User Manual - G602 Robotic Arm

Revision 3.2.2

2021-02-10

For G602 robots

(Hardware version number V3.0.0)

RobotAnno (Shenzhen) Co., Ltd

www.robotanno .com

Statement

Thank you for purchasing and using PROBOT G602 products. For your safety and benefit, please read the user manual and all the random information before using the product. If you do not operate and use the product in accordance with the user's use manual, resulting in any personal injury, property or other losses, RobotAnno (Shenzhen) Co., Ltd will not be liable.

Copyright Description

RobotAnno (Shenzhen) Co., Ltd

Address: Building 705, Building 1, Nanchang No.1 Industrial Zone, Xixiang Street, Baoan District, Shenzhen City, Guangdong Province

Sales Mail: robotanno@robotanno.com

After sale Mail: support@robotanno.com

Web: www.robotanno.com

强制
● G602 use of the robot arm is fully described in this specification. Please operate the robot arm on the basis of careful reading and full understanding. ● Before the arm is powered on, adjust the arm to near the initial state, and the operator is in a safe position outside the working range of the arm. ● In case of emergency, press the emergency stop button immediately. If the brake arm is not in time, it may cause personal injury or equipment damage.

注意
● The drawings and photos in the manual are representative examples and may be different from the purchased products. ● The instructions are sometimes modified due to product improvements, specifications changes and their easier use. The revised instructions will update the version number below the cover and be issued in a revised version. ● When you need to order the instructions due to breakage and loss, please contact our sales department to order according to the version number of the cover. ● The customer carries on the product transformation without authorization, is not within the scope of our warranty, our company is not responsible.

Directory

Directory.. 4
Document version.. 6
1. Security considerations.. 7
1.1 Symbols and their meanings.. 7
1.2 Hazard issues.. 7
1.3 Notes.. 9
1.4 Environment.. 10
2. System description.. 11
2.1 Product Overview.. 11
2.2 Description of control box.. 15
2.3 IO Interface Description.. 17
2.4 Hardware parameters.. 19
2.5 Structural dimensions.. 19
3 Quick Start Guide.. 22
3.1 Hardware interface and connection.. 22
3.2 Start ROS 2GO.. 23
3.3 Power on system.. 25
3.4 Setting up network connections.. 25
3.5 Launch ROS upper computer.. 27
3.6 Simulation/online switching.. 28
3.7 Arm enabling.. 28
3.8 Adjust the initial position of the robot arm.. 29
3.9 Dynamic point control of robot arm.. 29
Use of 4 mechanical arms.. 30
4.1 Basic operations.. 30
4.2 Host computer ROS environment.. 31
4.3 Introduction of Upper Computer Function.. 37
4.4 Single-machine simulation.. 49
4.5 Operation of the real machine.. 50
4.6 Claws used.. 50
4.7 Routine demonstration.. 52
5MoveIt! Programming interface.. 53
5.1 Setting the state of motion.. 53

5.2 Set Target Position........ 57
5.3 Cartesian space motion........ 62
6 One-click firmware upgrade........ 66
6.1 Directions for Use........ 66
6.2 Preparation for upgrade........ 66
6.3 One-click upgrade........ 68
7 Blockly graphical programming........ 70
7.1 Enable graphical programming........ 70
7.2 Introduction to the Programming Interface........ 71
7.3 Blockly Code block........ 74
7.4 Blockly Programming examples........ 77
7.5 Frequently Asked Questions........ 79
8 Common Fault Diagnosis........ 81
8.1 Unable to boot........ 81
8.2 Arm unable to return to initial position........ 81
8.3 Mechanical arms hold tight........ 81
9 Appendix........ 83
9.1 References........ 83
9.2 Contact information........ 83

Document version

Date	Version	Author	Summary
2019-4-13	1.0	RobotAnno	Initial version
2019-10-8	2.0	RobotAnno	Add teach point save and run function
2019-12-15	2.1	RobotAnno	Add Zero-bit Calibration Function
2020-2-23	2.2.0	RobotAnno	New one-click firmware update function; update zero-bit calibration function
2020-3-11	2.3.0	RobotAnno	Chapter 7 Blockly Graphical Programming
2020-05-05	2.3.1	RobotAnno	Zero calibration function upgrade, update related parameter description
2020-7-16	3.0.0	RobotAnno	Add V 3.0.0 update
2020-12-24	3.2.1	RobotAnno	Corresponding software and hardware system version V3.2.1
2021-1-16	3.2.2	RobotAnno	Corresponding software and hardware system version v3.2.2

Table 1： version history

1 Safety precautions

Thank you for purchasing PROBOT G602 robotic arm. For your safety and prevent damage to the robotic arm, please read and master this manual and other ancillary information before using the robot arm, and start using it after you are familiar with all equipment knowledge, safety knowledge and precautions, and pay special attention to the following safety signs.

1.1 Symbols and their meanings

误操作时有危险，可能发生死亡或重伤事故

误操作时有危险，可能发生中等程度伤害、轻伤事故或物件损坏

手册和文档中必须遵守的事项

手册和文档中明确禁止的事项

Even matters belonging to the category of "attention" will have serious consequences because of different circumstances, so any "attention" matters are extremely important, please strictly abide by.

1.2 Hazardous matters

(1) In case of emergency, press the emergency stop button immediately, if the arm can not brake be timely, it may cause personal injury or equipment damage accident.

Stop button

(2) When power supply is switched on after the emergency stop is lifted, the emergency stop button should be started after the accident causing the emergency stop.

Release from emergency

(3) When moving within the range of the robot arm, observe the following:

1) Consider the strain scheme when the robot arm suddenly moves to its position.
2) Make sure to set up a shelter, just in case.

⚠ 注意

The robot arm action caused by misoperation may cause personal injury accident.

(4) Make sure that no one is within the range of the action of the robot arm and that the operator is operating in a safe position when:

1)G602 the arm is powered on;
2) trial run time;
3) show again.

(5) Please do not move and maintain the arm while the arm is in operation. If you want to move and maintain, please turn off the power of the arm before doing this operation.

⚠ 注意

Accidental entry into the robot arm action range or contact with the robot arm may cause personal injury. If you find an exception, press the stop button immediately.

Emergency stop button is located on the right side of the front of the PROBOT G602 robot arm control box.

1.3 Notes

（1）PROBOT G602 arm owners, operators must be responsible for their own safety. Do not rebuild the mechanical arm, due to unauthorized product transformation caused by accidents or failures, not within the scope of our warranty, the company is not responsible.

（2）Based on understanding the "warning sign" of the instructions for the use of the PROBOT G602 robot arm, use the robot arm.

（3）Check the following items in front of the PROBOT G602 arm, repair or take other necessary measures in time if abnormal.

1) whether each axis of the robot arm is in zero position;

2) electrical cables are properly connected;

3) electrical cable skin damage;

4) the emergency stop switch is in the state of release;

5) arm movement is abnormal, abnormal noise.

(4) Do not approach when controlling the movement of the robot arm and:

1) in the operation of the robot arm to use a lower rate of speed to debug the movement of the robot arm, in order to increase the effective control of the robot arm;

Before pressing the power supply key, the 2) should take into account the motion trend of the robot arm, consider in advance the motion track of the robot arm, and confirm that the line is not interfered;

(5) Production and operation

Before 1) boot, you must know all the tasks the robot arm will perform according to the program.

2) shall know the position and status of all switches, sensors and control signals that will affect the movement of the robot arm.

3) shall know the position of the emergency stop buttons on the robot arm control equipment, ready to press these buttons in case of emergency.

4) never think that the robot arm has been completed without moving its program. Because the arm is probably waiting for the input signal to move on.

1.4 Environment

PROBOT G602 the arm is safe to use and can adapt to the environment to the maximum extent. Please follow the instructions. Please be sure to follow the precautions in this manual.

（1）Never force the shaft of the arm, otherwise it may cause personal injury and equipment damage.

（2）After using the mechanical arm, the power cord plug should be unplugged, and the mechanical arm should be placed in a dry, normal temperature.

（3）Avoid being in dirt and dusty environment, avoid long-term placement in direct sunlight position, away from high humidity and strong vibration environment, soil, waste debris, high temperature will damage internal devices.

（4）Do not use PROBOT G602 arm on the following occasions:

1) close to flammable substances

2) an explosion

3) water or other liquid

4) the presence of corrosive, flammable gases

Environment 5) temperatures above 40 degrees Celsius

6) other harsh environment

2 System description

2.1 Product Overview

RobotAnno (Shenzhen) Co., Ltd. was founded in April 2017, is a desktop robot arm solution professional R & D and manufacturer. Through continuous innovation, RobotAnno are committed to providing educational robots, light industrial robots and commercial robots with high performance-to-price ratio, easy to operate, and experienced desktop robot arm products and solutions.

PROBOT G602 is a six-axis robot arm product for scientific research and education. It is simple and easy to use. It can help you improve the efficiency of research and learning.

2.1.1 Product Description

PROBOT G602 is a light weight, fast, repeat positioning accuracy, high cost-effective consumption-grade desktop robot arm. Small shape, small volume, high-speed, high-precision completion of loading and unloading, sorting, assembly and other work.

PROBOT G602 provide serial port, network, IO and other rich communication methods, support ROS、Python、C++ secondary development, repeat positioning accuracy up to 0.1 mm, with simple and easy to use visual interactive software and simulation platform, support a variety of convenient switching end actuators to meet the needs of different tasks.

- Integration

Adopt integrated servo motor, small size, flexible work, fast speed, high precision;

- High performance

The controller uses the Xilinx Zynq chip with strong performance to ARM+FPGA the heterogeneous SoC so that the secondary development is flexible and efficient, and the multi-axis linkage interpolation control algorithm is adopted to improve the accuracy and stability of the system;

- Extensibility

Provide ROS/Python/C++SDK, support RS-232/Ethernet/Digital IO communication expansion to maximize PROBOT G602 potential;

- Accessibility

Supports the ROS、 ROS-I framework, the rich open source function package, helps the user to use and the development quickly on the handset person;

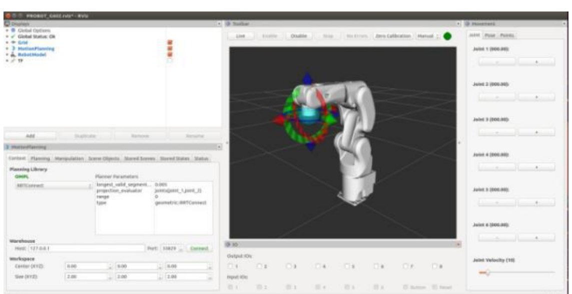
- Supporting courses

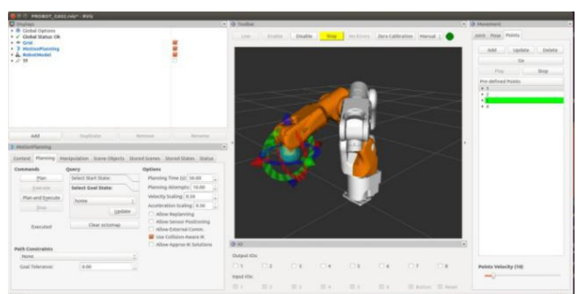
Choose the ROS arm development video course and teaching material, combine theory with practice, develop the mobile arm quickly; At the same time, optional course customization, robot online / offline training and other value-added services to provide comprehensive educational solutions;

- Ongoing update

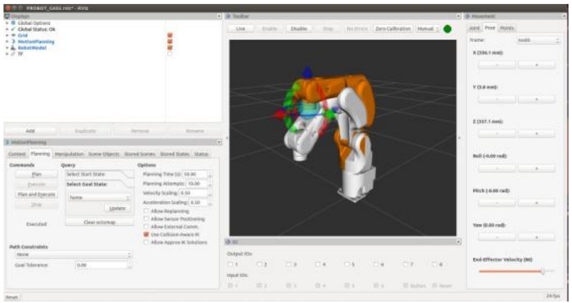
Support firmware update, software continuous iterative optimization, supporting applications and tutorials continue to upgrade.

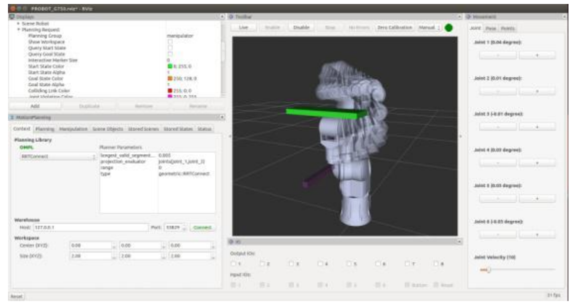
2.1.2 Functional characteristics





Virtualization Platform Support - PROBOT robotic arm complete The PROBOT arm can be described by dragging the 3D URDF model to support the on-line teaching function of the ROS、 Matlab and the model joint, and the robot arm CoppeliaSim carry out simulation analysis and real-time monitoring.position pose can be displayed in real time in Rivz.

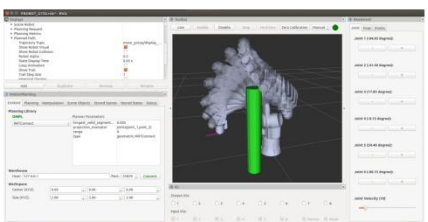


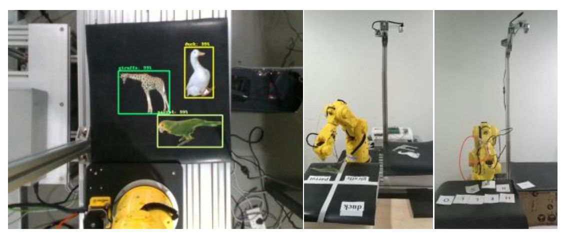


Click control - PROBOT arm upper support point movement   Obstacle avoidance planning - PROBOT arm supports dynamic obstacle avoidance,

Control, can realize the point movement to the joint and attitude, free through the path planning to automatically avoid the mechanical arm around

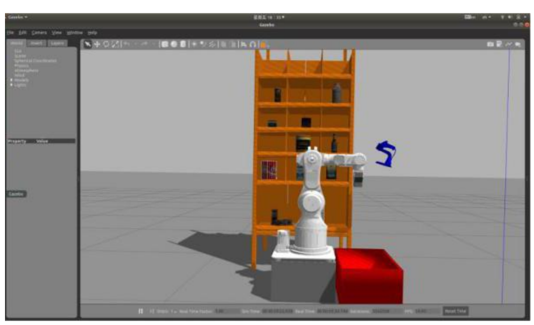
Adjust the attitude of the robot arm.Obstacles.





collision detection － P ROBOT robot arm supports grasping in working vision － PROBOT robot arm supports adding obstacle models through R GBD space and can identify, locate and grab objects in physical environments.

View collision areas in real time in Rvi z and Gazebo.



The application scene-intelligent shelf picking system supports digital twinning (Digital Twin), which can realize the comprehensive application of dynamic obstacle avoidance and cargo sorting of PROBOT arm.

2.1.3 Product appearance

- Figure 21 PROBOT G602 Mechanical arm appearance

- Figure 22 PROBOT G602 Appearance of the electronic control box of the robot arm

2.2 Description of control box

- Figure 23 Control box appearance (front)

On the front of the bottom control box, from left to right are: reset button (RESET), stop button (STOP), one button start button (BUTTON); there is a red power switch behind the control box.

(1) Stop button (STOP): in case of emergency, pressing down the stop button can stop the movement of the arm in time; when starting the arm again, the button should be twisted clockwise to relieve the emergency stop.

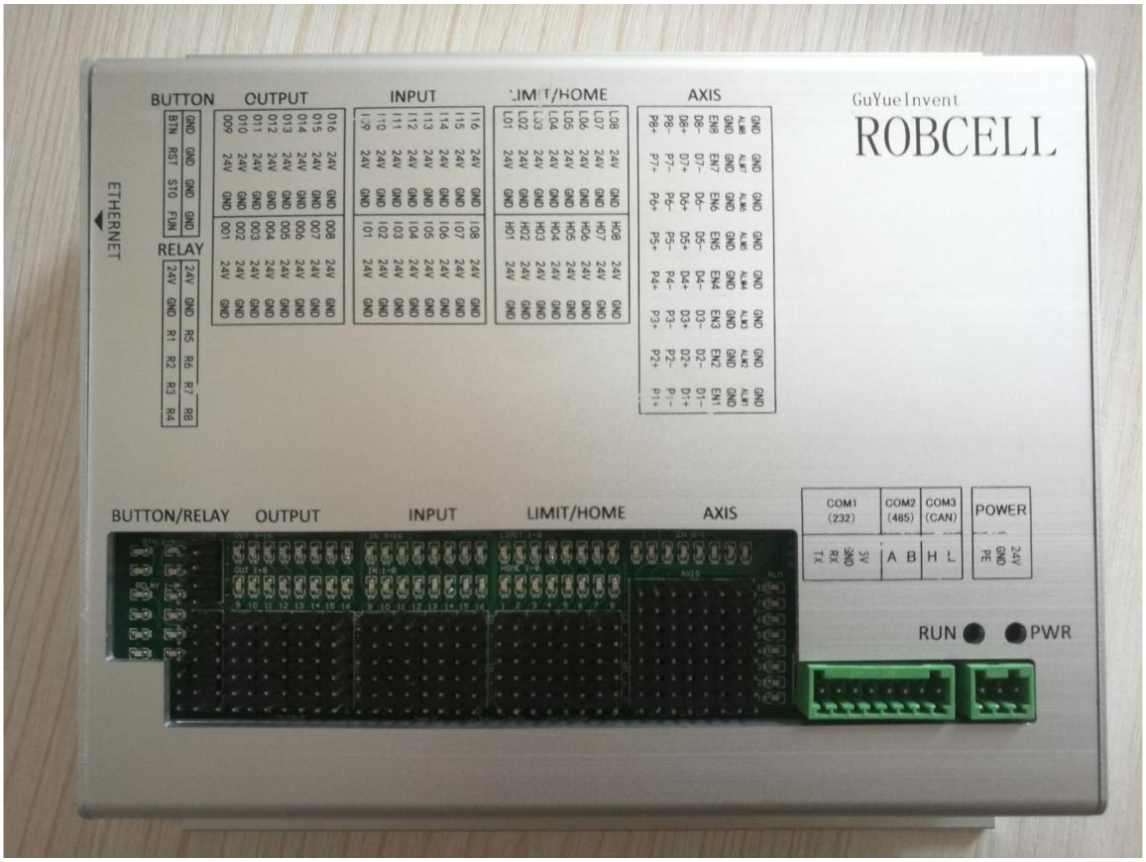
(2) One-click start button (BUTTON), reset button (RESET): connect to the digital input port of the controller, corresponding to the Button and Reset input port status in the host computer, for user programming.

Power switch: I is on and O is off.

- Figure 23 Control box appearance (after)

2.3 IO interface description

2.3.1 Controller IO Interface



- Figure 24 ROBCELL Controller (preinstalled in control cabinet)

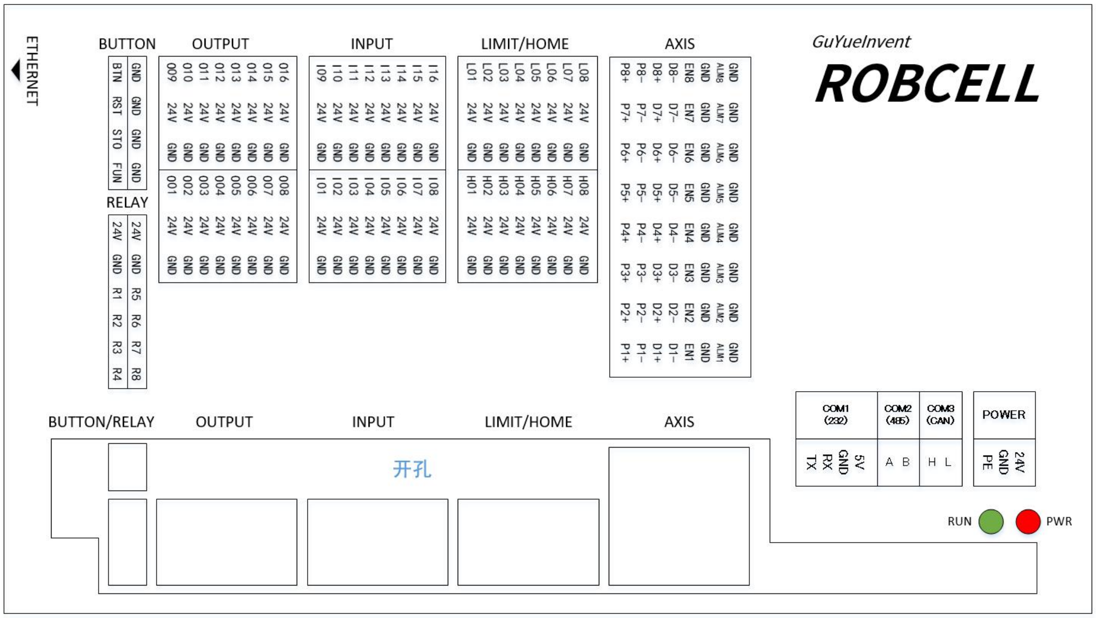
（1） BUTTON： connection control cabinet on the key, a total of 4 bits;

（2） RELAY： connection control cabinet relay use, a total of 8 bits;

（3） OUTPUT： digital IO output port, connect external output device, total 16 bits;

（4） INPUT： digital IO input port, connect external input device, total 16 bits;

（5） LIMIT/HOME： connecting the limit sensor on the robot arm ,8 bits each;



- Figure 25 ROBCELL Controller interface description (preinstalled in control cabinet)

2.3.2 Control cabinet IO interface

To facilitate the use of external equipment, the back end of the control cabinet through the DB 9 to connect four digital IO ports (2 output ,2 input), the specific pin distribution as follows:

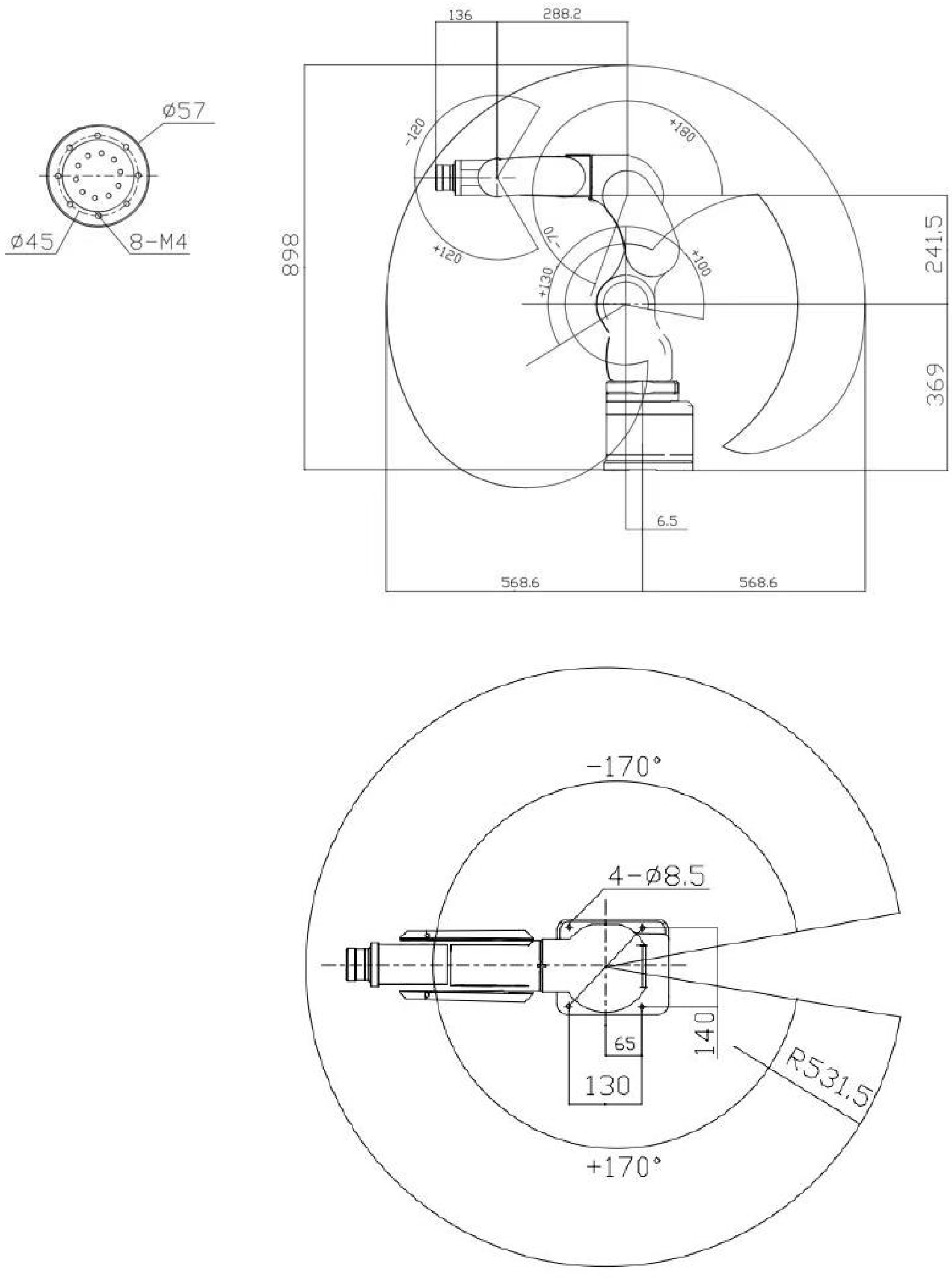
Serial number	Definition	Serial number	Definition	Serial number	Definition
1	RELAY1	4	I N2	7	G ND
2	RELAY2	5	V +24	8	G ND
3	I N1	6	V +24	9	G ND

2.4 Hardware parameters

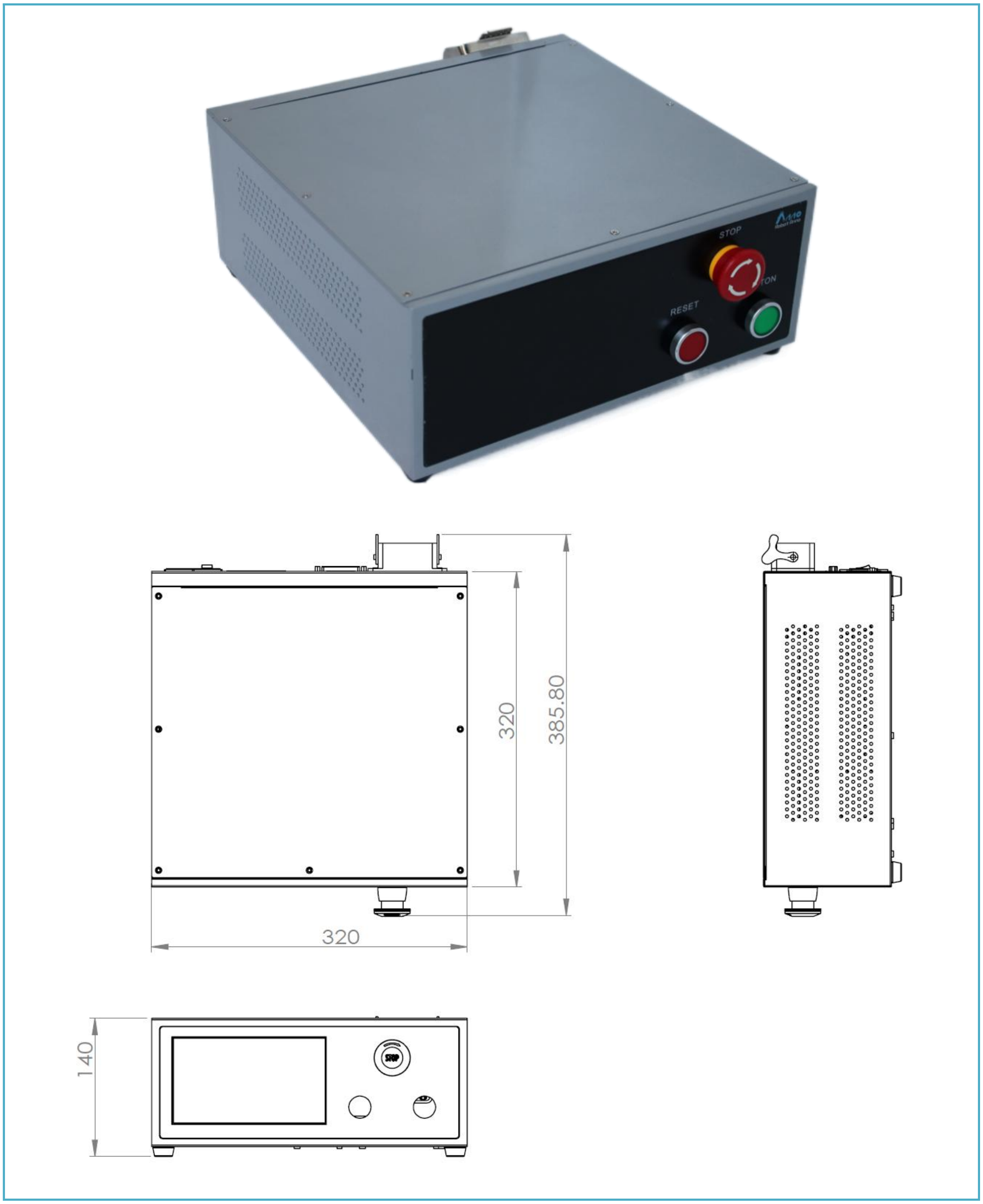
Mechanical arm type		**Six axes**	
Axis	Range of motion	Maximum rotation	
Axis 1	±170°	370°/ s	
Axis 2	±110°	370°/ s	
Axis 3	+40°~ -220°	° s 430	
Axis 4	±185°	300°/ s	
Axis 5	±125°	°/ s 460	
Axis 6	±360°	600°/ s	
Power supply voltage	220V/110V	**Weight**	28KG
Repeat positioning accuracy	±0.05 mm	**Reduction device**	Harmonic Reduction
Power supply voltage	220V/110V	**Rated load**	3KG
Maximum working radius	540 mm	**Maximum load**	4.5 KG
Type of motor	Servo Motor	**Controller**	ROBCELL A1

2.5 Structural dimensions

Dimensions



Size of Electronic control box	320mm*320mm*140mm
Size Chart of Electronic Control Box	


STOP
RESET
ON
320
385.80
320
140

3 Quick Start Guide

3.1 Hardware interface and connection

Before use, please refer to 2.2 control box instructions to understand the control box interface and panel functions, and then follow the following instructions to complete the hardware wiring.

Complete the connection between the control box and the robot arm, wire, digital IO interface as shown below:

（1）Connect the heavy load joint (male head) of the control box end to the joint end corresponding to the mechanical arm (female head);

(2) Connect the power cord of the robot arm control box to 220 V electricity;

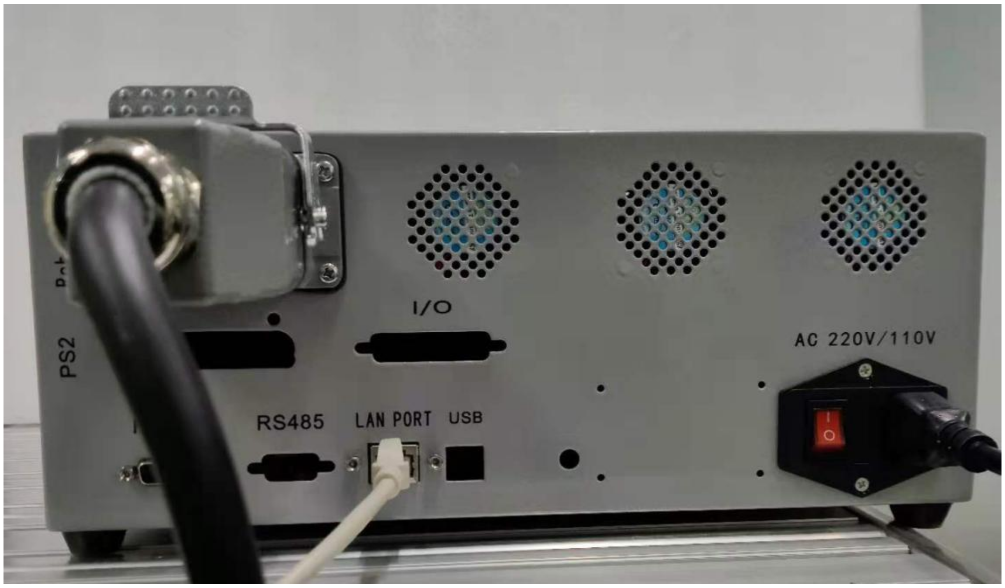
(3) Connect the PC net port with the control box net port of the robot arm by using the network wire;

(4) Connect external IO devices, such as claws, suckers, on DB 9 connectors as needed;

(5) Connect the round hole charger head to the round head of the control cabinet.

⚠ 注意

If IO function is not used, step 4 can be skipped.



- Figure 31 Physical drawing of cabinets

3.2 ROSs2GO start

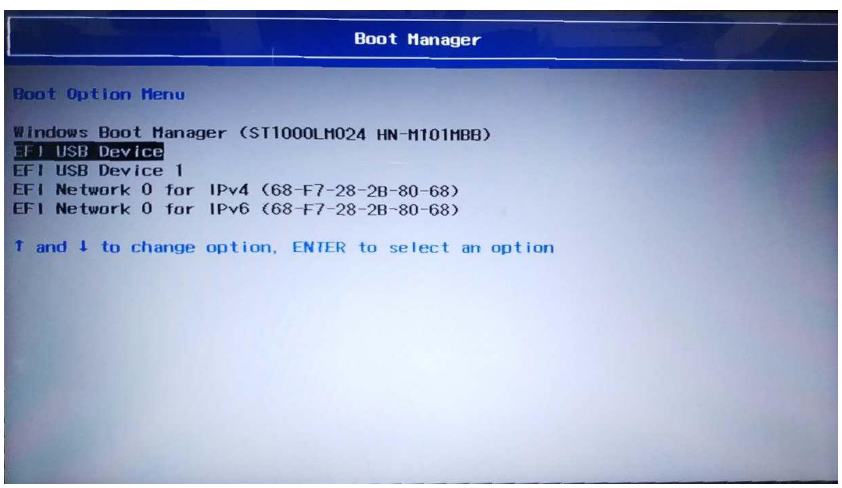
The company for the majority of users to provide plug and play ROS2GO mini U disk, and in the ROS2GO installation and deployment of this product required a set of ROS environment and host computer, users can directly use ROS2GO start the experience journey!

- **Figure 32 ROS2GO Mini U tray**

（1）Plug the ROS2GO into PC, start / restart PC, and enter the BIOS settings to boot the U disk using UEFI mode ,(different models of PC, enter the system in different ways, please refer to the use of PC startup mode).

注意

need to turn off Secure Boot (safe start) function in BIOS.



- **Figure 33 BIOS Setting U Disk Startup Diagram**

（2）After setting up, wait for the ROS2GO to complete the startup, after the successful start can see the following system desktop.

- Figure 34 ROS2GO Desktop system

3.3 Power on system

（1）Check the hardware connection again to make sure the connection is correct.

（2）Connect the back end power outlet of the control box to the external power supply, and press the front power switch to power the control box.

（3）Waiting for about 20 s, after power on will hear the buzzer ring twice in a row, the duration is short, indicating that the PROBOT G602 hardware and software system started successfully.

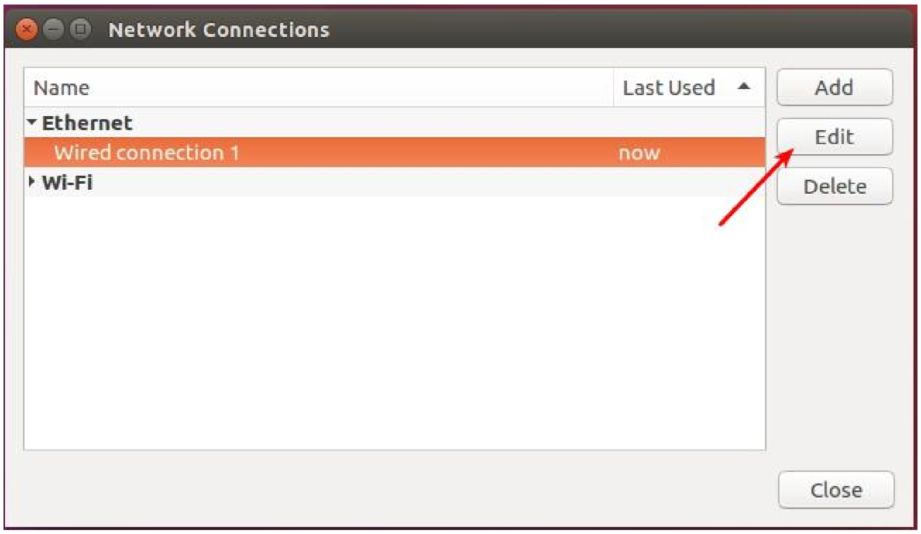
（4）Release the stop button.

3.4 Set Network Connection

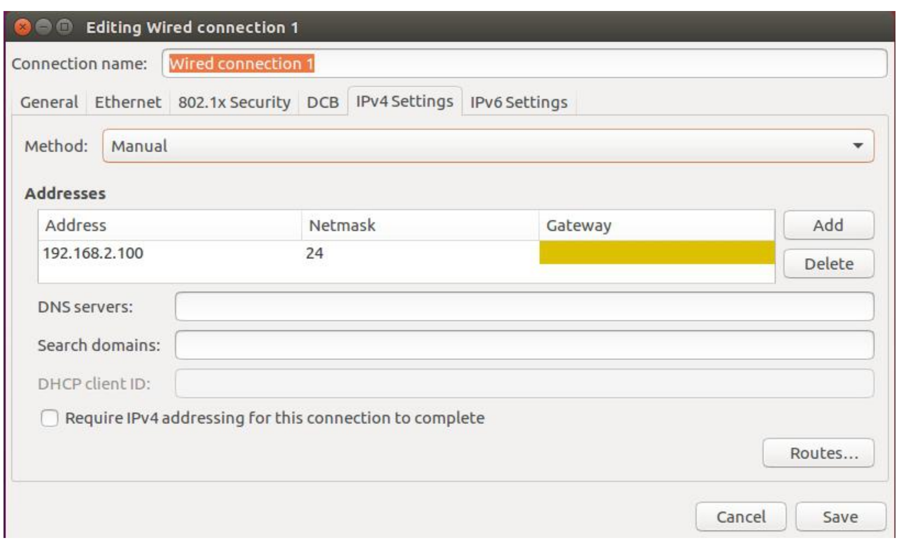
⚠ 注意

The controller and the upper computer are connected through the network cable, both need to be under the same LAN, the IP of the controller is fixed to 192.168.2.123, the ROS 2GO has been configured when it leaves the factory IP, but due to the different network cards of different hosts, there may be changes. Make sure the controller is connected successfully to the host computer network.

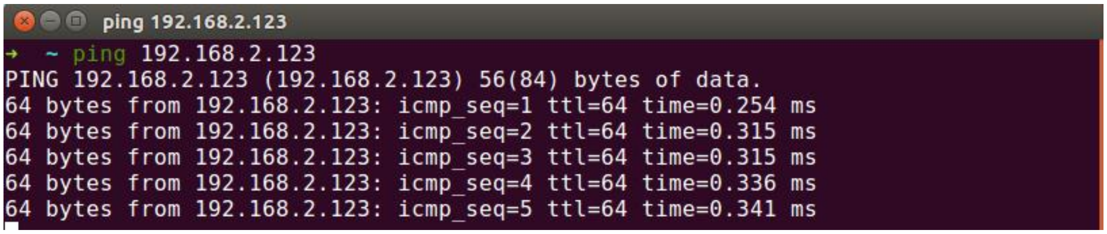
After entering the ROS 2GO desktop, click on the top right corner of the network logo, select the network configuration:



After entering the configuration interface, configure as shown below:



Save the configuration information and reconnect the wired network, see the successful connection information indicates that the configuration is complete, you can use the ping instructions to confirm:



3.5 Boot ROS Upper Machine

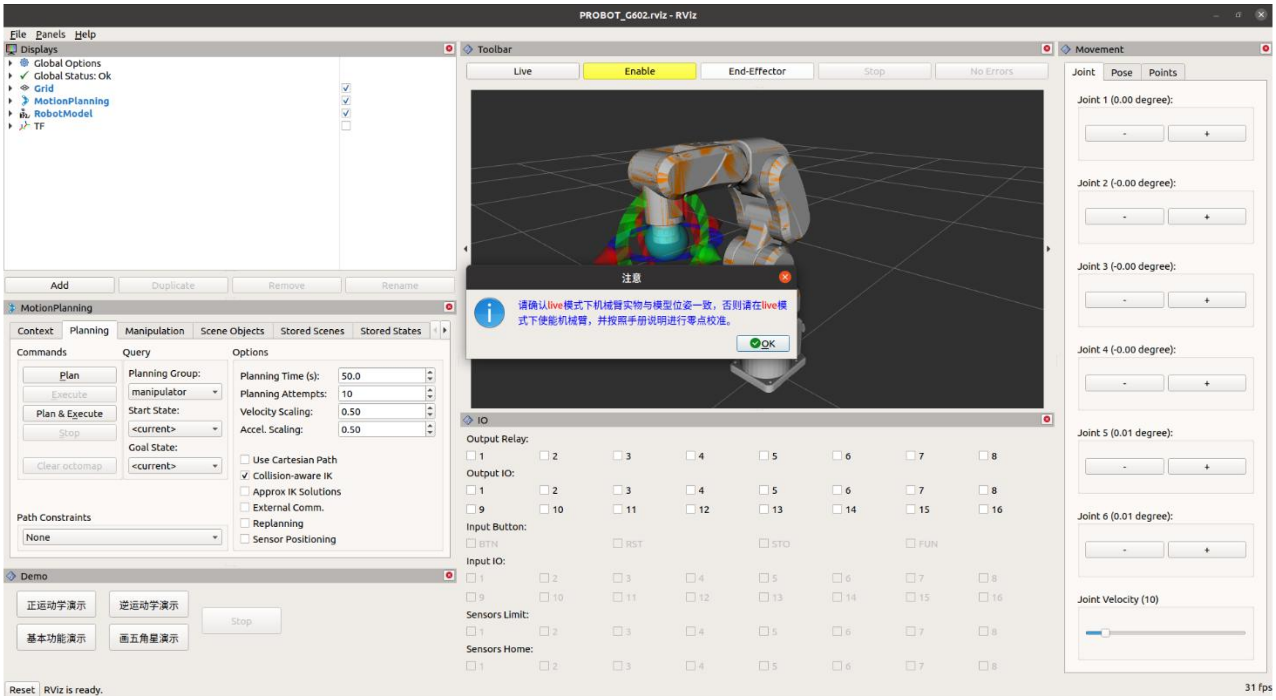
注意

When the controller is powered on, the PROBOT hardware and software system needs about 20 s, to start and the buzzer will ring twice continuously.Be sure to start the ROS PC after hearing the continuous noise!Otherwise, you need to restart the ROS host computer.

When the system starts, you can double-click the desktop icon "PROBOT Studio-G602"" to start the G602 arm ROS host computer.

Can also use the following command to start the ROS host:

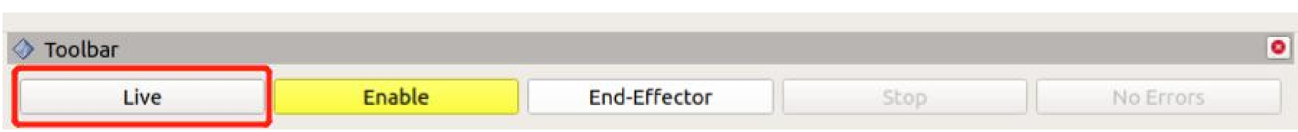
```
$roslaunch probot_bringup probot_g602_bringup.launch
```



- Figure 35 Operation of the real machine

3.6 Simulation/online switching

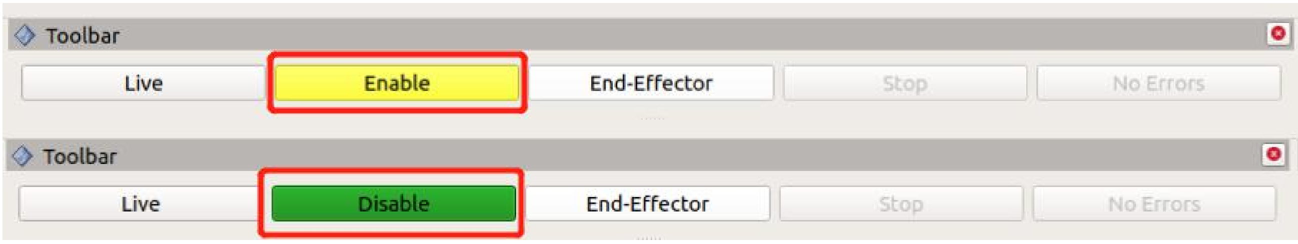
ROS the host computer starts, it enters the real machine control environment by default. It can click the simulation / real machine switch button in the control bar and switch to the simulation environment. Key display Live indicates that the current real-machine control environment, Sim the simulation environment:



- Figure 36 Switching to Real Machine Mode

3.7 Arm enabling

Click the Enable/Denergize button in the control bar to enable the robot arm to control, and the key yellow indicator light turns green:



- Figure 37 Enabling robot control

3.8 Adjust the initial position of the robot arm

PROBOT G602 the joints of the robot arm can not be pushed manually, if the physical robot does not conform to the pose of the model, please refer to the zero calibration section and use this function to re-calibrate the axis with position deviation.

3.9 Dynamic control of robot arm

Click on the "+" and "-" symbols of the corresponding axis / pose in the Move ment "area on the right side of the upper computer to control the point motion of the robot arm in the joint space / workspace. If the model and the object move synchronously, the system starts successfully.

For more use, please refer to the following instructions.

4 Use Of Robot Arm

4.1 Basic operations

4.1.1 Adjust the initial position of the robot arm

PROBOT G602 some joints of the robot arm can not be pushed manually, please refer to the zero calibration section and use this function to re-calibrate the axis with position deviation.

4.1.2 Start the robot arm

（1）Check the hardware connection again to make sure the connection is correct.

（2）Connect the back end power outlet of the control box to the external power supply, and press the front power switch to power the control box.

（3）Waiting for about 20 s, after power on will hear the buzzer ring twice in a row, the duration is short, indicating that the PROBOT G602 hardware and software system started successfully.

（4）Release the stop button.

注意

Before starting the arm, check the outer wire cover and the outer packing for breakage. If abnormal, repair or take other necessary measures in time.

You must wait for the buzzer to ring twice before you can continue the follow-up! Otherwise PROBOT G602 the hardware and software system has not been started, ROS the host computer will not be able to establish a connection with the controller.

The stop button is pressed to indicate that it is in a state of emergency, and the clockwise twist can relieve the state of emergency.

危险

Please do not move the arm while the arm is working. If you want to move, turn off the power of the arm and then move the arm.

4.1.3 Close the robot arm

After the operation, please follow the following instructions, control the robot arm back to the initial position "home", and then click" Disable", press the control cabinet stop button, turn off the control cabinet power, complete shutdown.

4.1.4 Stop

In case of emergency, pressing down the stop button can stop the operation of the arm in time; when running the arm again, twist the button clockwise to relieve the emergency stop state;

危险

In case of emergency, press the emergency stop button immediately, if the brake arm can not be timely, it may cause personal injury or equipment damage accident.

Stop key release emergency stop

• Please do not move and maintain the arm while the arm is working. If you want to move and maintain, turn off the power of the arm and perform these operations after the power is off.

4.2 Host computer ROS environment

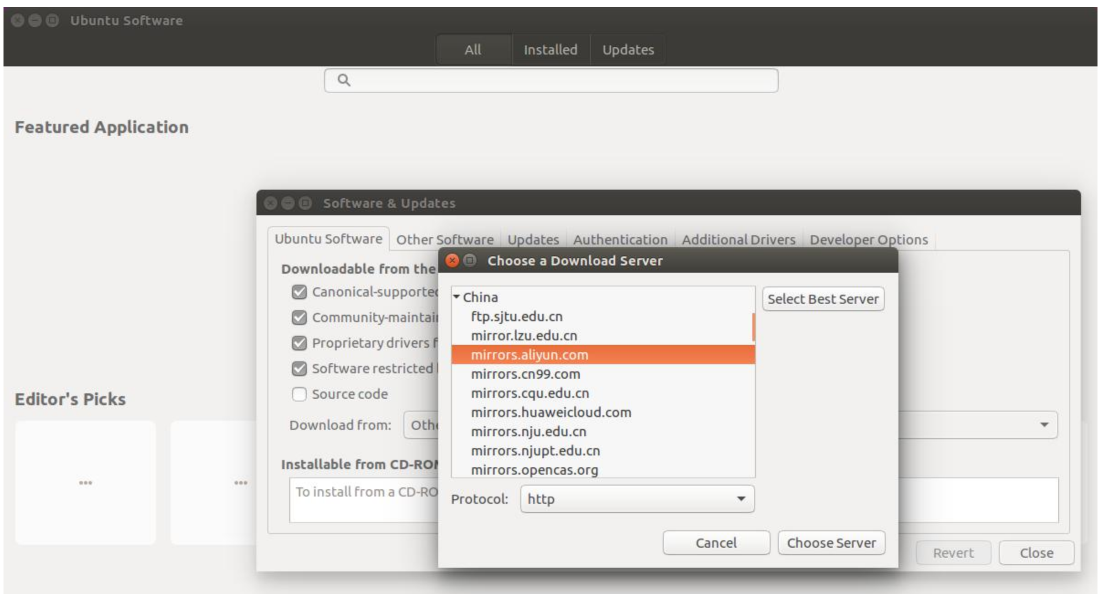
The use of PROBOT robot arm is based on Ubuntu and ROS environments. If you use this product equipped with customized ROS2GO, you can directly start ROS2GO, use pre-installed PC ROS environment. Otherwise, you need to install Ubuntu system and ROS related software on your own PC, please refer to the following steps.

4.2.1 Installation of Ubuntu 16.04/18.04 systems

Please skip this section if you use the customized ROS2GO, of this product and start ROS2GO use the pre-installed PC ROS environment directly.

(1) Dual system hard disk installation or virtual machine installation, hard disk space should be more than 20 GB, Ubuntu image download address is: http://www.ubuntu.com/download/

(2) After completion of the installation, it is recommended that domestic users modify the system software source as Ali Cloud / Huawei Cloud (selected according to specific circumstances):



- Figure 41 Modification of software source

(3) Enter the following command at the terminal to update the system software source:

```
$sudo apt-get update
$sudo apt-get upgrade
```

4.2.2 Use PC to install assistant

Please skip this section if you use the customized ROS2GO, of this product and start ROS2GO use the pre-installed PC ROS environment directly.

To install this product ROS the host computer, please download / obtain the PROBOT Studio host computer installation assistant probot_studio_setup.sh.x, as well as the installation package (such as: probot_studi o_v3_2_1-ubuntu 18_04.zip).

(1) Turn on the host computer installation assistant

Place the host computer installation assistant and installation package file under the same folder.

Confirm that the installation assistant has executable permissions. Check the probot_studio_setup.sh.x and right-click on "Properties "-" Permissions ", check" Allow to execute as a program file ".

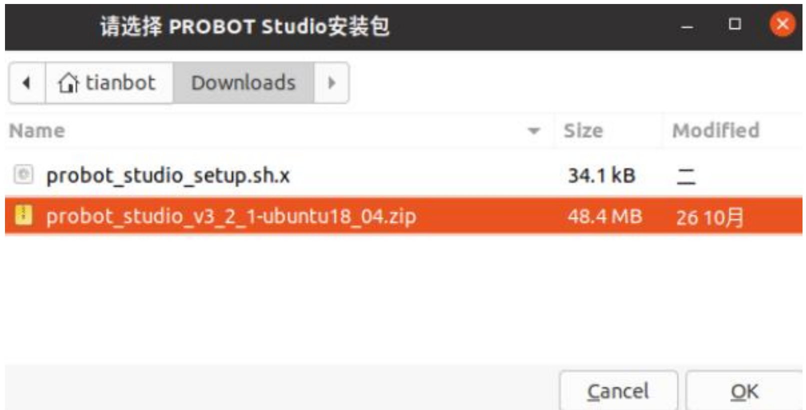
- Figure 42 View/modify executable permissions

Open the terminal input the following command, start the upper computer installation assistant. First start, will automatically install the program to run the required dependency package, please wait patiently.

```
$./ probot_studio_setup.sh.x
```

(2) Select the PC Installation Package

Note the selection of the installation package corresponding to the current Ubuntu system version, the current support Ubuntu 16.04/18.04. Click OK". to select



- Figure 43 Select Upper Locator Installation Package

(3) Select the installation item

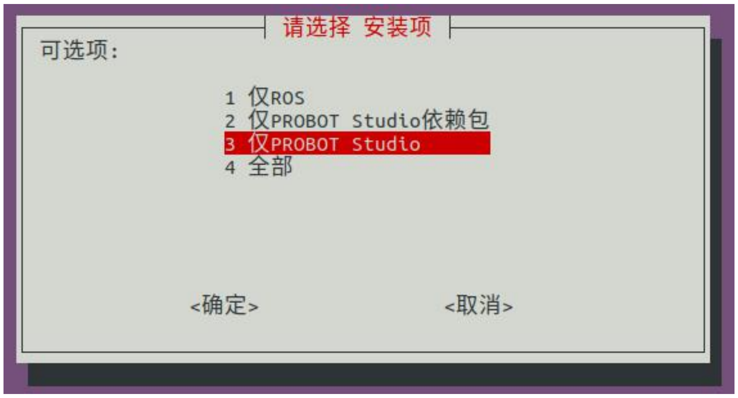
Installation of the host computer includes:

1. ROS： only select this to install ROS operating system only.

2. only PROBOT Studio dependency packages: select this only to install the dependency packages required PROBOT Studio the host computer.

3. PROBOT Studio： only select this to install PROBOT Studio upper computer only.

4. all: select this item to install ROS operating system, PROBOT Studio host computer and its required dependencies in turn.



- Figure 44 Select Setup

This interface only supports moving the cursor through the keyboard directional key. Press up / down to switch options, press left / right to switch to OK / cancel, press enter.

(4) Selection of mechanical arm models

⚠ 注意

Please select according to the actual robot arm model, after installation, the upper computer only supports the selected robot arm.

- Figure 45 Selection of Arm Type

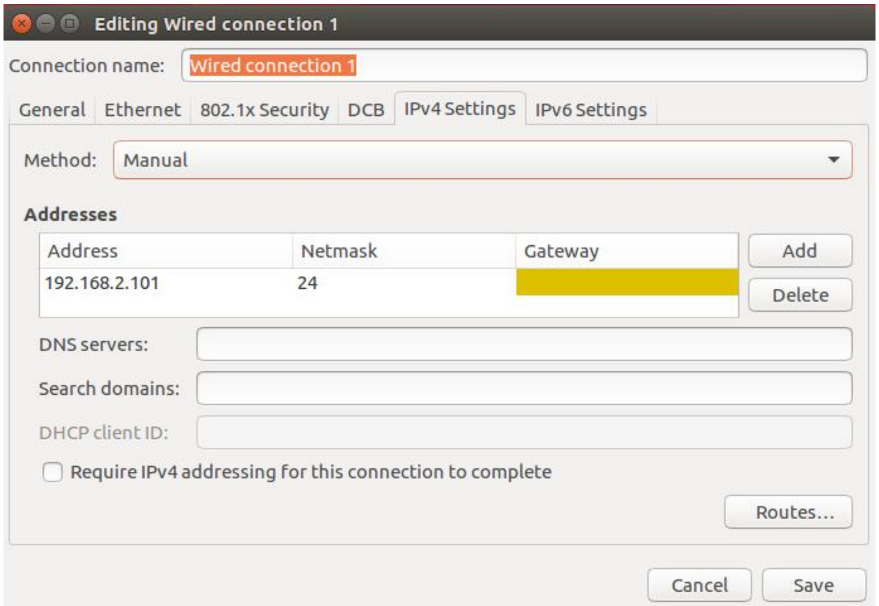
(5) Start installation

The installation time depends on the network environment, please wait patiently. If the upgrade is successful, the terminal will print the following information. If it takes too long, try replacing the network.

- Figure 46 Installation success

4.2.3 Set PC End IP Address

The controller communicates with the upper computer through the network cable. Before using, it is necessary to manually configure the address IP the PC end to make it in the same LAN as the controller. The default IP of the controller is 192.168.2.123, and the PC IP address should be configured to 192.168.2. XXX, such as 192.168.2.101.



- Figure 47 Setting static IP

4.3 Introduction to the Function of Upper Locator

First of all, the main interface partition and its functions are introduced, which are divided into six regions:

- Visual Plug-in Configuration Area:
- ROS MoveIt! Visual Control Area:
- Key Control Area:

  Simulation / real-machine environment switch key, robot control enable key, deenergize key, stop key, clear error key, zero calibration key, status indicator light;

- Model Dynamic Display Area:

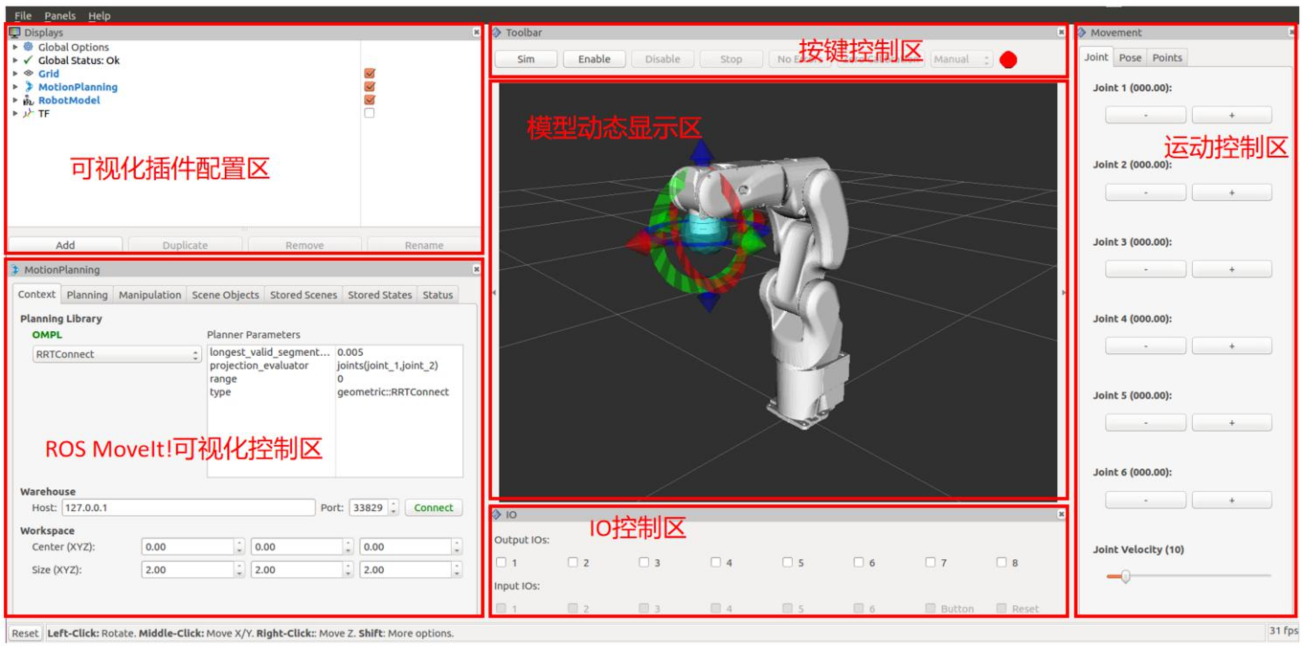
  Display the actual position and target position of the robot, use the mouse to drag the robot end drag ball can change the target position;

- IO Control Area:

Digital IO output control and input status display;

- Movement Control Area:

  Joint space point movement, workspace point movement and teaching point function.



- Figure 48 Main interface of host computer

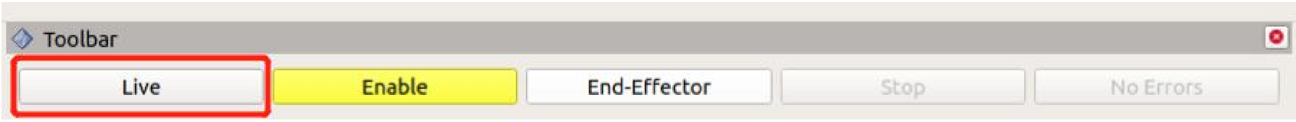
注意

According to the actual robot arm type, the corresponding robot arm model will be displayed in the dynamic display area of the model. But the host interface is basically the same as the above.

4.3.1 Control bar key function

（1）Simulation/real-machine environment switching

Click on this button to switch the simulation / real machine environment. Be sure to switch in Disable state. Sim indicates that the simulation environment is currently in place, Live that the real-machine control environment is currently in place:

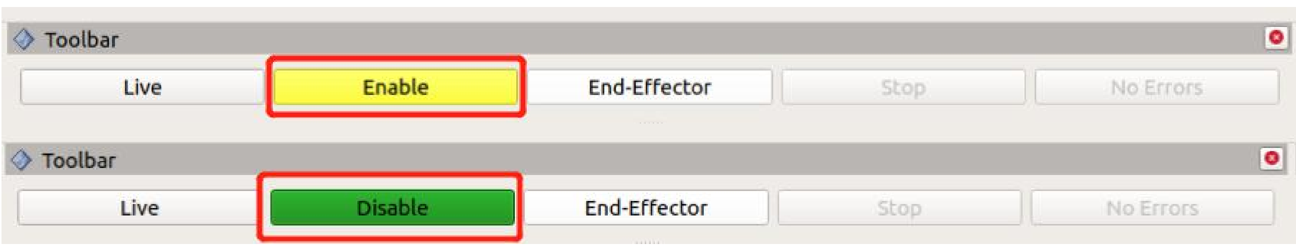


- Figure 49 Simulation/Treality Switching

（2）Robot control enable/dividing

Click on the Enable to enable robot control, click on the Disable division (in the simulation environment without enable).

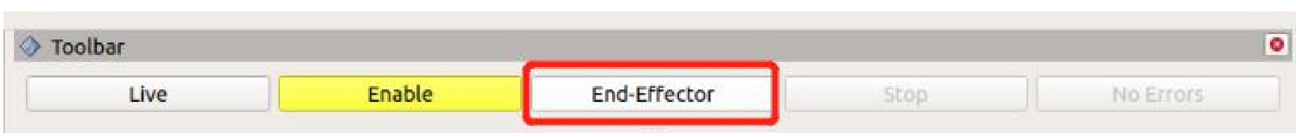
The status indicator shows that the red light indicates that the current robot is in a controlled deenergized state, and the green light indicates that the current robot is in a controlled enabling state:



- Figure 410 Robot control enable/degenerate

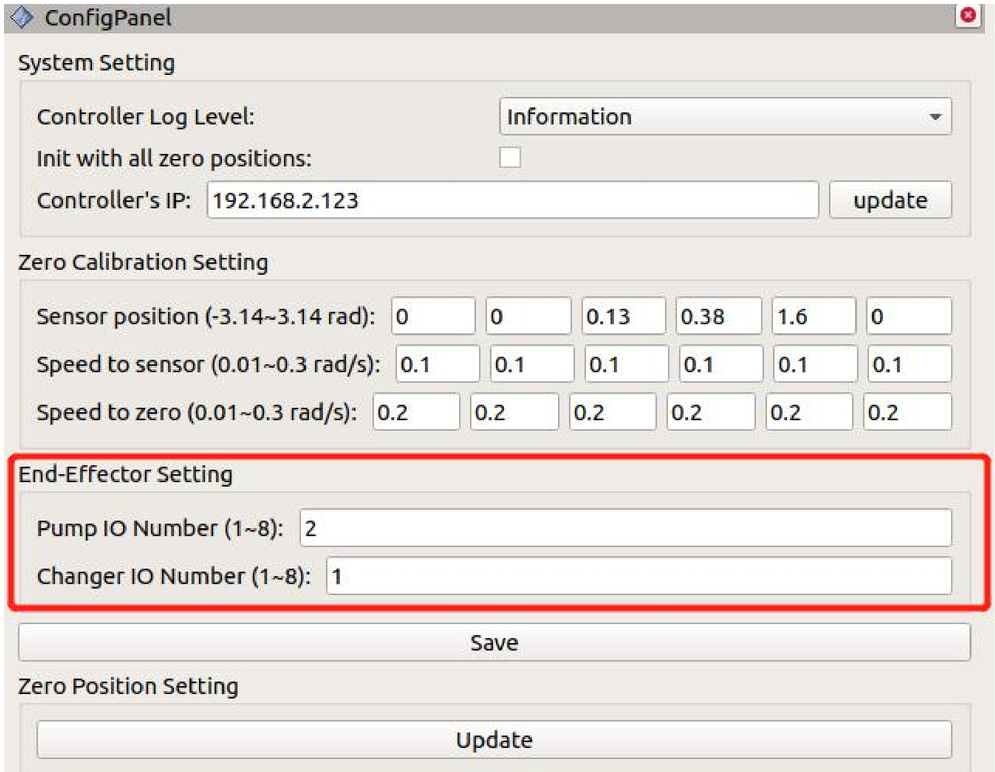
(3) End-effector

Click on the End-E ffector to control the working state of the robot terminal actuator-claw or sucker, such as the robot end installation claw, click on the End-E ffector, claw closed, click again, claw open (if the claw does not move, you can first open the air pump, Then open the solenoid valve).



- Figure 46 Robot Terminal Actuator

Under the Config P anel page before use, set the air pump and solenoid valve corresponding I O port, factory default air pump connection I O 1, solenoid valve connection I O 2, as shown below.

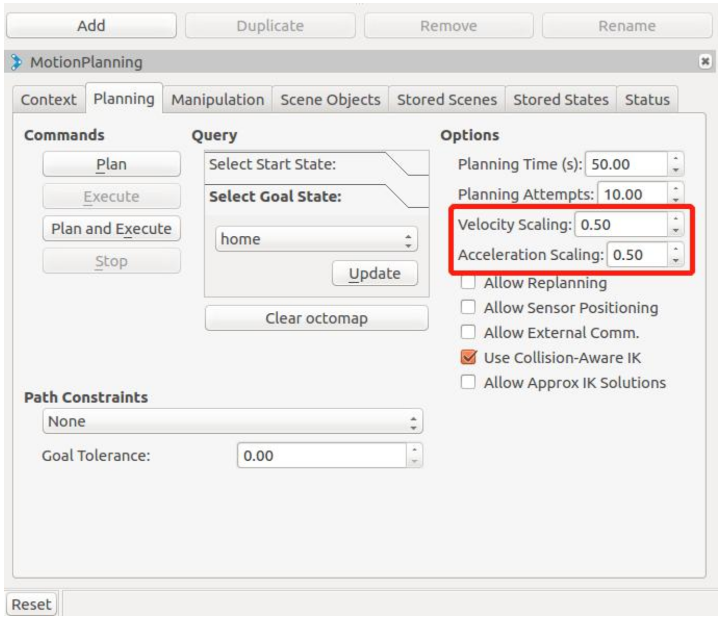


- Figure 47 I O Installation of Robot Terminal Actuator

4.3.2 Drag to teach

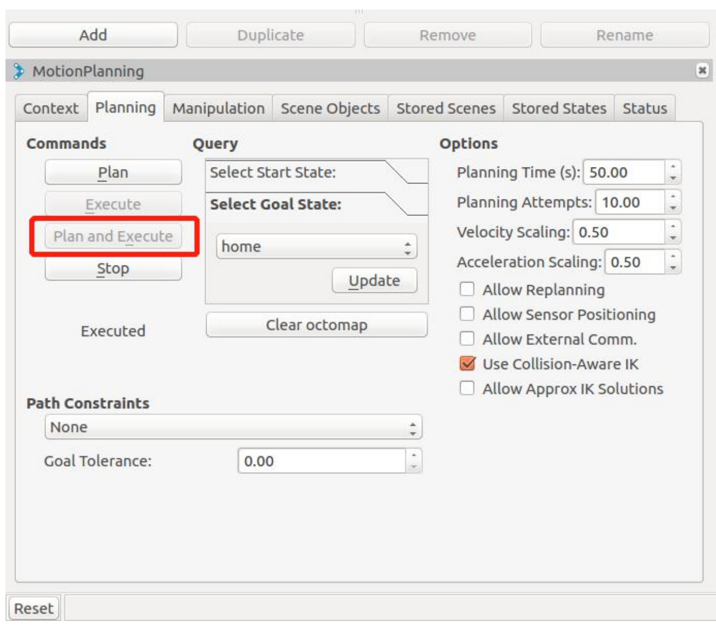
（1）The mouse is positioned on the control ball of the robot model terminal, and the robot can be dragged by the left mouse button. When the mouse is loosened, the yellow robot model is the target pose, while the silver robot model is the current actual pose.

（2）set the planned maximum time limit and the velocity and acceleration of the robot motion (percentage ,1 means 100%, i.e. set to the maximum).



- Figure 48 Drag to show the speed and acceleration settings

（3）Click ROS MoveIt! The robot model starts to move when the Plan and Execute button in the Planning label page of the visual control area (during the movement, the Plan and Execute button becomes gray and can not be operated, waiting for the current action to be completed).

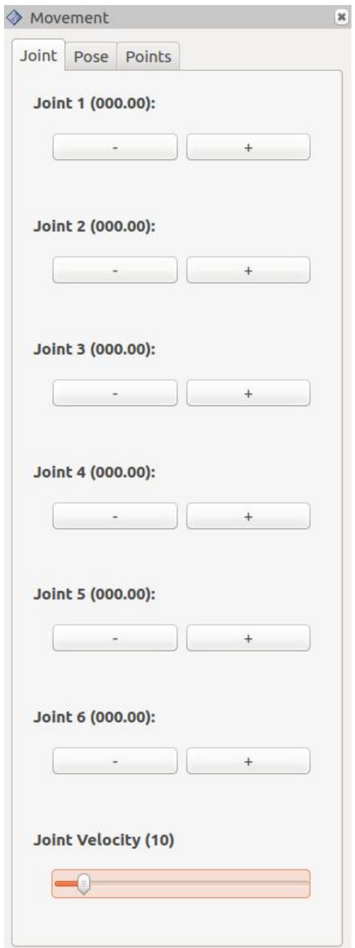


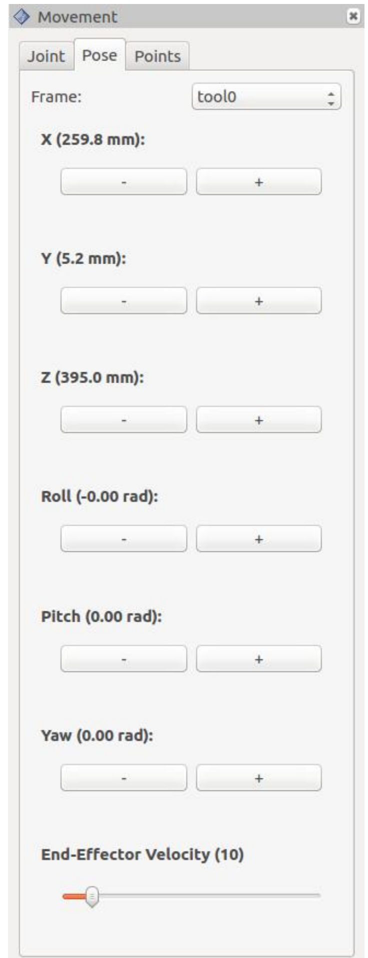
- Figure 49 Drag to teach execution

(4) The robot moves from the current pose to the target pose until the interface shows that the two models completely coincide and the robot motion ends.

4.3.3 Joint Space Point Movement

(1) All functions of joint motion are on the Joint tab of the right motion control area. The motion of the joint can be controlled by long press +, and the motion speed of the Joint Velocity slide bar on the lower side can be adjusted.





- Fig .410 Space Point Motion Control Figure 411 Workspace Point Motion Control Figure-

4.3.4 Workspace point movement

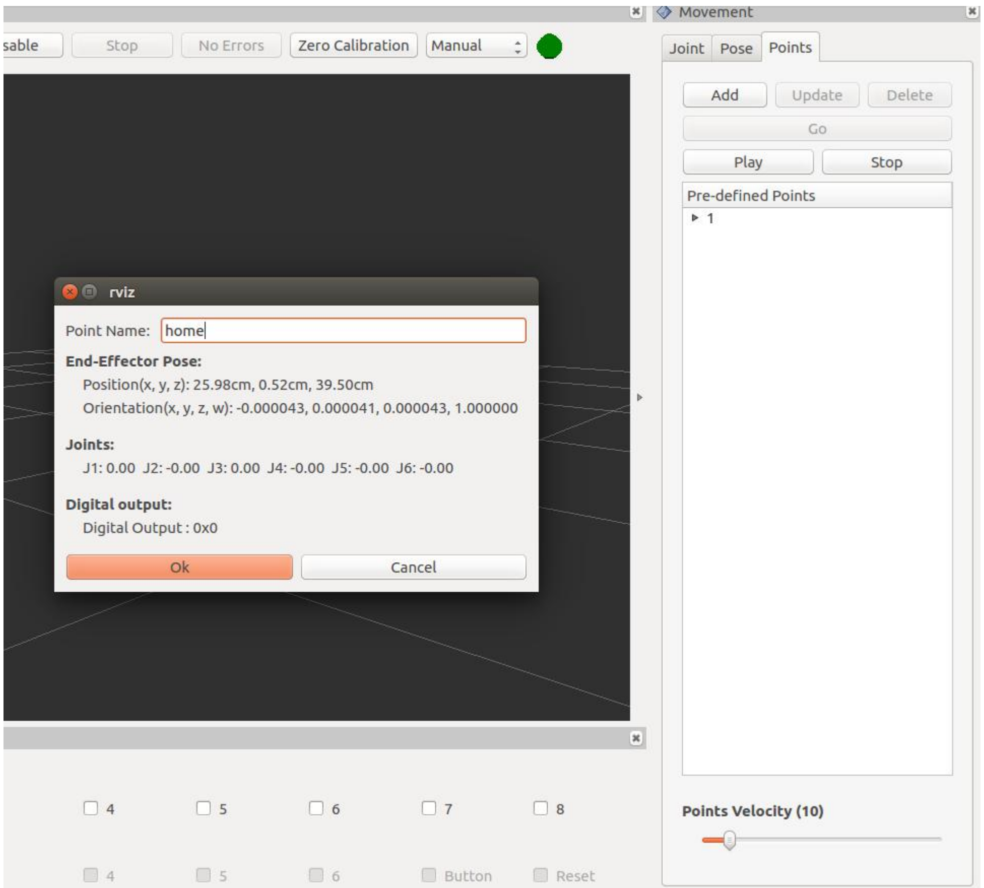
（1）All functions of spatial click are on the Pose tab of the right motion control area.

（2）Through the long press +,- can control the robot end to perform workspace point movement, the lower side of the End-Effector Velocity slider adjustable speed.

4.3.5 Save and run teaching points

Display the save and run function of the teaching point in the Points tab of the right motion control area.

（1） Add： add teaching points.



- Figure 412 Add Teaching Points

When you click the Add button, a dialog box will pop up to add the teaching point, automatically detect the current pose and IO status of the robot arm, at the same time, you can add the delay before the operation, and save the teaching point as a custom name.

（2） Update： update the current teaching point.

Under any position of the robot arm, select any teaching point in the list of teaching points, click the Update button, you can update the teaching point to the current position and IO state of the robot.

（3） Delete： Delete the current tutorial

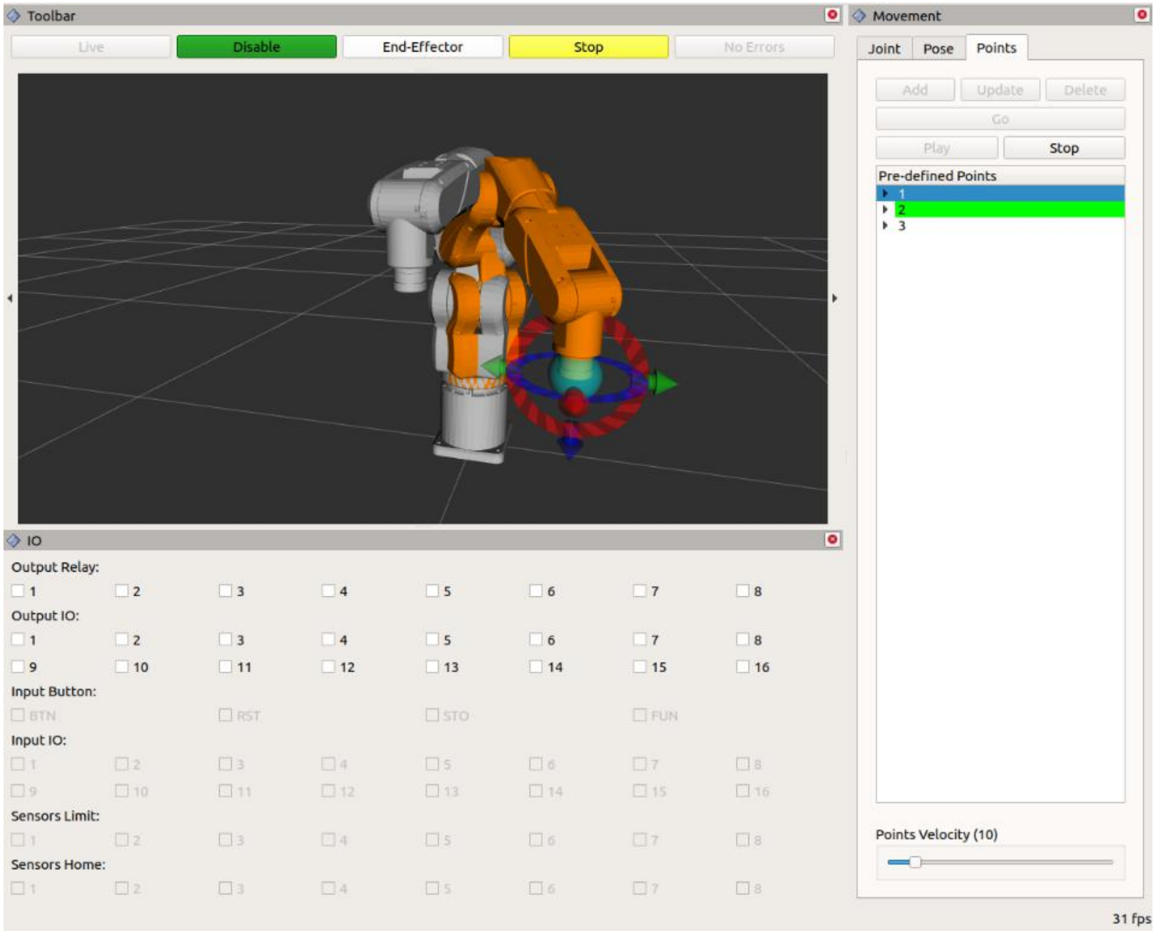
Select any teaching point in the list of teaching points, click the Delete button, you can delete the teaching point.

（4） Go： movement to the point of instruction.

Select any teaching points in the list of teaching points, click on the Go button, you can let the robot arm move to the teaching point saved posture and IO state.

（5） Play： loop runs a list of teaching points.

Click the Play button, start from the first teaching point to cycle the teaching point in the running list, click the Stop button to stop running, the current running target point will become a green background.



- Figure 413 running points

（6） Stop： stop loop running show point list.

（7） Points Velocity： adjust the speed of the teaching point.

4.3.6 Show that the teaching point is offline

Show point running supports offline mode. Refer to the previous section, add and save the teaching points through the host computer. After saving the teaching point information, you can support the offline operation of the teaching point.The operation is as follows:

（1） The control box is powered on without starting the upper computer.

（2） Long press (about 3 s) control box one-click start button (BUTTON), can enable robot arm control (and the upper computer control bar enable button function consistent).

(3) Click the control box button (BUTTON) to start the off-line operation function. The system will automatically read the information of the teaching points saved offline and execute one by one to realize the circular operation of the list of teaching points (which is consistent with the Play function of the Points label page of the motion control area on the right side of the upper computer).

(4) Short press control box reset button (RESET), robot arm motion deceleration stop. During an emergency, press the stop button (STOP) and the arm will stop moving immediately.

4.3.7 IO control

IO control function in the IO control area, digital IO is divided into the following categories:

(1)8 bits Output Relay ： relay used in the connecting control cabinet;

(2) Output IO ： the digital IO output port, connecting the external output device, a total of 16 bits;

(3) Input Button ： the keys on the control cabinet, a total of 4 bits;

Input IO： the digital IO input port and connect the external input device for a total of 16 bits;

(5) Sensors Limit/Home ： limit sensors ,8 bits each, connected to the robot arm;

- Figure 414 IO Control

The left click of the mouse to control the output port, will change its output level, the box in front of the port number hook represents the enable digital output (low level effective);

In the Input column, you can see the level signal of the digital input port, and the box in front of the port number indicates that the digital input port is valid (low level valid).

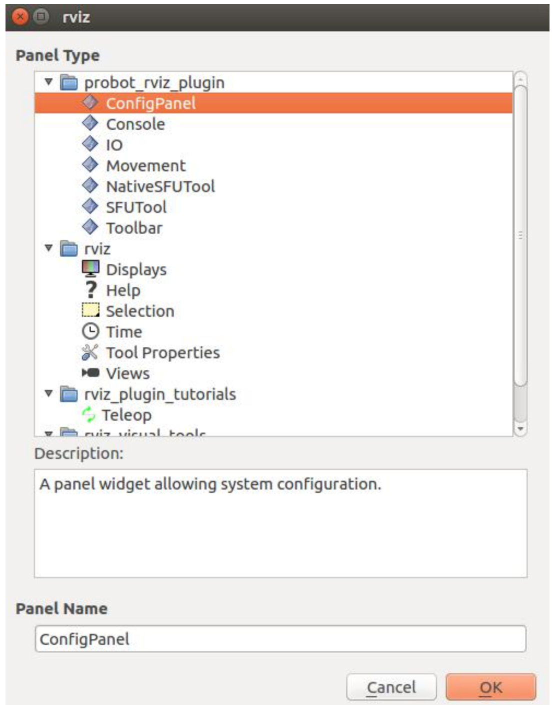
4.3.8 Zero calibration

The robot arm will be pre-calibrated before leaving the factory. If it is found that the position of the robot arm in the display area of the upper computer model is not in accordance with the actual situation, the zero bit calibration function can be used for synchronization. If in doubt, please contact the after-sales department immediately, do not disassemble the machine without authorization.

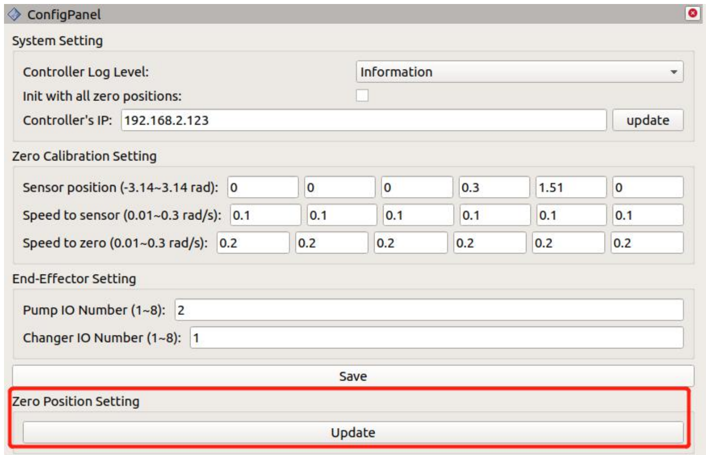
The steps of zero calibration are as follows:

1. use the joint space point motion control function (Joint) of the motion control area, the long press the +/- button to control the shaft movement, so that the joints of the robot arm are in zero position.

2. click on the upper left corner of the upper computer Panels, click on the Add new panel, selection ConfigPanel open the parameter configuration interface, click on the zero calibration completed, restart the control cabinet can take effect.



- Figure 415 Open ConfigPanel Configuration Panel



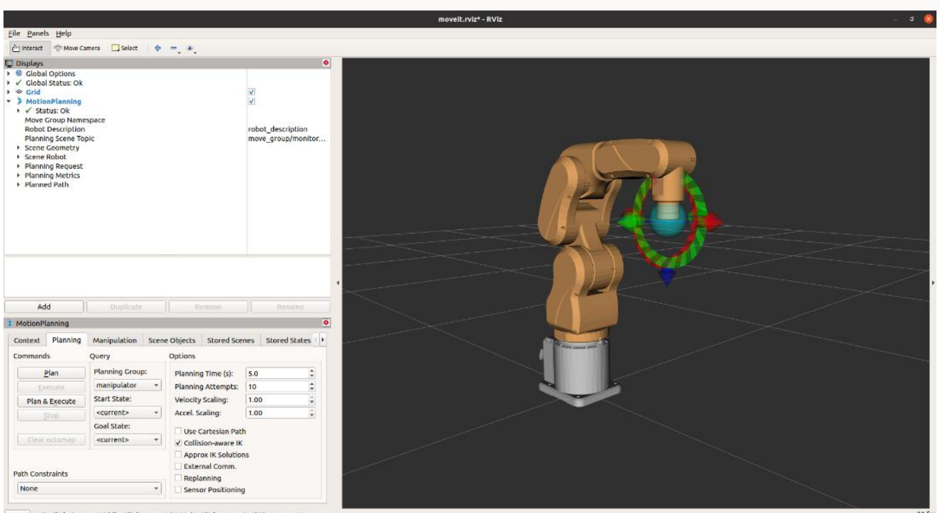
- Figure 416 ConfigPanel Configuration interface

4.4 Single Machine Simulation

The host computer ROS environment can be simulated on the PC without connecting controller and robot.

Use the following command to start the host computer and enter the simulation control environment:

```
$roslaunch probot _bringup probot_g602_bringup.launch sim：=true
```



- Figure 417 Single Machine Simulation Start

⚠ 注意

The upper computer interface started with the above command is different from the interface started under the real machine operation, and is more concise.

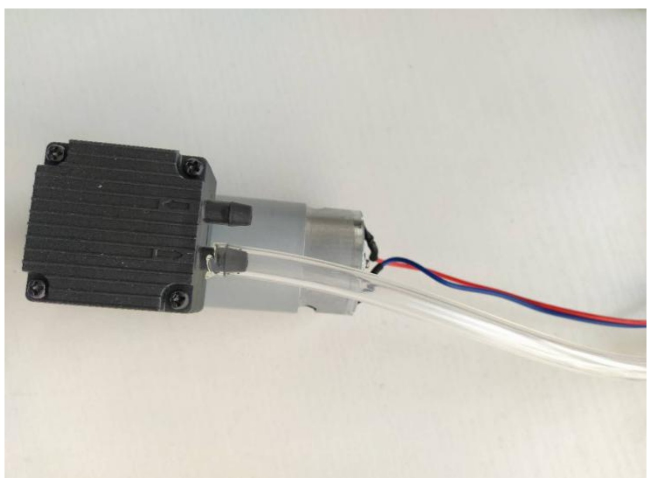
4.5 Operation of the real machine

Please refer to the 3 Quick Start Guide

For more routines, refer to the program in the probot _g602_demo function package.

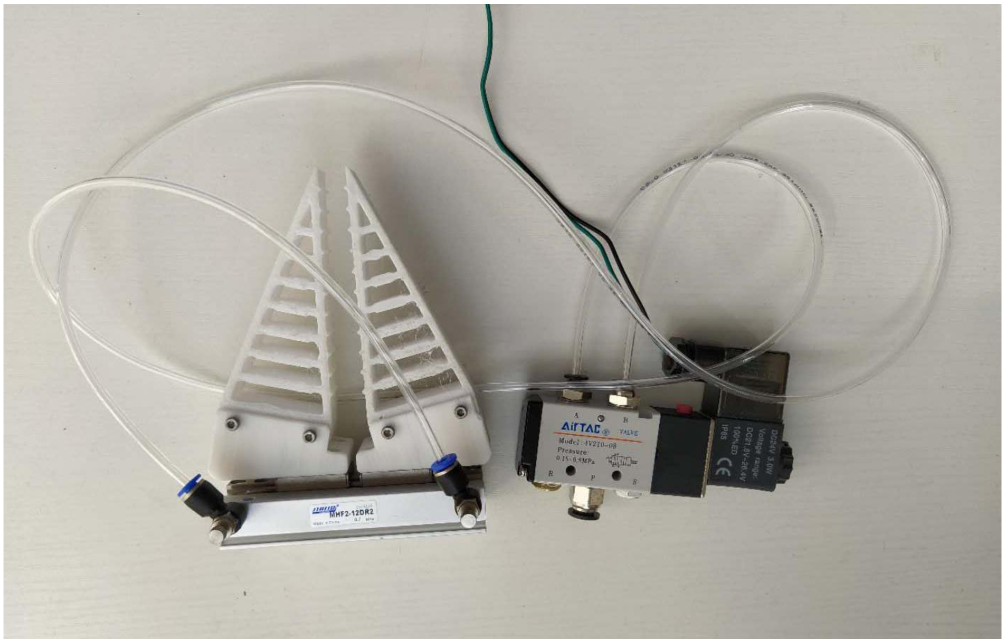
4.6 Use of Claws

（1） Add the power cord to the air pump bipolar welding and insert 8 mm trachea at the outlet with the arrow facing upward:

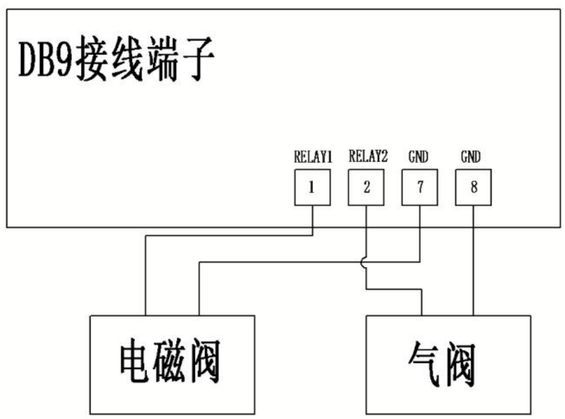


The other end is connected with the variable diameter joint, the other end of the joint is connected with the 6 mm trachea, and the 6 mm trachea is inserted into the solenoid valve P port, and the two 2.5 mm trachea is inserted into the A、B port,

(3) Connect the trachea of the solenoid valve A port to the right interface of the claw, and the B port to the left interface of the claw, as shown in the figure,



(4) Connect the solenoid valve, air pump and terminal according to the following figure:



Please run the real machine according to section 5.5 after the connection is successful, and refer to the IO control part, the output IO 1 and 2 are set to high power, the claw is closed; when the output IO 1 and 2 are set to low level, the claw is open.

⚠ 注意

Make sure the trachea is inserted into the deepest part of the airtight screw, otherwise the system leakage can not control the claw normally.

4.7 Routine demonstration

The upper computer supports the quick start of four demonstration routines, namely, forward kinematics demonstration, inverse kinematics demonstration, overall function demonstration and drawing pentagram demonstration. Click on any demo button and the robot arm starts executing the corresponding routine.

If you want to stop moving, click the stop button.

5 MoveIt! Programming Interface

PROBOT arm compatible ROS MoveIt! Programming interfaces can be used ROS MoveIt! environment Realize the application function development of robot.

注意

(1) The following function interfaces are polymorphic, see MoveIt! for detailed function descriptions Official presentation:

https://docs.ros.org/api/moveit_ros_planning_interface/html/namespacemoveit_1_1planning__interface.html

(2) See:

https://github.com/ps-micro,

Such as https://github.com/ps-micro / PROBOT_G602/tree/master/probot_demo/src

5.1 Setting the state of motion

(1) void moveit：：planning_interface：：MoveGroupInterface：：MoveGroupInterfaceImpl：：setMaxAccelerationScalingFactor (double max_acceleration_scaling_factor)

Function functions	Set the motion acceleration coefficient		
Function parameters	**Type**	**Name of name**	**Description**
	double	max_acceleration_scaling_factor	Acceleration upper limit coefficient, range :0~1

Return value	Value	Description
	-	-
Note		

Routine:

```
moveit：：planning_interface：：MoveGroupInterface arm (" administrator ");
arm.setMaxAccelerationScalingFactor(0.2);
```

(2) void moveit：：planning_interface：：MoveGroupInterface：：MoveGroupInterfaceImpl：：setMaxVelocityScalingFactor (double max_velocity_scaling_factor)

Function functions	Set the velocity coefficient		
Function parameters	**Type**	**Name of name**	**Description**
	double	max_velocity_scaling_factor	Velocity upper limit coefficient, range :0~1
Return value	**Value**	**Description**	
	-	-	
Note			

Routine:

```
moveit：：planning_interface：：MoveGroupInterface arm (" administrator ");
arm.setMaxVelocityScalingFactor(0.2);
```

(3) void moveit：：planning_interface：：MoveGroupInterface：：MoveGroupInterfaceImpl：：setGoalJointTolerance (double tolerance)

Function functions	Set the allowable error of joint motion		
Function parameters	**Type**	**Name of name**	**Description**
	double	tolerance	Allowable error, unit: Rad
Return value	**Value**	**Description**	
	-	-	
Note			

Routine:

```
moveit：：planning_interface：：MoveGroupInterface arm (" administrator ");
arm.setGoalJointTolerance(0.001);
```

(4) void moveit：：planning_interface：：MoveGroupInterface：：MoveGroupInterfaceImpl：：setGoalOrientationTolerance (double tolerance)

Function functions	Allowable error in setting motion attitude

Function parameters	Type	Name of name	Description
	double	tolerance	Allowable error
Return value	**Value**	**Description**	
	-	-	
Note			

Routine:

```
moveit：：planning_interface：：MoveGroupInterface arm (" administrator ");
arm.setGoalOrientationTolerance (0.001);
```

(5) void moveit：：planning_interface：：MoveGroupInterface：：MoveGroupInterfaceImpl：：setGoalPositionTolerance (double tolerance)

Function functions	Allowable error in setting the position of motion		
Function parameters	**Type**	**Name of name**	**Description**
	double	tolerance	Allowable error
Return value	**Value**	**Description**	
	-	-	
Note			

Routine:

```
moveit：：planning_interface：：MoveGroupInterface arm (" administrator ");
arm.setGoalPositionTolerance (0.001);
```

(6) void moveit：：planning_interface：：MoveGroupInterface：：stop (void)

Function functions	Stop all the motion of the robot		
Function parameters	**Type**	**Name of name**	**Description**
	-	-	-
Return value	**Value**	**Description**	
	-	-	
Note			

5.2 Set Target Position

bool moveit：：planning_interface：：MoveGroupInterface：：setJointValueTarget (const std:: vector <double >& group_variable_values)

Function functions	Set the position of each joint		
Function	**Type**	**Name of name**	**Description**

parameters	std : : vector<double >&	group_variable_values	Each joint position value, unit: Rad
Return value	**Value**	**Description**	
	T rue	Set up successfully	
	F alse	Setup failed	
Note			

Routine:

```
moveit: : planning_interface: : MoveGroupInterface arm (" administrator ");

   double targetPose[6]={0.391410,-0.676384,-0.376217,0.01.052834,0.454125};
std: : vector<double>joint_group_positions(6);
joint_group_positions[0]=targetPose[0];
joint_group_positions[1]=targetPose[1];
joint_group_positions[2]=targetPose[2];
joint_group_positions[3]=targetPose[3];
joint_group_positions[4]=targetPose[4];
joint_group_positions[5]=targetPose[5];

arm.setJointValueTarget (joint_group_positions);
arm.move();
```

(2) bool moveit: : planning_interface: : MoveGroupInterface: : setPoseTarget (const geometry_msgs: : Pose &) Annex end_effector_pose,const std : : string &end_effector_link ="")

Function functions	Setting the Target Position of the Robot Terminal		
Function parameters	**Type**	**Name of name**	**Description**
	geometry_msgs ： ： Pose &	end_effector_pose	Target attitude of the terminal
	std： ： string &	end_effector_link	Terminal name, default
Return value	**Value**	**Description**	
	T rue	Set up successfully	
	F alse	Setup failed	
Note			

Routine:

```
moveit： ： planning_interface： ： MoveGroupInterface arm (" administrator ");

   // Setting the target position of the robot terminal
geometry_msgs： ： Pose target_pose;
target_pose.orientation.x =0.70692;
target_pose.orientation.y =0.0;
target_pose.orientation.z =0.0;
target_pose.orientation.w =0.70729;

target_pose.position.x =0.2593;
```

```
target_pose.position.y =0.0636;
target_pose.position.z =0.1787;

  arm.setPoseTarget (target_pose);
arm.move();
```

(3) void moveit：：planning_interface：：MoveGroupInterface：：setRandomTarget()

Function functions	Setting Random Target Position		
Function parameters	**Type**	**Name of name**	**Description**
	-	-	-
Return value	**Value**	**Description**	
	-	-	
Note			

Routine:

```
moveit：：planning_interface：：MoveGroupInterface arm (" administrator ");

// Randomly generate a target location
arm.setRandomTarget();

// Start motion planning and move the arm to the target position
arm.move();
```

(4) bool moveit：：planning_interface：：MoveGroupInterface：：setPositionTarget (double x,double y,double z,const std：：string &end_effector_link ="")

Function functions	Setting the Target Position of the Robot Terminal		
Function parameters	**Type**	**Name of name**	**Description**
	double	x	X coordinates
	double	y	Y coordinates
	double	z	Z coordinates
	std：：string &	end_effector_link	Terminal name, default
Return value	**Value**	**Description**	
	T rue	Set up successfully	
	F alse	Setup failed	
Note			

(5) bool moveit：：planning_interface：：MoveGroupInterface：：setRPYTarget (double roll, double pitch,double yaw,const std：：string &end_effector_link ="")

Function functions	Setting Target Attitude of Robot Terminal

	Type	Name of name	Description
Function parameters	double	roll	X axis rotation value
	double	pitch	Y axis rotation value
	double	yaw	Z axis rotation value
	std：：string &	end_effector_link	Terminal name, default
Return value	**Value**	**Description**	
	T rue	Set up successfully	
	F alse	Setup failed	
Note			

5.3 Cartesian Space Movement

(1) double moveit：：planning_interface：：MoveGroupInterface：：computeCartesianPath (

const std：：vector<geometry_msgs：：Pose >& waypoints,

double eef_step,

double jump_threshold,

moveit_msgs：：RobotTrajectory &trajectory,

bool avoid_collisions =true,

moveit_msgs：：MoveItErrorCodes *error_code =NULL)

Function functions	Set the robot Cartesian path

<table>
<tr><td rowspan="7">Function parameters</td><td>Type</td><td>Name of name</td><td>Description</td></tr>
<tr><td>std : : vector<geometry_msgs : : Pose >&</td><td>waypoints</td><td>Descartes Trail</td></tr>
<tr><td>double</td><td>eef_step</td><td>Step value of the terminal</td></tr>
<tr><td>double</td><td>jump_threshold</td><td>Jump threshold</td></tr>
<tr><td>moveit_msgs : : RobotTrajectory &</td><td>trajectory</td><td>Descartes Paths from Planning</td></tr>
<tr><td>bool</td><td>avoid_collisions</td><td>Permissible collision detection</td></tr>
<tr><td>moveit_msgs : : MoveItErrorCodes *</td><td>error_code</td><td>Error code returned</td></tr>
<tr><td rowspan="2">Return value</td><td>Value</td><td colspan="2">Description</td></tr>
<tr><td>double</td><td colspan="2">Coverage of programmable paths, range :0~1</td></tr>
<tr><td>Note</td><td colspan="3"></td></tr>
</table>

Example:

```
moveit: : planning_interface: : MoveGroupInterface arm (" administrator ");

// Gets the name of the link of the terminal
std: : string end_effector_link =arm.getEndEffectorLink();
```

```
// Get current position data for the starting position of the most robot arm motion
geometry_msgs：：Pose start_pose =arm.getCurrentPose (end_effect_link). pose;

  std：：vector<geometry_msgs：：Pose>waypoints;

// Add the initial pose to the list of road points
  waypoints.push_back (start_pose);

start_pose.position.z -=0.2 per cent;
  waypoints.push_back (start_pose);

start_pose.position.x +=0.1;
  waypoints.push_back (start_pose);

start_pose.position.y +=0.1;
  waypoints.push_back (start_pose);

  // Path planning in Cartesian space
  moveit_msgs：：RobotTrajectory trajectory;
  const double jump_threshold =0.0;
  const double eef_step =0.01;
  double fraction =0.0;
int maxtries =100 per cent/maximum number of attempted planning
int attempts =0// Number of attempts to plan

while (action <1.0&& attempts <maxtries)
  {
fraction =arm.computeCartesianPath (waypoints, eef_step, jump_threshold, trailory);
attempts++;
```

```
if (attems %10==0)
ROS_INFO (" Still drying after% d attempts ...", attempts);
   }

if (fragment ==1)
   {
ROS_INFO (" Path computed successfully.Moving the arm .");

            // Generate motion planning data for robotic arms
            moveit：：planning_interface：：MoveGroupInterface：：Plan plan;
            plan.trajectory_=trajectory;

            // Implementation Campaign
            arm.execute (plan);
sleep(1);
   }
else
   {
ROS_INFO (" Path planning failed with only 0.6f success after% dattempts.", fragment, maxtries);
   }
```

6 One-click firmware upgrade

ROS host computer integrated PROBOT SFU firmware upgrade tool, can provide one-click firmware update service for ROBCELL controller remotely in network environment.

Follow the following steps to upgrade the controller firmware through the host computer.**Note that the following steps take PROBOT G750 as an example, other models of robot arm operation is consistent.**

6.1 Use Notes

⚠ 注意

(1) Please follow this instruction strictly for firmware upgrade, otherwise it may cause problems such as unable to boot, controller can not start.

(2) Do not perform firmware upgrade operations on a controller that is in operation, such as controlling the movement of the robot arm.After the upgrade is complete, the controller will be automatically restarted, which is likely to cause danger!

(3) During the firmware upgrade, make sure that the controller is powered properly and remains connected to the PC network (no power down, continuous network), otherwise it may cause the upgrade process to fail and can not be turned on.

(4) After the upgrade is successful, the controller will restart automatically, which may cause the buzzer to ring all the time. Don't worry! Just power up the controller again.

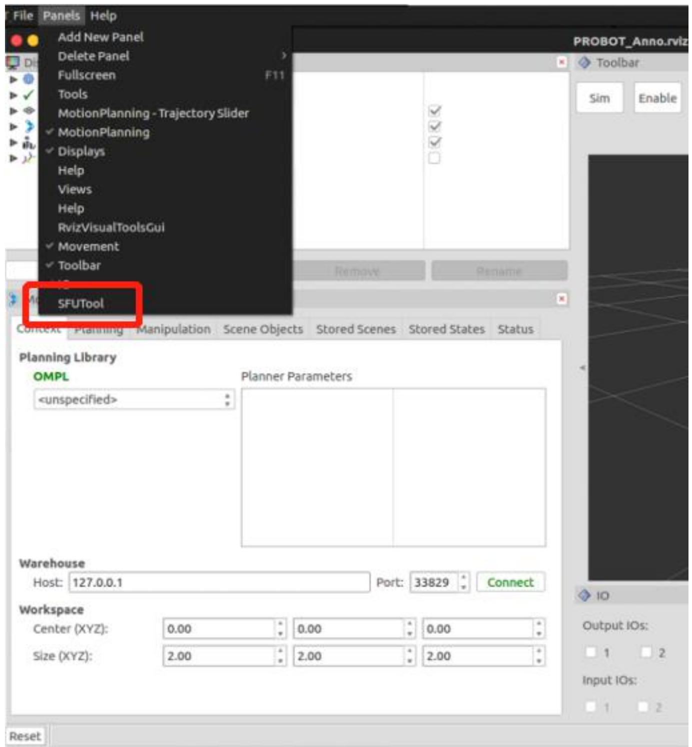
6.2 Prepare for upgrade

start the controller and confirm that the network connection between the PC and the controller is stable.

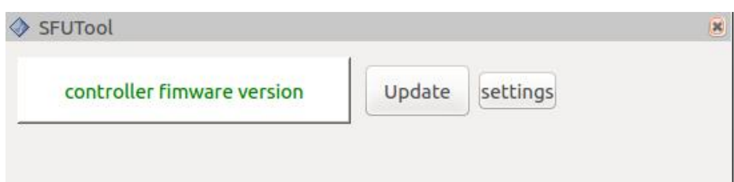
(1) Start the ROS host computer using the following command, add additional parameters at the end sfu：=enable, enable firmware upgrade.

$roslaunch probot_bringup probot_g750_bringup.launch sfu： =enable

(2) Click the host computer Panels option and select SFUTool, to open the firmware upgrade tool panel.



- Figure 61 Open SFUTool on host computer



- Figure 62 Firmware Upgrade Tool Panel

controller firmware version column will display the controller firmware version. Update for one-click firmware upgrade button. Click settings to open the configuration panel.

6.3 One-click upgrade

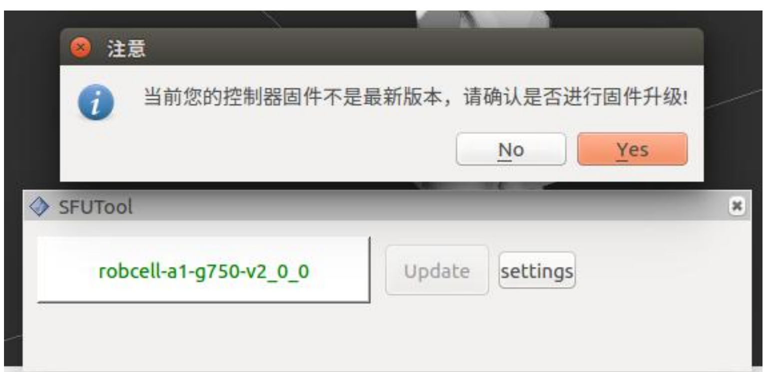
Click the Update button on the SFUTool panel and the system will automatically complete the following actions:

1. Obtain controller firmware version information;

2. According to the version information obtained above, query the latest firmware version to confirm whether the current controller firmware needs to be updated. The version display bar displays the version of the controller's current firmware;

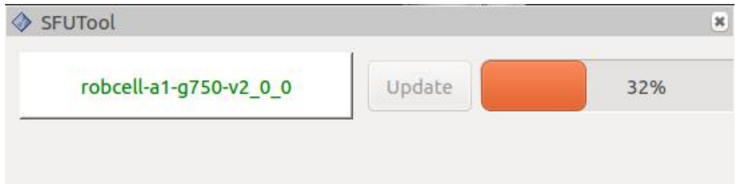
- Figure 63 Query Controller Current Version

3. If you need to update, the following prompt box pops up. Click Yes to confirm the upgrade;



- Figure 64 Firmware Upgrade Confirmation

4. After the firmware upgrade starts, will download the official latest firmware package, after the download completes the firmware update operation automatically;



- Figure 65 Firmware upgrade in progress

5. Firmware upgrade completed. If the upgrade is successful, the version display bar refreshes the firmware version number and automatically restarts the controller. At this time the upper computer and controller connection will be disconnected, please restart the upper computer.

- Figure 66 Firmware Upgrade Successful

7 Blockly graphical programming

ROS upper computer integrated PROBOT Blockly graphical programming tools, in simulation or real machine mode, you can drag and drop code block programming control arm, no manual code writing.

7.1 Enable graphical programming

7.1.1 Single Machine Simulation

The host computer ROS environment can be simulated on the PC without connecting controller and robot.

Activate the host computer and enable Blockly graphical programming using the following commands:

```
$roslaunch probot _bringup probot_g602_bringup.launch sim：=true blockly：=enable
```

Note: Blockly graphical programming function is not turned on by default, you need to specify the starting parameter blockly as enable.

7.1.2 Operation of the real machine

⚠ 注意

Check the hardware connection to power the control box to ensure that the PROBOT G602 hardware and software system starts successfully and the robot arm is in its initial pose before using this function.

Activate the host computer and enable Blockly graphical programming using the following commands:

```
$roslaunch probot _bringup probot_g602_bringup.launch blockly：=enable
```

Note: Blockly graphical programming function is not turned on by default, you need to specify the starting parameter blockly as enable.

The above command will start the rviz host interface by default at the same time, you can specify the start parameter rviz as disable, start will not open the interface:

```
$roslaunch probot _bringup probot_g602_bringup.launch rviz：=disable blockly：=enable
```

7.1.3 Open the programming interface

The above two sections have opened the graphical programming tool while starting the upper computer control, which runs in the background. Can open the browser, URL bar input 127.0.0.1:1234, press enter to access the Blockly programming interface.

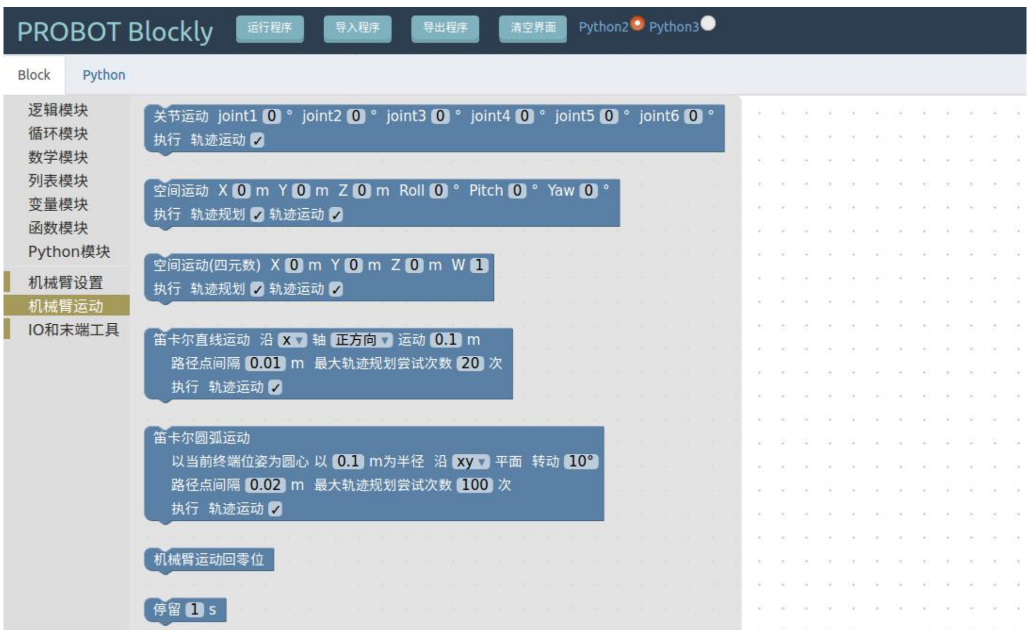
7.2 Introduction to Programming Interface

When the page is opened, you will see the following interface:

programming interface is mainly divided into control bar, programming view area and Python view area.

7.2.1 Programming View

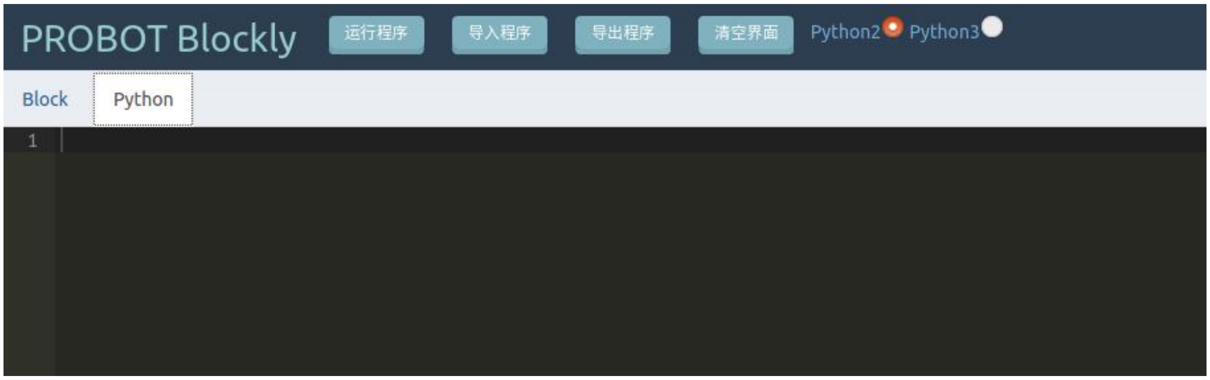
By default, you will enter the programming view area, the left side of which is the Blockly code block menu. If you click on the robot arm motion menu, you will open the Blockly code block related to the robot arm motion control. Mouse left click to select the required code block, drag and drop directly to the right blank area, you can complete the graphical programming of the code block.



- Figure 72 Blockly Graphical programming interface

7.2.2 Python area of view

Click on the interface on the Python will enter the Python view area, can view graphical programming automatically generated Python source code.



- Figure 73 Python View area

7.2.3 Control bar

After programming in the programming view area, the program is processed by clicking the control bar button. These include:

（1） Running program: running Blockly programming automatically generated program to achieve the control of the robot arm and system;

（2） Import program: import Blockly program to the programming view area;

（3） Export program: the Blockly program completed in the programming view area export file, can be saved to the local. This file can be used for importing programs;

（4） Empty interface: empty all Blockly programs in the programming view area.

7.3 Blockly code block

7.3.1 Mechanical arm setup

（1）Mechanical arm initialization. Can set the real machine or simulation control environment, and choose whether to enable robot arm control. In the real machine environment, the robot arm must be controlled, otherwise the program will not be able to control the movement of the robot arm;

（2）Set the motion velocity coefficient (percentage). Used to adjust the speed of all robot motion control, such as joint motion and Cartesian motion;

（3）Set the motion acceleration coefficient (percentage). Used to adjust the joint motion, Cartesian motion and other robot motion control acceleration;

（4）Set the allowable error of joint motion (angle). Allowable error for adjusting joint motion;

（5）sets whether spatial motion allows multiple trajectory planning. Check to allow multiple trajectory planning, that is, after the planning failure, whether to allow another attempt;

（6）Set the target attitude allowable error (angle);

（7）Set the allowable error of the target position of the spatial motion (m);

（8）Mechanical arm running environment unloading. Force the arm to stop all movement, select whether to return to the initial position, and finally turn off the robot arm control function.

7.3.2 Mechanical arm movement

（1）Joint motion (degree). Set each joint angle value of the target position;

（2）Space motion. sets the cartesian coordinate value of the target position xyz(m) and the terminal rotation angle (degree), and selects whether to perform trajectory planning and motion. If you check the track motion, you will certainly carry out track planning;

（3）spatial motion (quaternion). sets the cartesian coordinate value of the target position xyz(m) and the terminal pose (quaternion) and selects whether to perform trajectory planning and motion. If you check the track motion, you will certainly carry out track planning;

（4）Descartes runs straight. A straight line along the positive / negative direction of a coordinate axis (m). Allow the interval between the planned path points in a straight line and the maximum number of trajectory planning;

（5）Cartesian circular arc movement. At the center of the current terminal position, the circle motion is carried out along the plane formed by a certain two coordinate axes with the set radius (m) as the center. allows setting rotation angles, path point intervals, and maximum number of trajectory planning;

（6）The robot arm moves back to zero. Control the position and posture of the robot arm returning to each joint at 0 angle;

（7）For n seconds. The system is dormant for a period of time, and the robot arm remains stationary.

7.3.3 IO and end tools

（1）Setting up the controller digital IO OUTPUT. set output IO level (high/low);

（2）If you read the controller IO INPUT. number reads input IO, performs an action if the level value is high or low;

（3）Read controller digital IO INPUT. The return value is high / low;

（4）High / low level value.

7.3.4 Other code blocks

Besides the robot arm control, the PROBOT Blockly function package also provides code blocks such as program logic, loop, mathematics, list and so on, which makes graphical programming more powerful.

7.4 Blockly programming examples

Blockly official examples of programming are placed under the packages/functions/probot_blockly/blockly_demo folder and can be imported and used in the programming view area.

注意

For graphical programming using PROBOT Blockly, use robot arm initialization as the first code block of the program to initialize the robot arm running and control environment. Otherwise, you will not be able to use other code blocks to control the robot arm (the runtime terminal will report an error and prompt you to use the robot arm settings-the robot arm initialization).

At the end of the program, it is recommended to use the robot arm running environment to uninstall as the last code block of the program, but not mandatory.

7.4.1 Joint motion

jog_joint_demo：

- Figure 74 Blockly Graphical Programming Example - Joint Motion

7.4.2 Space motion

jog_pose_demo：

- Figure 75 Blockly Graphical Programming Example - Cartesian Motion

7.4.3 IO control

io_control_demo：

- Figure 76 Blockly Graphical Programming Example - IO Control

7.5 Frequently Asked Questions

7.5.1 Start Script No Execution Permission

The error is as follows:

```
ERROR: cannot launch node of type [ares_blockly/ares_blockly_backend.py]: can't locate node [ares_blockly_backend.py]in package [ares_blockly]
ERROR: cannot launch node of type [ares_blockly/pywebserver.py]: can't locate node [pywebserver.py]in package [ares_blockly]
```

The problem is that the startup script has no execution rights. Add executable permissions to the following command:

```
$cd packages/functions/probot_blockly/scripts
$sudo chmod +x .-R
```

7.5.2 Start script wrap format error

The error is as follows:

```
process[ares_blockly_backend-2]: started with pid [901]
usr/bin/env: \u2018python3\r\u2019: No such file or directory //
...
process[ares_blockly_webserver-3]: started with pid [904]
usr/bin/env: \u2018python3\r\u2019: No such file or directory //
```

This problem is due to the startup script line wrap format error. Replace the windows line breaks with the unix format by:

```
$cd packages/functions/probot_blockly/scripts
$dos2unix ./*
```

Before using this command, install the dos2unix tool.

7.5.3 Missing dependency packages for python3 versions

The error is as follows:

```
Traceback (most recent call last):
 File "/blockly_ws/src/ares_blockly/scripts/ares_blockly_backend.py",line 35,in <module>
...
 import yaml
ImportError: No module named'yaml'
```

the problem is due to PC lack of Python3 version of the yaml、rospkg and other dependency packages. Execute the following command to install the missing dependency package:

```
$sudo apt-get -y install python3-pip ros-melodic -mavros
$sudo pip3install pyyaml numpy rospkg autobahn
```

8 Common Fault Diagnosis

8.1 Unable to boot

Check that the power input cable is connected and the plug is plugged into the effective power grid. Chinese mainland supply voltage 220 V, China Taiwan, Japan, Europe and the United States and other regions supply voltage 110 V.

Whether the power switch is on (when the power switch is on, the power switch indicator light is on. If the emergency stop switch is in a state of emergency stop, if it is the above reason, you can diagnose and solve it by yourself. If not, please contact our maintenance department in time. Contact email: support@robotanno.com: after sale

8.2 The robotic arm can not return to its initial position

Treatment: if in the operation process, found that the robot arm can not return to the initial position, please refer to the joint space point movement part, using the joint space point movement function to adjust the robot arm to the initial position. Then turn on the power, reset the initial pose, debug several times to see if the problem is ruled out. If this problem occurs many times, please contact our after-sales department in time. Please don't disassemble the machine without being familiar with it.

8.3 The mechanical arm grabs the object

If the robot arm can not grasp the object, it may be caused by the overload of the robot arm or the wear of the parts caused by the long service time of the robot arm. If this problem occurs many times, please contact our maintenance department in time, please do not disassemble the machine without familiarity.

⚠ 注意

If it is not caused by the overweight of the caught object, please find the repairman to deal with it, do not deal with it yourself, otherwise we will not be responsible for any other problems.

9 Appendix

9.1 References

G602 1.Github -PROBOT： https://github.com/ps-micro/PROBOT_

2.MoveIt! Tutorials： http://docs.ros.org/melodic / api/moveit_tutorials/html/index.html

3. Guyue College: https://class.guyuehome.com/

4. Gu Yueju: https://www.guyuehome.com/

5. ROS Robotics Development Practice, edited by Hu Chunxu, Mechanical Industry Press

6. ROBOT SFU Firmware Upgrade Tool User Manual, Fine Front Microcontrol

9.2 Contact information

Email: support@robotanno.com

Website: www.robotanno.com

www .guyueho me .com