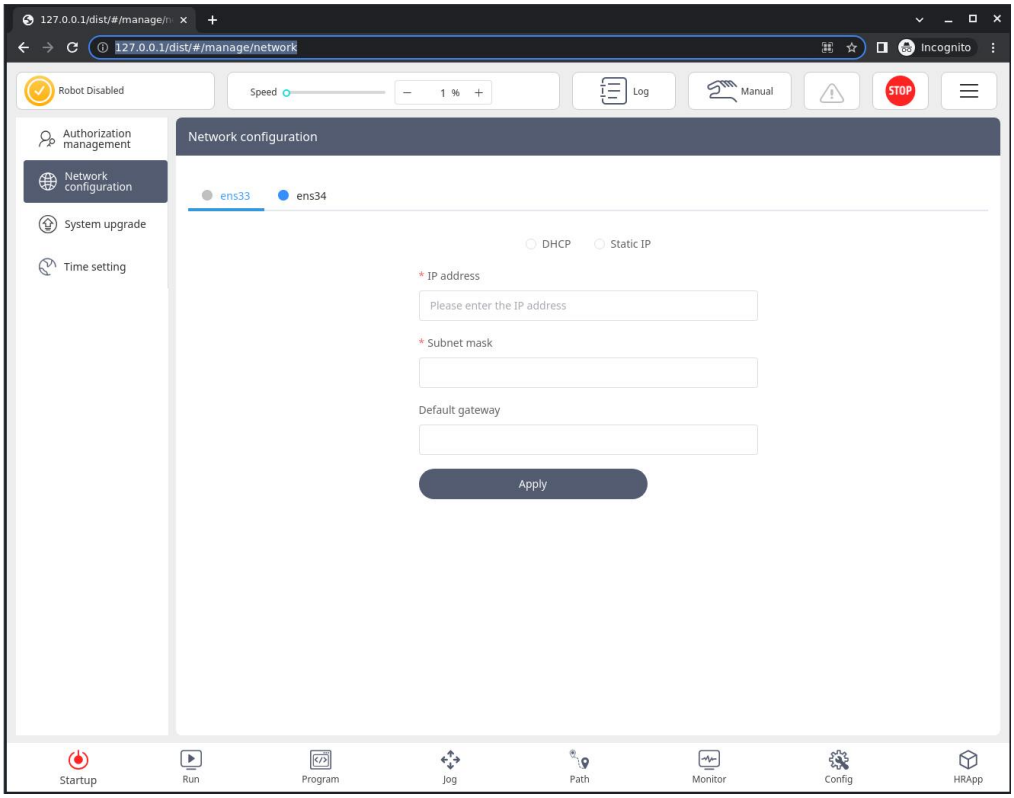
2. The browser inside the virtual machine will display the robot login page. After logging in, the user can operate the virtual robot through the software interface. The default username and password are admin/admin.
   If login fails, please contact the supplier or technical support personnel.

Fig.14.1.3.2 Login



3. Check the IP address of the virtual machine.
   After logging in, go to System Menu > Administration > Network to view the IP address of the virtual machine network adapter. If no IP address is obtained, the following information will be displayed:

Fig.14.1.3.3 View Virtual Machine IP Address

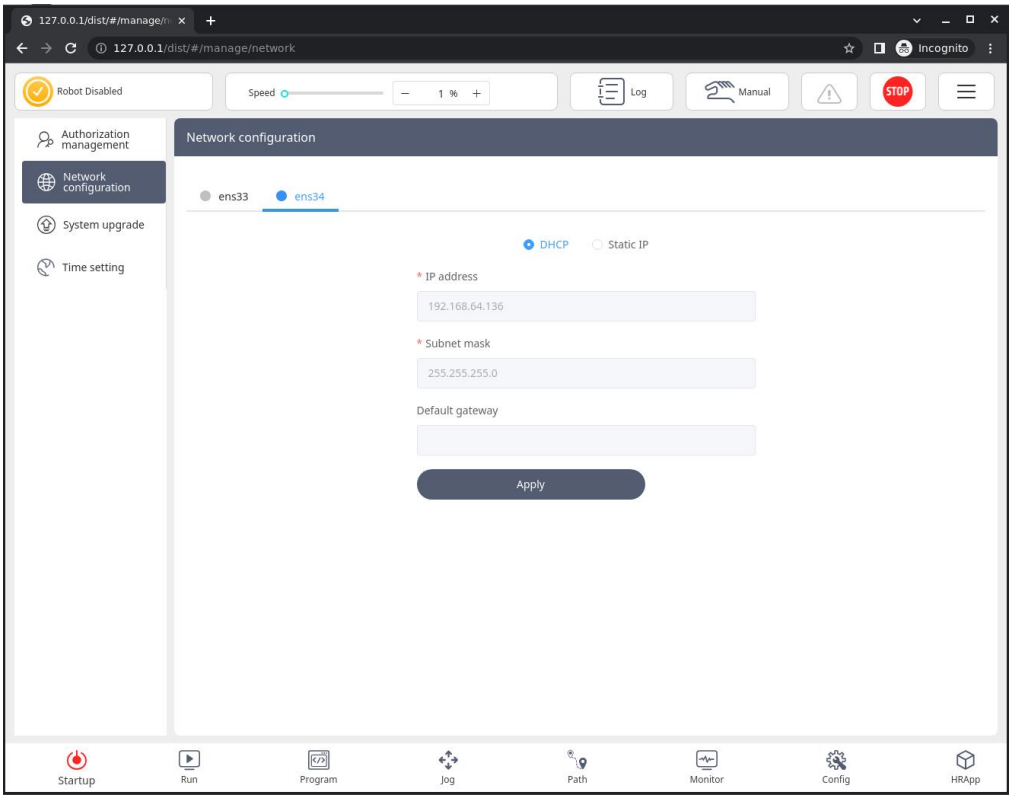


Then, you can handle it as follows:

- Change the IP configuration of the virtual machine to Static IP, fill in an arbitrary IP address and subnet mask, and then switch it back to DHCP;
- Click Apply and restart the virtual machine system.

4. Obtain the virtual machine's IP address.
   Select Stack Menu → Management → Network configuration to view the IP address.

Fig.14.1.3.4 View Virtual Machine IP Address



5. Remotely log in to the virtual machine.

   Launch a browser (e.g., Chrome, Edge) on the virtual machine host system, enter http://[IP]/dist in the address bar, where [IP] can be either of the two IP addresses of the virtual machine, for example, 192.168.64.136.

Note: If remote login fails with this IP address, try using the other one instead.

14.2 Simulation Control Box

Our company provides the RobotSimBox.exe simulation control box software, which can be used to monitor and debug the virtual robot’s control box.
This software works only with the virtual robot—do not use it to connect to a real robot.

After starting the simulation control box software, enter the virtual robot’s IP address and port number.
(The default port number for the simulation control box is 20502, but it can be modified if required. For modification instructions, please contact technical personnel.)

1. After entering the virtual robot IP address and port number, click OK to start the simulation control box.

Fig.14.2.1 Enter Virtual Machine IP Address

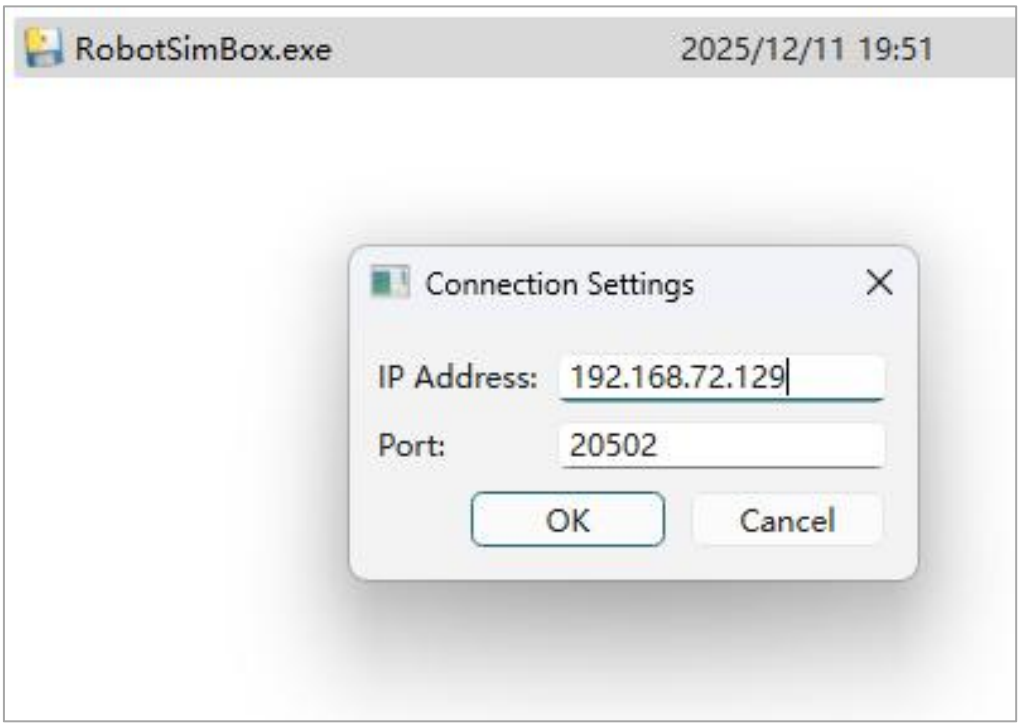
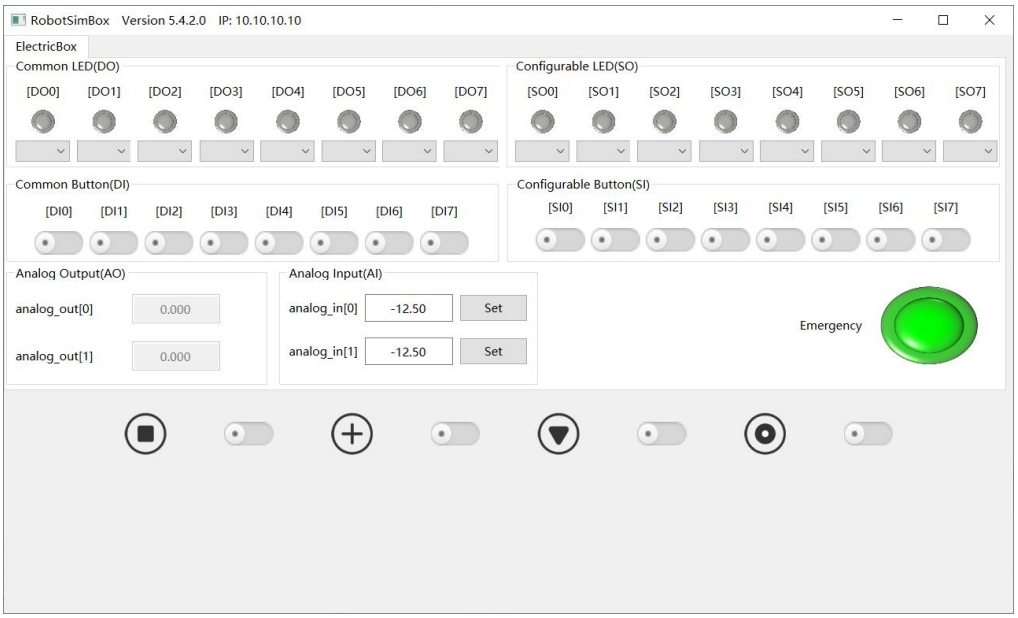
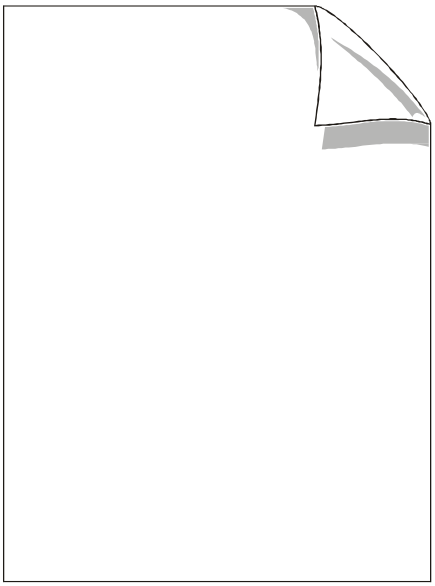
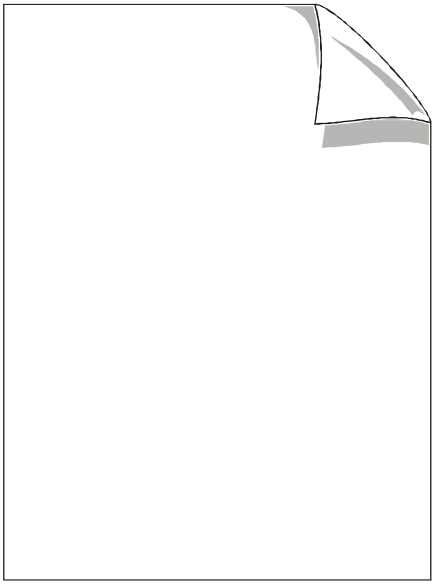


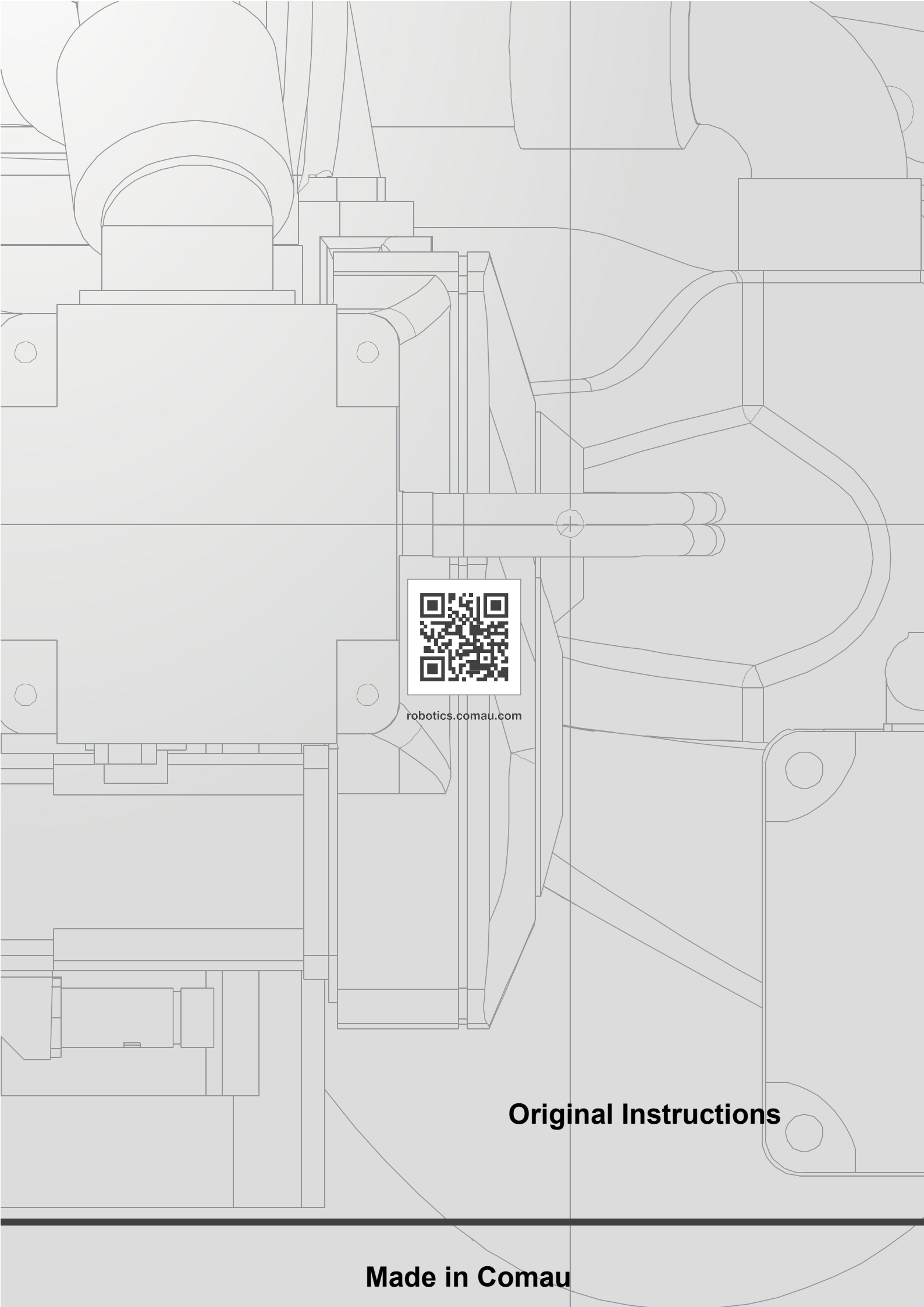
Fig.14.2.2 Simulated Control Cabinet



Users can configure input signal states and observe output port signals through this interface.







Original Instructions

Made in Comau