



# Collaborative robots

## Basic training

Software Version V3.0.0

---

START

## Part I

# Basic introduction to the software

1. Singularity
2. Quick start
3. User level
4. Engineering concepts
5. Overview of the interface
6. Log introduction
7. Multi-terminal connection





✦ Lecture 1 ✦

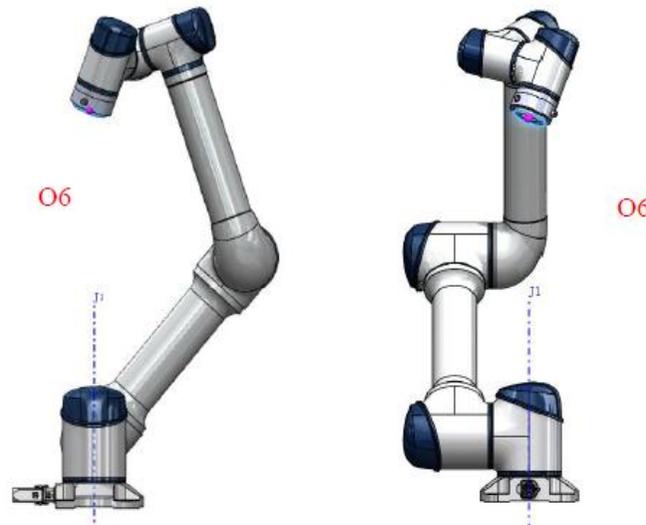
# Singularity

---

## ◆ 1.1 Shoulder singularity

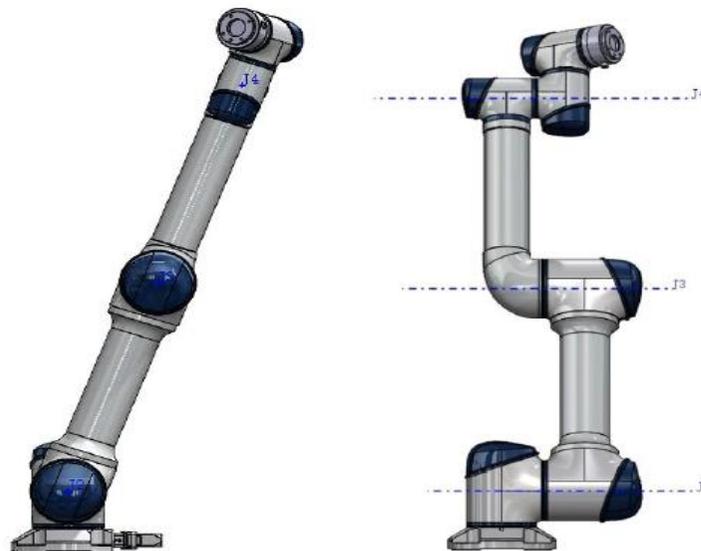
1. For the GCR series configuration, there are **shoulder singularities**, **elbow singularities**, and **wrist singularities**.

2. When the center of the wrist joint O6 is on the axis of one joint J1, the shoulder is singular at this time, resulting in the unsolved joint of 1. O6 is also affected by singularity when it is located very close to J1, where moving the end can **cause joint 1 to overdrive**. Refer to the following diagram for the singular posture of the adjacent shoulder:



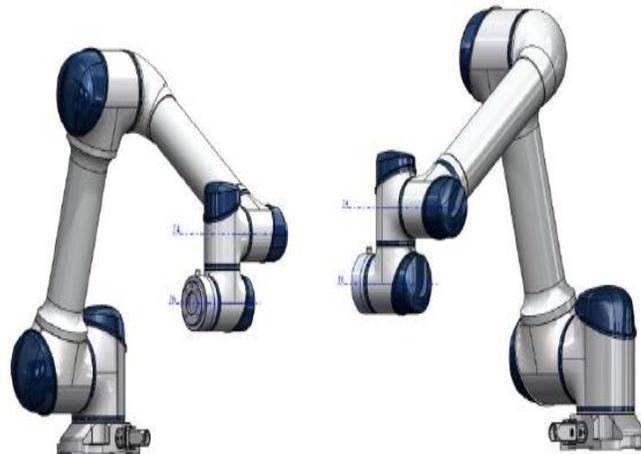
## ◆ 1.2 Elbow singular

When the axes of the second, third and fourth joints  $J_2$ ,  $J_3$  and  $J_4$  are coplanar, there is no solution for the two joints. Simply, when the joint 3 are close to 0 degrees and in a near singular position, moving the end may cause joint 2, 3, and 4 to overdrive. Refer to the following diagram for the singular pose of the near elbow:



## ◆ 1.3 Wrist singularity

When the joint 5 is 0 degrees, the joint 6 is not solved at this time, resulting in the singularity of the wrist. When the joint 5 are close to 0 degrees, it is a singular posture near the wrist, and moving the end at this time may cause the joint 4, 5, and 6 to overspeed. Refer to the diagram below:



## 1.4 Robot singularity stall risk

The robot will automatically slow down when performing motion planning (straight line, arc, etc., excluding joint movement) near the singularity, and should avoid the singularity or pass through the singularity with joint motion during teaching.

When the robot runs to or near the above-mentioned singularities (shoulder singularities, elbow singularities, wrist singularities), the planning motion based on Cartesian coordinates cannot be correctly inversely solved into the joint motion of each axis, and the motion planning will not be carried out correctly, and the joint jog motion or move j motion command can be adopted.



✦ Lecture 2 ✦

# Quick start

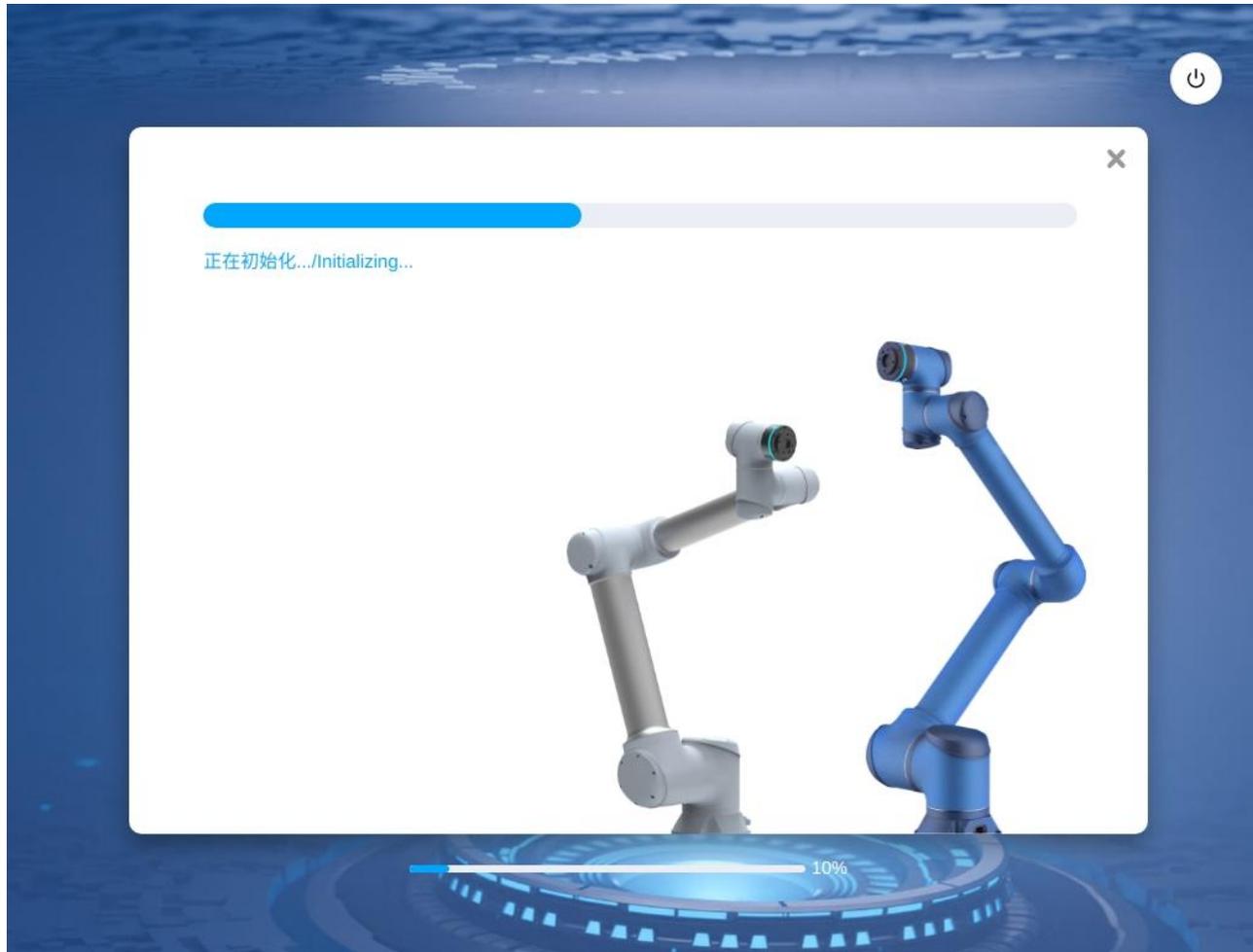
---

## ◆ 2.1 Turn on for the first time



1. By default, the boot system displays "Welcome to the AUCTECH collaborative robot" to enter the "Start" interface

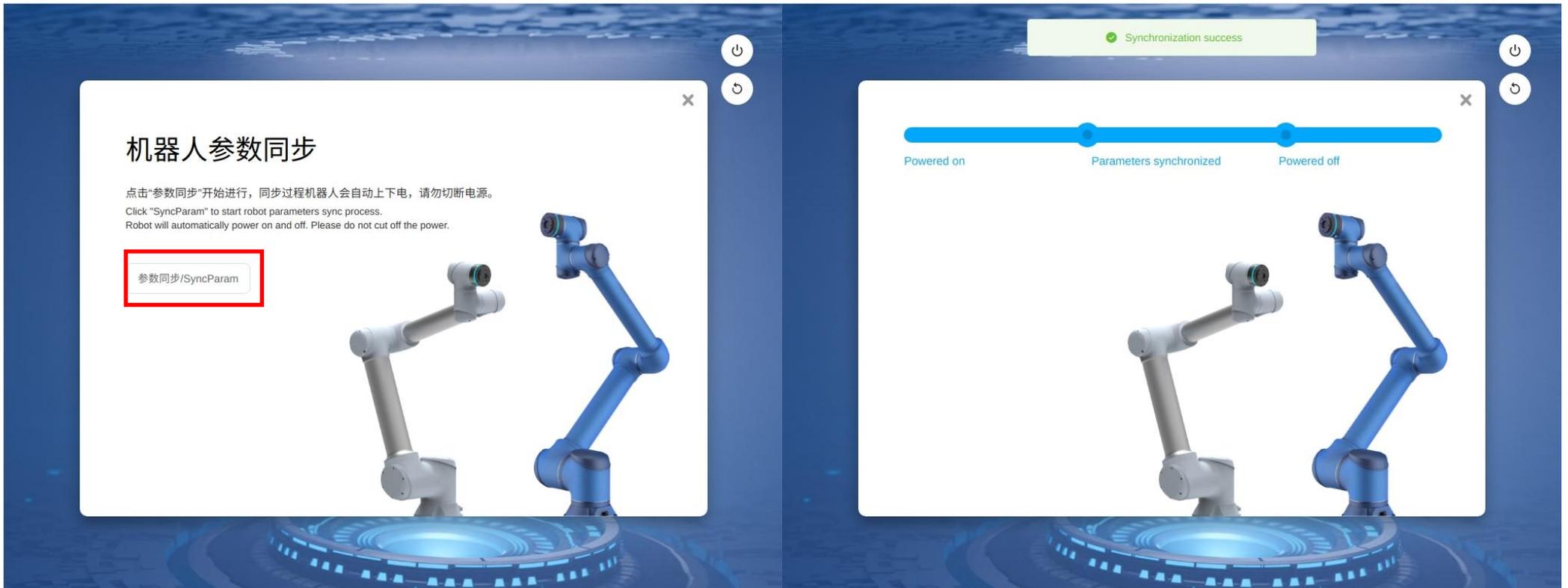
## ◆ 2.1 Turn on for the first time



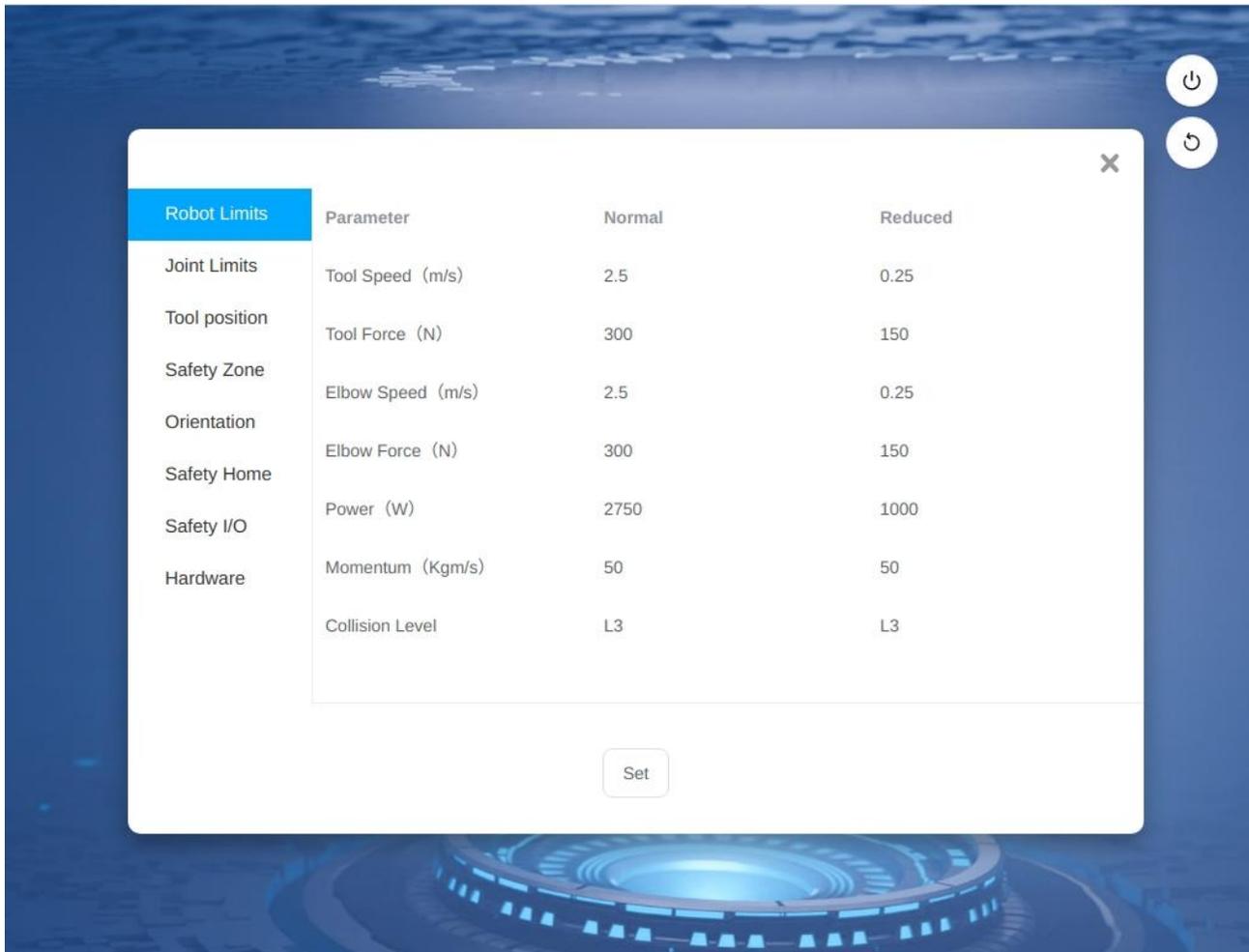
2. Click the **"Start"** button to initialize the robot. The progress bar of "Initializing..." is displayed, and "Initialization Completed" is displayed after initialization is completed.

## ◆ 2.1 Turn on for the first time

3. The initialization of the robot is completed, and the robot parameter synchronization page is entered. Click "**Synchronize Parameters**", the robot will power up, parameter synchronization, and power off operations.



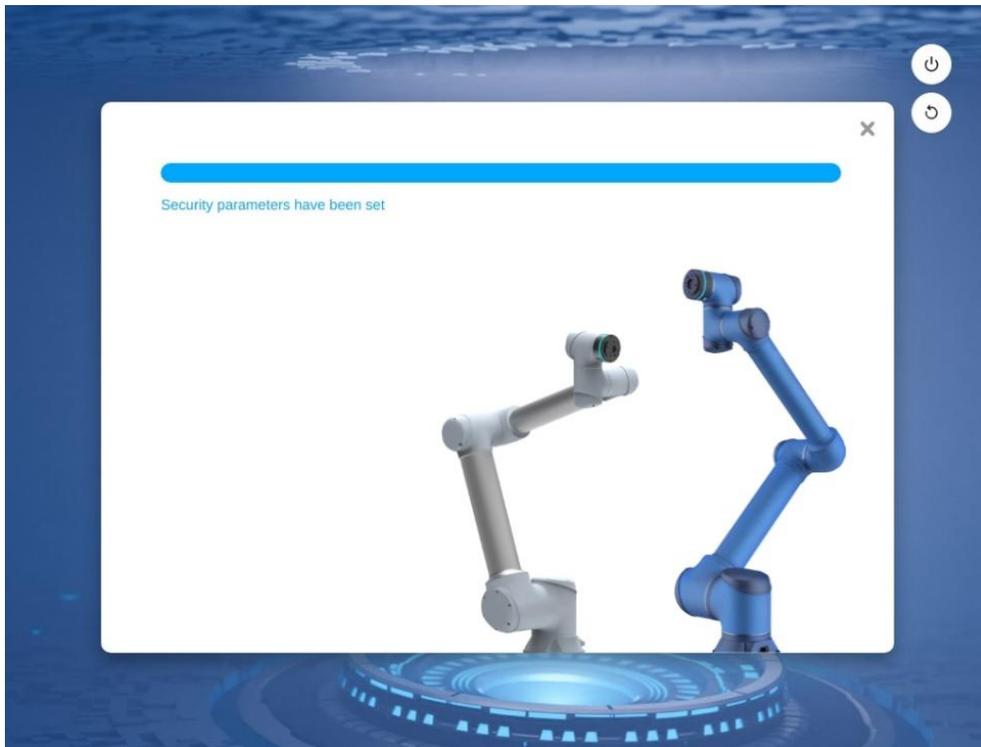
## 2.1 Turn on for the first time



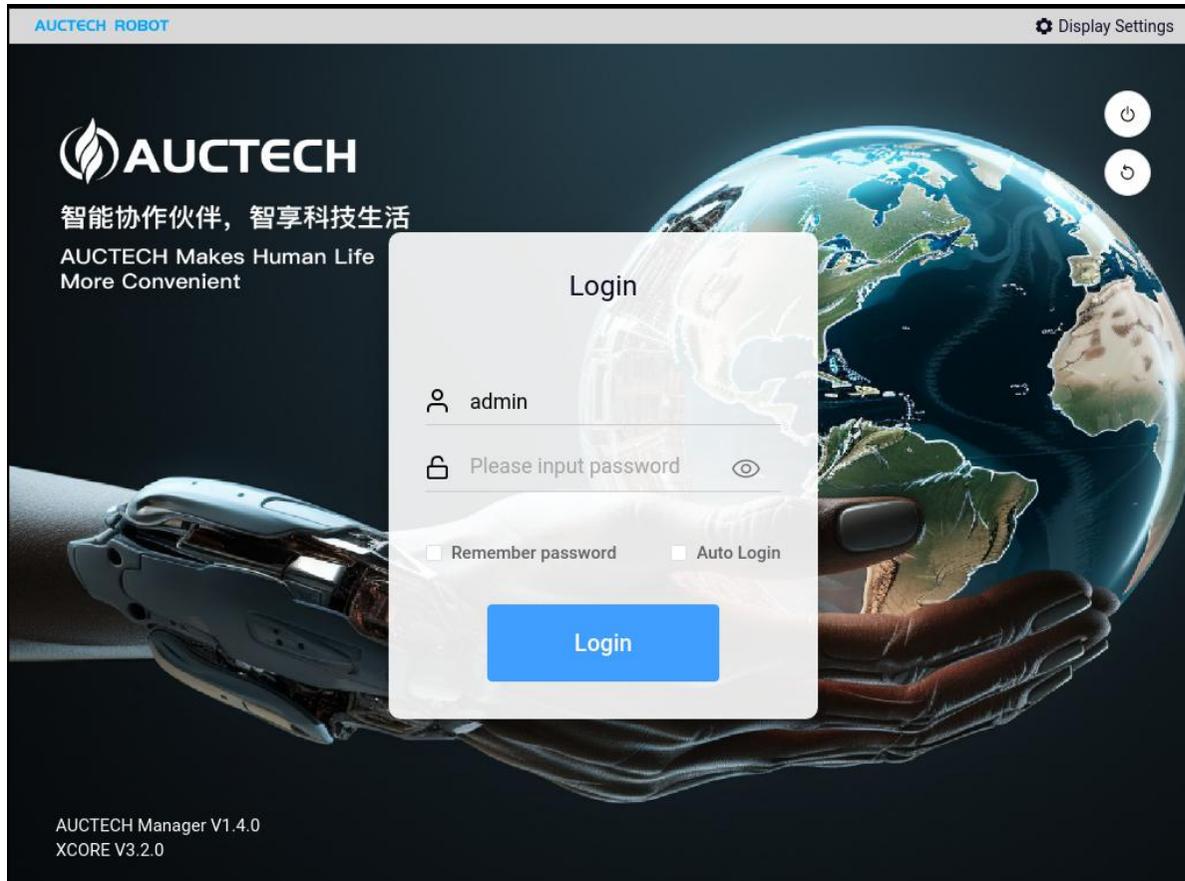
4. After the prompt synchronization is successful, **enter the security parameter setting page.**

## ◆ 2.1 Turn on for the first time

5. After confirming that the safety parameters are normal, click "**Set Safety Parameters**", and after the safety parameters are set, the initialization of the robot is completed, **and the control cabinet is restarted according to the prompts.**

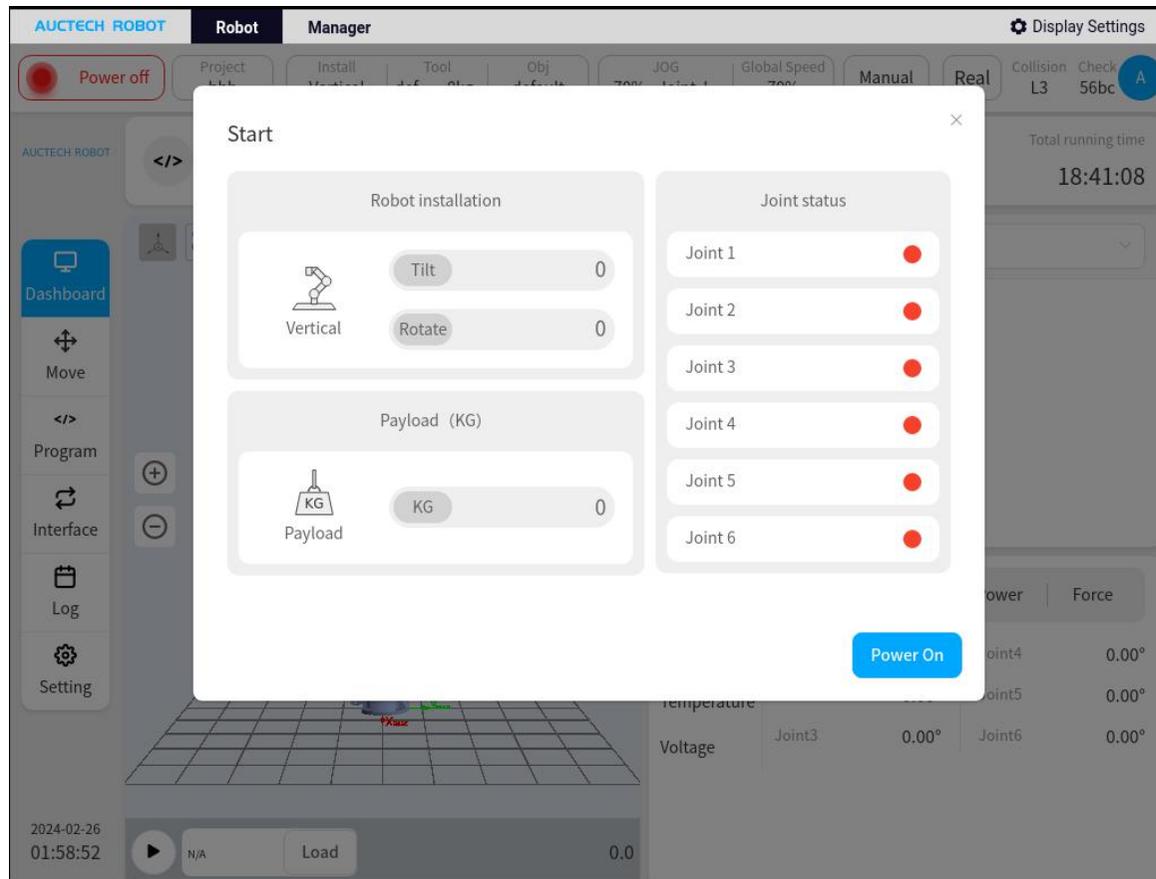


## 2.2 Log in

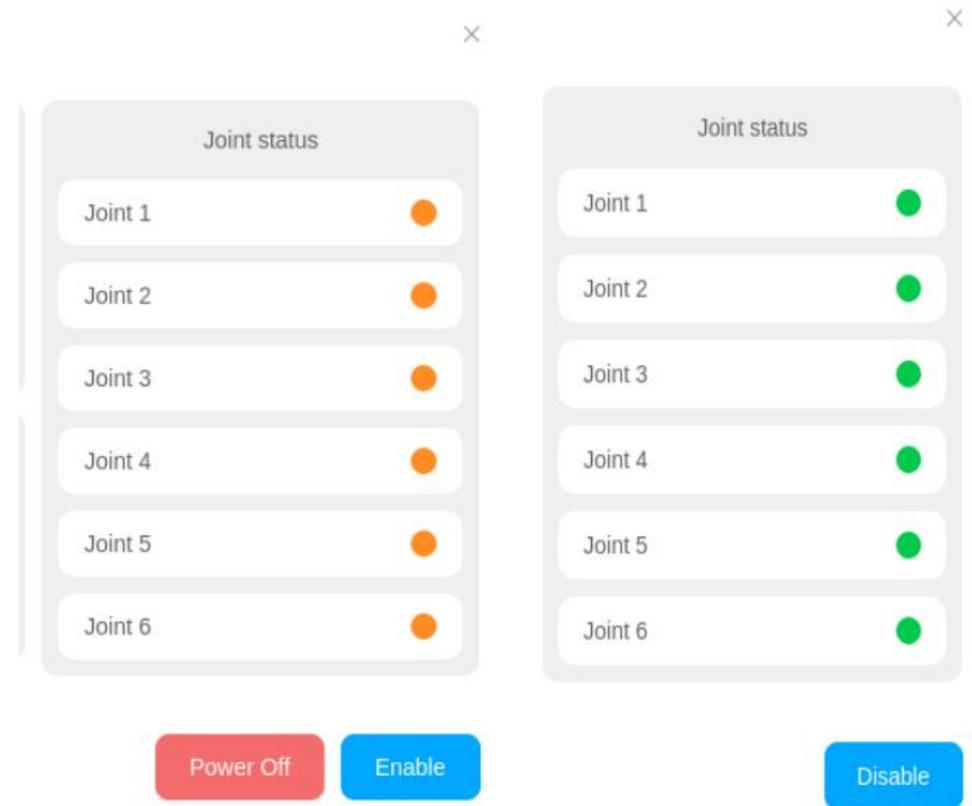


1. After connecting the teach pendant and turning it on, enter the login interface, **enter the username and password**, and click the "Login" button to log in successfully.
2. At the top right of the login interface is the shutdown button, and in the middle is the user input username and password area.
3. AUCTECH COBOT has **two accounts** out of the factory, one default account is **default**, one administrator account is **admin**, and the **password is 123**.

## 2.3 The bot starts



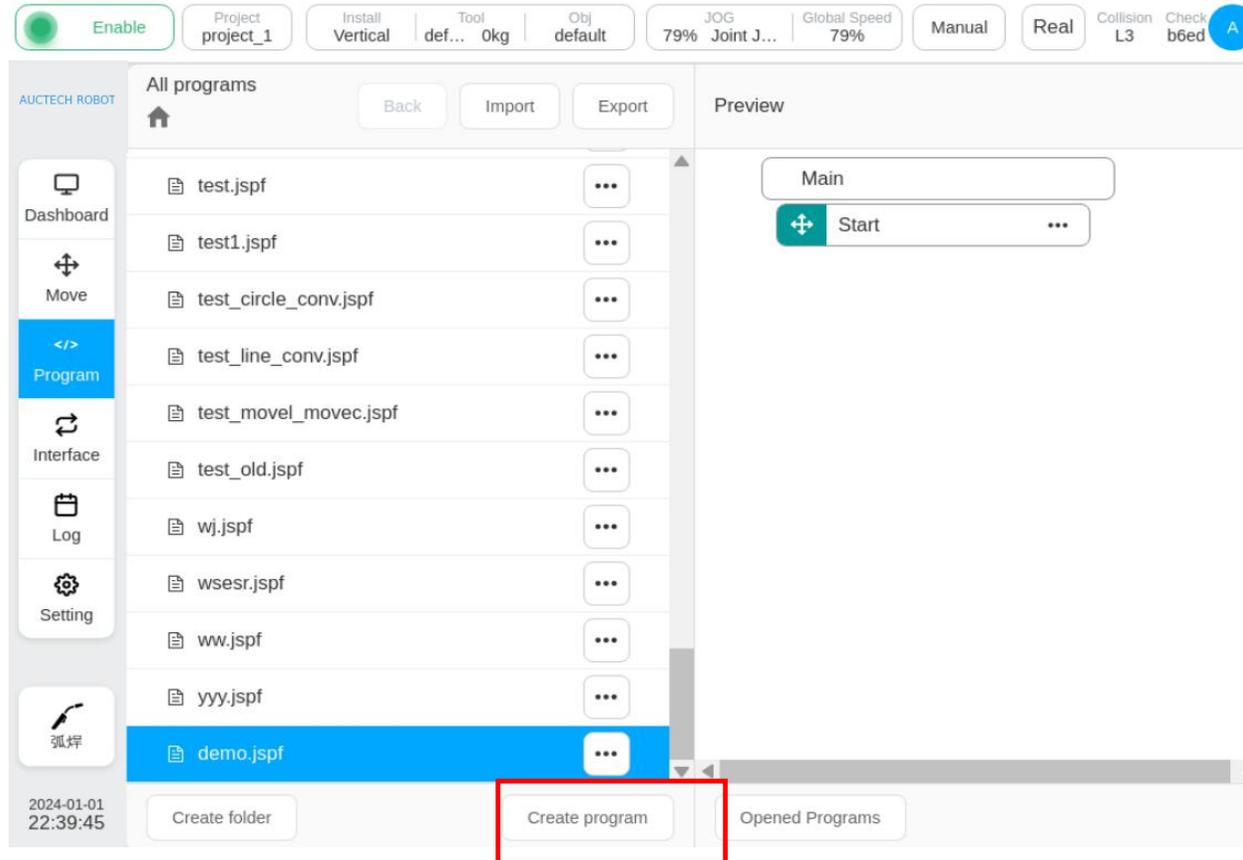
Startup interface (robot is not powered on)



The robot is powered on and not enabled

The enabled state of the bot

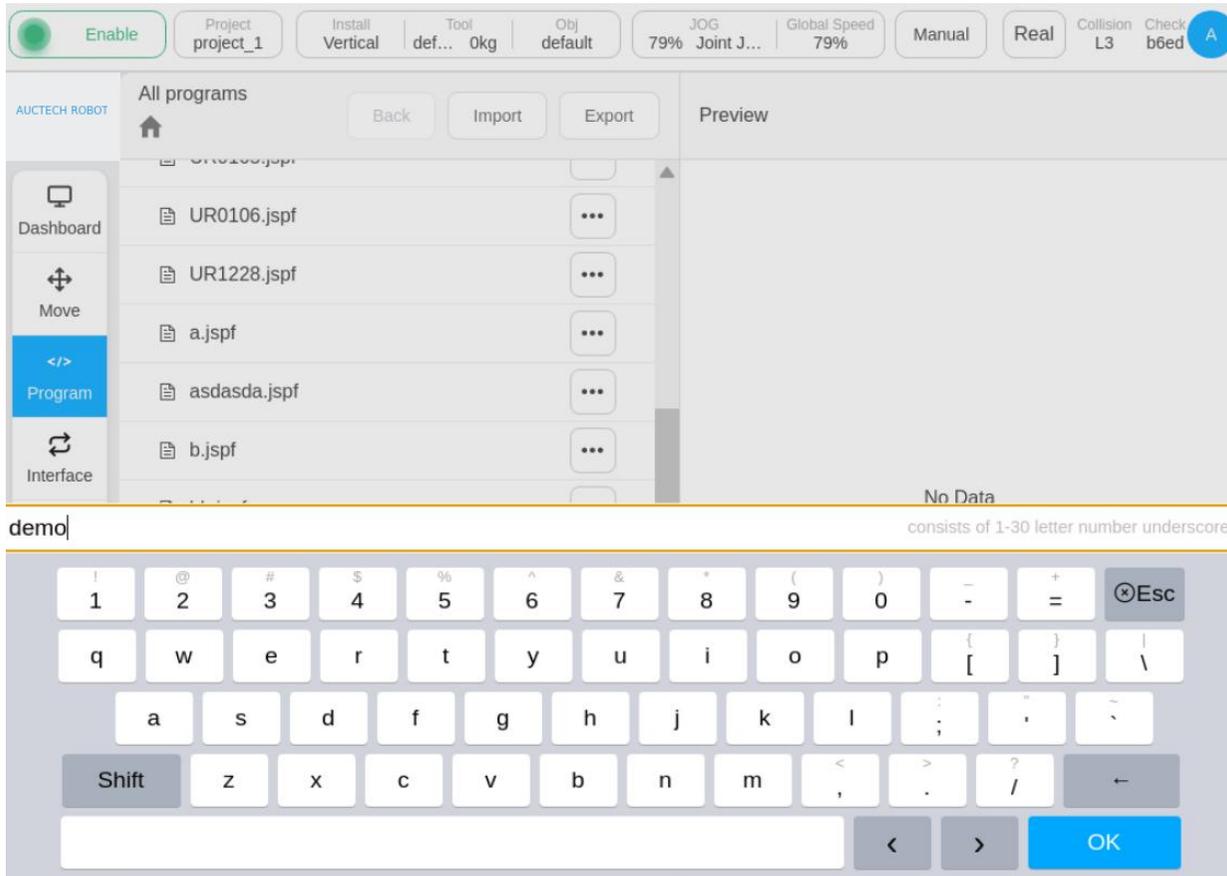
## 2.4 Create a program



The screenshot displays the AUCTECH ROBOT software interface. At the top, there is a control bar with buttons for 'Enable', 'Project project\_1', 'Install Vertical', 'Tool def...', 'Obj 0kg', 'Obj default', 'JOG 79%', 'Joint J...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check b6ed'. Below this is a sidebar with navigation options: 'Dashboard', 'Move', 'Program' (highlighted), 'Interface', 'Log', 'Setting', and '弧焊'. The main area is titled 'All programs' and contains a list of program files: test.jspf, test1.jspf, test\_circle\_conv.jspf, test\_line\_conv.jspf, test\_move1\_movec.jspf, test\_old.jspf, wj.jspf, wsesr.jspf, ww.jspf, yyy.jspf, and demo.jspf. The 'demo.jspf' file is selected. At the bottom of the interface, there are buttons for 'Create folder', 'Create program' (highlighted with a red box), and 'Opened Programs'. The 'Preview' window on the right shows a 'Main' block with a 'Start' button.

Note: The program file suffix is **.jspf**

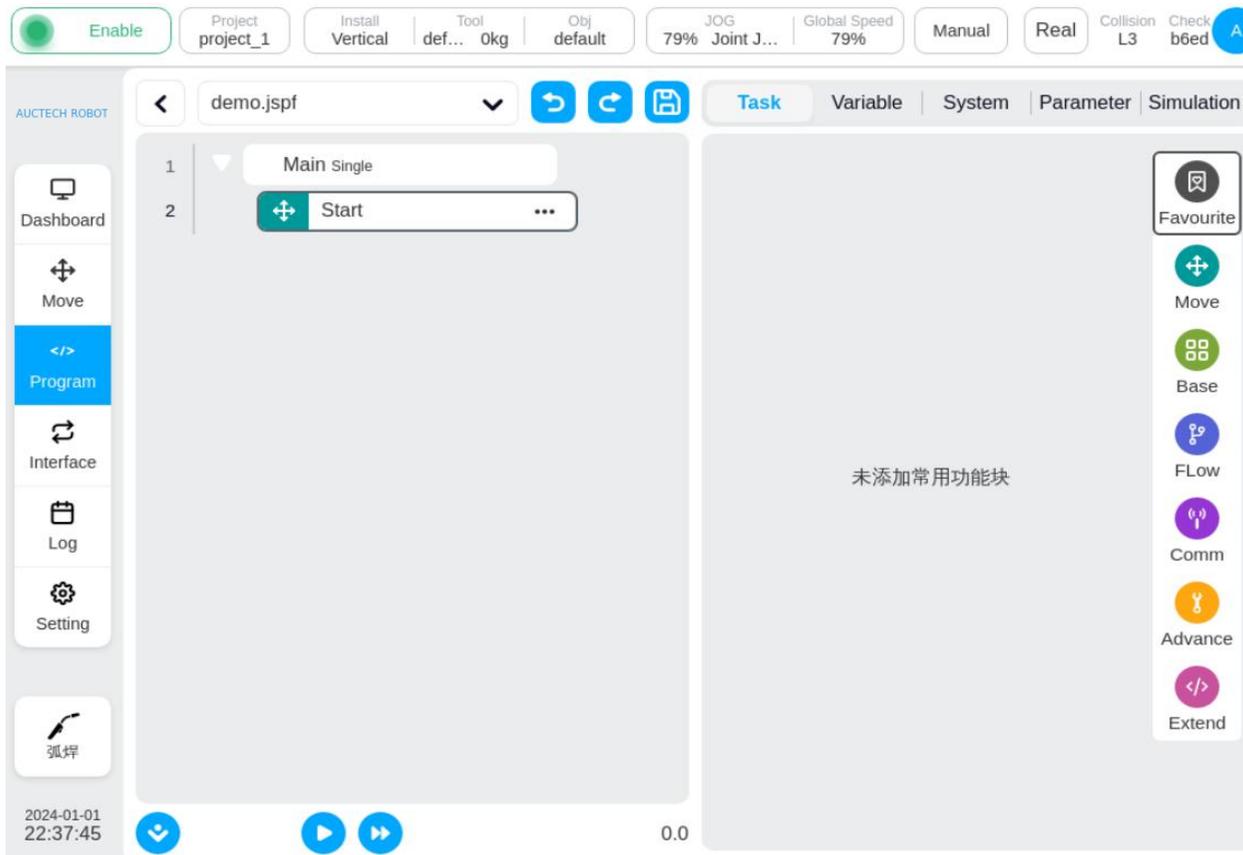
## 2.4 Create a program



Click "**Programs**" on the left navigation bar, switch to the program page, click "**New Program**" at the bottom, and enter the program name: demo. Click "OK" to create a new program **demo.jspf** and enter the programming page.

**Note:** The naming convention of program names is composed of 1-30 digits, letters, and underscores

## 2.4 Create a program



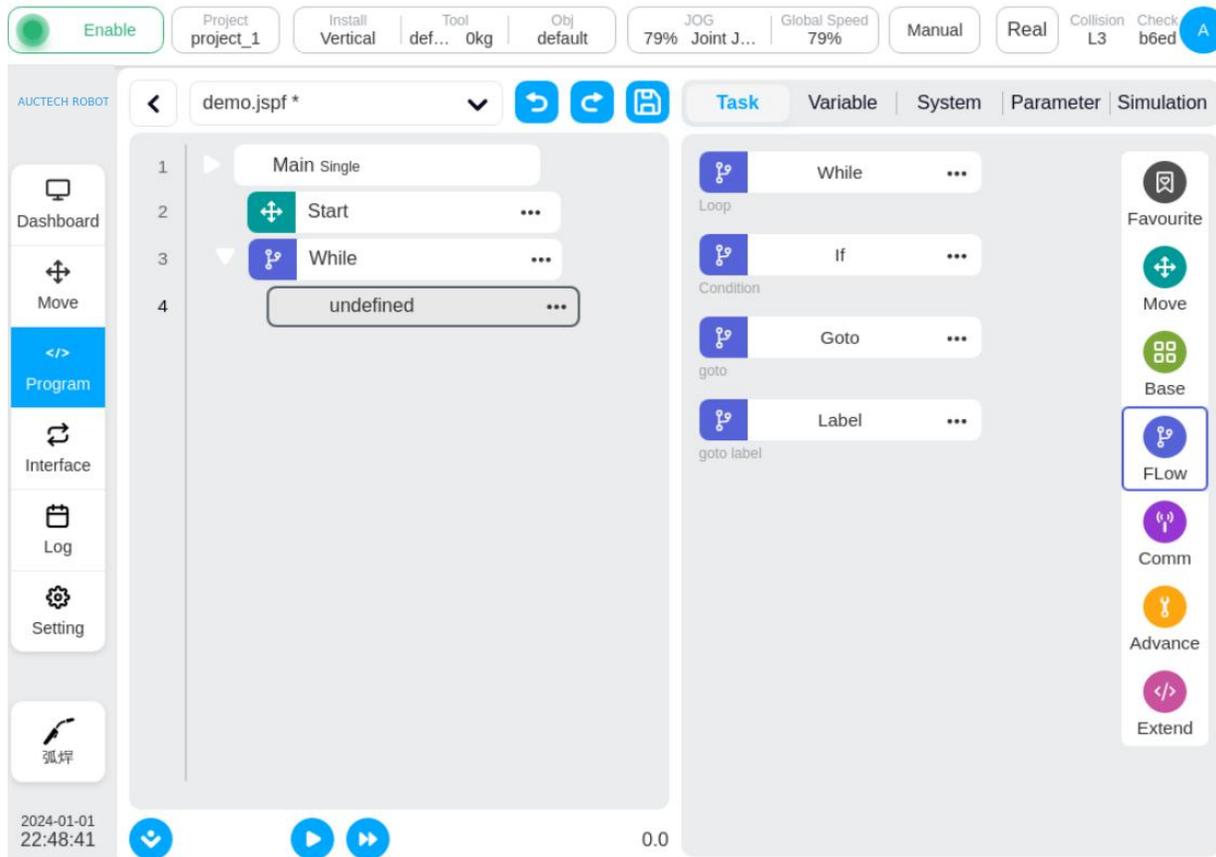
By default, the demo program contains a Start function block to set the starting point of the robot, double-click Start, or click "Parameters" on the right to view and set the starting point.

## 2.5 Exemple :Write a program

Now let's implement such a feature:

The robotic arm moves to the **preparation point**, moves down to a **working point**, **waits for 2s**, **set the digital output port 1 to high**, **waits for 1s**, set the digital output port 1 to low , and the process is cycled 3 times.

## 2.5 Write a program

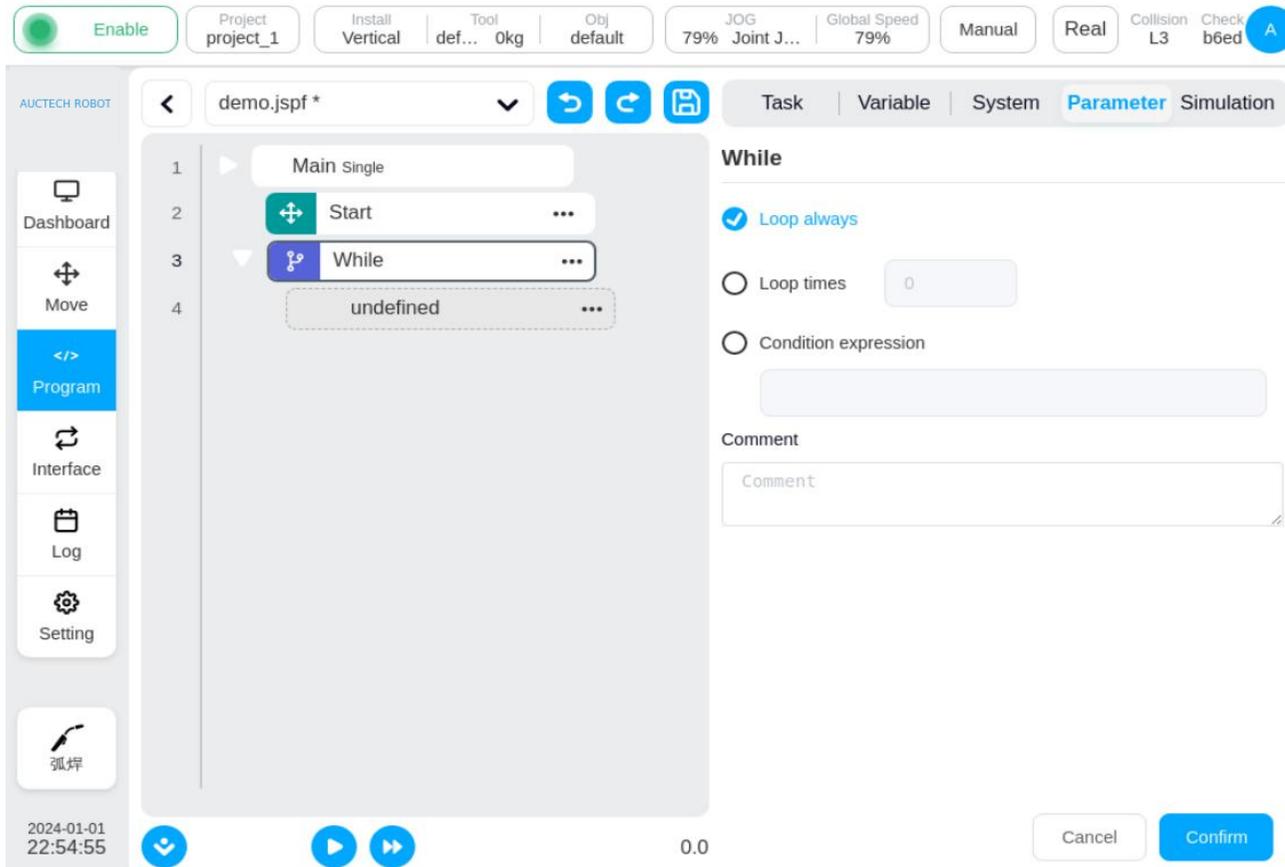


The screenshot displays the AUCTECH ROBOT software interface. At the top, there are control buttons for 'Enable', 'Project project\_1', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG 79% Joint J...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check b6ed'. Below this is a navigation bar with tabs for 'Task', 'Variable', 'System', 'Parameter', and 'Simulation'. The main workspace shows a program tree on the left with a 'Main Single' block containing a 'Start' block and a 'While' block. The 'While' block is expanded, showing an 'undefined' block. On the right, a 'Task' palette is visible, listing various function blocks: 'While' (Loop), 'If' (Condition), 'Goto' (goto), and 'Label' (goto label). A 'Flow' block is highlighted in the 'Flow' category of the palette. The bottom left corner shows the date and time: '2024-01-01 22:48:41'.

In the "Task" tab - "Flow", find the **While function** block, which can implement a circular process, **hold it down and drag it into the program tree on the left**, and release it to add the While function block to the program tree.

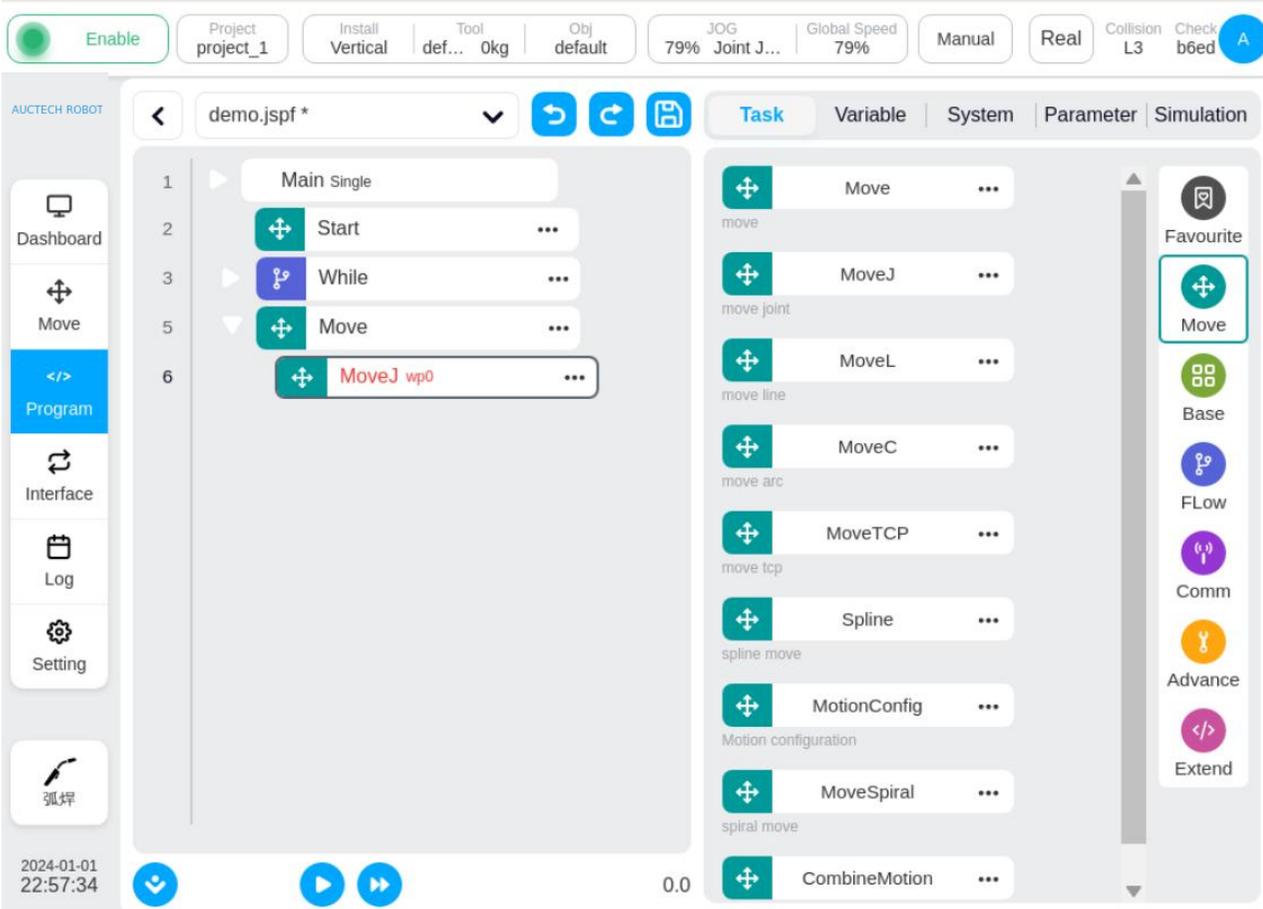
**Double-click** to add a function block as well.

## 2.5 Write a program



Double-click the While function block to configure its parameters, select Number of Cycles, and enter 3. Indicates that all program segments under the While block will be executed three times.

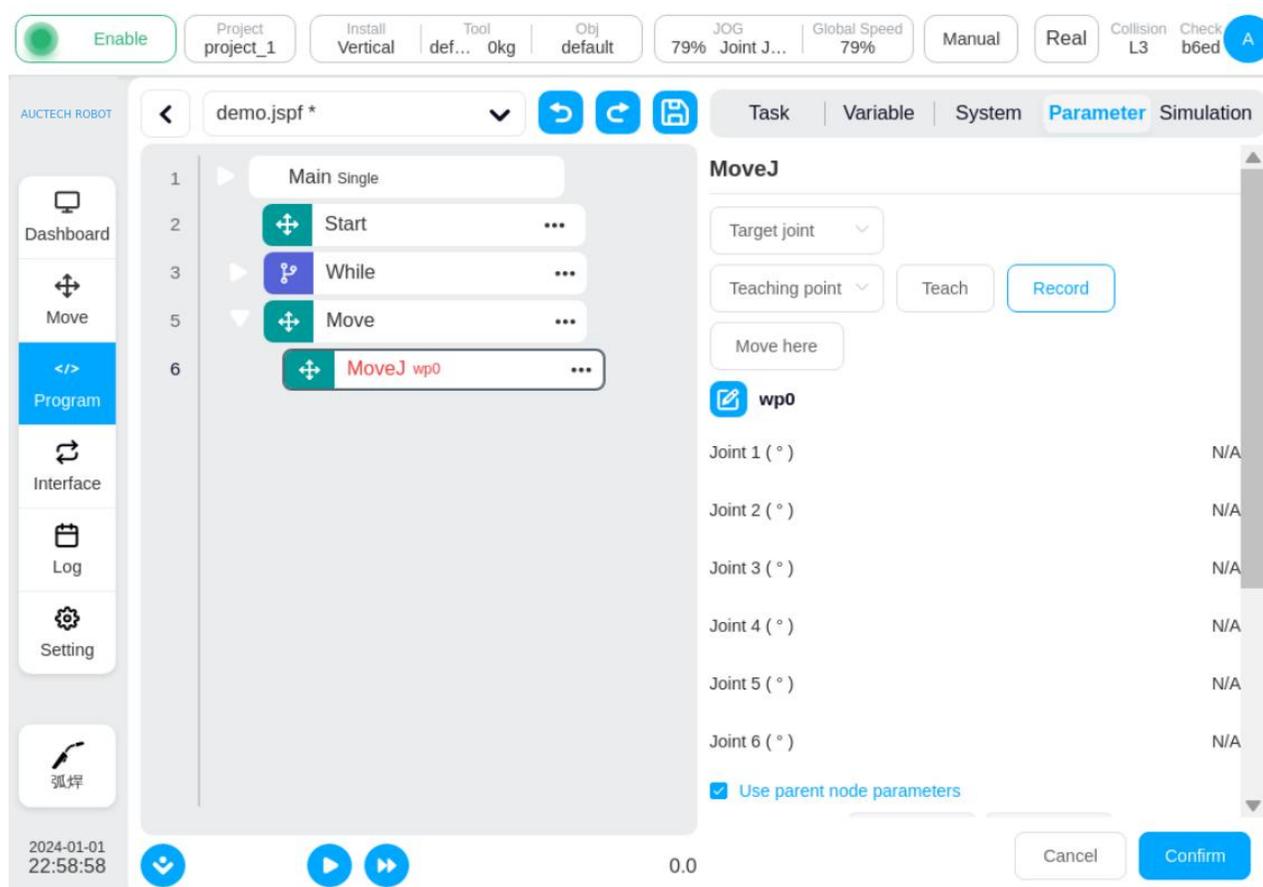
## 2.5 Write a program



The screenshot displays the AUCTECH ROBOT software interface. At the top, there is a status bar with various indicators: "Enable" (green), "Project project\_1", "Install Vertical", "Tool def... 0kg", "Obj default", "JOG 79% Joint J...", "Global Speed 79%", "Manual", "Real", "Collision L3", "Check b6ed", and a blue "A" button. Below the status bar, the interface is divided into several sections. On the left, there is a sidebar with icons for "Dashboard", "Move", "Program" (highlighted in blue), "Interface", "Log", "Setting", and "弧焊". The main area shows a program tree for "demo.jspf \*". The tree has a root node "Main Single" (line 1) with a play button. Underneath are "Start" (line 2), "While" (line 3), and "Move" (line 5). The "Move" block is expanded to show "MoveJ wp0" (line 6). To the right of the program tree is a "Task" tab with a list of function blocks: "Move", "MoveJ", "MoveL", "MoveC", "MoveTCP", "Spline", "MotionConfig", "MoveSpiral", and "CombineMotion". Each block has a plus icon and a three-dot menu. On the far right, there is a "Favourite" panel with icons for "Move", "Base", "Flow", "Comm", "Advance", and "Extend". At the bottom left, there is a timestamp "2024-01-01 22:57:34" and a "0.0" value. At the bottom center, there are play and stop buttons.

In the "Task" tab - "Move" - find the **MoveJ** function block and drag it into the While-sub-block in the program tree.

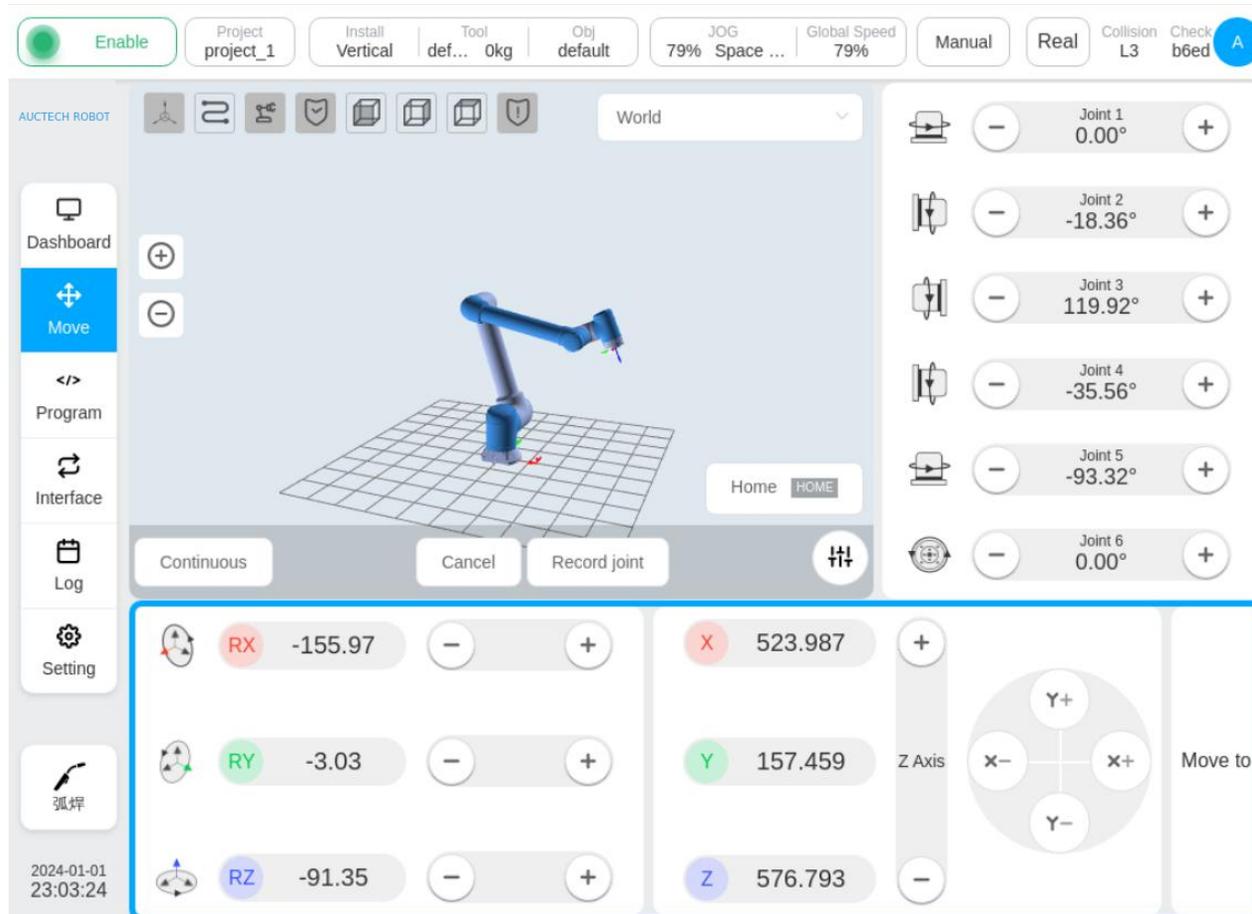
## 2.5 Write a program



The screenshot displays the AUCTECH robot programming software interface. The top status bar shows 'Enable', 'Project project\_1', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG 79%', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check b6ed'. The main workspace shows a program tree for 'demo.jspf \*' with a 'Main Single' block containing 'Start', 'While', 'Move', and 'MoveJ wp0'. The 'MoveJ' configuration panel is open, showing 'Target joint' (dropdown), 'Teaching point' (dropdown), 'Teach' button, and 'Record' button. The 'Move here' section lists joints 1 through 6, all with 'N/A' values. A 'wp0' button is visible. At the bottom, there are 'Cancel' and 'Confirm' buttons, and a timestamp '2024-01-01 22:58:58'.

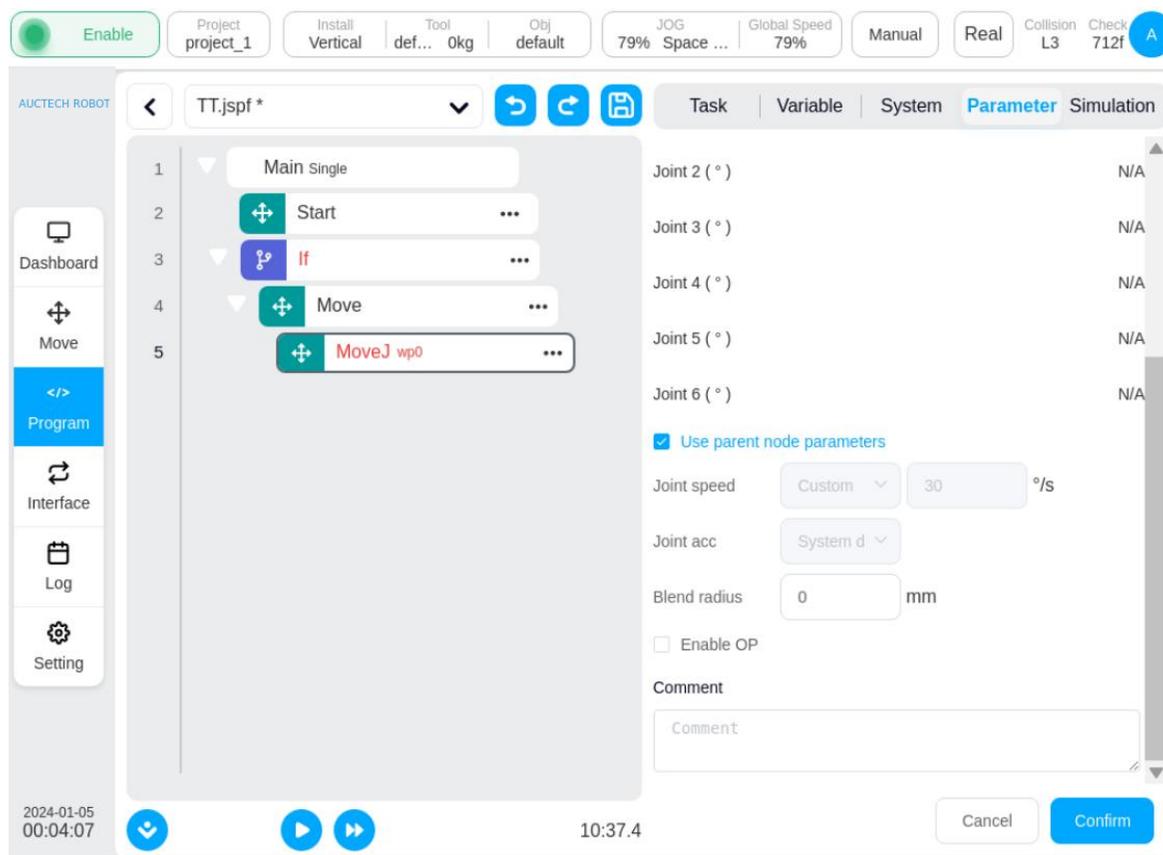
Double-click the MoveJ function block to configure its parameters. Click the "Select Point" button to switch to the point where the robot is taught on the mobile interface.

## 2.5 Write a program



After the point is set,  
click "**Record Joint**" to  
record the current robot  
posture.

## 2.5 Write a program



The screenshot displays the AUCTECH ROBOT software interface. At the top, there are status indicators: 'Enable' (green), 'Project project\_1', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG 79%' and 'Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check 712f'. The main workspace shows a program tree with a 'Main Single' block containing a 'Start' block, an 'If' block, a 'Move' block, and a 'MoveJ wp0' block. The 'MoveJ wp0' block is selected, and its parameter configuration area is visible on the right. The configuration area includes a table of joint set points:

Joint	Set Point
Joint 2 (°)	N/A
Joint 3 (°)	N/A
Joint 4 (°)	N/A
Joint 5 (°)	N/A
Joint 6 (°)	N/A

Below the table, there are several configuration options:

- Use parent node parameters
- Joint speed: Custom (dropdown), 30 (input), %/s
- Joint acc: System d (dropdown)
- Blend radius: 0 (input), mm
- Enable OP
- Comment: Comment (input field)

At the bottom of the configuration area, there are 'Cancel' and 'Confirm' buttons. The bottom status bar shows the date '2024-01-05', time '00:04:07', and a timer '10:37.4'.

In the parameter configuration area of the MoveJ function block, you can see the set point information.

You can further set the angular velocity and angular acceleration of the joint for the movement. Click the "Confirm" button below to save the parameters of the function block.

## 2.5 Write a program



The screenshot displays the AUCTECH robot programming software interface. The main window shows a program tree on the left with blocks: Main Single, Start, While, Move, MoveJ wp0, and MoveL wp1. The MoveL wp1 block is selected, and its configuration panel is open on the right. The configuration panel includes fields for X(mm), Y(mm), Z(mm), RX(\*), RY(\*), and RZ(\*), with values 523.987, 157.459, 576.793, -155.967, -3.031, and -91.351 respectively. There are also checkboxes for 'Use parent node coordinate' and 'Use parent node parameters', both checked. The tool speed is set to 100 mm/s. The interface also shows a top bar with 'Enable', 'Project project\_1', 'Install Vertical', 'Tool def...', 'Obj default', 'JOG 79%', 'Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', 'Check b6ed', and a 'Simulation' tab. The bottom bar shows a play button, a stop button, and a '0.0' timer.

In the same way, add the MoveL function block to the bottom of the MoveJ function block and set the point.

## 2.5 Write a program



The screenshot shows the AUCTECH ROBOT software interface. At the top, there is a status bar with various indicators: 'Enable' (green), 'Project project\_1', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG Space ... 79%', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', 'Check b6ed', and a blue 'A' button. Below this is a navigation bar with 'demo.jspf \*' and several icons. The main workspace is divided into a left sidebar with icons for 'Dashboard', 'Move', 'Program' (highlighted), 'Interface', 'Log', 'Setting', and '弧焊'. The central area shows a program tree with a 'Main Single' block containing 'Start', 'While', 'Move', and 'Wait' blocks. The 'Wait' block is selected, and its configuration panel is open on the right. The panel is titled 'Wait' and has the subtitle 'Wait until condition is met'. It features three radio buttons: 'Delays' (checked), 'When DI Port DI1 is LOW', and 'Wait thread end Select thread'. The 'Delays' option has a value of '2000'. There is also a 'Condition expression' option with an empty text field. At the bottom of the panel is a 'Comment' field. The bottom status bar shows the date '2024-01-02', time '00:37:24', and a '0.0' value. There are also 'Cancel' and 'Confirm' buttons at the bottom right of the configuration panel.

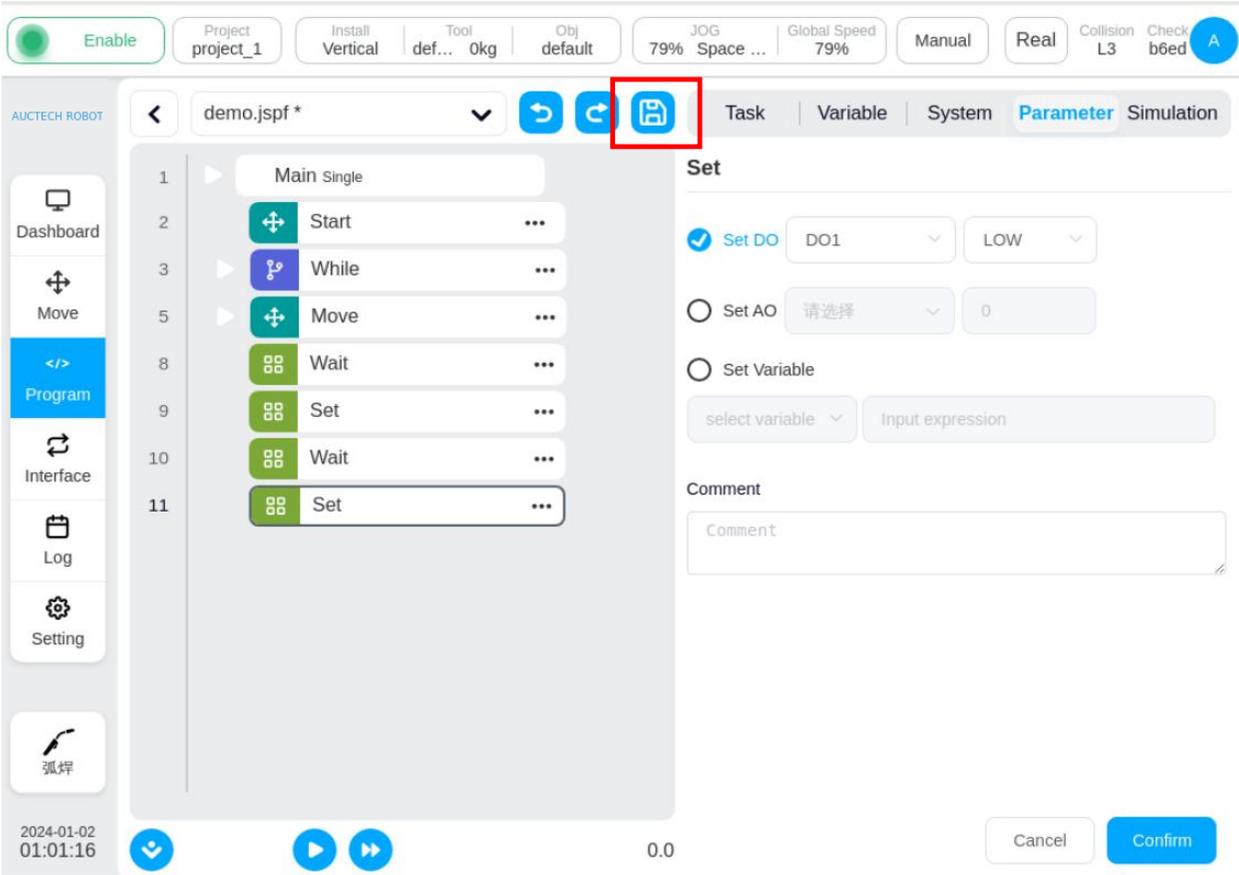
Find the Wait function block in the "Task" tab - "Basic" area, drag it to the position shown in the figure, select "Delay" in its parameter configuration area, and enter the parameter 2000, which means that the program will delay the execution of subsequent function blocks for 2s when the program is executed.

## 2.5 Write a program



In the "Task" tab - "Basics" area, select the Set function block and drag it below Wait. Select "Set Output" in the parameter configuration area, and select **DO1** and **HIGH**, which means that the digital output port 1 is set high when the program is executed to the function block.

## 2.5 Write a program



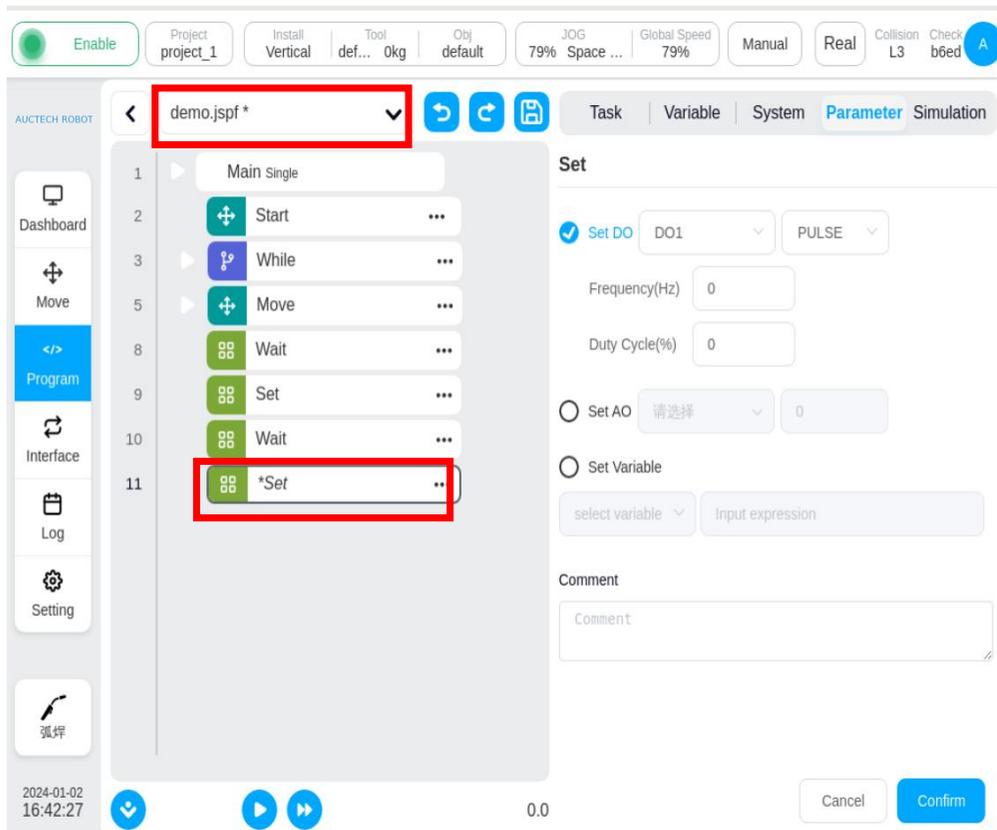
The screenshot displays the AUCTECH software interface for writing a program. The top bar includes various control buttons like 'Enable', 'Project project\_1', 'Install Vertical', 'Tool def...', 'Obj default', 'JOG 79%', 'Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check b6ed'. The main editor area shows a program named 'demo.jspf\*' with a sequence of blocks: 'Main Single', 'Start', 'While', 'Move', 'Wait', 'Set', 'Wait', and 'Set'. The 'Set' block configuration is visible on the right, showing 'Set DO' set to 'DO1' and 'LOW'. The 'Save Program' button is highlighted with a red box. The bottom status bar shows the date '2024-01-02', time '01:01:16', and a '0.0' value. There are also 'Cancel' and 'Confirm' buttons at the bottom right.

Similarly, add a Wait function block to set the delay of 1s, and add a Set function block to set DO1 to low.

Click the "**Save Program**" button below to save the program.

## 2.5 Write a program

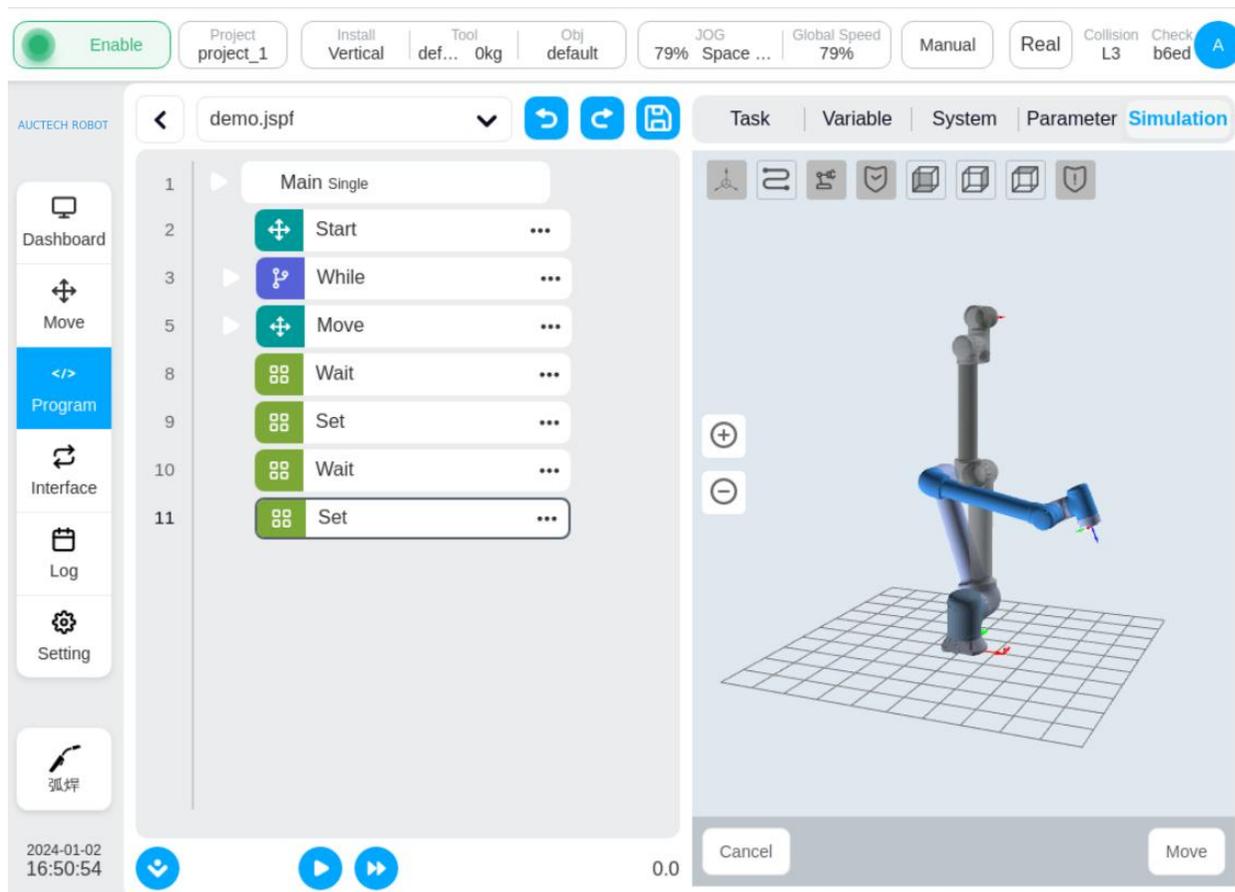
Note: **After** the function block, program, project, etc. are **changed**, there is a **"\*"** sign in the **corresponding position**. Disappear after saving.



1. If the project is not saved, it will not affect the current commissioning and production, but it will be restored to before the change after the power failure.

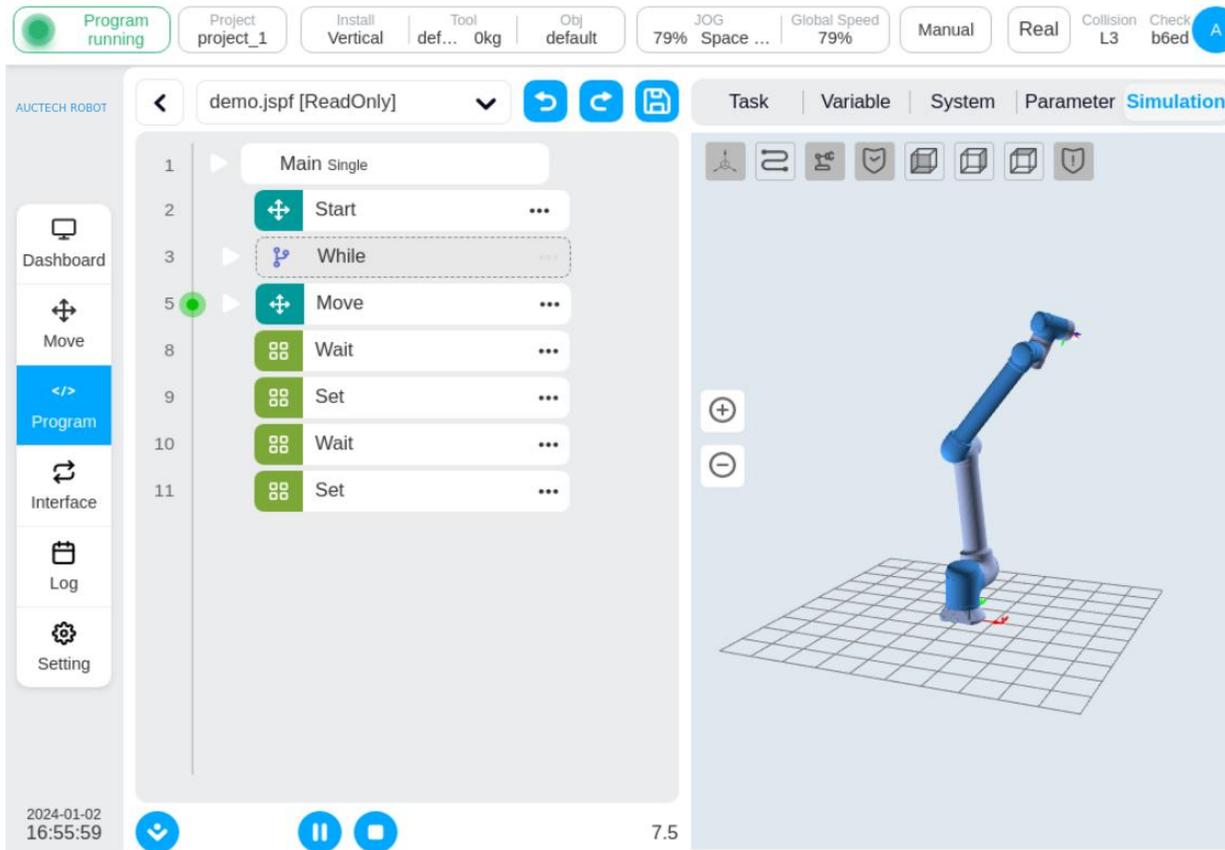
2. After the parameters in the function block are changed, you should **click "Confirm"** in the lower right corner first, and then perform other operations. If the function block is not confirmed, you cannot click "Save Program".

## 2.6 Run the program



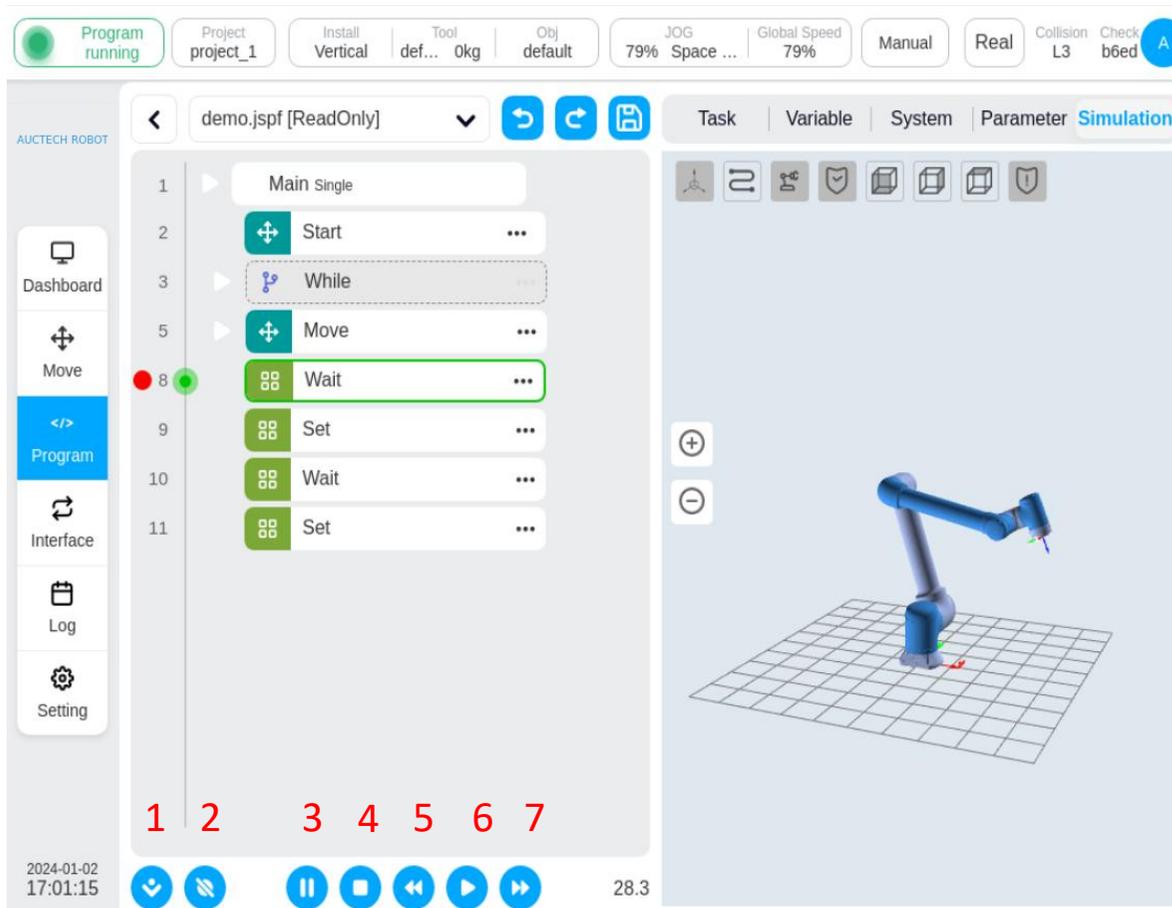
Switch to the “Simulation” tab, click the Run button at the bottom, **if the current position of the robot is different from the starting point of the program, the “Move” button will be displayed**, and the starting position of the robot will be displayed on the 3D display. After **long pressing and Hold “Move” button to move the robot to the starting position**. Click the Run button again to run the program.

## 2.6 Run the program



When the program is executed, the green dot in front of the program tree is used to indicate the function block that is currently being executed, and the real-time posture of the robot can be displayed in the 3D model. You can pause or stop the program while it is running.

## 2.6 Run the program



The screenshot displays the AUCTECH robot control software interface. At the top, there is a status bar with various indicators: 'Program running' (green), 'Project project\_1', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG 79% Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check b6ed'. Below this is a navigation menu with 'Task', 'Variable', 'System', 'Parameter', and 'Simulation' (selected). The main area shows a program tree for 'demo.jspf [ReadOnly]' with steps 1 through 11. Step 8, 'Wait', is highlighted with a green border and a red dot, indicating a breakpoint. A 3D simulation of a blue robotic arm is visible on the right. At the bottom, there are playback controls and a timestamp '2024-01-02 17:01:15'.

### Debugging Function: Breakpoint

You can add a breakpoint before the sequence number of the program tree.

The program runs to a breakpoint and pauses, and in the lower right corner, you can choose to continue or stop.

- 1: Auto-follow
- 2: Cancel the breakpoint
- 3: Pause the program
- 4: Stop the program
- 5: Backwards
- 6: Continue running
- 7: Forward



✦ Lecture 3 ✦

# User level

---

## 3.1 User level

User permission level	Permission description
Operator	<ul style="list-style-type: none"><li>Allows you to select projects, run programs, manually jog programs, view robot status, and more</li></ul>
Programmer	<ul style="list-style-type: none"><li>Operator user permissions</li><li>Programming robot programs, engineering configurations</li><li>The default username is default and the initial <b>password is 123</b></li></ul>
Maintainer	<ul style="list-style-type: none"><li>Operator user permissions</li><li><b>System updates</b></li></ul>
Admin	<ul style="list-style-type: none"><li>Maintainer user permissions</li><li><b>User management</b></li><li>The single username is admin and the initial <b>password is 123</b></li></ul>

# 3.1 User level



## AUCTECH Manager

Dashboard

Net

Log

Service

Update

**Account**

Index	Name	Role	Create
1	admin	Admin	
2	default	Programmer	

### Create User

Name \*

Role \*

Operator  
Programmer  
Maintainer

Password \*

Confirm Pass \*

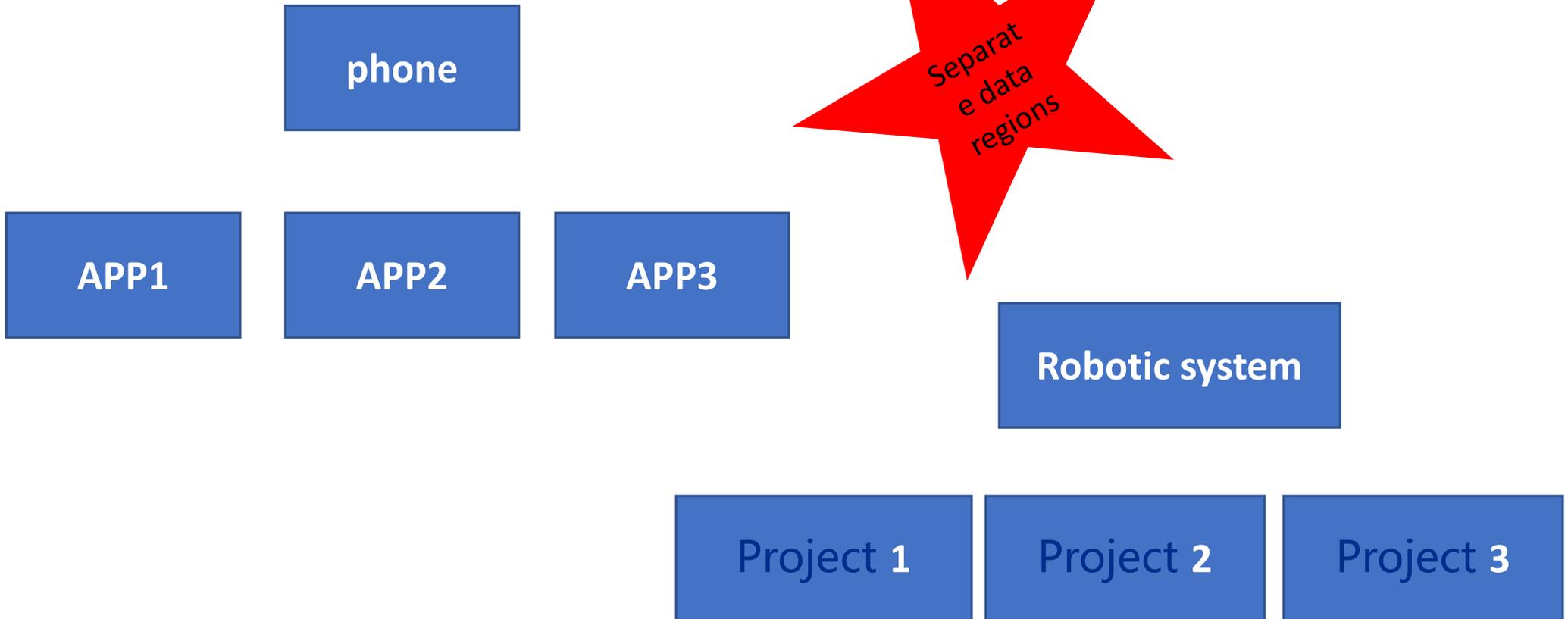


✦ Lecture 4 ✦

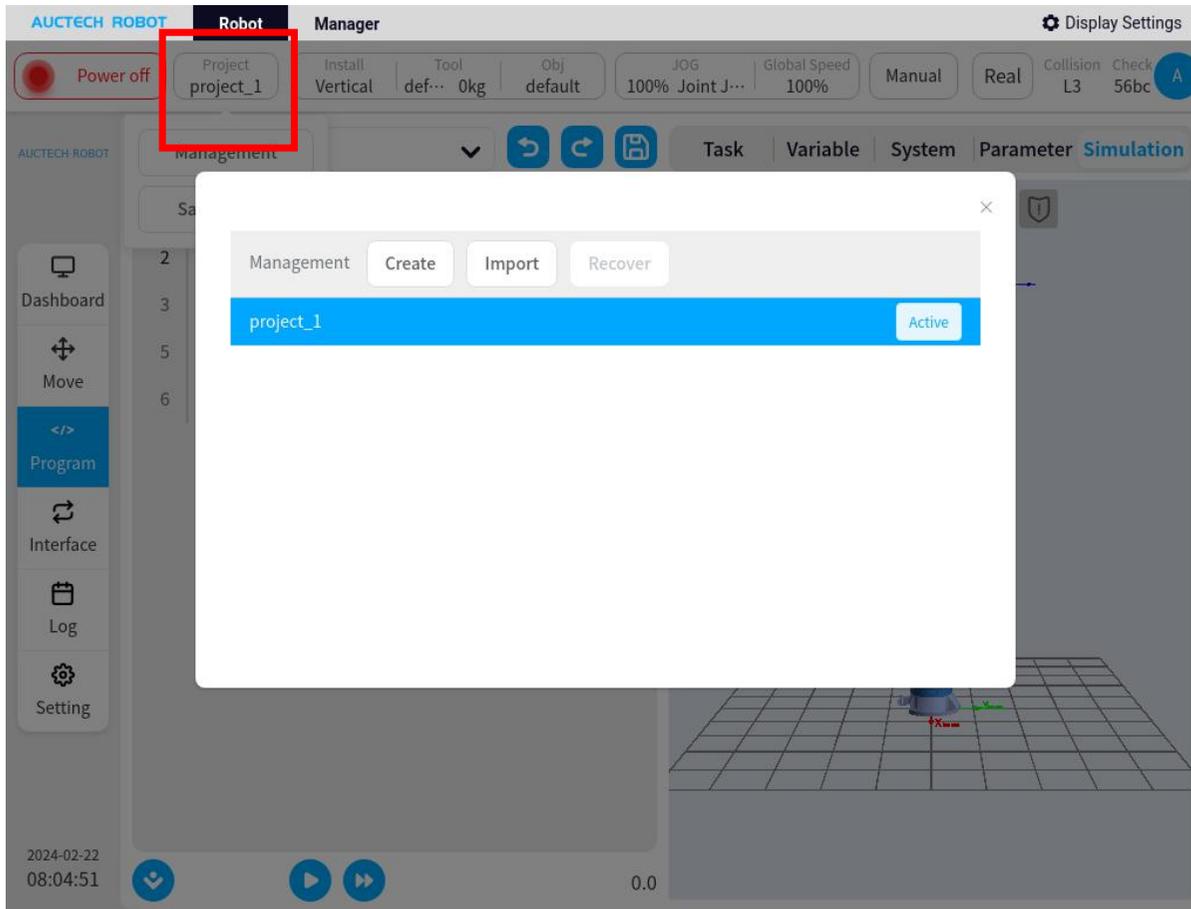
# Project concepts

---

## ◆ 4.1 Project concepts

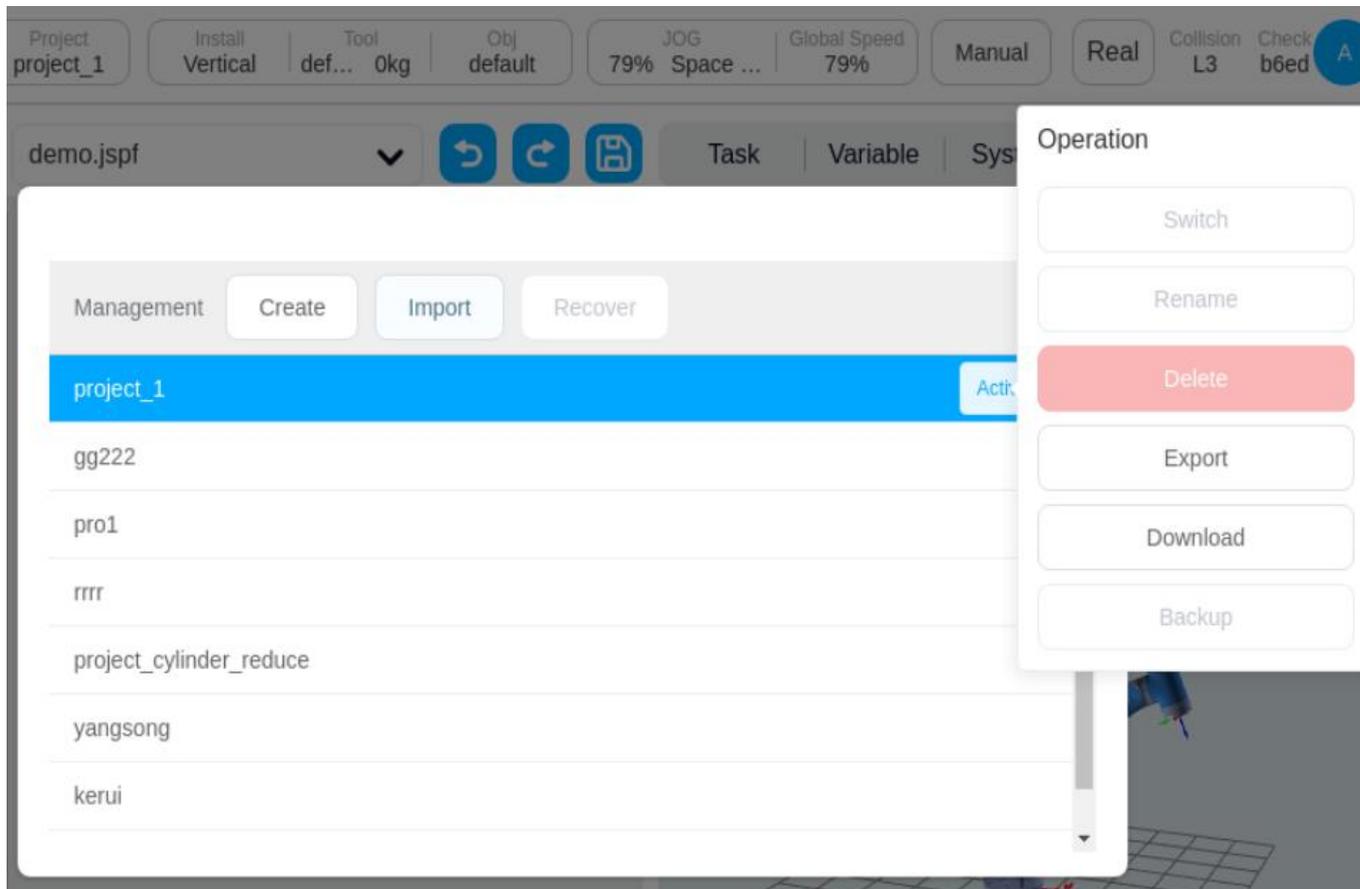


## 4.2 project management



Support multi-process management, engineering data includes robot programs, engineering global variables, engineering settings and other information. **When the system starts loading, the current project data will be loaded according to the system settings,** and other project data cannot be used in the current project.

## ◆ 4.3 project management



On the project management page, you can **create, import, rename, and export** projects.



✦ Lecture 5 ✦

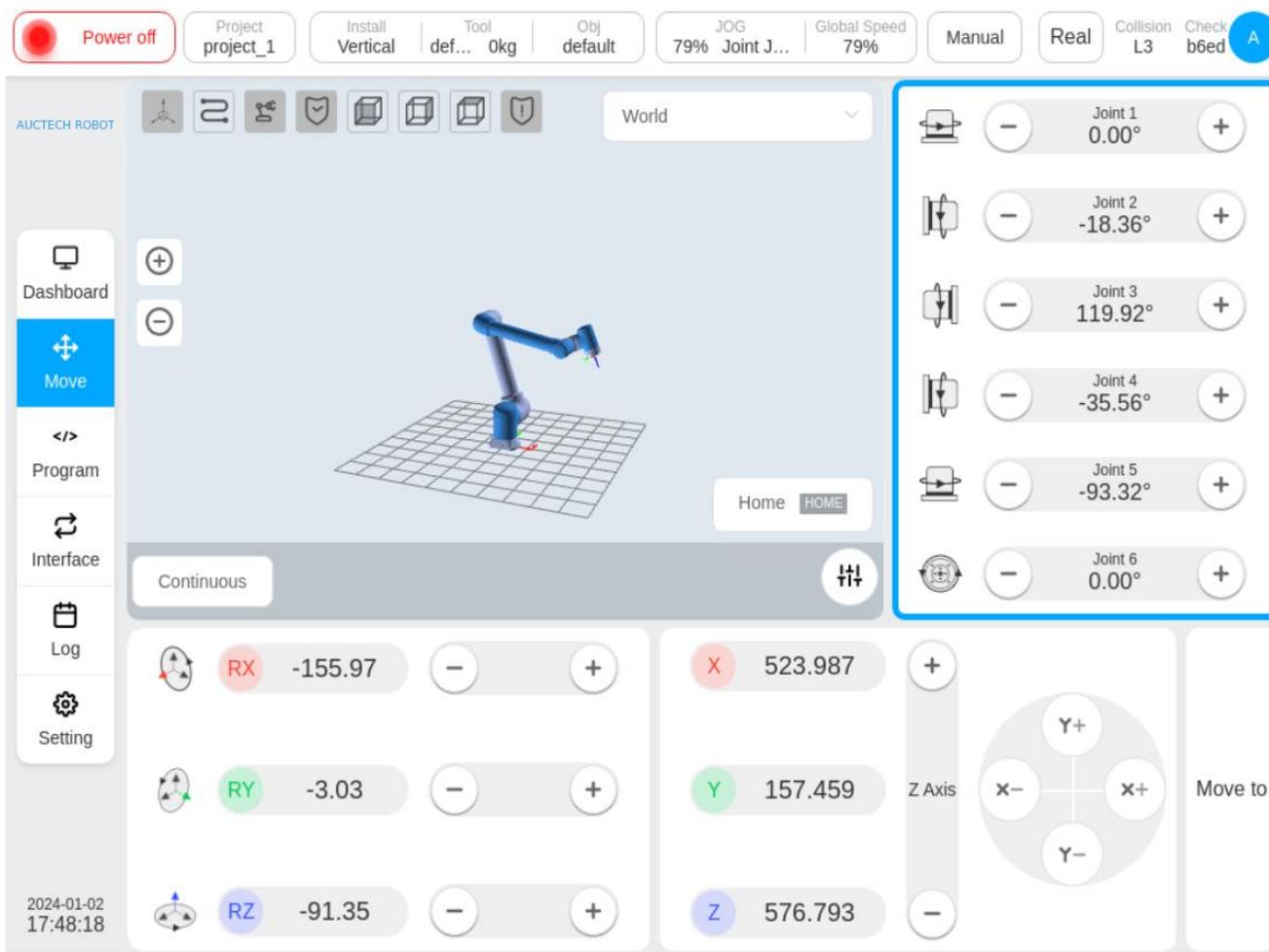
# Overview of the interface

---

# 5.1 Dashboard

Joint	Position	IO status	Power	Force
Position	Joint1	0.00°	Joint4	-35.56°
Temperature	Joint2	-18.36°	Joint5	-93.32°
Voltage	Joint3	119.92°	Joint6	0.00°

## 5.2 Move

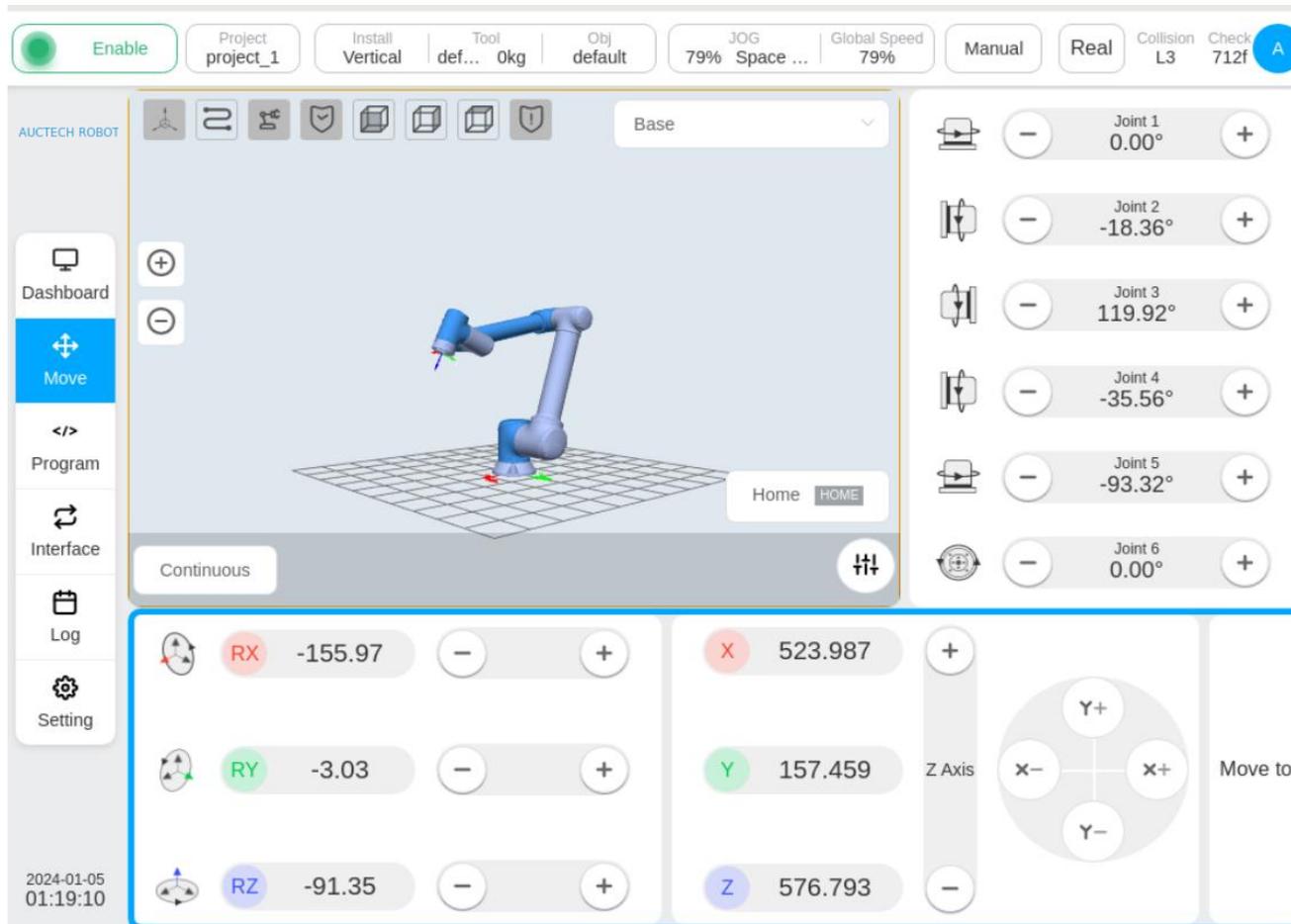


There are **two modes of movement** when manually moving the robot: **joint movement and end movement.**

When you click on a joint to move any area, **a blue border** appears in that area to select for joint movement.

**Note: The robot does not refer to the coordinate system when the joints move**

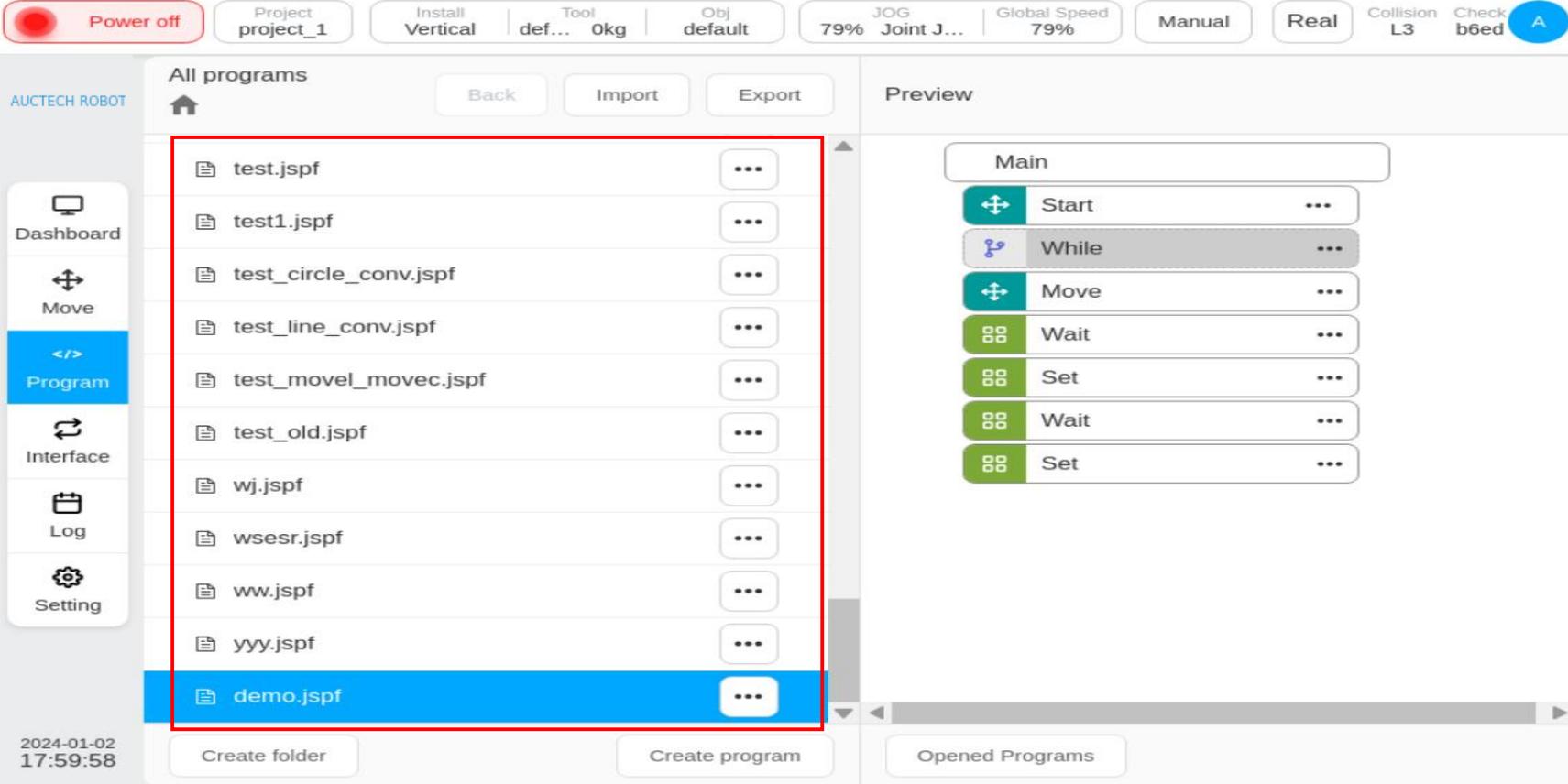
## 5.2 Move



The screenshot displays the AUCTECH robot control interface. At the top, there is a status bar with 'Enable', 'Project project\_1', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG 79% Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check 712f'. The main area shows a 3D view of the robot arm on a grid, with a 'Home HOME' button. The left sidebar contains 'Dashboard', 'Move' (highlighted), 'Program', 'Interface', 'Log', and 'Setting'. The right side features joint angle sliders for Joint 1 (0.00°), Joint 2 (-18.36°), Joint 3 (119.92°), Joint 4 (-35.56°), Joint 5 (-93.32°), and Joint 6 (0.00°). The bottom section, highlighted with a blue border, shows Cartesian coordinate inputs: RX (-155.97), RY (-3.03), RZ (-91.35), X (523.987), Y (157.459), and Z (576.793). A 'Z Axis' slider and a 'Move to' directional pad are also present.

When you click on the end to move any area, a blue border appears in that area, and the end move is selected.

## 5.3 program list



The screenshot displays the AUCTECH robot control software interface. At the top, there is a status bar with a red "Power off" button, a "Project project\_1" dropdown, and various operational modes like "Install Vertical", "Tool def...", "Obj 0kg", "JOG 79%", "Joint J...", "Global Speed 79%", "Manual", "Real", "Collision L3", and "Check b6ed".

The main interface is divided into two main sections: "All programs" and "Preview".

**All programs:** This section contains a list of program files. The files listed are: test.jspf, test1.jspf, test\_circle\_conv.jspf, test\_line\_conv.jspf, test\_movel\_movec.jspf, test\_old.jspf, wj.jspf, wsesr.jspf, ww.jspf, yyy.jspf, and demo.jspf. The "demo.jspf" file is highlighted in blue. A red box highlights the entire list of programs.

**Preview:** This section shows the structure of the selected program, "Main". It contains a sequence of steps: Start, While, Move, Wait, Set, Wait, and Set. Each step has a corresponding icon and a three-dot menu.

At the bottom of the interface, there is a date and time stamp "2024-01-02 17:59:58" and three buttons: "Create folder", "Create program", and "Opened Programs".

# 5.4 interface



Power off | Project project\_1 | Install Vertical | Tool def... 0kg | Obj default | JOG 79% Joint J... | Global Speed 79% | Manual | Real | Collision L3 | Check b6ed A

I/O | Register | CCI | TCI | TCP/IP | Industrial Bus

Digital In | Digital Out | Function In | Function Out | Analog In | Analog Out

Type	Name	Description	Modbus	Status
Digital In 1	DI1		176	<input type="checkbox"/> 0
Digital In 2	DI2		177	<input type="checkbox"/> 0
Digital In 3	DI3		178	<input type="checkbox"/> 0
Digital In 4	DI4		179	<input type="checkbox"/> 0
Digital In 5	DI5		180	<input type="checkbox"/> 0
Digital In 6	DI6		181	<input type="checkbox"/> 0
Digital In 7	DI7		182	<input type="checkbox"/> 0
Digital In 8	DI8		183	<input type="checkbox"/> 0

Dashboard | Move | Program | **Interface** | Log | Setting

2024-01-02 18:00:48

# 5.5 log



Enable | Project project\_1 | Install Vertical | Tool def... 0kg | Obj default | JOG 79% Joint J... | Global Speed 79% | Manual | Real | Collision L3 | Check b6ed | A

AUCTECH ROBOT

Log Level All History logs

01-02-18:01:20	info	Robot state changed:Enable
01-02-18:01:16	info	Robot state changed:Disable
01-02-17:36:53	info	Program state changed:Stopped
01-02-17:36:53	info	Program state changed:Stopping
01-02-17:36:33	info	Robot state changed:Poweroff
01-02-17:36:32	info	Robot state changed:Disable
01-02-17:18:16	info	Program state changed:Paused
01-02-17:18:16	info	Program state changed:Pausing
01-02-17:00:47	info	Program state changed:Running
01-02-17:00:42	info	Program state changed:Stopped
01-02-17:00:31	info	Program state changed:TaskRunning
01-02-16:56:17	info	Program state changed:Stopped
01-02-16:55:52	info	Program state changed:Running
01-02-16:54:48	info	Program state changed:Stopped
01-02-16:54:48	error	0x00C00000 Compiler error:@/home/ducocobot/siasun_cobot/run/dist/plugins_custom/weld/scripts/api.lua:3: undefined variable   none   none
01-02-16:54:48	info	Program state changed:Running
01-02-16:54:42	info	Program state changed:Stopped
01-02-16:54:42	info	Program state changed:Stopping

2024-01-02 18:02:44

Dashboard | Move | Program | Interface | **Log** | Setting

# 5.6 Setting

Enable | Project project\_1 | Install Vertical | Tool def... 0kg | Obj default | JOG 79% Joint J... | Global Speed 79% | Manual | Real | Collision L3 | Check b6ed

Tool | User | Install | Variable | Safety | System Event | Other | Plugins

TCP Add

Index	Name	Position	Mass	Centroid	Action
1	default	0,0,0,0,0,0	0	-0.15,0.09,22.13	...
2	tool1	100,100,100,0,90,0	0	0,0,0	...
3	tcp1	0,0,100,0,0,0	0	0,0,0	...

Dashboard | Move | Program | Interface | Log | Setting

2024-01-02 18:06:10



✦ Lecture 6 ✦

# Log introduction

---

## 6.1 Log introduction



Time	Level	Message
01-02-18:01:20	info	Robot state changed:Enable
01-02-18:01:16	info	Robot state changed:Disable
01-02-17:36:53	info	Program state changed:Stopped
01-02-17:36:53	info	Program state changed:Stopping
01-02-17:36:33	info	Robot state changed:Poweroff
01-02-17:36:32	info	Robot state changed:Disable
01-02-17:18:16	info	Program state changed:Paused
01-02-17:18:16	info	Program state changed:Pausing
01-02-17:00:47	info	Program state changed:Running
01-02-17:00:42	info	Program state changed:Stopped
01-02-17:00:31	info	Program state changed:TaskRunning
01-02-16:56:17	info	Program state changed:Stopped
01-02-16:55:52	info	Program state changed:Running
01-02-16:54:48	info	Program state changed:Stopped
01-02-16:54:48	error	0x00C00000 Compiler error:@/home/ducocobot/siasun_cobot/run/dist/plugins_custom/weld/scripts/api.lua:3: undefined variable   none   none
01-02-16:54:48	info	Program state changed:Running
01-02-16:54:42	info	Program state changed:Stopped
01-02-16:54:42	info	Program state changed:Stopping

Click Export Historical Logs to display the list of all historical log files. You can export one of these log files or export all logs.

**Note:** All historical log files exported to a USB flash drive (Fat32 format) are a compressed package named logs.tar



## 6.1 Log introduction



Enable Project project\_1 Install Vertical Tool def... 0kg Obj default JOG 79% Joint J... Global Speed 79% Manual Real Collision L3 Check b6ed A

AUCTECH ROBOT History logs Export Export all Current Log

safety_param.log
process_logger_2024-01-02
process_logger_2024-01-01
process_logger_2023-11-05.tar.gz
process_logger_2023-04-12.tar.gz
process_logger_2023-03-22
process_logger_2023-03-16
process_logger_2023-03-15
process_logger_2023-03-07
process_logger_2023-02-17
process_logger_2023-02-10
process_logger_2022-12-12
process_logger_2022-11-14
process_logger_2022-11-12
process_logger_2022-11-11
process_logger_2022-11-10
process_logger_2022-11-09

2024-01-02 18:16:37

There are three types of log information in the diachronic log:

1. **process\_logger** Process logging

2. **ec\_logger** Error logging

3. **brake\_test.log** Brake detection log



## 6.3 ec\_logger

```
[23-11-23-07:30:40][info]0x13000001: state changed Master state changed
[23-11-23-07:51:22][info]0x  0:
[23-11-23-07:51:22][info]0x13000001: state changed Master state changed
[23-11-23-07:51:30][info]0x  0:
[23-11-23-07:51:30][info]0x13000001: state changed Master state changed
[23-11-23-08:06:12][info]0x  0:
[23-11-23-08:08:39][info]0x13000001: state changed Master state changed
[23-11-23-08:25:34][info]0x13000001: state changed Master state changed
[23-11-23-08:34:39][info]0x13000001: state changed Master state changed
[23-11-23-08:37:02][info]0x  0:
[23-11-23-08:37:02][info]0x13000001: state changed Master state changed
[23-11-23-08:37:09][info]0x  0:
[23-11-23-08:37:09][info]0x13000001: state changed Master state changed
[23-11-23-09:28:09][info]0x13000001: state changed Master state changed
[23-11-23-09:31:26][info]0x  0:
[23-11-23-09:31:26][info]0x13000001: state changed Master state changed
[23-11-23-09:31:34][info]0x  0:
[23-11-23-09:31:34][info]0x13000001: state changed Master state changed
[23-11-23-09:40:45][info]0x13000001: state changed Master state changed
```

ec\_logger Error Logging:

The main real-time communication data is used for in-depth bug troubleshooting and analysis.



✦ Lecture 7 ✦

# Multi-terminal connection

---

## ◆ 7.1 Multi-terminal connection - wired connection

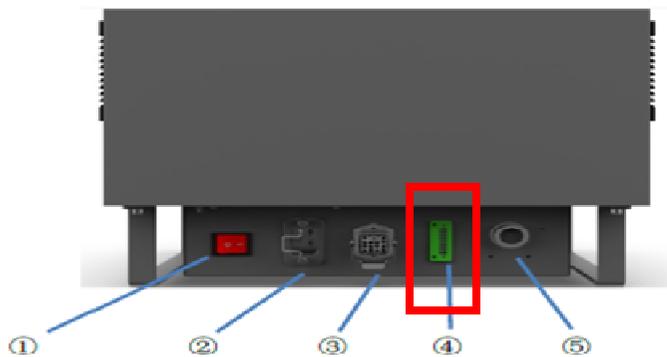
Users can **connect the robot by using PC or mobile device** with wired or wireless.

Wired Connection:

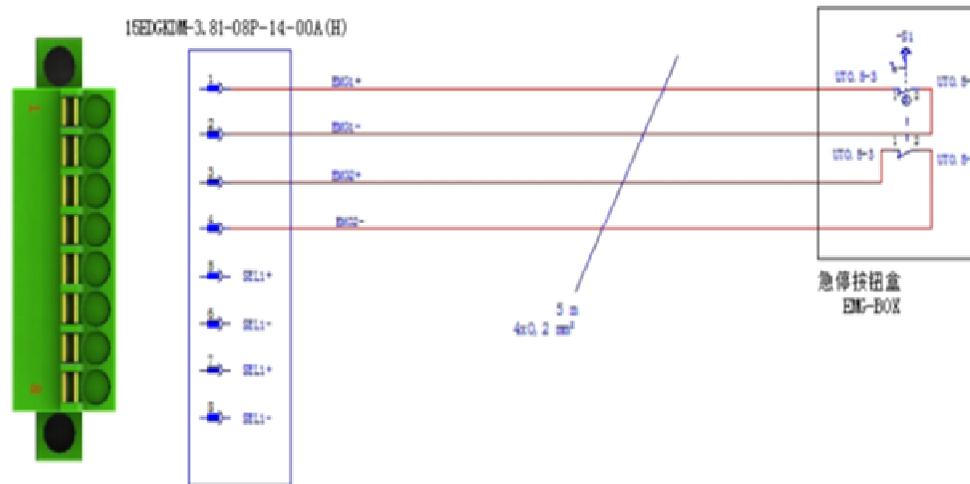
1. **Connect the network cable to the "LAN" interface** on the control cabinet, and the other end to the computer network port.
2. Set the robot IP according to the requirements, for example: the robot IP is set to: 192.168.1.10.  
(**Enter the system control - network settings page to view the IP address of the robot**)
3. Set the computer IP to be in the same network segment as the robot IP, for example, the computer IP is set to: 192.168.1.11.
4. Ping the robot with the computer to test the network on/off.
5. Open a browser, it is recommended to use Google Chrome (Chrome), and then enter the IP address and **port number 7000** just set by the robot in the address bar, **such as 192.168.1.10:7000 or http:AUCTECH-cobot.com:7000**, enter after the input is completed.

## 7.2 Multi-terminal connection - wireless connection

When using a mobile device (PAD or PC) to **wirelessly connect the robot**, It is possible to leave the teach pendant unattached, but at this time **the robot safety loop is in a disconnected state**, and the **emergency stop button needs to be connected at the shield interface of the teach pendant**.



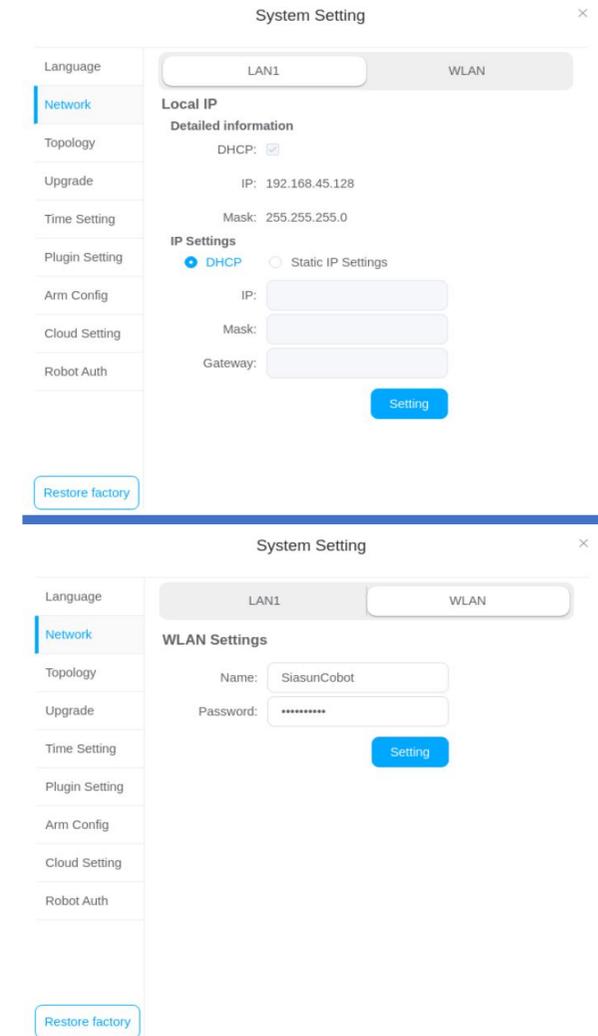
- ①: System power switch
- ②: AC input interface
- ③: Robotic arm connection interface
- ④: Shielded teach pendant interface
- ⑤: Teach pendant connection interface



## 7.2 Multi-terminal connection - wireless connection

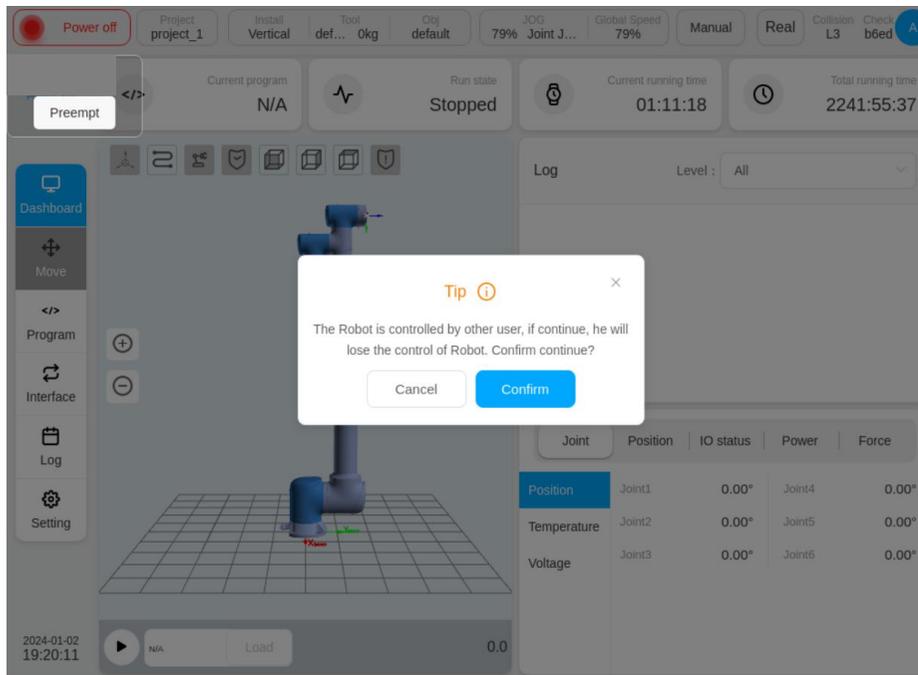
### Connection Steps:

1. Start the power supply of the control cabinet, use the mobile device to **find the wireless network of the robot**, according to the name of the wireless network of the robot control cabinet, click "Connect", and enter the **default password: 1234567890**.
2. After the mobile device is successfully connected to the wireless network of the robot control cabinet, open the browser (Google Chrome is recommended), **enter [http:AUCTECH-cobot.com:7000](http://AUCTECH-cobot.com:7000) in the address bar, or [192.168.117.66:7000](http://192.168.117.66:7000) (wireless port IP address) to confirm and connect to the robot.**



The image shows two screenshots of the 'System Setting' interface. The top screenshot displays the 'Local IP' configuration for LAN1, with fields for IP (192.168.45.128), Mask (255.255.255.0), and IP Settings (DHCP selected). The bottom screenshot displays the 'WLAN Settings' configuration, with fields for Name (SiasunCobot) and Password (masked with asterisks).

## 7.3 Multi-terminal connection



The system supports multiple terminals to be connected at the same time, but only one terminal has the right to control, which can run the program, configure the system and other functions. Other terminals only have the permission to view, and cannot run the robot arm or modify the setting information

When other terminals want to take control, you can click the "preempt" button in the upper left corner. The following dialog box pops up, click the "OK" button to obtain control of the system, and the terminal with control will lose control.

## Part II

# Introduction to software operation

1. **Mobile interface**
2. **Programming**
3. **API introduction**
4. **Set up the information**
5. **Security settings**
6. **Description of the error**



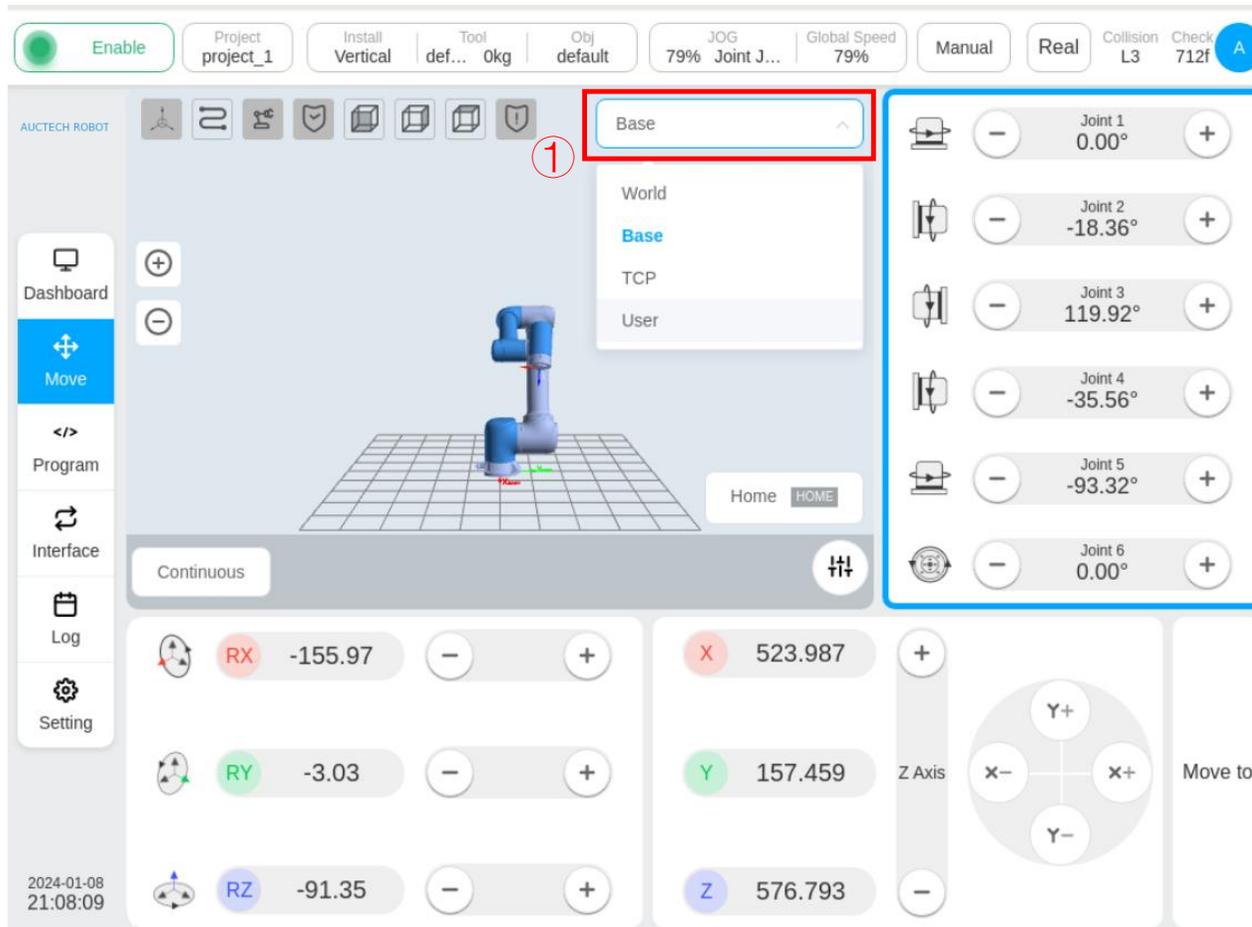


✦ Lecture 1 ✦

# Move Interface

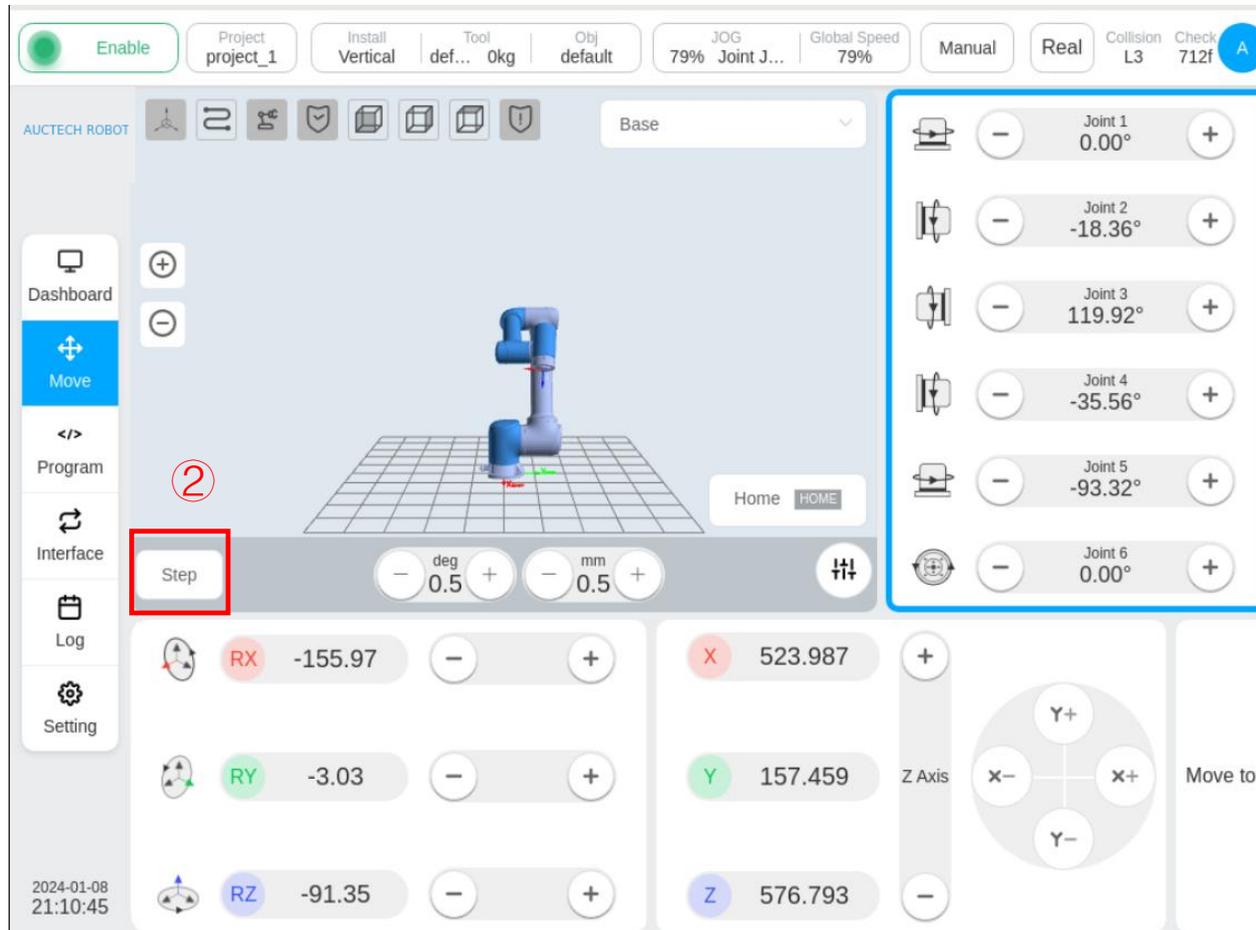
---

## ◆ 1.1 Coordinate system selection



logo①, You can select the coordinate system used by the current robot and the reference coordinate system of the manually operated robot .

## ◆ 1.2 Sport mode selection



logo@button to switch between "continuous mode" or "step mode" mode.

The range of settings for step angle and step distance in Step Mode:

The default values are 0.5deg and 0.5mm

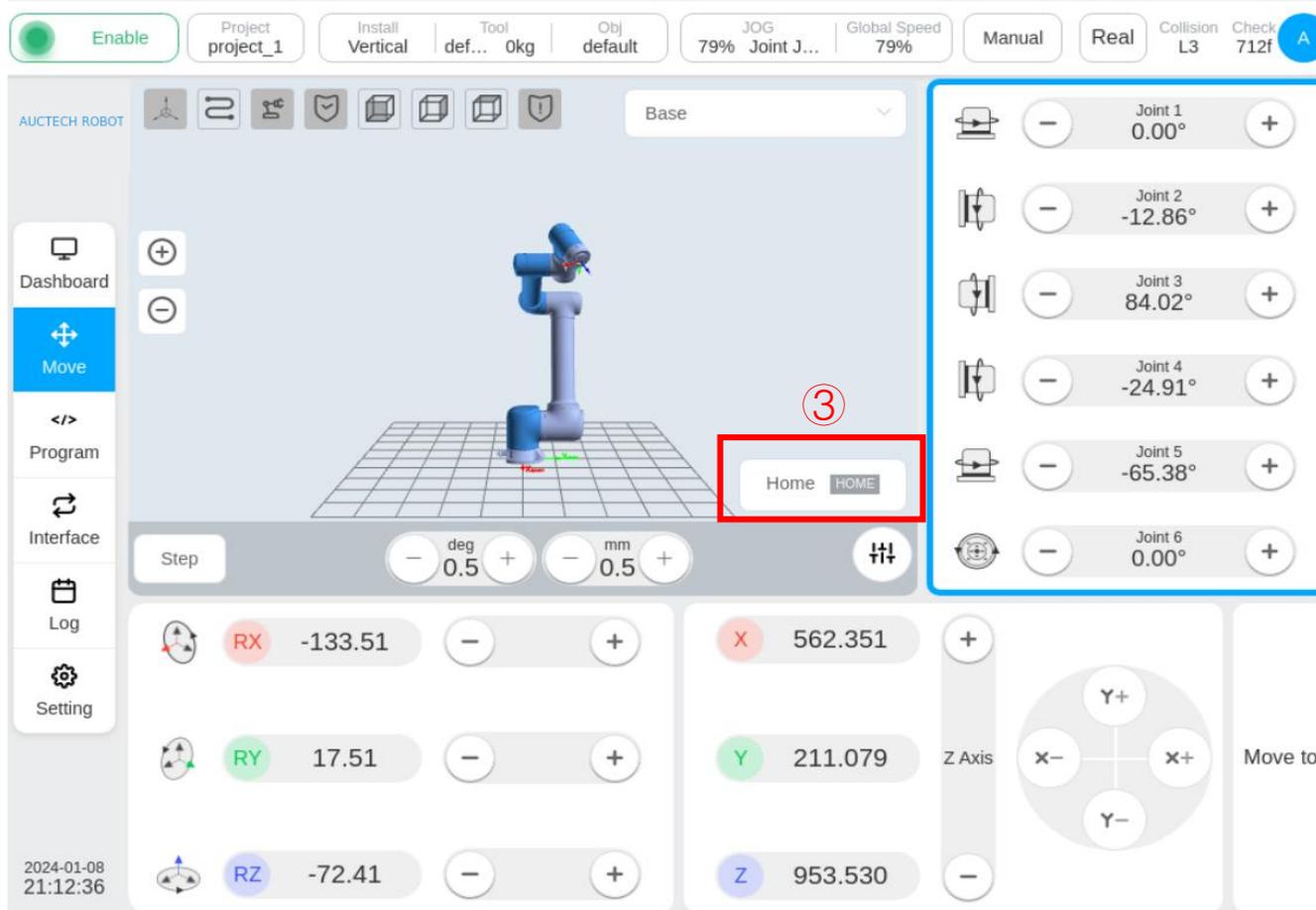
The minimum allowable settings are

**0.1deg and 0.1mm**

The maximum allowable setting is

**5deg and 5mm**

## 1.3 Move to home point



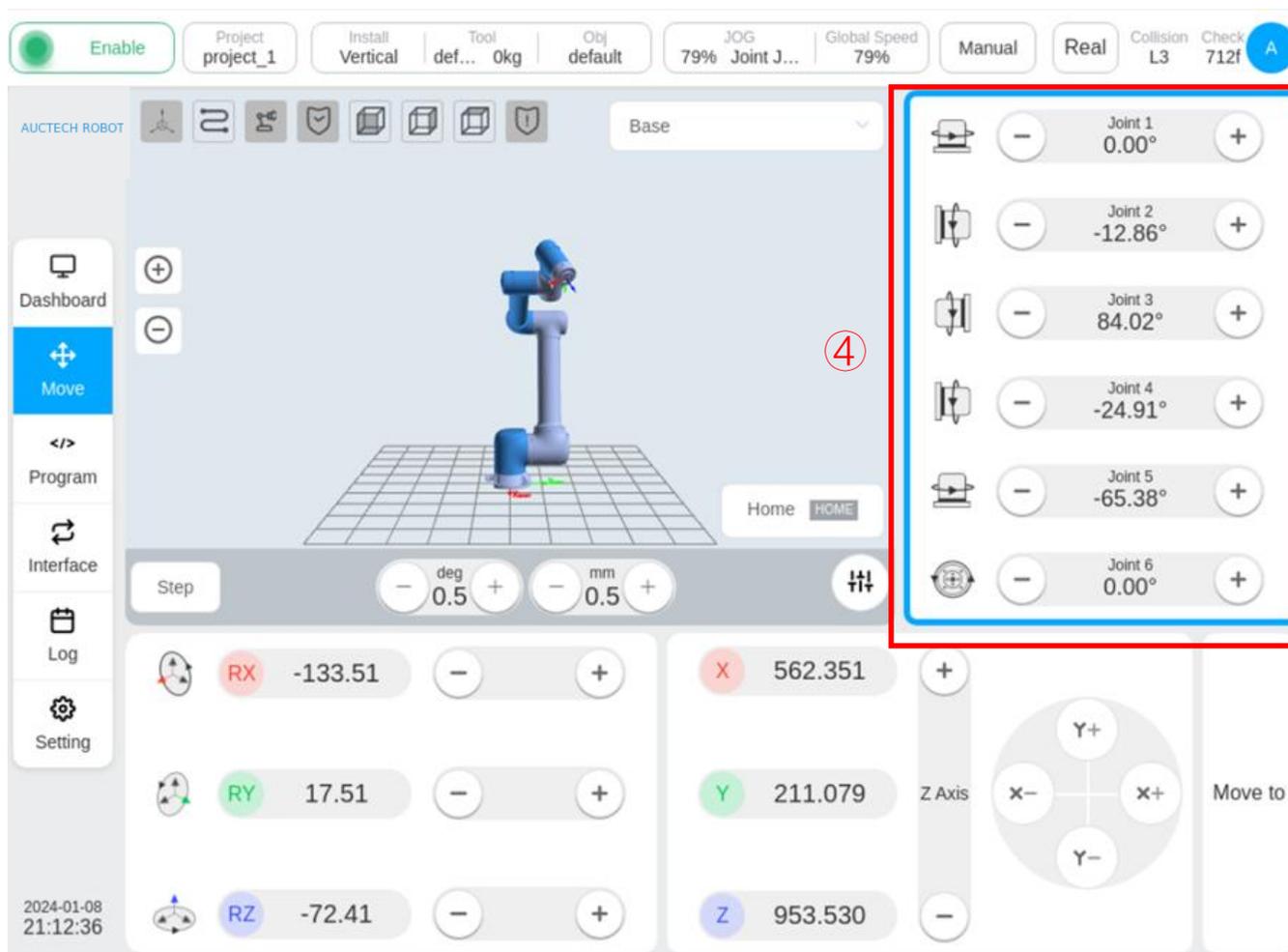
logo③, Pressing and holding the button will move the robot to the set HOME point.

button color

Green at the HOME point

Grey is not at the HOME point

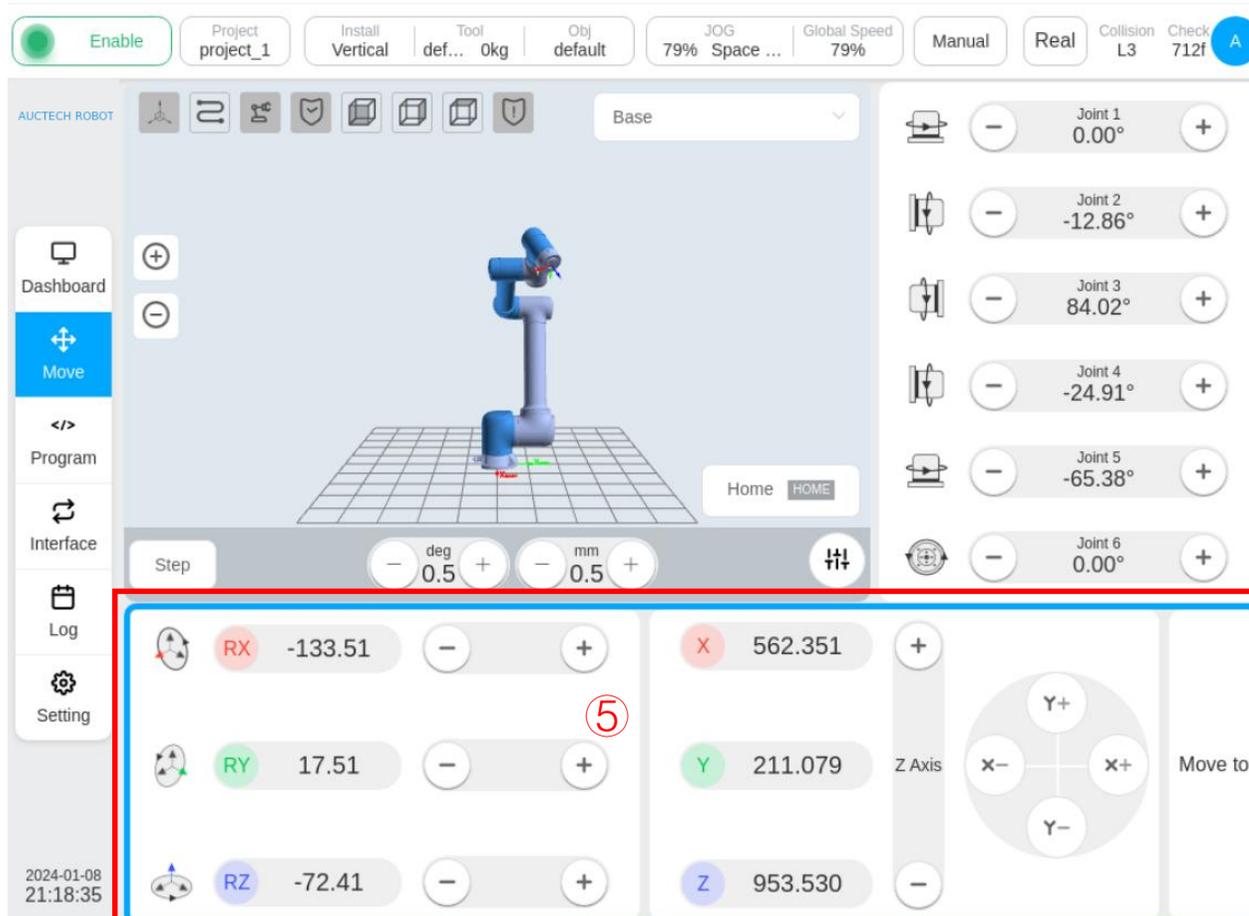
## ◆ 1.4 Joint move



logo ④, Manually operate the area where the robot's joints move.

When you click on a joint to move any area, a blue border appears in that area to select for joint movement.

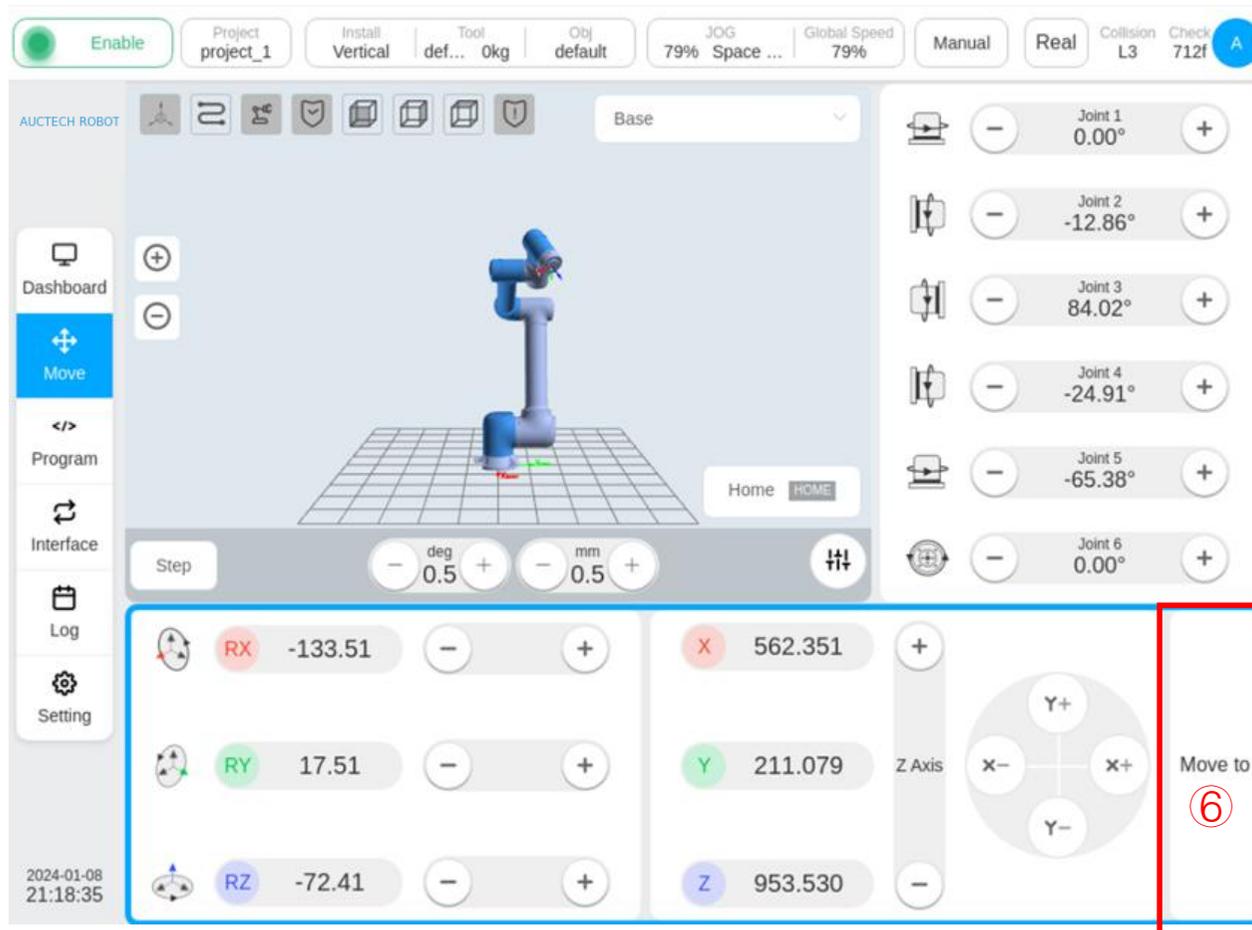
## ◆ 1.5 End movement



logo⑤, Manually operate the area where the end of the robot moves.

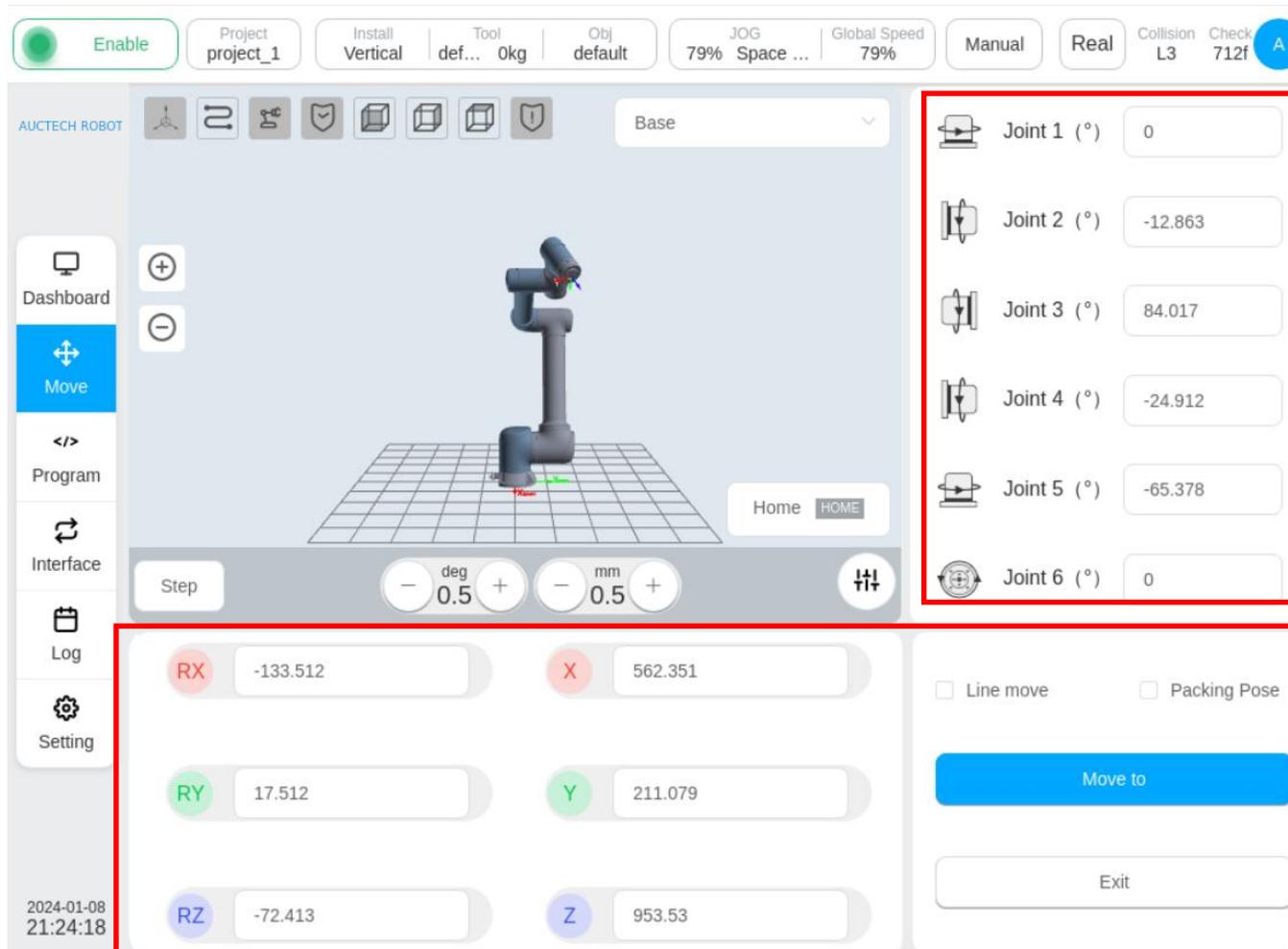
When you click on the end to move any area, a blue border appears in that area, and the end move is selected.

## ◆ 1.6 Move to



logo⑥, The user manually enters the **joint angle or pose value** and holds the button to operate the robot to move .

## ◆ 1.6 Move to



The screenshot displays the AUCTECH robot control interface. The top status bar shows 'Enable', 'Project project\_1', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG 79% Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check 712f'. The main area shows a 3D simulation of a robotic arm on a grid. A sidebar on the left contains 'Dashboard', 'Move', 'Program', 'Interface', 'Log', and 'Setting'. The 'Move to' screen features a list of joint angles (Joint 1 to Joint 6) with input fields, a 'Move to' button, and an 'Exit' button. A bottom panel shows coordinate inputs for RX, RY, RZ, X, Y, Z.

Joint	Angle (°)
Joint 1	0
Joint 2	-12.863
Joint 3	84.017
Joint 4	-24.912
Joint 5	-65.378
Joint 6	0

Coordinate	Value
RX	-133.512
RY	17.512
RZ	-72.413
X	562.351
Y	211.079
Z	953.53

After entering the joint angle or pose value, the position of the input is represented by a ghost in the simulation diagram. After confirming that the location is correct. Long press the "Move to" button.

**Note:** The movement speed of the robotic arm should not be too fast.

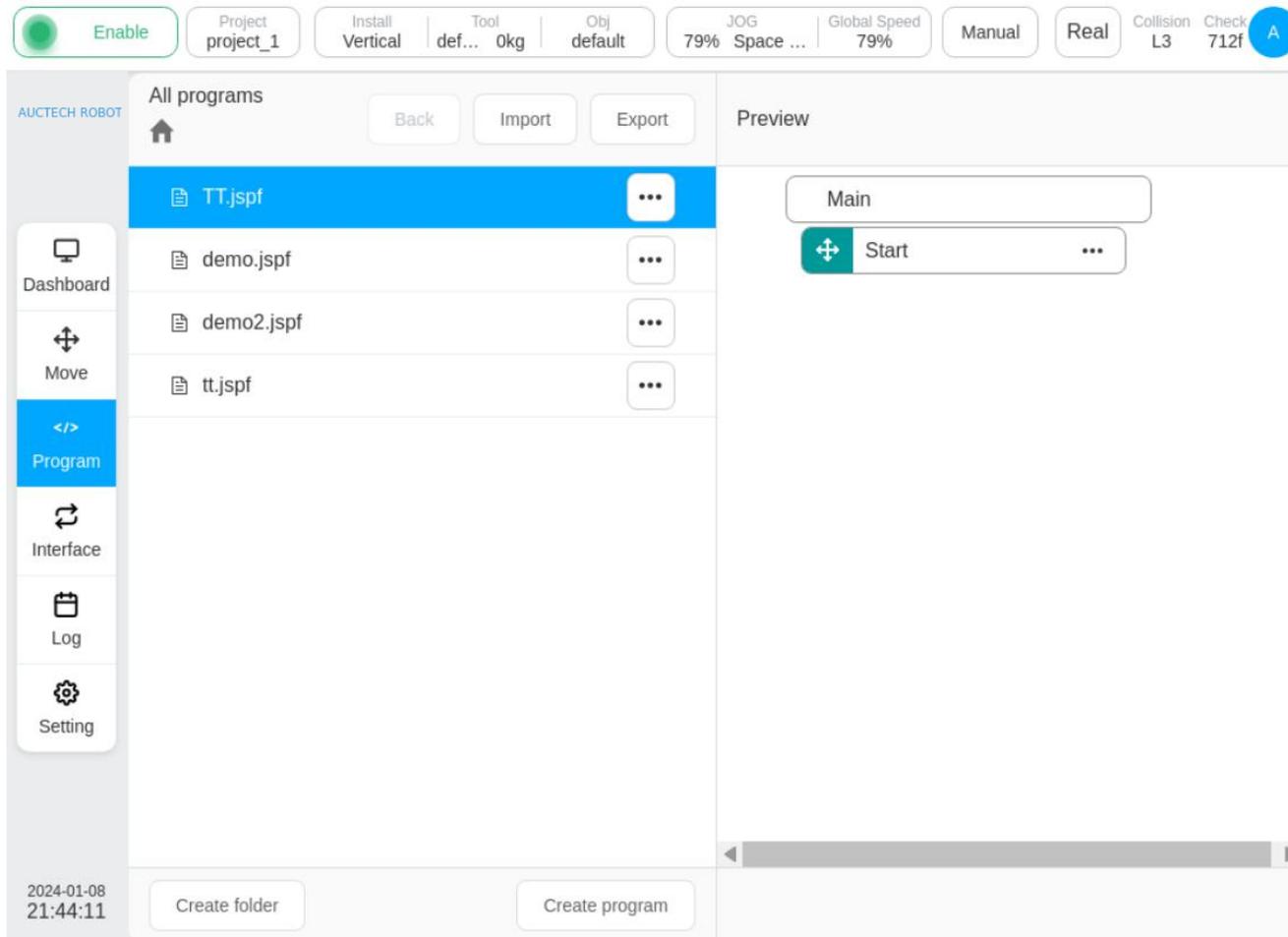


✦ Lecture 2 ✦

# Programming

---

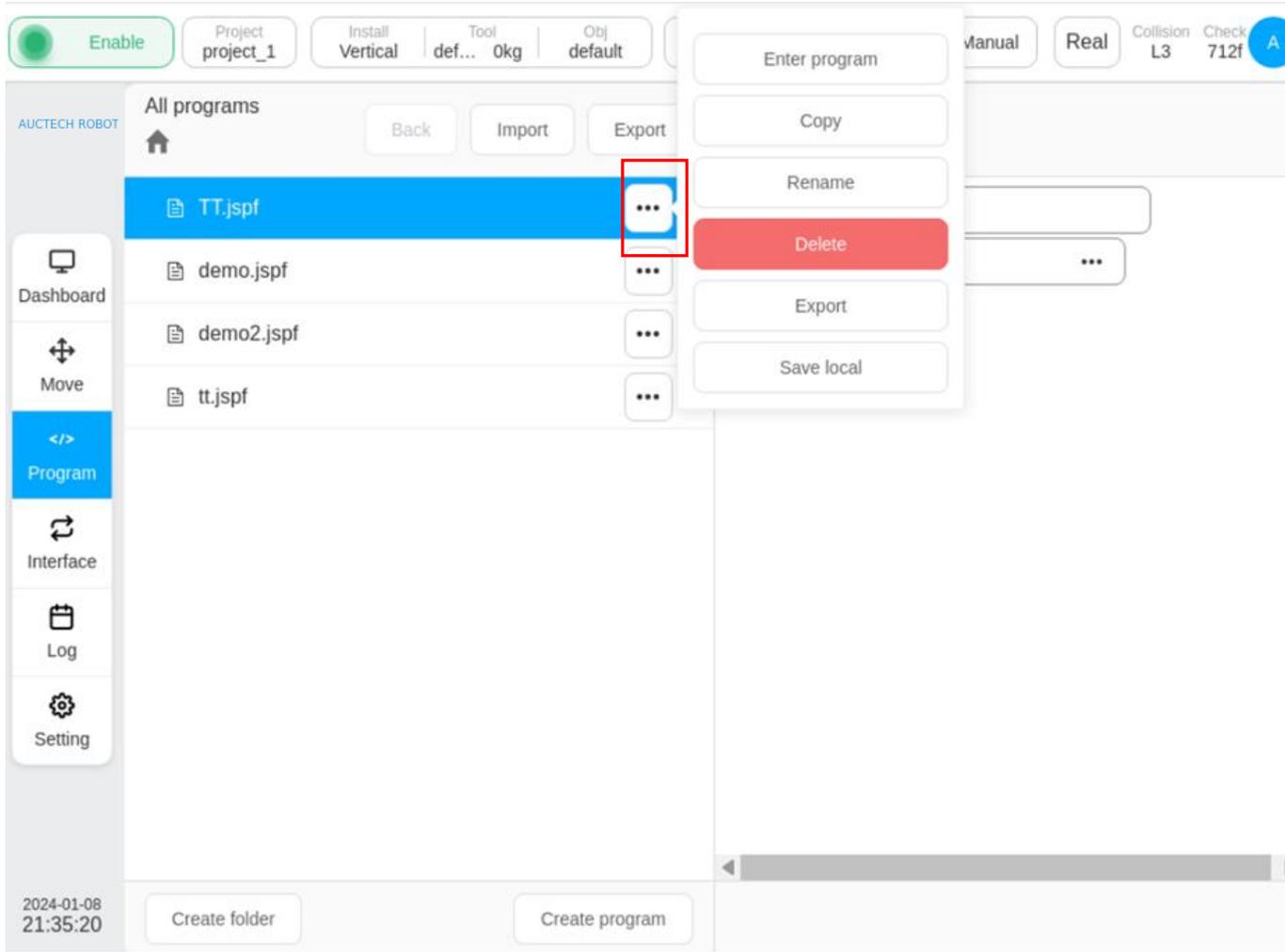
## 2.1 List of programs



The screenshot displays the AUCTECH ROBOT software interface. At the top, there is a status bar with various controls: a green 'Enable' button, 'Project project\_1', 'Install Vertical', 'Tool def...', 'Obj 0kg default', 'JOG 79% Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', 'Check 712f', and a blue 'A' button. The main interface is divided into a left sidebar and a main content area. The sidebar contains icons for 'Dashboard', 'Move', 'Program' (highlighted in blue), 'Interface', 'Log', and 'Setting'. The main content area is titled 'All programs' and features a list of programs: 'TT.jspf', 'demo.jspf', 'demo2.jspf', and 'tt.jspf'. Each program entry has a three-dot menu icon to its right. The 'TT.jspf' entry is selected and highlighted in blue. To the right of the list is a 'Preview' window showing a tree view with 'Main' and 'Start' (with a plus icon and a three-dot menu icon). At the bottom left, the date and time '2024-01-08 21:44:11' are displayed. At the bottom center, there are two buttons: 'Create folder' and 'Create program'.

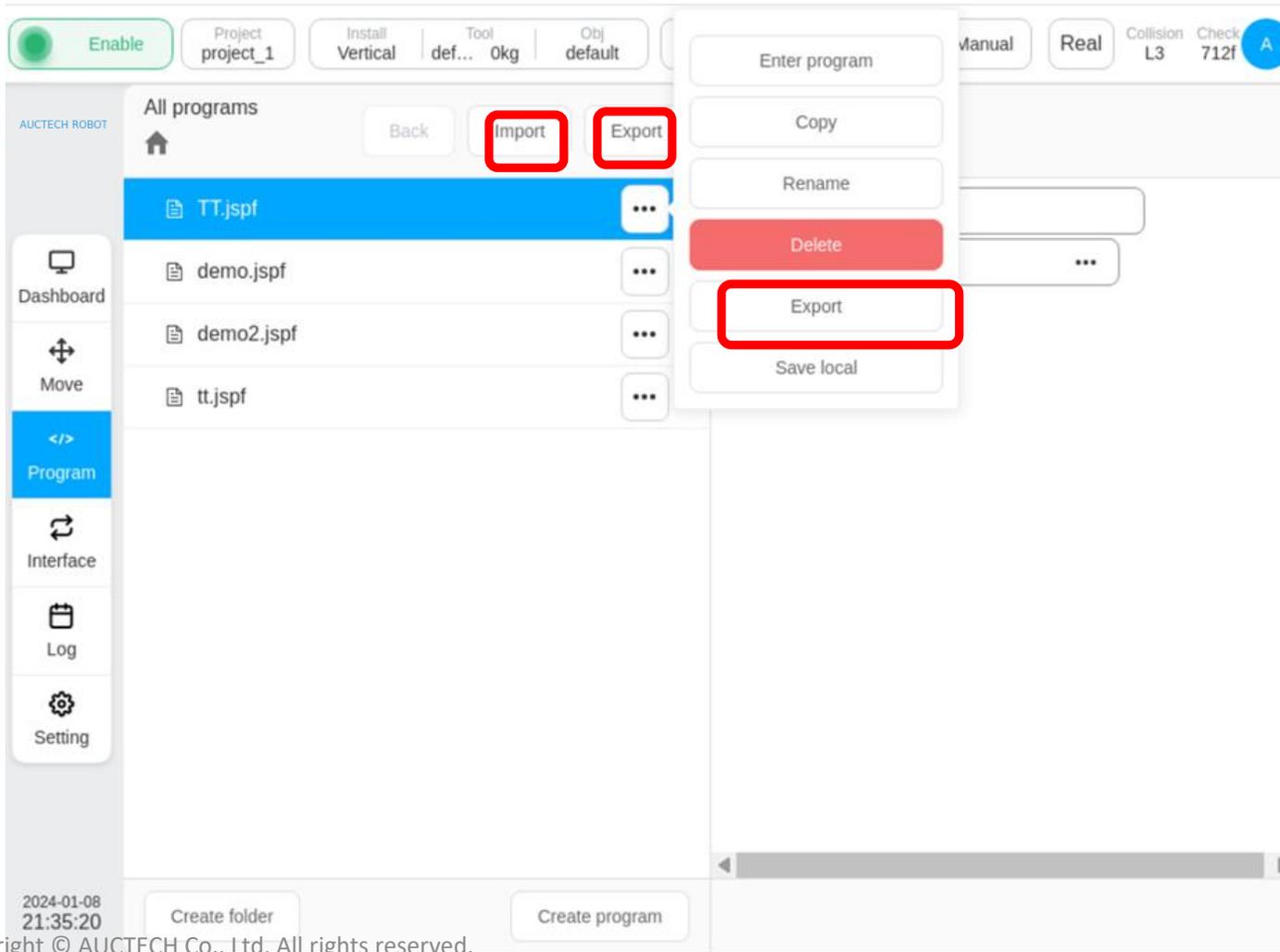
Select: Double-click or check, click "Enter Program" to enter the folder or program.

## 2.1 List of programs



Click on the "...", at the back of the program, to select different operations on the program.

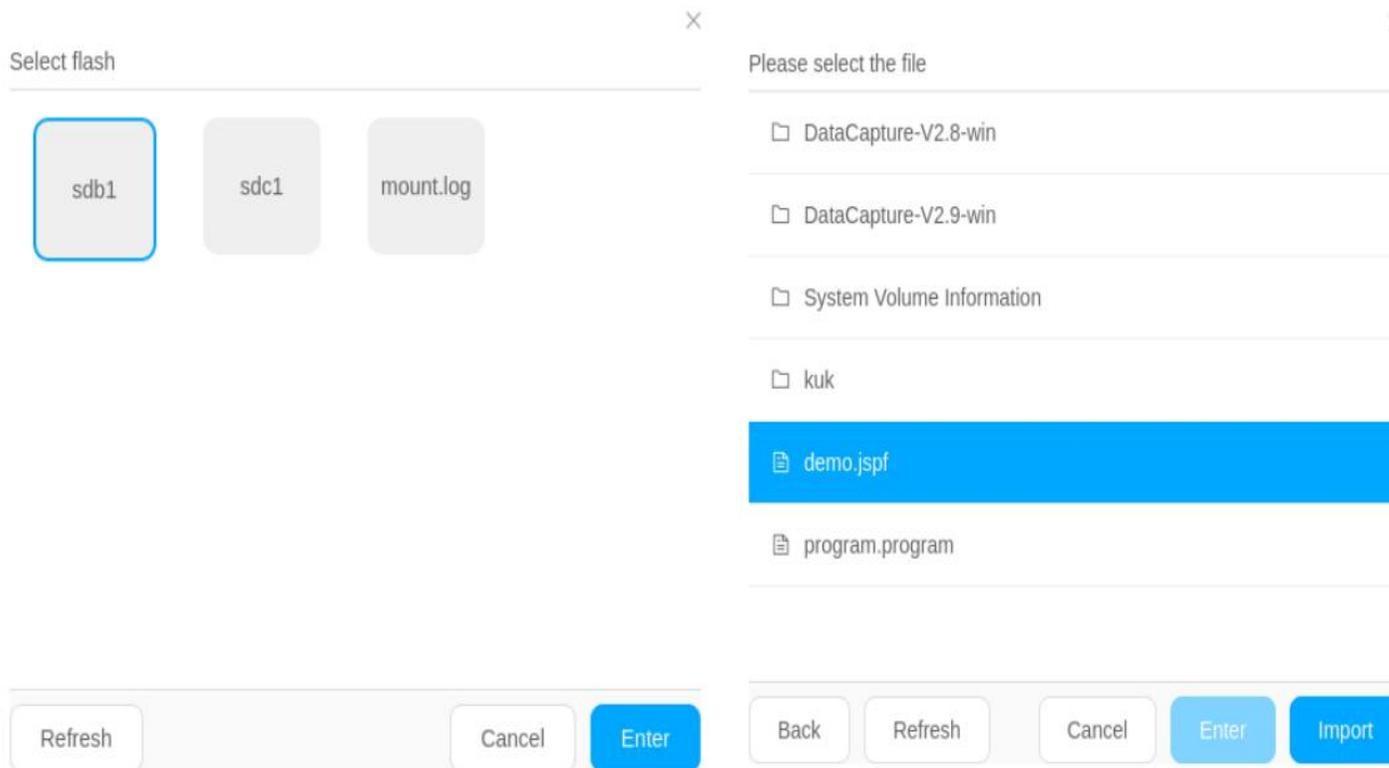
## ◆ 2.2 Importing and exporting programs



The screenshot displays the AUCTECH ROBOT software interface. At the top, there is a status bar with 'Enable' (green), 'Project project\_1', 'Install Vertical', 'Tool def...', 'Obj 0kg', 'Obj default', 'Manual', 'Real', 'Collision L3', 'Check 712f', and a blue 'A' button. Below this is the 'All programs' section, which includes a 'Back' button and 'Import' and 'Export' buttons (both highlighted with red boxes). The program list contains 'TT.jspf' (selected), 'demo.jspf', 'demo2.jspf', and 'tt.jspf'. A context menu is open for 'TT.jspf', showing options: 'Enter program', 'Copy', 'Rename', 'Delete', 'Export' (highlighted with a red box), and 'Save local'. The left sidebar contains 'Dashboard', 'Move', 'Program' (selected), 'Interface', 'Log', and 'Setting'. The bottom status bar shows the date '2024-01-08 21:35:20' and 'Create folder' and 'Create program' buttons.

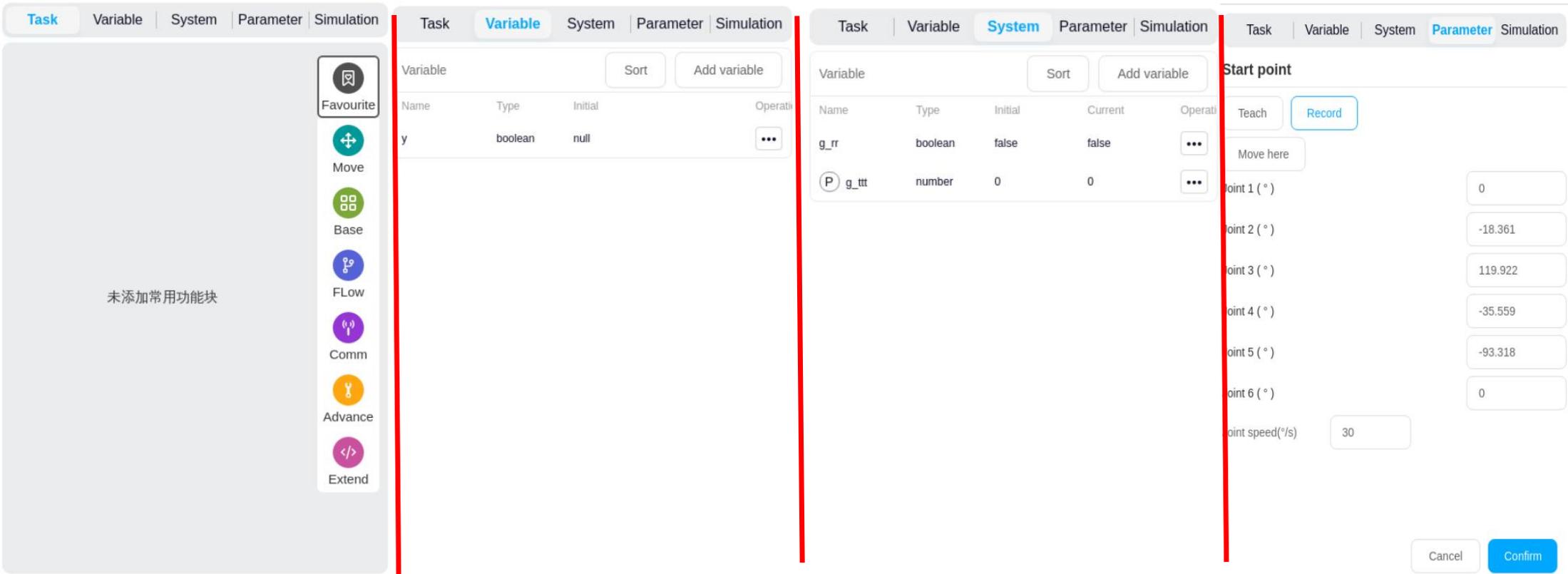
**Note:** The USB flash drive used for import and export **must be in FAT32** format

## ◆ 2.2 Importing and exporting programs



If the name of the program file on the control cabinet is the same as that of the control cabinet during import, the rename keyboard will pop up, and the renamed file will be imported into the control cabinet (**the suffix is jspf**)

## 2.3 programming



The screenshot shows the software interface with four tabs: Task, Variable, System Variable, and Parameter. Each tab displays a table of variables and their properties.

**Task Tab:** Shows a table with columns Name, Type, Initial, and Operati. A sidebar on the left contains icons for Favourite, Move, Base, Flow, Comm, Advance, and Extend. The text "未添加常用功能块" (No common function blocks added) is visible.

**Variable Tab:** Shows a table with columns Name, Type, and Initial. The variable "y" is listed with Type "boolean" and Initial "null".

**System Variable Tab:** Shows a table with columns Name, Type, Initial, Current, and Operati. Two variables are listed: "g\_rr" (boolean, false) and "g\_ttt" (number, 0).

**Parameter Tab:** Shows a "Start point" section with a "Teach" button and a "Record" button. Below are input fields for "Move here" and "Joint 1 (°)" through "Joint 6 (°)", and a "Joint speed(/s)" field set to 30. "Cancel" and "Confirm" buttons are at the bottom.

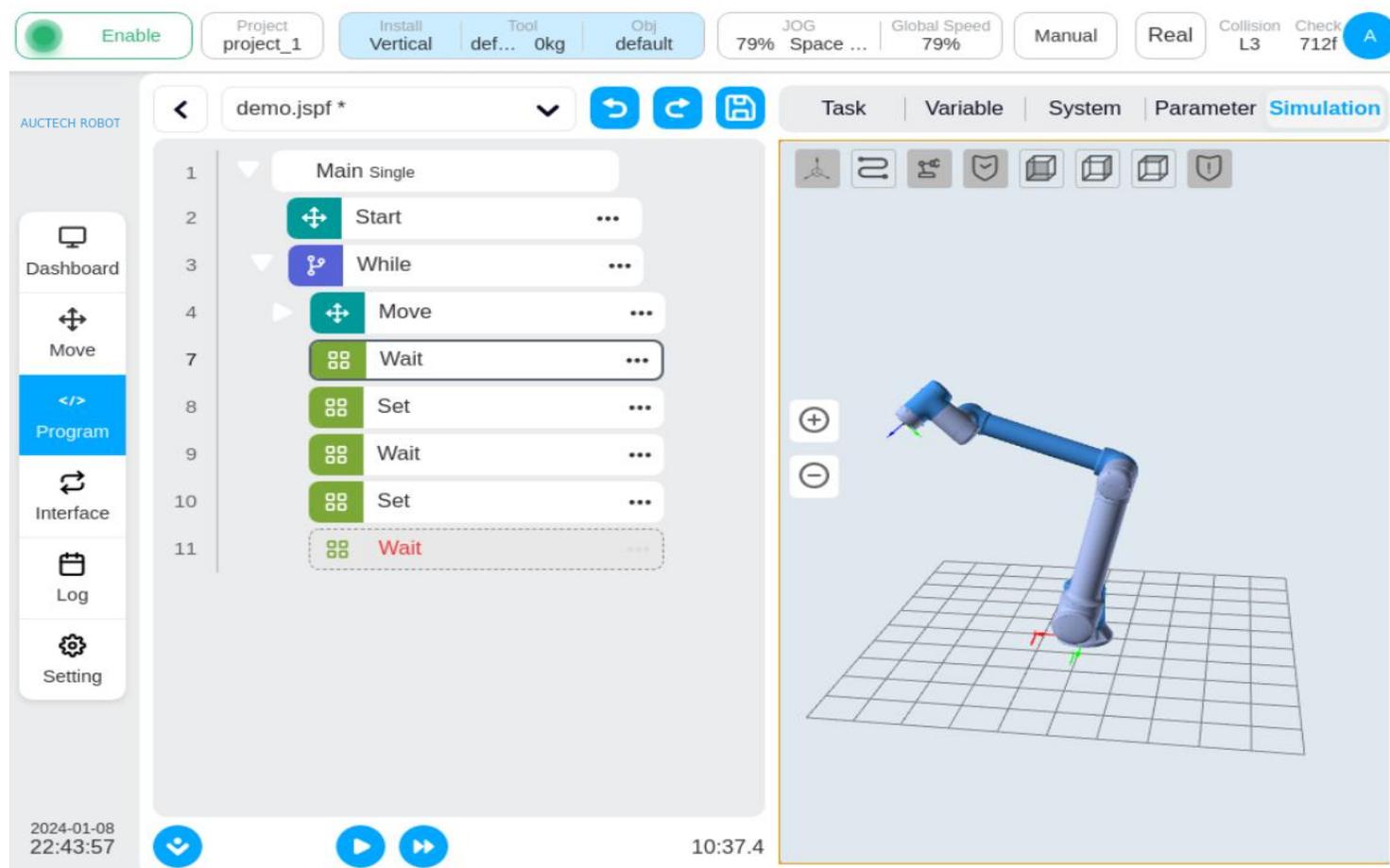
Task

Variable

System Variable

Parameter

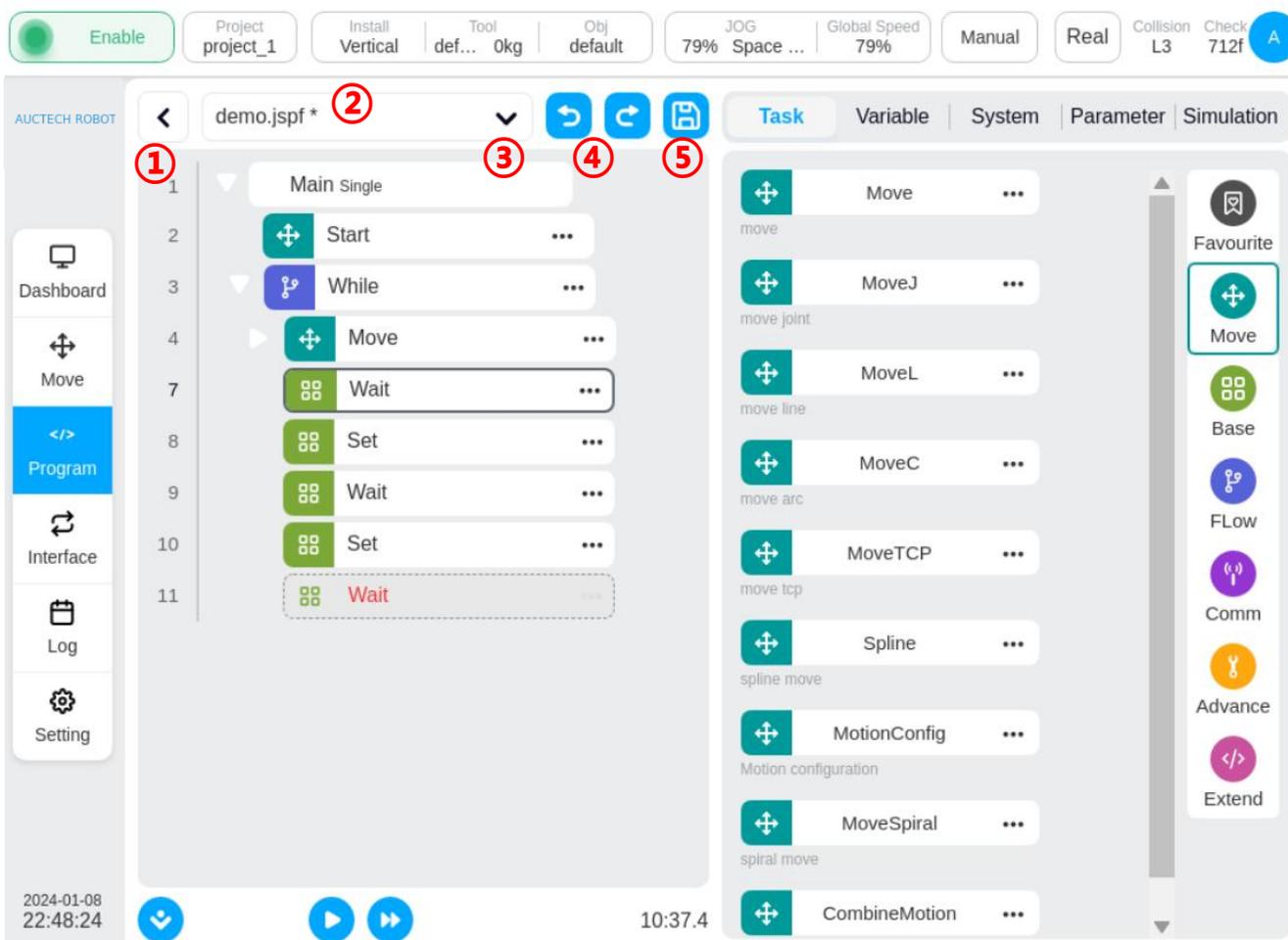
## 2.3 programming



The screenshot displays the AUCTECH ROBOT programming software interface. At the top, there is a status bar with an 'Enable' button, project name 'project\_1', tool settings, and various control buttons like 'Manual', 'Real', and 'Simulation'. The main workspace is divided into a left sidebar with navigation options (Dashboard, Move, Program, Interface, Log, Setting) and a central area for programming. The programming area shows a sequence of steps: 1. Main Single, 2. Start, 3. While loop, 4. Move, 7. Wait, 8. Set, 9. Wait, 10. Set, and 11. Wait. The right side of the interface features a 3D simulation of a blue robotic arm on a grid floor, with a toolbar for simulation controls and a time display of 10:37.4.

## Simulation

## 2.3 programming



The screenshot displays the AUCTECH robot programming interface. At the top, there is a status bar with various controls: 'Enable' (green), 'Project project\_1', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG 79% Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', 'Check 712f', and a blue 'A' button. Below this, the main interface is divided into several sections. On the left, there is a sidebar with icons for 'Dashboard', 'Move', 'Program' (highlighted in blue), 'Interface', 'Log', and 'Setting'. The central area shows a task list for 'demo.jspf \*' (indicated by a red circle 2). The list includes: 1. Main Single (expanded), 2. Start, 3. While, 4. Move, 7. Wait, 8. Set, 9. Wait, 10. Set, and 11. Wait. A red circle 1 is placed over the left sidebar, a red circle 3 is over the task list header, a red circle 4 is over the undo button, and a red circle 5 is over the save button. On the right, there is a 'Task' palette with various motion tasks: Move, MoveJ, MoveL, MoveC, MoveTCP, Spline, MotionConfig, MoveSpiral, and CombineMotion. A 'Favourite' list on the far right includes Move, Base, FLOW, Comm, Advance, and Extend. At the bottom left, the date and time '2024-01-08 22:48:24' are shown, and at the bottom center, the time '10:37.4' is displayed.

logo①>Returns the list of programs

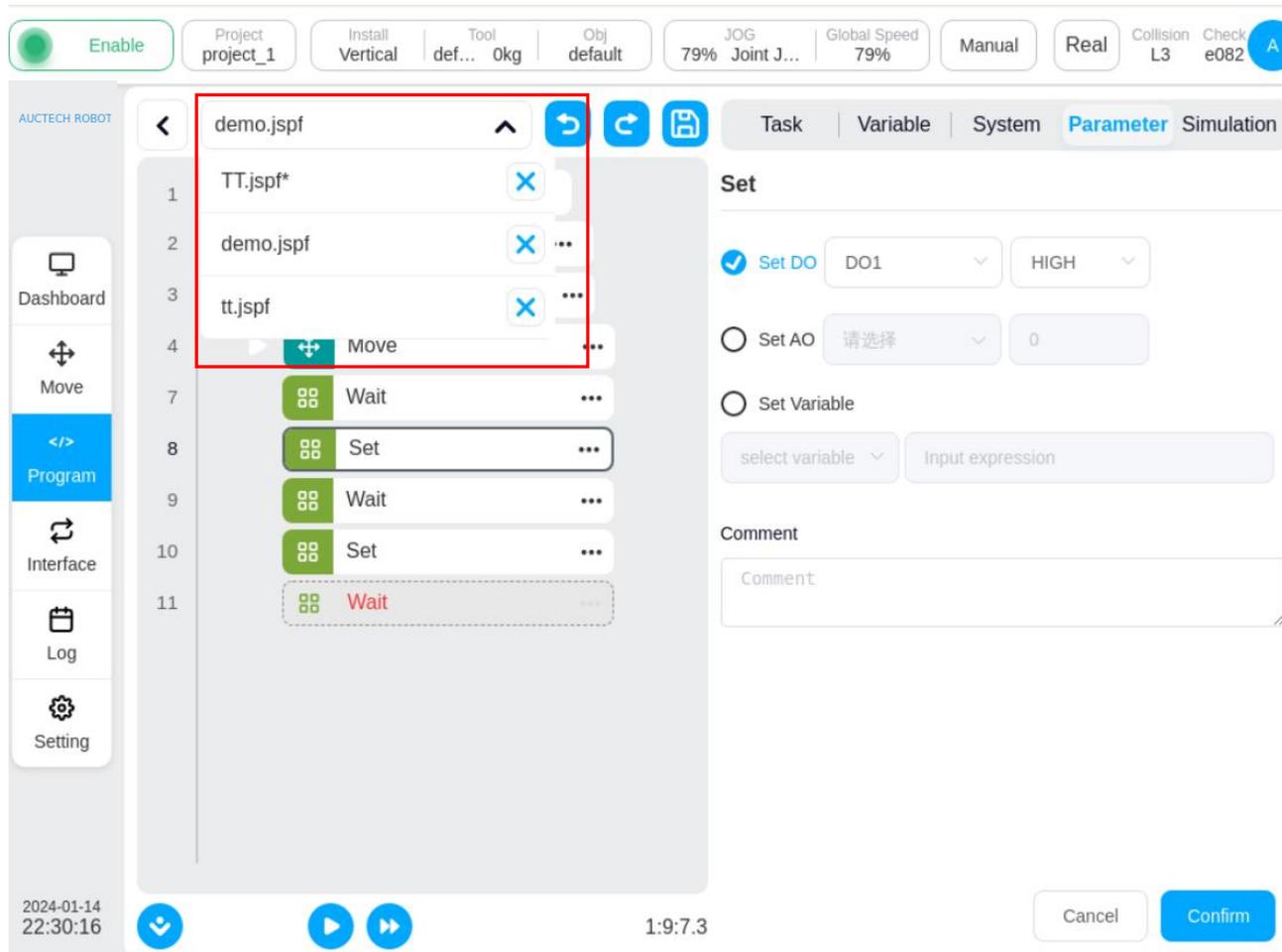
logo② :indicates the current program name

logo③:Expand the list of open programs

Logo④:Undo the changes of the current program

Logo⑤:Save the current program

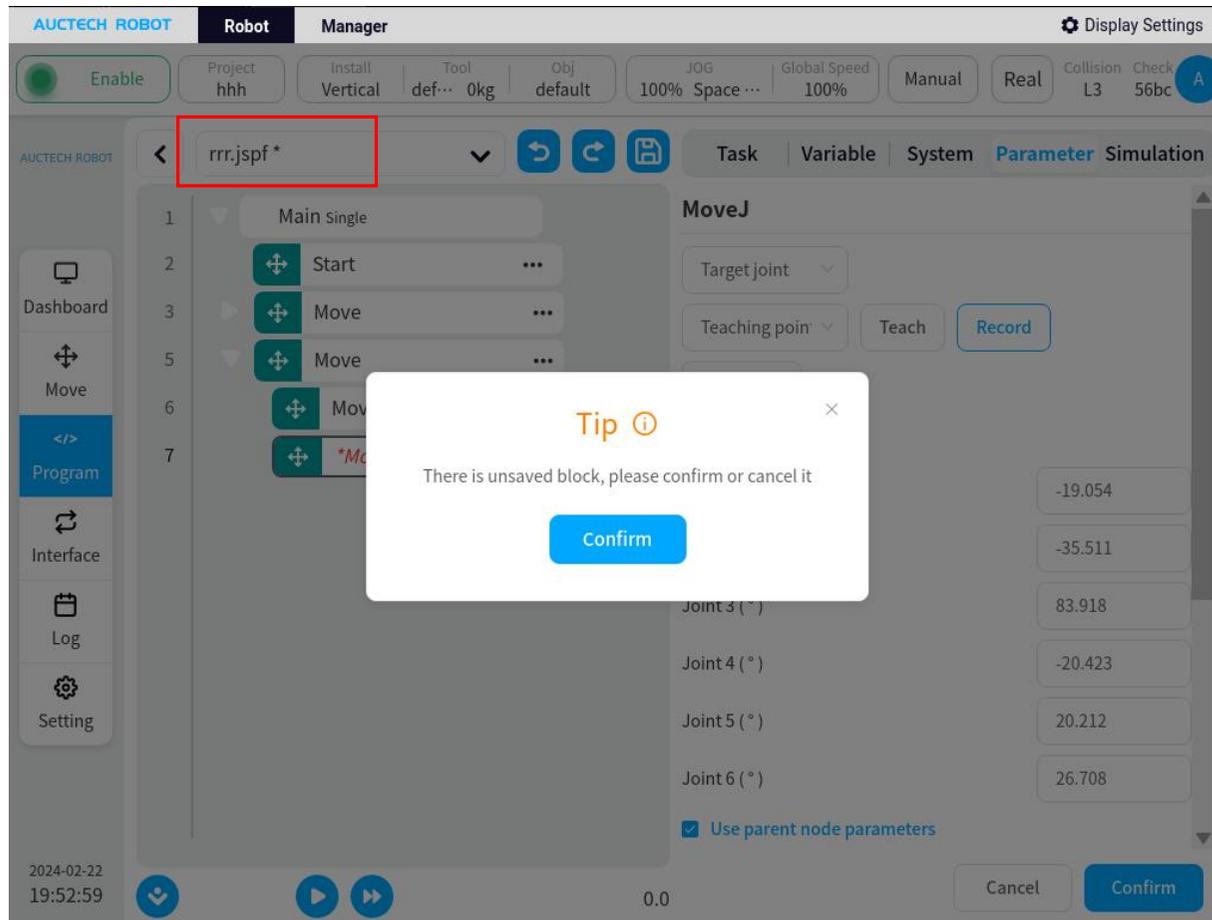
## 2.3 programming



The screenshot displays the AUCTECH robot programming software interface. At the top, there is a status bar with various indicators: 'Enable' (green), 'Project project\_1', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG 79% Joint J...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', 'Check e082', and a blue 'A' button. Below this, a navigation menu includes 'Task', 'Variable', 'System', 'Parameter' (selected), and 'Simulation'. The main workspace shows a list of programs on the left, with 'demo.jspf' highlighted in a red box. The right side of the workspace is a 'Set' configuration panel with options for 'Set DO' (checked, DO1, HIGH), 'Set AO' (unchecked, 请选择, 0), and 'Set Variable' (unchecked, select variable, Input expression). A 'Comment' field is also present. At the bottom, there are 'Cancel' and 'Confirm' buttons. The bottom status bar shows the date '2024-01-14', time '22:30:16', and a timer '1:9:7.3'.

Click on the list of open programs and **double-click on the program name to switch to that program.**

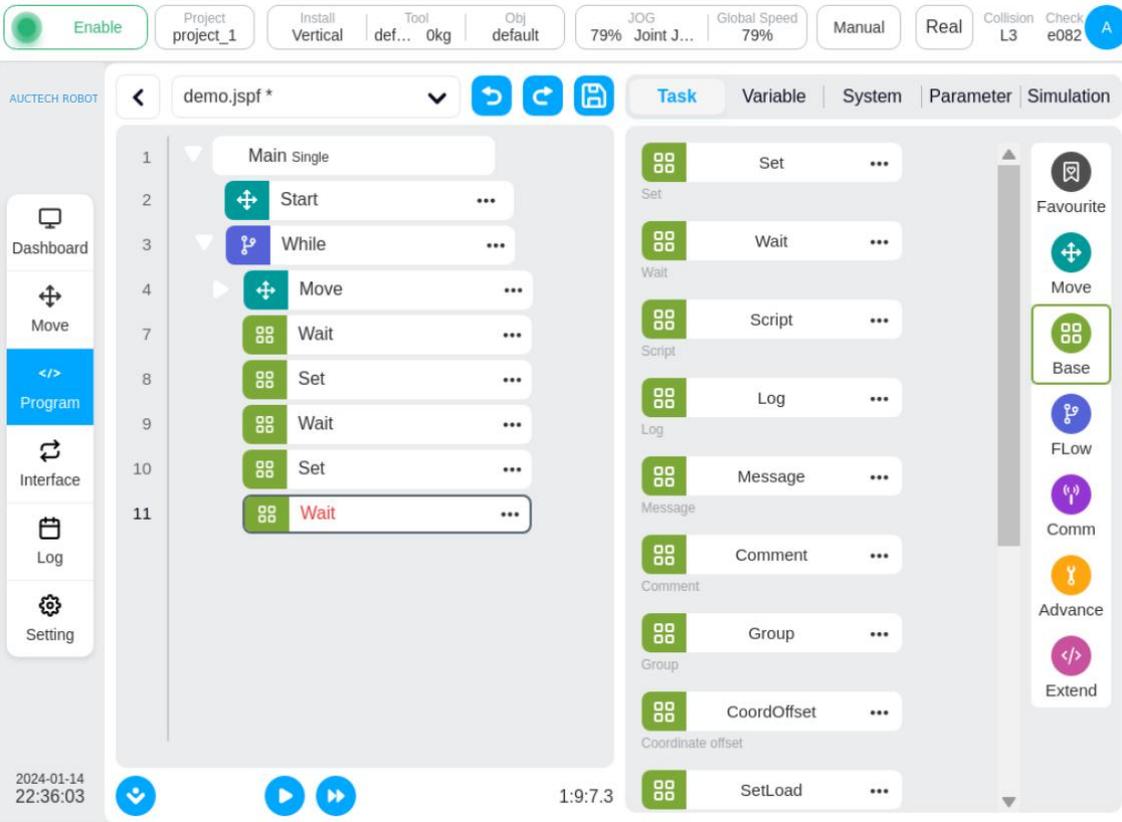
## 2.3 programming



If there are changes and the program is not saved, it is indicated by an \* sign after the program name.

If you select "Close Program", a dialog box will pop up prompting you to save the program or discard the changes

## 2.3 programming



The screenshot displays the AUCTECH programming software interface. At the top, there is a status bar with various indicators: 'Enable' (green), 'Project project\_1', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG 79%', 'Joint J...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check e082'. Below this, the main workspace is titled 'demo.jspf \*' and contains a task list with the following items:

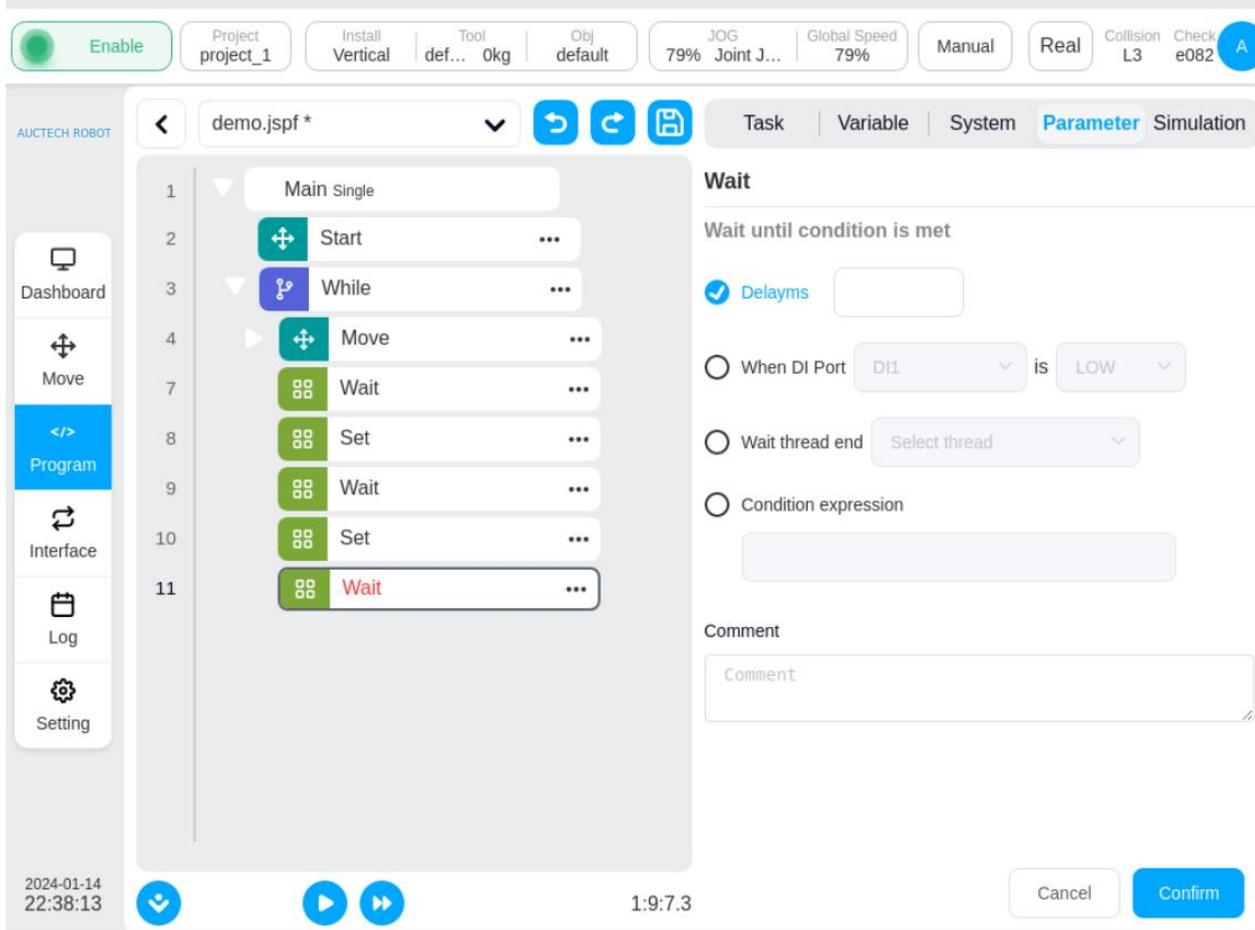
- 1 Main Single
- 2 Start
- 3 While
- 4 Move
- 7 Wait
- 8 Set
- 9 Wait
- 10 Set
- 11 Wait

To the right of the task list is a block palette with various function blocks, including 'Set', 'Wait', 'Script', 'Log', 'Message', 'Comment', 'Group', 'CoordOffset', and 'SetLoad'. A 'Base' block is highlighted in the palette. The interface also includes a sidebar with navigation options like 'Dashboard', 'Move', 'Program', 'Interface', 'Log', and 'Setting'. At the bottom left, the date and time are shown as '2024-01-14 22:36:03', and at the bottom center, the time '1:9:7.3' is displayed.

How to add function blocks:

1. **Drag and drop** the function block to the specified position.
2. **Double-click** the function block to add to add

## 2.3 programming



The screenshot displays the AUCTECH ROBOT programming environment. The top status bar shows 'Enable', 'Project project\_1', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG 79%', 'Joint J...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', 'Check e082', and a blue 'A' button. The main interface is divided into a left sidebar with 'Dashboard', 'Move', 'Program' (selected), 'Interface', 'Log', and 'Setting'. The central workspace shows a ladder logic program for 'demo.jspf \*' with a 'Main Single' block containing a sequence of blocks: Start, While, Move, Wait, Set, Wait, Set, and Wait. The right panel is titled 'Wait' and shows configuration options: 'Wait until condition is met', 'Delays' (checked), 'When DI Port DI1 is LOW', 'Wait thread end' (Select thread), and 'Condition expression'. A 'Comment' field is also present. At the bottom, there are 'Cancel' and 'Confirm' buttons.

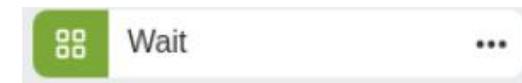
Configure the parameters of the function block:

(1): **Double-click** the function block

(2): **Select the function block** and **switch to the parameter** configuration area

## ◆ 2.3 programming

The function block in the program tree has the following three states, **red indicates that the function block is not configured with valid parameters; The italics plus \* indicate that the parameters of the function block have been changed**, but they have not been saved to the program tree. The white body indicates that the function block is valid and unchanged



Click on the right side of the function block in the program tree ... The following operation dialog box is displayed, including the following operations:

**Cut:** Cut the function block.

**Copy:** Copies the function block.

**Paste:** You can paste the copied function block below the selected function block.

**Disable/Enable:** Click to choose whether to enable the function block, when you choose not to enable, the function block will be grayed out, and the program will not execute the function block.

**Batch Operation:** Multiple rows can be selected. **Delete:** You can delete the function block.

**Set as the Start block:** Click to select the function block as the start line of run, and there will be an icon in front of the function block

Operation

Cut

Copy

Paste

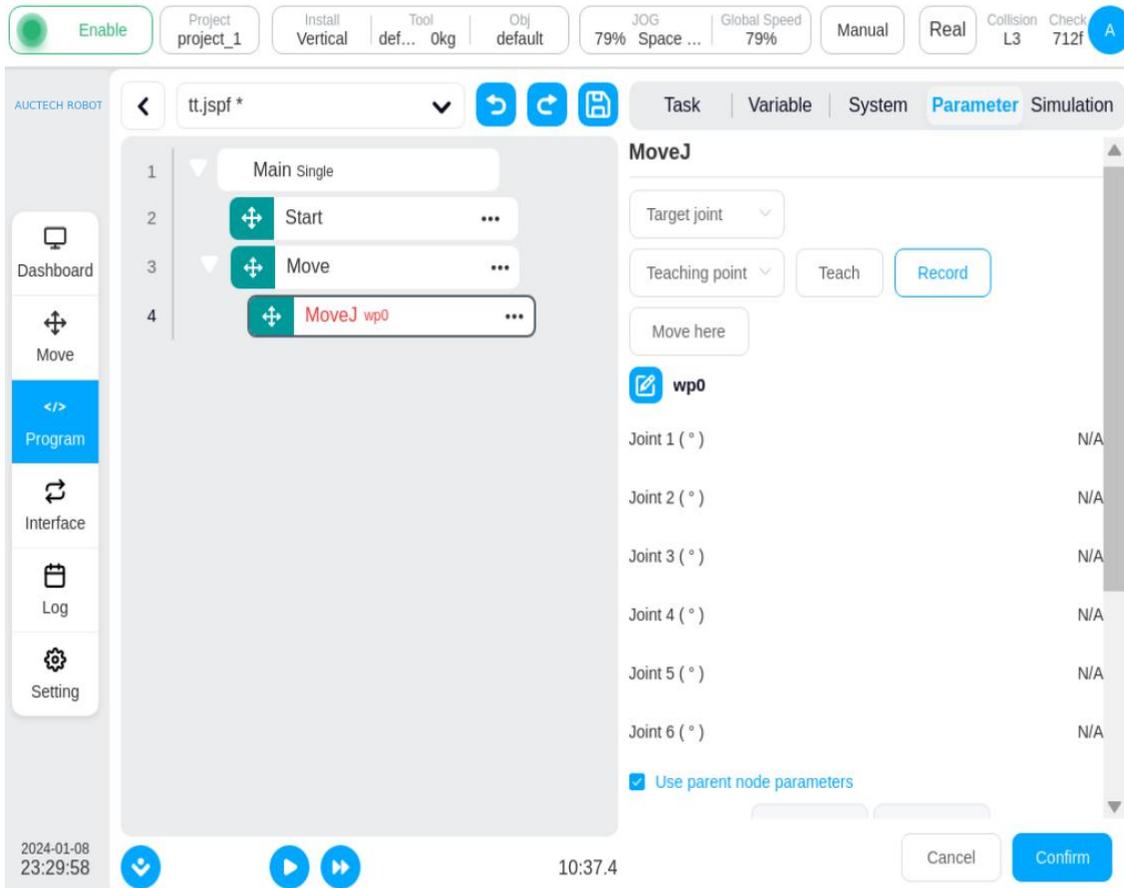
Disable

Batch Operation

Delete

Set as the Start block

## 2.4 Instruction blocks -Movej



The screenshot displays the AUCTECH robot programming software interface. At the top, there are control buttons for 'Enable', 'Project project\_1', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG 79%', 'Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check 712f'. The main workspace shows a program tree with 'Main Single' containing 'Start', 'Move', and 'MoveJ wp0' blocks. The 'MoveJ' block is selected, and its configuration panel is open on the right. The configuration panel includes fields for 'Target joint', 'Teaching point', and 'Move here'. Below these is a table for joint parameters:

Joint	Value
Joint 1 (°)	N/A
Joint 2 (°)	N/A
Joint 3 (°)	N/A
Joint 4 (°)	N/A
Joint 5 (°)	N/A
Joint 6 (°)	N/A

At the bottom of the configuration panel, there is a checkbox for 'Use parent node parameters' which is checked. The interface also shows a 'Dashboard' sidebar on the left and a 'Program' button. The bottom status bar displays the date '2024-01-08', time '23:29:58', and a timer '10:37.4'.

The robot **moves according to the joint movement**, and can choose to move to the target joint or target posture.

Configurable parameters:

**Target joint:** It can be set by teaching or set as a variable, and can be changed manually after teaching.

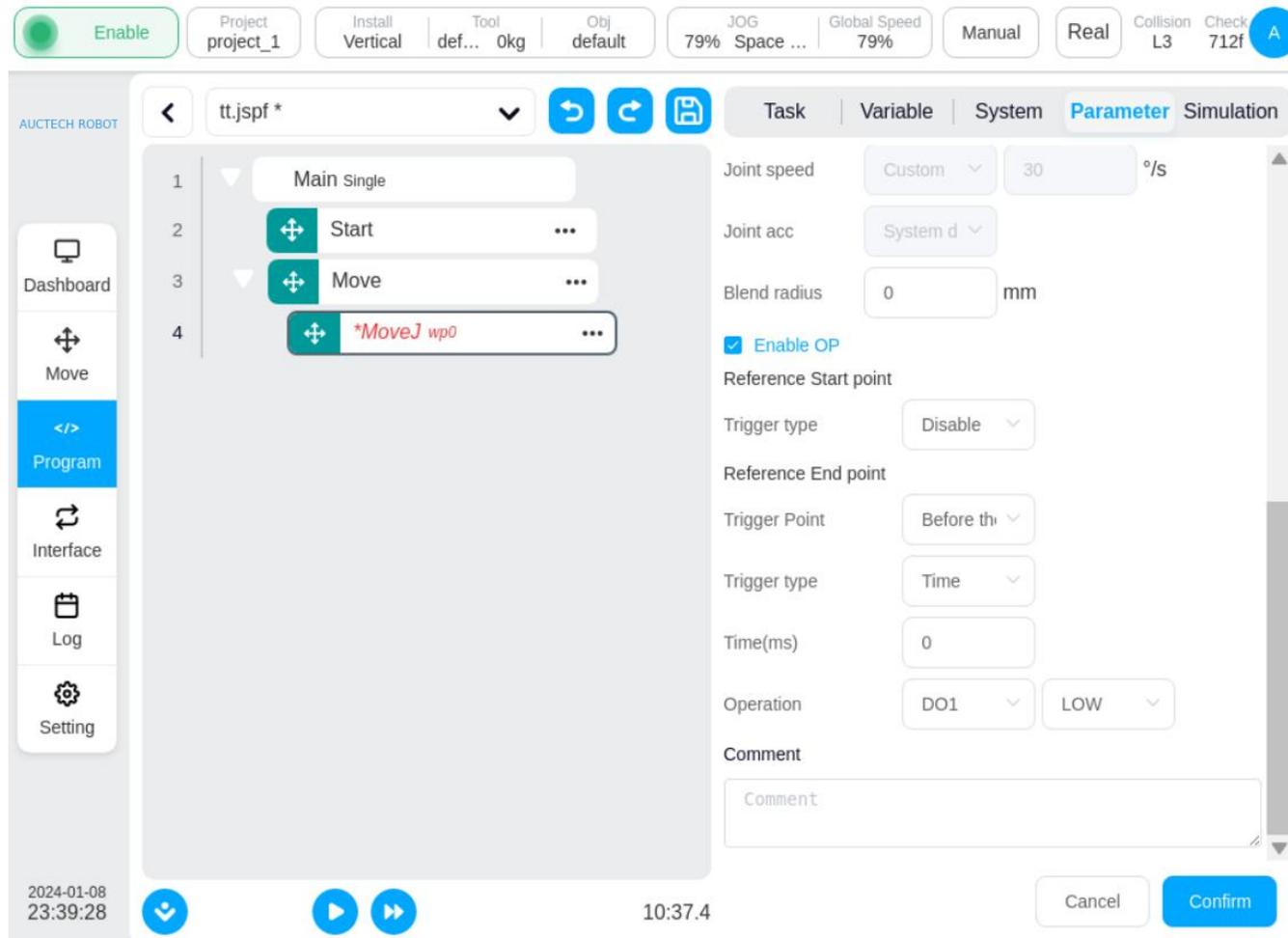
**Move a point:** You can set the point as a variable.

**Use Parent Node Parameters:** If selected, the function block uses the joint angular velocity and joint angular acceleration parameters set by the Move function block of the parent node. If this option is not selected, you need to set the joint angular velocity and joint angular acceleration for the function block, which is selected by default

**Joint angular velocity:** unit°/s; **Joint angular acceleration:** unit°/s<sup>2</sup> **Fusion radius:** The unit mm0 indicates no fusion

**Enable OP:** The OP function allows you to set the status of the universal digital output during trajectory execution.

## 2.4 Instruction blocks -Movej



The screenshot shows the AUCTECH software interface. At the top, there are status indicators: 'Enable' (green), 'Project project\_1', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG 79% Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check 712f'. The main workspace shows a program tree with a 'Main Single' block containing 'Start', 'Move', and '\*MoveJ wp0'. The right panel is set to 'Parameter' mode and shows the following configuration for the selected instruction:

- Joint speed: Custom, 30 %/s
- Joint acc: System d
- Blend radius: 0 mm
- Enable OP
- Reference Start point
- Trigger type: Disable
- Reference End point
- Trigger Point: Before th
- Trigger type: Time
- Time(ms): 0
- Operation: DO1, LOW
- Comment: Comment

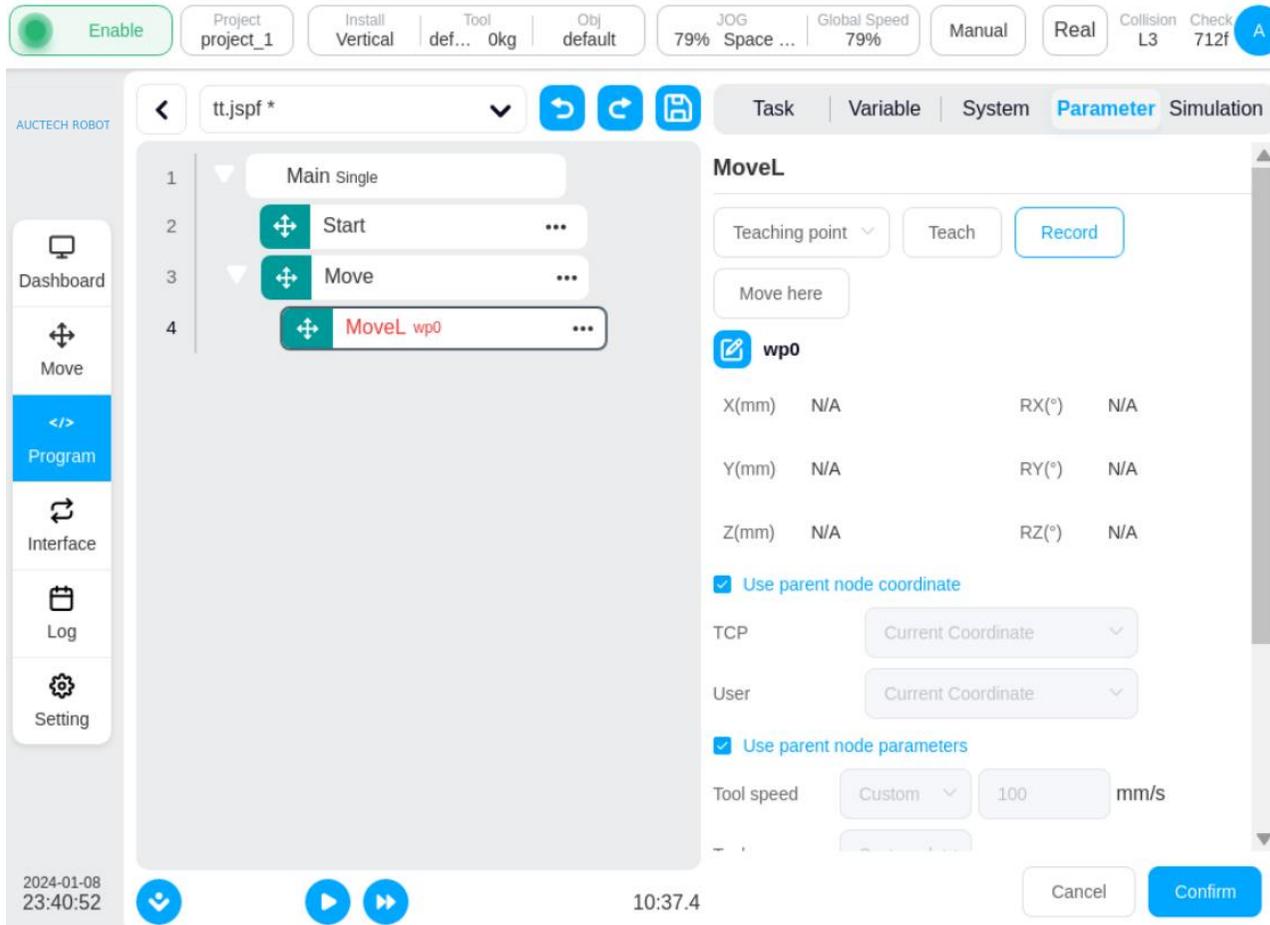
At the bottom, there are 'Cancel' and 'Confirm' buttons, along with a timestamp '2024-01-08 23:39:28' and a time '10:37.4'.

If OP is enabled, you need to configure the following:  
It can be triggered after the track starts and before the track ends

**Trigger Type:** You can select no trigger or time trigger  
Trigger a delay

**Output Port:** Select the port and port status

## 2.4 Instruction blocks -MoveL



The screenshot displays the AUCTECH robot programming interface. At the top, there is a status bar with various indicators: 'Enable' (green), 'Project project\_1', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG 79% Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check 712f'. Below this, the main workspace shows a program tree for 'tt.jspf \*' with a 'Main Single' block containing 'Start', 'Move', and 'MoveL wp0'. The 'MoveL' block is selected, and its configuration panel is open. The panel includes a 'Teaching point' dropdown, 'Teach' and 'Record' buttons, and a 'Move here' button. A table shows the target coordinates for 'wp0':

Coordinate	Value	Angle	Value
X(mm)	N/A	RX(°)	N/A
Y(mm)	N/A	RY(°)	N/A
Z(mm)	N/A	RZ(°)	N/A

Additional settings include checked options for 'Use parent node coordinate' and 'Use parent node parameters'. The 'TCP' and 'User' fields are set to 'Current Coordinate'. The 'Tool speed' is set to 'Custom' with a value of '100 mm/s'. At the bottom, there are 'Cancel' and 'Confirm' buttons. The interface also shows a sidebar with 'Dashboard', 'Move', 'Program', 'Interface', 'Log', and 'Setting' options, and a timestamp of '2024-01-08 23:40:52' and '10:37.4'.

The robot moves to the target attitude in a straight line, and the parameters can be set:

**Enable OP:** The OP function allows you to set the status of the universal digital output during trajectory execution.

**Move the setting point:** You can set the point as a variable, and you can manually change it after the variable teaching is set.

**Use Parent Node Coordinate System:** If selected, the parameter uses the reference coordinate system set by the Move function block of the parent node, which is selected by default.

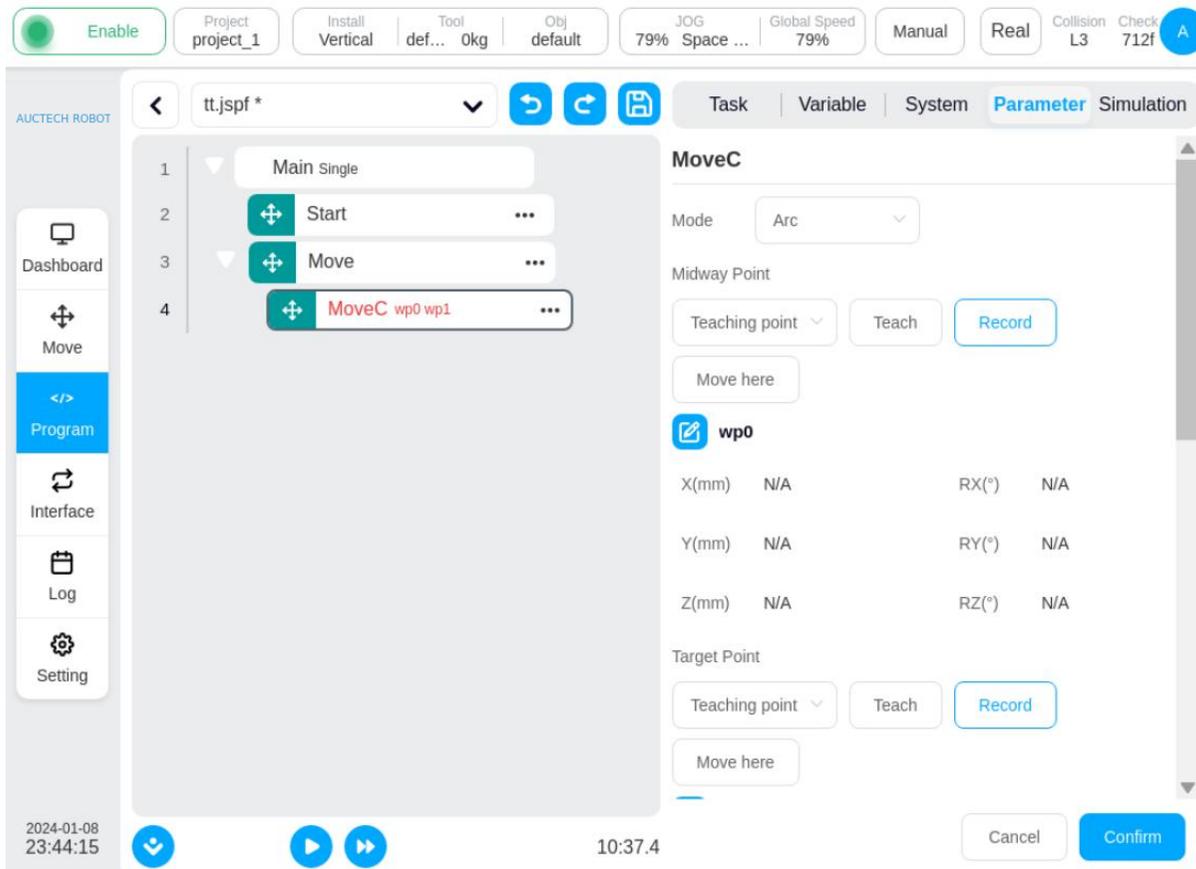
**Use the parent node parameter:** the same as the description in Movj, and do not repeat it.

**End velocity:** in mm/s

**End acceleration:** in mm/s<sup>2</sup>

**Fusion radius:** in mm, 0 means no fusion

## 2.4 Instruction blocks - MoveC



The screenshot shows the AUCTECH ROBOT software interface. At the top, there is a status bar with 'Enable' (green), 'Project project\_1', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG 79% Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check 712f'. Below this is a navigation bar with 'Task', 'Variable', 'System', 'Parameter', and 'Simulation' tabs. The main workspace shows a program tree with 'Main Single' containing 'Start', 'Move', and 'MoveC wp0 wp1' blocks. The 'MoveC' block is selected, and its configuration panel is visible on the right. The configuration panel includes a 'Mode' dropdown set to 'Arc', a 'Midway Point' section with 'Teaching point' and 'Teach' buttons, and a 'Target Point' section with 'Teaching point' and 'Teach' buttons. A table below shows the coordinates for the 'wp0' point:

Coordinate	Value	Unit	Value
X(mm)	N/A	RX(°)	N/A
Y(mm)	N/A	RY(°)	N/A
Z(mm)	N/A	RZ(°)	N/A

At the bottom of the interface, there is a date and time display '2024-01-08 23:44:15', a play button, a stop button, and a time display '10:37.4'. There are also 'Cancel' and 'Confirm' buttons at the bottom right of the configuration panel.

The robot moves in an arc or a circle, and the parameters can be set:

**Mode:** Arc or Circle

**Intermediate point:** It can be set or set as a variable by teach-in, and can be changed manually after teach-in.

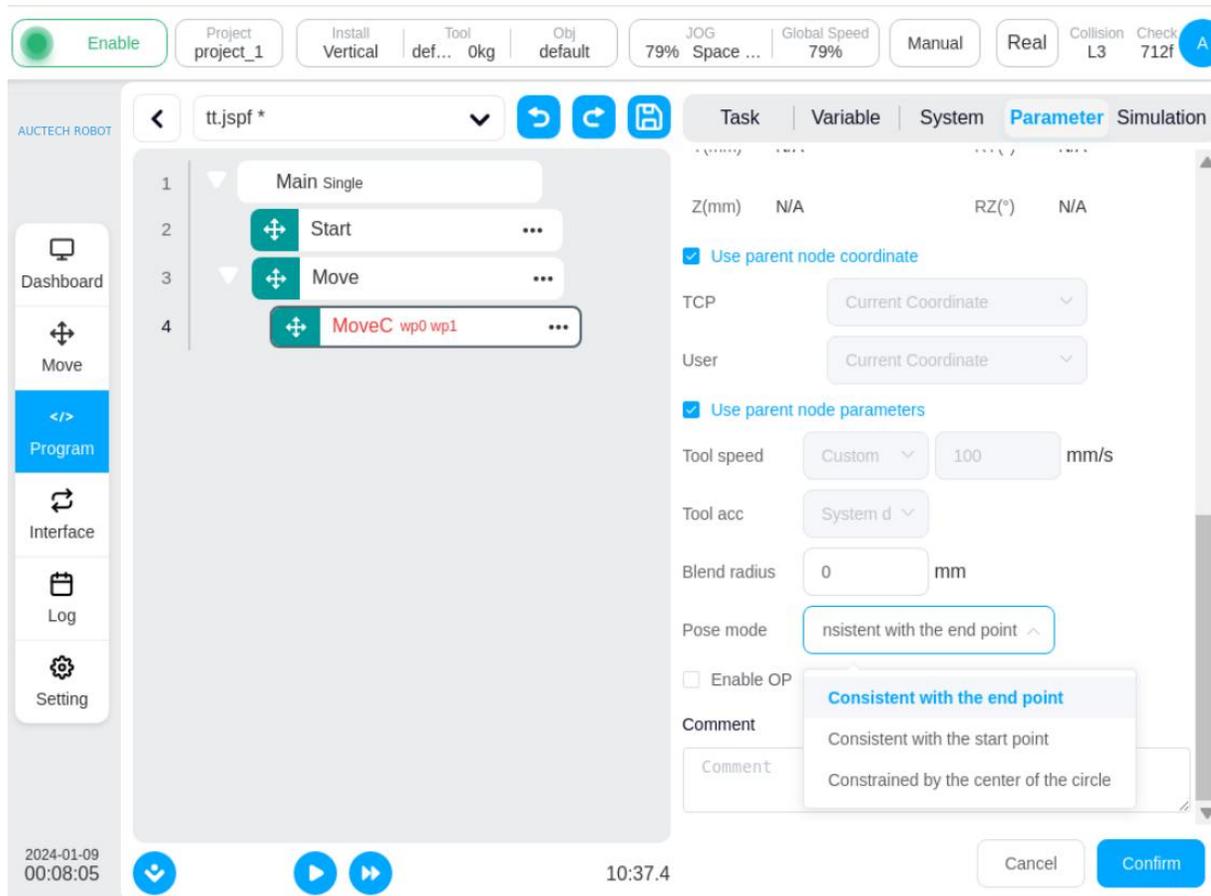
**Target point:** It can be set or set as a variable by teaching, and can be changed manually after teaching.

**Use the coordinate system of the parent node:** the same as the description at MovL

**Use the parent node parameter:** the same as the description in Movj

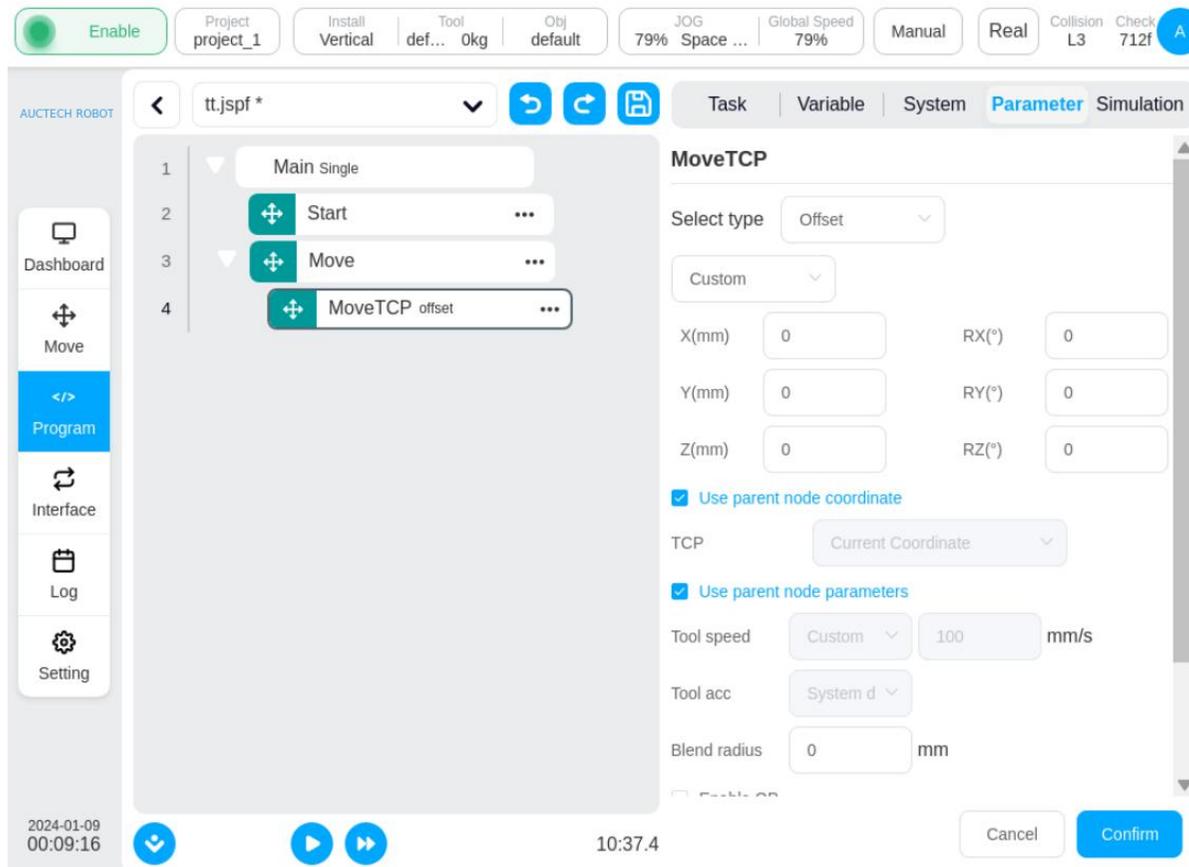
**Enable OP:** Same as described in Movj

## 2.4 Instruction blocks - MoveC



There are **three modes of attitude control**: consistent with the end point, consistent with the starting point, and constrained by the center of the circle.

## 2.4 Instruction blocks - MoveTCP



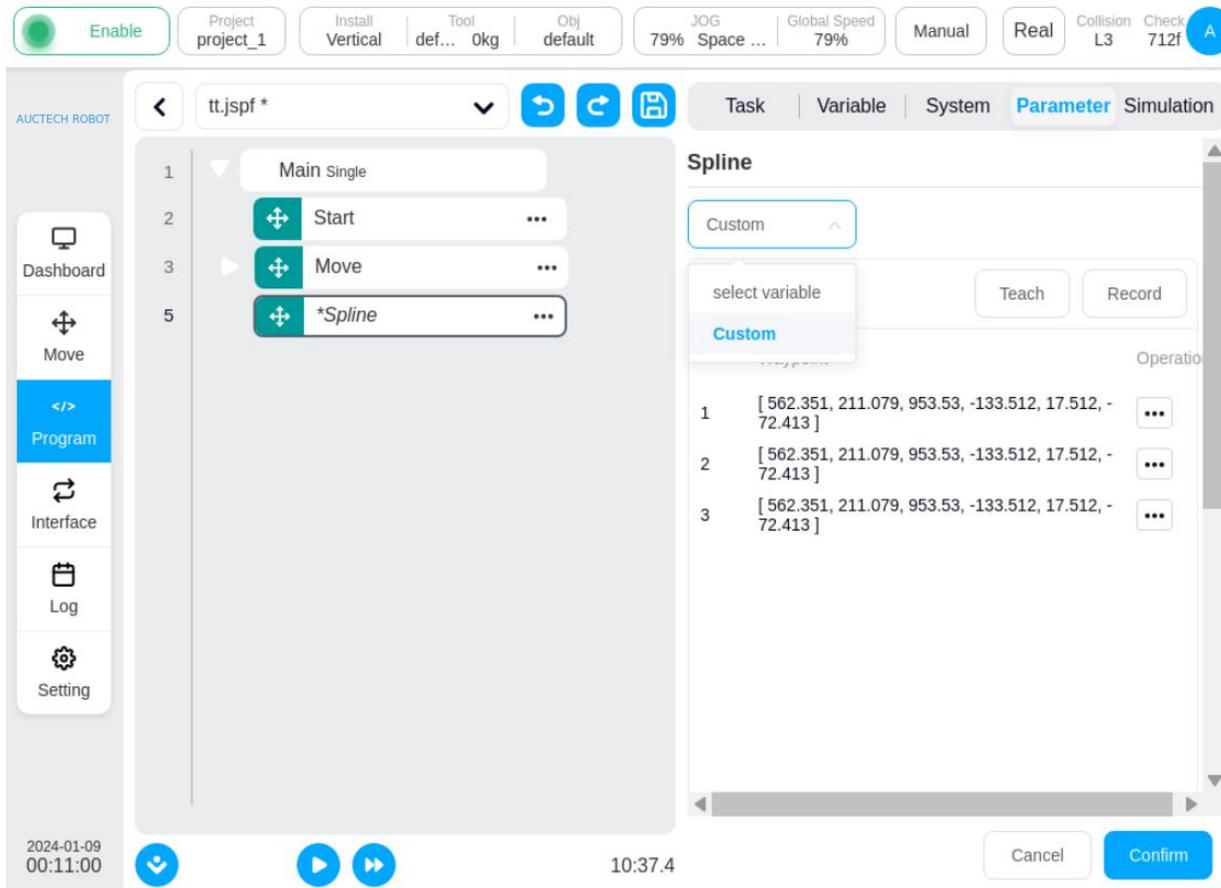
The robot **moves along the tool's coordinate system.**

You can enter offsets in each direction directly, or you can use variables. You can teach two points and use the offset between the two points as the offset value for the movement. Use the parent coordinate system: Same as described at MovL

Use parent node parameters: same as described at Movj

Enable OP: Same as described at Movj

## 2.4 Instruction blocks -Spline



Controls the end of the robot to move along a spline curve.

**Spline waypoints:** Add points by teaching or selecting variables, and generate splines based on points

**Reference Coordinate System:** When teaching the point, the default is the current tool and workpiece coordinate system, which can be changed manually

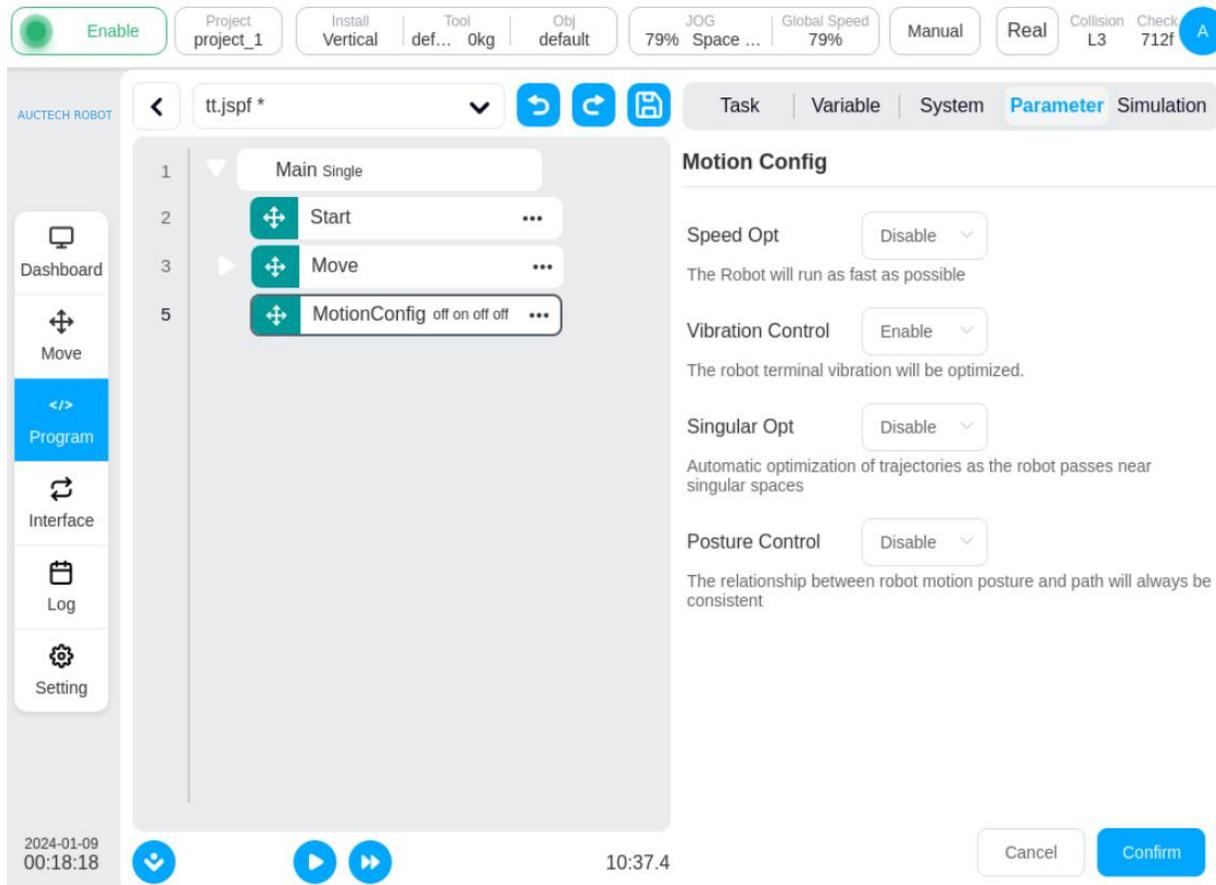
The robot moves along the tool's coordinate system.

You can enter offsets in each direction directly, or you can use variables. You can teach two points and use the offset between the two points as the offset value for the movement.

Use the parent coordinate system: Same as described at MovL

**Use parent node parameters:** same as described at Movj  
**Enable OP:** Same as described at Movj

## 2.4 Instruction blocks -MotionConfig



The screenshot shows the AUCTECH robot control interface. At the top, there is a status bar with 'Enable' (green), 'Project project\_1', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG 79% Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check 712f'. Below this is a navigation menu with 'Dashboard', 'Move', 'Program' (selected), 'Interface', 'Log', and 'Setting'. The main area displays a program 'tt.jspf\*' with a list of instruction blocks: 'Main Single', 'Start', 'Move', and 'MotionConfig off on off off'. The 'MotionConfig' block is expanded, showing the following settings:

- Speed Opt:** Disable (The Robot will run as fast as possible)
- Vibration Control:** Enable (The robot terminal vibration will be optimized.)
- Singular Opt:** Disable (Automatic optimization of trajectories as the robot passes near singular spaces)
- Posture Control:** Disable (The relationship between robot motion posture and path will always be consistent)

At the bottom, there are 'Cancel' and 'Confirm' buttons, and a timestamp '2024-01-09 00:18:18' and '10:37.4'.

(1): Turn motion-related configurations on or off.

Options:

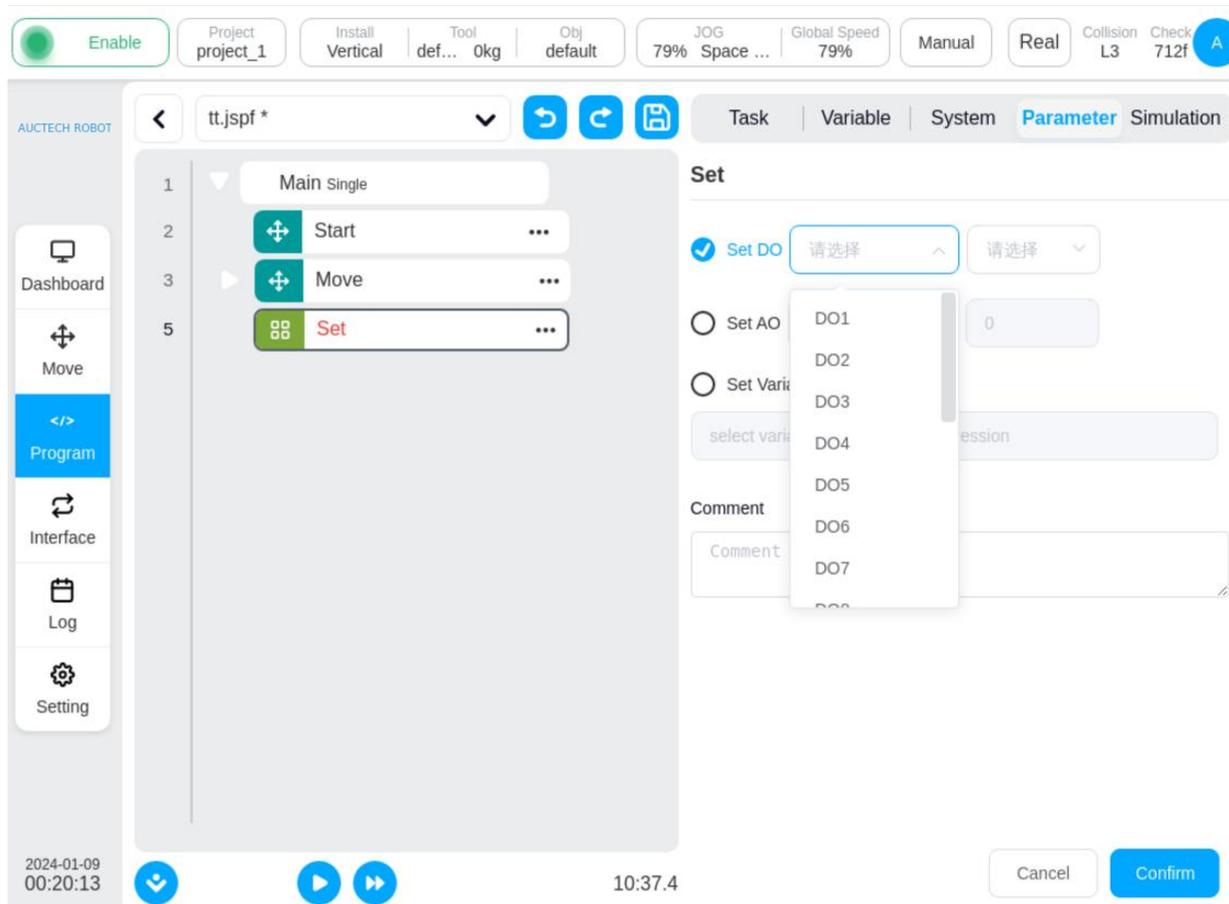
**Speed optimization:** When enabled, the robotic arm will follow the path at the highest possible speed while meeting the constraints of the system.

(2): **Acceleration optimization:** After enabled, the acceleration parameters in the script will be automatically optimized, and the acceleration will not be strictly followed.

(3): **Vibration control:** After it is turned on, it will be optimized for the vibration at the end of the robot.

(4): After enabled, the trajectory will be automatically optimized when the robot passes near the singular space.

## ◆ 2.5 Instruction blocks -Set



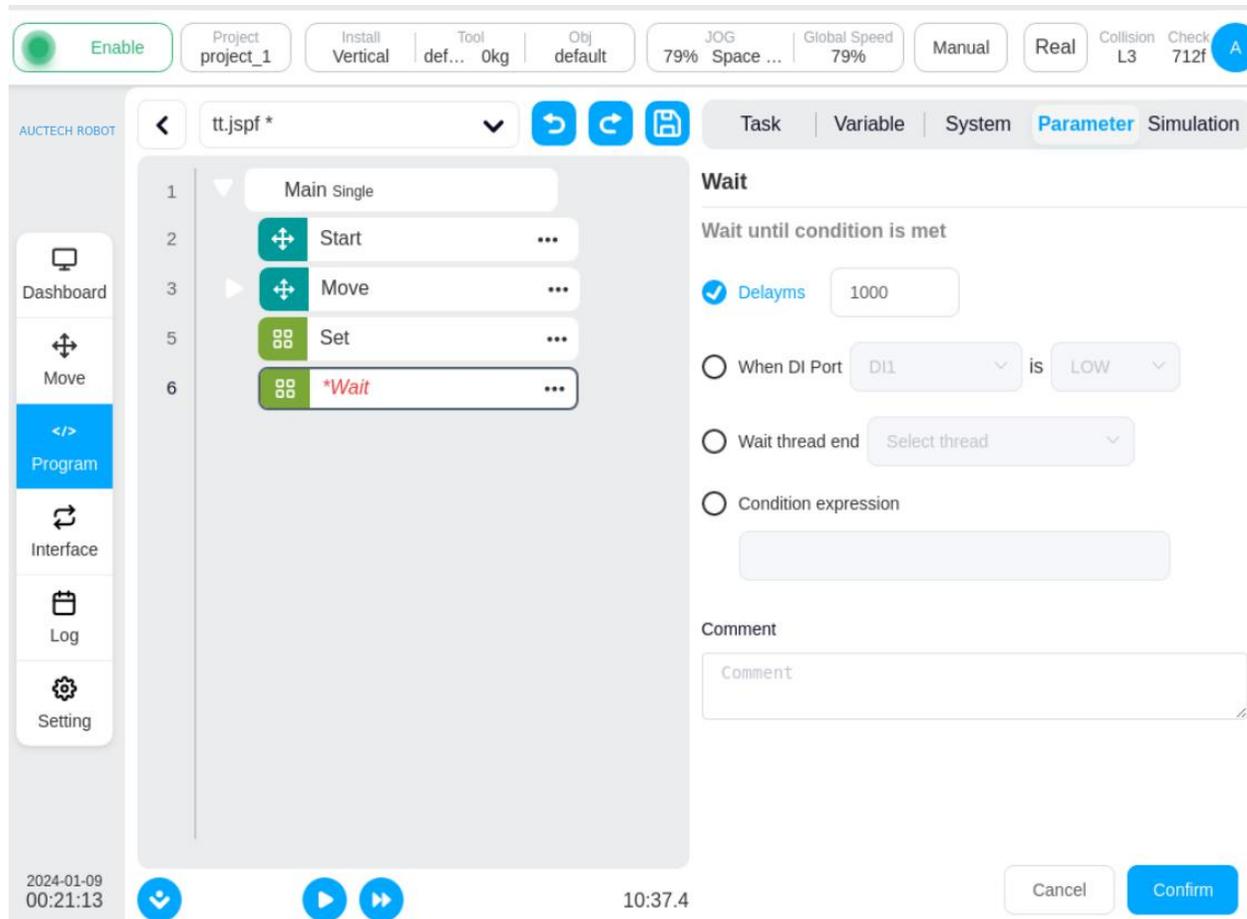
The screenshot displays the AUCTECH ROBOT software interface. At the top, there is a status bar with various indicators: 'Enable' (green), 'Project project\_1', 'Install Vertical', 'Tool def...', 'Obj 0kg', 'Obj default', 'JOG 79%' and 'Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check 712f'. Below this, a navigation bar shows 'Task', 'Variable', 'System', 'Parameter', and 'Simulation'. The main workspace is titled 'tt.jspf \*' and contains a sequence of instruction blocks: 'Main Single', 'Start', 'Move', and 'Set'. The 'Set' block is selected, and its configuration panel is open on the right. The 'Set' panel has three radio buttons: 'Set DO' (checked), 'Set AO', and 'Set Vari'. The 'Set DO' option is active, and a dropdown menu is open, listing digital output ports from DO1 to DO7. The 'Set AO' option has a value of '0' entered. The 'Set Vari' option has a 'select vari' dropdown and a 'ession' input field. There are also 'Comment' and 'Comment' input fields. At the bottom of the 'Set' panel are 'Cancel' and 'Confirm' buttons. The bottom status bar shows the date '2024-01-09', time '00:20:13', and a timer '10:37.4'.

(1): The digital output port of the control cabinet and the digital output port at the end of the robotic arm can be set to high level or low level.

(2): Assign values to program variables or system variables.

(3): Set the analog output

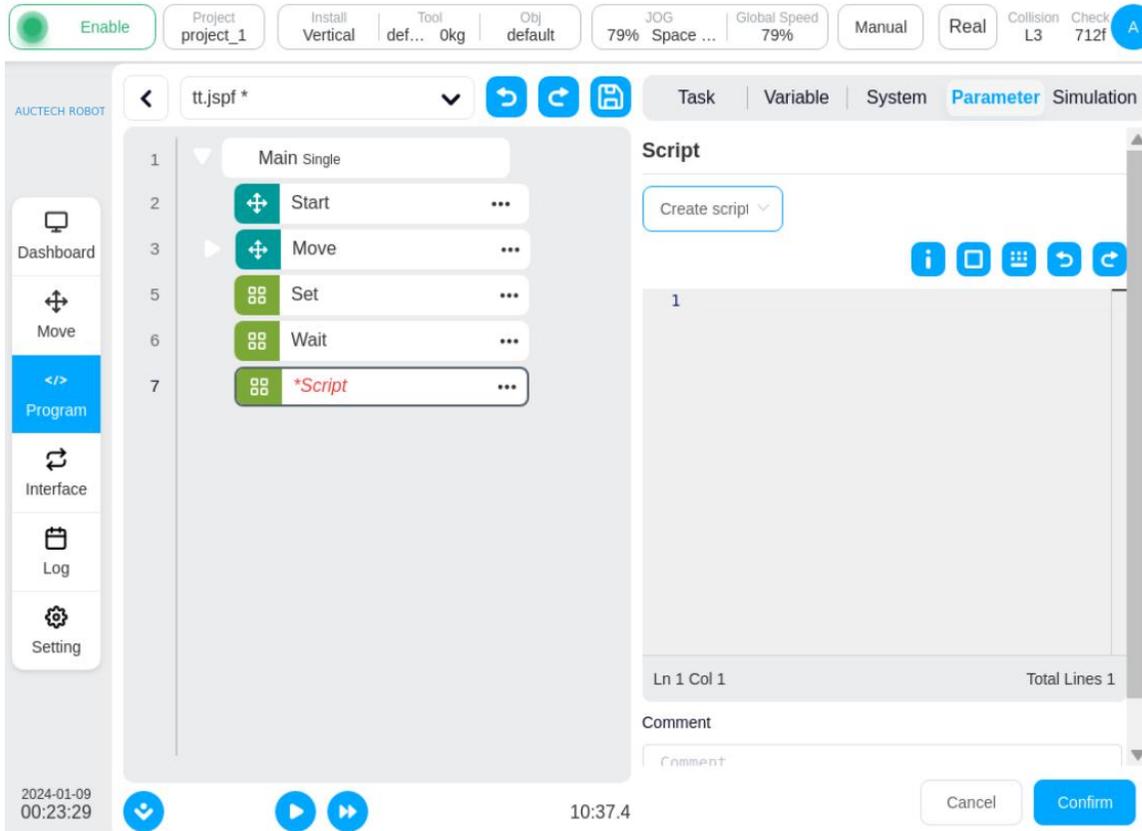
## 2.5 Instruction blocks -Wait



The screenshot displays the AUCTECH software interface. At the top, there is a status bar with various indicators: 'Enable' (green), 'Project project\_1', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG 79%', 'Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check 712f'. Below this, a navigation bar shows 'Task', 'Variable', 'System', 'Parameter', and 'Simulation'. The main workspace is titled 'tt.jspf \*' and contains a sequence of instruction blocks: 'Main Single', 'Start', 'Move', 'Set', and '\*Wait'. The '\*Wait' block is selected, and its configuration panel is open on the right. The panel is titled 'Wait' and includes the following options: 'Wait until condition is met', 'Delays' (checked, value 1000), 'When DI Port DI1 is LOW', 'Wait thread end' (Select thread), and 'Condition expression'. A 'Comment' field is also present. At the bottom of the configuration panel, there are 'Cancel' and 'Confirm' buttons. The bottom status bar shows the date '2024-01-09', time '00:21:13', and a timer '10:37.4'.

Optionally, wait for a period of time, wait for DI signals and expressions, and the program will wait until the set conditions are met when executing to this function block.

## 2.5 Instruction blocks -Script

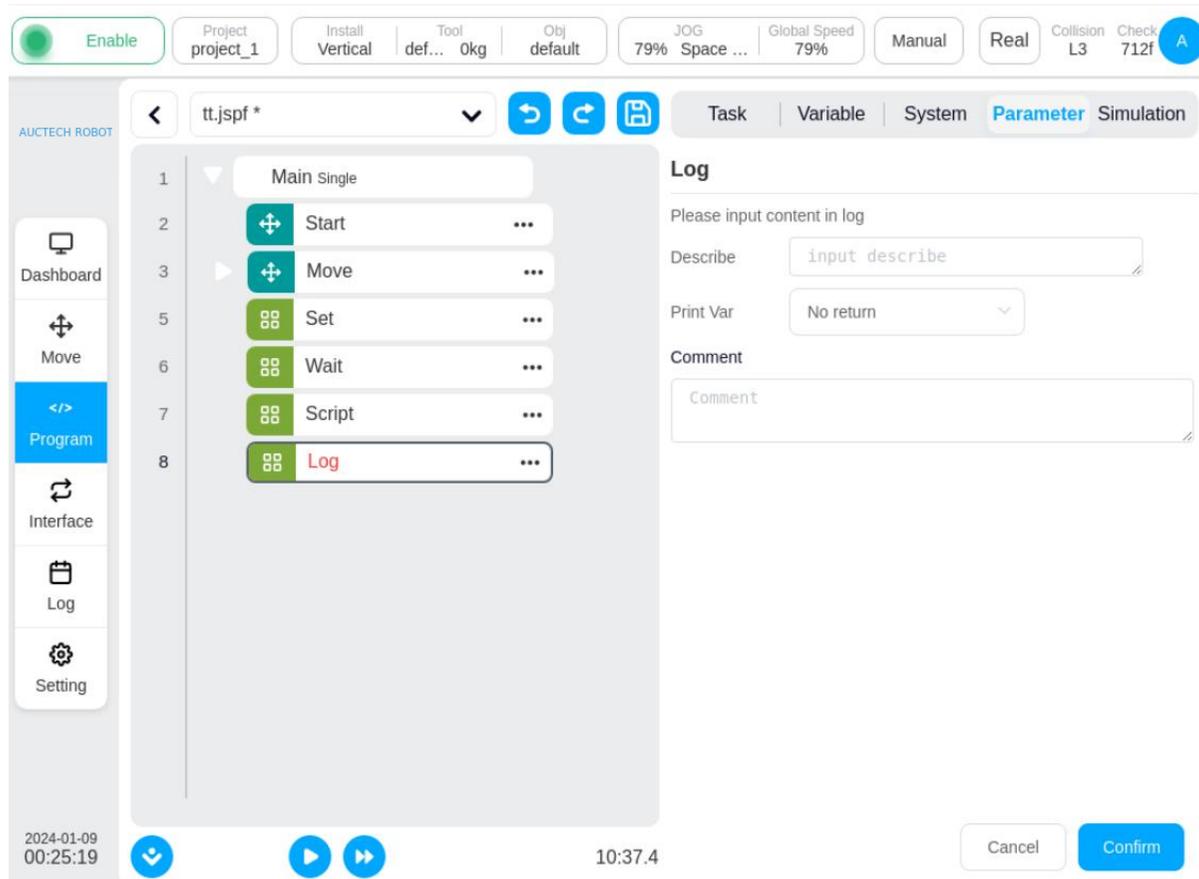


You can **choose an expression, a script, or a script file.**

- (1): Expressions can be used to create a line of scripts using the expression editor, and for the input of expressions, please refer to Section 8 Input Keyboard (Expression Input Keyboard);
- (2): Scripts can be used to write entire paragraphs of script code;
- (3): Script file can select a script from the file to a script file.

For more information about the script code, see the Scripting Manual.

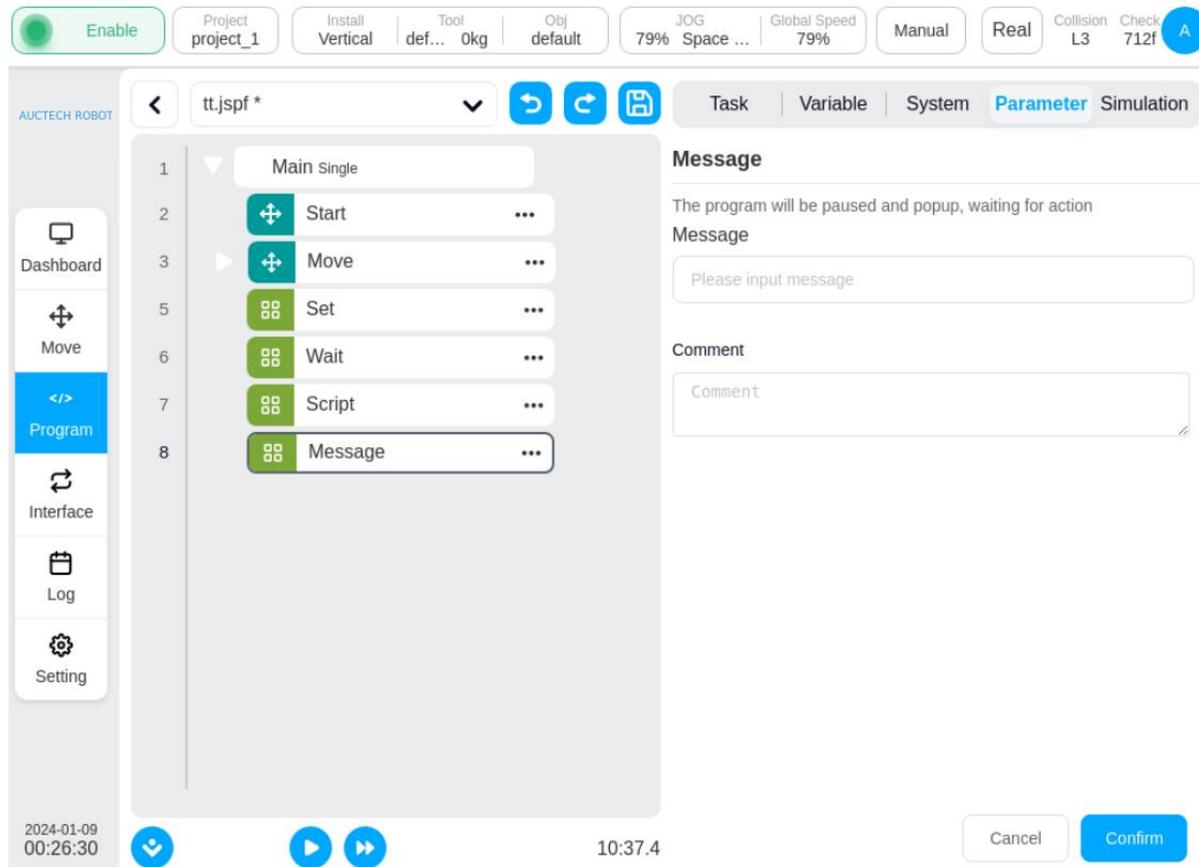
## 2.5 Instruction blocks -Log



The screenshot displays the AUCTECH ROBOT software interface. At the top, there is a status bar with various indicators: a green 'Enable' button, 'Project project\_1', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG 79% Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check 712f'. Below this, a navigation bar shows 'tt.jspf \*' and tabs for 'Task', 'Variable', 'System', 'Parameter', and 'Simulation'. The main workspace is divided into a left sidebar with icons for 'Dashboard', 'Move', 'Program' (highlighted), 'Interface', 'Log', and 'Setting'. The central area shows a sequence of instruction blocks: 'Main Single', 'Start', 'Move', 'Set', 'Wait', 'Script', and 'Log'. The 'Log' block is selected, and its configuration panel is visible on the right. This panel includes a 'Log' title, a prompt 'Please input content in log', a 'Describe' field with the value 'input describe', a 'Print Var' dropdown menu set to 'No return', and a 'Comment' text area. At the bottom of the interface, there is a timestamp '2024-01-09 00:25:19', a time display '10:37.4', and 'Cancel' and 'Confirm' buttons.

Log function block. You can print the values of messages or variables to a log file.

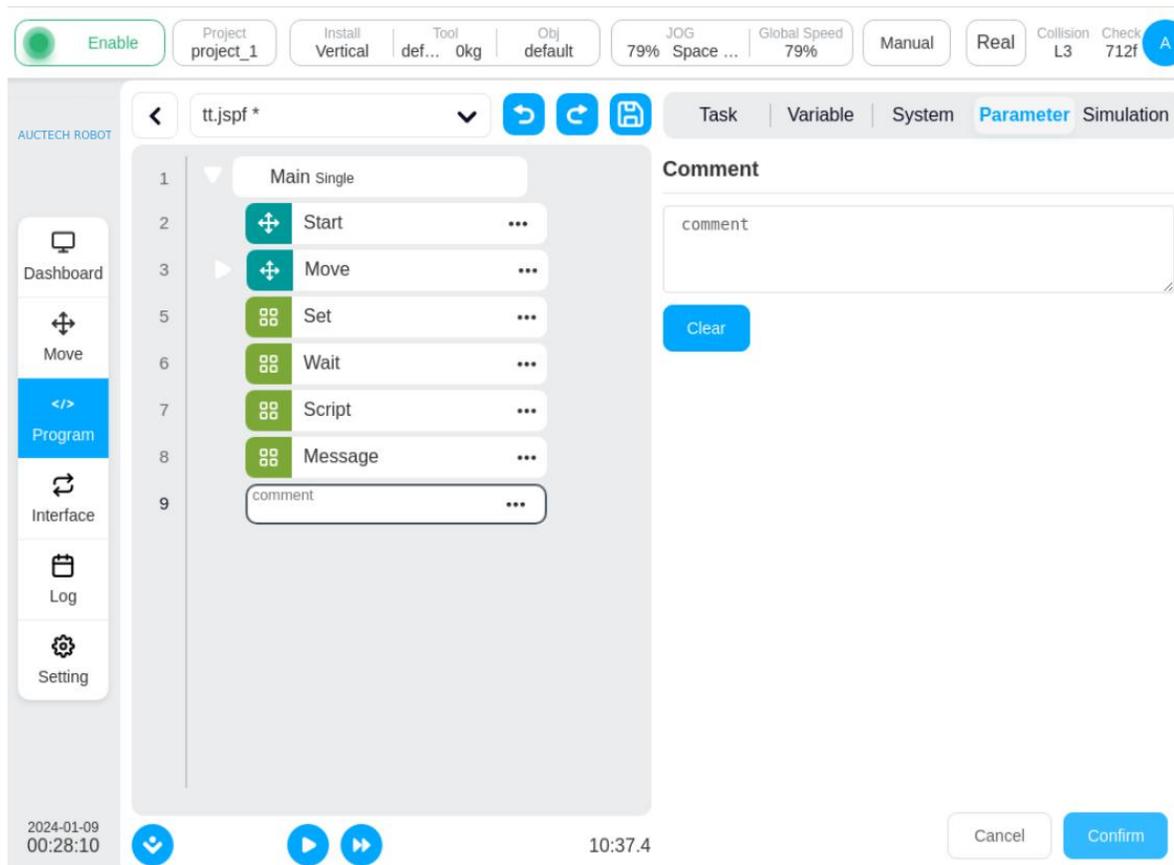
## ◆ 2.5 Instruction blocks -Message



The screenshot displays the AUCTECH software interface. At the top, there is a status bar with various indicators: 'Enable' (green), 'Project project\_1', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG 79% Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check 712f'. Below this, a navigation menu includes 'Dashboard', 'Move', 'Program' (highlighted), 'Interface', 'Log', and 'Setting'. The main workspace shows a program tree with a 'Main Single' block containing 'Start', 'Move', 'Set', 'Wait', 'Script', and 'Message' blocks. The 'Message' block is selected, and its configuration panel is open on the right. The panel includes a 'Message' field with the text 'The program will be paused and popup, waiting for action', a 'Message' input field containing 'Please input message', and a 'Comment' input field containing 'Comment'. At the bottom of the interface, there are playback controls (stop, play, next) and a timestamp of '10:37.4'. A 'Cancel' button and a 'Confirm' button are also visible.

**Message pop-up function block.** A message can be set, and when the program runs to this function block, a dialog box can pop up to display the message and the program is suspended, and the user can choose to stop or continue running the program.

## ◆ 2.5 Instruction blocks -Comment



The screenshot displays the AUCTECH ROBOT software interface. The top bar shows the system status: 'Enable', 'Project project\_1', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG 79%', 'Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', 'Check 712f', and a blue 'A' button. The main workspace shows a program tree for 'tt.jspf \*' with the following blocks:

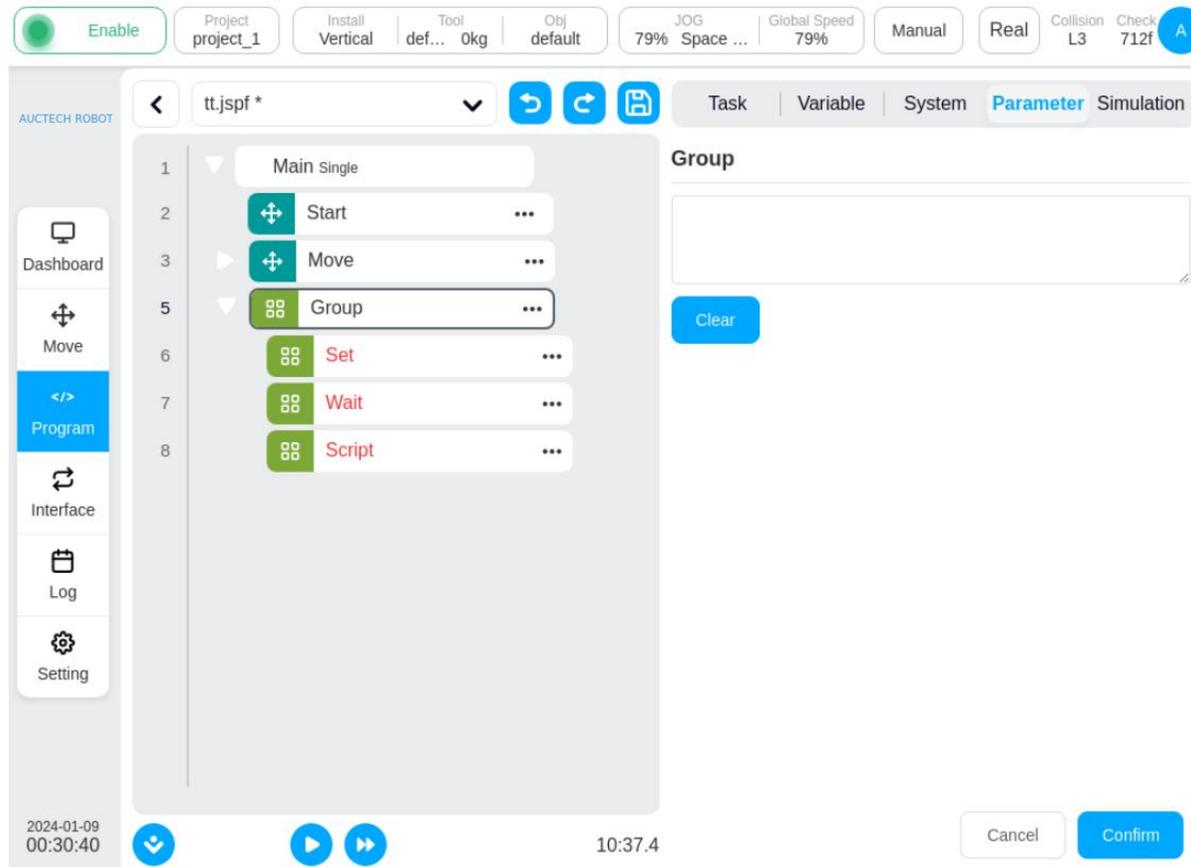
- 1 Main Single
- 2 Start
- 3 Move
- 5 Set
- 6 Wait
- 7 Script
- 8 Message
- 9 comment

The 'comment' block at line 9 is selected, and a 'Comment' dialog box is open, showing the text 'comment'. The dialog box has a 'Clear' button. At the bottom of the interface, there are 'Cancel' and 'Confirm' buttons. The bottom left corner shows the date and time: '2024-01-09 00:28:10'. The bottom right corner shows the time: '10:37.4'.

Annotate function blocks.

Add a comment to the program tree. While the program is running, this function block does nothing.

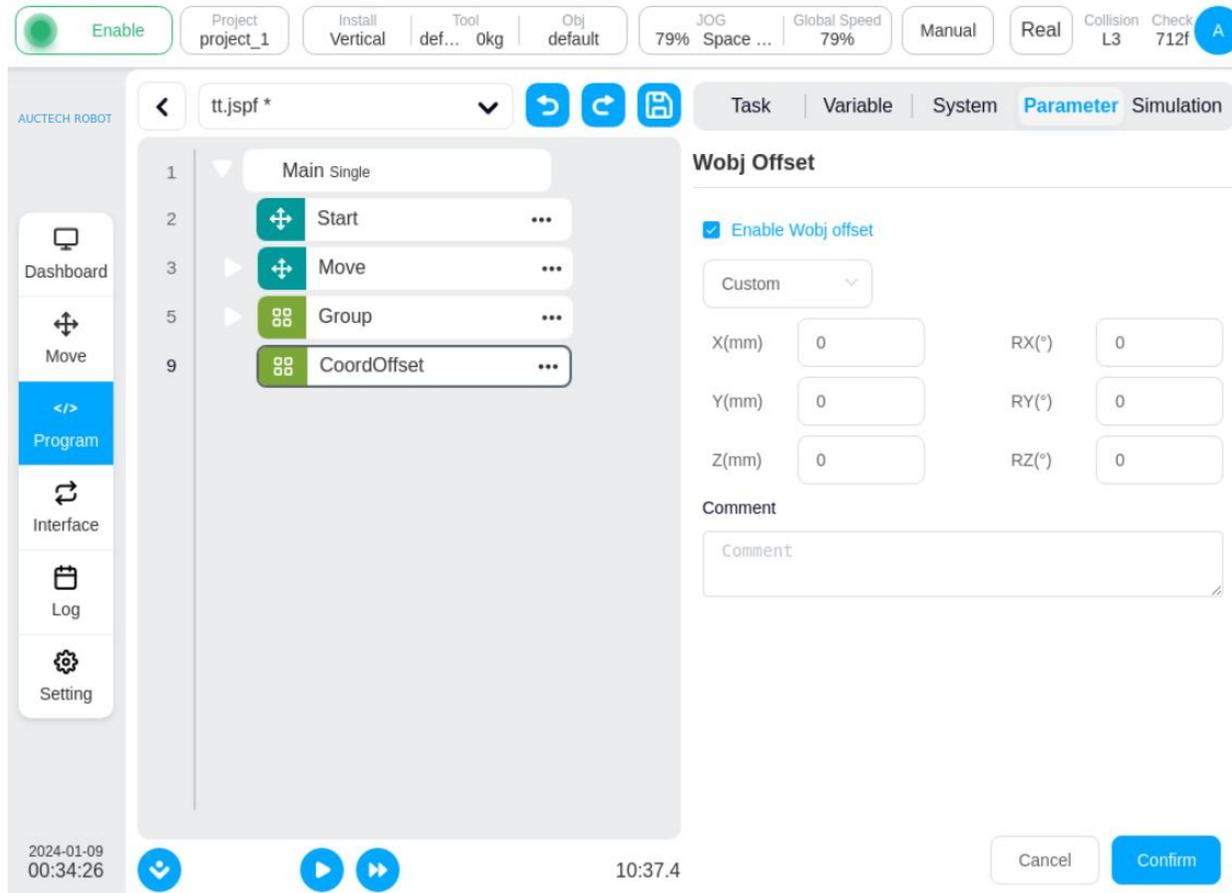
## ◆ 2.5 Instruction blocks -Group



Group function blocks.

It is used to organize the program, and some function blocks can be placed under a group to facilitate the organization and reading of the program. **There is no impact on the execution of the program.**

## 2.5 Instruction blocks -CoordOffset



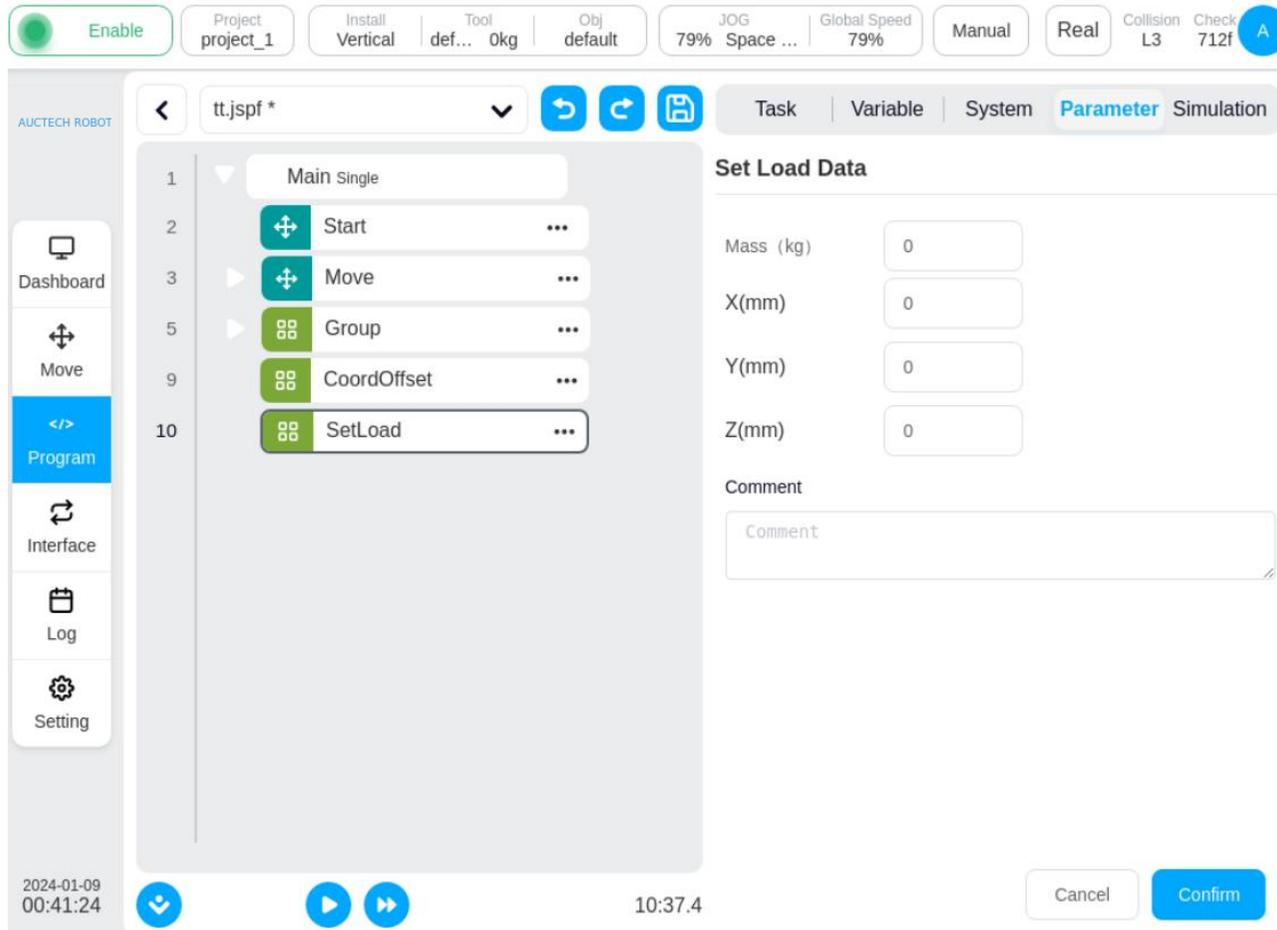
The screenshot displays the AUCTECH ROBOT software interface. At the top, there is a status bar with various indicators: 'Enable' (green), 'Project project\_1', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG Space ... 79%', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check 712f'. Below this, the main workspace shows a program tree on the left with blocks: 'Main Single', 'Start', 'Move', 'Group', and 'CoordOffset'. The 'CoordOffset' block is selected, and its configuration panel is open on the right. The 'Wobj Offset' section is active, showing a checked 'Enable Wobj offset' option and a 'Custom' dropdown menu. Below this, there are input fields for X(mm), Y(mm), Z(mm), RX(\*), RY(\*), and RZ(\*), all set to 0. A 'Comment' field is also present. At the bottom of the configuration panel, there are 'Cancel' and 'Confirm' buttons. The bottom status bar shows the date '2024-01-09', time '00:34:26', and a timer '10:37.4'.

The coordinate system offset function block **sets an offset based on the coordinate system of the workpiece.**

This offset will be added to the reference artifact coordinate system of subsequent Move function blocks.

When the program is running, it takes effect, and the program stops and cancels the coordinate system offset.

## 2.5 Instruction blocks -SetLoad



The screenshot displays the AUCTECH ROBOT software interface. At the top, there is a status bar with various indicators: a green 'Enable' button, 'Project project\_1', 'Install Vertical', 'Tool def...', 'Obj 0kg default', 'JOG 79% Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check 712f'. Below this, the main workspace shows a program tree on the left with a 'Main Single' folder containing 'Start', 'Move', 'Group', 'CoordOffset', and 'SetLoad' blocks. The 'SetLoad' block is selected, and its configuration panel is visible on the right. The 'Set Load Data' panel includes input fields for 'Mass (kg)', 'X(mm)', 'Y(mm)', and 'Z(mm)', all currently set to '0'. There is also a 'Comment' text area. At the bottom of the interface, there are navigation buttons for 'Cancel' and 'Confirm', along with a timestamp '2024-01-09 00:41:24' and a time display '10:37.4'.

Set the gripping load function block. The current load of the robot (mass, centroid) can be set during the program run .

## 2.6 Instruction blocks -While

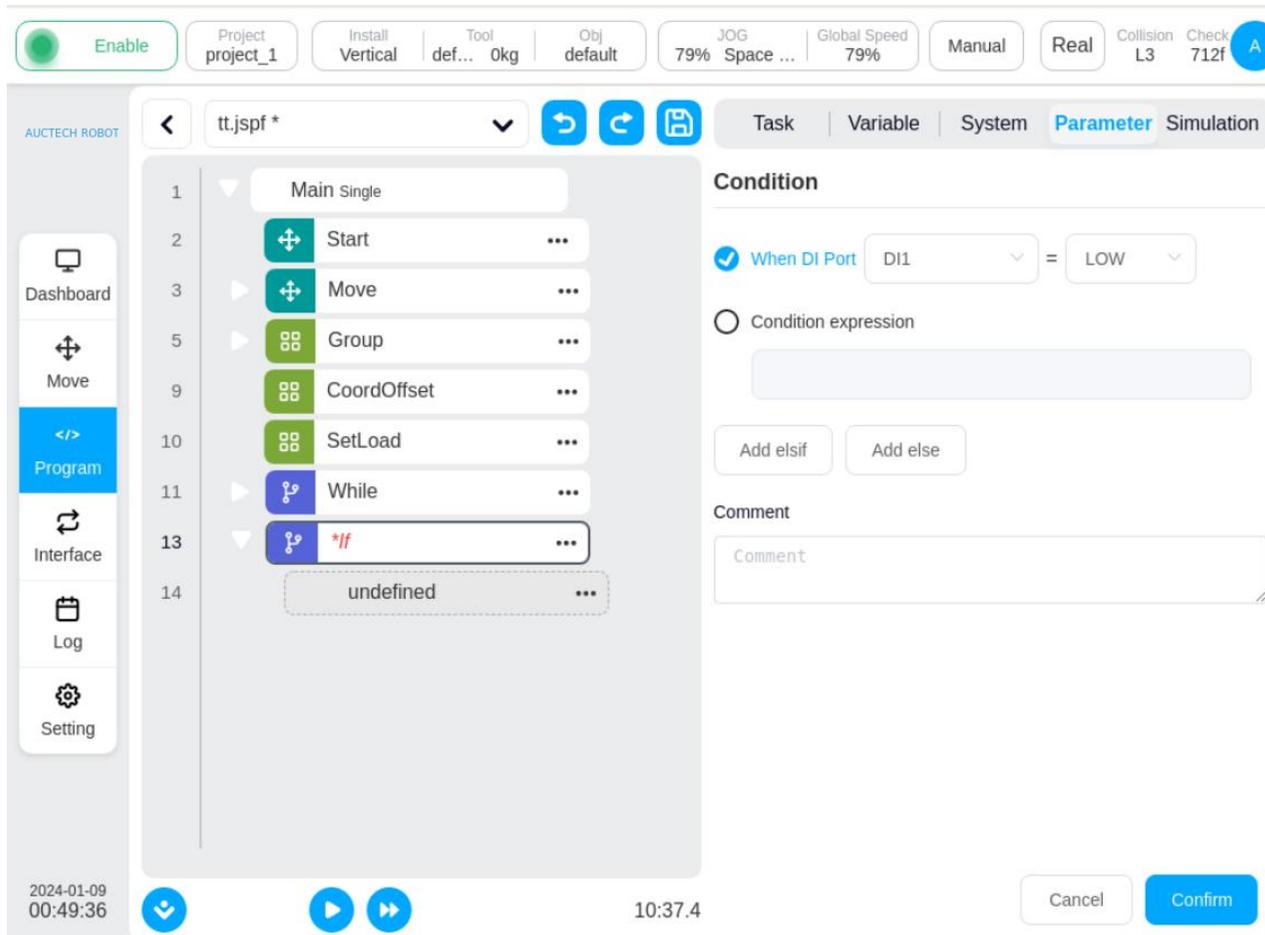


The screenshot shows the AUCTECH software interface. At the top, there's a status bar with 'Enable', 'Project project\_1', 'Install Vertical', 'Tool def...', 'Obj default', 'JOG Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check 712f'. Below this, the main workspace shows a program tree for 'tt.jspf \*'. The tree includes a 'Main Single' block at line 1, followed by 'Start' (line 2), 'Move' (line 3), 'Group' (line 5), 'CoordOffset' (line 9), 'SetLoad' (line 10), and a 'While' block (line 11). The 'While' block is expanded, showing an 'undefined' block at line 12. The right-hand panel is titled 'While' and contains the following configuration options: 'Loop always' (checked), 'Loop times' (0), and 'Condition expression' (unchecked). There is also a 'Comment' field with the text 'Comment'. At the bottom of the interface, there are 'Cancel' and 'Confirm' buttons, along with a timestamp '2024-01-09 00:48:24' and a time '10:37.4'.

Loop function blocks.

The loop executes the function blocks inside it. It can be set to keep looping; Specify the number of cycles; Specify the loop condition, and as long as the loop condition is true, the loop runs.

## 2.6 Instruction blocks -If

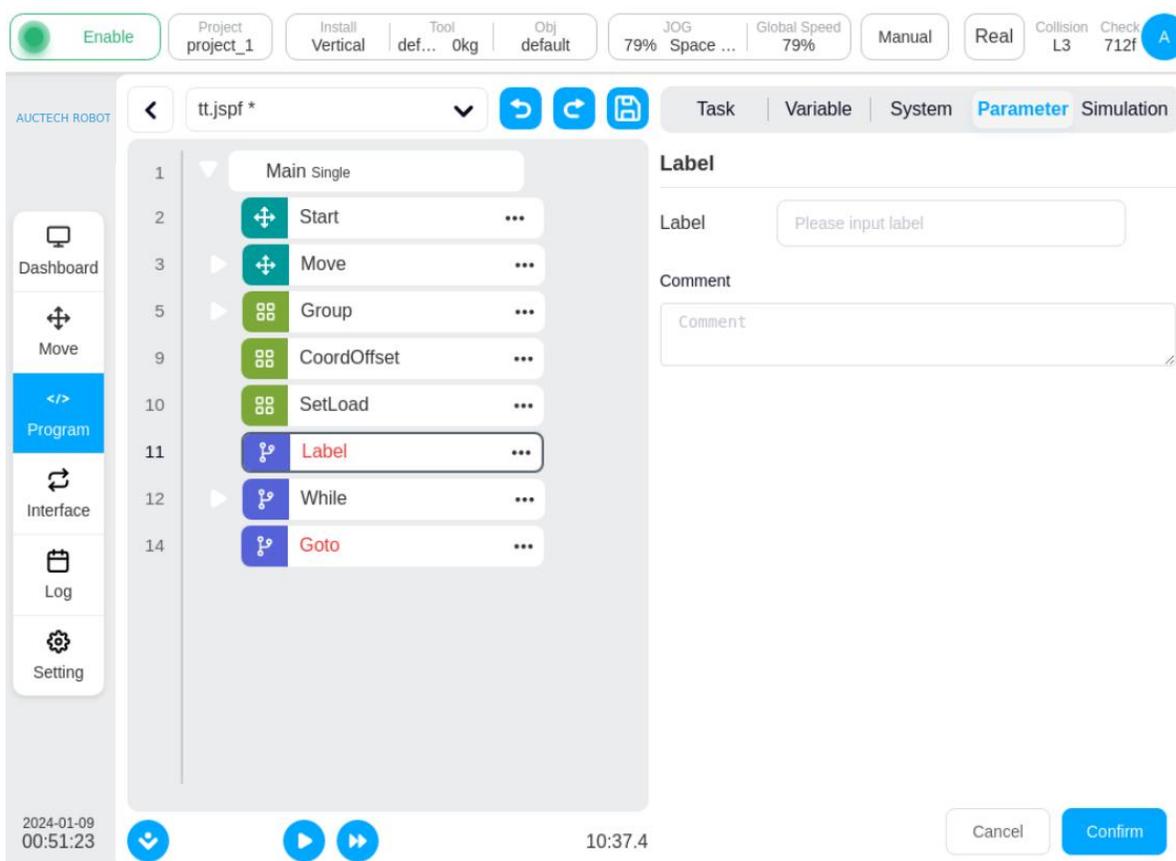


The screenshot displays the AUCTECH ROBOT software interface. The top bar includes a status bar with 'Enable', 'Project project\_1', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG 79%', 'Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check 712f'. The main workspace shows a program tree on the left with a list of instruction blocks: Main Single, Start, Move, Group, CoordOffset, SetLoad, While, and an If block. The If block is selected, and its configuration panel is visible on the right. The 'Condition' section is active, showing 'When DI Port' with 'DI1' selected and 'LOW' as the condition. Below this, there are 'Add elsif' and 'Add else' buttons. The 'Comment' section has a text input field. At the bottom, there are 'Cancel' and 'Confirm' buttons. The bottom left corner shows the date '2024-01-09' and time '00:49:36', and the bottom right corner shows the time '10:37.4'.

Conditional function blocks.

You can set the condition or condition expression of the digital input port, and if the condition is met, the function block in the If will be executed, and the subsequent **elsif or else function block can be added.**

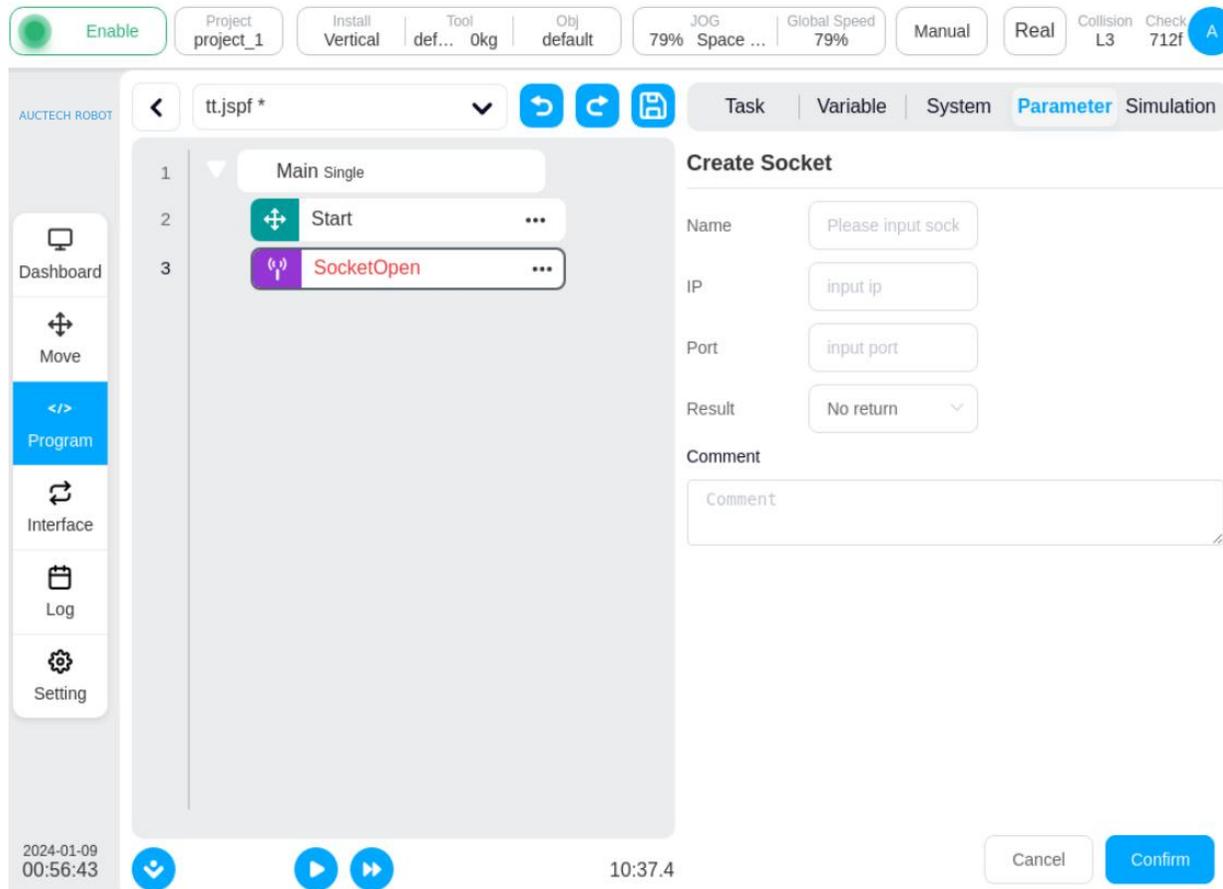
## 2.6 Instruction blocks -Goto&Label



The Goto function which works in conjunction with Label, allows you to transfer the control point of the program to the Label.

The example program shown in the figure shows that when the program runs to line 14, it will jump to label on line 11 and then run downward.

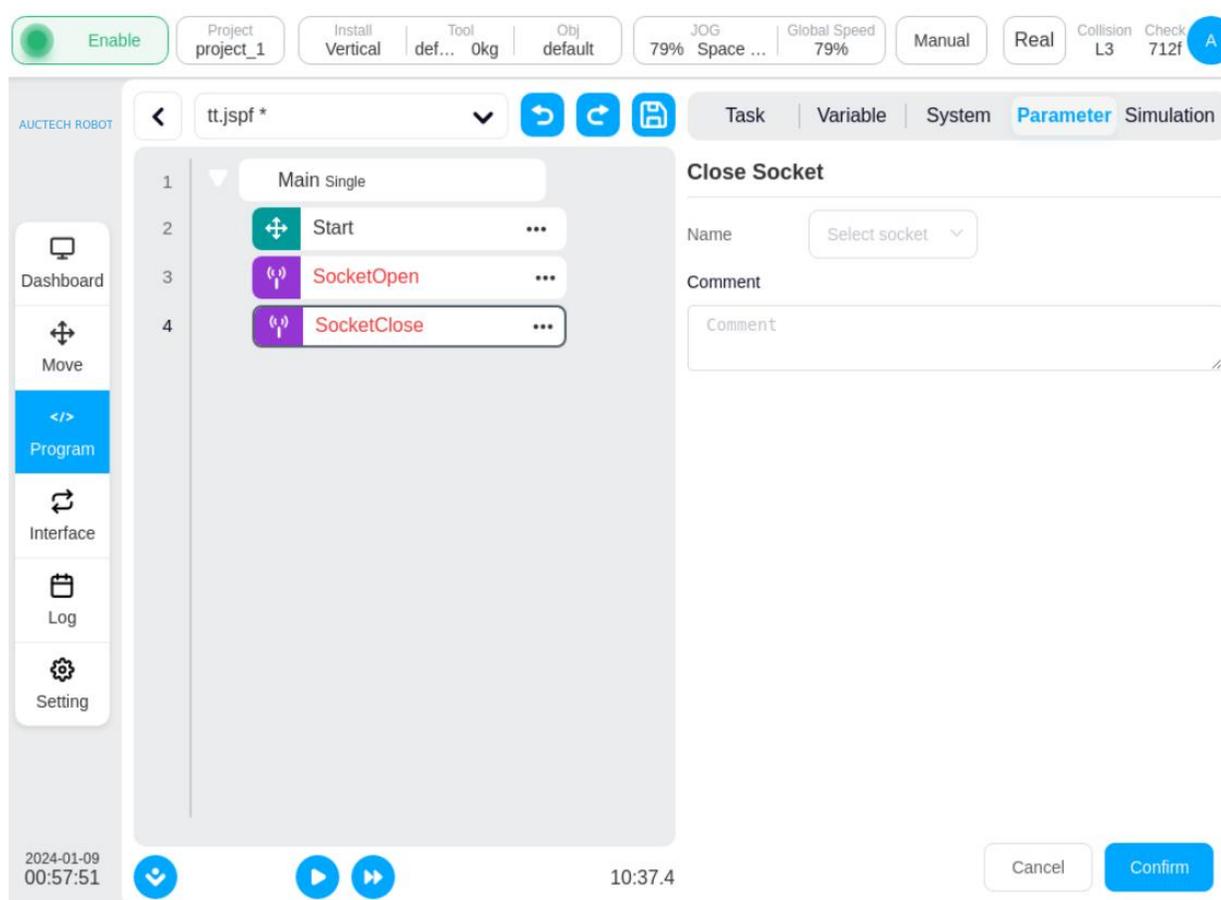
## 2.7 Instruction blocks -SocketOpen



The screenshot displays the AUCTECH ROBOT software interface. At the top, there are status indicators: 'Enable' (green), 'Project project\_1', 'Install Vertical', 'Tool def...', 'Obj 0kg', 'default', 'JOG 79%', 'Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', 'Check 712f', and a blue 'A' icon. The main workspace shows a ladder logic program with a 'Main Single' block containing a 'Start' block and a 'SocketOpen' block. A 'Create Socket' dialog box is open, with the following fields: 'Name' (Please input sock), 'IP' (input ip), 'Port' (input port), 'Result' (No return), and 'Comment' (empty). The bottom of the interface shows a date and time '2024-01-09 00:56:43', a play button, a stop button, and a time '10:37.4'. There are also 'Cancel' and 'Confirm' buttons at the bottom right of the dialog.

To create a socket, you need to set the name of the connection, configure the IP address and port number of the destination server, and select whether to bind the return value to a variable.

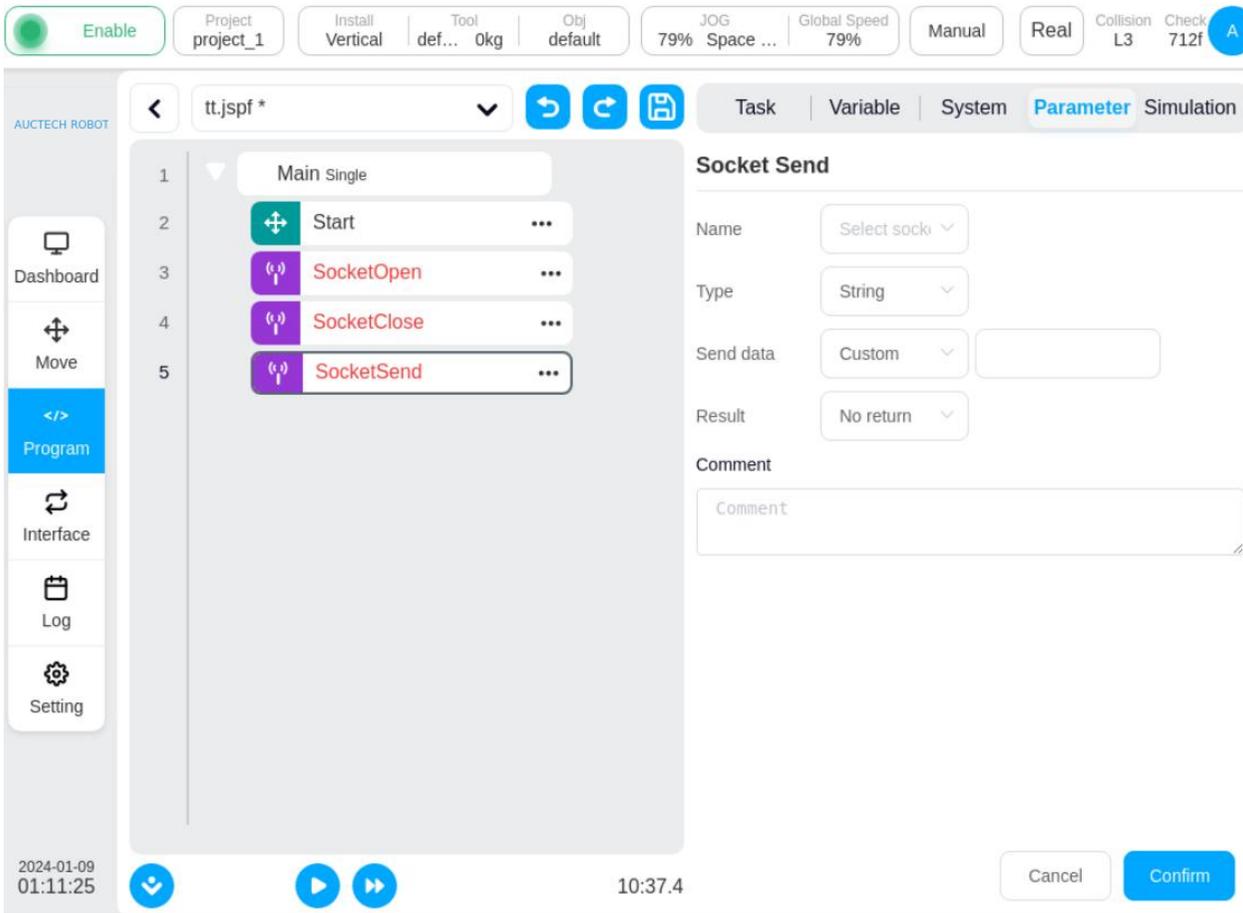
## 2.7 Instruction blocks -SocketClose



The screenshot displays the AUCTECH ROBOT software interface. At the top, there is a status bar with various controls: 'Enable' (green), 'Project project\_1', 'Install Vertical', 'Tool def...', 'Obj 0kg default', 'JOG 79% Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check 712f'. Below this, the main workspace shows a program named 'tt.jspf \*' with a list of instruction blocks: 'Main Single', 'Start', 'SocketOpen', and 'SocketClose'. The 'SocketClose' block is selected, and its configuration panel is visible on the right. The configuration panel includes a 'Name' field with a dropdown menu set to 'Select socket', a 'Comment' field, and a 'Comment' label. At the bottom of the interface, there are navigation buttons: 'Cancel' and 'Confirm', along with a timestamp '2024-01-09 00:57:51' and a time display '10:37.4'.

To close a socket connection, you need to select a name for the connection.

## 2.7 Instruction blocks -SocketSend



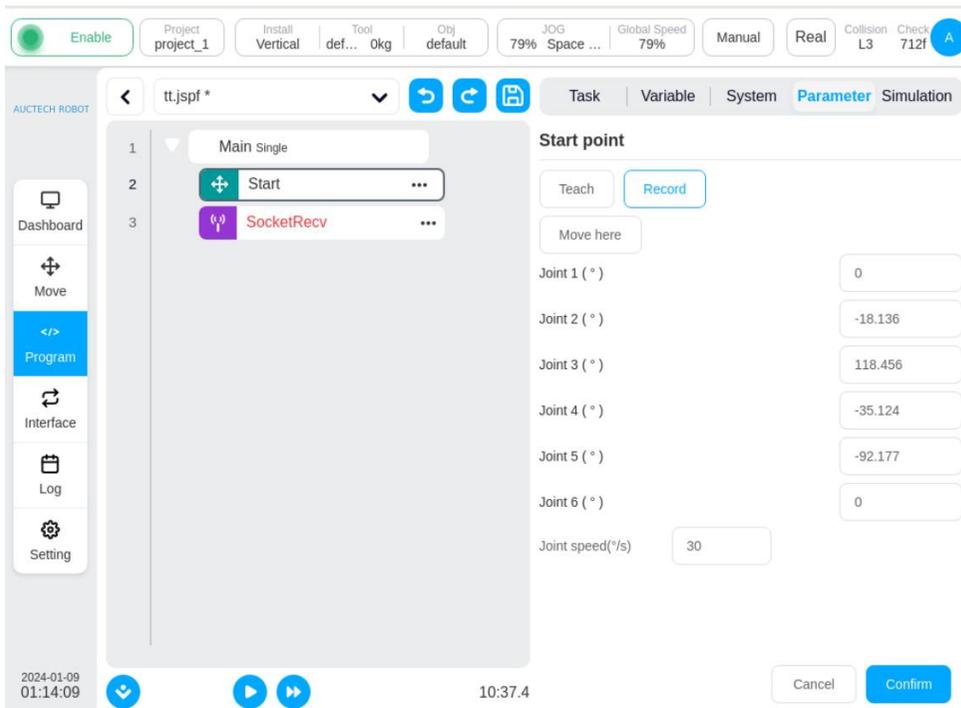
The screenshot shows the AUCTECH software interface. At the top, there are control buttons for 'Enable', 'Project project\_1', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG 79% Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check 712f'. The main workspace is titled 'tt.jspf \*' and contains a sequence of instruction blocks: 1. Main Single, 2. Start, 3. SocketOpen, 4. SocketClose, and 5. SocketSend. The 'Socket Send' panel is open, showing the following configuration options:

- Name: Select sock
- Type: String
- Send data: Custom
- Result: No return
- Comment: Comment

At the bottom of the interface, there are buttons for 'Cancel' and 'Confirm', along with a timestamp '2024-01-09 01:11:25' and a time '10:37.4'.

socket sends data. To send data to an established socket connection, you can **send string or floating-point array as the sending type**, you can select **a variable or direct input for sending data**, and you can bind the return value to a variable to get the sending status

## 2.7 Instruction blocks -SocketRecv



socket receives data

1. Receive data from the established socket connection, the **receiving type can be string, string array, floating point array**, and the receiving variable is configured for receiving information.

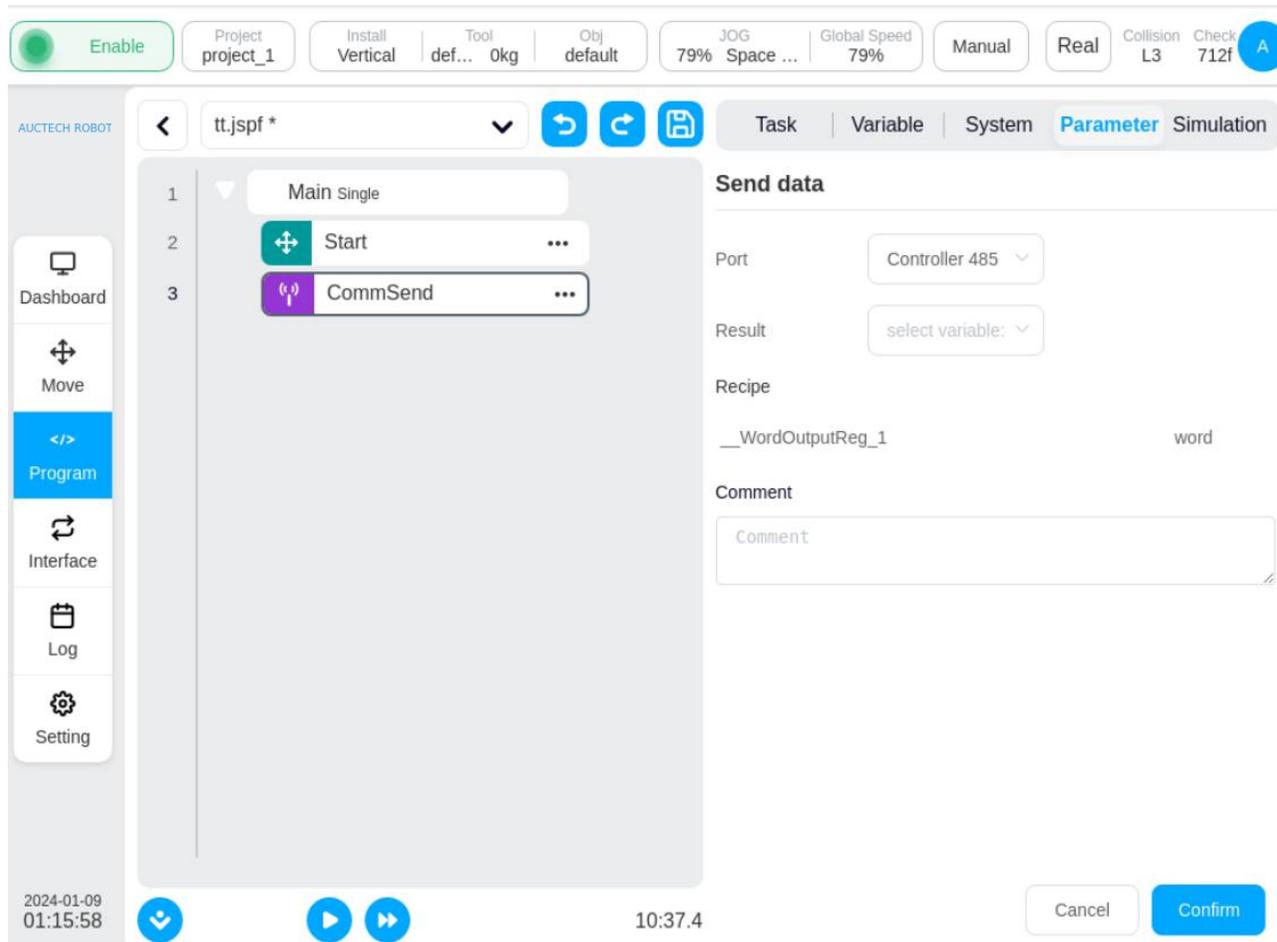
2. **When the string type is given**, the program will receive the data of the length and process it according to the string, and save it to the variable configured in "Receive Information".

3. **When the string array type is used**, the program will parse the received string into numeric values, and all the values are in "()", and the values are separated by ",". **For example, if a string of strings "(12,1.23)" is received from the socket, the function block converts it to a num\_list type with a value of {12,1.23} and saves it to the variable configured for "Receive Information".**

4. **When the floating-point array type is required**, the program will convert the received data into a set of single-precision floating-point numbers and save it to the variable configured in "Receive Information".

You can configure the receive timeout period, and if the data that meets the rule is not received within the timeout period, the next statement is executed.

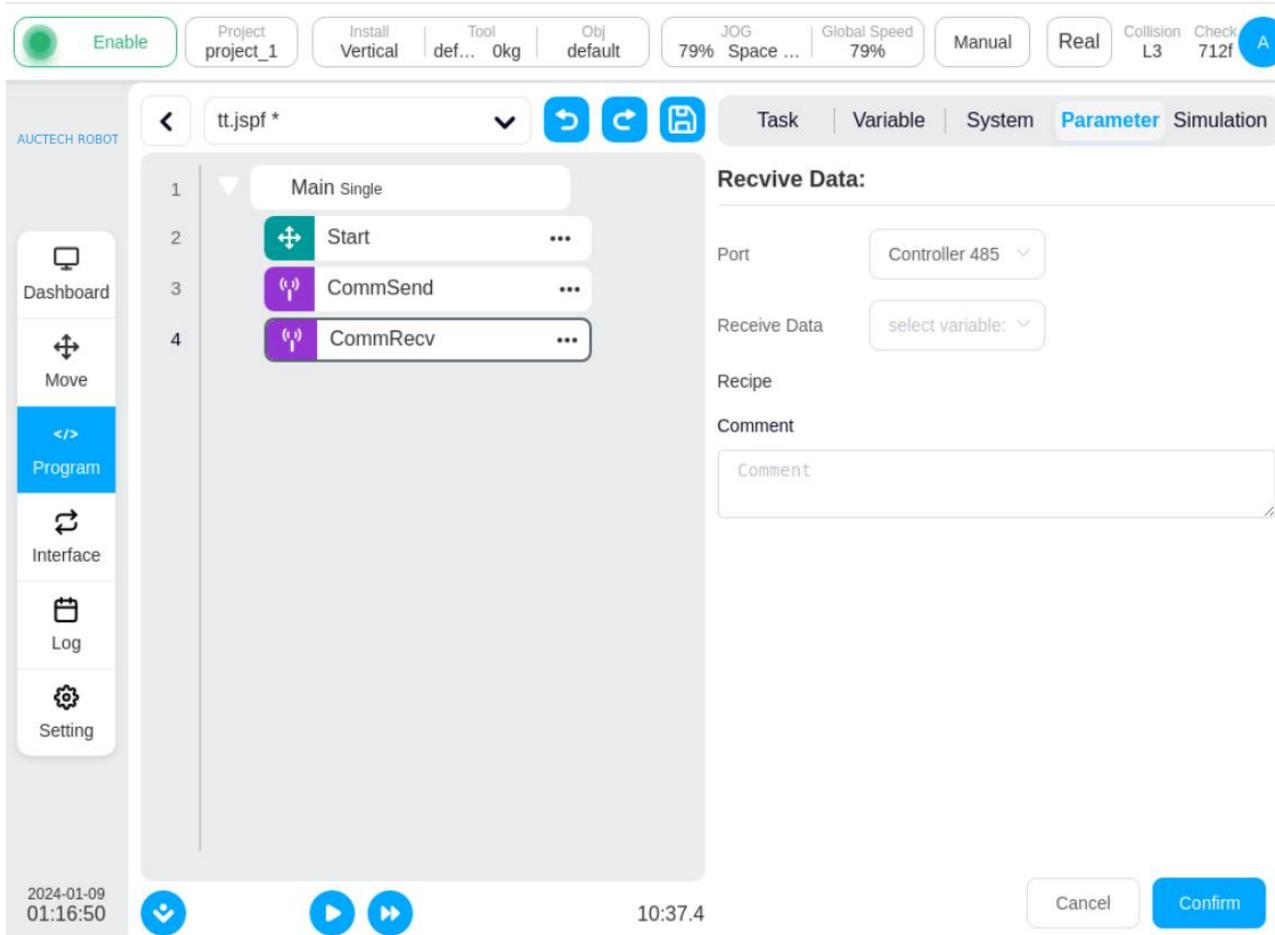
## 2.7 Instruction blocks -CommSend



The screenshot displays the AUCTECH software interface. At the top, there is a status bar with various indicators: 'Enable' (green), 'Project project\_1', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG 79%', 'Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', 'Check 712f', and a blue 'A' button. Below this is a navigation bar with 'Task', 'Variable', 'System', 'Parameter' (selected), and 'Simulation' tabs. The main workspace shows a ladder logic diagram with three steps: 1. 'Main Single', 2. 'Start' (green block), and 3. 'CommSend' (purple block). The 'CommSend' block is expanded, showing configuration options: 'Port' (Controller 485), 'Result' (select variable:), 'Recipe' (\_\_\_WordOutputReg\_1 word), and 'Comment' (empty text field). At the bottom, there are 'Cancel' and 'Confirm' buttons. The bottom left corner shows the date '2024-01-09' and time '01:15:58'. The bottom center shows a play button and a time '10:37.4'.

When the robot port 485 or CAN port is set with a recipe, you can use this function block to set the value for the data in the recipe and send it. You can set the status of the return variable to get the recipe data sent.

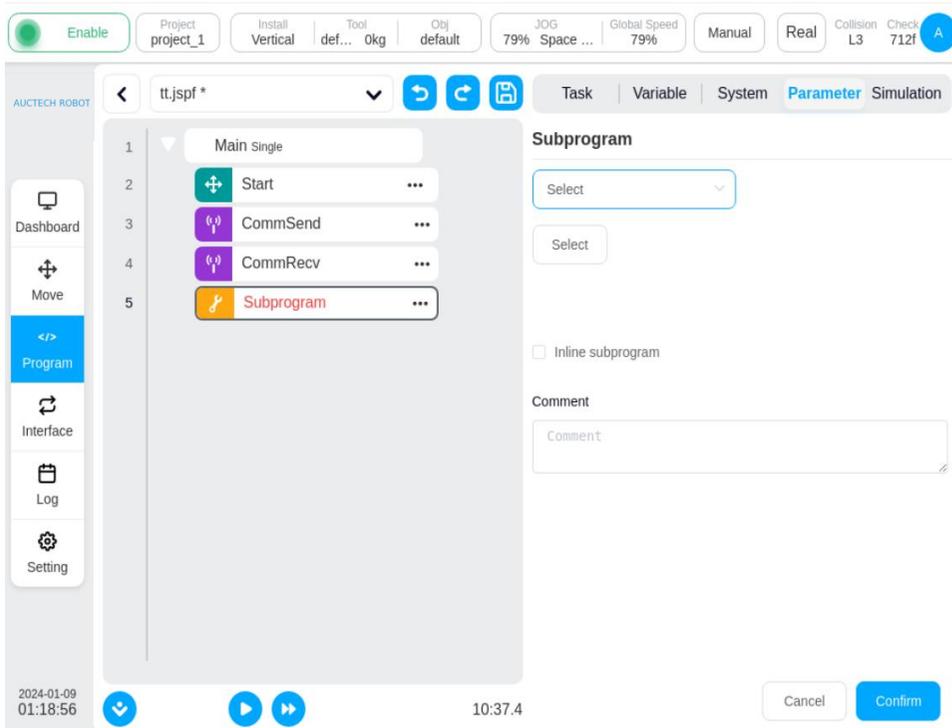
## 2.7 Instruction blocks -CommRecv



The screenshot displays the AUCTECH software interface. At the top, there is a control bar with buttons for 'Enable', 'Project project\_1', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG Space ... 79%', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check 712f'. Below this is a navigation bar with 'Task', 'Variable', 'System', 'Parameter', and 'Simulation' tabs. The main workspace shows a ladder logic diagram with a 'Main Single' block containing 'Start', 'CommSend', and 'CommRecv' blocks. The 'CommRecv' block is selected, and its configuration panel is open on the right. The 'Recv Data' panel includes a 'Port' dropdown set to 'Controller 485', a 'Receive Data' dropdown set to 'select variable:', and a 'Recipe' field. A 'Comment' text area is also present. At the bottom, there are 'Cancel' and 'Confirm' buttons. The status bar at the bottom left shows the date '2024-01-09' and time '01:16:50', and the bottom right shows the time '10:37.4'.

Recipe data receiving. When the formula is **set on port 485 or CAN port of the robot**, the function block is used to receive the data, and after processing according to the formula, the obtained num\_list data is assigned to the receiving variable.

## 2.8 Instruction blocks -Subprogram



Subroutine. Other programs can be embedded into the current program. There are two ways:

Select the embedded subroutine directly from the program list, if you do not check the “inline subroutine” option, **the subroutine will be loaded from the file every time the program is running**, that is, the change of the subroutine file will affect the main program. If checked, **the subroutine is copied directly to the main program**, and the changes to the subroutine file have no effect on it.

If it is set to a string variable, the program will dynamically load the corresponding subroutine according to the variable value as the subroutine name when it is running.

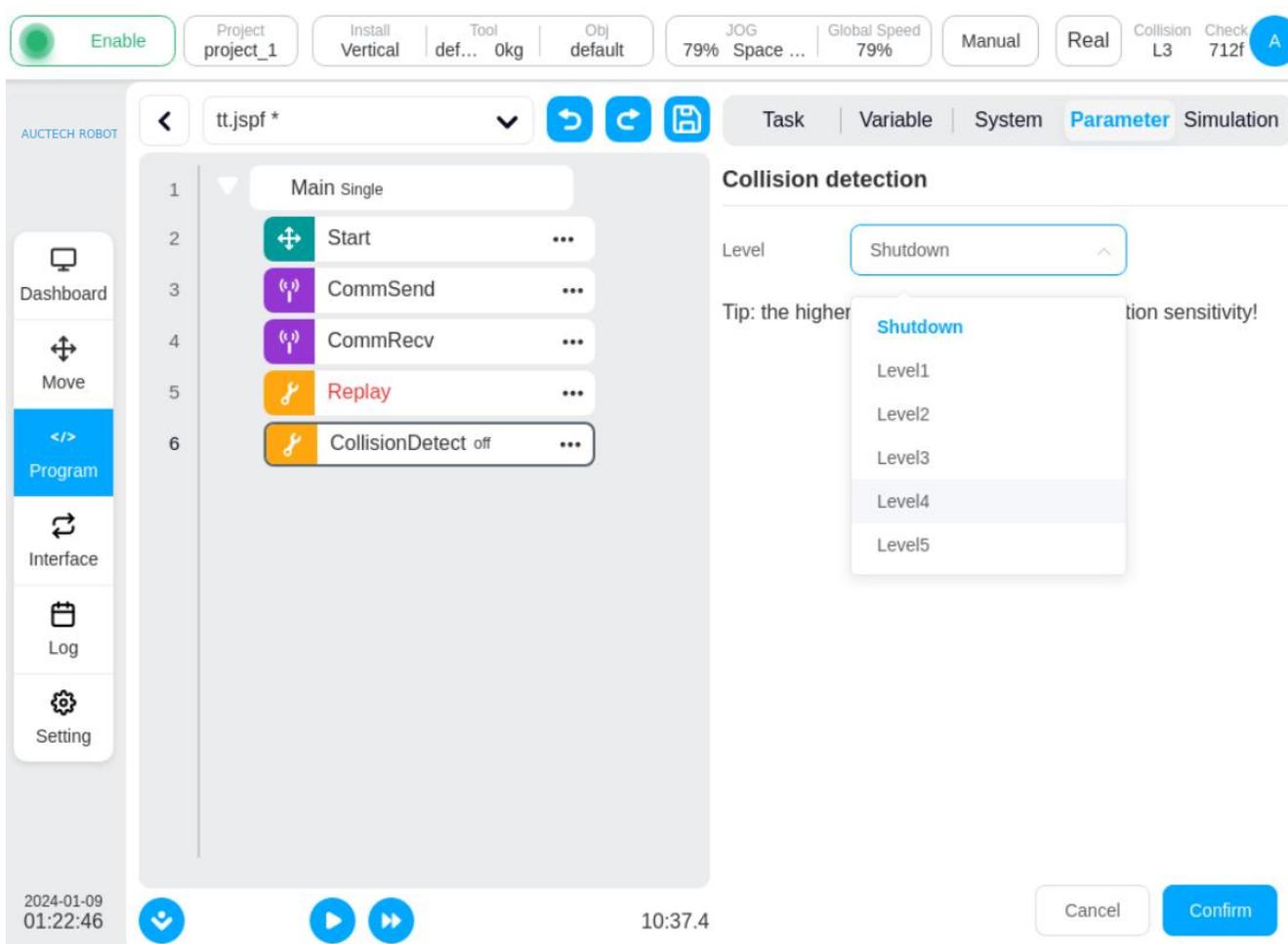
## 2.8 Instruction blocks -Replay



The screenshot displays the AUCTECH ROBOT software interface. At the top, there is a status bar with various indicators: 'Enable' (green), 'Project project\_1', 'Install Vertical', 'Tool def...', 'Obj 0kg default', 'JOG 79% Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check 712f'. Below this is a navigation menu with 'Task', 'Variable', 'System', 'Parameter', and 'Simulation'. The main workspace shows a program tree for 'tt.jspf \*' with a 'Main Single' block containing 'Start', 'CommSend', 'CommRecv', and 'Replay' blocks. The 'Replay' block is selected, and its configuration panel is open on the right. The 'Replay' panel includes a 'Track' field with 'Select track' and 'Create track' buttons, a 'Replay mode' dropdown set to 'Joint space', a 'Speed(%)' input field set to '30', and a 'Comment' text area. At the bottom of the interface, there are playback controls (play, stop, next) and a timestamp '2024-01-09 01:19:40' and '10:37.4'. A 'Cancel' button and a 'Confirm' button are also visible.

Trajectory recurrence. You can create a trajectory of the robot's motion, and the robot will move according to the trajectory when the program executes to this function block. You can select an existing track file or create a new track. How to create a new track, click "New Track", the "Track Recording" box will be displayed after entering the track name, click "Start Recording" to start recording the trajectory data, at this time, the page will display a translucent floating box indicating that the trajectory is being recorded, the user can use traction or jog to move the robot, click "Stop Recording" or the stop button on the floating frame, then complete the creation of the trajectory file.

## 2.8 Instruction blocks -CollisionDetect



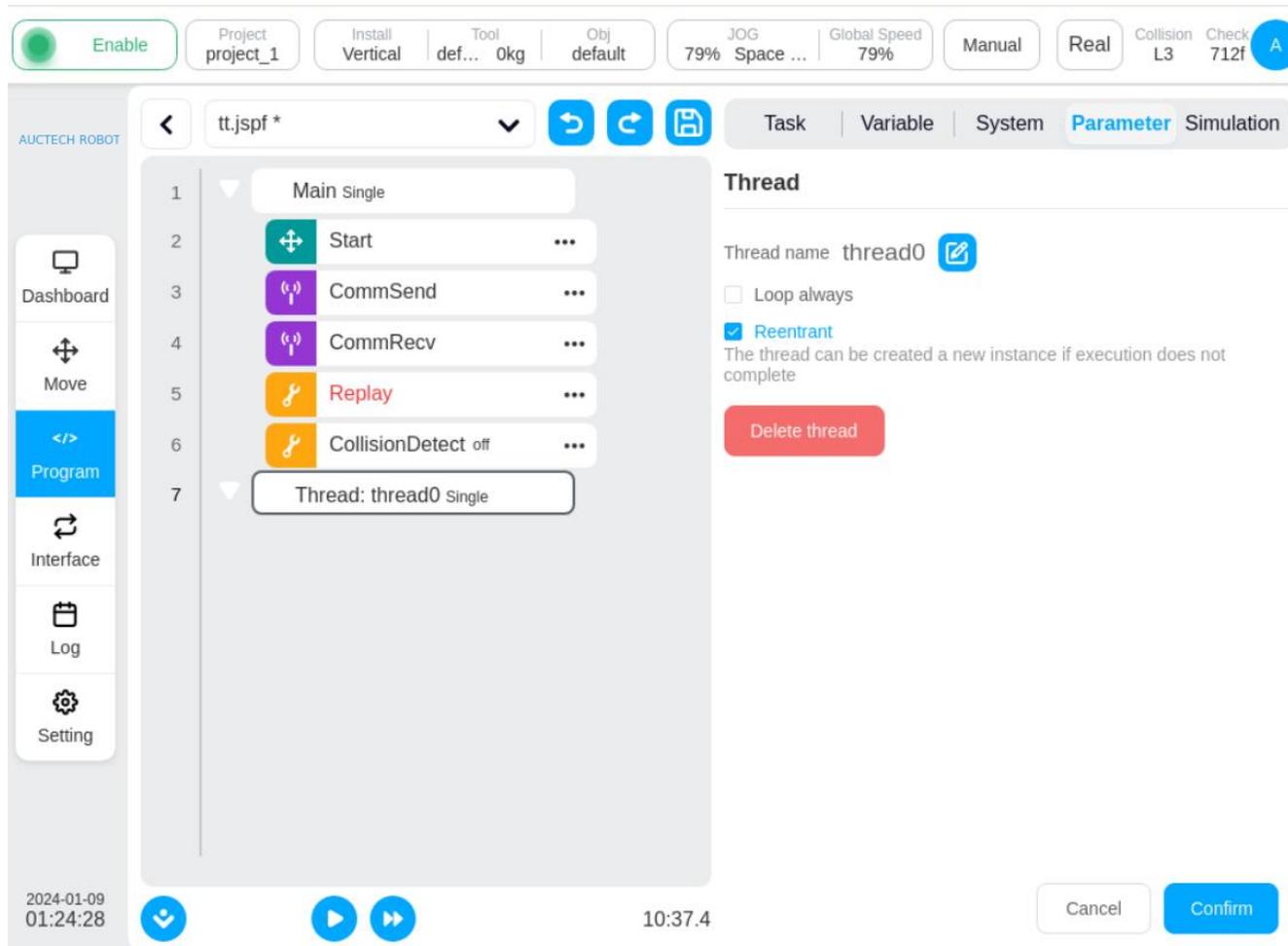
The screenshot displays the AUCTECH software interface. At the top, there is a status bar with various indicators: 'Enable' (green), 'Project project\_1', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG 79% Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check 712f'. Below this, a navigation bar shows 'Task', 'Variable', 'System', 'Parameter', and 'Simulation'. The main workspace is titled 'tt.jspf \*' and contains a list of instruction blocks under 'Main Single':

- 1. Main Single
- 2. Start
- 3. CommSend
- 4. CommRecv
- 5. Replay
- 6. CollisionDetect off

The 'CollisionDetect' block is selected, and its configuration panel is open. The 'Level' dropdown menu is set to 'Shutdown'. A tip below the dropdown reads: 'Tip: the higher the level, the higher the collision sensitivity!'. The dropdown menu lists the following options: Shutdown, Level1, Level2, Level3, Level4, and Level5. At the bottom of the configuration panel, there are 'Cancel' and 'Confirm' buttons. The bottom status bar shows the date '2024-01-09', time '01:22:46', and a timer '10:37.4'.

Collision detection level. This script can be used to set the sensitivity of collision detection during the program run.

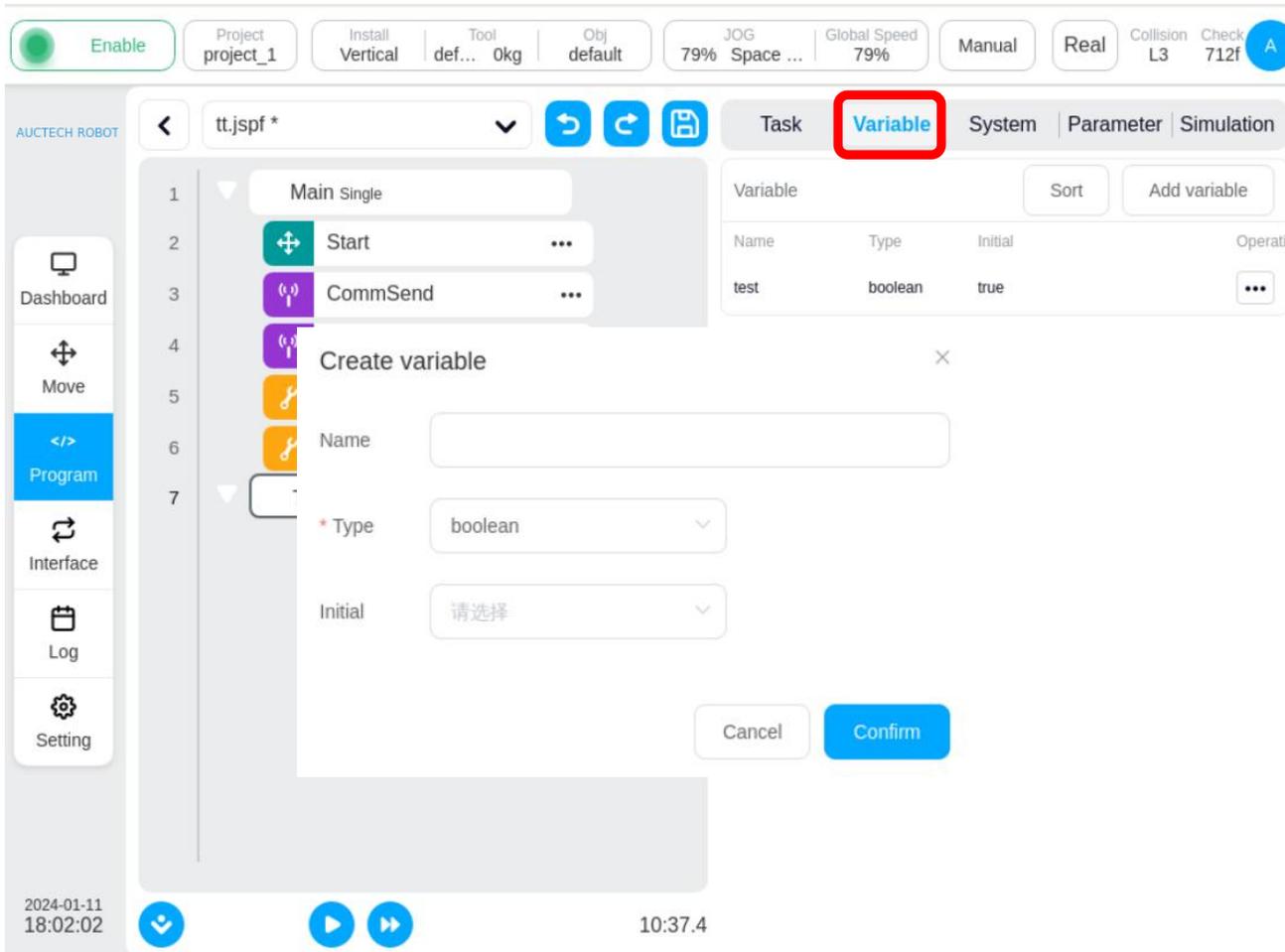
## 2.8 Instruction blocks -Thread



The screenshot displays the AUCTECH software interface for configuring a thread. The main program is 'tt.jspf \*'. The thread configuration area shows a thread named 'thread0' with the 'Reentrant' option checked. The thread configuration area also includes options for 'Loop always' and a 'Delete thread' button. The main program area shows a sequence of instruction blocks: Start, CommSend, CommRecv, Replay, and CollisionDetect off. The thread configuration area also shows a 'Thread: thread0 Single' block.

Threads, executed in parallel with the main program of the robot, interact with the main program through variables. In the parameter configuration area, you can set whether the thread keeps looping. A maximum of 10 threads are allowed to be created in a program. Note: Robot motion instructions are not allowed in the thread

## 2.9 Variable



Program variables are of the following types:

**boolean**: boolean num\_list: array type

**number**: Number type string: String type

**pose**: The type of data that represents the Cartesian position of the bot

**joint**: The type of data that represents the position of the robot's joints

**pos\_list**: Cartesian position array Joint\_list: joint position array

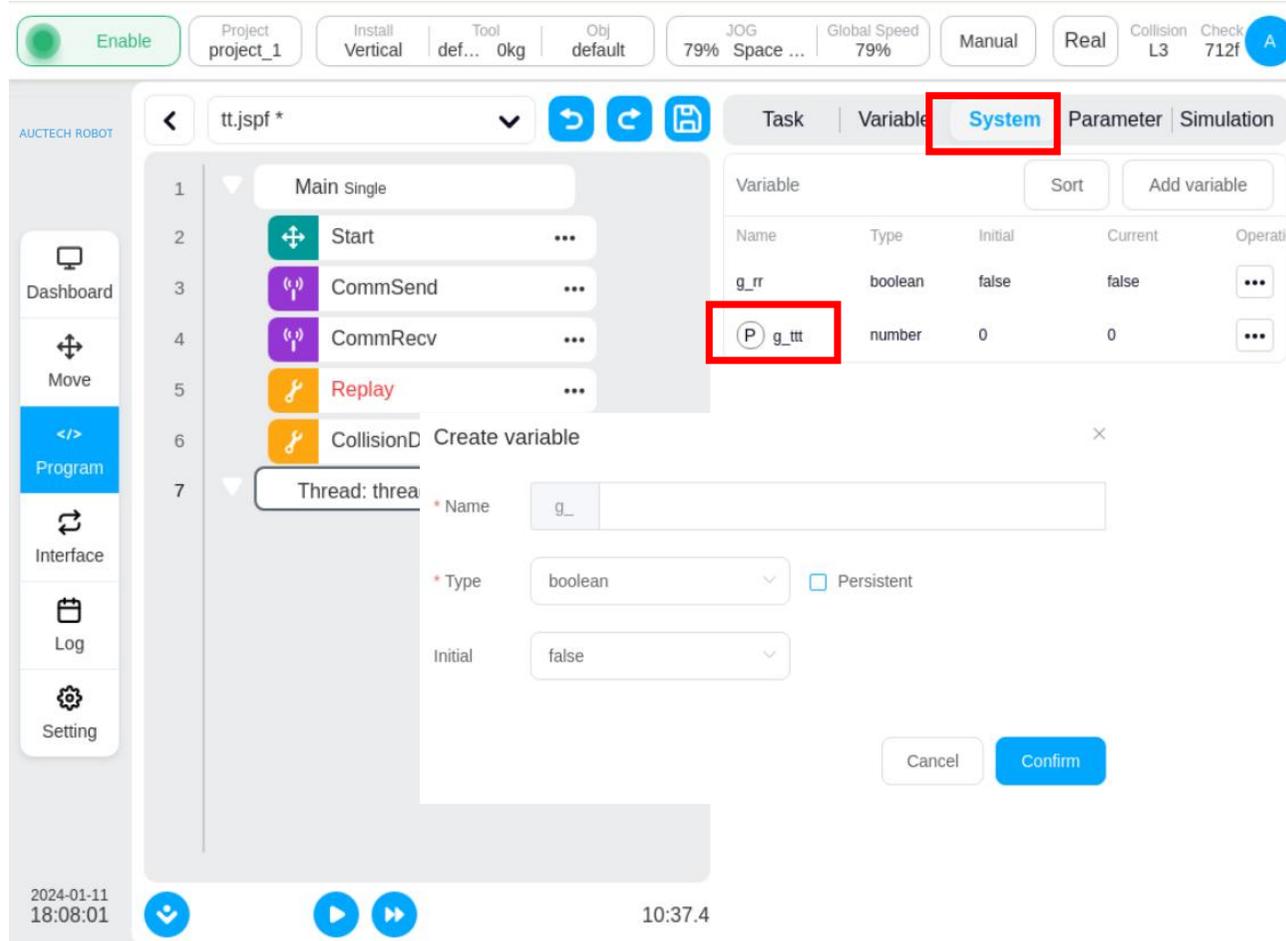
**pose\_speed**: Cartesian position velocity

**joint\_speed**: Joint velocity

**pose\_acc**: Cartesian position acceleration

**joint\_acc**: Joint acceleration

## 2.9 System variable



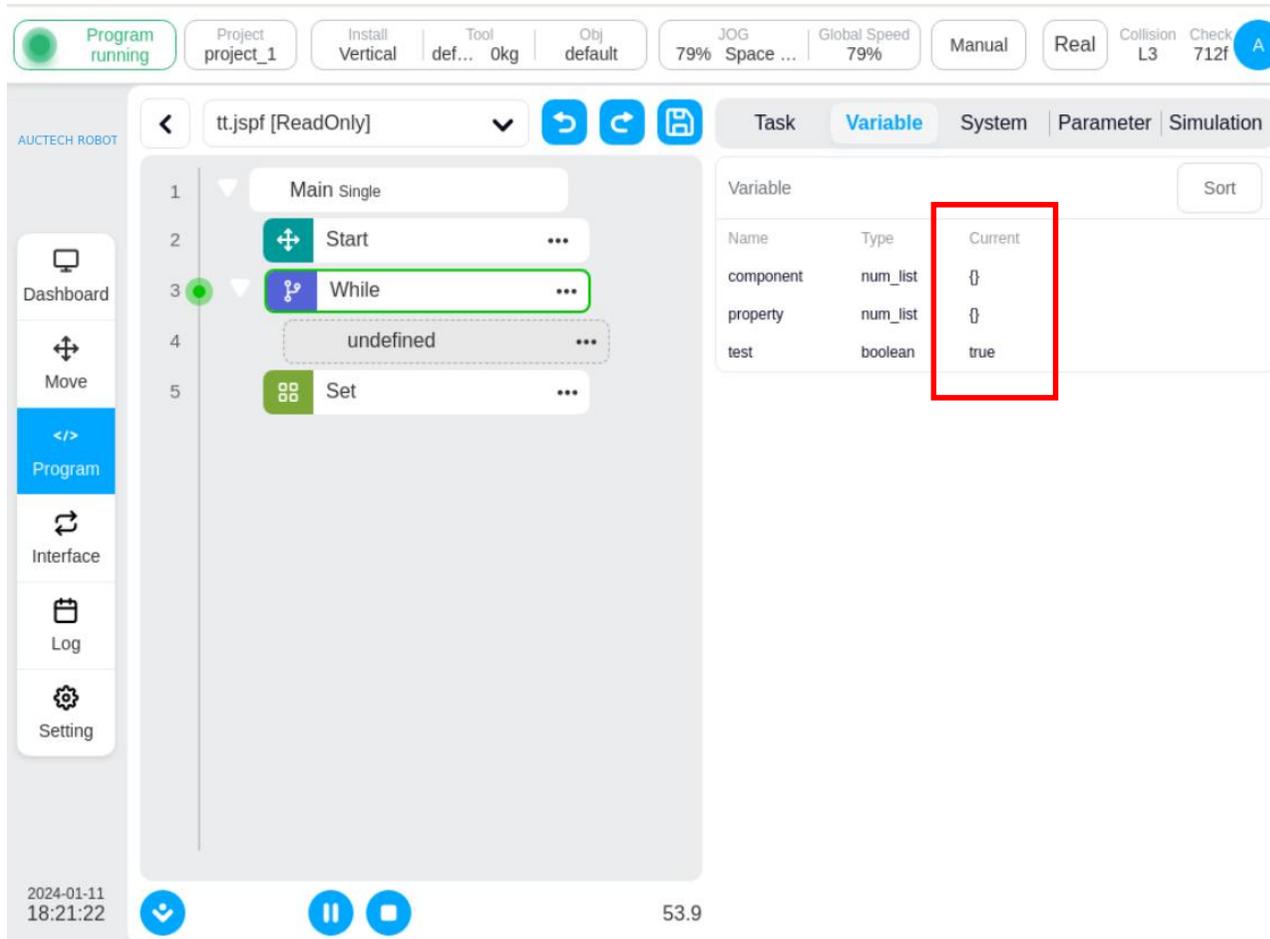
The screenshot shows the AUCTECH software interface. At the top, there are status indicators for 'Enable', 'Project project\_1', 'Install Vertical', 'Tool def...', 'Obj 0kg default', 'JOG 79%', 'Space ...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check 712f'. Below this, the 'System' tab is selected, and a table of variables is displayed. The table has columns for Name, Type, Initial, Current, and Operati. The variable 'g\_ttt' is highlighted with a red box, and a 'P' icon is visible next to its name, indicating it is a persistent variable. A 'Create variable' dialog is also visible, showing fields for Name (g\_), Type (boolean), Initial (false), and a checkbox for Persistent.

Name	Type	Initial	Current	Operati
g_rr	boolean	false	false	...
P g_ttt	number	0	0	...

The **word P** is displayed in front of the name of the **persistent variable**.

The variables added in the settings are system variables, which can be used by each program in the current project.

## 2.10 Variable monitoring



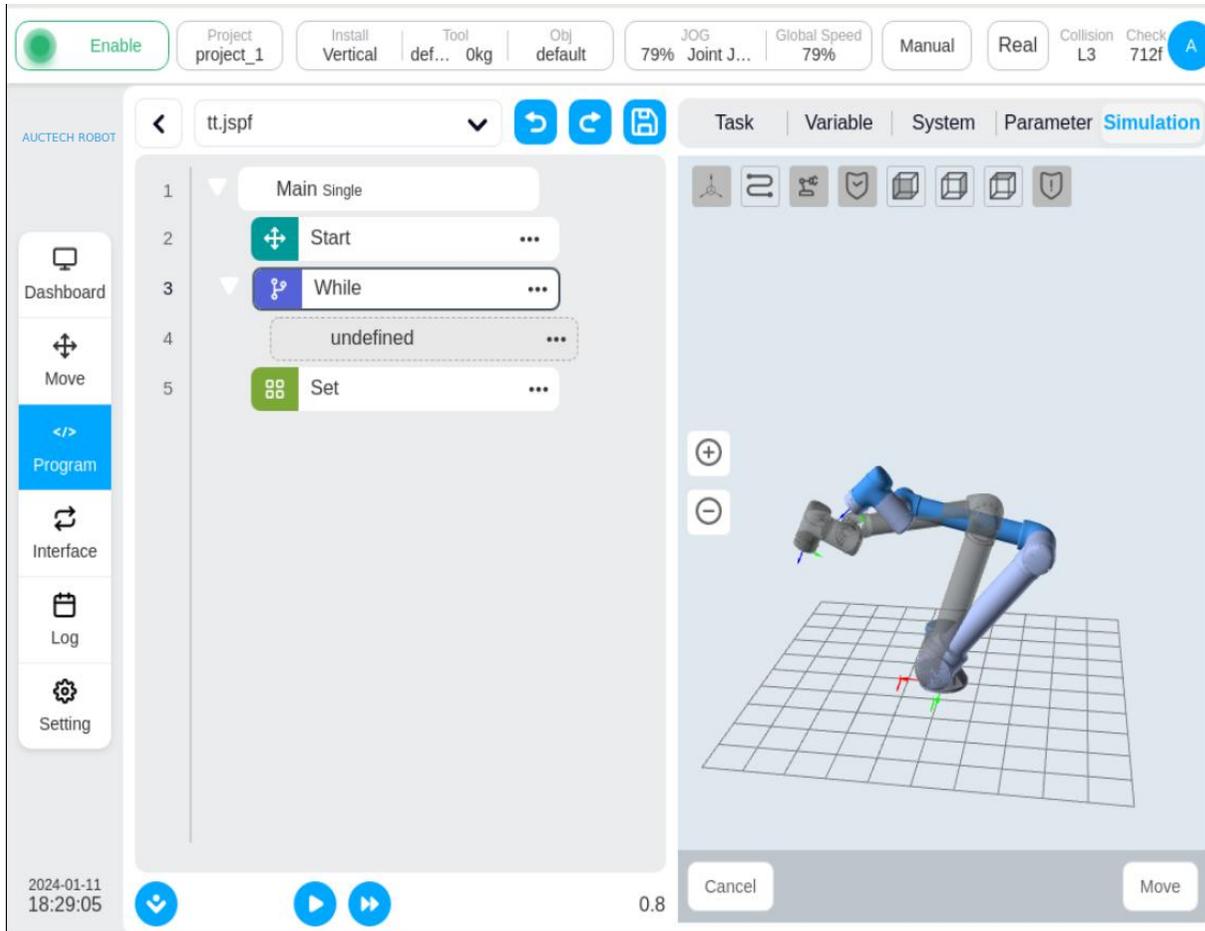
The screenshot shows the AUCTECH software interface with the 'Variable' tab selected. The interface displays a table of current values for various variables. A red box highlights the 'Current' column in the table.

Variable	Type	Current
component	num_list	{}
property	num_list	{}
test	boolean	true

While the program is running, this page can display the current values of program variables and system variables.

**Note:** Variables of type JOINT and POSE are both displayed as **num\_list** during the monitoring period, and the data unit is displayed in m/rad. If the data is too long, you can use the "log" function block to print the variable to the log for viewing.

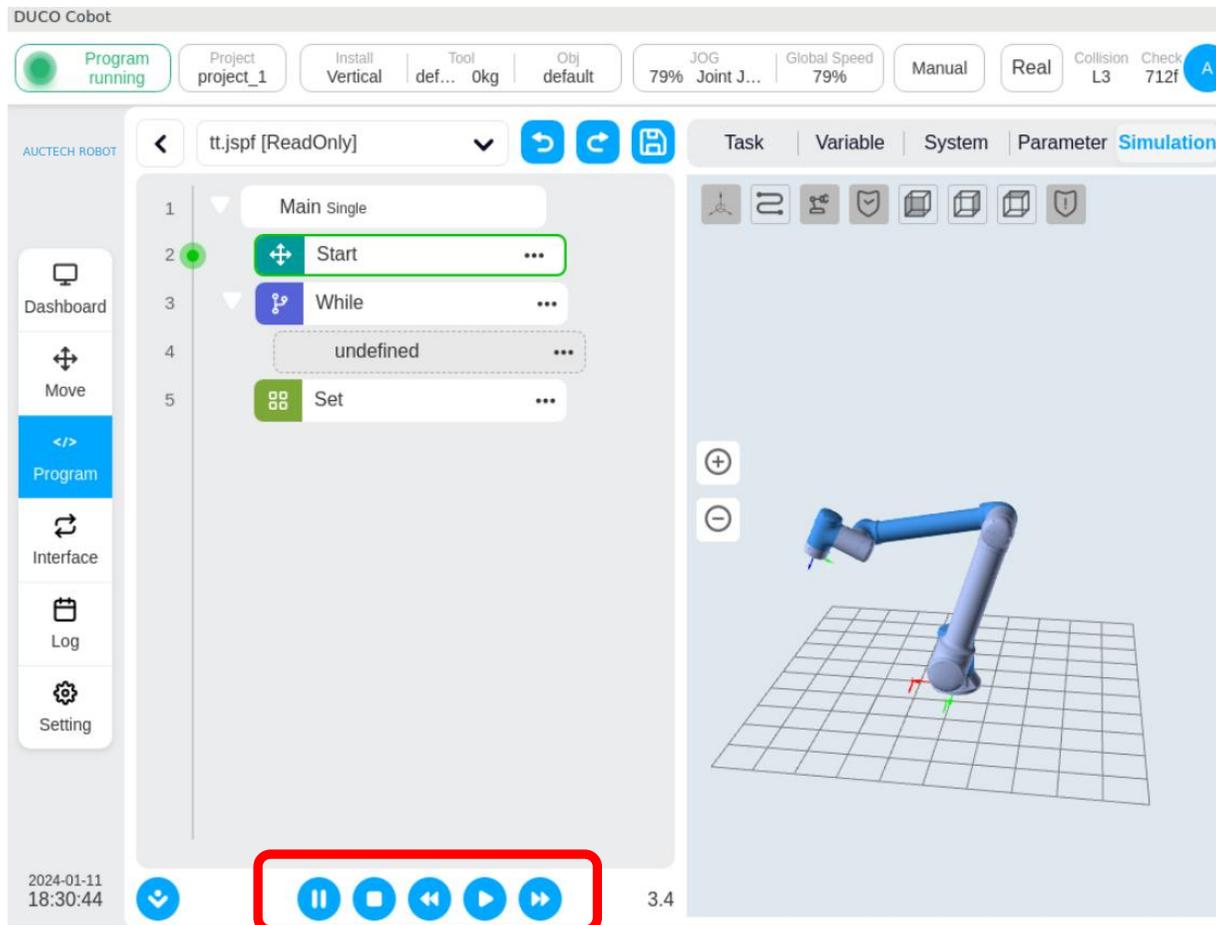
## 2.11 Run



Click on the "Run" tab to enter the running page, which controls the start and stop of the program and displays the 3D model of the robot. Click the button to run the program, and click the button to run the program in a step-by-step manner. You can pause or stop the program at any time while it is running.

**Note:** If the actual position of the robotic arm is inconsistent with the Start point, you need to press and hold "Press and Hold" to manually move to the Start point.

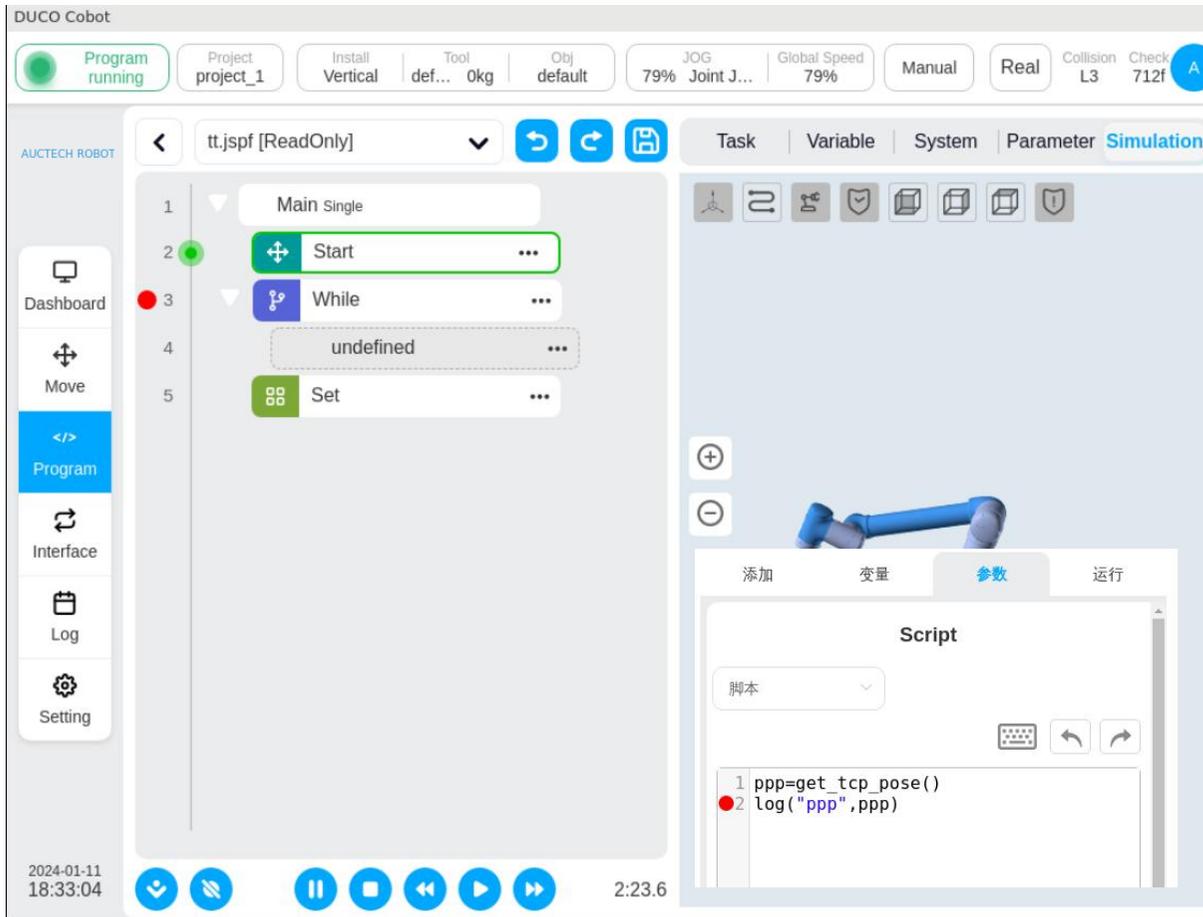
## 2.12 Run - Single-step



◀◀ Go back to the previous step

Note: For sports function  
blocks only

## 2.13 Breakpoint debugging



The screenshot shows the AUCTECH robot control software interface. The top status bar indicates the program is running. The main workspace displays a program tree for 'tt.jspf [ReadOnly]' with a 'Main Single' block containing a 'Start' block at line 2, a 'While' block at line 3, an 'undefined' block at line 4, and a 'Set' block at line 5. A red dot on line 2 indicates a breakpoint. The right panel shows a 'Script' block with the following code:

```

1 ppp=get_tcp_pose()
2 log("ppp",ppp)

```

The interface also includes a sidebar with navigation options (Dashboard, Move, Program, Interface, Log, Setting) and a bottom control bar with playback buttons and a timestamp of 2:23.6.

When the program runs to a breakpoint, you can choose to continue running, or run forward and backward step by step.

Note:

1. "Cancel Breakpoint" is all canceled!!
2. Breakpoints can also be added to the "Script" script function block.
3. When calling a subroutine, adding a breakpoint to the subroutine is invalid.



✦ Lecture 3 ✦

# Interface

---

## 3.1 Digital\_I/O



Type	Name	Description	Modbus	Status
Digital In 1	DI1		176	0
Digital In 2	DI2		177	0
Digital In 3	DI3		178	0
Digital In 4	DI4		179	0
Digital In 5	DI5		180	0
Digital In 6	DI6		181	0
Digital In 7	DI7		182	0
Digital In 8	DI8		183	0

There are a total of **16 sets of input and output IOs**, of which **8 groups are configurable IOs**, which can be configured as functional IOs. Functional IO currently supports the following functions:

**Inputs:** Run Program, Pause Program, Stop Program and Traction Teaching, Collision Detection Reset, Robot Power-On, Robot Enable, Robot Disable, Robot Power-off.

**Outputs:** Idle Status, Program Running Status, Paused Status, Program End, Profinet Disconnect, Home Position, Collision Trigger, Auto Mode, Robot Power-Off Status, Robot Disable Enabled State

# 3.1 Analog in/out



Program running | Project project\_1 | Install Vertical | Tool def... 0kg | Obj default | JOG 79% Joint J... | Global Speed 79% | Manual | Real | Collision L3 | Check 712f | A

I/O | Register | CCI | TCI | TCP/IP | Industrial Bus

Digital In | Digital Out | Function In | Function Out | **Analog In** | Analog Out

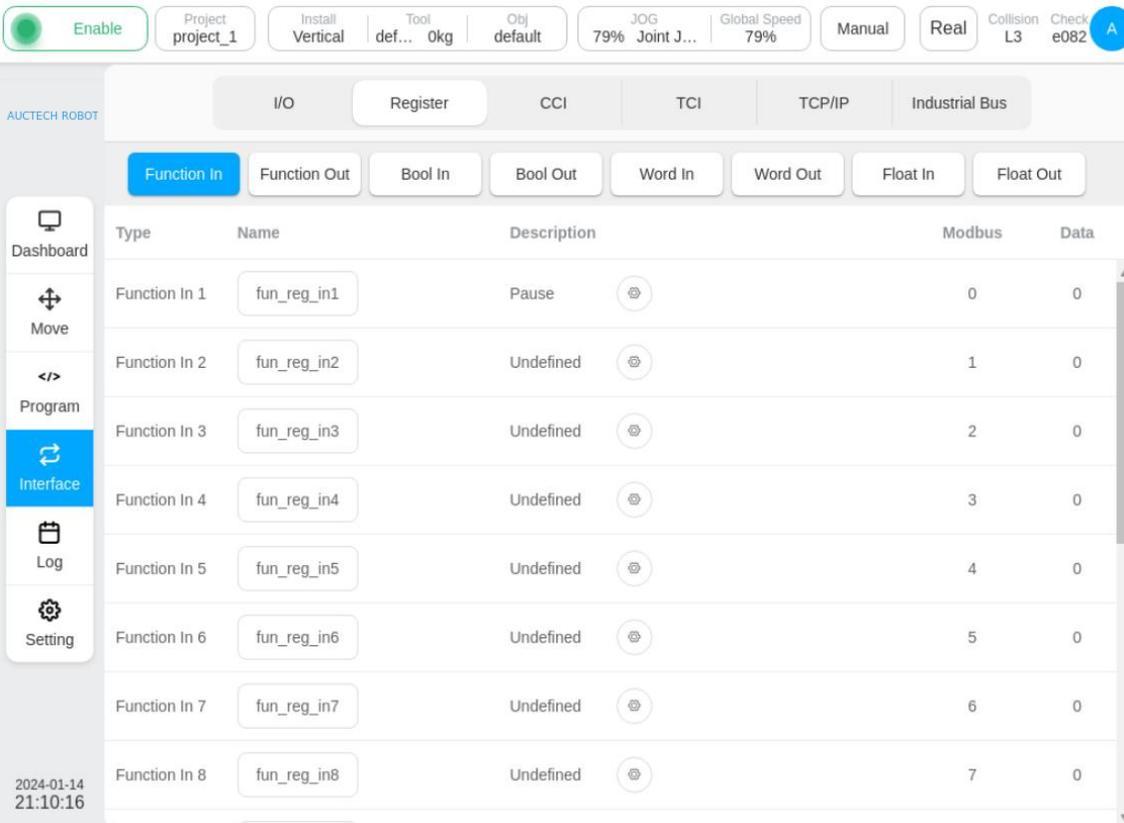
Type	Name	Description	Modbus	Data(mA/V)
Analog In 1	AI_C1	Current In	72	0.00
Analog In 2	AI_V2	Voltage In	77	0.00

Dashboard | Move | Program | **Interface** | Log | Setting

2024-01-11 18:36:01

2 configurable analog inputs and outputs

## 3.2 Register-Function in



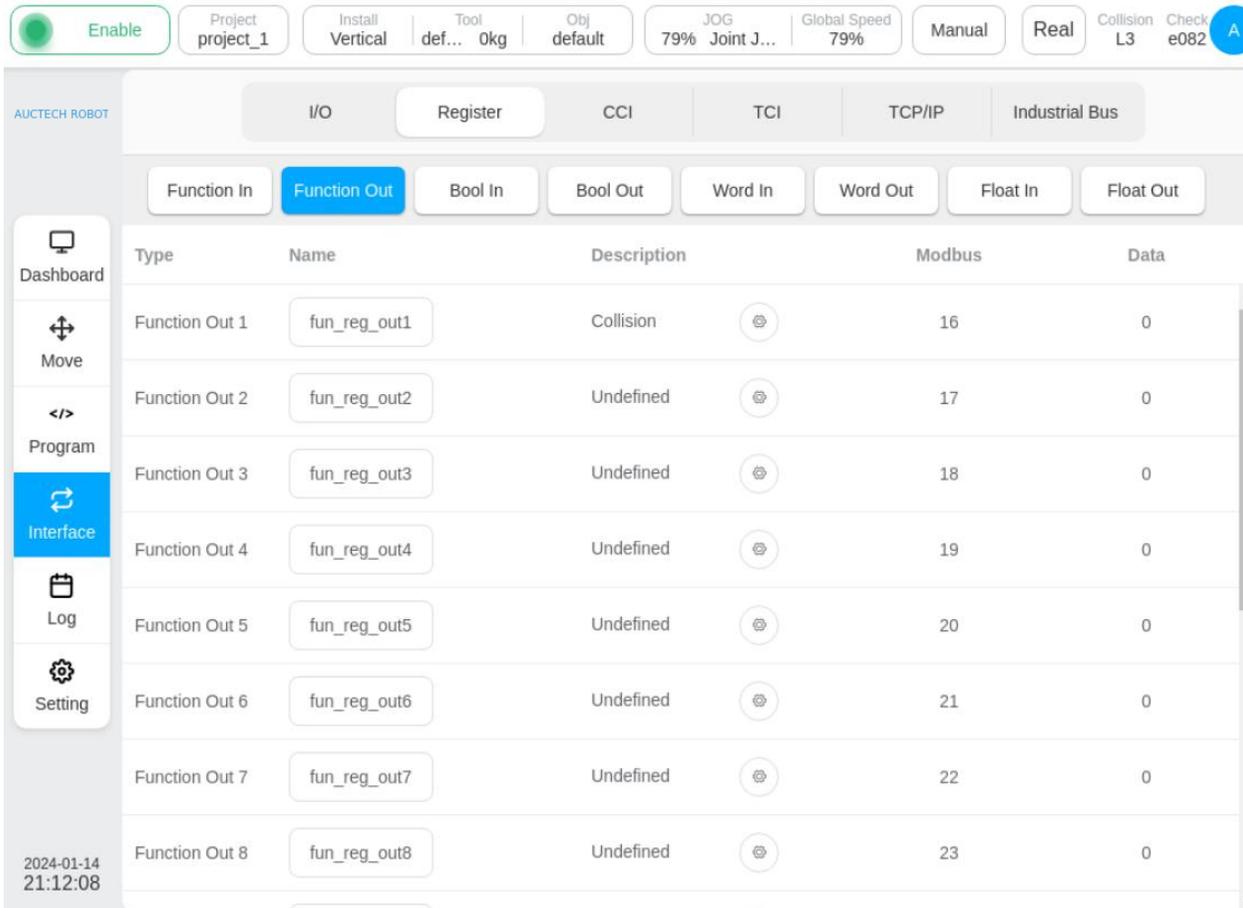
Type	Name	Description	Modbus	Data
Function In 1	fun_reg_in1	Pause	0	0
Function In 2	fun_reg_in2	Undefined	1	0
Function In 3	fun_reg_in3	Undefined	2	0
Function In 4	fun_reg_in4	Undefined	3	0
Function In 5	fun_reg_in5	Undefined	4	0
Function In 6	fun_reg_in6	Undefined	5	0
Function In 7	fun_reg_in7	Undefined	6	0
Function In 8	fun_reg_in8	Undefined	7	0

The **16-channel function input** of the register currently supports the following functions: running program, pausing program, stopping program and traction teaching, collision detection reset, robot power-on and power-off, robot up and down enablement, system shutdown, and recording point.

### Notes:

1. **Change the register name** to conform to the naming rule "2-24 digits starting with **a letter. Letter. Underline combination**"
2. When defined as "running program", the following selection program cannot be empty
3. **The signal does not need to be triggered continuously.** For example: "Power off", "Pause program", etc.

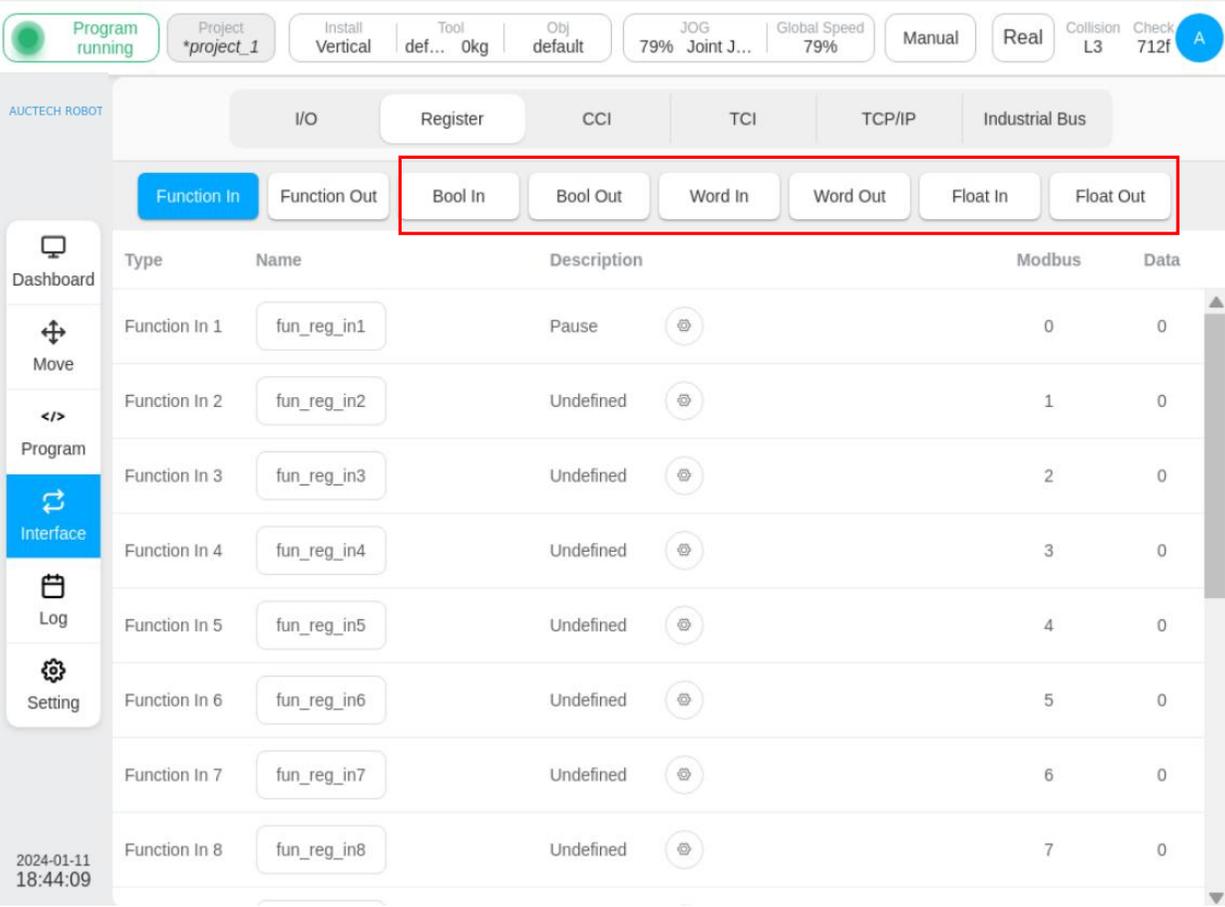
## 3.2 Register-Function out



Type	Name	Description	Modbus	Data
Function Out 1	fun_reg_out1	Collision	16	0
Function Out 2	fun_reg_out2	Undefined	17	0
Function Out 3	fun_reg_out3	Undefined	18	0
Function Out 4	fun_reg_out4	Undefined	19	0
Function Out 5	fun_reg_out5	Undefined	20	0
Function Out 6	fun_reg_out6	Undefined	21	0
Function Out 7	fun_reg_out7	Undefined	22	0
Function Out 8	fun_reg_out8	Undefined	23	0

The 16 function outputs of the register currently support the following states: idle state, program running state, paused state, program end, profinet disconnect, home position, collision trigger, automatic mode, not powered on, enabled, robot arm in motion, and ethernet disconnected

## 3.2 Register



Program running | Project \*project\_1 | Install Vertical | Tool def... 0kg | Obj default | JOG 79% Joint J... | Global Speed 79% | Manual | Real | Collision L3 | Check 712f | A

I/O | Register | CCI | TCI | TCP/IP | Industrial Bus

Function In | Function Out | Bool In | Bool Out | Word In | Word Out | Float In | Float Out

Type	Name	Description	Modbus	Data
Function In 1	fun_reg_in1	Pause	0	0
Function In 2	fun_reg_in2	Undefined	1	0
Function In 3	fun_reg_in3	Undefined	2	0
Function In 4	fun_reg_in4	Undefined	3	0
Function In 5	fun_reg_in5	Undefined	4	0
Function In 6	fun_reg_in6	Undefined	5	0
Function In 7	fun_reg_in7	Undefined	6	0
Function In 8	fun_reg_in8	Undefined	7	0

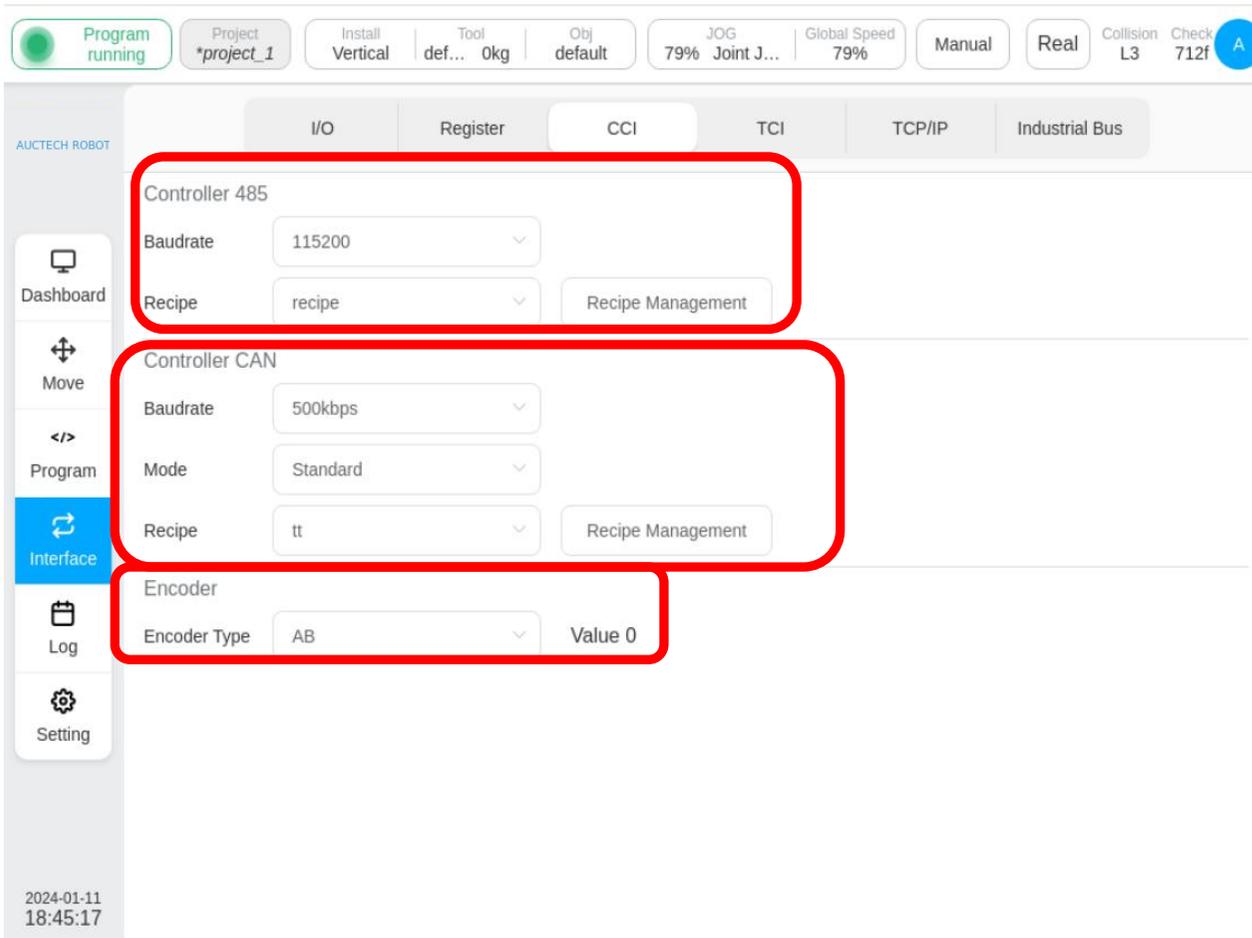
2024-01-11 18:44:09

64-way Boolean type input and output  
32 byte input and output  
32 floating-point inputs and outputs

## ◆ 3.2 Register\_Precautions

1. Change the register name to conform to the naming rule "2-24 digits starting **with a letter. Letter. Underline Combination**".
2. When the function input is defined as "Run Program":
  - (1) If the backloading program is empty, then this register will continue to run after pausing, and when the robotic arm is idle, when this register is 1, the robotic arm will not respond.
  - (2) If the loading program is set later. When the robot arm is idle, set this register to 1, the robot arm runs the currently loaded program.
3. **The function input (including function IO) signal should not be triggered for a long time**, such as triggering "power off" for a long time, "running the program", etc.
4. **Be sure to save the project after the change!**
5. In the registers and IO signals, the output signal can be manually changed (except for the function output).

## 3.3 CCI



The screenshot displays the AUCTECH control interface with the 'CCI' tab selected. The interface includes a top status bar with 'Program running' and various system parameters. The main configuration area is divided into three sections, each highlighted with a red box:

- Controller 485:** Baudrate is set to 115200, and Recipe is set to 'recipe'.
- Controller CAN:** Baudrate is set to 500kbps, Mode is set to 'Standard', and Recipe is set to 'tt'.
- Encoder:** Encoder Type is set to 'AB' and Value is set to 0.

The left sidebar contains navigation options: Dashboard, Move, Program, Interface (selected), Log, and Setting. The bottom left corner shows the date and time: 2024-01-11 18:45:17.

**Port 485** supports the following baud rates: 9600, 19200, 38400, 57600, and 115200

The **CAN port** can be configured with baud rates of 10kbps, 20kbps, 50kbps, 100kbps, 125kbps, 250kbps, 500kbps, and 1000kbps

There are **two types of encoder** types in the control cabinet, namely AB and ABZ (the number of ABZ encoder lines, default is 0)

## 3.4 TCI



Program running | Project \*project\_1 | Install Vertical | Tool def... 0kg | Obj default | JOG 79% Joint J... | Global Speed 79% | Manual | Real | Collision L3 | Check 712f

I/O | Register | CCI | **TCI** | TCP/IP | Industrial Bus

Tool IO information

Type	Name	Description	Modbus	Status
Tool In1	Tool_DI1		208	<input type="checkbox"/>
Tool In2	Tool_DI2		209	<input type="checkbox"/>
Tool Out1	Tool_DO1	PNP pattern	216	<input type="checkbox"/>
Tool Out2	Tool_DO2	PNP pattern	217	<input type="checkbox"/>

ResuePort

Function: Analog input

Tool_AI1	0.00 V	Modbus: 88
Tool_AI2	0.00 V	Modbus: 89

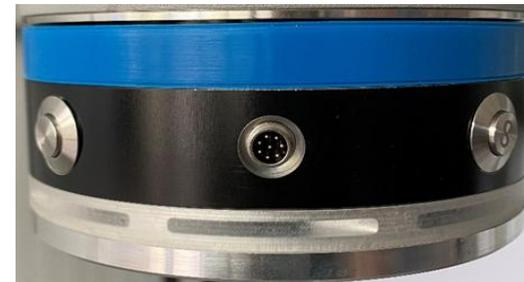
2024-01-11 18:49:04

On the end of the robotic arm there are a total of:

2-way tool inputs

2 tool outputs, **PNP or NPN mode can be selected.**

Reusable ports that can be used for **analog inputs or 485 communication** ports.



## 3.5 TCP/IP



Port 2000 can receive control commands from the robot. The information received by port 2000 is displayed in the lower area of the page.

The 2001 port sends information about the current status of the robot at a frequency of 10Hz.

The 2011 port is a UDP port that allows the robot to be controlled through the data items configured by the recipe.

For details about the use of ports 2000 and 2001, see [Cobot External Communication Interface](#).

For details about how to use the 2011 port, see [Recipes](#).



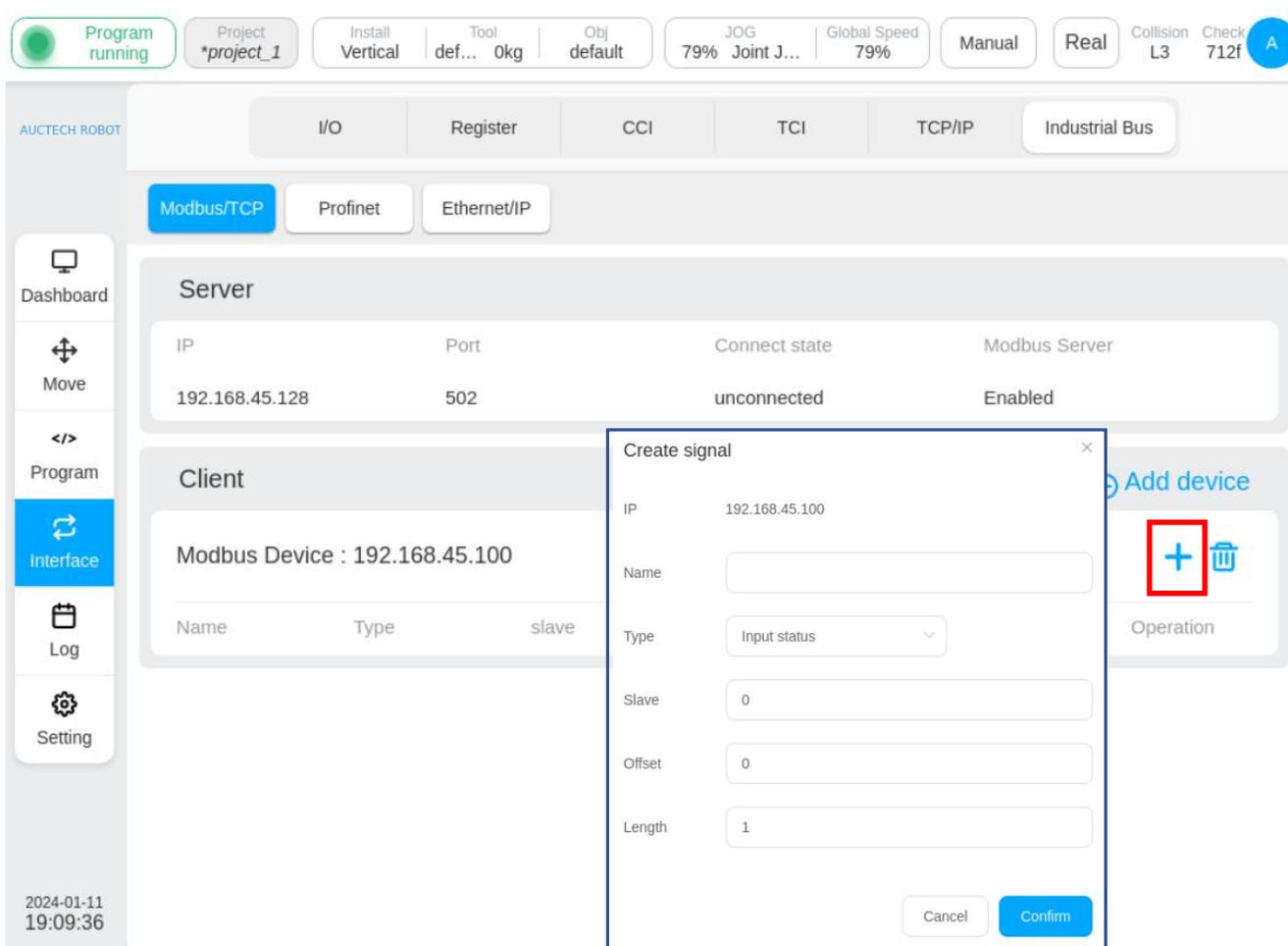
# 3.5 TCP/IP

## Port 2000



Instructions received	Return value	
run(Program name) or run(Program name, speed percentage)	Feedback at the beginning: run start	Execution failure feedback: run fail
speed	Perform success feedback: set success	Execution failure feedback: set fail
stop	Perform success feedback: stop success	Execution failure feedback: stop fail
pause	Perform success feedback: pause success	Execution failure feedback: pause fail
resume	Perform success feedback: resume success	Execution failure feedback: resume fail
login	Perform success feedback: login success	Execution failure feedback: login fail
logout	Perform success feedback: logout success	Execution failure feedback: logout fail
poweron	Perform success feedback: poweron success	Execution failure feedback: poweron fail
poweroff	Perform success feedback: poweroff success	Execution failure feedback: poweroff fail
enable	Perform success feedback: enable success	Execution failure feedback: enable fail
disable	Perform success feedback: disable success	Execution failure feedback: disable fail
shutdown	Perform success feedback: shutdown success	Execution failure feedback: shutdown fail
state	Feedback on the status of each system, format: (1):(2):(3):(4) concentrate: (1) Robot status, (2) Program status, (3) Safety controller status, (4) Robot operation mode	
isprogfinish	Whether the feedback program runs at the end of the program, the end feedback character is 1, and the unfinished feedback character is 0	
getlasterror	Feedback detailed error information for the bot, format: error_id:error_message\n	

## 3.6 MODBUS/TCP



Program running

Project \*project\_1

Install Vertical

Tool def... 0kg

Obj default

JOG 79% Joint J...

Global Speed 79%

Manual

Real

Collision L3

Check 712f

AUCTECH ROBOT

I/O Register CCI TCI TCP/IP Industrial Bus

Modbus/TCP Profinet Ethernet/IP

Dashboard

Move

Program

Interface

Log

Setting

2024-01-11 19:09:36

IP	Port	Connect state	Modbus Server
192.168.45.128	502	unconnected	Enabled

Name	Type	slave
Modbus Device : 192.168.45.100		

Create signal

IP 192.168.45.100

Name

Type Input status

Slave 0

Offset 0

Length 1

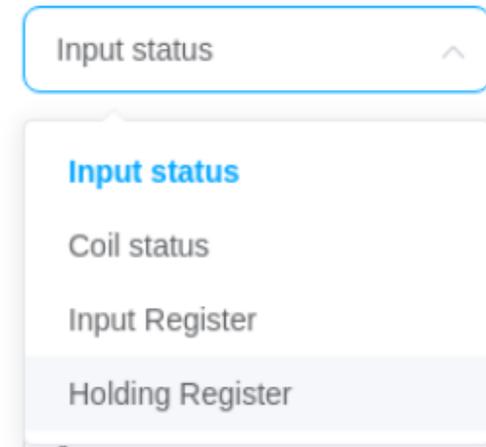
Cancel Confirm

Add device

+

Operation

Modbus Communication:  
When the robotic arm is used as a client, external devices can be added, such as modbus to io, etc.



Input status

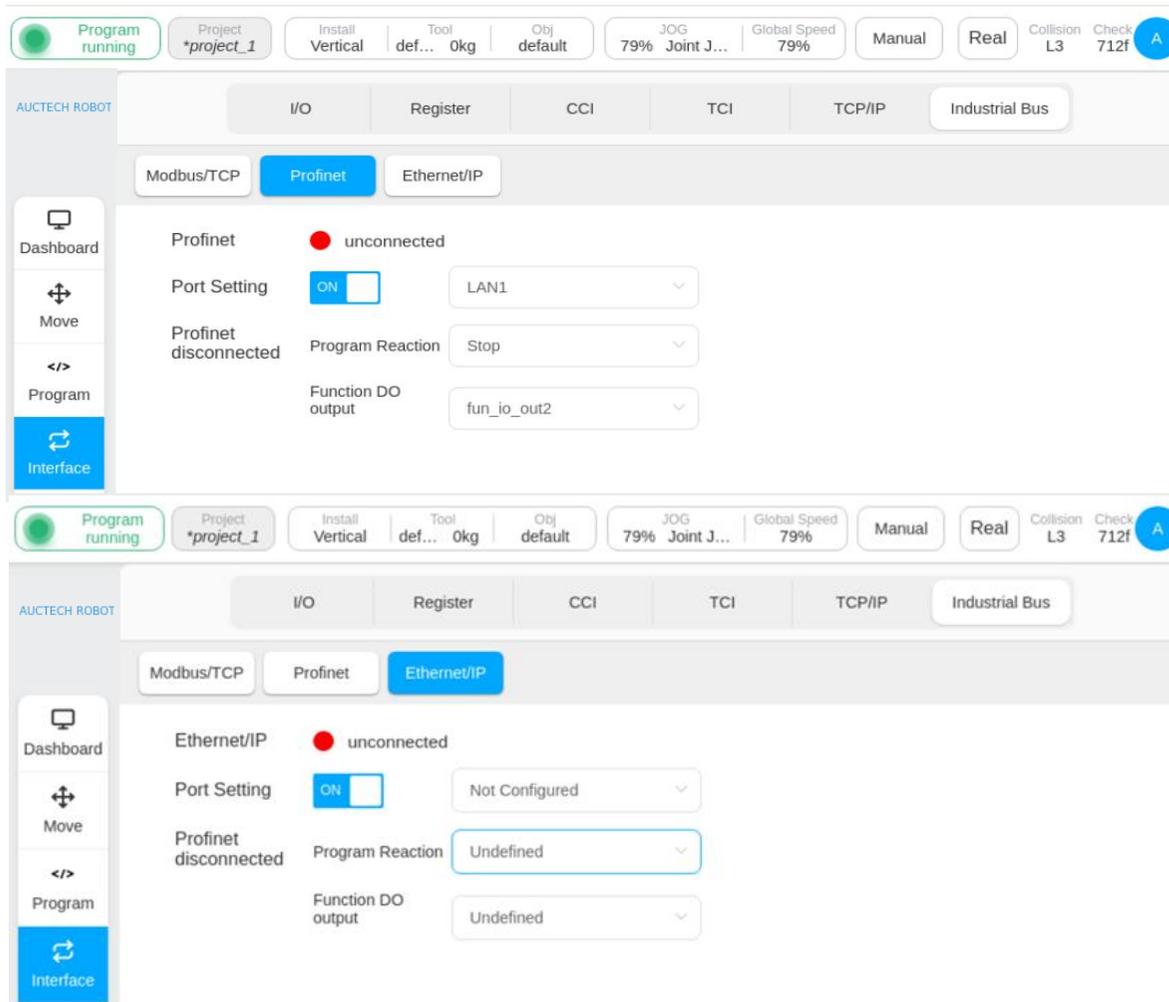
Input status

Coil status

Input Register

Holding Register

## 3.7 Profinet & Ethernet/IP



The image displays two screenshots of the AUCTECH robot control interface, showing the configuration settings for Profinet and Ethernet/IP. The interface includes a top status bar with 'Program running', 'Project \*project\_1', and various operational modes like 'Manual', 'Real', and 'Collision L3'. The left sidebar contains navigation options: 'Dashboard', 'Move', 'Program', and 'Interface'. The main content area shows the configuration for the selected communication protocol.

**Top Screenshot (Profinet):**

- Protocol: Profinet (unconnected)
- Port Setting: ON (LAN1)
- Profinet disconnected: Program Reaction (Stop)
- Function DO output: fun\_io\_out2

**Bottom Screenshot (Ethernet/IP):**

- Protocol: Ethernet/IP (unconnected)
- Port Setting: ON (Not Configured)
- Profinet disconnected: Program Reaction (Undefined)
- Function DO output: Undefined

1. Program Response: Undefined, Suspend Program, Stop Program

2. The function DO output is synchronized with the function output configuration in the I/O subpage

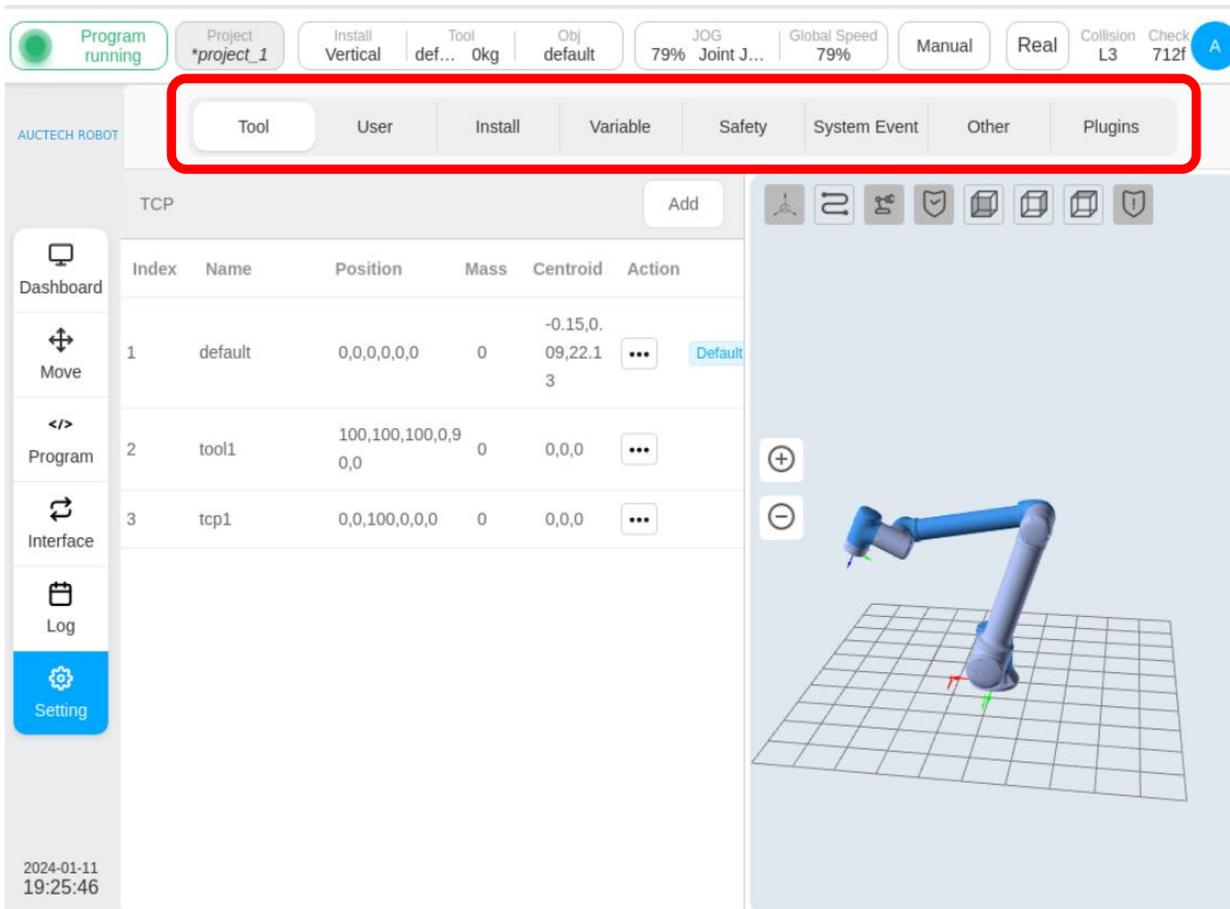


✦ Lecture 4 ✦

# Set up

---

## 4.1 Set up



Index	Name	Position	Mass	Centroid	Action
1	default	0,0,0,0,0,0	0	-0.15,0.09,22.13	...
2	tool1	100,100,100,0,90,0	0	0,0,0	...
3	tcp1	0,0,100,0,0,0	0	0,0,0	...

Tool settings, workpiece coordinate system, installation settings, variable settings, security settings, system settings, other settings, plug-in settings

# 4.1 Tool setting

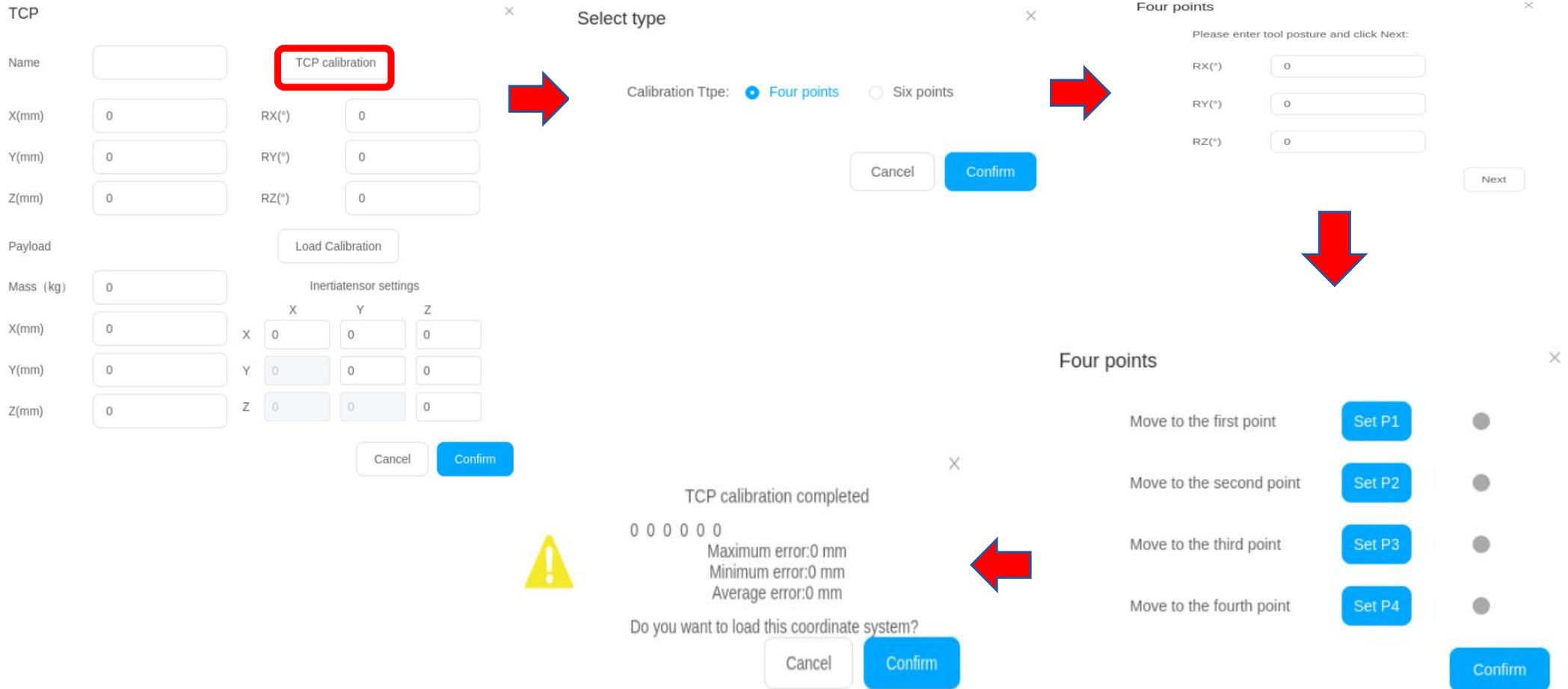
Index	Name	Position	Mass	Centroid	Action
1	default	0,0,0,0,0,0	0	-0.15,0.09,22.13	...
2	tool1	100,100,100,0,90,0	0	0,0,0	...
3	tcp1	0,0,100,0,0,0	0	0,0,0	...

Click the Add button to create

Dialog box for TCP calibration with the following fields:

- Name:
- X(mm):  0
- Y(mm):  0
- Z(mm):  0
- Payload:
- Mass (kg):  0
- RX(°):  0
- RY(°):  0
- RZ(°):  0
- Inertiatensor settings:
  - X:  0
  - Y:  0
  - Z:  0

# 4.1 Tool setting- 4-point calibration process

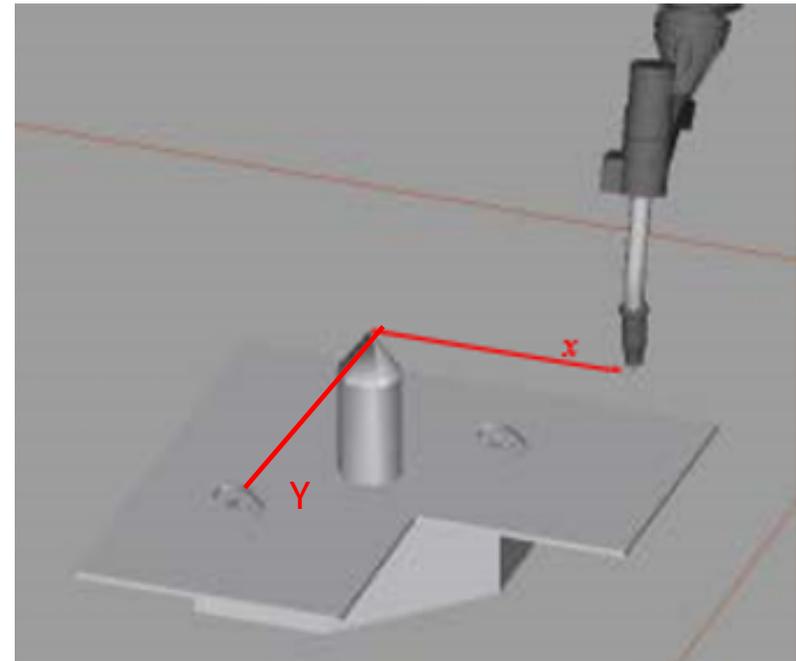


## ◆ 4.1 Tool setting- TCP calibration method

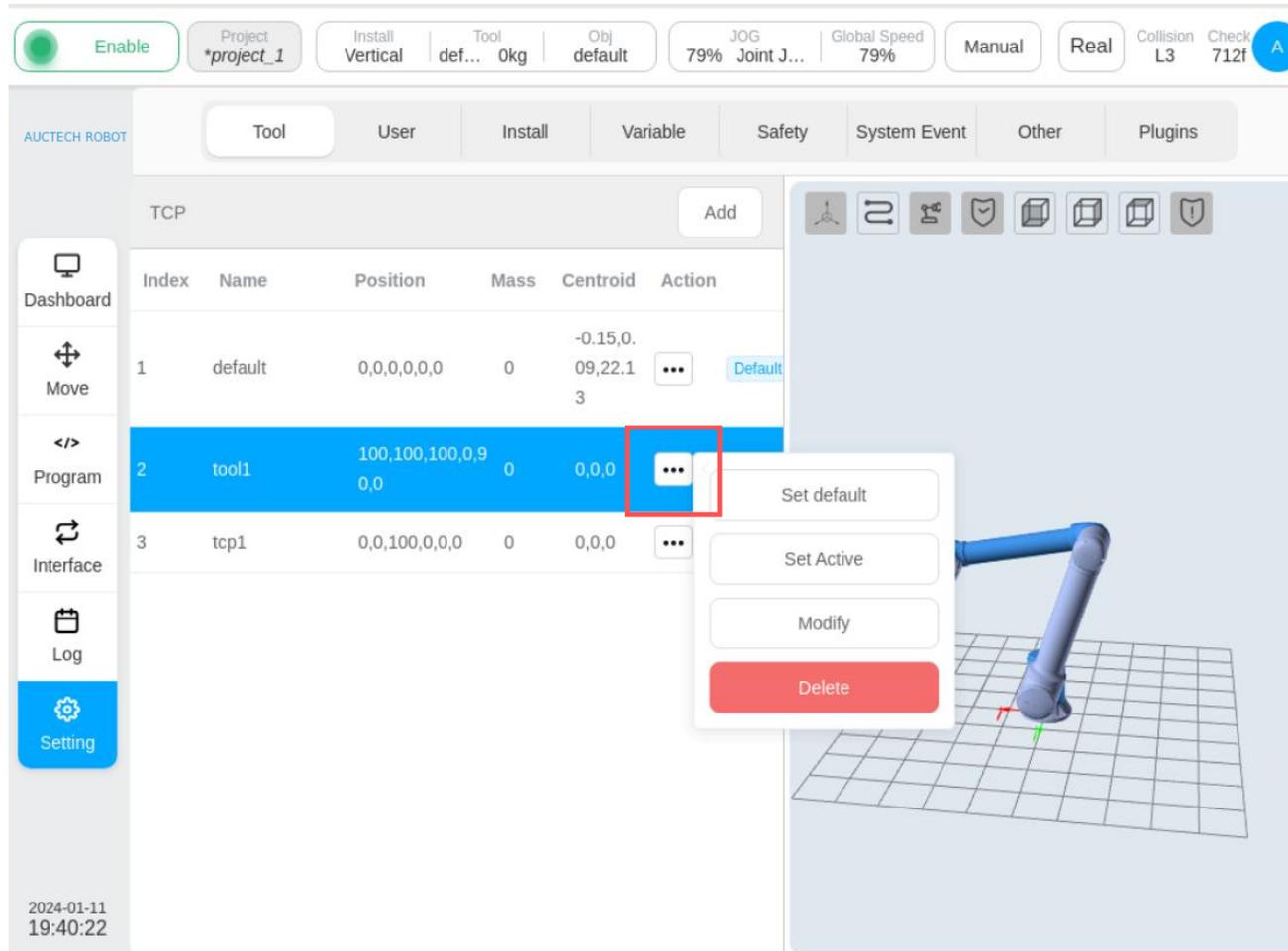
**4-point method:** take the calibration points  $P_1$ ,  $P_2$ ,  $P_3$ ,  $P_4$ , and the **difference between the 4 points is  $90^\circ$  and not in the same plane.**



**6-point method:** On the basis of the 4-point method, two calibration points in the  $X+$  and  $Y+$  directions are added.



## 4.1 Tool setting



Enable Project \*project\_1 Install Vertical Tool def... 0kg Obj default JOG 79% Joint J... Global Speed 79% Manual Real Collision L3 Check 712f A

AUCTECH ROBOT

Tool User Install Variable Safety System Event Other Plugins

TCP Add

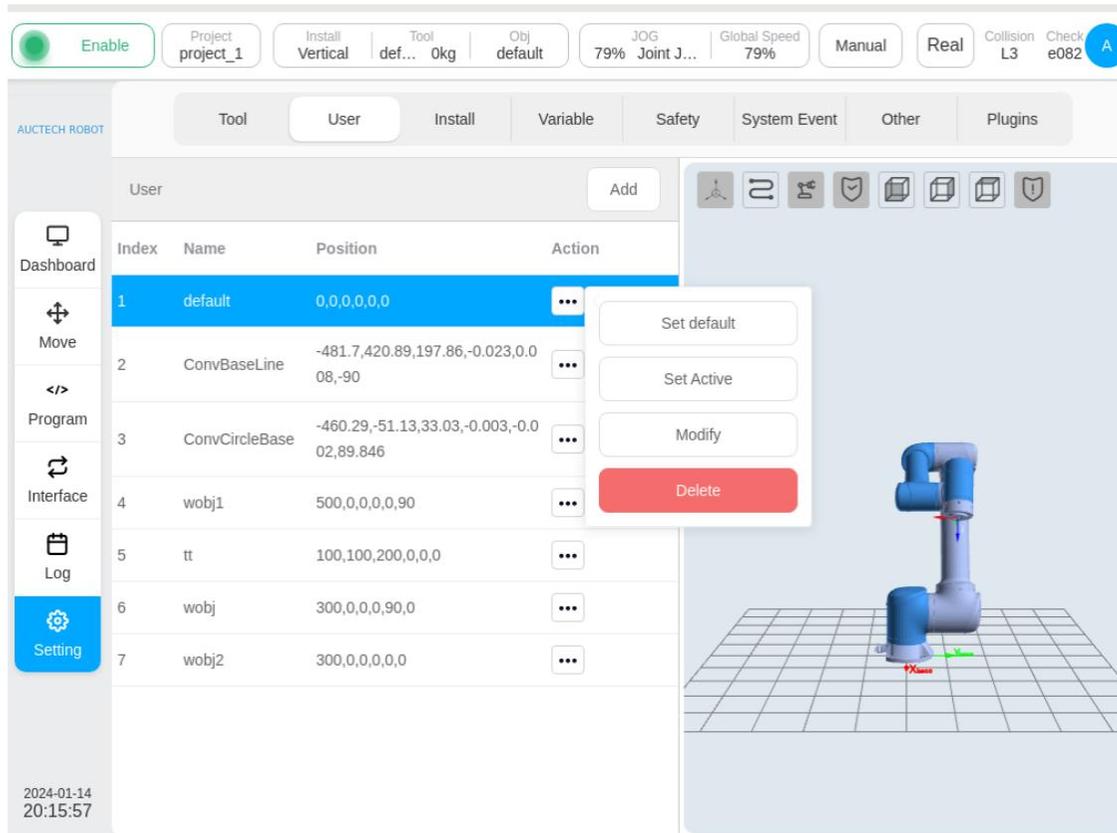
Index	Name	Position	Mass	Centroid	Action
1	default	0,0,0,0,0,0	0	-0.15,0.09,22.13	... Default
2	tool1	100,100,100,0,90,0	0	0,0,0	... <b>Delete</b>
3	tcp1	0,0,100,0,0,0	0	0,0,0	...

Dashboard Move Program Interface Log Setting

2024-01-11 19:40:22

Note: Tools that have been set as default or current cannot be deleted!

## 4.2 User setting

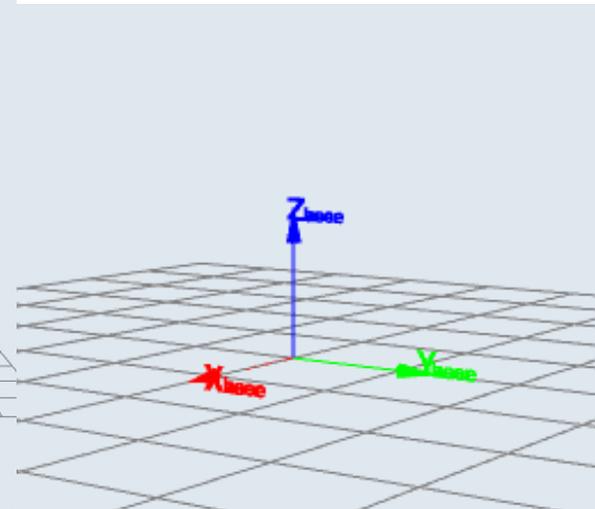


Index	Name	Position	Action
1	default	0,0,0,0,0,0	...
2	ConvBaseLine	-481.7,420.89,197.86,-0.023,0,0 08,-90	...
3	ConvCircleBase	-460.29,-51.13,33.03,-0.003,-0,0 02,89.846	...
4	wobj1	500,0,0,0,0,90	...
5	tt	100,100,200,0,0,0	...
6	wobj	300,0,0,0,90,0	...
7	wobj2	300,0,0,0,0,0	...

Red indicates the X-axis

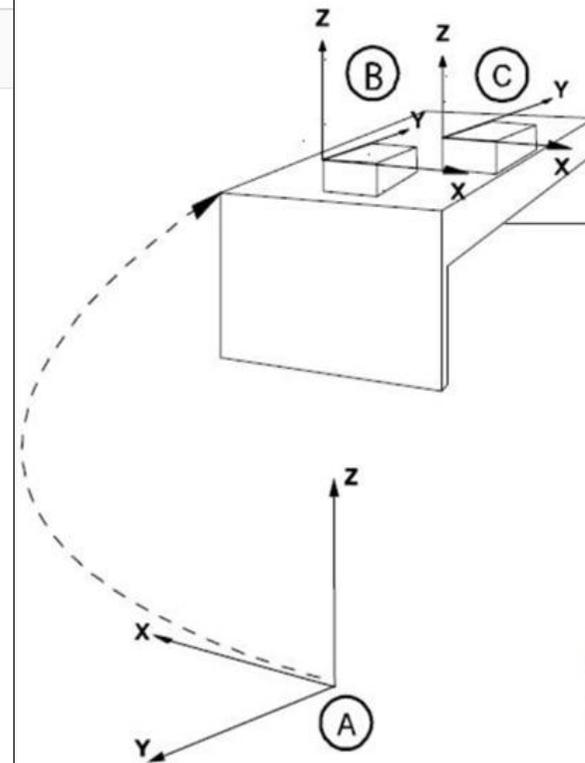
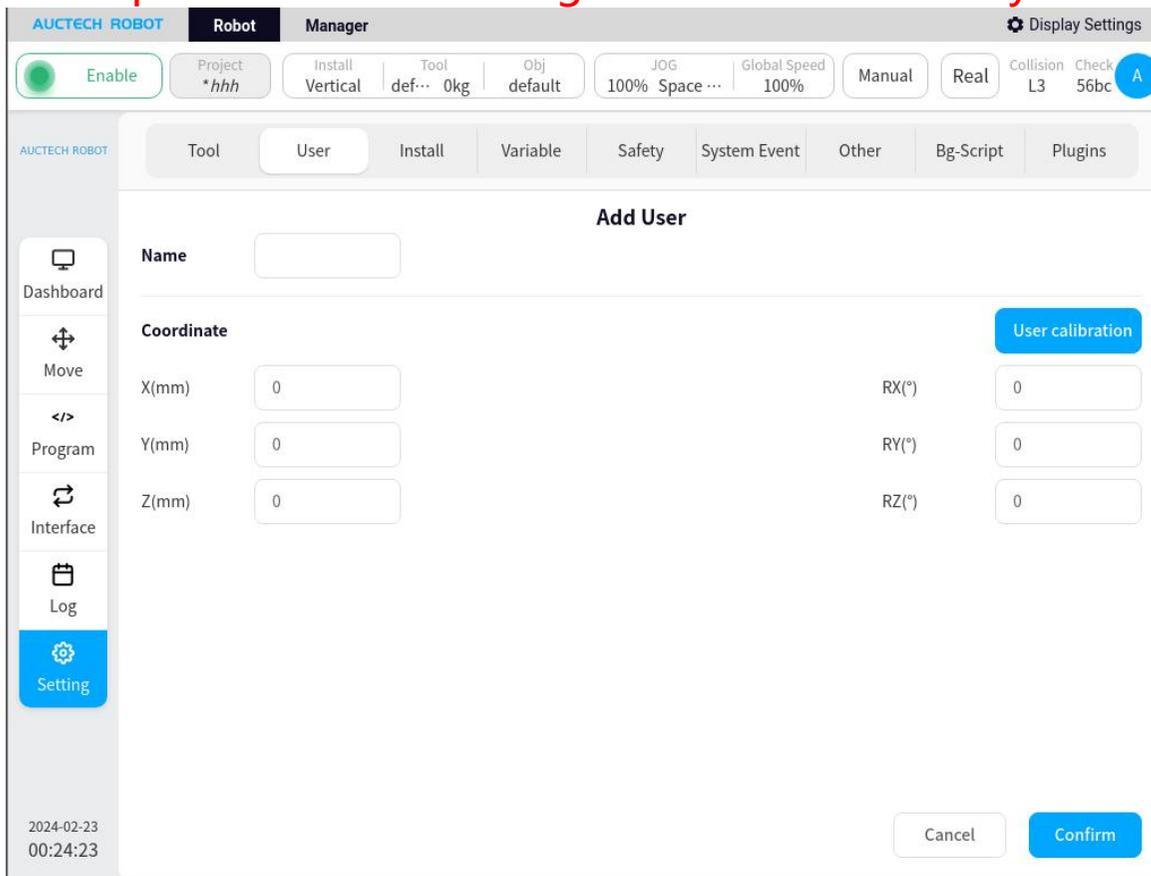
Green indicates the Y-axis

Blue indicates the Z-axis



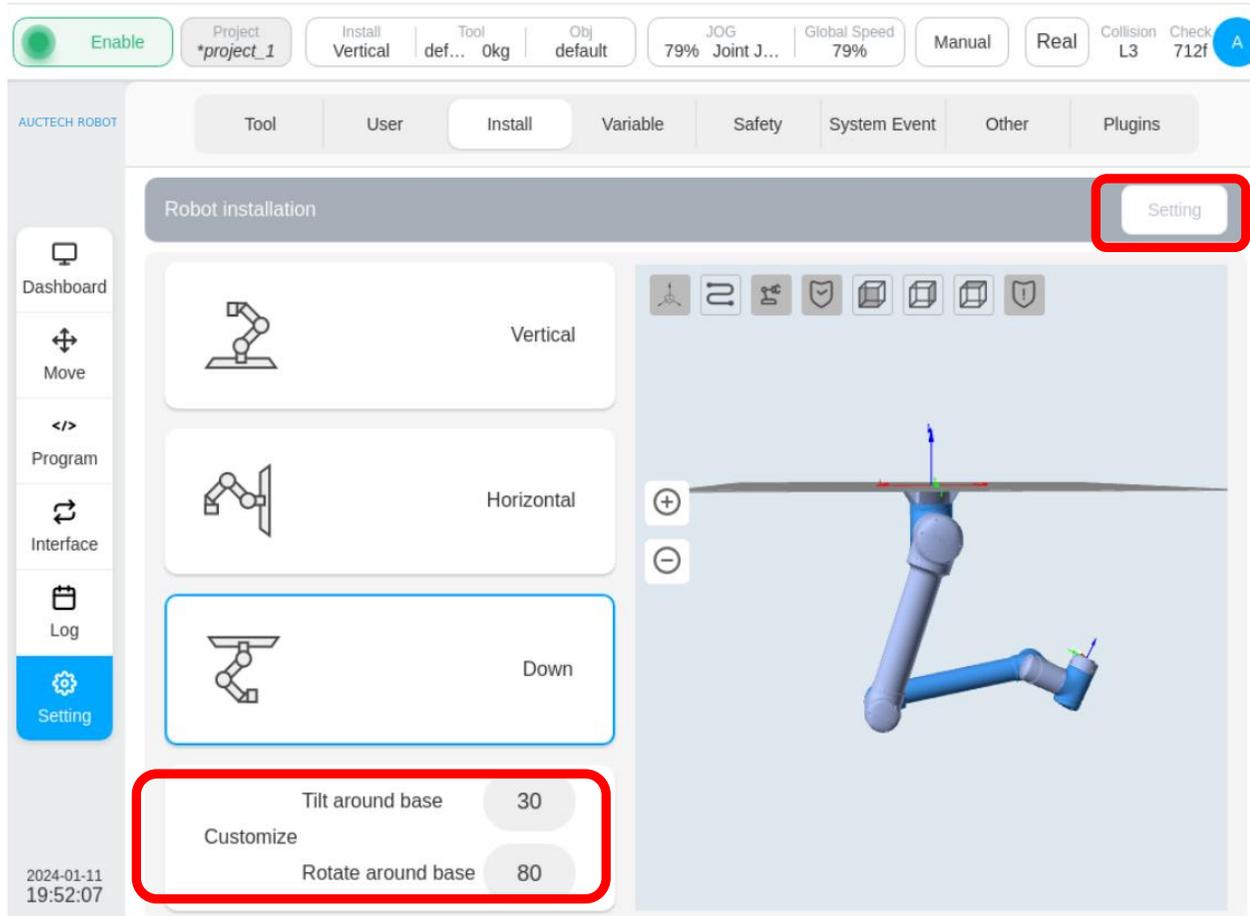
## 4.2 User setting

The workpiece coordinate system corresponds to the workpiece: **it defines the position of the workpiece relative to the geodetic coordinate system** (or other coordinate system).



A: 大地坐标系  
B: 工件坐标系1  
C: 工件坐标系2

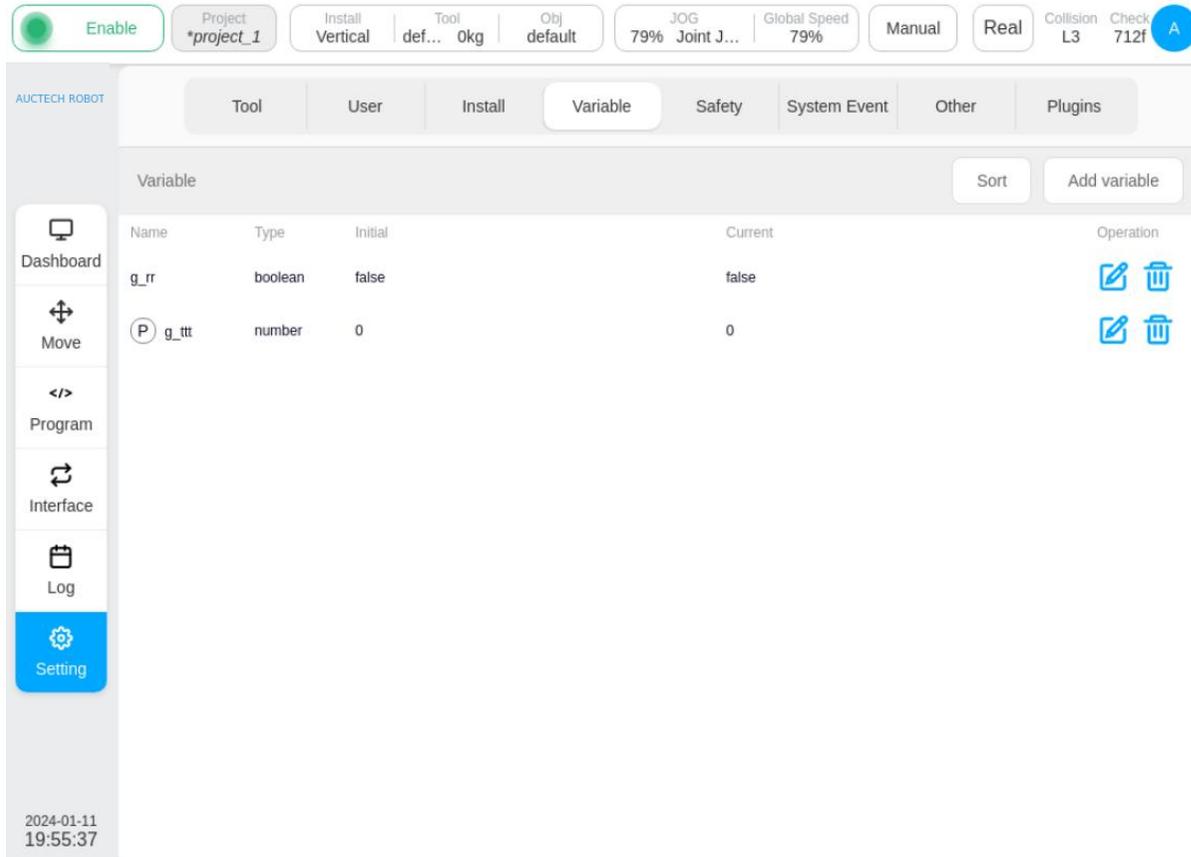
## 4.3 Install setting



The screenshot displays the AUCTECH ROBOT software interface. At the top, there is a status bar with 'Enable' and 'Project \*project\_1'. Below this, a navigation bar includes 'Tool', 'User', 'Install', 'Variable', 'Safety', 'System Event', 'Other', and 'Plugins'. The 'Install' section is active, showing 'Robot installation' with a 'Setting' button highlighted in red. The interface lists three installation modes: 'Vertical', 'Horizontal', and 'Down', with 'Down' selected. A 3D model of the robot arm is shown in the center. At the bottom, a settings panel is highlighted with a red box, showing 'Tilt around base' set to 30 and 'Rotate around base' set to 80. The date and time '2024-01-11 19:52:07' are visible in the bottom left corner.

The RY and RZ value  
setting range is  
 $-180^{\circ} \sim 180^{\circ}$

## 4.4 Variable settings



Name	Type	Initial	Current	Operation
g_rr	boolean	false	false	 
(P) g_ttt	number	0	0	 

There are **13 types of variables**, including boolean, number, string, num\_list, pose, joint, timer, pose\_speed, joint\_speed, pose\_acc, and joint\_acc, pose\_list and joint\_list

The word **P** is displayed in front of the name of the persistent variable

Note: The variables added in the settings are **system variables**, which can be used by each program in the current project.

## 4.5 Security settings



Enable Project \*project\_1 Install Vertical Tool def... 0kg Obj default JOG 79% Joint J... Global Speed 79% Manual Real Collision L3 Check 712f

AUCTECH ROBOT

Tool User Install Variable Safety System Event Other Plugins

Robot Limits

Robot Limits	Normal	Reduced	Tolerance
Tool Speed (m/s)	2.5	0.25	±0.01
Tool Force (N)	300	150	±10
Elbow Speed (m/s)	2.5	0.25	±0.01
Elbow Force (N)	300	150	±10
Power (W)	2750	1000	±100
Momentum (Kgm/s)	100	50	±10
Collision Level	Level3	Level3	

2024-01-11 19:57:45

Unlock

Safe mode:

- (1) **Normal mode**: the security mode activated by default;
- (2) **Reduction mode**: This mode can be activated with Secure Input IO

Security Precautions:

1. It must be set when the mechanical arm is powered off.
2. **When the project is switched, the parameters in the security settings will be invalid and need to be set anew.**

**Unlock password: 123**

## 4.6 System Event



The screenshot displays the AUCTECH robot control interface. At the top, there is a status bar with an 'Enable' button and various system parameters such as 'Project \*project\_1', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG 79%', 'Joint J...', 'Global Speed 79%', 'Manual', 'Real', 'Collision L3', and 'Check 712f'. Below this, a navigation bar includes 'Tool', 'User', 'Install', 'Variable', 'Safety', 'System Event', 'Other', and 'Plugins'. The 'System Event' configuration screen is active, showing a table with columns for 'Event' and 'Action'. A single event is listed: '1. Program Status = Running DO1 HIGH'. There are edit and delete icons for this event. The left sidebar contains navigation options: 'Dashboard', 'Move', 'Program', 'Interface', 'Log', and 'Setting'. The bottom left corner shows the date and time: '2024-01-11 20:02:53'.

**System Event Description:** When the event is true, the response will be triggered.

For example, when the program is running, the BOOL output register 1 is set to true.

**The current system events are not affected by the program.**

## 4.7 Other –Boot Setting



The screenshot displays the AUCTECH robot control interface. At the top, there is a status bar with an 'Enable' button, project information ('Project \*project\_1'), tool settings ('Install Vertical', 'Tool def... 0kg', 'Obj default'), speed settings ('JOG 79%', 'Joint J...', 'Global Speed 79%'), and mode buttons ('Manual', 'Real', 'Collision L3', 'Check 712f', 'A'). Below this is a navigation menu with tabs for 'Tool', 'User', 'Install', 'Variable', 'Safety', 'System Event', 'Other', and 'Plugins'. The 'Other' tab is selected, and the 'Boot Setting' sub-menu is active. The 'Boot Setting' panel includes: 'Home Setting' with a warning icon; 'Force sensor'; 'Tool Setting'; 'Import Model'; 'Interference Zone'; 'Default Global speed' set to 79% with a slider; 'Default jog speed' set to 79% with a slider; 'Auto power on' set to OFF with a toggle switch; and 'Set default program' with a 'Select program' button. A sidebar on the left contains icons for 'Dashboard', 'Move', 'Program', 'Interface', 'Log', and 'Setting' (which is highlighted). The bottom left corner shows the date and time: '2024-01-11 20:05:02'.

**Default Global/JOG Speed:** Sets the default speed when the computer is first turned on

**Automatic power-on of the robot:** After the control cabinet is turned on, the robotic arm body will automatically power on and enable the operation after the control cabinet is turned on.

**Set Default Program:** Set the default program.

## 4.7 Other – Home Setting

Tool	User	Install	Variable	Safety	System Event	Other	Plugins
Boot Setting	HOME position						
Home Setting 	Joint1 ( ° )	<input type="text" value="0"/>					
Force sensor	Joint2 ( ° )	<input type="text" value="0"/>					
Tool Setting	Joint3 ( ° )	<input type="text" value="30"/>					
Import Model	Joint4 ( ° )	<input type="text" value="0"/>					
Interference Zone	Joint5 ( ° )	<input type="text" value="0"/>					
	Joint6 ( ° )	<input type="text" value="0"/>					
<input type="button" value="Select position"/>		<input type="button" value="Confirm"/>					

Change or teach the Home point,  
and **after the change is completed,**  
it **needs to be synchronized in the**  
**safety Home.**

## 4.7 Other –Force sensor

Tool User Install Variable Safety System Event **Other** Plugins

Boot Setting  
Home Setting ⓘ  
**Force sensor**  
Tool Setting  
Import Model  
Interference Zone

**Force sensor**

**Base force sensor** ● unconnected

Input source: Box etherCat  Enable  OFF

Installation

X(mm)	<input type="text" value="0"/>	RX(°)	<input type="text" value="0"/>
Y(mm)	<input type="text" value="0"/>	RY(°)	<input type="text" value="0"/>
Z(mm)	<input type="text" value="0"/>	RZ(°)	<input type="text" value="0"/>

---

**Tool force sensor** ● unconnected

Input source: Box etherCat  Enable  OFF

Installation

X(mm)	<input type="text" value="0"/>	RX(°)	<input type="text" value="0"/>
Y(mm)	<input type="text" value="0"/>	RY(°)	<input type="text" value="0"/>

Settings for the force sensor: including settings such as importing topology files.

## 4.7 Other – Tool Setting



The screenshot shows the 'Tool Setting' interface with the 'Other' tab selected. The left sidebar contains a list of settings: Boot Setting, Home Setting (with a warning icon), Force sensor, Tool Setting (highlighted in blue), Import Model, and Interference Zone. The main content area is titled 'Tool Button' and contains the following settings:

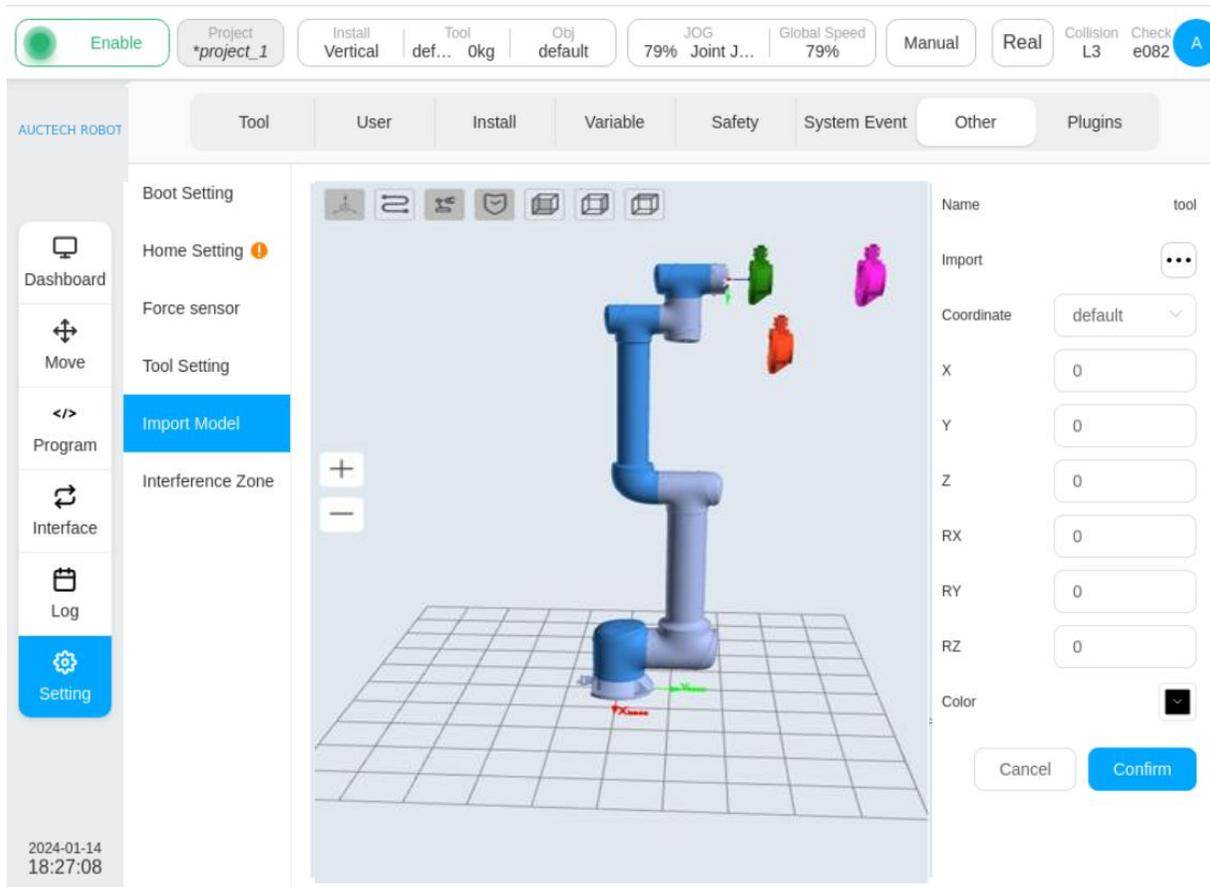
- Enable T Button: ON
- Enable S Button: ON
- S Button Function: Add variable & n
- Name Prefix: mydata

**Start the T button:** long press this button to drag and teach.

**Start the S button:** S button function: add variables; Add motion commands, as well as add variables and motion commands

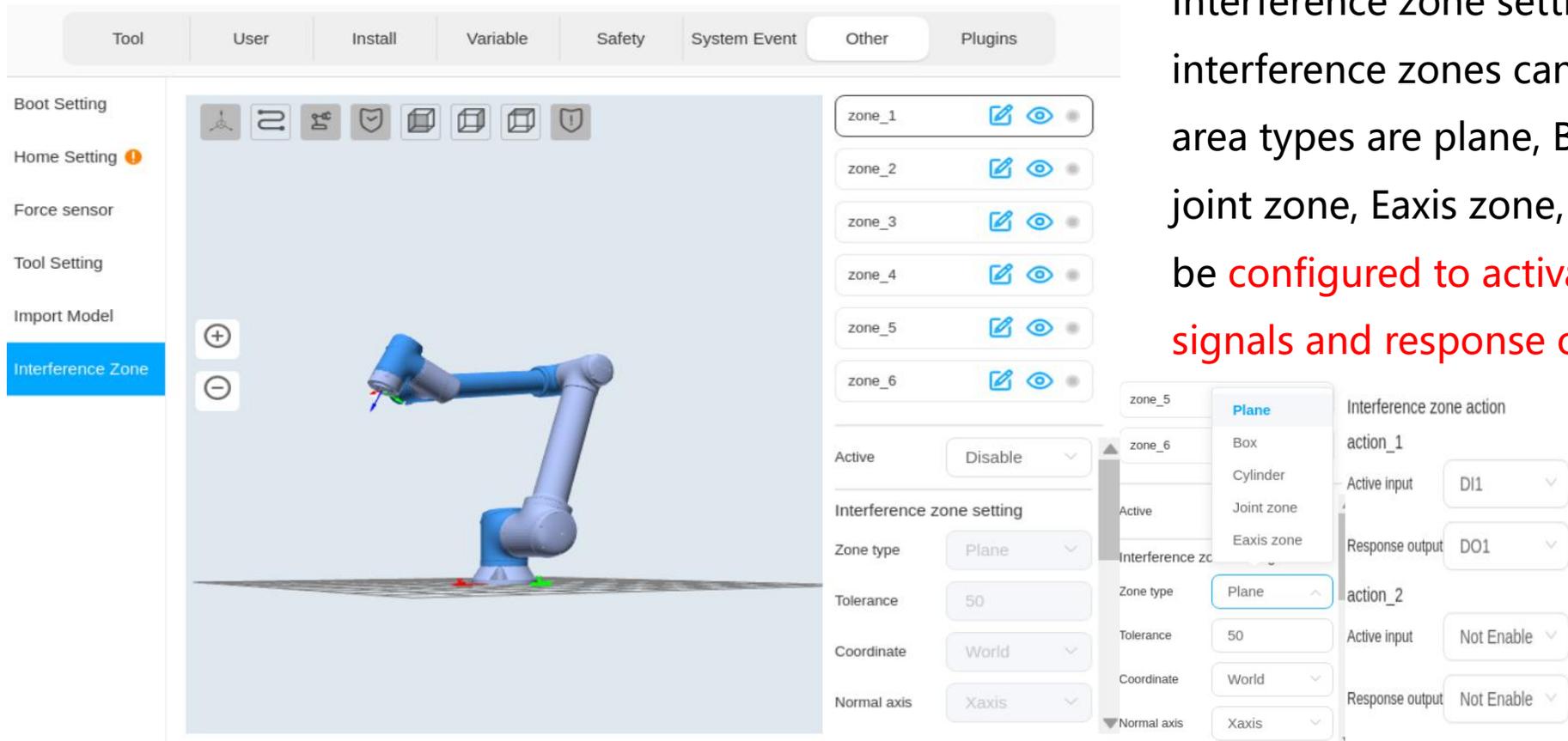
When **selecting the s button to add a variable or adding a variable and motion command**, you also **need to configure the variable name prefix.**

## 4.7 Other – Import Model



Model Import Settings: To import a model in a 3D image, the model must be a **TSL** file.

## 4.7 Other – Interference Zone



zone\_1 zone\_2 zone\_3 zone\_4 zone\_5 zone\_6

Active Disable

Interference zone setting

Zone type Plane

Tolerance 50

Coordinate World

Normal axis Xaxis

zone\_5 zone\_6

Plane

Box

Cylinder

Joint zone

Eaxis zone

Interference zone action

action\_1

Active input D11

Response output DO1

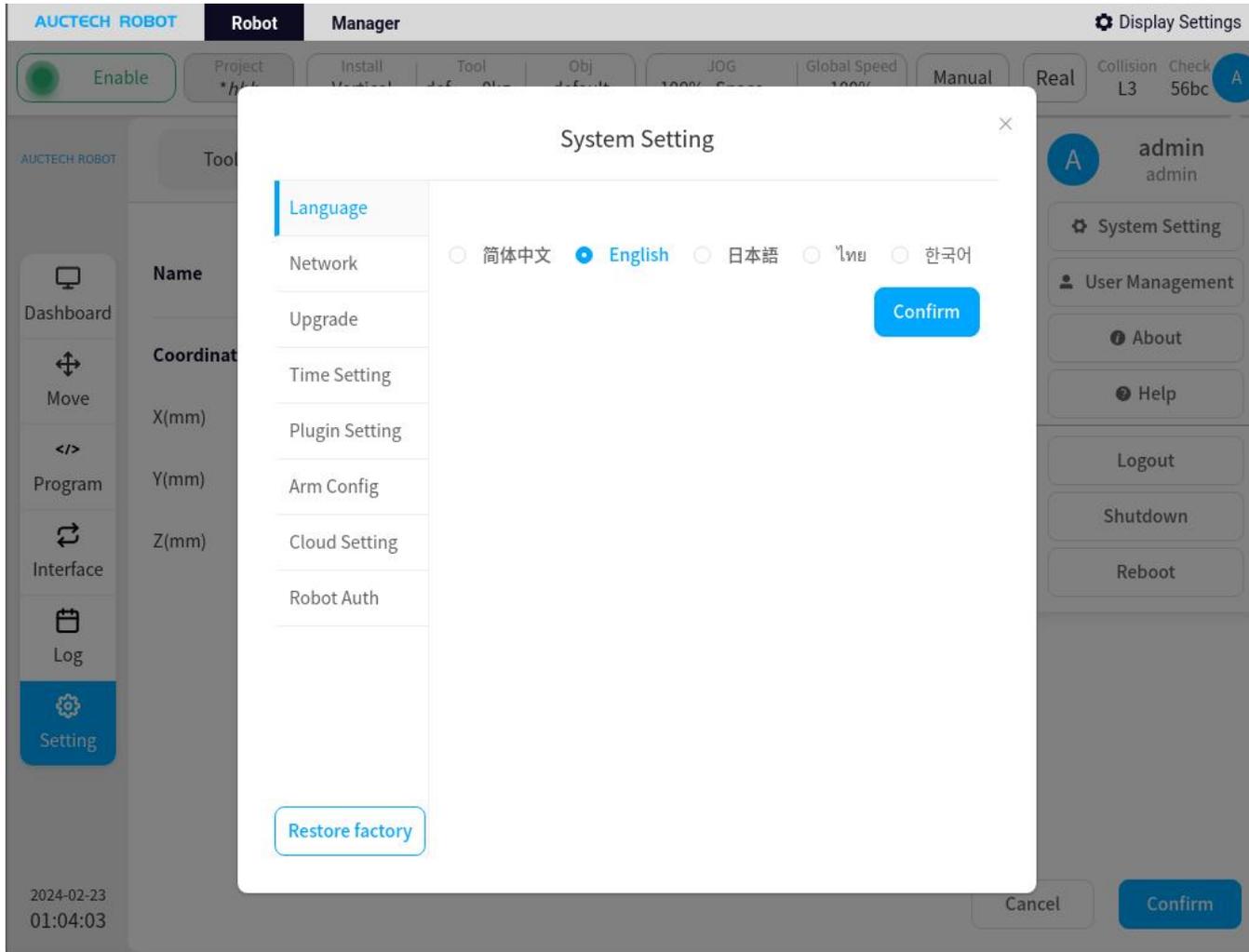
action\_2

Active input Not Enable

Response output Not Enable

Interference zone setting: 6 interference zones can be set, the area types are plane, Box, cylinder, joint zone, Eaxis zone, each area can be configured to activate by input signals and response output signals

## 4.8 System Settings - Language



Supported languages  
include: Chinese, English,  
Japanese, Thai

## ◆ 4.8 System Settings - Network

The network settings include LAN, WLAN settings.

### System Setting

Language

**Network**

Topology

Upgrade

Time Setting

Plugin Setting

Arm Config

Cloud Setting

Robot Auth

Restore factory

LAN1 | WLAN

#### Local IP

Detailed information

DHCP:

IP: 192.168.45.128

Mask: 255.255.255.0

#### IP Settings

DHCP  Static IP Settings

IP:

Mask:

Gateway:

Setting

### System Setting

Language

**Network**

Topology

Upgrade

Time Setting

Plugin Setting

Arm Config

Cloud Setting

Robot Auth

Restore factory

LAN1 | WLAN

#### WLAN Settings

Name:

Password:

Setting

## ◆ 4.8 System Settings - Topology

System Setting ×

Language

Network

**Topology**

Upgrade

Time Setting

Plugin Setting

Arm Config

Cloud Setting

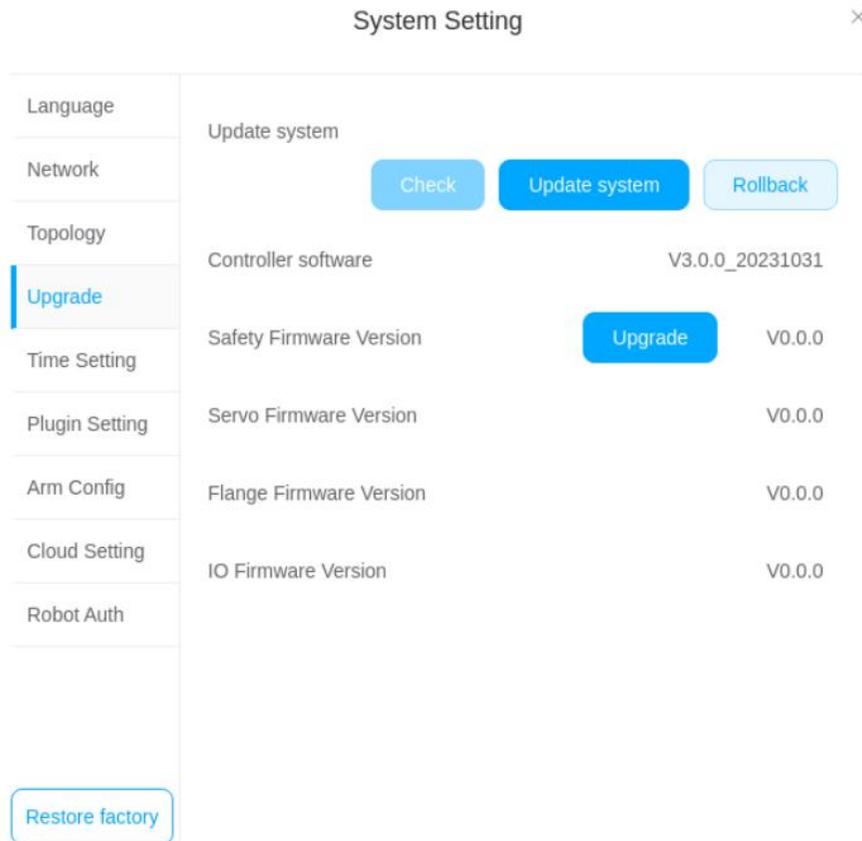
Robot Auth

Import topology

Select topology

Topology management is used to import topology files, **such as expanding external axes and 6D force control sensors.**

## ◆ 4.8 System Settings - Upgrade



Put the upgrade package in a USB flash drive in FAT32 format and insert it into the USB port of the control cabinet. Select Local Update. Find the file on the USB flash drive and OK.

Software version update or rollback files end with the **.update** suffix, select the required file and click the "Rollback" button will prompt the reboot to take effect (make sure the robot arm is in a power-off state)

Note: The control cabinet needs to be restarted after the update, and **the USB flash drive should not continue to be plugged into the USB port.**

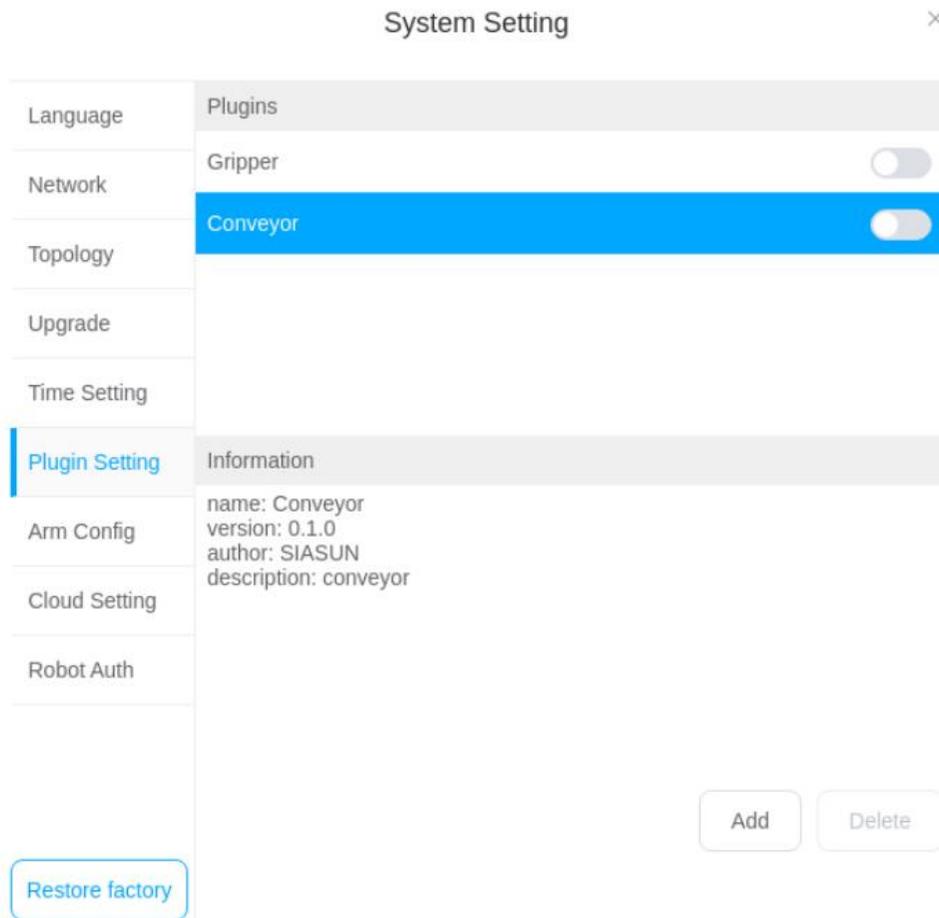
## ◆ 4.8 System Settings – Time Setting

System Setting ×

Language	
Network	Date <input type="text" value="2024-01-11"/>
Topology	
Upgrade	Time <input type="text" value="21:11:23"/>
<b>Time Setting</b>	<input type="button" value="Confirm"/>
Plugin Setting	
Arm Config	
Cloud Setting	
Robot Auth	

Time Setting: Set the current time (the log records are based on this time)

## ◆ 4.8 System Settings – Plugin Setting



System Setting

Language	Plugins
Network	Gripper <input type="checkbox"/>
Topology	Conveyor <input checked="" type="checkbox"/>
Upgrade	
Time Setting	
Plugin Setting	Information
Arm Config	name: Conveyor version: 0.1.0 author: SIASUN description: conveyor
Cloud Setting	
Robot Auth	

Restore factory

Add Delete

Plugin management: Manage the plugins used.

The supported plug-ins are visual, welding, palletizing, and external shafts.

## ◆ 4.8 System Settings – Arm Config



System Setting ×

Language

Network

Topology

Upgrade

Time Setting

Plugin Setting

**Arm Config**

Cloud Setting

Robot Auth

Export arm configs Export

Import arm configs

config Select

robot\_sn Select

Sync to the tool board

Import

Synchronize parameters in box to tool board

Restore factory

Robot parameters:

1. The configuration of the robotic arm can be imported and exported
2. Synchronize the parameters of the control cabinet to the terminal version

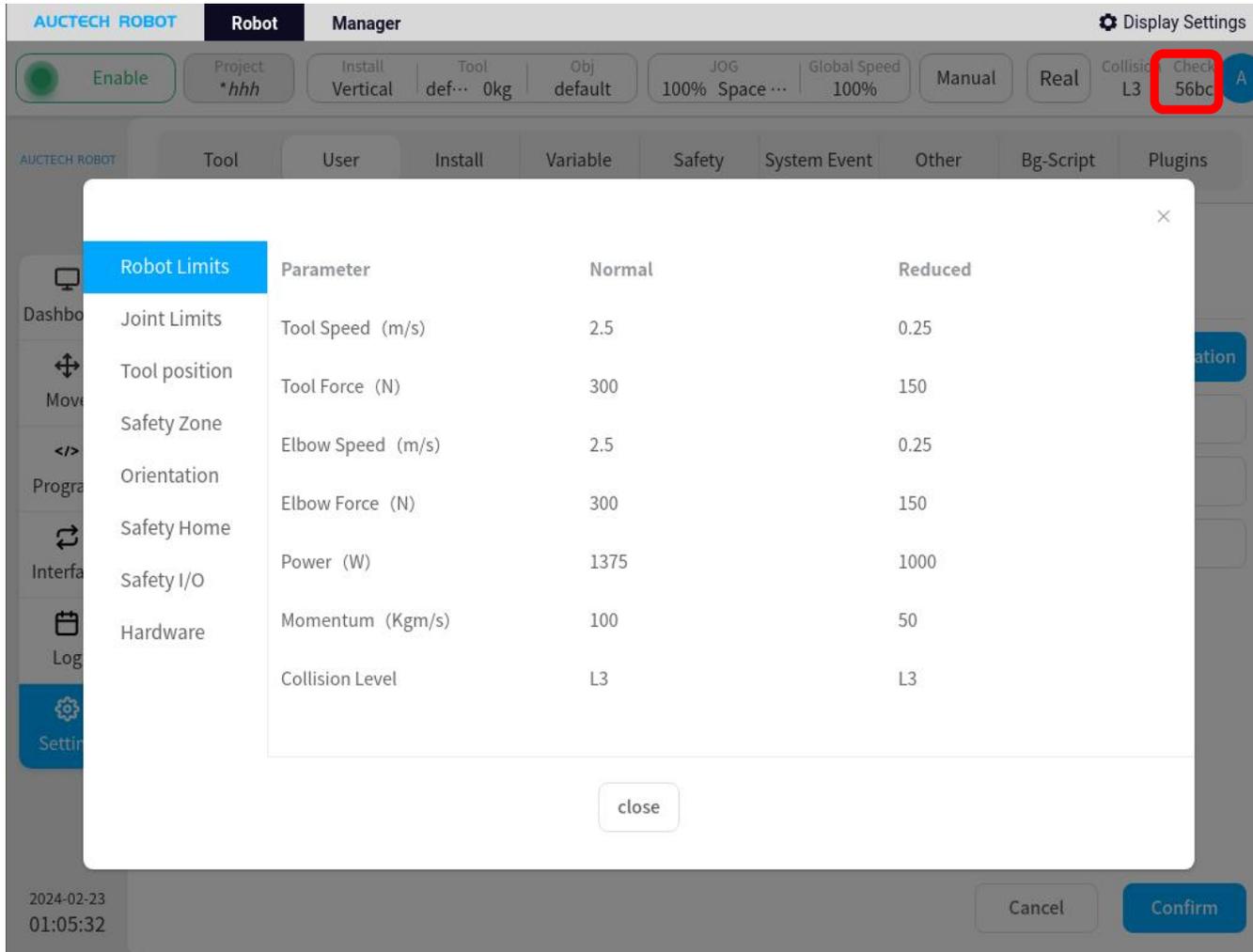


✦ Lecture 5 ✦

# Security settings

---

## 5.1 Modify the security configuration



The screenshot shows the AUCTECH ROBOT Manager interface. In the upper status bar, the 'Security Check' button is highlighted with a red box. A dialog box titled 'Robot Limits' is open, displaying a table of security parameters.

Robot Limits	Parameter	Normal	Reduced
Joint Limits	Tool Speed (m/s)	2.5	0.25
Tool position	Tool Force (N)	300	150
Safety Zone	Elbow Speed (m/s)	2.5	0.25
Orientation	Elbow Force (N)	300	150
Safety Home	Power (W)	1375	1000
Safety I/O	Momentum (Kgm/s)	100	50
Hardware	Collision Level	L3	L3

Click "Security Check" on the upper status bar, and the following dialog box will pop up, you can view the currently activated security configuration parameters.

Security configuration parameters can also be viewed in the settings page - security settings.

## 5.2 Modify the security configuration

**Normal Mode:** The security mode that is activated by default

**Reduced Mode:** This mode can be activated using the Safe Input IO

## 5.2 Modify the security configuration



The screenshot shows the AUCTECH robot control interface. At the top, there are various status indicators including 'Enable', 'Project \*hhh', 'Install Vertical', 'Tool def... 0kg', 'Obj default', 'JOG 100% Space ...', 'Global Speed 100%', 'Manual', 'Real', 'Collision L3', and 'Check 56bc'. Below this, there are tabs for 'Tool', 'User', 'Install', 'Variable', 'Safety', 'System Event', 'Other', 'Bg-Script', and 'Plugins'. The 'Robot Limits' tab is selected, and the 'Robot Limits' configuration page is displayed. A modal dialog box is shown in the center with the text 'Tip: Please power off first' and a 'Confirm' button. The 'Unlock' button at the bottom left is highlighted with a red box.

	Normal	Reduced	Tolerance
Tool Speed (m/s)	2.5	0.25	±0.01
Tool Force	150	150	±10
Elbow Speed	0.25	0.25	±0.01
Elbow Force	150	150	±10
Power	1000	1000	±100
Momentum (Kgm/s)	100	50	±10
Collision Level	Level3	Level3	

Before changing the safety configuration, you must use the password to unlock when the power of the robotic arm body is powered off, and the unlock

password: 123



## 5.3 Modify the security configuration –Robot Limits



Robot safety parameters are **used to limit general robot movements**. It is possible to configure its parameter values in normal and reduced modes.

	Normal	Reduced	Tolerance
Tool Speed (m/s)	2.5	0.25	±0.01
Tool Force (N)	300	150	±10
Elbow Speed (m/s)	2.5	0.25	±0.01
Elbow Force (N)	300	150	±10
Power (W)	2750	1000	±100
Momentum (Kgm/s)	100	50	±10
Collision Level	Level3	Level3	

**Maximum speed at the end:** Limit the maximum speed at the end of the robot.

**Maximum Force at the End:** Limit the maximum force exerted at the outermost end of the robot.

**Maximum elbow speed:** Limit the maximum elbow speed of the robot.

**Elbow Maximum Force:** Limits the maximum force exerted by the outermost part of the robot's elbow.

**Power:** Limit the maximum mechanical work that the robot can do to the outside.

**Maximum Momentum:** Limit the maximum momentum of the robot.

**Collision Level:** Set the level of the robot collision, the higher the level, the more sensitive.



## 5.4 Modify the security configuration - Joint Limits



Enable Project \*project\_1 Install Vertical Tool def... 0kg Obj default JOG 79% Joint J... Global Speed 79% Manual Real Collision L3 Check 712f A

AUCTECH ROBOT

Tool User Install Variable Safety System Event Other Plugins

Robot Limits

Joint Limits

Normal

Joint	min position (°)	max position (°)	Speed (°/s)	Torque (Nm)
Joint 1	-360	360	119.99	840
Joint 2	-360	360	119.99	840
Joint 3	-360	360	179.99	430
Joint 4	-360	360	224.99	205
Joint 5	-360	360	224.99	205
Joint 6	-360	360	224.99	205

Reduced

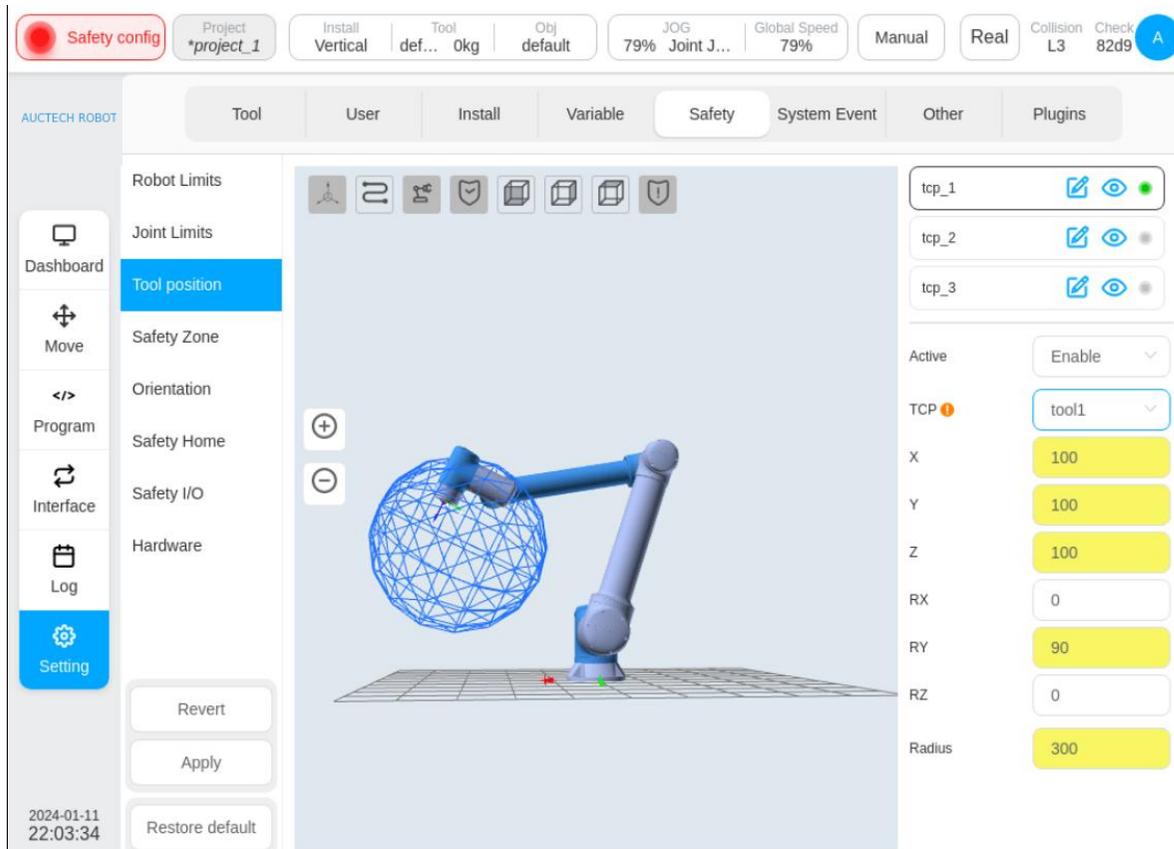
Joint	min position (°)	max position (°)	Speed (°/s)	Torque (Nm)
Joint 1	-360	360	29.99	400

Dashboard Move Program Interface Log Setting

2024-01-11 21:58:00 Unlock

Joint safety parameter limits are used to limit the position range, maximum speed, and maximum torque of each joint of the robot. It is possible to configure the values of the parameters in normal and reduced modes.

## 5.5 Modify the security configuration-Tool position

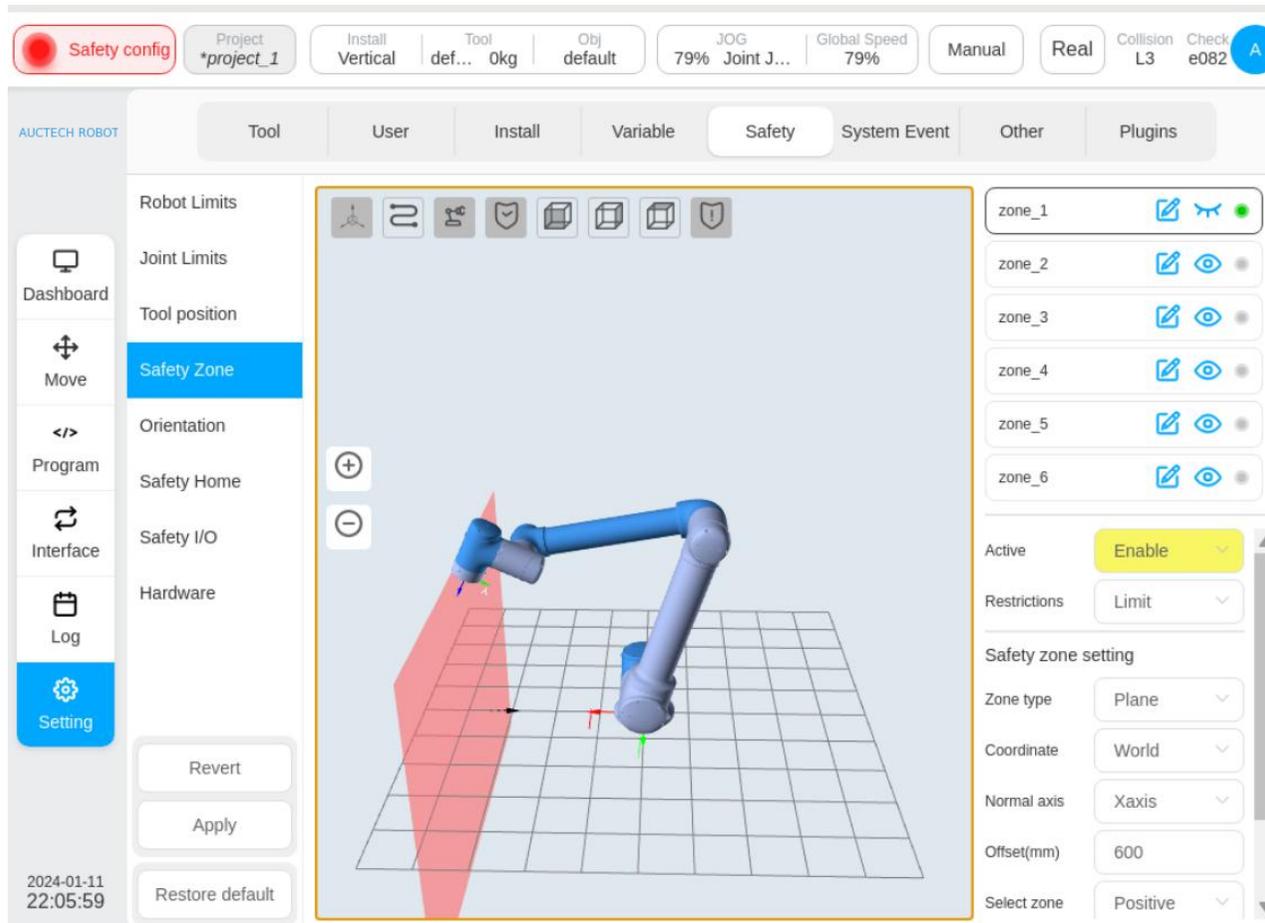


The security system can **define three sets of TCP offsets**, and after configuration, the robot will use these three sets of TCP to do speed and position monitoring. **Any TCP location where the speed exceeds the security setting will trigger a security violation.**

Note:

- (1) The inactive configuration is indicated in gray
- (2) This safety setting should be used in conjunction with the "safe area".

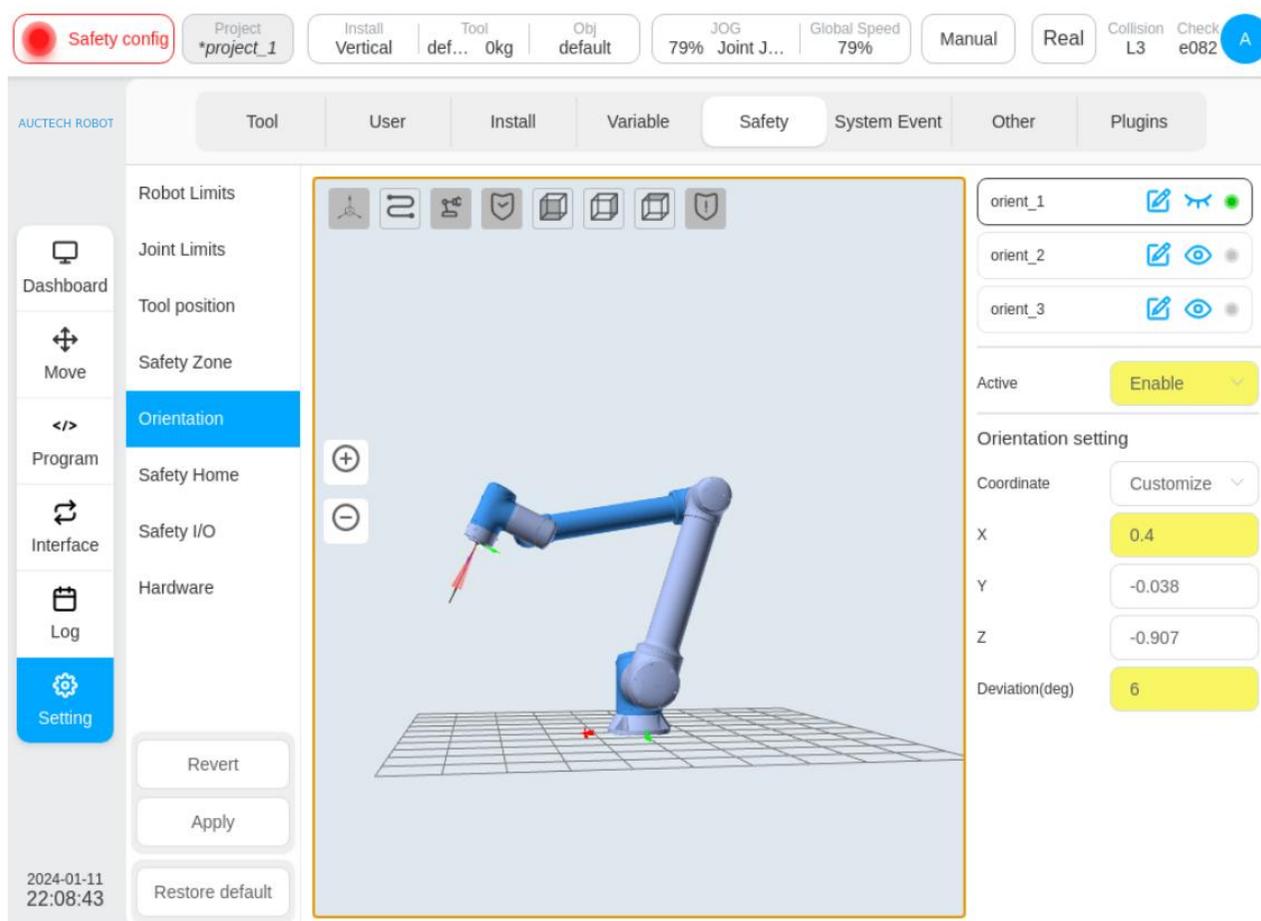
## 5.6 Modify the security configuration-Safety Zone



The definition of safety zone includes three forms: planar, rectangular and cylindrical. **Up to 6 independent space areas can be set up.**



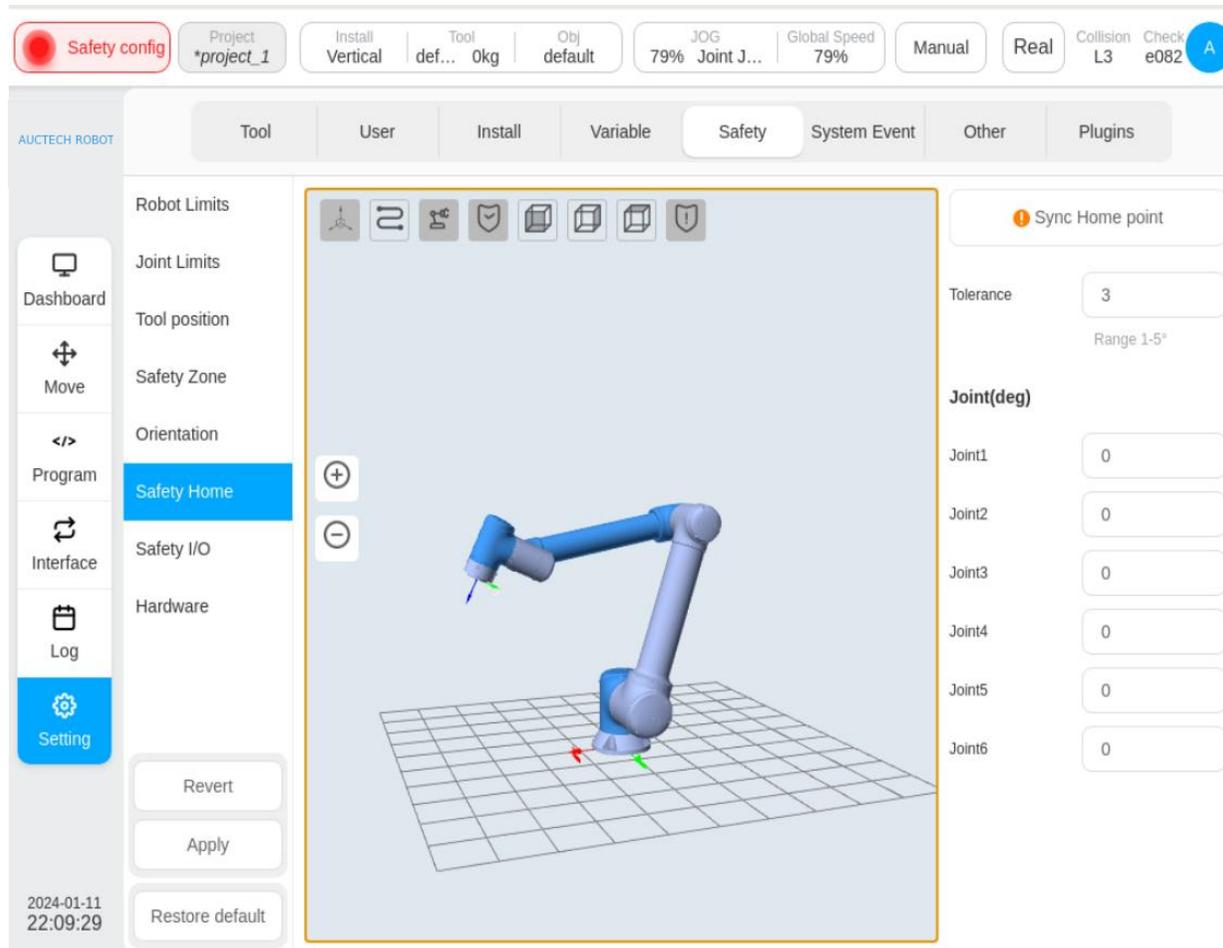
## 5.7 Modify the security configuration-Orientation



The safe attitude area is set up, and the attitude area refers to the formation of a conical angle around a certain direction vector under the robot base coordinate system. **The attitude restriction only restricts the Z-axis of the robot TCP to the range of the attitude region.** Triggering a protective stop beyond the area.



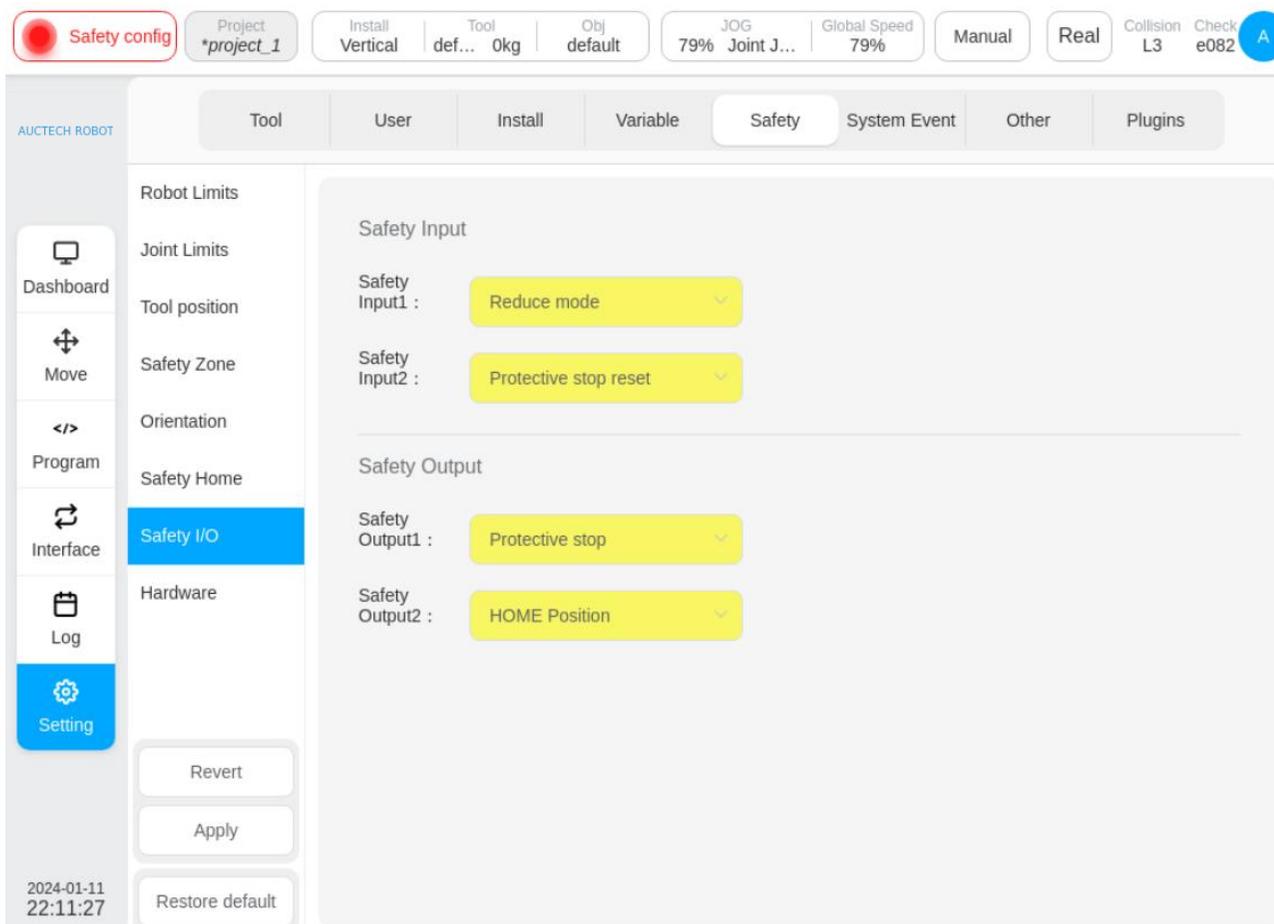
## 5.8 Modify the security configuration – Safety home



If the location in the security home settings is inconsistent with the home position in the "Other settings", a reminder icon will be displayed in the "Sync home settings".

If the settings are not synchronized, the home location in the system is subject to the security settings. The operation of "press and hold home" back to the home point in the mobile interface, and the output of the home signal of other ports of the robot (such as 2001) are subject to the safety settings.

## 5.9 Modify the security configuration - Safety I/O



The screenshot displays the AUCTECH robot control interface. At the top, there is a status bar with a red "Safety config" indicator, a project name "\*project\_1", and various operational parameters like "Install Vertical", "Tool def...", "Obj default", "JOG 79%", "Joint J...", "Global Speed 79%", "Manual", "Real", "Collision L3", and "Check e082". Below this is a navigation menu with tabs for "Tool", "User", "Install", "Variable", "Safety", "System Event", "Other", and "Plugins". The "Safety" tab is selected, showing a configuration screen for "Safety I/O". The screen is divided into two main sections: "Safety Input" and "Safety Output". Under "Safety Input", there are two dropdown menus: "Safety Input1" set to "Reduce mode" and "Safety Input2" set to "Protective stop reset". Under "Safety Output", there are two dropdown menus: "Safety Output1" set to "Protective stop" and "Safety Output2" set to "HOME Position". At the bottom of the configuration area, there are three buttons: "Revert", "Apply", and "Restore default". On the left side, there is a vertical sidebar with icons for "Dashboard", "Move", "Program", "Interface", "Log", and "Setting". The "Setting" icon is highlighted in blue. At the bottom left, the date and time "2024-01-11 22:11:27" are displayed.

There are two configurable inputs and outputs in the safety I/O.

## ◆ 5.10 Introduction to security features

Security features	description	Trigger an action	Security level
Joint position limitations	Set the upper and lower limits of the allowed joint position, which will be triggered when the current position of the robot exceeds the limit.	Safe stop 1	PLd
Joint speed limits	Set the allowable joint speed limit, which is triggered when the robot joint speed exceeds the limit.	Safe stop 1	PLd
Joint acceleration limitations	Set the allowable joint acceleration limit, which is triggered when the robot joint acceleration exceeds the limit.	Safe stop 1	PLd

## ◆ 5.11 Introduction to security features

Secure input	description	Trigger an action	Security level
Emergency stop	There are three emergency stop inputs: emergency stop button for <b>control cabinet</b> , emergency stop button for <b>teach pendant</b> , and <b>emergency stop interface</b> reserved for control cabinet. Triggered when any input source is disconnected.	Perform a Type 1 shutdown first, trigger a timer at the same time, and perform a Type 0 shutdown after 500ms.	PLd
Configurable safety input protective stops2	There are two configurable security inputs, <b>which need to be configured on the interface to take effect after being connected</b> . Takes effect only in automatic mode.	Safe stop 2	PLd
Configurable safety input protective stops1		Safe stop 1	PLd
Mode of operation	You can choose to enter manual mode or automatic mode. <b>The speed is 250 mm/s in manual mode</b> , and the safety function is triggered by switching between the operating modes while the robot is moving.	Safe stop 2	PLd
3 - position enabling device	When the robot is <b>in manual mode</b> , you need to <b>press the 3rd gear to the middle position to move the robot</b> . The safety function will be triggered when the 3rd gear is enabled at any time during the robot movement and is in a non-intermediate gear. <b>You can choose to turn this feature off</b> .	Safe stop 2	PLd

## ◆ 5.12 Introduction to security features

Secure output	description	Security level
Emergency feedback	When any E-stop is triggered, the path is disconnected.	PLd
Configurable safety outputs	There are <b>two configurable security outputs</b> , which need to be configured on the interface to take effect after being turned on. <b>Takes effect only automatic mode.</b>	PLd

## ◆ 5.13 Downtime category

Stop category	description
0	Immediately <b>cut off the power</b> of the robotic arm
1	Immediately <b>reduce the speed of each joint to 0</b> with the fastest acceleration, and after the joint is stationary, <b>the brake will be snapped up</b> , and the <b>motor will be enabled</b>
2	While maintaining the trajectory, <b>the robot is slowed down to a standstill</b> , and each joint remains enabled after stationation, and there is <b>no action in holding the brake</b> .



✦ Lecture 6 ✦

# Script Example

---

## ◆ 6.1 Example1

### Socket communication

```
ret= false
a="123"
b=""
socket_close("aa")
while(ret== false ) do
  ret=socket_open("192.168.233.1",50000,"aa",3000)
  sleep(200)
end
while(1) do
  socket_write(a,"aa",1000)
  sleep(1000)
End
```

## ◆ 6.2 Example2

Posture millimeters, degree units converted into meters, radians

```
a={100,200,300,90,0,45}
b={0,0,0,0,0,0}
function fun(data)
  local a={0,0,0,0,0,0}
  for i=1,3 do
    a[i]=data[i]/1000
  end
  for i=4,6 do
    a[i]=deg2rad(data[i])
  end
  return a
end
b=fun(a)
log( "b" ,b)
```



## 6.3 Example3

Pose meters, radians converted to millimeters, degrees

```
a={0.100,0.200,0.300,1.57,0,0.785}
```

```
b={0,0,0,0,0,0}
```

```
function fun(data)
```

```
  local a={0,0,0,0,0,0}
```

```
  for i=1,3 do
```

```
    a[i]=data[i]*1000
```

```
  end
```

```
  for i=4,6 do
```

```
    a[i]=rad2deg(data[i])
```

```
  end
```

```
  return a
```

```
end
```

```
b=fun(a)
```

```
log("b",b)
```

## ◆ 6.4 Example4

Empty list expressions

```
a=1
b=nil ----- equated b{}
if next(b) == nil then
  a=a+1
end
log("a",a)
```



## 6.5 Example5

# Palletizing and depalletizing procedures



```
--movej({0,0,1.57079601,0,-1.57079601,0},90,90)
a={4,3,2}           -- Layers, rows, columns
b={0.1,0.1,0.1}    --Workpiece dimensions, length, width and height (in mm)
c={0,0,0}          --Record Layer Rows and Columns
while(c[1]<a[1]) do  -- Judgment layer
start_dian={0.69200003,0.16400003,0.6600002,-3.14159226,3.2737059996179596e-7,-1.57079637}
    if(c[2]<a[2]) then -- Judgment line
        if(c[3]<a[3]) then
movej({1.57079601,0,1.57079601,6.442085975777445e-9,-1.57079601,3.947477900112517e-9},90,90)
start_dian[1]=start_dian[1]-b[1]*c[3] --column
start_dian[2]=start_dian[2]-b[2]*c[2] -- row
start_dian[3]=start_dian[3]-b[3]*c[1]
movej(start_dian,1,1)
    c[3]=c[3]+1
else
    c[3]=0
    c[2]=c[2]+1
End
else
    c[2]=0
    c[1]=c[1]+1
end
End
```

## ◆ 6.6 Example6

String data is extracted based on the head and end of frames

```
data="152,,295,26262,aaa,100,200,300,400,500,600,bbb,152,295,26262,aaa"  
tou="aaa,"  
wei=",bbb,"  
qqq=""  
a=0  
b=0  
a=str_find(data,tou)+4  
b=str_find(data,wei)  
qqq=str_substr(data,a,b-a)  
log("qqq",qqq)
```

## ◆ 6.7 Example7

Add parentheses to the string and turn it into an array

```
data="100,200,300,400,500,600"
```

```
cc={0,0,0,0,0,0}
```

```
data=str_cat("(",data)
```

```
data=str_cat(data,")")
```

```
cc=str2list(data)
```

```
log("cc",cc)
```

**Thank you for using Auctech Cobots**



**THANKS**

