Collaborative Robot Elfin Series

HansRobotSoftware instruction manual

(E03/E05/E10)
(V3.5.405.521)

Introduction

How to Use This Manual

This manual is intended for operators of Elfin robots, who should have some electrical and programming knowledge. The manual will provide instructions for Elfin operators from the following ways:

- ☐Safety: The operator should keep all safety instructions in mind.
- ☐Mechanical Installation: The operator should follow the instructions when installing the robot.
- ☐Electrical interface: Open ports of Elfin are introduced for the convenience of secondary development.
- ☑Software Control: It can guide the operator to install software and run the robot.
- ☑Security Configuration: It introduces the basic safety settings.

Technical Support

Shenzhen Han's Robot Co., Ltd will provide you with long-term technical services. If you have any technical problems or other needs during using our robot, you are welcome to visit our company website: www.hansrobot.com, or directly contact our technical engineers.

Contact Information

Company Address: 5F, Block B, Han's Innovation Mansions, No. 9018 Beihuan Boulevard, High-tech Park, Nanshan District, Shenzhen, China

Factory Address: 4F, Building 4, Han's Laser Industrial Park, No.128 Chongqing Road, Fuyong Town, Baoan District, Shenzhen, China

Mobile: +86 75526983668 Fax: +86 7552698278

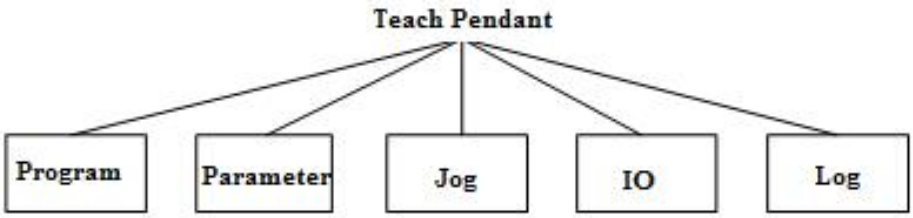
Contents

Chapter 1 HansRobot Software Operation

1.1 Software Overview

HansRobot Teach Pendant is an interface operation software that can perform a manual operation, programming, parameter setting and user monitoring of the robot. The user can control the robot to move according to the path set by the user through the operation interface function button so that the robot can complete the expected action. It can be said that the Teach Pendant is the steering wheel for controlling the movement of the robot.

The Teach Pendant is divided into the following modules:



1.2 Software startup

1) Check if DCS and Teach Pendant are both powered on and started successfully.

（1）DCS successfully launched logo：

（2）Teach Pendant successfully launched logo：

2） If the DCS and the Teach Pendant are not self-starting successfully, you need to manually start the DCS and the Teach Pendant separately.

Startup steps:

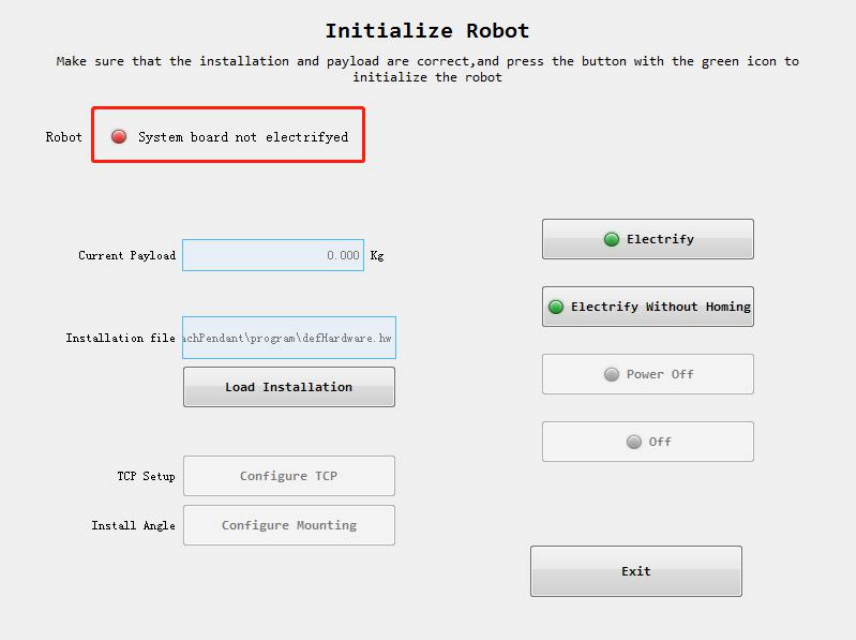
(1)Start DCS: Double-click "hmRbDCS" to start the software on the desktop；

(2)DCS successfully launched logo;

(3)Start Teach Pendant

Open the Teach Pendant software: Double-click the "HansRobot" icon to launch the Teach Pendant；

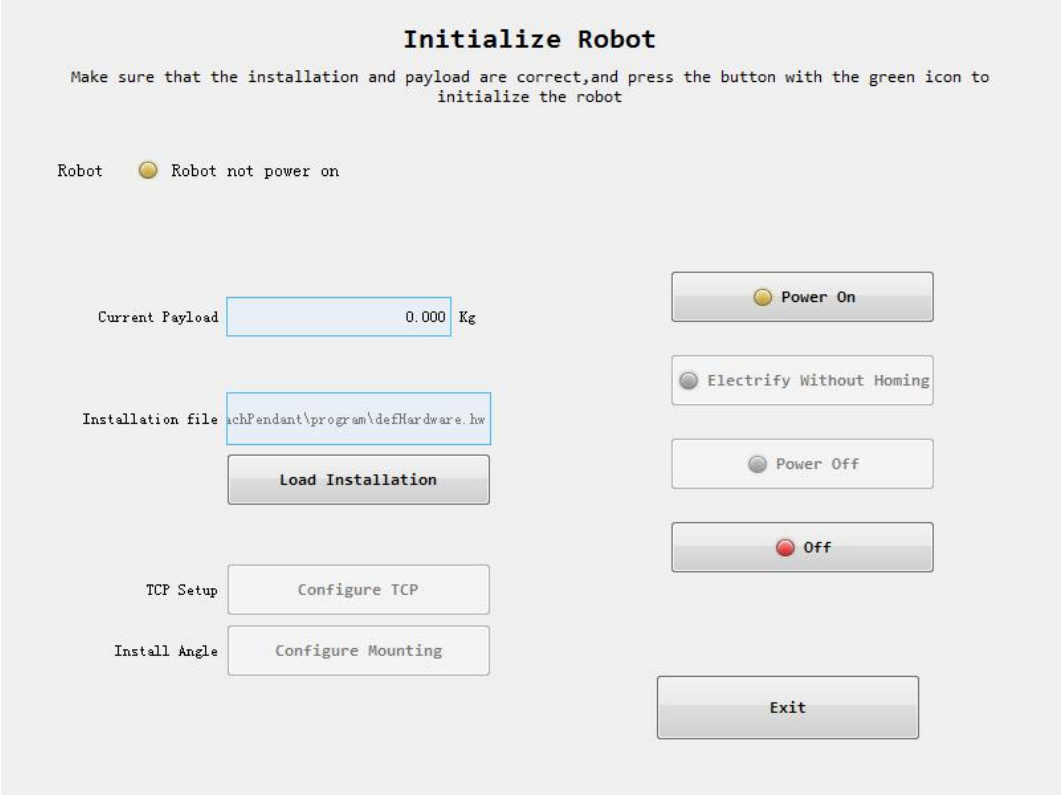
The “Initialize Robot” interface as shown in the figure below is displayed. The interface prompts “System board not electrified” and the status indicator is red. (Under normal circumstances, if the controller software is started normally, the robot is already electrified, no need to make the following Electrify steps);



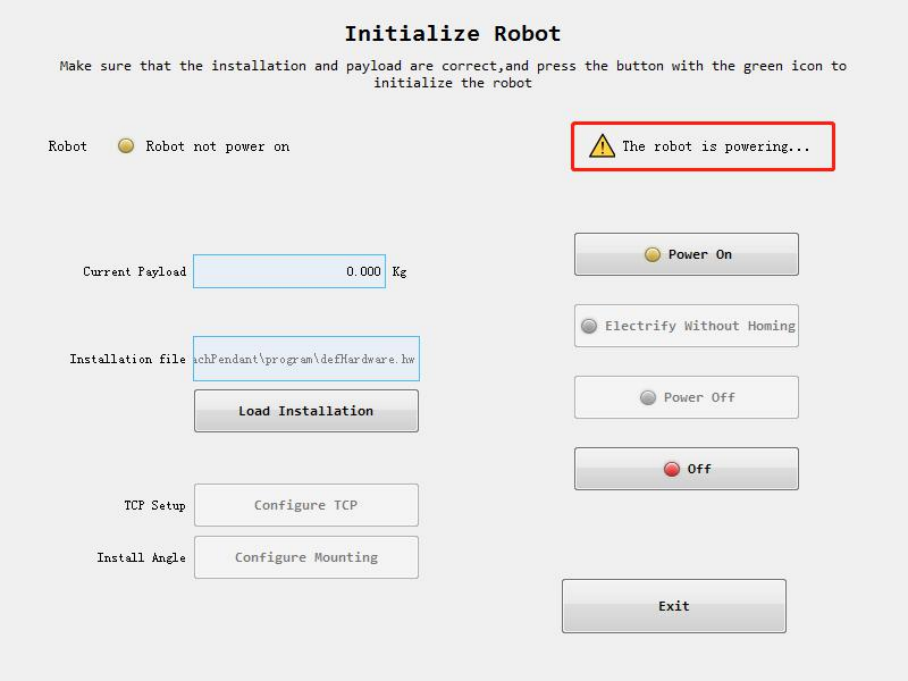
(4)click 

, the interface shows" The robot is electrifying";

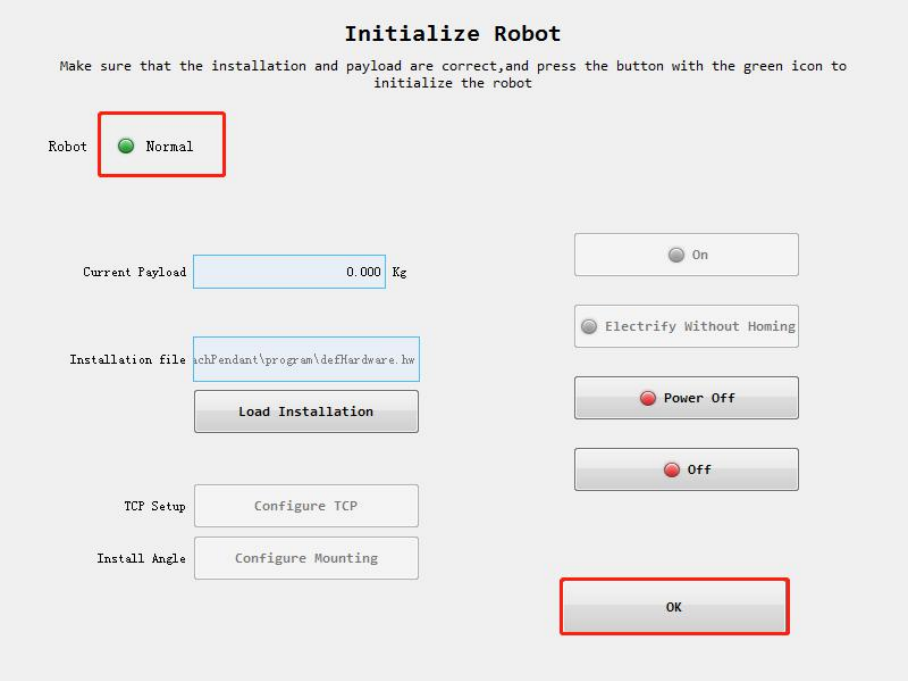
(5)after the robot is electrified successfully if the robot is still not powered on， the status indicator is Yellow， Please check whether the robot setting information (current payload, installation angle position, TCP, machine origin, etc.) is correct. If it is wrong, please change it to the correct parameter setting:



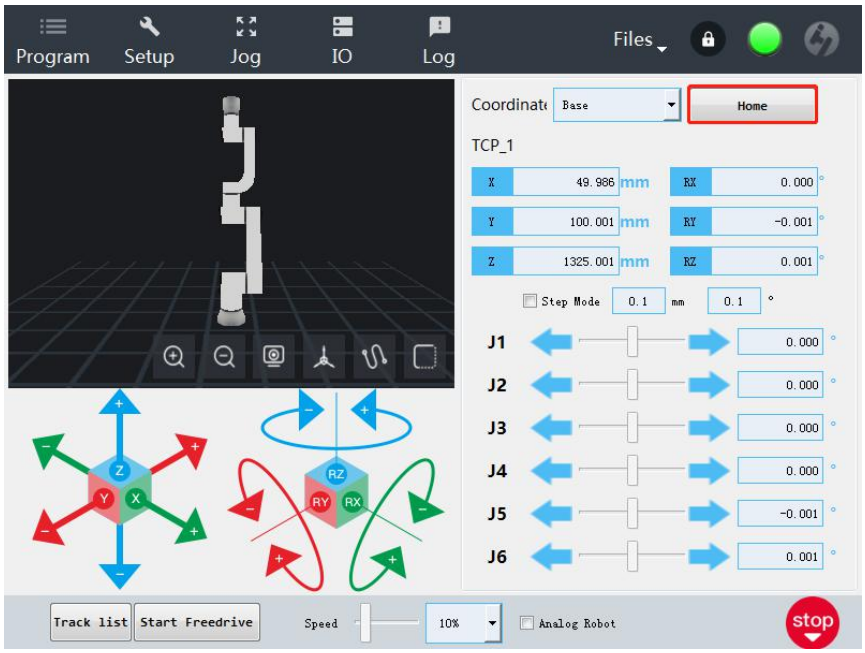
（6）After checking the robot information, click “Power on” button, the interface will prompt “The robot is powering” as shown below：



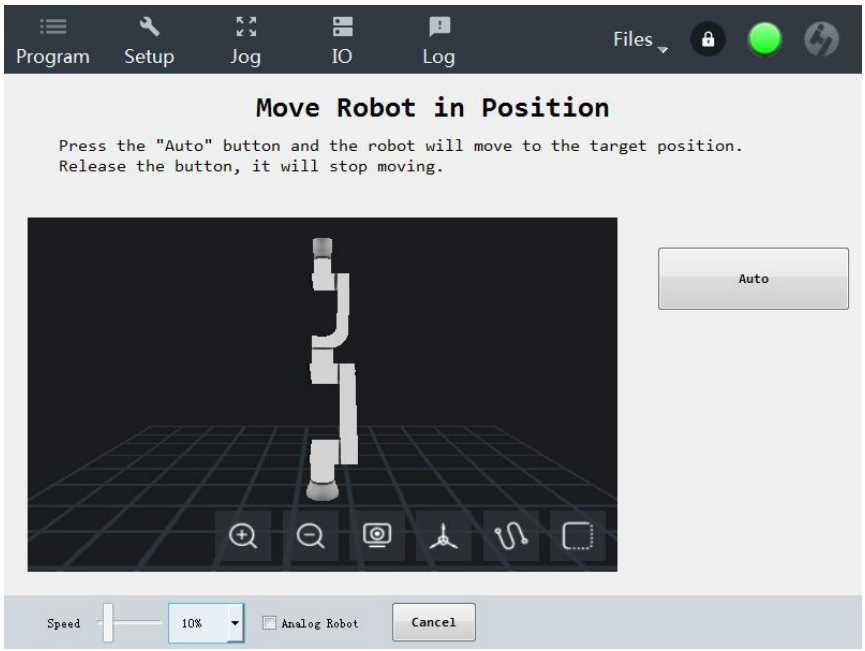
(7)The robot is successfully electrified as below interface, the indicator is green and the status is normal:



（8）Then click“OK”to control the robot. Then press the “Home” button to move the robot to the original position. The figure below shows the “Jog interface” when the robot returns to the original position. At this time, the joint angle coordinates are “0,0,0,0,0,0”.

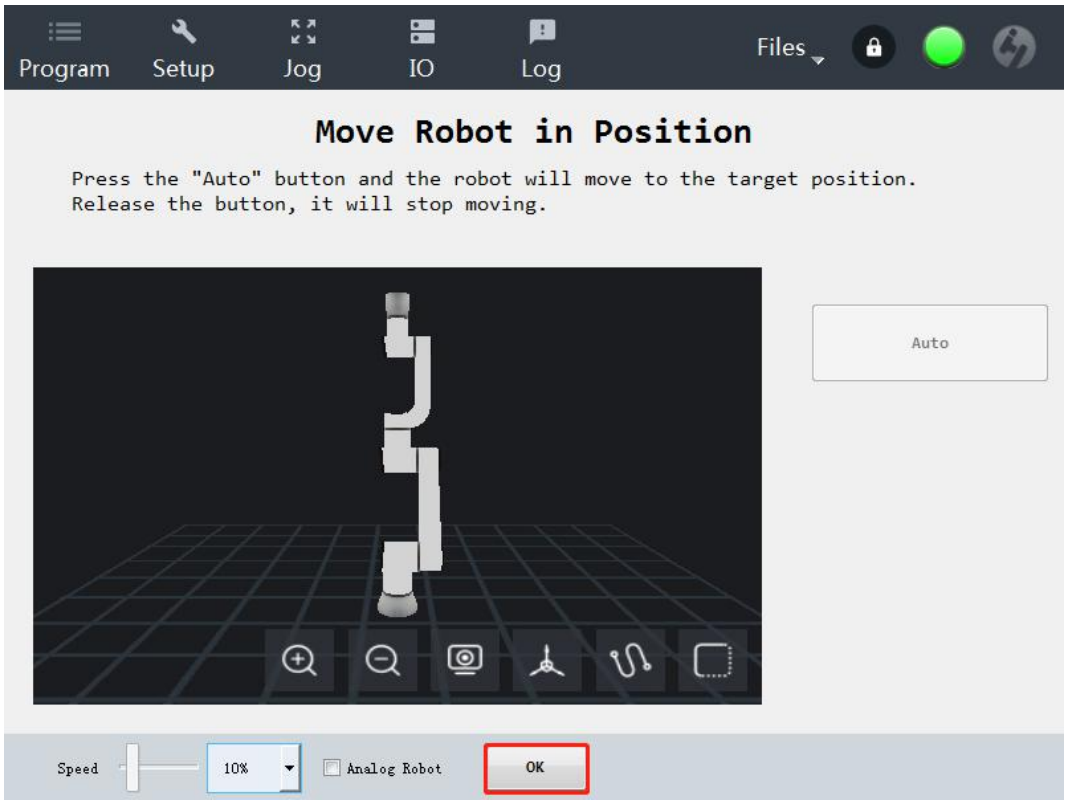


After clicking the “Home” button, the “Move robot in position” interface will be displayed, and the user can choose to press and hold the “Auto” button to the original position.



1.3 Move Robot in Position

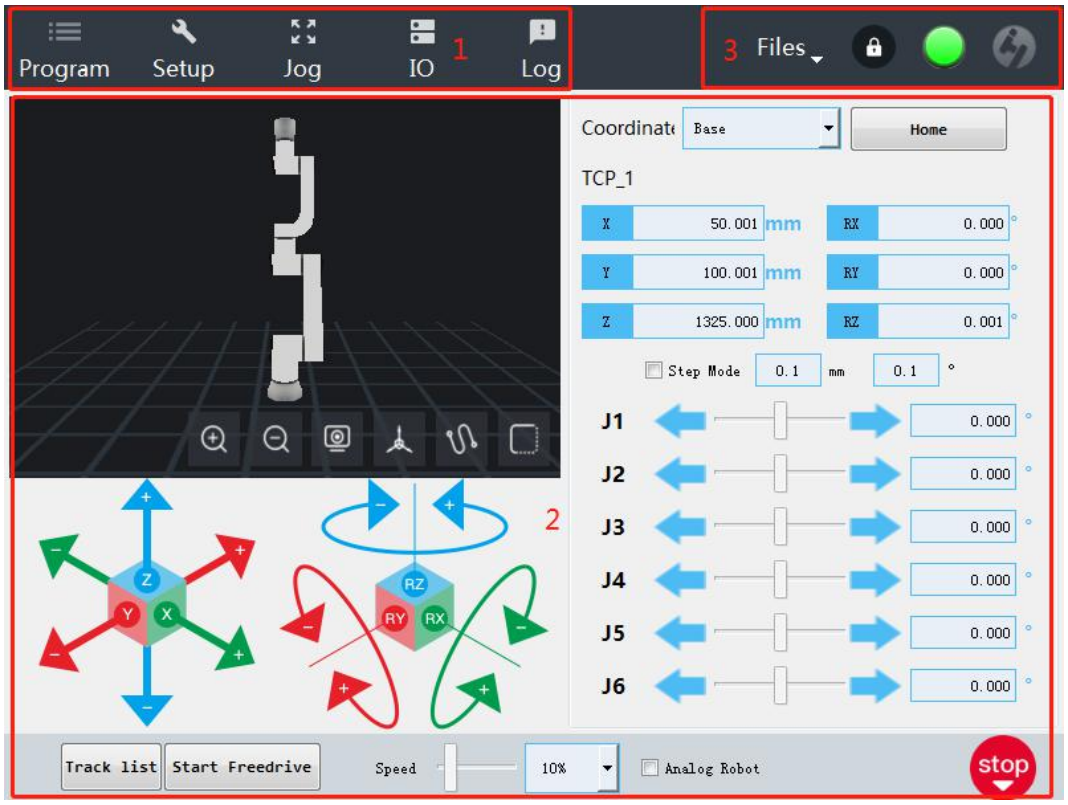
When the robot arm has to be moved to a position in its workspace, the "Move Robot in position" tab is required. For example, if the robot arm needs to move to the start position of the program before starting to run, or if the robot moves to the original position as described above, it will switch to the interface. Hold the "Auto" button to move the robot to the target position, then the "Auto" button becomes unclickable,click the "OK" button at the bottom of the interface. If you want to exit the page before moving to the target position, you can click the "Cancel" button at the bottom of the interface.



1.4 Operation Interface

There are three parts on the main interface，including：

- Functions
- Functions subpage
- Control panel

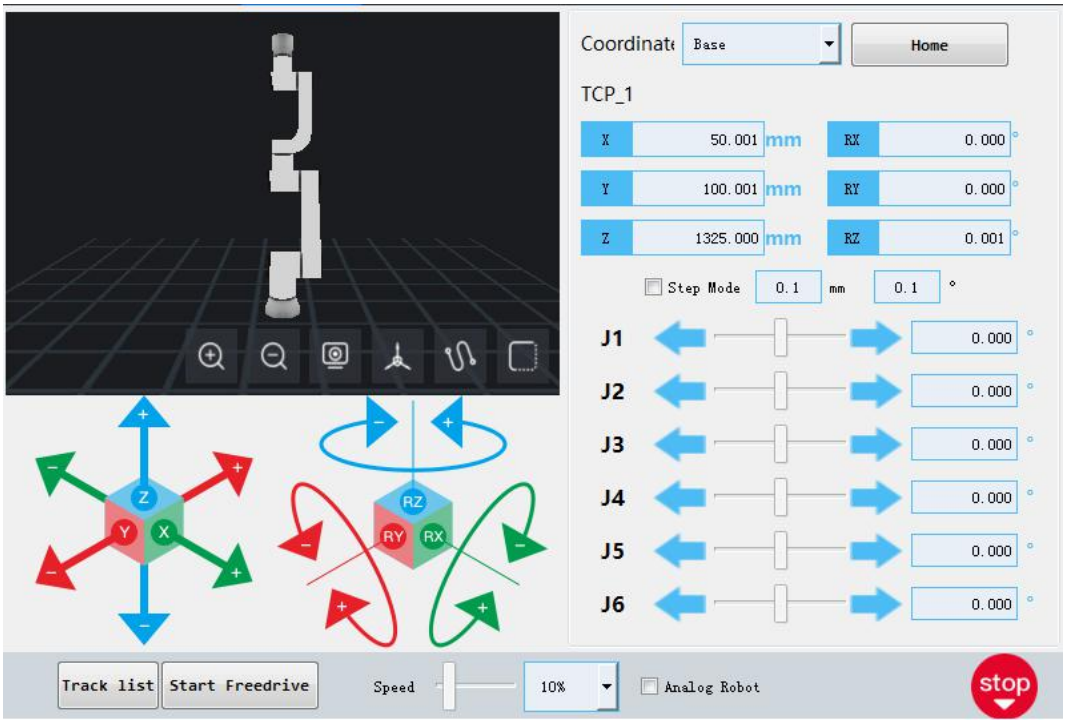


Functions

Click to switch between different function module tabs, dark blue indicates the currently selected tab

Functions subpage

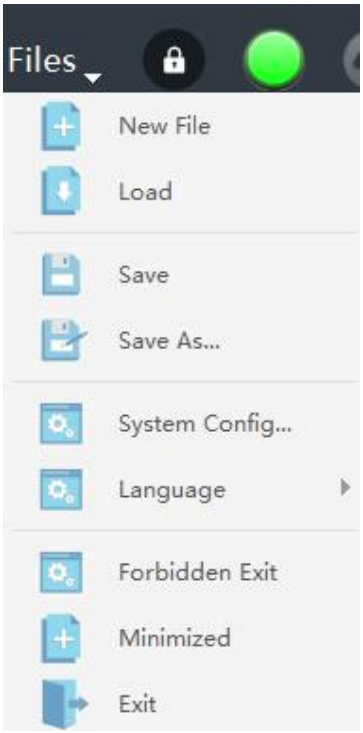
The function subpage area is a specific operation interface corresponding to the function module. As shown in the figure below, this function sub-page corresponds to the "Jog" tab of the function module.



Control panel

The control panel area contains: “Files”,” Interface Lock”, “Status Indicator”, “Version information”.

➔ Files



Click the “Files” button to display the following subdirectory, which has the following meanings:

1）New file：Create a new script to load an existing program or create an empty program.

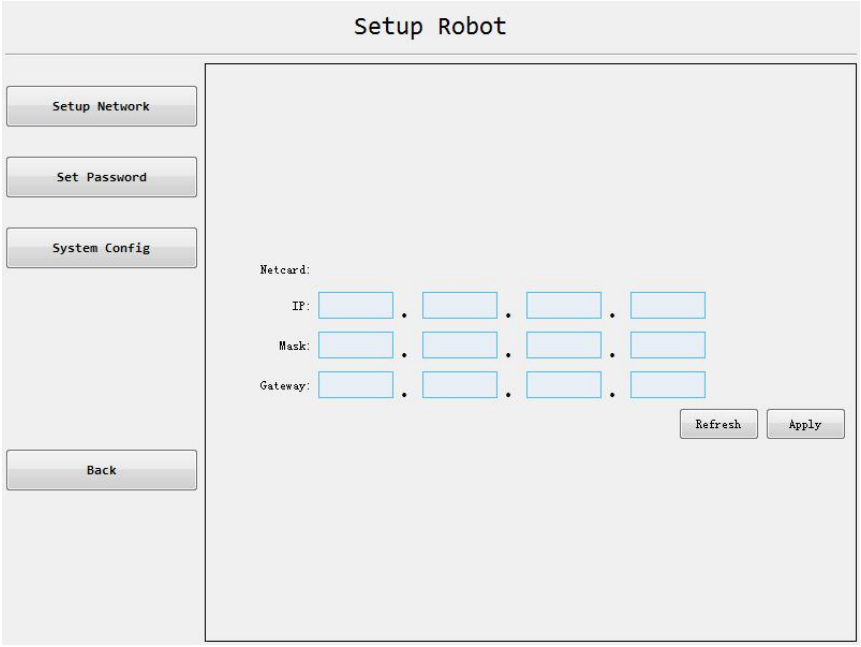
2）Load：Load an existing program.

3）Save：Save the currently edited program to the default program folder.

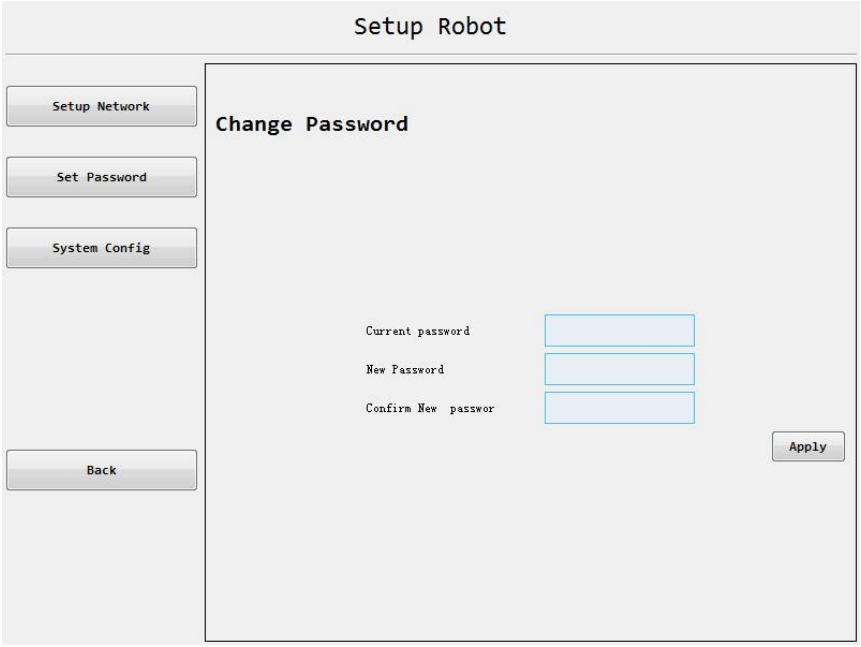
4）Save as ：Save the currently edited program to a user-defined location and rename it.

5）System configuration：Setup Network, password, etc.

a、Setup Network: Set IP to communicate with external TCP/IP. Input IP、Mask information and click the “Apply” button.

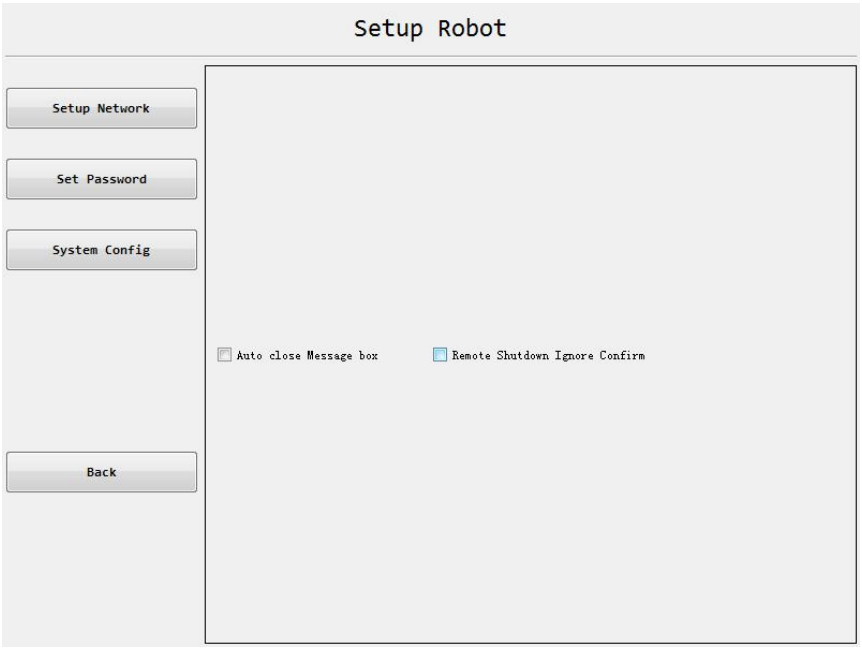


b、Set Password: change controller password, which is used for interface locks, security settings, etc., to distinguish user permissions.



C、System Config: Popup display settings."Auto close Message box"： After the pop-up window reports an error, whether the popup window is automatically closed after the function IO or interface is provided to clear the error.

Remote Shutdown Ignore Confirm： whether to confirm if the remote shutdown



6) Language: Chinese, English, German, Spanish, and Korean, etc.

7) Forbidden exit：the teach pendant will be launched next time, and the "Forbidden Exit " and "Minimized" options will not be displayed

8) Minimized： Minimize interface.

9) Exit： Restart/shutdown controller: restart the industrial computer; exit: close the teach pendant.

➔ Interface Lock

Interface lock is to prevent accidental touch and distinguish user permissions. After the interface is locked, only the stop button can be operated. The password can be changed in the system settings option in the File menu.

➔ Status Indicator

LED status indicator indicates the status of the current robot arm：

•Red: The robot is currently in a stopped state. There are many reasons for a stop.

•Yellow: The robot is not powered on and cannot be operated.

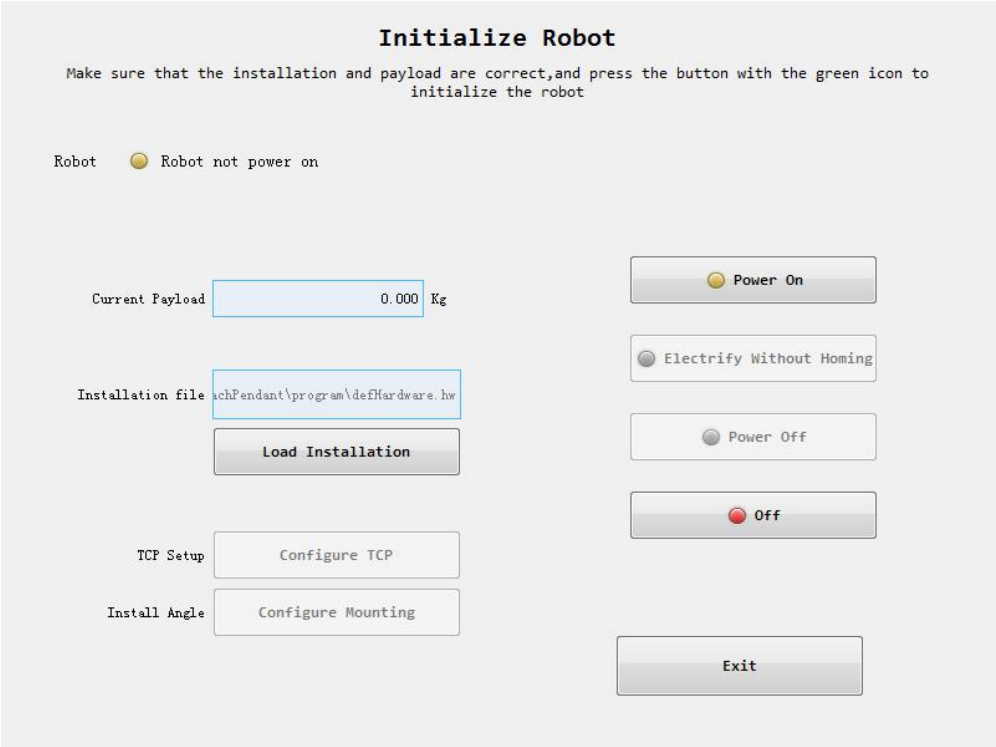
•Green: The robot is electrified and powered on, is ready for normal operation, such as movement.

•Purple: The robot is currently in a free drive state and can manually drag the robot.。

•Blue: The robot is currently in the script State and can run the script program. No other motion operations can be performed at this time.

Click the LED status indicator to enter the initialization interface.

➔ Initialization interface



Initialize the screen as shown, where you can control the initialization of the robot arm: power on/off, electrify, set the payload, load the installation settings file (.hw)

Robot arm status display LED status light，The text next to the LEDs further illustrates the current state of the robot arm.

1） If the robot is not electrified, click the “Electrify” button (the master station will be automatically started during electrifying).

2） If the master station is not started, click the “Start Master” button。

3） If the robot is not powered on, click the “Power on” button.

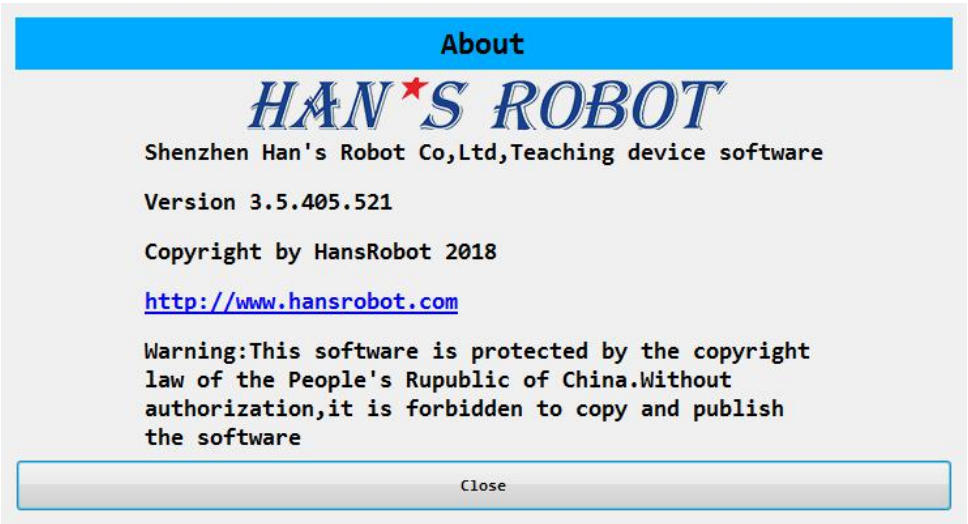
4）If the robot is out of the safe space, clear the error according to the operation prompts and restore the robot.

5） If the system board is not connected, it indicates that the selected system board information is incorrect. You need to select the correct information in the corresponding box on the DCS interface.。

6） If the controller is not initialized, it means that DCS is not connected, and to confirm whether the DCS is started.。

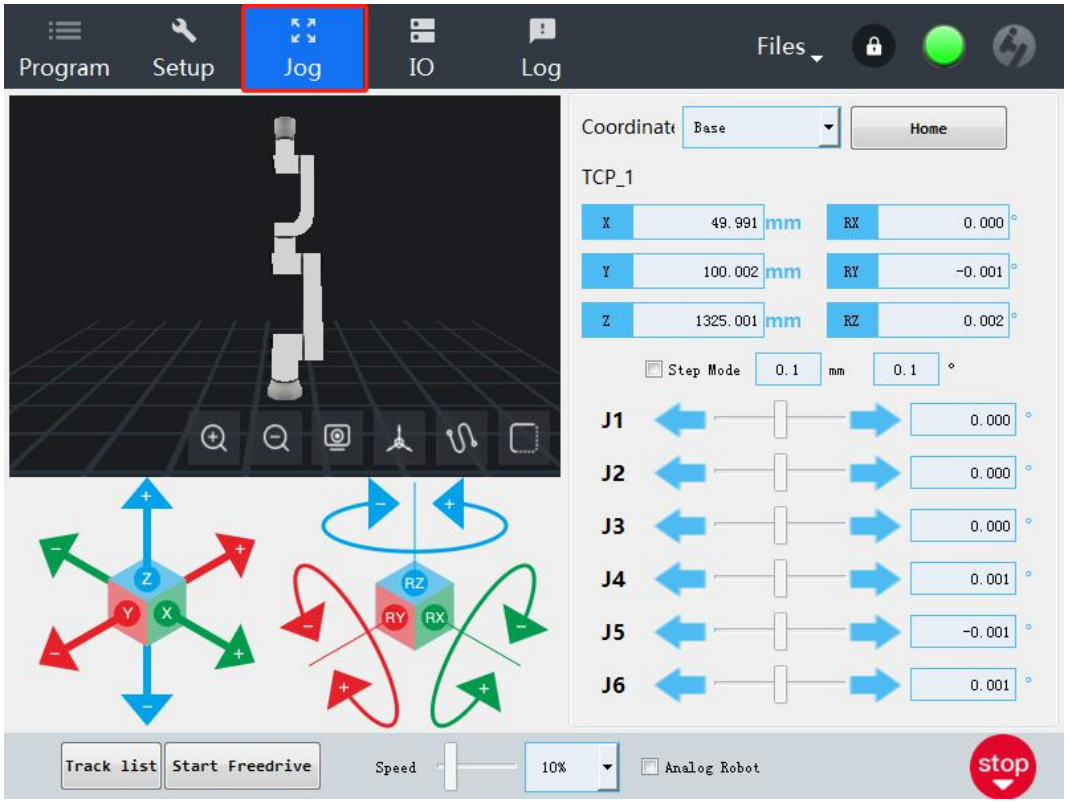
➔ Version information

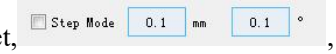
Click the LED status indicator to enter the initialization interface.



1.5 Manually controlling the robot

This section focuses on how to manually control the robot to reach the target position in the expected pose. This part of the content can be found in the "Jog" interface. The "Jog" interface is mainly used for the basic operation of robot movement, enabling the robot to perform angle and space motion and to teach teaching.



1）Step mode： Motion offset, 

, this step function takes effect only when this check box is selected, and the user can perform jog teaching at the set step offset.

2）Speed： 

The display bar shows the current rate 

ratio of the robot；Drag the square drag block on the display bar to adjust the speed ratio；Click the drop-down button in the options box to select the robot speed ratio. The speed ratio adjustment range is 1%-100%.

3）Analog Robot：Switch to the analog robot, i.e. the controlled robot is not the actual robot.

4）Coordinate： 

Defines the coordinate system currently used to control the robot. You can switch between the following coordinate system modes: Base、User, Tool. **If the correct coordinate system is not selected, the robot may appear to be inconsistent with the expected action, resulting in an unpredictable danger.**。

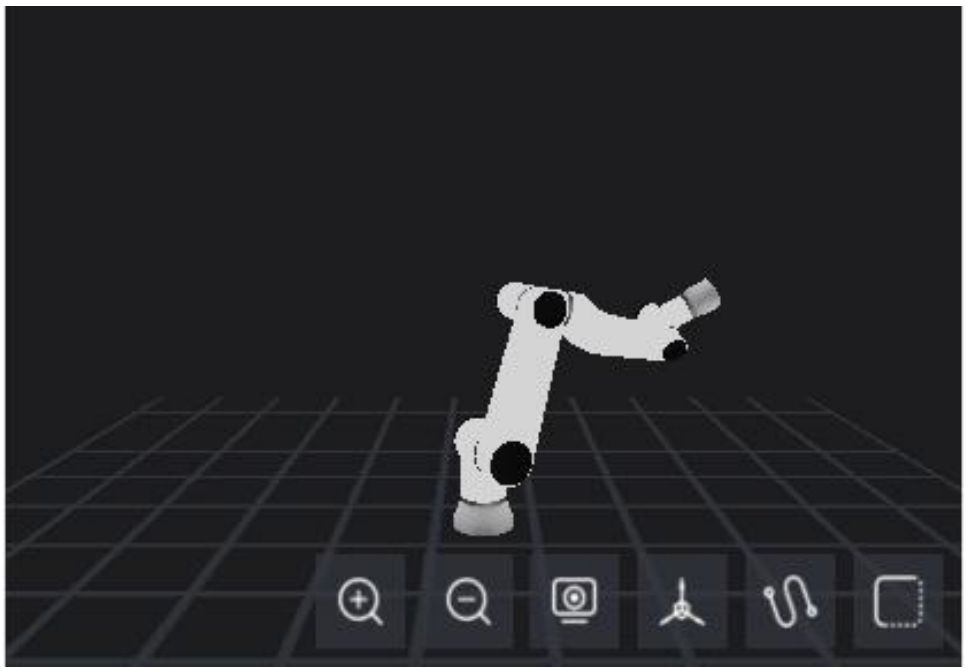
5）Home： 

Click the button to enter "Move Robot in Position" interface，hold "Auto" button， Then the body moves from the current posture to the original position (vertical state) by the joint angle motion.

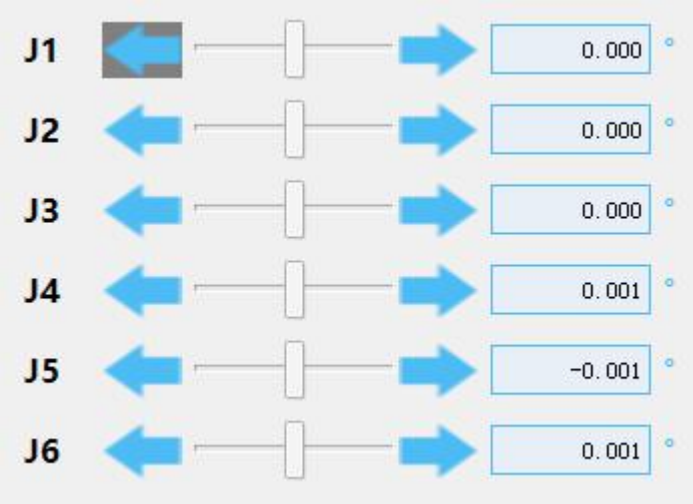
Two modes of motion, one is joint angle motion and the other is space coordinate motion.

3D Map

The robot 3D map, corresponding to the current position of the robot, press the zoom in/out mirror icon to zoom the view, and drag the finger icon to change the angle of view.



Joint angle motion



J1-J6 means the six Axles of the robot arm.

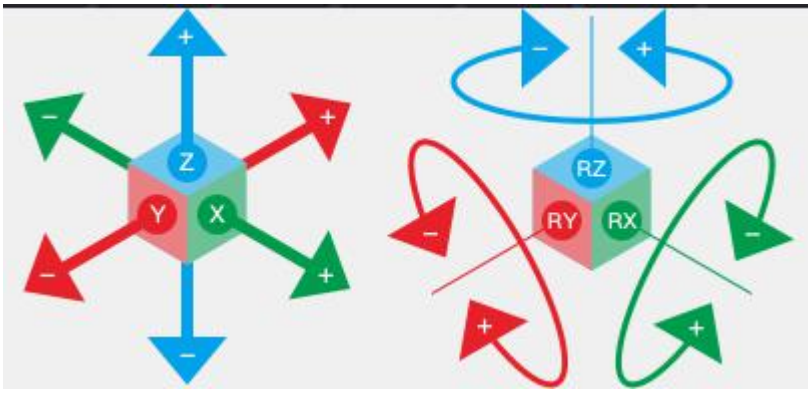
The text box displays the angular coordinate value of each corresponding axis position of the current robot, the unit is ° (angle).

1）J1-、J2-、J3-、J4-、J5-、J6- corresponding  **buttons：** In the safe space, the robot moves in the negative direction corresponding to the axis, and the text box display the current position of the corresponding axis. Each time the button is clicked, it moves 2° in the negative direction of the axis. Press and hold the button, the corresponding axis moves continuously in the negative direction, release the button to stop the movement. If the relative movement of the step is selected, the step distance of the user input corresponding to the axial negative direction is moved.

2）J1+、J2+、J3+、J4+、J5+、J6+ corresponding  **buttons：** In the safe space, the robot moves the corresponding axis in the forward direction, and the text box displays the current position of the corresponding axis. Each time the button is clicked, it moves 2° in the positive direction of the axis. Press and hold the button, the corresponding axis moves continuously in the positive direction, release the button to stop the movement, if the relative movement of the step is selected, the step distance of the user input corresponding to the axial positive direction is moved.

Note: The joint angle motion here corresponds to the MoveJ, MoveRelJ, and ACS motion modes in scripting.

Cartesian space motion

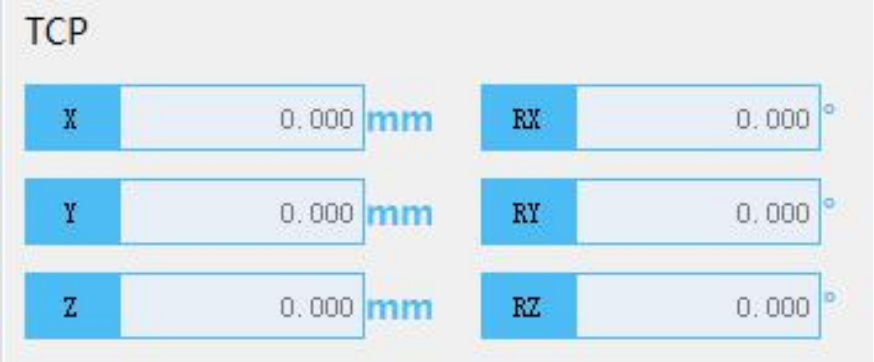


1）X-、Y-、Z- corresponding buttons： In the safe space, the robot end TCP moves in the negative direction of this component. Each time the button is clicked, the robot end TCP moves 2 mm in the negative direction. Press and hold the button, the end of the robot TCP moves continuously in the negative direction, release the button to stop the movement. If the starting step relative motion is checked, the robot end TCP moves the step distance input by the user in the negative direction.

2）X+、Y+、Z+ corresponding buttons： In the safe space, the robot moves in the positive direction of the component, and each time the button is clicked, the corresponding axial direction moves 2mm. Press and hold the button, the corresponding axis moves continuously in a positive direction, and release the button to stop the movement. If the relative movement of the starting step is checked, the step distance of the user input corresponding to the axial positive direction is selected.

3）RX-、RY-、RZ- corresponding buttons： RX- indicates that the robot end TCP rotates in the negative direction of the X-axis; RY- indicates that the robot end TCP rotates in the negative direction of the Y-axis; RZ- indicates that the robot end TCP rotates in the negative direction of the Z axis. Within the safe space, each time the button is clicked, the robot end TCP rotates 2° in the negative direction of the corresponding axis. Press and hold the button, the end TCP of the robot continues to rotate in the negative direction of the corresponding axis, and the button is released to stop the movement. If the starting step relative motion is checked, the robot end TCP rotates the step distance input by the user in the negative direction of the corresponding axis.

4）RX+、RY+、RZ+ corresponding buttons： RX+ indicates that the robot end TCP rotates in the positive direction of the X-axis; RY+ indicates that the robot end TCP rotates in the positive direction of the Y-axis; RZ+ indicates that the robot end TCP rotates in the positive direction of the Z axis. Within the safe space, each time the button is clicked, the robot end TCP rotates 2° in the positive direction of the corresponding axis. Press and hold the button, the end TCP of the robot continues to rotate in the positive direction of the corresponding axis, and the button is released to stop the movement. If the starting step relative motion is checked, the robot end TCP rotates the step distance input by the user in the positive direction of the corresponding axis.



TCP: Display the currently used tool coordinate position.

X、Y、Z、RX、RY、RZ Represents the current spatial coordinate position vector of the robot.

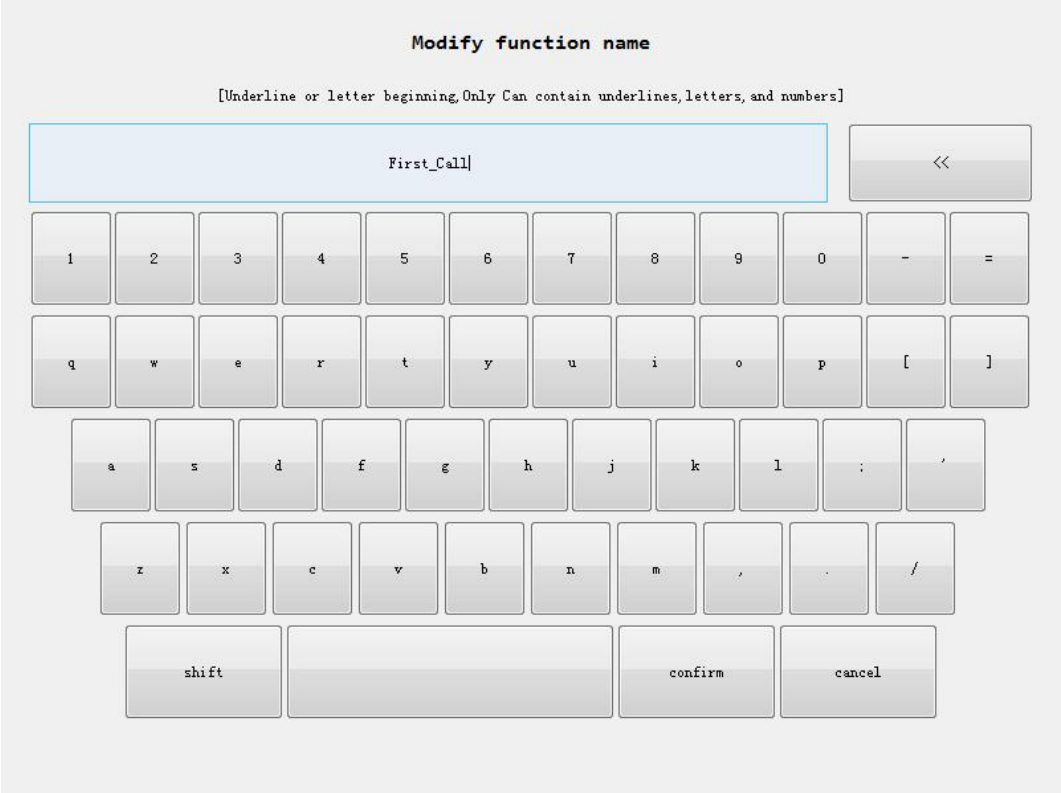
As shown in the figure above, the value displayed in the text box indicates the spatial coordinate value of the current robot end TCP relative to the currently selected feature coordinate system. The units of X, Y, and Z are mm, and the units of RX, RY, and RZ are °.

Note: The spatial coordinate motion of the jog interface corresponds to the MoveL, MoveRelL, and PCS motion modes in script programming.

Home

Click the" Home" button to move the robot to the original position. For more details, refer to chapter 7.2 (8).

Screen keyboard



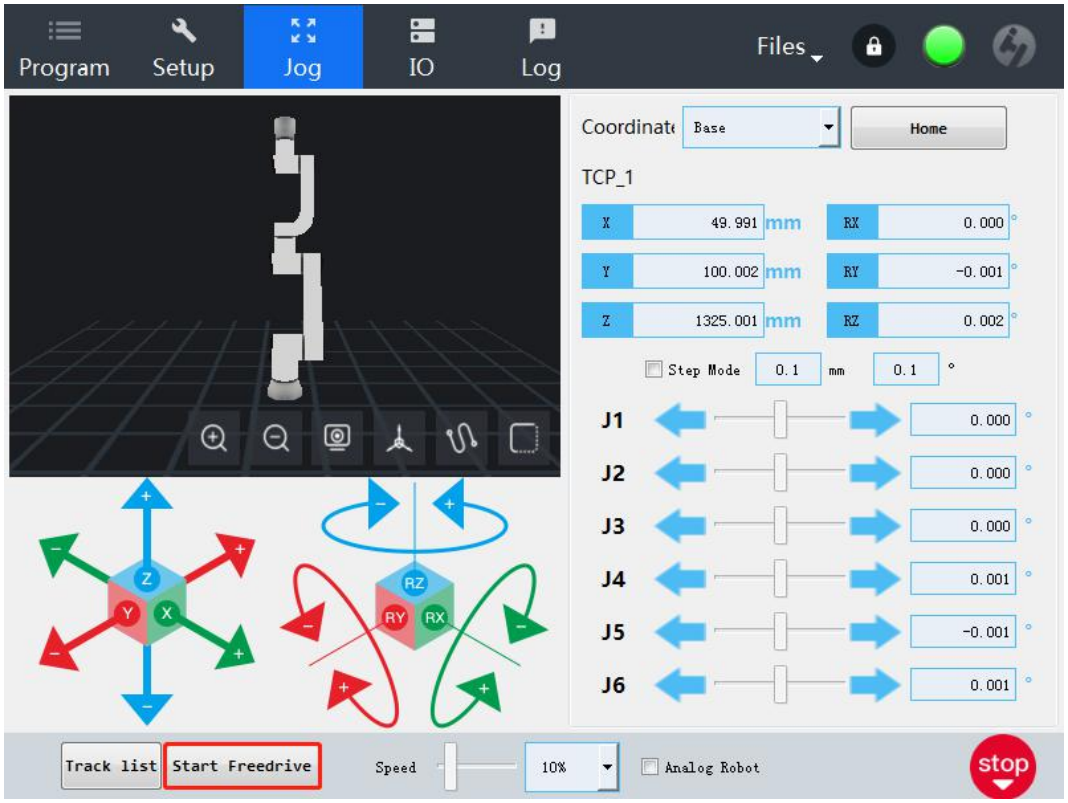
Click on the input box of the software interface, the input keyboard is shown above will appear, use the “shift” key to change the case, and use the “<<” key to delete.

Start Freedrive

Confirm that the original position, the payload, and installation angle parameters are all correct before starting freedrive. If the parameter is wrong, it will automatically turn off the freedrive and prompt the error immediately.

After starting freedrive, the button is displayed as “Close Freedrive ”, the robot can be freely dragged. When the robot approaches the safety boundary, a force in the opposite direction will be felt. At this time, continuing to the safety margin will cause the robot to report an error. So please drag within the security boundary.

Click “Close Freedrive” to end the freedrive state.



Tracklist

Click “Tracklist ” to enter “Free Drive Operation”. The drag track can be recorded on this interface, and a taught tracklist can be used in the script program.

Definition of each button in the interface:

1.Start Freedrive: Turn on the freedrive mode. Warning: Before turning on t freedrive mode, be sure to check whether the mechanical origin, installation angle, actual load, and other parameters are correct.

2.Close Freedrive:Turn off the freedrive mode.

3.New Traject: A new track name will be created in the track list. When the freedrive mode is enabled, the motion track can be obtained by manual sampling or timed sampling.

4. Delete Traject: Delete the traject in the current traject list (including traject name and traject point).

5. Traject Name: Show all current traject names.

6. Rename (Traject): Rename the traject name in the current traject list.

7. Clear Points: Clear all points in the traject in the current traject list.

8. Monitor Digital IO: When this option is checked, the sampling phase will record the digital IO change at the end of the robot and the stop time of the sampling process.

9. DI to DO: When this option is checked, during the sampling phase, the software interface will display the end output signal corresponding to the change in the end input signal.

10. Sampling Method: Two sampling methods, one is manual sampling; the other is timing sampling.

11. Sampling Interval: Set the sampling interval in the timed sampling mode.

12. Start Sampling: When the timing sample is selected and the sample interval is set, clicking the Start Sample button will start timing sampling.

13. Sampling current point: To set the manual sampling mode, drag the robot to the point and click “Sampling Current Point”.

14. Points list: Display all points recorded under the current traject.

15. Rename (Points list): Modify the default point name in the current sample point list.

16. Delete Point: Delete the point in the current sample point list.

17. Move to: Move to the point in the current sample point list

18. Motion type: Currently only supports joint angle motion, namely DragMoveJ.

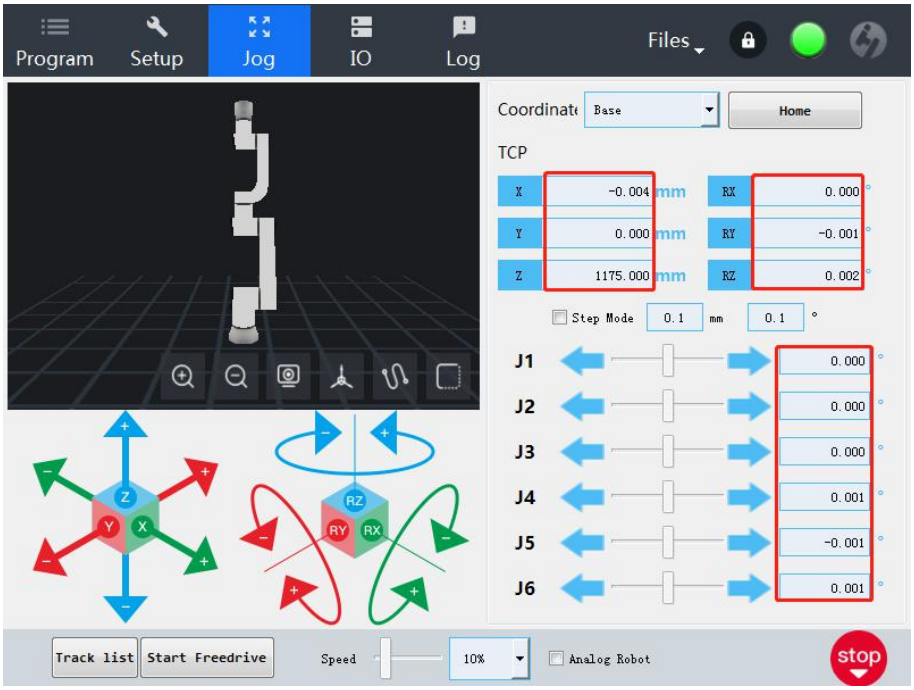
19. Move Trajectory: When the freedrive state is turned off, click this button to make the robot reproduce the selected traject in the traject list.

20. Stop Move: When the robot reproduces the trajectory, clicking this button will stop the movement.

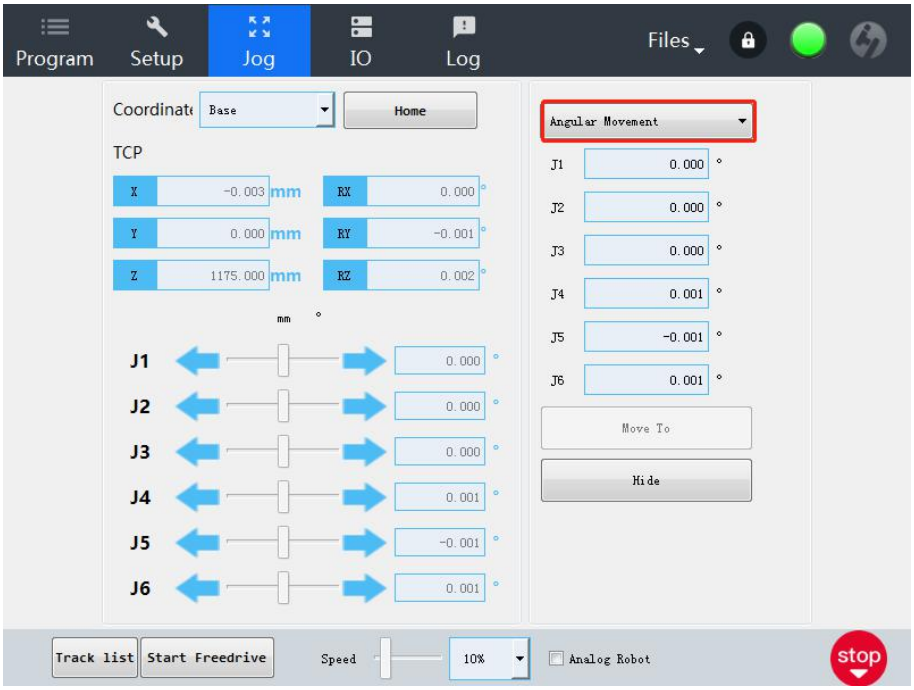
21. Exit: Exit the free drive operation interface.

Jog Editing

Click “Coordinates” and “TCP ” to enter the edit interface



The jog editing interface is shown below:



Jog editing function interface, for the user to manually input the known spatial

coordinate point coordinate value or the joint angle point angle value, and then long click the "Move to" button to directly move to the target point:

1) Angular movement: Select the angle movement to edit the data value of the angle motion;

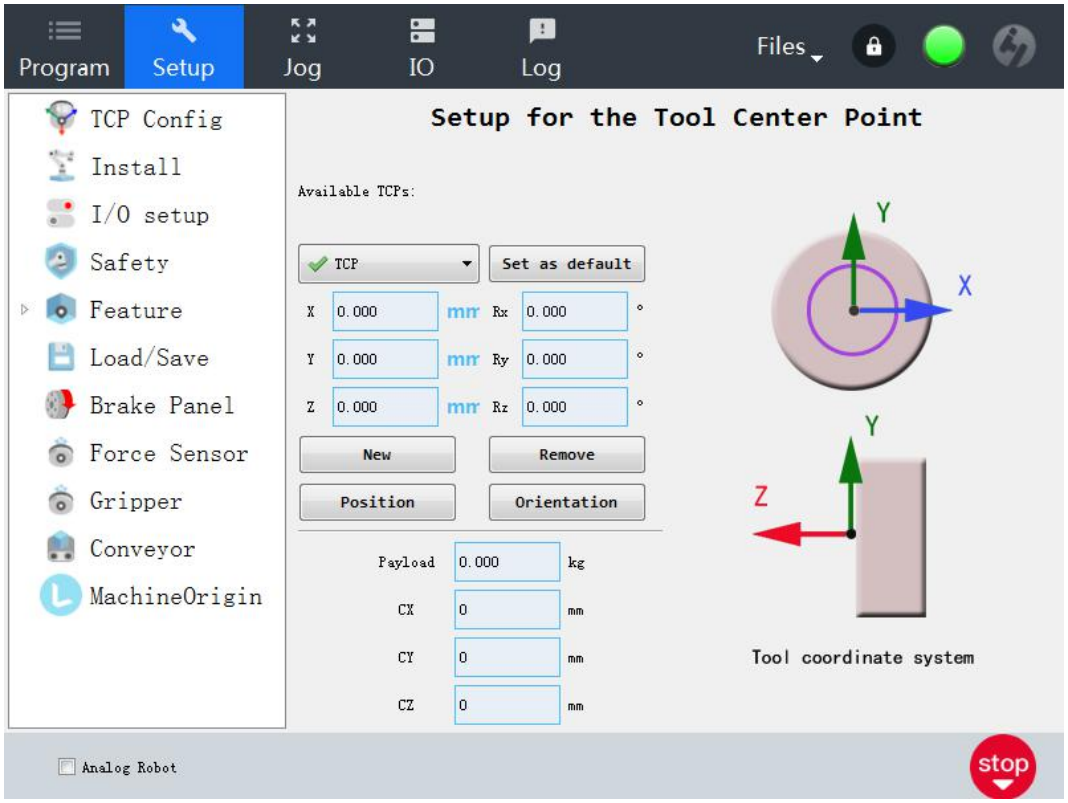
2） Space movement: Select the space movement to edit the data value of the space motion;

3）Move to: Hold the button to move to the target point after the movement is in place, the button becomes unclickable;

3）Hide: Click this button to exit the jog editing interface and return to the jog interface.

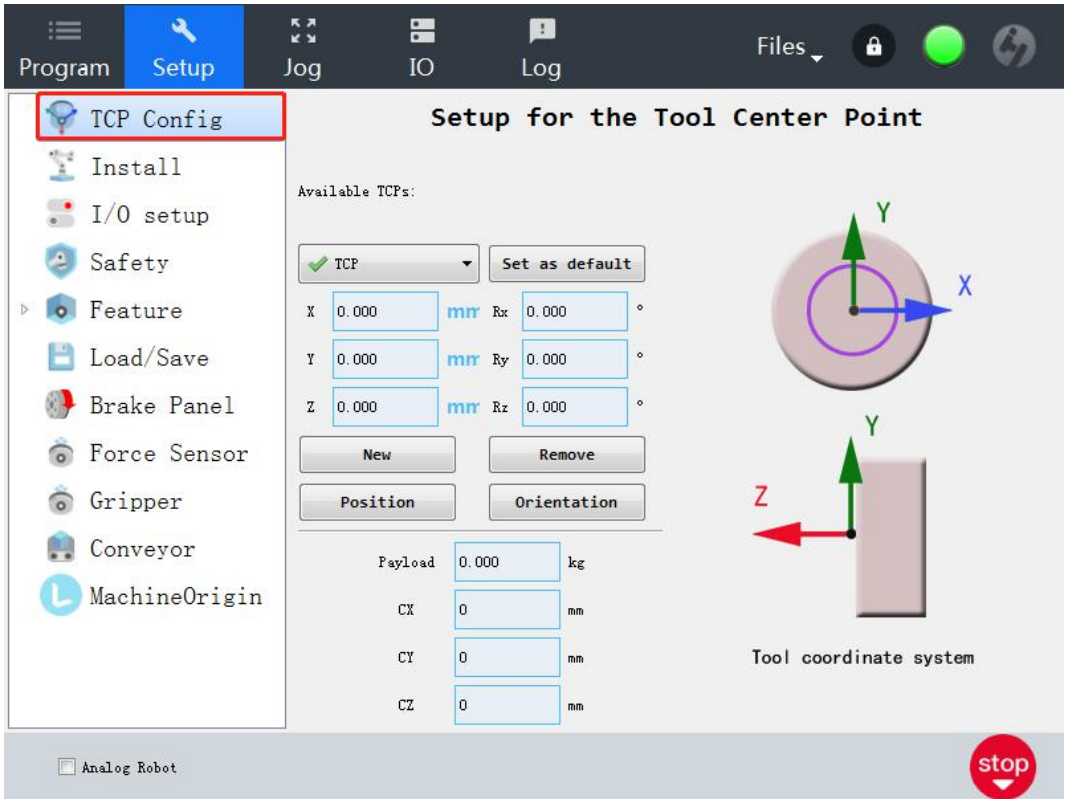
1.6 Setup

This area is used to set parameters such as TCP coordinate system, installation angle, safety space, safe parking, and grippers.



TCP Config

Set the tool coordinate system, you can use the manual input tool coordinate parameter value or manually teach the tool coordinate system.



Definition of each button in the interface:

1）New： New tool coordinates.

2）Remove： Delete selected tool coordinates.

3）Edit： When the user already knows the offset of the new TCP relative to the center of the flange, you can manually modify the tool coordinate system by manually clicking on the data frame and entering parameters in the pop-up interface.

4）Set as default： One and only one of the configured TCP's is the default TCP. In the TCP drop-down menu, there is a green circle icon to the left of the name that is set to the default TCP. The default TCP name is displayed at the TCP of the jog interface. To set the currently selected tool coordinate as the default TCP, click the Set as default button.

5）Position： The position of the origin of the newly created tool coordinate system is determined by teaching four points.

6）Oritation： The TCP direction can be automatically calculated by the following steps：

- Click" New' button
- Click the" Orientation" button
- Select a coordinate system from the drop-down list as prompted.
- Click on the teaching point to teach a point so that the tool direction is the same as the Z-axis direction of the selected coordinate system, and the teaching in the TCP direction can be completed.

Verify the calculated TCP direction and set it to the selected TCP using the Settings button.

The teaching principle of the four-point method: the end point of the tool needs to be moved from four different angles to the same position for teaching;

7）**Payload**

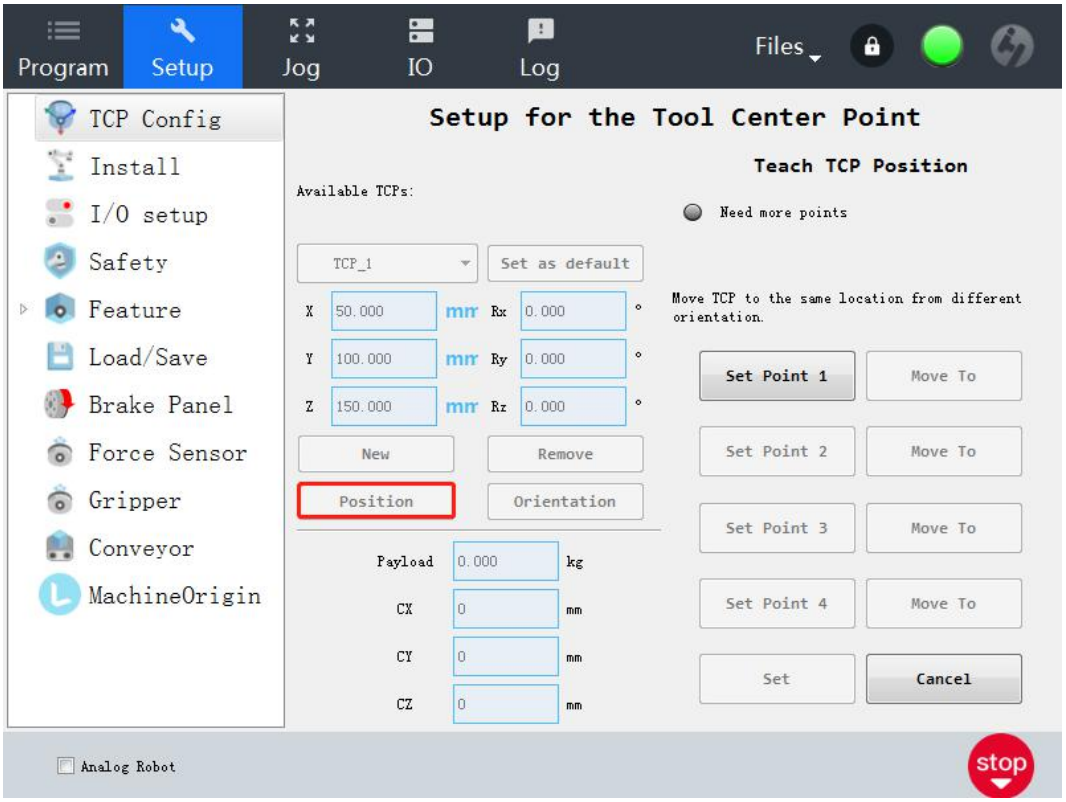
The user can set the mass of the load installed at the actual Elfin end and its offset from the center point of the flange in the X, Y, and Z directions. Click on the corresponding input box, the virtual keyboard will pop up, enter the correct value, click OK. The load setting range of the 5kg robot is [0~5] kg.

➔ TCP Examples

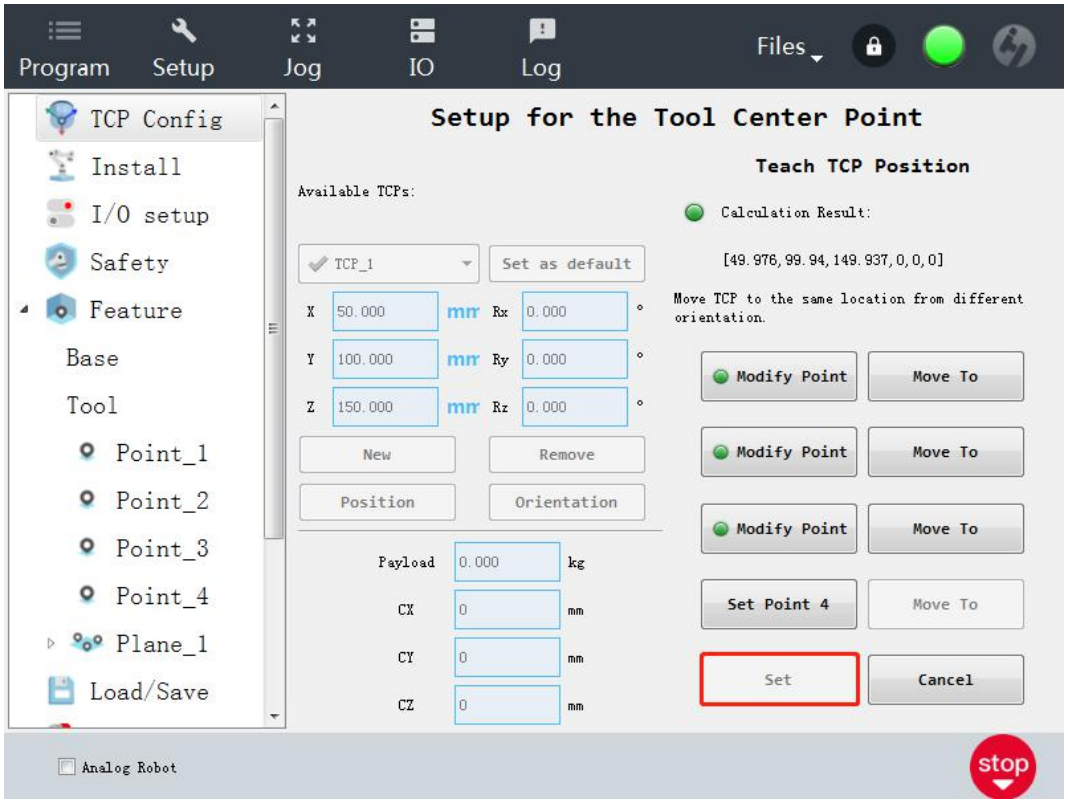
Position

① Click" New" button to automatically generate a new tool coordinate system TCP_1;

②Click the" Position" button to enter the four-point teaching interface； Move the robot tool center point to the same position from different angles according to the tips.



③When the teaching completes the interface after the first three points, the green light in front of the point indicates that the point meets the standard;



④After completing the teaching of the fourth point, click the “Set” button in the above

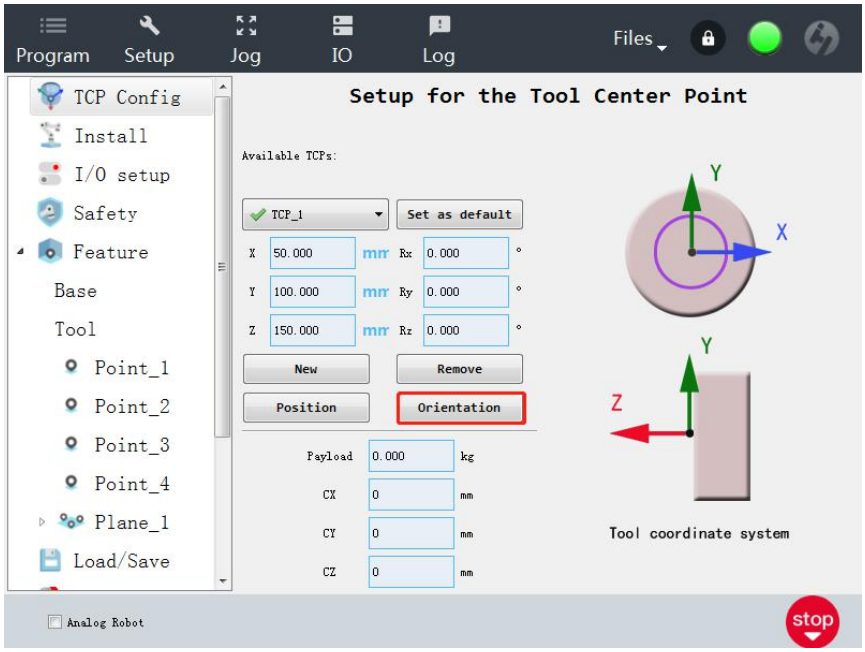
figure to complete the teaching of the tool position.

Note: The teaching points must be sufficiently diverse to ensure that the calculations are correct. Otherwise, the indicator light above the button will turn red. The quality of each taught point is indicated by the light of the corresponding button. Green = excellent, yellow = normal, red = unqualified.

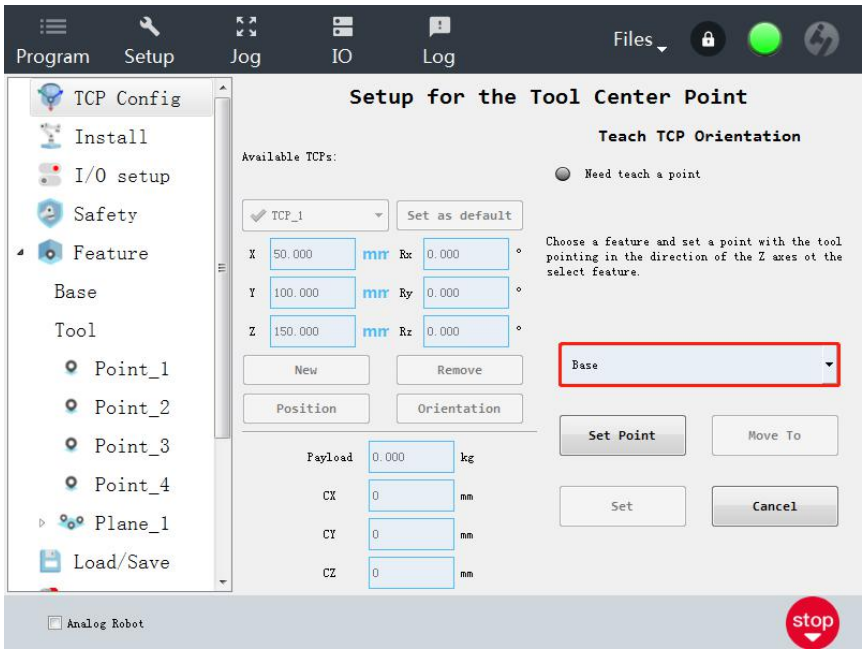
Orientation：

① Click" New" button to automatically generate a new tool coordinate system TCP_1;

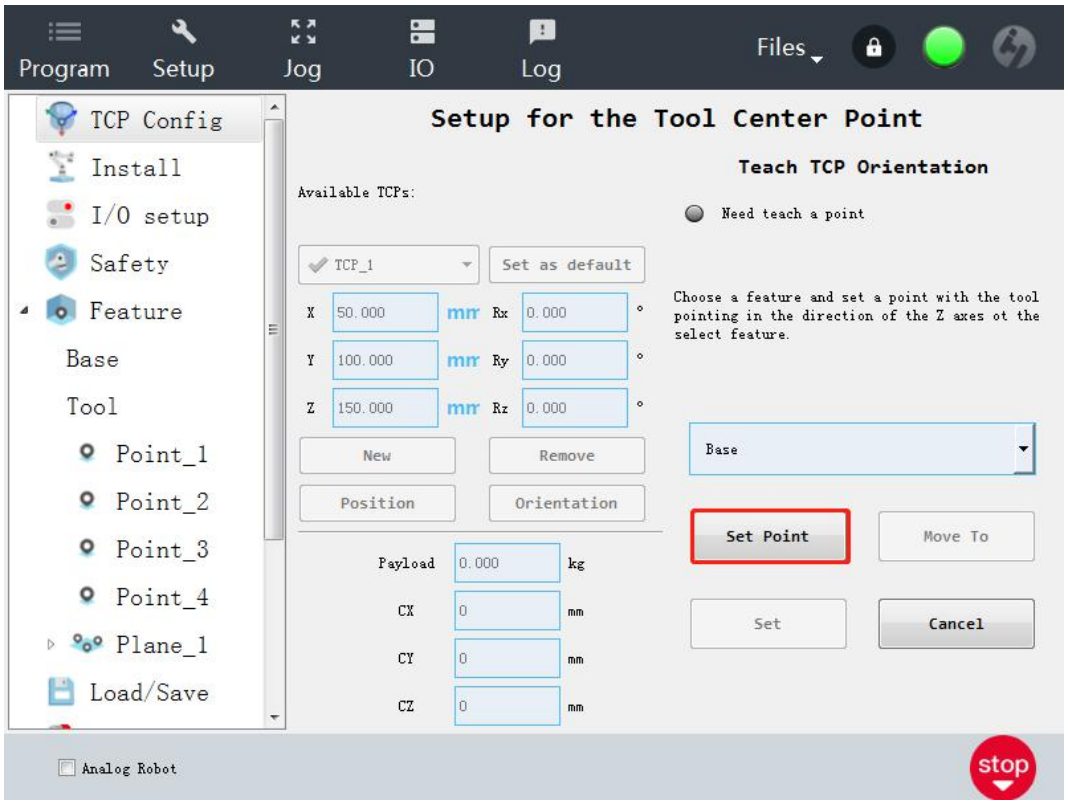
② Click the "Orientation" button in the interface to enter the TCP_1 direction teaching interface;



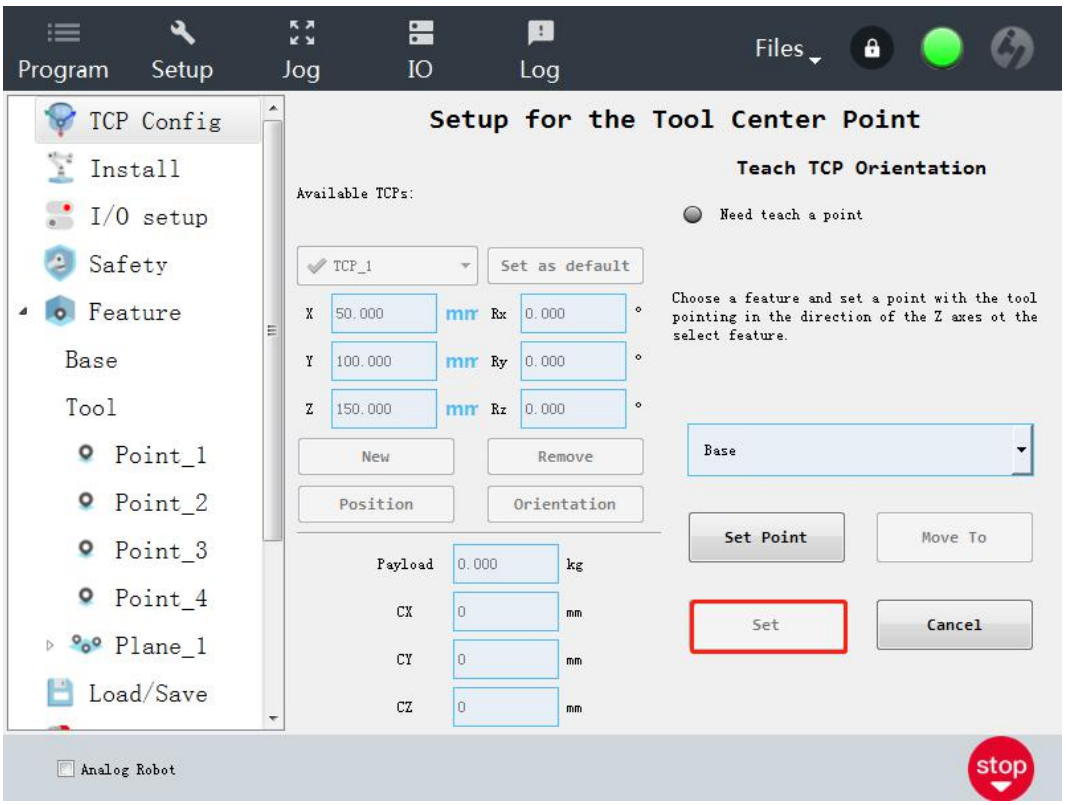
② According to the tips, select a coordinate system from the drop-down list, such as the "Base" coordinate system, the direction of the newly created TCP_ is the same as the direction of the "Base" coordinate system;



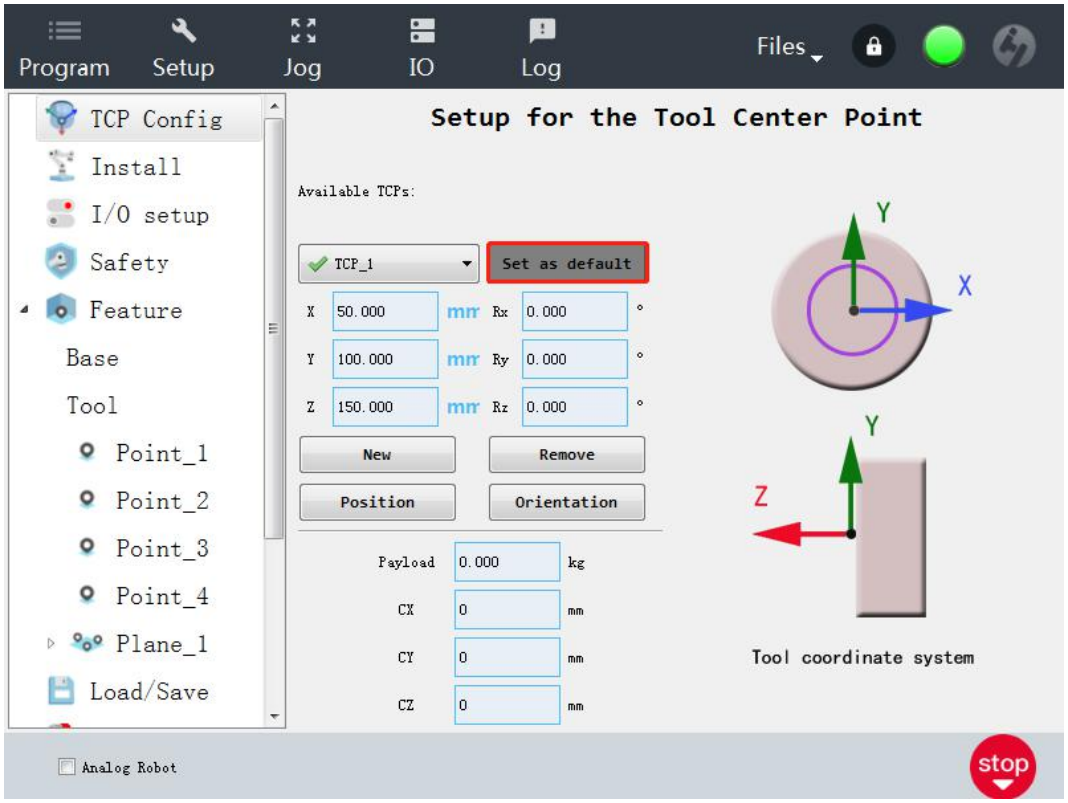
③ Click “Set Point”： Teach a point so that the tool direction is the same as the Z-axis direction of the selected coordinate system.



④ Click the “Set” button in the figure below to complete the teaching in TCP_1 direction；

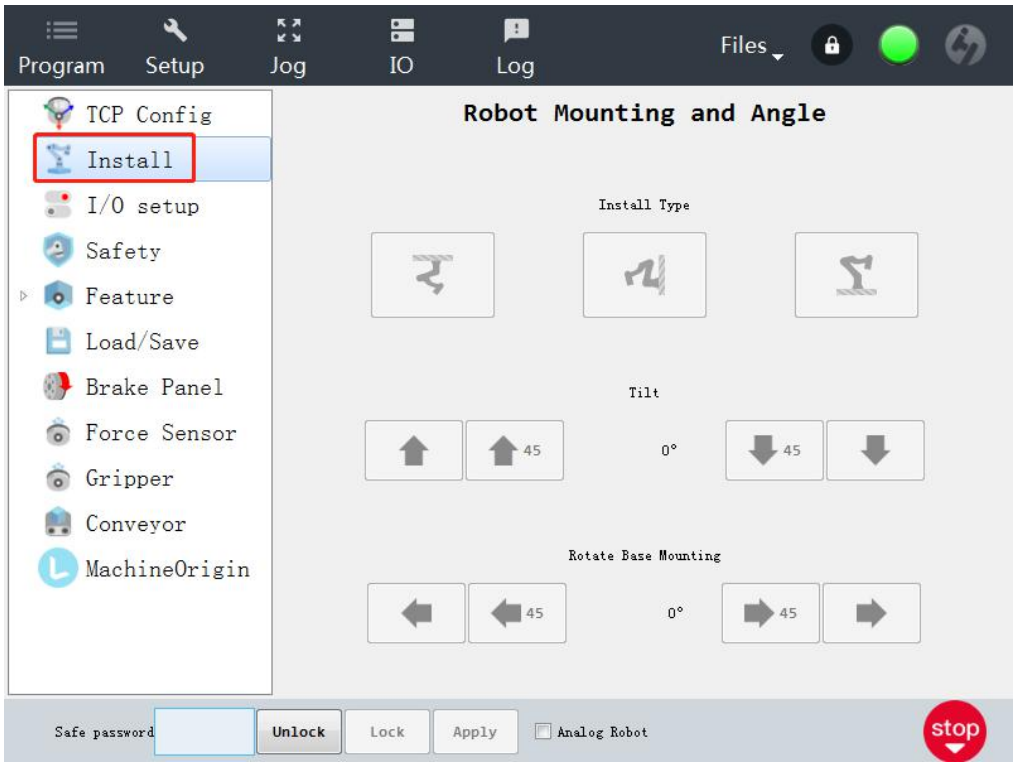


⑤If you want to set the newly created TCP as the default TCP coordinate system, you need to click the “Set as Default” button in the interface, as shown in the figure below, set TCP_1 as the default TCP:



Install

This page configures the installation location and angle of the robot and requires a security code to operate.



The system default robot arm is mounted on a smooth parallel floor or on the ground, in which case no changes to this interface are required. If the robot arm is mounted ceiling-mounted, wall-mounted or angled, it needs to be adjusted using the buttons in the interface. The interface installation type of button can be used to set the mounting angle of the robot arm. The three buttons of the installation type set the angle to ceiling (180◦), wall (90◦), ground (0◦), and the tilt button can be used to set any angle. The rotary button at the bottom of the screen is used to rotate the mounting angle of the robot arm to match the actual mounting angle.

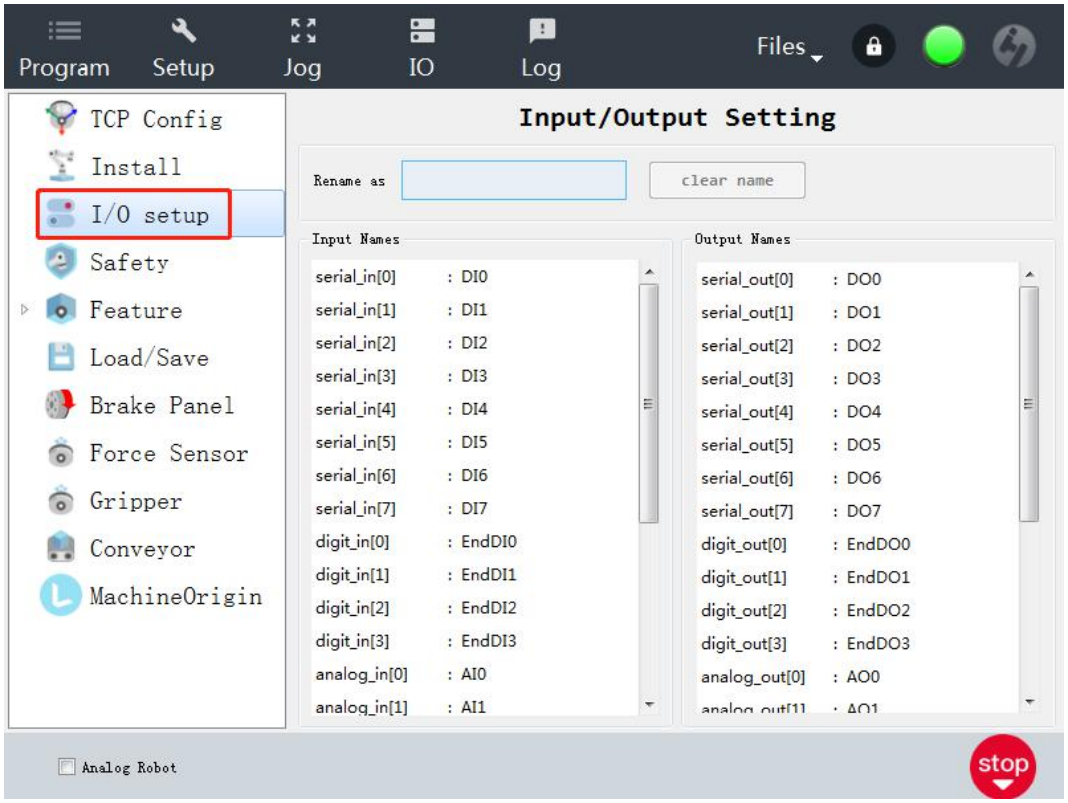
Tilt: Rotate around the Y coordinate of the Base coordinate, counterclockwise is positive;

Rotate Base Mounting: Rotate around the Z coordinate of the Base coordinate, counterclockwise is positive.

Warning: Be sure to use the correct installation settings. If the robot arm installation information is not set correctly, this will result in frequent safety stops or the possibility that the robot arm will lose weight when the zero force teach button is pressed.

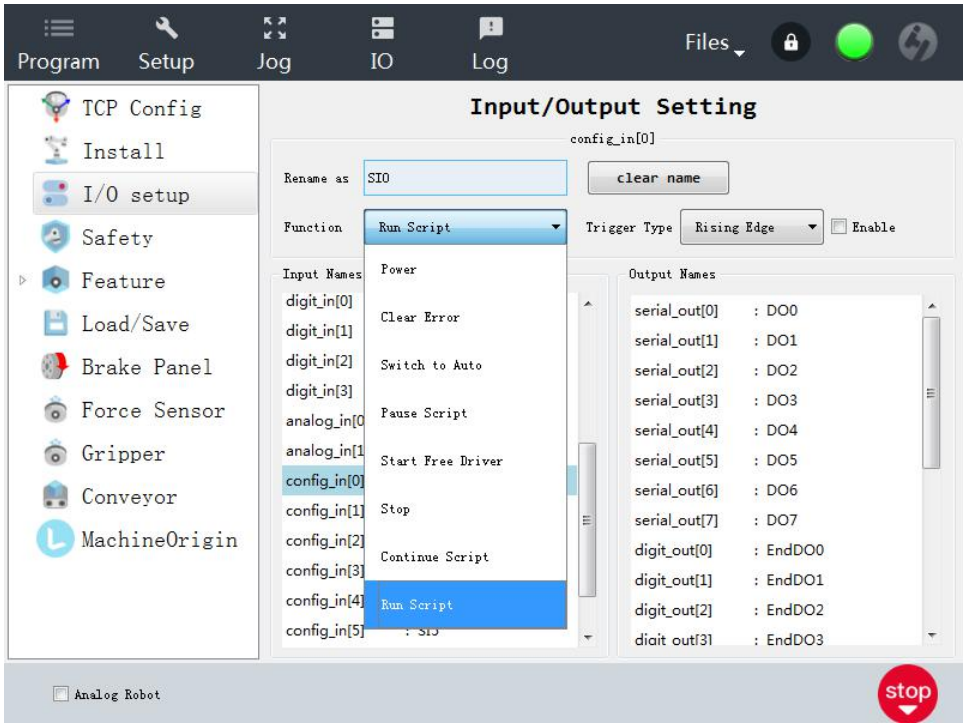
The controller uses an advanced dynamics model to ensure that the robot arm moves smoothly and accurately and that the robot arm can self-support while in the zero-force teaching mode. Therefore, it is important that the mounting angle and position of the robot arm be set correctly.

I/O Setup



1) Names can be assigned to the input and output signals, making it easier to remember the role of each signal when operating the robot. Click to select the appropriate I/O and use the on-screen keyboard to set the signal name. You can restore the name by setting the name to an empty string.

2) The configurable power box can be configured with eight configurable input IO signals to trigger the corresponding action. Config_in[0]~ Config_in[7] corresponds to SI0~SI7 of the robot IO interface. Available actions include "Power", "Clear Error", "Switch to Auto", "Pause Script", "Start Free Driver", "Stop", "Continue Script", "Run Script".



a) Power: The trigger type can be selected as the rising edge or the falling edge. After the configuration function is enabled, the controller continuously monitors the IO state, and the enable command is issued when the selected trigger condition is met.

b) Clear error: The trigger type can be selected as a rising edge or a falling edge. After the configuration function is enabled, the controller continuously monitors the IO state and issues a clear command when the selected trigger condition is met.

c) Switch to Auto: The trigger type can be selected as a high level or low level. After the configuration function is enabled, the controller continuously monitors the IO state. If the selected trigger condition is met, the script mode is entered, and the script mode is terminated.

d) Pause Script: The trigger type can be selected as a rising edge or a falling edge. After the configuration function is enabled, the controller continuously monitors the IO state, and the pause script command is issued when the selected trigger condition is met.

e) Start Free Driver: The trigger type can be selected as a high level or low level. After the configuration function is enabled, the controller continuously monitors the IO state. When the selected trigger condition is met, the zero force teaching mode is entered, and the zero force teaching mode is ended.

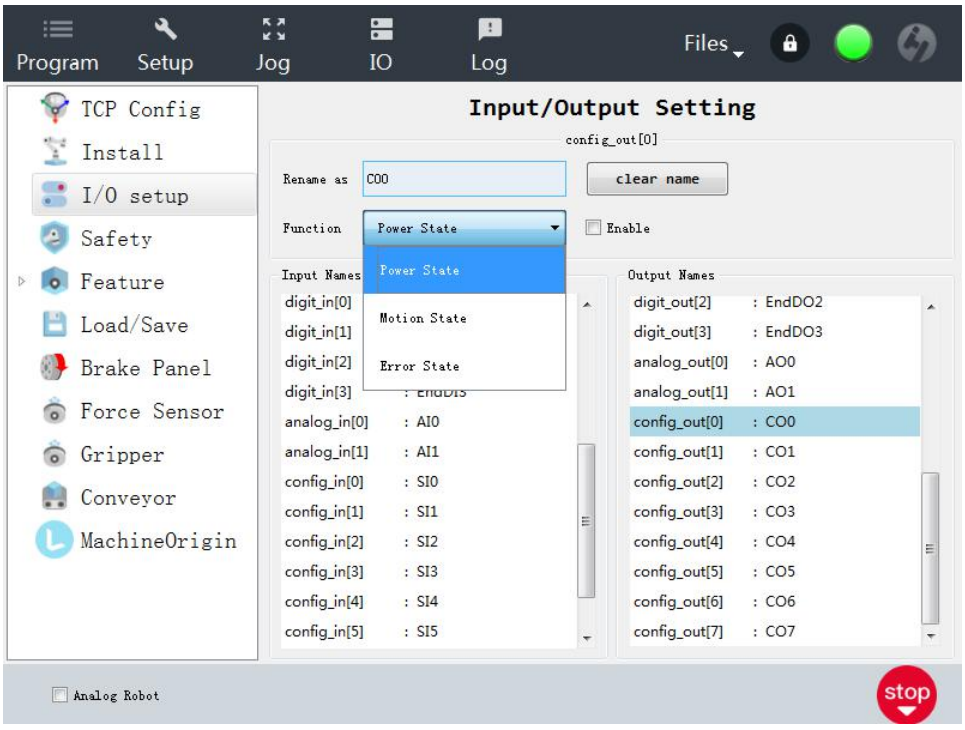
F) Stop: The trigger type can be selected as the rising edge or the falling edge. After the configuration function is enabled, the controller continuously monitors the IO state, and the stop command is issued when the selected trigger condition is met.

g) Continue Script: The trigger type can be selected as the rising edge or the falling edge. After the configuration function is enabled, the controller continuously monitors the IO state, and the continuous trigger command is issued when the selected trigger condition is met.

h) Run Script: The trigger type can be selected as the rising edge or the falling edge. After the configuration function is enabled, the controller continuously monitors the IO state. When the selected trigger condition is met, the command to run the “F_main” function and start the timer is issued. That is, you need to use this function to run the script. You need to write a script named F_mian function, compile the script and switch to script mode to run the script through IO.

3) In addition, eight configurable output IO signals can be configured to reflect the current state of the robot. Config_out[0]~ Config_out[7] corresponds to SO0~SO7 of the robot IO interface. The states include "Power State", "Motion State", and “Error State". When the "Enable" option is checked, the corresponding output IO outputs IO information according to the selected state. For example, Config_out[4] enables the motion state function. When the robot is in motion, the corresponding SO4 is high and there is no motion state, then SO4 is low.

If this option is unchecked, there is no difference between configurable IO and general IO. It is also possible to rename the eight configurable output IO signals of the electrical control box;

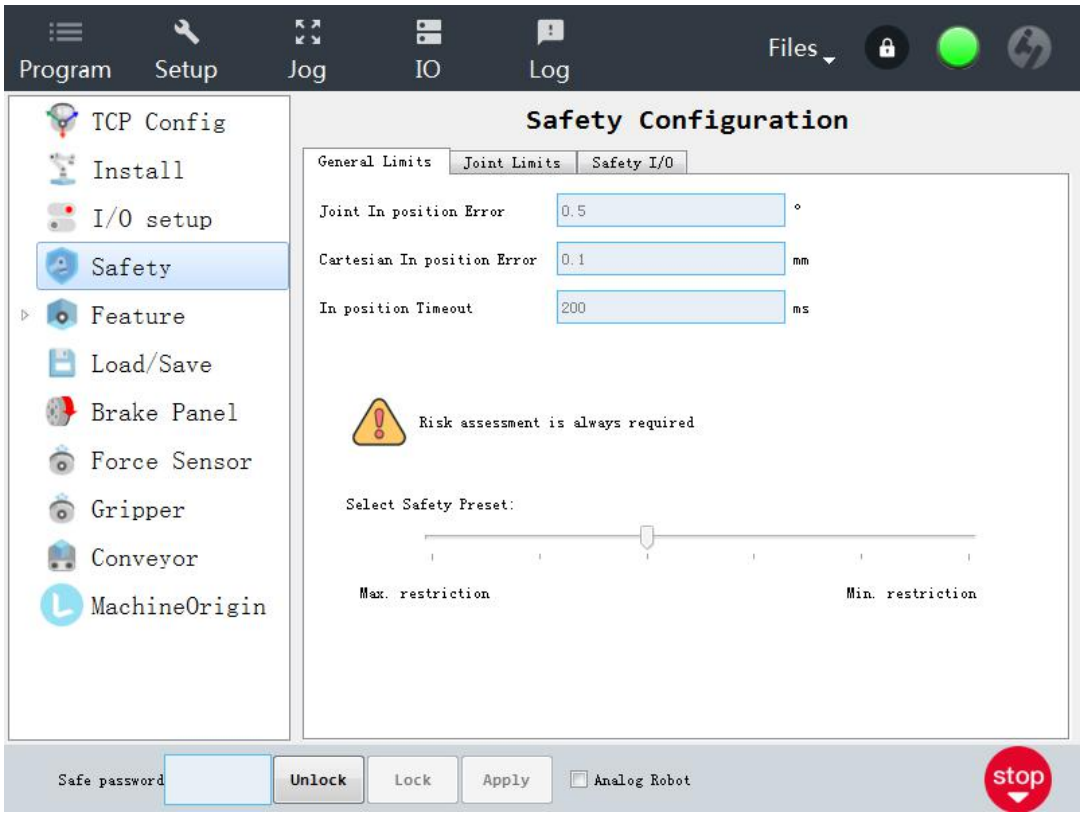


If two IOs are configured with one function at the same time, only the IO with the largest serial number is valid, that is, Config_in[2] and Config_in[5] are both configured with the power function. Only when Config_in[5] meets the trigger condition will it be sent.

Safety

This page configures the robot's safe collision threshold range, joint limits, and Safety I/O settings. You need to enter a security code to operate.

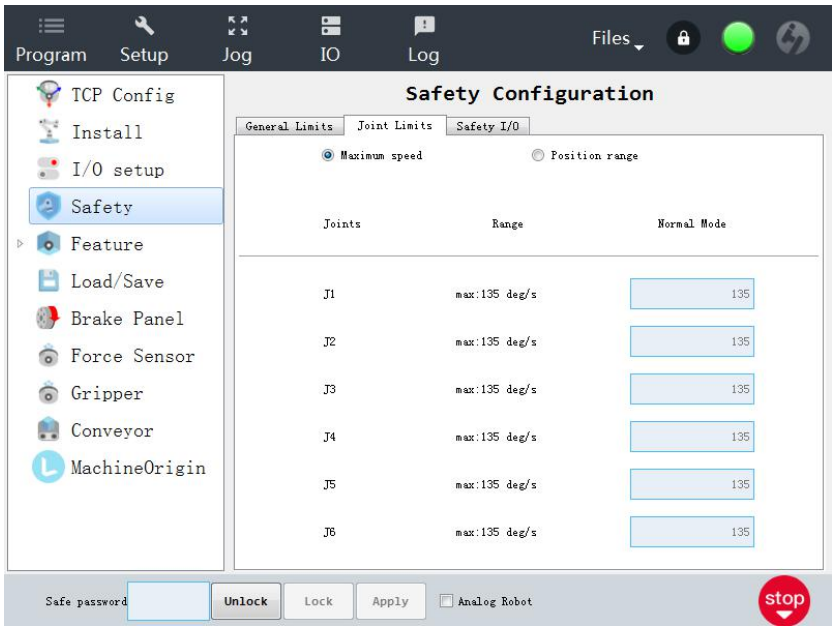
General Limits

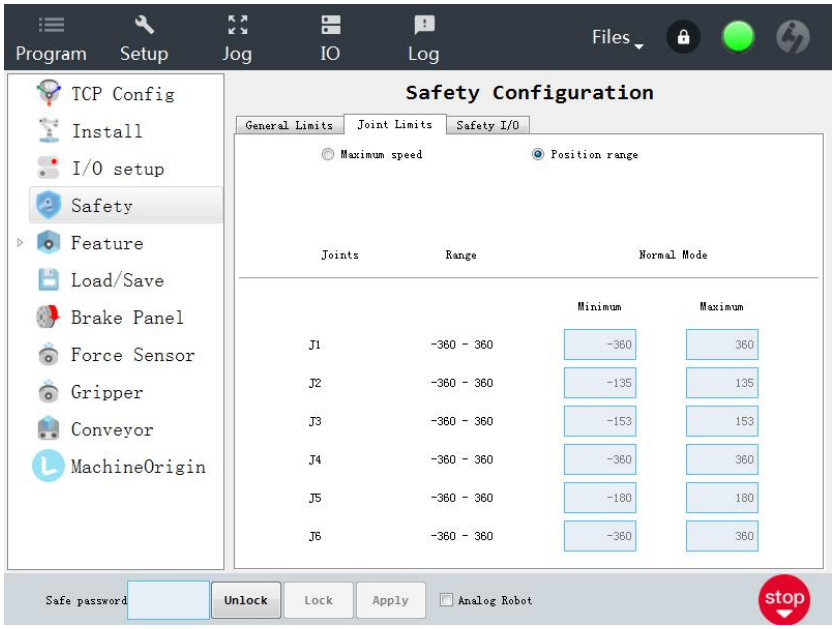


As shown in the figure above, pull the selection safety preset block to adjust the collision speed of the safe stop during the operation of the robot. The "Max.restriction" sets a smaller collision force, and the "Min.restriction" sets a larger collision force. A total of 6 gears are selected on the safety preset axis, and the required collision force is gradually increased from left to right. If it is not necessary, do not choose to limit the minimum gear position. This gear position does not guarantee the safety of operators and robots.

Joint Limits

As shown in the figure below, select “Maximum speed”, enter the security code, click “Unlock”, you can set the maximum speed of each joint (Elfin's maximum joint speed is 135°/s, the setting value can't exceed the limit), the setting is completed, click “Apply”,”then click ”Lock”.



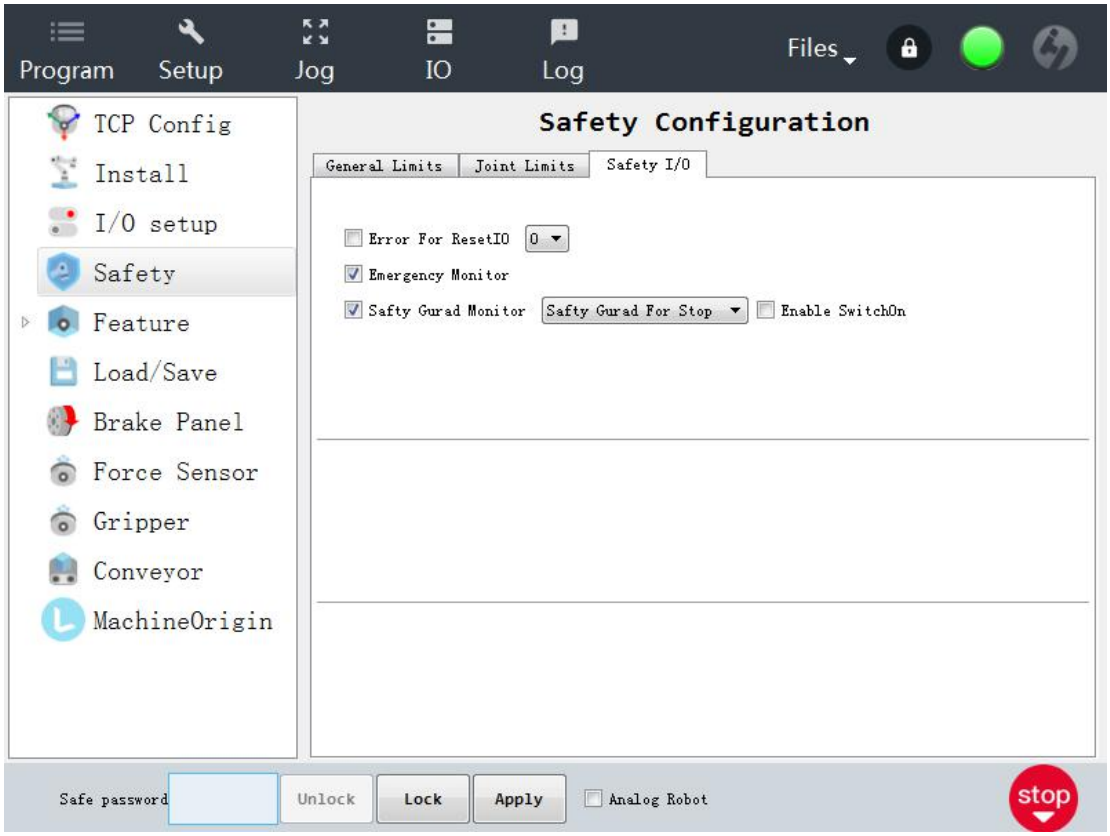


The same can be set for the location range.

Elfin's joint angular motion range is:J1：±360°;J2：±135°;J3：±153°;

J4：±360°;J5：±360°;J6：±360°

Safety I/O



Error For ResetIO: Whether the electronic control box output IO is set to the selected IO state after the robot reports an error.

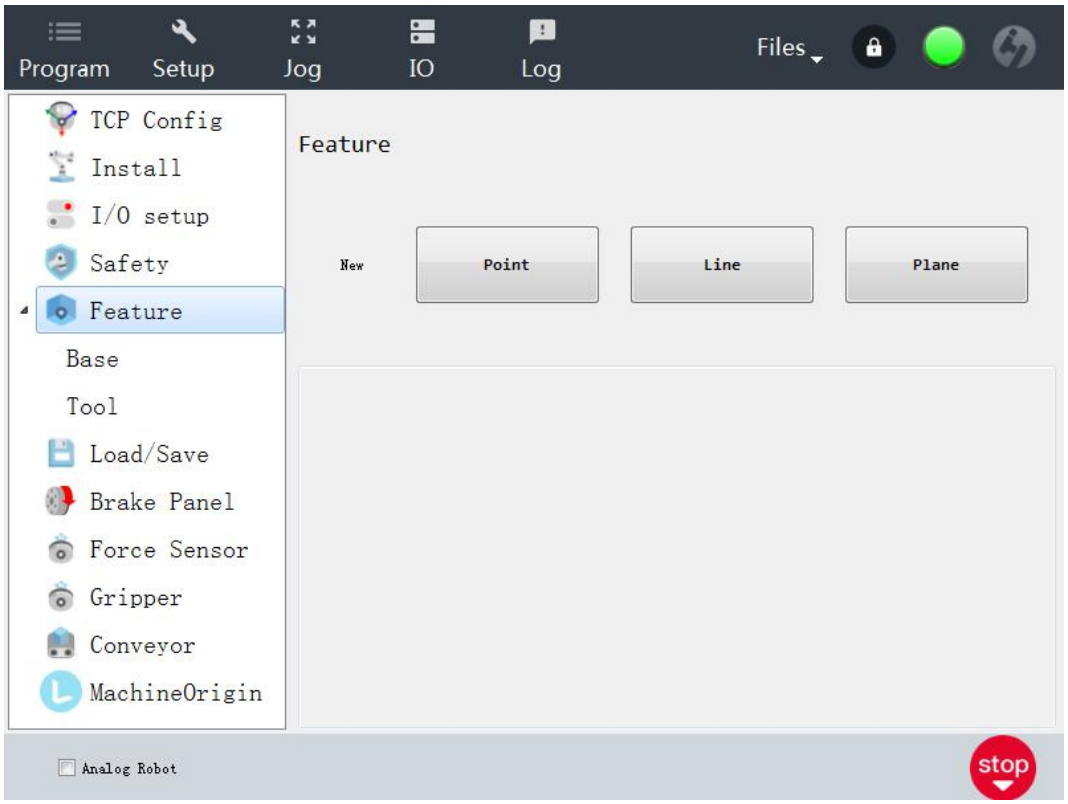
Emergency Monitor: Whether to start emergency stop monitoring, if it is a complete set of equipment, do not turn off emergency stop monitoring. After closing, pressing the emergency stop does not stop the robot immediately.

Safety Guard Monitor: Whether to start the safety light curtain monitoring, if the safety light curtain is started to stop, the safety IO signal is monitored, the movement is stopped, and the pop-up window is in a safe protection state, until the corresponding IO signal is not reset and the error is reset to restart the robot. If the safety light curtain is paused, the safety IO signal is monitored and the script is paused. After the reset is cleared, the previous script can be continued.

Enable SwitchOn: Whether to start the three-stage switch, after starting, you need to press the three-stage switch to operate the robot movement, switch the freedrive, and so on.

Feature

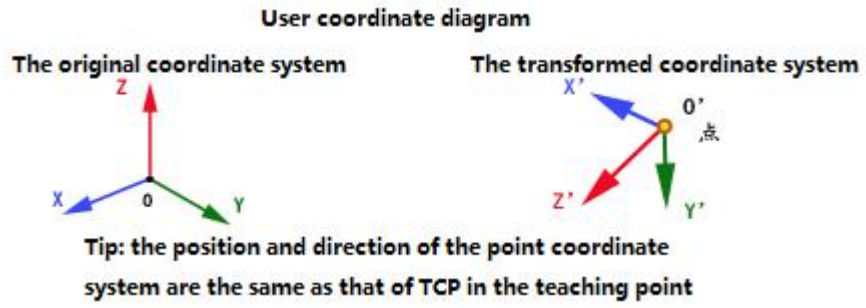
To set the user coordinate system (Point, Line, Plane), and set the user coordinate system by teaching.

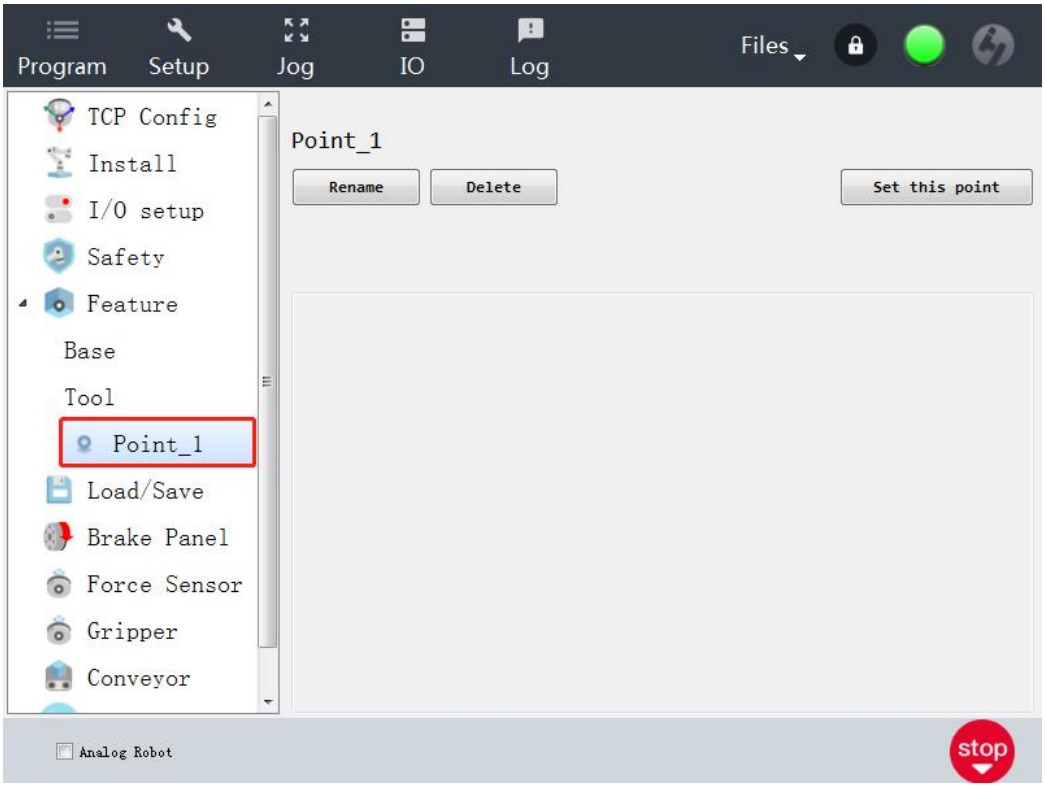


1) Point coordinate system

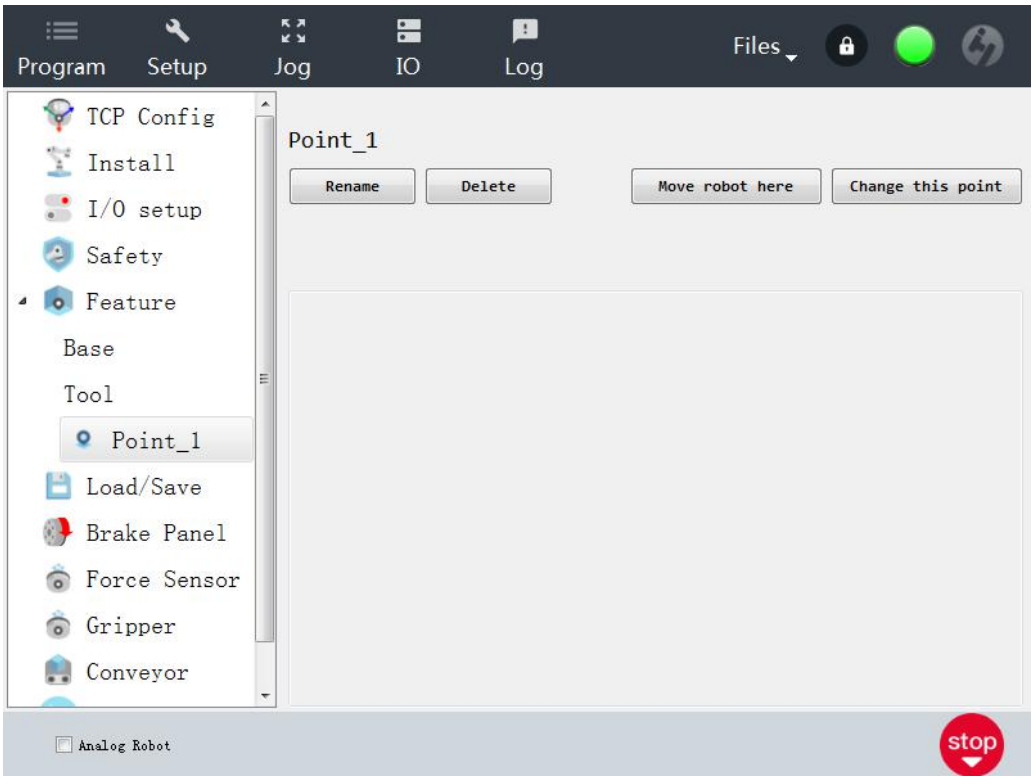
Click the "Point" button, the "Point_N" point coordinate system will be generated automatically, as shown in the figure below: "Point_1":

Click the "Set this point" button to set the point coordinate system. The position and direction of the point coordinate system are the same as the position and direction of the TCP at the teaching point.





After the point coordinate system is taught, the interface information is displayed as follows (such as Point_1):



(1)**Rename**:Use this button to rename a feature.

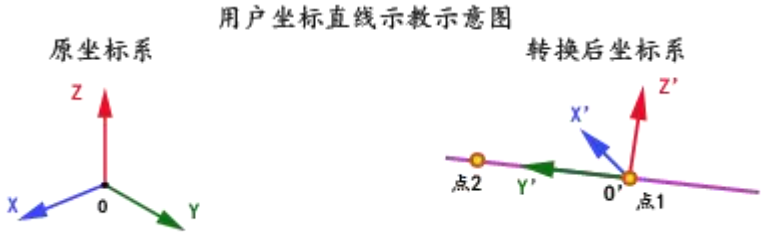
(2)**Delete**:Use this button to delete the selected feature.

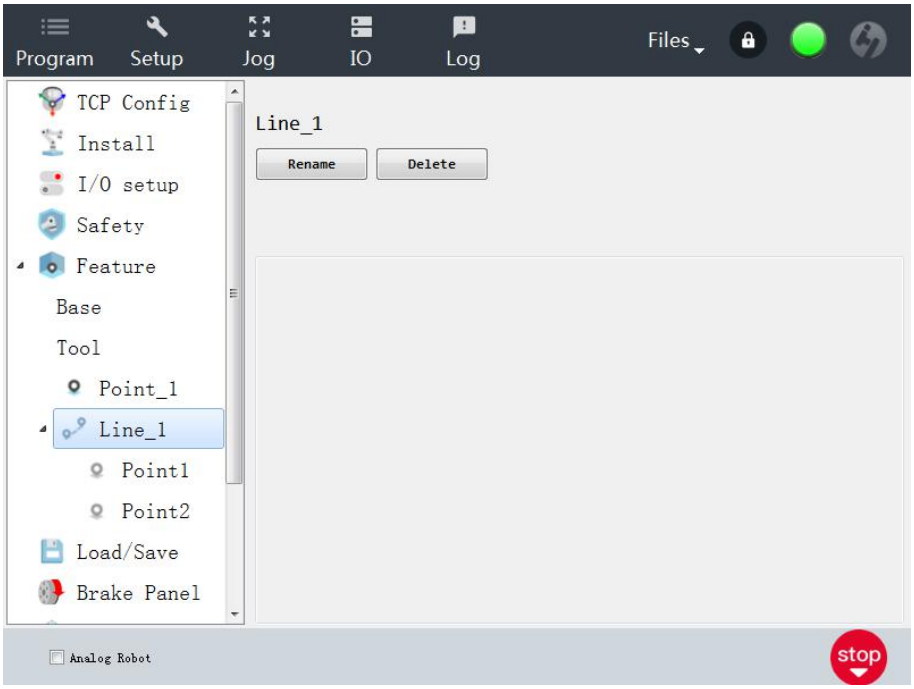
(3)**Move robot here**:Press this button to move the robot arm to the selected feature teach a point. After the move is over, the coordinate system of the feature and the coordinate system of TCP will coincide.

(4)**Change this point**:If it is found that the "point coordinate system" of the original teaching completion does not meet the requirements, the button can be clicked to re-teach the point coordinate system.

2)Line coordinate system

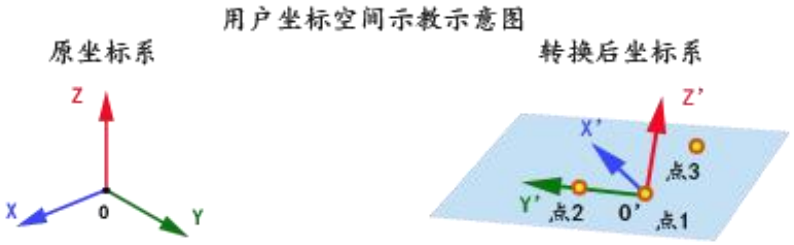
Add a line: Add a line feature as shown below. Teach two points to calculate the line feature. The first point points to the direction of the second point, which is the positive direction of the y-axis of the line coordinate system. The z-axis of the linear coordinate system is the same as the projection of the Z-axis of the first point in a straight line. The origin position of the linear coordinate system is the same as the position of the first teaching point.

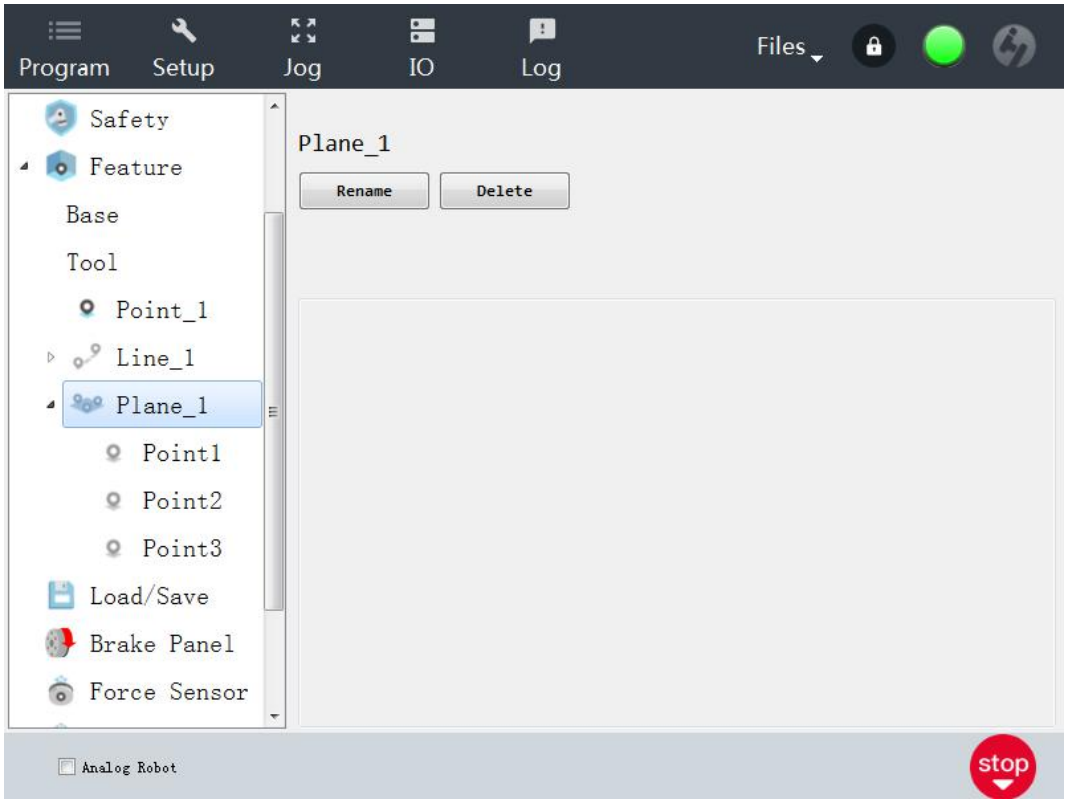




3) Plane coordinate system

Add plane: Add a plane feature to the installation settings as shown below. The plane is defined by three teach point features. The position of the coordinate system is the same as the position of the first teach point. The z-axis is the plane normal, and the axis from the first point to the second point is the Y-axis forward and reverse. The positive direction of the z-axis is set so that the angle between the z-axis of the plane and the z-axis of the first point is less than 180 degrees.





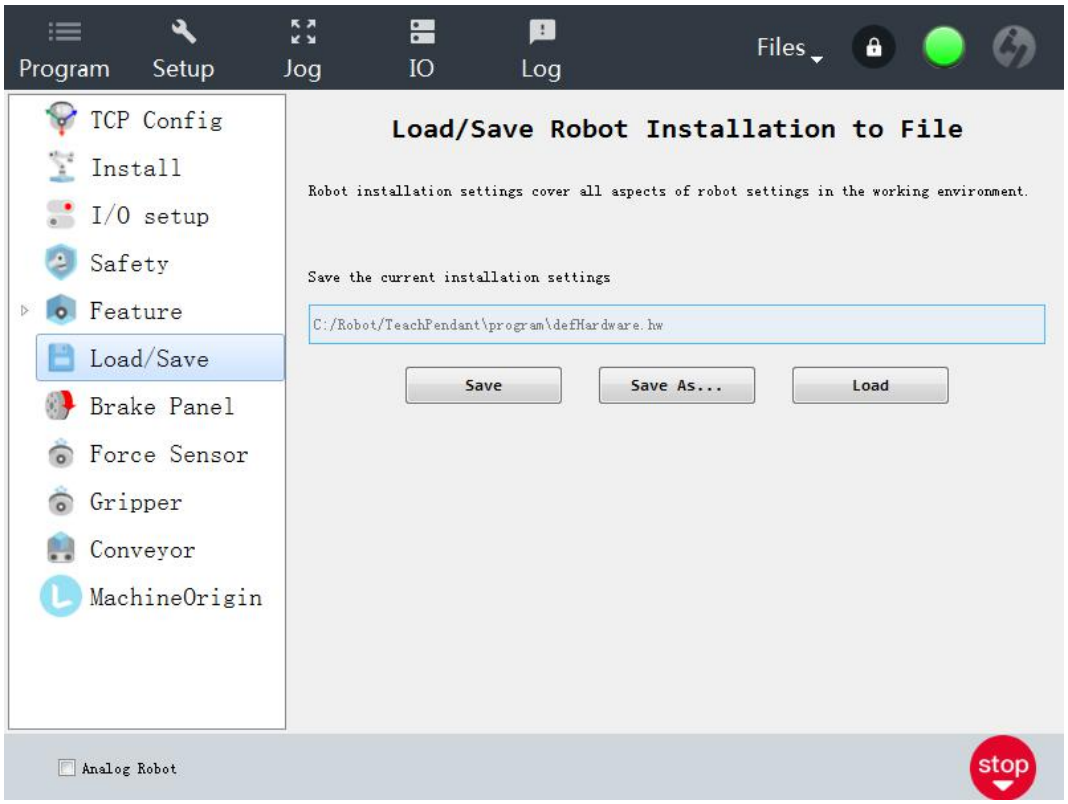
Load/Save

Used to load or save the robot's installation settings file (.hw file type). The installation information includes machine origin, dynamic parameters, and so on. The meaning of the interface button is as follows:

(1)**Save:**Save the current installation settings to the path shown in the path box.

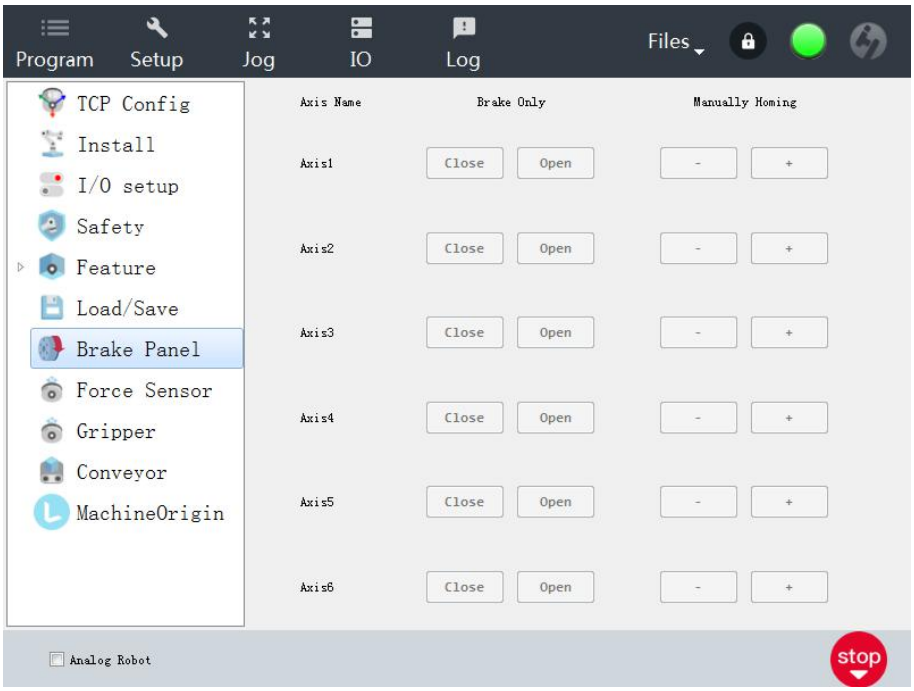
(2)**Save as:**Save the current installation settings to a custom path.

(3)**Load:** Load (under the /program path) to install the settings file. When the file is loaded, the robot will lose power and need to be powered on again.



Brake Panel

This interface can be used to brake a single axis and do HomeStep2 operation, but both operations can be performed when the robot is powered off.



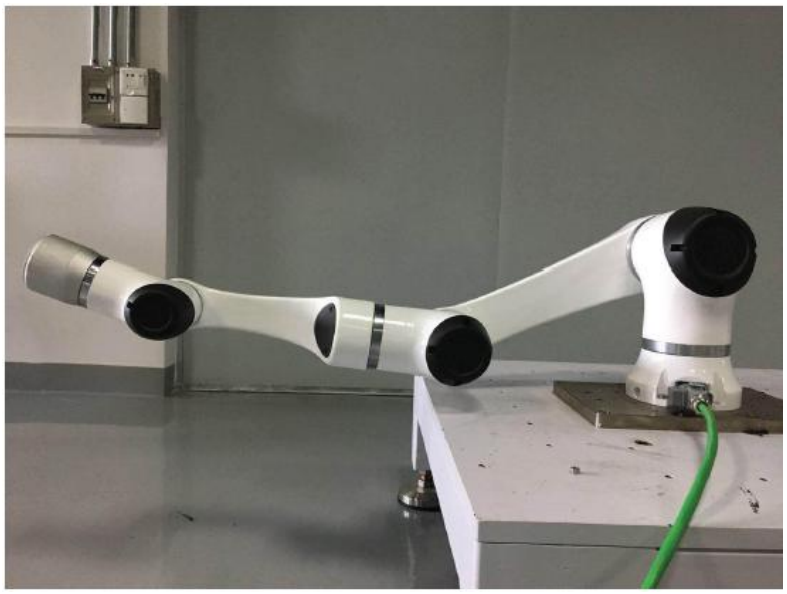
Open/close the axis brake

In some special cases, this function can be used when the brake of a single axis needs to be turned on for debugging.

➔ Application scenario：

The brake setting function is mainly used when the robot is self-collision or is activated when it is unable to reset.

① The 2nd axis collides with an external object



② The 3rd axis collides with an external object

③ Robot self-collision (between 2nd and 3rd axis)

▸Manually open (close) the brake operation method:

When the robot is powered off, let a person lift the corresponding axis:

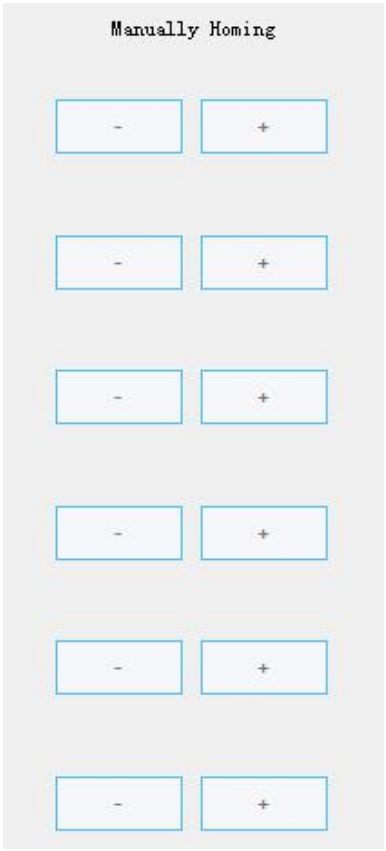
① If the 2nd axis collides with an external object, please lift the 2nd axis manually and you will hear a "giggle" sound. In this case, please move the robot arm a little in the opposite direction, then switch to the software “Brake” interface (hands must hold the 2nd axis all the time), click the “Open” button of Axis 2 to open the J2 axis brake and manually lift the J2 axis to the safe position. Then click on the "Close" button of Axis2 to close the brake;

② If the 3rd axis collides with an external object, please lift the 3rd axis manually and you will hear a "giggle" sound. In this case, please move the robot arm a little in the opposite direction, then switch to the software “Brake” interface (hands must hold the 3rd axis all the time), click the “Open” button of Axis 3 to open the J3 axis brake and manually lift the J3 axis to the safe position. Then click on the "Close" button of Axis3 to close the brake；

③ If the robot is self-collision (2nd or 3rd axis collide), please raise the 3r axis manually, you will hear the “giggle” sound of the brake when the J3 axis is lifted, and then switch to the software “Brake” interface (hands must hold the 3rd axis all the time, click the “Open” button of Axis 3 to open the J3 axis brake, manually lift the J3 axis and J2 axis to the normal distance, then click the “Close” button of Axis3 to close the brake;；

Manually Homing

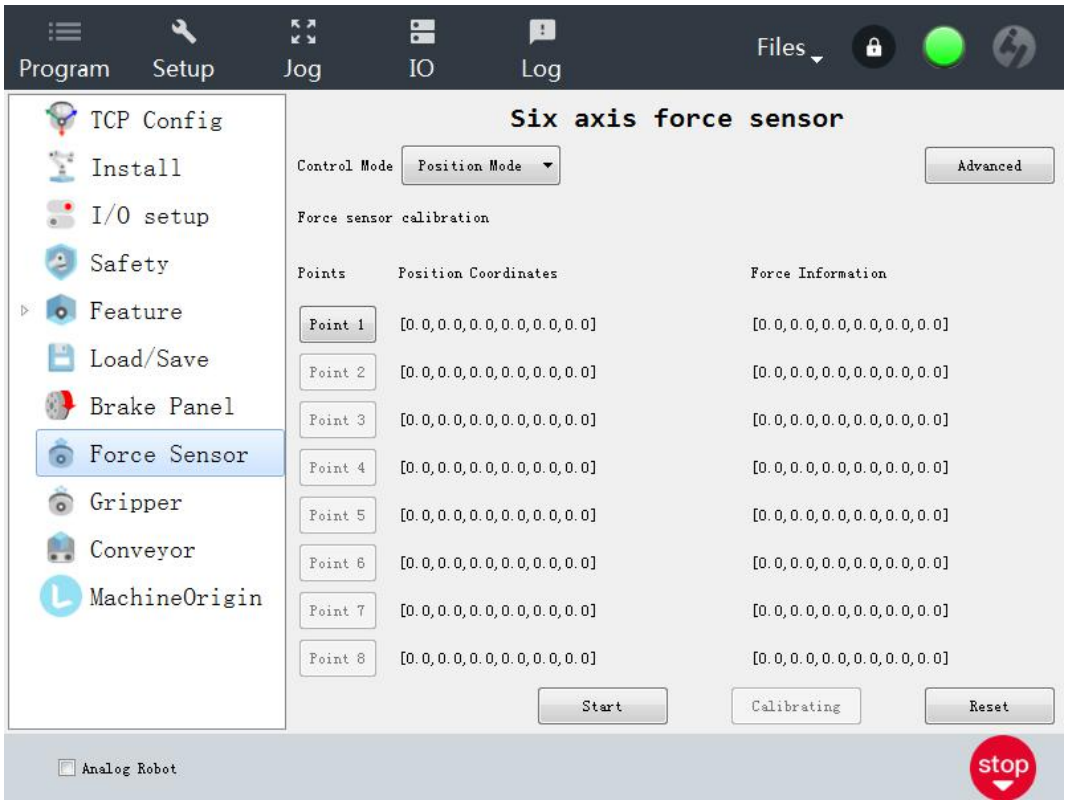
When the original position is lost，power off the robot arm firstly，then switch to Manually Homeing,click ”-” or ”+” to find the algorithm.



After performing the “Manually Homing” operation, power on the robot, and then return the robot to the original position at a slow speed (about 10%) to confirm whether the original position is correctly retrieved.

Force Sensor

This interface is applied to the calibration of the six-dimensional force sensor, and the application of the force sensor is completed by selecting the control mode.

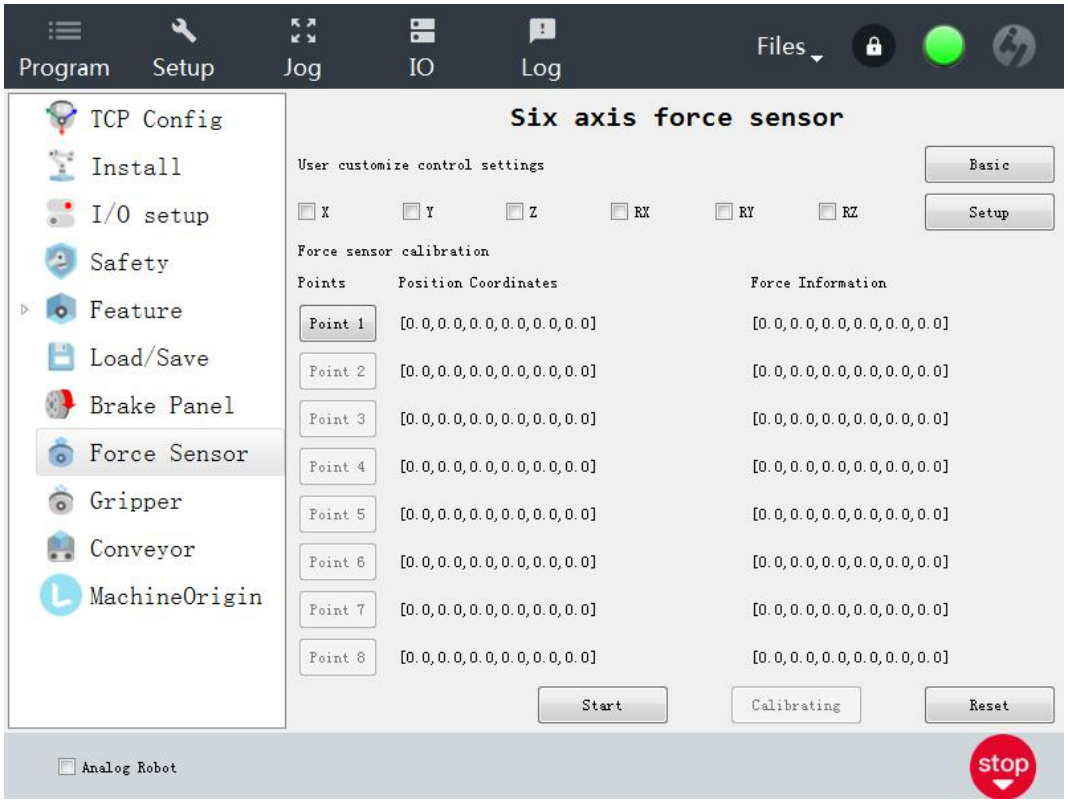


Position Mode:The motion space based on the force sensor is set in the X, Y, Z direction in the Cartesian space coordinate system.

Attitude Mode:The motion space based on the force sensor is set in the RX, RY, RZ direction in the Cartesian space coordinate system.

Freedom Mode:The motion space based on the force sensor is set in the X, Y, Z, RX, RY, RZ directions in the Cartesian space coordinate system.

Advanced:User-defined selection combines the elements in the Cartesian space coordinate system, select the corresponding mode and click the “Setup”button to complete the settings, as shown below.



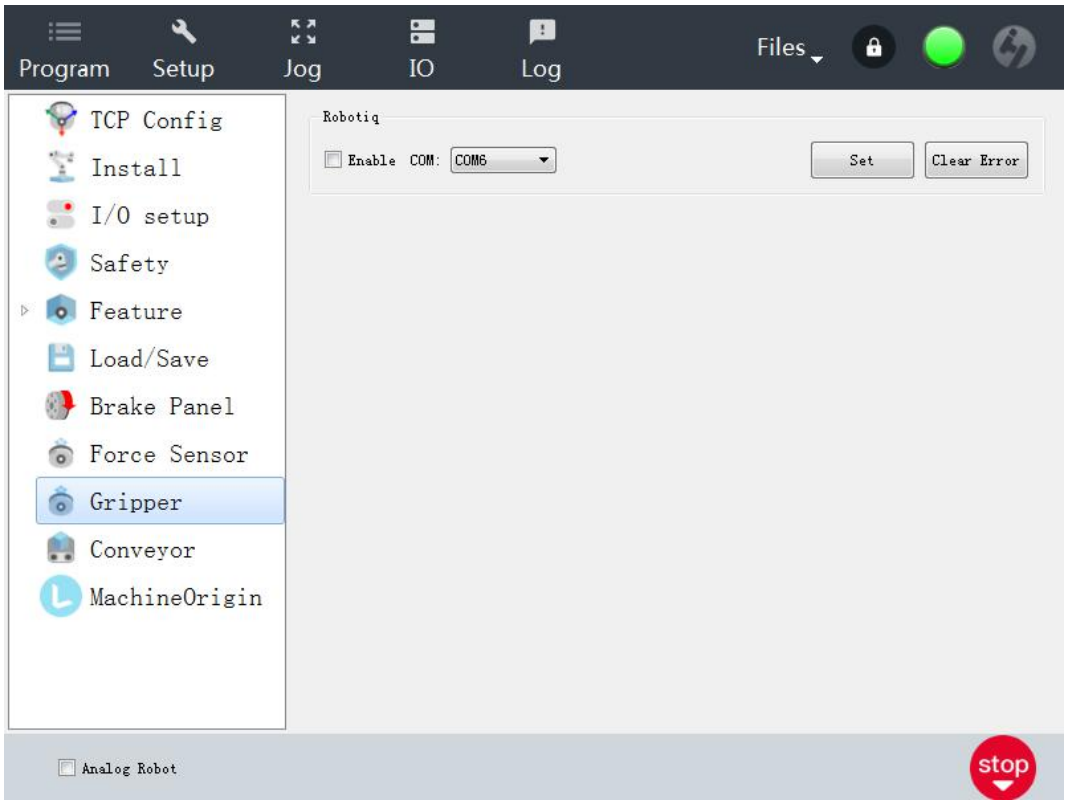
Calibrating: Calibrate the sensor by teaching the calibration points of 8 different poses.

Reset: Reset the calibration point.

Start: Start using force sensors.

Gripper

This interface is applied to the Robotiq gripper and controls the Robotiq gripper via the electrical box serial ports.



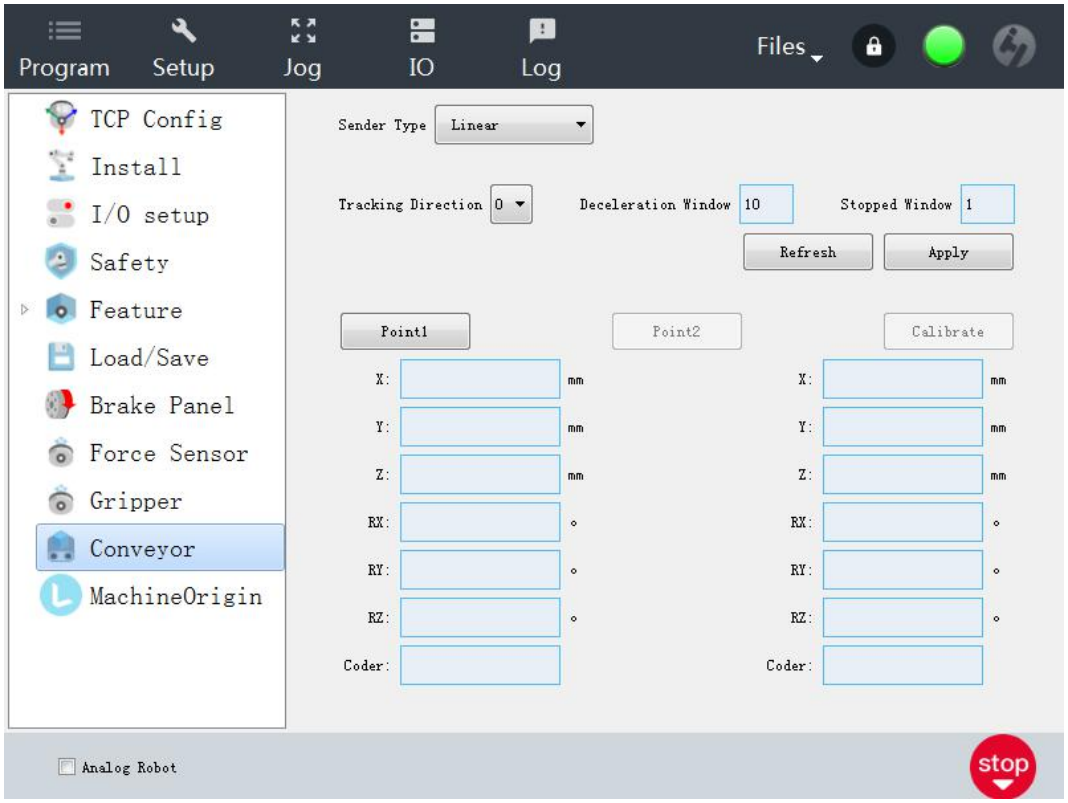
Enable:Enable Robotiq gripper.

Set:Connect the Robotiq gripper with the selected serial port.

Clear Error:Clear the Robotiq griper error.

Conveyor

When using a conveyor, the robot can be configured to track the movement and speed of the conveyor relative to the "tool center point." Conveyor Tracking Setup provides configuration options that allow the robot to work with multiple common conveyors.



Sender type:Linear is optional

Point 1 Point 2:Teaching two calibration points according to the running direction of the conveyor.

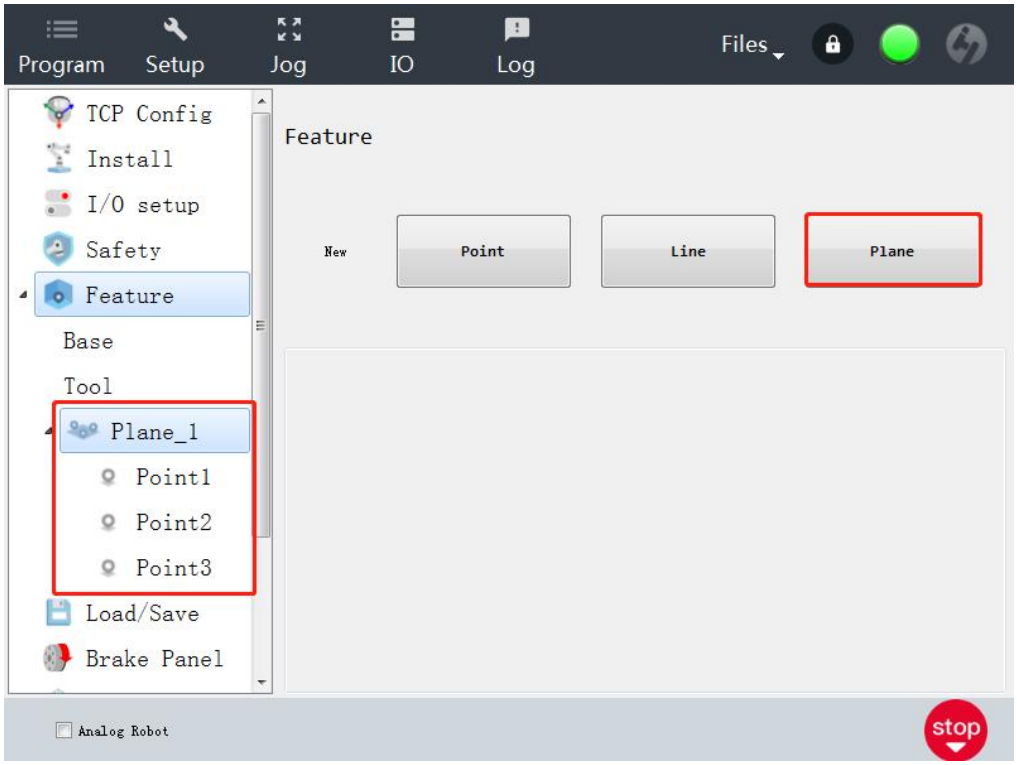
Calibrate:Start calibration.

➔ Calibration example

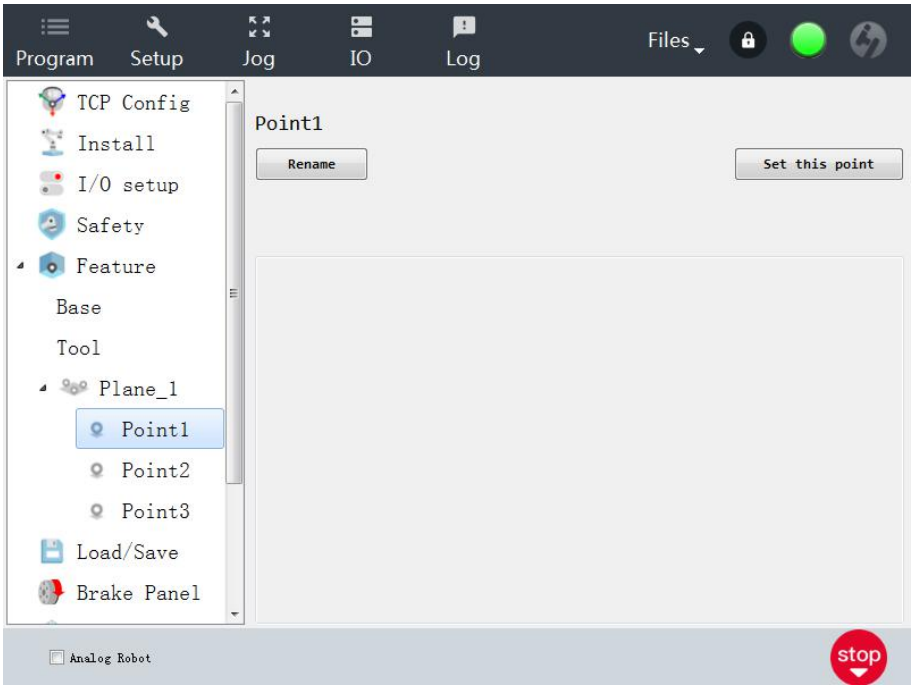
Preparation for conveyor calibration:

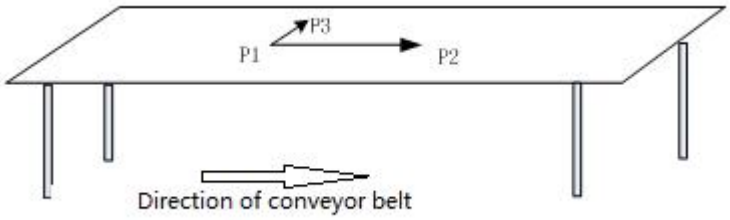
Teaching the robot user coordinates so that the Y-axis movement direction is consistent with the conveyor belt movement direction

1.Add line user coordinates, enter the Feature interface, click the "Plane" button;



2.Teach the user coordinates, move the end of the robot to the P1 point of the conveyor belt, teach the point to Point1, run the conveyor belt, move the P point to P2, stop the conveyor belt, move the end of the robot to the P2 point of the conveyor belt, and teach the point to Point2 . Then the positive direction of the Y-axis of this user coordinate is the same as the running direction of the conveyor belt, and the third point is taught to make the 3-point plane parallel to the plane of the conveyor belt.

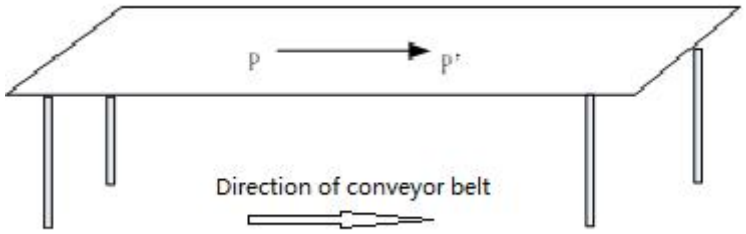


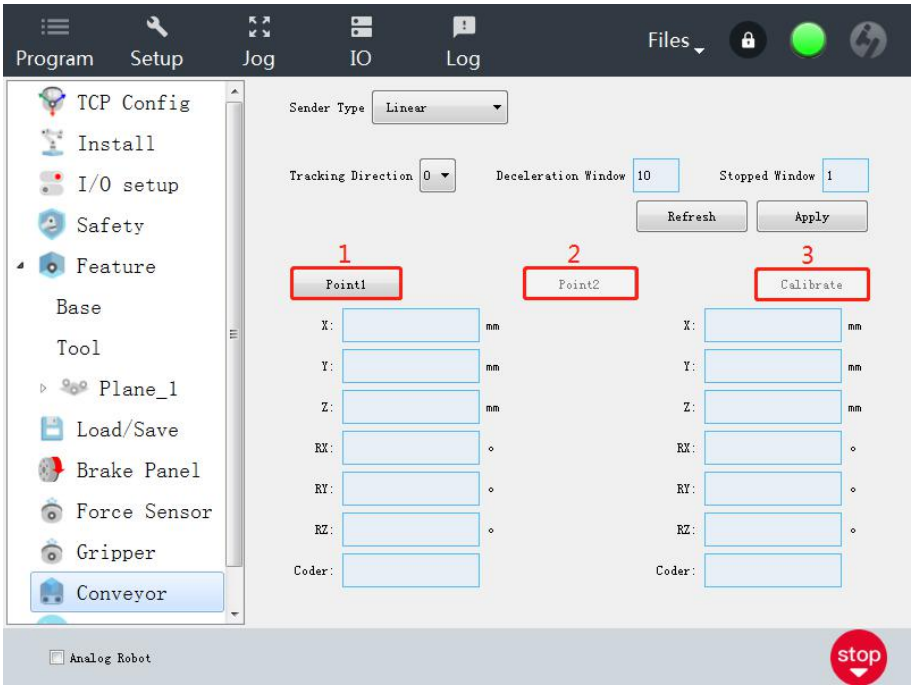


Conveyor calibration parameters

(1)Set Plane_1 to current user coordinates;

(2)Move the end of the robot to the P point of the conveyor belt, click "point 1", run the conveyor belt, move the P point to P', stop the conveyor belt, move the Y-axis, make the end of the robot to the P' point of the conveyor belt, click point 2. Click "calibrate" again, then the conveyor calibration is completed.



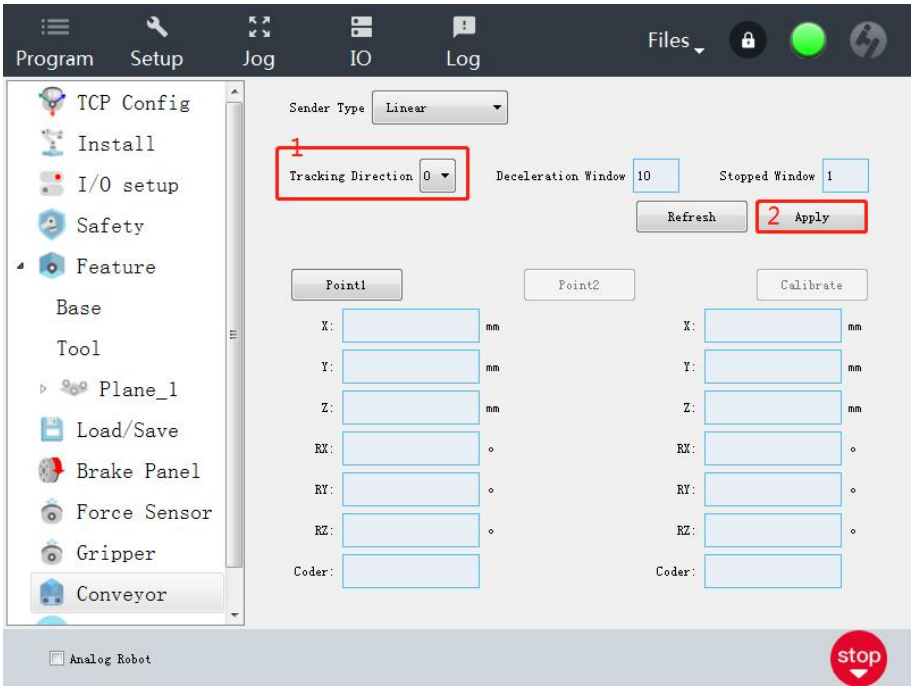


(3)Tracking direction:

1.The Y+ direction of Plane_1 is in the same direction as the direction of the conveyor belt. Select 1;

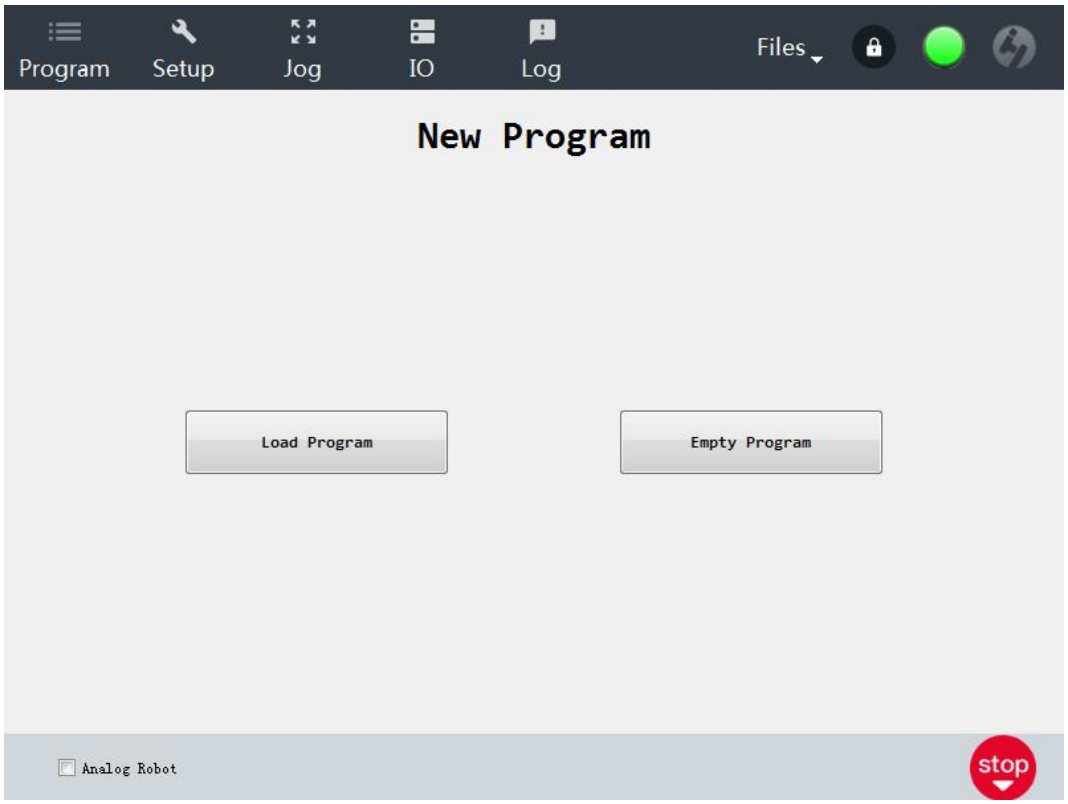
2.Plane_1's Y+ direction is opposite to the direction of the conveyor belt, select 0;

3.Click the "Apply" button;

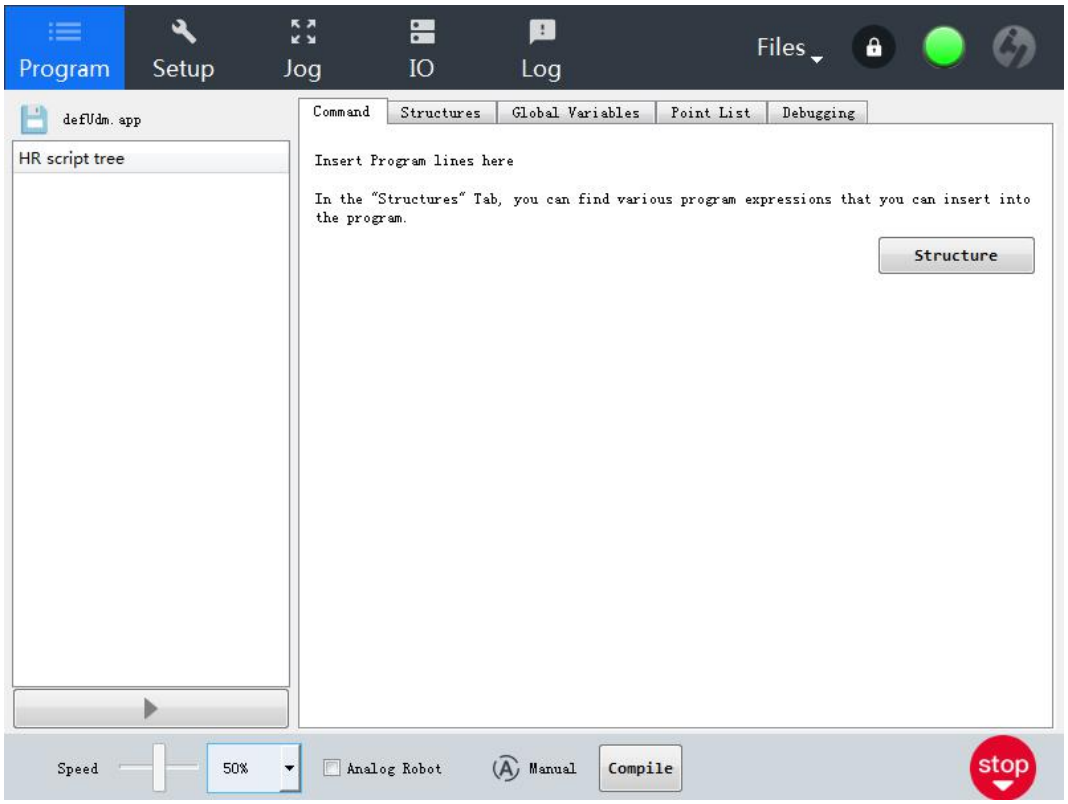


1.7 New Program

A new program can be created by an empty program or by referring to existing (saved) robot programs. Click the "Files" in the upper right corner to select "New" to enter the following interface, click "empty program" to create a new empty program, "Load program" can load the script program saved by the user.

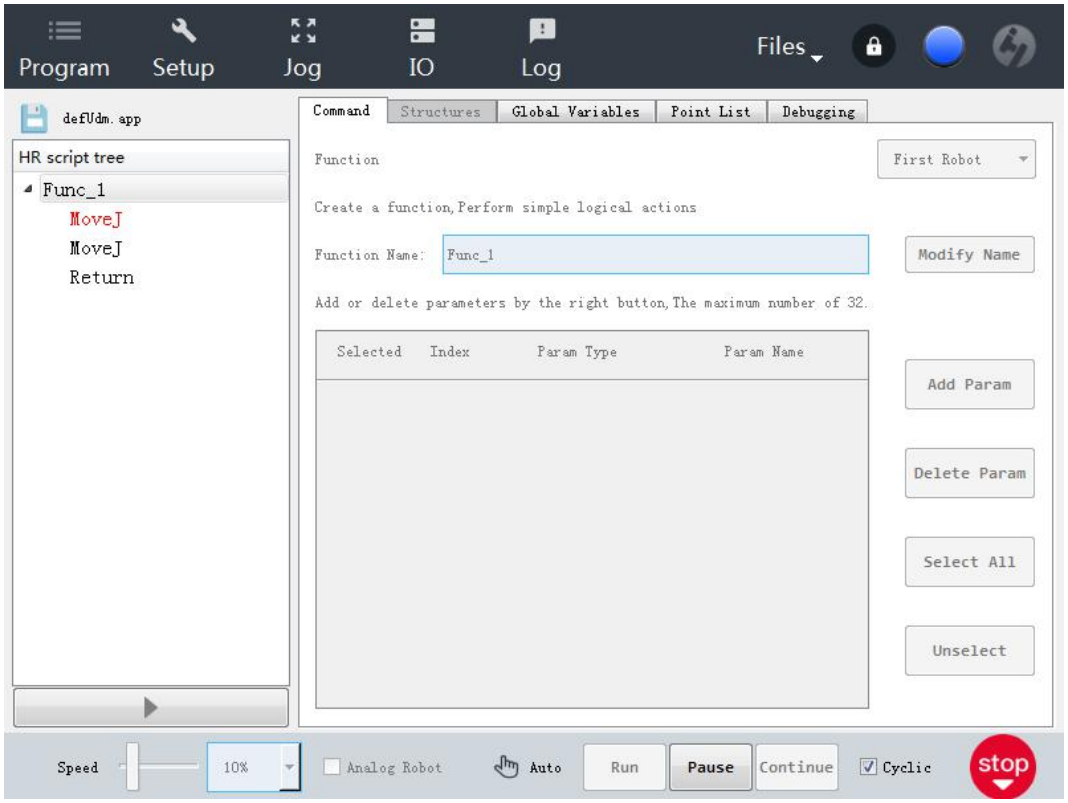


Program



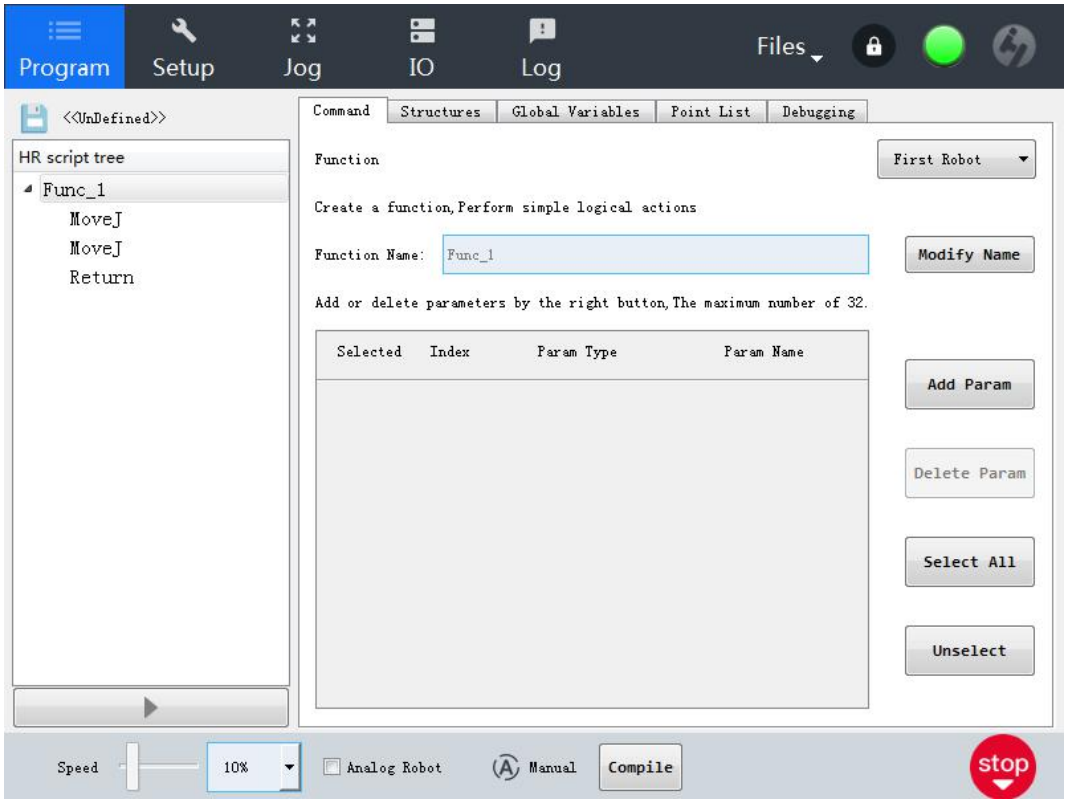
Script tree:The script tree on the left side of the screen displays the program as a command list, and the right area of the screen displays information related to the current command. The current command can be selected by clicking on the command list. You can use the Structure tab to insert or remove commands. Arrow button below the script tree  ,Click this button to expand the display of the script tree. The program name is displayed directly above the script tree.

In the script tree, the currently executing command is displayed in red, as shown in the following figure.

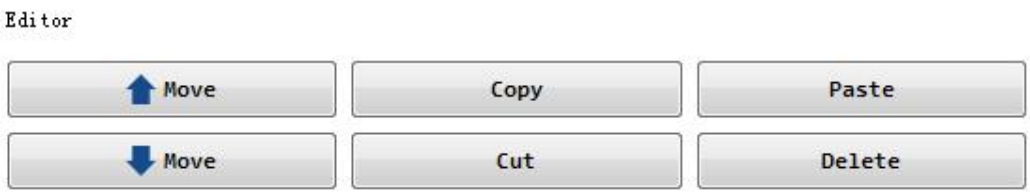

Program
Setup
Jog
IO
Log
Files
defUdm.app
HR script tree
Func_1
MoveJ
MoveJ
Return
Command
Structures
Global Variables
Point List
Debugging
Function
First Robot
Create a function, Perform simple logical actions
Function Name:
Func_1
Modify Name
Add or delete parameters by the right button, The maximum number of 32.
Selected
Index
Param Type
Param Name
Add Param
Delete Param
Select All
Unselect
Speed
10%
Analog Robot
Auto
Run
Pause
Continue
Cyclic
stop

Program--Structures

Click the Structure button to go to the Structure tab, where you can find a variety of optional commands, from which the commands in the program tree need to be inserted. After specifying and defining all program lines, switch to auto mode. After the compilation is successful, the program can run.



In “Structures”, the user can insert a number of commands into the program tree, and the user can freely edit the custom application script. You can also use the Copy, Delete, Cut, Paste and Move to operate the command line accordingly.



Copy: Copy a node or function and allow it to be used for other operations (for example, paste it elsewhere in the program tree).

Cut: Cut a node or function and allow it to be used for other operations (for example:

paste it elsewhere in the program tree).

Paste: Paste the node that was cut or copied before.

Delete: Remove a node or function from the program tree.

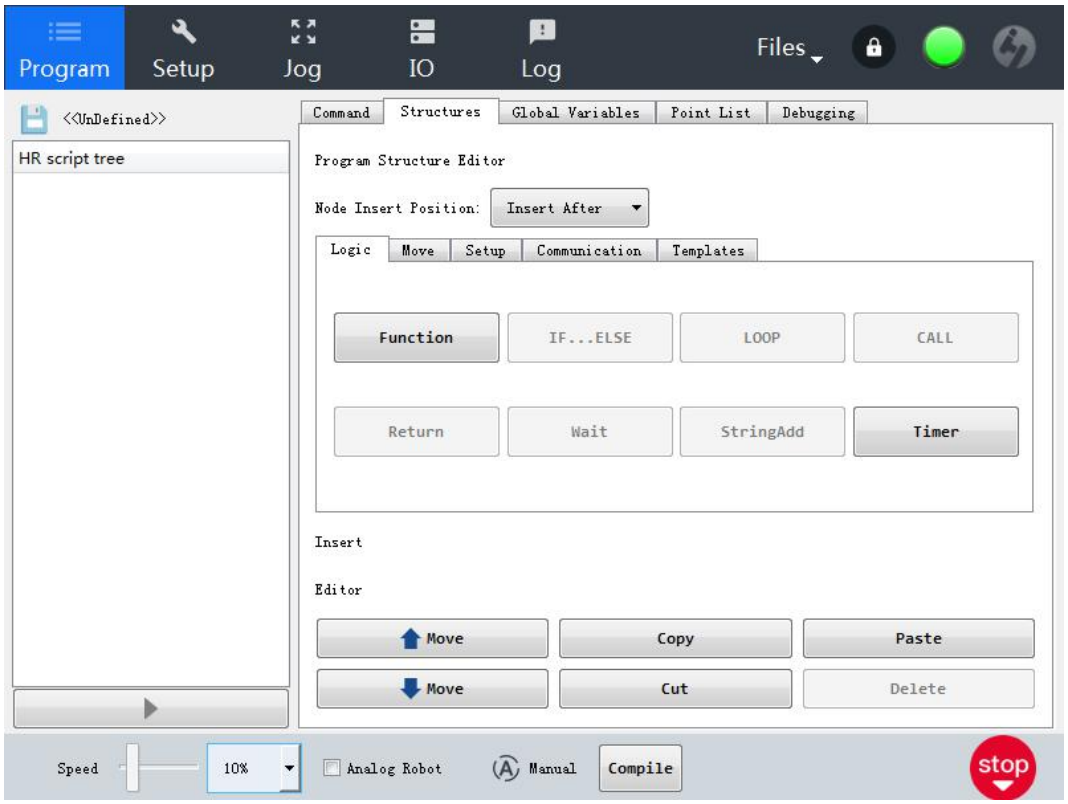
Move **Move:**Change the position of the node.

In the "Structure" column, you can add the following instructions to the program tree by clicking the desired command button. After adding the command, you need to edit the contents of the command in the "Command" column. The meanings of the buttons and instructions in the "Structure" column are as follows:

Node Insert Position:"Insert Before" will insert an instruction above the currently selected command line, and "Insert After" will insert an instruction below the currently selected command line.

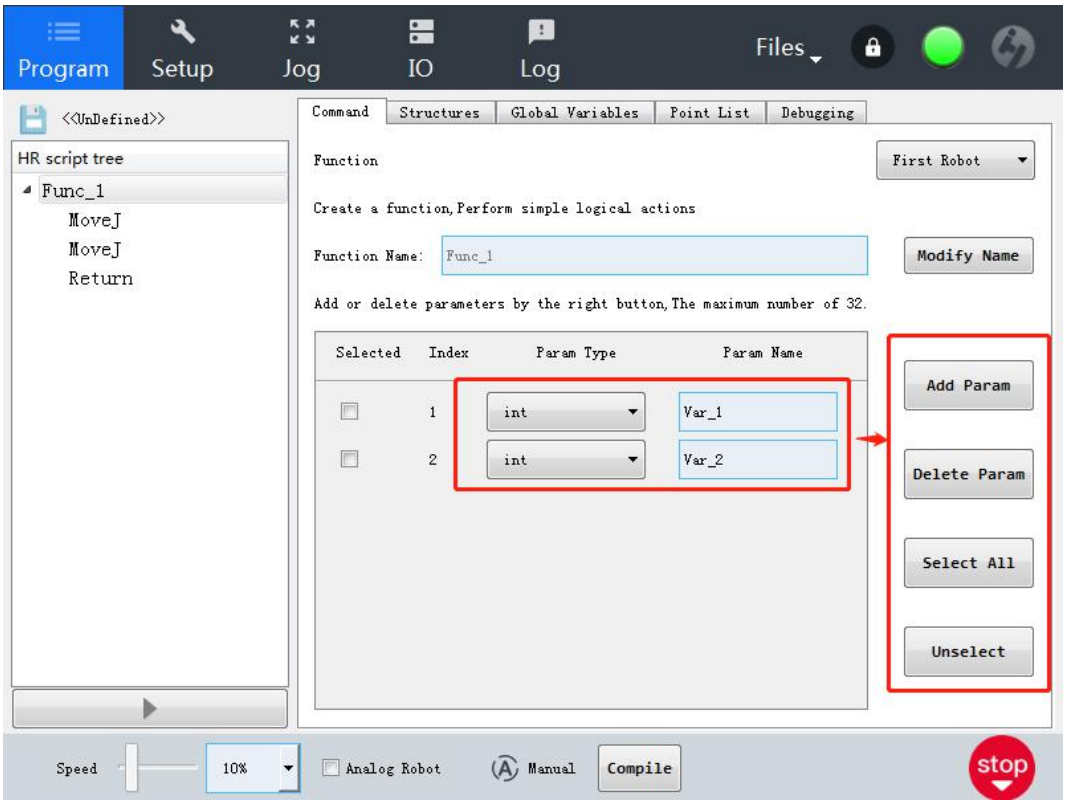
Logic

This part is a logical correlation function, you need to add logic instructions to insert instructions from this page.



➔ Function

Before adding a new function, you need to add a function. You can add instructions to the function. Each function ends with a Return command. A script can have multiple functions, and functions can call other functions by using a call instruction (Call).



Modify Name: The default function name is Func_num (num=1, 2, 3...), which can be modified to describe the function name of the function.

Add Param:Add a function local variable (int/doubot), when calling this function, you need to pass the corresponding number of parameters.

Delete Param:Delete the function local variable, check the "selected" option of the corresponding parameter, click the delete parameter button to delete the selected local parameter.

Select All:Select all local variables.

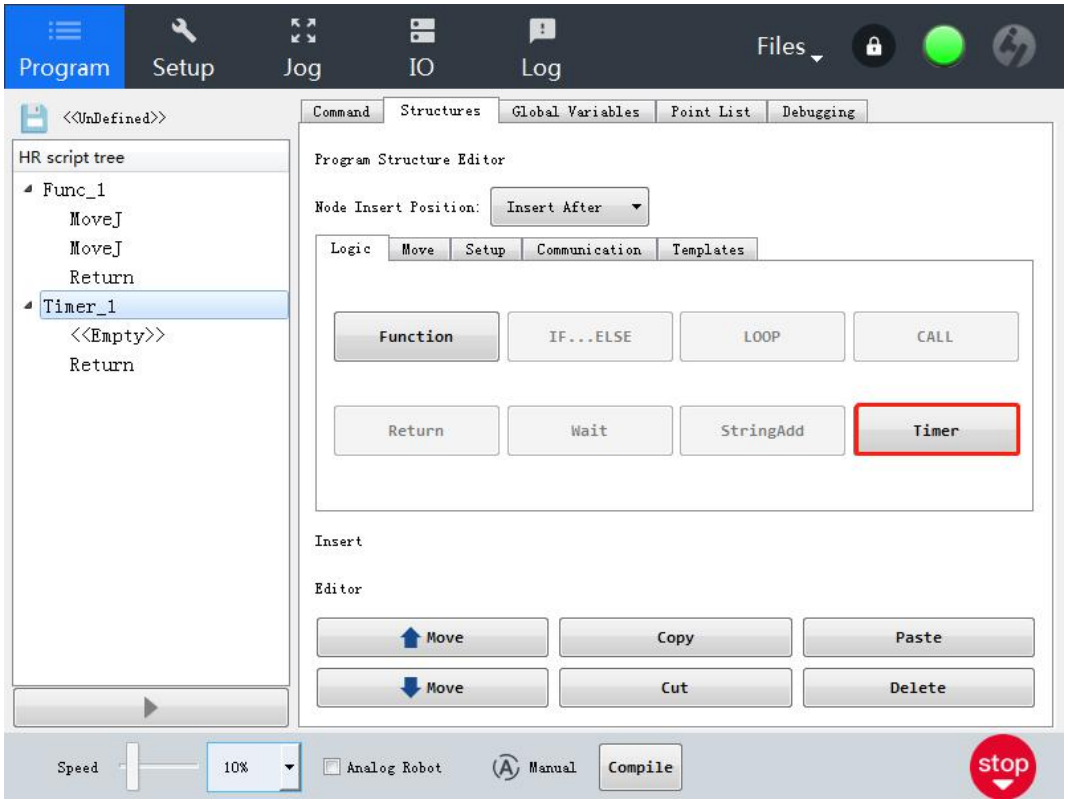
Unselect:All local variables are not selected.

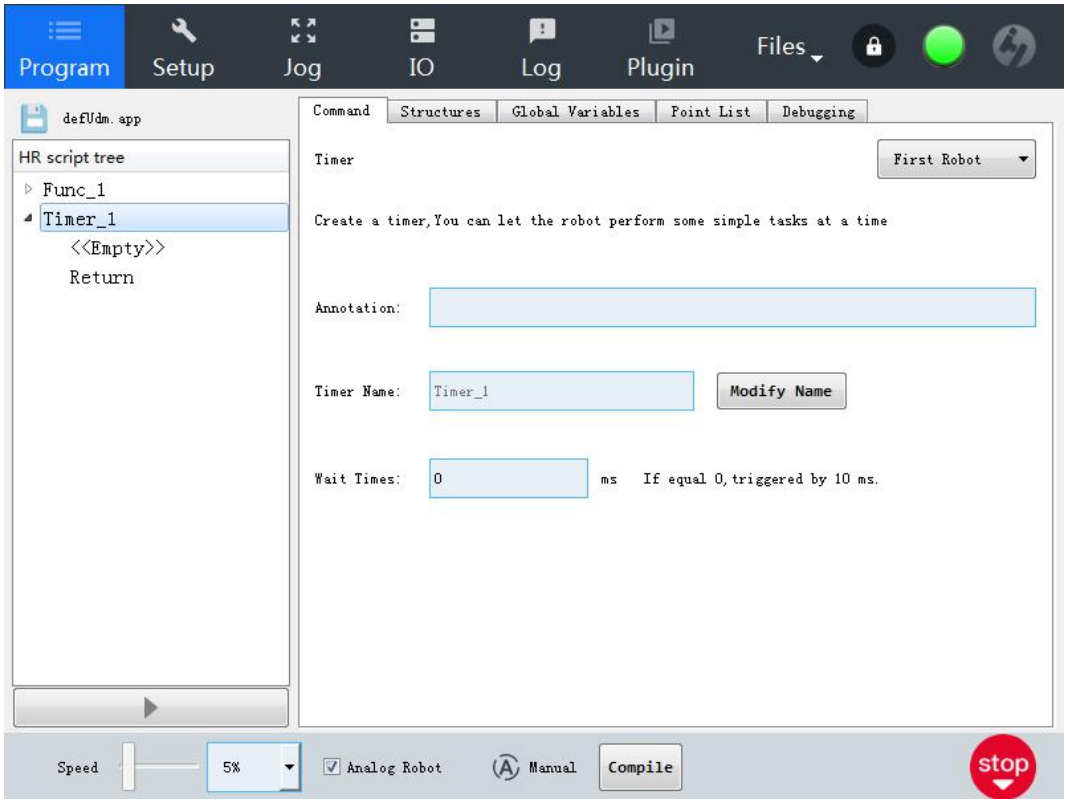
➔ Timer

Regularly execute the instruction of the timer structure to realize the real-time acquisition of the IO state. The timer function can be run in parallel with the Function function (up to 5 Timer functions can be run simultaneously). In the program tree, the timer and function are in the same structure.

Timer prohibits adding motion commands.

Note: The timer can be paused after the timer is turned on. The fifth timer is the exception and will not be suspended.





Annotation:Optional, describing the role of this timer.

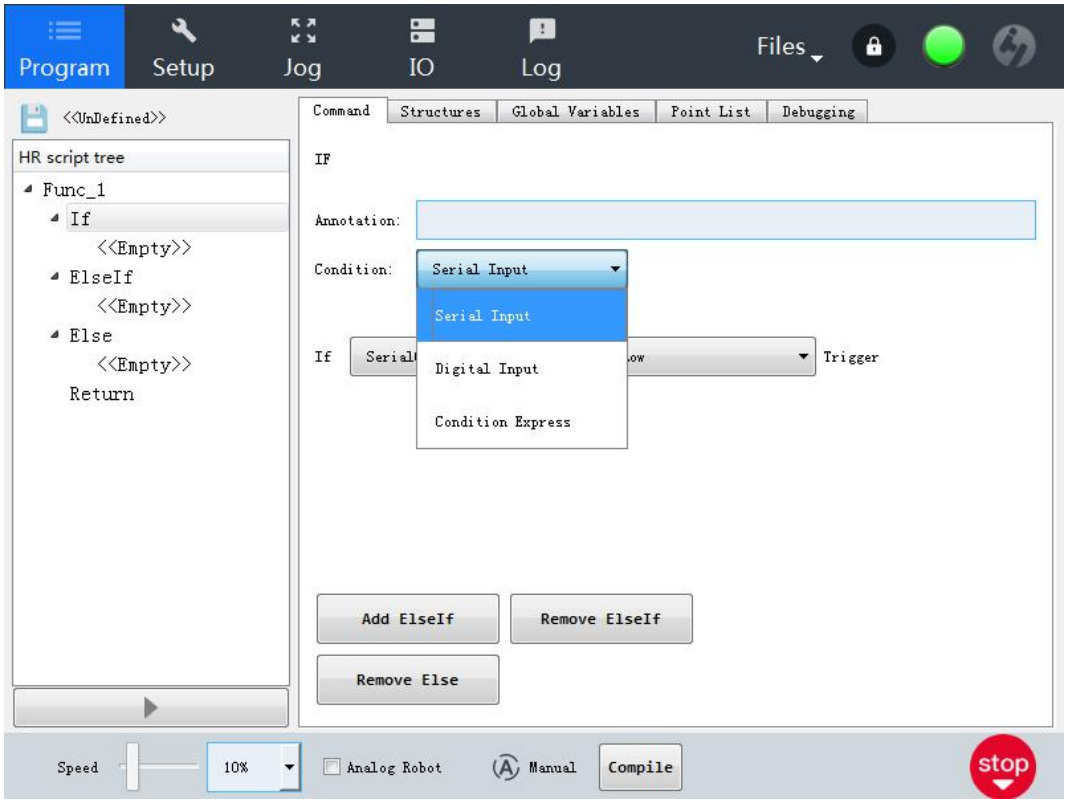
Modify Name:The default timer name is Timer_num (num=1, 2, 3...), which can be modified to describe the name of the timer function.

Wait Times:The timer execution period, if the waiting time is longer than the execution time of the timer structure instruction time, the execution period is the set waiting time. If the execution time of the timer structure instruction time is longer than the waiting time, the execution period is the execution time of the timer structure instruction time.

➔ IF...ELSE

A regular judgment command is added, and the "if...else" structure can instruct the robot to change its behavior according to whether the IO input state or the set judgment condition is satisfied or not. If the condition is true, the command line inside this If command will be executed. Each If command can contain multiple ElseIf and one Else commands. If the expression evaluates to False during the execution of the If part, it will go to the next ElseIf or Else.

The If command can have multiple ElseIf statements, which can be added and removed using the Add ElseIf and Remove ElseIf buttons. However, the If command can only have one Else statement.



3 kinds of judgment conditions provided:

Serial Input:check the input IO status of the electric box.

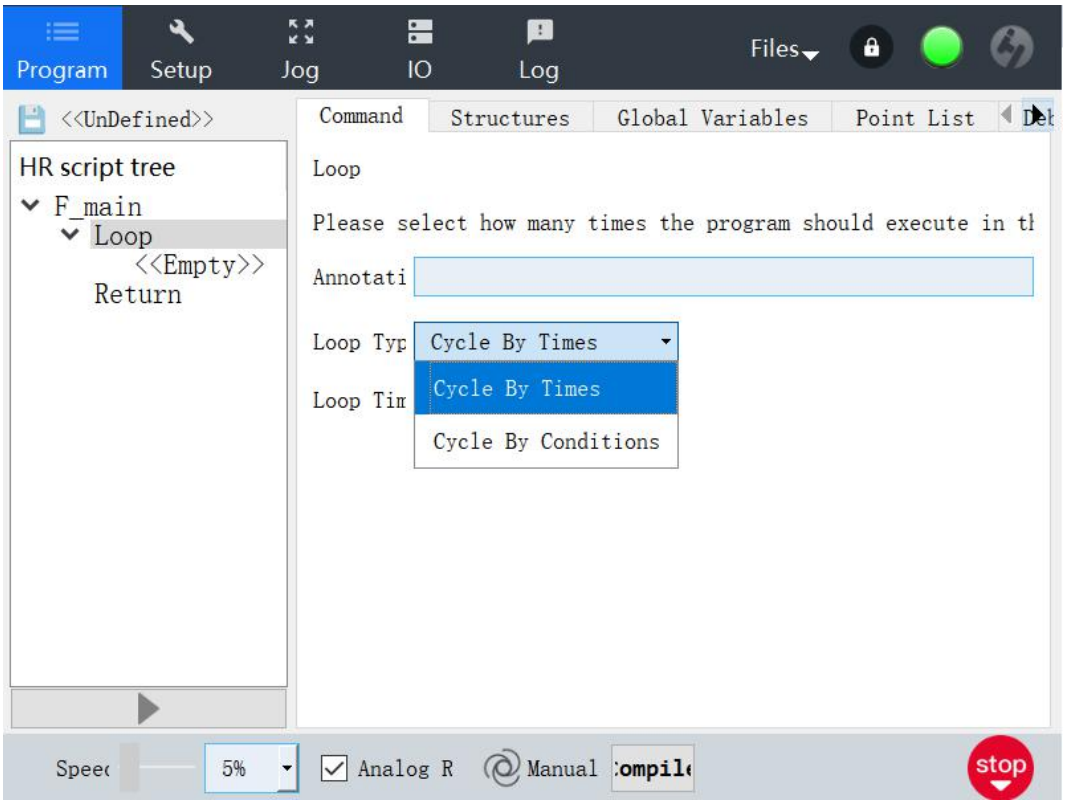
Digital Input:check the input IO state at the end of the robot.

Condition Express:Custom conditional expression judgment.

➔ Loop

Add a loop instruction, which is used in the internal loop of the program until the condition judged by LOOP is false, then jump out of the loop.

Loop conditions can choose the number of loops or a custom conditional expression.

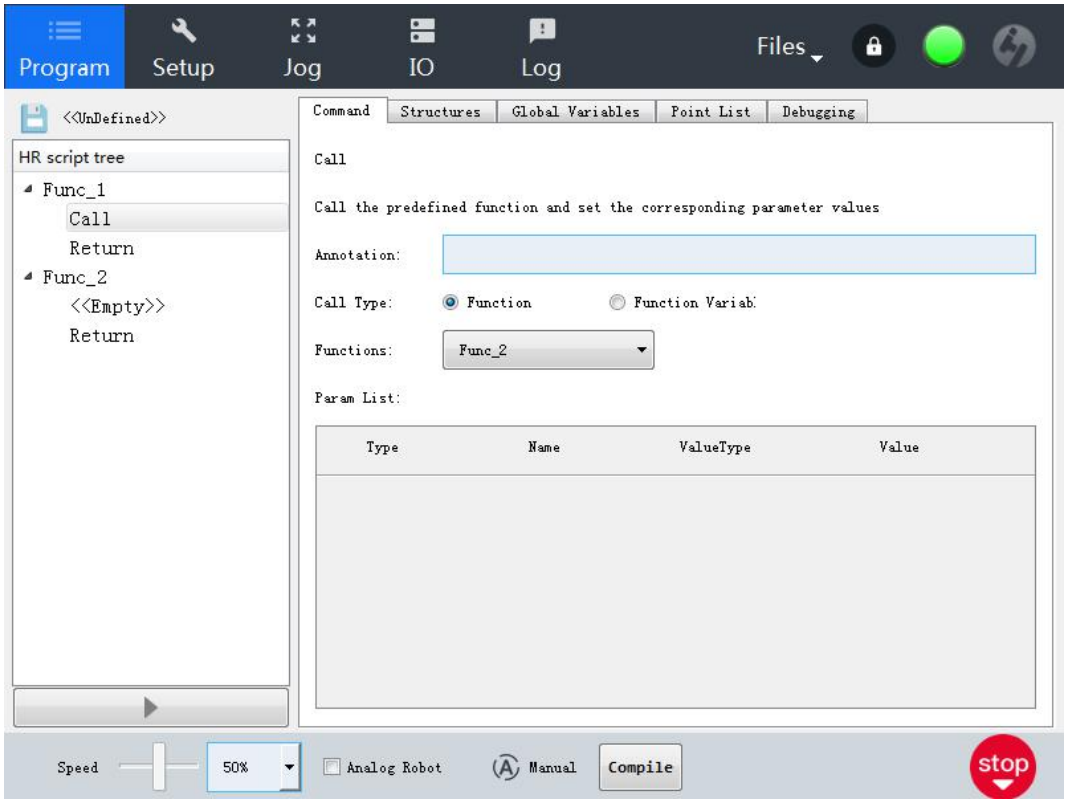


➔ Call

The CALL command,the logic is to call other functions in the function. If the called function has local variables,the calling function needs to pass in matching parameters.

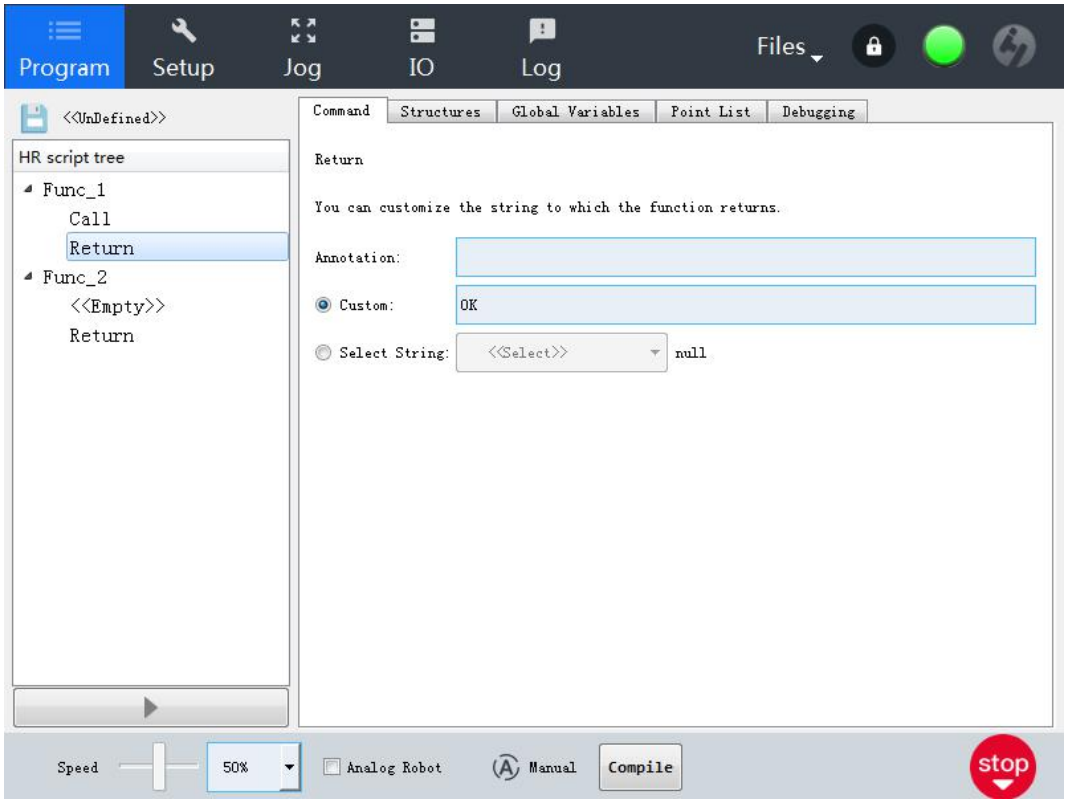
You can write a reusable operation as a function for repeated calls. Or decompose the process into multiple single module functions, improve the script readability by calling the function integration in the main function.

Call type:The function, the list includes other functions in addition to the function of the call instruction; the function variable, the list is a string, the function corresponding to the value of the string is called, and the use with StringAdd can simplify the function with higher repeatability. See section 7.6 for a simplified script method.



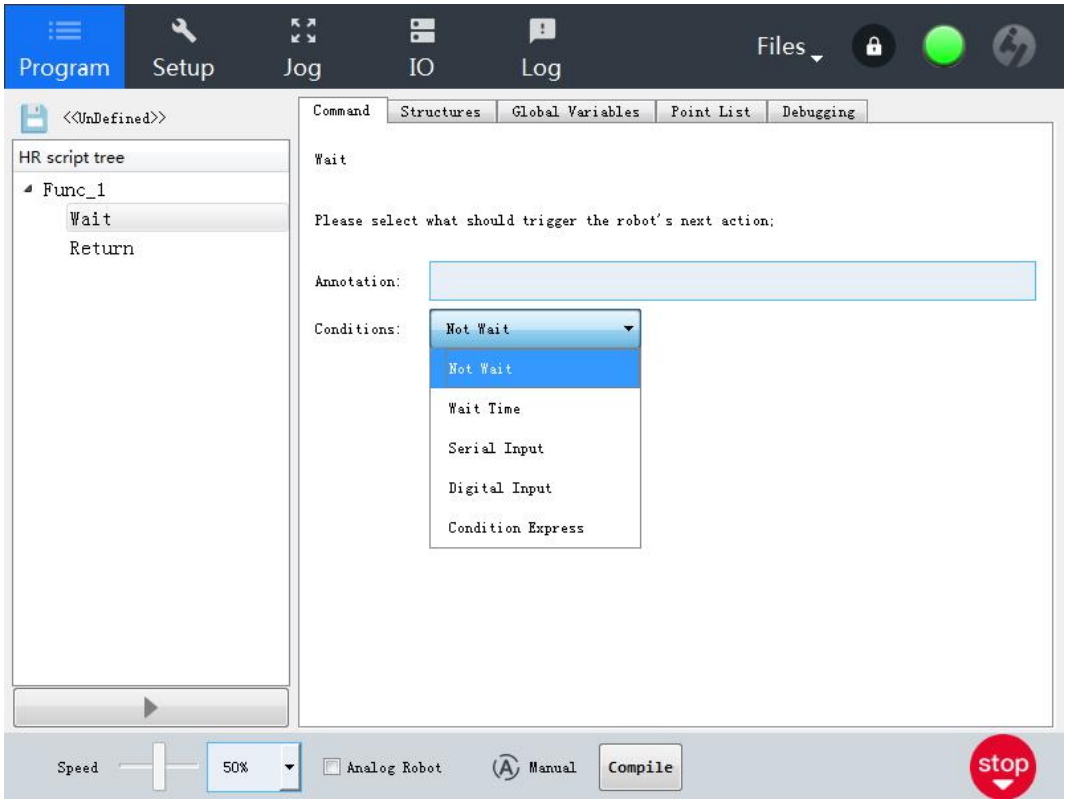
➔ Return

This logic is used at the end of the function to end the program. Can choose to return a custom string or variable value.



➔ Wait

Add a wait instruction to the function, wait for the end of the waiting time when the program is run, or wait until the set condition is met before the program can continue to execute.



4 kinds of waiting conditions provided:

Wait Time: Waiting time for setting.

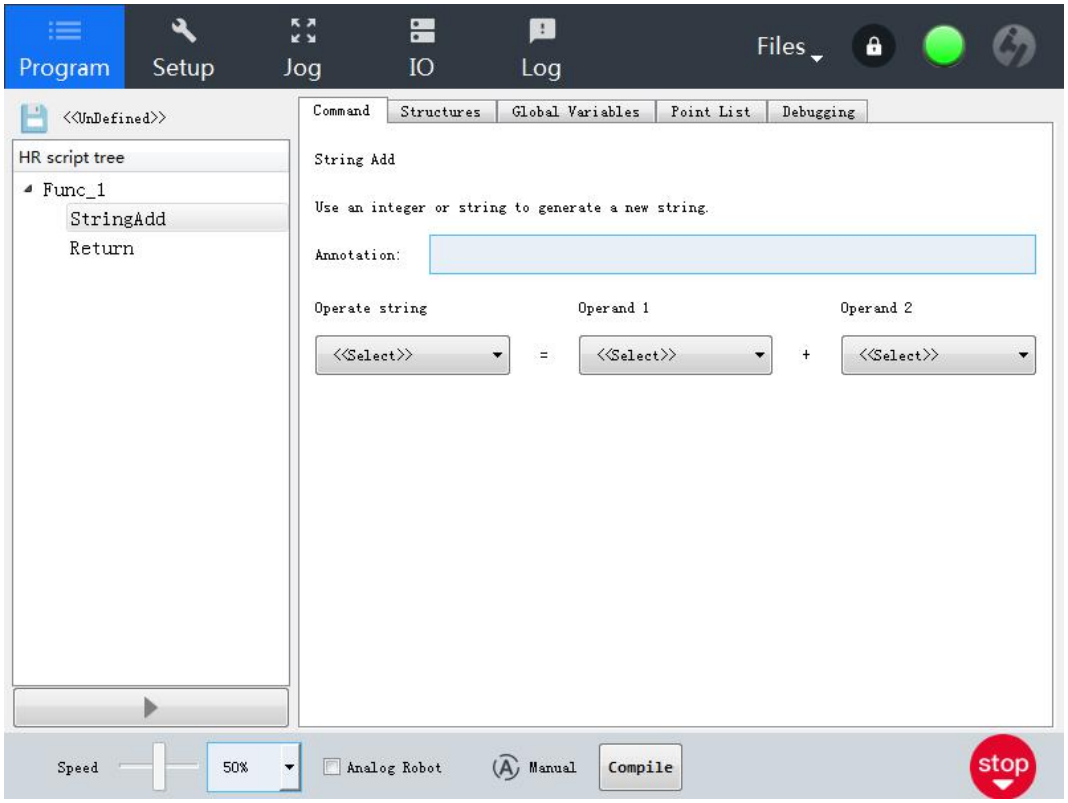
Serial Input:check the input IO status of the electric box.

Digital Input:check the input IO state at the end of the robot.

Condition Express:Custom conditional expression judgment.

➔ StringAdd

The result of adding two variables (string/integer) is assigned to a new string variable, usually used with the Move/Call directive. Simplify scripting by changing the value of the variable Num, in conjunction with the Move/Call instruction, moving different points/calling different functions. See section 7.6 for a simplified script method.

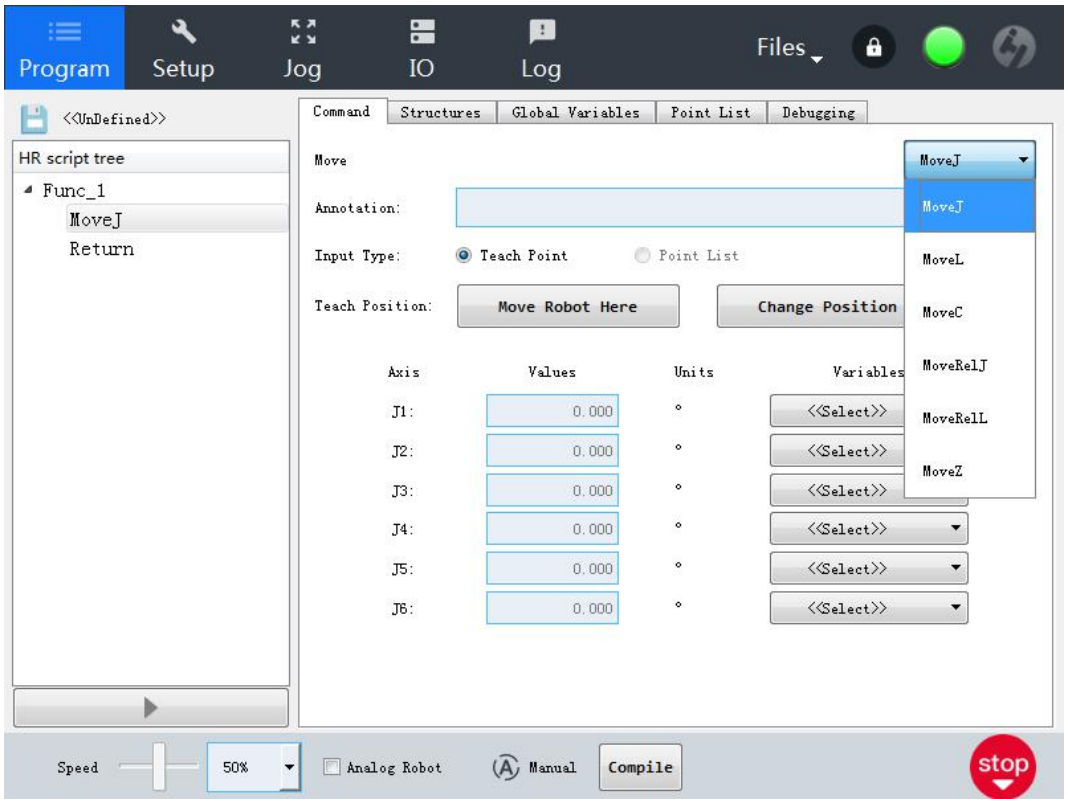


Move command

This part is a move-related function, you need to add move commands to insert commands from this page.

➔ Move

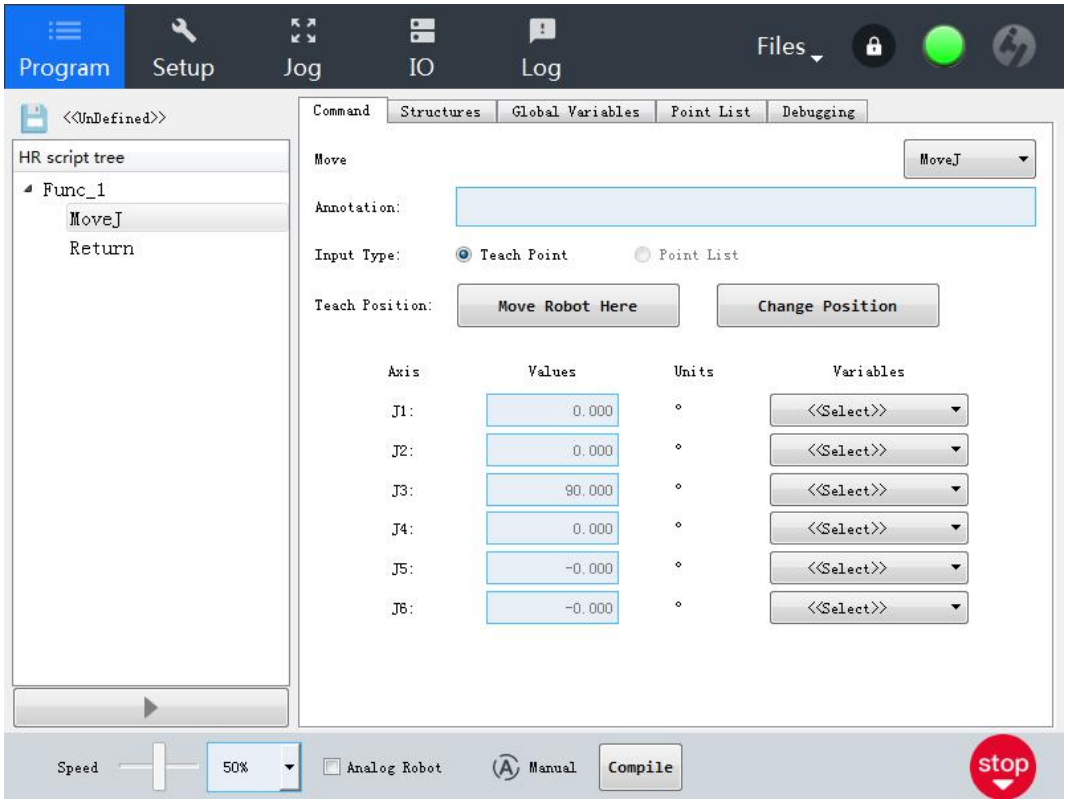
By teaching, add spatial linear move command (MoveL), joint move command (MoveJ), circular move command (MoveC), Z-move command (MoveZ), and single-axis move command (MoveRelL, MoveRelJ).



MoveJ

All of the robot's joint axis move quickly from one point to another. Applicable to occasions where there is no requirement for the end motion track of the robot.

The end of the robot moves along an irregular curve, so be aware of the danger of collision. If you want the robot to move in a straight line, you can use the spatial linear motion command(MoveL)



Input Type:Manual teaching will jump to the jog interface, the teaching will display the point below;point list, select the point in front of the point column to indicate the taught point or string (the variable value corresponds to the point list);

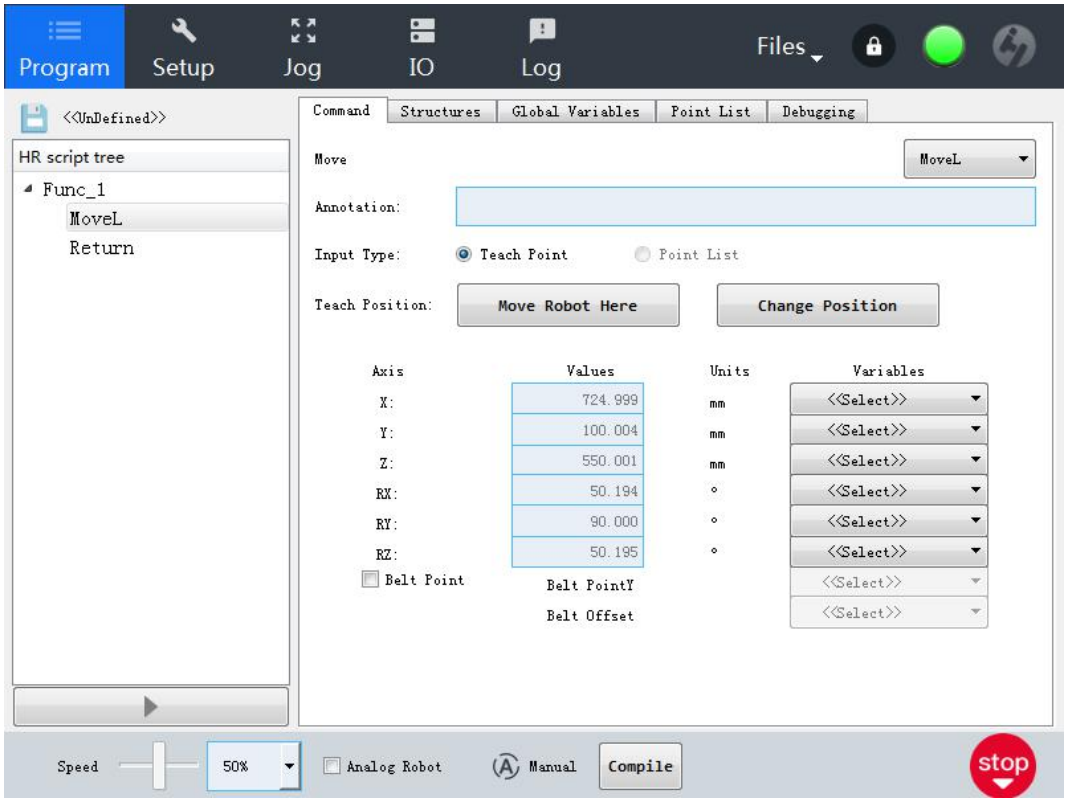
Move Robot Here: Move to the teaching point;

Change Position:Re-teaching the points;

Variables:If a variable is selected, the motion to point position when the command is executed is the axis position value plus the variable value.

MoveL

The robot moves the tool center point TCP from one point to another in a linear motion. When the starting point and ending point postures are different, the posture will be rotated to the endpoint in synchronization with the position.



Belt Point:Check to turn on the conveyor track. The target point is the position value instead of the position value plus the variable value.

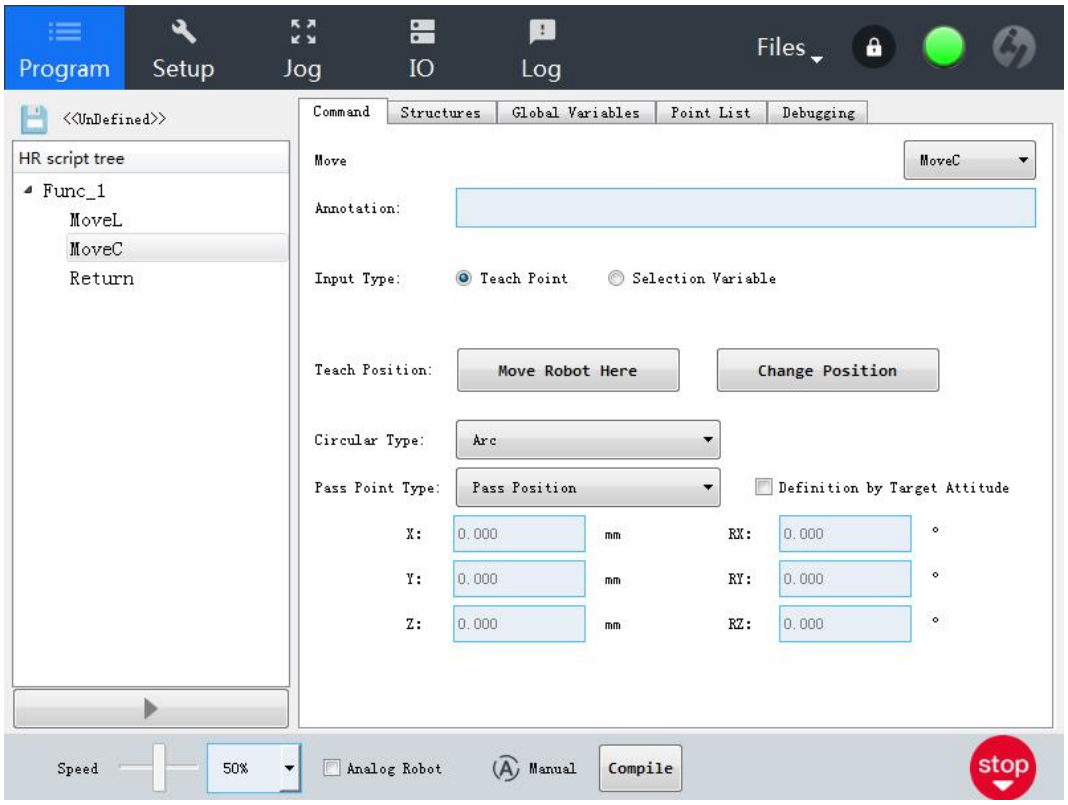
Belt PointY:Encoder value,unit:mm;

Belt Offset:Position compensation caused by calibration of the camera;

Detailed instructions for the use of the conveyor belt are provided in additional documents, which are available upon request.

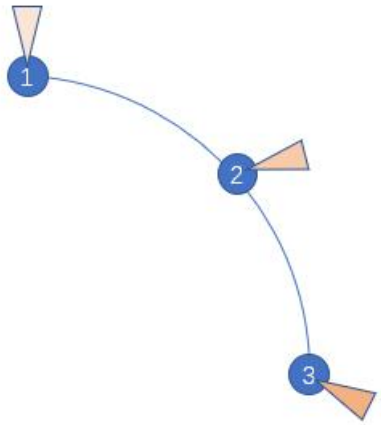
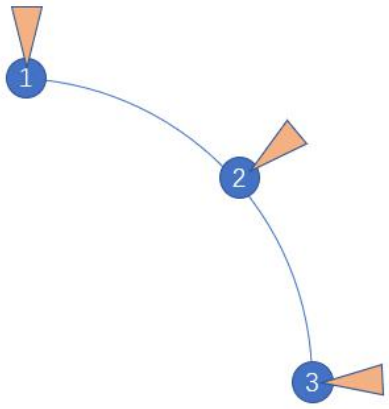
MoveC

The MoveC command realizes the circular path motion and uses a three-point determination circle to determine a circular path (full circle or arc). Three points cannot be in the same position or on a straight line, otherwise, a circular path cannot be determined.



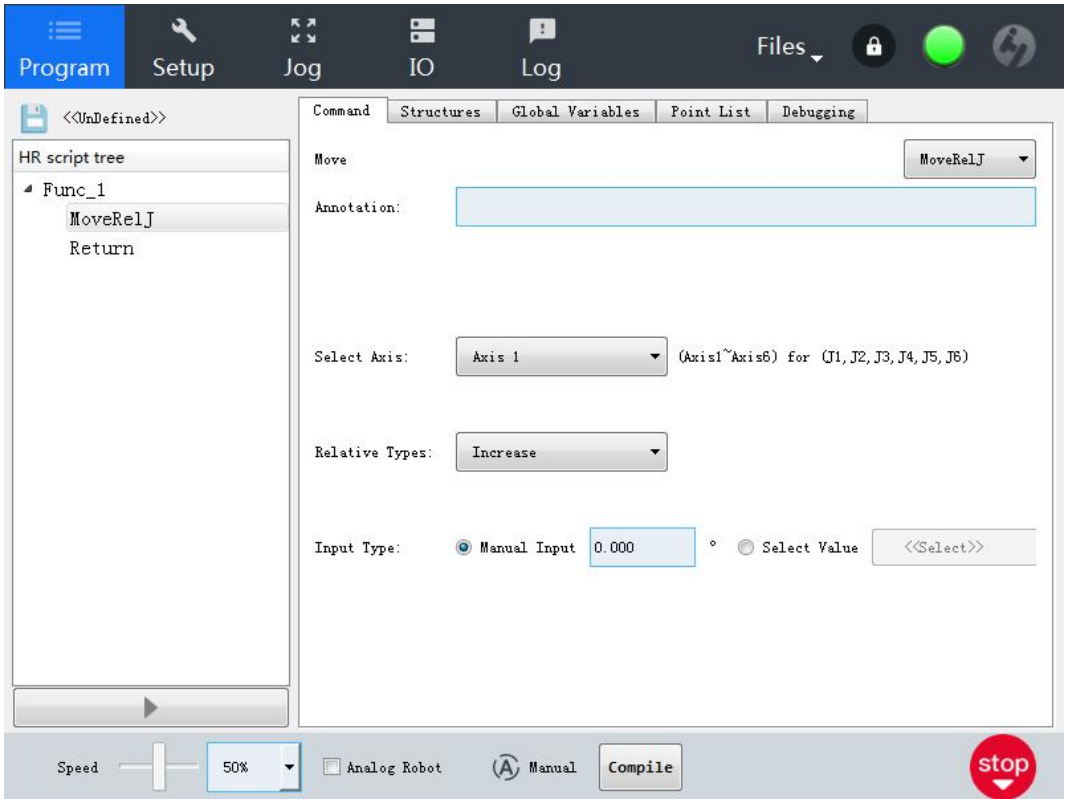
The function description is shown in the table below.

Button	Description
Circular type	There are two types to choose from Arc Full circle
Current Position	The current position of the robot, the starting point of the arc
Pass Position	A point through which a circular path passes can be used to "manually teach" PCS coordinate values, or to "select variables" to pass in coordinate values.
Target Position	The target point of a circular trajectory, if it is a full circle, is also considered as a passing point. Can be taught as PCS coordinate values, or can be passed in via variables.
Circle Number	If the arc type is selected as a full circle, the number of times the

	robot repeats the track
Attitude type	Two types: ① If not select "Definition by Target Attitude", the posture of the attitude transition from the initial point to the end point has nothing to do with the attitude of the second point. The effect is as follows, point 1 is the starting point, and point 2 and 3 are the passing point of the arc, the target point.  ② If not select "Definition by Target Attitude", point 1 is the starting point, and point 2 and 3 are the passing points of the arc, the target point. Based on the tool attitude of the initial point, from the starting point to the end point, the effect is as shown below: 

MoveRelJ

Specify the movement of a single joint axis of the robot, you can specify the fixed distance of motion or move to a fixed position.



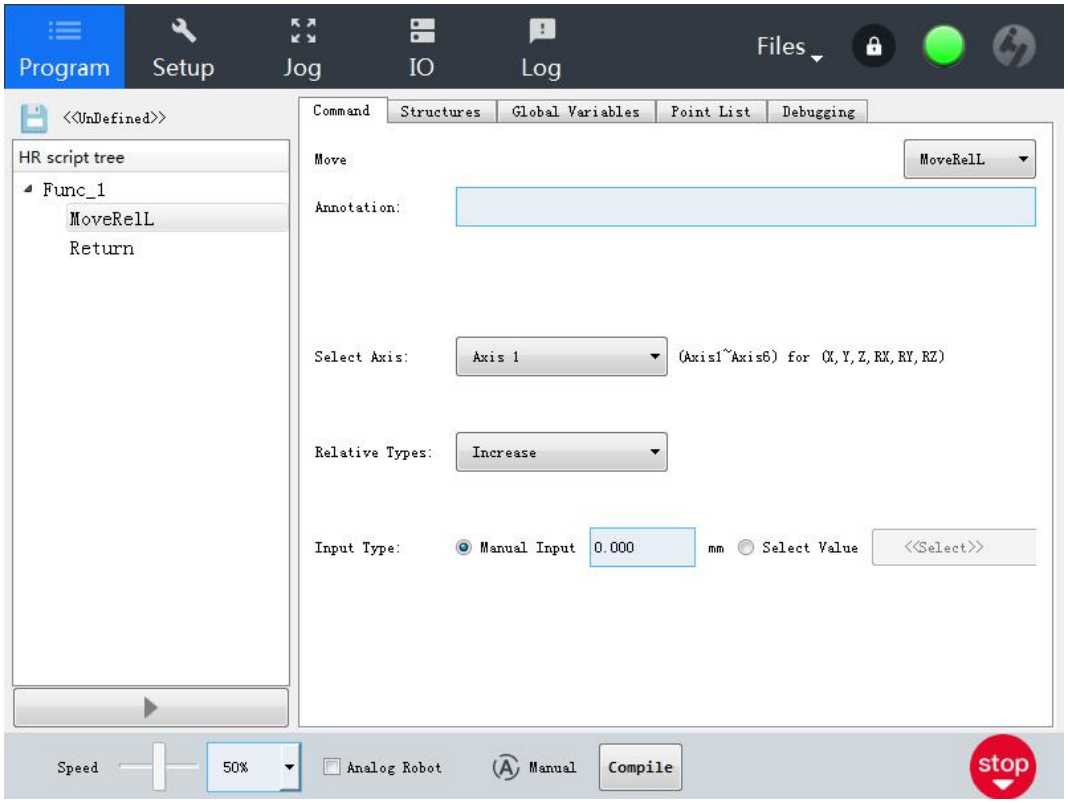
Select Axis: Axis1~Axis6 means joint axis 1~6;

Relative Types: Increase, the distance of increase value of motion at the current position; absolute value, motion to absolute position;

Input Type:custom value or variable value is available;

MoveRelL

Specify the linear motion of the robot's single spatial axis, you can specify the fixed distance of motion or move to a fixed position.



Select Axis: Axis1~Axis6 are X、Y、Z、RX、RY、RZ;

Relative Types: The Increase value, the distance of the increased value of the motion movement at the current position; the absolute value, the motion to the absolute value position;

Input Type:custom value or variable value is available;

MoveZ

The MoveZ command is used to implement a Z-shaped trajectory between two points. The schematic diagram is as follows, and the red trajectory is a Z-shaped trajectory. (Note: At the peak of the Z-shaped trajectory, the actual trajectory is smoother than the illustrated peak)

Teaching 3 points：

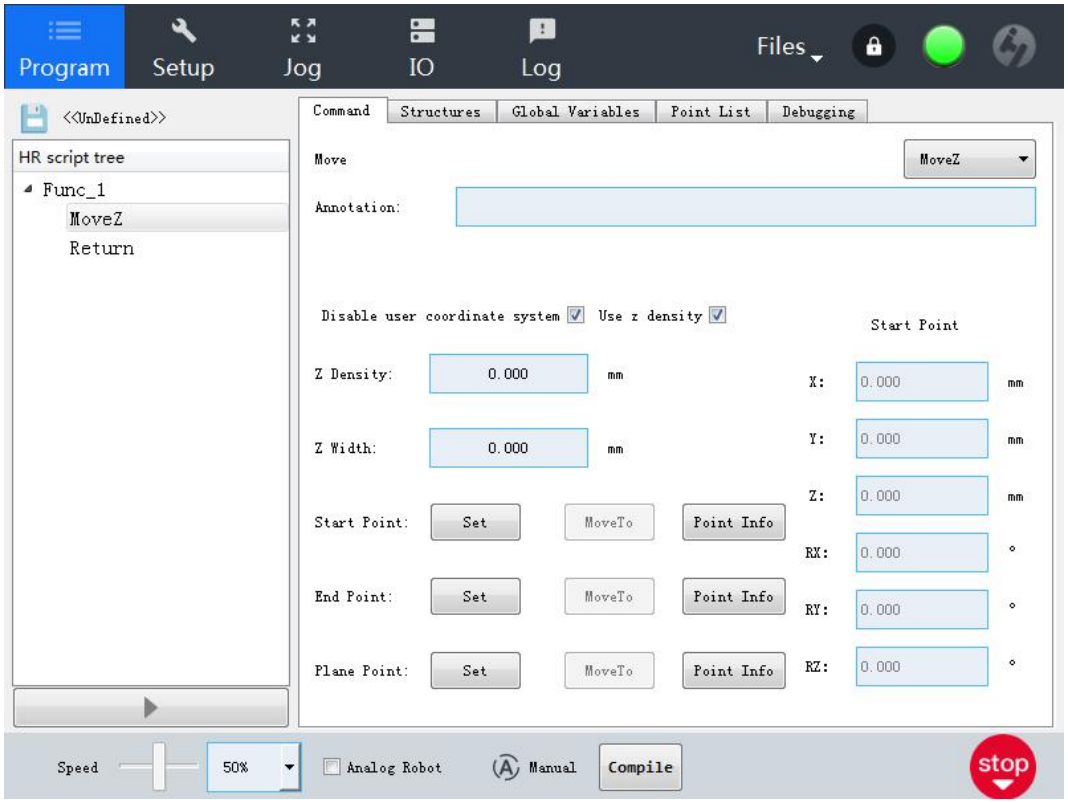
① Start Point： p1

② End Point： p2 （The line connecting p1 and p2 is the center line of the Z-track width line.）

③ Plane Point： p3（Used to determine the plane of the MoveZ track）

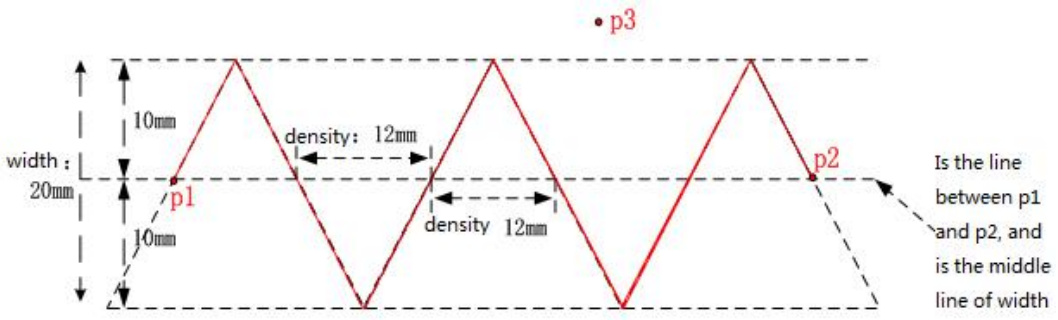
Z Width:As shown in the figure below, the vertical distance between the two peaks of the Z-shaped track;

Z Density:There are two ways to set up. One is to use the Z-density, the density is the artificial setting value. If it is not checked, the density is determined by the movement speed of the robot. The two are linear, and the ratio can be set in the configuration file. The faster, the denser the density; the slower the speed, the denser the density):



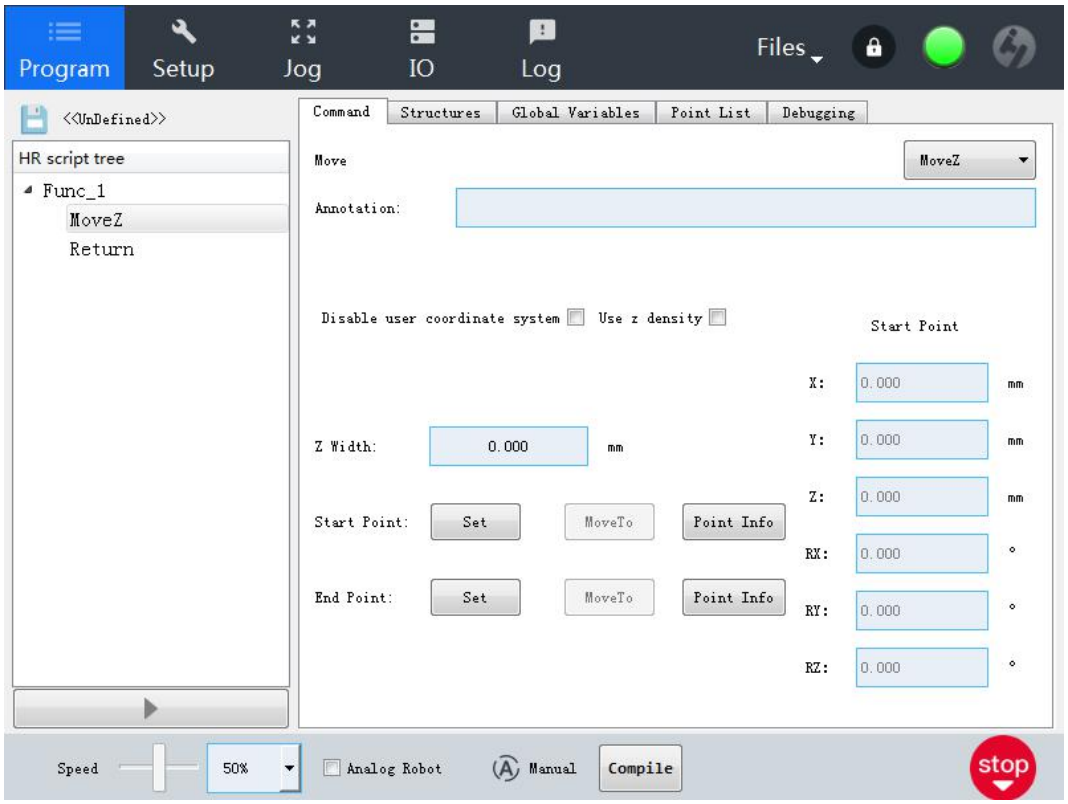
The following table takes the density parameter set to 0.02cm as an example to show the relationship between speed and density.

	Robot speed	Z Density
2^1	2%	8mm
2^2	4%	4mm
2^3	8%	2mm
2^4	16%	1mm

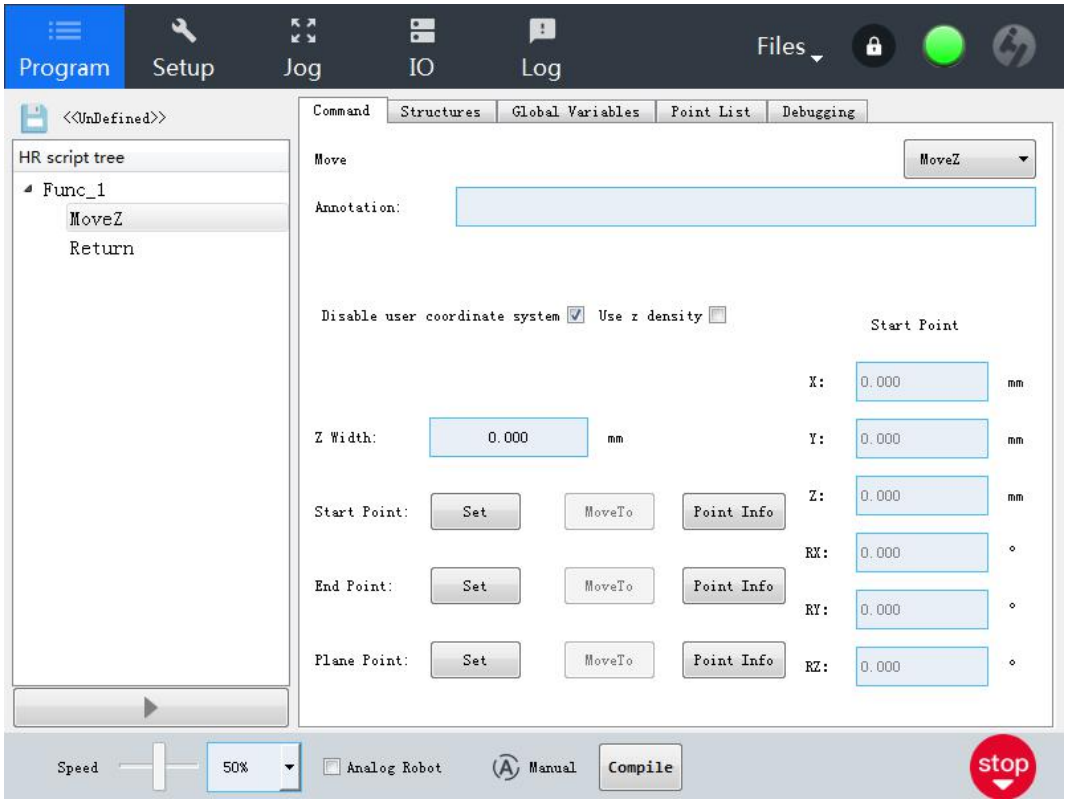


There are two ways to edit the MoveZ command: Whether to select "Disable User Coordinate System".

When "Disable User Coordinate System" is not selected, use the plane as the XY plane of the current coordinates. The Z-shaped trajectory in the current user coordinate system is realized by setting "Z-shaped width", "starting point" and "endpoint".

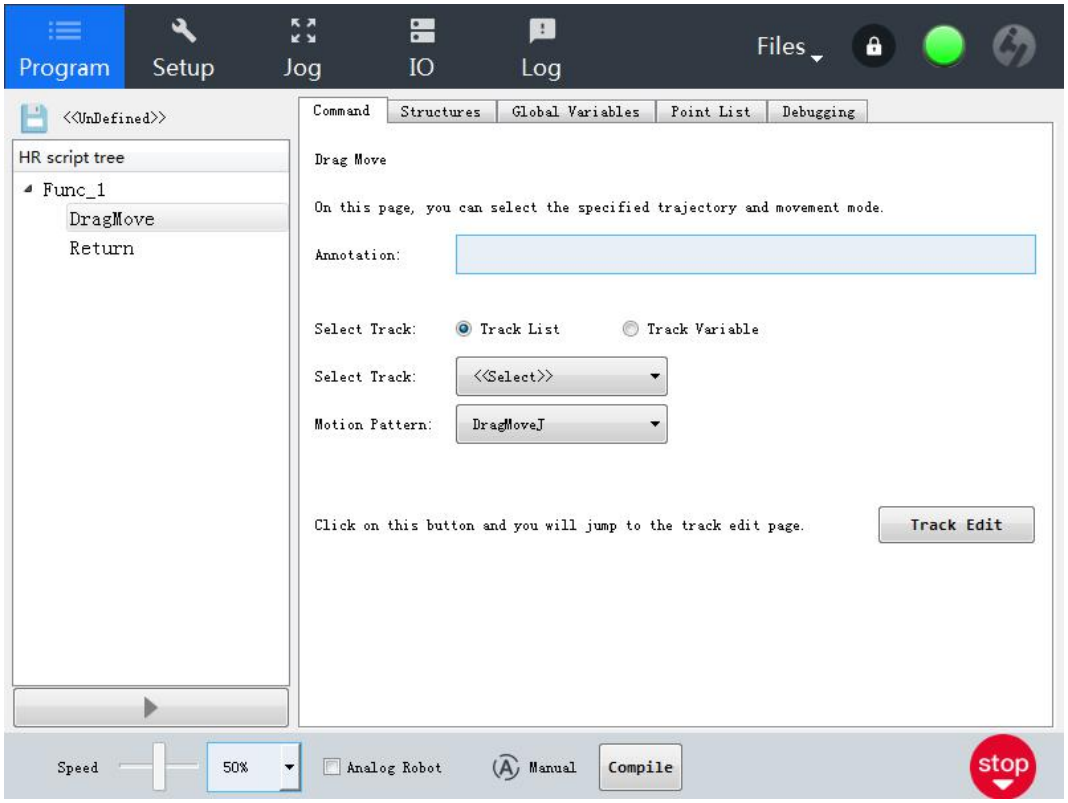


When "Disable User Coordinate System" is selected， will add 3 points to determine the place: Start Point, End Point, and Plane Point.



➔ DragMove

After the Freedrive function is turned on, the user drags the robot to record the motion trajectory at a certain sampling frequency. When the robot runs the command, the robot can repeat the trajectory of the user manually dragging;

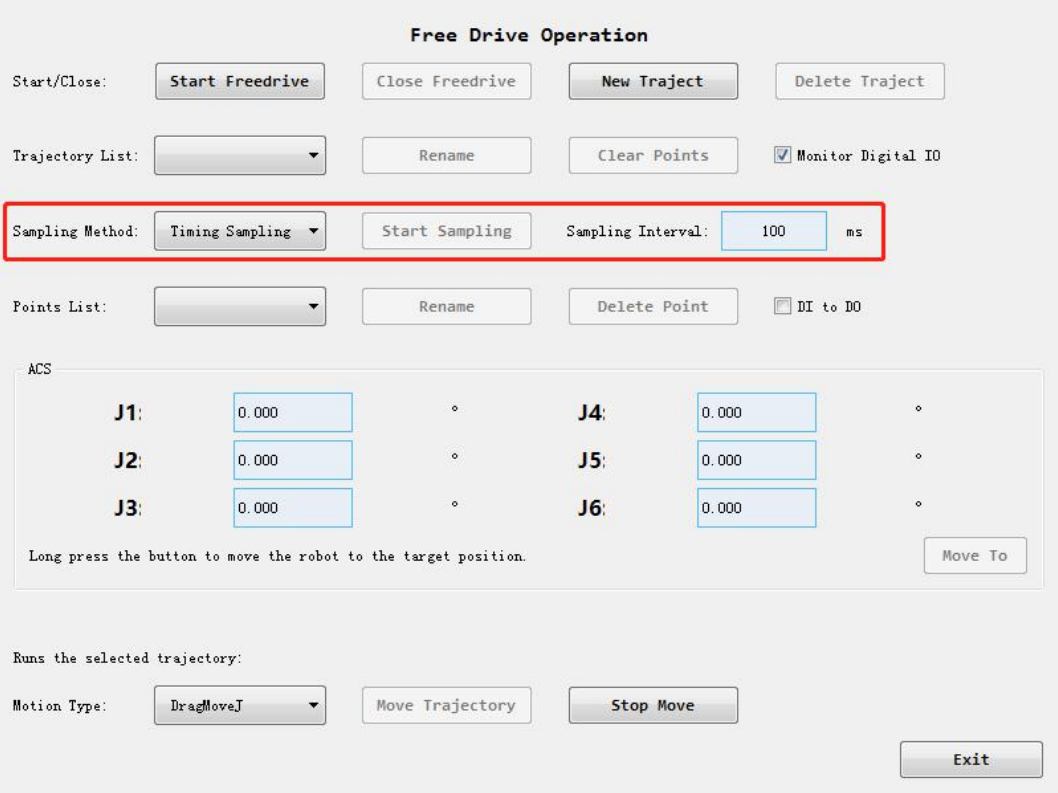


As shown in the figure above, “Select Track” has two modes: Track List, Track Variable.

1)Track List:You can select a taught track to run from the track list;

2)Track Variable:Create a new string variable string,set the string variable name to the name of the track to be run; when the instruction calls the variable, run the track.

Click the “Track Edit” button to enter the Free Drive Operation interface, as shown below;



(1)New Tracject:Click this button to create a new track for storing the points of the DragMove. You can query the newly created track in the Track List.

(2)Two Sampling Method:

①Timing Sampling:Select the Timing Sampling, input the sampling interval time (in ms) in the sampling interval edit field,click “Start Freedrive” button, open the sampling timer, that is, the controller will collect the user drag track at the interval set by the user.

②Manual Sampling:Select Manual Sampling, click “Start Freedrive” button, drag the robot to the target point and click the “Current Point Sampling” button to record the point. This method can continuously collect multiple target points.

(3)Points List:The points obtained by dragging are displayed in this table.

(4)Motion Type: Select DragMoveJ to execute points in the Dragmove command

(5)Delete point:After selecting the point in the sampling points list, click “Delete Point” to delete the corresponding points.

(6)Clear points:click ” Clear Points” to delete all points in the sampling points list.

(7)ACS,PCS display bar:ACS, PCS data value used to display the selected point.

(8)Move To:After selecting the point in the Points List, click the “Move To” button to move to the point in ACS or PCS mode.

(9)Monitor Digital IO: after clicking the “Start Freedrive”, the robot will not only

record the position of the drag track but also record the change of the IO signal during the drag and drag process and the waiting time of the drag process.

(10)DI to DO:Reflect the effect of the DI signal to the associated DO signal.

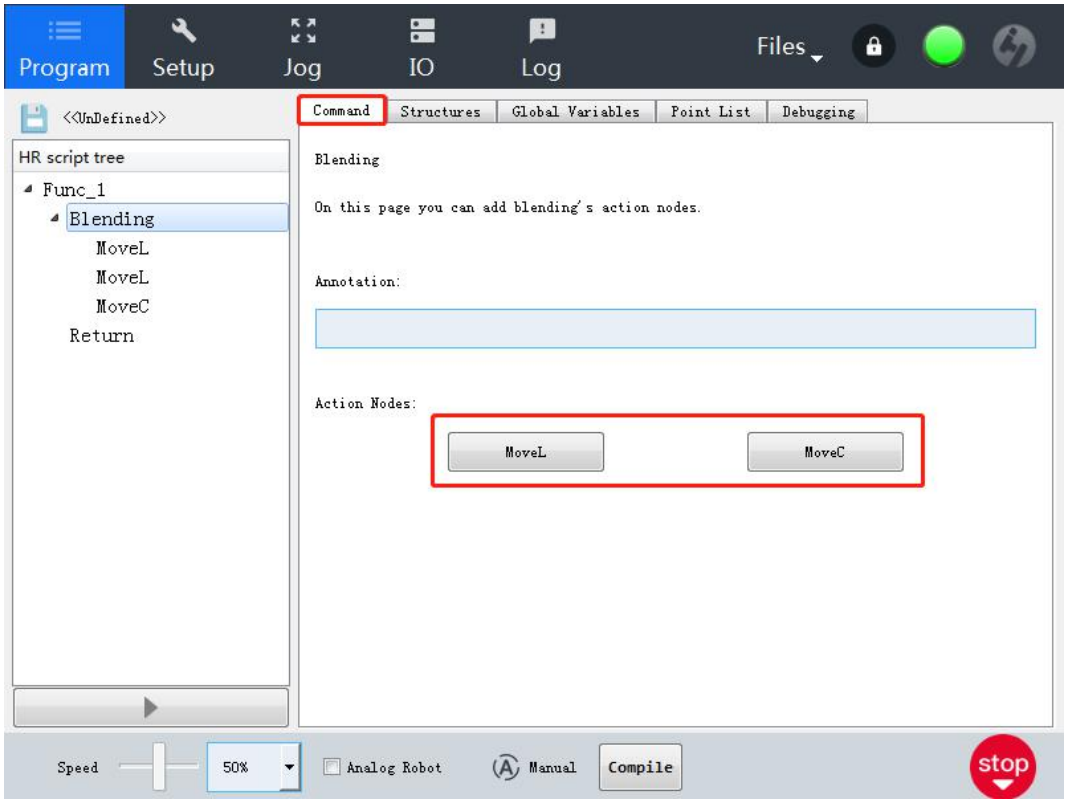
DragMove Operation Steps：

- Add the DragMove command in the script tree, click the "Track Edit" button to enter the command detail interface for editing;
- Click the "New Traject" button to create a new empty traject;
- Select the sampling method in the instruction details, such as "Timing Sampling", set the Sampling Interval (such as 500ms);
- Click ” Start Freedrive” and “Start Sampling” button,then use hands to drag to a defined trajectory;
- After finishing the dragging,Click” Stop Sampling” and “close Freedrive” to finish the job;
- Click the “Move Trajectory” button, the robot will reproduce the track traced by the user.

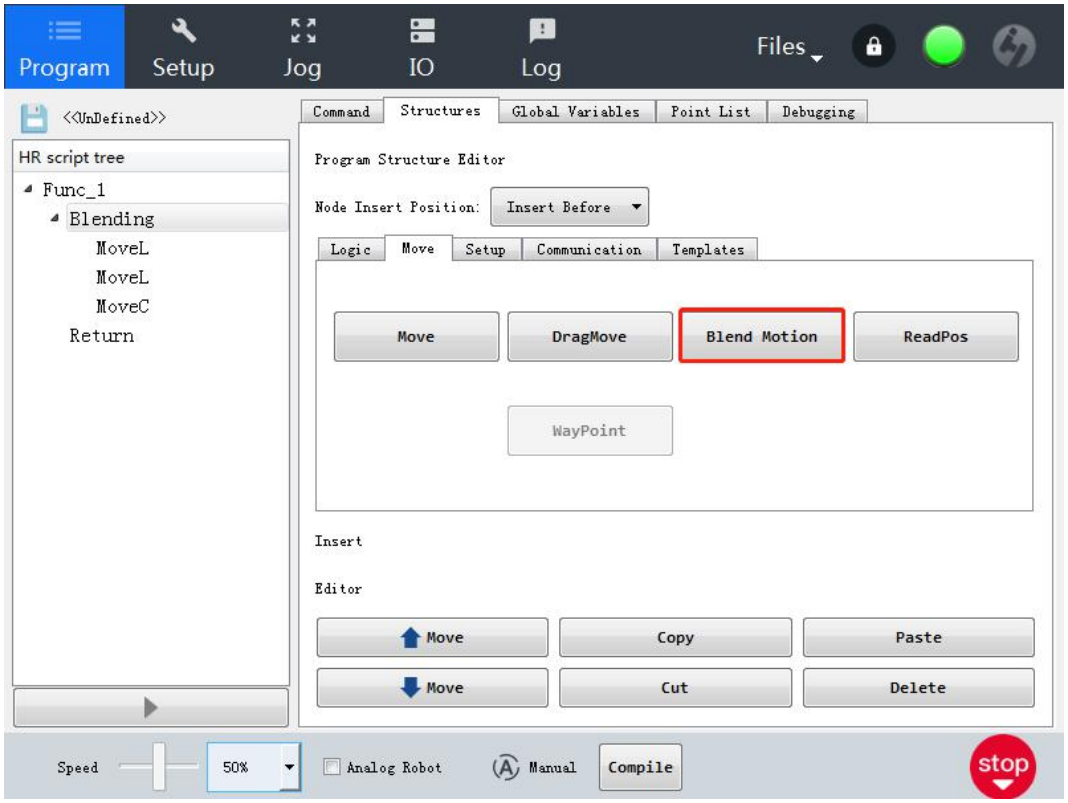
➔ Blend Motion

The adjacent two straight lines make a circular arc smooth transition with the set blending radius. The running behavior of Blending is continuous motion and will not stop at the waypoint. The smaller the blending radius value, the larger the corner of the path. Conversely, the larger the blending radius value, the smaller the corner of the path.

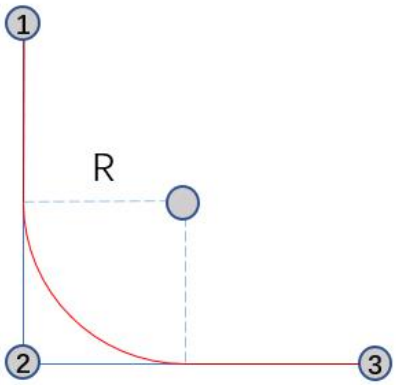
Unlike other ways of adding instructions, the instructions in the Blending structure are added in the command interface of Blending.



MoveL Command of Blending Structures:

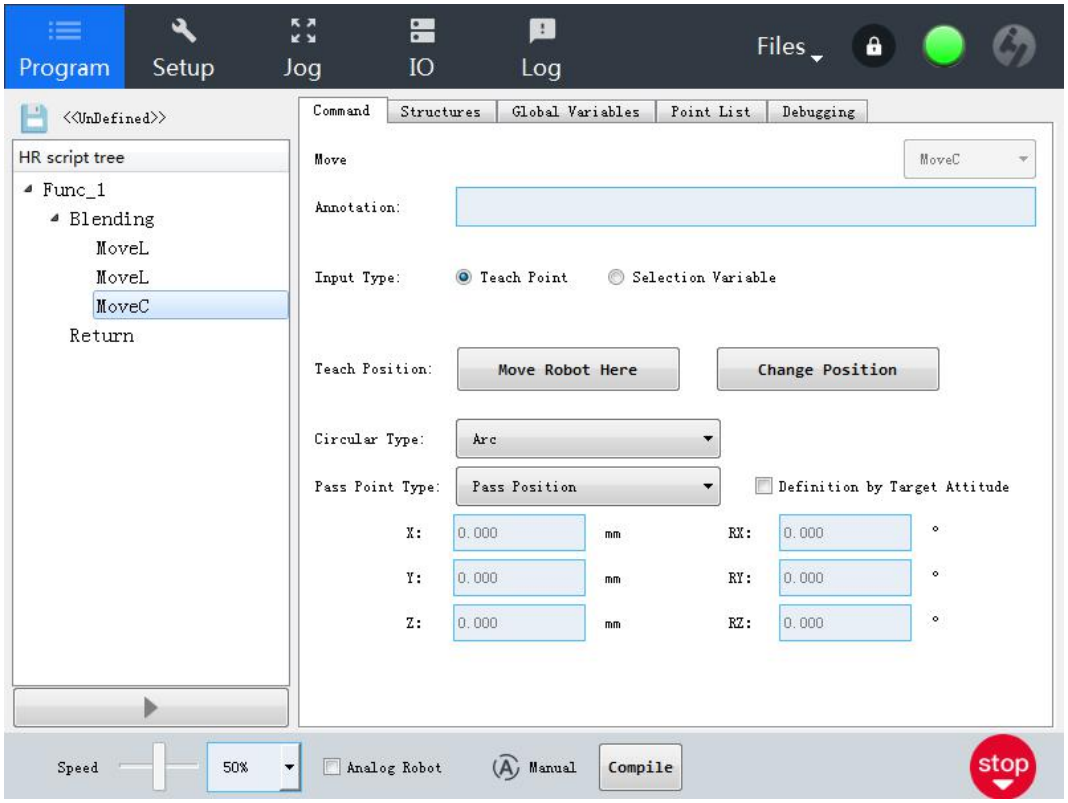


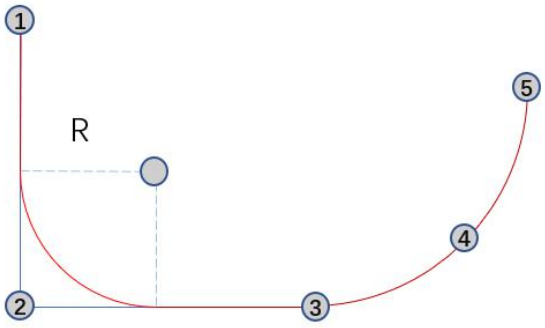
R radius: the blending radius performs a smooth arc transition. The set radius cannot exceed the straight line distance, otherwise, the calculation is incorrect. If two lines can be connected as a straight line, the set radius is invalid.



Point 1 is the current position of the robot, point 2 is the target point of the first MoveL command, and point 3 is the target point of the second MoveL instruction. The red trajectory is the calculated trajectory.

MoveC Command of Blending Command:





The red trajectory is the calculated running trajectory, the two straight lines are smoothly smoothed by the arc with the set blending radius, and MoveL+MoveC makes the straight line and the circular arc continuously move.

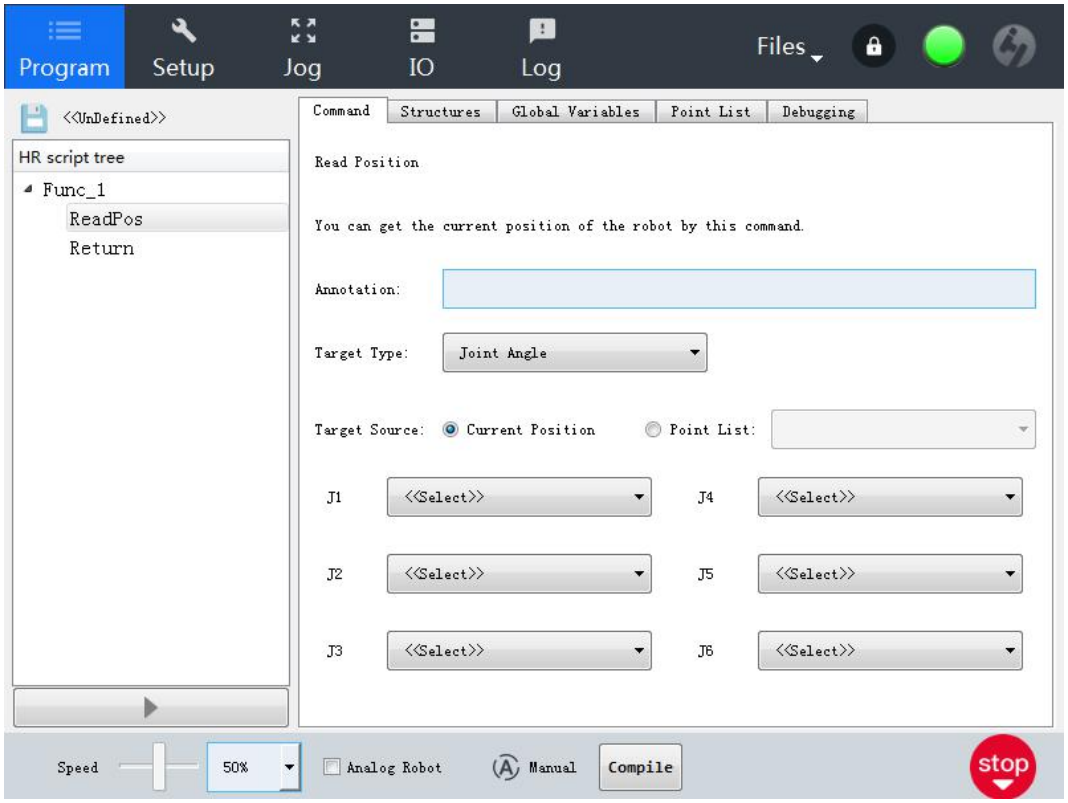
Point 1 is the current position of the robot, point 2 is the target point of the first MoveL command, and point 3 is the target point of the second MoveL command and also the starting point of the arc. Point 4 is the arc passing position, and point 5 is the target position of the arc. Specific parameters can be found in the MoveC section.

➔ ReadPos

Get the robot's position, current encoder value, and system time.

Get robot position:

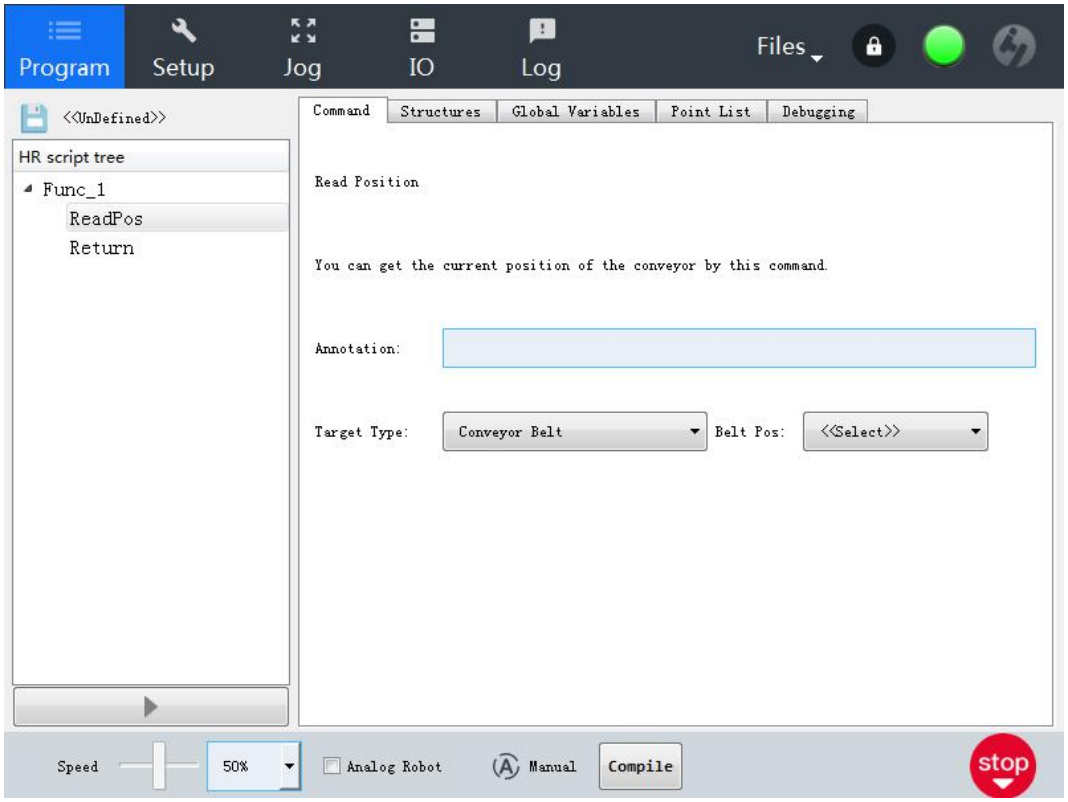
You can choose to get the joint angle value or space coordinate value and assign these values to the Double variable.



Target Source：Current Postion：Get the coordinates of the current robot，Point List：Get the coordinate value of the selected point.

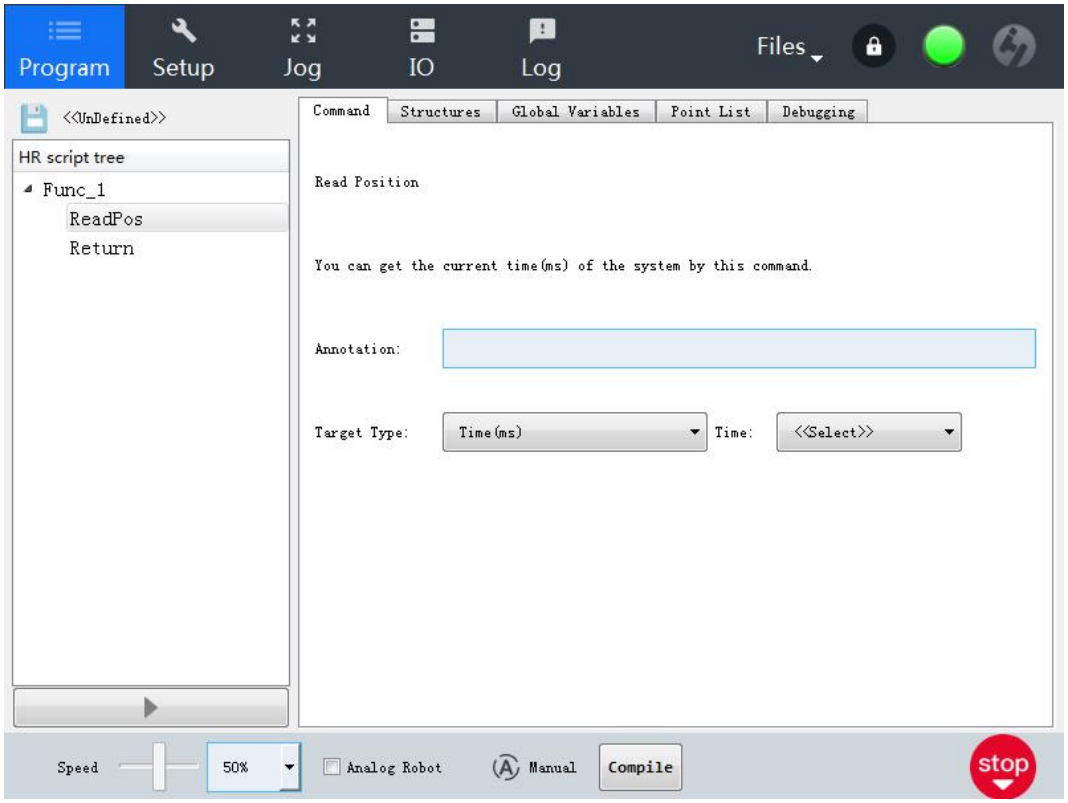
Get the encoder position：

Get the encoder values and assign them to the Double variable. Generally used for conveyor belts, the acquisition value is related to the conveyor belt calibration value, the unit is mm.



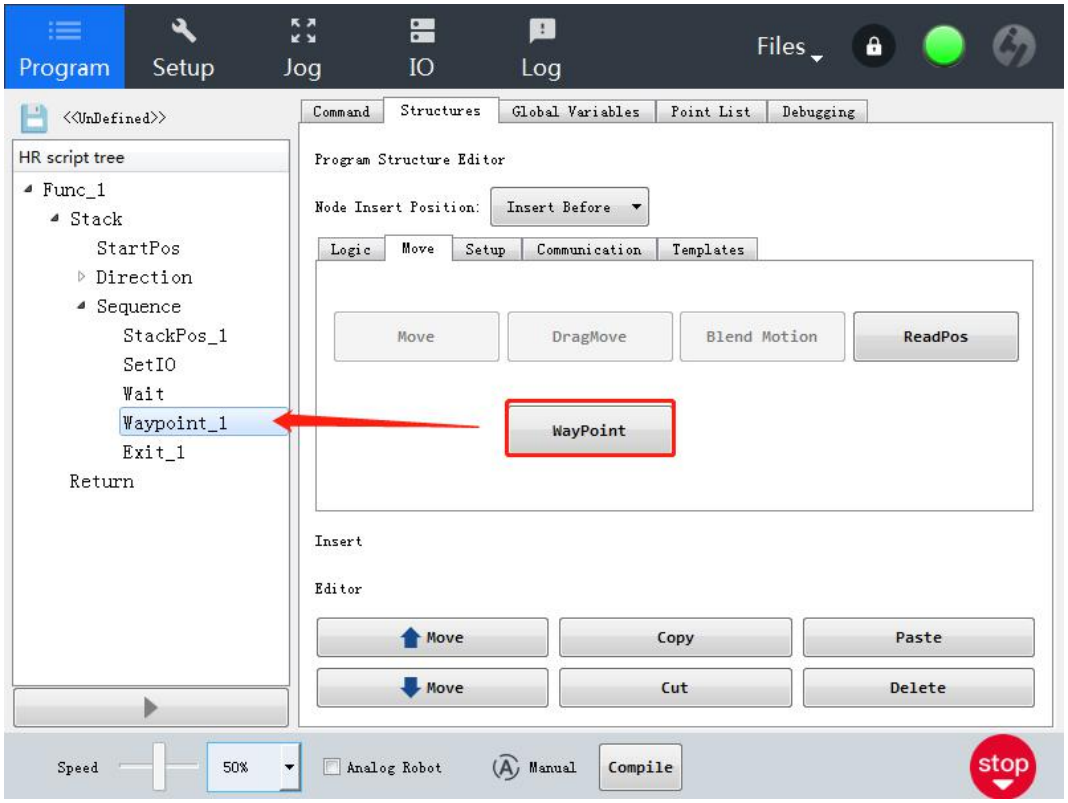
Get system time：

Get the system time and assign these values to the Int variable. Unit: ms.



➔ WayPoint

Waypoint can set the speed at the same time, can only be used in the “seek” function.



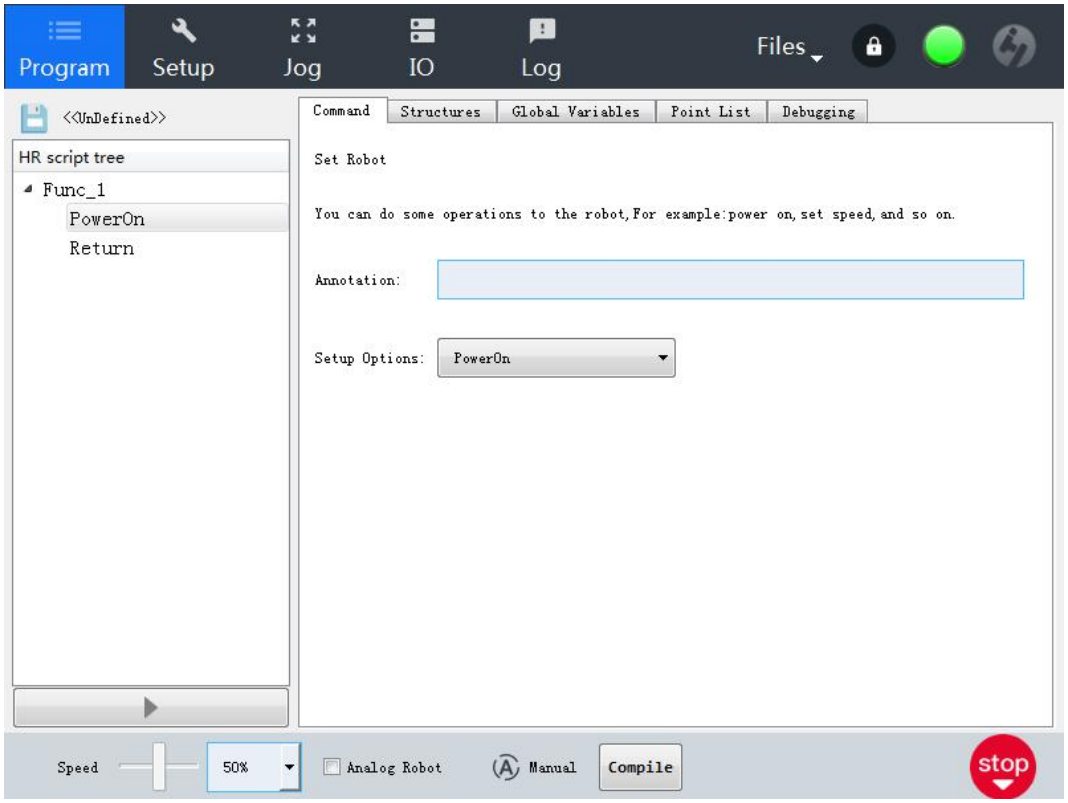
Setup

This part is to set the robot parameters and IO related functions. You need to add this type of instructions to insert instructions from this page.

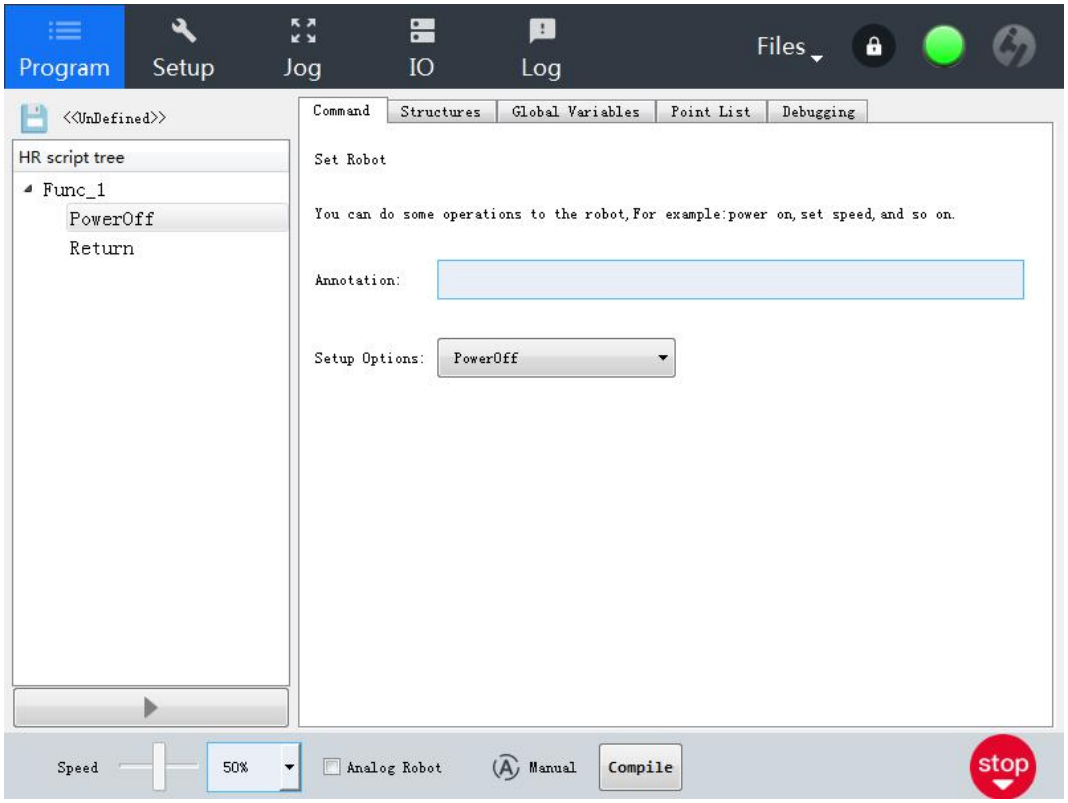
➔ Set Robot

This command can control the robot's PowerOn, PowerOff, Reset and Stop; set the robot's user coordinates, tool coordinates; set the speed ratio during the movement; set the robot payload; set conveyor belt.

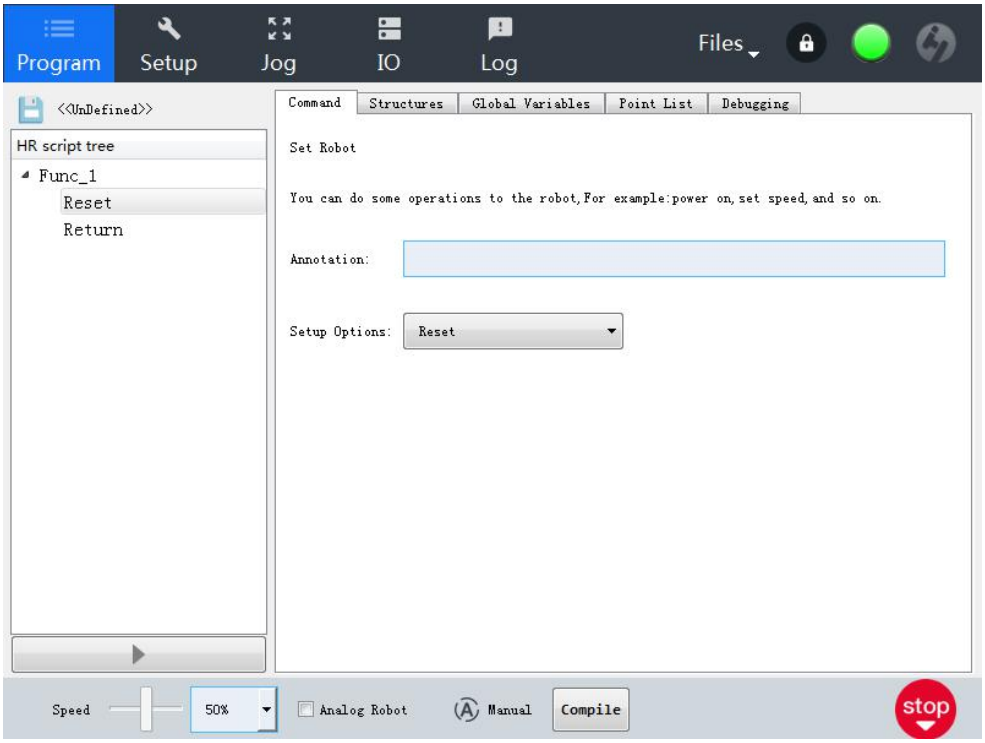
PowerOn



PowerOff

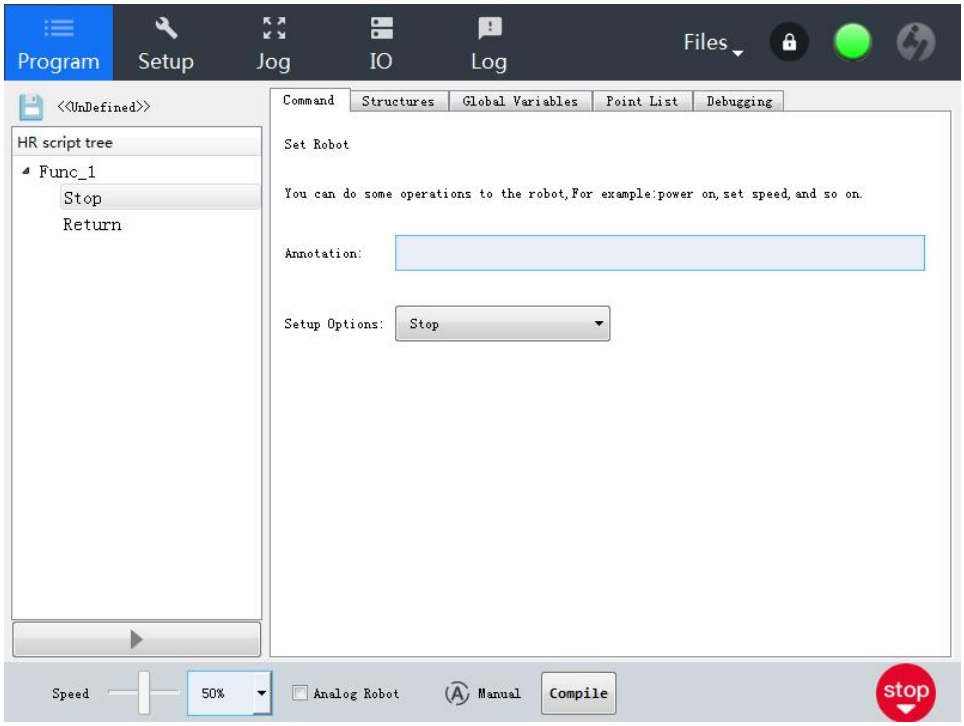

Program
Setup
Jog
IO
Log
Files
<<UnDefined>>
HR script tree
Func_1
PowerOff
Return
Command
Structures
Global Variables
Point List
Debugging
Set Robot
You can do some operations to the robot, For example:power on, set speed, and so on.
Annotation:
Setup Options:
PowerOff
Speed
50%
Analog Robot
Manual
Compile
stop

Reset



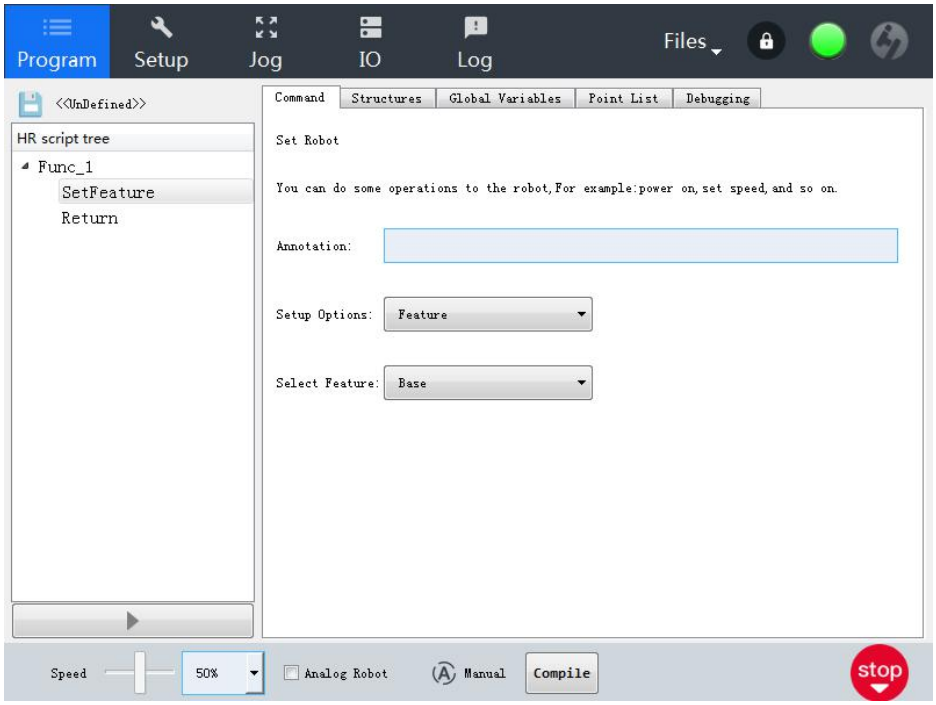
Stop

The script commands are executed in order, so if you want to stop during motion, it is recommended to execute the stop command by conditional judgment in the timer.



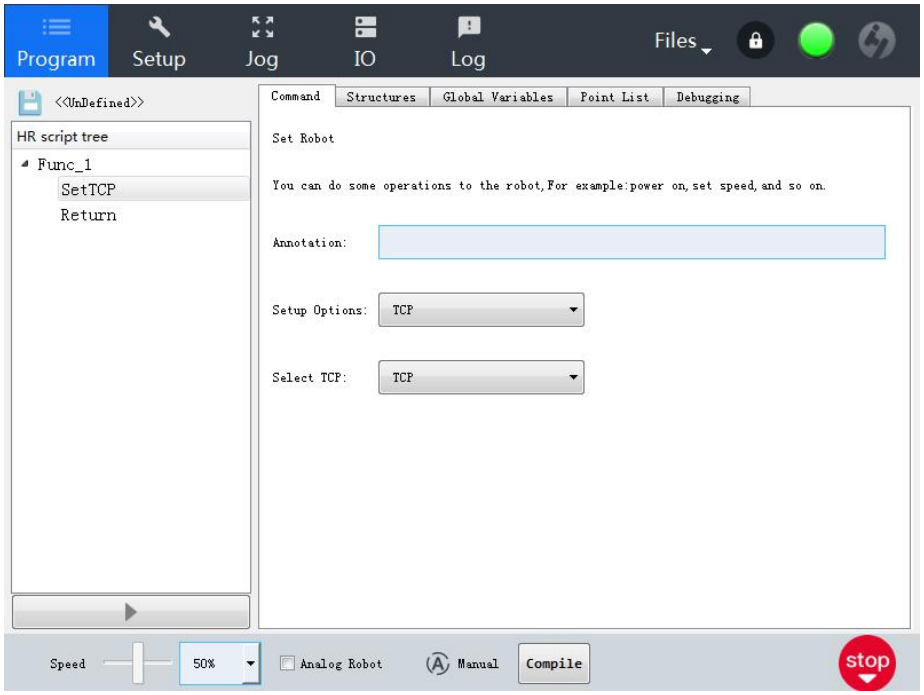
SetFeature

The user coordinates of the feature list can be configured in the feature interface of the parameter setting menu.



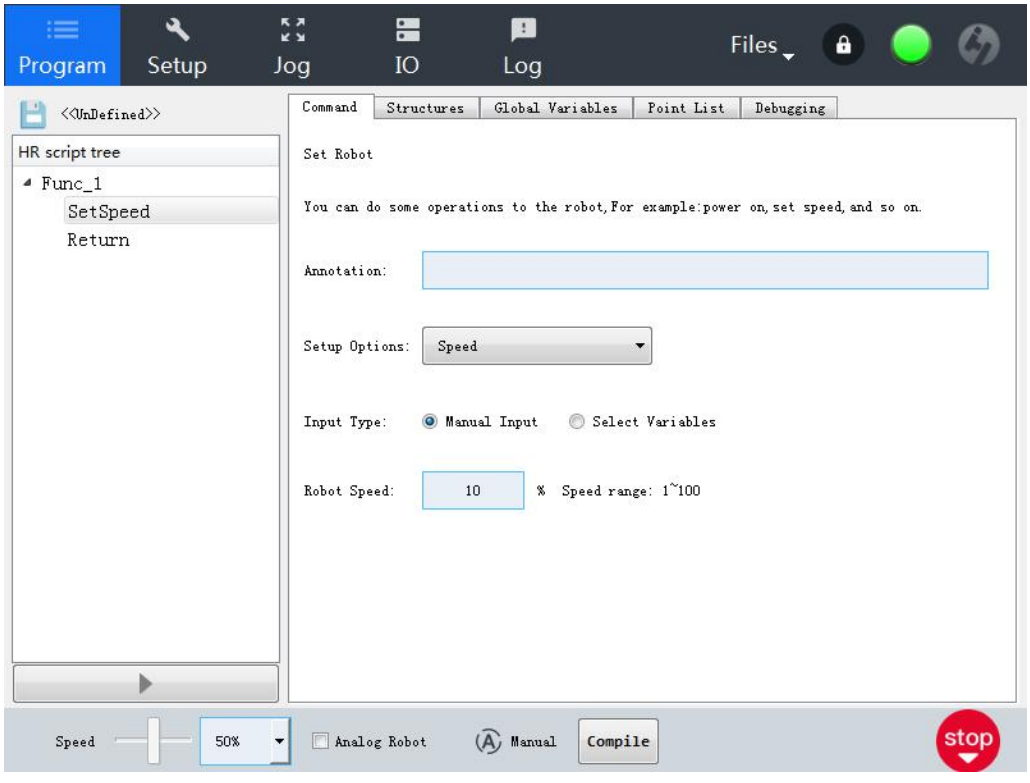
Set TCP

The tool coordinates of the TCP list can be configured in the TCP configuration interface of the parameter setting menu.



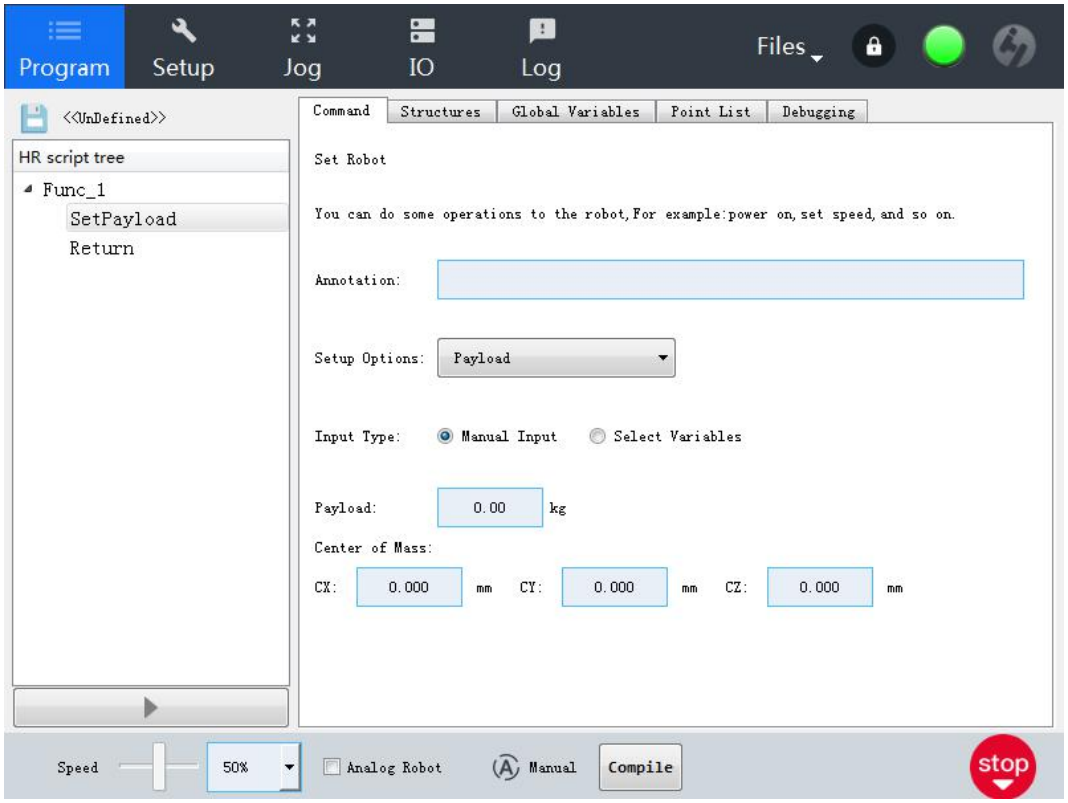
SetSpeed

Set the speed ratio of the robot movement, you can choose the way to customize the value or variable. If the set rate ratio is not the bottom frame speed list value, the speed is set successfully but no display.



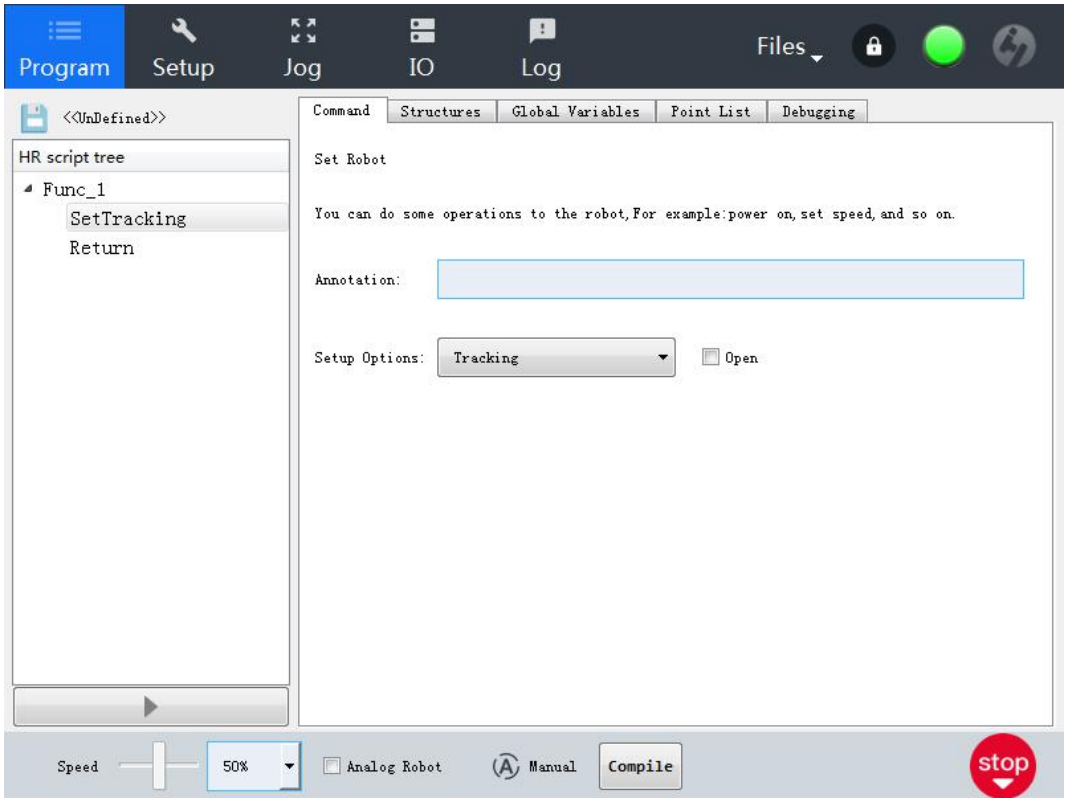
SetPayload

Set the rate ratio of the robot movement, you can choose Manual Input or Select Variables, and it is generally used for the capture process.



Open/Close Tracking

Open/Close Tracking function. Select" Open" means Open Tracking,(It is recommended to use the Open Tracking function in the MoveL command to optimize the use)Uneselect means Close Tracking.

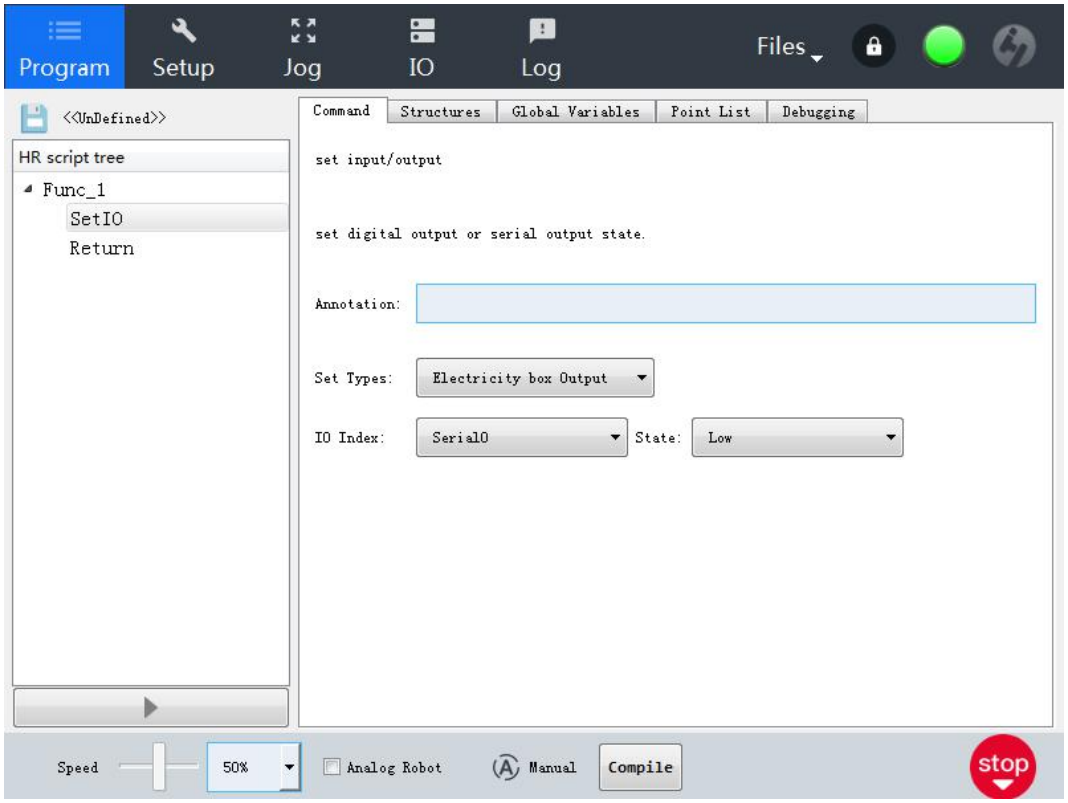


➔ SetIO

Set the End Output IO, the Electricity box output IO status and the Electricity box analog output.

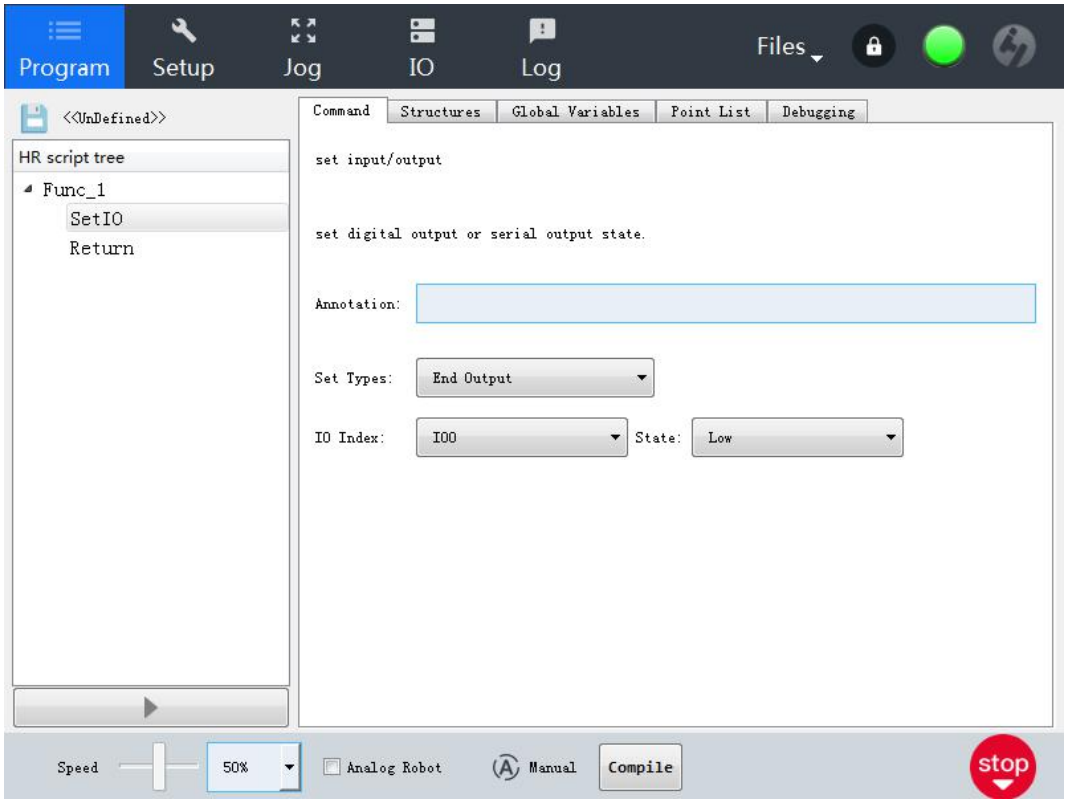
Electricity box Output

Set the digital output IO status of the electricity box serial port to set the corresponding IO signal high or low. Serial0~ Serial7 corresponds to DO0~DO7 of the electricity box interface of the robot IO menu, and Serial8~ Serial15 corresponds to SO0~SO7 of the robot IO interface.



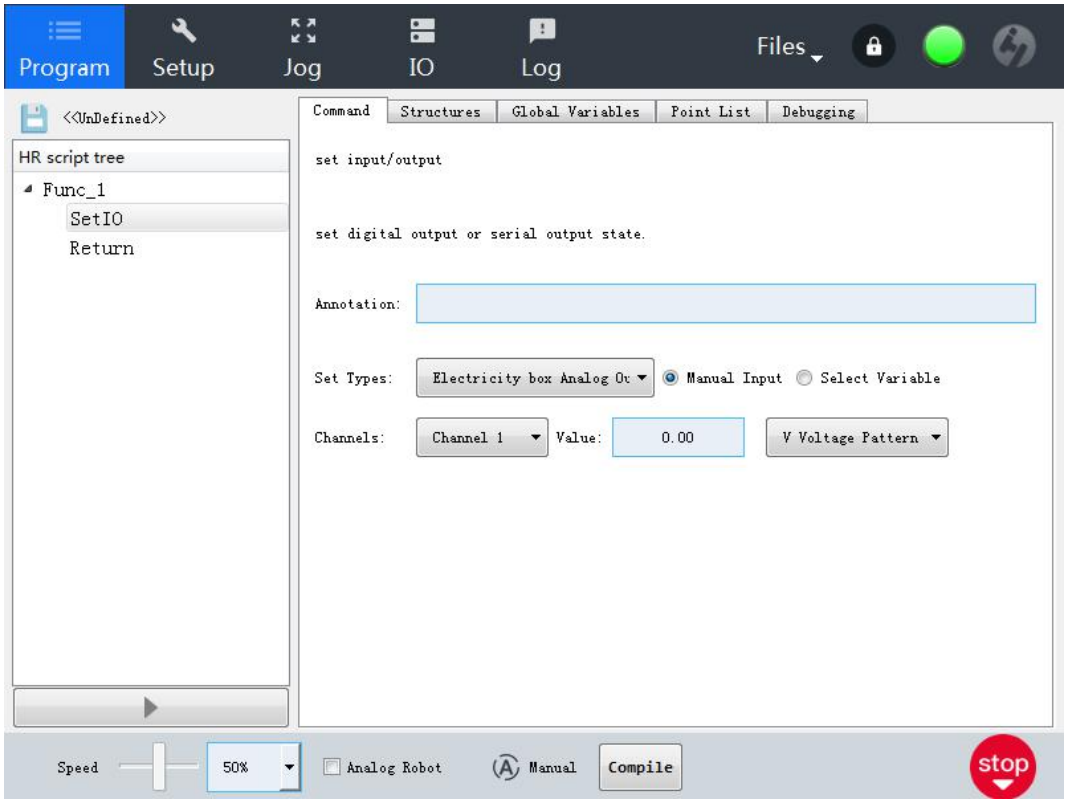
End Output

Set the end serial port digital output IO status to set the corresponding IO signal high or low. IO corresponding to the end interface of the robot IO menu.



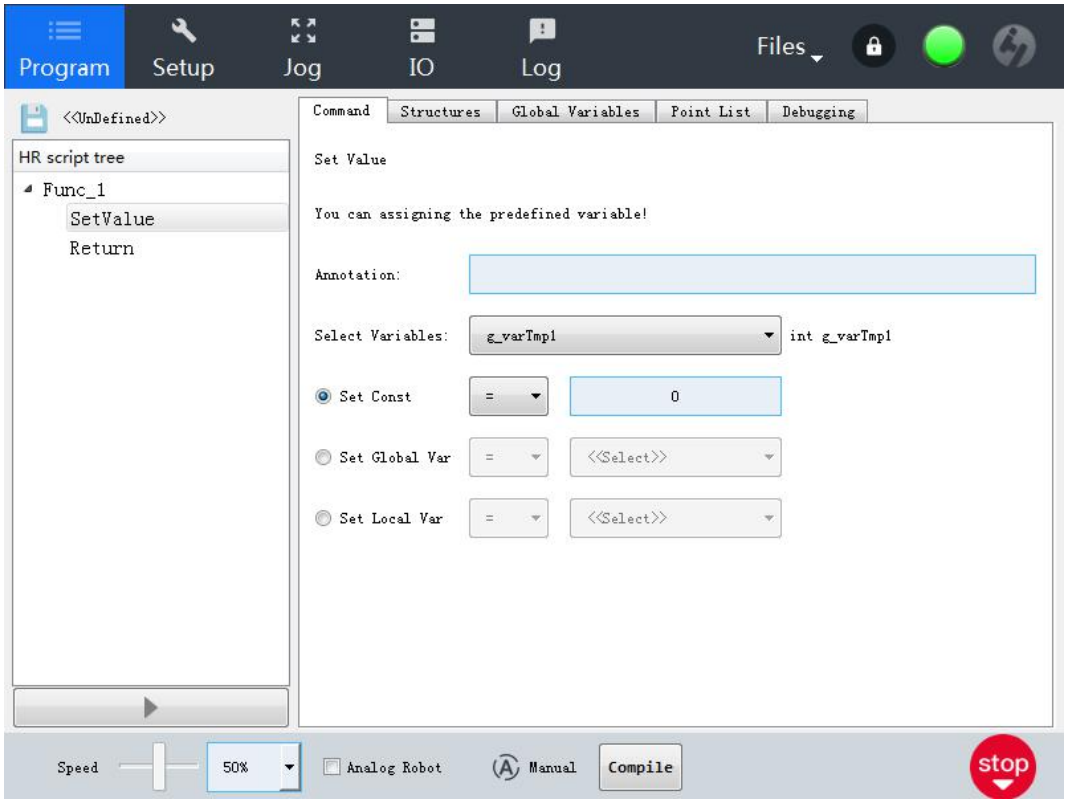
Electricity box Analog Output

You can select the current or voltage mode by setting the analog output value of the electrical control box by selecting a custom value or variable.



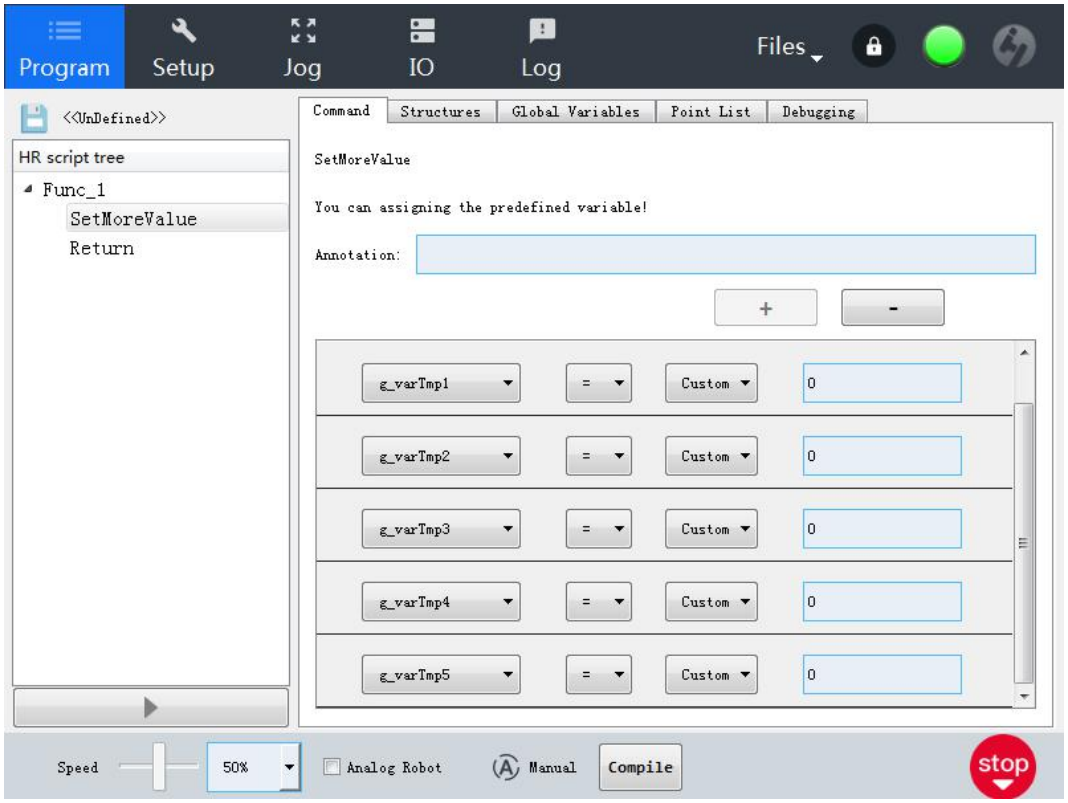
➔ SetVar

Commands for assigning global variables and local variables can be added, subtracted, multiplied, and divided.



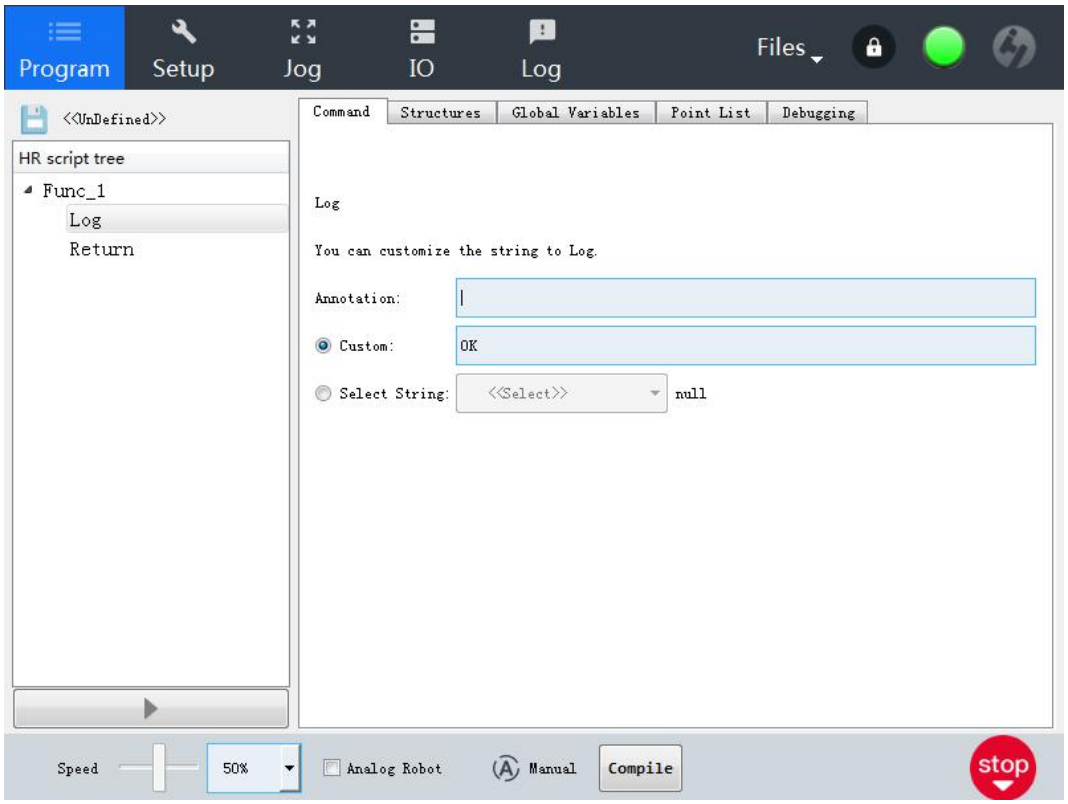
➔ SetValues

A command that assigns multiple global variables can perform addition, subtraction, multiplication, and division. Up to 5 operations can be added.



➔ Log

Print content to the diary for anomaly positioning problems.

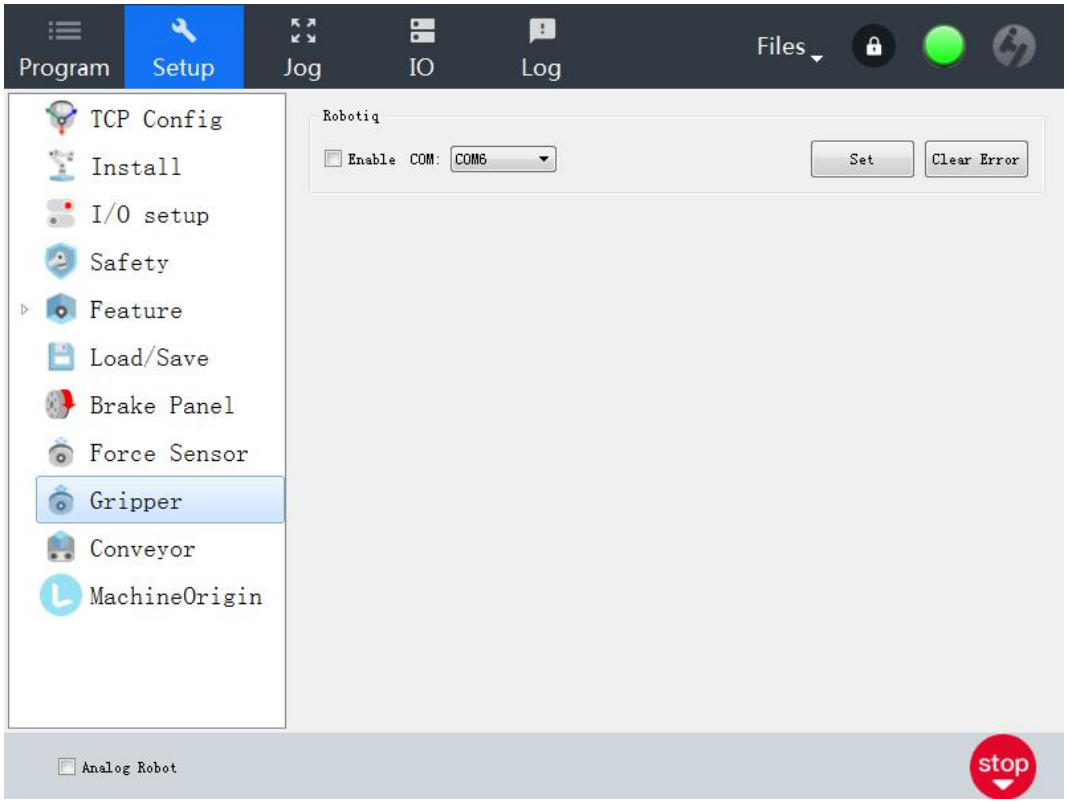


SetRobotiq Command

The robot end supports the installation of the Robotiq grippers. There are two communication methods: the com port of the electrical box, the operation steps include the following steps 1, 2; KPA485 communication, the operation steps only need step 2.

The steps are as follows:

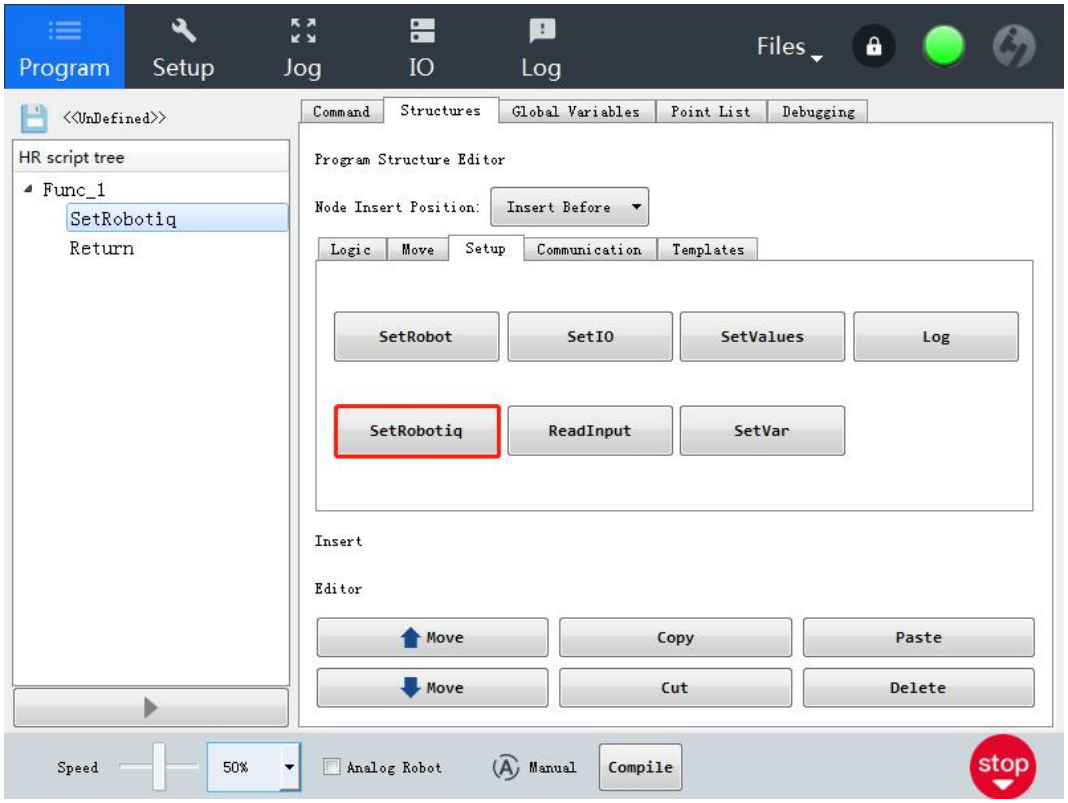
1)Change to “Gripper” interface， select “Enable” ,Com's choice depends on the interface of the actual gripper connection.



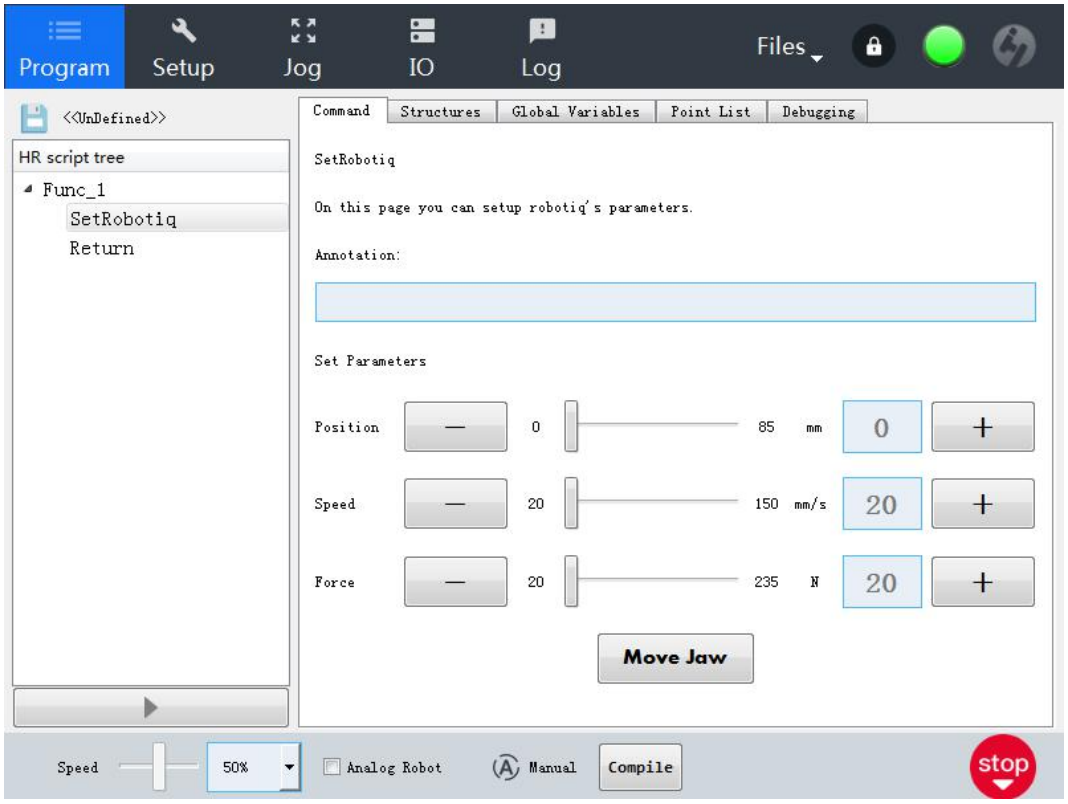
2)Command application

(1)Add the SetRobotiq command to the script tree:

It is generally necessary to add two SetRobotiq commands, corresponding to the state whether the gripper is open or closed.



(2)Edit SetRobotiq Command:



In the command interface, the meaning of each button is as follows:

Annotation:Add a note to the instruction.

Set the parameters (take the Robotiq gripper with the 5KG load as an example) as follows;

The position:the range is 0~85mm. When the gripper is closed, its position value is 0 mm. When the gripper is fully opened, the position value is 85mm.

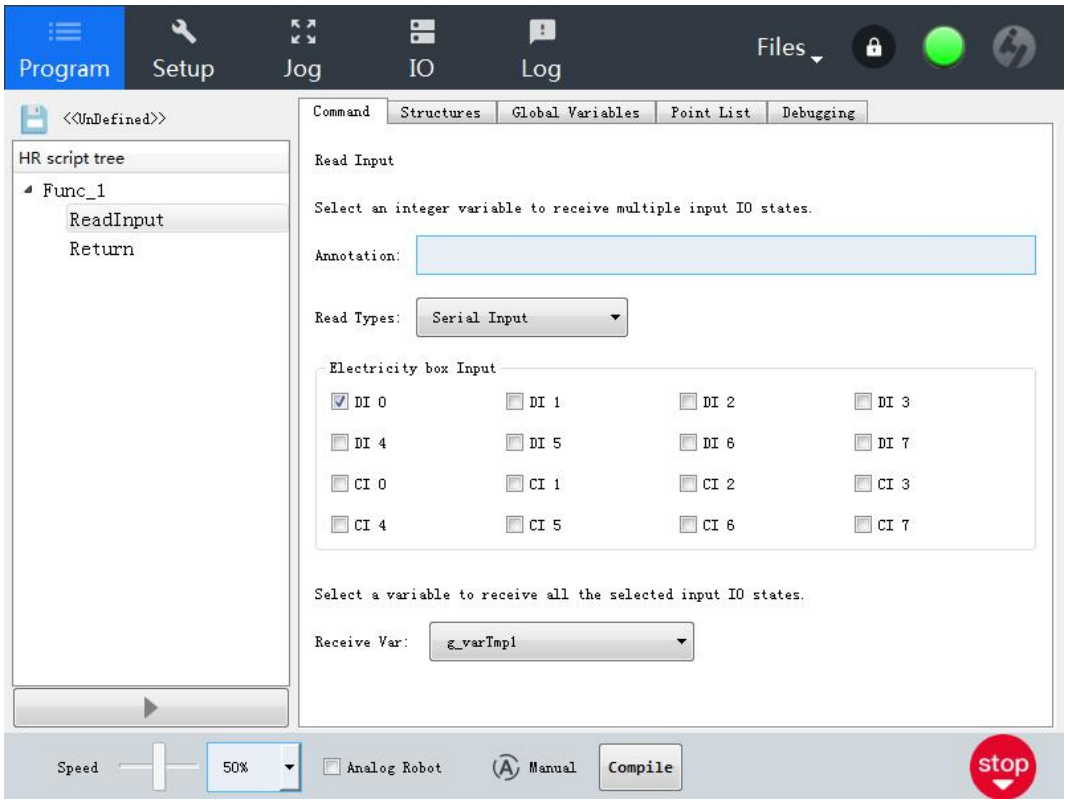
Speed:The range is 20~150mm/s, and the movement speed of the gripper state change (open to closed, closed to open) is set.

Force:Range 20~235N set the amount of holding force when the gripper is closed.

Move Jaw:To start the gripper and check the movement under the setting.

➔ ReadInput

Reads the state of one or more input IOs, stores their state in an int integer variable, and reads the electrical control box or the end analog input.

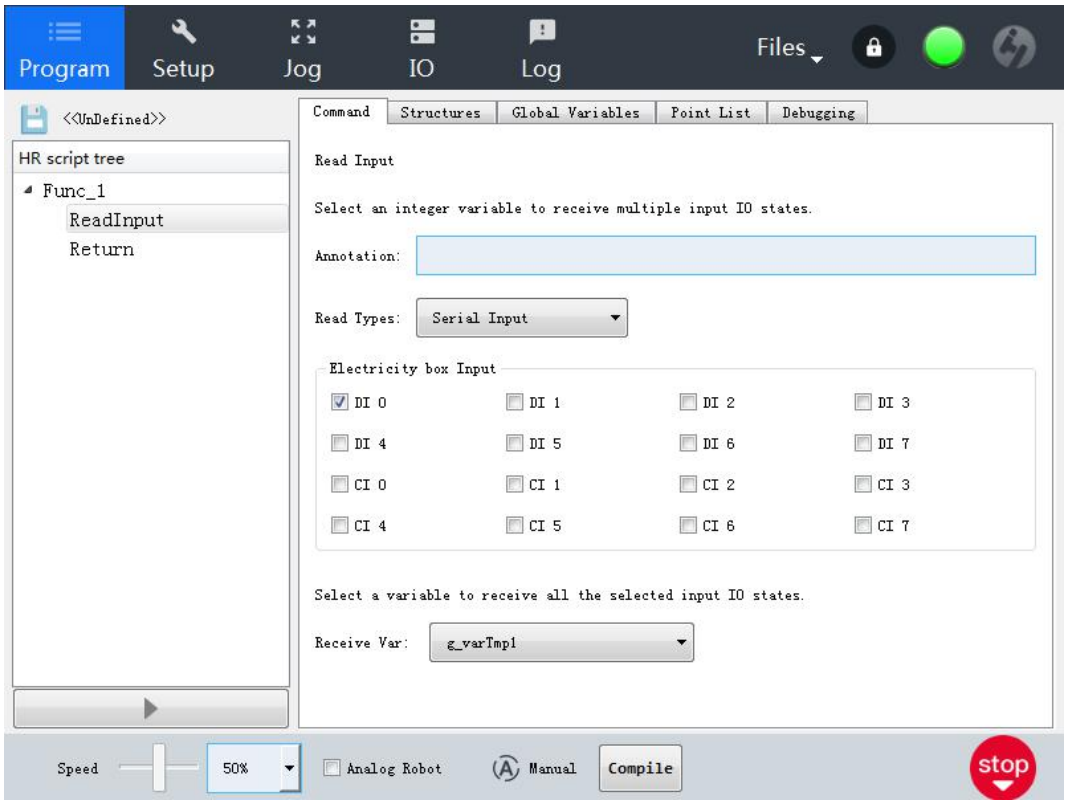


Serial Input

The serial port IO represents the electric control box IO, the digital IO represents the end IO, the serial port IO DI0~DI7 corresponds to the DI0~DI7 of the robot IO menu electronic control box interface, CI0~CI7 corresponds to the robot IO interface SI0~SI7; the digital IO end0 ~End3 corresponds to IO EndDI0~EndDI3 of the end interface of the robot IO menu.

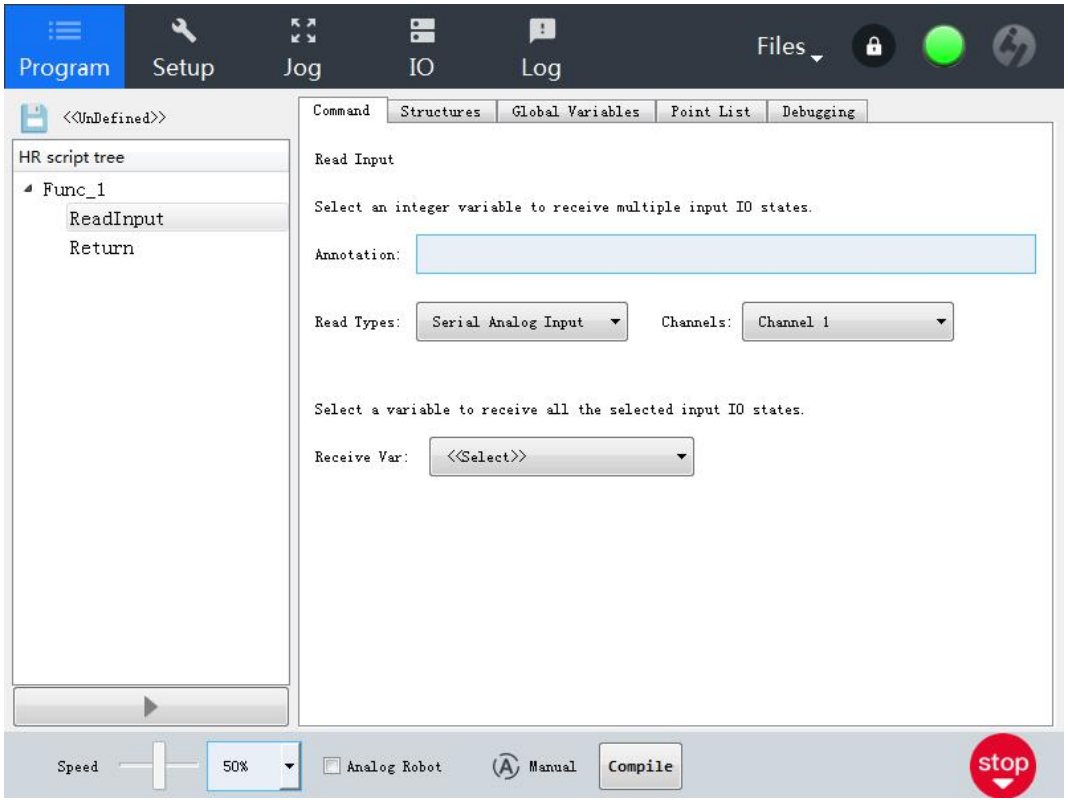
The value of the int variable is determined by:

If S0, S3, S6 are selected,the IO signal state is written into the int variable. When the signals of S0,S3, and S6 are both set high, that is, int=2^0 +2^1 +2^2 =7;if the signals of S0 and S6 are set high, S6 When the signal is de-asserted, int=2^0 +2^2 =5;



Serial Analog Input

The read analog value is stored in the Double variable, the serial port analog input represents the electronic control box analog quantity, and the digital-analog input represents the end analog input.



Communication

Elfin Teach Pendant Communication Module supports the two most popular communication methods: network port (TCP protocol) and serial port (MODBUS TCP protocol). The communication module of the teach pendant software is used in script editing as follows.

➔ Socket TCP/IP

★TCP protocol

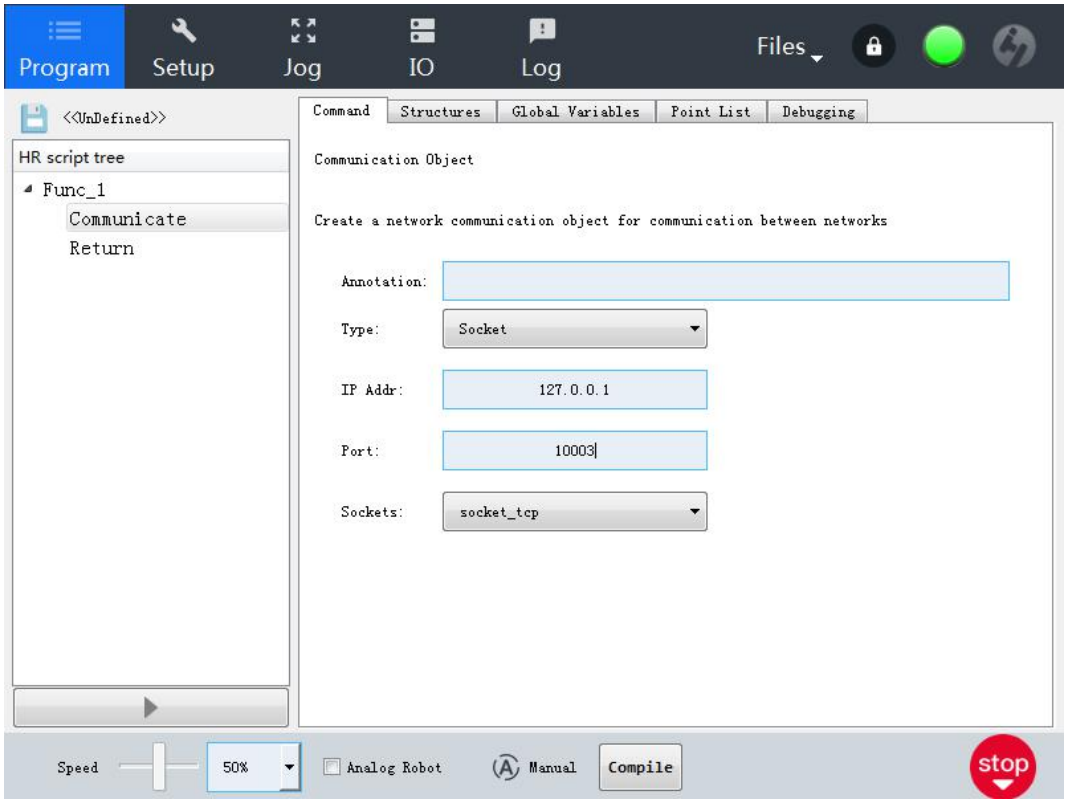
❶Variant

Added socket type variable socket_TCP (for new connection slot) and string type variable string_TCP_recieve (for receiving message string).

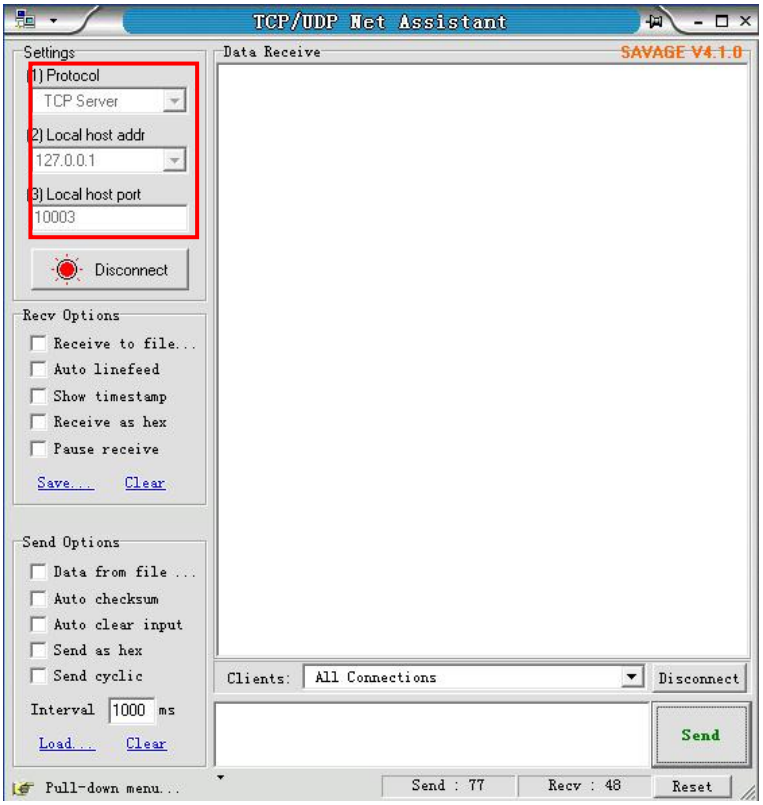
❷Establish a configuration connection

The following is an example of a communication assistant.

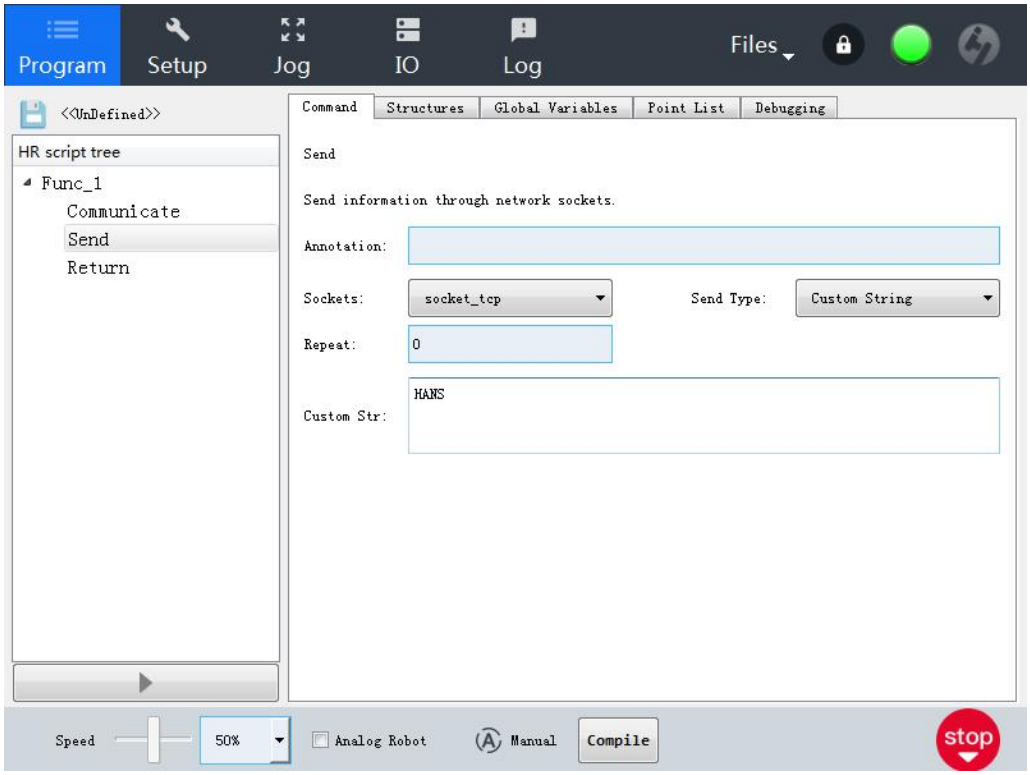
As a client(Teach Pendant), the TCP communication uses the Socket connection method, configures the IP (using the loopback address 127.0.0.1 or the remote host address) and Port, and the socket type created before the socket is used (socket_TCP), using the communication assistant as the server.



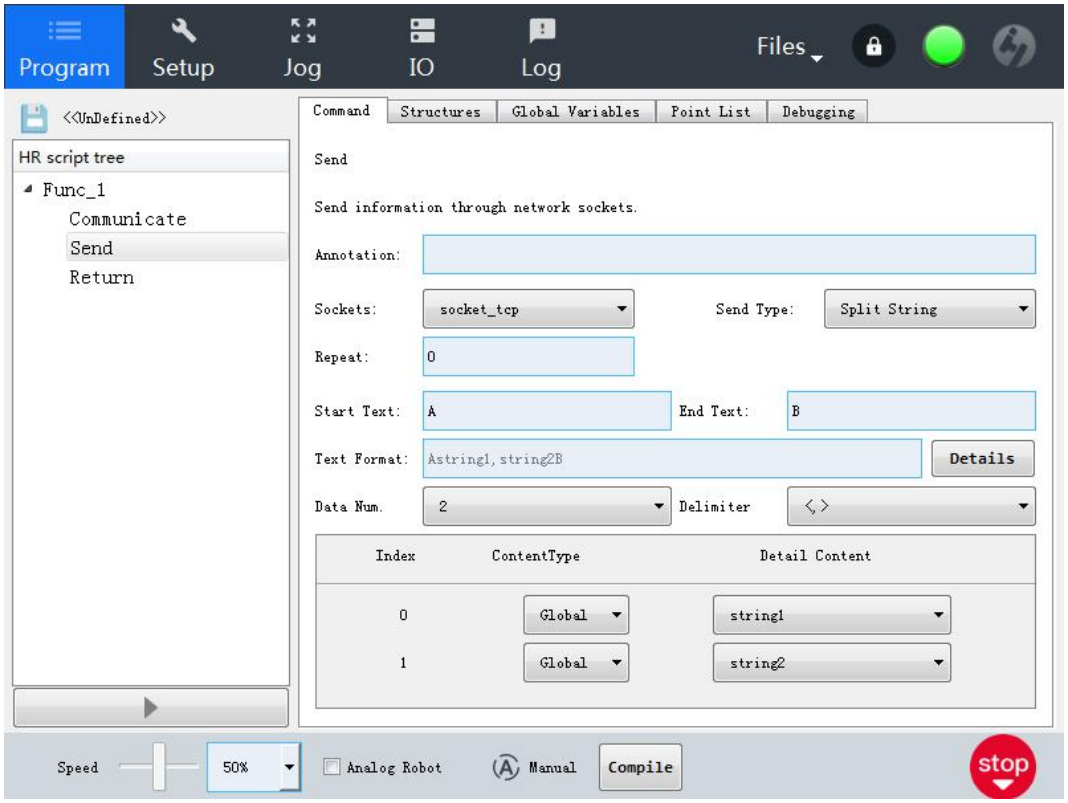
Note: If you use 127.0.0.1 as the loopback address, the port number must be 10003.



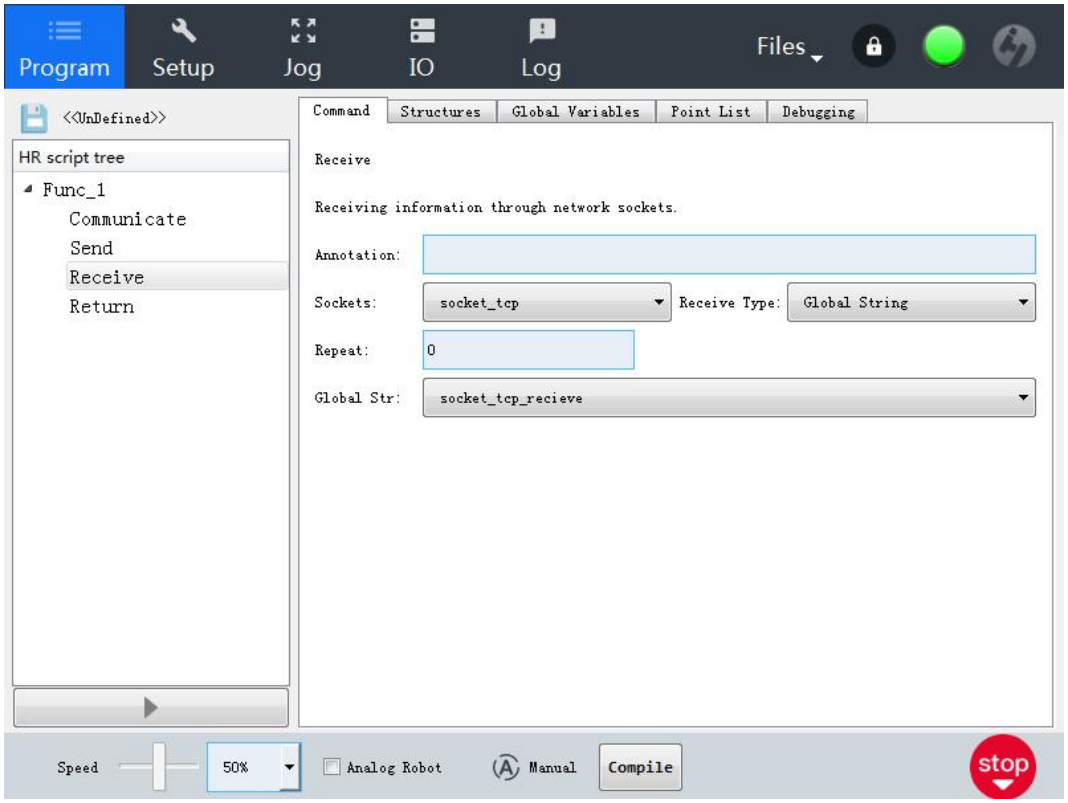
❸**Send Type:**As shown in the figure below, you can select the form of custom string sending,string variable sending,and multiple string variable sending; select "custom string" to send here, send the content as "HANS";



When "Split String" is selected for transmission, as shown in the following figure, "Start Text" and "End Text" are the opening and ending characters of the sent string, and whether the "Start Text" and "End Text" are required according to the actual application of the client. "Delimiter" is the separator character between the strings, “Data Num." is the number of strings sent, as shown in the following figure, the number is 2; the data format sent is Custom (Custom string) and Global (string variable) two forms.

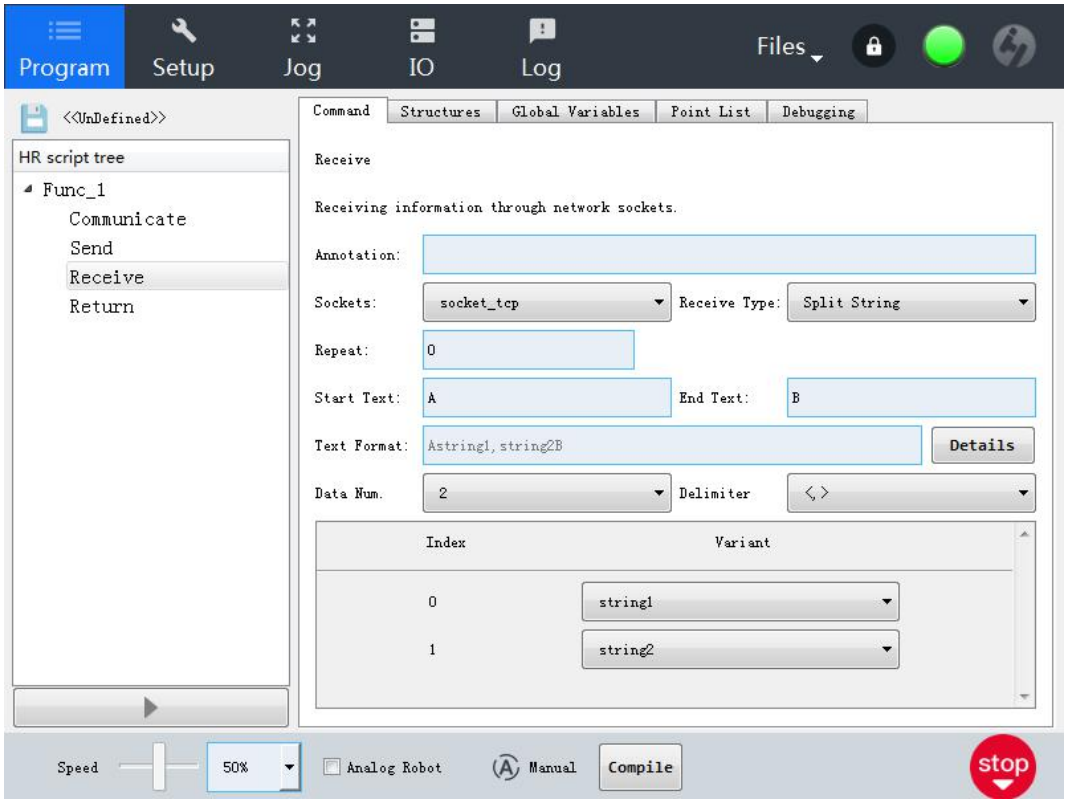

Program
Setup
Jog
IO
Log
Files
<<UnDefined>>
HR script tree
Func_1
Communicate
Send
Return
Command
Structures
Global Variables
Point List
Debugging
Send
Send information through network sockets.
Annotation:
Sockets:
socket_tcp
Send Type:
Split String
Repeat:
0
Start Text:
A
End Text:
B
Text Format:
Astring1, string2B
Details
Data Num.
2
Delimiter
Index
ContentType
Detail Content
0
Global
string1
1
Global
string2
Speed
50%
Analog Robot
Manual
Compile
stop

❹**Receive:**Accept the message in string format (string), here select the string variable (string_TCP_recieve) created before, as shown below:



When the “Split String" is selected for Receive Type, as shown in the figure below, the "Start Text" and "End Text" forms of the received data can be set. The "Delimiter" is the separator between the received strings and the number of data. For the number of received strings, the received variable types can be three types: Int, Double, and String.

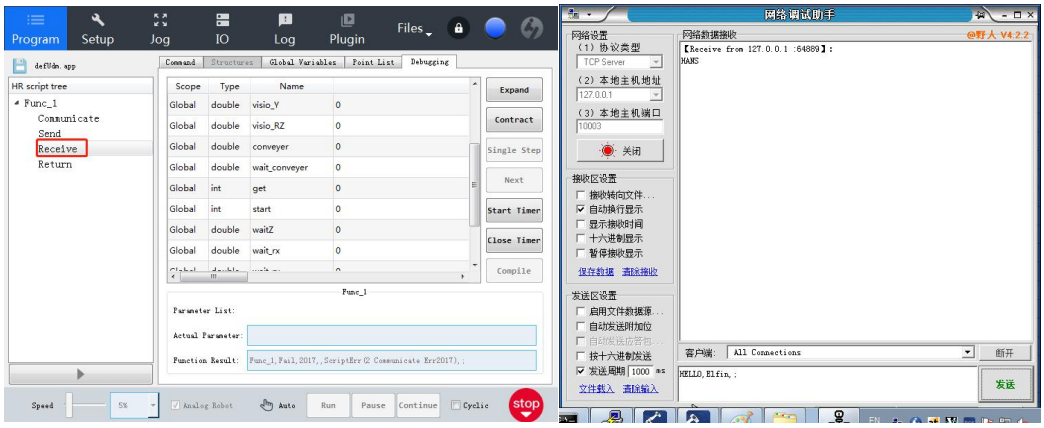
If the string format sent by the server is different from the set receiving format, the parameter will be reported incorrectly and the script will be finished.



❺**Return:**End function (Return), return a custom string to indicate that the function (Function) has been executed.

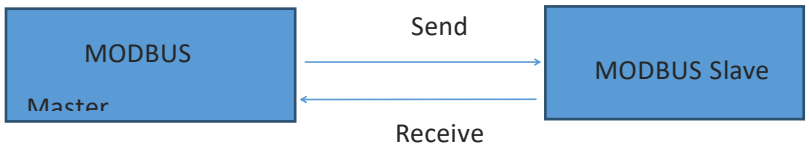
❻Run the function after the configuration is complete(F_TCP_communicate):

❼After the teach pendant sends (HANS) to the communication assistant, it waits to receive the message of the communication assistant; the teacher receives the message "HELLO, Elfin, " of the communication assistant, and the corresponding variable string_TCP_recieve is Change, the effect is as shown below:



➔ Modbus TCP

★MODBUS TCP Protocol



HansRobot Client　　Server (user equipment)

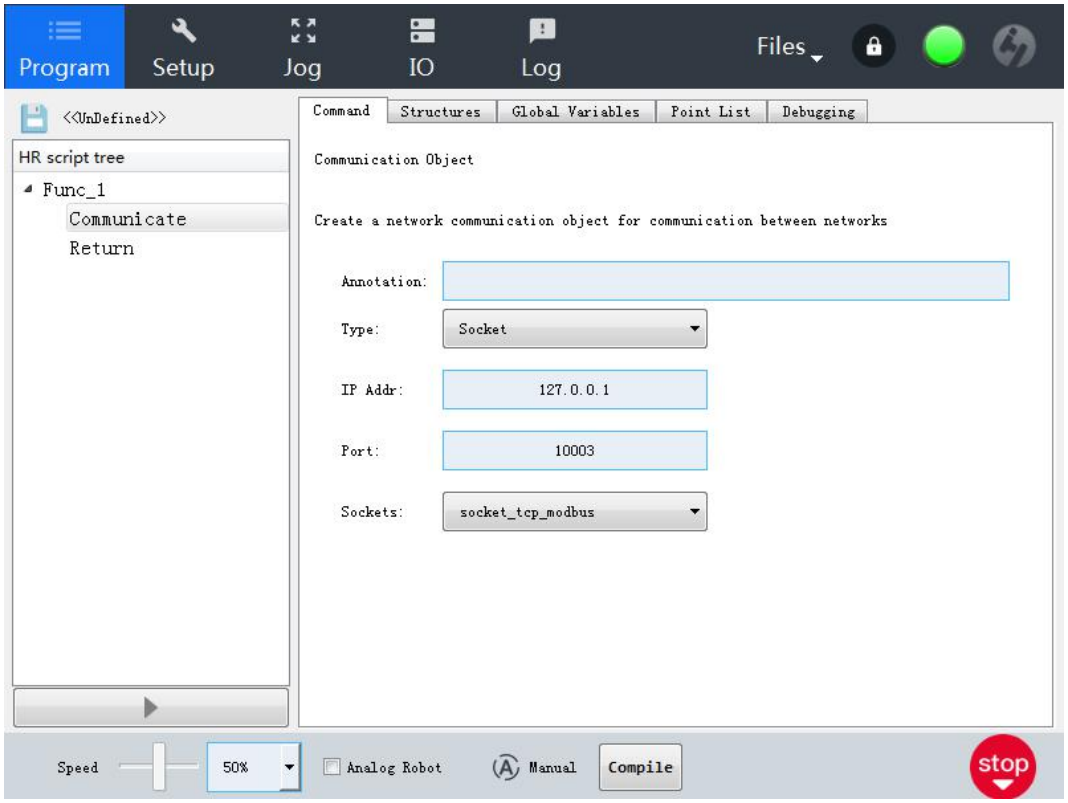
The following is a simulation of the user equipment with the ModSim32.exe tool.

❶Variant:

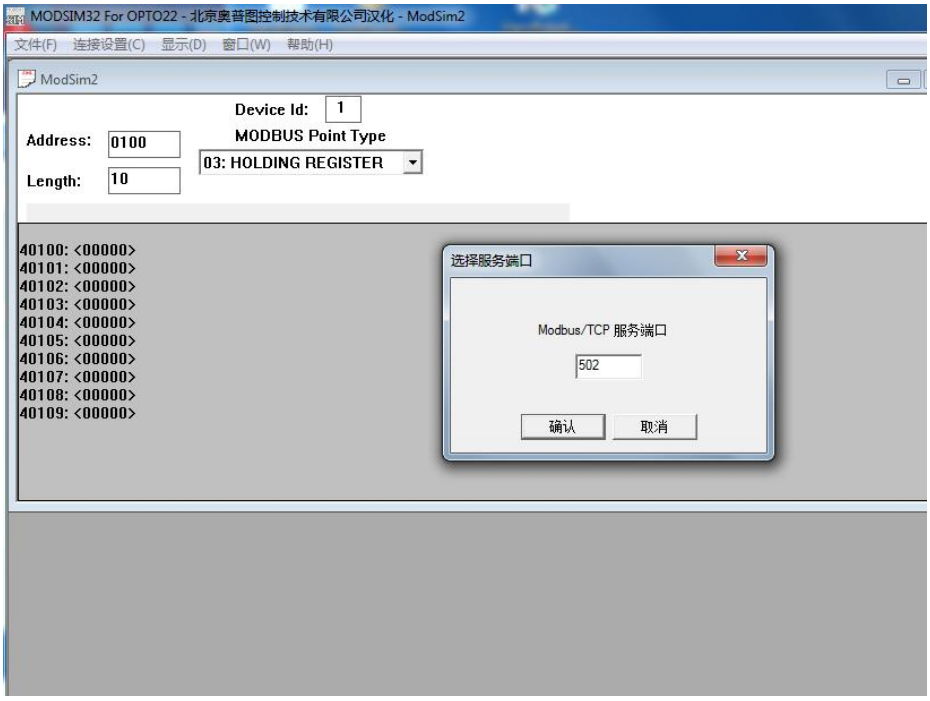
Added socket type variable, named socket_MODBUS (for new connection slot) and ten integer int variables (for reading serial port data).

❷Establish a configuration connection

MODBUS communication selects MODBUS, configures IP (using loopback address Loopback Address127.0.0.1 or host address) and Port (sets consistently), socket type variable (socket_MODBUS) created before Socket use.

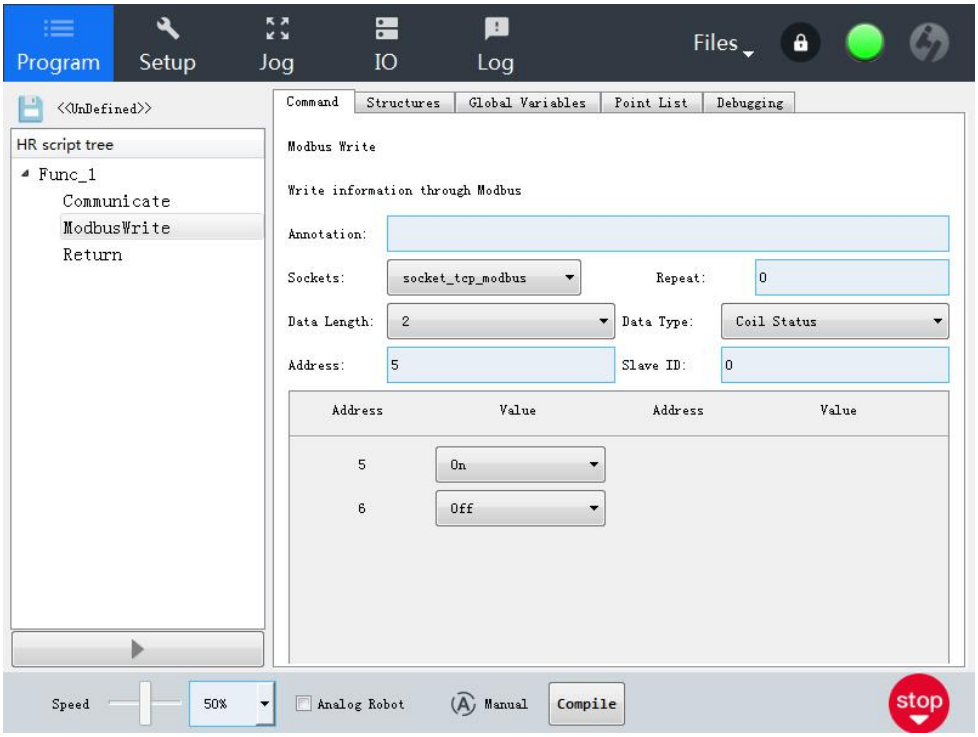


❸The ModSim32.exe tool simulates user equipment, as shown in the following figure to configure virtual serial port:



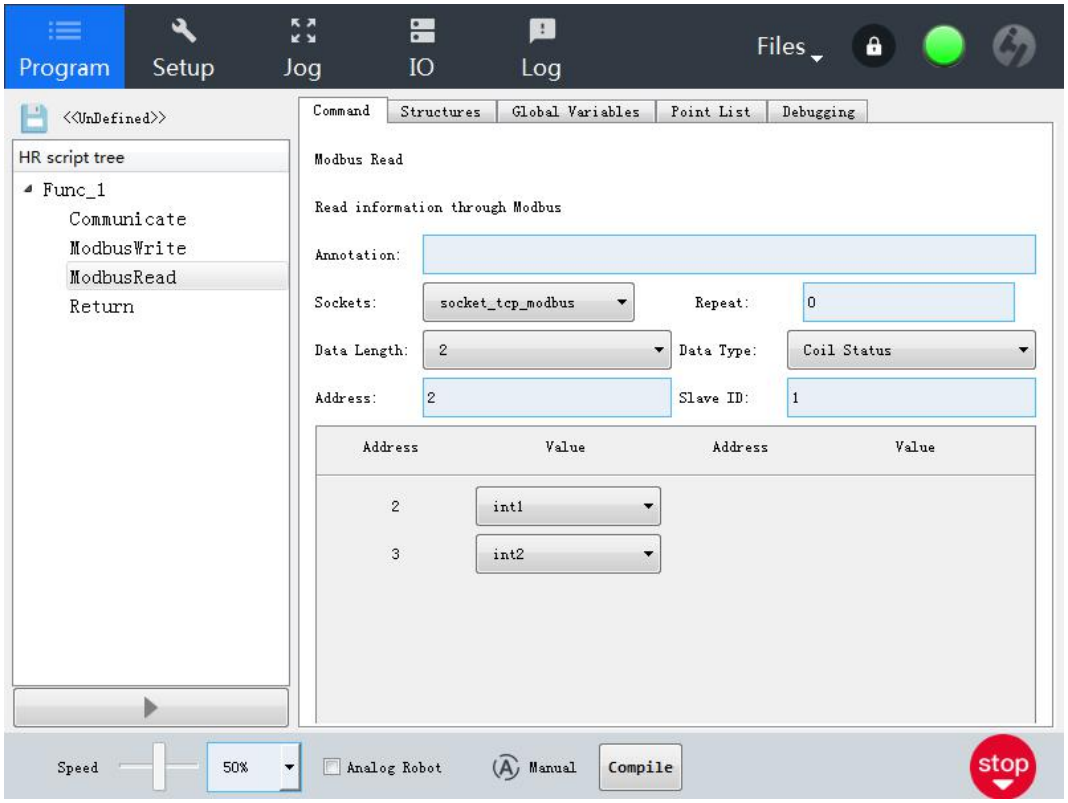
❹**Modbus write**

Note: The Data Type, Address, and Slave ID need to be consistent with the MODBUS slave information.



❺**ModbusRead:**Accept the message in the integer format (int), select the previously created integer variablein(t_3,int_4,...,int_12)

Note: The Data Type, Address, and Slave ID need to be consistent with the MODBUS slave information



Application template

➔ Seek

The Seek Operation includes stacking and de-stacking, and the sequence of motions can be performed on a given set of positions on a straight line. The seek command can use the sensor to determine if the robot tool has reached the target position to grab or lower the workpiece. The sensor can be a push button switch, a pressure sensor or a capacitive sensor. Suitable for handling cases where the thickness of the workpiece or the precise position of the workpiece is not clear.

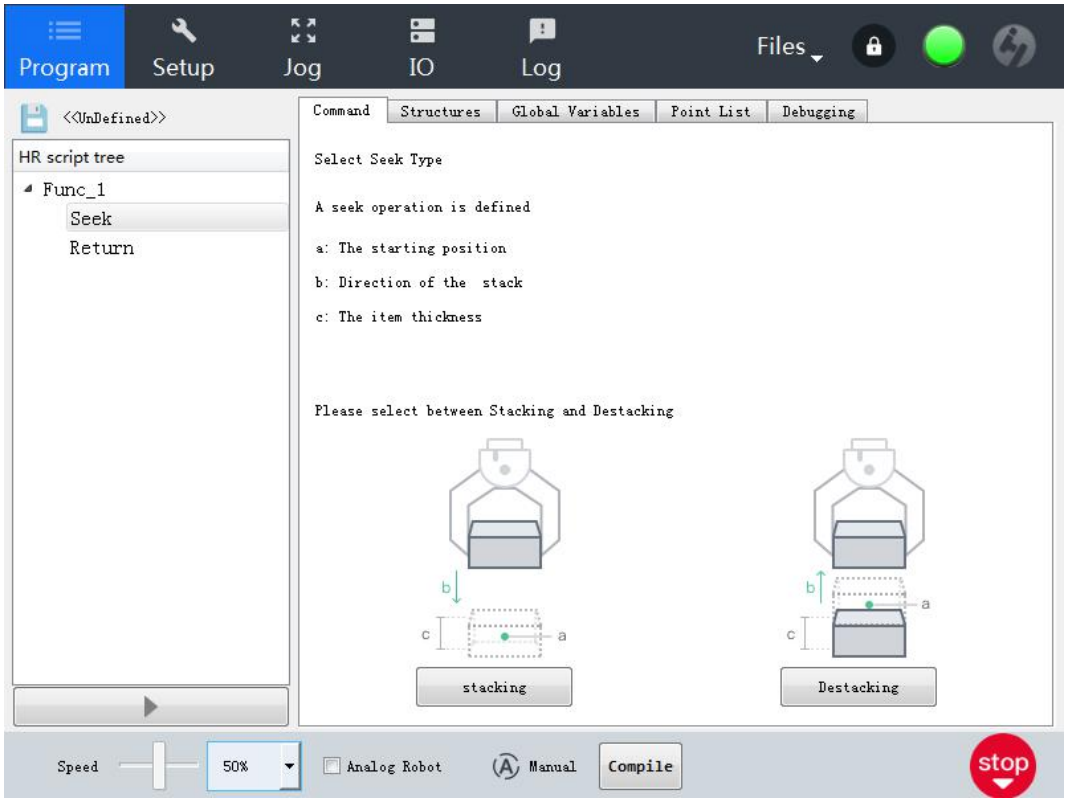
The following information is needed for writing the stacking and de-stacking function:

1、 a（the starting position）、d（direction of the stack）and c（the item thickness）;

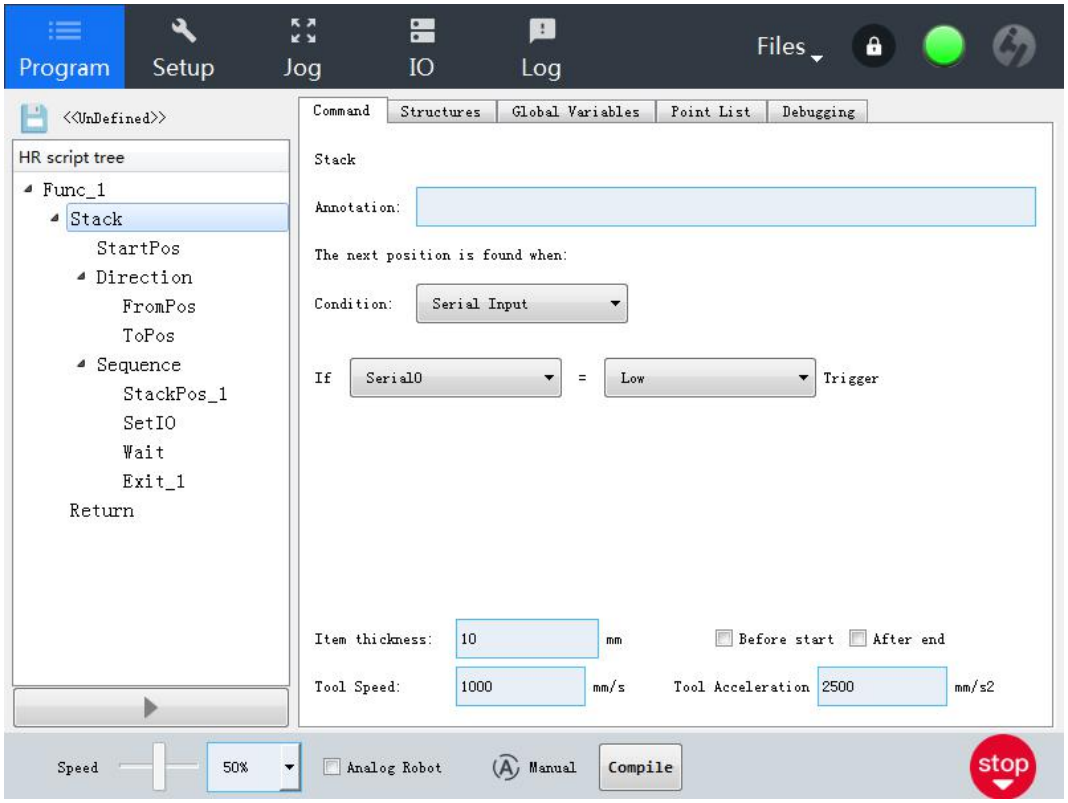
2、 When to reach the next stacking/de-stacking position;

3、 Sequence program to be executed for each stacking/de-stacking position;

4、 When to end the stacking/de-stacking;



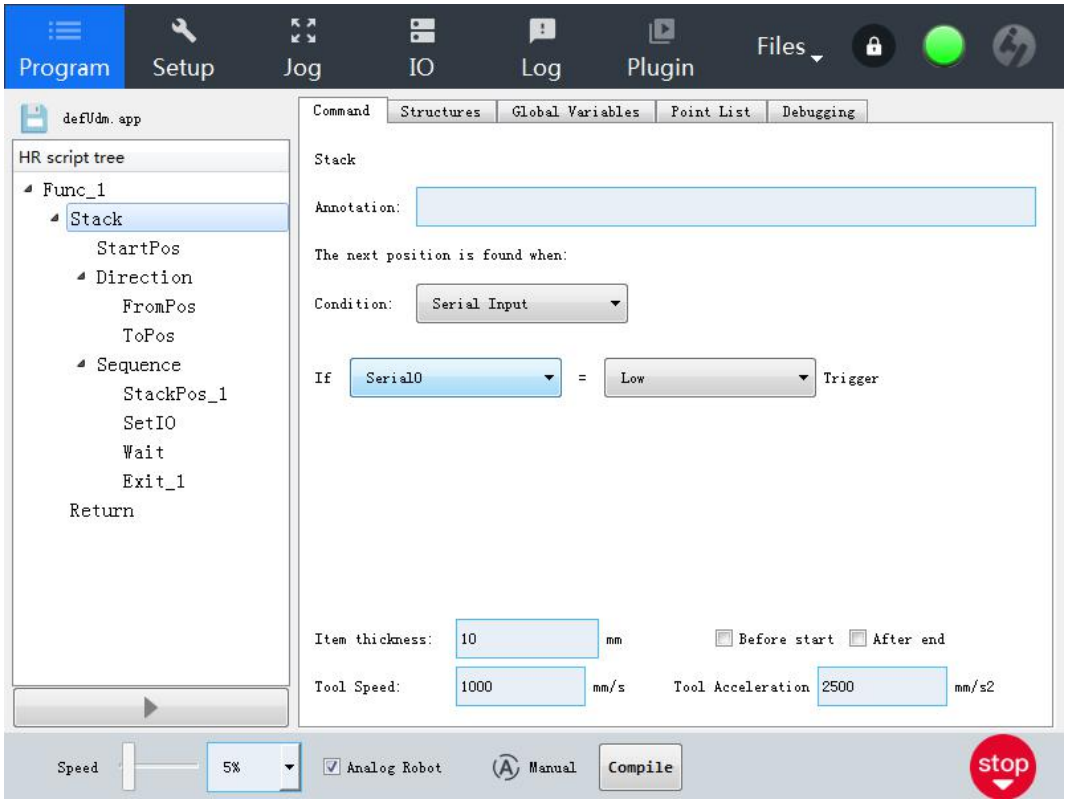
Stack:



Stacking process: The robot arm will move to the start position and then move back to the next stacking position in increments of the workpiece thickness. The robot remembers the point position and executes the sequence structure program. In the next round, the robot will start moving from the position you remember. If it is satisfied to find the next position condition, move to the next stacking position, if it is not satisfied, move in the stacking direction at a speed of 10 mm/s until the condition is satisfied, and then execute the sequence structure program. When the pallet height exceeds the defined value, or the stop condition is satisfied, the stacking is ended.

If "Before start" or "After end" is selected, the call function will be executed before or after the stacking process starts.

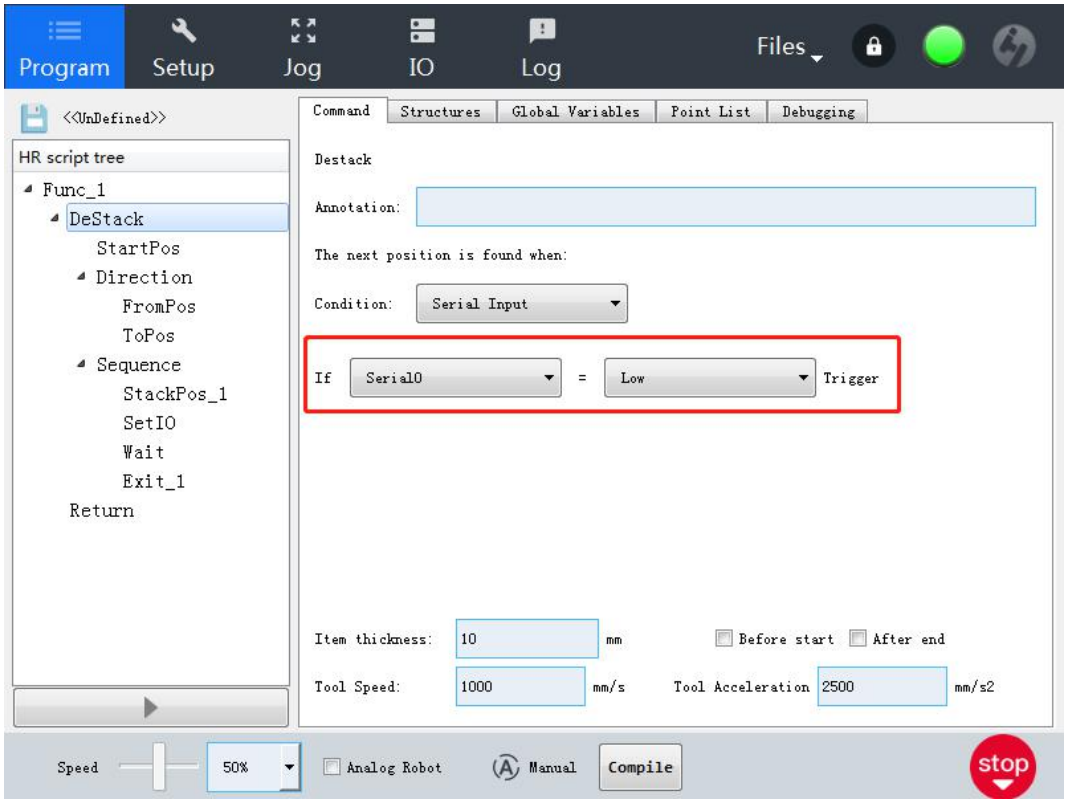
Unstack:



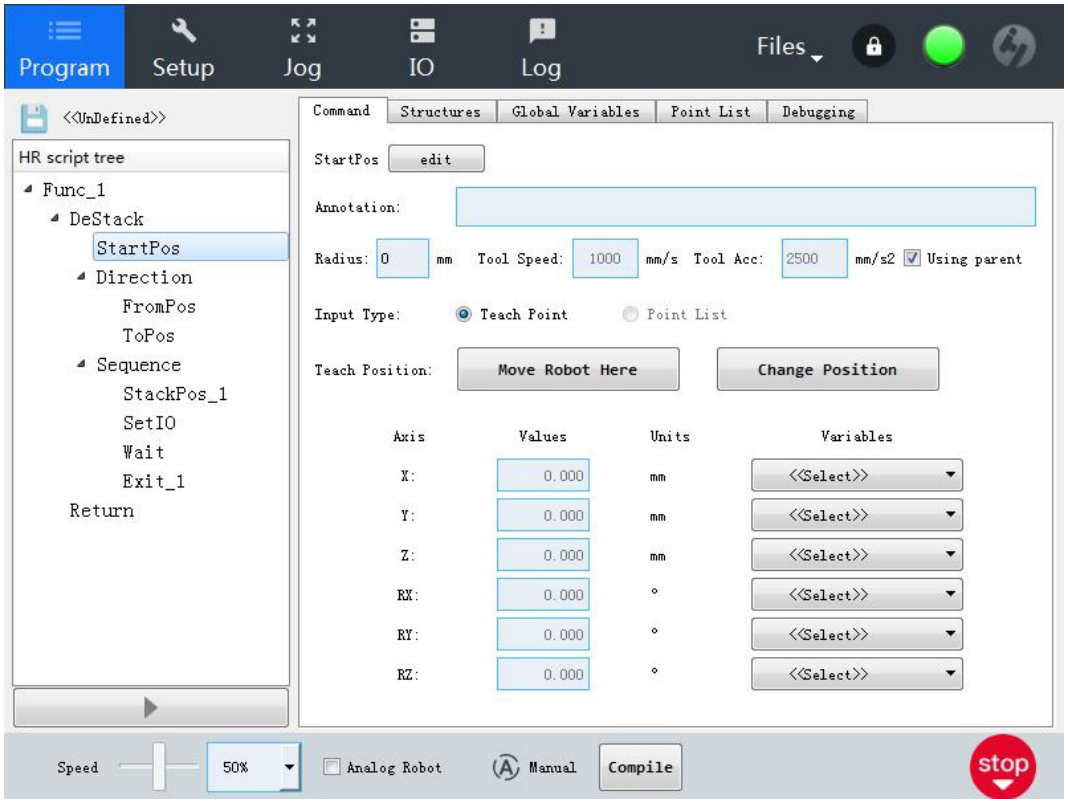
Destacking process: The robot arm will move to the start position and then move back to the next de-stacking position in increments of the workpiece thickness. The robot remembers the position and executes the sequence structure program. In the next round, the robot will start moving from the position you remember. If it is satisfied to find the next position condition, move to the next de-stacking position, if it is not satisfied, move in the opposite direction of the de-stacking at 10 mm/s until the condition is satisfied, and then execute the sequence structure program. End stacking when the pallet height exceeds the defined value, or if the stop condition is met.

If "Before start" or "After end" is selected, the call function will be executed before or after the de-stacking process starts.

The next stack position expression: if it is satisfied to find the next position condition, move to the next stacking/de-stacking position, if it is not satisfied, it will move in the stacking direction/removal direction at the speed of 10mm/s until the condition is met, and then execute.



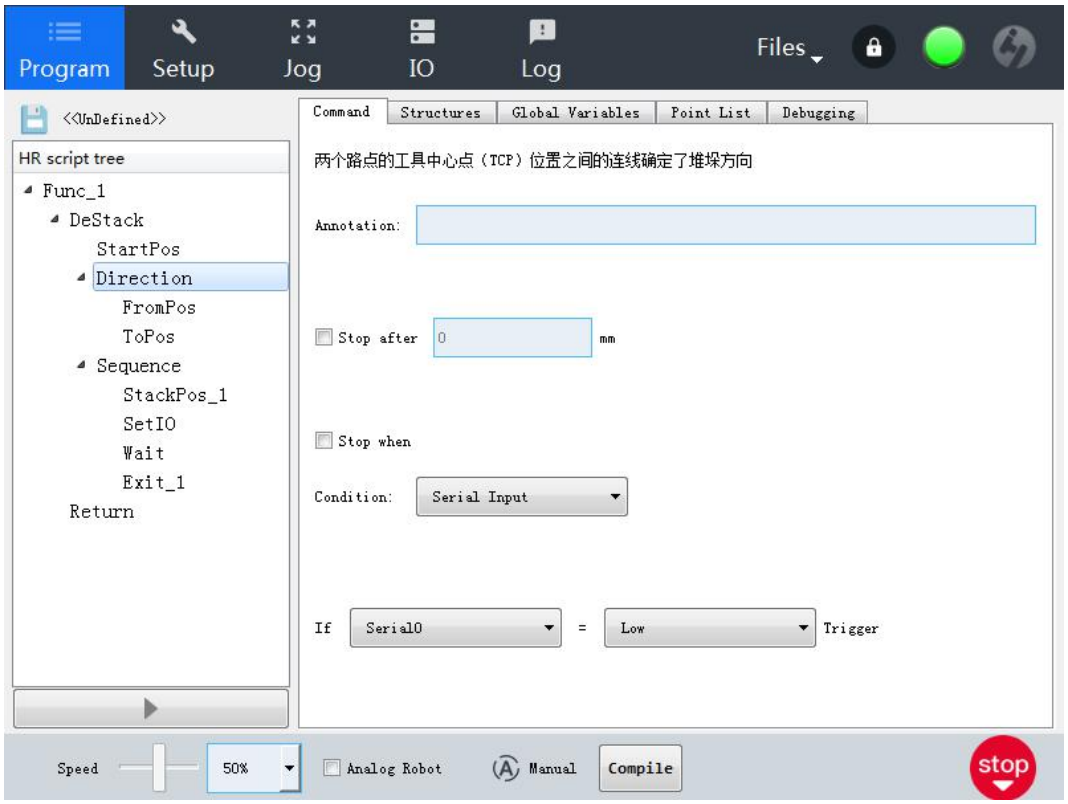
Start Position：Refers to the location where the stacking/de-stacking operation is started.



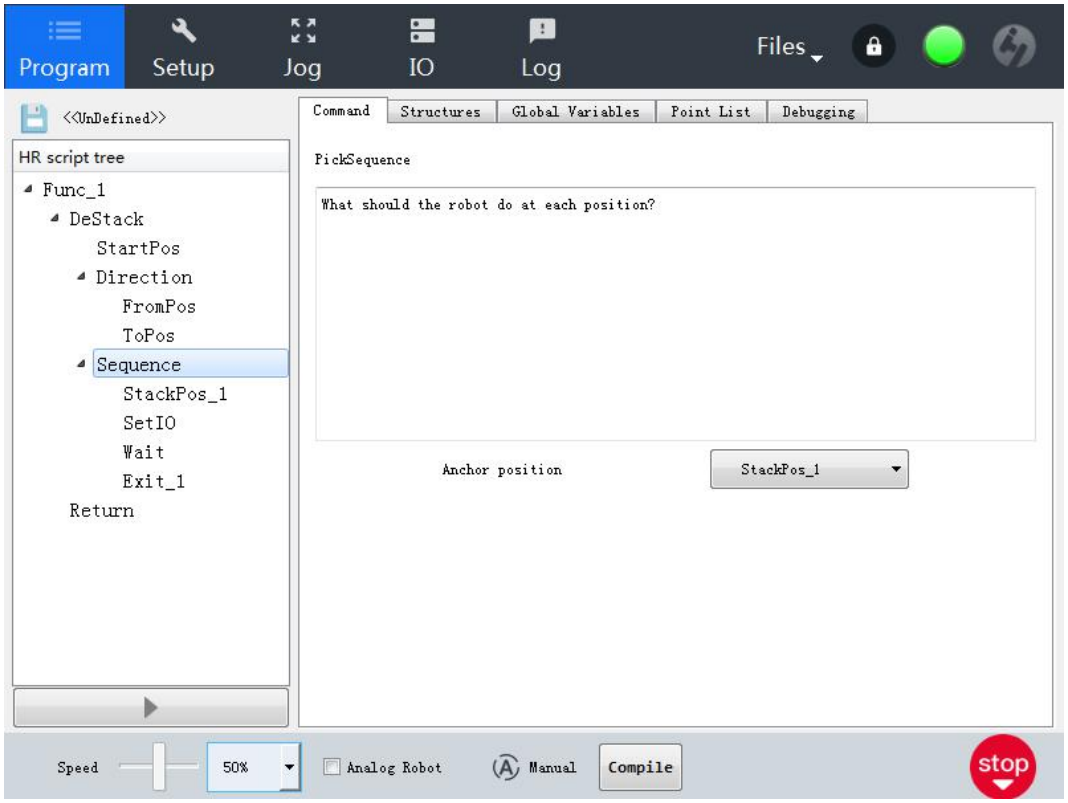
The direction: the direction of stacking/de-stacking is determined by two positions, and the position of FromPos points to the direction of ToPos in a positive direction. Note: the direction does not take the pointing gesture into account.

Stop when: There are two kinds of judgment conditions. 1. Determine whether the distance between the next stacking/de-stacking position and the starting point is greater than or equal to the set stopping distance; 2. The custom condition judgment can be selected by IO state or expression judgment.

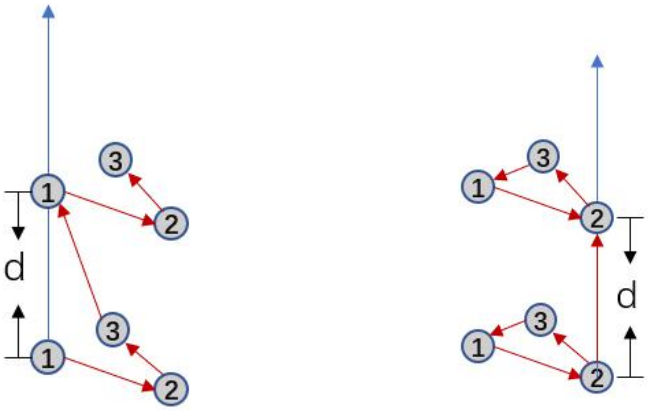
Choose two conditions to satisfy one of them and stop.



Sequence: The sequence describes the operations that should be done at each mode location. In the sequence node, the robot arm moves relative to the anchor position.



Taking stacking as an example:



Red is the motion tracking, and the running track is different according to the selected different anchor position.

The left picture shows the motion trajectory with Point_1 as the anchor point. The motion trajectory of the running sequence is 1->2->3; the right picture shows the motion trajectory with Point_2 as the anchor point. The motion trajectory of the running sequence is 2->3. ->1->2. The sequence motion instruction is executed. The motion point

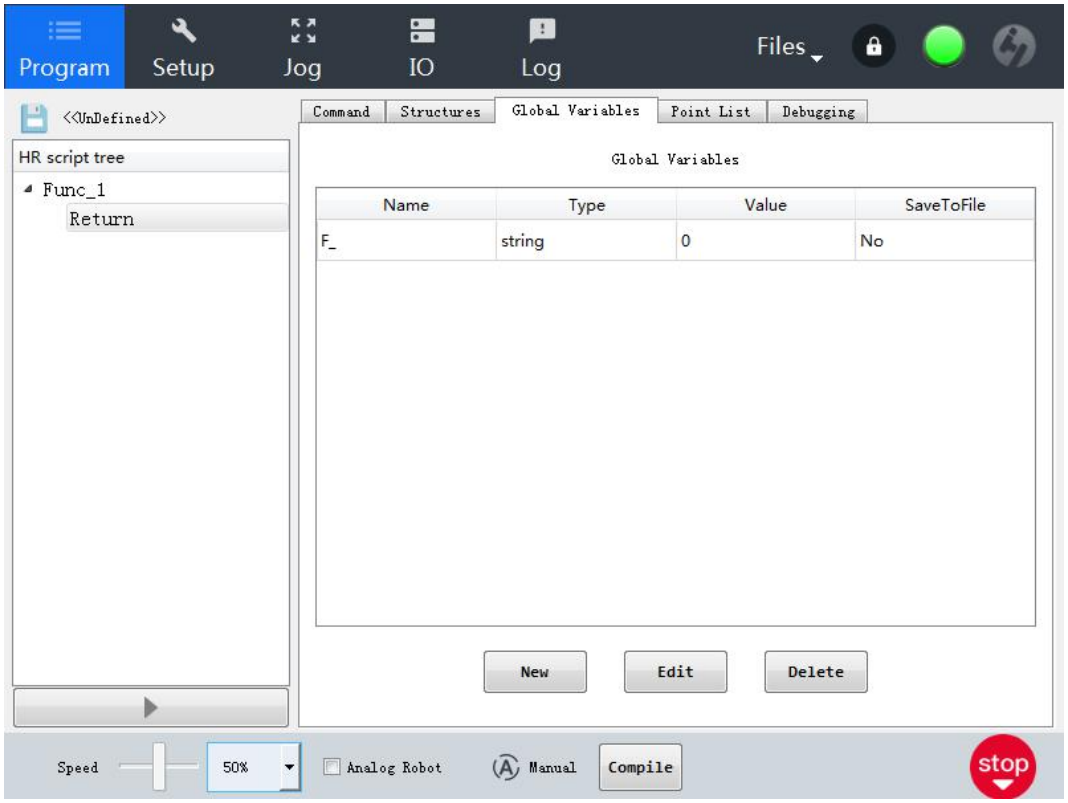
is not the point of teaching, but the relative position. The anchor point is the sticking point, and the other points are the relative positions of the anchor points.

Simplified Script Method

When the script logic has a lot of repetitive If~elseif judgments and the different functions are called, the following methods are provided to simplify the script.

➔ 1、ReadSerial、StringAdd、Call Command

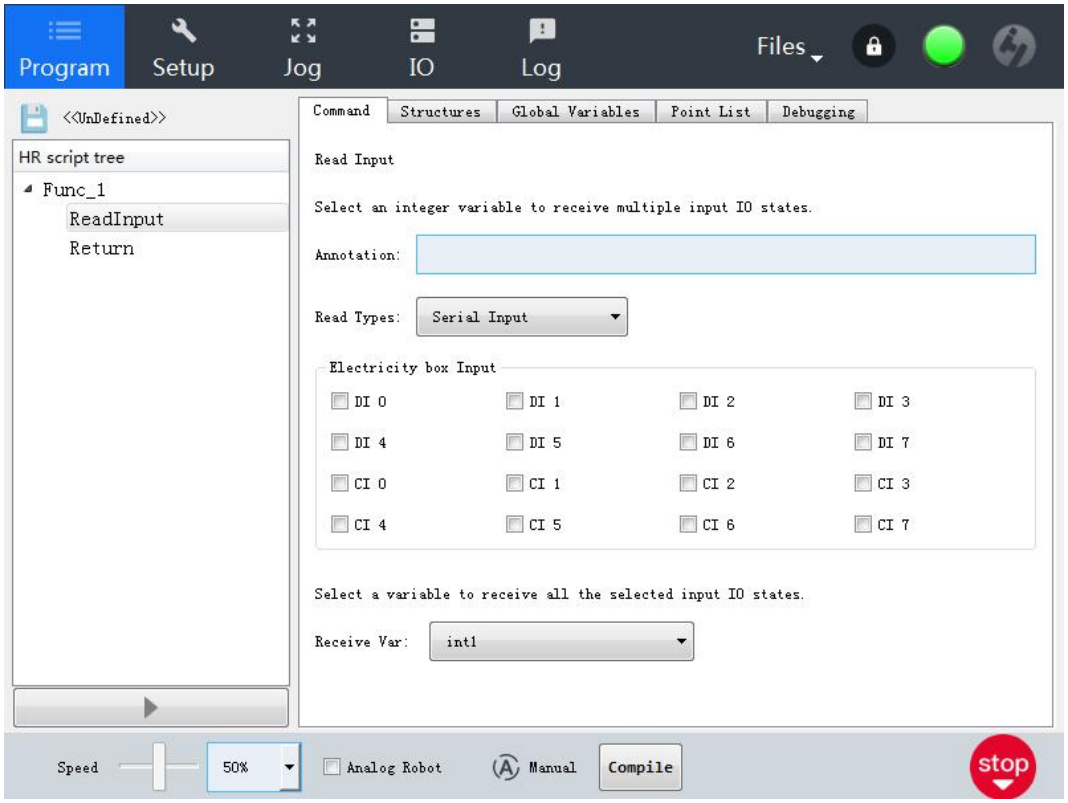
❶Click to enter the "command edit" interface, switch to the Global Variables interface, create a new string variable, and set the variable name to "F_" (name can only be F_), then click the "OK", "Save" button to complete the new variable;



❷Create a second string variable, the name can be freely defined, such as string1;

❸Create a second int variable, the name can be freely defined, such as int1;

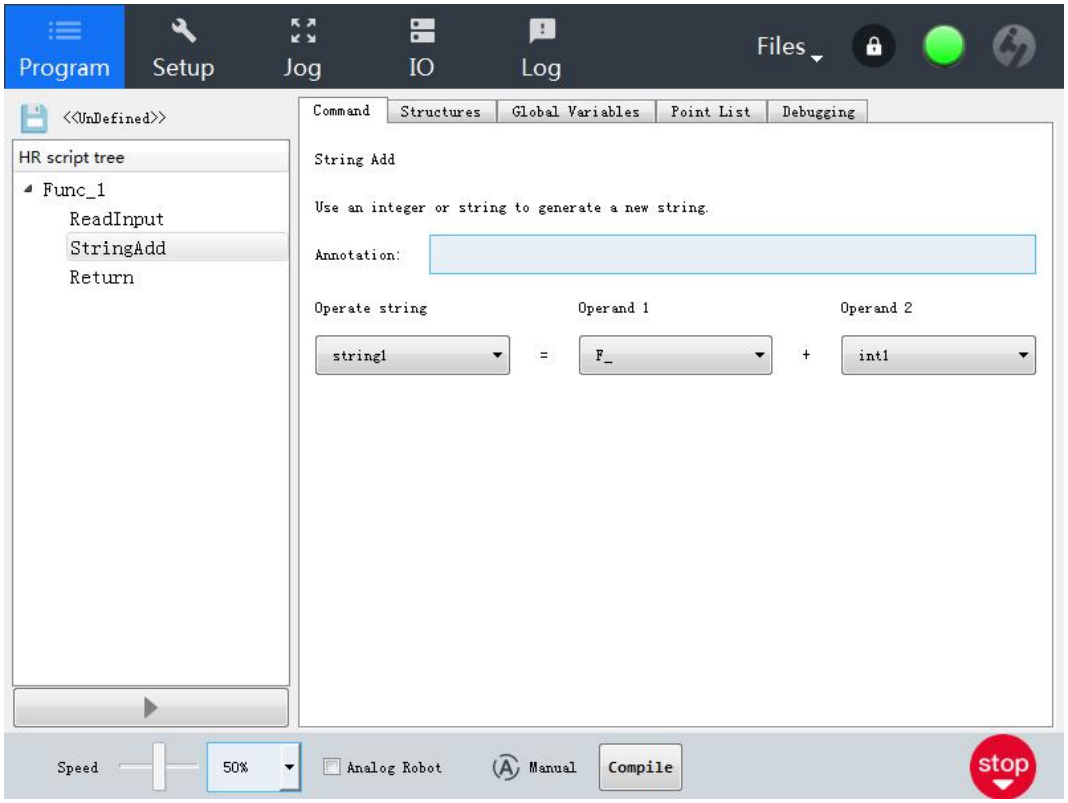
❹Add the ReadInput command in the program, check the option in front of the serial port IO to be monitored, and select the new int variable in step 3 in the Write Variable option, and then click the "Save" button.



The value of the int variable is determined by:

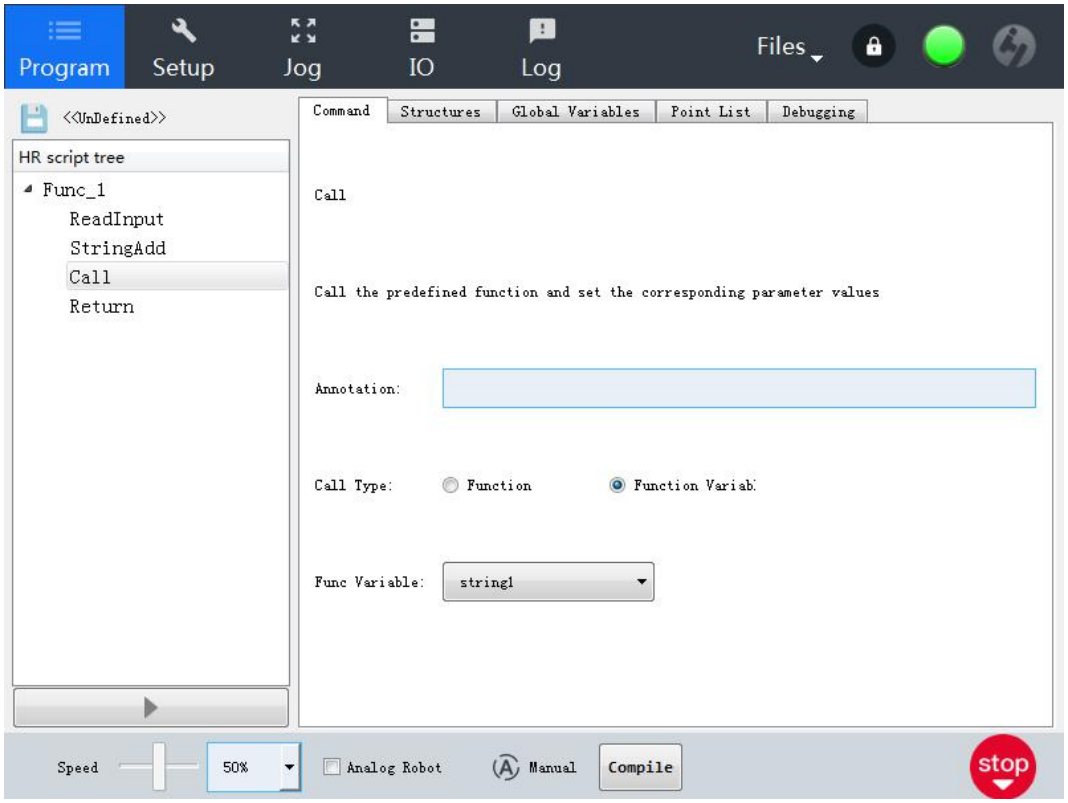
If S0, S3, S6 are selected, the IO signal state is written into the int variable. When the signals of S0, S3, and S6 are both set high, that is, int=2^1 +2^1 +2^2=7; if the signals of S0 and S3 are set high, S6 When the signal is de-asserted, int=2^0 +2^1 =3.

❺Add the StringAdd command to the program, select the string variable named "F_" in step 1 from the parameter 1 drop-down list in the instruction details, select the new int1 variable in the drop-down list axis of parameter 2, and select in the drop-down list of the result. Step 2 New string1 variable, then click the "Save" button;



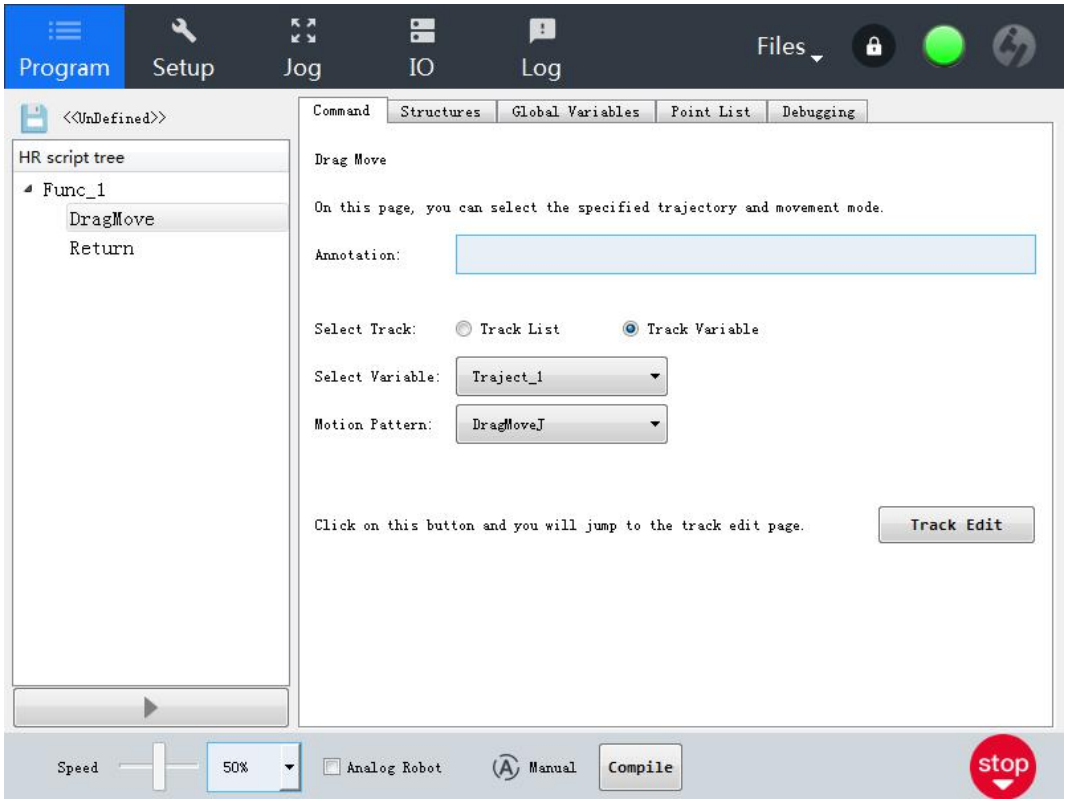
As shown in the figure above, because the variable name of parameter 1 is "F_", when the value of the parameter 2int variable is equal to 7, the call of the Call command is the F_7 function; if the value of the parameter 2int variable is equal to 8, the call of the Call command is the F_8 function;

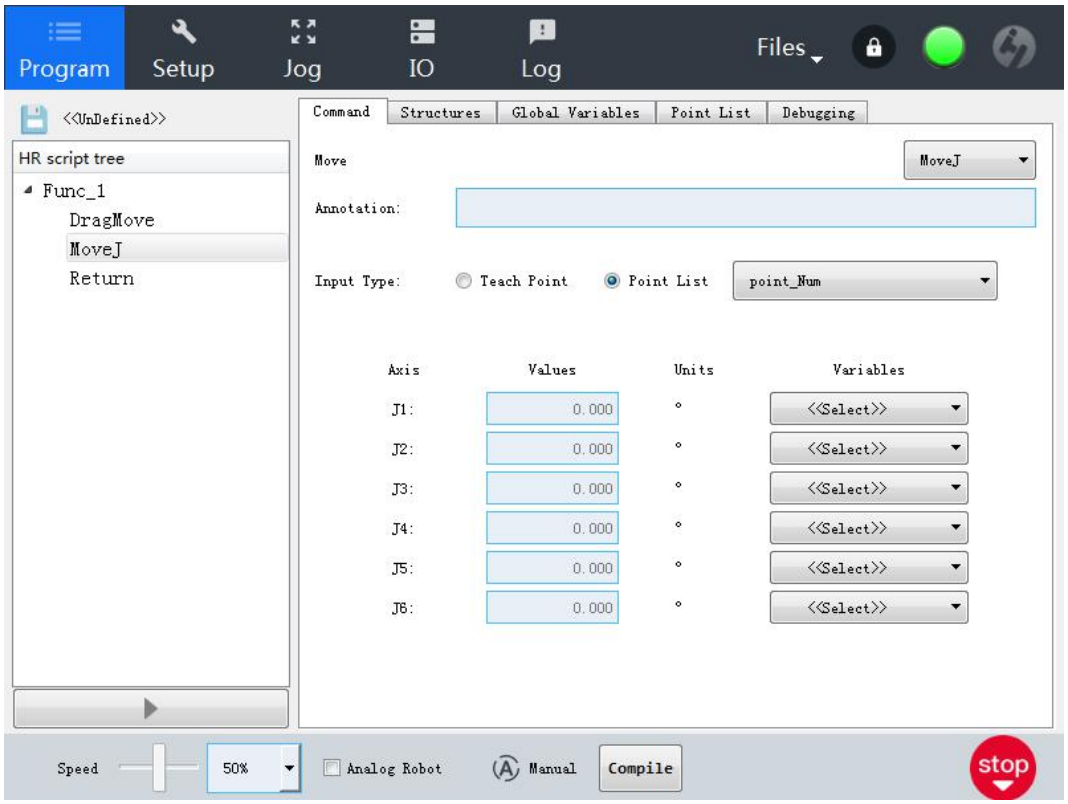
❻Add the Call command to the program, select the "Func Variable" option in the command interface, select the result variable set in step 4 in the drop-down list that follows, and then click the "Save" button.



➔ 2、Trajectory

DragMove and MoveJ/MoveL also provide call variables to run the track, providing the ability to change the value of the variable to reach an instruction to call a different track. For specific operations, please refer to the previous one.


Program
Setup
Jog
IO
Log
Files
<<UnDefined>>
HR script tree
Func_1
DragMove
Return
Command
Structures
Global Variables
Point List
Debugging
Drag Move
On this page, you can select the specified trajectory and movement mode.
Annotation:
Select Track:
Track List
Track Variable
Select Variable:
Traject_1
Motion Pattern:
DragMoveJ
Click on this button and you will jump to the track edit page.
Track Edit
Speed
50%
Analog Robot
Manual
Compile
stop


Program
Setup
Jog
IO
Log
Files
<<UnDefined>>
HR script tree
Func_1
DragMove
MoveJ
Return
Command
Structures
Global Variables
Point List
Debugging
Move
MoveJ
Annotation:
Input Type:
Teach Point
Point List
point_Num
Axis
Values
Units
Variables
J1:
J2:
J3:
J4:
J5:
J6:
0.000
<<Select>>
Speed
50%
Analog Robot
Manual
Compile
stop

➔ Conveyor belt application

一、Start with punctuation of the conveyor belt， refer to section 7.6.

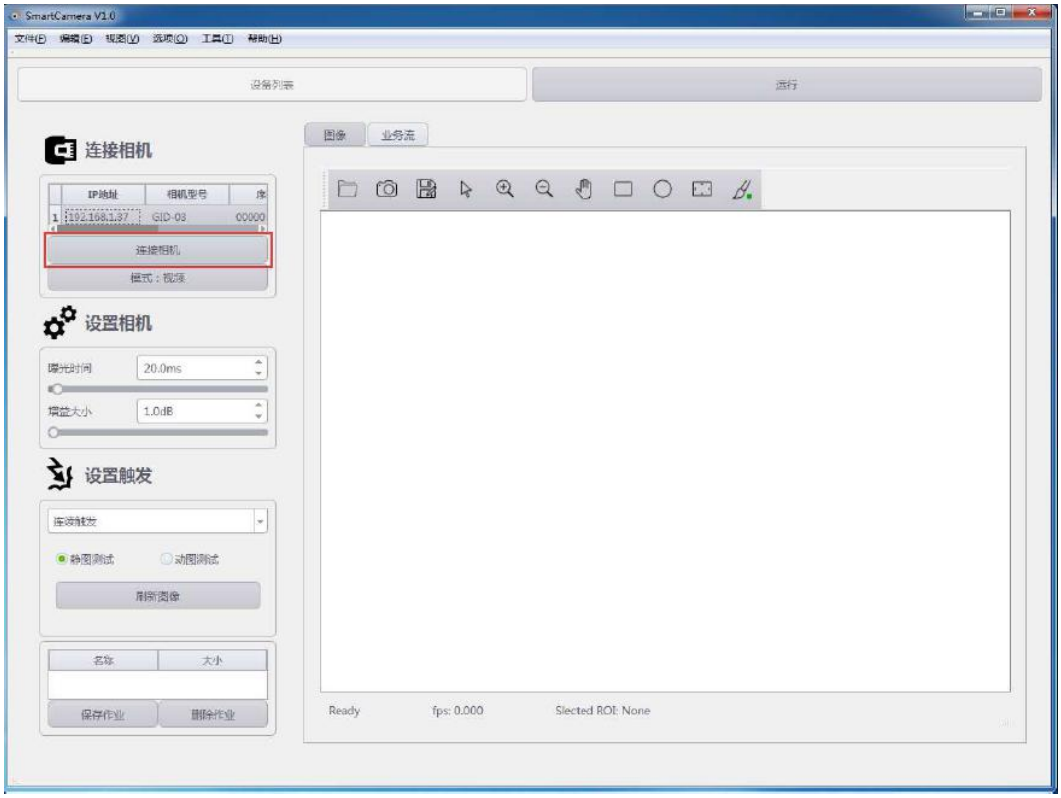
二、Camera calibration

Take Yuanqi smart camera as an example：

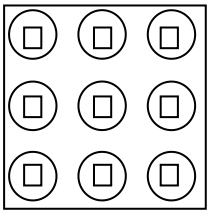
First connect the IP， making them in the same domain.

Controller client IP：192.168.1.100

Smart camera IP：192.168.1.38， port：3001



Enter the camera SmartCamera host computer system, click on the service flow, expand the right image processing bar and enter the Conveyor Location Tools interface to take the calibration.



Calibration rules (from camera perspective);

Location:Calibration paper on the conveyor belt.

Nine-point calibration:P0→P8;

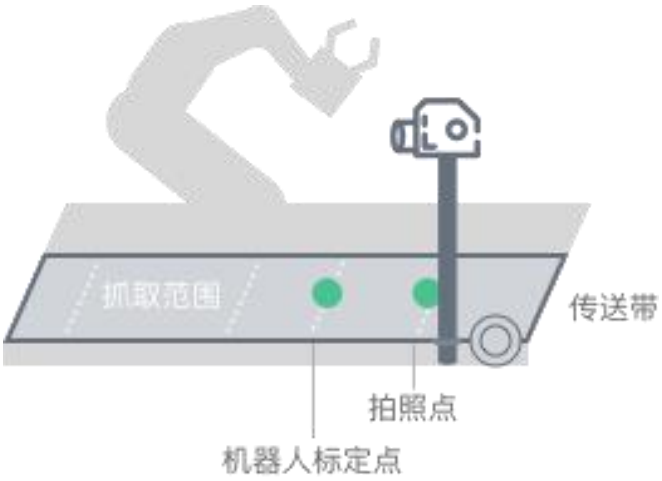
Let the calibration needle be the black point of the center of each calibration circle

The Z axis is unchanged, only the XY axis is changed to ensure that they are in the same plane;

When the position is calibrated, click the corresponding point to get the coordinates,

and after completing the teaching of 9 points, complete the camera calibration.

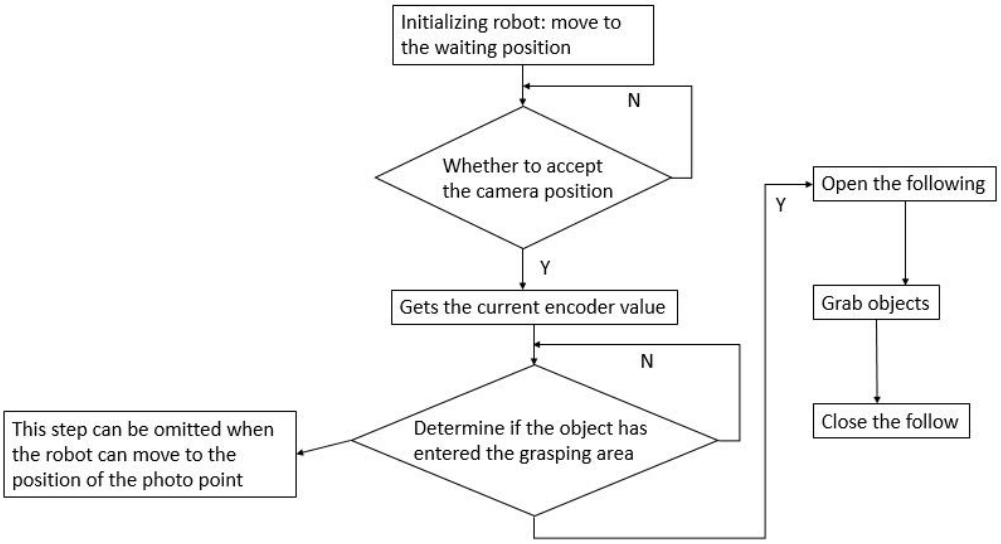
Note: If the robot can't move to the bottom of the camera, you need to record the distance offset between the “photographing point” and the “robot calibration point” when punctuating. If the robot can move to the punctuation below the camera, no need to set the parameter.



三、Grab process

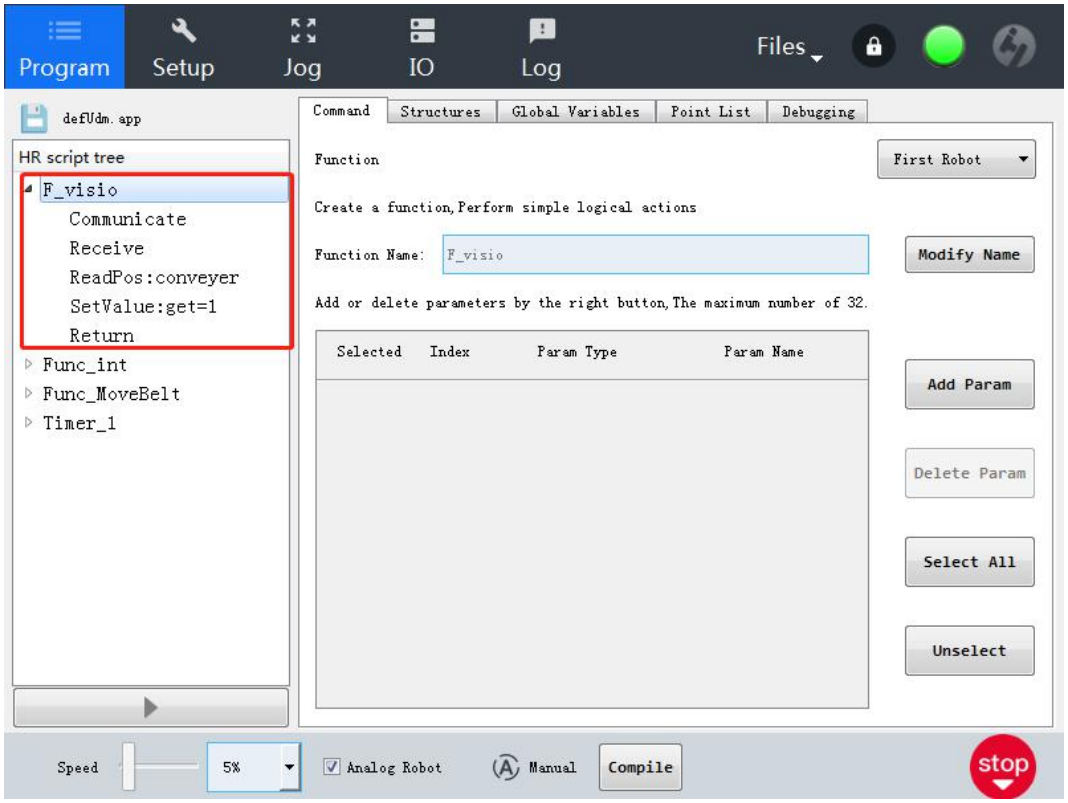
Provide a script for the conveyor to grab objects, and explain how the HansRobot follows the script by scripting.

The following is a script flow chart:

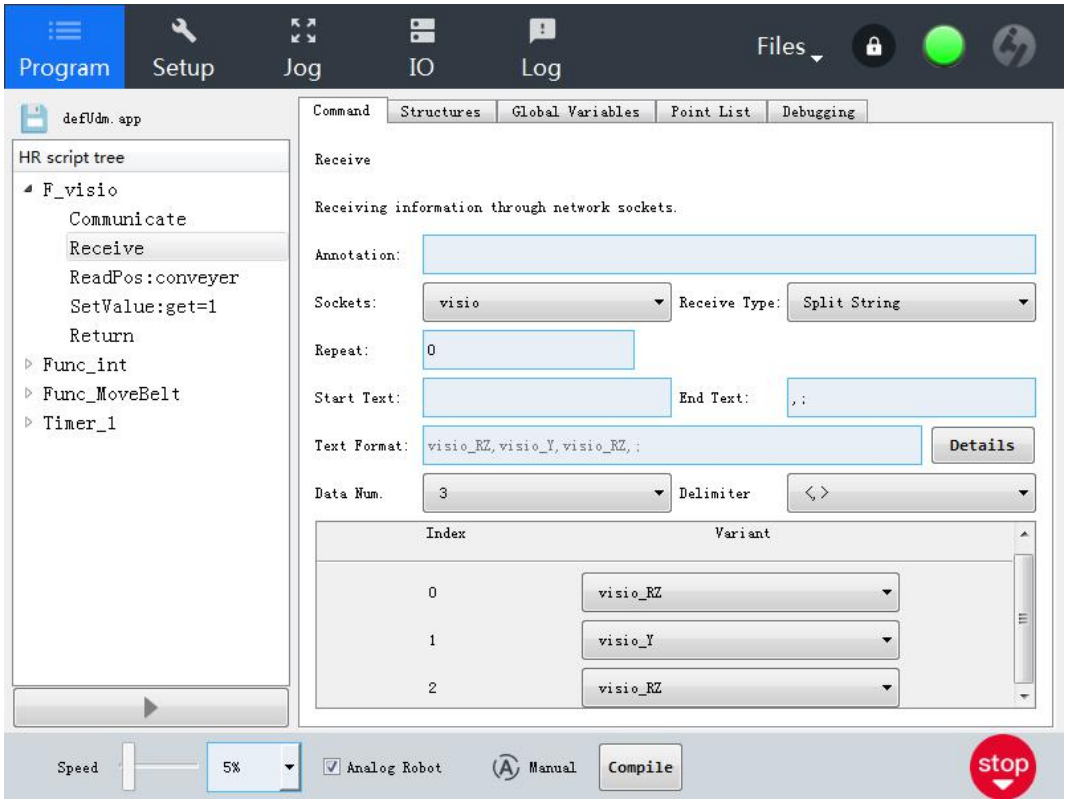


The following is a detailed description of the script:

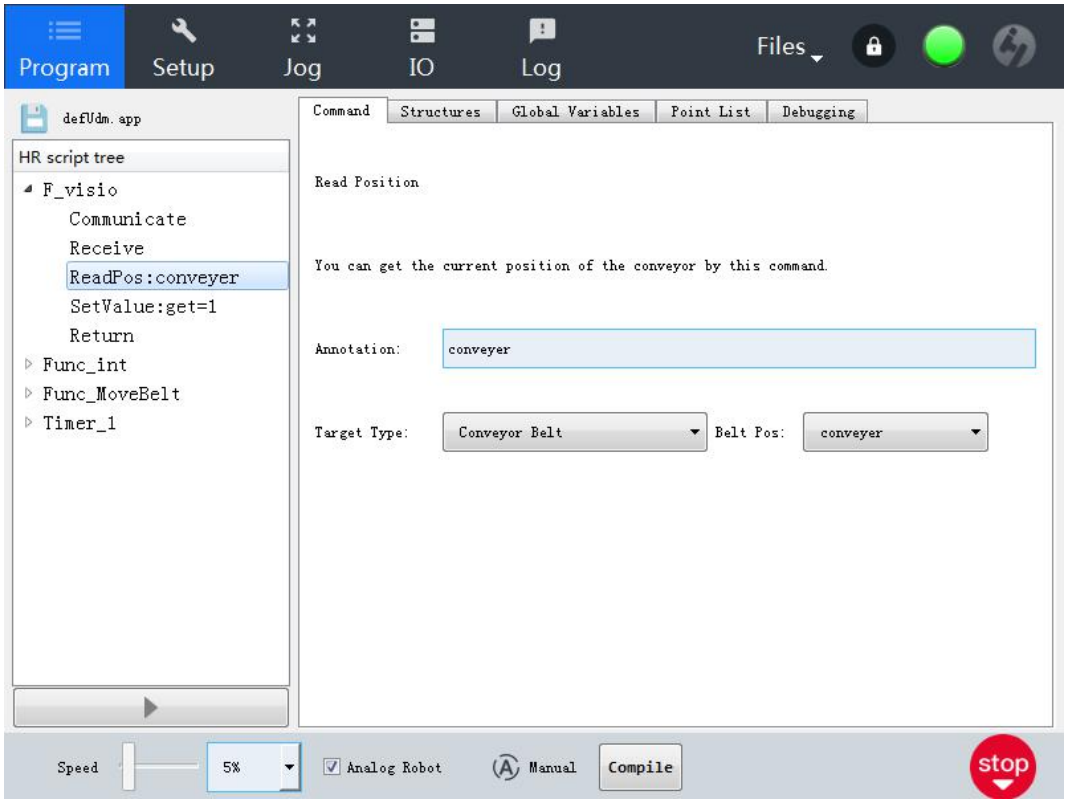
1.Communicate with the camera via TCP/IP:



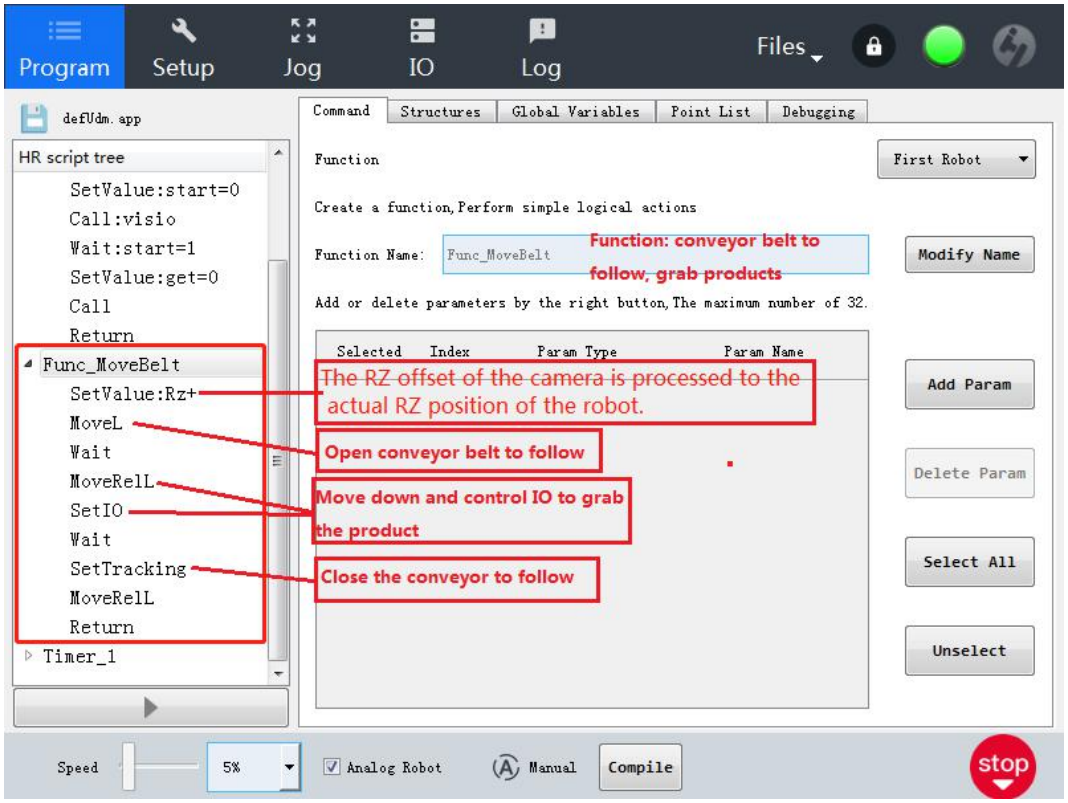
2.Obtain product location information according to the camera communication protocol. The X, Y position sent by the scripted camera is an absolute position, and RZ is a relative position.



3.Get the current encoder value of conveyor, which is required for subsequent follow-up functions.



4.Write following function.

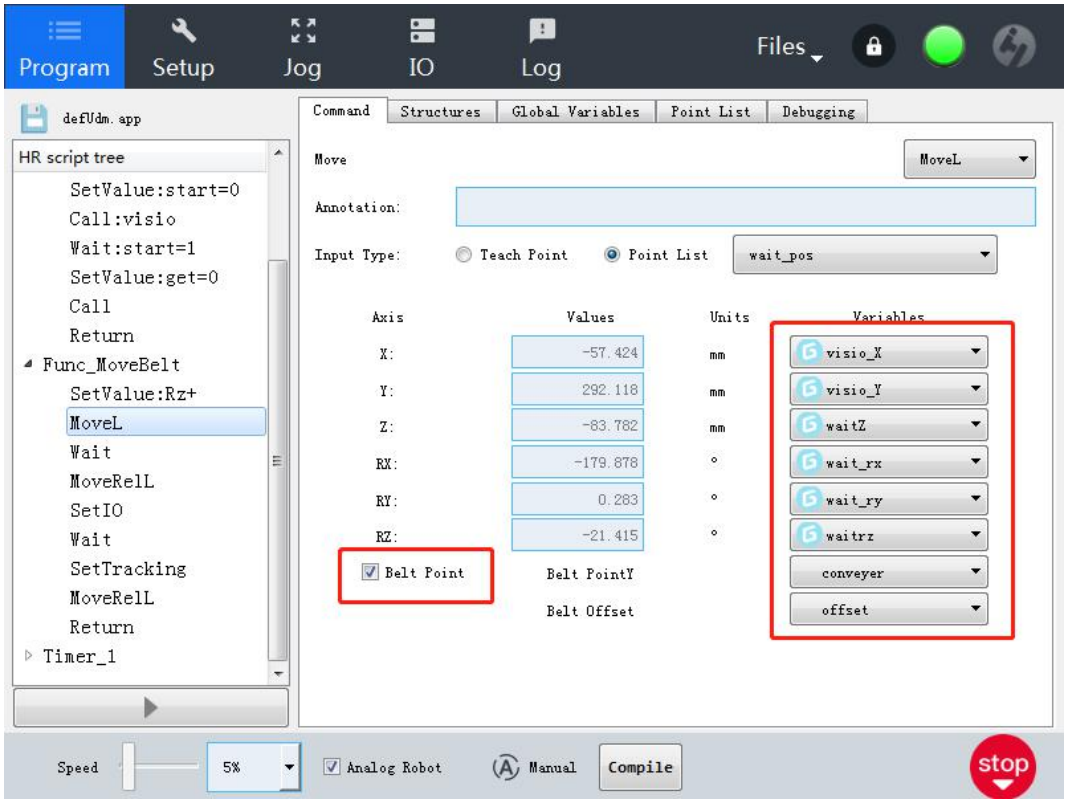


5.Following commands explain:

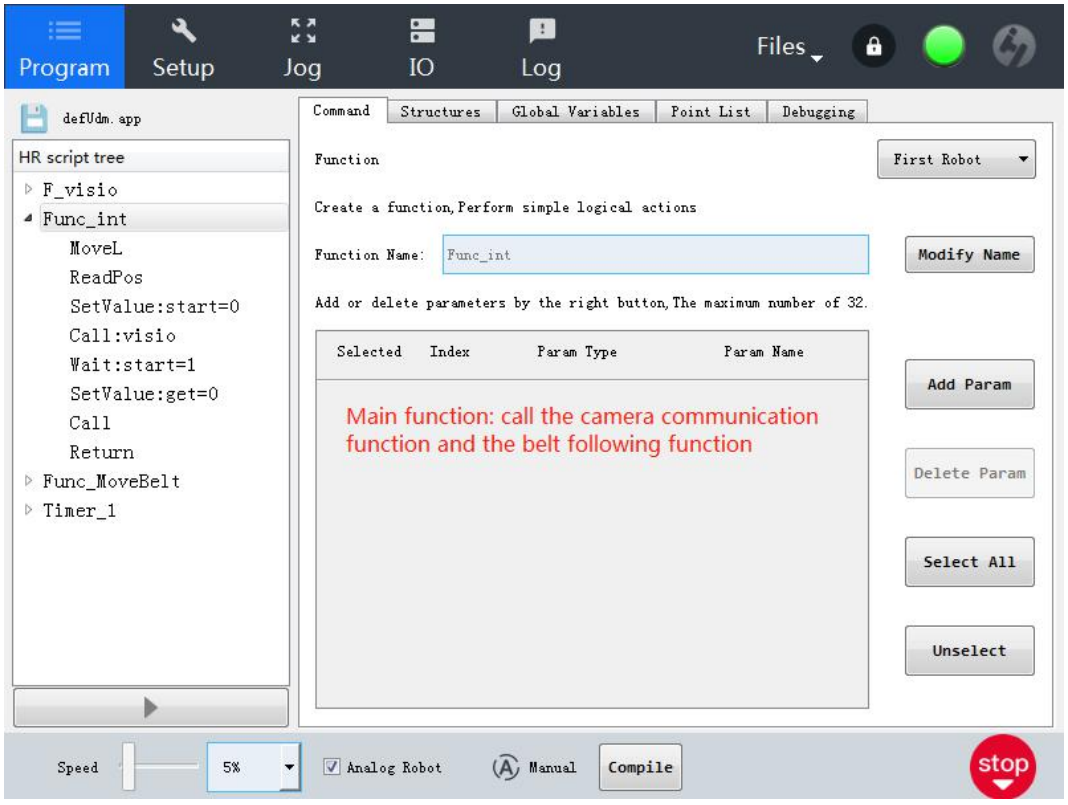
(1)select “Belt Point” to start the following function,and use the right red square data.

(2)Visio_X, Visio_Y is the product location sent by the camera. WaitrZ is the RZ position integration of the robot-calibrated RZ and camera-sending products. Other location information is a location that can be artificially taught to grab objects.

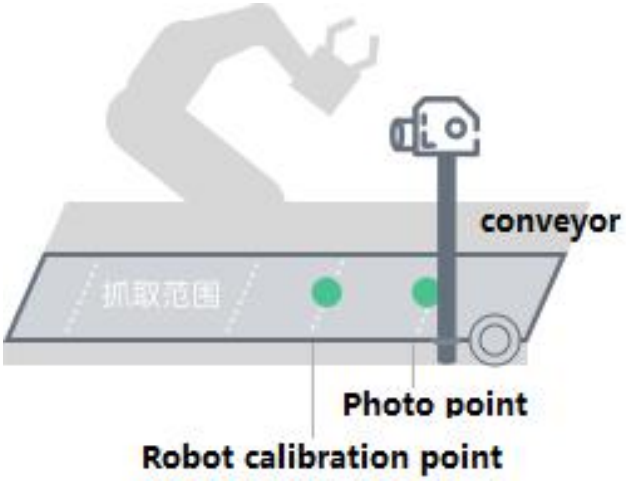
(3)Conveyer is the encoder value obtained when taking photos. When the offset is punctuation, the distance between the “photo point” and the “robot calibration point” needs to be recorded.



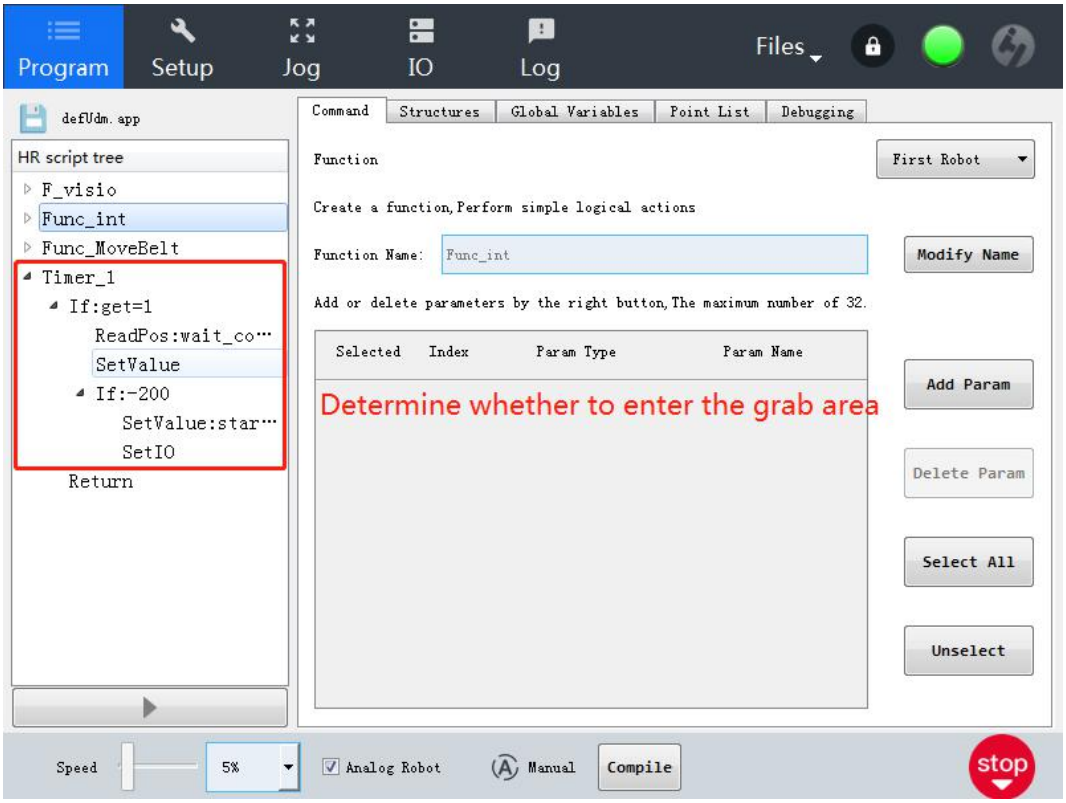
6.The main function calls the machine communication function to get the product location and then calls the following function to grab the product.

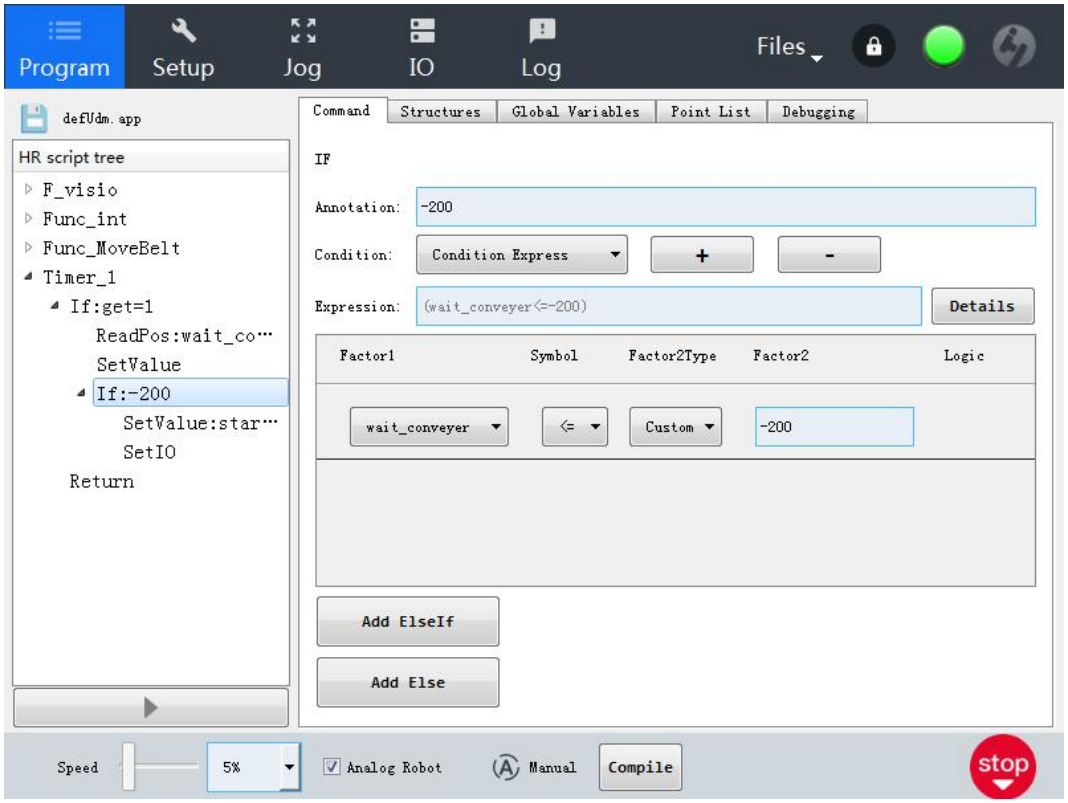


7.If the distance between the photo point and the robot position exceeds the range of motion of the robot, the gripping area needs to be set, and the following is entered in the grabbing. The next step is to set the grabbing area.



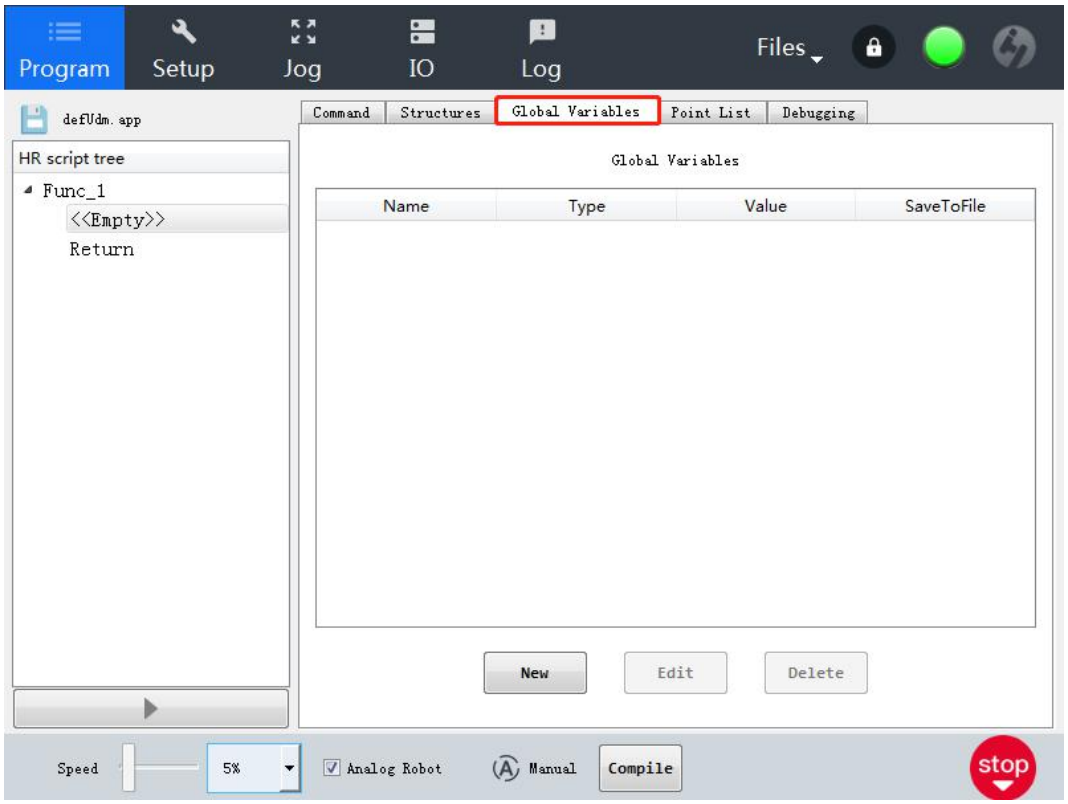
8.Loop to obtain the current encoder value, determine whether the encoder value of the current encoder value and the camera photo position is greater than the set distance if it is greater, it is considered to enter the grab area, and open to follow.





Run the main function to complete a process that grabs objects from the conveyor.

Global Variables



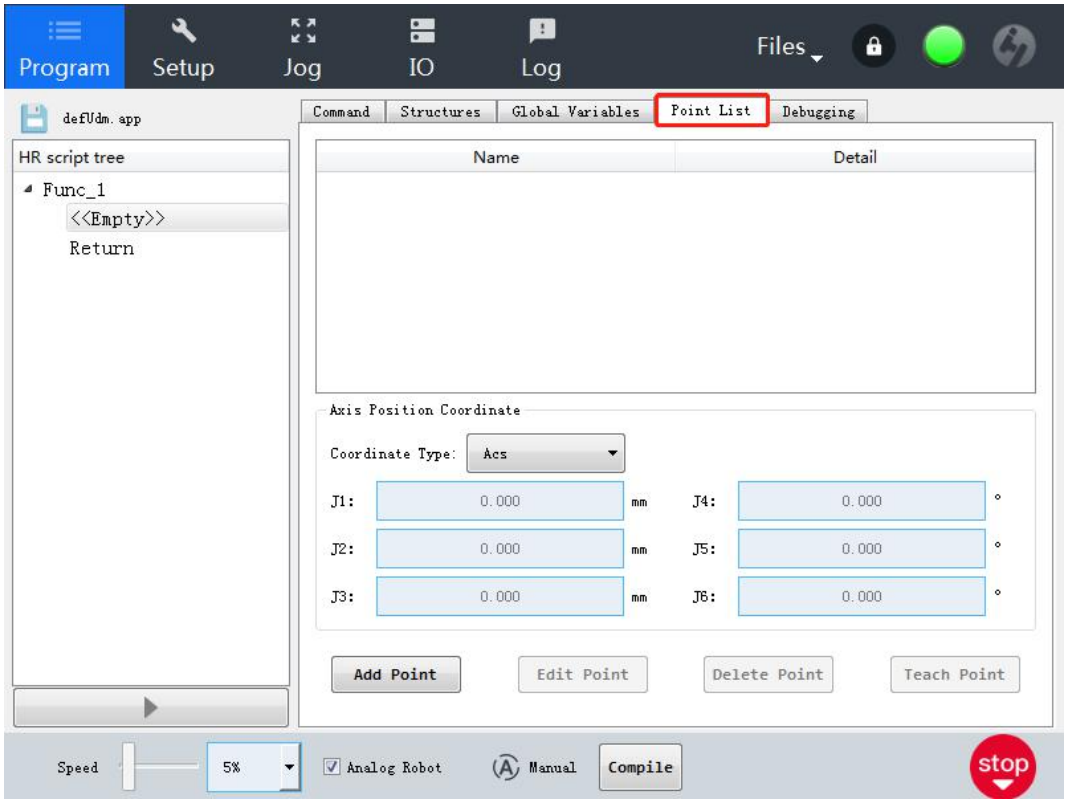
This area edits global variables up to a maximum of 128 variables. The "New" button is used to create a new global variable, the "Edit" button is used to modify the selected variable, and the "Delete" button is used to delete the selected variable. New robot programs can update variable values at runtime via variable assignments such as SetVar.

Types of Variables:

- int variable:range：−32768-32767.
- Double floating point value: floating point number (decimal).
- String:sequence of characters.
- Socket:Communication connection.
- int[]: int array, the corresponding value is the length of the array, the default value of the element is 0.
- doubot[]: doubt array, the corresponding value is the length of the array, the default value of the element is 0.
- string[]: str array, the corresponding value is the length of the array, the default value of the element is a blank character.
- “SaveToFile” function:When "YES" is selected for this option,the value of

the variable is saved in the configuration file in real time.

Point List



This area is a list of edit points. The meanings of the buttons in the interface are as follows:

Name: points name.

Detail: description of related points.

Axis Position Coordinate:Displays the joint coordinates (J1~J6) or spatial position (X, Y, Z, RX, RY, RZ) of the selected point.

Coordinate Type:The ACS indicates that the joint coordinates or the joint angle mode are moved to the teach point; the PCS indicates that the spatial position coordinates are displayed or moved in the spatial position direction to the teaching point.

Add Point:add points to the Point List.

Edit Point:edit the selected point in the Point List to modify its Name and Detail.

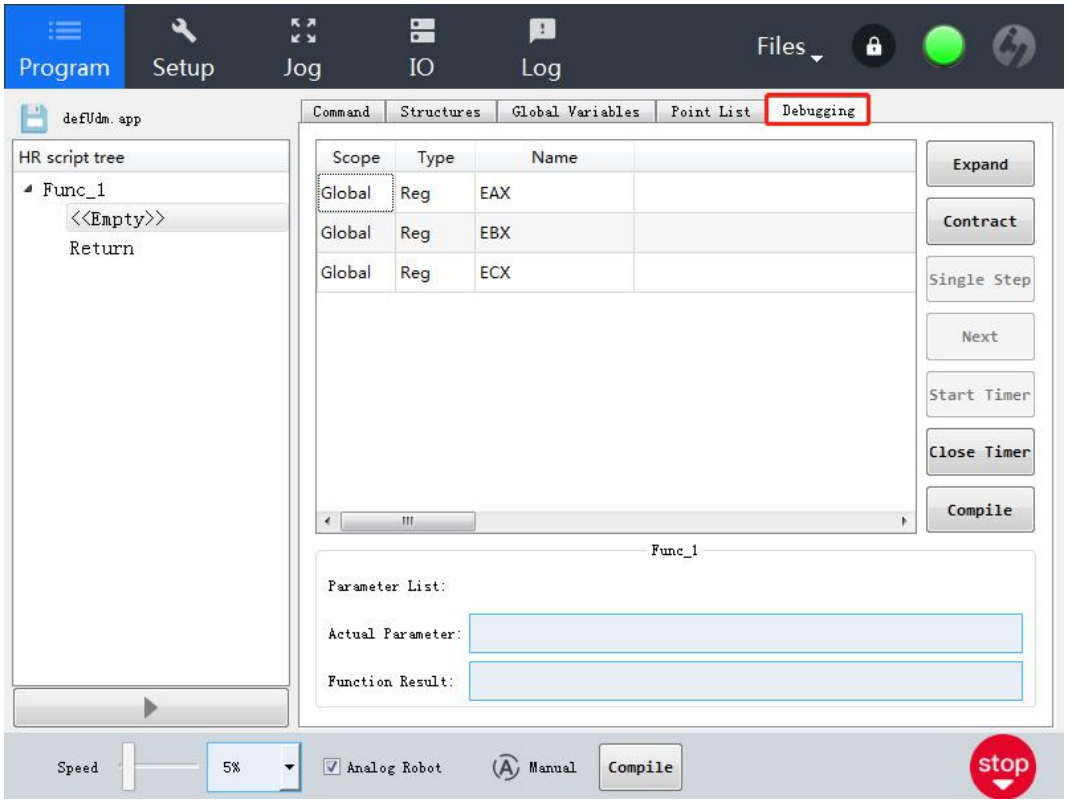
Delete Point:delet points in the Point List.

Teach Point:teach the selected point in the Point List.

(1)Teaching method 1: Manual teaching. The user can select the articulation button of the page, the space motion button, or enter the target point position in the data field to teach the point.

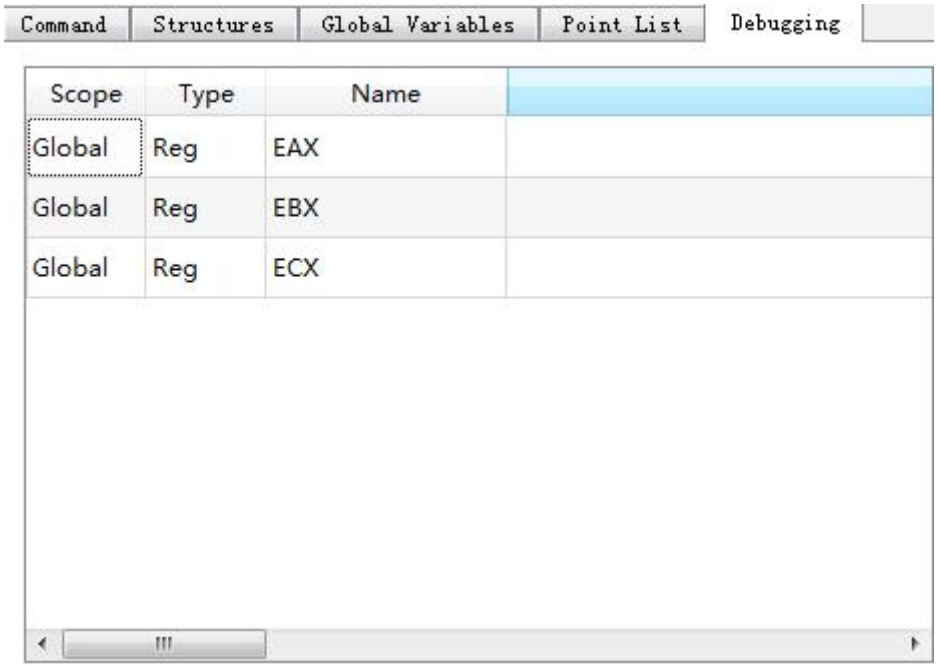
(2)Teaching method 2: Freedrive teaching. Click the “Start Freedrive” button to enter the freedrive teaching operation interface. The user can manually drag and drop the robot to make the robot reach the target position, and then teach the record to save the points.

Debugging



The meaning of each area and button in the debugging interface is as follows:

(1)Debugging list: As shown in the following figure, this list shows the values of global variables and local variables during the running of the function. Global variables contain registers and variables defined in the global variable list.



(2)Expand: Expand the script tree so that each command line in the program is displayed, which is convenient for viewing the program.

(3)Contract: Fold the program tree so that only the function name is displayed in the program, and the command line is hidden.

(3)Single Step:Used for single-step running programs, generally used for debugging.

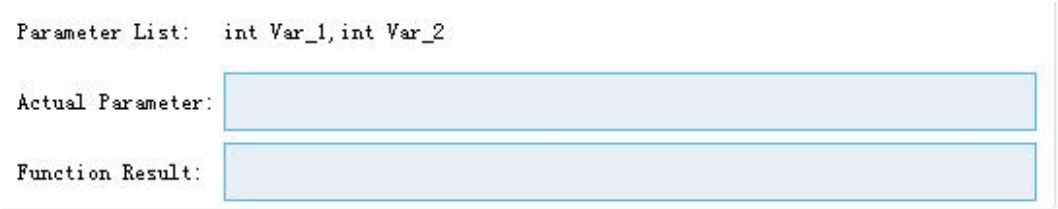
(4)Next:When single-step programs are running, jump to the next command line in the program.

(5)Start Timer:When the timer independent thread is started, and the commands in Command and Function in Timer are running at the same time.(Note: According to the time sequence of creating timers, the current timer can be turned on at most 5 [1~5], and the 5th timer cannot be suspended.)

(6)Close Timer:Close timer thread.

(7)Compile:Compile the current program script. If the compilation is successful, the program can be run. If the compilation fails, the reason for the failure will be displayed.

(8)Fun_1:As shown below



As shown in the figure above, the function Func_1 selected in the current program tree has the parameters Var_1, Var_2. When the user wants to run this function, the actual parameters need to be input. For the parameter type and format, refer to the parameter list. The above parameters are int variables. Note: This parameter is valid only for the function Func_1 and belongs to the local variable; the function returns the result after the function is executed.

(9)Run the robot:

Analog Robot:The robot control mode will be changed to the analog state, and the actual robot body will no longer move. When the user operates the software interface, the data value of the software interface will undergo an analog change.

The manual:the robot is in a non-script state, and the user can perform programming, free driving, IO control, etc..

Auto:The robot is in the scripting stage, the user can only run the program, other operations cannot be performed.

Run:When in "Auto" mode, after selecting a function, click on the "Run" button and the function will start running.

Pause:While the program is running, click the “Pause” button to pause the current motion; for the timer, the pause button is valid for the 1st to 4th timers, and the 5th timer is disabled.(Note: The wait command in the timer cannot be paused. The 1~5 timer ordering is relative to the time sequence in which the timer is created.)

Continue:After the program is paused, click the "Continue" button and the program will be executed on the suspended command line.

The cyclic:the program will run the selected function in a loop.

Speed:Adjust the speed of the robot movement.

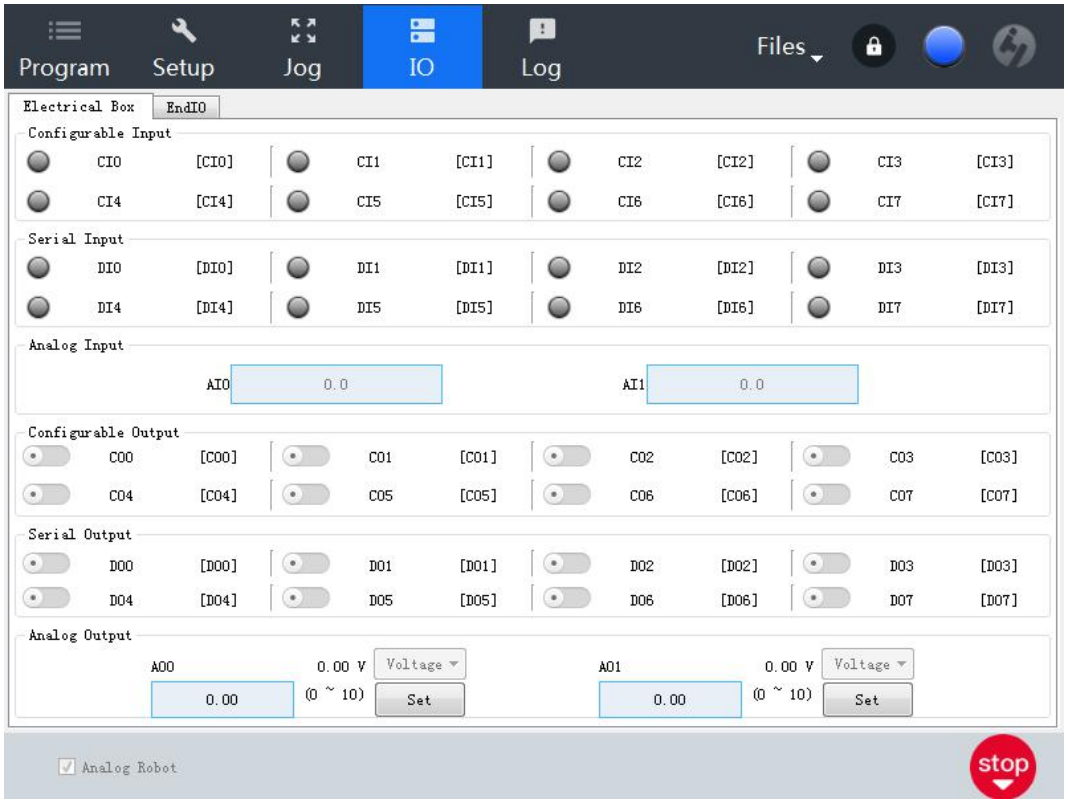
Stop:Stop the program, including the stop function and timer, and exit the script mode to switch to manual mode.

1.8

1.9 IO

In this interface, you can monitor and set the real-time I/O signals of the electrical box. The interface displays the current status of the I/O, including the I/O status during program execution. The program will stop running if there are any changes during the program run. All output signals will remain in their state when the program stops running. The screen is updated at 10 Hz, so signals that change very fast may not display correctly.

IO of Electrical Box

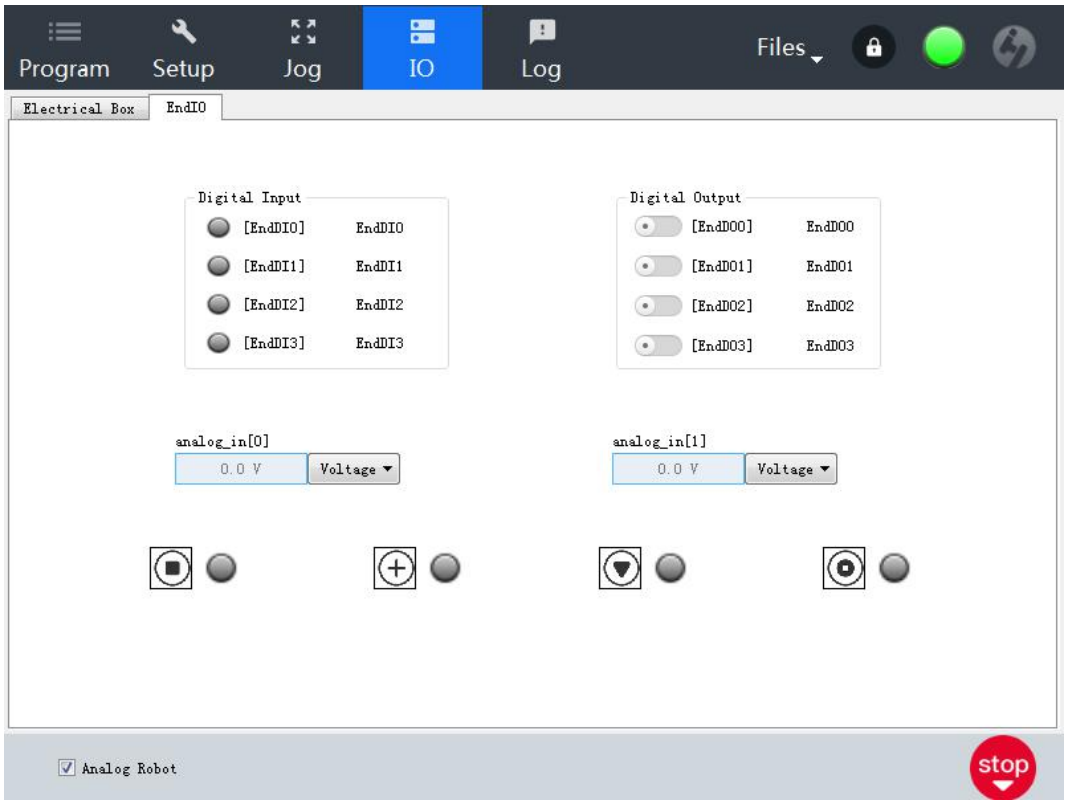


Configurable IO:Configurable input CI0~CI7, if the serial port input is high level, the corresponding input serial port indicator will be bright; configurable output CO0~CO7, select the corresponding output serial port, then the output level can be set high, corresponding output serial port indicator will be bright.

Serial IO: Serial port input DI0~DI7, if the serial port input is high level, the corresponding input serial port indicator light becomes bright; the serial port outputs DO0~DO7, select the corresponding output serial port, the serial port output level is set high, and the corresponding output serial port indicator lights up.

Analog IO:2 channels, analog current range: [4-20mA], analog voltage range: [0-10V].

End IO



Digital IO: The digital input EndDI0~EndDI3 corresponds to the IO input signal at the end of the robot. When the input is high, the corresponding digital indicator lights up; the digital output EndDO0~EndDO3 corresponds to the IO output signal at the end of the robot, and the output is high level. Digital indicator lights up.

Analog input:2 channels, analog voltage range: [0-10V].

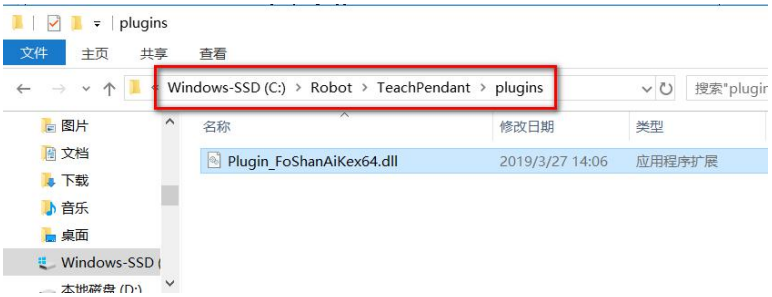
Function button status:Corresponds to the status of the new end IO button.

1.10 Plugins

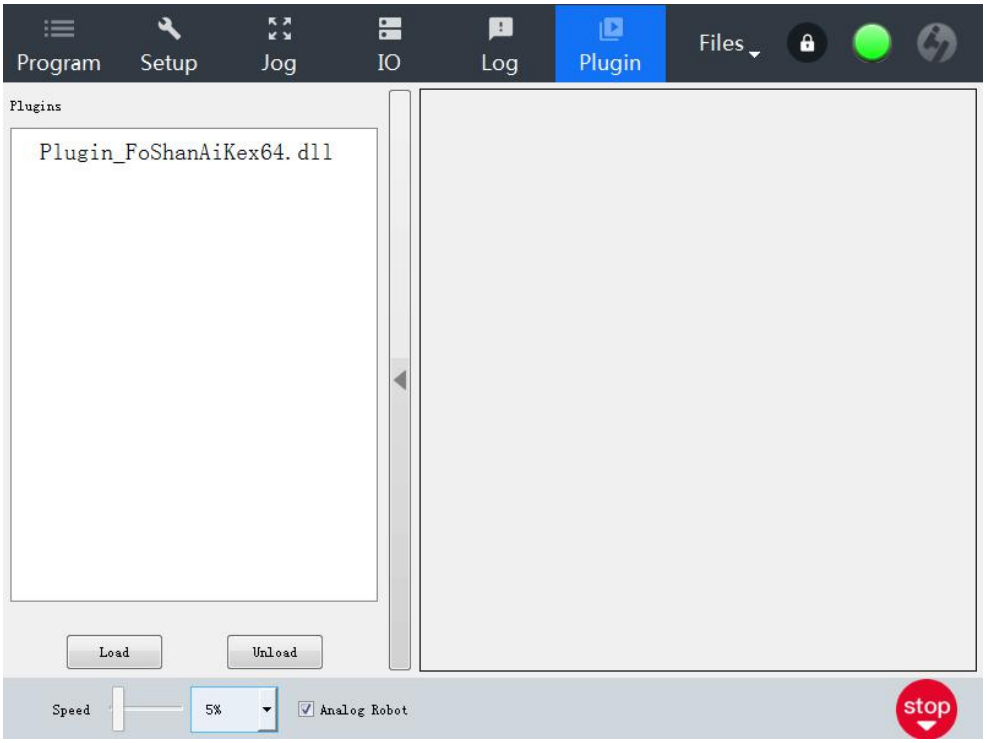
At present, many customers in the market want to add one or two pages to the interface of the teach pendant, so that the teach pendant can realize the function of the host computer. Based on this, Elfin software provides a plug-in page---"plug-in". For some simple applications, the client-specific process interface can be mounted into the framework of the teach pendant through a plug-in. After the setup is complete, you can load the plug-in interface written by the client to the “Run” interface after restarting the software.

Operation steps:

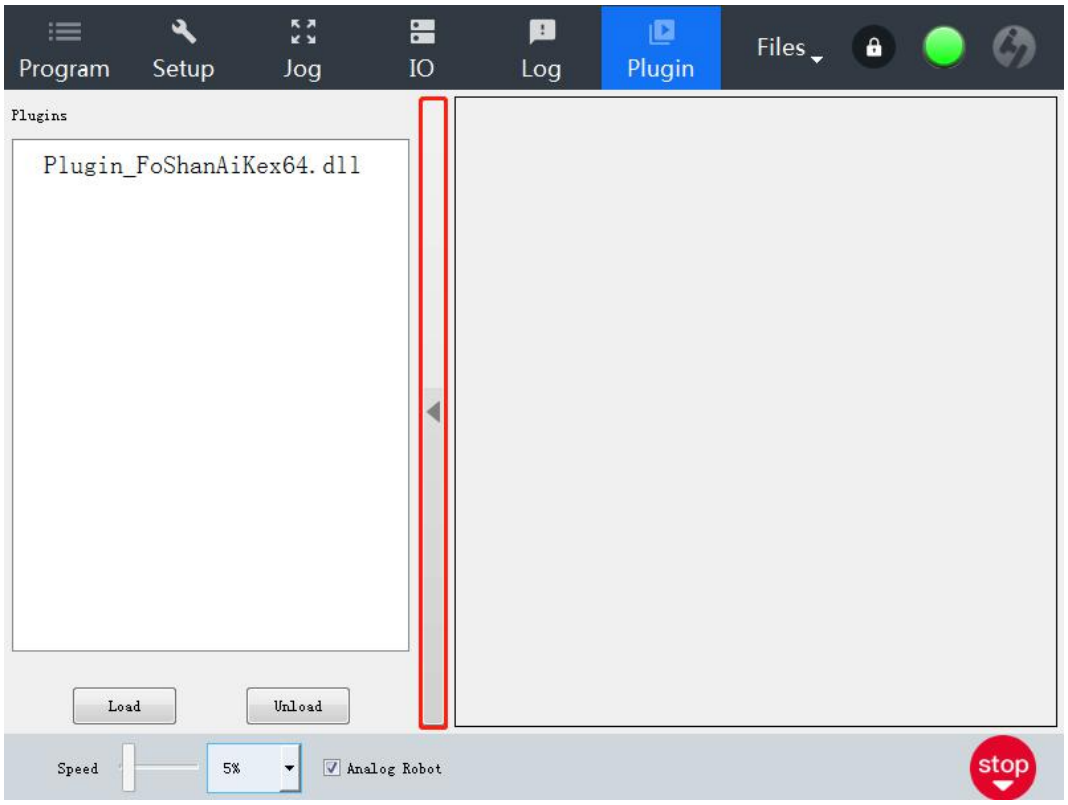
1)After the process plugin is written, the corresponding version of the dll generated by the plugin is placed in C:\Robot\TeachPendant\plugins;

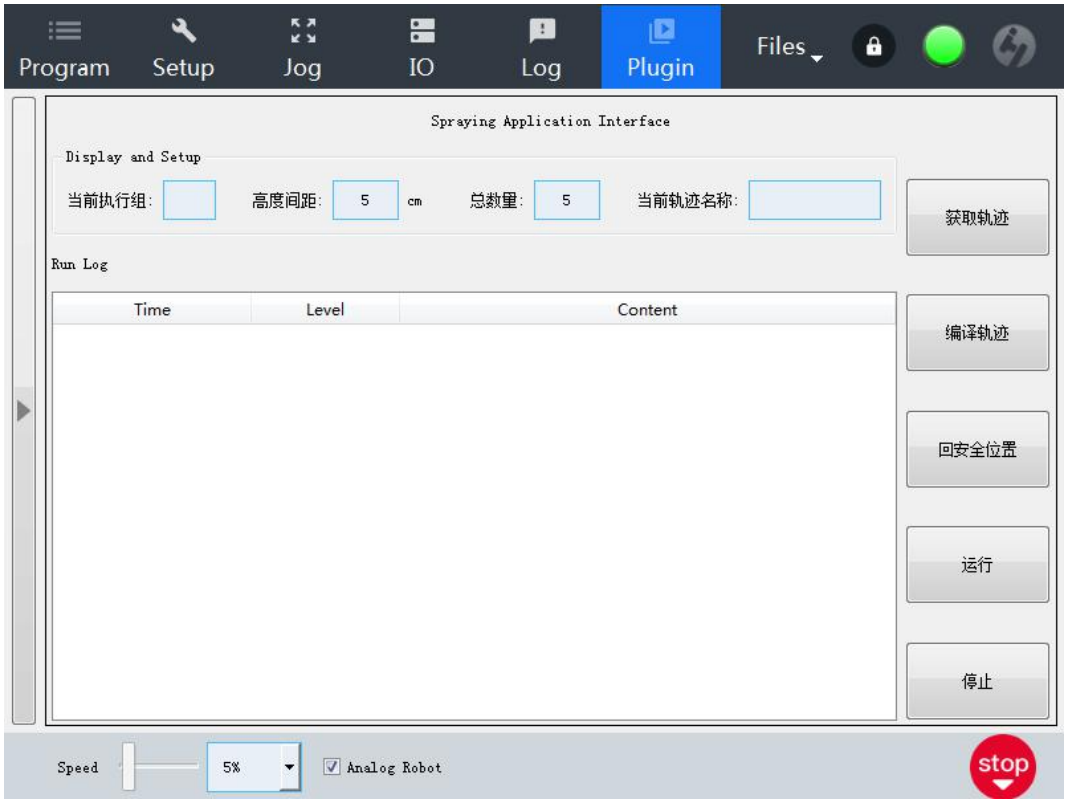


2)Start the teach pendant, add the plugin option to the function module;



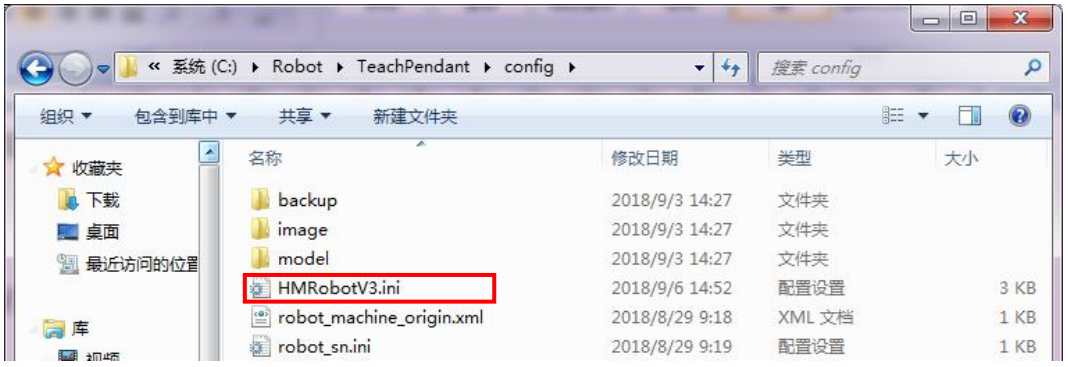
3)Select the dll file and click Load to load the plugin into the interface. Click the red button to hide the loading interface.

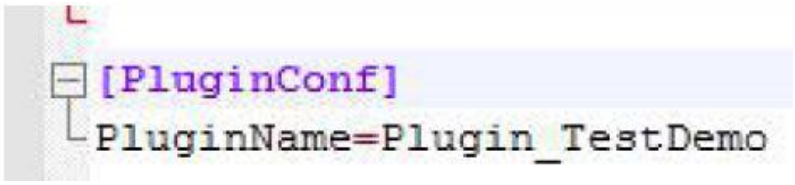




4)If want to automatically load the dll every time when starting the teach pendant, then in the execution directory of the teach pendant V3.0, modify the "[PluginConf]" option in HMRobotV3.ini and change its "PluginName" to the name of the plugin.

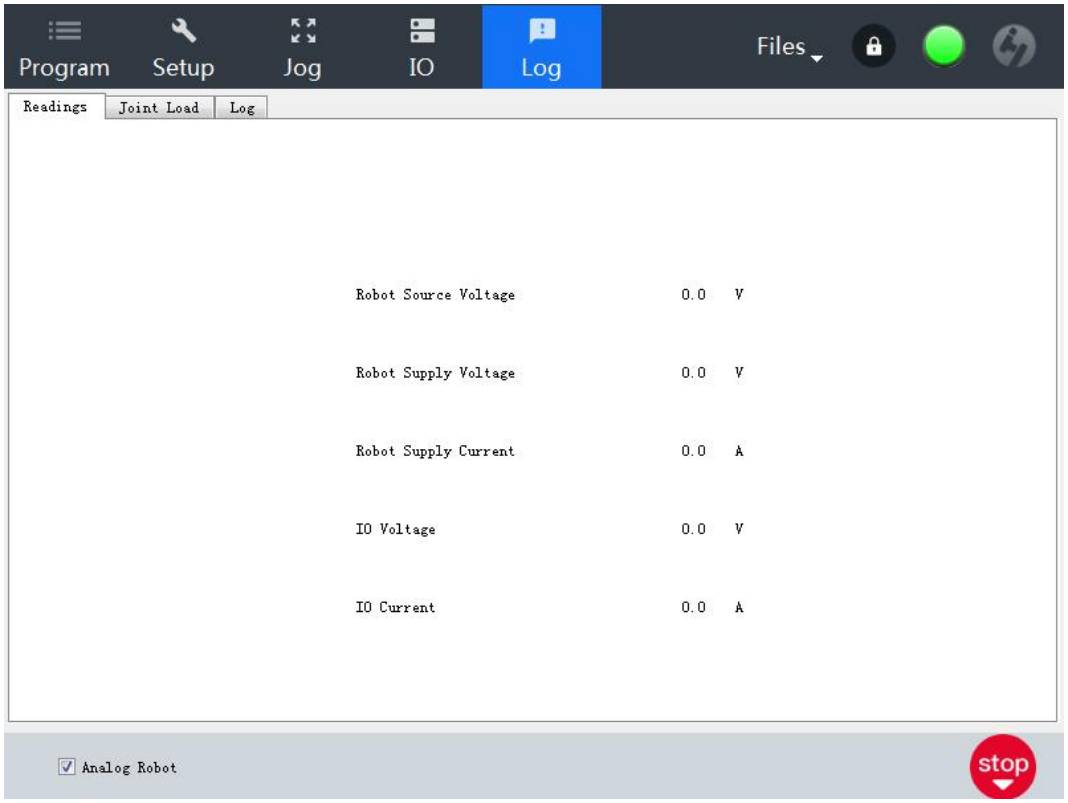
Note: The plugin name does not contain the suffix "x64", "d", etc., just the plugin name, as shown below:





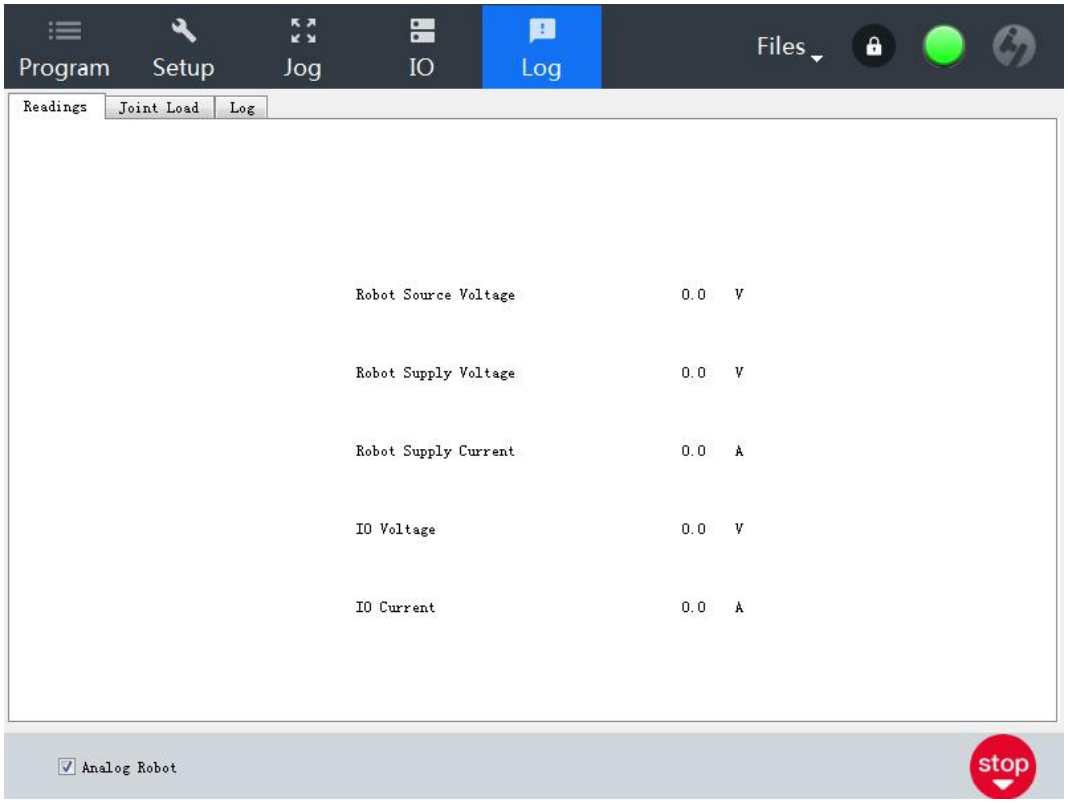
1.11 Log

This section shows the robot arm and control box status and log messages.



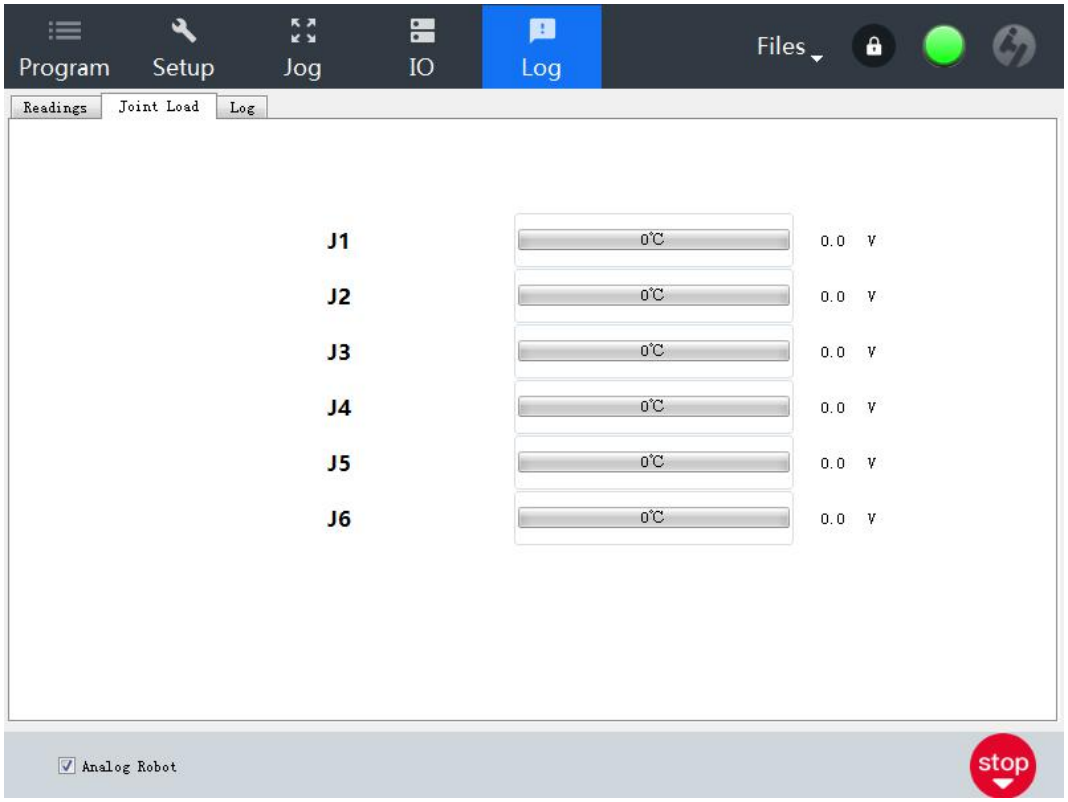
Readings

Real-time display of the power supply of the robot's electric control box.

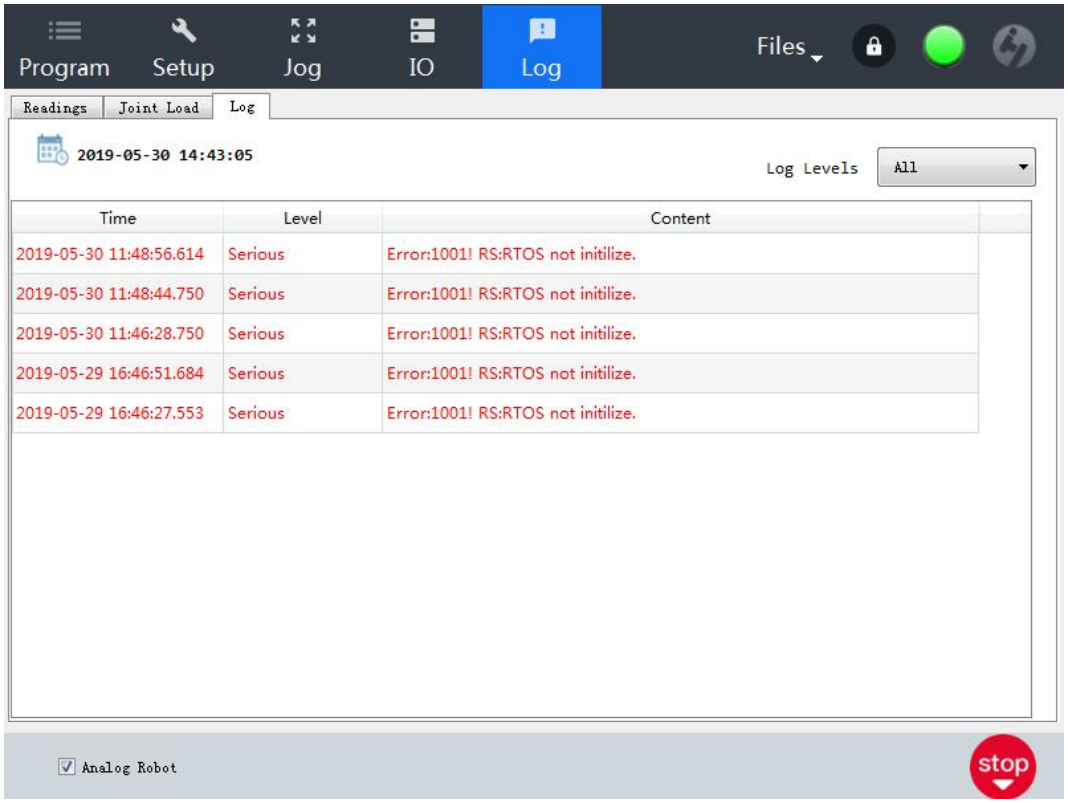


Joint Load

Display information about each joint of the robot. The information about each joint of the robot includes the temperature of the motor and electronics, and the voltage at the joint.



Log



The first column will log the arrival time of the message. The second column shows the severity classification of the message. The next column shows the specific message content. Message content contains:

(1)Command sender:client

(2)Command sent:cmd

(3)Error code:For example, Error: 1035 or err: 1041. For details of the error code, please refer to the communication protocol file.

Error explanation: for example RS: DCS Status Error

Log Level:This button filters messages by log severity level.

1.12 DCS Interface

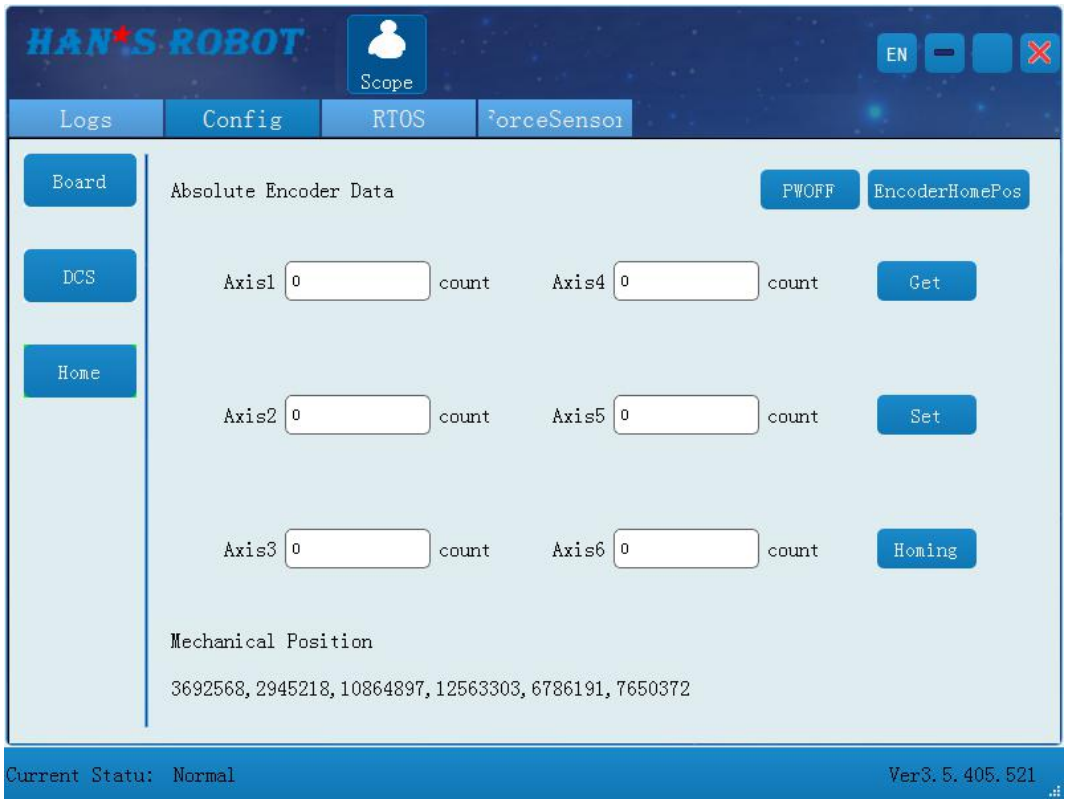
Double-click the DCS icon at the bottom right of the screen to open the DCS interface.

Mechanical Position

As shown in the figure below, it is the "Mechanical Position" interface in the DCS ;

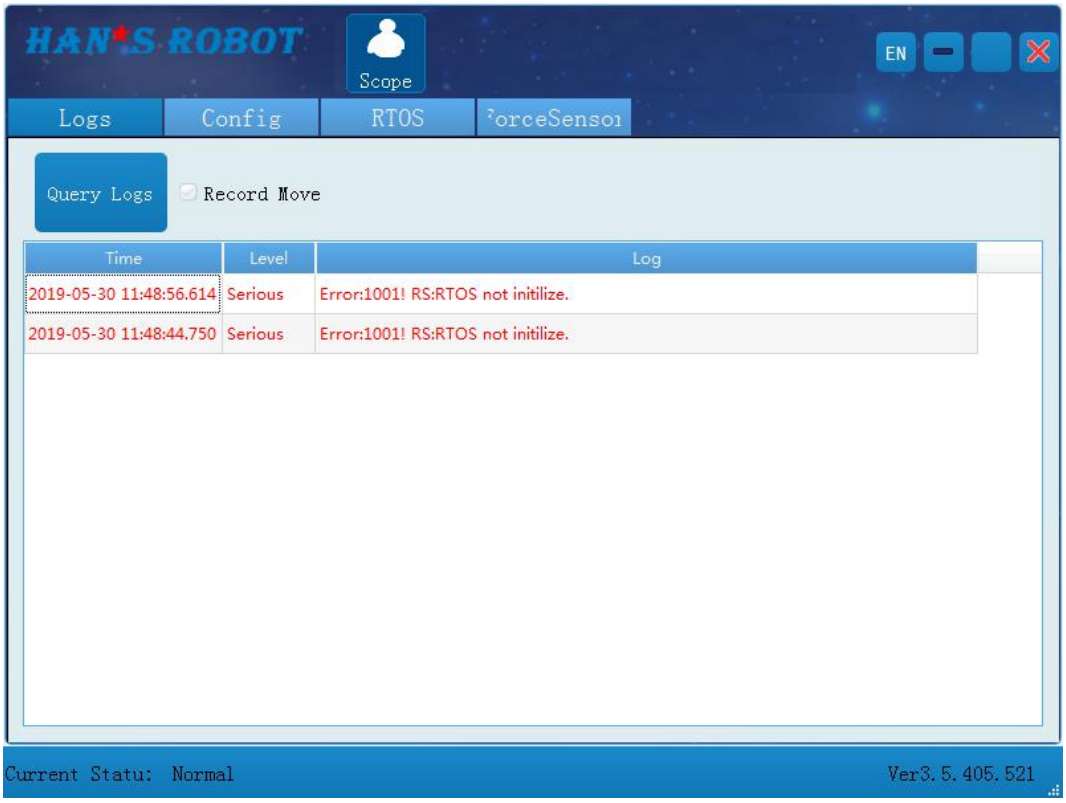
Generally, the users do not need to operate the interface, the only situation where the interface needs to be used: mechanical position lost.

If the mechanical position is lost, please contact our technical engineer and we will help to solve as soon as possible.



Logs

In this interface, the user can check the error status of the robot in real time; you can check the status of the robot movement by selecting the option of “Record Move”.



Chapter 2 ROS Control

For standard products (Elfin body, electric control box, teach pendant), the robot is controlled based on Windows system. However, Elfin also provides the ROS platform control interface. If you want to use the ROS platform to control the robot, please do the following:

Software Preparation:

1. Download the Ubuntu system image from this page: http://releases.ubuntu.com/14.04/, Select 64bit or 32bit under desktop image, 64bit is recommended.

2. https://jingyan.baidu.com/article/76a7e409bea83efc3b6e1507.html

Follow this tutorial for dual system installation.

3. http://wiki.ros.org/indigo/Installation/Ubuntu

Install ROS according to this tutorial and recommend Desktop-Full install

Hardware Connection:

1. Power the robot with an electric cabinet:

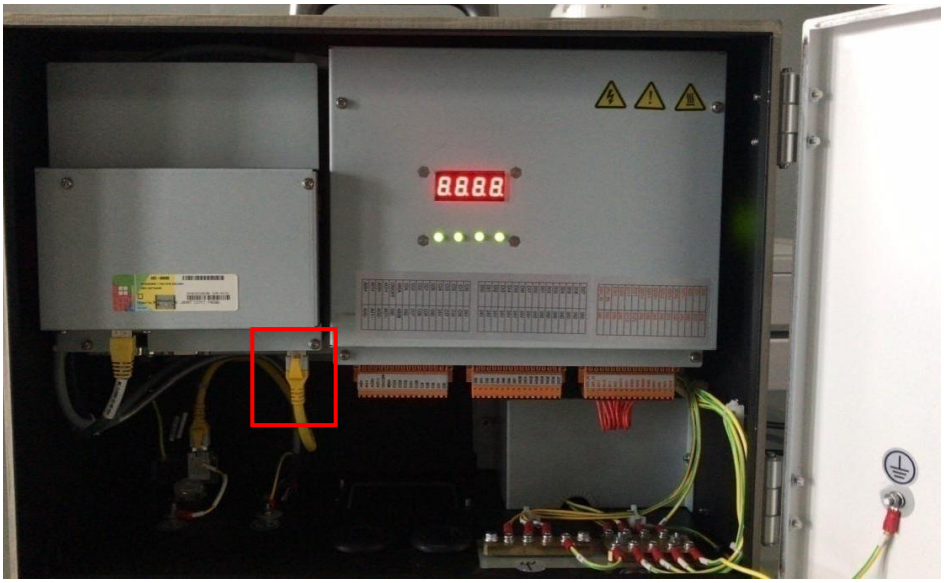
1) Hardware preparation

A network cable, a computer

2) Operation procedures

①Open the door of the electric cabinet (each electric cabinet has corresponding key when shipping);

②Connect the network cable: unplug the yellow port of the red box below, and connect the external network line to the network line with a network port extension module;



③Connect the other end of the external network cable to the network port of the computer;

④Press the power button on the teach pendant to power the robot.

2. Use external power to power the robot

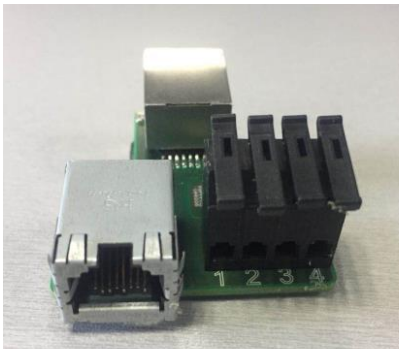
When you only buy our Elfin body or do not want to use the electric control box to power the robot, you can use the external power supply to power the robot.

1) Hardware preparation:

External power supply, network cable, computer, body/control box adapter

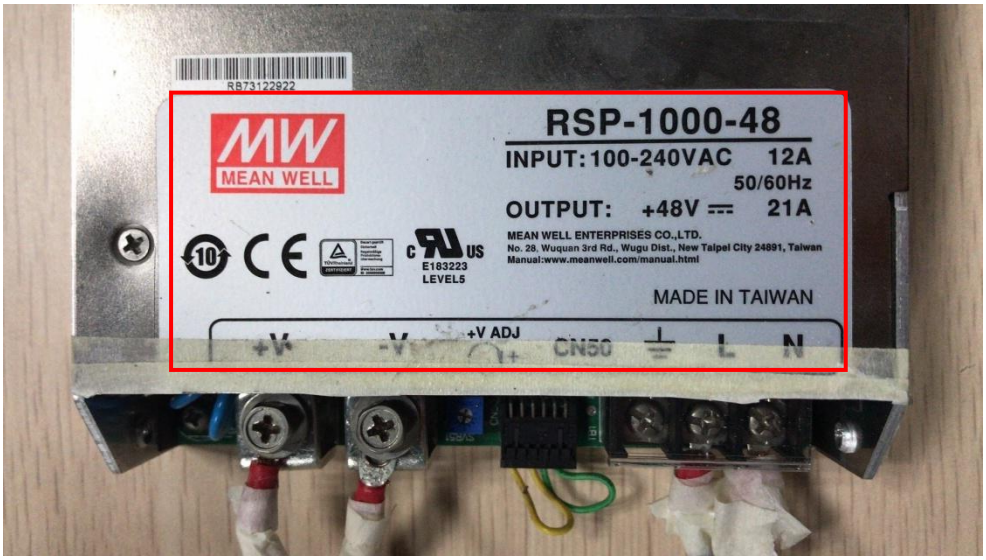
Note:

①"Body/control box adapter" is a product developed by our company, which needs to be purchased from our company for use.



②The normal voltage of the Elfin body is 42V~54V, and the peak current cannot be lower than 19A.

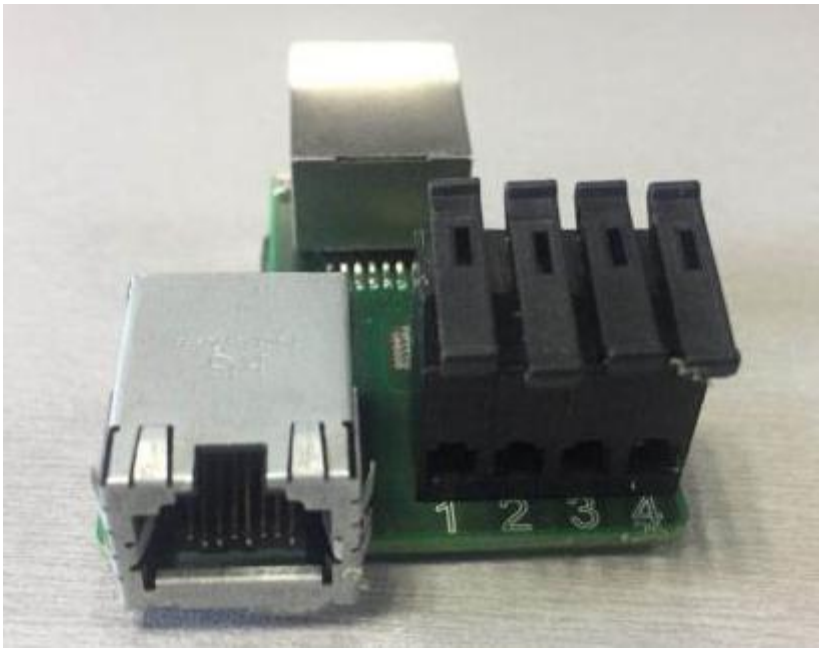
The external power supply illustrated here is as follows:



2) Operation procedures

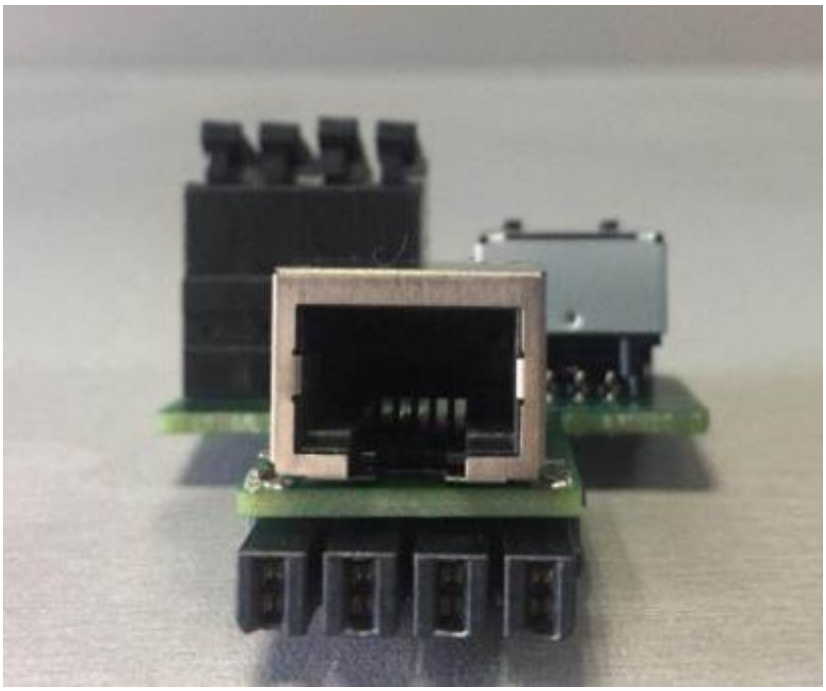
(1) Hardware connection

1) Power supply wiring



The front end of the adapter is shown in the figure above. The left side of this end

USES the network cable to connect the network port of the computer and plug it in. There are 1234 four ports on the right side, corresponding digital code is written on the interface, where 123 corresponds to 1.48V+ 2.48V- 3. GND.The fourth port is not processed, and it can be fastened after being connected.



The rear end of the adapter is shown in the figure above. The end can be connected with the Elfin body and buckled.

(2) Power on the external power supply, that is, power on the robot.

3. ROS operation：

•Tutorials on the ROS official website: http://wiki.ros.org/ROS/Tutorials

•Chinese tutorials can also be found from baidu, such as the following tutorial is good

https://www.ncnynl.com/category/ros-junior-tutorial/1/

•Instructional website for ROS control Elfin

github.com/hans-robot/elfin_robot

Chapter 3 Shutdown Procedures

1. Save the program: If there is a program being edited before shutting down, please save the program first;

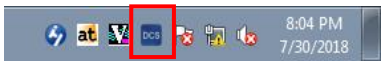
2. PWOFF: Move the robot to a relatively safe position, then switch to the DCS interface, click the “PWOFF” button in the upper right corner to disable the robot;

3. OFF: switch to the "initialization" interface in the indicator, and click "OFF" button to power off the robot;

4. Close the software:

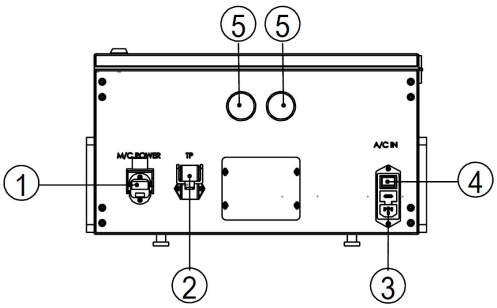
1) Turn off the teach pendant: Click on "File" and select the "Exit" option in its drop-down menu to close the controller;

2) Close DCS: Right click the "DCS" icon in the lower right corner and select "Exit".



5. Close Windows system:

6. Turn off the electric control box power supply: after the system is completely shut down, disconnect the electric control box power supply (press the button of control box power supply as shown below②);



Chapter 4 Quality Warranty

1.13 Product Quality Warranty

In the principle of no prejudice to any claim agreement that may be reached between users (customers) and distributors or retailers, the manufacturer shall give customers a product quality warranty according to the following terms:

If any defect occurs due to defective manufacturing or materials within 12 months after new equipment and its components are put into operation (not more than 15 months if transportation time is included), Han's Robot shall provide necessary spare components while users (customers) shall provide labor for replacement with spare components. Related components shall be maintained or replaced with another component embodying the up-to-date technological level. This product quality warranty is invalid provided that equipment defects are caused by improper handling or failure to observe related information described in the user manual. This product quality warranty does not apply to or extend to any maintenance performed by authorized distributors or customers such as installation and software downloading. Users (Customers) must provide a purchase receipt and purchase date as valid evidence of enjoying the product quality warranty. According to this product quality warranty, any claim must be made within two months when the product quality warranty is not obviously fulfilled. Any equipment or components replaced or returned to Han's Robot shall be owned by Han's Robot. Any other claim arising from or in connection with equipment is not within the scope of this product quality warranty. Any terms of this product quality warranty do not try to limit or exclude customers' legal rights as well as the manufacturer's liability for any casualties due to its negligence. The duration of this product quality warranty shall not be extended due to any services provided according to the terms of this product quality warranty. Han's Robot reserves the right to collect replacement or maintenance costs to customers without violating the principles of this product quality warranty. The preceding regulations do not imply any change of burden of proof, harming the interests of customers.

If equipment shows any defect, Han's Robot shall not bear any resulting damage or loss, e.g. production loss or damage to other production equipment.

1.14 Disclaimer

Han's Robot reserves the right to upgrade products without prior notice because it devotes itself to continual improvement on product reliability and performance. Han's Robot does its best to ensure the accuracy and reliability of the contents of this manual, but disclaim any liability for any error or missing information.

Chapter 5 Appendix

1.15 Technical Specifications

Name	Parameter
Weight	40kg
Payload	10 kg
Reach	1000 mm
Joint range	+/-360°
Joint speed	90°/s
Tool	1 m/s
Repeatability	+/- 0.05 mm
Installation area	Φ 180mm
Degree of freedom	6 rotating joints
Control box dimensions	445.2*236.5*536mm
I/O ports	Digital input 4
	Digital output 4
	Analog input 2
I/O power supply	Electric cabinet: 24V 2A Terminal IO: 24V 1.5A
Communication	TCP/IP
Programming	On-screen manipulation director; remote access
IP class	IP54
Power	About 350 W
Collaboration	10 advanced security configuration functions
Main material	Aluminum alloy
Operating ambient temperature	0-50℃
External power input	200-240 VAC, 50-60 Hz
Cables	Control box connecting cable: 5 m
	Demonstrator connecting cable: 5 m

1.16 Limiting Security-Related Functions

Limiting Safety Function	Description
Joint position	Min. and Max. joint angular positions
Joint speed	Max. joint angular speed
TCP position	Plane where the TCP position of the robot is limited in Cartesian space
TCP speed	Max. speed of robot TCP
TCP force	Max. thrust of robot TCP

1.17 Electrical Safety Specifications

Safety Input	Description
Robot emergency stop	Execute Class 1 stop* and use system emergency stop to notify the body
Emergency stop button	Execute Class 1 stop* and use system emergency stop to notify the body
System emergency stop	Execute Class 1 stop*
Protective stop	Execute Class 2 stop*
Emergency stop reset	Restore the robot from protective stop state

*** Term explanation:**

- Class 0 stop: The robot stops operation immediately after the robot power supply is cut off. This is an uncontrolled stop. The robot may deviate from a route set in the program because every joint is braked at the fastest speed.

- This protective stop can be used only when a safety evaluation limit is exceeded or when an error occurs in the safety evaluation part of the control system.

- Class 1 stop: The robot stops when you stop it for power supply to it. The power supply is cut off after the robot stops. This is a controlled stop. The robot will follow a route compiled in the program. The power supply is cut off after 1s or once the robot keeps its feet.

- Class 2 stop: Controlled stop during power-on of the robot. The robot stops all actions within 1s. The control of the safety evaluation control system enables the robot to stay in the stop position.

1.18 Robot Status Box

	Robot Status	1	0
	Moving	Moving	
	Homing	Homing	
	ErrStop	Error in the movement process	
	StandBy	Be ready and wait for an instruction	Moving
	Stopping	Stop state	
	Disable	Disabled	Enabled
	Power	Enabled	Disabled
	Error	Error state	After resetting and clearing
	HandType	Left hand type	Right hand type
	Ready	The startup of the controller is finished	
	SafeSpace	Exceed safe space	Not exceed safe space
	MovSpace	Standby	

1.19 Elfin Function Return Error Code Table

Error Code	State Description	English Display	Solution
0	Normal	RS:Operatorsuccessed.	
1001	The robot is not initialized	RS:RTOS not init.	Click the "RTOS" icon to start the RTOS.
1002	The master is not started	RS:Master status is closed.	Click Start master to start the master.
1003	The slave is disconnected	RS:Slave dropped.	Click Start master to restart the master.
1004	The robot is safely locked	RS:Robot on safe locked.	Set the serial port Input17 to low.
1005	Physical emergency stop	RS:Robot on Emergency stop.	Release the emergency stop button and click the Power on button of the software to power on again.
1006	The robot is disabled	RS:Robot not ServoOn	Click the Servo On button to enable the robot.
1007	Slave error	RS:Slave on error	(1) Shut down the master; (2) Start the master.
1008	The robot exceeds safe space	RS:Robot out safe space.	(1) In enabled state, let the axis move back to safe space in the reverse direction of the axis that exceeds safe space due to long jog movement. Click Reset to perform resetting and clearing; (2) In disabled state, Click Reset to perform clearing. Click Servo On. Let the

Error Code	State Description	English Display	Solution
			axis move back to safe space in the reverse direction of the axis that exceeds safe space due to long jog movement.
1009	The robot is moving	RS:Robot on Moving.	Prompting error code: Movement instructions are received in the movement process.
1010	Commands are invalid	RS:Command Invalid.	Prompting error code: The user sends invalid commands. Please check details of commands.
1011	Parameter error	RS:Parameter error.	Check and correct parameters set in the program
1012	Function call format error	RS:Function call format error.	Check and correct the format of called functions.
1013	Commands wait to be executed	RS:Command waiting execute.	Prompting error code: The DCS is in standby state and cannot read any movement state.
1014	I/O does not exist	RS:IO not exist.	Prompting error code: No digital I/O module set in a corresponding program is available in the hardware. Check and modify it to an I/O number corresponding to the hardware.

Error Code	State Description	English Display	Solution
1015	The robot does not exist	RS:Robot not exist.	Prompting error code: No corresponding robot is available. Check that the software is opened normally.
1016	No DCS server is connected	RS:Not connected to server.	Check whether the DCS interface is started. Click the Login button of the software to reconnect the DCS.
1017	Communicatio n timeout	RS:Communication timeout.	Check the network. Re-click the Login button to reconnect the network.
1018	Connection fails	RS:Connected failed.	Check whether the DCS interface is started. Click the Login button of the software to reconnect the DCS.
1019	Serial port connection fails	RS:Communication Serial Failed.	Check whether the settings of the serial port module and the selection of the Com port are correct.
1020	Home position is not set	RS:Home position not set.	Set correct home data.
1021	The last command is not finished	RS:Last command not finished.	Wait until the execution of the command is finished before sending an instruction.
1022	The serial port DI is empty	RS:Serial DI is NULL.	No corresponding serial port I/O is available in the external hardware.

Error Code	State Description	English Display	Solution
1023	The serial port DO is empty	RS:Serial DO is NULL.	No corresponding serial port I/O is available in the external hardware.
1024	Waiting timeout	RS:Wait Timeout.	Prompting error code: The execution of a command times out. Re-execute this command.
1025	Error state	RS:Robot On Fault.	(1) Prompting error code: The robot is in error state and cannot execute any command; (2) Click Reset to perform clearing; (3) Click the Servo On button to re-enable the robot.
1026	The robot is stopping	RS:Robot is Stopping.	Prompting error code: The robot is stopping.
1027	The robot is being disabled	RS:Robot is Disabling.	Prompting error code: The robot is being disabled.
1028	The robot is being enabled	RS:Robot is Enabling.	Prompting error code: The robot is being enabled.
1029	The function is not enabled	RS:Option is not Enabled.	Prompting error code: It indicates that this function is not enabled.
1030	Master startup times out	RS:Robot start master time out!	Prompting error code: Master startup times out. (1) Click Shut down

Error Code	State Description	English Display	Solution
			master; (2) Click Start master.
1031	The robot is not powered on	RS:Robot not electrify!	Click Power on to power on the robot.
1032	The serial port is not started	RS:Serial not Enabled!	Open the serial port settings in the DCS interface.
1033	The simulation state command is invalid	RS:Beening simulation!Command Invalid!	Prompting error code: This command is invalid in simulate state.
1034	The RTOS library does not exist	RS:RTOS Library not exist!	(1) Prompting error: The RTOS is not started; (2) Double-click the "RTOS" icon to start the RTOS.
1035	The DCS processing command thread crashes	RS:DCS Handle Command thread crash!	Restart the DCS software.
1036	Option is disabled	RS:Option forbidden!	Error prompt: It indicates that this operation is disabled at this moment.
1037	Emergency stop state is not cleared	RS:Emergency Stop Not clear!	Release the E-stop button and power on using the controller.
1038	Safe stop I/O is not cleared	RS:Safe Stop Not clear!	Move the object away and click Reset to perform clearing.
1039	The script is running	RS:Script running!	Prompting error code: Commands are invalid when a script is running.

Error Code	State Description	English Display	Solution
1040	xml parameter error	RS:Xml Param Error!	Prompting error: xml file format error
1041	The system board is not connected	RS:System Board Not Connect!	Open the system board on the DCS interface. Set a correct serial port module and COM port. After they are set to defaults, open it.
1042	The controller is not started	RS:Controller Not Start!	Start the controller.
1043	Controller state error	RS:Controller Statu Error!	Prompting error code: Any issued command cannot be executed in this state.
1044	The robot is in effortless demonstration mode	RS:Robot in TeachMode!	Prompting error code: The robot is in effortless demonstration mode and commands are invalid. To execute commands normally, disable the effortless demonstration function.
1045	The robot has been powered on	RS:Robot Already Electrify!	Prompting message: It indicates that the system has been powered on and it is unnecessary to power on again.
1046	Connecting the ModBus fails	EC: Connect to Modbus Failed!	Check whether the ModBus communication settings are correct. Check whether communication hardware is normal.

Error Code	State Description	English Display	Solution
1047	The master has been started	EC: Master is Started!	Prompting message: It indicates that the master has been started.
1048	Load parameters exceed limits	EC:Parameter over specified payload!	Check whether load set values exceed system limits. For example, The load limit of E5 is 5 kg.
1049	DCS state error	EC: DCS Status Error!	Restart the DCS.
1050	The target position is invalid	EC: Target position invalid!	Check whether the target position exceeds safe space.
2000	Library loading fails.	RS:Load library failed.	Check whether the library file format is correct. Reload the library.
2001	The script is empty	RS:Script is empty.	Prompting error: The contents of the script is empty.
2002	Compilation error	RS:Compile error.	Prompting error: Check whether program instructions are incorrect.
2003	Script reloading error	RS:Reload script error.	Check whether the loaded text format is correct.
2004	The function does not exist	RS:Function not exist.	(1) The function instruction does not exist; (2) Check whether the sent command is incorrect.
2005	Function return type error	RS:Function return type is error.	(1) Function return value type error; (2) Check that the

Error Code	State Description	English Display	Solution
			function return value is consistent with the definition.
2006	MissSignal1	RS:MissSignal.	This function is not enabled.
2007	MissSignal2	RS:MissSignal.	
2008	Parameter type error	RS:Function param type is error	Check that the function definition parameter is consistent with the incoming parameter type and quantity.
2009	No header file is contained	RS:Can't not find head file.	The tcc is provided with a header file. Version V2.5 does not use this error code.
2010	No return value is available	RS:Function do not have return value.	Add a function return value.
2011	Parameter count error	RS:Function param count is error	Check whether the entered number of parameters is consistent with the set number of parameters.
2012	UDM stack error	RS:UDM Stack Err..	Test error code used by developers
2013	The script is locked and may be being compiled.	RS:Script been lock,maybe compiling.	Wait until the compilation is finished.
2014	Not in RunScript state	RS:Not In RunScript Statu.	Prompting error code: The DCS is not in RunScript state. This cannot be solved by running the script. Change the program

Error Code	State Description	English Display	Solution
			mode to AUTO on the debugging interface of the demonstrator script programming.
2015	The serial port is not connected	RS:Serial Close.	Switch to the system setting subinterface of the DCS interface and open the serial port connection.
2016	The controller is not started	RS:Controller not started.	Start the controller normally.
2017	The socket is not connected	RS:Socket Not Connected.	Check whether the socket communication settings are correct.
2018	The function name cannot have any space	REC_FuncNamehave Space	Delete the space from the function name.
2020	The function is interrupted	RS:Function broken stop.	Prompting error code: It indicates that the Stop button is clicked in the running process of the program.
2021	Socket error	RS:Socket Error.	Check the socket in the program is selected correctly.

1.20 Elfin Controller Error Processing

Description	Error Display	Solution
Normal	Normal	
Short circuit error	Error: Short circuit error!	Click Reset to perform resetting and clearing. Click the Servo On button to enable the robot.
Overvoltage error	Error: Over voltage limit error!	
Undervoltage error	Error: Under voltage limit error!	
Overspeed error	Error: Over velocity limit error!	
Execution error	Error: Execute error!	
RMS overcurrent error	Error: Over current error!	
Encoder error	Error: Encoder error!	
Position following error	Error: Following position error!	
Speed following error	Error: Following velocity error!	
Negative limit error	Error: Negative limit error!	
Positive limit error	Error: Positive limit error!	
Servo overtemperature error	Error: Server over heating error!	
Peak	Error: Max current error!	

Description	Error Display	Solution
current error		
Hardware braking error	Error: Emergency stop error!	
UDM error	Error: UDM error!	
Servo parameter error	Error: Server parameter error!	
Enablement timeout	Error: Robot enable time out!	Check whether the robot body drive UDM operates normally. Click the Servo On button to re-enable the robot.
SDK collision detection	Error: Robot Collide with body!	(1) Manually open the brake using the demonstrator; (2) Manually drag the robot so that it leaves the self-collision posture; (3) Close the brake and perform clearing and enablement.
Joint limit	Error: Over joint limit error!	(1) In enabled state, let the axis move back to safe space in the reverse direction of the axis that exceeds safe space due to long jog movement. Click Reset to perform resetting and clearing; (2) In disabled state, Click Reset to perform clearing. Click Servo On. Let the axis move back to safe space in the reverse direction of the axis that exceeds safe space due to long jog movement.
Singularity	Error: Singularity error!	Click Reset to perform clearing. Click Servo On to enable

Description	Error Display	Solution
		the robot.
Abnormal stop	Error: General stopping criterion	Click Reset to perform clearing.
SDK calculation error	Error: calculate failed	Click Reset to perform clearing.
UDM state error	Error: UDM Statu Error!	Shut down the master; Start the master.
Slave error	Error: Init slave Error!	Click Reset to perform clearing.
HomeStep2 error	Error: Home Step2Error!	Click Reset to perform clearing.