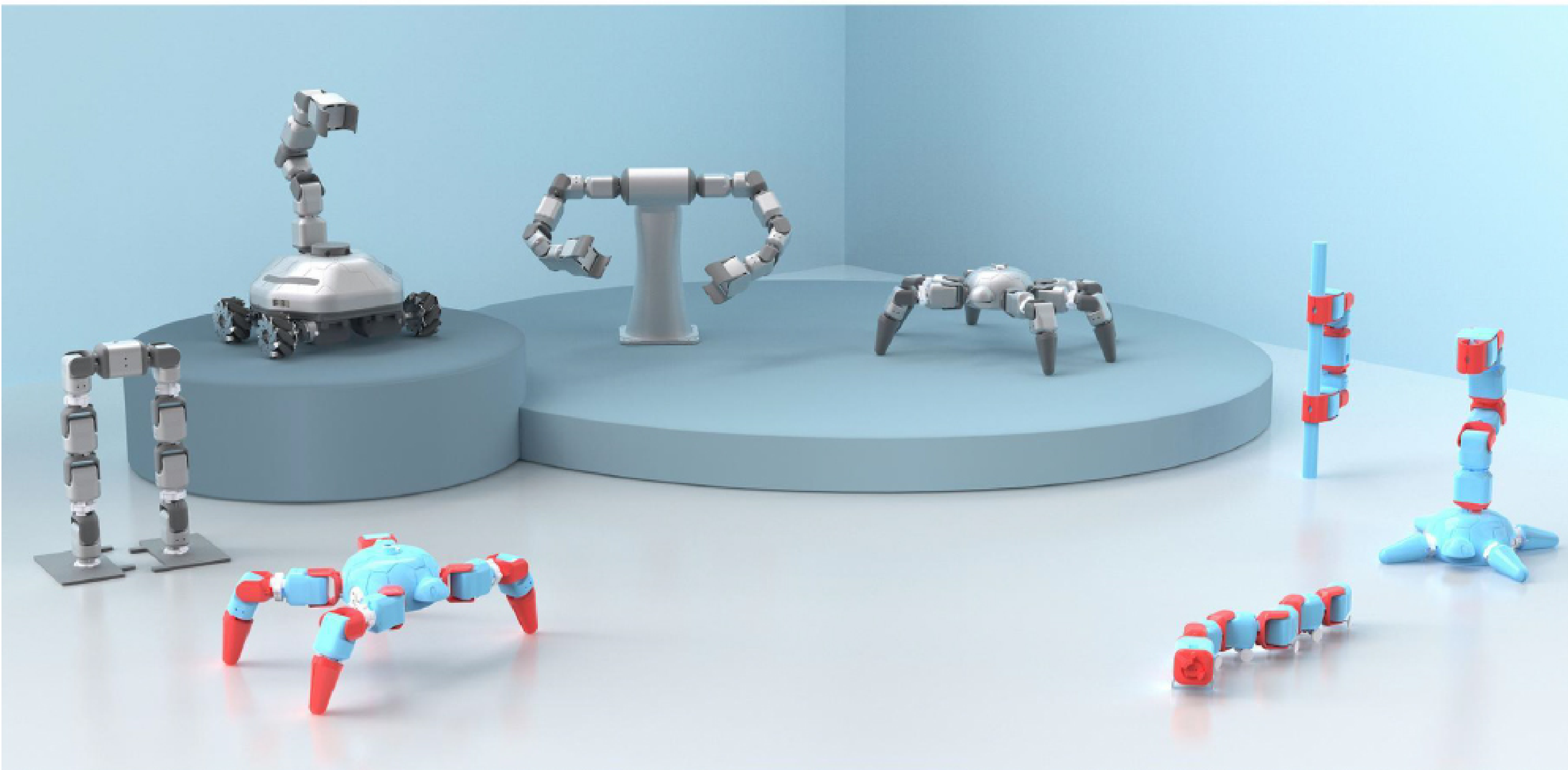
Shenzhen COREBINGO Co.,Ltd



Desktop modular intelligent robot system user manual

Version: V1.0

Tel: +86-755 28450127
Email: cs@corebingo.com
Website: www.corebingo.com
Add: Room A446, Qianjiang Arts Center, No.58, Bulan Road,Longgang District, Shenzhen City, PRC

CONTENTS

修改记录：

Version	Date	Author	Changing Remarks
V1.0	2022-9-9	Biowin Robotics	

Disclaimer

To the maximum extent permitted by law, the products described in this manual (including its hardware, software, firmware, etc.) are provided "as is". Biowin Robot does not provide any form of express or implied warranty for possible defects, errors or malfunctions of the products, including but not limited to the warranties of merchantability, quality satisfaction, fitness for a particular purpose, and non-infringement of the rights of third parties, etc; No compensation will be made for any special, incidental or consequential damages resulting from the use of this manual or the use of our products.

Before using this product, please read this user manual and relevant technical documents published on the Internet in detail and understand the relevant information. Ensure that you fully understand the product and related knowledge on the premise of use. Biowin Robot recommends that you use this manual under the guidance of a professional. All safety information contained in this manual shall not be considered as a safety assurance for this product. Even if this manual and related instructions are followed, hazards or losses caused during use may still occur.

The user of this product has the responsibility to ensure that the applicable laws and regulations of relevant countries are followed and that there is no major danger in the use of the robot.

Foshan Biowin Robotics and Automation Co.,Ltd

Address：A414, Research Institute of Guangdong University of Technology, Shishan Town, Nanhai District, Foshan

Website：www.biowinedu.com

1 Safety Precautions

This chapter introduces the safety precautions that should be paid attention to when using this product. Please read the manual carefully before using it for the first time. This product should be used in the environment in accordance with the requirements. Do not modify the product without authorization, otherwise it may cause product failure, even personal injury, electric shock, fire, etc. The assembly, operation, teaching and programming personnel of the product must read the manual carefully and use the product in strict accordance with the specifications of the operation manual.

1.1 General Safety

△ Danger
The equipment of this product belongs to live equipment, and users are not allowed to disassemble and change the line at will, otherwise it is easy to cause the equipment itself and personal safety damage.

This product is a small robot based on modular design. The following safety rules shall be followed when using it:

1. When operating the robot, the local laws and regulations shall be strictly observed. The safety precautions described in the manual are only a supplement to the local safety regulations.
2. The “danger ”, “warning ” and “precautions ” described in the manual are only supplementary instructions for all safety precautions.
3. Please use the robot within the specified environment. The use of the robot beyond the specification and load will shorten the service life of the robot and even damage the equipment.
4. The operator of the robot must first understand all kinds of safety precautions and

master the correct operation and maintenance methods before operating and maintaining the robot.

5. Highly corrosive cleaning is not applicable to the cleaning of this product, and anodized parts are not applicable to immersion cleaning.
6. Without professional training personnel, it is not allowed to repair the faulty product or disassemble the robot without authorization. If the product is faulty, please contact Biowin Robot in time.
7. If the product is scrapped, please comply with relevant laws to properly dispose of industrial wastes and protect the environment.
8. The product packaging box contains small parts. Please do not let children play to prevent accidental swallowing.
9. Children must be monitored by adults during use. Please turn off the equipment in time when the operation is completed.
10. During the movement of the robot, please do not extend your hand into the movement range of the robot arm, and be careful of bumping and pinching.
11. It is forbidden to change or remove the nameplate, description, icon and mark of the robot and related equipment.
12. Please be careful during handling and installation. Put the robot with care according to the instructions on the packing box and place it correctly according to the arrow direction, otherwise the machine will be easily damaged.
13. Before operating the robot, please refer to the user manual attached with the box.

1.2 Matters Needing Attention

The load of each joint module has certain limits. When the robot suddenly loses power, it is proved that the module may be in the overload state. Please reevaluate the robot configuration.

The control module is a live module and equipped with power supply inside. Please do not disassemble and assembly it to avoid damaging the equipment.

2 Hardware introduction

2.1 Module Type

2.1.1 Control Module

F Module 	Size	58*48*84 mm^3
	Output voltage	DC 7.4V
	Communication mode	WIFI
	IO port	1
	Weight	114 g
P Module 	Size	130*30*68 mm^3
	Output voltage	DC 7.4V
	Communication mode	WIFI
	Module port	7
	IO port	4 (There is a serial port.)
	Weight	446 g
M Module 	Size	260*260*110 mm^3
	LiDAR	1
	Operating system	ROS Melodic 18.04
	Output voltage	DC 7.4V
	Module port	7
	Communication mode	WIFI
	IO port	4（There is a serial port. ）
	Weight	2200 g

Note: The control module is the power supply and communication module. The

M module is equipped with LiDAR and is configured to run ROS operating system.

2.1.2 Joint Module

T module 	Degree of freedom	1
	Size	48*36*81 mm^3
	Locked-rotor torque	16 kg.cm
	Voltage	DC 7.4V
	Weight	100 g
	Module type	Joint module
I Module 	Degree of freedom	1
	Size	57*40*72 mm^3
	Locked-rotor torque	16 kg.cm
	Voltage	DC 7.4V
	Weight	105 g
	Module type	Joint module
G Module 	Degree of freedom	1
	Size	58*57*79 mm^3
	Locked-rotor torque	16 kg.cm
	Voltage	DC 7.4V
	Weight	123 g
	Module type	End module

Note: The rotation axis of the joint of T module is perpendicular to the center line of the joint, and the rotation axis of the joint of I module is parallel to the center line of the joint.

2.1.3 Sensor Module

MINI PRO main control board: Sensor data detection and setting.

Sound sensor: It is used to detect and feedback external sound.

Dot matrix: It displays patterns through 5X7 squares.

Ultrasonic sensor: It is used to detect whether there are obstacles ahead.

Gesture sensor: It is used to recognize gesture movements such as up, down, left, right, front and back.

Line patrol sensor: It is used to detect black and white lines.

RGB lights: It is used to display lights of different colors.

Color sensor: It is used to detect different colors and feedback results.

Infrared remote receiving sensor: It is used to receive button information of the infrared remote control.

2.1.4 Auxiliary Module

Auxiliary mobility wheel: It is used for auxiliary mobility of robots, such as snake-like robots, mounted on T module and F module.

Size: φ17-38 mm

Auxiliary universal wheel: It is used for the auxiliary universal movement of robots, such as the variable wheelpitch two-wheeled robot, mounted on the F module.

Size: φ40-31 mm

I module wrapping piece: It makes I module fixed on the mobile chassis to realize wheeled rotation, and is used to build a wheel mobile platform.

Size: 85-76.5-46.5 mm

Ordinary wheeled module: It is used to build the wheeled mobile platform, and it is combined with I module to form the wheeled mobile module to achieve the mobile function, for example, it is used to build the two-wheeled mobile platform.

Size: φ68-44.5mm

Mecanum wheeled module: It is used to build the wheeled mobile platform, and it is combined with I module to form the wheeled mobile module. It is used to build the four-wheel mobile platform, which can realize the function of transverse and oblique translation.

Size: φ84-39.5 mm

Auxiliary module of universal wheel: It is used to build the mobile platform and realize the auxiliary support movement function, such as the front and rear auxiliary support for the two-wheel moving platform.

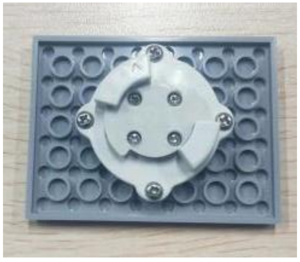
Size: 86-60-55.5 mm

F module fixing seat: It is used to fix F module; For example, when using F module to build a manipulator, the fixed seat of F module should be used as the base to build.

Bionic foot: It is used as a support for the base of P module when building the operating arm of P module.

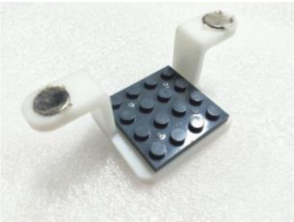
I module camera fastener: It is used to fix the camera to the wider side of the I module.

T module camera fastener: It is used to fix the camera to the end of the T module.



The sensor fixing plate of the P module: It is fixed to the top interface of P module for easy installation of sensors.

The fixing plate of the sensor main control board of the M module: It is used to fix the sensor main control board at the bottom of the M module.



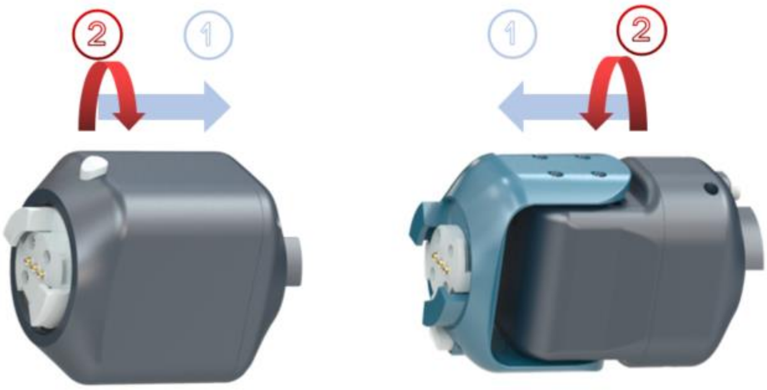
The sensor fixing plate of the M module: It fixes the ultrasonic and patrol sensors in front of the M module.

USB Camera: It is used for the AI example and part of the ROS example.

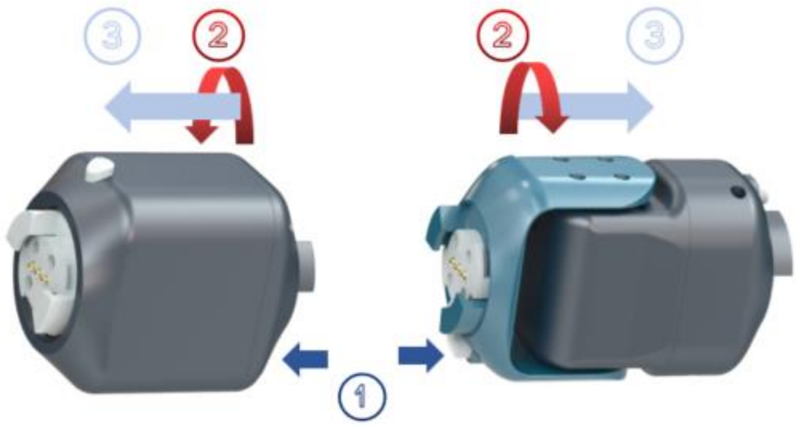
2.2 Module Assembly

The assembly of modules is the beginning of the use of MiniBot. Each module realizes simple and reliable connection between modules through a unique unified mechanical and electrical interface. The user only needs 2 seconds to complete the disassembly and assembly of modules.

- Module Assembly: During assembly, the interfaces of any two modules shall be aligned closely (if there are anti-release buttons at both ends, the buttons shall be staggered for docking), and the connection can be completed by rotating a certain angle.



- Module Disassembly: Press the anti-release buttons of the two modules at the same time (if any) and rotate them by a certain Angle to complete the disassembly.



2.3 Assembly and Disassembly of Sensor Fasteners

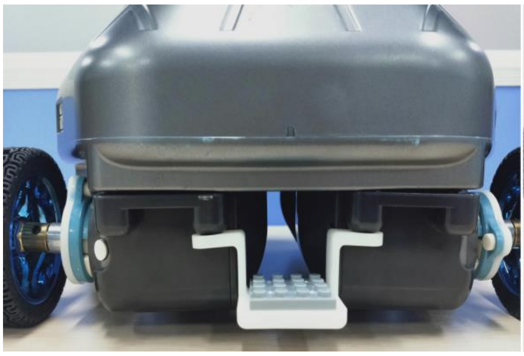
We use Lego connectors to fix and disassemble sensors. Now we introduce the assembly and disassembly of sensor auxiliary fasteners respectively.

The camera can be installed on the I module and the T module respectively. When the camera is installed on the I module, the camera fasteners of I module is used. Firstly, the suction cup bolt is fixed to the camera by rotating, and then the suction cup is fixed on the larger plane of the I module by pressing, as shown in the figure:

The camera is fixed at the end of the T module. First, the camera and the camera fasteners of T module need to be fastened with M6 bolts, and then the camera is fixed at the end of the T module through the unified two-leaf quick interface, as shown in the figure:

If the sensor needs to be installed on the P module, the sensor fixing plate of the P module needs to be fixed to the top interface of the P module through the unified two-leaf quick interface, and then the sensor is fixed to the sensor fixing piece of the P module through Lego accessories, as shown in the figure:

If the sensor needs to be installed on the M module, the fixing plate of the sensor main control board of the M module needs to be fixed at the middle of the bottom of the M module through the unified two-leaf quick interface, the ultrasonic sensor or the line patrol sensor should be fixed at the front of the M module (I module wrapping piece) through the fixing plate of the M module sensor, and then the sensor should be fixed on the fixing piece of the M module sensor through Lego accessories, as shown in the figure:



3 Development Introduction

For some users with Python foundation, the company has built a MiniBot platform under the Python environment, which supports Python compilation and run MiniBot series robots, providing users with secondary development opportunities.

3.1 Purpose

The purpose of this instruction is to fully describe how our MiniBot series robots runs in Python environment, so that users can quickly understand and use the MiniBot robot and the secondary development of the robot.

3.2 Development Environment

linux	Ubuntu 18.04
ROS	Melodic
Window	Window 10
Python	Python 3.0, pip3
External libraries	Numpy and Keyboard

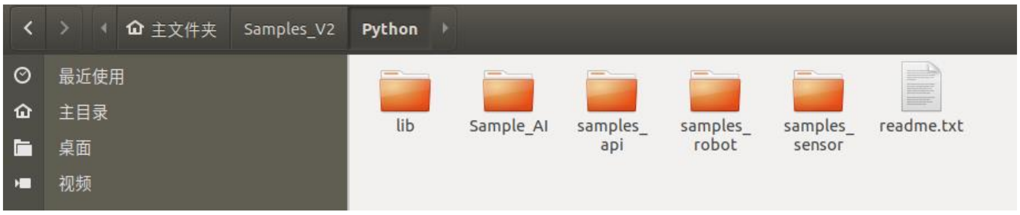
How should we get the code?

Website: https://gitee.com/biowinrobotics/Samples_V2

3.3 Sample Directory Structure

In the content of this article, the development and running environment we prepared is Ubuntu 18.04, and the code package is placed under the main directory.

Enter the "Samples_V2"（The version number changes with the update） folder, and then enter the "Python" directory. We can view the following directory structure.



It can be seen that there are mainly four folders, which encapsulate the dynamic libraries and experimental codes needed for running. The specific contents are as follows:

The “lib” folder stores the dynamic libraries required by the program.

The “samples_api” folder encapsulates API interface programs such as module information and mode switching.

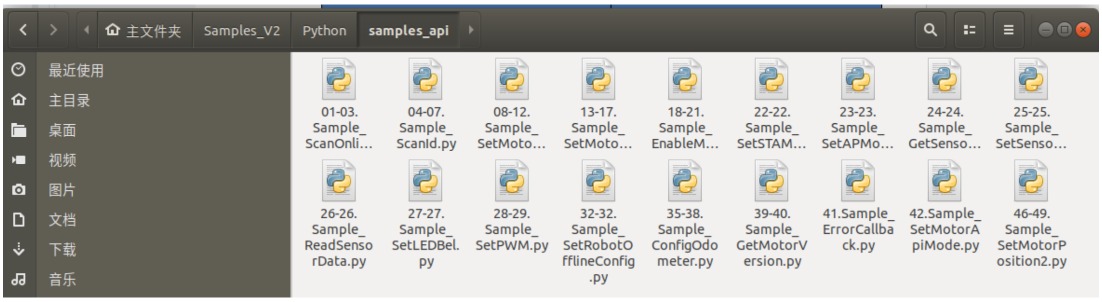
The “samples_robot” folder stores the py files that control the robot motion and demo.

The “samples_sensor” folder stores some py files related to sensors.

The “Sample_AI” folder contains the code for some AI samples.

4 API Basic Example

Enter the “/Home/Samples_V2/Python/samples_api” directory, you can see that we provide users with a lot of py files for api applications.



Functional introduction of each document:

Filename	Function
01-03.Sample_ScanOnlineRobots.py	Scan online devices
04-07.Sample_ScanId.py	Scan and set the module ID and module category
08-12.Sample_SetMotorMode.py	Set the position mode of the joint
13-17.Sample_SetMotorTargetSpeed.py	Set the speed mode of the joint
18-21.Sample_EnableMotor.py	Set joint enablement and obtain steering gear information
22-22.Sample_SetSTAMode.py	Set the current device to STA communication mode
23-23.Sample_SetAPMode.py	Set the current device to AP communication mode
24-24.Sample_GetSensorData.py	Get external device (sensor) data
25-25.Sample_SetSensorData.py	Set external device (sensor) parameters

Filename	Function
26-26.Sample_ReadSensorData.py	Read device data from the data buffer
27-27.Sample_SetLEDBel.py	Set light strip mode
28-29.Sample_SetPWM.py	Set PWM run time
32-32.Sample_SetRobotOfflineConfig.py	Set offline mode
35-38.Sample_ConfigOdometer.py	Set up the odometer
39-40.Sample_GetMotorVersion.py	Read servo and controller version
41.Sample_ErrorCallback.py	Monitor module alarm status
42.Sample_SetMotorApiMode.py	Set the servo to API mode
46-49.Sample_SetMotorPosition2.py	Set the specified position of the servo (float)

These are files for users to view robot joint module information and detect robot information.

4.1 Scanning and Connection of Equipment

（1） Functional Description

Using the computer to connect the robot main control module, and scan the IP addresses of the modules already connected.

（2） Hardware Preparation

1 F module.

(3) Operation Steps

Now take the F module as an example, connect the F module to the T module, turn on the F module switch, and connect the PC to the hotspot corresponding to the F module, which is generally biowinF_xxx. It can be viewed by the robot ID on the silver label paper of the F module.

(4) Example Execution

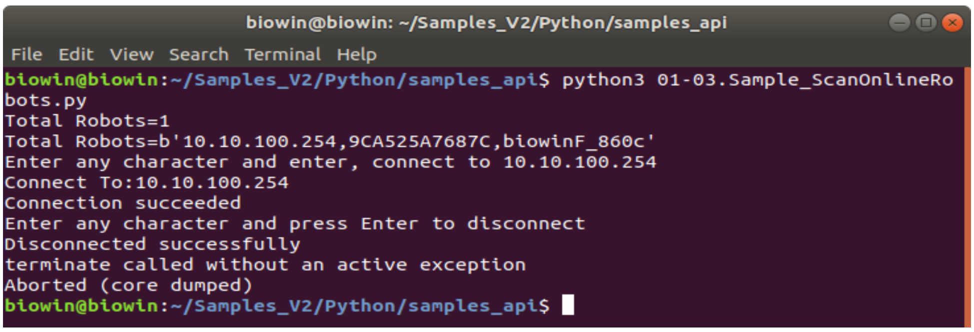
Execute Python file, the file path is:

/Samples_V2/Python/samples_api/01-03.Sample_ScanOnlineRobots.py

Specific operation: Right-click in the samples_api directory to open the terminal, and enter the run command:

$ python3 01-03.Sample_ScanOnlineRobots.py

Operation result:



We can see that a device named biowinF_860c was scanned.

Note: When running a Python command, you need to open the terminal in the directory of the py file you want to execute, and then execute the Python run command.

4.2 The Module ID Scanning

(1) Functional Description

1. Scan the ID number of the joint module connected to the robot main control module;

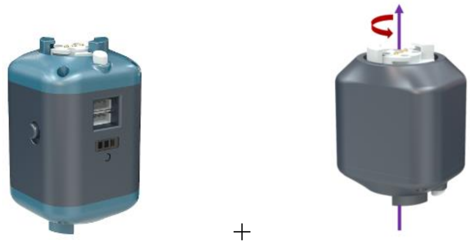
2. Change the ID number of the joint module connected to the robot main control module;

3. Set the joint module as the I module (different types of modules can set different motion angles);

4. Get the module category to which the joint module belongs.

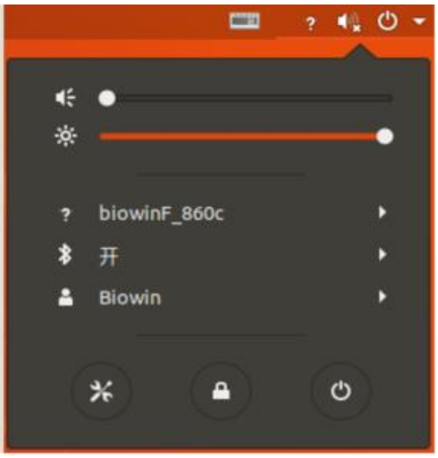
(2) Hardware Preparation

1 F module and 1 I module.



(3) Operation Steps

After connecting the joint module, turn on the power of the F module, and connect the PC to the hotspot of the F module.

(4) Example Execution

Execute Python file, the file path is:

/Samples_V2/Python/samples_api/04-07.Sample_ScanId.py

Specific operation: Right-click in the samples_api directory to open the terminal, and enter the run command:

$ python3 04-07.Sample_ScanId.py

Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_api
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_api$ python3 04-07.Sample_ScanId.py
Connect To:10.10.100.254
Connection succeeded
Total Ids Count=1
[22]
[1]
Enter any character and press Enter to set the ID of module 22 to 23
Set up successfully
Enter any character and press Enter to set the module 23 to G-module
Set up successfully
Enter any character and press Enter to get the category of the module
module 23 is G-module
Enter any character and press Enter to disconnect
Disconnected successfully
terminate called without an active exception
Aborted (core dumped)
biowin@biowin:~/Samples_V2/Python/samples_api$
```

Note: The ID number of the joint module can be changed here, and the ID number is incremented by 1 for each time it is run. For example, if the I module is 22, it will become 23 after running the program. At the same time, if the entire program is run completely, the category and number of the I module remain unchanged. If you need to change the joint ID, it is recommended to use the Bw-Studio Suite software.

4.3 Joint Module Settings

(1) Functional Description

1. Set the joint module mode;

2. Set the position angle of the joint module (position mode);

3. Set the position angle and target running speed of the joint module (position mode);

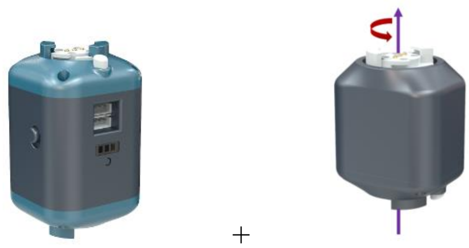
4. Simultaneously set the position of multiple joint modules (position mode);

5. Simultaneously set the position of multiple joint modules with target speed (position mode);

6. Get the current position and angle of the joint module.

(2) Hardware Preparation

1 F module and 2 I modules.



(3) Operation Steps

After connecting the joint module, turn on the power of the F module, and connect the PC to the hotspot of the F module.

（4） Example Execution

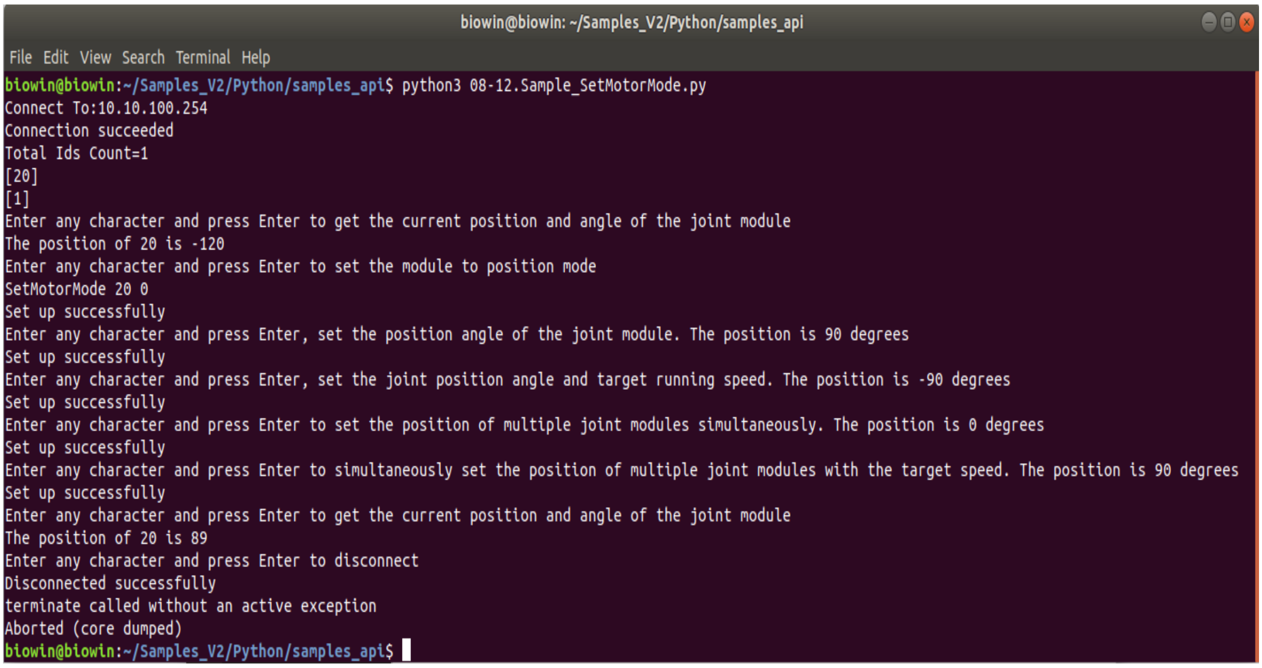
Execute Python file, the file path is:

/Samples_V2/Python/samples_api/08-12.Sample_SetMotorMode.py

Specific operation: Right-click in the samples_api directory to open the terminal, and enter the run command:

$ python3 08-12.Sample_SetMotorMode.py

Operation result:



```
biowin@biowin: ~/Samples_V2/Python/samples_api
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_api$ python3 08-12.Sample_SetMotorMode.py
Connect To:10.10.100.254
Connection succeeded
Total Ids Count=1
[20]
[1]
Enter any character and press Enter to get the current position and angle of the joint module
The position of 20 is -120
Enter any character and press Enter to set the module to position mode
SetMotorMode 20 0
Set up successfully
Enter any character and press Enter, set the position angle of the joint module. The position is 90 degrees
Set up successfully
Enter any character and press Enter, set the joint position angle and target running speed. The position is -90 degrees
Set up successfully
Enter any character and press Enter to set the position of multiple joint modules simultaneously. The position is 0 degrees
Set up successfully
Enter any character and press Enter to simultaneously set the position of multiple joint modules with the target speed. The position is 90 degrees
Set up successfully
Enter any character and press Enter to get the current position and angle of the joint module
The position of 20 is 89
Enter any character and press Enter to disconnect
Disconnected successfully
terminate called without an active exception
Aborted (core dumped)
biowin@biowin:~/Samples_V2/Python/samples_api$
```

Run the program to see the setting process of the joint's position mode.

4.4 Module Speed Settings

（1） Functional Description

1. Set the joint module target speed (speed mode);

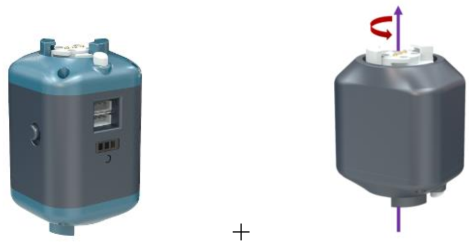
2. Simultaneously set multiple speeds of multiple joint modules (speed mode);

3. Get the module target speed;

4. Get the current speed of the module.

（2） Hardware Preparation

1 F module and 2 I modules.



(3) Operation Steps

After connecting the joint module, turn on the power of the F module, and connect the PC to the hotspot of the F module.

(4) Example Execution

Execute Python file, the file path is:

/Samples_V2/Python/samples_api/13-17.Sample_SetMotorTargetSpeed.py

Specific operation: Right-click in the samples_api directory to open the terminal, and enter the run command:

```
$    python3 13-17.Sample_SetMotorTargetSpeed.py
```

Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_api
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_api$ python3 13-17.Sample_SetMotorTargetSpeed.py
Connect To:10.10.100.254
Connection succeeded
Total Ids Count=1
[20]
[1]
Enter any character and press Enter to set the module to speed mode [please pay attention to selecting I module, other modules may cause damage]
SetMotorMode 20 1
Set up successfully
Enter any character and press Enter to set the joint module target speed to 50%
Set up successfully
Enter any character and press Enter, and set multiple speeds to 85% synchronously
Set up successfully
Enter any character and press Enter to get the target speed of the module
The speed of 20 is 85
Enter any character and press Enter to get the current speed of the module
The speed of 20 is 85
Enter any character and press Enter to get the current speed steps of the module
The speed steps of 20 is 2900
Enter any character and press Enter, the synchronization setting stops
Set up successfully
Enter any character and press Enter to disconnect
Disconnected successfully
terminate called without an active exception
Aborted (core dumped)
biowin@biowin:~/Samples_V2/Python/samples_api$
```

Run the program to see the setting process of speed mode.

Note: When performing this experiment, it is best to use the I module, if using the T and G modules may damage them.

4.5 Setting of whether the Joint Module Is Enabled

(1) Functional Description

1. Set whether the joint module is enabled;

2. Set whether the joint module is locked;

3.Set the servo parameters of the module (the example is setting the target position parameters);

4. Get the servo parameters of the module (the example is getting the target position parameters).

(2) Hardware Preparation

1 F module and 1 T module.

+

(3) Operation Steps

After connecting the joint module, turn on the power of the F module, and connect the PC to the hotspot of the F module.

(4) Example Execution

Execute Python file, the file path is:

/Samples_V2/Python/samples_api/18-21.Sample_EnableMotor.py

Specific operation: Right-click in the samples_api directory to open the terminal, and enter the run command:

$ python3 18-21.Sample_EnableMotor.py

Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_api
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_api$ python3 18-21.Sample_EnableMotor.py
Connect To:10.10.100.254
Connection succeeded
Total Ids Count=1
[33]
[1]
Enter any character and press Enter, set the joint module enable
Set up successfully
Enter any character and press Enter to set the joint module disable
Set up successfully
Enter any character and press Enter to set the joint module to unlock
Set up successfully
Enter any character and press Enter to set the module to position mode
SetMotorMode 33 0
Set up successfully
Enter any character and press Enter, set the servo parameters of module 33's target position
Set up successfully
Enter any character and press Enter, get the servo parameters of module 33's target position
The current position of 33 is 2048
Enter any character and press Enter to disconnect
Disconnected successfully
terminate called without an active exception
Aborted (core dumped)
biowin@biowin:~/Samples_V2/Python/samples_api$
```

After running the program, when enabled, the T module cannot be broken. When the module is disabled, the T module can be moved.

Note: It is best to use the T module for this experiment, as it is easier to see if the module is enabled or not.

4.6 STA Mode and AP Mode Settings

(1) Functional Description

Set the connection mode of the current device to STA mode.

(2) Hardware Preparation

1 F module and 1 T module.

+

(3) Operation Steps

After connecting the joint module, turn on the power of the F module, and connect the PC to the hotspot of the F module.

(4) Example Execution

Here to add a little knowledge: If you want to use a PC to connect and control the F, M, and P modules, there are two methods. One is to use the PC to directly connect the hotspot of the main control module (F, P, M), which is called AP mode. The other is to make the main control module and the PC in the common network mode, that is, when they belong to the same subnet, the effect can also be achieved. This method is called STA mode.

Before running the program, the code needs to be modified so that the user's own F module is connected to the hotspot. The modification is as follows:

Modifying the 34-35 lines of the py file to the network name and password you need to connect to.

```
31 #************************
32 #   2.1、Set the connection mode of the current device to STA mode
33 #************************#
34 ssid = "biowin_2.4G"
35 password = "12345678"
36 input("Enter any character and press Enter, connect the device to the hotspot of "+str(ssid)
37 r =  mlib.SetSTAMode(ssid.encode(),password.encode());
38 if(r):
39   print("Set up successfully")
40 else:
41   print("Set up failed")
```

After modification, save it, and then run the program. At this time, you can see that the power light of the F module is flashing, indicating that the setting is successful. At this time, it can also be found in the PC that the hotspot of the F module originally connected has disappeared, and the PC is also connected to the same network and can control the F module.

Execute Python file, the file path is:

/Samples_V2/Python/samples_api/22-22.Sample_SetSTAMode.py

Specific operation: Right-click in the samples_api directory to open the terminal, and enter the run command:

$　python3 22-22.Sample_SetSTAMode.py

Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_api
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_api$ python3 22-22.Sample_SetSTAMode.py
Connect To:10.10.100.254
Connection succeeded
Enter any character and press Enter, connect the device to the hotspot of biowin_2.4G, the password used is 12345678
Set up successfully
Enter any character and press Enter to disconnect
Disconnected successfully
terminate called without an active exception
Aborted (core dumped)
biowin@biowin:~/Samples_V2/Python/samples_api$ 
```

So what if we want to switch from STA mode back to AP mode? We provide a simple and quick method, just press and hold the button next to the power light of the F module until you see the power light flashing.

4.7 Reading of Ultrasonic Sensor Data

(1) Functional Description

Read data from external devices (take the ultrasonic sensor as an example, refer to API document 2.25 for details).

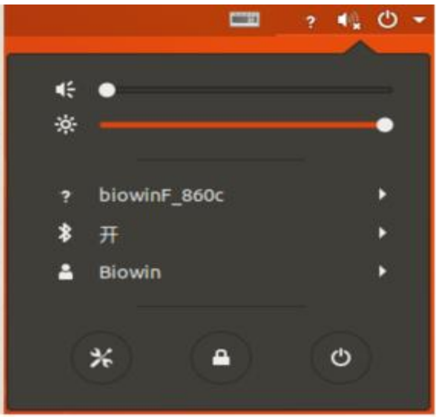
(2) Hardware Preparation

1 F module, 1 ultrasonic sensor module, 1 main control board, 1 4Pin sensor connection line and 1 sensor ordinary connection line.



(3) Operation Steps

Connect the main control board to the F module. For the specific connection method, please refer to Chapter 5 Sensor Control Example. Among them, the ultrasonic sensor is connected to the D3 port of the main control board. After connecting the wires as shown in the figure below, turn on the power of the F module and connect the PC to the hotspot of the F module.



(4) Example Execution

Execute Python file, the file path is:

/Samples_V2/Python/samples_api/24-24.Sample_GetSensorData.py

Specific operation: Right-click in the samples_api directory to open the terminal, and enter the run command:

$ python3 24-24.Sample_GetSensorData.py

Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_api
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_api$ python3 24-24.Sample_GetSensorData.py
Connect To:10.10.100.254
connection succeeded
rs=477
Enter any character and press Enter to disconnect
Disconnected successfully
terminate called without an active exception
Aborted (core dumped)
biowin@biowin:~/Samples_V2/Python/samples_api$
```

After running the program, place your hand or other object in front of the ultrasonic sensor, and the ultrasonic will return the readings.

4.8 Dot Matrix Module Settings

(1) Functional Description

Get data from external devices (take the dot matrix module as an example, refer to API document 2.26 for details).

(2) Hardware Preparation

1 F module, 1 dot matrix module, 1 main control board, 1 4Pin sensor connection line and 1 sensor ordinary connection line.



(3) Operation Steps

Connect the main control board to the F module. For the specific connection method, please refer to Chapter 5 Sensor Control Example. Among them, the dot matrix is connected to the D4 port of the main control board. After connecting the wires as shown in the figure below, turn on the power of the F module and connect the PC to the hotspot of the F module.

（4） Example Execution

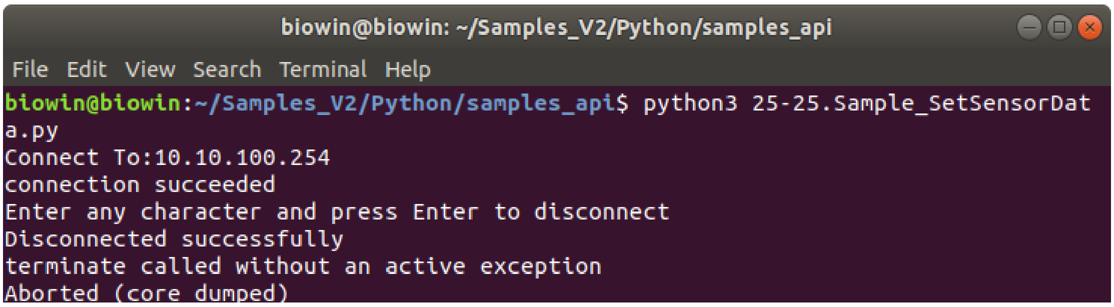
Execute Python file, the file path is:

/Samples_V2/Python/samples_api/25-25.Sample_SetSensorData.py

Specific operation: Right-click in the samples_api directory to open the terminal, and enter the run command:

$ python3 25-25.Sample_SetSensorData.py

Operation result:



Running the code, the dot matrix will display the numbers 0-99.

4.9 Reading of Remote Sensor Data

（1） Functional Description

Read the device data from the data buffer (take the ultrasonic sensor as an example, refer to API document 2.27 for details).

（2） Hardware Preparation

1 F module, 1 main control board, 1 remote control receiving module and 1

remote control.

(3) Operation Steps

Connect the main control board to the F module. For the specific connection method, please refer to Chapter 5 Sensor Control Example. Among them, the remote control receiving module is connected to the D3 port of the main control board. After connecting the wires as shown in the figure below, turn on the power of the F module and connect the PC to the hotspot of the F module.

(4) Example Execution

Execute Python file, the file path is:

/Samples_V2/Python/samples_api/26-26.Sample_ReadSensorData.py

Specific operation: Right-click in the samples_api directory to open the terminal, and enter the run command:

$ python3 26-26.Sample_ReadSensorData.py

Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_api
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_api$ python3 26-26.Sample_ReadSensorData.py
Connect To:10.10.100.254
connection succeeded
Please press the 1~9 keys on the remote control
rs=16
Enter any character and press Enter to disconnect
Disconnected successfully
terminate called without an active exception
Aborted (core dumped)
biowin@biowin:~/Samples_V2/Python/samples_api$
```

Due to the coding system problem, the relationship between input and output is as follows:

Input	Output
1	10
2	11
3	12
4	13
5	14
6	15
7	16
8	17
9	18

4.10 Light Belt Mode Settings

(1) Functional Description

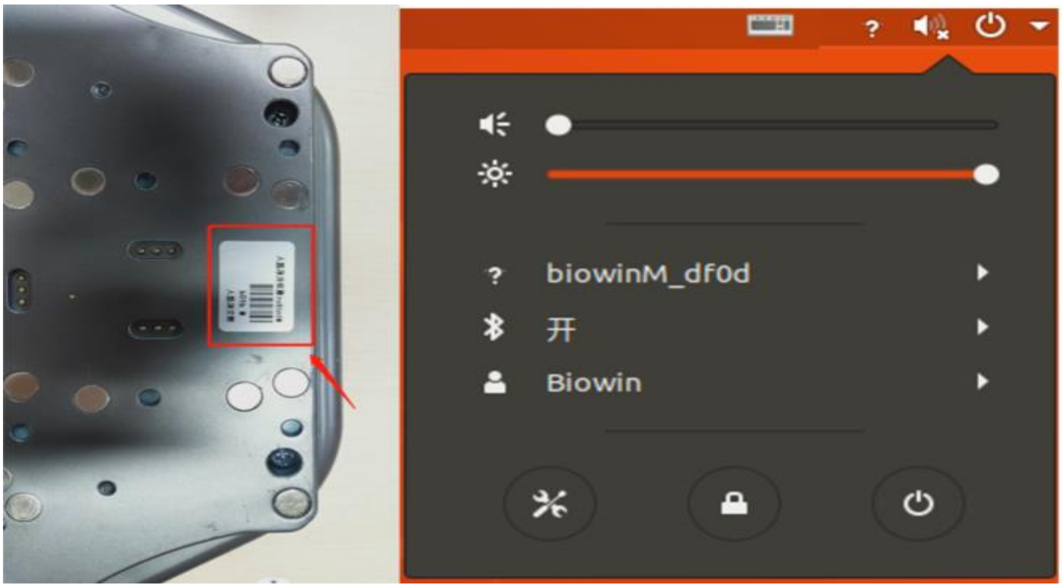
Set two light belts of M module.

(2) Hardware Preparation

1 M module.

（2）Operation Steps

Turn on the power of the M module, and connect the PC to the hotspot of the M module, which is generally biowinM_xxx. For details, please view the silver-white label of the M module.



（3）Example Execution

Execute Python file, the file path is:

/Samples_V2/Python/samples_api/27-27.Sample_SetLEDBel.py

Specific operation: Right-click in the samples_api directory to open the terminal, and enter the run command:

$ python3 27-27.Sample_SetLEDBel.py

Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_api
File  Edit  View  Search  Terminal  Help
biowin@biowin:~/Samples_V2/Python/samples_api$ python3 27-27.Sample_SetLEDBel.py
Connect To:10.10.100.254
Connection succeeded
Enter any character and press Enter, set the device to 0x01: turn left light
setup successfully
Enter any character and press Enter to disconnect
Disconnected successfully
terminate called without an active exception
Aborted (core dumped)
biowin@biowin:~/Samples_V2/Python/samples_api$
```

After running the program, and you can see that the light on the right side of the M module lights up yellow and flashes.

4.11 PWM Time Settings

Before running the program, here is a brief introduction to what PWM is.The full name of PWM is Pulse Width Modulation, which is a method of digitally encoding the level of an analog signal. Through the use of high-resolution counters, the duty cycle of the square wave is modulated to encode the level of a specific analog signal. The PWM signal is still digital because at any given moment, the full-scale DC supply is either fully present (ON) or completely absent (OFF). The voltage or current source is applied to the analog load in a repetitive pulse train of ON or OFF. Simply put, the high bit is valid (1) and the low bit is invalid (0). The time set here is the electrical pulse cycle time.

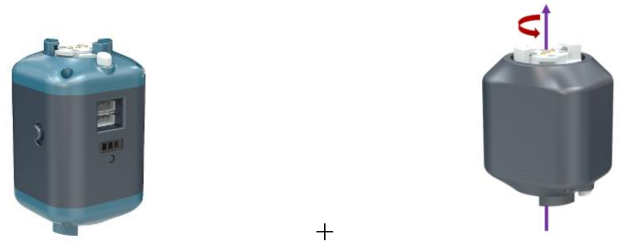
（1） Functional Description

1. Set PWM (run time);

2. Get PWM (run time).

（2） Hardware Preparation

1 T module and 1 I module numbered 21.



(3) Operation Steps

Connect the F module and I module, turn on the power of the F module, and connect the PC to the hotspot of the F module.

(4) Example Execution

Execute Python file, the file path is:

/Samples_V2/Python/samples_api/28-29.Sample_SetPWM.py

Specific operation: Right-click in the samples_api directory to open the terminal, and enter the run command:

$ python3 28-29.Sample_SetPWM.py

Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_api
File  Edit  View  Search  Terminal  Help
biowin@biowin:~/Samples_V2/Python/samples_api$ python3 28-29.Sample_SetPWM.py
Connect To:10.10.100.254
Connection succeeded
Enter any character and press Enter, Set the module 21 to open loop mode
SetMotorMode 21 2
setup successfully
Enter any character and press Enter to set the PWM of the module 21 to 30
setup successfully
Enter any character and press Enter to get the PWM of the module 21
The PWM value of the module 21 is 30
Enter any character and press Enter to set the PWM of the module 21 to 0
setup successfully
Enter any character and press Enter to disconnect
Disconnected successfully
terminate called without an active exception
Aborted (core dumped)
biowin@biowin:~/Samples_V2/Python/samples_api$
```

Note: In the experiment, please select the module numbered 21, otherwise the effect will not be achieved.

4.12 Offline Status Settings

(1) Functional Description

1. Set the offline operation information;

2. Set whether to enable offline operation mode.

(2) Hardware Preparation

1 F module, 1 I module numbered 20, 1T module numbered 36 and 1 G module numbered 41.

(3) Operation Steps

Connect each module, as shown in the following figure:

(4) Example Execution

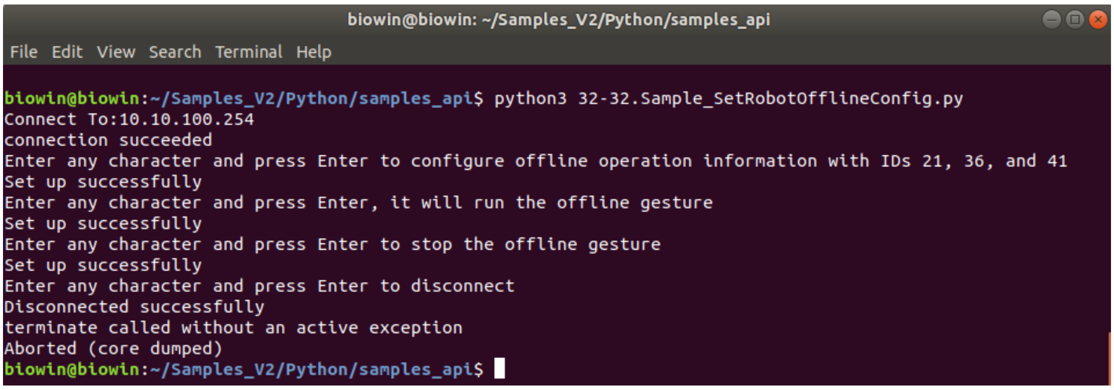
Execute Python file, the file path is:

/Samples_V2/Python/samples_api/32-32.Sample_SetRobotOfflineConfig.py

Specific operation: Right-click in the samples_api directory to open the terminal, and enter the run command:

$ python3 32-32.Sample_SetRobotOfflineConfig.py

Operation result:



Run the program, here you can see that this robot arm is doing offline motion, you can continue to press Enter to stop it.

4.13 Enable and Disable Odometer

(1) Functional Description

1. Configure the odometer;

2. Reset odometer data;

3. Enable or disable odometer reporting;

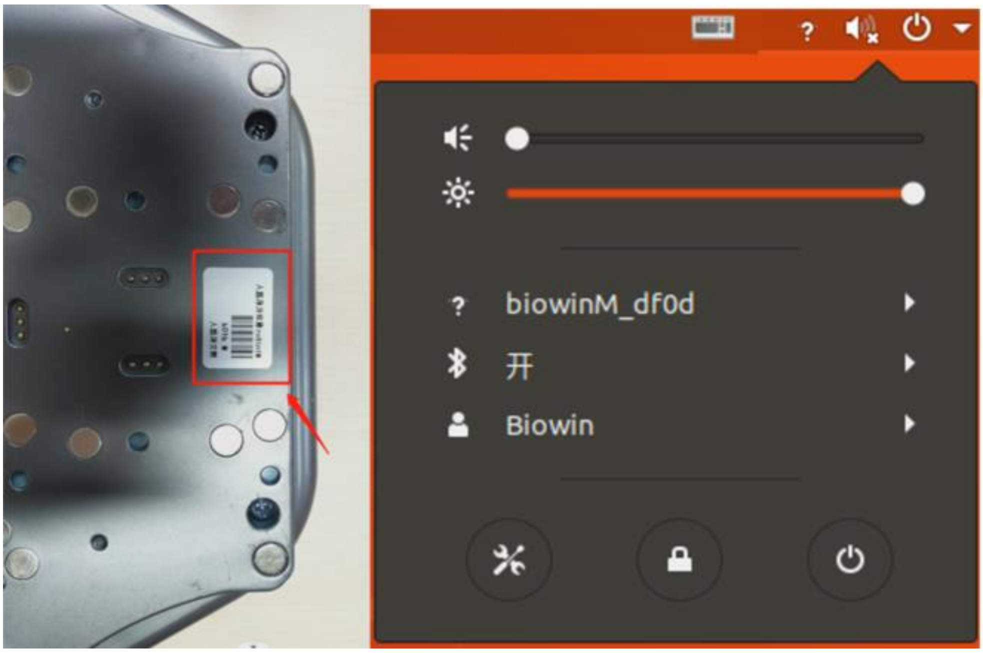
4. Read the reported odometer information.

（2）Hardware Preparation

1 four-wheel car with rubber wheels.

（3）Operation Steps

Turn on the power of the car, and connect the PC to the hotspot of the M module.



(4) Example Execution

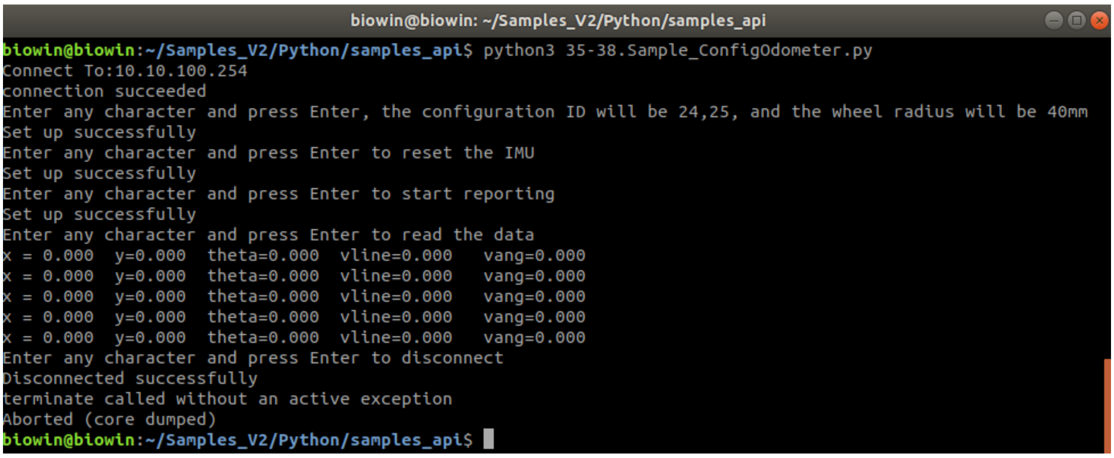
Execute Python file, the file path is:

/Samples_V2/Python/samples_api/35-38.Sample_ConfigOdometer.py

Specific operation: Right-click in the samples_api directory to open the terminal, and enter the run command:

$ python3 35-38.Sample_ConfigOdometer.py

Operation result:



Note: The car must have wheels numbered 24 and 25, one on the left and one on the right.

4.14 Viewing of Servo and Controller Version

（1）Functional Description

1. Read the version number of the servo;

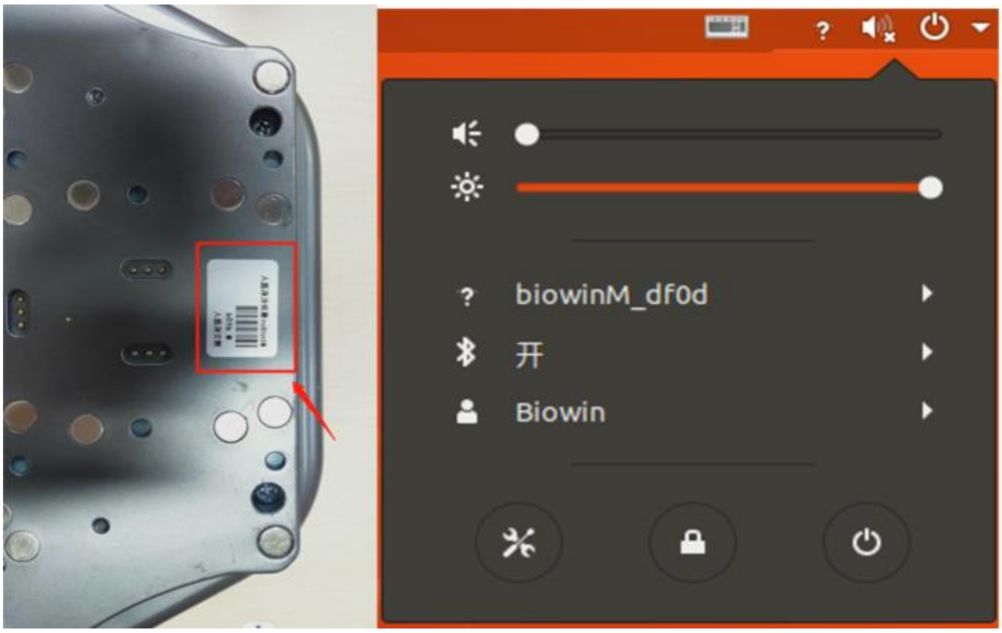
2. Read the controller version number.

（2）Hardware Preparation

1 four-wheel car with rubber wheels.

（3）Operation Steps

Turn on the power of the car, and connect the PC to the hotspot of the M module.



（4） Example Execution

Execute Python file, the file path is:

/Samples_V2/Python/samples_api/39-40.Sample_GetMotorVersion.py

Specific operation: Right-click in the samples_api directory to open the terminal, and enter the run command:

$ python3 39-40.Sample_GetMotorVersion.py

Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_api
File  Edit  View  Search  Terminal  Help
biowin@biowin:~/Samples_V2/Python/samples_api$ python3 39-40.Sample_GetMotorVersion.py
Connect To:10.10.100.254
connection succeeded
Enter any character and press Enter, the firmware version number and servo version number with ID 22 will be obtained
Firmware version number:3.8
Servo version number:9.3
Enter any character and press Enter to get the hardware version, software version and serial number of the controller
hardware version:1.6.1
Software version:1.1.8
serial number:0.0.0
Enter any character and press Enter to disconnect
Disconnected successfully
terminate called without an active exception
Aborted (core dumped)
biowin@biowin:~/Samples_V2/Python/samples_api$
```

Run the program, you can see that the version and controller information of one I module of the car are scanned.

Note: It is also possible to test with other modules. You can only view one joint module at a time when running the code.

4.15 Monitoring Module Alarm Status

（1）Functional Description

1. Set the alarm callback method, and execute this function when a fault occurs;

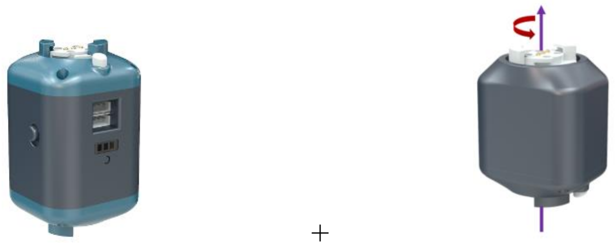
2. Modify the maximum input voltage threshold of the servo to trigger the overvoltage alarm;

3. Read back the current servo position and monitor servo error information;

4. Restore the highest input voltage threshold of the servo.

（2）Hardware Preparation

1 F module and 1 I module numbered 22.



（3）Operation Steps

Connect the F module and I module, turn on the power of the F module, and connect the PC to the hotspot of the F module.

（4）Example Execution

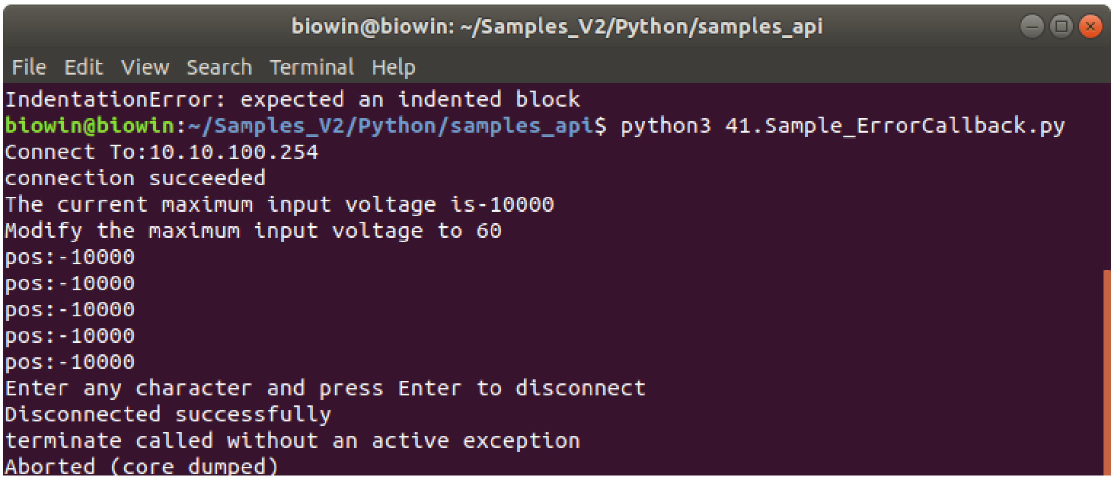
Execute Python file, the file path is:

/Samples_V2/Python/samples_api/41.Sample_ErrorCallback.py

Specific operation: Right-click in the samples_api directory to open the terminal, and enter the run command:

$ python3 41.Sample_ErrorCallback.py

Operation result:



Run the program, you can see that after modifying the maximum input voltage threshold of the I module, the current position will be read back and the current fault information will also be reported.

4.16 Setting the Servo to API Mode

(1) Functional Description

Set the servo to API mode (here divided into high and low configuration servo, scientific research version is high configuration servo).

(2) Hardware Preparation

1 F module and 1 I module.

+

(3) Operation Steps

Connect the F module and I module, turn on the power of the F module, and connect the PC to the hotspot of the F module.

(4) Example Execution

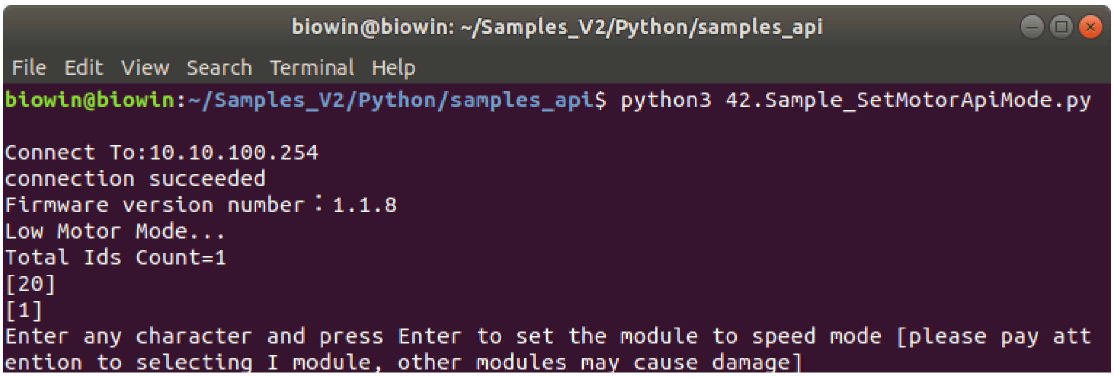
Execute Python file, the file path is:

/Samples_V2/Python/samples_api/42.Sample_SetMotorApiMode.py

Specific operation: Right-click in the samples_api directory to open the terminal, and enter the run command:

$ python3 42.Sample_SetMotorApiMode.py

Operation result:



Run the program, and change the API control mode of the servo to the high-end version of the servo control mode (default high-end servo).

4.17 Setting of Specified Position (Floating Point Type) of Servo

(1) Functional Description

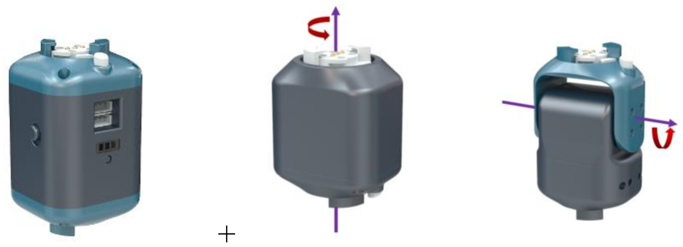
1. Set the position angle of the joint module (position mode - floating point type);

2. Set the position angle and target running speed of the joint module (position mode - floating point type);

3. Simultaneously set the position of multiple joint modules(position mode - floating point type).

(2) Hardware Preparation

1 F module, 1 I module and 1 T module.



(3) Operation Steps

Connect the F module and I module, turn on the power of the F module, and connect the PC to the hotspot of the F module.

(4) Example Execution

Execute Python file, the file path is:

/Samples_V2/Python/samples_api/46-49.Sample_SetMotorPosition2.py

Specific operation: Right-click in the samples_api directory to open the terminal, and enter the run command:

$ python3 46-49.Sample_SetMotorPosition2.py

Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_api
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_api$ python3 46-49.Sample_SetMotorPosition2.py
Connect To:10.10.100.254
connection succeeded
Total Ids Count=2
[20, 30]
[1, 1]
Enter any character and press Enter to get the current position and angle of the joint module
20position26
30position80
Enter any character and press Enter to set the module to position mode
SetMotorMode 20 0
20Set up successfully
SetMotorMode 30 0
30Set up successfully
Enter any character and press Enter, set the position angle of the joint module to 80.5 degrees
20Set up successfully
30Set up successfully
Enter any character and press Enter to set the joint position angle and target running speed. The pos
ition is -80.5 degrees
20Set up successfully
30Set up successfully
Enter any character and press Enter, and set the position of multiple joint modules simultaneously. T
he position is 10.5 degrees
Set up successfully
Enter any character and press Enter, and set the position of multiple joint modules simultaneously wi
th the target speed. The position is 80.5 degrees
Set up successfully
Enter any character and press Enter to get the current position and angle of the joint module
20position26
30position79
Enter any character and press Enter to disconnect
Disconnected successfully
terminate called without an active exception
Aborted (core dumped)
```

Run the program and set the servo to a floating-point position angle.

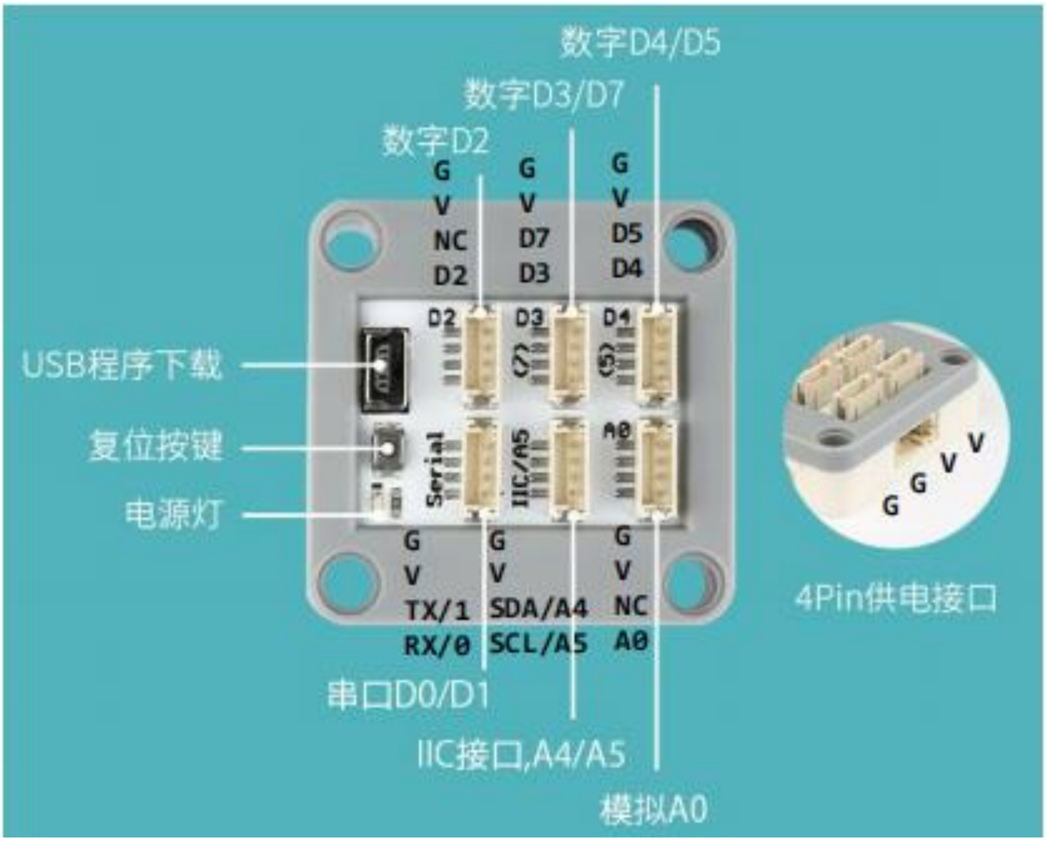
5 Sensor Control Example

Enter the “/Home/Samples_V2/Python/samples_sensor/” directory, you can see that we provide users with a lot of py files for sensor applications.

The following is a brief description of these files:

Filename	Sensor	Function
01.Sample_RgbSensor.py	RGB light (full color LED module)	The RBG lights turn from red to green to blue in turn, then go out, and then cycle 1 more time.
02.Sample_VoiceSensor.py	Sound sensor and RGB light	If the sound sensor detects sound, the RGB light will be red, otherwise it will always be blue.
03.Sample_MatrixSensor.py	Dot matrix module	Show numbers.
04.Sample_GestureSensor.py	Gesture sensor and RGB light	When the gesture sensor detects the corresponding gesture, the RGB lights display different colors.
05.Sample_ColorSensor.py	Color sensor and RGB light	RGB lights identify colors and light up the corresponding colors.
06.Sample_TrackingSensor.py	Tracing module and dot matrix module	Line patrol sensor detects and displays different expressions according to the detection results.
07.Sample_RemoteSensor.py	Remote sensor and dot matrix module	Press the corresponding button on the remote control sensor, and the dot matrix displays the corresponding pattern.
08.Sample_UltrasonicSensor.py	Ultrasonic sensor and dot matrix module	The ultrasonic sensor measures different distances, and the dot matrix displays different patterns.

In the "samples_sensor" directory, sensor-related programs are mainly stored, and users can quickly master the use of various sensors through them. In the sensor experiment, each sensor has a corresponding interface, and the effect can only be achieved if it is successfully connected to the corresponding interface. The following is the structure diagram of the MINI PRO main control board.



The connection table between the sensor category and the main control board interface:

Category	Interface
Digital sensor	D2
Ultrasonic sensor	D3
Dot matrix	D4
RGB light (full color LED module)	D2
Infrared remote control module	D3
Gesture sensor	IIC/A5
Sound sensor	D3

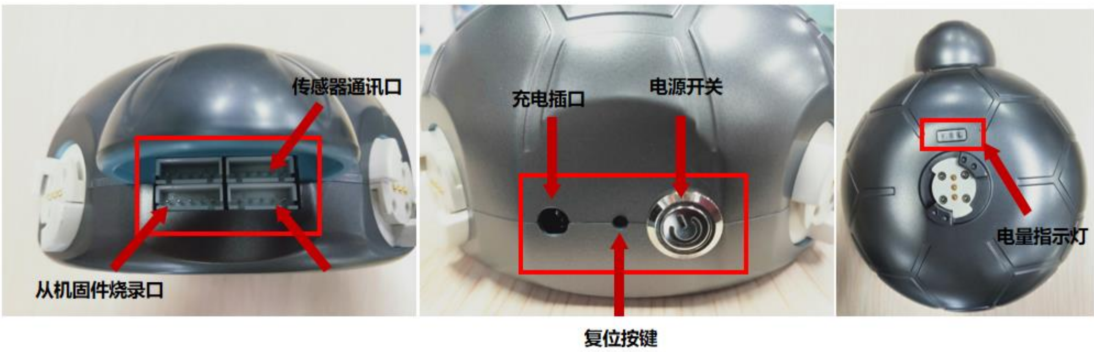
Category	Interface
Tracking module	D3
Main control module F, M, P	D0/D1

The sensor can only be used when it is connected with the main control module F, P and M modules. Here we introduce the wiring positions of these three modules.

F module:

Interface Name	Function
Sensor communication port	Connect the 4Pin cable to the robot end.
Firmware burning port	Update the firmware program to use.
Power indicator	When the power light shows three bars of power, it is full power, and when there is no power, the power light flashes.
Reset button	Press and hold the reset button for 3 seconds, the power light flashes when the reset is successful.
Charging port	Connect the charger adapter cable to charge.
Power switch	Push up for "on", push down for "off".

P module:



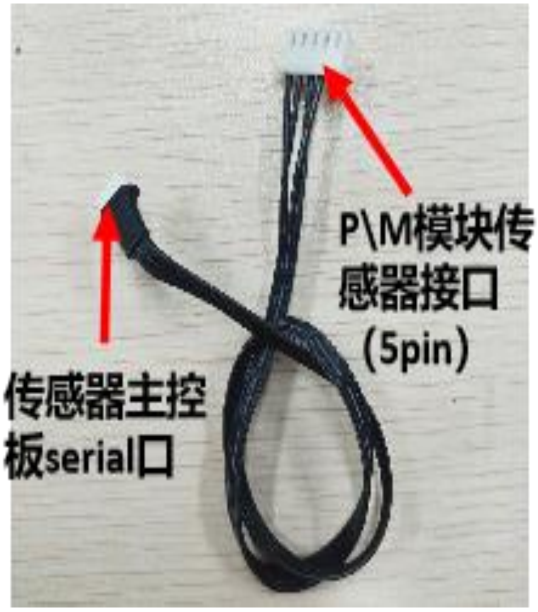
Interface Name	Function
Sensor communication port	Connect the 5Pin cable to the robot end.
Host firmware burning port	Update the host firmware program using.
Slave firmware burning port	Update the slave firmware program using.
Power indicator	When the power light shows three bars of power, it is full power, and when there is no power, the power light flashes.
Reset button	Press and hold the reset button for 3 seconds, the power light flashes when the reset is successful.
Charging port	Connect the charger adapter cable to charge.
Power switch	Press is "on", protrusion is "off".

M module:

Interface Name	Function
Sensor communication port	Connect the 5Pin cable to the robot end.
Host firmware burning port	Update the host firmware program using.
Slave firmware burning port	Update the slave firmware program using.
Power indicator	When the power light shows three bars of power, it is full power, and when there is no power, the power light flashes.
Serial communication port	It can be connected to the serial port of the control module for serial communication.
Reset button	Press and hold the reset button for 3 seconds, the power light flashes when the reset is successful.
Charging port	Connect the charger adapter cable to charge.
Power switch	Press is "on", protrusion is "off".
USB 3.0 port	Connect the keyboard, mouse, camera and other external devices to the internal computer of the M module.
HDMI port	Externally connect a monitor to the internal computer of the M module.

MiniBot has three sensor connection lines, including sensor connection line (sensor-sensor main control board), 4Pin connection line (F module-sensor main control board), and 5Pin connection line (P\M module-sensor main control board). The interface is as follows:

Sensor Cable



4Pin Cable 5Pin Cable

The following will demonstrate the usage of the sensor through some examples.

5.1 RGB Light Control

（1）Functional Description

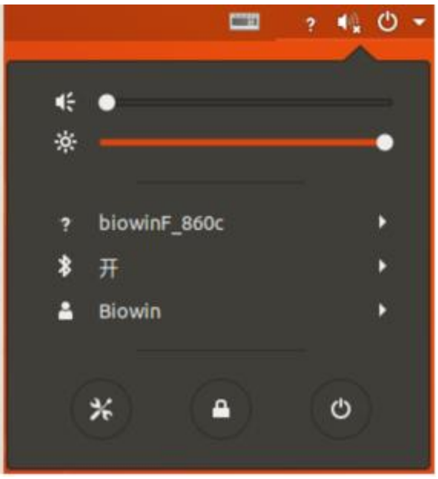
Control the full color LED light to light up three colors of red, green and blue in turn.

（2）Hardware Preparation

1 F module, 1 full color LED module and 1 MINI PRO main control board module.

（3）Operation Steps

Now take the F module as an example, connect the RGB sensor (full-color LED) to the corresponding port of the MINI PRO main control board (connected to the D2 port), and the MINI PRO main control board is connected to the F module. Turn on the power switch of the F module, and connect the PC to the hotspot corresponding to the F module.



（4）Example Execution

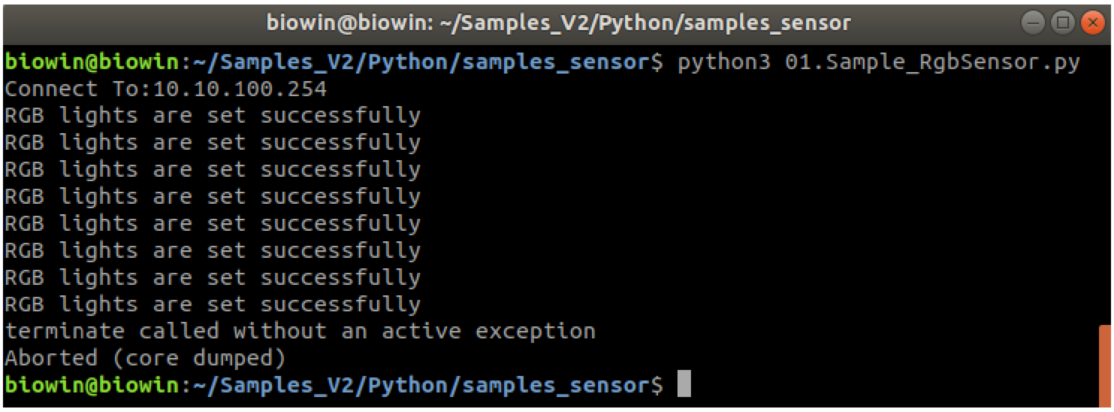
Execute Python file, the file path is:

/Home/Samples_V2/Python/samples_sensor/01.Sample_RgbSensor.py

Specific operation: Right-click in the samples_sensor directory to open the terminal, and enter the run command:

$ python3 01.Sample_RgbSensor.py

Operation result:



```
biowin@biowin: ~/Samples_V2/Python/samples_sensor
biowin@biowin:~/Samples_V2/Python/samples_sensor$ python3 01.Sample_RgbSensor.py
Connect To:10.10.100.254
RGB lights are set successfully
RGB lights are set successfully
RGB lights are set successfully
RGB lights are set successfully
RGB lights are set successfully
RGB lights are set successfully
RGB lights are set successfully
RGB lights are set successfully
terminate called without an active exception
Aborted (core dumped)
biowin@biowin:~/Samples_V2/Python/samples_sensor$
```

The RBG light turns from red to green to blue in turn, then goes out, then loops 1 more time and exits the program.

Note: The interface between the sensor and the main control board needs to be connected correctly. Here the interface between the LED and the main control board is the D2 port. If the interface is connected incorrectly, the sensor will not be able to return data correctly, and the experimental effect will not be achieved.

5.2 Use of Sound Sensor

(1) Functional Description

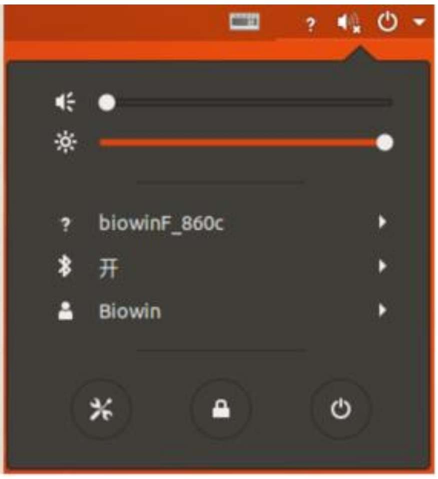
The sound sensor is used to judge whether there is sound in the external environment. If the sound is detected, the full-color LED light will light up in red, otherwise the full-color LED light will always light up in blue.

(2) Hardware Preparation

1 F module, 1 MINI PRO main control board module, 1 sound sensor module, 1 full color LED module.

(3) Operation Steps

Now take the F module as an example, connect the RGB sensor to the corresponding port of the MINI PRO main control board (connected to the D2 port), the sound sensor to the D3 port, and the MINI PRO main control board to the F module. Turn on the power switch of the F module, and connect the PC to the hotspot corresponding to the F module.



（4）Example Execution

Execute the Python file, the file path is:

/Home/Samples_V2/Python/samples_sensor/02.Sample_VoiceSensor.py

Specific operation: Right-click in the samples_sensor directory to open the terminal, and enter the run command:

$ python3 02.Sample_VoiceSensor.py

Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_sensor
biowin@biowin:~/Samples_V2/Python/samples_sensor$ python3 02.Sample_VoiceSensor.py
Connect To:10.10.100.254
The sensor was turned on successfully
1
RGB lights are set successfully
1
RGB lights are set successfully
-1
RGB lights are set successfully
-1
RGB lights are set successfully
1
RGB lights are set successfully
```

When the data collected by the sound sensor is greater than or equal to 1, the LED lights up red. If no sound is received or the sound is less than 1, the LED will always light blue.

Note: If the blue light is always on, usually the sound is low or no sound data is received at all (the terminal will print -1 at this time), then please check whether the sensor wiring is correct.

5.3 Use of Dot Matrix Module

（1）Functional Description

The control dot matrix displays the numbers 00-99 (integer).

（2）Hardware Preparation

1 F module, 1 MINI PRO main control board module and 1 dot matrix module.

（3）Operation Steps

Now take the F module as an example, connect the dot matrix to the corresponding port of the MINI PRO main control board (connected to the D4 port), and the MINI PRO main control board is connected to the F module. Turn on the power switch of the F module, and connect the PC to the hotspot corresponding to the F module.

（4）Example Execution

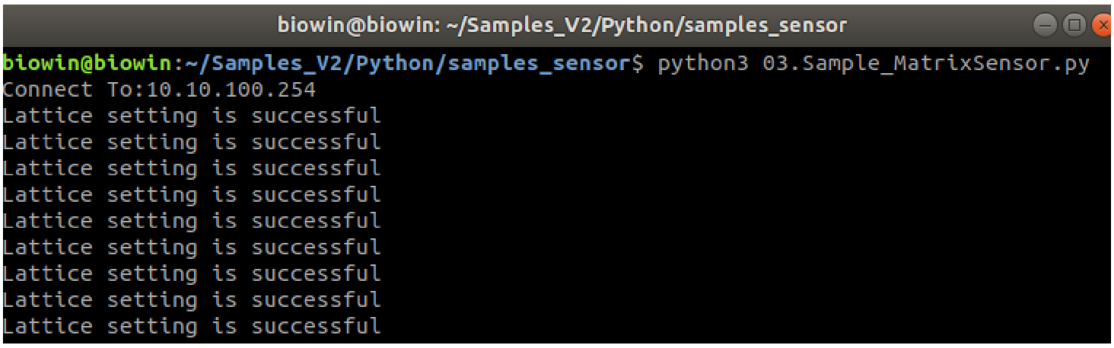
Execute the Python file, the file path is:

/Home/Samples_V2/Python/samples_sensor/03.Sample_MatrixSensor.py

Specific operation: Right-click in the samples_sensor directory to open the terminal, and enter the run command:

$ python3 03.Sample_MatrixSensor.py

Operation result:



After running the program, the dot matrix displays 00, then gradually increases by 1 until 99, and then again from 00 to 99.

5.4 Use of Gesture Sensor

(1) Functional Description

Hand movements are detected by the gesture sensor. If a gesture is detected, the full-color LED lights will light up in different colors. Specifically, the up gesture will turn on red, the down gesture will turn on green, the left gesture will turn on blue, and the right gesture will turn off.

(2) Hardware Preparation

1 F module, 1 MINI PRO main control board module, 1 Gesture Sensor Module,

1 full color LED module.

(3) Operation Steps

Now take the F module as an example, connect the full-color LED light to the corresponding port of the MINI PRO main control board (connected to the D2 port), the gesture sensor is connected to the IIC/A5 port, and the MINI PRO main control board is connected to the F module.Turn on the power switch of the F module, and connect the PC to the hotspot corresponding to the F module.

(4) Example Execution

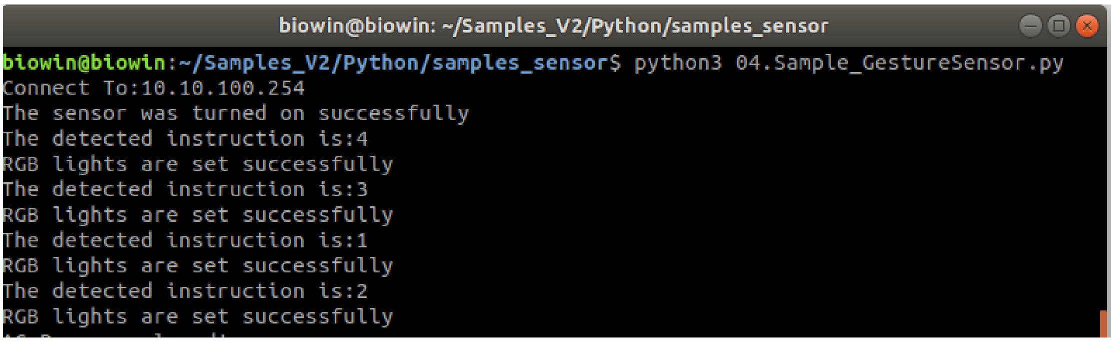
Execute the Python file, the file path is:

/Home/Samples_V2/Python/samples_sensor/04.Sample_GestureSensor.py

Specific operation: Right-click in the samples_sensor directory to open the terminal, and enter the run command:

$ python3 04.Sample_GestureSensor.py

Operation result:



Among them, the up gesture turns on red, the down gesture turns on green, the left gesture turns on blue, and the right gesture turns off.

Note: The sensitivity of the gesture sensor is not high, please try several times during the experiment.

5.5 Use of Color Sensor

(1) Functional Description

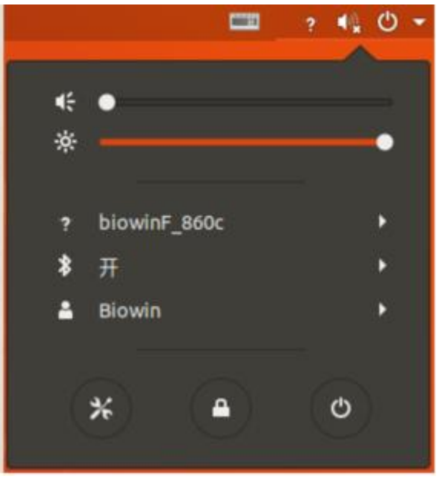
The color of the square is detected by the color sensor, and if a red square is detected, the full-color LED light turns red. When a blue square is detected, the full-color LED lights up in blue. When a green square is detected, the full-color LED lights up green.

(2) Hardware Preparation

1 F module, 1 MINI PRO main control board module, 1 color sensor, 1 full color LED module and 3 different colored squares.

（2）Operation Steps

Now take the F module as an example, connect the color sensor to the corresponding port of the MINI PRO main control board (connected to the IIC/A5 port), the full-color LED light is connected to the D2 port, and the MINI PRO main control board is connected to the F module. Turn on the power switch of the F module, and connect the PC to the hotspot corresponding to the F module.



（3）Example Execution

Execute the Python file, the file path is:

/Home/Samples_V2/Python/samples_sensor/05.Sample_ColorSensor.py

Specific operation: Right-click in the samples_sensor directory to open the terminal, and enter the run command:

$ python3 05.Sample_ColorSensor.py

Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_sensor
biowin@biowin:~/Samples_V2/Python/samples_sensor$ python3 05.Sample_ColorSensor.py
Connect To:10.10.100.254
The detected color numbers are:1
RGB lights are set successfully
The detected color numbers are:1
RGB lights are set successfully
The detected color numbers are:1
RGB lights are set successfully
The detected color numbers are:4
RGB lights are set successfully
The detected color numbers are:4
RGB lights are set successfully
The detected color numbers are:4
RGB lights are set successfully
```

Put any color square above the color sensor, the color sensor recognizes what color it is, and the full-color LED light receives data to display the same color as the square. Among them, No. 1 corresponds to red; No. 2 corresponds to green; No. 3 corresponds to blue; No. 4 is the light off. Note: The recognition time of the color sensor takes about 1.5 seconds.

5.6 Use of Line Patrol Sensor

(1) Functional Description

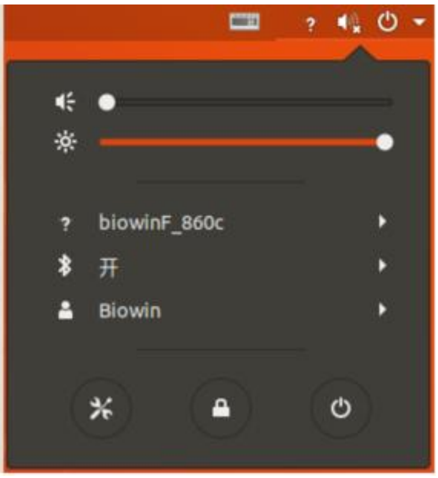
Switch detection between black lines and non-black lines through the tracking module, and control the dot matrix to display different patterns.

(2) Hardware Preparation

1 F module, 1 MINI PRO main control board module, 1 tracking module, 1 dot matrix module and 1 roll of black line patrol tape.

(3) Operation Steps

Now take the F module as an example, connect the dot matrix to the corresponding port of the MINI PRO main control board (connected to the D4 port), the tracing module is connected to the D3 port, and the MINI PRO main control board is connected to the F module. Turn on the power switch of the F module, and connect the PC to the hotspot corresponding to the F module.



(4) Example Execution

Execute the Python file, the file path is:

/Home/Samples_V2/Python/samples_sensor/06.Sample_TrackingSensor.py

Specific operation: Right-click in the samples_sensor directory to open the terminal, and enter the run command:

$ python3 06.Sample_TrackingSensor.py

Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_sensor
biowin@biowin:~/Samples_V2/Python/samples_sensor$ python3 06.Sample_TrackingSensor.py
Connect To:10.10.100.254
[0, 0]
Lattice setting is successful
[0, 0]
Lattice setting is successful
[0, 0]
Lattice setting is successful
[0, 0]
Lattice setting is successful
```

After running the program, use the tracing module to switch back and forth between the black line and the non-black line area (the detection distance is about 3cm), you can see the corresponding expression displayed by the dot matrix.

5.7 Use of Remote Control Sensor

(1) Functional Description

The remote control receiving module is controlled by the remote control, so as to control the dot matrix to display different patterns.

(2) Hardware Preparation

1 F module, 1 MINI PRO main control board module, 1 remote control receiving module, 1 remote control and 1 dot matrix module.

(3) Operation Steps

Now take the F module as an example, connect the dot matrix to the corresponding port of the MINI PRO main control board (connected to the D4 port), the remote control receiving module is connected to the D3 port, and the MINI PRO main control board is connected to the F module.Turn on the power switch of the F module, and connect the PC to the hotspot corresponding to the F module.

(4) Example Execution

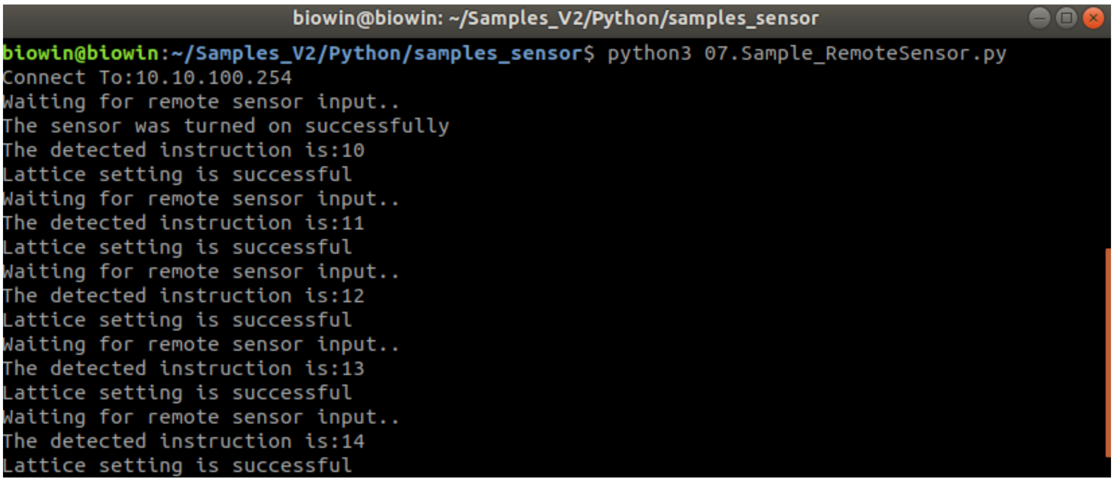
Execute the Python file, the file path is:

/Home/Samples_V2/Python/samples_sensor/07.Sample_RemoteSensor.py

Specific operation: Right-click in the samples_sensor directory to open the terminal, and enter the run command:

$ python3 07.Sample_RemoteSensor.py

Operation result:



```
biowin@biowin:~/Samples_V2/Python/samples_sensor$ python3 07.Sample_RemoteSensor.py
Connect To:10.10.100.254
Waiting for remote sensor input..
The sensor was turned on successfully
The detected instruction is:10
Lattice setting is successful
Waiting for remote sensor input..
The detected instruction is:11
Lattice setting is successful
Waiting for remote sensor input..
The detected instruction is:12
Lattice setting is successful
Waiting for remote sensor input..
The detected instruction is:13
Lattice setting is successful
Waiting for remote sensor input..
The detected instruction is:14
Lattice setting is successful
```

Enter numbers on the remote control, and the dot matrix will display the entered number information.

5.8 Use of Ultrasonic Sensor

(1) Functional Description

The distance from the square is detected by the ultrasonic module, so as to control the dot matrix to display different patterns.

(2) Hardware Preparation

1 F module, 1 MINI PRO main control board module, 1 ultrasonic sensor module, 1 dot matrix module and 3 different colored squares.

(3) Operation Steps

Now take the F module as an example, connect the dot matrix to the corresponding port of the MINI PRO main control board (connected to D4 port), the ultrasonic sensor is connected to the D3 port, and the MINI PRO main control board is connected to the F module. Turn on the power switch of the F module, and connect the PC to the hotspot corresponding to the F module.

(4) Example Execution

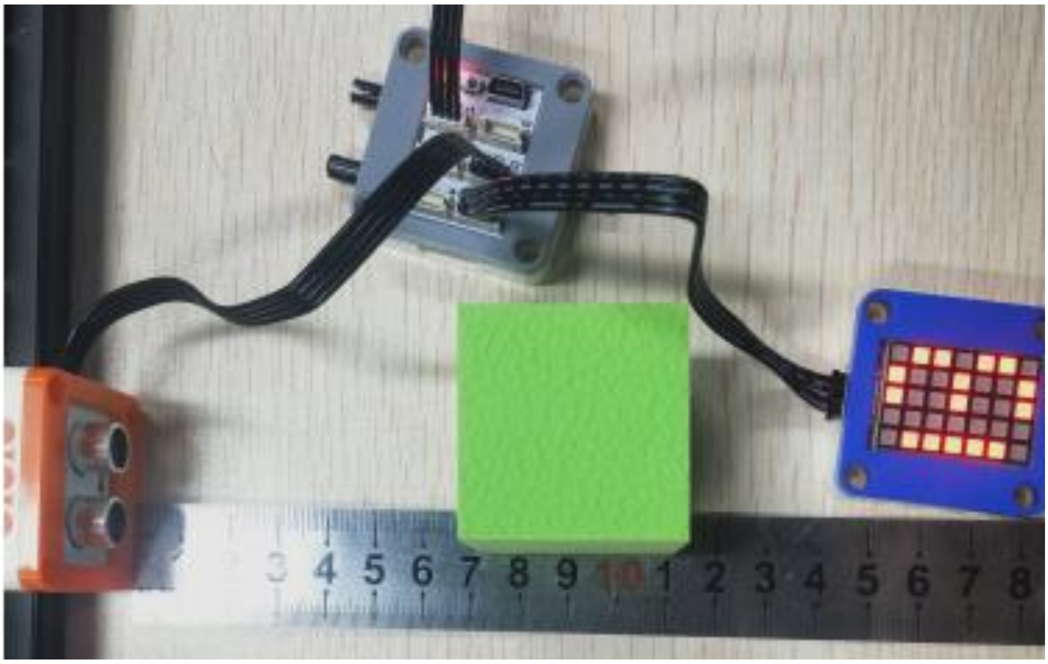
Execute the Python file, the file path is:

/Home/Samples_V2/Python/samples_sensor/08.Sample_UltrasonicSensor.py

Specific operation: Right-click in the samples_sensor directory to open the terminal, and enter the run command:

$ python3 08.Sample_UltrasonicSensor.py

Operation result:



After running the program, ultrasonic test to different distances, the dot matrix shows different expressions.

Note: The ultrasonic detection distance should not exceed 400 cm.

6 Robot Comprehensive Case

Through the content of the previous two chapters, I believe that you have a certain foundation for modules and sensors. Now let us combine the previous content to carry out some robot configurations and experiments.

6.1 Modular Operation Arm Experiment

6.1.1 Robot Arm Basic Experiment

(1) Realization function: We set up three small experiments for the basic configuration of the robot arm, which are the joint control experiment, the object grasping experiment, and the forward and reverse angle solution experiments. Next, we first build the robot arm and then carry out the experiment.

(2) Hardware Preparation

1 F module fixing seat, 1 F module, 2 I modules, 1 G module and 3 T modules.

(3) Machine configuration: The robotic arm (F I T T T I G) in this experiment is a five-degree-of-freedom robotic arm, where the numbering sequence is 20, 30, 31, 32, 21, and 40, respectively. Note: 20~29 are I modules, 30~39 are T modules, and 40~49 are G modules. Build as shown below:

After building the configuration of the robot, it is time to experiment.

6.1.1.1 Joint Module Control Experiment

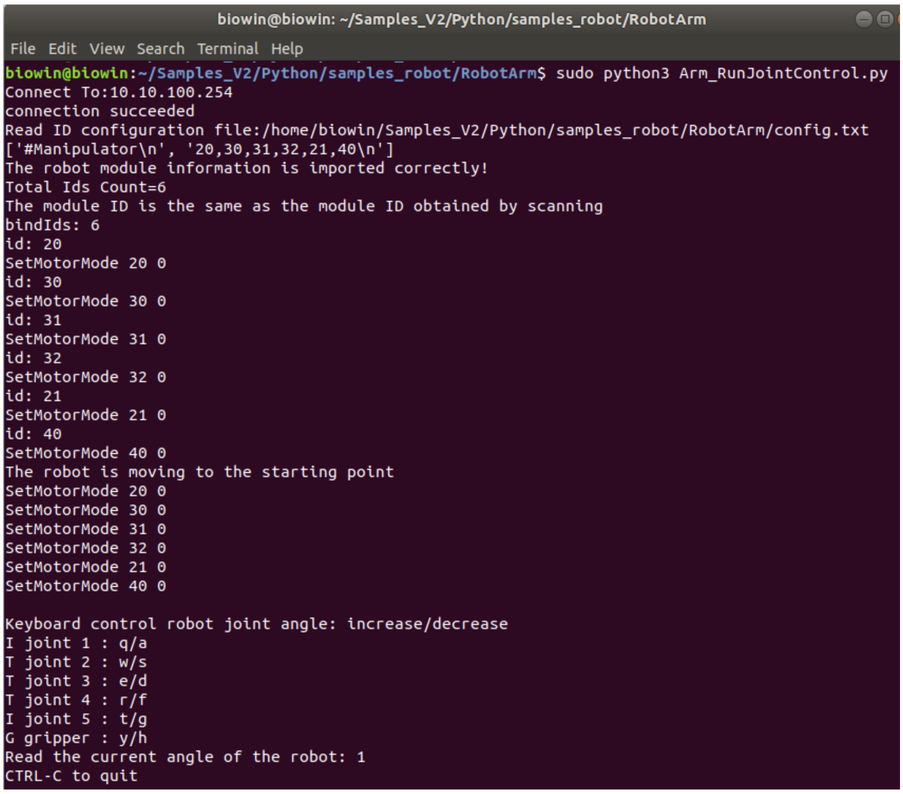
Realization function: Control a single joint module through the keyboard.

Steps:

Go to: /Home/Samples_V2/Python/samples_robot/RobotArm directory, right-click to open the terminal and run the following command.

$ sudo python3 Arm_RunJointControl.py

Operation result:



Users can control the movement of a single joint by typing letters such as "q, w" on the keyboard. Press "ctrl+c" to exit.

Note: If there is a problem of ID mismatch, please modify it according to the joint number you actually built in the "config.txt" file in the same directory.

Tip: In all experiments of the robot, if the ID number is incorrect, it is necessary to check whether the ID number of the built robot is consistent with the number of the config.txt file in the same directory. If not, it is necessary to change the ID number in the config.txt file to the ID number that you actually built.

6.1.1.2 Grasping Experiment

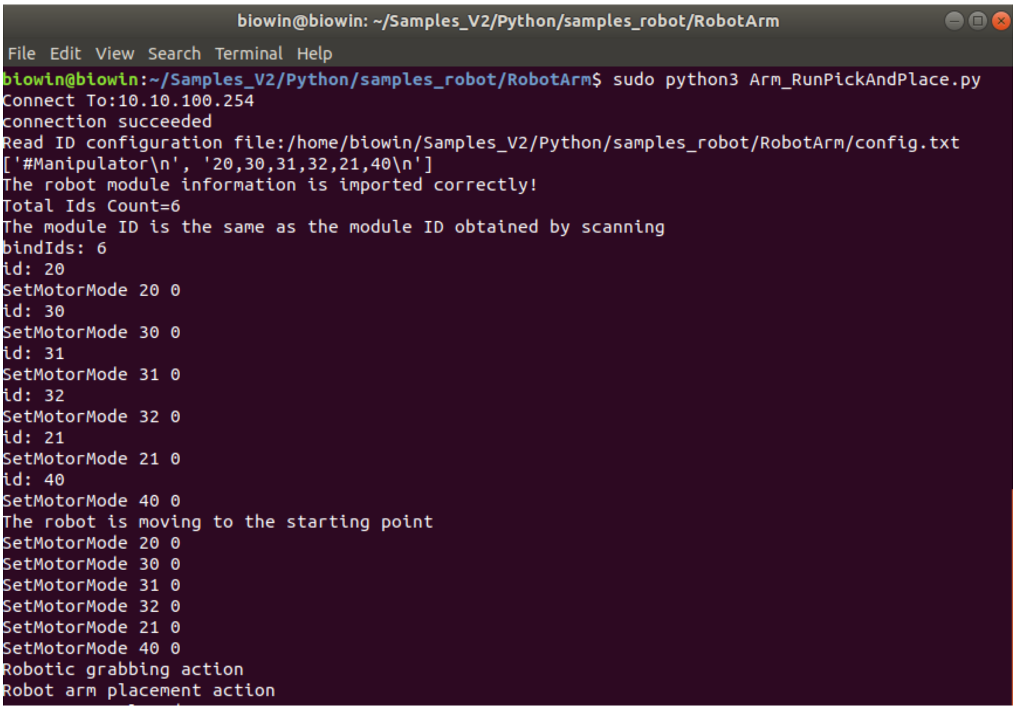
Realization function: View the process of picking and placing objects of the robotic arm.

Steps:

Go to: /Home/Samples_V2/Python/samples_robot/RobotArm directory, right-click to open the terminal and run the following command.

$ sudo python3 Arm_RunPickAndPlace.py

Operation result:



After running the program, the robotic arm will automatically grab the object and move it to a certain distance before putting it down, and repeat the grabbing action until you press "ctrl+c" to exit.

Note: If there is a problem of ID mismatch, please modify it according to the joint number you actually built in the "config.txt" file in the same directory.

6.1.1.3 Forward and Reverse Angle Experiment of Robot Arm

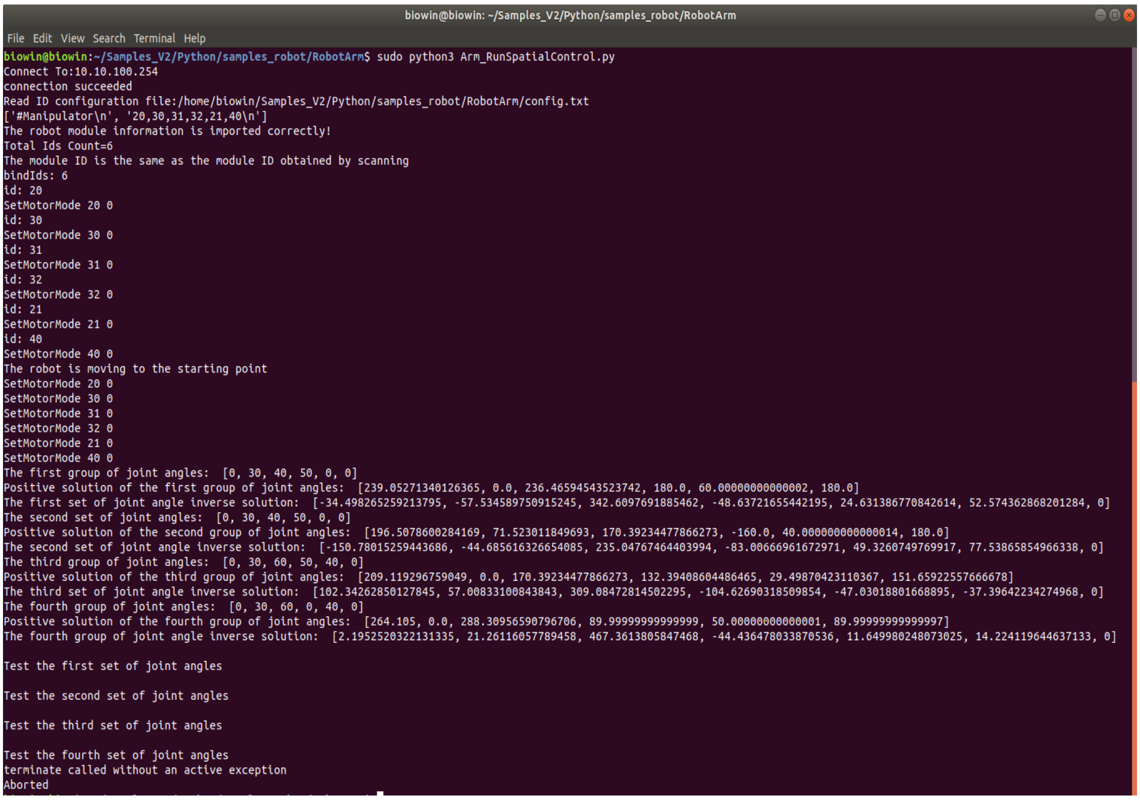
Realization function: View the forward and reverse joint angles of the robotic arm.

Steps:

Go to: /Home/Samples_V2/Python/samples_robot/RobotArm directory, right-click to open the terminal and run the following command.

$ sudo python3 Arm_RunSpatialControl.py

Operation result:



```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotArm
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotArm$ sudo python3 Arm_RunSpatialControl.py
Connect To:10.10.100.254
connection succeeded
Read ID configuration file:/home/biowin/Samples_V2/Python/samples_robot/RobotArm/config.txt
['#Manipulator\n', '20,30,31,32,21,40\n']
The robot module information is imported correctly!
Total Ids Count=6
The module ID is the same as the module ID obtained by scanning
bindIds: 6
id: 20
SetMotorMode 20 0
id: 30
SetMotorMode 30 0
id: 31
SetMotorMode 31 0
id: 32
SetMotorMode 32 0
id: 21
SetMotorMode 21 0
id: 40
SetMotorMode 40 0
The robot is moving to the starting point
SetMotorMode 20 0
SetMotorMode 30 0
SetMotorMode 31 0
SetMotorMode 32 0
SetMotorMode 21 0
SetMotorMode 40 0
The first group of joint angles:  [0, 30, 40, 50, 0, 0]
Positive solution of the first group of joint angles:  [239.05271340126365, 0.0, 236.46594543523742, 180.0, 60.00000000000002, 180.0]
The first set of joint angle inverse solution:  [-34.498265259213795, -57.534589750915245, 342.6097691885462, -48.63721655442195, 24.631386770842614, 52.574362868201284, 0]
The second set of joint angles:  [0, 30, 40, 50, 0, 0]
Positive solution of the second group of joint angles:  [196.5078600284169, 71.523011849693, 170.39234477866273, -160.0, 40.000000000000014, 180.0]
The second set of joint angle inverse solution:  [-150.78015259443686, -44.685616326654085, 235.04767464403994, -83.00666961672971, 49.3260749769917, 77.53865854966338, 0]
The third group of joint angles:  [0, 30, 60, 50, 40, 0]
Positive solution of the third group of joint angles:  [209.119296759049, 0.0, 170.39234477866273, 132.39408604486465, 29.49870423110367, 151.65922557666678]
The third set of joint angle inverse solution:  [102.34262850127845, 57.00833100843843, 309.08472814502295, -104.62690318509854, -47.03018801668895, -37.39642234274968, 0]
The fourth group of joint angles:  [0, 30, 60, 0, 40, 0]
Positive solution of the fourth group of joint angles:  [264.105, 0.0, 288.30956590796706, 89.99999999999999, 50.00000000000001, 89.99999999999997]
The fourth group of joint angle inverse solution:  [2.1952520322131335, 21.26116057789458, 467.3613805847468, -44.436478033870536, 11.649980248073025, 14.224119644637133, 0]

Test the first set of joint angles

Test the second set of joint angles

Test the third set of joint angles

Test the fourth set of joint angles
terminate called without an active exception
Aborted
```

The program pre-sets the calculation of forward and inverse solutions for four groups of joint angles, and then automatically calculates and outputs the forward and inverse solutions of these four groups of joint angles.

Note: The inverse solution does not calculate the angle of the gripper, it defaults to 0.

6.1.2 Sensor Application Example

After completing the basic experiments, we provide users with more advanced experiments. Below we will implement some interesting experiments through the combination of sensors and main control modules.

6.1.2.1 Color Sorting

（1）Realization function: Control the robotic arm to grab the squares from the A/B/C area of the map to the color sensor for detection, and place the squares in the corresponding color area according to the results of the detected color, and the full-color LED lights are on the same color as the squares.

（2）Hardware Preparation

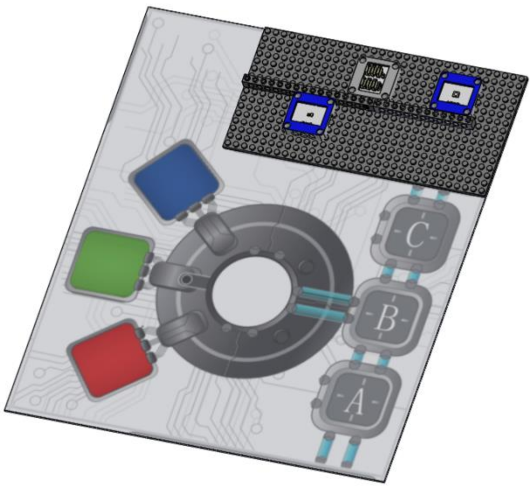
1 F module fixing seat, 1 F module, 2 I modules, 1 G module, 3 T modules, 1 MINI PRO main control board module, 1 color sensor module (connected to IIC/A5 port), 1 full color LED light module (connected to D2 port) and 3 different colored squares.

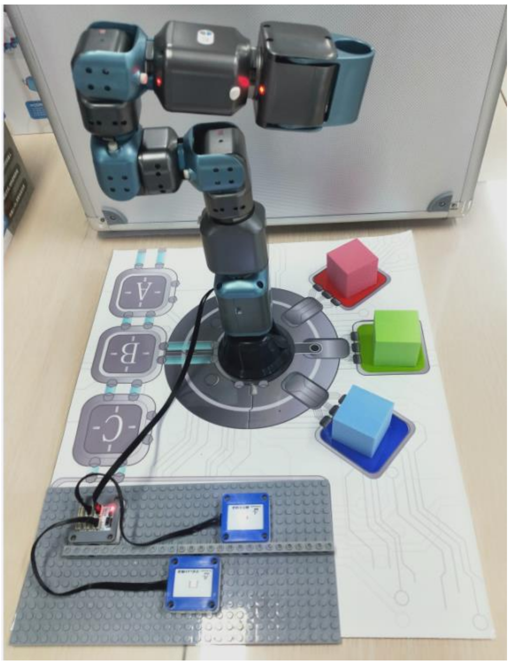


(3) Construction of the Configuration

Machine configuration: Build a (the ID of the F I T T T I G module is determined according to Section 6.1.1) type robotic arm, and connect the sensor according to the corresponding interface. Fix the robotic arm on the base of the map, the color sensor module is connected to the IIC/A5 port, and the full-color LED light module is connected to the D2 port. In the experiment, the color sensor should be placed in the sensor area on the map (fixed with black pins). Note that the location of the color sensor should be placed in the corresponding position according to the construction diagram. If the position is not accurate, it will affect the color recognition result.

Build as shown below:





Note: The fixed base of the F module should be fixed according to the corresponding position of the map mark (the mark of the handle of the fixed base).

(4) Example Execution

Turn on the power switch of the F module, and connect the PC to the hotspot corresponding to the F module.

Go to:/Home/Samples_V2/Python/samples_robot/RobotArm/SensorsApp/

01.ColorSorting/

Right-click to open the terminal and execute the Python command:

$ sudo python3 ColorSorting_Run.py

Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotArm/SensorsApp/01.ColorSorting
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotArm/SensorsApp/01.ColorSorting$ sudo python3 ColorSorting_Run.py
Connect To:10.10.100.254
connection succeeded
Read ID configuration file:/home/biowin/Samples_V2/Python/samples_robot/RobotArm/SensorsApp/01.ColorSorting/config.txt
['#Manipulator\n', '20,30,31,32,21,40\n']
The robot module information is imported correctly!
Total Ids Count=6
The module ID is the same as the module ID obtained by scanning
[20, 30, 31, 32, 21, 40]
bindIds: 6
id: 20
SetMotorMode 20 0
id: 30
SetMotorMode 30 0
id: 31
SetMotorMode 31 0
id: 32
SetMotorMode 32 0
id: 21
SetMotorMode 21 0
id: 40
SetMotorMode 40 0
sorting!
The detected color numbers are:1
Detected red
RGB lights are set successfully
```

Use the operating arm to grab the squares in areas A-C to the color sensor to detect the color, place the squares in the corresponding color area according to the detection results, and use the LED lights to light up the corresponding color. By grabbing squares of different colors and placing them on the color sensor, the color sensor can identify the squares of different colors and return the data, while the full color LED light receives the data and displays the same color as the squares.

Note: Check and make sure that there is no error in the connection between the sensor and the module.

Tip: In the sensor experiment of the robotic arm, the configuration and number of the robotic arm do not need to be changed, so in the experiment, it is the sensors that are actually added and modified.

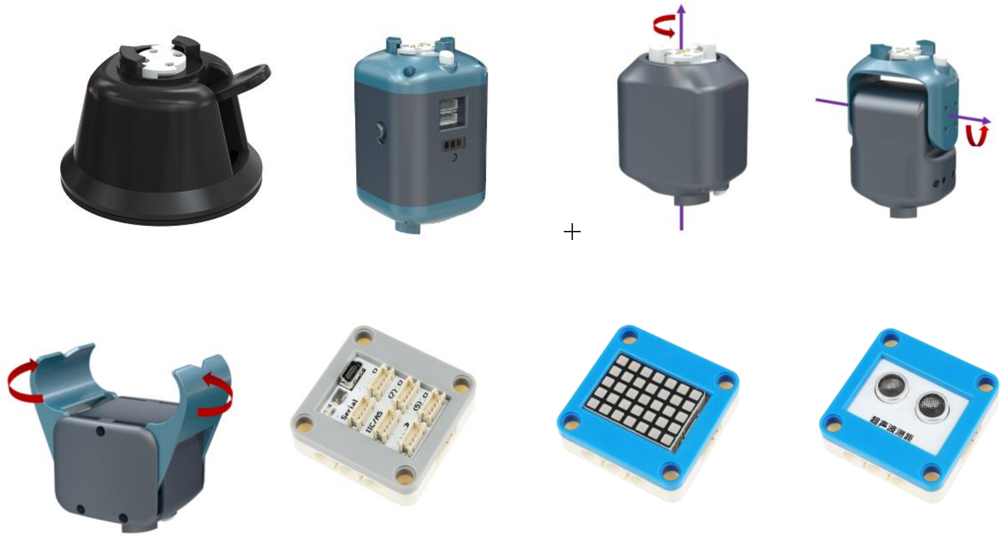
6.1.2.2 The Robot Arm Grabs the Square You Put

（1）Realization function: Place the square randomly in the A/B/C area, and the ultrasonic sensor detects the position of the square, so as to control the robotic arm to grab the square where the square is placed, and place it in the red, green and blue area (currently the logic is: A area- red area, B area - green area, C area - blue area), and the dot matrix displays different expressions at the same time.

（2）Hardware Preparation

1 main control module (F module), 1 F module fixing seat, 3 T modules, 2 I

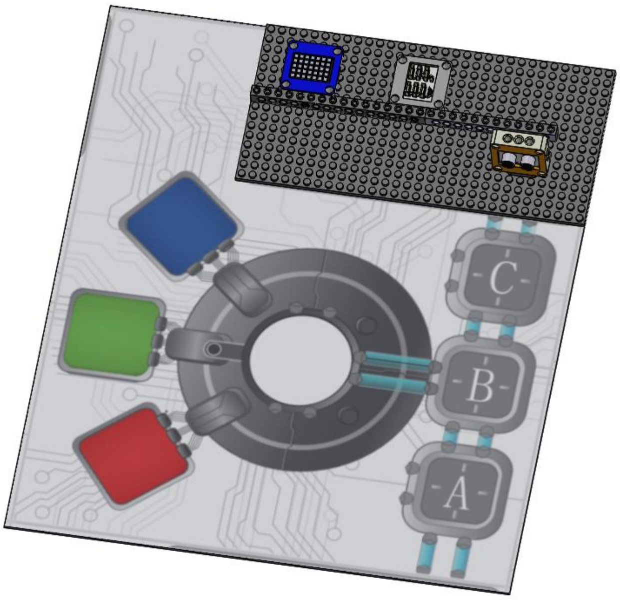
modules, 1 G module, 1 MINI PRO main control board module, 1 ultrasonic ranging module, 1 dot matrix module and 3 different colored squares.



（3）Construction of the Configuration

Build (the ID of the F I T T T I G module is determined according to Section 6.1.1) a five-degree-of-freedom robotic arm and fix it on the map base. Connect the cables between the sensor, the main control board and the F module. Among them, the ultrasonic sensor is connected to the D3 port, and the dot matrix is connected to the D4 port. The ultrasonic sensor should be in the same line with the A, B, and C areas. At the same time, three squares are placed in the A-C area, and the color sensor should be placed in the 5-9 column of the sensing area fixed plate (from the ABC area to the red, green and blue area), the 4th row (from the direction close to the robot arm to the direction away from the robot arm) .

Build as shown below:



(4) Example Execution

Turn on the power switch of the F module, and connect the PC to the hotspot corresponding to the F module.

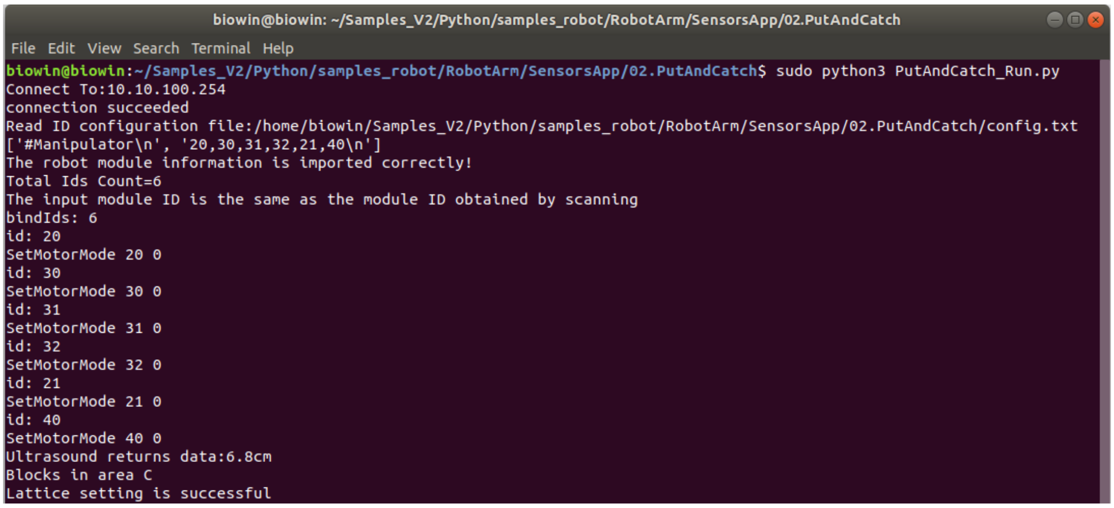
Go to:/Home/Samples_V2/Python/samples_robot/RobotArm/SensorsApp/

02.PutAndCatch/

Right-click to open the terminal and execute the Python command:

$ sudo python3 PutAndCatch_Run.py

Operation result:



```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotArm/SensorsApp/02.PutAndCatch
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotArm/SensorsApp/02.PutAndCatch$ sudo python3 PutAndCatch_Run.py
Connect To:10.10.100.254
connection succeeded
Read ID configuration file:/home/biowin/Samples_V2/Python/samples_robot/RobotArm/SensorsApp/02.PutAndCatch/config.txt
['#Manipulator\n', '20,30,31,32,21,40\n']
The robot module information is imported correctly!
Total Ids Count=6
The input module ID is the same as the module ID obtained by scanning
bindIds: 6
id: 20
SetMotorMode 20 0
id: 30
SetMotorMode 30 0
id: 31
SetMotorMode 31 0
id: 32
SetMotorMode 32 0
id: 21
SetMotorMode 21 0
id: 40
SetMotorMode 40 0
Ultrasound returns data:6.8cm
Blocks in area C
Lattice setting is successful
```

After running the program, the ultrasonic sensor detects that there is a square in area A, and the robotic arm grabs the square from area A and places it in the red area. If it detects a square in area B, grab it to the green area. If it detects a square in area C, grab it to the blue area. At the same time, the expression of the dot matrix will change according to different detection results.

6.1.2.3 Control of Gripper Tension

(1) Realization function: Artificially place the squares randomly in the range of A-C area, the ultrasonic sensor detects the position of the square, according to the distance between the square and the ultrasonic wave (Note: The square can only be placed in the same line as the ultrasonic wave and only in the range of A-C area inside), so as to control the size of the opening of the gripper of the robotic arm, and the dot matrix displays the expression of the opening of the gripper.

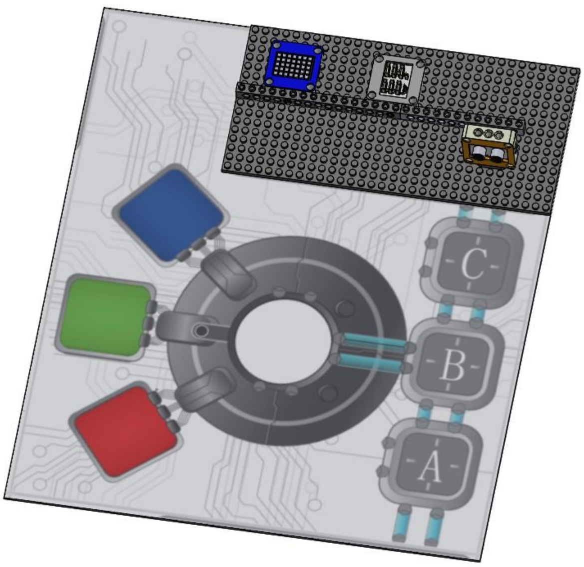
(2) Hardware Preparation

1 main control module (F module), 1 F module fixing seat, 3 T modules, 2 I modules, 1 G module, 1 MINI PRO main control board module, 1 ultrasonic ranging module, 1 dot matrix module and 3 different colored squares.

(3) Construction of the Configuration

Build (the ID of the F I T T T I G module is determined according to Section 6.1.1) a five-degree-of-freedom robotic arm and fix it on the map base. Connect the cables between the sensor, the main control board and the F module. Among them, the ultrasonic sensor is connected to the D3 port, and the dot matrix is connected to the D4 port. The ultrasonic sensor should be in the same line with the A, B, and C areas.

Build as shown below:



(4) Example Execution

Turn on the power switch of the F module, and connect the PC to the hotspot corresponding to the F module.

Go to:/Home/Samples_V2/Python/samples_robot/RobotArm/SensorsApp/

03.GripperControl/

Right-click to open the terminal and execute the Python command:

$ python3 GripperControl_Run.py

Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotArm/SensorsApp/03.GripperControl
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotArm/SensorsApp/03.GripperControl$ python3 GripperControl_Run.py
Connect To:10.10.100.254
connection succeeded
Read ID configuration file:/home/biowin/Samples_V2/Python/samples_robot/RobotArm/SensorsApp/03.GripperControl/config.txt
['#Manipulator\n', '20,30,31,32,21,40\n']
The robot module information is imported correctly!
Total Ids Count=6
The input module ID is the same as the module ID obtained by scanning
bindIds: 6
id: 20
SetMotorMode 20 0
id: 30
SetMotorMode 30 0
id: 31
SetMotorMode 31 0
id: 32
SetMotorMode 32 0
id: 21
SetMotorMode 21 0
id: 40
SetMotorMode 40 0
Lattice setting is successful
Ultrasound returns data:0.6cm
Lattice setting is successful
Ultrasound returns data:0.5cm
Lattice setting is successful
Ultrasound returns data:8.7cm
Lattice setting is successful
Ultrasound returns data:11.6cm
Lattice setting is successful
Ultrasound returns data:17.2cm
Lattice setting is successful
Ultrasound returns data:56.3cm
Lattice setting is successful
```

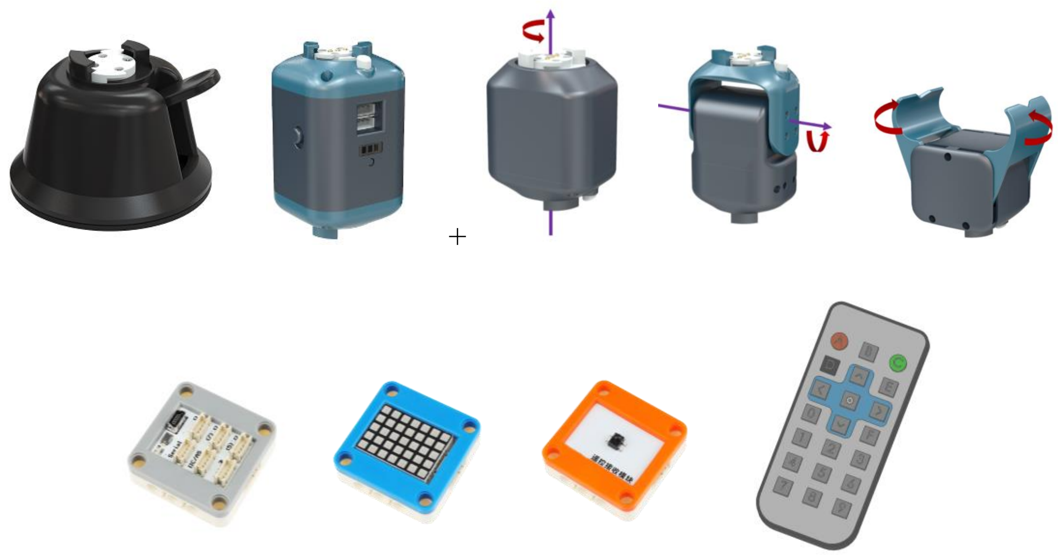
In this case, when the ultrasonic sensor detects whether there is an object in the B and C areas, here we can use a square as an obstacle. When we put the square in the B (far away from the ultrasonic) area, the angle of the robotic arm expands. When we put the square in the C area (closer to the ultrasonic wave), the angle of the robotic arm is closed. It actually simulates the process of a robotic arm grabbing an object.

6.1.2.4 Random Draw

（1）Realization function: The robotic arm swings back and forth on the three squares, waiting for the command from the remote control. After pressing "1" with the remote control, which square the robotic arm grabs is determined according to the random number generated, and the dot matrix displays different expressions.

（2）Hardware Preparation

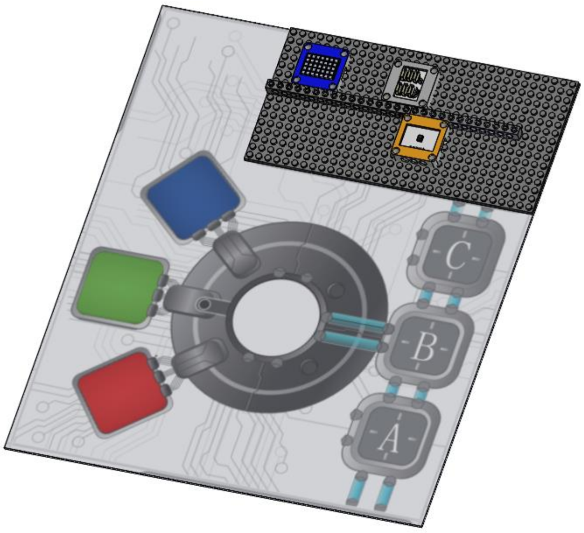
1 F module, 1 F module fixing seat, 3 T modules, 2 I modules, 1 G module, 1 remote control receiving module, 1 MINI PRO main control board module, 1 remote control, 1 dot matrix module and 3 different colored squares.



（3）Construction of the Configuration

Build (the ID of the F I T T T I G module is determined according to Section 6.1.1) a five-degree-of-freedom robotic arm and fix it on the map base. Connect the cables between the sensor, the main control board and the F module. Among them, the remote control receiving module is connected to the D3 port, and the dot matrix is connected to the D4 port. The robotic arm is aimed at the red, blue and green areas.

Build as shown below:



(4) Example Execution

Turn on the power switch of the F module, and connect the PC to the hotspot corresponding to the F module.

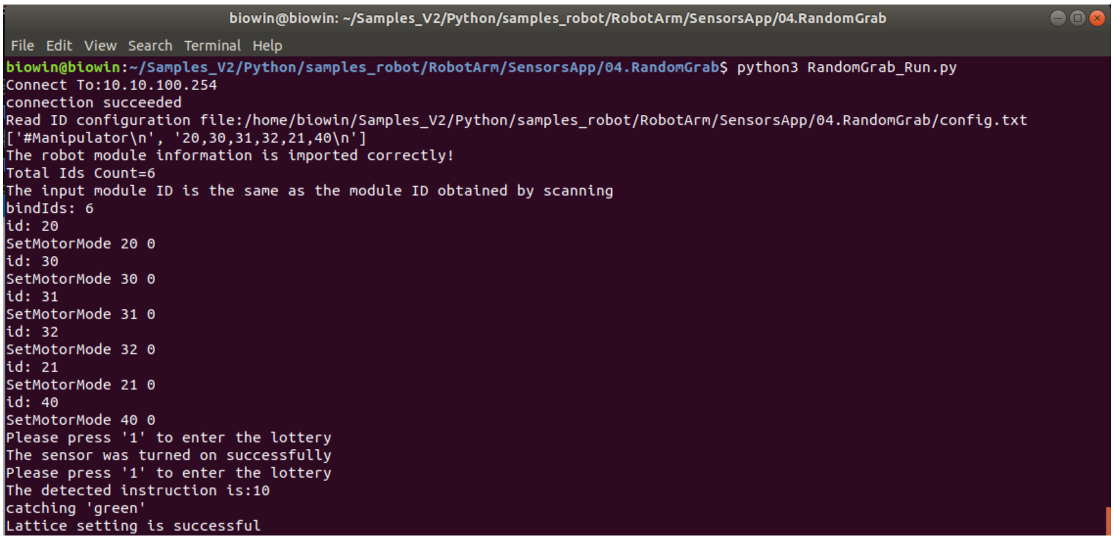
Go to:/Home/Samples_V2/Python/samples_robot/RobotArm/SensorsApp/

04.RandomGrab/

Right-click to open the terminal and execute the Python command:

$ python3 RandomGrab_Run.py

Operation result:



```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotArm/SensorsApp/04.RandomGrab
File  Edit  View  Search  Terminal  Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotArm/SensorsApp/04.RandomGrab$ python3 RandomGrab_Run.py
Connect To:10.10.100.254
connection succeeded
Read ID configuration file:/home/biowin/Samples_V2/Python/samples_robot/RobotArm/SensorsApp/04.RandomGrab/config.txt
['#Manipulator\n', '20,30,31,32,21,40\n']
The robot module information is imported correctly!
Total Ids Count=6
The input module ID is the same as the module ID obtained by scanning
bindIds: 6
id: 20
SetMotorMode 20 0
id: 30
SetMotorMode 30 0
id: 31
SetMotorMode 31 0
id: 32
SetMotorMode 32 0
id: 21
SetMotorMode 21 0
id: 40
SetMotorMode 40 0
Please press '1' to enter the lottery
The sensor was turned on successfully
Please press '1' to enter the lottery
The detected instruction is:10
catching 'green'
Lattice setting is successful
```

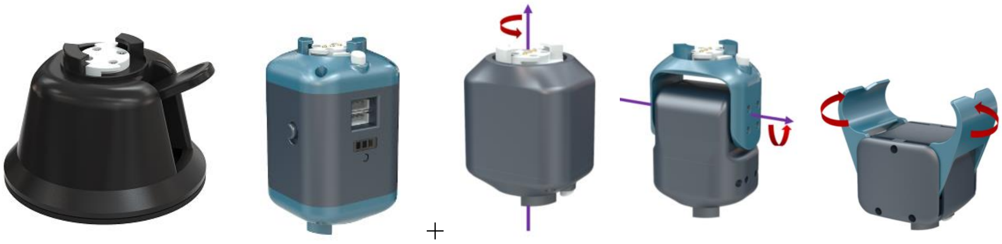
After running the program, the robotic arm swings back and forth in the red, blue and green areas. When the "1" button of the remote control is pressed, the robotic arm randomly grabs a square from one of the areas and shakes it (simulating a prize draw). The dot matrix displays different emoticons, representing the mood of winning different prizes. Then put it down for the next draw.

6.1.2.5 Remote Control

（1）Realization function: By pressing different buttons on the remote control (buttons "1"-"6"), the robotic arm is controlled to move the squares left and right (A area - red area, B area - green area, C area - blue area).

（2）Hardware Preparation

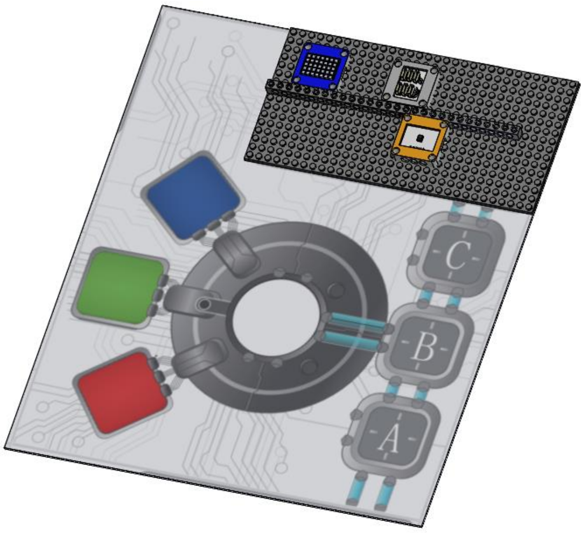
1 F module, 1 F module fixing seat, 3 T modules, 2 I modules, 1 G module, 1 remote control receiving module, 1 MINI PRO main control board module, 1 remote control, 1 dot matrix module and 3 different colored squares.



(3) Construction of the Configuration

Build (the ID of the F I T T T I G module is determined according to Section 6.1.1) a five-degree-of-freedom robotic arm and fix it on the map base. Connect the cables between the sensor, the main control board and the F module. Among them, the remote control receiving module is connected to the D3 port, and the dot matrix is connected to the D4 port. The robotic arm is aimed at the red, blue and green areas, and the blocks can be placed arbitrarily in the designated area.

Build as shown below:



(4) Example Execution

Turn on the power switch of the F module, and connect the PC to the hotspot corresponding to the F module.

Go to:/Home/Samples_V2/Python/samples_robot/RobotArm/SensorsApp/

05.RemoteControl/

Right-click to open the terminal and execute the Python command:

$ python3 RemoteControl_Run.py

Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotArm/SensorsApp/05.RemoteControl
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotArm/SensorsApp/05.RemoteControl$ python3 RemoteControl_Run.py
Connect To:10.10.100.254
connection succeeded
Read ID configuration file:/home/biowin/Samples_V2/Python/samples_robot/RobotArm/SensorsApp/05.RemoteControl/config.txt
['#Manipulator\n', '20,30,31,32,21,40\n']
The robot module information is imported correctly!
Total Ids Count=6
The input module ID is the same as the module ID obtained by scanning
bindIds: 6
id: 20
SetMotorMode 20 0
id: 30
SetMotorMode 30 0
id: 31
SetMotorMode 31 0
id: 32
SetMotorMode 32 0
id: 21
SetMotorMode 21 0
id: 40
SetMotorMode 40 0
Please enter a key
The sensor was turned on successfully
The detected instruction is:10
Return: 10
Press '1' - from A to red
Lattice setting is successful
Please enter a key
The detected instruction is:11
Return: 11
Press '2' - from B to green
Lattice setting is successful
```

Run the program and use the remote control to input numbers to control the robotic arm to perform corresponding actions. The relationship between keys and actions is as follows:

Key	Command
1	Move the object from the A area to the red area.
2	Move the object from the B area to the green area.
3	Move the object from the C area to the blue area.
4	Move the object from the red area to the A area.
5	Move the object from the green area to the B area.
6	Move the object from the bule area to the C area.

6.2 Experiment of Biped Wheeled Robot

6.2.1 Demo Experiment of Biped Robot Motion

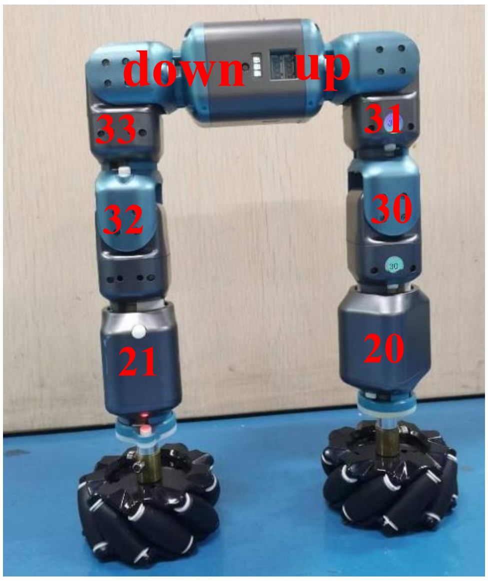
(1) Realization function: Control the biped robot to take three steps forward and backward.

(2) Hardware Preparation

1 F module, 2 I modules, 4 T modules and 1 pair of Mecanum wheels.

(3) Construction of the Configuration

Build the bionic biped robot according to the joint numbers below, and then place the robot on the ground. If the ID does not match, modify the "config.txt" file in the same directory.



(4) Example Execution

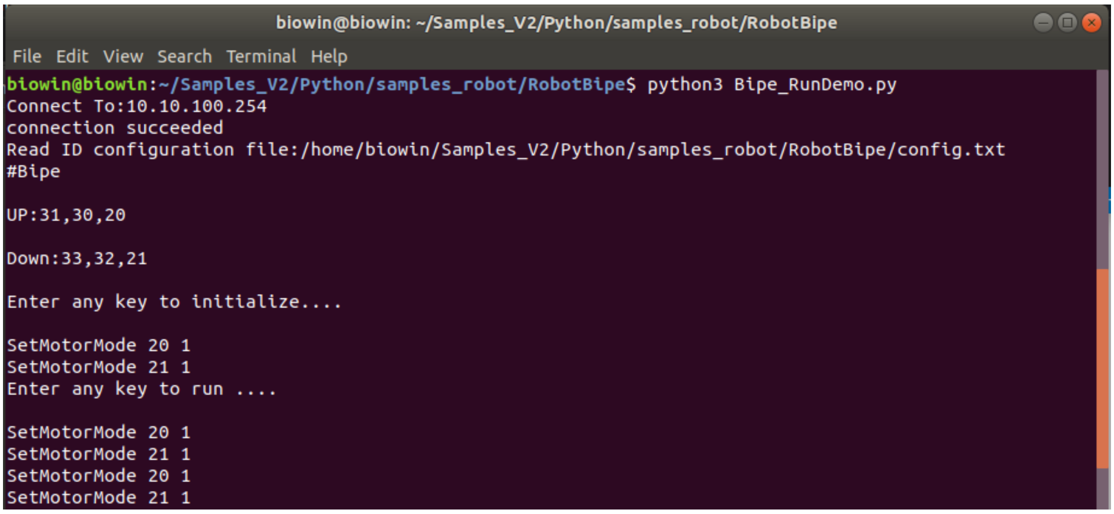
Steps: Turn on the F module switch, and connect the PC to the hotspot corresponding to the F module, which is generally biowinF_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotBipe/

Right-click to open the terminal and execute the Python command:

$ python3 Bipe_RunDemo.py

Operation result:



After running the program, the motion function is triggered by pressing the Enter key, and the biped robot will move forward and backward according to the preset motion.

Note: After running the program, the biped robot will be initialized. The speed of this process is relatively high, so be careful to avoid accidental injury.

6.2.2 Control Experiment of Biped Robot

(1) Realization function: Control the biped robot to walk forward and backward through the keyboard keys.

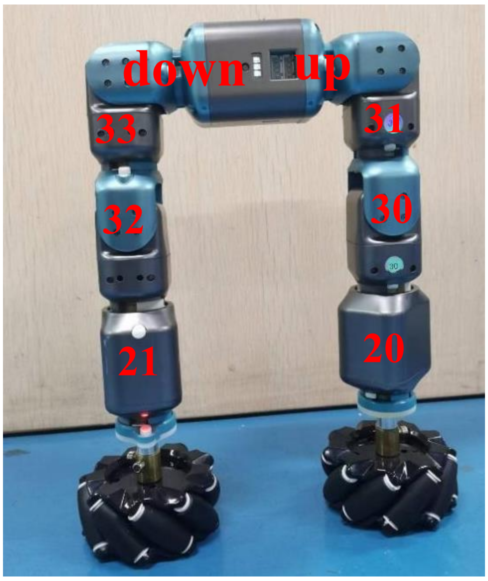
(2) Hardware Preparation

1 F module, 2 I modules, 4 T modules and 1 pair of Mecanum wheels.




(3) Construction of the Configuration

Build the bionic biped robot according to the joint numbers below, and then place the robot on the ground. If the ID does not match, modify the "config.txt" file in the same directory.



(4) Example Execution

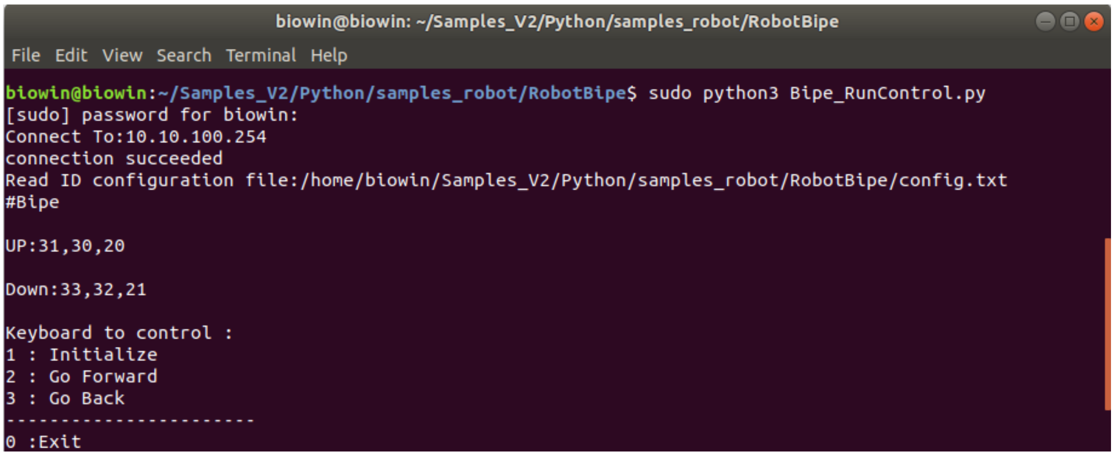
Steps: Turn on the F module switch, and connect the PC to the hotspot corresponding to the F module, which is generally biowinF_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotBipe/

Right-click to open the terminal and execute the Python command:

$ sudo python3 Bipe_RunControl.py

Operation result:



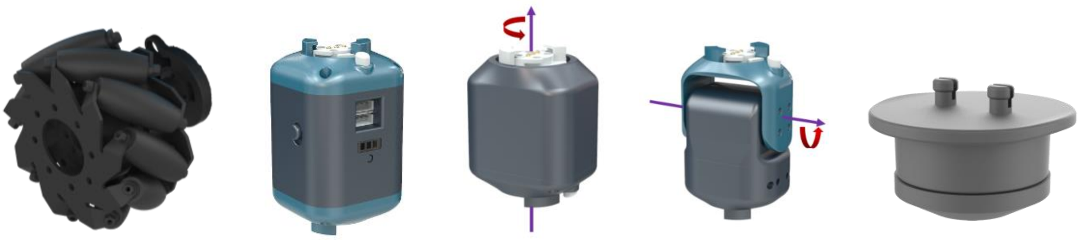
After running the program, press the number "1" on the keyboard, the biped robot will be initialized at this time, the robot speed is relatively fast in this process, in order to avoid accidental injury, it is recommended that the user lay it flat on the ground. Among them, pressing the number "2", the robot moves forward, and pressing the number "3", the robot retreats.

6.2.3 Demo Experiment of Bi-Wheel Robot Motion

（1） Realization function: Control the two-wheeled robot to realize the movement of forward, backward, left turn and right turn.

（2） Hardware Preparation

1 F module, 2 I modules, 4 T modules, 1 pair of Mecanum wheels and 1 auxiliary universal wheel.

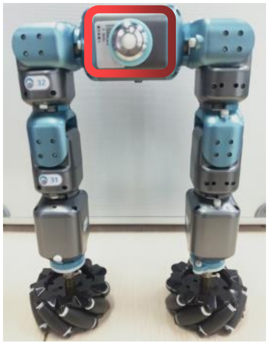


（3） Construction of the Configuration

Build a bionic biped robot, and add an auxiliary universal wheel to it above the F module. After building, place the robot flat on the ground.

Note: the auxiliary universal wheel is facing down, that is, aiming at the ground.

If the ID does not match, modify the "config.txt" file in the same directory.




(4) Example Execution

Steps: Turn on the F module switch, and connect the PC to the hotspot corresponding to the F module, which is generally biowinF_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotBipe/

Right-click to open the terminal and execute the Python command:

$ python3 WheelBipe_RunDemo.py

Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotBipe
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotBipe$ python3 WheelBipe_RunDemo.py
Connect To:10.10.100.254
connection succeeded
Read ID configuration file:/home/biowin/Samples_V2/Python/samples_robot/RobotBipe/config.txt
#Bipe

UP:31,30,20

Down:33,32,21

Enter any key to initialize....

SetMotorMode 20 1
SetMotorMode 21 1
Enter any key to run ....
```

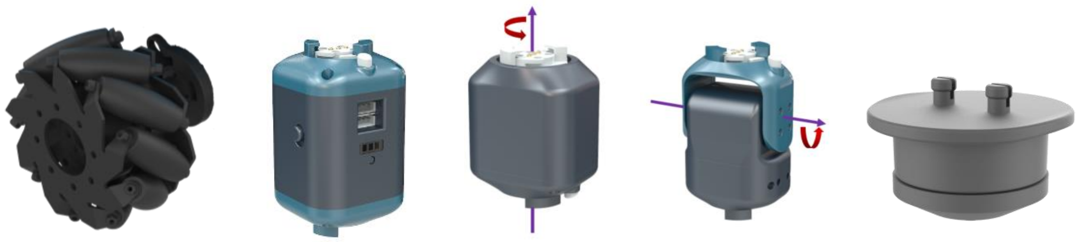
Run the program and press Enter to see the motion of the two-wheeled robot. The robot will go forward and backward, then turn left and right.

6.2.4 Control Experiment of Bi-Wheel Robot

(1) Realization function: Control the two-wheeled robot to move forward and backward, turn left and right, and stop through the keys of the keyboard.

(2) Hardware Preparation

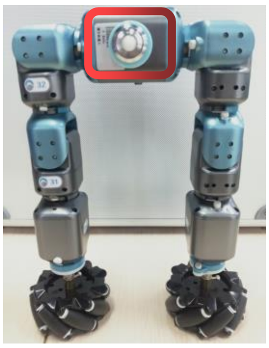
1 F module, 2 I modules, 4 T modules, 1 pair of Mecanum wheels and 1 auxiliary universal wheel.



(2) Construction of the Configuration

Build a bionic biped robot, and add an auxiliary universal wheel to it above the F module. After building, place the robot flat on the ground.

Note: the auxiliary universal wheel is facing down, that is, aiming at the ground. If the ID does not match, modify the "config.txt" file in the same directory.



(3) Example Execution

Steps: Turn on the F module switch, and connect the PC to the hotspot corresponding to the F module, which is generally biowinF_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotBipe/

Right-click to open the terminal and execute the Python command:

$ sudo python3 WheelBipe_RunControl.py

Operation result:

Run the program and input different numbers through the keyboard to control the movement of the two-wheeled robot. Note: The robot will only move if you press and hold the number key.

Key	Command
1	The robot initializes and starts to switch to two-wheel mode.
2	Two-wheeled robot moves forward.
3	The two-wheeled robot retreats.
4	Stop the movement.
5	The two-wheeled robot turns left.
6	The two-wheeled robot turns right.
0	Exit.

6.3 Experiment of Humanoid Biped Robot

6.3.1 Control Experiment of Humanoid Biped Robot

(1) Realization function: Control the humanoid biped robot to move forward, backward, slide left, slide right, turn left, turn right and dance through the keyboard

keys.

（1）Hardware Preparation

1 F module, 8 T modules, 4 orthogonal modules and 2 base plates.

（2）Construction of the Configuration

Build a humanoid biped robot and place the robot on the ground after it is built. If the ID does not match, modify the "config.txt" file in the same directory.

（3）Example Execution

Steps: Turn on the F module switch, and connect the PC to the hotspot corresponding to the F module, which is generally biowinF_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotBipe4T/

Right-click to open the terminal and execute the Python command:

$ sudo python3 Bipe4T_RunControl.py

Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotBipe4T
File  Edit  View  Search  Terminal  Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotBipe4T$ sudo python3 Bipe4T_RunControl.py
Connect To:10.10.100.254
connection succeeded
Read ID configuration file:/home/biowin/Samples_V2/Python/samples_robot/RobotBipe4T/config.txt
#Bipe-4T

UP:33,32,31,30

Down:37,36,35,34

SetMotorMode 33 0
SetMotorMode 32 0
SetMotorMode 31 0
SetMotorMode 30 0
SetMotorMode 37 0
SetMotorMode 36 0
SetMotorMode 35 0
SetMotorMode 34 0
Robot initialization setting is successful
Keyboard to control :
1 : Go Forward
2 : Go Back
3 : Dance
4 : Turn Left
5 : Turn Right
6 : Run Left
7 : Run Right
-----------------------
0 :Exit
```

Run the program and enter numbers on the keyboard to control the movement of the robot. The correspondence between numbers and motion is as follows:

Key	Command
1	The robot moves forward.
2	The robot steps back.
3	Robot dancing.
4	The robot turns left.
5	The robot turns right.
6	The robot walks to the left.
7	The robot walks to the right.
0	Exit.

6.3.2 Demo Experiment of Humanoid Biped Robot Motion

（1）Realization function: Control the humanoid biped robot to perform continuous motion of forward walking, backward walking, left walking, right walking, left turning, right turning and dancing.

（2）Hardware Preparation

1 F module, 8 T modules, 4 orthogonal modules and 2 base plates.

（3）Construction of the Configuration

Build a humanoid biped robot and place the robot on the ground after it is built. If the ID does not match, modify the config.txt file in the same directory.

（4）Example Execution

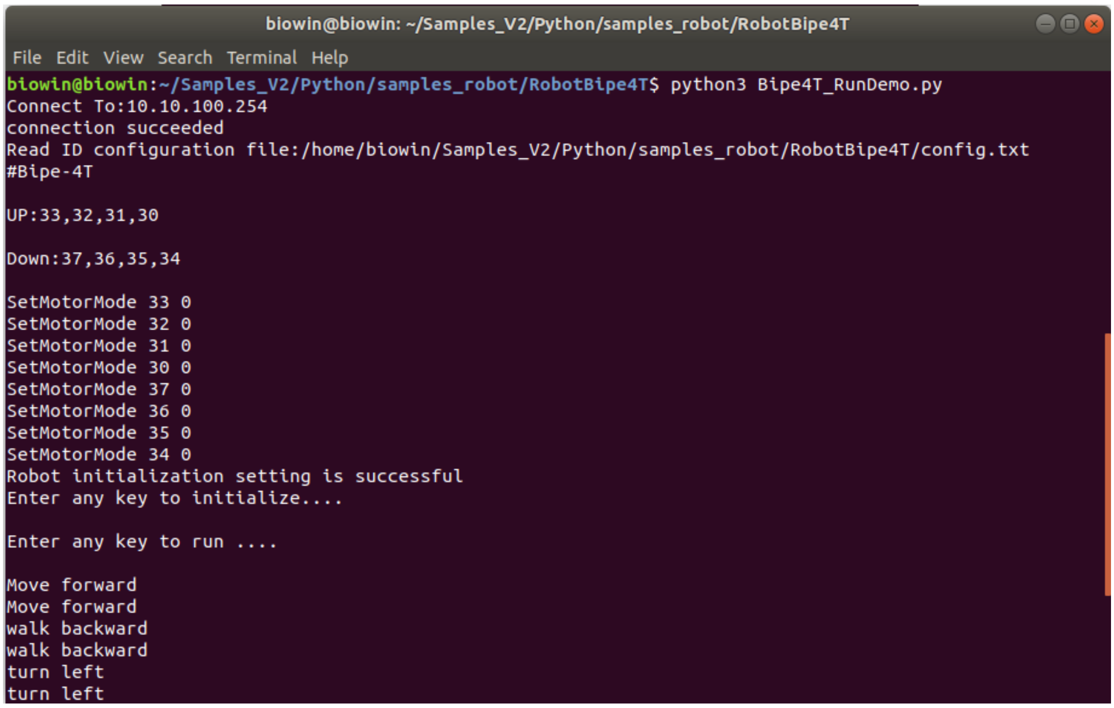
Steps: Turn on the F module switch, and connect the PC to the hotspot corresponding to the F module, which is generally biowinF_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotBipe4T/

Right-click to open the terminal and execute the Python command:

$ python3 Bipe4T_RunDemo.py

Operation result:



```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotBipe4T
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotBipe4T$ python3 Bipe4T_RunDemo.py
Connect To:10.10.100.254
connection succeeded
Read ID configuration file:/home/biowin/Samples_V2/Python/samples_robot/RobotBipe4T/config.txt
#Bipe-4T

UP:33,32,31,30

Down:37,36,35,34

SetMotorMode 33 0
SetMotorMode 32 0
SetMotorMode 31 0
SetMotorMode 30 0
SetMotorMode 37 0
SetMotorMode 36 0
SetMotorMode 35 0
SetMotorMode 34 0
Robot initialization setting is successful
Enter any key to initialize....

Enter any key to run ....

Move forward
Move forward
walk backward
walk backward
turn left
turn left
```

Running the program, the humanoid biped robot moves according to the preset movements.

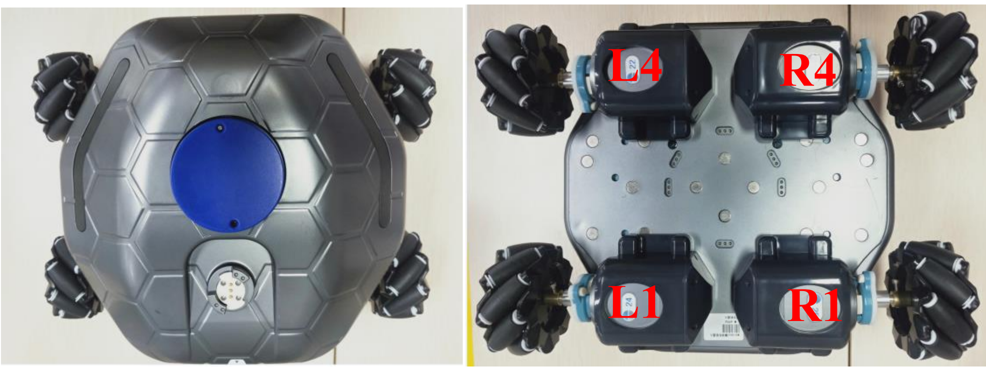
6.4 Mobile Vehicle Experiment

In the mobile car, we provide users with a variety of configurations of the car. As follows:

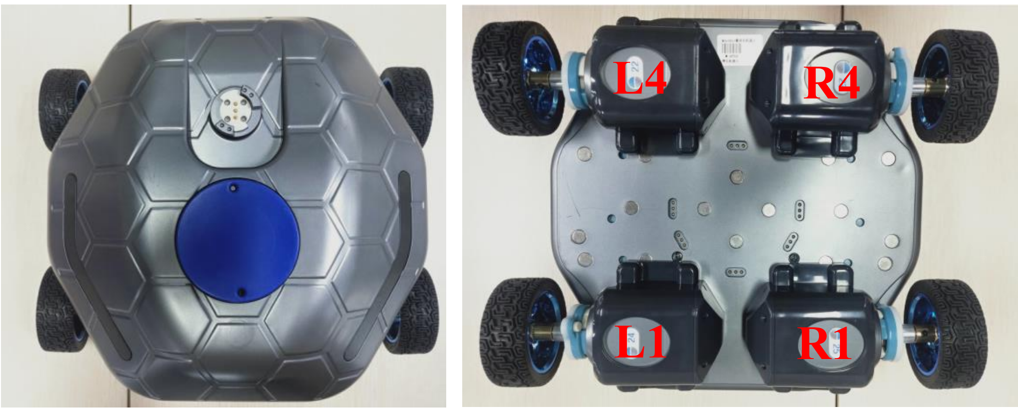
Two-wheeled car:

Three-wheeled car:

Four Mecanum wheels car:



Four rubber wheels car:



Through the configuration diagram of the mobile car above, I believe that you already have a certain understanding of various types of cars. Let's try to run and control them through the program. Here we only take the ordinary four-rubber wheel car as an example. Interested friends can try other configuration cars by themselves after completing the four-rubber wheel experiment.

6.4.1 Mobile Vehicle Basic Experiment

6.4.1.1 Control Experiment of Four Rubber Wheels

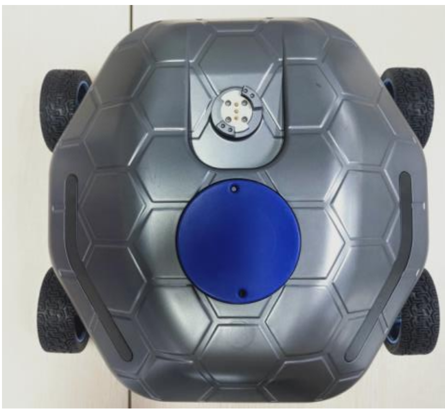
（1）Realization function: Control the movement of the mobile car through the keyboard (the distance, speed, etc. can be set).

（2）Hardware Preparation

1 M module, 4 rubber wheels, 4 I modules and 4 I module wrapping piece.

（3）Construction of the Configuration

Put the four I modules into the I module wrapping piece, and then connect a rubber wheel to each I module. After the assembly is completed, connect it to the M module. As shown in the figure:



（4）Example Execution

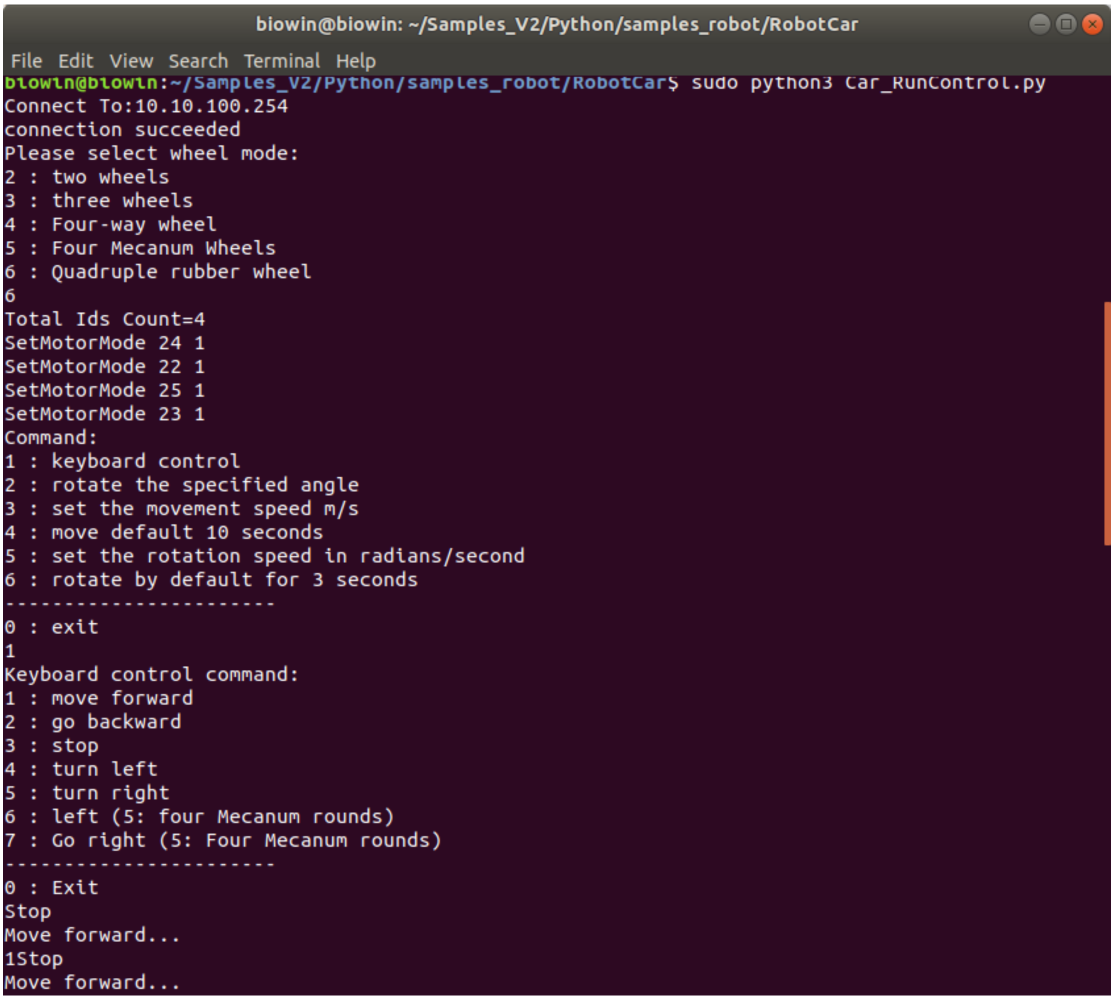
Turn on the M module switch, and connect the PC to the hotspot corresponding to the M module, which is generally biowinM_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotCar/

Right-click to open the terminal and execute the Python command:

$ sudo python3 Car_RunControl.py

Operation result:



Note: Run the program, because we are building a four-rubber-wheeled car here, so you must first enter "6" on the keyboard to select the corresponding car configuration. Of course, if you use other car types, you can enter the corresponding number. After selecting the corresponding configuration, the following command selection will appear:

Key	Command
1	Keyboard control.
2	Rotate the specified angle.

Key	Command
3	Set the movement speed: m/s.
4	Move default 10 seconds.
5	Set the rotation speed: radians/second.
6	Rotate by default for 3 seconds.
0	Exit.

From the screenshot of the operation, we entered the number "1", which is to control the movement of the car on the keyboard. Of course, you can choose other operations as well. After selecting "Keyboard Control", the car movement can be controlled by pressing the numbers on the keyboard again. The correspondence between buttons and control commands is as follows:

Key	Command
1	Go forward
2	Go backward
3	Stop
4	Turn left
5	Turn right
6	Go left
7	Go right
0	Exit

Note: This is the control command for the four rubber-wheeled car. The operation method of other configurations is similar to the picture, but some buttons are different, which will be prompted at runtime.

6.4.1.2 Demo Experiment of Four Rubber Wheels Motion

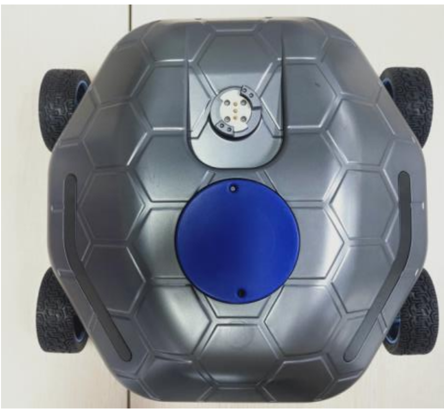
(1) Realization function: Control the moving car to perform continuous motion of forward, backward, left turn and right turn.

(2) Hardware Preparation

1 M module, 4 rubber wheels, 4 I modules and 4 I module wrapping piece.

(3) Construction of the Configuration

Put the four I modules into the I module wrapping piece, and then connect a rubber wheel to each I module. After the assembly is completed, connect it to the M module. As shown in the figure:



(4) Example Execution

Turn on the M module switch, and connect the PC to the hotspot corresponding to the M module, which is generally biowinM_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotCar/

Right-click to open the terminal and execute the Python command:

$ sudo python3 Car_RunDemo.py

Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotCar
File  Edit  View  Search  Terminal  Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotCar$ sudo python3 Car_RunControl.py
Connect To:10.10.100.254
connection succeeded
Please select wheel mode:
2 : two wheels
3 : three wheels
4 : Four-way wheel
5 : Four Mecanum Wheels
6 : Quadruple rubber wheel
6
Total Ids Count=4
SetMotorMode 24 1
SetMotorMode 22 1
SetMotorMode 25 1
SetMotorMode 23 1
Command:
1 : keyboard control
2 : rotate the specified angle
3 : set the movement speed m/s
4 : move default 10 seconds
5 : set the rotation speed in radians/second
6 : rotate by default for 3 seconds
----------------------
0 : exit
1
Keyboard control command:
1 : move forward
2 : go backward
3 : stop
4 : turn left
5 : turn right
6 : left (5: four Mecanum rounds)
7 : Go right (5: Four Mecanum rounds)
----------------------
0 : Exit
Stop
Move forward...
1Stop
Move forward...
```

Running the program, the four rubber-wheeled car will go forward and backward, then turn left and right. Tip: If you feel that the car movement time is short, you can modify the code yourself.

Note: In the Demo experiment of the car, you also need to use root privileges, that is, add sudo and then execute the python command.

6.4.2 Sensor Application Example

6.4.2.1 Obstacle Avoidance Experiment

(1) Realization function: The ultrasonic sensor starts to detect obstacles when the car is moving. If there is no obstacle, the car will cycle through a square route (go straight first and then turn right). If an obstacle is detected, the car will turn left to

avoid the obstacle. , to realize the function of ultrasonic obstacle avoidance.

（2）Hardware Preparation

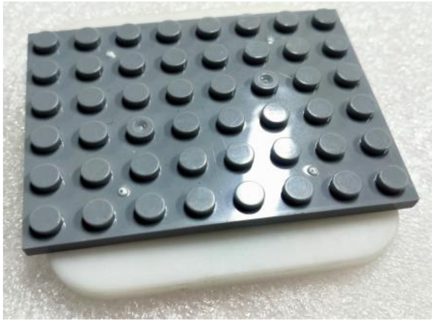
1 M module, 4 rubber wheels, 4 I modules, 4 I module wrapping piece, 1 sensor fixing plate of the M module, 1 fixing plate of the sensor main control board of the M module, 1 ultrasonic sensor module and 1 main control board.





The sensor fixing plate of the M module is used for fixing, and the ultrasonic and line tracking sensors are fixed in front of the M module (fixed on the I module wrapping piece).

The sensor main control board fixing plate of the M module is used for fixing, and the sensor main control board is fixed at the bottom of the M module.



(3) Construction of the Configuration

Put the four I modules into the I module wrapping piece, and then connect a rubber wheel to each I module. After completion, connect it to the M module, and connect the sensor and main control board to the car.

Note: The ultrasonic module is connected to the D3 port of the main control board, and the main control board is connected to the upper right port of the M module. If you forget, please refer to the introduction of the M module in the sensor control example in Section 5. As shown below:

(4) Example Execution

Turn on the M module switch, and connect the PC to the hotspot corresponding to the M module, which is generally biowinM_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotCar/SensorsApp/

Right-click to open the terminal and execute the Python command:

$ sudo python3 01.AvoidObstacle.py

Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotCar/SensorsApp
File  Edit  View  Search  Terminal  Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotCar/SensorsApp$ sudo python3 01.AvoidObstacle.py
[sudo] password for biowin:
Connect To:10.10.100.254
connection succeeded
Total Ids Count=4

Please select wheel mode:
2 : Two Wheels
3 : Three Wheels
4 : Four-way Wheels
5 : Four Mecanum Wheels
6 : Four rubber wheels
CTRL-C to quit

6
SetMotorMode 24 1
SetMotorMode 22 1
SetMotorMode 25 1
SetMotorMode 23 1
Ultrasound returns data:1135.4cm
Move forward...
Ultrasound returns data:11.0cm
There is an obstacle ahead, turn left to avoid it!!
Turn left...
Move forward...
Ultrasound returns data:1135.4cm
Move forward...
Ultrasound returns data:1135.4cm
Move forward...
```

After running the program, the car moves forward, and when the sensor detects an obstacle, it turns to avoid it.

6.4.2.2 Line Patrol Experiment

(1) Realization function: The tracking module detects the black line on the ground when the trolley moves, so as to control the trolley to follow the black line on the ground to move (note that the arc of the black line cannot be too large, otherwise it will affect the line patrol effect).

(2) Hardware Preparation

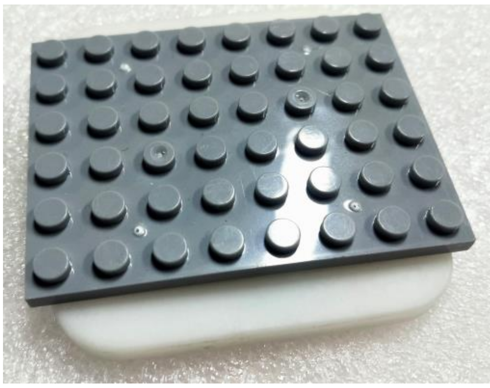
1 M module, 4 rubber wheels, 4 I modules, 4 I module wrapping pieces, 1 sensor fixing plate of the M module, 1 fixing plate of the sensor main control board of the M module, 1 tracking module, 1 main control board and 1 roll of black line patrol tape.





The sensor fixing plate of the M module is used for fixing, and the ultrasonic and line tracking sensors are fixed in front of the M module.

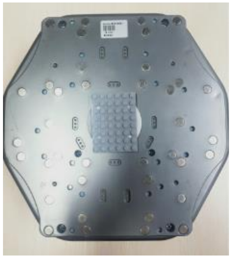
The sensor main control board fixing plate of the M module is used for fixing, and the sensor main control board is fixed at the bottom of the M module.

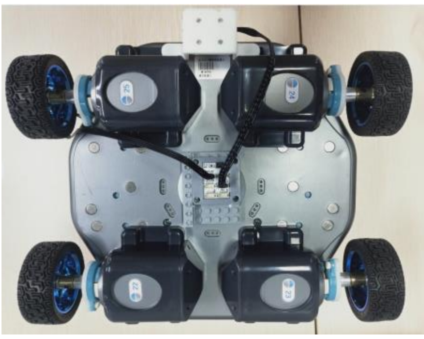


(3) Construction of the Configuration

Put the four I modules into the I module wrapping piece, and then connect a rubber wheel to each I module. After completion, connect it to the M module, and connect the sensor and main control board to the car.

Note: The tracking module is connected to the D3 port of the main control board, and the main control board is connected to the upper right corner of the M module. If you forget, please refer to the introduction of the M module in the sensor control example in Section 5. As shown below:





(4) Example Execution

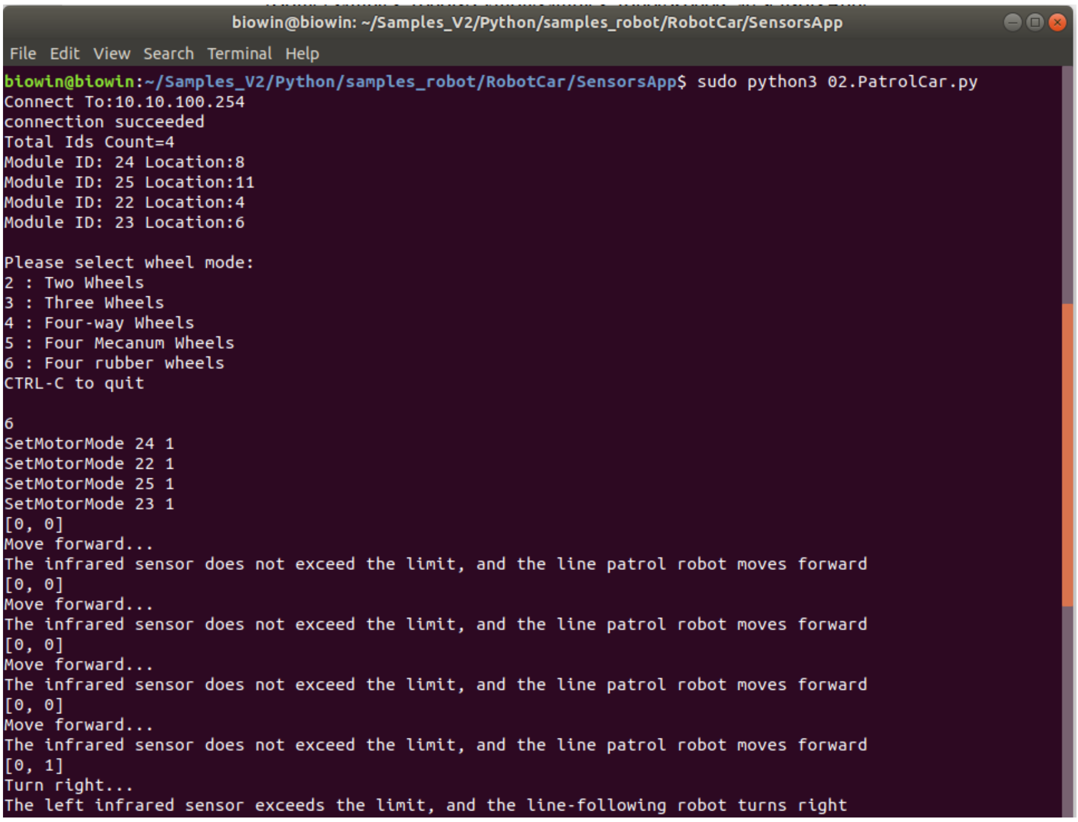
Turn on the M module switch, and connect the PC to the hotspot corresponding to the M module, which is generally biowinM_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotCar/SensorsApp/

Right-click to open the terminal and execute the Python command:

$ sudo python3 02.PatrolCar.py

Operation result:



```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotCar/SensorsApp
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotCar/SensorsApp$ sudo python3 02.PatrolCar.py
Connect To:10.10.100.254
connection succeeded
Total Ids Count=4
Module ID: 24 Location:8
Module ID: 25 Location:11
Module ID: 22 Location:4
Module ID: 23 Location:6

Please select wheel mode:
2 : Two Wheels
3 : Three Wheels
4 : Four-way Wheels
5 : Four Mecanum Wheels
6 : Four rubber wheels
CTRL-C to quit

6
SetMotorMode 24 1
SetMotorMode 22 1
SetMotorMode 25 1
SetMotorMode 23 1
[0, 0]
Move forward...
The infrared sensor does not exceed the limit, and the line patrol robot moves forward
[0, 0]
Move forward...
The infrared sensor does not exceed the limit, and the line patrol robot moves forward
[0, 0]
Move forward...
The infrared sensor does not exceed the limit, and the line patrol robot moves forward
[0, 0]
Move forward...
The infrared sensor does not exceed the limit, and the line patrol robot moves forward
[0, 1]
Turn right...
The left infrared sensor exceeds the limit, and the line-following robot turns right
```

The car will move along the black line according to the detection result of the tracking module. If there is no black line ahead, the car will back up, and if it deviates from the black line, it will automatically adjust by turning left or right.

6.4.2.3 Experiment that Cannot Go Out of the Designated Area

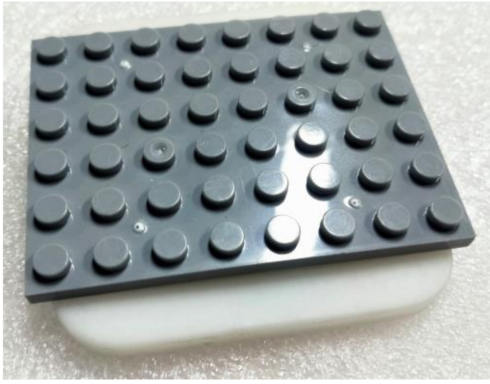
(1) Realization function: Stick a closed space on the ground with black line patrol tape, put the car in the closed space to move, so that it cannot go out of the closed space.

(2) Hardware Preparation

1 M module, 4 rubber wheels, 4 I modules, 4 I module wrapping pieces, 1 sensor fixing plate of the M module, 1 fixing plate of the sensor main control board of the M module, 1 tracking module, 1 main control board and 1 roll of black line patrol tape.

The sensor fixing plate of the M module is used for fixing, and the ultrasonic and line tracking sensors are fixed in front of the M module.

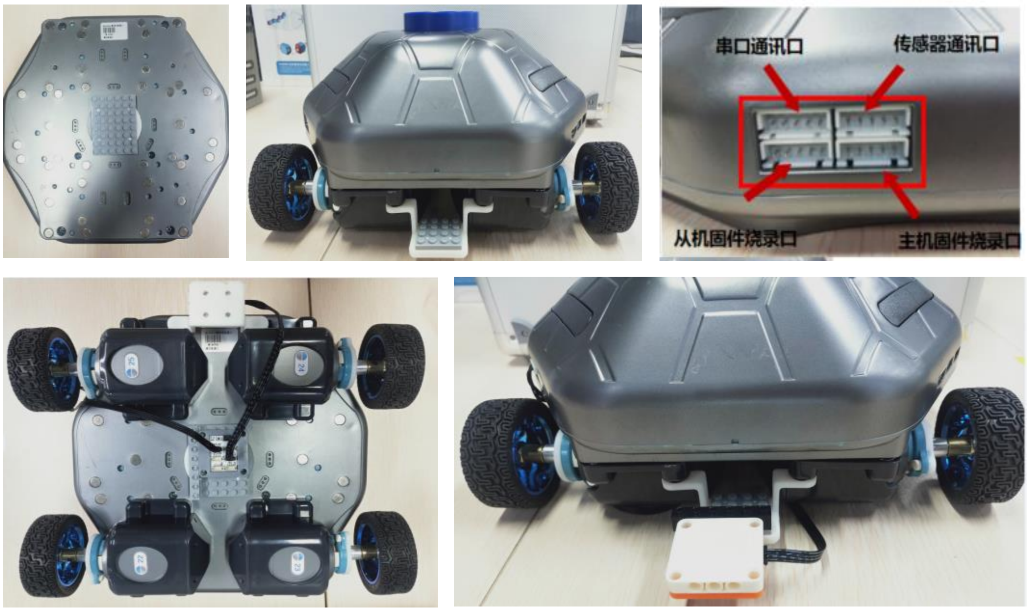
The sensor main control board fixing plate of the M module is used for fixing, and the sensor main control board is fixed at the bottom of the M module.



（3）Construction of the Configuration

Put the four I modules into the I module wrapping piece, and then connect a rubber wheel to each I module. After completion, connect it to the M module, and connect the sensor and main control board to the car.

Note: The tracking module is connected to the D3 port of the main control board, and the main control board is connected to the upper right corner of the M module. If you forget it, please refer to the introduction of the M module in the sensor control example in Section 5. As shown below:



（4）Example Execution

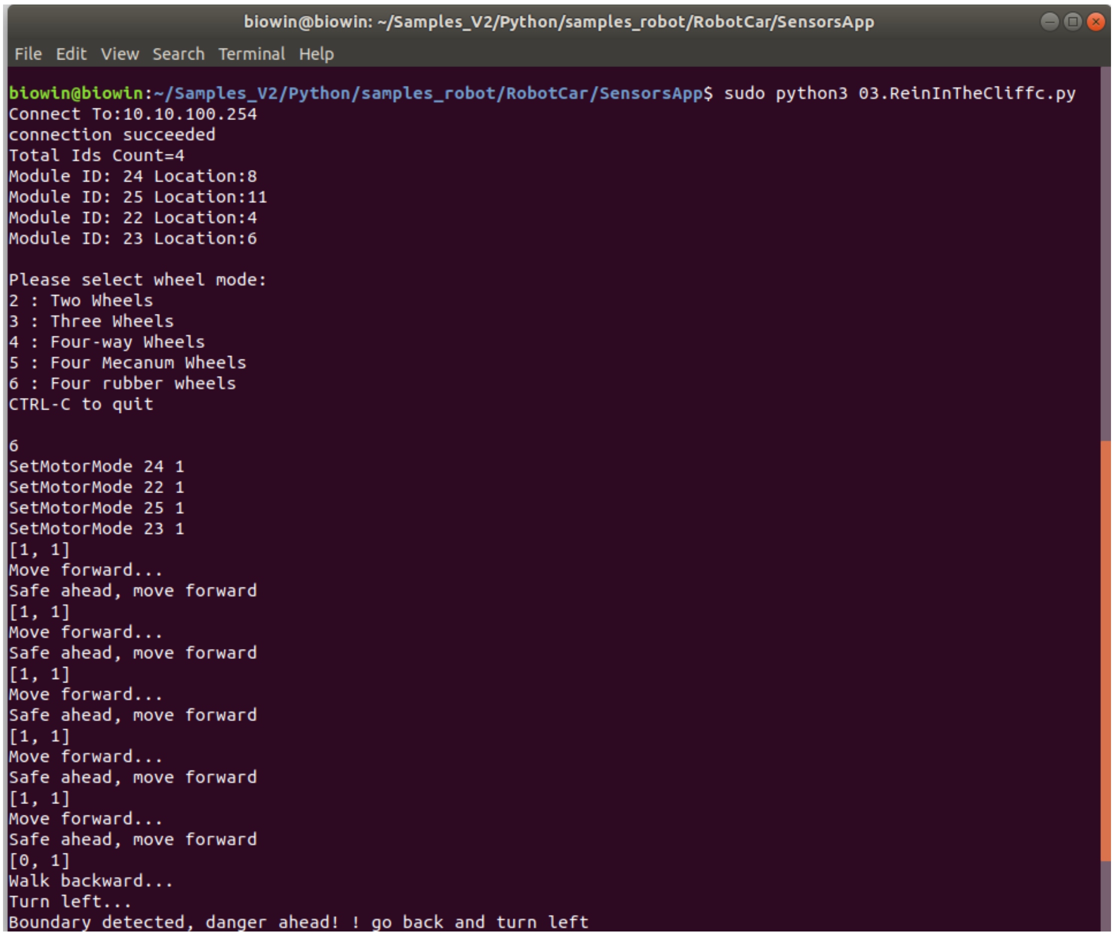
Turn on the M module switch, and connect the PC to the hotspot corresponding to the M module, which is generally biowinM_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotCar/SensorsApp/

Right-click to open the terminal and execute the Python command:

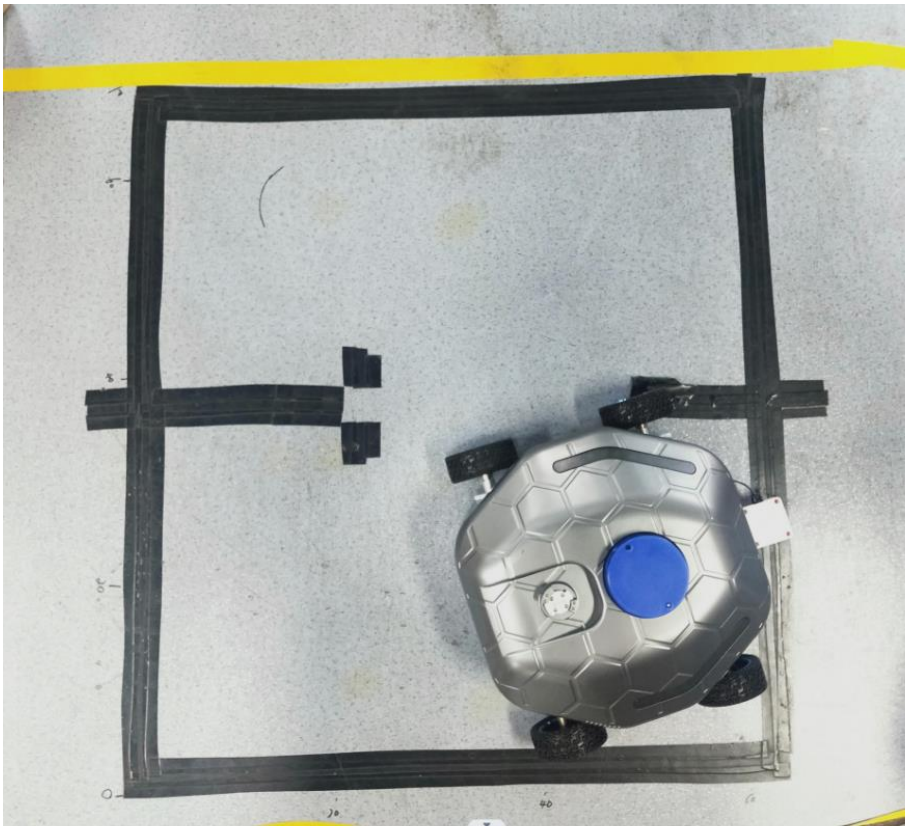
$ sudo python3 03.ReinInTheCliffc.py

Operation result:



```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotCar/SensorsApp
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotCar/SensorsApp$ sudo python3 03.ReinInTheCliffc.py
Connect To:10.10.100.254
connection succeeded
Total Ids Count=4
Module ID: 24 Location:8
Module ID: 25 Location:11
Module ID: 22 Location:4
Module ID: 23 Location:6

Please select wheel mode:
2 : Two Wheels
3 : Three Wheels
4 : Four-way Wheels
5 : Four Mecanum Wheels
6 : Four rubber wheels
CTRL-C to quit

6
SetMotorMode 24 1
SetMotorMode 22 1
SetMotorMode 25 1
SetMotorMode 23 1
[1, 1]
Move forward...
Safe ahead, move forward
[1, 1]
Move forward...
Safe ahead, move forward
[1, 1]
Move forward...
Safe ahead, move forward
[1, 1]
Move forward...
Safe ahead, move forward
[1, 1]
Move forward...
Safe ahead, move forward
[0, 1]
Walk backward...
Turn left...
Boundary detected, danger ahead! ! go back and turn left
```

Post a black quad on the ground, put the car in it, run the program, and the car starts to move. When the sensor detects the black line, the car backs up and turns left.



6.5 Experiment of Vehicle-Mounted Robot Arm

In the previous content, we know how to build a mobile car and a robotic arm. Now, we combine them together to build a car with a robotic arm, so that the function of moving and grabbing objects can be realized at the same time.

6.5.1 Control Experiment of Vehicle-Mounted Robot Arm

(1) Realization function: Control the car through the keyboard to move forward, backward, left and right and the movement of each joint of the mechanical arm.

(2) Hardware Preparation

A car with four rubber wheels and a robotic arm (that is, a robotic arm with five degrees of freedom configured as I T T T I G).

(3) Construction of the Configuration

Connect the robotic arm with five degrees of freedom (I T T T I G) to the car If the actual joint ID is different from the example, you can modify it in the "config.txt" file in the same directory. As the picture shows:

（4）Example Execution

Turn on the M module switch, and connect the PC to the hotspot corresponding to the M module, which is generally biowinM_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotCarArm/

Right-click to open the terminal and execute the Python command:

$ sudo python3 CarArm_RunControl.py

Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotCarArm
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotCarArm$ sudo python3 CarArm_RunControl.py
Connect To:10.10.100.254
connection succeeded
Total Ids Count=10
Module ID: 24 Location:8
Module ID: 20 Location:9
Module ID: 21 Location:9
Module ID: 25 Location:11
Module ID: 30 Location:9
Module ID: 31 Location:9
Module ID: 32 Location:9
Module ID: 40 Location:9
Module ID: 22 Location:4
Module ID: 23 Location:6

Please select wheel mode:
2 : Two Wheels
3 : Three Wheels
4 : Four-way Wheels
5 : Four Mecanum Wheels
6 : Four rubber wheels
CTRL-C to quit

6
SetMotorMode 24 1
SetMotorMode 22 1
SetMotorMode 25 1
SetMotorMode 23 1
Read ID configuration file:/home/biowin/Samples_V2/Python/samples_robot/RobotCarArm/config.txt
['#Manipulator-ForCar\n', '20,30,31,32,21,40\n']
The robot module information is imported correctly!
Total Ids Count=10
The module ID input by the robotic arm is the same as the module ID obtained by scanning
```

```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotCarArm
File Edit View Search Terminal Help
The robot module information is imported correctly!
Total Ids Count=10
The module ID input by the robotic arm is the same as the module ID obtained by scanning
bindIds: 6
id: 20
SetMotorMode 20 0
id: 30
SetMotorMode 30 0
id: 31
SetMotorMode 31 0
id: 32
SetMotorMode 32 0
id: 21
SetMotorMode 21 0
id: 40
SetMotorMode 40 0
The robot is moving to the starting point
SetMotorMode 20 0
SetMotorMode 30 0
SetMotorMode 31 0
SetMotorMode 32 0
SetMotorMode 21 0
SetMotorMode 40 0

Please enter: mobile car control command
i : go forward , : go backward
j : turn left l : turn right
k : stop
h : Left walk (5 : Four Mecanum rounds)
; : Go right (5 : Four Mecanum rounds)


Keyboard control robot joint angle: increase/decrease
I joint 1 : q/a
T joint 2 : w/s
T joint 3 : e/d
T joint 4 : r/f
I joint 5 : t/g
G gripper : y/h
Read the current angle of the robot: 1
CTRL-C to quit

Four parallel rubber wheels go forward...
Stop
Four parallel rubber wheels turn left...
Stop
Four parallel rubber wheels turn right...
Stop
Four parallel rubber wheels go backward...
Stop
```

After running the program, you can control the car through the keys on the keyboard, and you can also control the robotic arm on the car.

Note: Because the base of the trolley has four I modules, and the robotic arm also has two I modules. At this time, if there are two I modules with the same number, there may be problems. It is recommended to set the numbers of all I modules to be different. At the same time the code is buggy and it is easy to identify multiple modules.

6.5.2 Demo Experiment of Vehicle-Mounted Robot Arm Motion

（1）Realization function: Realize the combined motion of controlling the mobile car and the robotic arm to grasp objects.

（2）Hardware Preparation

A car with four rubber wheels and a robotic arm (that is, a robotic arm with five degrees of freedom configured as I T T T I G).

（3）Construction of the Configuration

Connect the robotic arm with five degrees of freedom (I T T T I G) to the car If the actual joint ID is different from the example, you can modify it in the "config.txt" file in the same directory. As the picture shows:

（4）Example Execution

Turn on the M module switch, and connect the PC to the hotspot corresponding to the M module, which is generally biowinM_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotCarArm/

Right-click to open the terminal and execute the Python command:

$ python3 CarArm_RunDemo.py

Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotCarArm
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotCarArm$ python3 CarArm_RunDemo.py
Connect To:10.10.100.254
connection succeeded
Total Ids Count=10
Module ID: 24 Location:8
Module ID: 20 Location:9
Module ID: 21 Location:9
Module ID: 25 Location:11
Module ID: 30 Location:9
Module ID: 31 Location:9
Module ID: 32 Location:9
Module ID: 40 Location:9
Module ID: 22 Location:4
Module ID: 23 Location:6

Please select wheel mode:
2 : Two Wheels
3 : Three Wheels
4 : Four-way Wheels
5 : Four Mecanum Wheels
6 : Four rubber wheels
CTRL-C to quit

6
SetMotorMode 24 1
SetMotorMode 22 1
SetMotorMode 25 1
SetMotorMode 23 1
Read ID configuration file:/home/biowin/Samples_V2/Python/samples_robot/RobotCarArm/config.txt
['#Manipulator-ForCar\n', '20,30,31,32,21,40\n']
The robot module information is imported correctly!
Total Ids Count=10
The module ID input by the robotic arm is the same as the module ID obtained by scanning
```

```
bindIds: 6
id: 20
SetMotorMode 20 0
id: 30
SetMotorMode 30 0
id: 31
SetMotorMode 31 0
id: 32
SetMotorMode 32 0
id: 21
SetMotorMode 21 0
id: 40
SetMotorMode 40 0
The robot is moving to the starting point
SetMotorMode 20 0
SetMotorMode 30 0
SetMotorMode 31 0
SetMotorMode 32 0
SetMotorMode 21 0
SetMotorMode 40 0
Four parallel rubber wheels go forward...
Four parallel rubber wheels go backward...
Four parallel rubber wheels turn left...
Four parallel rubber wheels turn right...
Stop
```

After running the program, the trolley first walks forward, then backward, then turns left and then right, while the robotic arm grabs objects.

6.6 Climbing Robot Experiment

6.6.1 Control Experiment of Climbing Robot

（1）Realization function: Control the climbing robot to climb up, climb down, open and close the claws through the keyboard.

（2）Hardware Preparation

1 F module, 2 T modules and 2 G modules.

（3）Construction of the Configuration

Connect each joint module as shown below.

（4）Example Execution

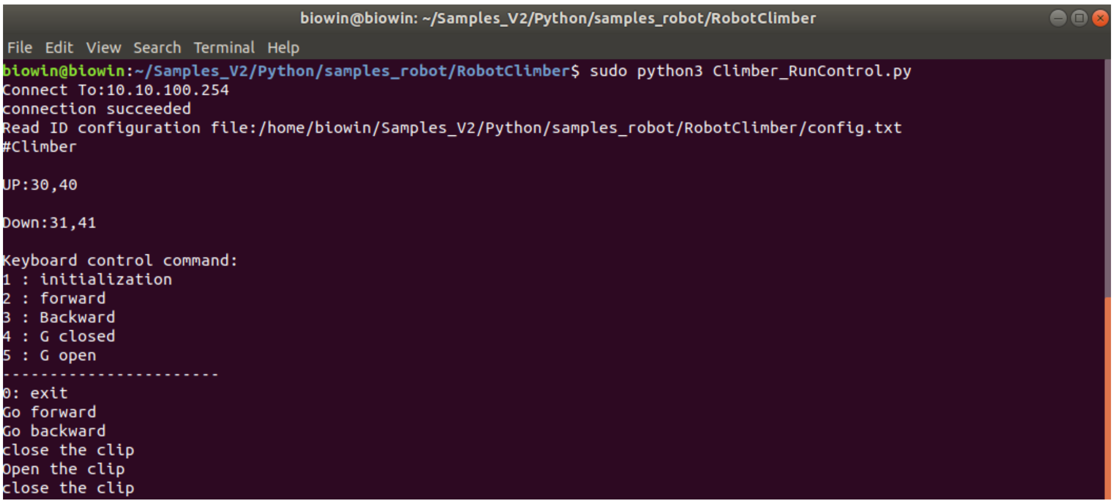
Steps: Turn on the F module switch, and connect the PC to the hotspot corresponding to the F module, which is generally biowinF_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotClimber/

Right-click to open the terminal and execute the Python command:

$ sudo python3 Climber_RunControl.py

Operation result:



After running the program, control the movement of the robot on the keyboard. The control commands corresponding to the keyboard are as follows:

Key	Action
1	Initialization
2	Forward (climb up)
3	Backward(climb down)
4	G closed
5	G open
Ctrl+C	Exit

Note: If the expected result does not appear, there may be a problem of no initialization. The solution is to open the joint module configuration in the Bw-Studio Suite software, set all joints to position mode, change the position angle to 0, and then click "Execute" " button, run the python command again after the initialization is complete.

6.6.2 Demo Experiment of Climbing Robot Motion

(1) Realization function: Control the climbing robot to perform the combined movement of opening and closing the gripper, climbing up and down.

(2) Hardware Preparation

1 F module, 2 T modules and 2 G modules.

(3) Construction of the Configuration

Connect each joint module as shown below.

(4) Example Execution

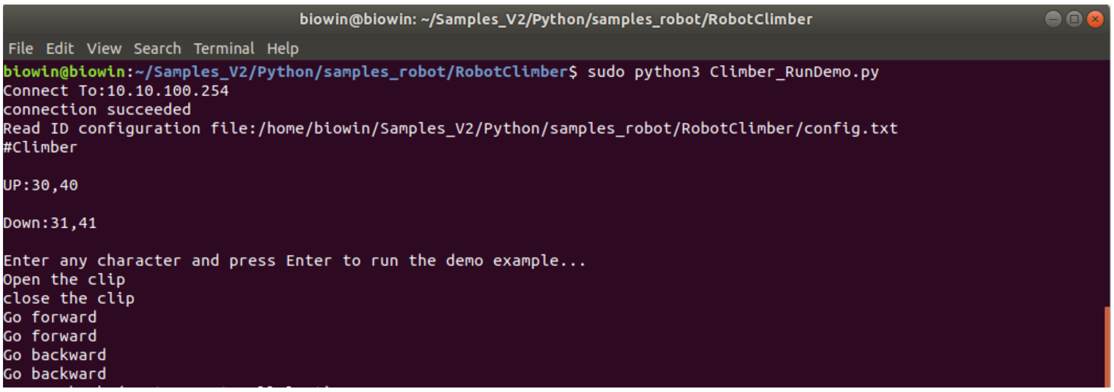
Steps: Turn on the F module switch, and connect the PC to the hotspot corresponding to the F module, which is generally biowinF_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotClimber/

Right-click to open the terminal and execute the Python command:

```
$	python3 Climber_RunDemo.py
```

Operation result:



After running the program, the climbing robot automatically executes the preset program.

Note: If the expected result does not appear, there may be a problem of no initialization. The solution is to open the joint module configuration in the Bw-Studio Suite software, set all joints to position mode, change the position angle to 0, and then click "Execute" " button, run the python command again after the initialization is complete.

6.7 Experiment of Snake-Like Robot

6.7.1 Control Experiment of Snake-Like Robot

(1) Realization function: Control the forward and backward, left and right turns of the snake-like robot through the keyboard.

(2) Hardware Preparation

1 F module, 5 T modules, 6 auxiliary moving wheels (5 small and 1 large).

(3) Construction of the Configuration

Connect the F module and the T module, and add auxiliary moving wheels to each module at the same time. Build as shown:

(4) Example Execution

Steps: Turn on the F module switch, and connect the PC to the hotspot corresponding to the F module, which is generally biowinF_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotSnake/

Right-click to open the terminal and execute the Python command:

$ sudo python3 Snake_RunControl.py

Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotSnake
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotSnake$ sudo python3 Snake_RunControl.py
Connect To:10.10.100.254
connection succeeded
Read ID configuration file:/home/biowin/Samples_V2/Python/samples_robot/RobotSnake/config.txt
#Snake

DOWN:30,31,32,33,34

Enter any character and press Enter to start initialization...
Stop
Command:
1 : control
2 : set rate parameter
3 : set amplitude
4 : set frequency
----------------------
0 : exit
Please input:1
Keyboard control command:
1 : forward
2 : back
3 : stop
4 : turn left
5 : turn right
----------------------
0 : exit
Go forward
Stop
Go backward
Stop
Turn left
Stop
Turn right
Stop
```

After running the program, enter numbers on the keyboard to control the movement of the snake-like robot. The control commands corresponding to the numbers on the keyboard are as follows:

Key	Command
1	Forward
2	Backward
3	Stop
4	Turn left
5	Turn right
0	Exit

Tip: Pay attention to distinguish the position of the head and tail of the snake-like robot.

6.7.2 Demo Experiment of Snake-Like Robot Motion

（1）Realization function: Control the combined movement of the snake-like robot forward, backward, left and right turns.

（2）Hardware Preparation

1 F module, 5 T modules, 6 auxiliary moving wheels (5 small and 1 large).

（3）Construction of the Configuration

Connect the F module and the T module, and add auxiliary moving wheels to each module at the same time. Build as shown:

(4) Example Execution

Steps: Turn on the F module switch, and connect the PC to the hotspot corresponding to the F module, which is generally biowinF_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotSnake/

Right-click to open the terminal and execute the Python command:

$ python3 Snake_RunDemo.py

Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotSnake
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotSnake$ python3 Snake_RunDemo.py
Connect To:10.10.100.254
connection succeeded
Read ID configuration file:/home/biowin/Samples_V2/Python/samples_robot/RobotSnake/config.txt
#Snake

DOWN:30,31,32,33,34

Enter any character and press Enter to initialize...
Stop
Enter any character and press Enter to run the demo example...
Go forward
Go backward
Turn left
Turn right
Stop
```

After running the program, press Enter, and the snake-like robot will move according to the preset actions.

6.8 Modular Manipulator Experiment

6.8.1 Control Experiment of Manipulator

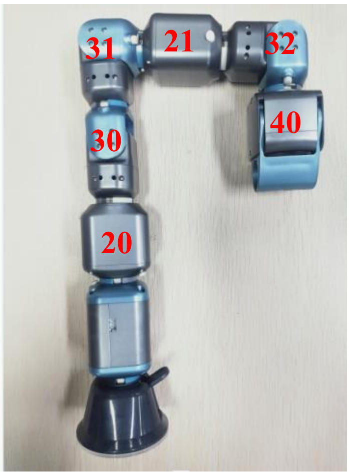
（1）Realization function: Control the movement of each joint of the robotic arm and control the movement of the robotic arm on the Cartesian coordinate system through the keyboard.

（2）Hardware Preparation

1 F module, 2 I modules, 3 T modules and 1 G module.

（3）Construction of the Configuration

Connect each module according to the picture below:



（4）Example Execution

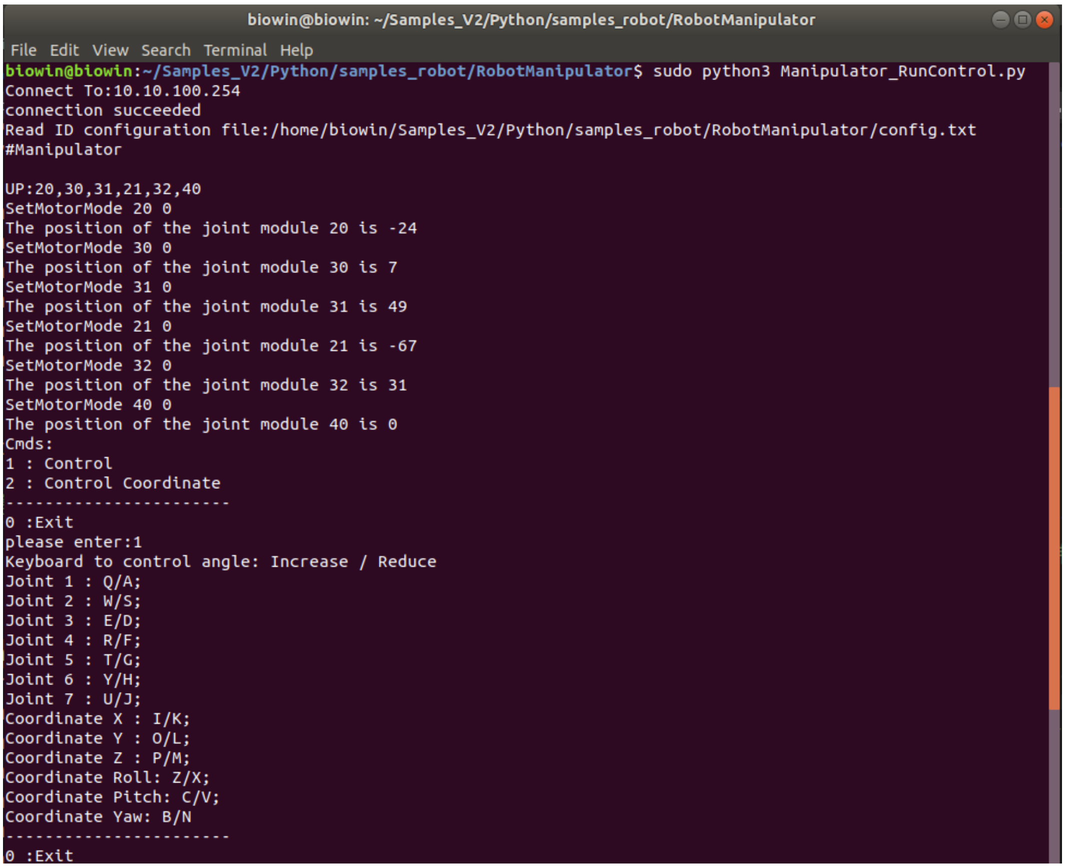
Steps: Turn on the F module switch, and connect the PC to the hotspot corresponding to the F module, which is generally biowinF_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotManipulator/

Right-click to open the terminal and execute the Python command:

$ sudo python3 Manipulator_RunControl.py

Operation result:



```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotManipulator
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotManipulator$ sudo python3 Manipulator_RunControl.py
Connect To:10.10.100.254
connection succeeded
Read ID configuration file:/home/biowin/Samples_V2/Python/samples_robot/RobotManipulator/config.txt
#Manipulator

UP:20,30,31,21,32,40
SetMotorMode 20 0
The position of the joint module 20 is -24
SetMotorMode 30 0
The position of the joint module 30 is 7
SetMotorMode 31 0
The position of the joint module 31 is 49
SetMotorMode 21 0
The position of the joint module 21 is -67
SetMotorMode 32 0
The position of the joint module 32 is 31
SetMotorMode 40 0
The position of the joint module 40 is 0
Cmds:
1 : Control
2 : Control Coordinate
----------------------
0 :Exit
please enter:1
Keyboard to control angle: Increase / Reduce
Joint 1 : Q/A;
Joint 2 : W/S;
Joint 3 : E/D;
Joint 4 : R/F;
Joint 5 : T/G;
Joint 6 : Y/H;
Joint 7 : U/J;
Coordinate X : I/K;
Coordinate Y : O/L;
Coordinate Z : P/M;
Coordinate Roll: Z/X;
Coordinate Pitch: C/V;
Coordinate Yaw: B/N
----------------------
0 :Exit
```

After running the program, you can see that in this case, we can choose the control joint angle and the control space coordinate. First, we enter the number "1" to select the control joint angle, and then the letter button corresponding to the control command will appear below. Pressing these letters on the keyboard controls the corresponding individual joints. details as follows:

Key	Command
Q/A	Control joint 1
W/S	Control joint 2
E/D	Control joint 3
R/F	Control joint 4

Key	Command
T/G	Control joint 5
Y/H	Control joint 6
U/J	Control gripper

```
please enter:2
Keyboard to control coordinate: Increase / Reduce
Coordinate X : I/K;
Coordinate Y : O/L;
Coordinate Z : P/M;
Coordinate Roll: Z/X;
Coordinate Pitch: C/V;
Coordinate Yaw: B/N
-----------------------
0 :Exit
```

Earlier, we chose to enter the number "1" to open the command interface for controlling joints. Now we enter "0" to exit and return to the start interface, enter "2" to select the control space coordinates. This space coordinate actually takes the center of the base as the origin, and the end of the manipulator is regarded as a moving point. The control space coordinate is actually the coordinate that controls this moving point.

6.8.2 Demo Experiment of Manipulator Motion

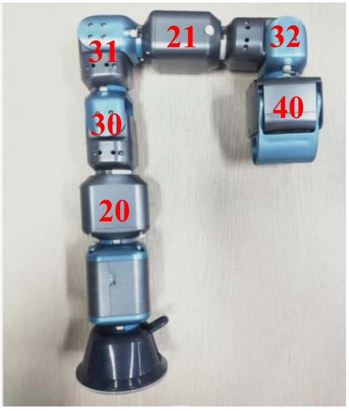
(1) Realization function: The control manipulator imitates the palletizing manipulator to grasp, move, and place the objects in combination.

(2) Hardware Preparation

1 F module, 2 I modules, 3 T modules and 1 G module.

（3）Construction of the Configuration

Connect each module according to the picture below:



（4）Example Execution

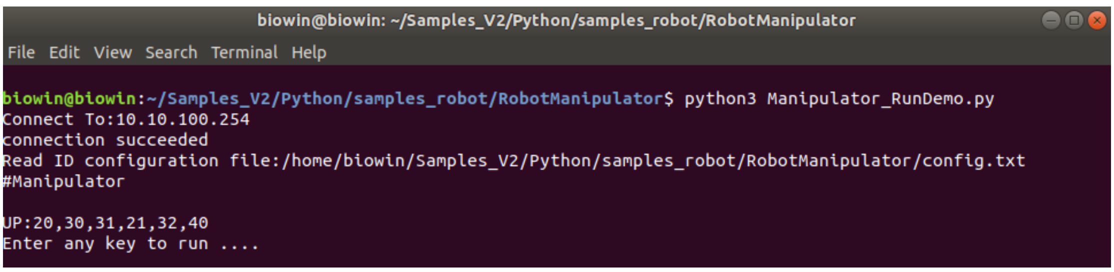
Steps: Turn on the F module switch, and connect the PC to the hotspot corresponding to the F module, which is generally biowinF_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotManipulator/

Right-click to open the terminal and execute the Python command:

$ python3 Manipulator_RunDemo.py

Operation result:



```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotManipulator
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotManipulator$ python3 Manipulator_RunDemo.py
Connect To:10.10.100.254
connection succeeded
Read ID configuration file:/home/biowin/Samples_V2/Python/samples_robot/RobotManipulator/config.txt
#Manipulator

UP:20,30,31,21,32,40
Enter any key to run ....
```

After running the program, press Enter, the manipulator initializes and grabs the object, rotates 90 degrees and puts the object down.

6.8.3 Sensor Control Experiment of Manipulator

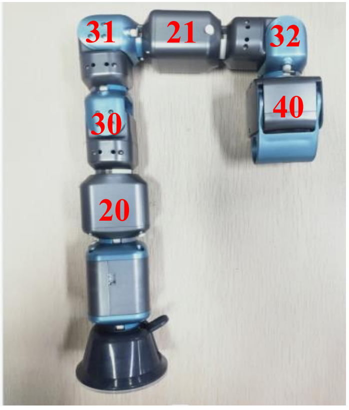
（1）Realization function: Understand the process of adding an action group and remotely execute the action group.

（2）Hardware Preparation

1 F module, 2 I modules, 3 T modules and 1 G module.

（3）Construction of the Configuration

Connect each module according to the picture below:



（4）Example Execution

Steps: Turn on the F module switch, and connect the PC to the hotspot corresponding to the F module, which is generally biowinF_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotManipulator/

Right-click to open the terminal and execute the Python command:

$ sudo python3 Manipulator_RunActions.py

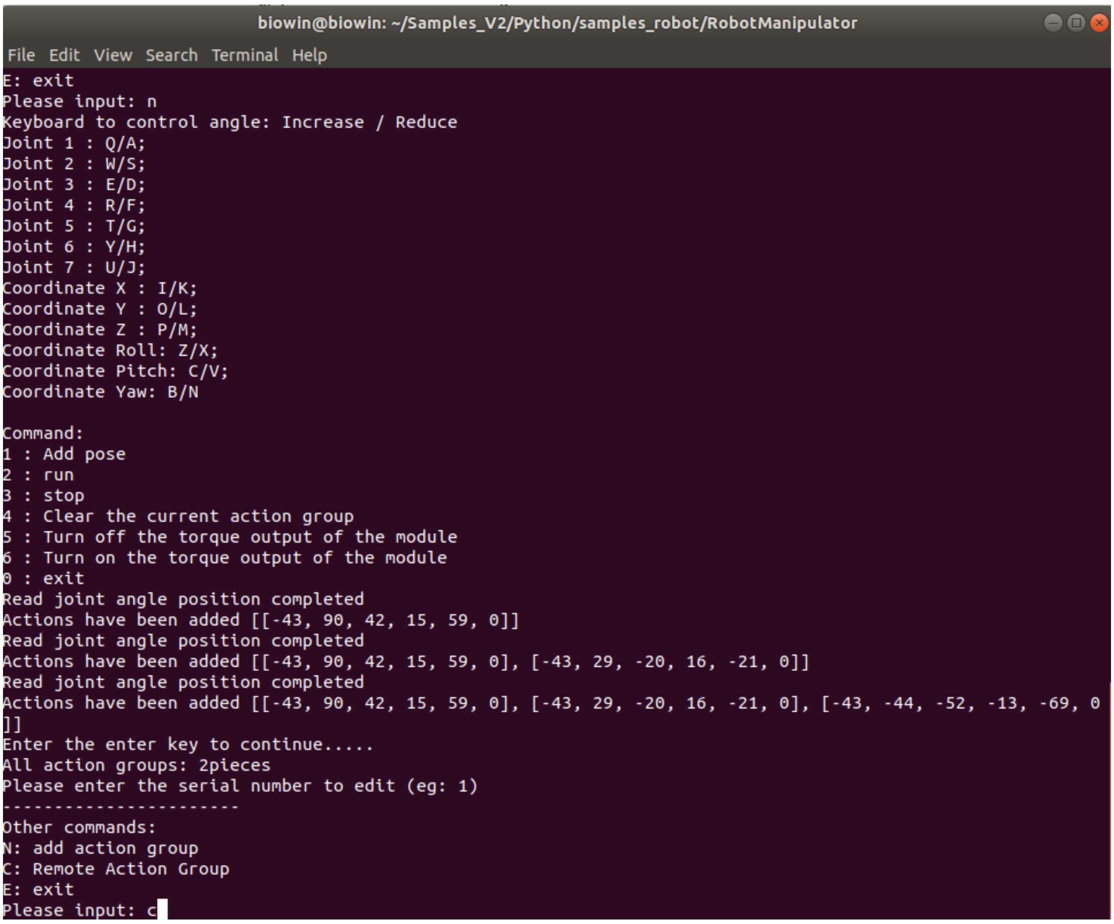
Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotManipulator
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotManipulator$ sudo python3 Manipulator_RunActions.py
Connect To:10.10.100.254
connection succeeded
Read ID configuration file:/home/biowin/Samples_V2/Python/samples_robot/RobotManipulator/config.txt
#Manipulator

UP:20,30,31,21,32,40
SetMotorMode 20 0
The position of the joint module 20 is 5
SetMotorMode 30 0
The position of the joint module 30 is 12
SetMotorMode 31 0
The position of the joint module 31 is 60
SetMotorMode 21 0
The position of the joint module 21 is 0
SetMotorMode 32 0
The position of the joint module 32 is 39
SetMotorMode 40 0
The position of the joint module 40 is 0
All action groups: 0pieces
Please enter the serial number to edit (eg: 1)
----------------------
Other commands:
N: add action group
C: Remote Action Group
E: exit
Please input: n
```

After running the program, you can see the above interface, then we enter "N/n" to choose to add an action group.

```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotManipulator
File Edit View Search Terminal Help
E: exit
Please input: n
Keyboard to control angle: Increase / Reduce
Joint 1 : Q/A;
Joint 2 : W/S;
Joint 3 : E/D;
Joint 4 : R/F;
Joint 5 : T/G;
Joint 6 : Y/H;
Joint 7 : U/J;
Coordinate X : I/K;
Coordinate Y : O/L;
Coordinate Z : P/M;
Coordinate Roll: Z/X;
Coordinate Pitch: C/V;
Coordinate Yaw: B/N

Command:
1 : Add pose
2 : run
3 : stop
4 : Clear the current action group
5 : Turn off the torque output of the module
6 : Turn on the torque output of the module
0 : exit
The servo enable is turned off
Read joint angle position completed
Actions have been added [[-43, 9, -12, 0, 9, 0]]
Read joint angle position completed
Actions have been added [[-43, 9, -12, 0, 9, 0], [-43, -18, -42, 16, -11, 0]]
Read joint angle position completed
Actions have been added [[-43, 9, -12, 0, 9, 0], [-43, -18, -42, 16, -11, 0], [-43, -6, 39, 16, 26, 0]]
Enter the enter key to continue.....
All action groups: 1pieces
Please enter the serial number to edit (eg: 1)
----------------------
Other commands:
N: add action group
C: Remote Action Group
E: exit
Please input:
```

The process of adding an action group is not complicated. After we enter the interface for adding an action group, in order to add an action, we need to input English letters to control the movement of the joints. This step has actually been done in the previous experiments. Without further elaboration, after controlling the joint to get the desired action, we can press the number "1" on the keyboard to add the action. In the above screenshot, we entered the number "1" at the beginning, so the starting position was recorded at the beginning, and then we entered a string of English letters, at this time the joints are controlled to move, and then pressing the number "1" will Once again record the robot position after being controlled. So two actions are recorded. Then click "0" to exit.



After exiting back to the initial interface, we enter "n" again, repeat the process just now, and create a new action group. Well, now we have two action groups, and then select the remote control to execute the action group. In the front, we recorded two groups of action groups. Now press the number on the remote control to execute the action group just now. Among them, the number "1" represents the execution of the action group 1; the number "2" represents the execution of the action group 2.

```
All action groups: 2 pieces
Please enter the serial number to edit (eg: 1)
----------------------
Other commands:
N: add action group
C: Remote Action Group
E: exit
Please input: c
All action groups have 2
Please press the infrared remote control to control the action group...
(Remote control 1. Indicates the execution of action group 1; 2. Indicates the e
xecution of action group 2...)
Remote switch: exit
Detected commands:11
loading 1/2
Action list:
[[-127, 12, -90, 174, 80, 0], [-127, 0, -75, 176, 83, 1], [-127, 0, -75, 180, 72
, 34]]
execute action[-127, 12, -90, 174, 80, 0]
execute action[-127, 0, -75, 176, 83, 1]
execute action[-127, 0, -75, 180, 72, 34]
run is complete
```

6.9 Experiment of Vehicle-Mounted Manipulator

6.9.1 Control Experiment of Vehicle-Mounted Manipulator

(1) Function realization: Control the movement of each joint of the mobile car and the manipulator through the keyboard.

(2) Hardware Preparation

1 ordinary four-rubber wheel car and 1 manipulator with five degrees of freedom (I T T I T G).

(3) Construction of the Configuration

Connect the manipulator (configured as I T T I T G) to the four rubber wheel car, as shown in the figure:

(4) Example Execution

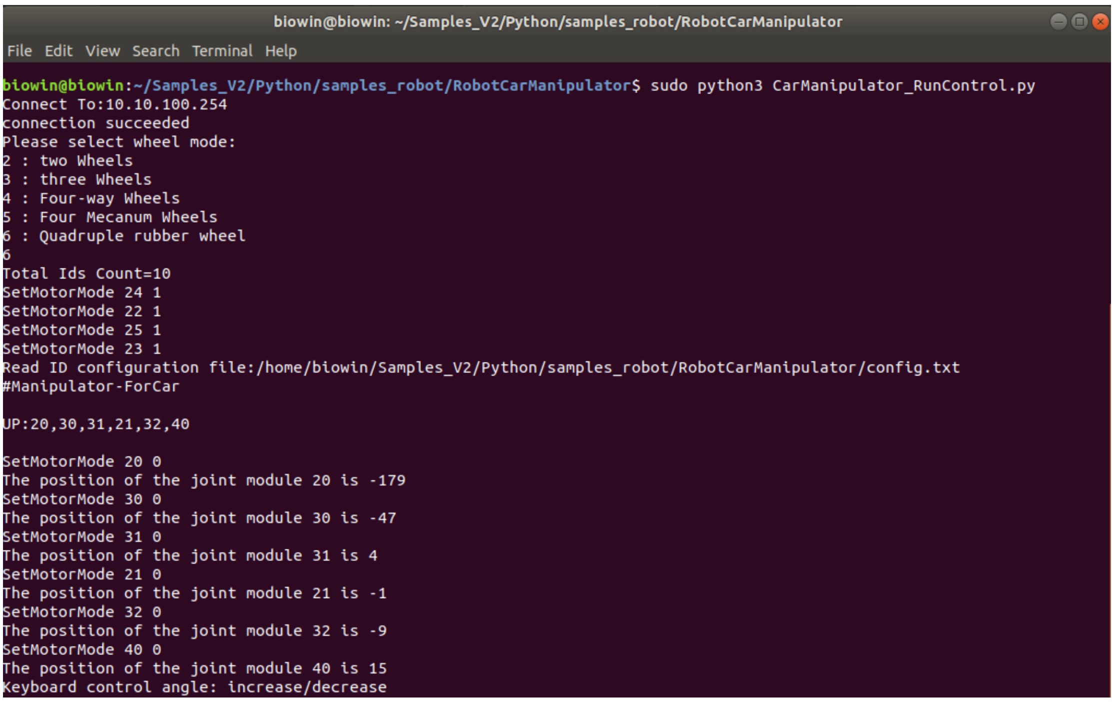
Turn on the M module switch, and connect the PC to the hotspot corresponding to the M module, which is generally biowinM_xxx.

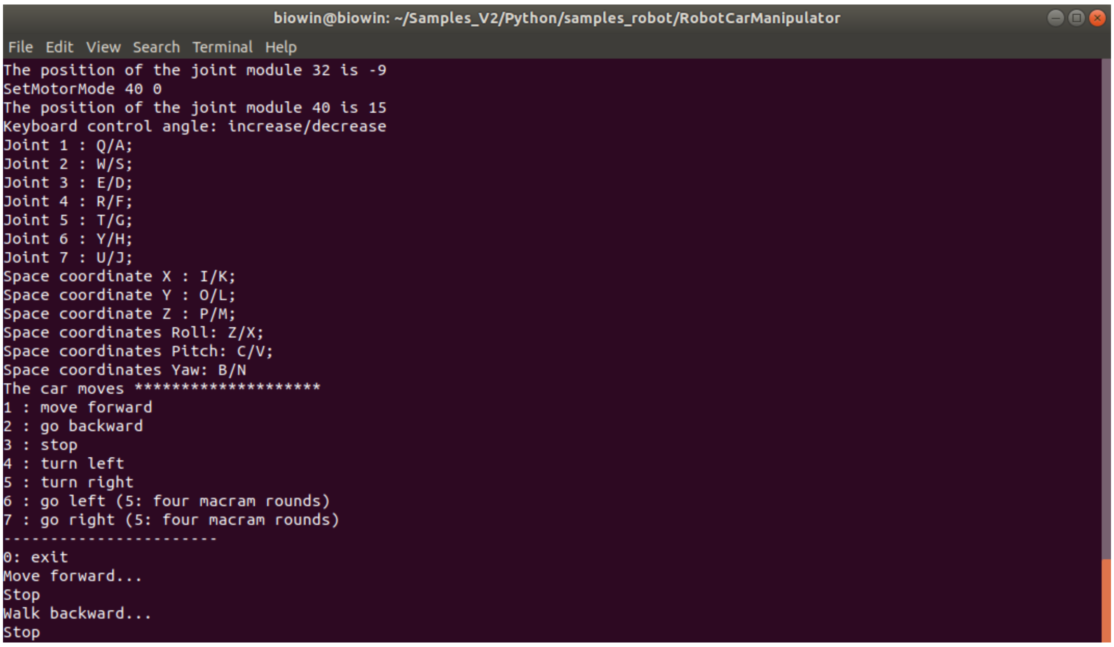
Go to:/Home/Samples_V2/Python/samples_robot/RobotCarManipulator/

Right-click to open the terminal and execute the Python command:

$ sudo python3 CarManipulator_RunControl.py

Operation result:


```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotCarManipulator
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotCarManipulator$ sudo python3 CarManipulator_RunControl.py
Connect To:10.10.100.254
connection succeeded
Please select wheel mode:
2 : two Wheels
3 : three Wheels
4 : Four-way Wheels
5 : Four Mecanum Wheels
6 : Quadruple rubber wheel
6
Total Ids Count=10
SetMotorMode 24 1
SetMotorMode 22 1
SetMotorMode 25 1
SetMotorMode 23 1
Read ID configuration file:/home/biowin/Samples_V2/Python/samples_robot/RobotCarManipulator/config.txt
#Manipulator-ForCar

UP:20,30,31,21,32,40

SetMotorMode 20 0
The position of the joint module 20 is -179
SetMotorMode 30 0
The position of the joint module 30 is -47
SetMotorMode 31 0
The position of the joint module 31 is 4
SetMotorMode 21 0
The position of the joint module 21 is -1
SetMotorMode 32 0
The position of the joint module 32 is -9
SetMotorMode 40 0
The position of the joint module 40 is 15
Keyboard control angle: increase/decrease
```



After running the program, select the four rubber wheel mode. In this case, we can enter letters to control the movement of the joints and numbers to control the movement of the car.

6.9.2 Demo Experiment of Vehicle-Mounted Manipulator Motion

(1) Function realization: Control the mobile car and realize the combined movement of the robot arm on the car to grasp the object.

(2) Hardware Preparation

1 ordinary four-rubber wheel car and 1 manipulator with five degrees of freedom (I T T I T G).

(3) Construction of the Configuration

Connect the manipulator (configured as I T T I T G) to the four rubber wheel car, as shown in the figure:

(4) Example Execution

Purpose: To run the motion demo of the car and the manipulator on the car.

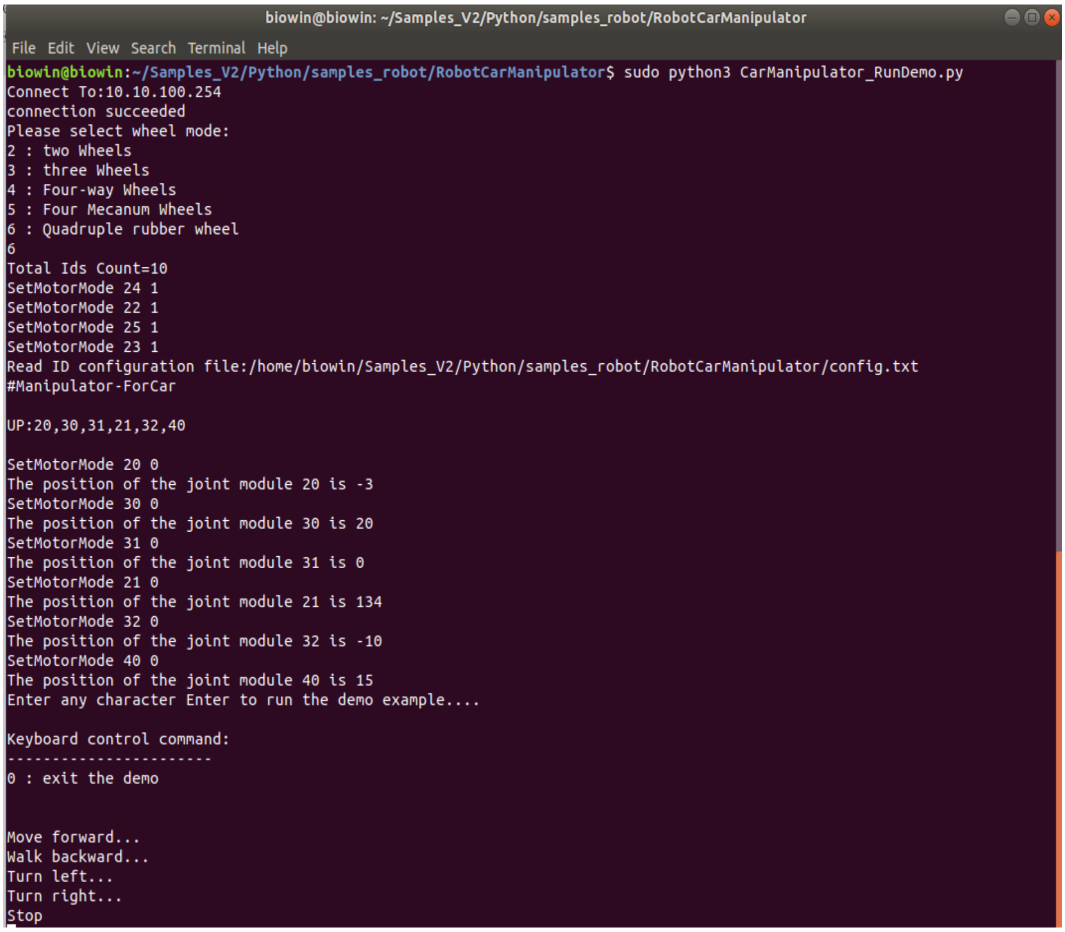
Turn on the M module switch, and connect the PC to the hotspot corresponding to the M module, which is generally biowinM_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotCarManipulator/

Right-click to open the terminal and execute the Python command:

$ sudo python3 CarManipulator_RunDemo.py

Operation result:



```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotCarManipulator
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotCarManipulator$ sudo python3 CarManipulator_RunDemo.py
Connect To:10.10.100.254
connection succeeded
Please select wheel mode:
2 : two Wheels
3 : three Wheels
4 : Four-way Wheels
5 : Four Mecanum Wheels
6 : Quadruple rubber wheel
6
Total Ids Count=10
SetMotorMode 24 1
SetMotorMode 22 1
SetMotorMode 25 1
SetMotorMode 23 1
Read ID configuration file:/home/biowin/Samples_V2/Python/samples_robot/RobotCarManipulator/config.txt
#Manipulator-ForCar

UP:20,30,31,21,32,40

SetMotorMode 20 0
The position of the joint module 20 is -3
SetMotorMode 30 0
The position of the joint module 30 is 20
SetMotorMode 31 0
The position of the joint module 31 is 0
SetMotorMode 21 0
The position of the joint module 21 is 134
SetMotorMode 32 0
The position of the joint module 32 is -10
SetMotorMode 40 0
The position of the joint module 40 is 15
Enter any character Enter to run the demo example....

Keyboard control command:
-----------------------
0 : exit the demo


Move forward...
Walk backward...
Turn left...
Turn right...
Stop
```

After running the program, the manipulator on the car first performs the grabbing action, then the car moves back and forth, left and right, and finally returns to the origin, and the manipulator does the action of putting down the object.

6.10 Experiment of Spider Robot

6.10.1 Basic Experiment of Spider Robot

6.10.1.1 Control Experiment of Spider Robot

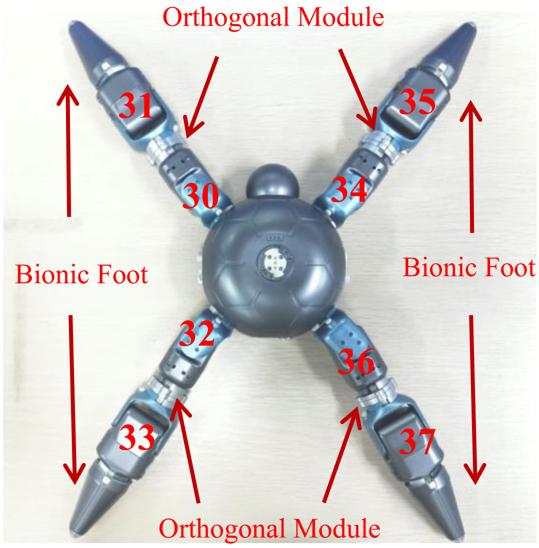
(1) Function realization: Control the spider robot to move forward and backward, turn left and right and various display actions through the keyboard.

(2) Hardware Preparation

1 P module, 8 T modules, 4 bionic feet and 4 orthogonal modules.

(3) Construction of the Configuration

Connect the P module, T module, bionic foot and orthogonal module according to the diagram below.



(4) Example Execution

Purpose: To control the movement of the spider robot.

Steps: Turn on the P module switch, and connect the PC to the hotspot corresponding to the P module, which is generally biowinP_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotSpider/

Right-click to open the terminal and execute the Python command:

$ sudo python3 Spider_RunControl.py

Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotSpider
File  Edit  View  Search  Terminal  Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotSpider$ sudo python3 Spider_RunControl.py
[sudo] password for biowin:
Connect To:10.10.100.254
connection succeeded
Read ID configuration file:/home/biowin/Samples_V2/Python/samples_robot/RobotSpider/config.txt
#Spider

L1:30,31

L3:32,33

R1:34,35

R2:36,37

SetMotorMode 30 0
SetMotorMode 31 0
SetMotorMode 32 0
SetMotorMode 33 0
SetMotorMode 34 0
SetMotorMode 35 0
SetMotorMode 36 0
SetMotorMode 37 0
Robot initialization setting succeeded
Keyboard control command:
1 : forward
2 : back
3 : crawl forward
4 : Crawling back
5 : turn left
6 : turn right
7 : Crawling left turn
8 : Crawling right turn
9 : squat and stand up
a : say hello
b : push up
c : one-handed push-up
d : dance
e : wagging tail
f : combat defense
g : twist body defense
----------------------
0 : Exit
```

After running the program, the above interface appears. Now you can control the movement of the spider by entering keyboard commands. The control commands are as follows:

Command	Action
1	Forward
2	Backward
3	Crawling forward
4	Crawling back
5	Turn left

Command	Action
6	Turn right
7	Crawling left turn
8	Crawling right turn
9	Squat and stand
a	Greet
b	Push-up
c	One-handed push-up
d	Dance
e	Wagging tail
f	Combat defense
g	Wiggle body defense
0	Exit

6.10.1.2 Demo Experiment of Spider Robot Motion

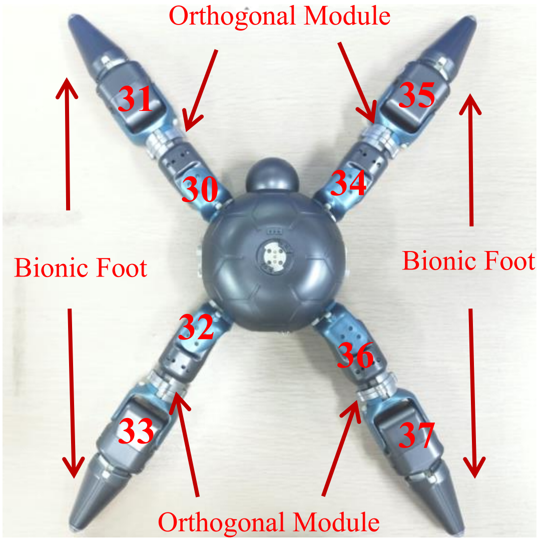
（1） Function realization: Control the spider robot to do combined movements, perform forward and backward movements, left and right steering and various display actions.

（2） Hardware Preparation

1 P module, 8 T modules, 4 bionic feet and 4 orthogonal modules.

(3) Construction of the Configuration

Connect the P module, T module, bionic foot and orthogonal module according to the diagram below.



(4) Example Execution

Purpose: To control the movement of the spider robot.

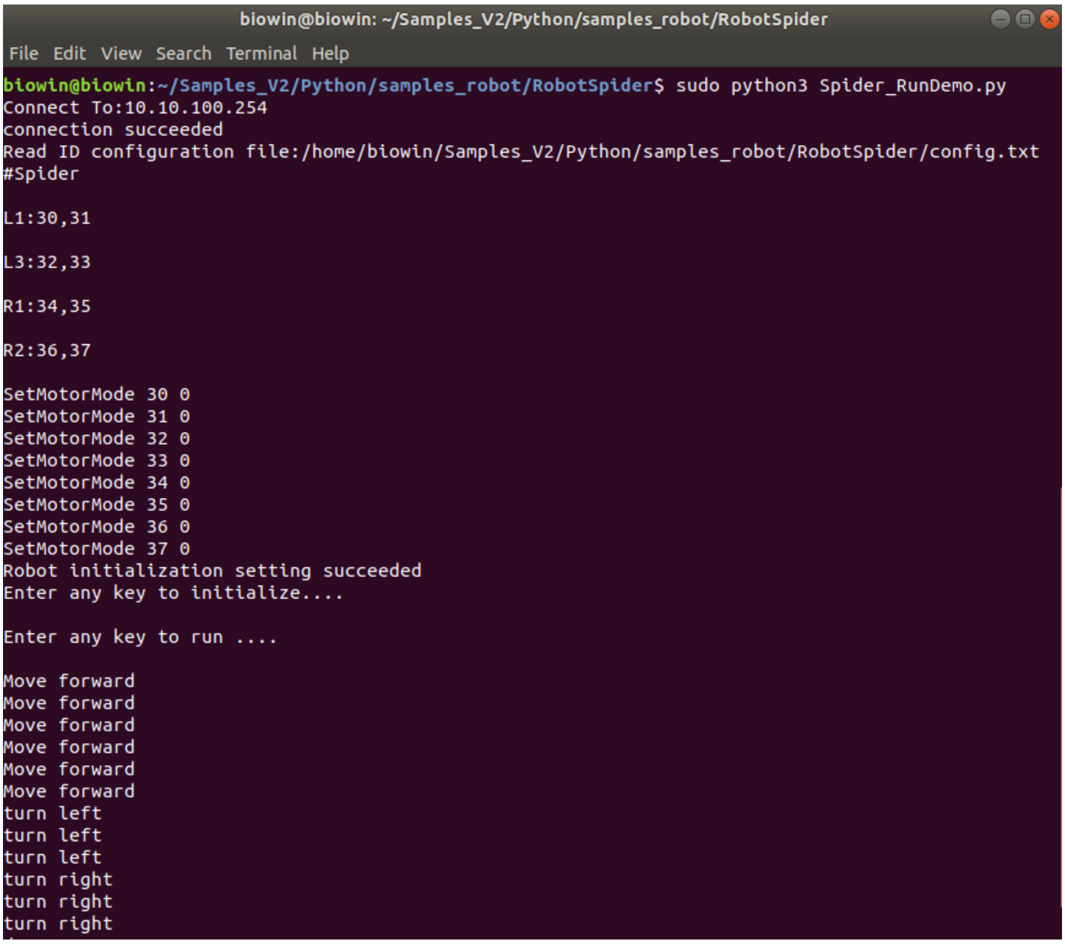
Steps: Turn on the P module switch, and connect the PC to the hotspot corresponding to the P module, which is generally biowinP_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotSpider/

Right-click to open the terminal and execute the Python command:

$ python3 Spider_RunDemo.py

Operation result:



After running the program, the spider robot performs all actions three times.

6.10.2 Experiment of Spider Robot Combined with Sensor

Note: In the spider's sensor experiment, if you press the "ctrl+c" key once and the program does not exit, please press the "ctrl+c" key again until it exits.

6.10.2.1 Crossing the Fireline Experiment

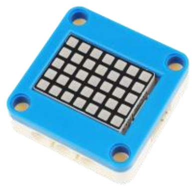
(1) Function realization: The spider robot uses ultrasonic sensors to detect higher obstacles in front (such as bridges) during the forward movement. After detection, the spider crawls forward to cross, and the dot matrix displays different expressions.

(2) Hardware Preparation

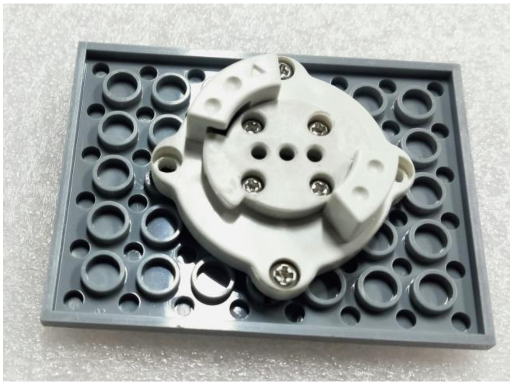
1 P module, 8 T modules, 4 bionic feet, 4 orthogonal modules, 1 MINI PRO

sensor main control board module, 1 ultrasonic sensor module and 1 dot matrix module.



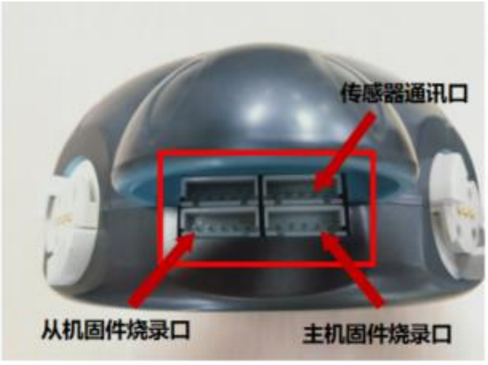



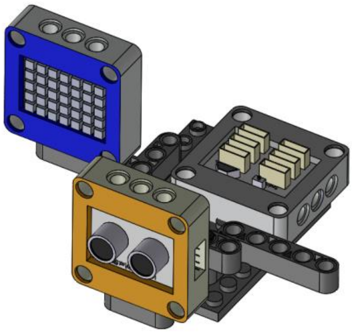

The sensor fixing plate of the P module is used to be fixedly connected to the top interface of the P module to facilitate the installation of the sensor.

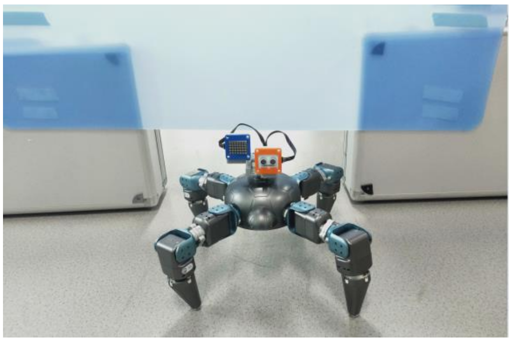


（3） Construction of the Configuration

Connect the module and sensor according to the diagram below. Among them, the ultrasonic is connected to the D3 port of the main control board, and the dot matrix is connected to the D4 port of the main control board. The interface between the P module and the main control board is the serial port. For details, see the sensor control example in Chapter 5.







(4) Example Execution

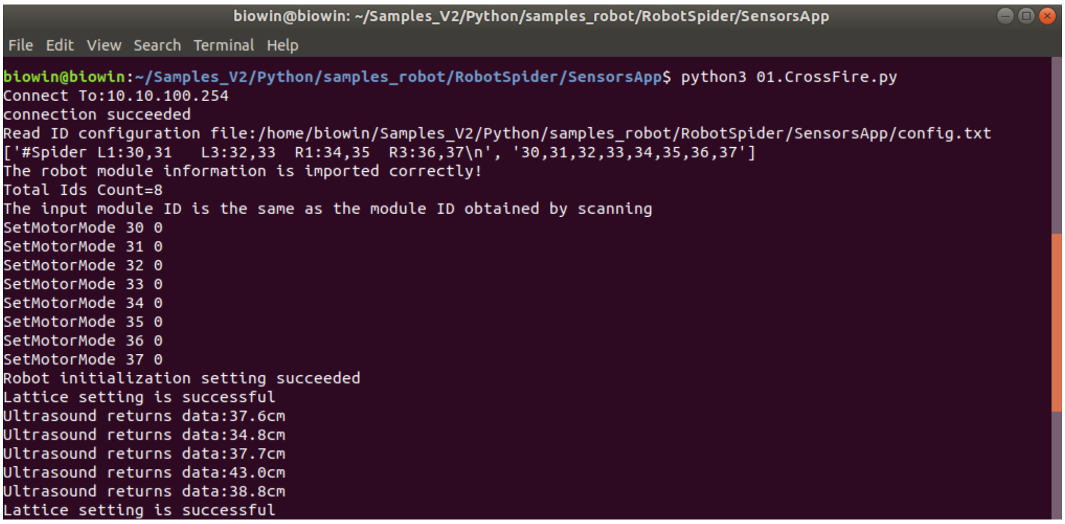
Steps: Turn on the P module switch, and connect the PC to the hotspot corresponding to the P module, which is generally biowinP_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp/

Right-click to open the terminal and execute the Python command:

$ python3 01.CrossFire.py

Operation result:



```
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp$ python3 01.CrossFire.py
Connect To:10.10.100.254
connection succeeded
Read ID configuration file:/home/biowin/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp/config.txt
['#Spider L1:30,31   L3:32,33  R1:34,35  R3:36,37\n', '30,31,32,33,34,35,36,37']
The robot module information is imported correctly!
Total Ids Count=8
The input module ID is the same as the module ID obtained by scanning
SetMotorMode 30 0
SetMotorMode 31 0
SetMotorMode 32 0
SetMotorMode 33 0
SetMotorMode 34 0
SetMotorMode 35 0
SetMotorMode 36 0
SetMotorMode 37 0
Robot initialization setting succeeded
Lattice setting is successful
Ultrasound returns data:37.6cm
Ultrasound returns data:34.8cm
Ultrasound returns data:37.7cm
Ultrasound returns data:43.0cm
Ultrasound returns data:38.8cm
Lattice setting is successful
```

After running the program, the spider starts to move. When it encounters a higher object, the spider crawls forward, and the dot matrix displays the expression at the same time. Note: The interface of the sensor, the interface between the main control board and the P module should be connected correctly.

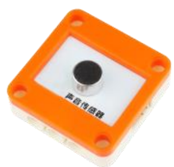
6.10.2.2 Dancing Spider Robot

(1) Function realization: The sound sensor is used to detect the sound of the environment. If the sound is detected, the spider makes a dancing action, and the dot matrix displays different expressions.

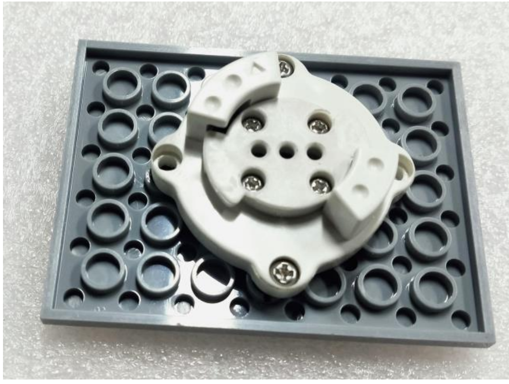
(2) Hardware Preparation

1 P module, 8 T modules, 4 bionic feet, 4 orthogonal modules, 1 MINI PRO sensor main control board module, 1 dot matrix module and 1 sound sensor module.



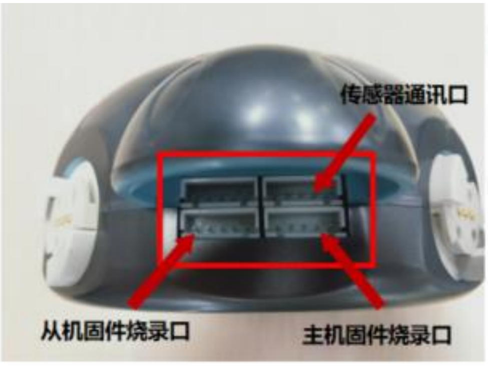



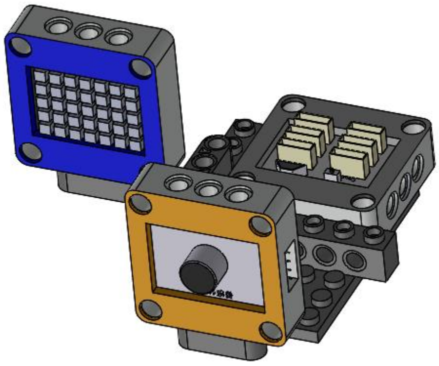
The sensor fixing plate of the P module is used to be fixedly connected to the top interface of the P module to facilitate the installation of the sensor.



(3) Construction of the Configuration

Connect the module and sensor according to the diagram below. Among them, the sound sensor is connected to the D3 port of the main control board, and the dot matrix is connected to the D4 port of the main control board. The interface between the P module and the main control board is the serial port. For details, see the sensor control example in Chapter 5.





(4) Example Execution

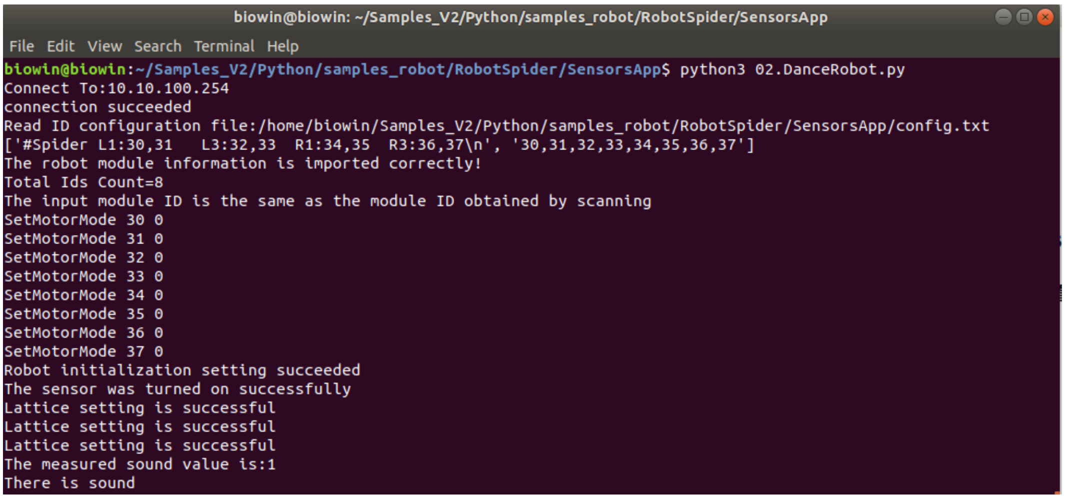
Steps: Turn on the P module switch, and connect the PC to the hotspot corresponding to the P module, which is generally biowinP_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp/

Right-click to open the terminal and execute the Python command:

$ python3 02.DanceRobot.py

Operation result:



```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp
File  Edit  View  Search  Terminal  Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp$ python3 02.DanceRobot.py
Connect To:10.10.100.254
connection succeeded
Read ID configuration file:/home/biowin/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp/config.txt
['#Spider L1:30,31   L3:32,33  R1:34,35  R3:36,37\n', '30,31,32,33,34,35,36,37']
The robot module information is imported correctly!
Total Ids Count=8
The input module ID is the same as the module ID obtained by scanning
SetMotorMode 30 0
SetMotorMode 31 0
SetMotorMode 32 0
SetMotorMode 33 0
SetMotorMode 34 0
SetMotorMode 35 0
SetMotorMode 36 0
SetMotorMode 37 0
Robot initialization setting succeeded
The sensor was turned on successfully
Lattice setting is successful
Lattice setting is successful
Lattice setting is successful
The measured sound value is:1
There is sound
```

After running the program, when the sound sensor detects the sound and the sound is greater than the threshold, the spider robot performs the dancing action, and the dot matrix displays the expression at the same time.

Note: The sound sensor is connected to the D3 port of the main control board.

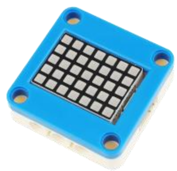
6.10.2.3 Greeting Spider Robot

(1) Function realization: The sound sensor is used to detect the sound of the environment. If it is detected, the spider makes a greeting action, the dot matrix displays different expressions, and the full-color LED lights up in different colors.

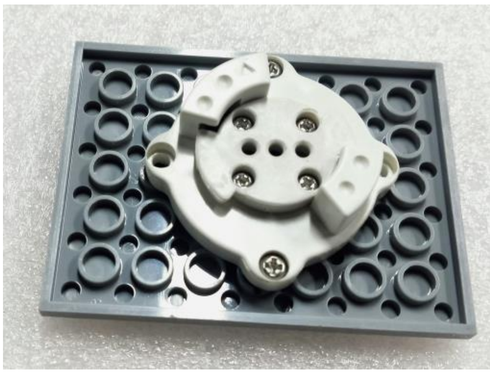
(2) Hardware Preparation

1 P module, 8 T modules, 4 bionic feet, 4 orthogonal modules, 1 MINI PRO sensor main control board module, 1 dot matrix module, 1 full color LED module and 1 sound sensor module.



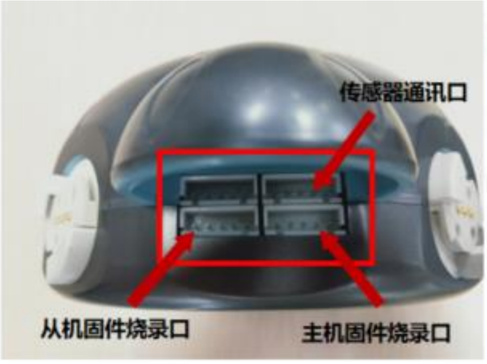


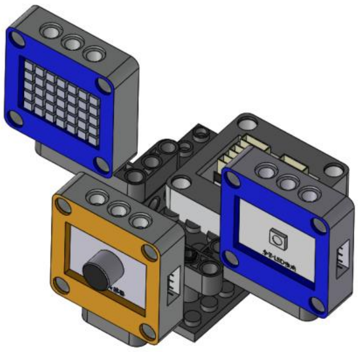
The sensor fixing plate of the P module is used to be fixedly connected to the top interface of the P module to facilitate the installation of the sensor.



(3) Construction of the Configuration

Connect the module and sensor according to the diagram below. Among them, the sound sensor is connected to the D3 port of the main control board, the dot matrix is connected to the D4 port of the main control board, and the full-color LED module is connected to the D2 port of the main control board. The interface between the P module and the main control board is the serial port. For details, see the sensor control example in Chapter 5.





（4）Example Execution

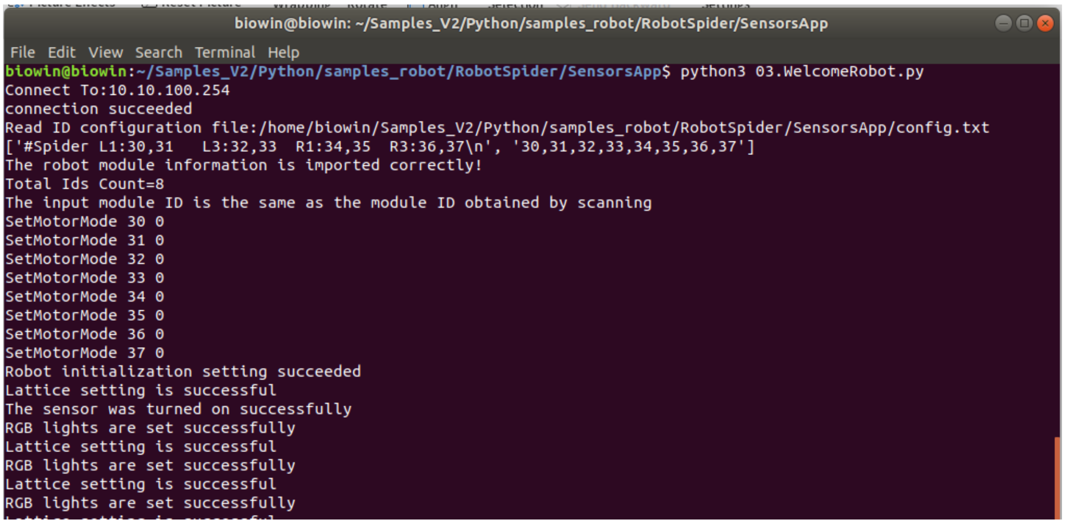
Steps: Turn on the P module switch, and connect the PC to the hotspot corresponding to the P module, which is generally biowinP_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp/

Right-click to open the terminal and execute the Python command:

$ python3 03.WelcomeRobot.py

Operation result:



After running the program, when the sound sensor detects the sound and the sound is greater than the threshold, the spider robot starts to do the welcome action, at the same time, the dot matrix displays a happy expression, and the LED lights light up in different colors.

Note: The appearance color of the sensor may be different, but the name is the same, mainly for the function.

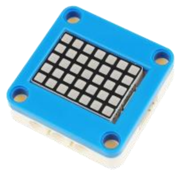
6.10.2.4 Following Spider Robot

(1) Function realization: The ultrasonic sensor detects whether there is an object in front of the spider robot and the spider robot maintains a certain distance from the object in front. If the object moves forward, the spider moves forward. If the object moves back, the spider moves back, and the dot matrix displays different expressions.

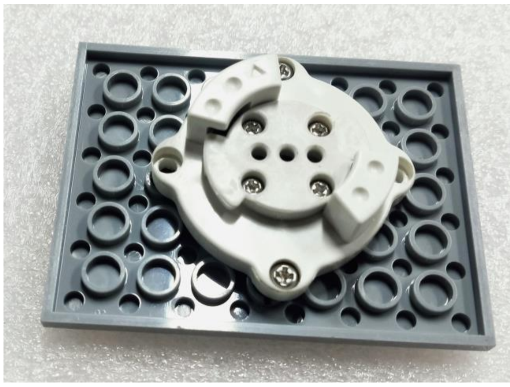
(2) Hardware Preparation

1 P module, 8 T modules, 4 bionic feet, 4 orthogonal modules, 1 MINI PRO sensor main control board module, 1 dot matrix module and 1 ultrasonic sensor module.



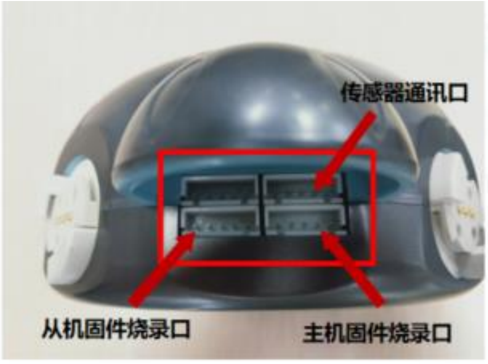


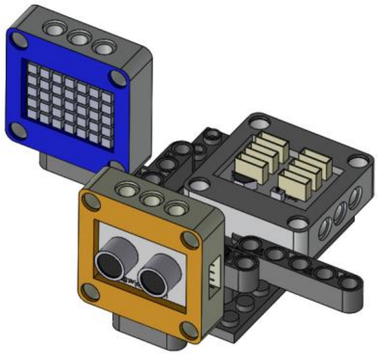

The sensor fixing plate of the P module is used to be fixedly connected to the top interface of the P module to facilitate the installation of the sensor.

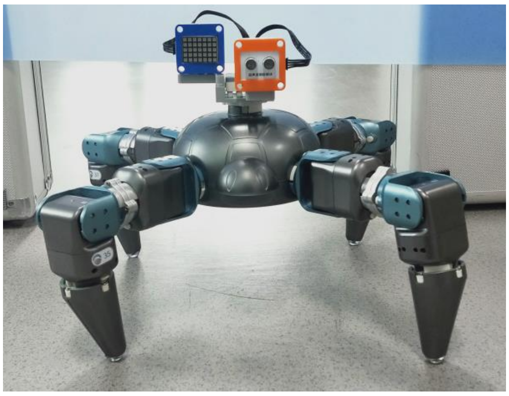


（3）Construction of the Configuration

Connect the module and sensor according to the diagram below.Among them, the ultrasonic sensor is connected to the D3 port of the main control board, and the dot matrix is connected to the D4 port of the main control board. The interface between the P module and the main control board is the serial port. For details, see the sensor control example in Chapter 5.







（4）Example Execution

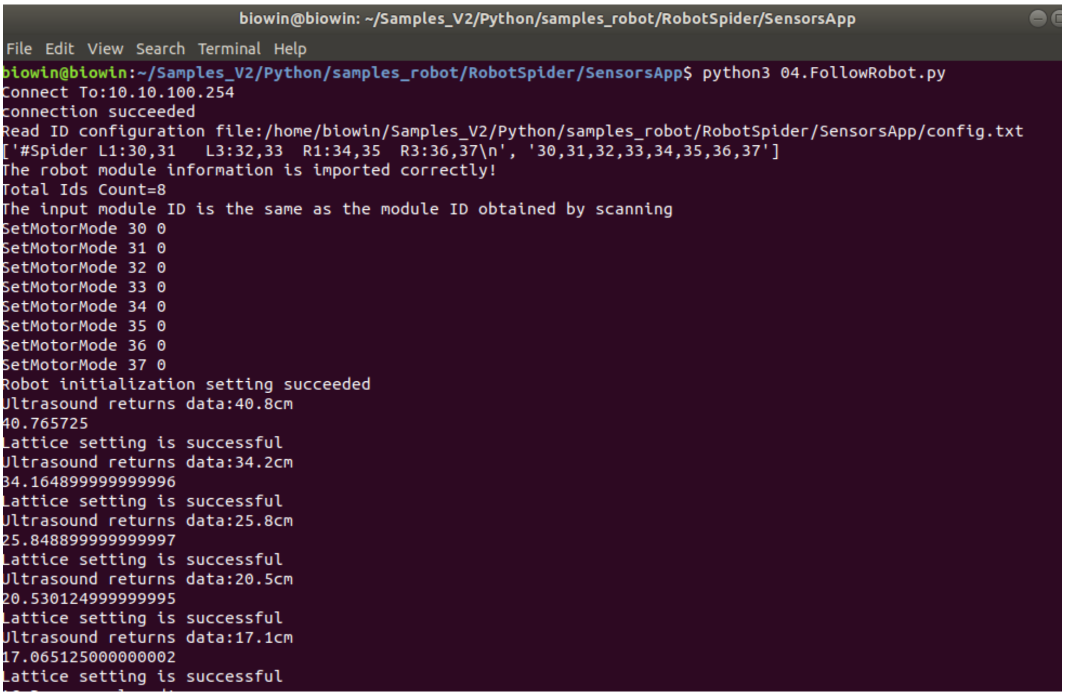
Steps: Turn on the P module switch, and connect the PC to the hotspot corresponding to the P module, which is generally biowinP_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp/

Right-click to open the terminal and execute the Python command:

$ python3 04.FollowRobot.py

Operation result:



After running the program, the spider robot starts to move. When the distance between the spider robot and the object is greater than 30cm, the spider robot moves forward, and if it is less than 30cm, it retreats. To put it simply, the spider robot always keeps a certain distance from the object.

Tip: You can approach the sensor with your hand, the spider robot will move back, the hand will move forward, and the spider robot will also move forward. At the same time, the dot matrix shows the moving direction of the spider robot with arrows.

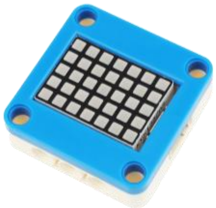
6.10.2.5 Gesture Control

(1) Function realization: The gesture sensor is used to detect gesture actions, and control the robot to move forward, backward, left and right according to the detected gesture commands, and the dot matrix displays the direction of the movement.

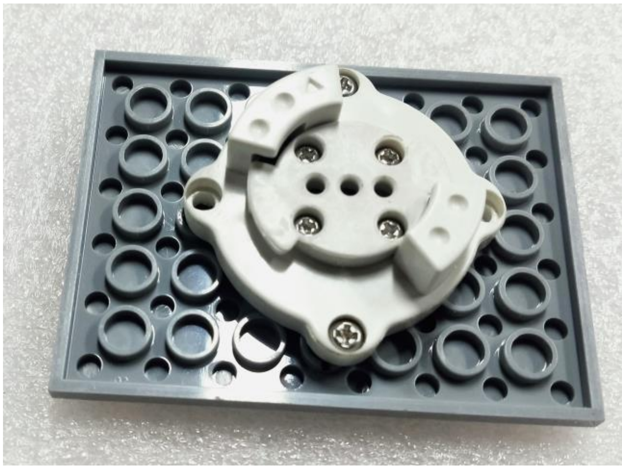
(2) Hardware Preparation

1 P module, 8 T modules, 4 bionic feet, 4 orthogonal modules, 1 MINI PRO sensor main control board module, 1 dot matrix module and 1 Gesture Sensor Module.



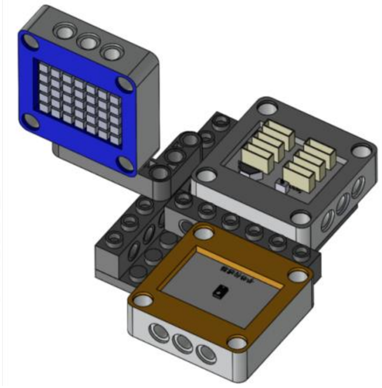



The sensor fixing plate of the P module is used to be fixedly connected to the top interface of the P module to facilitate the installation of the sensor.



（3）Construction of the Configuration

Connect the module and sensor according to the diagram below. Among them, the gesture sensor is connected to the IIC/A5 port of the main control board, and the dot matrix is connected to the D4 port of the main control board. The interface between the P module and the main control board is the serial port. For details, see the sensor control example in Chapter 5.



（4）Example Execution

Purpose: To test the control of spider movement by different gestures.

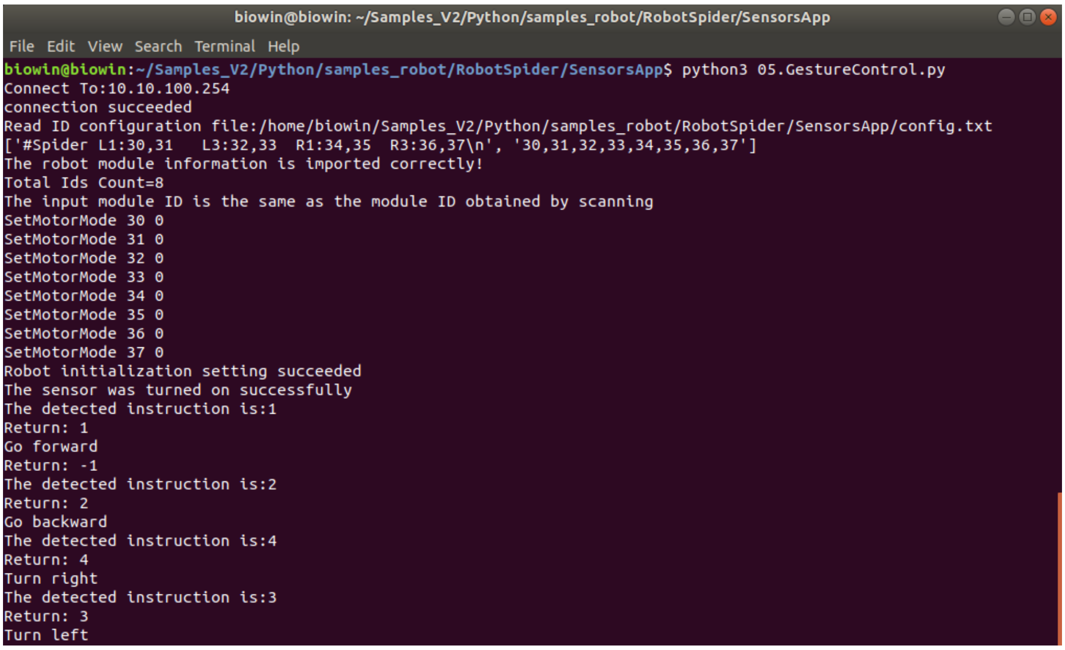
Steps: Turn on the P module switch, and connect the PC to the hotspot corresponding to the P module, which is generally biowinP_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp/

Right-click to open the terminal and execute the Python command:

$ python3 05.GestureControl.py

Operation result:



```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp$ python3 05.GestureControl.py
Connect To:10.10.100.254
connection succeeded
Read ID configuration file:/home/biowin/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp/config.txt
['#Spider L1:30,31   L3:32,33  R1:34,35  R3:36,37\n', '30,31,32,33,34,35,36,37']
The robot module information is imported correctly!
Total Ids Count=8
The input module ID is the same as the module ID obtained by scanning
SetMotorMode 30 0
SetMotorMode 31 0
SetMotorMode 32 0
SetMotorMode 33 0
SetMotorMode 34 0
SetMotorMode 35 0
SetMotorMode 36 0
SetMotorMode 37 0
Robot initialization setting succeeded
The sensor was turned on successfully
The detected instruction is:1
Return: 1
Go forward
Return: -1
The detected instruction is:2
Return: 2
Go backward
The detected instruction is:4
Return: 4
Turn right
The detected instruction is:3
Return: 3
Turn left
```

After running the program, the gesture sensor starts to detect gestures. When the gesture is detected, the spider robot makes corresponding actions, and the dot matrix shows the movement direction of the spider robot.

Gesture	Movement direction
Up gesture	Go forward
Down gesture	Go backward
Left gesture	Turn left
Right gesture	Turn right

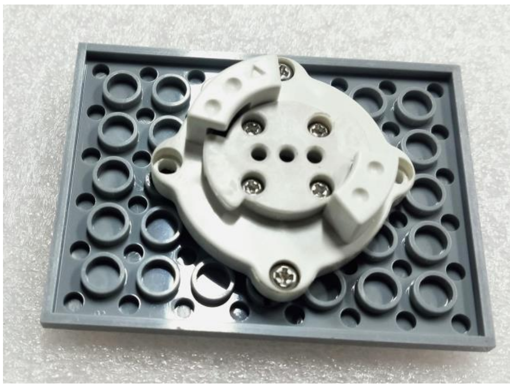
6.10.2.6 Remote Control

(1) Function realization: Control the spider robot to move forward, backward, left turn, right turn, push-up and dance by pressing the buttons on the remote control, and each movement dot matrix displays different expressions, and the full-color LED displays different colors of lights.

(2) Hardware Preparation

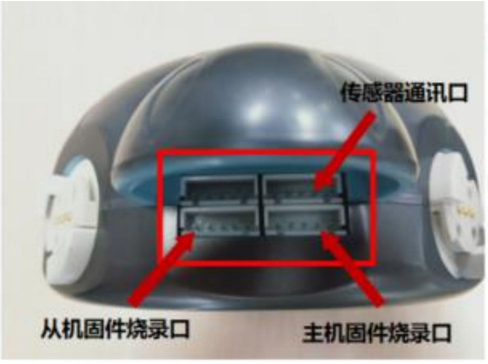
1 P module, 8 T modules, 4 bionic feet, 4 orthogonal modules, 1 MINI PRO sensor main control board module, 1 dot matrix module, 1 full color LED module, 1 remote control receiving module and 1 remote control.

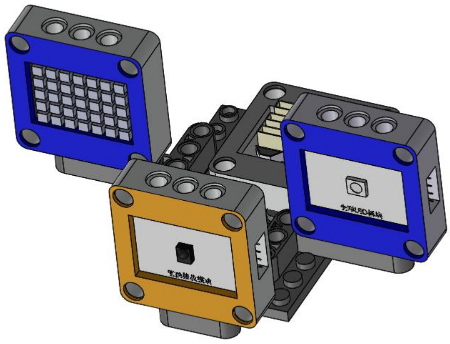
The sensor fixing plate of the P module is used to be fixedly connected to the top interface of the P module to facilitate the installation of the sensor.



(3) Construction of the Configuration

Connect the module and sensor according to the diagram below. Among them, the remote control receiving module is connected to the D3 port of the main control board, the dot matrix is connected to the D4 port of the main control board, and the full-color LED light is connected to the D2 port. The interface between the P module and the main control board is the serial port. For details, see the sensor control example in Chapter 5.





（4）Example Execution

Steps: Turn on the P module switch, and connect the PC to the hotspot corresponding to the P module, which is generally biowinP_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp/

Right-click to open the terminal and execute the Python command:

$ python3 06.RemoteControl.py

Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp
File  Edit  View  Search  Terminal  Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp$ python3 06.RemoteControl.py
Connect To:10.10.100.254
connection succeeded
Read ID configuration file:/home/biowin/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp/config.txt
['#Spider L1:30,31   L3:32,33  R1:34,35  R3:36,37\n', '30,31,32,33,34,35,36,37']
The robot module information is imported correctly!
Total Ids Count=8
The input module ID is the same as the module ID obtained by scanning
SetMotorMode 30 0
SetMotorMode 31 0
SetMotorMode 32 0
SetMotorMode 33 0
SetMotorMode 34 0
SetMotorMode 35 0
SetMotorMode 36 0
SetMotorMode 37 0
Robot initialization setting succeeded
Please press the button:
The sensor was turned on successfully
The detected instruction is:10
Return: 10
Press "1" - dance
Lattice setting is successful
RGB lights are set successfully
Please press the button:
The detected instruction is:11
Return: 11
Press "2" - push ups
Lattice setting is successful
RGB lights are set successfully
```

After running the program, press keys on the remote control to control the movement of the spider robot. The relationship between commands and actions is as follows:

Command	Action
Press ↑	Go forward, the dot matrix displays the up arrow, and the red light is on.
Press ↓	Go backward, the dot matrix displays the down arrow, and the green light is on.
Press ←	Turn left, the dot matrix displays the left arrow, and the blue light is on.
Press →	Turn right, the dot matrix displays the right arrow, and the yellow light is on
Press 1	Dancing, the dot matrix displays the surprise expression, and the purple light is on.
Press 2	Push-up, the dot matrix displays right wink expression, and the blue light is on.

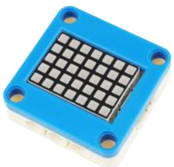
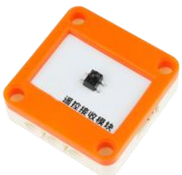
6.10.2.7 Sporting Spider Robot

Function realization: Control the spider robot to complete push-up, one-hand push-up and squat-up movements by pressing the buttons on the remote control, and each movement dot matrix displays different expressions, and the full-color LED displays different colors of lights.

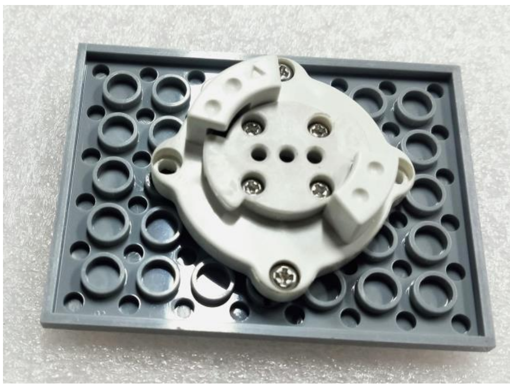
（2）Hardware Preparation

1 P module, 8 T modules, 4 bionic feet, 4 orthogonal modules, 1 MINI PRO sensor main control board module, 1 dot matrix module, 1 full color LED module, 1 remote control receiving module and 1 remote control.



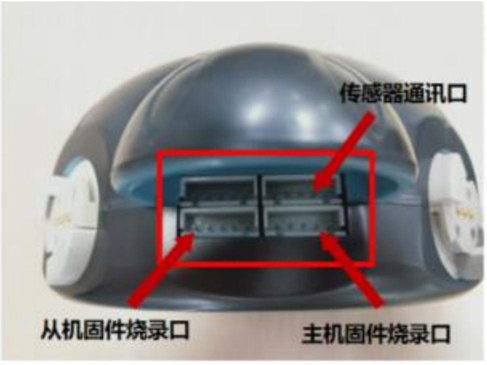




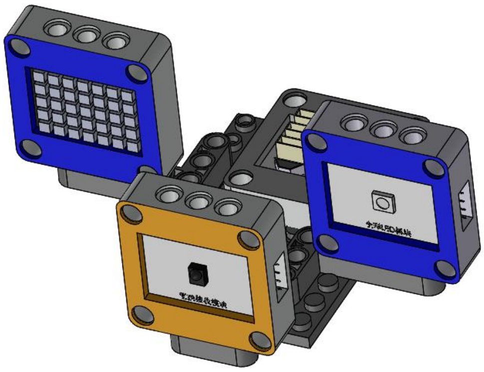
The sensor fixing plate of the P module is used to be fixedly connected to the top interface of the P module to facilitate the installation of the sensor.



(3) Construction of the Configuration

Connect the module and sensor according to the diagram below. Among them, the remote control receiving module is connected to the D3 port of the main control board, the dot matrix is connected to the D4 port of the main control board, and the full-color LED light is connected to the D2 port. The interface between the P module and the main control board is the serial port. For details, see the sensor control example in Chapter 5.





(4) Example Execution

Steps: Turn on the P module switch, and connect the PC to the hotspot corresponding to the P module, which is generally biowinP_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp/

Right-click to open the terminal and execute the Python command:

$ python3 07.SportRobot.py

Operation result:

```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp
File  Edit  View  Search  Terminal  Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp$ python3 07.SportRobot.py
Connect To:10.10.100.254
connection succeeded
Read ID configuration file:/home/biowin/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp/config.txt
['#Spider L1:30,31   L3:32,33  R1:34,35  R3:36,37\n', '30,31,32,33,34,35,36,37']
The robot module information is imported correctly!
Total Ids Count=8
The input module ID is the same as the module ID obtained by scanning
SetMotorMode 30 0
SetMotorMode 31 0
SetMotorMode 32 0
SetMotorMode 33 0
SetMotorMode 34 0
SetMotorMode 35 0
SetMotorMode 36 0
SetMotorMode 37 0
Robot initialization setting succeeded
The sensor was turned on successfully
The detected instruction is:10
Return: 10
Press '1' - one-handed push-up
Lattice setting is successful
RGB lights are set successfully
The detected instruction is:11
Return: 11
Press '2' - push up with both hands
Lattice setting is successful
RGB lights are set successfully
The detected instruction is:12
Return: 12
Press '3' - squat down and stand up
Lattice setting is successful
RGB lights are set successfully
Lattice setting is successful
```

After running the program, use the buttons on the remote control to control the movement of the spider robot. The details are as follows:

Command	Action
Press 1	One-handed push-up, the dot matrix displays the one-handed push-up pattern, and the yellow light is on.
Press 2	Push-up, the dot matrix displays the push-up pattern with both hands, and the purple light is on.
Press 3	Squat up, the dot matrix displays the squat up pattern, and the cyan light is on.

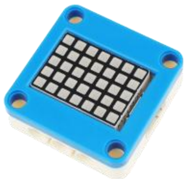
6.10.2.8 Face Changing Spider Robot

(1) Function realization: The spider robot realizes the pacing movement forward and backward, and the object appears in front of the ultrasonic sensor (15-30cm), the dot matrix displays different facial expressions, and the full-color LED lights light up different colors to realize the face-changing effect.

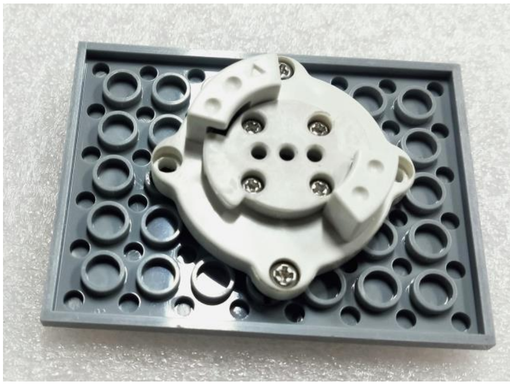
(2) Hardware Preparation

1 P module, 8 T modules, 4 bionic feet, 4 orthogonal modules, 1 MINI PRO sensor main control board module, 1 dot matrix module, 1 full color LED module and 1 ultrasonic sensor module.



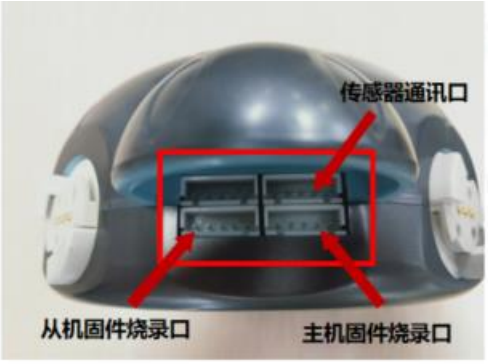



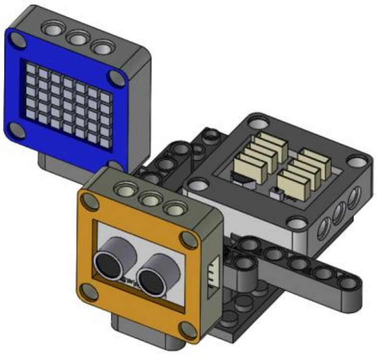
The sensor fixing plate of the P module is used to be fixedly connected to the top interface of the P module to facilitate the installation of the sensor.

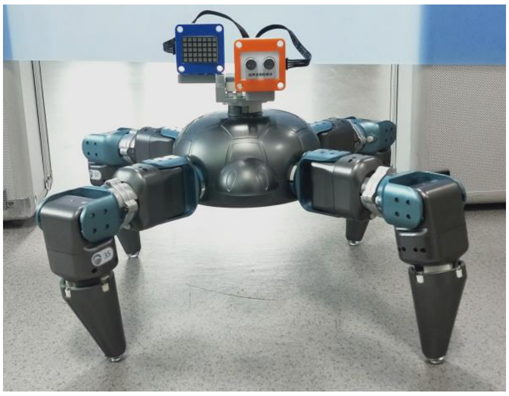


(3) Construction of the Configuration

Connect the module and sensor according to the diagram below. Among them, the ultrasonic sensor is connected to the D3 port of the main control board, the dot matrix is connected to the D4 port of the main control board, and the full-color LED module is connected to the D2 port of the main control board. The interface between the P module and the main control board is the serial port. For details, see the sensor control example in Chapter 5.







（4）Example Execution

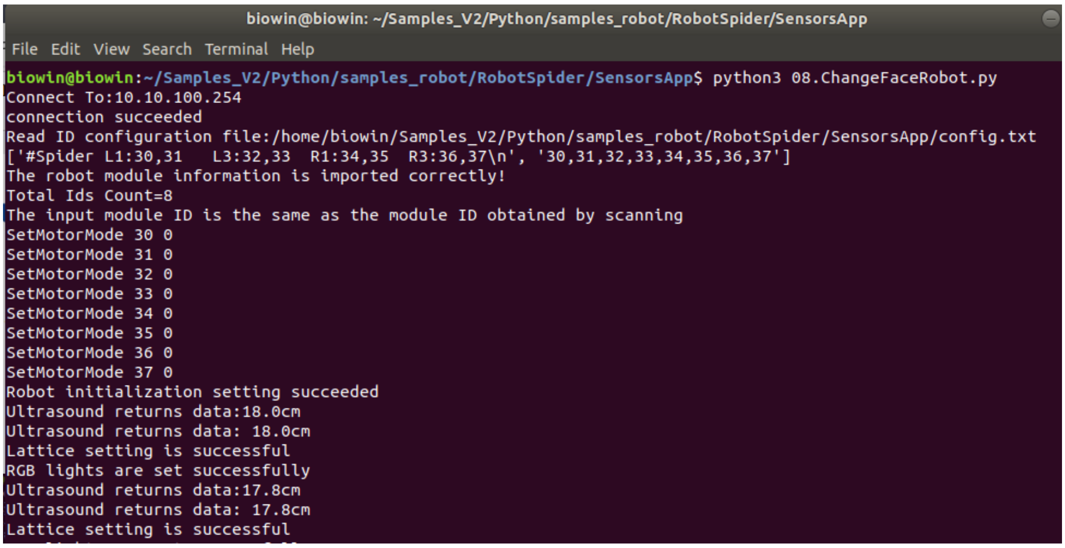
Steps: Turn on the P module switch, and connect the PC to the hotspot corresponding to the P module, which is generally biowinP_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp/

Right-click to open the terminal and execute the Python command:

$ python3 08.ChangeFaceRobot.py

Operation result:



After running the program, the ultrasonic sensor starts to detect obstacles. When the obstacle is 15-30cm in front of the spider robot, the spider robot changes its face, that is, the dot matrix shows different expressions.

6.10.2.9 Defense against Spider Robot

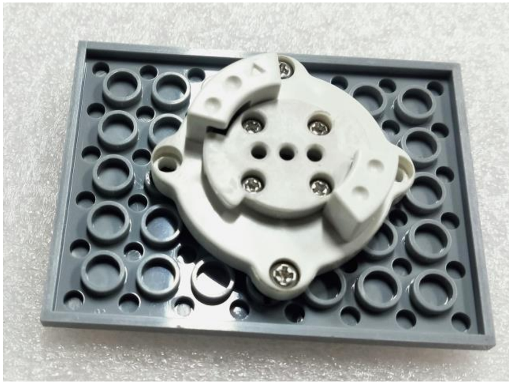
(1) Function realization: The color sensor can identify blocks of different colors, and control the spider robot to make different defensive actions according to the danger level of the identified color (from low to high: green, blue, and red). The full-color LED lights correspond to the identified colors, and the dot matrix displays the capital letters of the corresponding colors.

(2) Hardware Preparation

1 P module, 8 T modules, 4 bionic feet, 4 orthogonal modules, 1 MINI PRO sensor main control board module, 1 dot matrix module, 1 full color LED module and 1 color sensor module.

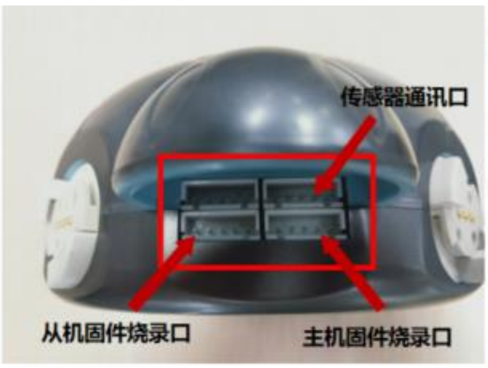


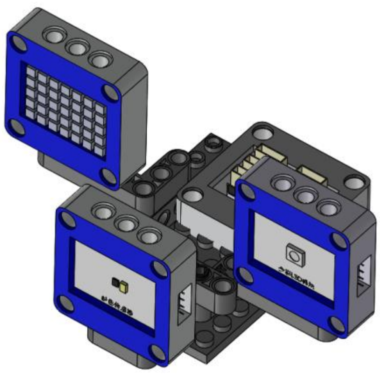
The sensor fixing plate of the P module is used to be fixedly connected to the top interface of the P module to facilitate the installation of the sensor.



（3）Construction of the Configuration

Connect the module and sensor according to the diagram below.Among them, the full-color LED light is connected to the D2 port of the main control board, the dot matrix is connected to the D4 port of the main control board, and the color sensor is connected to the IIC/A5 port of the main control board. The interface between the P module and the main control board is the serial port. For details, see the sensor control example in Chapter 5.





(4) Example Execution

Purpose: To test the response of the spider robot when it encounters squares of different colors.

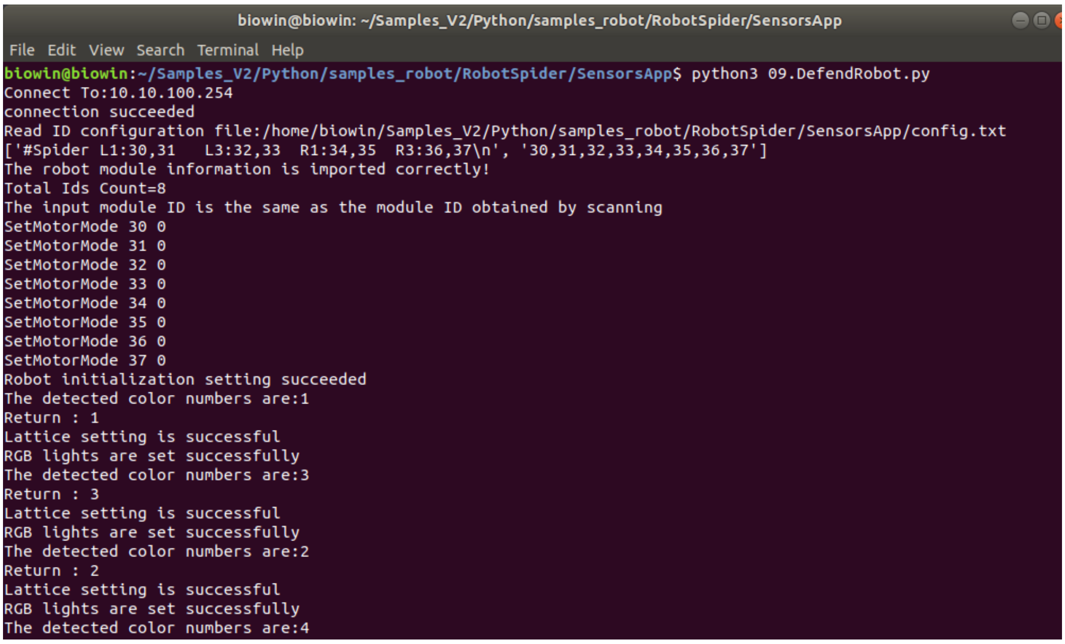
Steps: Turn on the P module switch, and connect the PC to the hotspot corresponding to the P module, which is generally biowinP_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp/

Right-click to open the terminal and execute the Python command:

$ python3 09.DefendRobot.py

Operation result:



```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp
File Edit View Search Terminal Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp$ python3 09.DefendRobot.py
Connect To:10.10.100.254
connection succeeded
Read ID configuration file:/home/biowin/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp/config.txt
['#Spider L1:30,31   L3:32,33  R1:34,35  R3:36,37\n', '30,31,32,33,34,35,36,37']
The robot module information is imported correctly!
Total Ids Count=8
The input module ID is the same as the module ID obtained by scanning
SetMotorMode 30 0
SetMotorMode 31 0
SetMotorMode 32 0
SetMotorMode 33 0
SetMotorMode 34 0
SetMotorMode 35 0
SetMotorMode 36 0
SetMotorMode 37 0
Robot initialization setting succeeded
The detected color numbers are:1
Return : 1
Lattice setting is successful
RGB lights are set successfully
The detected color numbers are:3
Return : 3
Lattice setting is successful
RGB lights are set successfully
The detected color numbers are:2
Return : 2
Lattice setting is successful
RGB lights are set successfully
The detected color numbers are:4
```

After running the program, the spider robot will make different actions and expressions when facing squares of different colors.

Color of squares	Action
Red square	The spider robot enters the combat state, the RGB light is red, and the dot matrix displays the letter "R".
Green square	The spider robot twists its body, the RGB lights turn green, and the dot matrix displays the letter "G".
Blue square	The spider robot wagging its tail, the RGB light turns blue, and the dot matrix displays the letter "B".

6.10.2.10 Obstacle Avoidance Spider Robot

(1) Function realization: The spider robot walks forward, and the ultrasonic sensor detects whether there is an obstacle ahead. After identifying the obstacle, the spider robot bypasses the obstacle through the pre-set path planning and continues to walk in the initial direction of movement, and the dot matrix displays different faces facial expressions. Note that the obstacles identified by this program should be selected from the aluminum leather case that comes with the product. If other obstacles are selected, the spider's path planning gait needs to be adjusted by itself.

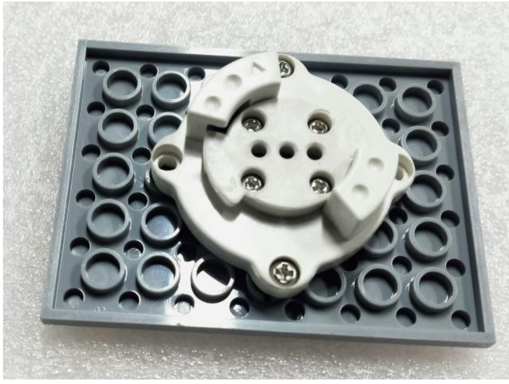
(2) Hardware Preparation

1 P module, 8 T modules, 4 bionic feet, 4 orthogonal modules, 1 MINI PRO sensor main control board module, 1 ultrasonic sensor module and 1 dot matrix module.



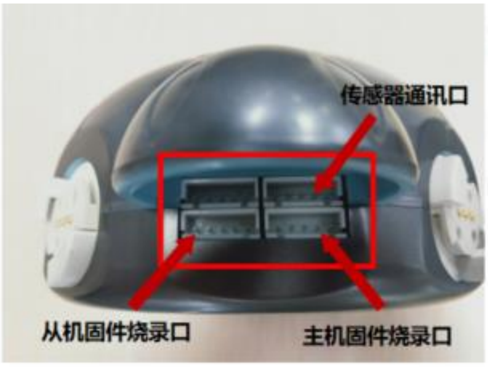


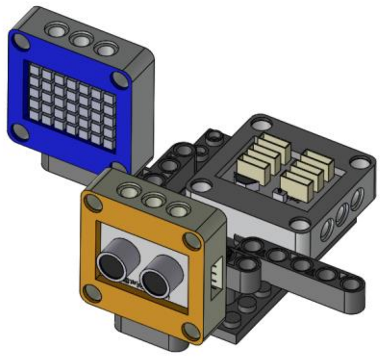
The sensor fixing plate of the P module is used to be fixedly connected to the top interface of the P module to facilitate the installation of the sensor.

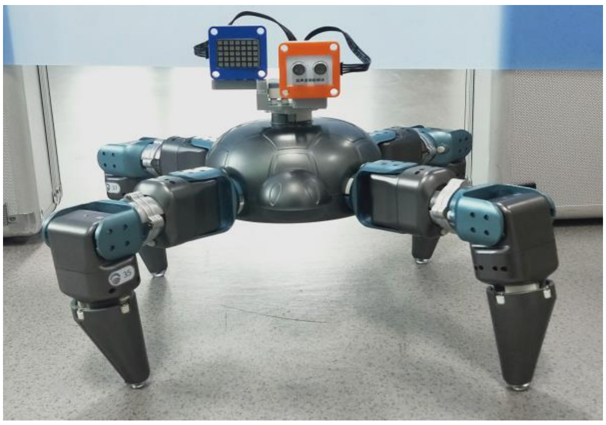


(3) Construction of the Configuration

Connect the module and sensor according to the diagram below. Among them, the ultrasonic is connected to the D3 port of the main control board, and the dot matrix is connected to the D4 port of the main control board. The interface between the P module and the main control board is the serial port. For details, see the sensor control example in Chapter 5.







(4) Example Execution

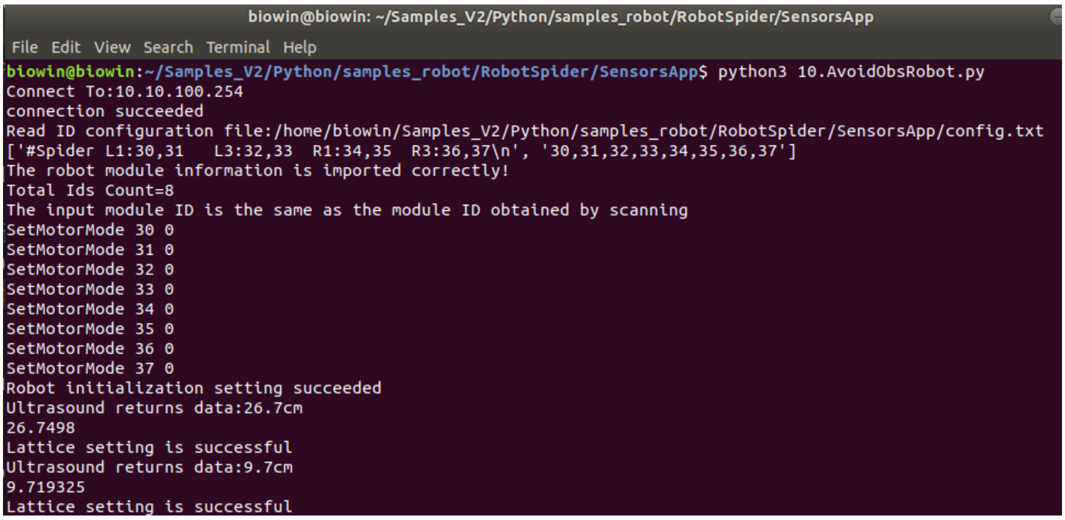
Steps: Turn on the P module switch, and connect the PC to the hotspot corresponding to the P module, which is generally biowinP_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp/

Right-click to open the terminal and execute the Python command:

$ python3 10.AvoidObsRobot.py

Operation result:



After running the program, the ultrasonic sensor starts to detect whether there is an obstacle ahead, and if so, it will follow the preset action to avoid the obstacle. If not, move forward, and the dot matrix displays the corresponding expression at the same time.

6.10.2.11 Turn Right to Avoid Obstacles

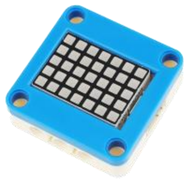
(1) Function realization: The spider robot walks forward, and the ultrasonic sensor detects whether there is an obstacle ahead. After identifying the obstacle, the spider robot turns to the right, avoids the obstacle and continues to move forward, and the dot matrix displays different expressions.

(2) Hardware Preparation

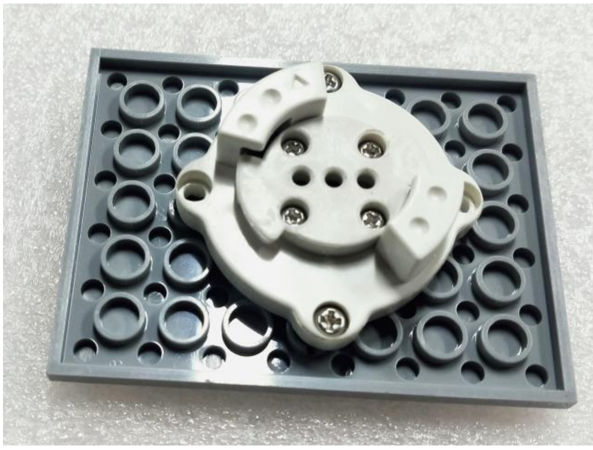
1 P module, 8 T modules, 4 bionic feet, 4 orthogonal modules, 1 MINI PRO

sensor main control board module, 1 ultrasonic sensor module and 1 dot matrix module.



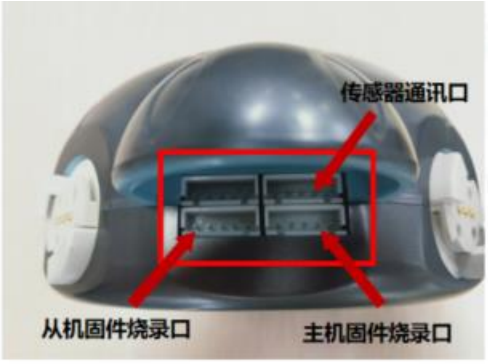


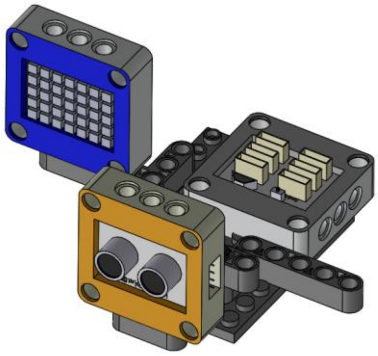

The sensor fixing plate of the P module is used to be fixedly connected to the top interface of the P module to facilitate the installation of the sensor.

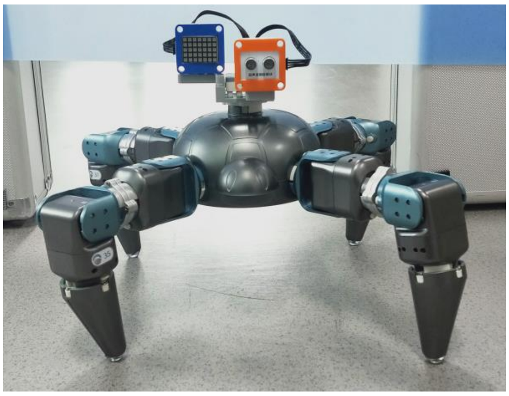


(3) Construction of the Configuration

Connect the module and sensor according to the diagram below. Among them, the ultrasonic is connected to the D3 port of the main control board, and the dot matrix is connected to the D4 port of the main control board. The interface between the P module and the main control board is the serial port. For details, see the sensor control example in Chapter 5.







(4) Example Execution

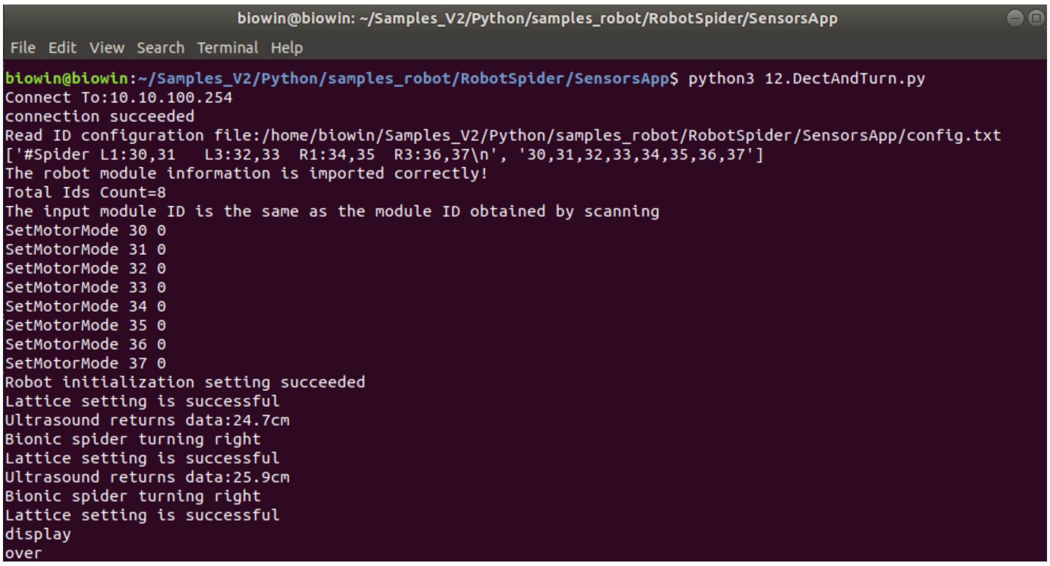
Steps: Turn on the P module switch, and connect the PC to the hotspot corresponding to the P module, which is generally biowinP_xxx.

Go to:/Home/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp/

Right-click to open the terminal and execute the Python command:

$ python3 12.DectAndTurn.py

Operation result:


```
biowin@biowin: ~/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp
File  Edit  View  Search  Terminal  Help
biowin@biowin:~/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp$ python3 12.DectAndTurn.py
Connect To:10.10.100.254
connection succeeded
Read ID configuration file:/home/biowin/Samples_V2/Python/samples_robot/RobotSpider/SensorsApp/config.txt
['#Spider L1:30,31   L3:32,33  R1:34,35  R3:36,37\n', '30,31,32,33,34,35,36,37']
The robot module information is imported correctly!
Total Ids Count=8
The input module ID is the same as the module ID obtained by scanning
SetMotorMode 30 0
SetMotorMode 31 0
SetMotorMode 32 0
SetMotorMode 33 0
SetMotorMode 34 0
SetMotorMode 35 0
SetMotorMode 36 0
SetMotorMode 37 0
Robot initialization setting succeeded
Lattice setting is successful
Ultrasound returns data:24.7cm
Bionic spider turning right
Lattice setting is successful
Ultrasound returns data:25.9cm
Bionic spider turning right
Lattice setting is successful
display
over
```

After running the program, the spider robot starts to walk forward, and when the ultrasonic sensor detects an obstacle ahead, it turns to the right to avoid the obstacle instead of crawling forward.

Note: Enter the "ctrl+c" key twice to stop the movement.

7 AI Example

7.1 Face Recognition

(1) Function realization: The robotic arm rotates back and forth within a certain range to scan the face. If a face appears, the robotic arm will make a greeting.

(2) Hardware Preparation:

1 USB camera, 1 camera fixing suction cup, 1 F main control module, 1 ITTTIG 5-DOF operating arm and 1 F module base.

(3) Software requirements: Linux system, python3, python-opencv-contrib, PIL, numpy, threading environment and experimental code (including configuration path).

(4) Configuration construction: Complete the construction of the F module, robotic arm, and F module base according to the diagram. Fix the camera fixing suction cup to the camera, and attach the suction cup to the I module (the I module connected with the G module), and connect the USB interface of the USB camera to the computer.

(5) Example Execution:

Steps:

1. Use the "Ctrl+Alt+T" keys to open a terminal;

Go to:/Home/Samples-V2/Python/samples_AI/1.Face_recognition(The actual path needs to be modified according to the code storage location);

2. Go to the "1.Face_recognition" folder;

3. Note: If you need to record a specific face database, go to step 4. If there is no need to record a specific face database, go to step 7;

4. Run 1_Saveimagedata.py with pycharm, enter the name and number of the imported portrait data;

```
biowin@biowin:~/Samples-V2/Python/Sample_AI/1.Face_recognition$ python3 1_Saveim
agedata.py
Enter name:
xie
Enter number:
1
```

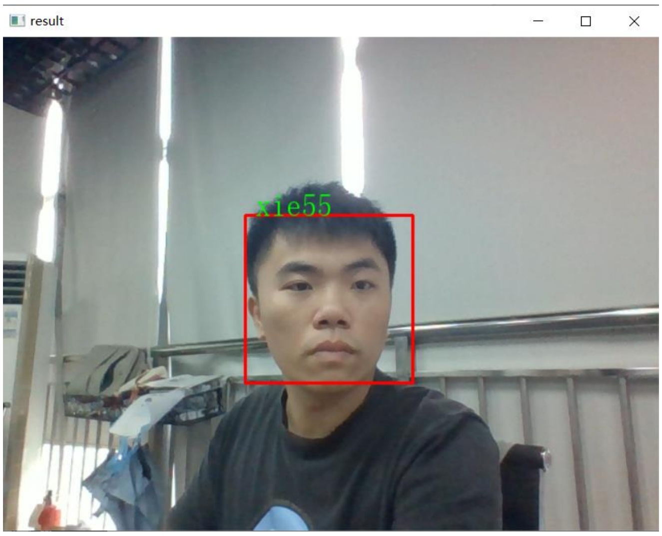
5. Click on the camera display window with the mouse and press the "S" key to take pictures and save the portrait data to be imported. Portrait data should be taken from multiple angles (the more the number, the higher the recognition accuracy). Press the "Q" key to end taking pictures;

6. Run 2_CreatLBPH.py on the terminal to perform feature value recognition on the imported face data;

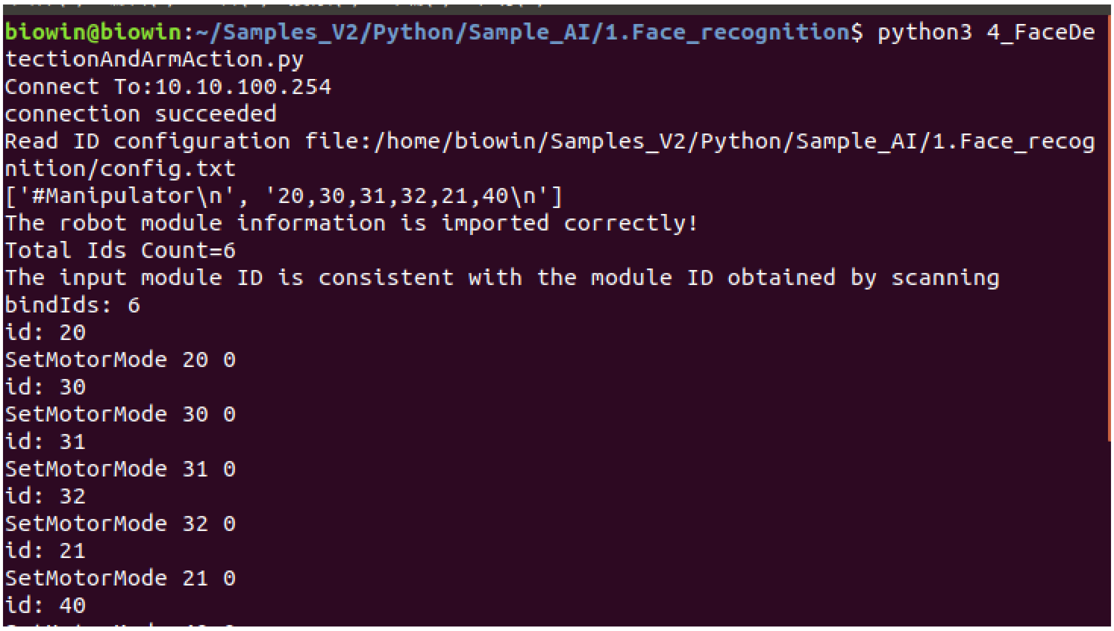
```
biowin@biowin:~/Samples-V2/Python/Sample_AI/1.Face_recognition$ python3 2_CreatL
BPH.py
```

7. Run 3_OpenRecognition.py on the terminal to recognize the face, and display the corresponding name for the face with imported data. Display "Other Person" for faces without imported data;

```
biowin@biowin:~/Samples-V2/Python/sample_AI/1.Face_recognition$ python3 3_OpenRe
cognition.py
```

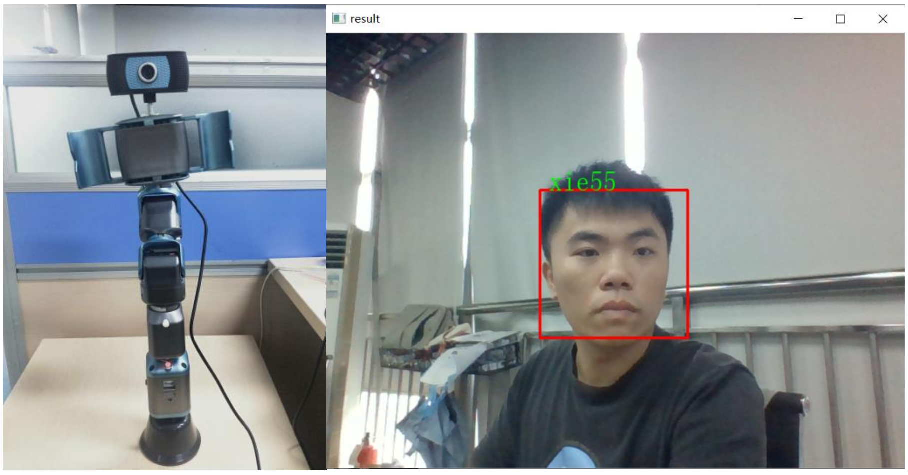


8. Turn on the switch of the F module and connect the WIFI of the F module with a computer, run 4_FaceDetectionAndArmAction.py on the terminal, identify the face, display the corresponding name for the face with imported data, and display "other people" for the face without imported data. When a face is recognized, the robotic arm will perform two greeting actions;



9. Click the image window with the mouse and press the "Q" key to exit.

(6) Experimental effect



The robotic arm rotates back and forth to scan the face within a certain range. When the face is detected, it will use a display frame to frame the face (by taking a photo of a specific face in advance and importing it into the data area, the specific face can be visually recognized and displayed in the image. The name of a specific face is displayed in the image, and the face without the imported face image is displayed as other person), and the robotic arm makes a greeting after recognizing the face.

(7) Precautions

1. After inserting the USB camera, since each computer sorts the cameras differently, if you find that the field of view that appears after running the program is not the field of view of the USB camera you are using, try to change "cap = cv2.VideoCapture(1)" in the code to "cap = cv2.VideoCapture(0)";

2. It is recommended that the recognized face should be no more than 2 m away from the camera, and make sure that the entire face appears in the camera's field of view.

7.2 Color Recognition

(1) Function realization: Record the HSV parameters of the color by adjusting the HSV range of different colors by yourself, and identify the corresponding color by changing the parameters.

(2) Hardware requirements: 1 USB camera, 1 camera fixing suction cup, 1 F main control module, 1 ITTTIG 5-DOF operating arm, 1 F module base, 1 red, green and blue squares each.

(3) Software requirements: Linux system, python3, python-opencv-contrib, PIL, numpy data package environment and experimental code (including configuration path).

(4) Configuration construction: Complete the construction of the F module, robotic arm, and F module base according to the diagram. Fix the camera fixing suction cup to the camera, and attach the suction cup to the I module (the I module connected with the G module), and connect the USB interface of the USB camera to the computer.


(5) Example Execution:

Steps:

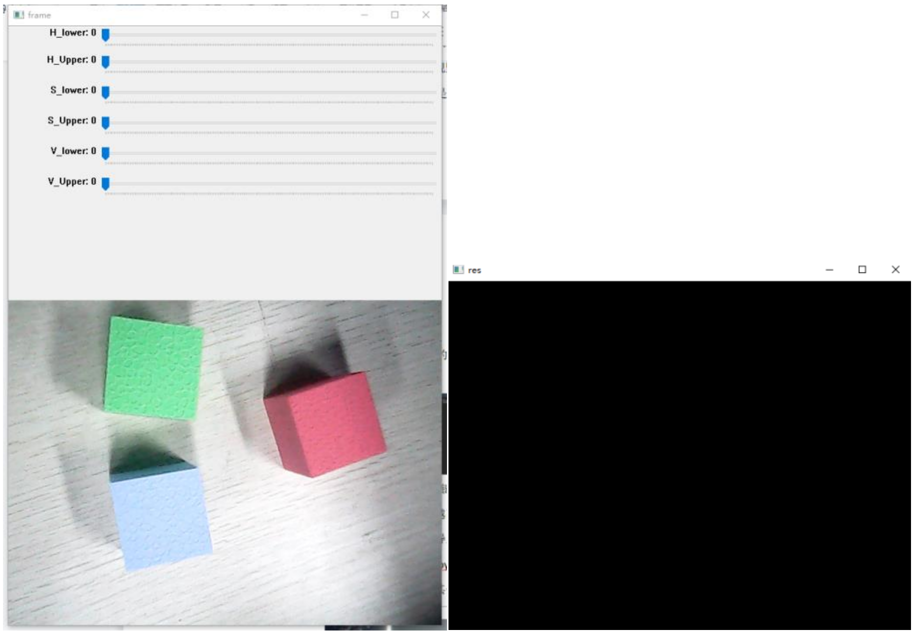
1. Use the "Ctrl+Alt+T" keys to open a terminal;

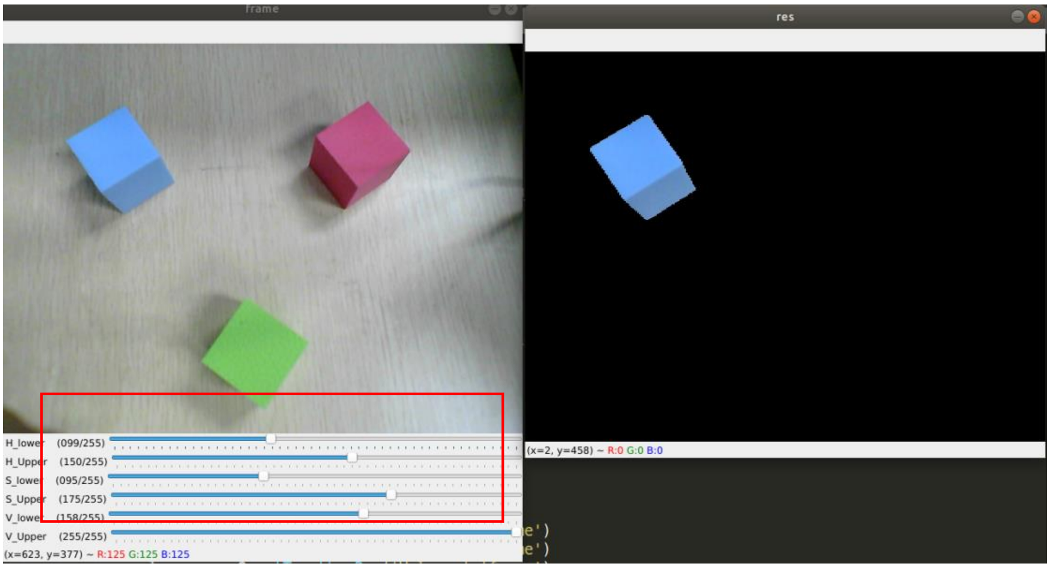
2. Go to:/Home/Samples-V2/Python/samples_AI/2.Color_recognition(The actual path needs to be modified according to the code storage location);

3. Turn on the switch of the F module and connect the WIFI of the F module with a computer, and run Color_RecordAndRecognitionWithArm.py in the terminal;

```
biowin@biowin:~/Samples-V2/Python/sample_AI/2.Color_recognition$ python3 Color_RecordAndRecognitionWithArm.py
```

4．Adjust the HSV high and low thresholds by dragging the 6 sliders until only the colors you want to record appear in the "res" window. Click the image window and press the "S" key to record the HSV value of a specific color, enter the name of the color on the terminal and press Enter to confirm;

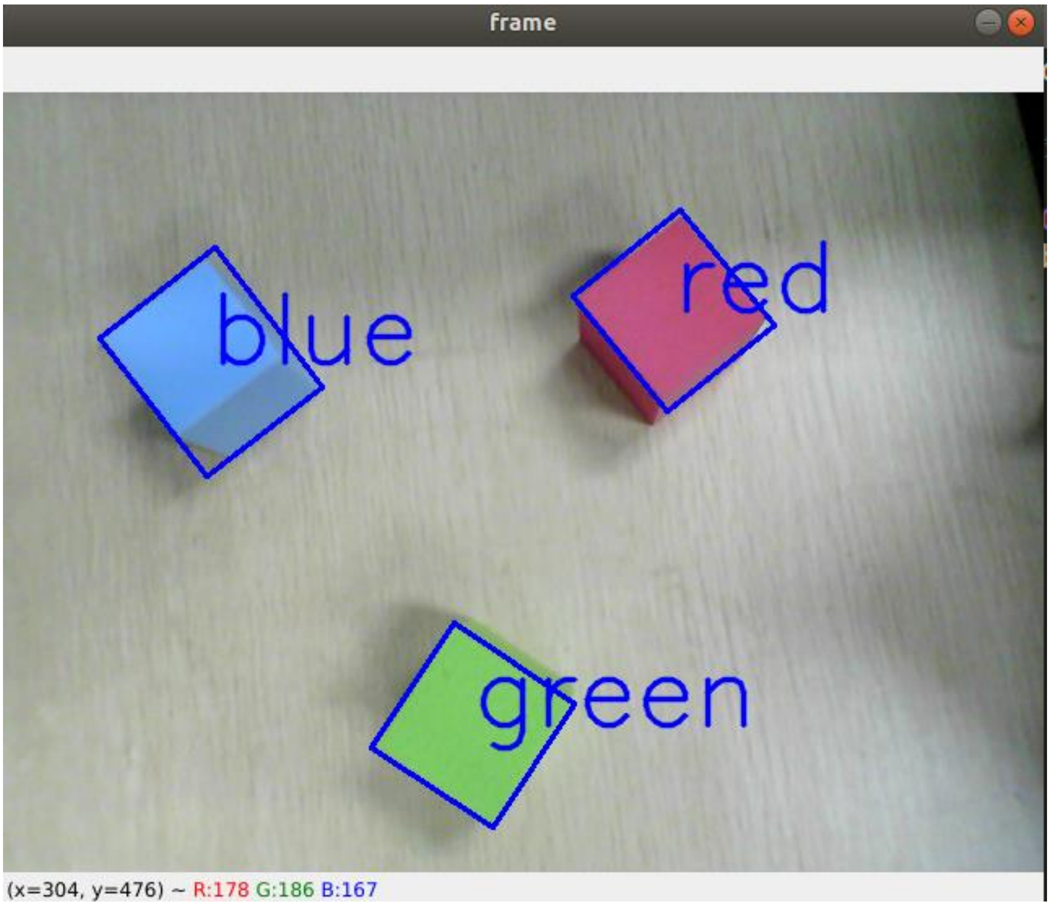




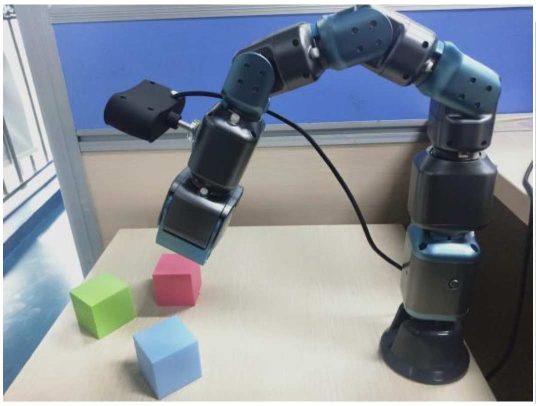
```
biowin@biowin:~/Samples_V2/Python/Sample_AI/2.Color_recognition$ python3 Color_RecordAndRecognitionW
ithArm.py
Connect To:10.10.100.254
Read ID configuration file:/home/biowin/Samples_V2/Python/Sample_AI/2.Color_recognition/config.txt
['#Manipulator\n', '20,30,31,32,21,40\n']
The robot module information is imported correctly!
Total Ids Count=6
The input module ID is the same as the module ID obtained by scanning
bindIds: 6
id: 20
SetMotorMode 20 0
id: 30
SetMotorMode 30 0
id: 31
SetMotorMode 31 0
id: 32
SetMotorMode 32 0
id: 21
SetMotorMode 21 0
id: 40
SetMotorMode 40 0
Adjust the slider so that the color to be recorded is displayed in the res window, and press 's' to
save, and press 'esc' to exit the recording:
```

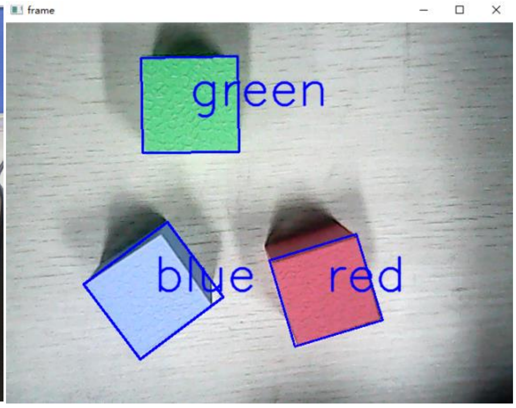
5. After recording the desired color, click the image window and press the "Esc" key to end the photo recording (no need to record and press the "Esc" key to jump to the color recognition step directly);

6. Put the three colored squares in the camera's field of view, the image will recognize the recorded color, and display the recognized color name, press the "Esc" key to exit the recognition.



(6) Experimental effect:





The robotic arm bends downwards so that the camera's field of view is on the table, and the red, green and blue squares are placed on the table, and specific colors are identified by dragging the high and low thresholds of HSV.

(7) Precautions

1.After inserting the USB camera, since each computer sorts the cameras differently, if you find that the field of view that appears after running the program is not the field of view of the USB camera you are using, try to change "cap = cv2.VideoCapture(1)" in the code to "cap = cv2.VideoCapture(0)";

2.When recording colors, try to ensure that there are sufficient light sources (white lights), and the lighting conditions during color recognition should be consistent with those during color recording, otherwise there will be color recognition errors;

3.Please do not use Chinese characters for the recorded color names.

7.3 Color Sorting

(1) Function implementation: Randomly place squares of different colors in the color screening area on the map (record the HSV value of the color of the square in advance through Section 7.2). Use the mouse to select the square to be clamped, and the robot grabs the square into the square placement area of the corresponding color.

(2) Hardware requirements: 1 USB camera, 1 camera fixing bracket, 1 color sorting map, 1 F main control module, 1 ITTTIG 5-DOF operating arm, 1 F module

base, 1 red, green and blue squares each, 1 sheet of 11x7 checkerboard (1.5cm side length, in the doc folder).

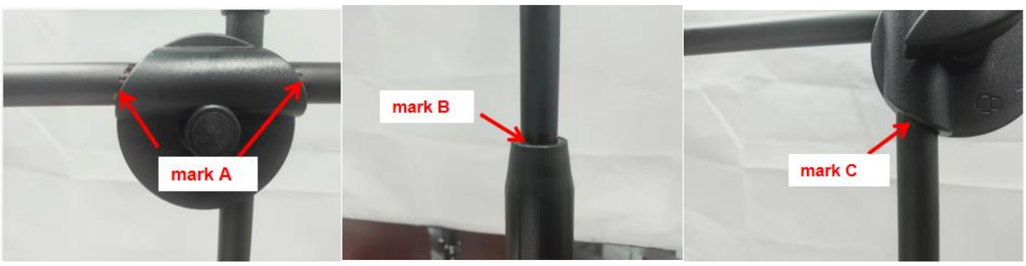
(3) Software requirements: Linux system, python3, python-opencv-contrib, numpy data package environment, experimental code (including configuration path).

(4) Configuration construction: Complete the construction of the F module, robotic arm, and F module base according to the diagram. Fix the camera on the camera stand as shown below (pay attention to the height and width of the camera stand), and connect the USB port of the USB camera to the computer.

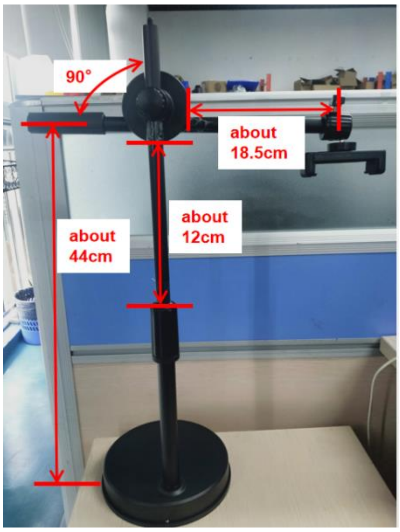
1.Take out the four accessories of the camera stand (the accessories are named 1-4 respectively);

2.Complete the build of the camera fixing bracket according to the following figure, which needs to be noted:

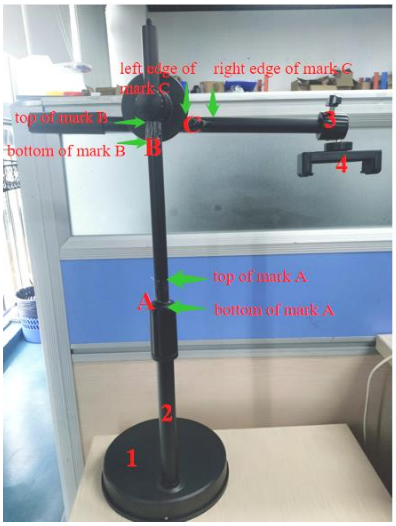
①There are three marking lines in the camera bracket (marking A, B, C respectively, after testing, this position can better achieve the experimental effect);



②Fit the bracket parts of the fixed bracket to the marks A, B, and C according to the above figure. The specific effects are as follows:



③Part 4 needs to be straight down.



3.Pull the camera part 4, use it to clamp the camera, pay attention to the installation direction of the camera, as shown in the figure below. Make sure that the direction of the camera's USB cable and the vertical rod of the camera's fixing bracket are not on the same side (otherwise the photos taken are opposite);

4.Fix the 5-DOF robotic arm on the map through the F module base, and place the built camera fixing bracket (including the camera) in the "D area" of the map;

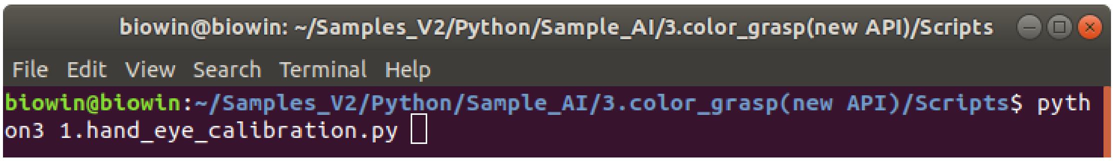
(5) Example Execution:

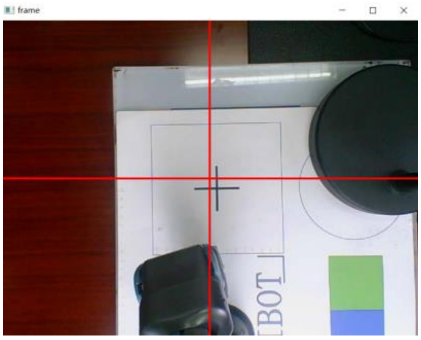
Steps:

1. Use the "Ctrl+Alt+T" keys to open a terminal;

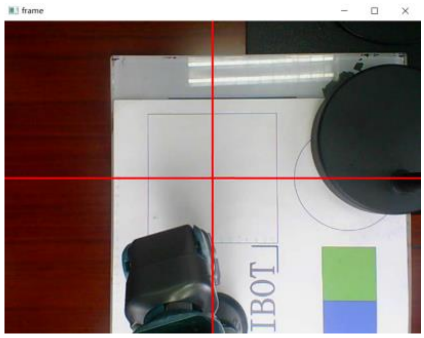
Go to:/Home/Samples-V2/Python/samples_AI/3.color_grasp/Scripts(The actual path needs to be modified according to the code storage location);

2. Run 1.hand_eye_calibration.py in the terminal, and the "red cross" image as shown in the figure appears in the window. Adjust the position of the camera stand base so that the "red cross" exactly coincides with the "black cross" on the map. After the adjustment is completed, click the image display window, and press the "Q" key to exit;



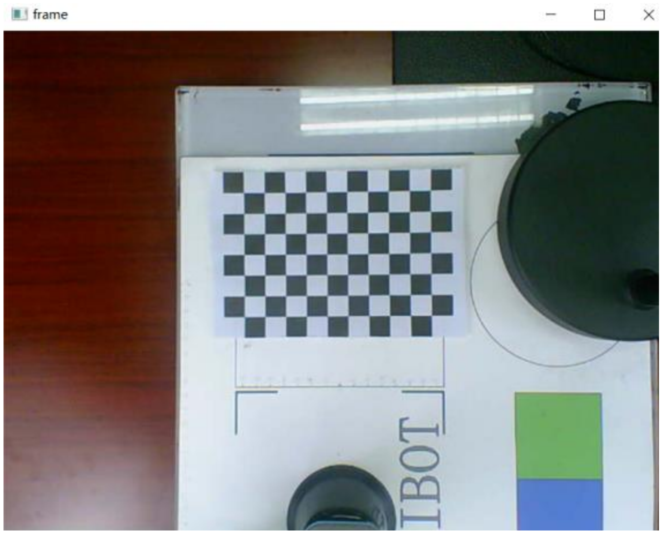


Before adjustment



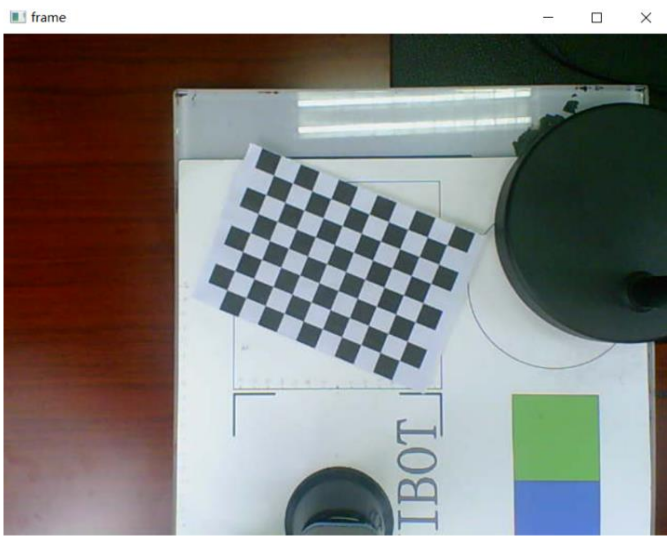
After adjustment

3. Run 2.save_picture.py on the terminal, and put the entire checkerboard in the camera's field of view, and try to ensure that the center of the checkerboard coincides with the "black cross" on the map;

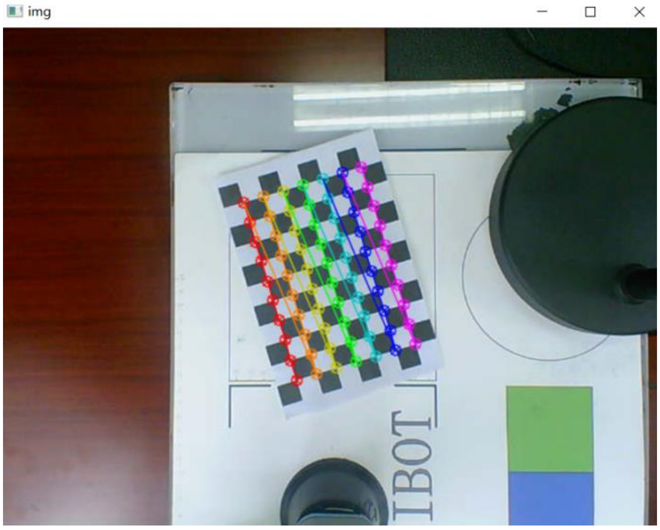


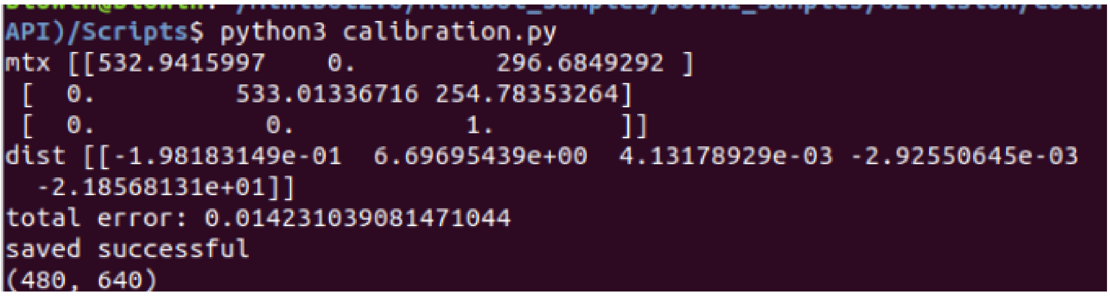
4. Click on the image display window and press the "S" key to save the image. Enter the number "1" on the output terminal, name the photo just taken, and press the "Enter" key to end the saving of the first photo;

5. According to the operation of the previous step, take pictures of the checkerboard 14 times(Note: Each time you press the "S" key to take a photo, you need to click the image display window to take a photo to be effective. There is also the need to rotate the angle of the checkerboard before each photo is taken, so that the angle of the checkerboard in the 14 photos is different), after saving 14 pictures, click the image display window and press the "Q" key to end;

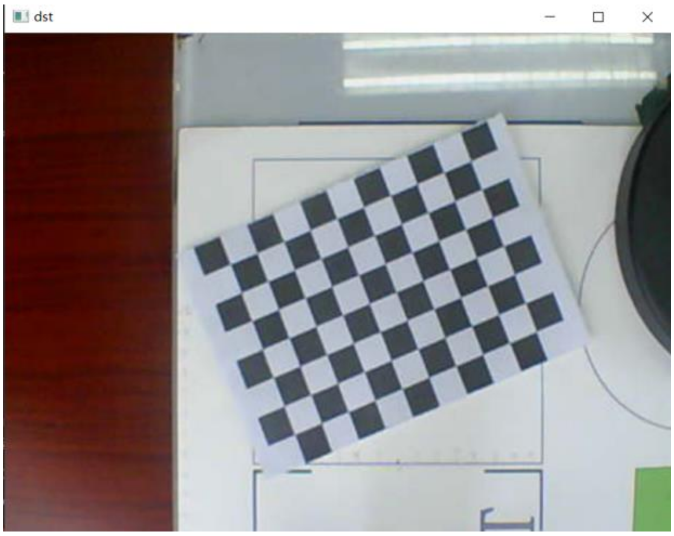


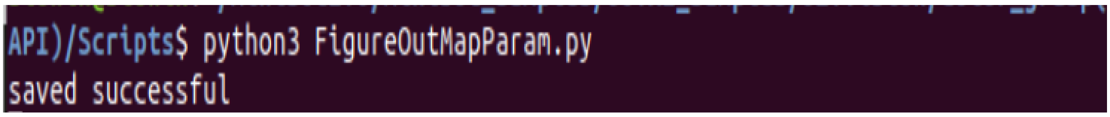
6. Run 3.calibration.py on the terminal to calibrate the internal and external parameters of the camera;





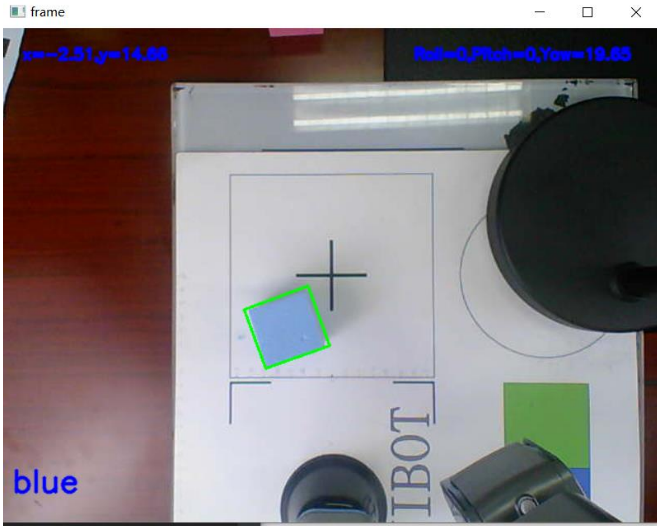
7. Run 4.FigureOutMapParam.py on the terminal to make the entire checkerboard appear in the image display window. Click the image display window, press "Space" to calculate the length of the checkerboard, if "saved successful" appears, the calculation is successful, and press "Esc" to end;





8. Turn on the switch of the F module and connect the WIFI of the F module with a computer, and run 5.color_grasp.py on the terminal;

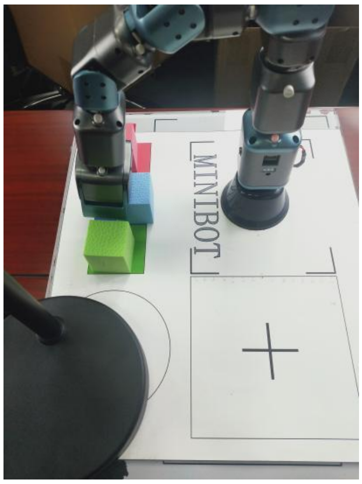
9. Double-click the square to be grabbed in the image window with the mouse. After the selection is successful, the image will use a green square to frame the square to be grabbed (sometimes the square recognition is not accurate, you can consider re-running the "color recognition" program in section 7.2 , redefine the HSV value of the color, and ensure that the lighting brightness of the environment when recording the color is consistent with the lighting brightness of the visual capture), after the recognition is successful, press the "Enter" key, and the robotic arm will grab the selected square. And place the square in the corresponding color area (this DEMO provides the identification of red, green and blue squares, if you want to identify other squares, you need to run the "color recognition" program to record the HSV file of the square);




```
    <color name="blue">
        <X>-5.84</X>
        <Y>13.1</Y>
        <yaw>-29.54</yaw>
    </color>
    <color name="green">
        <X>-3.94</X>
        <Y>15.01</Y>
        <yaw>-15.42</yaw>
    </color>
</colordata>
{'red': (0.0, 0.0, 0.0), 'blue': (-5.84, 13.1, -29.54), 'green': (-3.94, 15.01, -15.42)}
Posture to be solved:[-58.4, 131.0, 30, 114.02738576632223, 180, 0]
[-58.4, 131.0, 30, 114.02738576632223, 180, 0]-The pose of this group has no solution
===============================
Posture to be solved:[-58.4, 131.0, 30, 114.02738576632223, 179, 0]
Alternative solution:[-65.97261423367777, 95.82606471245703, 8.109614455994185, 68.7811355243692, 0]
Posture to be solved:[-58.4, 131.0, 30, 114.02738576632223, 178, 0]
Alternative solution:[-65.97261423367777, 88.66832306731993, 23.073918065630973, 59.974573559869505, 0]
Available solutions:[-65.97261423367777, 88.66832306731993, 23.073918065630973, 59.974573559869505, 0]
angle2 = %f [-65.97261423367777, 88.66832306731993, 23.073918065630973, 59.974573559869505, -36.432614233677775]
color is %s blue
2
```

10. After finishing the experiment, click on the image display window and press "Esc" to exit.

(6) Experimental effect:



Place squares of different colors anywhere in the square grab area (the box with the "cross" below the camera). The robotic arm grabs the square by identifying the position of the square, and places it on the corresponding color placement area according to the color of the grabbed square.

(7) Precautions

1. After inserting the USB camera, since each computer sorts the cameras differently, if you find that the field of view that appears after running the program is not the field of view of the USB camera you are using, try to change "cap = cv2.VideoCapture(1)" in the code to "cap = cv2.VideoCapture(0)";

2. When recording colors, try to ensure that there are sufficient light sources (white lights), and the lighting conditions during color recognition should be consistent with those during color recording, otherwise there will be color recognition errors;

3. After testing, in the grabbing area of the "black cross", the closer the distance between the square and the robotic arm, the higher the success rate of grabbing;

4. When the mouse selects the square to be grabbed, it is necessary to ensure that the green box in the image display area can fit the square, otherwise the grabbing effect will be affected. When identifying, it is necessary to ensure that the ambient light of the "visual capture" and "color identification" experiments are basically the same, otherwise, the "color identification" program needs to be re-run to record the HSV value of the color.

7.4 Gesture Recognition

(1) Function realization: Control the movement of each joint of the robot by recognizing gestures, and make greetings and nods.

(2) Hardware requirements: 1 USB camera, 1 camera fixing suction cup, 1 F main control module, 1 ITTTIG 5-DOF operating arm, 1 F module base.

(3) Software requirements: Software requirements: Linux system, python3, python-opencv-contrib, sklearn, threading data package environment, experimental

code (including configuration path).

(4) Configuration construction: Complete the construction of the F module, robotic arm, and F module base according to the diagram. Fix the camera fixing suction cup to the camera, and attach the suction cup to the I module (the I module connected with the G module), and connect the USB interface of the USB camera to the computer.

(6) Example Execution:

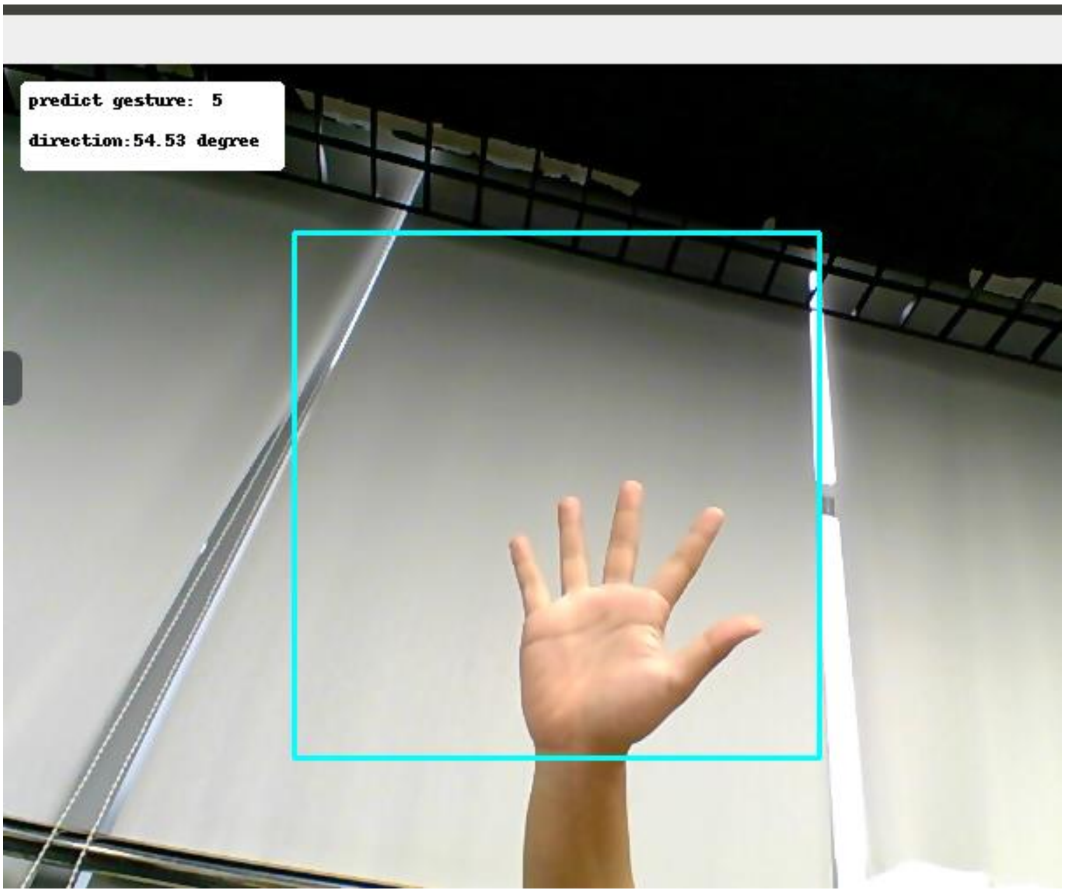
Steps:

1. Use the "Ctrl+Alt+T" keys to open a terminal;

Go to:/Home/Samples-V2/Python/samples_AI/4.Gesture_recognition(The actual path needs to be modified according to the code storage location);

2. Enter "pip install sklearn" in the terminal to install the corresponding library (necessary to run the program for the first time);

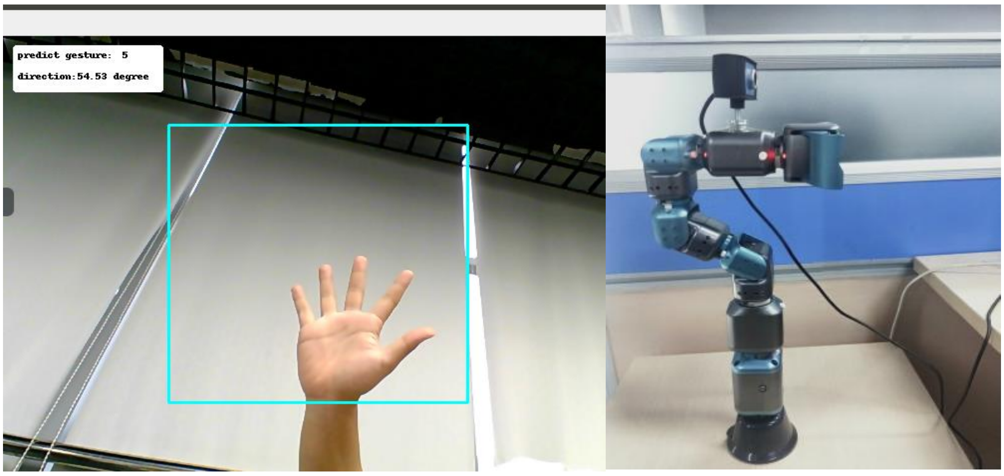
3. Run 1_inference_gesture.py in the terminal (for gesture recognition);

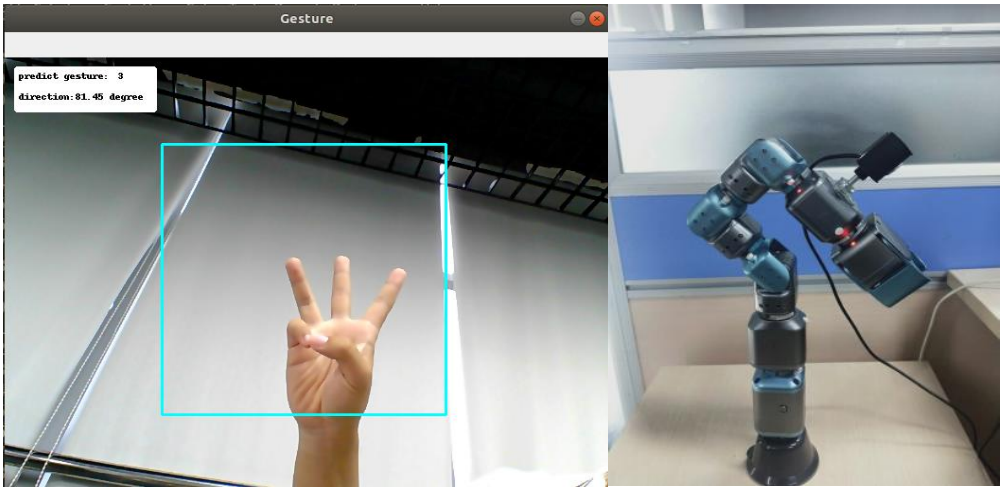


4. Turn on the switch of the F module and connect the WIFI of the F module with a computer, and run 2_inference_gesture_with_arm.py on the terminal (gesture recognition combined with robot motion);

5. Click on the image window and press "Esc" to exit.

(6) Experimental effect:





The robotic arm performs initial actions and recognizes the gestures in the camera's field of view. The recognition results are displayed in the upper left of the image window, and the robotic arm will perform corresponding actions for different gestures recognized. The specific actions are as follows.

Gesture	Action
"1"	The first module movement
"2"	The second module movement
"3"	The third module movement
"4"	The fourth module movement
"5"	The fifth module movement

(7) Precautions

After inserting the USB camera, since each computer sorts the cameras differently, if you find that the field of view that appears after running the program is not the field of view of the USB camera you are using, try to change "cap = cv2.VideoCapture(1)" in the code to "cap = cv2.VideoCapture(0)".

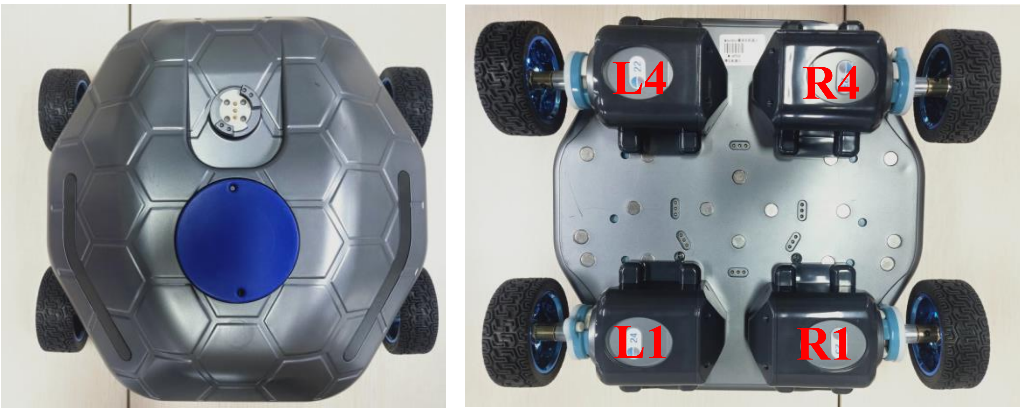
7.5 Speech Recognition

(1) Function realization: The program recognizes the voice content in the audio file, and controls the movement of the car according to the voice content.

(2) Hardware requirements: M module, 4 Mecanum wheel assemblies, 4 ordinary wheel assemblies, 2 universal wheels, 4 I packages, 4 I modules.

(3) Software requirements: Linux system, python3 environment, pygame, numpy, wine environment, experimental code (including configuration path)

(4) Experimental hardware connection diagram: By installing the ordinary wheel (including I module) on the L1, L4, R1, R4 serial port positions, as shown in the figure:



(5) Execution example:

step:

1. Use the "Ctrl+Alt+T" keys to open a terminal;

2. Go to:/Home/Samples-V2/Python/samples_AI/5.Voicekeywords_lite (the actual path needs to be modified according to the code storage location)

3. Enter sudo apt-get install wine64 in the terminal to install the corresponding library (necessary to run the program for the first time);

4. Turn on the switch of the M module and connect the WIFI of the M module with a computer, and run main.py on the terminal (speech recognition combined with robot motion);

(6) Experimental effect

```
biowin@biowin:~/Samples_V2/Python/Sample_AI/5.Voicekeywords_lite$ python3 main.py
pygame 2.1.2 (SDL 2.0.16, Python 3.6.9)
Hello from the pygame community. https://www.pygame.org/contribute.html
Connect To:10.10.100.254
连接成功
Total Ids Count=7
SetMotorMode 23 1
SetMotorMode 24 1
SetMotorMode 25 1
SetMotorMode 22 1
Z:\home\biowin\Samples_V2\Python\Sample_AI\5.Voicekeywords_lite\HTK_EndPointedVoice\hco
py.exe -A -D -T 1 -C tr_wav.cfg -S ./list_command.scp

HTK Configuration Parameters[21]
  Module/Tool      Parameter                  Value
# HREC             FORCEOUT                    TRUE
# HNET             TRACE                          1
# HLABEL           TRACE                          8
# HPARM            TRACE                         65
# HSHELL           TRACE                          2
#                  FORCECXTEXP                 TRUE
#                  ALLOWXWRDEXP                TRUE
#                  ENORMALIZE                  TRUE
#                  NUMCEPS                       12
#                  CEPLIFTER                     22
#                  NUMCHANS                      26
#                  PREEMCOEF               0.970000
#                  USEHAMMING                  TRUE
#                  WINDOWSIZE         250000.000000
#                  SAVEWITHCRC                 TRUE
#                  SAVECOMPRESSSED              TRUE
#                  TARGETRATE         100000.000000
#                  TARGETKIND            MFCC_E_D_A_Z
#                  ZMEANSOURCE                FALSE
#                  SOURCEFORMAT                 WAV
#                  SOURCEKIND              WAVEFORM

HParm: Parm tab type MFCC_E_D_A_K_Z saved to ..\commands\mfcc\command.mfc [sampSize=156
,nSamples=86] with CRC
..\commands\voice_endpoint\command.wav -> ..\commands\mfcc\command.mfc
```

```
HTK Configuration Parameters[21]
  Module/Tool       Parameter                  Value
# HREC              FORCEOUT                    TRUE
# HNET              TRACE                          1
  HLABEL            TRACE                          8
  HPARM             TRACE                         65
  HSHELL            TRACE                          2
#                   FORCECXTEXP                 TRUE
#                   ALLOWXWRDEXP                TRUE
#                   ENORMALIZE                  TRUE
                    NUMCEPS                       12
                    CEPLIFTER                     22
                    NUMCHANS                      26
                    PREEMCOEF               0.970000
                    USEHAMMING                  TRUE
                    WINDOWSIZE         250000.000000
                    SAVEWITHCRC                 TRUE
#                   SAVECOMPRESSSED              TRUE
                    TARGETRATE         100000.000000
                    TARGETKIND           MFCC_E_D_A_Z
                    ZMEANSOURCE                FALSE
                    SOURCEFORMAT                 WAV
                    SOURCEKIND              WAVEFORM

        识别结果：右转

四并橡胶轮右转...
Z:\home\biowin\Samples_V2\Python\Sample_AI\5.Voicekeywords_lite\HTK_EndPointedVoice\hco
py.exe -A -D -T 1 -C tr_wav.cfg -S ./list_command.scp
```

```
HTK Configuration Parameters[21]
  Module/Tool       Parameter                  Value
# HREC              FORCEOUT                    TRUE
# HNET              TRACE                          1
  HLABEL            TRACE                          8
  HPARM             TRACE                         65
  HSHELL            TRACE                          2
#                   FORCECXTEXP                 TRUE
#                   ALLOWXWRDEXP                TRUE
#                   ENORMALIZE                  TRUE
                    NUMCEPS                       12
                    CEPLIFTER                     22
                    NUMCHANS                      26
                    PREEMCOEF               0.970000
                    USEHAMMING                  TRUE
                    WINDOWSIZE         250000.000000
                    SAVEWITHCRC                 TRUE
#                   SAVECOMPRESSSED              TRUE
                    TARGETRATE         100000.000000
                    TARGETKIND           MFCC_E_D_A_Z
                    ZMEANSOURCE                FALSE
                    SOURCEFORMAT                 WAV
                    SOURCEKIND              WAVEFORM

        识别结果：向左

四并橡胶轮左转...
Z:\home\biowin\Samples_V2\Python\Sample_AI\5.Voicekeywords_lite\HTK_EndPointedVoice\hco
py.exe -A -D -T 1 -C tr_wav.cfg -S ./list_command.scp
```

```
HTK Configuration Parameters[21]
  Module/Tool      Parameter                  Value
# HREC             FORCEOUT                    TRUE
# HNET             TRACE                          1
  HLABEL           TRACE                          8
  HPARM            TRACE                         65
  HSHELL           TRACE                          2
#                  FORCECXTEXP                 TRUE
#                  ALLOWXWRDEXP                TRUE
#                  ENORMALIZE                  TRUE
                   NUMCEPS                       12
                   CEPLIFTER                     22
                   NUMCHANS                      26
                   PREEMCOEF               0.970000
                   USEHAMMING                  TRUE
                   WINDOWSIZE         250000.000000
                   SAVEWITHCRC                 TRUE
#                  SAVECOMPRESSSED              TRUE
                   TARGETRATE         100000.000000
                   TARGETKIND           MFCC_E_D_A_Z
                   ZMEANSOURCE                FALSE
                   SOURCEFORMAT                 WAV
                   SOURCEKIND              WAVEFORM

        识别结果: 前进

四并橡胶轮向前走...
Z:\home\biowin\Samples_V2\Python\Sample_AI\5.Voicekeywords_lite\HTK_EndPointedVoice\hco
py.exe -A -D -T 1 -C tr_wav.cfg -S ./list_command.scp
```

```
HTK Configuration Parameters[21]
  Module/Tool      Parameter                  Value
# HREC             FORCEOUT                    TRUE
# HNET             TRACE                          1
  HLABEL           TRACE                          8
  HPARM            TRACE                         65
  HSHELL           TRACE                          2
#                  FORCECXTEXP                 TRUE
#                  ALLOWXWRDEXP                TRUE
#                  ENORMALIZE                  TRUE
                   NUMCEPS                       12
                   CEPLIFTER                     22
                   NUMCHANS                      26
                   PREEMCOEF               0.970000
                   USEHAMMING                  TRUE
                   WINDOWSIZE         250000.000000
                   SAVEWITHCRC                 TRUE
#                  SAVECOMPRESSSED              TRUE
                   TARGETRATE         100000.000000
                   TARGETKIND           MFCC_E_D_A_Z
                   ZMEANSOURCE                FALSE
                   SOURCEFORMAT                 WAV
                   SOURCEKIND              WAVEFORM

        识别结果: 后退

四并橡胶轮向后走...
Z:\home\biowin\Samples_V2\Python\Sample_AI\5.Voicekeywords_lite\HTK_EndPointedVoice\hco
py.exe -A -D -T 1 -C tr_wav.cfg -S ./list_command.scp
```

```
HParm: Parm tab type MFCC_E_D_A_K_Z saved to ..\commands\mfcc\command.mfc [sampSize=156
,nSamples=80] with CRC
..\commands\voice_endpoint\command.wav -> ..\commands\mfcc\command.mfc

HTK Configuration Parameters[21]
  Module/Tool      Parameter                  Value
# HREC             FORCEOUT                    TRUE
# HNET             TRACE                          1
  HLABEL           TRACE                          8
  HPARM            TRACE                         65
  HSHELL           TRACE                          2
#                  FORCECXTEXP                 TRUE
#                  ALLOWXWRDEXP                TRUE
#                  ENORMALIZE                  TRUE
                   NUMCEPS                       12
                   CEPLIFTER                     22
                   NUMCHANS                      26
                   PREEMCOEF               0.970000
                   USEHAMMING                  TRUE
                   WINDOWSIZE         250000.000000
                   SAVEWITHCRC                 TRUE
#                  SAVECOMPRESSSED              TRUE
                   TARGETRATE         100000.000000
                   TARGETKIND           MFCC_E_D_A_Z
                   ZMEANSOURCE                FALSE
                   SOURCEFORMAT                 WAV
                   SOURCEKIND              WAVEFORM

        识别结果: 停止

停止
```

The car recognizes different voice commands and makes corresponding actions respectively. The specific actions are as follows.

voice	action
'forward'	move forward
'back'	go back
'left'	turn left
‘turn right’	turn right
'stop'	stop

(7) Precautions

If the battery light flashes, the robot may move abnormally due to insufficient battery, please charge it in time.

7.6 text recognition

（1）Function realization: The program recognizes the text in the picture, and controls the spider robot to perform different actions according to the content. The specific actions are as follows.

(2) Hardware requirements: P module, 8 T modules, 4 orthogonal modules, 4 bionic feet.

(3) Software requirements: Linux system, python3 environment, python-opencv-contrib, PIL, numpy and other environments, experimental code (including configuration path)

(4) Experimental hardware connection diagram:

Spider Numbering Diagram

(5) Execution example:

Step:

1. Use the "Ctrl+Alt+T" keys to open a terminal;

2. Go to:/Home/Samples-V2/Python/samples_AI/6.Words_recogniton_lite1. (The actual path needs to be modified according to the code storage location)

3. Turn on the switch of the P module and connect the WIFI of the P module with a computer, and run main.py on the terminal (text recognition combined with robot motion);

(6) Experimental effect

```
biowin@biowin:~/Samples_V2/Python/Sample_AI/6.Words_recogniton_lite$ python3 main.py
Connect To:10.10.100.254
连接成功
读取ID配置文件:/home/biowin/Samples_V2/Python/Sample_AI/6.Words_recogniton_lite/config.txt
#Spider

L1:30,31

L3:32,33

R1:34,35

R2:36,37

SetMotorMode 30 0
SetMotorMode 31 0
SetMotorMode 32 0
SetMotorMode 33 0
SetMotorMode 34 0
SetMotorMode 35 0
SetMotorMode 36 0
SetMotorMode 37 0
机器人初始化设置成功
【系统提示】 识别结果为  你好
【系统提示】 识别结果为  前进
【系统提示】 识别结果为  后退
【系统提示】 识别结果为  跳舞
【系统提示】 识别结果为  停止
terminate called without an active exception
已放弃 (核心已转储)
biowin@biowin:~/Samples_V2/Python/Sample_AI/6.Words_recogniton_lite$
```

Recognize the text in the picture, and recognize that different text spiders make corresponding actions. The specific actions are as follows.

Word	action
'Hello'	greet
'go ahead'	Move forward
'back'	Go back
'Dance'	Swing
'stop'	Stop and exit

(7) Precautions

If the action performed by the robot is inconsistent with the description, check whether the sequence of IDs in the ID configuration file is consistent with the IDs actually constructed. The configuration file path is:

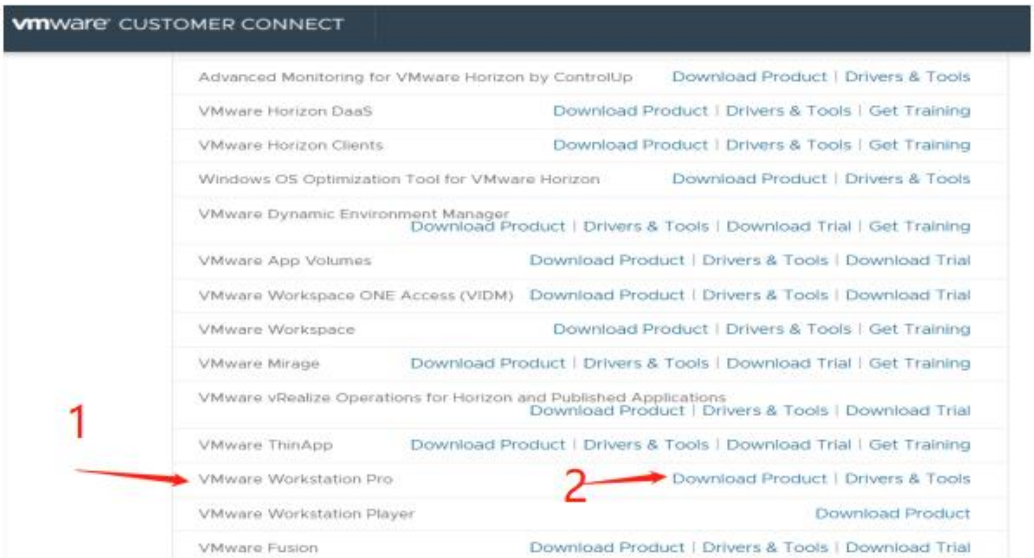
/Home/Samples_V2/Python/samples_AI/6.Words_recogniton_lite/config.txt

8 ROS environment construction and installation

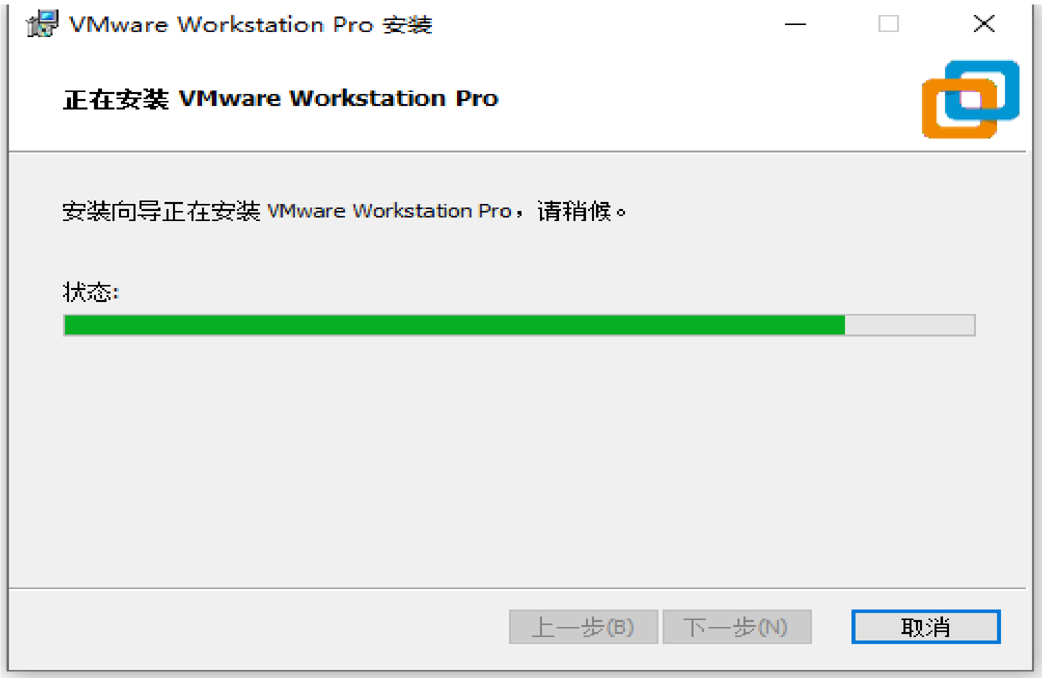
This article introduces how to install a virtual machine, how to install a Linux system in a virtual machine, and how to install ROS (melodic version). If you use the control box to control the robot, you can skip this chapter, because the control box has already configured the environment and does not need to build the environment.

8.1Virtual machine installation

Step 1: Enter the VMware official website (address: https://www.vmware.com/cn.html), click "Resources" in the upper navigation bar, and then click "Product Download" in the figure, as shown in the following figure.



Step 2: After the download is successful, you can install it, and click "Next" all the time.



8.2 Ubuntu installation

Step 1: After installing Vmware, you can create a new virtual machine. First download the ISO file: Download address: http://mirrors.aliyun.com/ubuntu-releases/18.04/.

Index of /ubuntu-releases/18.04/

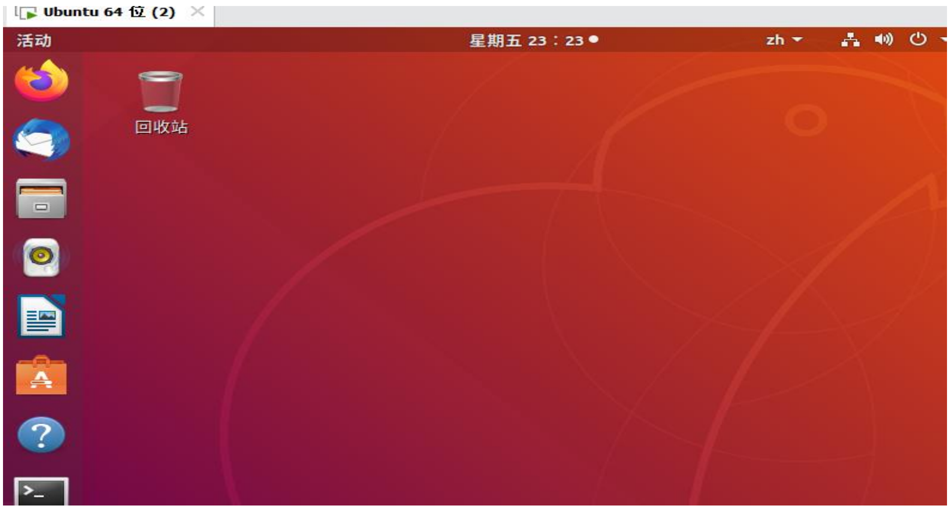
File Name	File Size	Date
Parent directory/	-	-
FOOTER.html	810.0 B	2021-09-17 05:46
HEADER.html	3.9 KB	2021-09-17 05:46
MD5SUMS-metalink	296.0 B	2020-02-12 21:42
MD5SUMS-metalink.gpg	916.0 B	2020-02-12 21:42
SHA256SUMS	202.0 B	2021-09-17 05:58
SHA256SUMS.gpg	833.0 B	2021-09-17 05:58
ubuntu-18.04.6-desktop-amd64.iso	41.0 B	2021-09-17 05:46
ubuntu-18.04.6-desktop-amd64.iso.torrent	187.7 KB	2021-09-17 05:46
ubuntu-18.04.6-desktop-amd64.iso.zsync	47.0 B	2021-09-17 05:46
ubuntu-18.04.6-desktop-amd64.list	42.0 B	2021-09-17 05:46
ubuntu-18.04.6-desktop-amd64.manifest	46.0 B	2021-09-17 05:46
ubuntu-18.04.6-live-server-amd64.iso	45.0 B	2021-09-17 05:45

Step 2: Open Vmware and click New Virtual Machine.

Step 3: Choose to install the Ubuntu mirror source, download the iso file with version 18.0 in the previous step, and select it.

Step 4: Configure the virtual machine size, user name and login password, and installation location.

Step 5: The screenshot of successful installation is as follows:



8.3 ROS system installation

Step 1: After creating the virtual machine, you can install the ROS system. Different Ubuntu installations have different ROS versions. The ROS version corresponding to Ubuntu 18.04 is ROS Melodic.
Configure Ubuntu Software and Updates to allow installation of uncertified software. First open the "Software and Updates" dialog, which can be searched in the Ubuntu search button.
After opening, configure it as shown below (make sure "restricted", "universe", and "multiverse" are checked).

Step 2: Set the installation source and copy the following commands to the terminal.

The installation source of Tsinghua University in China:

$ sudo sh -c '. /etc/lsb-release && echo "deb http://mirrors.ustc.edu.cn/ros/ubuntu/ $DISTRIB_CODENAME main" > /etc/apt/sources.list.d/ros-latest.list'

Step 3: Set the key and enter the command in the terminal:

$ sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654

Step 4: Install and update apt, and then you can install ROS. terminal command：

$ sudo apt-get update

#install ROS

$ sudo apt install ros-melodic-desktop-full

Note: At this time, the connection may time out due to network reasons, and the installation may fail. At this time, you can repeatedly call "update" and "installation command" until it succeeds.

Step 5: Configure environment variables to facilitate the use of ROS in any terminal. You can call the following two commands.

Terminal command:

Fist：$ echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc

Second：$ source ~/.bashrc

If you need to uninstall ROS, you can call the following command:

$ sudo apt remove ros-melodic-*

Step 6: Install the build dependencies, first install the build dependencies related tools:

$ sudo apt install python-rosdep python-rosinstall python-rosinstall-generator python-wstool build-essential

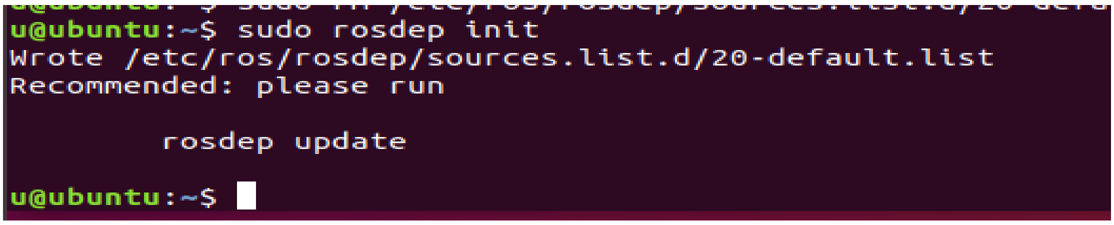
Then install rosdep (you can install system dependencies):

$ sudo apt install python-rosdep

初始化 rosdep：

$ sudo rosdep init

Successful screenshots:



After rosdep is successful, update it (if it fails, you can refer to this solution:https://blog.csdn.net/leida_wt/article/details/115120940）：

$ rosdep update

Step 7: Test ROS. ROS has some built-in small programs, you can run these small programs to check whether the ROS environment can run normally. Note: The cursor must be focused in the keyboard control window, otherwise the turtle movement cannot be controlled.

First open three terminals and launch the following three command lines:

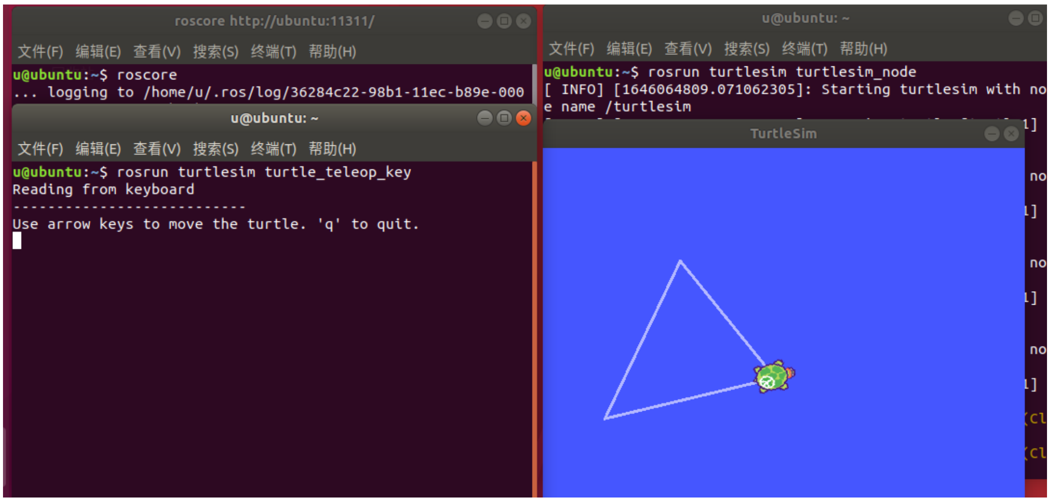
Command line 1 input: $ roscore

Command line 2 input: $ rosrun turtlesim turtlesim_node (a graphical interface will pop up at this time)

Command line 3 input: $ rosrun turtlesim turtle_teleop_key

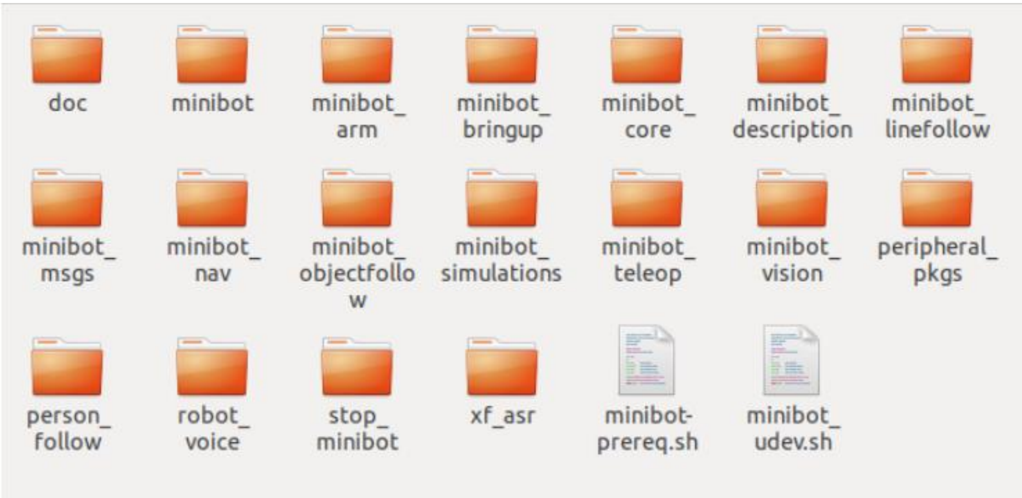
Note: "↑" is to control the turtle to move forward, "↓" is to control the turtle to move backward, "←" is to rotate counterclockwise, and "→" is to rotate clockwise.

The results are as follows:



8.4 MiniBot ROS environment configuration

Create a ROS workspace. Open a terminal in Ubuntu and type the following command to add dependencies for the workspace.
(1) Create a workspace
$ mkdir -p ~/MiniBot_ws/src && cd ~/MiniBot_ws/src
(2) Download the code to ~/MiniBot_ws/src
$ git clone https://gitee.com/biowinrobotics/MiniBot_ros.git
(3) Install dependent packages
$ sudo sh ~/MiniBot_ws/src/MiniBot_ros/MiniBot-prereq.sh
(4) Compile the MiniBot_ws workspace
$ cd ~/MiniBot_ws && catkin_make



(5) Configure the bashrc system file

$ gedit ~/.bashrc

Copy the following content to bashrc and add it directly at the end of the file:

workspace path

source /opt/ros/melodic/setup.bash

source ~/MiniBot_ws/devel/setup.bash

editor

export EDITOR='nano -w' # Use when ssh

export EDITOR='gedit' # When sublime and vscode are installed, replace with subl and code

#terminator computer & hostname color settings

PS1="\[\e[32;36m\]\u\[\e[37;33m\]@\h \[\e[36;32m\]\w\[\e[0m\]\\ $ "

#declare the default workspace of ros

ROS_WORKSPACE=~/MiniBot_ws/

#Declare your own computer name and ROS_MASTER_URI to ROS, multi-machine communication

#export ROS_HOSTNAME=hostname.local

#export ROS_MASTER_URI=http://hostname.local:11311

edit shortcut path

alias cw='cd ~/MiniBot_ws/'

alias cs='cd ~/MiniBot_ws/src/MiniBot_ros/'

alias cm='cd ~/MiniBot_ws/ && catkin_make'

(6) After the copy is completed, remember to source it in the terminal, or reopen a new terminal to run the program (each time a new terminal is opened, the system will reload the bashrc file)

$ source ~/.bashrc

9 Use of ROS

This chapter introduces the method of remotely controlling the robot under the ROS system and uses some cases to facilitate the user to use the MiniBot robot.

9.1 WiFi Connection Problem

If you want to control the mobile car robot with a PC, you need to connect the PC to the built-in WiFi module of the car computer by direct connection. The WiFi of the car computer generally starts with ROS_Mxx by default, but be careful not to connect to biowin_Mxx. If it is another P module or F module, it can be directly connected to the corresponding WiFi, such as biowin_Pxx. For example, in the picture below, use your own computer to directly connect to ROS_M3e09.

When connected to the WiFi of the car ROS, you can enter the PC to refresh the IP address. Enter the command in the terminal:

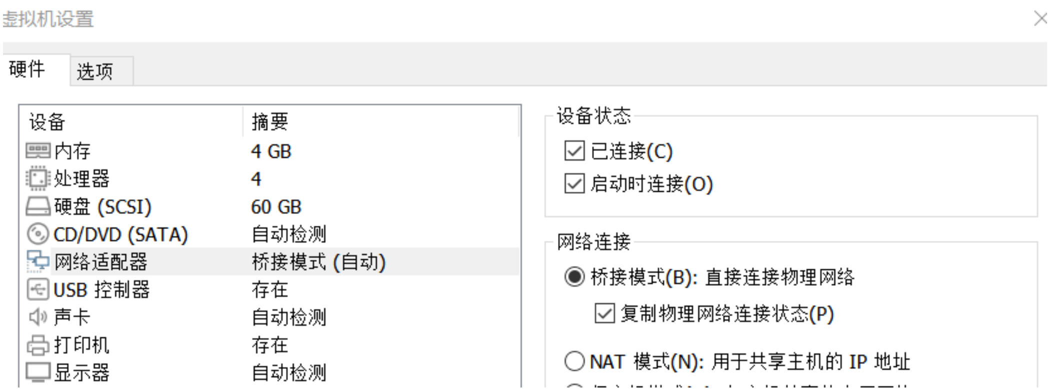
$ sudo service network-manager reatart && ifconfig

The IP of the PC can be refreshed and configured by using the above command multiple times, so that the car and the PC are located in the same subnet. The factory default IP address of the car is 192.168.1.100, so the IP address of the PC needs to be 192.168.1.xxx.

Note: When you need to use a PC to control the car, the PC must be connected to the car's WiFi. When you need to download files from the outside (or when you need to use an available network), the PC needs to disconnect the car's WiFi and connect to the external WiFi is only available. Under normal circumstances, the content of each section below is not specially explained, and the default is to connect the WiFi of the car ROS.

Supplementary note: Because the default address of the car is 192.168.1.100, but the IP address of the PC does not necessarily belong to the same subnet as the car system, so it may happen that the PC can ping the car, but the car may not be able to ping the PC. Happening. The workaround is as follows:

In the virtual machine, open Settings, select Network Adapters, select Bridged Mode.



Then open Virtual Network Editor from Edit, click Change Settings, select Bridged Mode, and select Automatic Settings.

Save and use the ifconfig command again to view the IP address of the virtual machine, refresh the IP, and test whether the car and the virtual machine are in the same subnet.

9.2 SSH Remote Connection Robot

（1） Use the ifconfig command to view the addresses of the robot and the PC. The IP address of the car is 192.168.1.100 by default, and the address of the PC is 192.168.1.110. Note: The car and the PC must be in the same subnet.

$ ifconfig

```
biowin@biowin: ~
biowin@biowin:~$ ifconfig
eno1: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 1c:69:7a:08:4e:53  txqueuelen 1000  (以太网)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 16  memory 0xc0b00000-c0b20000

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (本地环回)
        RX packets 286  bytes 21220 (21.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 286  bytes 21220 (21.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp0s20f3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.19  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::a0bb:3e0a:5af6:d9d5  prefixlen 64  scopeid 0x20<link>
        ether 04:ea:56:49:ac:73  txqueuelen 1000  (以太网)
        RX packets 93  bytes 17331 (17.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 150  bytes 22604 (22.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

(2) Execute the $ sudo gedit /etc/hosts/ command, and add the other party's IP address and corresponding computer name to the file.

$ sudo gedit /etc/hosts/

That is to add "192.168.1.100 minibot" to the PC.

```
打开(O)
1 127.0.0.1      localhost
2 127.0.1.1      biowin
3 192.168.1.100     ubuntu
```

Don't forget to add "192.168.1.19 biowin" (the name of the PC) to the car.

After the setup is complete, use the ping command on the two machines to test whether the network is connected.

```
biowin@biowin:~$ ping 192.168.1.100
PING 192.168.1.100 (192.168.1.100) 56(84) bytes of data.
64 bytes from 192.168.1.100: icmp_seq=1 ttl=64 time=5.15 ms
64 bytes from 192.168.1.100: icmp_seq=2 ttl=64 time=3.79 ms
64 bytes from 192.168.1.100: icmp_seq=3 ttl=64 time=7.28 ms
64 bytes from 192.168.1.100: icmp_seq=4 ttl=64 time=17.3 ms
64 bytes from 192.168.1.100: icmp_seq=5 ttl=64 time=11.3 ms
```

```
ubuntu@ubuntu:~$ ping 192.168.1.19
PING 192.168.1.19 (192.168.1.19) 56(84) bytes of data.
64 bytes from 192.168.1.19: icmp_seq=1 ttl=64 time=6.61 ms
64 bytes from 192.168.1.19: icmp_seq=2 ttl=64 time=76.5 ms
64 bytes from 192.168.1.19: icmp_seq=3 ttl=64 time=98.6 ms
64 bytes from 192.168.1.19: icmp_seq=4 ttl=64 time=120 ms
```

(4) Master-slave configuration. Open and modify the ~/.bashrc file. Both the master and slave machines need to modify the ~/.bashrc file and add the following:

$ gedit ~/.bashrc

Add to the host:

export ROS_HOSTNAME= Host IP

export ROS_MASTER_URI=http://192.168.1.100:11311

Add to the slave:

export ROS_HOSTNAME= Slave IP

export ROS_MASTER_URI=http://192.168.1.100:11311

(5) Install openssh-server

#Start openssh-server

$ sudo apt install openssh-server

$ sudo service ssh start

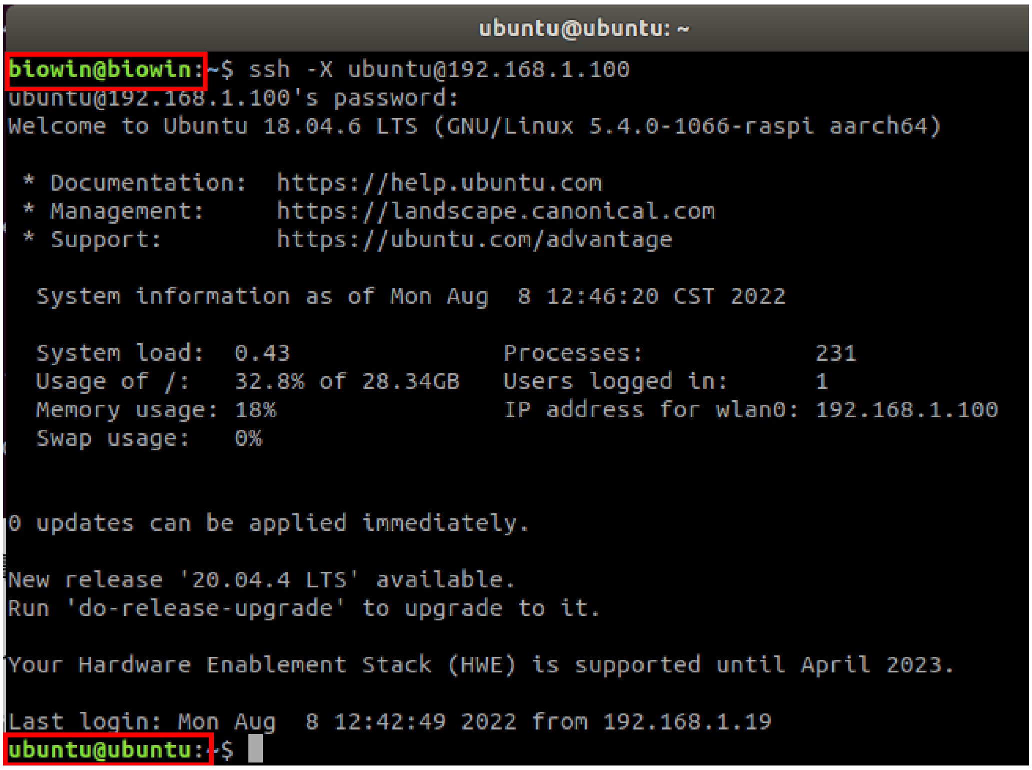
#View ssh running status

$ sudo service ssh status

(6) Log in to the robot remotely. Robot account: ubuntu. Password: biowinbot. Where before the @ sign is the username of the host, and after the @ sign is the IP address of the host (192.168.1.100). According to the factory settings of the robot, the default username is ubuntu.

Execute the command in the terminal of the PC:

$ ssh -X ubuntu@192.168.1.100

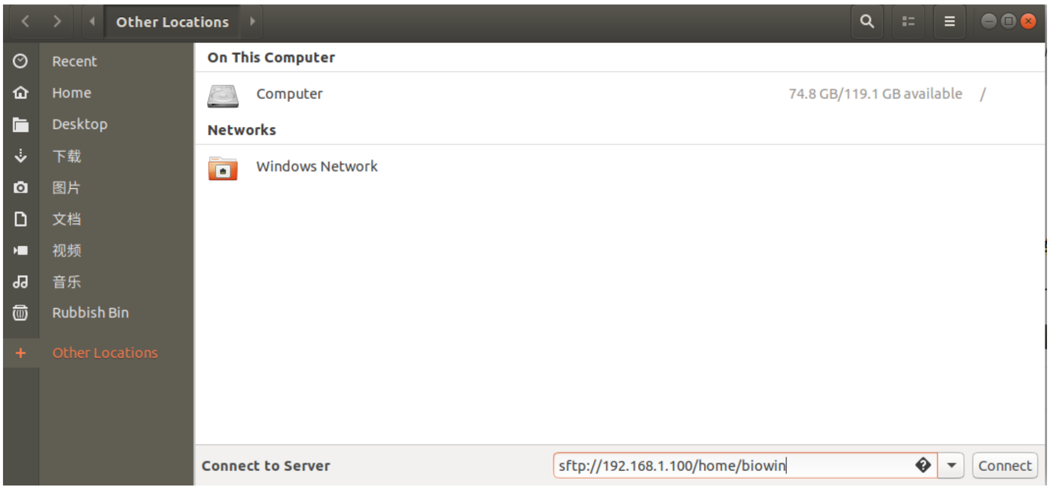


After the ssh remote login is successful, you can see the information shown in the above figure. Especially the most important thing is to look at the green font part of the bottom line (representing "user@machine name"), which has changed from the user name and machine name of the PC computer (biowin@biowin) to the user name and machine name of the host computer (ubuntu@ ubuntu), which is the most important verification method for successful ssh remote login.

The above method is the simplest ssh remote login. Every time you open a new terminal, you need to enter the command and password. To simplify the operation, you can also configure the ssh public key and private key to set up password-free login. Find out for yourself.

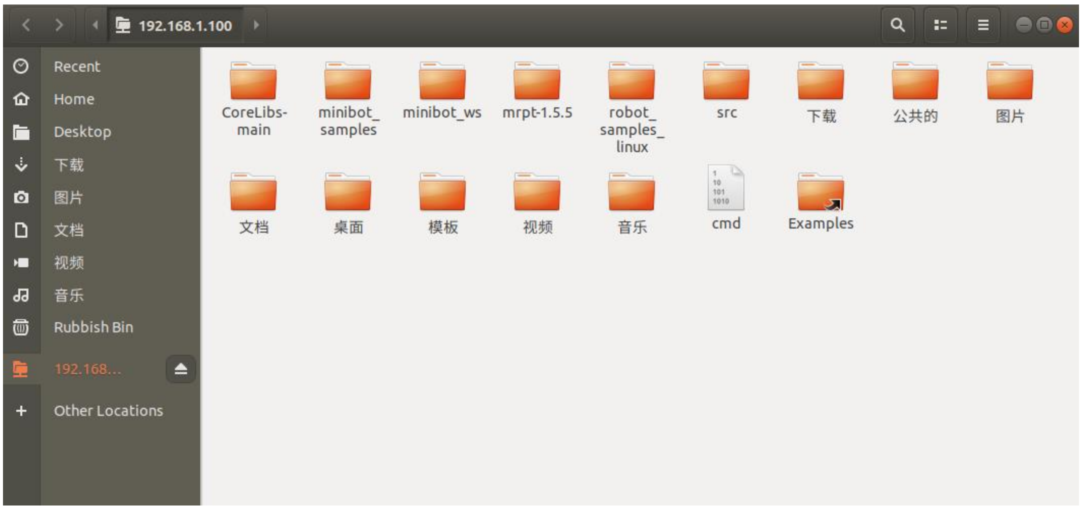
9.3 Remote Access to Robot Files

Open the file directory in the PC, select "Other Location", and enter the IP address of the car to connect, so that the PC can remotely control the files in the car robot.



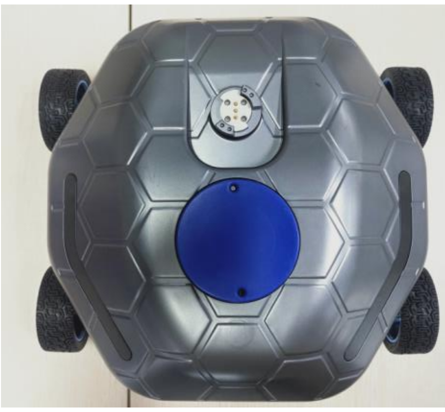
Click Connect and enter your account and password. The account is ubuntu and the password is biowinbot.

You can see that there will be one more directory in the file directory at this time, and this directory stores the files in the car.



The content of this section will enable users to quickly master the use of Minibot robot and enhance their understanding of Minibot robot through some experimental cases.

9.4 Mobile Car Robot with Four Rubber Wheels



In Section 8.5, the ssh connection has been created to realize the multi-machine communication of ROS. Now start using the SSH service to remotely control the car from your PC. Follow the steps in Section 6 to build the car robot, log in to the car system at the terminal, and then use the ROS system to control and operate the robot. Note: The premise of the experiments in this section is that the PC has been successfully connected to the mobile car platform remotely via SSH. That is, first enter the $ ssh-X ubuntu@192.168.1.100 command in the terminal of the PC, enter the system password biowinbot of the car, and log in to the car system. The terminal command with ssh in the following example refers to the terminal command running on the car, generally under ubuntu@ubuntu, and other terminal commands run on the PC.

9.4.1 The Keyboard Controls Robot Movement

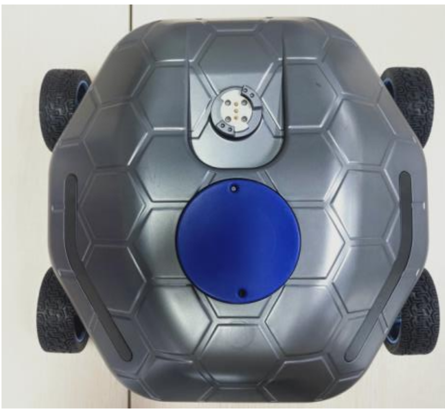
(1) Realization function: Based on the ROS system, the movement of the car is controlled through the keyboard.

(2) Hardware Preparation

1 M module, 4 rubber wheels, 4 I modules and 4 I module wrapping pieces.

（3）Construction of the Configuration

Put the four I modules into the I module wrapping piece, and then attach a rubber wheel to each I module, and then attach it to the M module after completion. As shown in the figure:



（4）Example Execution

Open the PC or control box computer, and run the following command to start the node of the four-wheeled car.

（ssh）$ roslaunch minibot_driver car_driver.launch

Start keyboard control node.

$ roslaunch minibot_teleop keyboard_control.launch

Long press the following buttons to control the car:

Key	Command
i	Move forward

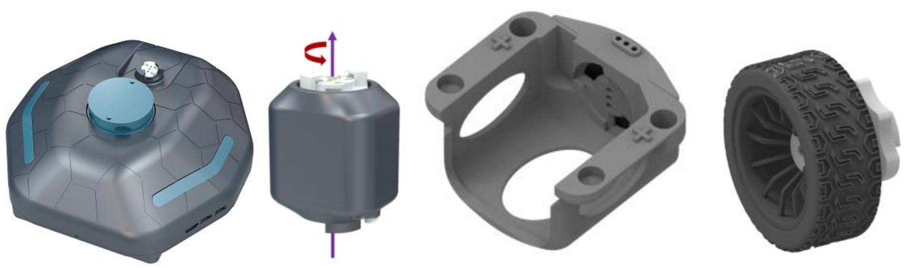
,	Move backward
j	Turn left
l	Turn right
u	Go left
o	Go right
m	Back left
.	Back right
k	Stop
q	Accelerate
z	Slow down

9.4.2 The Car Follows a Track of a Particular Shape

(1) Realization function: Based on the ROS system, the car is controlled to run according to the trajectory of a specific shape.

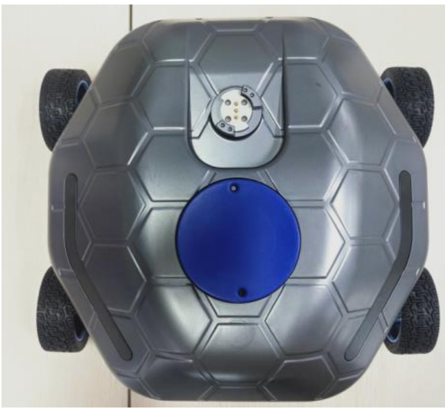
(2) Hardware Preparation

1 M module, 4 rubber wheels, 4 I modules and 4 I module wrapping pieces.



(3) Construction of the Configuration

Put the four I modules into the I module wrapping piece, and then attach a rubber wheel to each I module, and then attach it to the M module after completion. As shown in the figure:



（4）Example Execution

Open the PC or control box computer and run the following code, first start the car ROS driver.

（ssh）$　roslaunch minibot_driver car_driver.launch

Start the LiDAR odometer.

（ssh）$　roslaunch minibot_bringup laser_odom.launch

Run the following command to control the car to follow a circular trajectory.

（ssh）$　rosrun minibot_application move_circle.py

Run the following command to control the car to take a square trajectory.

（ssh）$　rosrun minibot_application nav_square.py

Run the following commands to control the car to move forward and backward (based on odometer).

（ssh）$　rosrun minibot_application odom_out_and_back.py

9.4.3 Startup of the Robot Camera

（1）Realization function: Use the ROS visualization tool RVIZ to view the screen of the USB camera.

（2）Hardware Preparation

1 M module, 4 rubber wheels, 5 I modules, 4 I module wrapping pieces, 2 T modules, 1 USB camera and 1 camera fixing piece.







（3）Construction of the Configuration

Put the four I modules into the I module wrapping piece, and then attach a rubber wheel to each I module, and then attach it to the M module after completion. Connect the (configured as I T T) PTZ to the car, fix the USB camera to the camera fixing piece of the T module, connect the fixing piece to the end of the PTZ, and finally connect the USB cable of the camera to the USB interface on the side of the car. As shown in the figure:

（4）Example Execution

Start the USB camera.

（ssh）$ rosrun minibot_bringup start_camera.sh

If "Cannot open '/dev/video0': 13, Permission denied" appears.

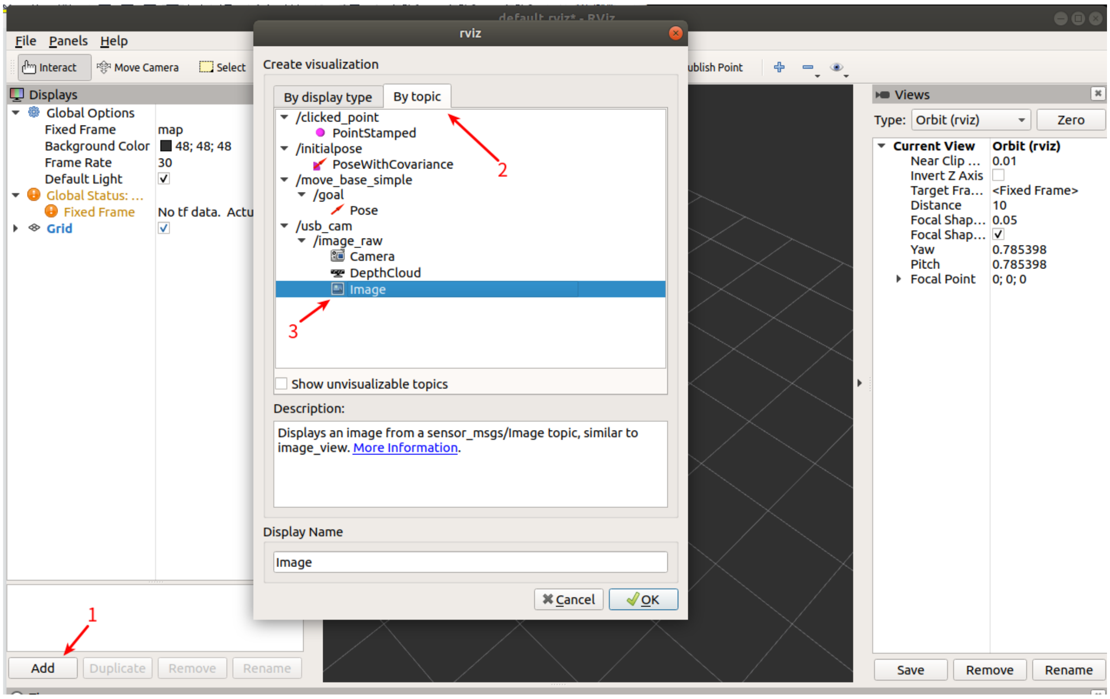
Workaround: Add permissions, use the following command:

（ssh）$ sudo chmod 666 /dev/video0

Launch RVIZ

$ rviz

Click "add" in the opened rviz, and then select "Image" from "By topic" in the opened interface.



9.4.4 Realize the Function of Car Line Patrol

(1) Realization function: The vision library based on OpenCV controls the car for color tracking.

(2) Hardware Preparation

1 M module, 4 rubber wheels, 5 I modules, 4 I module wrapping pieces, 2 T modules, 1 USB camera, 1 camera fixing piece, and 1 roll of yellow line patrol tape.








(3) Construction of the Configuration

Put the four I modules into the I module wrapping piece, and then attach a rubber wheel to each I module, and then attach it to the M module after completion. Connect the (configured as I T T) PTZ to the car, fix the USB camera to the camera fixing piece of the T module, connect the fixing piece to the end of the PTZ, and finally connect the USB cable of the camera to the USB interface on the side of the car. As shown in the figure:

(4) Environment construction

In this example the trace color is yellow, a specific line is pasted with yellow tape on the ground, and the car is placed in front of the line.

Note: Try not to place yellow objects near the car to avoid affecting the operation of the car.

As shown below:

(5) Example Execution

In the PTZ of this experiment, the modules used are I module 21, t module 36 and t module 37. The launch can be modified according to the actual built module ID.

(ssh) $ roscd minibot_bringup/launch

(ssh) $ gedit minibot_with_arm.launch

The PTZ modifies the correct ID of the module from bottom to top, as shown in the figure:

```
<launch>
<!-- wheelMode: 1-two-wheel mode, 2-two-wheel mode 3-three-wheel mode
4-four wheels(four-wheel mode)  5-four wheels(mecanum wheel mode) 6-
four wheels(normal wheel mode) -->
<!-- connect_mode: "udp" or "serial" -->
<!-- armIdL: Fill in according to the number of operating arms, if it
does not match the actual number of construction, an error will be
reported -->
<arg name="wheelMode"  default="6"/>
<arg name="connectMode" default="serial"/>
<arg name="armIdL" default="21,36,37"/>
<param name="connectMode" value="$(arg connectMode)"/>
<param name="wheelMode" value="$(arg wheelMode)"/>
<param name="armIdL" value="$(arg armIdL)"/>
```

(ssh) $ roslaunch minibot_bringup minibot_with_arm.launch

Start the USB camera.

（ssh）$ rosrun minibot_bringup start_camera.sh

Move to line patrol attitude.

（ssh）$ rosrun minibot_driver move_to_linefollow_pose.py

Start the visual line patrol program, [0.Black][1.Red][2.Orange][3.Yellow][4.Green][5.Cyan][6.Blue][7.Purple]. If not filled, the default is 3 yellow.

（ssh）$ roslaunch minibot_application linefollow.launch

Start "rqt_image_view" on the PC and view the topic "line_follow/image".

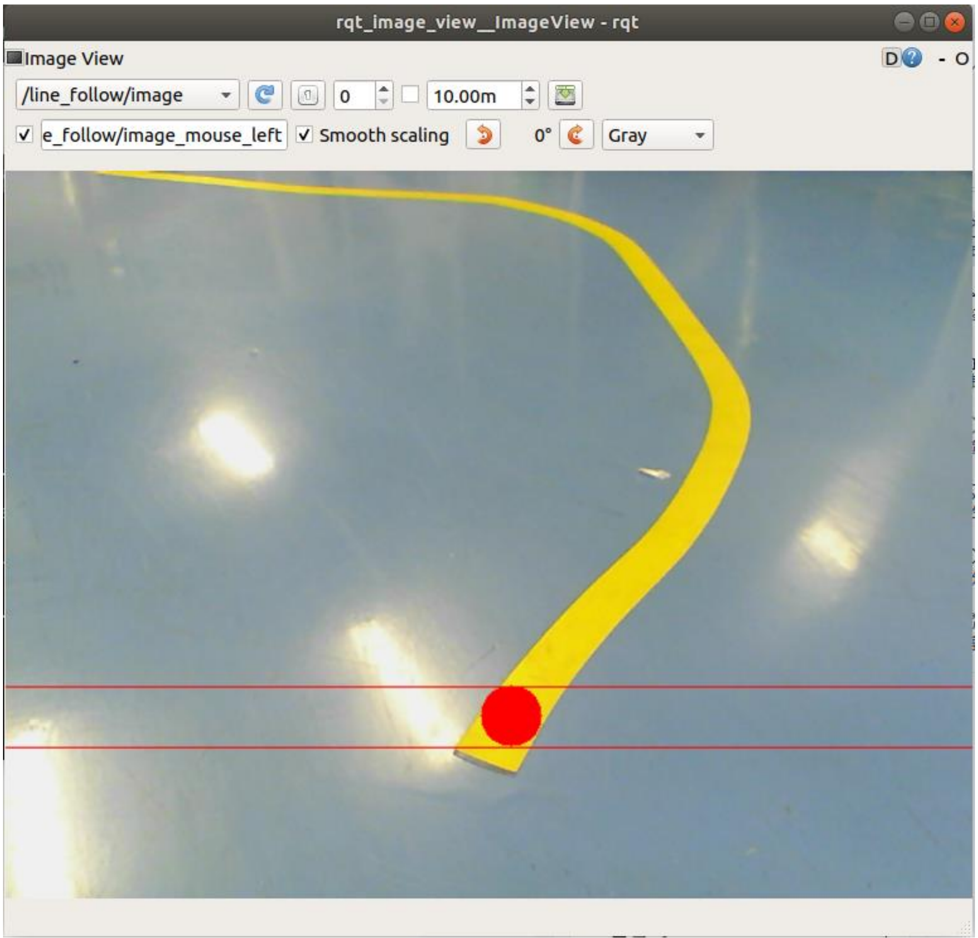
$ rqt_image_view

Start line patrol.

$ rosservice call /line_follow/trigger

（6）Operation effect

Start the line patrol through "/line_follow/trigger", then the car will start to move.

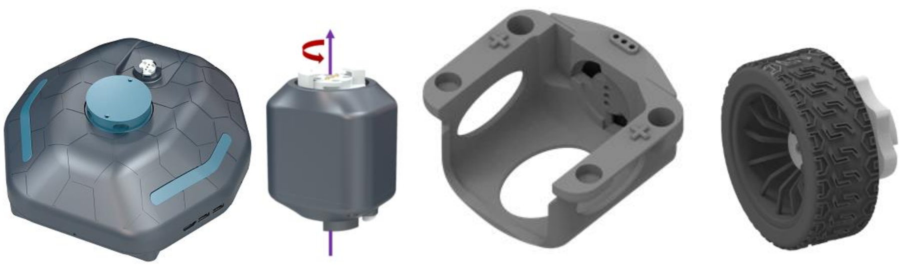


9.4.5 Face Tracking

(1) Realization function: The vision library based on OpenCV recognizes the face, and the car will track according to the position change of the face. When in use, the distance between the car and the person should be kept at 1 meter.

(2) Hardware Preparation

1 M module, 4 rubber wheels, 5 I modules, 4 I module wrapping pieces, 2 T modules, 1 USB camera and 1 camera fixing piece.





(3) Construction of the Configuration

Put the four I modules into the I module wrapping piece, and then attach a rubber wheel to each I module, and then attach it to the M module after completion. Connect the (configured as I T T) PTZ to the car, fix the USB camera to the camera fixing piece of the T module, connect the fixing piece to the end of the PTZ, and finally connect the USB cable of the camera to the USB interface on the side of the car. As shown in the figure:

(4) Example Execution

Run the robot + PTZ ROS driver

(ssh) $ roslaunch minibot_bringup minibot_with_arm.launch

Start the USB camera.

(ssh) $ rosrun minibot_bringup start_camera.sh

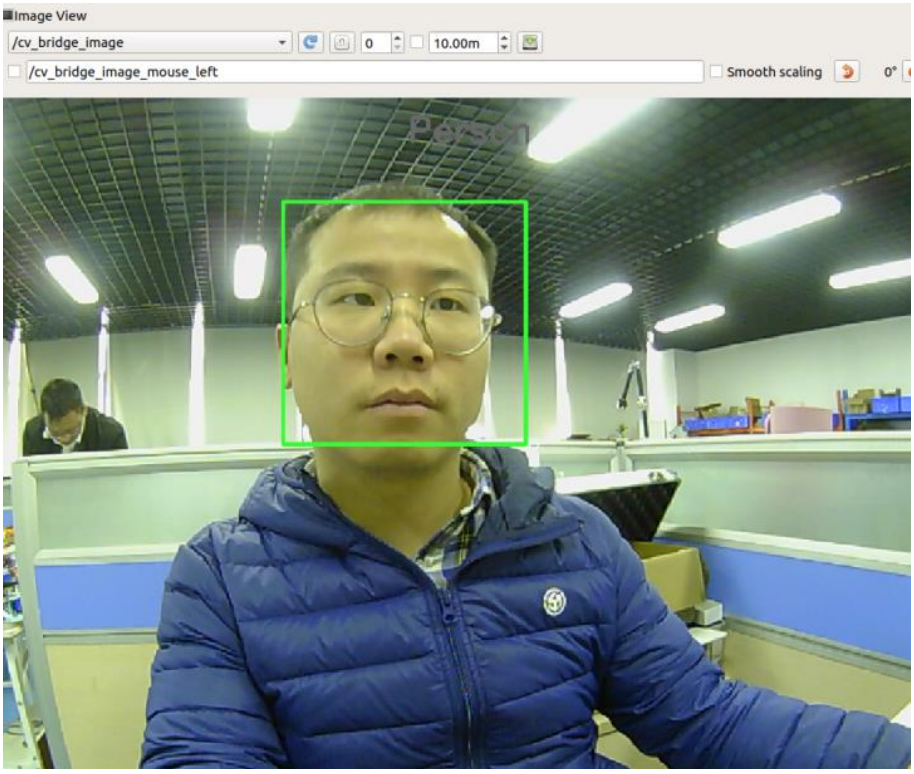
Start the face recognition and tracking program.

（ssh）$ roslaunch minibot_application face_detector.launch

Start "rqt_image_view" on the PC.

$ rqt_image_view

（5）Operation effect

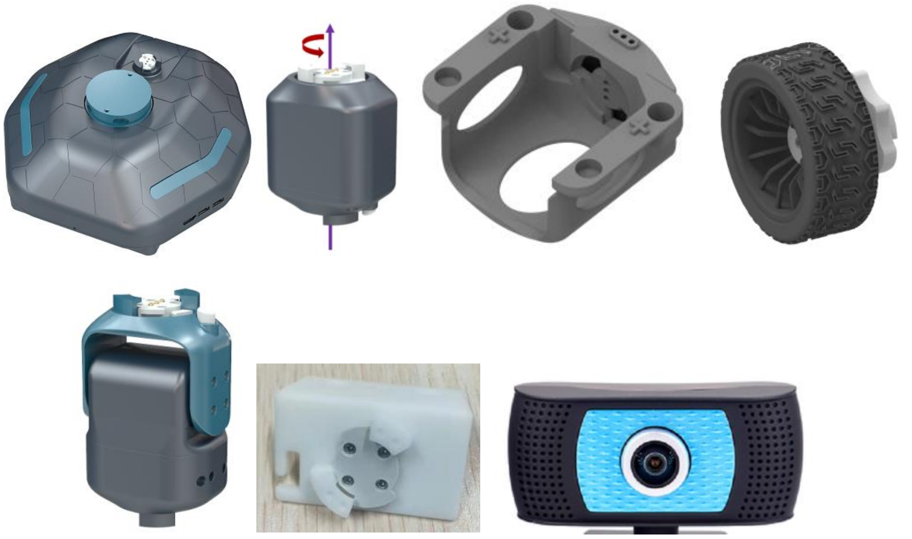


9.4.6 Object Following Based on QR Code

（1）Realization function: Place the specified QR code in front of the camera, and after the car obtains the image, the OpenCV-based vision library calculates the position change of the QR code for motion tracking.

（2）Hardware Preparation

1 M module, 4 rubber wheels, 5 I modules, 4 I module wrapping pieces, 2 T modules, 1 USB camera and 1 camera fixing piece.



(3) Construction of the Configuration

Put the four I modules into the I module wrapping piece, and then attach a rubber wheel to each I module, and then attach it to the M module after completion. Connect the (configured as I T T) PTZ to the car, fix the USB camera to the camera fixing piece of the T module, connect the fixing piece to the end of the PTZ, and finally connect the USB cable of the camera to the USB interface on the side of the car. As shown in the figure:

(4) Example Execution

Run the robot + PTZ ROS driver

(ssh)$ roslaunch minibot_bringup minibot_with_arm.launch

Start the USB camera.

(ssh)$ rosrun minibot_bringup start_camera.sh

Start the QR code tracking program.

(ssh)$ roslaunch minibot_application ar_follow.launch

Launch RVIZ on PC.

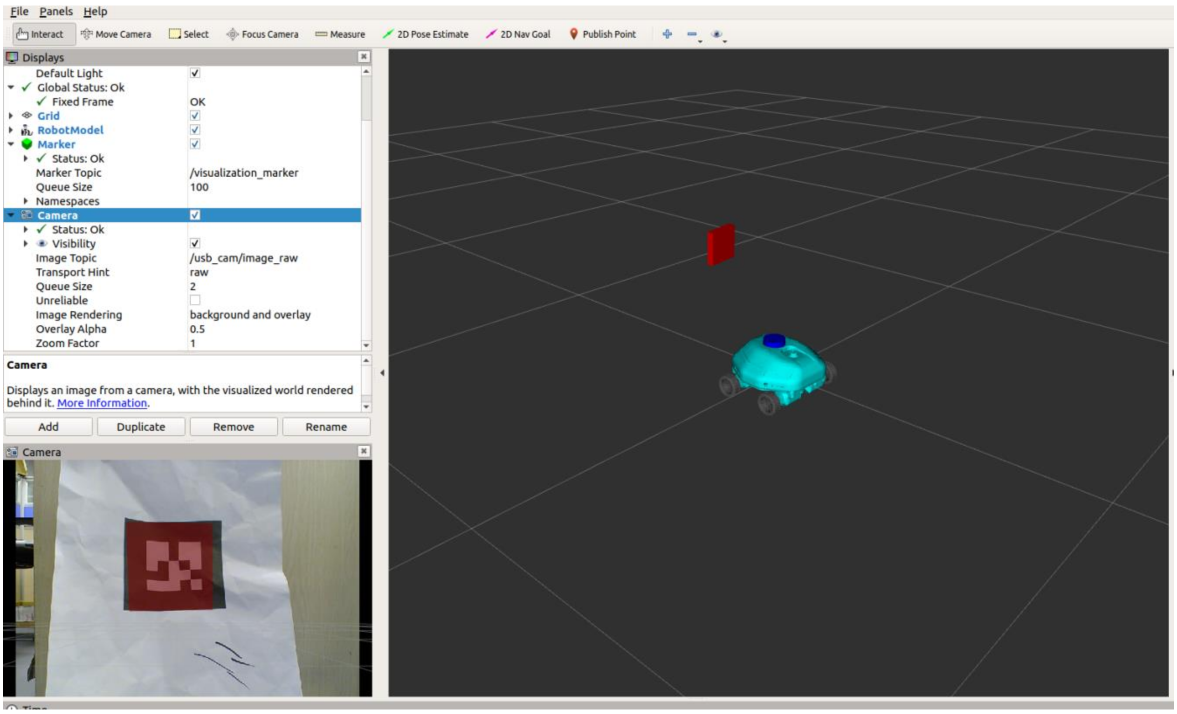
$ rosrun rviz rviz -d `rospack find minibot_application`/rviz/ar_follow.rviz

If you get "ImportError: No module named ar_track_alvar_msgs.msg" error.

Workaround:

(ssh)$ sudo apt-get install ros-melodic-ar-track-alvar ros-melodic-ar-track-alvar-msgs

(5) Operation effect



9.4.7 LiDAR Mapping

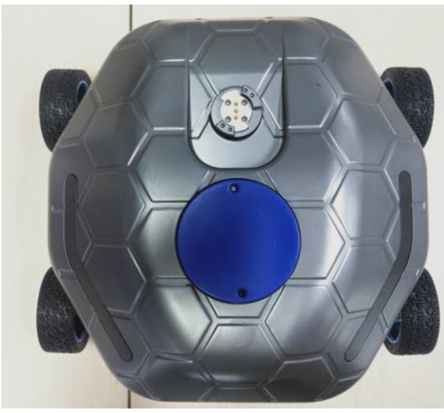
(1) Realization function: The obstacles near the car are scanned by LiDAR, and a map is created and saved according to the real environment based on SLAM technology.

(2) Hardware Preparation

1 M module, 4 rubber wheels, 4 I modules and 4 I module wrapping pieces.

(3) Construction of the Configuration

Put the four I modules into the I module wrapping piece, and then attach a rubber wheel to each I module, and then attach it to the M module after completion. As shown in the figure:



(4) Example Execution

Enter the following commands in the terminal to start the car drive node and LiDAR node.

(ssh)$ roslaunch minibot_bringup minibot_with_ekf_odom.launch

Enter the command in the terminal, gmapping the graph node.

(ssh)$ roslaunch minibot_nav mb_gmapping.launch

View the mapping results in rviz on the PC side.

$ rosrun rviz rviz -d `rospack find minibot_nav`/rviz/gmapping.rviz

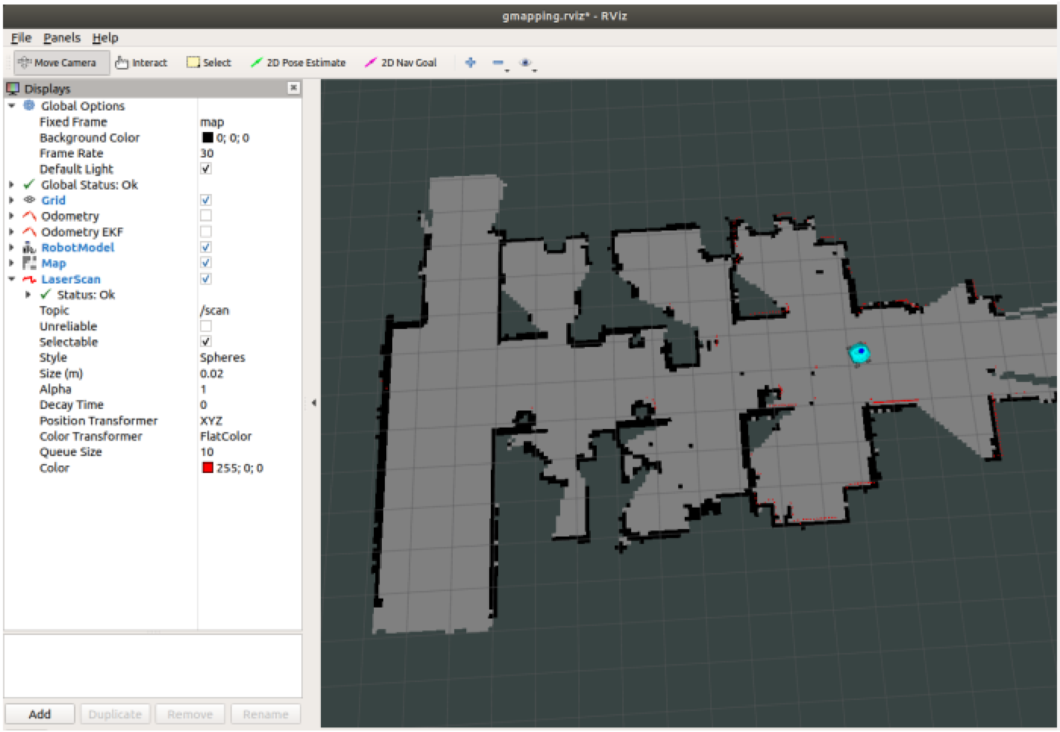
Start the keyboard control node, use the keyboard to control the movement of the car, and gradually improve the map during the movement. In the process of building a map, if we feel that the map we built is not satisfactory, we can choose to build a map multiple times until we are satisfied. Note: Move slowly or the map may drift.

$ roslaunch minibot_teleop keyboard_control.launch

After the map is built, you can save the created map with the command below. If you want to view the map, you can find the map we just created in the /minibot_nav/maps folder named: curMap.pgm and curMap.yaml.

(ssh)$ roslaunch minibot_nav mb_mapSaving.launch

(5) Operation effect



9.4.8 LiDAR Autonomous Navigation

(1) Realization function: The obstacles near the car are scanned by LiDAR,

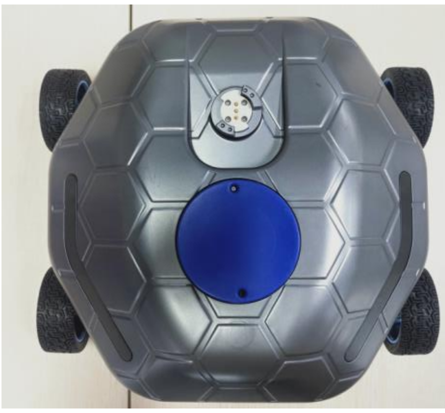
and autonomous navigation is carried out according to the real environment based on SLAM technology.

（2）Hardware Preparation

1 M module, 4 rubber wheels, 4 I modules and 4 I module wrapping pieces.

（3）Construction of the Configuration

Put the four I modules into the I module wrapping piece, and then attach a rubber wheel to each I module, and then attach it to the M module after completion. As shown in the figure:



（4）Example Execution

Enter the following commands in the terminal to start the car drive node and LiDAR node.

(ssh)$ roslaunch minibot_bringup minibot_with_ekf_odom.launch

Start the navigation node.

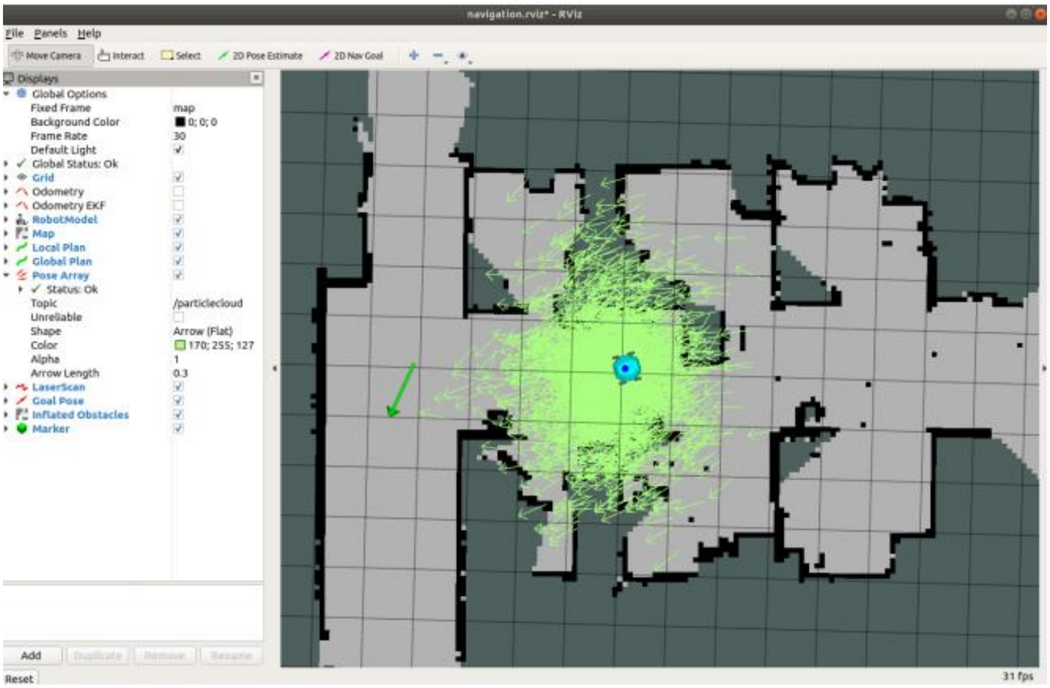
(ssh)$ roslaunch minibot_nav mb_navigation.launch

Open rviz on the PC side to view the results.

$ rosrun rviz rviz -d `rospack find minibot_nav`/rviz/navigation.rviz

Use the 2D Pose Estimate button on rviz to determine the matching position of the robot in Rviz according to the real environment where the robot is located, then click the 2D NAV Goal button, and click the target point and direction of the robot on the map. After clicking, it will generate an arrow, and the cart moves towards the target point.

(5) Operation effect



9.4.9 Walking Different Tracks in the Gazebo Simulation Environment

(1) Realization function: The control car model runs in the Gazebo simulation environment. This example runs on the PC side, and the PC side needs to be set as the ROS host.

(2) Example Execution

$ gedit ~/.bashrc

Find

```
#Check your IP address with ifconfig
export ROS_HOSTNAME=192.168.1.19
export ROS_MASTER_URI=http://192.168.1.100:11311
```

and modify it to ROS_MASTER_URI as your own IP, assuming that the current PC

IP is 192.168.1.151, as shown below:

```
#Check your IP address with ifconfig
export ROS_HOSTNAME=192.168.1.151
export ROS_MASTER_URI=http://192.168.1.151:11311
```

Run the following command to start the keyboard remote control node to control the simulation car to run.

$ roslaunch minibot_gazebo minibot_gazebo.launch

$ roslaunch minibot_sim_demo keyboard_control.launch

Run the following command to control the simulation car to walk the circular trajectory.

$ roslaunch minibot_gazebo minibot_gazebo.launch

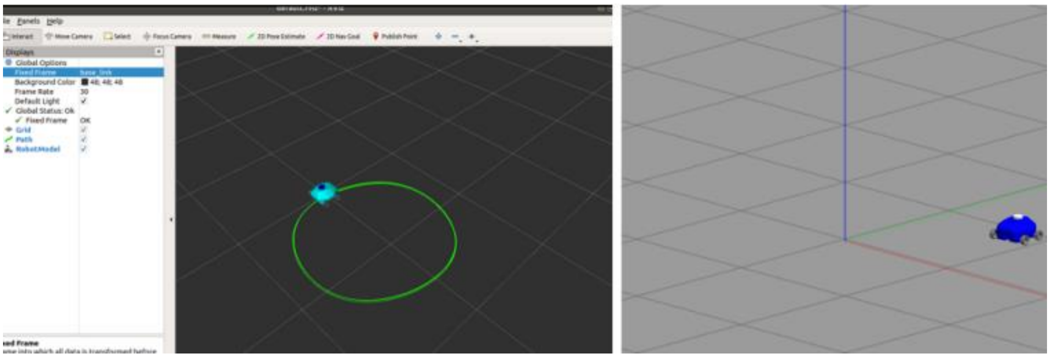
$ rosrun minibot_sim_demo move_circle.py

Run the following command to control the simulation car to walk a square trajectory.

$ roslaunch minibot_gazebo minibot_gazebo.launch

$ rosrun minibot_sim_demo nav_square.py

(5) Operation effect



10 Exception Handling

10.1 Mechanical Part

①When using BW-RobotStudio to control the robot, the modules need to be built according to the software prompts in the construction stage, and the number of modules to be built is one at a time.

②When disassembling the module, if pressing the button still cannot rotate the module to complete the disassembly, press the button several times or directly press the button at the interface to rotate and disassemble the module. In the same way, if it is found that the modules cannot be directly locked during assembly, press the button several times to make the button rebound to realize the locking between the modules.

③When building the module, you should assemble the module according to the prompts on the 3D simulation interface, pay attention to the directions of the "red" and "blue" ports, and do not install them in reverse.

10.2 Electrical Part

①When the robot runs abnormally or does not move, observe whether the battery light of the robot flashes or there is only one bar of power left (three bars when fully charged). If so, the robot needs to be charged in time before use.

②After eliminating the power problem, if the module still does not move normally, observe whether the red indicator light of each module is on. If it does not light up, observe whether the interface is loose and the connection between the two modules is unsuccessful.