Table of Contents

Introduction

Preparation

Development

4 Others

myCobot: From 0 to 1

myCobot
小象机械臂
LEARNING
从入门到精通
人人都可以学习玩耍的入门级协作机器人
世界最小最轻的六轴协作机器人 Worlds' Smallest Collaborative Robotic Arm
大象机器人编著

1.1 Why do we design myCobot

> **An entry-level collaborative robot arm** that everyone can learn and play

The original design of myCobot is to help friends who are interested in 6-axisseries robot to learn it from entry to master, creating unprecedented experience and teaching value.

What you can learn

> Robotics is based on rigid body kinematics and dynamics, but also an interdisciplinary subject that combines hardware, software, algorithm and control.

With myCobot, you can learn that

- **Hardware**
  - **Embedded Microcontroller Based on ESP32**
  - **Motor and Steering Gear**
  - **M5Stack Basic/ Atom**
- **Software**
  - **Arduino**开发环境
  - **C++**
  - **Python**

    - **ROS，MoveIt**
    - **Communication Data**
    - **Virtual Machines & Linux (visual system)**
- **Algorithm**
    - **Series Manipulator**
    - **Coordinate and Coordinate Transformation**坐标与坐标转换
    - **DH Parameters**
    - **Kinematics**
    - **Manipulator Algorithm (e.g. dynamics)**
- **Machine Vision (Vision Set)**
    - **Color Recognition**
    - **Image Recognition**
    - **Hand-Eye Calibration**
    - **See and Grab**
- **Extended Applications**
    - **End-effector: gripper,suction pump, etc.**
    - **Robot Suit & Industry 4.0 Applications**

Parts of Gitbook

> View the directory on the left to jump

There are four major parts of Gitbook :

- **Introduction & Quick Start**
    - **Introduction** -- introduce what myCobot is and its main features, etc.
    - **How to Read** -- help you read Gitbook efficiently according to your learning level and knowledge background
    - **Use Cases** -- you can know exactly what use cases you can accomplish with
    - **Quick Start** -- learn the unboxing of your myCobot, and its first boot and use
- **Preparation before Development**

    - **Background Knowledge** -- learn about tools, industrial robots, algorithms, software, hardware,etc.
    - **Hardware Learning** -- learn about embedded hardware, structural components, electronic components, etc.
    - **Purpose of Use** -- identify the purpose you want to use it for, and complete the study related to your task
- **Development and Use**
    - **Development Environment** -- learn to use Arduino, ROS, uiFlow, roboFlow, python and others development environment to develop myCobot
    - **Accessories** -- learn to use myCobot with different accessories, such us bases, grippers, suction pumps and so on
    - **Machine Vision** -- learn to control myCobot under the guidance of machine vision
    - **Robot Modification** -- learn how to modificate myCobot into a 4 or 5 axis manipulator
- **myCobot Suit**
    - **Intelligent Warehouse:** learn how to use myCobot to carry different objects
    - **Artificial Intelligence:** learn how to control myCobot to grasp objects intelligently under the guidance of machine vision
    - **Industry 4.0:** learn how to grasp and place objects intelligently by simulating production line

---

Information Source

- **Official website:** www.elephantrobotics.com
- **Tutorial video:** https://www.youtube.com/channel/UC68l2RaRF2Mp8fzpCTzNBfA
- **Shop website:** https://shop.elephantrobotics.com

Contact Us

> **If you have any other questions, you can contact us as follows.**
>
> **We will answer you as soon as possible (working day 9:30-18:30)**

- Twitter: myCobot Official\@CobotMy
- Facebook: https://www.facebook.com/MyCobot-116558893805177
- Mail: support@elephantrobotics.com

myCobot Introduction

myCobot
世界最小最轻的六轴协作机器人
Worlds' Smallest Collaborative Robotic Arm
myCobot

1 Design Background

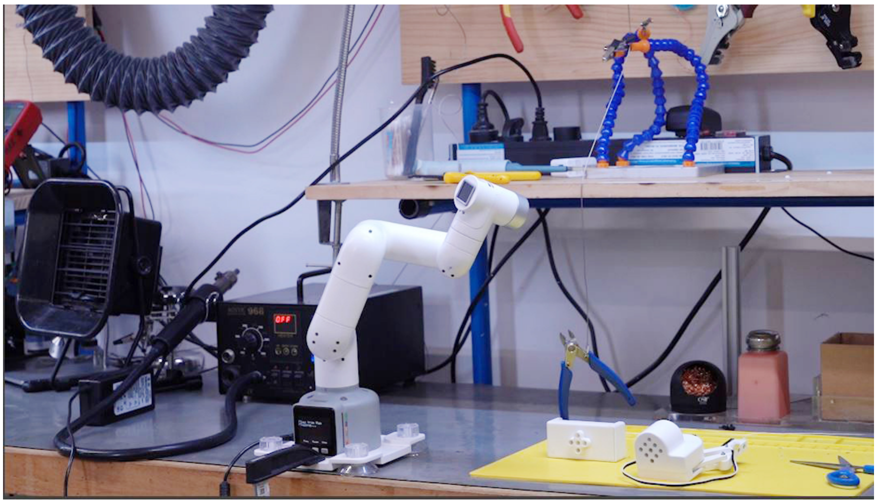
Upholding the mission of **"Enjoy Robots World"**, Elephant Robotics designed and developed myCobot, the world's smallest and lightest collaborative robot, retaining most functions of industrial robots. With compact and elegant industrial design, excellent and powerful performance, and huge software and hardware development space, myCobot has unlimited possibilities in applications.

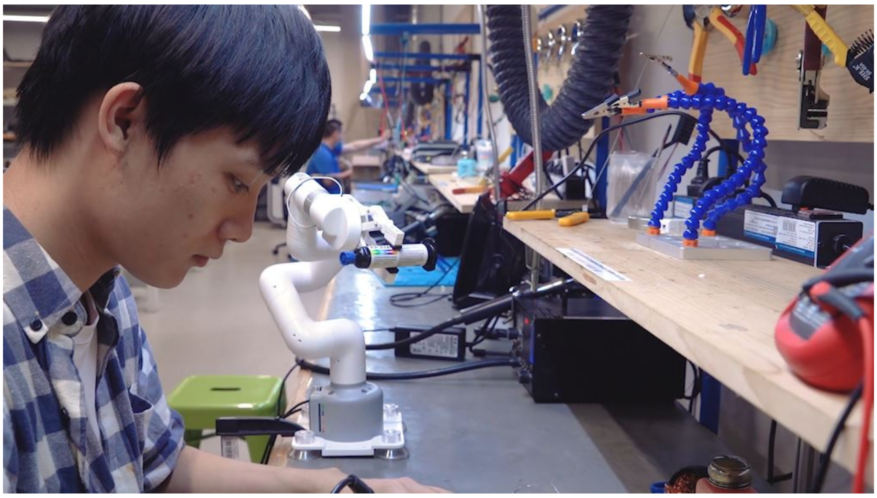
The design prototype of myCobot is from All-in-one Robot launched by Elephant Robot in China in 2018. As the first **all-in-one collaborative robot** in China, it has won "2019 CAIMRS Industrial Robot Innovation Award" and 2019 High-tech Robot Annual "Innovation Technology Award", and has been sold to more than 30 countries at home and abroad, receiving unanimous praise and recognition from the world's top 500 enterprises.

2 Introduction

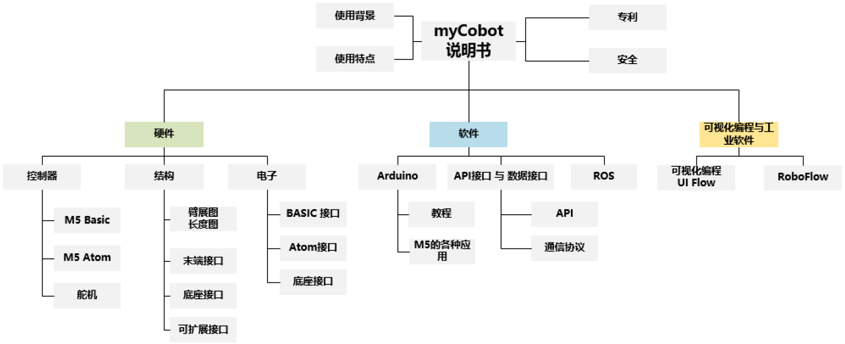
myCobot is the world's **smallest and lightest six-axis collaborative robot**, jointly produced by **Elephant Robotics** and **M5STACK**. It is more than a productivity tool full of imaginations, can carry on the secondary development according to the demands of users to achieve personalized customization.

With a weight of 850g, a payload of 250g and an arm length of 350mm, myCobot is compact but powerful, can not only be matched with a variety of end effectors to adapt to different kinds of application scenarios also support the secondary development of multi-platforms software to meet the needs of various scenarios such as **scientific research and education, smart home, and commercial applications**.





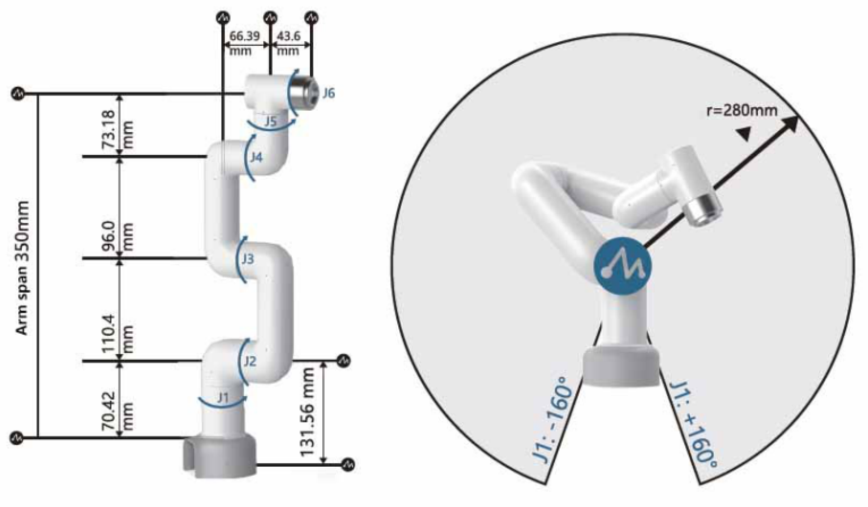
3 Framework



4 Parameters

表1： myCobot 280 Product Parameters

Parameter	Data
Model	myCobot-280
Working radius	280mm
Payload	250g
Arm Span	350mm
Repeatability	±0.5mm
Weight	850g
Power input	8V,5A
Working Conditon	-5°~45°
Communication	USB Type-C



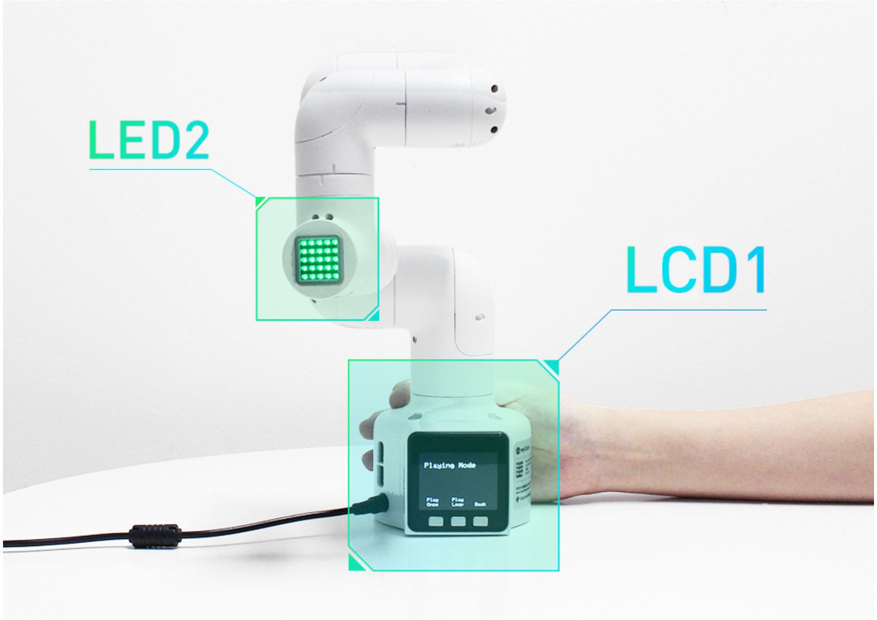
5 Accessories

- myCobot
- End Effectors
    - Parallel Gripper

    - Adaptive Gripper
    - Angled Gripper
    - Suction Pump
- Base
    - G Base
    - Flap Base
- Others
    - Rocker
    - Battery Case

6 Features

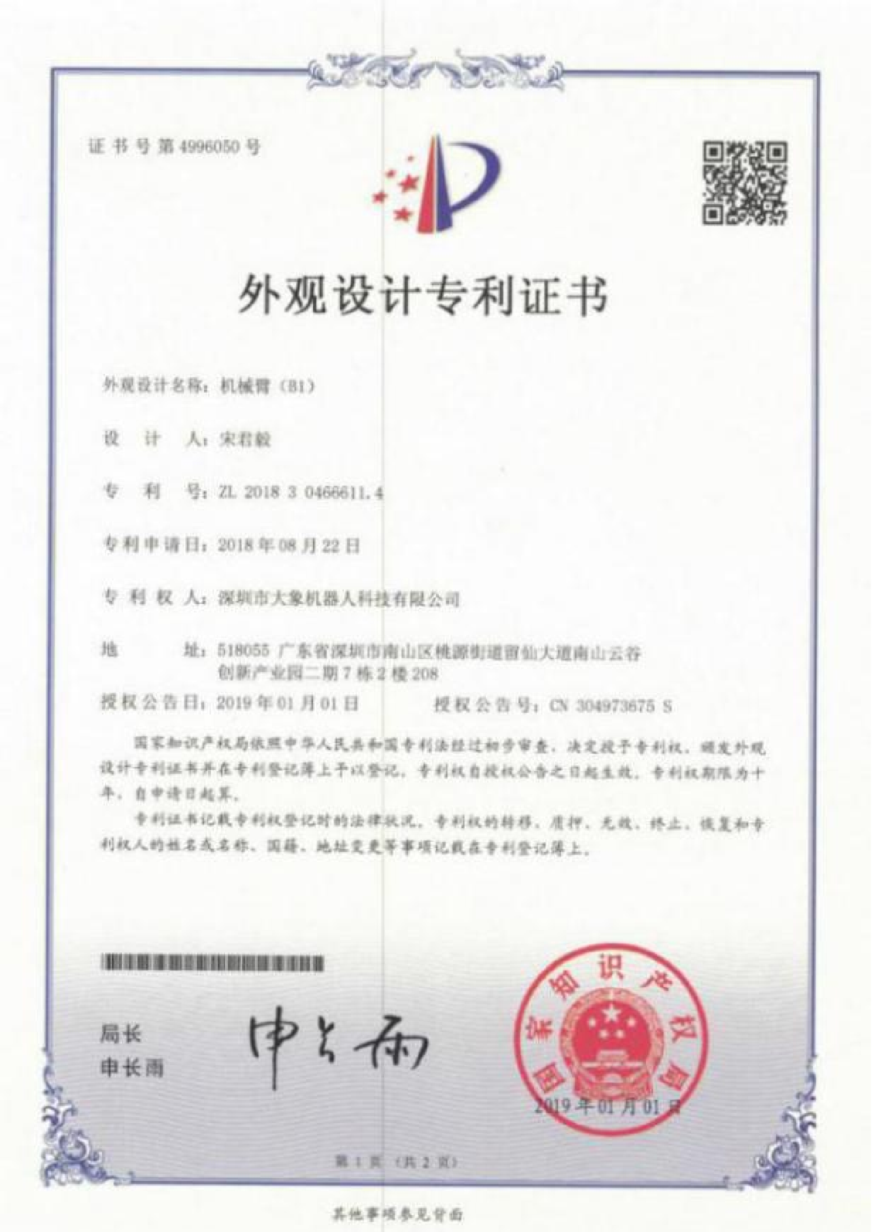
- **Unique Industrial Design & Extremely Compact**
  myCobot is an integrated modular design and only weighs 850g which is very easy to carry. Its overall body structure is compact with less spare parts and can be quickly disassembled and replaced to realize plug and play.
- **High configuration & Equipped with 2 Screens**
  myCobot contains 6 high-performance servo motors with fast response, small inertia and smooth rotation, and carries 2 screens supporting fastLED library to show the expanded application scene more easily and clearly.
- **Lego Connector & Thousands of M5STACK Ecological Application**
  The base and end of myCobot are equipped with Lego Connector, which is suitable for the development of various miniature embedded equipment. Its base is controlled by M5STACK Basic, and thousands of application cases can be use directly.
- **Bloky Programming & Supporting Industrial ROS**
  Using UIFlow visual programming software, programming myCobot is simple and easy for everyone. You can also Arduino, ROS, or other multiple functional modules of open source system, even RoboFlow, software of industrial robots from Elephant Robotics.
- **Track Recording & Learn by hand**
  Get rid of the traditional point saving mode, myCobot supports drag teaching to record the saved track and can save up to 60mins different tracks making it easy and fun for new players to learn.



7 Patents

> myCobot is protected by patents

NO.	Patent Name	Patent No.
1	Collaborative robotic arm	2020030683471.3
2	Mechanical arm linkage and mechanical arm	CN 208196791 U
3	Mechanical arm joint connector and mechanical arm	CN 208196840 U
4	Method and system for robot posture maintaining, dragging and teaching	ZL 2018 1 1634649.3
5	A robot online collision detection method and system based on momentum model	ZL 2019 1 0030748.9
6	A Kind of Robot Dynamic Parameter Identification Method Independent of Joint Angular Acceleration	ZL 2019 1 0773865.4



证书号第4996050号

外观设计专利证书

外观设计名称：机械臂（B1）

设　计　人：宋君毅

专　利　号：ZL 2018 3 0466611.4

专利申请日：2018年08月22日

专利权人：深圳市大象机器人科技有限公司

地　　址：518055 广东省深圳市南山区桃源街道留仙大道南山云谷
创新产业园二期7栋2楼208

授权公告日：2019年01月01日　　　授权公告号：CN 304973675 S

国家知识产权局依照中华人民共和国专利法经过初步审查，决定授予专利权，颁发外观设计专利证书并在专利登记簿上予以登记。专利权自授权公告之日起生效。专利权期限为十年，自申请日起算。

专利证书记载专利权登记时的法律状况。专利权的转移、质押、无效、终止、恢复和专利权人的姓名或名称、国籍、地址变更等事项记载在专利登记簿上。

局长
申长雨

申长雨

国家知识产权局
2019年01月01日

第1页（共2页）

其他事项参见背面

Reading Instruction

Reading Objectives

Gitbook is designed to help you to achieve goals

> Major Goals

- Understand the basic use of mechanics, electronics and software related to myCobot
- Understand the basic principle, joints, coordinates, terms, control of mechanical arm, and master simple forward and inverse kinematics calculations
- Understand the basic operation of API, uiFlow Visual Programming
- Have all skills to complete the intelligent warehouse suit, and can use myCobot for simple waypoint movement, handling, control, etc.

> Extended Goals

- Understand the image recognition algorithms related to machine vision
- Understand the construction of robot visual scene and the coordination methods and strategies between vision and manipulator
- Have all skills to complete the AI suit

Your Knowledge Level

Gitbook needs to be read according to your actual learning level and knowledge background. We divide them into three levels:

Level	Learning Background	Skills and Qualifications Requires	Estimated Learning Time	Sugg Develc Plat
Freshman	Major in Information, Electronics, Mechanical, Automation	Have a knowledge of programming languages Basic knowledge of electronics	100h	uiFlow
Superior	Knowledge of Arduino or other similar hardware products Knowledge of steering gear and programming Knowledge of IO interface, etc.	Know how to debug API interface Have a knowledge of communication	50h	Arduinc
Professional	Experience with at least one kind of industrial robot or consumer robot Ability to develop hardware and software	Understand the Cartesian coordinates Understand joint control Understand the basic use of robots	30h	all

Learning Schedule

NO	Target	Theory	Practice	Hour
1	Unboxing	1.track recording & learn by hand	1.accessories of myCobot 2.drive myCobot to track recording	1h
2	Background Knowledge Learning	1.application background of industrial robot 2.learning of coordinate and space, cartesian 3d coordinate and rotation, xyz 3.joint and coordinate control of industrial robot	1.joint control and reproduction 2.speed control 3.coordinate point position control and circulation	5h
3	Hardware Learning	1.principle and operation of embedded electronics 2.principles of steering gear and motor 3.actuator learning	1.control and drive of basic/atom 2.drive and motion of steering gear 3.study of robot accessories	5h
4	Software: Firmware and Updates	1.identify different software platforms and its purposes for use 2.principles of firmware loading and adaptation	1. choose the platform that works for you 2.download and update the firmware	2h
5	Software: Development Environment	1.building of arduino 2.library file download and update of arduino 3. have a knowledge of serial communication	1.be familiar with arduino 2.load the library 3.program and run the first line of code	2h
6	Learning and Development of Robot Library	1.basic communication and operation types of robots 2.common operation of robot 3.control of direction mode and coordinate mode	1.communicate with myCobot 2.control myCobot to move 3.operate I0 port, gripper and other signals	5h

NO	Target	Theory	Practice	Hour
7	uiFlow	1.understand the basic architecture and relationships of visual programming interface: sensors, actuators and processes 2.variables, loops, and judgments 3.control method of manipulator arm	1.displays different fonts in basic 2.make myCobot move in different positions according to the three buttons of basic 3.control myCobot to cycle multiple points	10h
8	roboFlow	1.learning of common industrial operating system for robots 2.common modules of roboFlow : point position, fast movement,IO control and input 3.advanced modules of roboFlow : looping, judging, and pallet procedures	1.control the movement of myCobot 2.basic control of IO input and output 3.cycle control and judgment	5h
9	Intelligent Warehouse	1.learning of robot point motion 2.rules for the placement of pallets 3.principle of end-effector control	1.operate the robot to move at different points 2.operate the robot to grab and place 3.control and recognition of gripper	10h

NO	Target	Theory	Practice	Hour
10	Algorithm about Image Recognition	1.introduction to the visual sensor of stickV 2.introduction and programming environment of maixpy 3.methods and strategies of common color recognition 4.methods and strategies of common shape recognition 5.methods and strategies of common area identification 6.json data transfer	1.build virtualbox environment 2.read different colors 3.recognition of different shapes 4.data transfer and reading in uiFlow	20h
11	Vision and Robotics Control	1.correlation of world and camera coordinates 2.standardization of qr code image 3. movement and correction	1.operate myCobot to camera coordinate system 2.the motion of myCobot in the camera coordinate system 3.re-calibration and setting	10h
12	Artificial Intelligence	1. learn and make the flow chart 2. learn and make electrical connection diagram 3. description and operation of shape classification	1.sensor connection 2.connection and drive of gripper 3.robot actuator & joint debugging of uiFlow	20h

In addition, you can also buy our **Industry 4.0 suit** to learn how to build and use the simulation industrial applications.

Service Support

If the above schdule cannot meet your needs, you can contact the **custom service** for further communication. We provide software and hardware customization service.

We will answer you as soon as possible (working day 9:30-18:30)

- Twitter: myCobot Official\@CobotMy

- Facebook: https://www.facebook.com/MyCobot-116558893805177
- Mail: support@elephantrobotics.com

4 Use cases

mycobot can be used for both **personal learning applications** and **educational applications**.

Personal Applications

Leisure & entertainment
dance with rhythm, play an instrument, take pictures and video, write painting, play games and so on

Intelligent furniture
remote control of myCobot to grab and carry objects with gripper and suction pump, such as mixing coffee, grabing bread, etc.

Teaching AIDS
simulating the teaching of industrial robot, K12 teaching through entertainment experience, cost-effective research laboratory assistant

Scientific and technological equipment
carry AGV car to realize automatic navigation and handling, combined with vision to do infrared thermal imager, face tracking, etc.

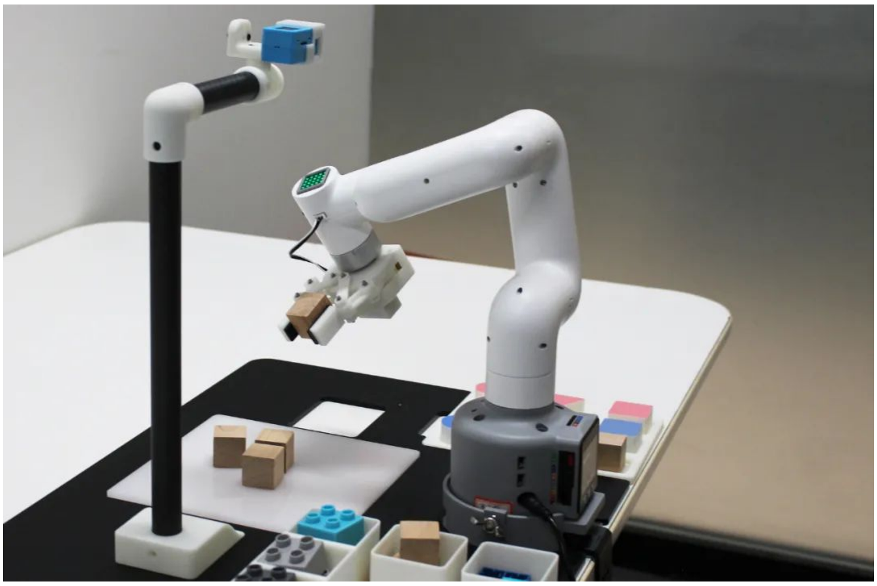
Commercial Exhibition
cooperate with other products to do commercial scene demonstration

stickT

Educational Applications

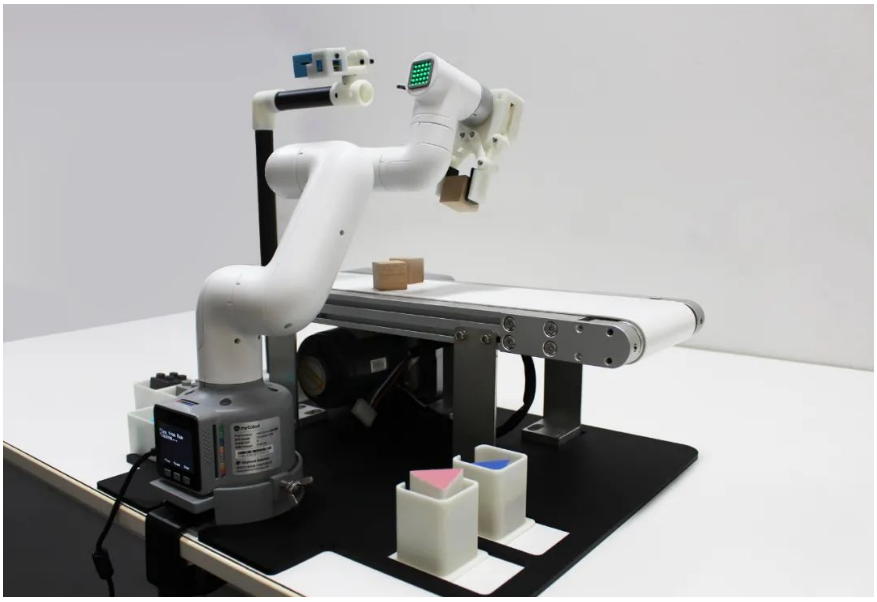
1 Maker Education /K12 Education

- Learning Platform: myCobot + Arduino/ROS + UIFlow
- Learning Purpose: independent development, creative learning
- Corresponding Suit: basic suit



2 Artificial Intelligence Course

- Learning Platform: myCobot + StickV camera
- Learning Purpose: robot algorithm, visual recognition algorithm
- Corresponding Suit: artificial intelligence suit
- Suit Content: visual system, feeding table



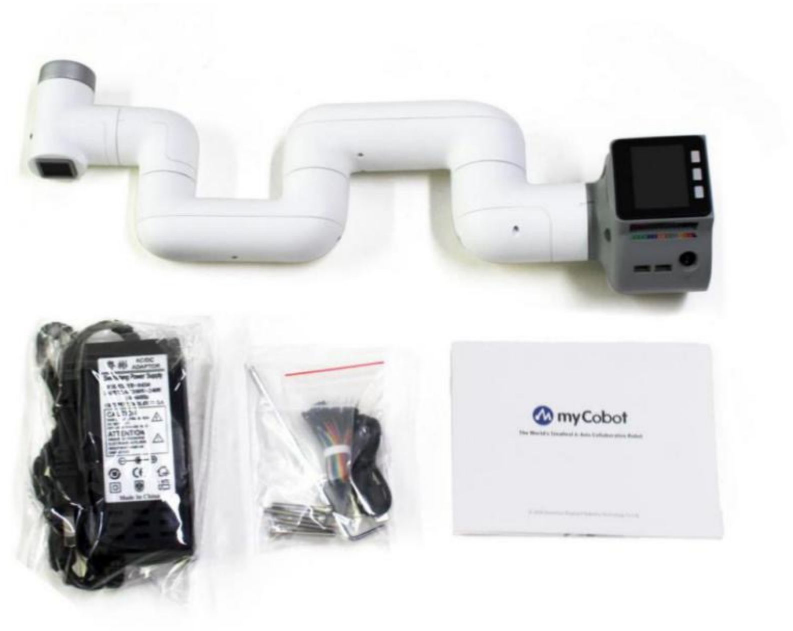
3 Vocational Education/Higher Education

- Learning platform: myCobot + RoboFlow
- Learning Purpose: robot algorithm, robot movement, robot simulation
- Corresponding Suit: artificial intelligence suit
- Suit Content: visual system, conveyor belt, feeding table, material block

4 Quick Start

Step 1: What’s in the Box

After the packaging box is in place, please confirm that the robot packaging is intact and undamaged. **If there is any damage, please contact the logistics company and the local supplier in time.**



1 myCobot 280【standard set】

- myCobot-280
- Brochure
- Power Supply
- USB-Type C
- Jumper
- M4*35, stainless steel screw
- Hexagon wrench

2 Operating Environment and Conditions

Please install the robot system in an environment that meets the conditions described in the table in order to exert and maintain the performance of the machine and use it safely.

Environment	Target
Temperature	-10℃~45℃
Relative Humidity	20%~70%
Indoor/Outdoor	Indoor
Another Environmental Requirement	-Avoid sunlight. -Keep away from dust, oily smoke, salt, iron filings, etc. -Keep away from flammable and corrosive liquids and gases. -Do not contact with water. -Does not transmit shock, vibration, etc. -Keep away from strong electromagnetic interference sources.

Step 2: Fix the robot

You can fix the robot by our base, or design your own customized base to fix.

The specific fixation method can refer to the fixation of this机械臂的固定

Step 3: Power Supply

myCobot must use an external power source to provide enough electricity.

- **Rated Voltage:** 7-9V
- **Rated Current:** 3-5A
- **The type of plug:** DC 5.5mm×2.1

Note: You can’t simply supply with typeC inserted into Basic.

Please use the official power supply to avoid damage to myCobot.

Step 4: Drag Teaching Demonstration

Before Recording:

> After entering the recording mode, select the recording storage location

- Button A: Store to Ram
- Button B: Store to Memory Card
- Button C: Exit the Recording Mode

Start Recording:

> Click record and select the storage location, then manually drag the robotic arm to complete your target action and remeber to save it, the action will be recorded and stored .
>
> Note: The default recording time is 100s.If the recording time is too long that will exceed the memory, you can customize it by modifying the code, or record it on the computer.

Play:

Click Button A to play the recored action

- Button A: Start Playing the Recorded Action
- Button B: Pause
- Button C: Exit Playback

Open source code: The code above is open, named MainControl, you can download and customize the code via github, refer to the link below:

https://github.com/elephantrobotics/myCobot/tree/main/Arduino/MycobotBasic/examples

Video Tutorials: https://youtu.be/WzrbOrdQop0

Raspberry myCobot

Begin your myCobot

Firstly, you need to connect power supply, display, keyboard and mouse as shown in the following image, then initiate myCobot by the red switch.

The use of python

Start quickly

We've pre-loaded the python API package in the Raspberry Pi version of myCobot `pymycobot` , just use it in your code.

You can create a python file at any position, for example: `light_led.py` , then write the following code in the file:

```
from pymycobot.mycobot import MyCobot
from pymycobot import PI_PORT, PI_BAUD

mc = MyCobot(PI_PORT, PI_BAUD)
mc.set_color(255, 0, 0)
```

After saving, run it. You'll see the light board on top of myCobot lit in red.

Update the library

If our library is updated, it can also be easily synchronized on Raspberry Pi.

python2 update:

```
[sudo] pip install pymycobot --upgrade
```

python3 update:

```
[sudo] pip3 install pymycobot --upgrade
```

More examples

In addition to the above content, we provide more examples to help users use the python API.

You can download it at the following address:

https://www.elephantrobotics.com/wp-content/uploads/2021/04/PythonAPI tutorials.zip

The use of ROS

We've pre-installed ROS kinetic in the Raspberry version of myCobot and provide the package `myCobotROS` , you can use it easily.

1. Go to the working directory:

```
cd ~/ros_catkin_ws
source devel/setup.bash
```

1. Go to myCobotROS's redeme, follow `3.Visualization in RViz` , the address is as follows:

https://github.com/elephantrobotics/myCobotROS#3-visualization-in-rviz

Python API of myCobot

```
这是用于与mycobot进行串行通信并对其进行控制的python API。
```

Installation

Notes：

Make sure that you flash `Atom` to the top Atom，flash `Transponder` to the Basic. Firmware `Atom` and `Transponder` Download link：https : //github.com/elephantrobotics/myCobot/tree/main/Software You can also use myStudio to update，download link of myStudio：https : //github.com/elephantrobotics/myStudio/releases

Pip

```
pip install pymycobot --upgrade --user
```

Source code

```
git clone https://github.com/elephantrobotics/pymycobot.git <your-path>
cd <your-path>/pymycobot
# Install
[sudo] python2 setup.py install
# or
[sudo] python3 setup.py install
```

Usage:

```
from pymycobot.mycobot import mycobot
from pymycobot.mycobot import Angle, Coord
from pymycobot import PI_PORT, PI_BAUD # For raspberry pi version of mycobot.
```

The `demo` directory stores some test case files.

You can find out which interfaces pymycobot provides in `pymycobot/README.md`.

Please go to here.

1、The robot arm swings left and right

Introduction of API

API used to control the robot arm swings left anf right is:

1、 `MyCobot(port)`

Function: Initialize a MyCobot object.

Parameter Description: `port`： The type of data is String, is the port number that controls the robot arm, and the windows system can view it at the port in Device Manager.

2、 `get_angles()`

Function: Get the angle of the six joint points of the robot arm

Return Value: The type of return value is list, with six elemental data, corresponding to joints 1 to 6.

3、 `send_angles(degrees,speed)`

Function: Set the angle of six joint points at a time.

Parameter Description:

`degrees`： The type of parameter is list. The angle data for the six joint points must be included. The angle value range of the six joint points is -180 to 180.

`speed`： The type pf data is int, value range 0~100. Represents the speed at which the robot arm runs to the specified position, and the higher the value, the greater the speed.

4、 `send_angle(id, degree, speed)`

Function: Set the angle of a single joint.

Parameter Description

`id`： Represents the joints of the robotic arm, a total of six joints, with a specific representation. For example, joint1 can be: `Angle.J1.value`。

`degree`： Represents the angle of the joint, value range -180 to 180.

`speed`： Represents the speed of the robot arm

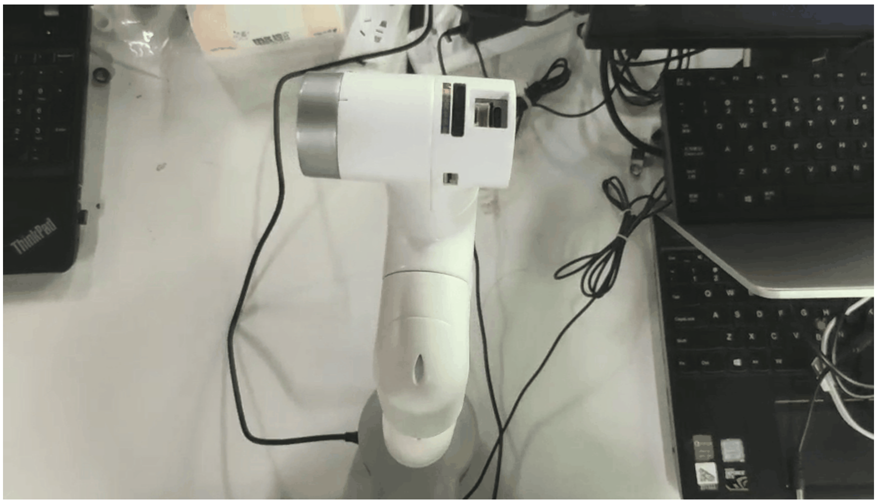
5、 `release_all_servos()`

Function: Release the robot arm, let it swing manually at will.

Content of code

```
from pymycobot.mycobot import MyCobot
from pymycobot.genre import Angle
from pymycobot import PI_PORT, PI_BAUD  # When using the Raspberry Pi version of myCob
import time

# Initialize a myCobot object
mc = MyCobot(PI_PORT, PI_BAUD)

# Get the coordinates of the current location
angle_datas = mc.get_angles()
print(angle_datas)

# Pass coordinate parameters with a number of columns, let the robot arm move to the s
mc.send_angles([0, 0, 0, 0, 0, 0], 50)
print(mc.is_paused())
# Set the wait time to ensure that the robot arm has reached the specified position
# while not mc.is_paused():
time.sleep(2.5)

# Move joint 1 to the position of 90
mc.send_angle(Angle.J1.value, 90, 50)

# Set the wait time to ensure that the robot arm has reached the specified position
time.sleep(2)

# Set the number of cycles
num = 5

# Let the robot arm swing left and right
while num > 0:
    # Move joint 2 to the position of 50
    mc.send_angle(Angle.J2.value, 50, 50)

    # Set the wait time to ensure that the robot arm has reached the specified positic
    time.sleep(1.5)

    # Move joint 2 to the position of -50
    mc.send_angle(Angle.J2.value, -50, 50)

    # Set the wait time to ensure that the robot arm has reached to the specified posi
    time.sleep(1.5)

    num -= 1

# Let the robot arm shrink. You can swing the robot arm manually and then use the get_
# Use this function to get the robot arm to where you want it to be.
mc.send_angles([88.68, -138.51, 155.65, -128.05, -9.93, -15.29], 50)

# Set the wait time to ensure that the robot arm has reached to the specified positior
time.sleep(2.5)

# Release the robot arm, let it swing manually at will
mc.release_all_servos()
```

Effects as shown



2、The head of the robot arm intelligently plans the route

Knowledge preparation for API

`send_coords([x,y,z,rx,ry,rz],speed,model)` is used to control the movement of the head of the robot arm to a specified point in a specified manner.It is primarily used to enable intelligent planning of the head of the robot arm from one position to another.X,Y,Z represents the position of the head of the robot arm in space（The coordinate system is a right-angle coordinate system），rx,ry,rz represents the posture of the head of the robot arm at that point.（The coordinate system is Euler coordinates）。The implementation of the algorithm and the representation of Euler coordinates require a certain degree of academic knowledge, we won't make too much explanation here, we only need to understand the right angle coordinate system can we use this function well.

Introduction of API

1、`send_coords([x,y,z,rx,ry,rz],speed,model)`

Function：Intelligently plan the route to move the head of the robot arm from the original point to the specified point.

Parameter Description

`[x,y,z,rx,ry,rz]`：The parameters are combined into a spatial right-angle coordinate system with x, y, z. The bottom of the robot arm is as the origin, x-positive axis in front, y-positive axis on the right and z-axis on the top. `[x,y,z]` represented as the position of the head of the robot arm. `[rx,ry,rz]` indicate the posture of the head of the robot arm. You can adjust the posture of the head of the robot arm and then use `get_coords()` to get the posture in the position, In this way you don't need to know the Euler coordinate system can you also understand the function.

`speed`：Represents the speed of the robot arm. The range of values is 0 to 100, the higher the value, the faster the speed.

`model`：Values are limited to 0 and 1. 0 indicates that the path to the head of the robot arm is nonlinear, means the route is randomly planned, as long as the head of the robot arm moves to the specified point in a specified manner. 1 indicates the movement of the head of the robot arm is linear, means the intelligent planning route allows the head of the robot arm to move in a straight line to the specified point.

2、`get_coords()`

Function: Get the spatial coordinates of the head of the robot arm at this time and the current posture.

Return value：The returned type is a list collection of six float elements, the first three coordinates x,y,z representing the coordinates of the head of the robot arm, and the last three coordinates rx, ry, rz representing the posture of the head of the robot arm.

3、 `send_coord(id,coord,speed)`

Function：Only one of the coordinates of the x, y, z axis in the head space coordinates of the robot arm is modified separately.

Parameter Description：

`id` ：The type of data is `genre.Coord` , it represents the X, Y, Z axis of the head of the robot arm. For example: `Coord.X.value` 。

`coord` ：The type of data is float, Represents modifying the coordinate value.

`speed` ：Represents the speed of the robot arm. Value range 0~100, the higher the value, the faster it is.

Content of code

```
from pymycobot.mycobot import MyCobot
from pymycobot.genre import Coord
from pymycobot import PI_PORT, PI_BAUD  # When using the Raspberry Pi version of myCob
import time

# Initialize a mycobot object
mc = MyCobot(PI_PORT, PI_BAUD)
# Gets the angle and posture of the current head
coords = mc.get_coords()
print(coords)
# Intelligently plan your route, Let the head linearly arrive at the coordinate[59.9,-
mc.send_coords([59.9, -65.8, 250.7, -50.99, 83.14, -52.42], 80, 1)
# Set the wait time
time.sleep(1.5)

# Intelligently plan your route, Let the head linearly arrive at the coordinate[59.9,-
mc.send_coords([59.9, -65.8, 350.7, -50.99, 83.14, -52.42], 80, 1)
# Set the wait time
time.sleep(1.5)
# Change only the x coordinates of the head,Set the x coordinate of the head to -40. L
mc.send_coord(Coord.X.value, -40, 70)
```

Effects as shown

3、Safety control of th robot arm

Introduction of API

1、 `is_power_on()`

Function: Determine whether the robot arm is powered.

Return calue：1 indicates powered，0 indicates outage，-1 indicates an error

2、 `power_on()`

Function: Power the robot arm

3、 `power_off()`

Function: Power is lost to the robot arm and all functions will fail.

Note: The robot arm cannot be relaxed after outage，means `set_free_mode()` is invalid.

4、 `pause()`

Function: Pause the movement of the robot arm.

5、 `resume()`

Function: Restore the movement of the robot arm.

6、 `stop()`

Function: The robot arm stops moving

7、 `is_in_position(data,flag)`

Function: Determines whether the robot arm has reached the specified position

Parameter Description:

`data` ： A list collection of six elements, representing an angle collection, or a collection of robot arm head data.

`flag` ： 1 Represents the head data of the robot arm，0 Represents the angle collection data of the robot arm.

8、 `is_paused()`

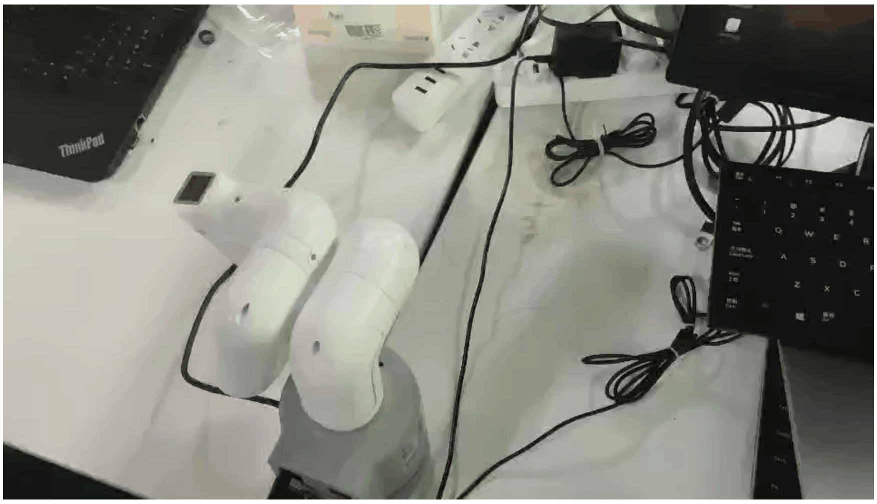
Function: Determine if the robot arm is suspended

Return value：1 indicates a pause, 0 indicates that it is not paused, and -1 indicates an error.

Content of code

```
from pymycobot.mycobot import MyCobot
from pymycobot import PI_PORT, PI_BAUD  # When using the Raspberry Pi version of myCob
import time

# Initialize a MyCobot object
mc = MyCobot(PI_PORT, PI_BAUD)
# Determine whether the robot arm is powered or not, and if there is no power supply,
if not mc.is_power_on():
    # Power the robot arm
    mc.power_on()
# The robot arm arrive at position[0,0,0,0,0,0] at 30 speeds
mc.send_angles([0, 0, 0, 0, 0, 0], 30)
# Get the current time
start = time.time()
# Check whether the robot arm has arrived at the position [0,0,0,0,0,0] or not
while not mc.is_in_position([0, 0, 0, 0, 0, 0], 0):
    # Restore the robot arm movement
    mc.resume()
    # Let the robot arm move 0.5s
    time.sleep(0.5)
    # Pause the robot arm movement
    mc.pause()
    # Determine if the movement timed out
    if time.time() - start > 9:
        # Stop movement of the robot arm
        mc.stop()
        break
# Get the current time
start = time.time()
# The robot arm arrive at position[88.68, -138.51, 155.65, -128.05, -9.93, -15.29] at
mc.send_angles([88.68, -138.51, 155.65, -128.05, -9.93, -15.29], 30)
# Check whether the robot arm has arrived at the position [88.68, -138.51, 155.65, -12
while not mc.is_in_position([88.68, -138.51, 155.65, -128.05, -9.93, -15.29], 0):
    # Restore the movement of the robot arm
    mc.resume()
    # Let the robot arm move 0.5s
    time.sleep(0.5)
    # Pause the movement of the robot arm You can use is_paused() to check whether the
    mc.pause()
    # Check if the movement timed out
    if time.time() - start > 9:
        mc.stop()
        # Stop the movement of the robot arm
        break
# Power off the robot arm after operation
mc.power_off()
```

Effects as shown



ThinkPad

4、Test the robot arm

Introduction of API

1、 `is_all_servo_enable()`

Function：Determine if six joint points are working properly.

Return parameter：1 indicates working properly，0 means that it doesn't work，-1 indicates an error alarm.

2、 `jog_angle(joint_id,direction,speed)`

Function：Let a joint point move continuously.

Parameter Description：

`joint_id`： value range1-6, six integers represent joint points 1 to 6.

`direction`： 1 indicates increase in angle，0 indicates decrease in angle.

`speed`： Indicates the speed of increase and decrease.

3、 `jog_stop()`

Function：Stop the movement of joint points.

4、 `release_servo(servo_id)`

Function：Relax the specified joint points.

Parameter Description：

`servo_id`： valuer range 1-6, six integers represent joint points 1 to 6.

5、 `focus_servo(servo_id)`

Function：Secure the specified joint points

Parameter Description

`servo_id`： valuer range 1-6, six integers represent joint points 1 to 6.

Content of code

```
from pymycobot.mycobot import MyCobot
from pymycobot import PI_PORT, PI_BAUD  # When using the Raspberry Pi version of myCob
import time

# Initialize a MyCobot object
mc = MyCobot(PI_PORT, PI_BAUD)
# Determine whether the robot arm is powered or not, and if there is no power supply,
if not mc.is_power_on():
    # Power the robot arm
    mc.power_on()

# Check that the six joints are working properly
# You can also use is_servo_enable(servo_id) to change single calibration
if mc.is_all_servo_enable():
    # Power off the robot arm
    mc.power_off()
    # Check whether the robot arm is power off
    if mc.is_all_servo_enable() == -1:
        print("机臂
```

Effects as shown

5、Dance of the robot arm

Introduction of API

1、 `set_color(R, G, B)`

Function：Set the color of the headlights of the robot arm

Parameter Description：

R,G,B：Corresponding to the value of color array[R,G,B]

Content of code

```
from pymycobot.mycobot import MyCobot
from pymycobot import PI_PORT, PI_BAUD  # When using the Raspberry Pi version of myCob
import time

if __name__ == '__main__':
    # Initialize a MyCobot object
    mc = MyCobot(PI_PORT, PI_BAUD)
    # Set the start time
    start = time.time()
    # Let the robot arm move to the specified position
    mc.send_angles([-1.49, 115, -153.45, 30, -33.42, 137.9], 80)
    # Check whether it move to the specified positon
    while not mc.is_in_position([-1.49, 115, -153.45, 30, -33.42, 137.9], 0):
        # Restore the movement of the robot arm
        mc.resume()
        # Let the robot arm move 0.5s
        time.sleep(0.5)
        # Pause the movement of the robot arm
        mc.pause()
        # Check if the movement timed out
        if time.time() - start > 3:
            break
    # Set start time
    start = time.time()
    # Let the movement last 30 seconds
    while time.time() - start < 30:
        # Let the robot arm reach this position quickly
        mc.send_angles([-1.49, 115, -153.45, 30, -33.42, 137.9], 80)
        # Set the color of the light to[0,0,50]
        mc.set_color(0, 0, 50)
        time.sleep(0.7)
        # Let the robot arm reach this position quickly
        mc.send_angles([-1.49, 55, -153.45, 80, 33.42, 137.9], 80)
        # Set the color of the light to[0,50,0]
        mc.set_color(0, 50, 0)
        time.sleep(0.7)

    # mc.release_all_servos()
```

Effects as shown

6、Use of gripper

Introduction of API

1、 `is_gripper_moving()`

Function：Determine if the gripper is running.

Return parameter：

1 means running, 0 means no, and -1 indicates an error

2、 `set_encoder(joint_id, encoder)`

Function：Turn the specified joint point to the specified position

Parameter Description

`joint_id` ： value range 1-7, Represents 1 to 6 joint points and grippers respectively.

`encoder` ： Value range 0~4096，2048 means 0 when the value range is -180--180

3、 `set_encoders(encoders, sp)`

Function：Let the robot arm move to the specified position.

Parameter Description：

`encoders` ： A list collection of six int elements, with the order of six encoder data representing the position of 1 to 6 joint points.

`sp` ： Indicates the rotate speed of the robot arm.

4、 `get_encoder(joint_id)`

Function：Gets encoder data for the specified joint point.

Parameter Description

`joint_id` ： Value range 1-7, reprents 1-6 joint points and grippers respectively.

Return value：

`encoder` ： Indicate the encoder data information of this joint

5、 `set_gripper_value(value, speed)`

Function：Let the gripper turn to the specified position at the specified speed.

Parameter Description:

`value` ： Indicates where the claws are to be reached, value range 0~4096。

`speed` ： Indicates how much speed it turns, value range 0~100。

6、 `get_gripper_value()`

Function：Get the `encoder` data information of gripper

Return value：

`encoder`： the data information of gripper

7、 `set_gripper_state(flag, speed)`

Function：Let the gripper reach the specified state at the specified speed.

Parameter Description

`flag`： 1 means claws are closed and 0 means claws are open.

`speed`： Indicates how quickly the specified state has been reached，value range 0~100。

Content of code

```
from pymycobot import PI_PORT, PI_BAUD  # When using the Raspberry Pi version of myCob
from pymycobot.mycobot import MyCobot
import time


def gripper_test(mc):
    print("Start check IO part of api\n")
    # Check if the gripper is moving
    flag = mc.is_gripper_moving()
    print("Is gripper moving: {}".format(flag))
    time.sleep(1)

    # Set the current position to (2048).
    # Use it when you are sure you need it.
    # Gripper has been initialized for a long time. Generally, there
    # is no need to change the method.
    # mc.set_gripper_ini()
    # Set joint point 1, let it rotates to the position 2048
    mc.set_encoder(1, 2048)
    time.sleep(2)
    # Set 6 joint position, let the robot arm rotates to the position at 20 speeds.
    mc.set_encoders([1024, 1024, 1024, 1024, 1024, 1024], 20)
    time.sleep(3)
    # Get location information for joint point 1
    print(mc.get_encoder(1))
    # Set the gripper rotate to the position 2048
    mc.set_encoder(7, 2048)
    time.sleep(3)
    # Set the gripper rotate to the position 1300
    mc.set_encoder(7, 1300)
    time.sleep(3)

    # Set the gripper to the state 2048 at 70 speeds.
    mc.set_gripper_value(2048, 70)
    time.sleep(3)
    # Set the gripper to the state 1500 at 70 speeds.
    mc.set_gripper_value(1500, 70)
    time.sleep(3)

    # Set the state of gripper，Let it open its paws quickly at 70 speeds
    mc.set_gripper_state(0, 70)
    time.sleep(3)
    # Set the state of gripper，Let it close its paws quickly at 70 speeds
    mc.set_gripper_state(1, 70)
    time.sleep(3)

    # Get the value of gripper
    print("")
    print(mc.get_gripper_value())


if __name__ == "__main__":
    # Initialize a MyCobot object
    mc = MyCobot(PI_PORT, PI_BAUD)
    # Let it move to the zero position
    mc.set_encoders([2048, 2048, 2048, 2048, 2048, 2048], 20)
    time.sleep(3)
    gripper_test(mc)
```

Effects as shown

7、Calibration of the robot arm

Introduction of API

1、 `set_servo_calibration(servo_no)`

Function：When the position is adjusted to zero, there is impalignment of the port of the robot arm, and the wrong joint can be calibrated using this method.

Parameter Description

`servo_no` ： Value range 1~6，represents 6 joint points.

2、 `is_controller_connected()`

Function: Determines whether the current robot arm is in a writeable state

Return value： 1 means writeable, 0 means you can't write, -1 indicates an error.

3、 `set_gripper_ini()`

Function: Set 6 joints to the initial position.

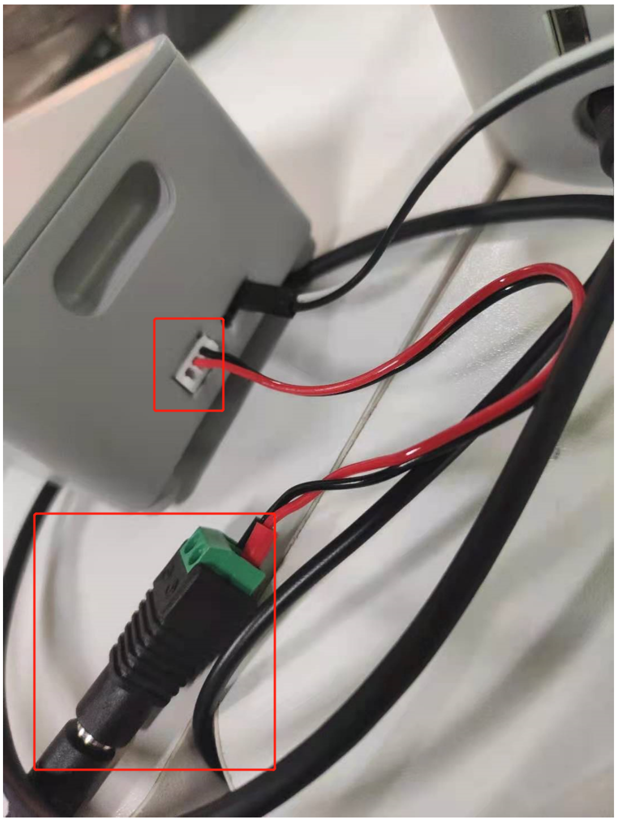
Content of code

```
from pymycobot.mycobot import MyCobot
from pymycobot import PI_PORT, PI_BAUD  # hen using the Raspberry Pi version of myCobo

# Initailize a MyCobot object
mc = MyCobot(PI_PORT, PI_BAUD)

# Detect whether the robot arm can burn into the program
if mc.is_controller_connected() != 1:
    print("Connect the robot arm correctly for program writing")
    exit(0)

# Fine-tune the robot arm to ensure that all snaps in the adjusted position are aligne
# Based on the alignment of the robot arm port, only a case is given here
mc.send_angles([0, 0, 18, 0, 0, 0], 20)

# Calibrate the position at this time, and the angle position after calibration is rep
# The for loop is equivalent to the set_gripper_ini() method
for i in range(1, 7):
    mc.set_servo_calibration(i)
```

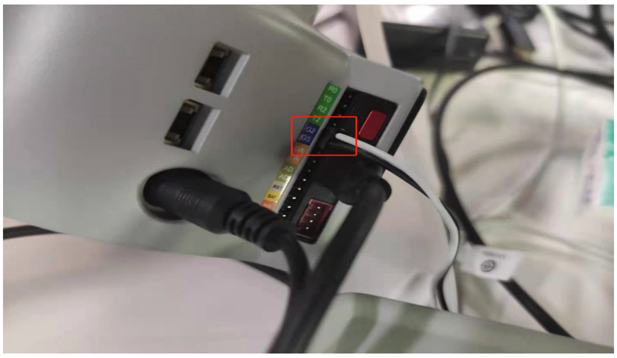
8、Control suction pump

Connection

Firstly, we need to power the sunction pump as shown in the picture



Next we need to connect sunction pump and myCobot, using control cables. Connect 2 and 5 in the pin port. As shown in the picture.



Introduction of API

`set_basic_output(pin_no, pin_signal)`

Function：By connecting `pin_no` and control signal `pin_signal` to control external devices.

Parameter Description:

`Pin_no`： The int type parameter, the number of the label at the bottom of the device takes only the numeric portion.

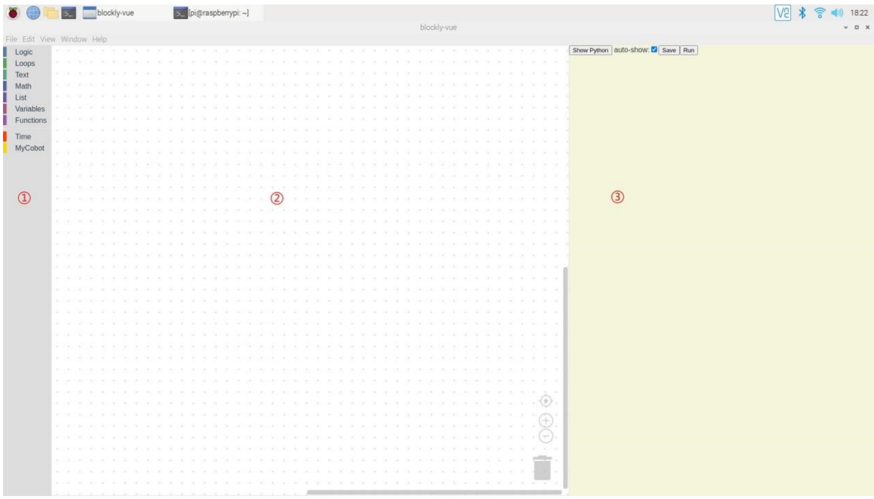
`Pin_signal`： 1 means stop running, 0 means run.

Content of code

```python
from pymycobot.mycobot import MyCobot
from pymycobot import PI_PORT, PI_BAUD  # When using the Raspberry Pi version of myCob
import time

# Initialize a MyCobot object
mc = MyCobot(PI_PORT, PI_BAUD)

# Open sunction pump
def pump_on():
    # Let 2 work
    mc.set_basic_output(2, 0)
    # Let 5 work
    mc.set_basic_output(5, 0)

# Stop sunction pump
def pump_off():
    # Let 2 stop working
    mc.set_basic_output(2, 1)
    # Let 5 stop working
    mc.set_basic_output(5, 1)


pump_off()
time.sleep(3)
pump_on()
time.sleep(3)
pump_off()
time.sleep(3)
```

Use Instruction of Myblockly

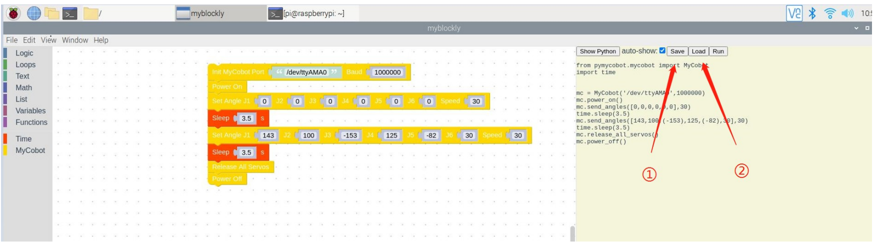
Interfacce introduction



1-1interface

As 1-1 indicates， ① represents the Puzzle Toolbar, which contains logical control puzzles, variable settings puzzles, mathematical function puzzles, text type puzzles, and control robot arm method puzzles. ② represents a jigsaw puzzle board, pulls the method module in the puzzle toolbar into the puzzle board, and the method module will appear in the drawing board. ③ represents the code display area, and the method module stitched into the drawing board automatically generates python code in the code display area.

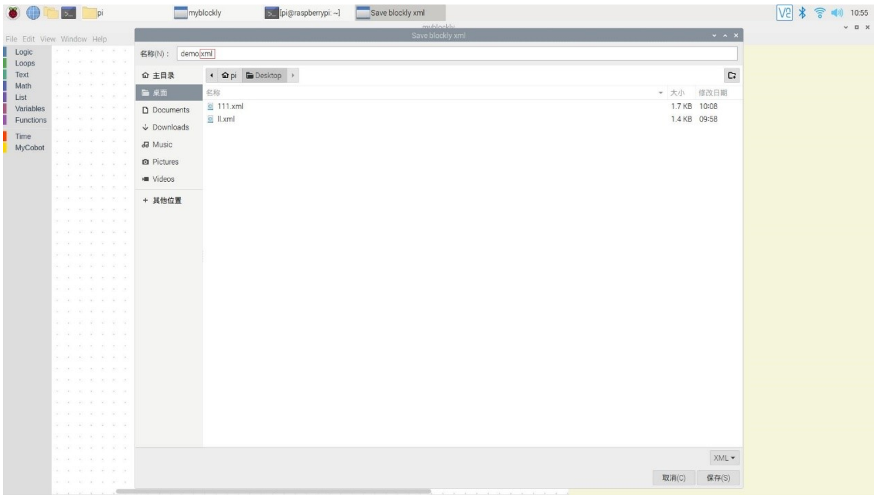
Introduction of save and load


myblockly
[pi@raspberrypi: ~]
File Edit View Window Help
Logic
Loops
Text
Math
List
Variables
Functions
Time
MyCobot
Init MyCobot Port "/dev/ttyAMA0" Baud 1000000
Power On
Set Angle J1 0 J2 0 J3 0 J4 0 J5 0 J6 0 Speed 30
Sleep 3.5 s
Set Angle J1 143 J2 100 J3 -153 J4 125 J5 -82 J6 30 Speed 30
Sleep 3.5 s
Release All Servos
Power Off
Show Python
auto-show: ☑
Save
Load
Run
①
②

```
from pymycobot.mycobot import MyCobot
import time


mc = MyCobot('/dev/ttyAMA0',1000000)
mc.power_on()
mc.send_angles([0,0,0,0,0,0],30)
time.sleep(3.5)
mc.send_angles([143,100,(-153),125,(-82),30],30)
time.sleep(3.5)
mc.release_all_servos()
mc.power_off()
```

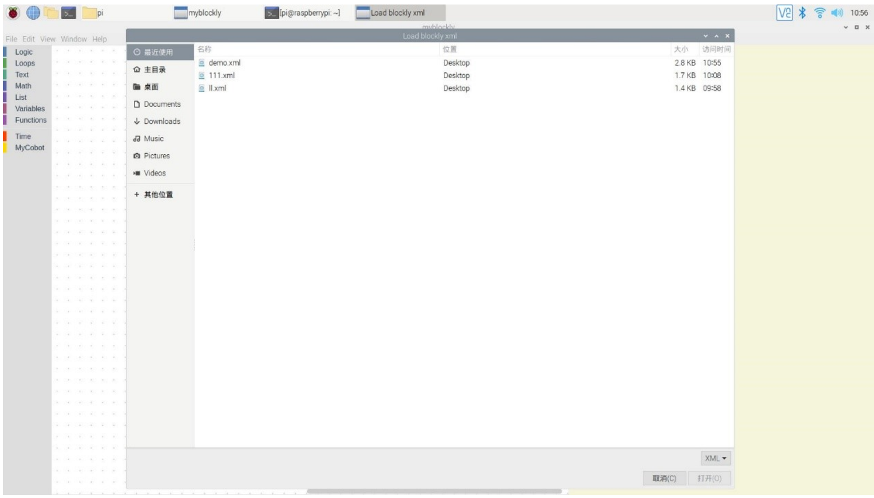
1-2Save and Load

After programming, you can click ①save in the picture1-2 to save, the effect after clicking is shown in the picture1-3.



1-3Save Interface

In particular, it is important to add a suffix .xml when defining a saved file name, otherwise the saved file will not be loaded. Of course, if you accidentally forget to add the suffix.xml. It doesn't matter, just rename the file before loading it and add .xml suffix.



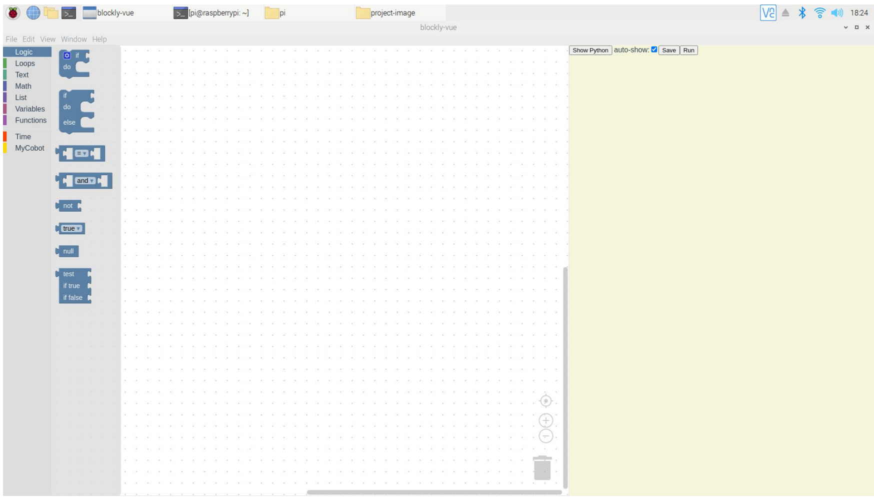
1-4Load Interface

You can click ②Load in the picture1-2 to load the file, the effect after clicking is shown in the picture1-4. The file that is loaded at this time can only be a file with a .xml suffix.

Intrduction of Myblockly

Logic

1、As shown in Figure 1-1, all methods contained in the Logic module are included.

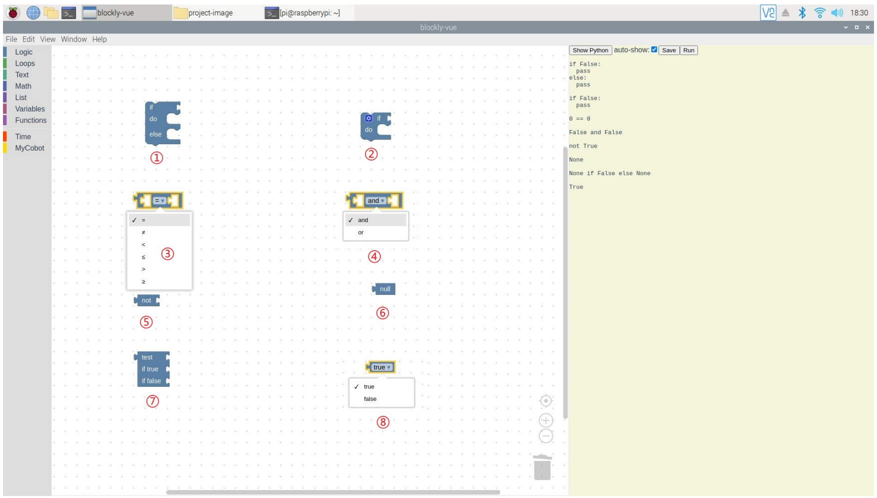


1-1 Logic Module Presentation

2、Method is explained in detail

1-2 detailed method（一）

```
As 1-2 indicates, ① indicates if（condition）do（program）method. If the conditions ar
```



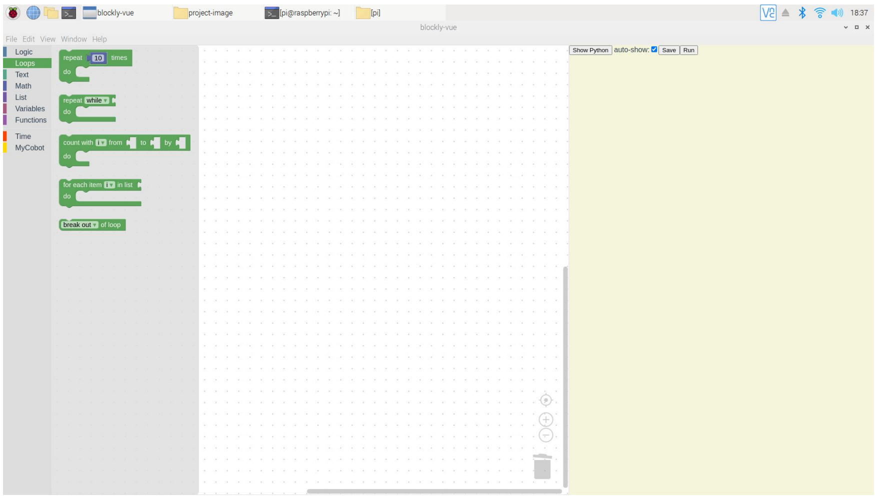
```
if False:
  pass
else:
  pass

if False:
  pass

0 == 0

False and False

not True

None

None if False else None

True
```

1-3 方法详细（二）

```
如图1-3所示，①表示if（条件）do（程序1）else（程序2），若满足条件则执行程序1，否则执行程序2。
```

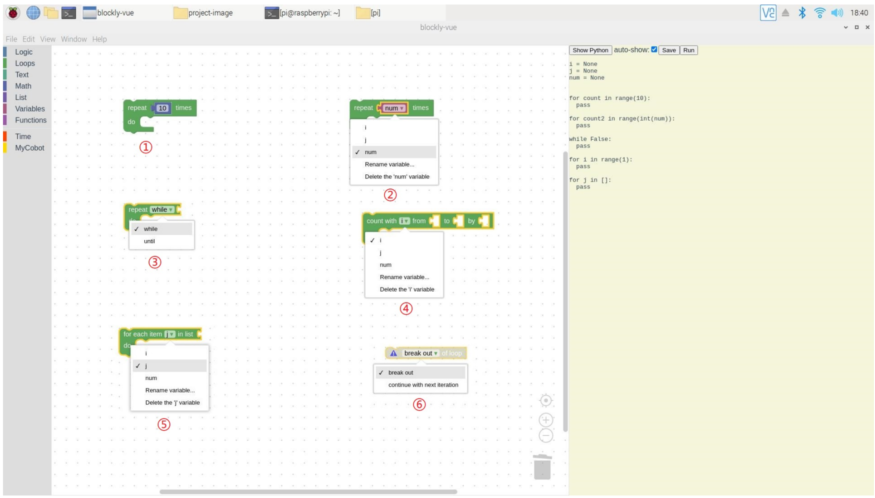
Loops

1、如图2-1所示即为Loops模块所包含的所有方法。



2-1 Loops模块

2、Method is explained in detail



```
i = None
j = None
num = None


for count in range(10):
  pass

for count2 in range(int(num)):
  pass

while False:
  pass

for i in range(1):
  pass

for j in []:
  pass
```

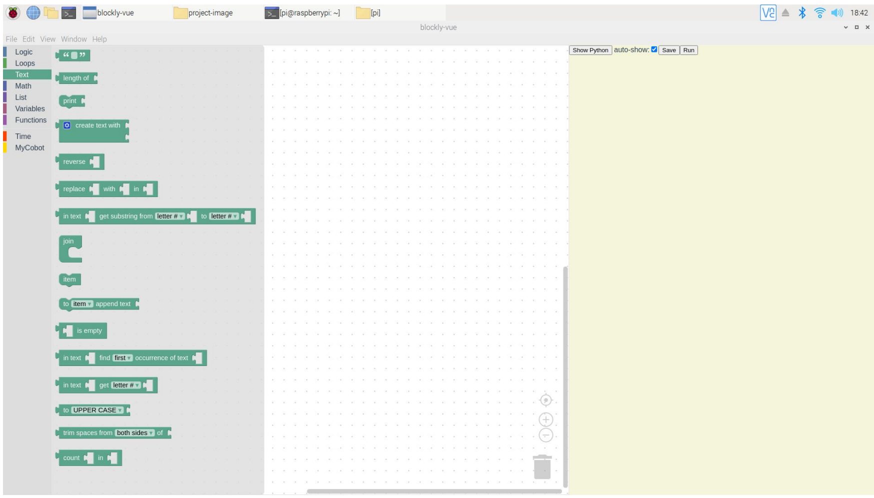
2-2 Loops模块方法详细

如图2-2所示，①表示重复执行10次do里面的程序。②表示重复变量num次do中的程序（do被遮挡）。点击②中

注：在循环中想使用循环中的变量需要设置一致的变量。

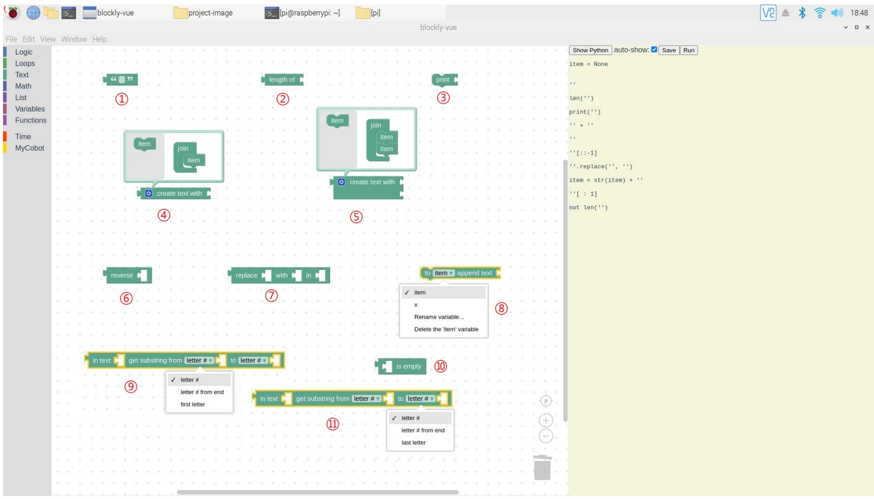
Text

1、如图3-1所示即为Text模块所包含的所有方法。



3-1 Text Module

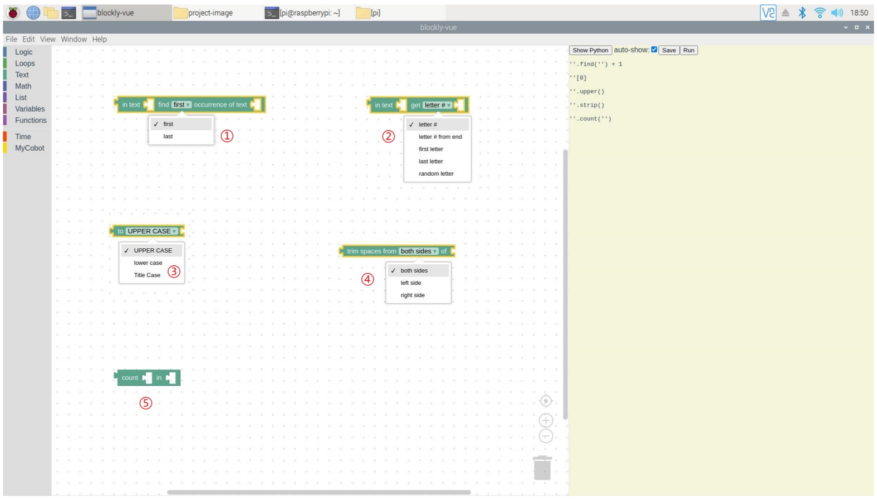
2、Method is explained in detail



```
item = None

''
len('')
print('')
'' + ''
''
''[::-1]
''.replace('', '')
item = str(item) + ''
''[ : 1]
not len('')
```

3-2 Text模块方法详细（一）

如图3-2所示，①表示文本内容，可以自定义文本内容。②表示计算指定文本内容的长度。③表示输出文本内容



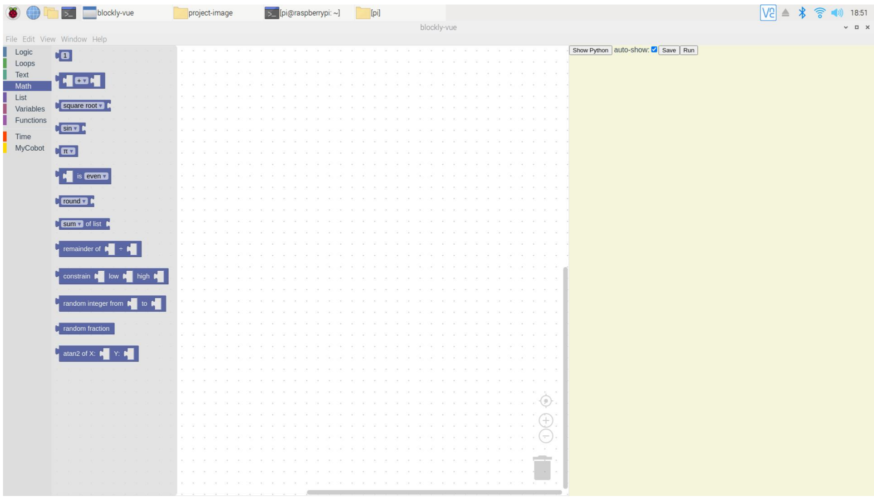
```
''.find('') + 1
''[0]
''.upper()
''.strip()
''.count('')
```

3-3 Text模块方法详细（二）

```
如图3-3所示，①表示指定字符串在选定字符串中第一次或最后一次出现的位置，可以点击下拉框选择是第一
```

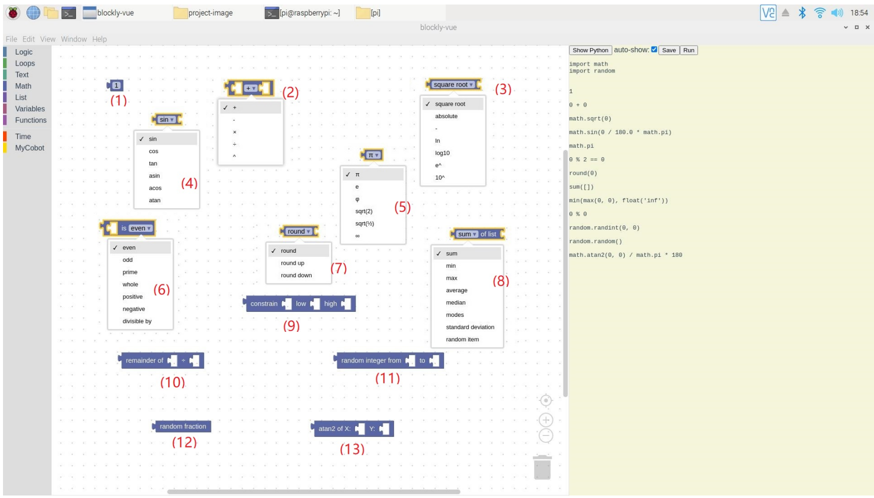
Math

1、如图4-1所示即为Math模块所包含的所有方法。



4-1 Math Module

2、Method is explained in detail



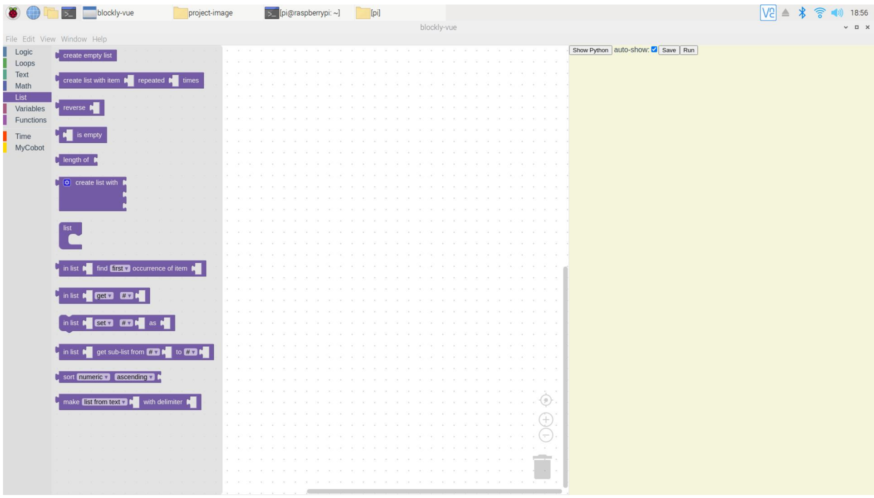
```
import math
import random


1

0 + 0

math.sqrt(0)

math.sin(0 / 180.0 * math.pi)

math.pi

0 % 2 == 0

round(0)

sum([])

min(max(0, 0), float('inf'))

0 % 0

random.randint(0, 0)

random.random()

math.atan2(0, 0) / math.pi * 180
```

4-2 Math模块方法详细

```
如图4-2所示，(1)表示数字常量，该数值常量是可以自定义的。(2)表示两个变量逻辑相加减等运算操作，
```

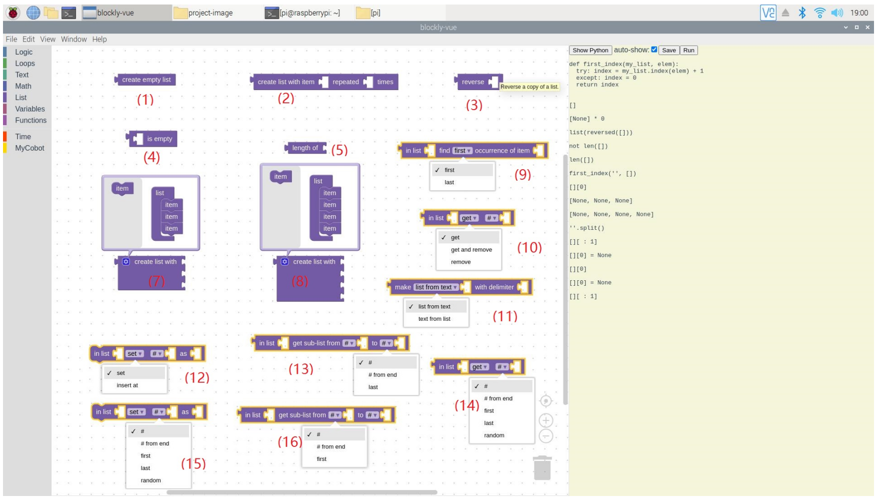
List

1、如图5-1所示即为List模块所包含的所有方法。



5-1 List Module

2、Method is explained in detail



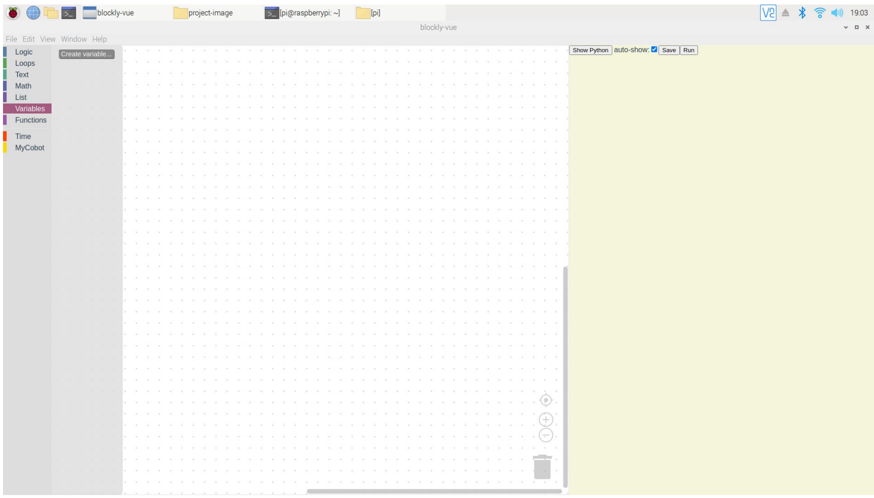
```
def first_index(my_list, elem):
  try: index = my_list.index(elem) + 1
  except: index = 0
  return index


[]

[None] * 0

list(reversed([]))

not len([])

len([])

first_index('', [])

[][0]

[None, None, None]

[None, None, None, None]

''.split()

[][ : 1]

[][0] = None

[][0]

[][0] = None

[][ : 1]
```

5-2 List模块方法详细

如图5-2所示，(1)表示创建一个空的list数组。(2)表示创建一个数组，该数组为指定一个数重复多少次后

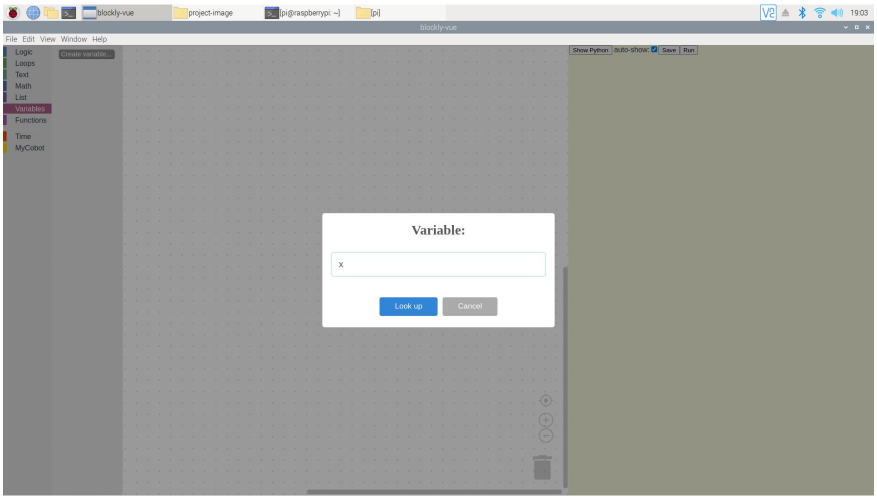
Variables

1、如图6-1所示即为Variables模块



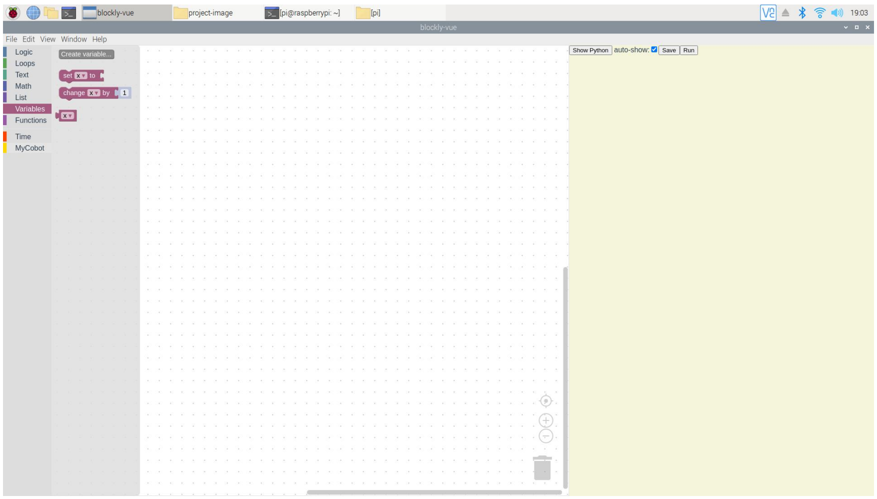
6-1 Variables模块

2、如图6-2所示，点击箭头所指处即可开始创建变量。



6-2 自定义变量名

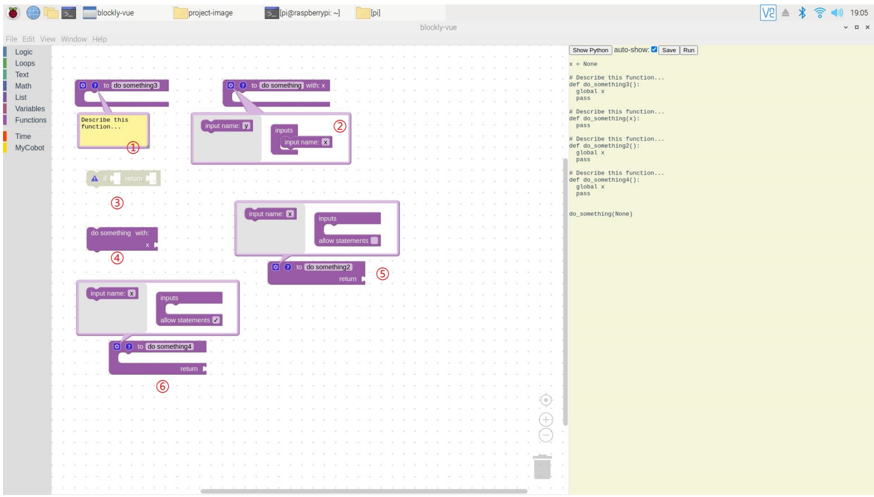
如图6-2所示，在输入框中输入自定义的变量名，点击Look up即可创建。



6-3 变量生成效果

如图6-3所示即为创建好后的变量。

Functions

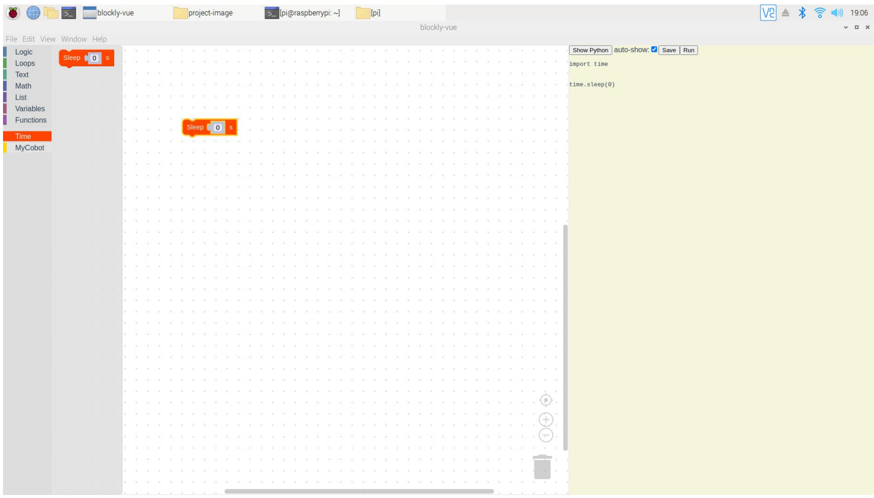


```
x = None

# Describe this function...
def do_something3():
  global x
  pass

# Describe this function...
def do_something(x):
  pass

# Describe this function...
def do_something2():
  global x
  pass

# Describe this function...
def do_something4():
  global x
  pass


do_something(None)
```

7-1 Functions Module

```
如图7-1所示，Functions模块包含两类函数，第一种如①所示是没有返回值的，第二种如⑥所示有返回值的。
```

Time

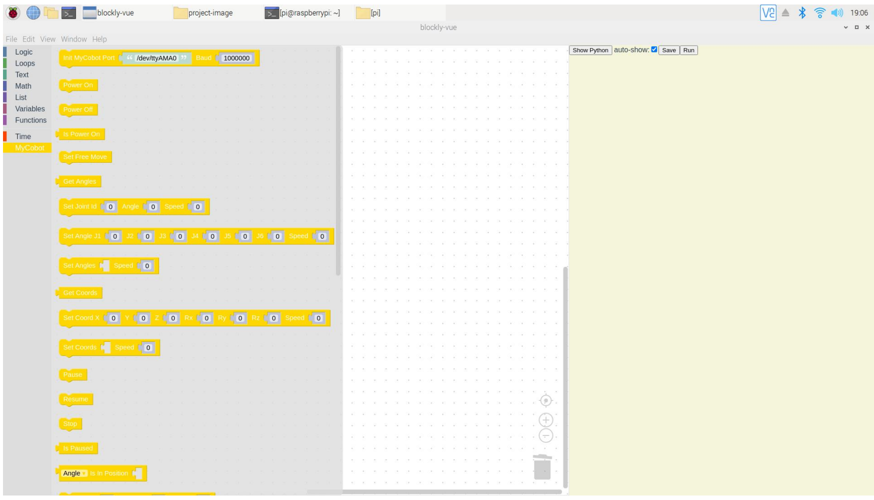


```
import time


time.sleep(0)
```

8-1 Time Module

```
As 8-1 indicates, sleep (time_num) represents waiting `time_num` seconds before execut
```

Mycobot



9-1 Mycobot Module

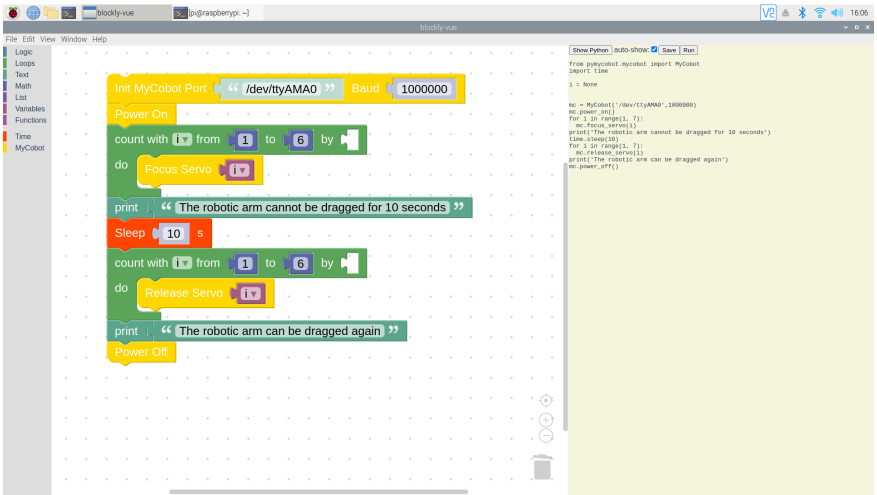
The method of use of this module, please refer to pymycobot。

Release and fixation of the robot arm

Case Introduction

In this case, by using loop call, the method of `focus servo` to fix 6 joint points， and use `time` method to fix it for 10 seconds. Finally use loop call `release` method to release 6 joint points.

二、 Content of demo



```
Init MyCobot Port " /dev/ttyAMA0 " Baud 1000000
Power On
count with i from 1 to 6 by
do  Focus Servo i
print " The robotic arm cannot be dragged for 10 seconds "
Sleep 10 s
count with i from 1 to 6 by
do  Release Servo i
print " The robotic arm can be dragged again "
Power Off
```

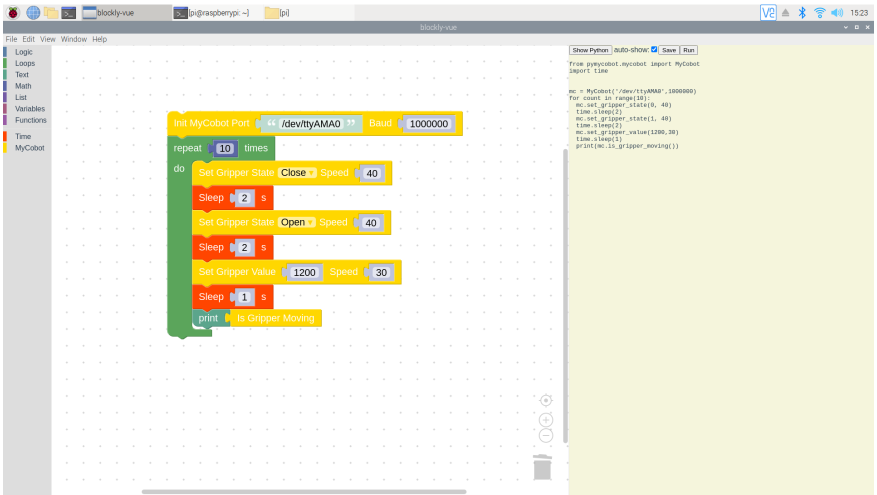
```
from pymycobot.mycobot import MyCobot
import time

i = None


mc = MyCobot('/dev/ttyAMA0',1000000)
mc.power_on()
for i in range(1, 7):
  mc.focus_servo(i)
print('The robotic arm cannot be dragged for 10 seconds')
time.sleep(10)
for i in range(1, 7):
  mc.release_servo(i)
print('The robotic arm can be dragged again')
mc.power_off()
```

Gripper detection of the robot arm

Case Introduction

By using `Set_Gripper_State` function of myCobot to let the gripper open and close 10 times, and adjust angle after each group close.

demo



blockly-vue
File Edit View Window Help
Logic
Loops
Text
Math
List
Variables
Functions
Time
MyCobot
Init MyCobot Port " /dev/ttyAMA0 " Baud 1000000
repeat 10 times
do
Set Gripper State Close Speed 40
Sleep 2 s
Set Gripper State Open Speed 40
Sleep 2 s
Set Gripper Value 1200 Speed 30
Sleep 1 s
print Is Gripper Moving
Show Python
auto-show: ☑
Save
Run

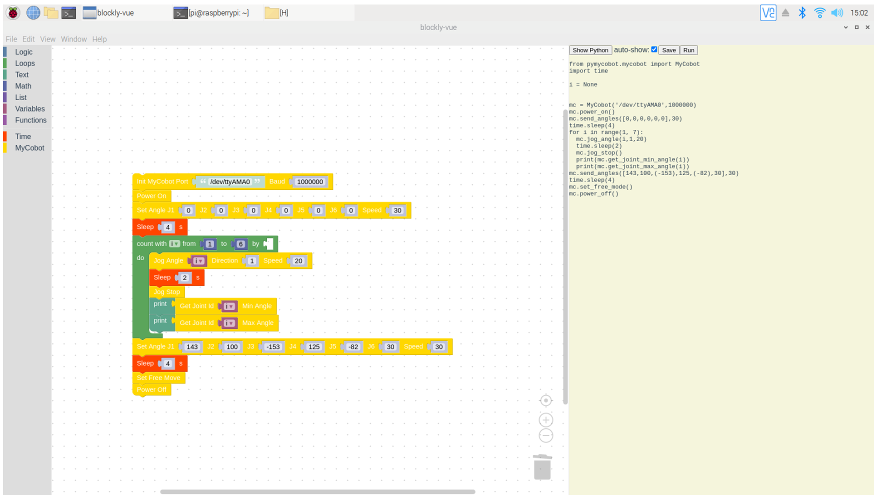
```
from pymycobot.mycobot import MyCobot
import time


mc = MyCobot('/dev/ttyAMA0',1000000)
for count in range(10):
  mc.set_gripper_state(0, 40)
  time.sleep(2)
  mc.set_gripper_state(1, 40)
  time.sleep(2)
  mc.set_gripper_value(1200,30)
  time.sleep(1)
  print(mc.is_gripper_moving())
```

Set the movement time of the robot arm

Case content

The main experimental content of this case is to call the `jog_angle` function to keep the six joints moving continuously through a loop. Stop its motion with the `jog_stop` function.Finally, the robot arm is moved to a safer position and the joint is released and powered off.

Content of program



```
Init MyCobot Port “ /dev/ttyAMA0 ” Baud 1000000
Power On
Set Angle J1 0 J2 0 J3 0 J4 0 J5 0 J6 0 Speed 30
Sleep 4 s
count with i from 1 to 6 by
do  Jog Angle i Direction 1 Speed 20
    Sleep 2 s
    Jog Stop
    print Get Joint Id i Min Angle
    print Get Joint Id i Max Angle
Set Angle J1 143 J2 100 J3 -153 J4 125 J5 -82 J6 30 Speed 30
Sleep 4 s
Set Free Move
Power Off
```

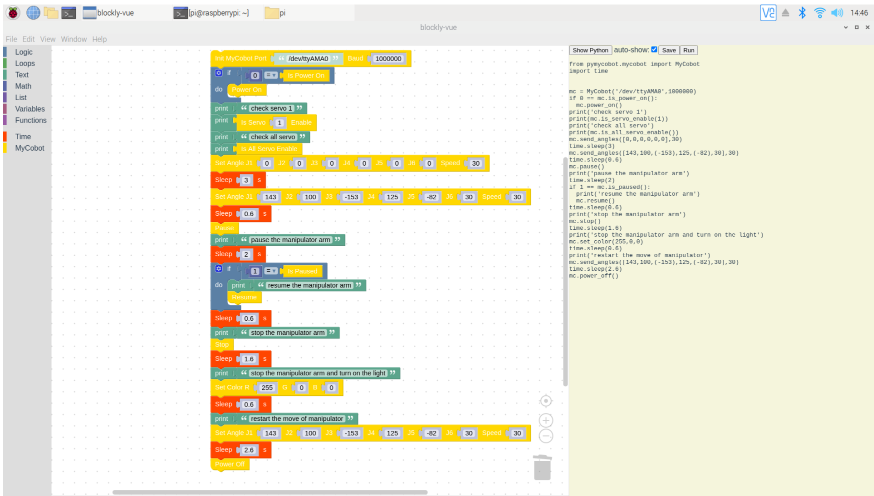
```python
from pymycobot.mycobot import MyCobot
import time

i = None


mc = MyCobot('/dev/ttyAMA0',1000000)
mc.power_on()
mc.send_angles([0,0,0,0,0,0],30)
time.sleep(4)
for i in range(1, 7):
  mc.jog_angle(i,1,20)
  time.sleep(2)
  mc.jog_stop()
  print(mc.get_joint_min_angle(i))
  print(mc.get_joint_max_angle(i))
mc.send_angles([143,100,(-153),125,(-82),30],30)
time.sleep(4)
mc.set_free_mode()
mc.power_off()
```

Control mechanism of the robot arm

Case content

This case mainly calls some of the commonly used control mechanism functions of the robot arm to control the robot arm, such as power off the robot arm, power supply, suspension of motion, restore movement and other control mechanism functions, as well as the control of the headlights of the robot arm.

Demo



```
from pymycobot.mycobot import MyCobot
import time


mc = MyCobot('/dev/ttyAMA0',1000000)
if 0 == mc.is_power_on():
  mc.power_on()
print('check servo 1')
print(mc.is_servo_enable(1))
print('check all servo')
print(mc.is_all_servo_enable())
mc.send_angles([0,0,0,0,0,0],30)
time.sleep(3)
mc.send_angles([143,100,(-153),125,(-82),30],30)
time.sleep(0.6)
mc.pause()
print('pause the manipulator arm')
time.sleep(2)
if 1 == mc.is_paused():
  print('resume the manipulator arm')
  mc.resume()
time.sleep(0.6)
print('stop the manipulator arm')
mc.stop()
time.sleep(1.6)
print('stop the manipulator arm and turn on the light')
mc.set_color(255,0,0)
time.sleep(0.6)
print('restart the move of manipulator')
mc.send_angles([143,100,(-153),125,(-82),30],30)
time.sleep(2.6)
mc.power_off()
```

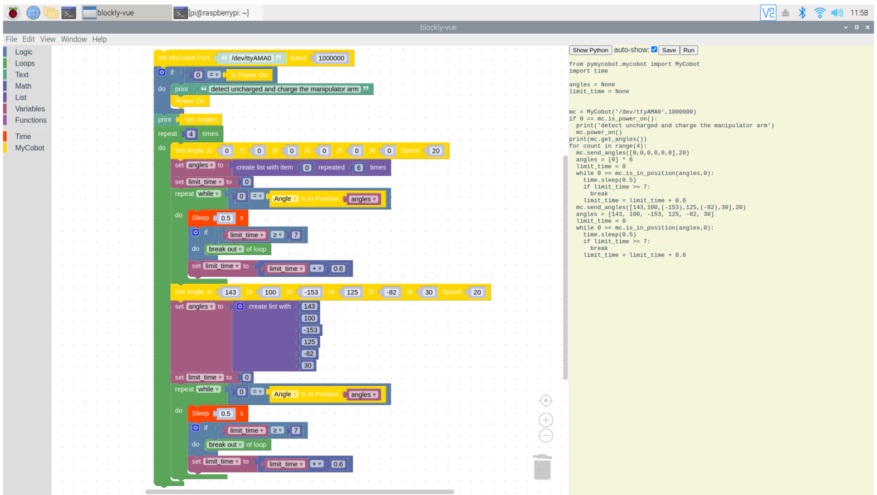
Adavanced operation of the robot arm

Case Introduction

Mainly realize that the robot arm intelligent judge the function that the robot arm has already arrived the specified position, based on this function, simply let the robot arm repeat two arrival instructions.

- Firstly use `if do` module to judge if the robot arm is powered, if not, you should power the robot arm. Output the current angle node information.
- Transfer the robot arm to zero. Define `angles` variable. Use the `repeated` method in the creation of `list` type data to assign zero node information to `angles` . Define `limit_time` to determine if the movement times out.
- Pass `angles` into the `is in position` method to determine whether the robot arm has reached the specified position. Use the `repeat` module 0.5s per movement to detect whether the robot arm has reached the specified position and time out the timer, and if more than 7s the robot arm has not yet reached the specified position, determine that it has arrived and execute the next instructions.
- The principle is similiar, so no more explanation

demo

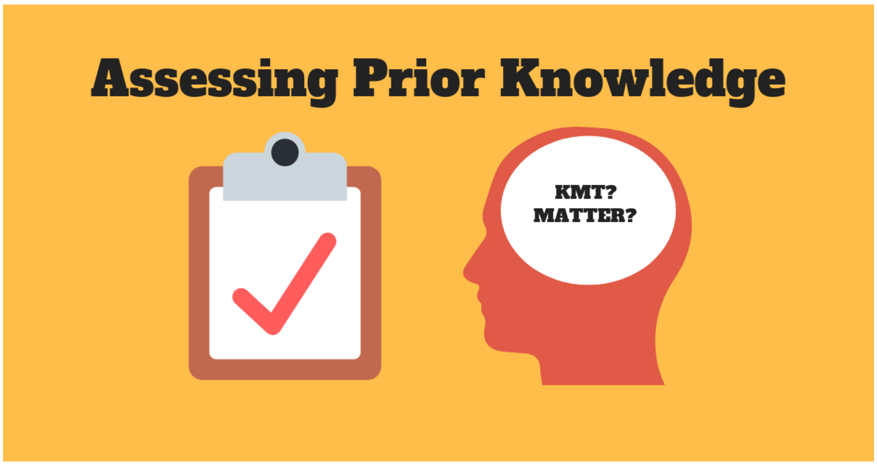


```
from pymycobot.mycobot import MyCobot
import time

angles = None
limit_time = None


mc = MyCobot('/dev/ttyAMA0',1000000)
if 0 == mc.is_power_on():
  print('detect uncharged and charge the manipulator arm')
  mc.power_on()
print(mc.get_angles())
for count in range(4):
  mc.send_angles([0,0,0,0,0,0],20)
  angles = [0] * 6
  limit_time = 0
  while 0 == mc.is_in_position(angles,0):
    time.sleep(0.5)
    if limit_time >= 7:
      break
    limit_time = limit_time + 0.6
  mc.send_angles([143,100,(-153),125,(-82),30],20)
  angles = [143, 100, -153, 125, -82, 30]
  limit_time = 0
  while 0 == mc.is_in_position(angles,0):
    time.sleep(0.5)
    if limit_time >= 7:
      break
    limit_time = limit_time + 0.6
```

When you use the Raspberry Pi version of mycobot, you should already have a Raspberry Pi system equipped with ROS Kinetic.

For detailed usage, please refer to the ROS part of Chapter 3

1 Background



It is necessary to have an in-depth understanding of myCobot from the aspects of hardware, software and robot algorithm. At the beginning, we can understand the progress of robots by understanding the history of robots.

- Robot
- Software
- Electronics
- Mechanics
- Motor

1.1 Robot

This chapter is excerpted from Introduction to robotics mechanics and control by J.Craig. If you want to read more about it, please buy it online.

1 Background

The history of industrial automation is characterized by the rapid renewal of technological means. The renewal of such automation technology is closely related to the world economy, whether as an inducement or a result of the development of the world economy. Industrial robot in the 1960s is undoubtedly a unique equipment, it will be combined with the computer aided design (CAD) system, computer aided manufacturing (CAM) system application, this is the modern manufacturing automation of the latest development trend. These technologies are leading the transition to a new field of industrial automation.

Manipulator is one of the most important types of industrial robots. Whether the manipulator can be called an industrial robot is controversial. The equipment shown here is generally considered to belong to the category of industrial robots, while CNC (NC) grinders are usually outside this category.

Generally speaking, the research of manipulator mechanism and control theory is not a new science, it is only a synthesis of traditional theory.Mechanical engineering theory provides a methodology for the study of manipulator in static and dynamic environments.The mathematical method is used to describe the spatial motion of the manipulator and its characteristics.Control theory provides various design methods and evaluation algorithms for the realization of desired motion or force.Electrical engineering technology can be used in sensor and industrial robot interface design;Computer technology provides the programming platform needed to perform the desired task.

2 Basic Concepts

Mechanical Arm

Mechanical Arm can also be called industrial robot, cooperative robot, manipulator arm, bionic arm, series robot, etc.

Position & Pose

In robot research, we usually study the position of objects in a three-dimensional space. The objects referred to here include not only the lever, parts and grasping tools of the manipulator, but also other objects in the workspace of the manipulator. Usually these objects can be described by two very important properties: position and pose. Naturally, we will first study how to express and calculate these parameters mathematically.

In order to describe the position and posture of a space object, we usually place the object firmly in a space coordinate system, that is, the reference frame, and then we study the position and posture of the space object in this reference coordinate system.

Direct Kinematics

Kinematics is the study of the motion of objects without regard to the forces causing such motion. In kinematics, we study higher-order derivatives of position, velocity, acceleration, and position variables with respect to time or other variables. Thus, the research object of manipulator kinematics is all the geometric and temporal characteristics of motion. Almost all manipulators are composed of rigid links, adjacent links connected by joints that allow for relative motion. If it's a revolute joint, its displacement is called the joint Angle. These joints are usually fitted with position sensors to measure the relative position of adjacent bars. If you have a revolute joint, this displacement is called the joint Angle. Some manipulators have sliding (or moving) joints, so the displacement of two adjacent links is a linear motion, which sometimes called the joint offset.

A typical problem in the study of manipulator kinematics is manipulator forward kinematics. It is a static geometry problem to calculate the position and posture of its end-effector of the manipulator. Specifically, given a set of values for joint angles, the forward kinematics problem is to compute the position and posture of the tool coordinate system relative to the base coordinate system. In general, we refer to this process as the representation of manipulator position from joint space description to Cartesian space description.

Degree of Freedom (DOF)

The number of DOF is the number of manipulator position variables in the coordinate system (reference frame) with figure 1-5 in the manipulator, which determines the position of all components in the mechanism. DOF is universal to all mechanisms. For example, a four-bar mechanism has only one DOF (although it has three movable rods). For a typical industrial robot, the number of joints is equal to the number of DOF because manipulator arms are mostly open motion chains and each joint position is defined by an independent variable.

End-effector

End-effector is installed at the free end of the manipulator. Depending on different applications of robot, it may be a fixture, a welding torch, an electromagnet, or some other devices. We usually describe the position of the manipulator in terms of a tool coordinate system attached to its end-effector, and the corresponding tool coordinate system is the base coordinate system connected to the fixed base of the manipulator.

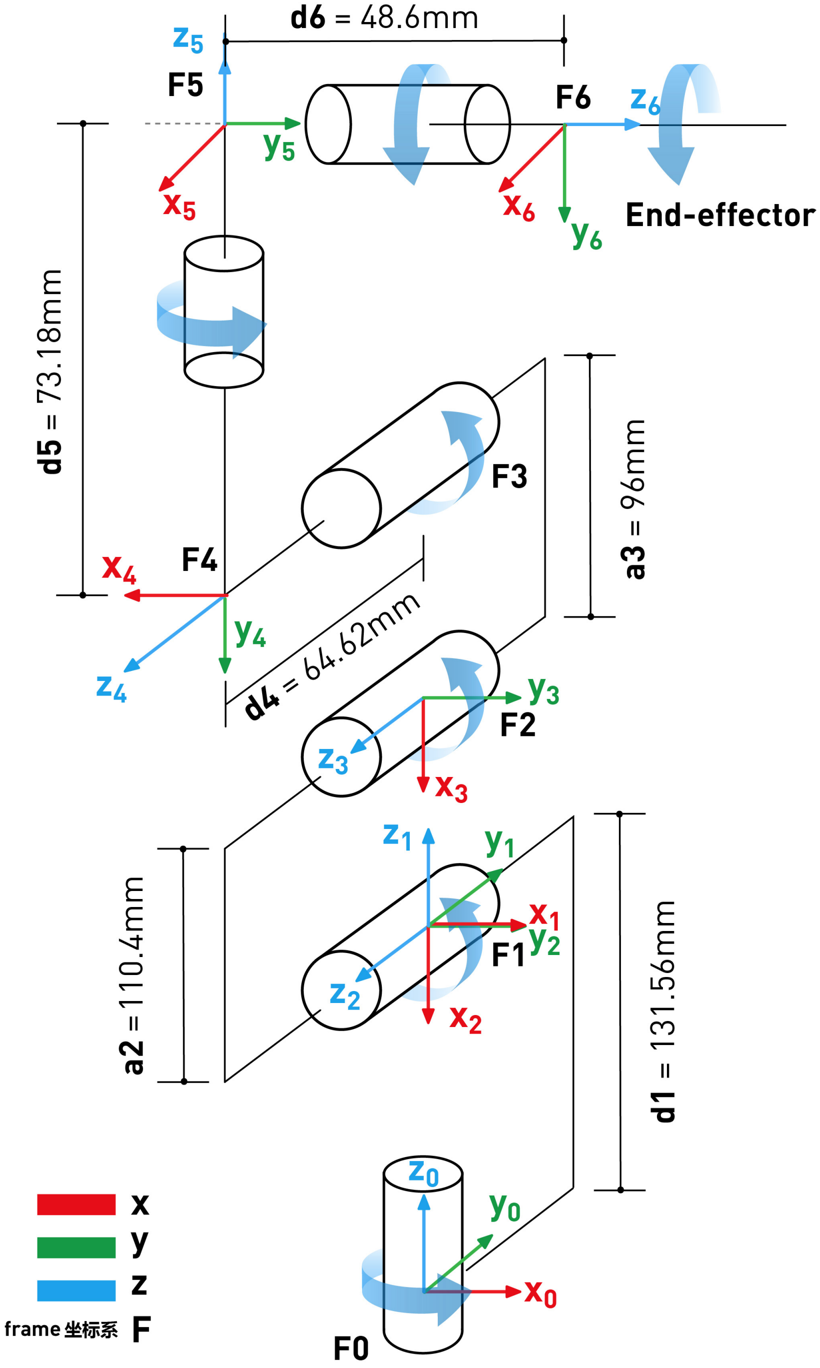
Inverse Kinematic

Given the position and posture of the end-effector of the manipulator, calculating all joint angles that can reach the given position and attitude.

3 Space description

Position

Once the coordinate system is established, we can locate any point in the world coordinate system with a position vector of 3x1. Since many coordinates areoften defined in the world coordinate system, a piece of information must beattached to the position vector indicating which coordinate system is defined.In this book, the position vector has a leading superscript to indicate thecoordinate system to which it refers.



Posture

We find that it is often necessary not only to represent points in space, but also to describe the posture of objects in space. For example, if the vector "P" in Figure 2-2 directly determines a point between the fingers of the manipulator hand, the position of the hand can only be fully determined if the posture of the hand is known. Assuming that the manipulator has a sufficient number of joints, the manipulator can be in any position and the position of the points between the fingers remains constant. To describe the posture of an object, we will fix a coordinate system on the object and give the representation of this coordinate system with respect to the reference system. In Figure 2-2, the coordinate system {B} is known to be fixed to the object in some way. The description in {B} relative to {A} is sufficient to indicate the attitude of object (A).

Coordinate System

A reference frame can be described in terms of the relation of one coordinate system with respect to another. The reference frame includes the concepts of position and posture, which is considered to be a combination of these two concepts in most cases. The position can be represented by a frame of reference in which the rotation matrix is the identity matrix and the position vector in this frame of reference determines the position of the described point. Similarly, if the position vector in the frame of reference is the zero vector, then it represents the posture.

4 DH Parameters

Definition

For rotational joint n, set 0=0.0, the direction of X axis is the same as that of X, axis, the origin position of coordinate system (n) is selected to satisfy d.=0.0.For prismatic joint n, the direction of axis 8 is set to meet 0.=0.0. When d.=0.0, the origin of the coordinate system {n) is selected to be located at the intersection of axis XN-1 and joint axis n.

In the link coordinate system, if the link coordinate system is fixedly attached to the link as described above, the link parameters can be defined as follows:

- a_i-1 ： along x_i-1 : move from the distance of z_i-1 to z_i
- alpha_i-1 ： aroundx_i-1 ： rotate from the Angle of z_i-1 to z_i
- d_i ： along z_i ： move from the distance of x_i-1 to x_i
- theta_i ： aroundz_i ： rotate from the Angle of x_i-1 tox_i

myCobot DH parameter

Joint	alpha	a	d	theta	offset
1	0	0	131.56	theta_1	0
2	PI/2	0	0	theta_2	-PI/2
3	0	-110.4	0	theta_3	0
4	0	-96	64.62	theta_4	-PI/2
5	PI/2	0	73.18	theta_5	PI/2
6	-PI/2	0	48.6	theta_6	0

1.2 Software

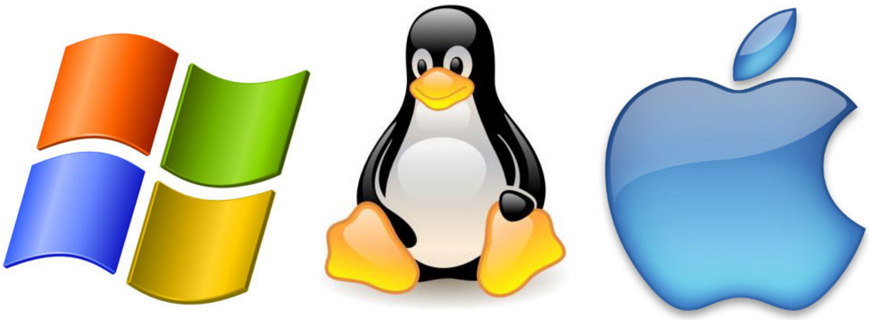
For myCobot users, software operation requires basic C/C++ knowledge to drive microcontrollers Basic and Atom.

Meanwhile, the related software is mainly required by GitHub and Arduino

- **Github**: download the latest myCobot code and update the related users instructions
- **Arduino**: IDE (integration development environment) of the core development of myCobot need to program in C/C++

You can also directly use Python, ROS, UIFlow, and other development tools, which are described in the chapter of Development and Use .

If you want to learn programming languages, you can learn from books, open classes, online videos, etc.. And your development platforms can be Windows, MacOS, or Linux.

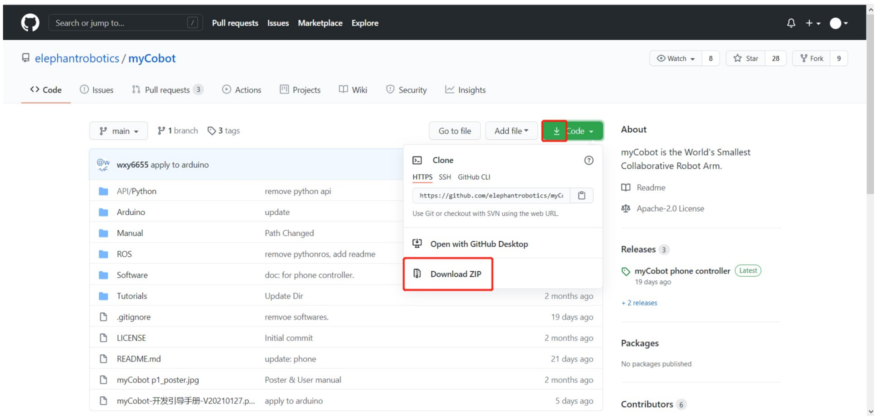


Github

GitHub is a hosting platform for open source and private software projects, it houses our software (Python, ROS, Arduino), APP, industrial visual programming software - RoboFlow, firmware, user manual and development guide manual.

Github：https://github.com/elephantrobotics/myCobot

Please click to download as shown in the figure



wxy6655 apply to arduino

API/Python	remove python api	
Arduino	update	
Manual	Path Changed	
ROS	remove pythonros, add readme	
Software	doc: for phone controller.	
Tutorials	Update Dir	2 months ago
.gitignore	remvoe softwares.	19 days ago
LICENSE	Initial commit	2 months ago
README.md	update: phone	21 days ago
myCobot p1_poster.jpg	Poster & User manual	2 months ago
myCobot-开发引导手册-V20210127.p...	apply to arduino	5 days ago

```
https://github.com/elephantrobotics/myC
```

Use Git or checkout with SVN using the web URL.

myCobot is the World's Smallest Collaborative Robot Arm.

Arduino

Arduino

Arduino is an open source electronic prototyping platform that is convenient, flexible and easy to use, contains hardware (various types of Arduino boards) and software (Arduino IDE). The hardware part (or development board) consists of a microcontroller (MCU), flash memory (FLASH), and a set of general input/output interfaces (GPIO), etc., you can think of it as a microcomputer motherboard. The software part is mainly composed of Arduino IDE on PC, related Board Support

Package (BSP) and abundant third-party function library. Users can easily download the BSP related to the development board you own and the library of functions by Arduino IDE to write your programs.

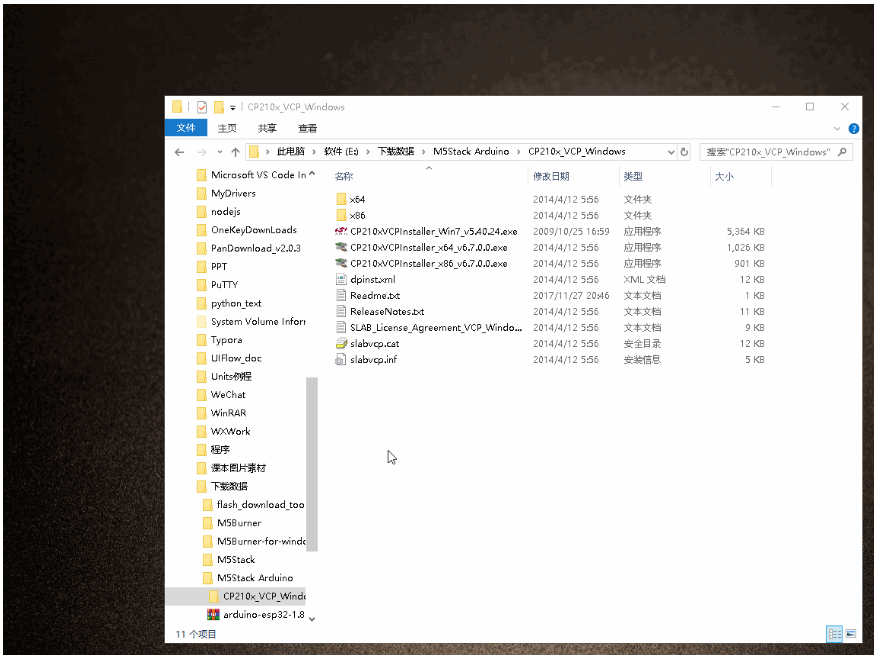
How to install Arduino IDE？

- Driver Installation

> M5Core host (including BASIC/GRAY/M5GO/FIRE/FACES). Before burning program, please click the button below to download the corresponding CP210X driver zip package according to the operating system you are using. After unpacking the package, select the installation package corresponding to the operating system number to install.

Download CP2104 driver

- Windows10
- MacOS
- Linux

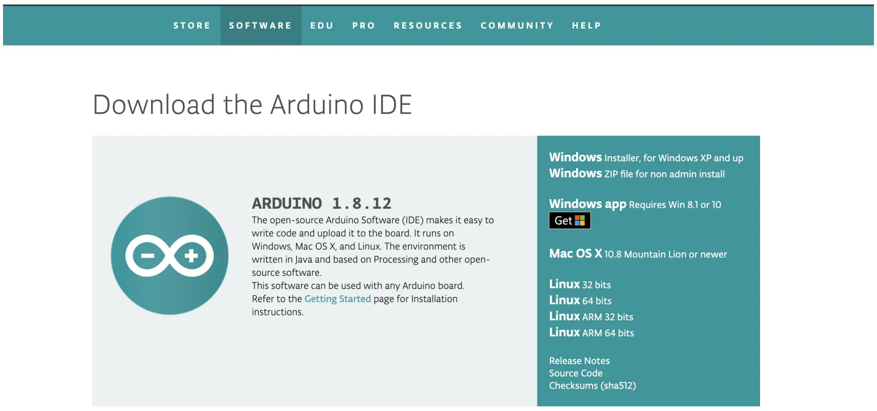


名称	修改日期	类型	大小
x64	2014/4/12 5:56	文件夹	
x86	2014/4/12 5:56	文件夹	
CP210xVCPInstaller_Win7_v5.40.24.exe	2009/10/25 16:59	应用程序	5,364 KB
CP210xVCPInstaller_x64_v6.7.0.0.exe	2014/4/12 5:56	应用程序	1,026 KB
CP210xVCPInstaller_x86_v6.7.0.0.exe	2014/4/12 5:56	应用程序	901 KB
dpinst.xml	2014/4/12 5:56	XML 文档	12 KB
Readme.txt	2017/11/27 20:46	文本文档	1 KB
ReleaseNotes.txt	2014/4/12 5:56	文本文档	11 KB
SLAB_License_Agreement_VCP_Windo...	2014/4/12 5:56	文本文档	9 KB
slabvcp.cat	2014/4/12 5:56	安全目录	12 KB
slabvcp.inf	2014/4/12 5:56	安装信息	5 KB

Arduino IDE

如果需要下载Arduino IDE 可以点击Aeduino 官网下载安装与电脑系统对应的版本。

If you need to download Arduino IDE, you can click Arduinoto download and install the version corresponding to your computer system.



Download the Arduino IDE

ARDUINO 1.8.12

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
This software can be used with any Arduino board.
Refer to the Getting Started page for Installation instructions.

Windows Installer, for Windows XP and up
Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10
Get

Mac OS X 10.8 Mountain Lion or newer

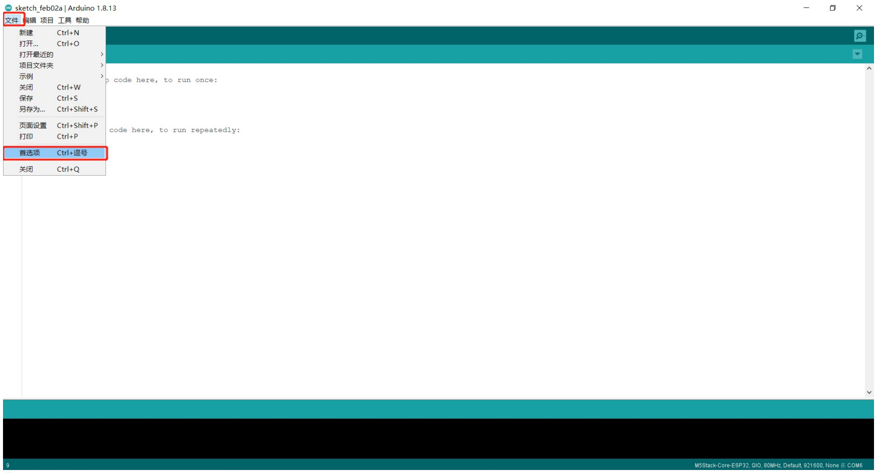
Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

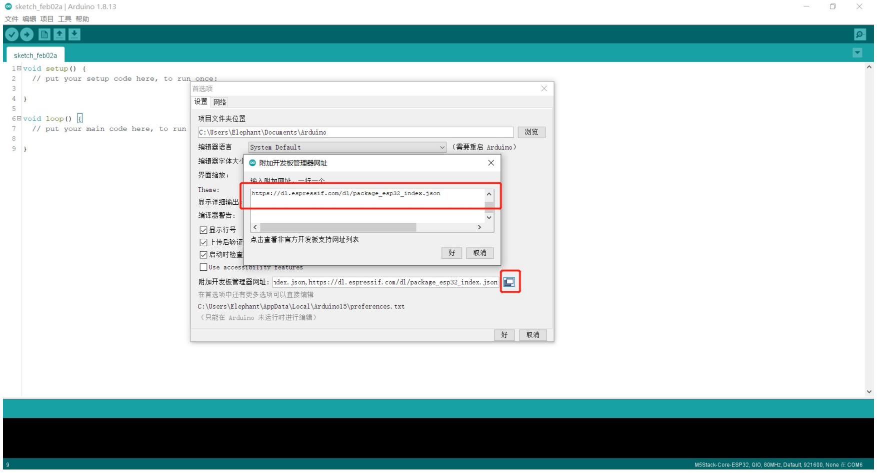
Release Notes
Source Code
Checksums (sha512)

Configure Arduino development environment required by myCobot

1. Add the required development board

- Open Arduino IDE, select *File -> Preferences ->setting* Then copy the URL below to link to the attached development board manager https://dl.espressif.com/dl/package_esp32_index.json





```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run
}
```

首选项

设置 网络

项目文件夹位置 C:\Users\Elephant\Documents\Arduino 浏览

编辑器语言 System Default （需要重启 Arduino）

编辑器字体大小

界面缩放：

Theme:

显示详细输出

编译器警告：

☑ 显示行号

☑ 上传后验证

☑ 启动时检查

□ Use accessibility features

附加开发板管理器网址： ıdex.json,https://dl.espressif.com/dl/package_esp32_index.json

在首选项中还有更多选项可以直接编辑

C:\Users\Elephant\AppData\Local\Arduino15\preferences.txt

（只能在 Arduino 未运行时进行编辑）

好 取消

附加开发板管理器网址

输入附加网址，一行一个

https://dl.espressif.com/dl/package_esp32_index.json

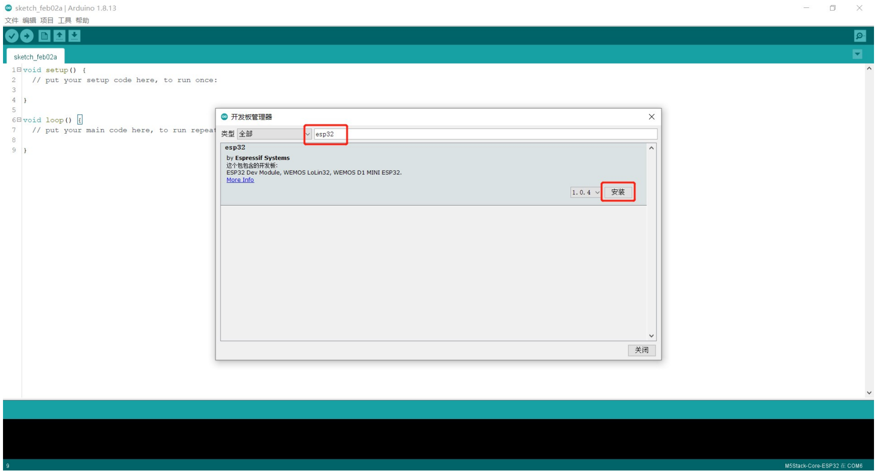
点击查看非官方开发板支持网址列表

好 取消

- Select *Tools -> Development Board: -> Development Board Manager*

In the new dialog box that pops up, enter and search *ESP32* , then click *Install*

(If the search fails, try restarting Arduino as below)



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeat
}
```

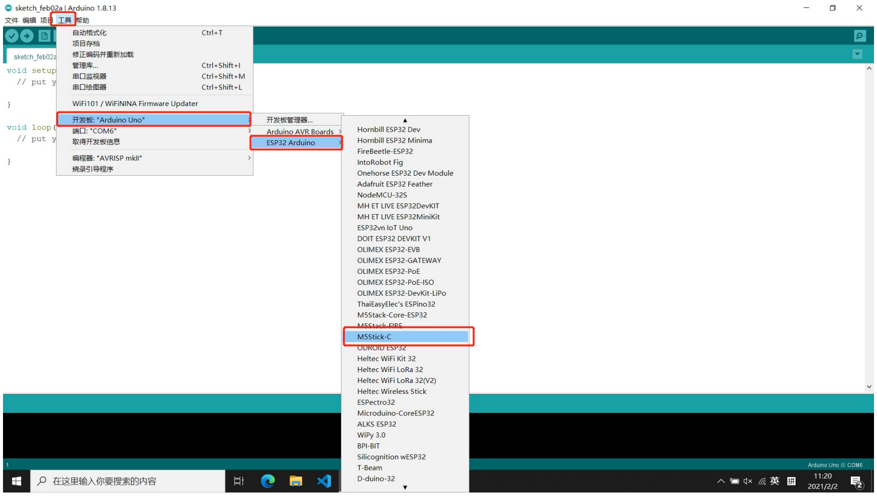
开发板管理器

类型 全部 esp32

esp32
by **Espressif Systems**
这个包包含的开发板：
ESP32 Dev Module, WEMOS LoLin32, WEMOS D1 MINI ESP32.
More Info

1.0.4 安装

- After the installation, select *Tools -> Development Board:* to check whether it was successful As shown in the figure below:



```
void setup
  // put y

}

void loop(
  // put y

}
```

2. Add Libraries

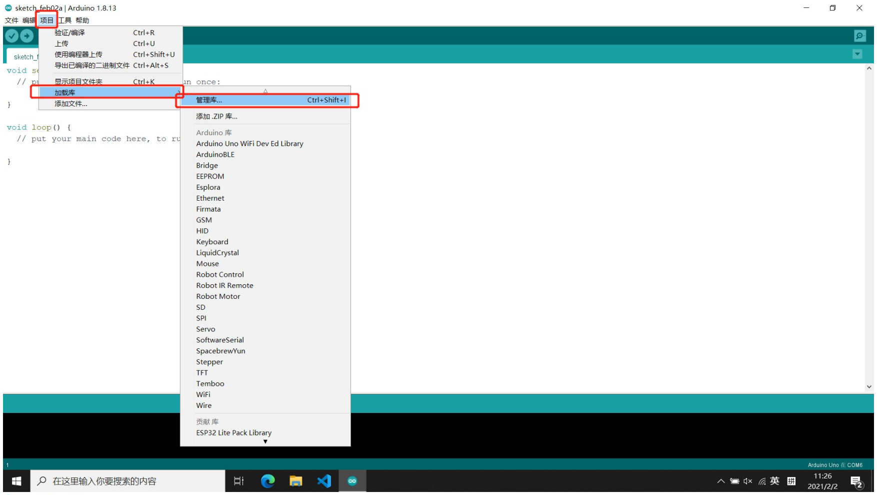
**Different hardware devices have different case libraries, please choose to

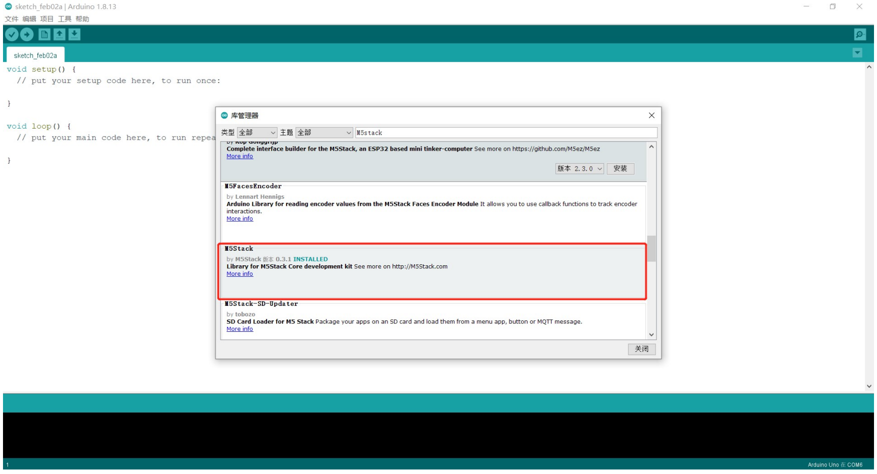
> download it according to the device you are using.**

- You can use project libraries added by IDE.

Open Arduino IDE, select *Project -> Load Library -> Management Library...*

> Search and install the following project libraries separately: M5Stack、M5Core2、M5Atom、ESP32_Lite_Pack_Library, etc.. Steps are as follows:





```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repea
```

M5stack

Complete interface builder for the M5Stack, an ESP32 based mini tinker-computer See more on https://github.com/M5ez/M5ez
More info

M5FacesEncoder
by Lennart Hennigs
Arduino Library for reading encoder values from the M5Stack Faces Encoder Module It allows you to use callback functions to track encoder interactions.
More info

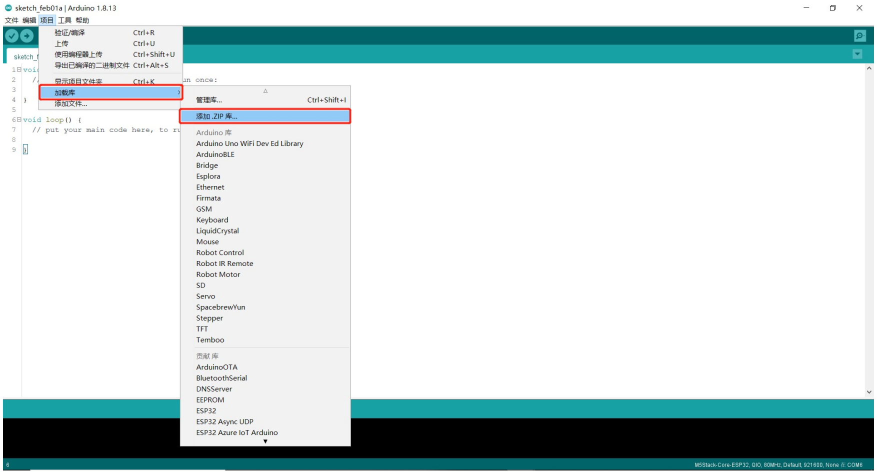
M5Stack
by M5Stack 版本 0.3.1 INSTALLED
Library for M5Stack Core development kit See more on http://M5Stack.com
More info

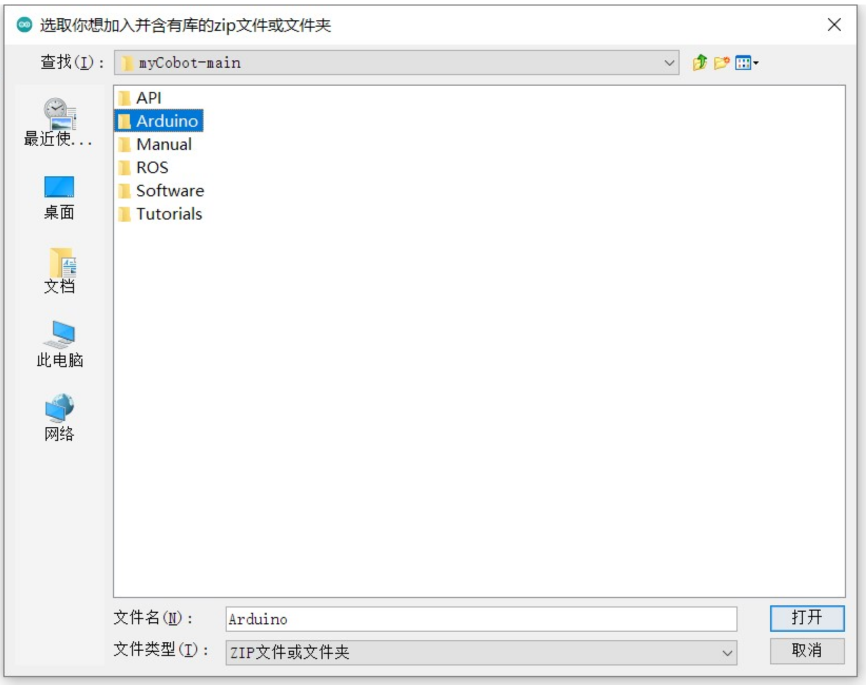
M5Stack-SD-Updater
by tobozo
SD Card Loader for M5 Stack Package your apps on an SD card and load them from a menu app, button or MQTT message.
More info

- Project libraries that need to be downloaded from GitHub, such as MycobotBasic

Download link: https://github.com/elephantrobotics/myCobot

After downloading, unzip and add it according to the index below





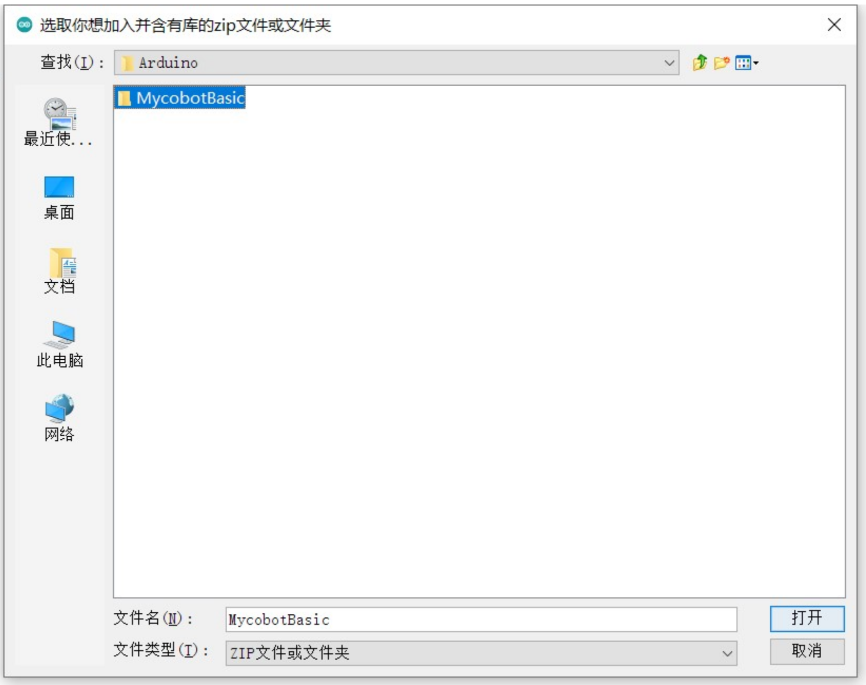
选取你想加入并含有库的zip文件或文件夹

查找(I): myCobot-main

API
Arduino
Manual
ROS
Software
Tutorials

文件名(N): Arduino

文件类型(T): ZIP文件或文件夹



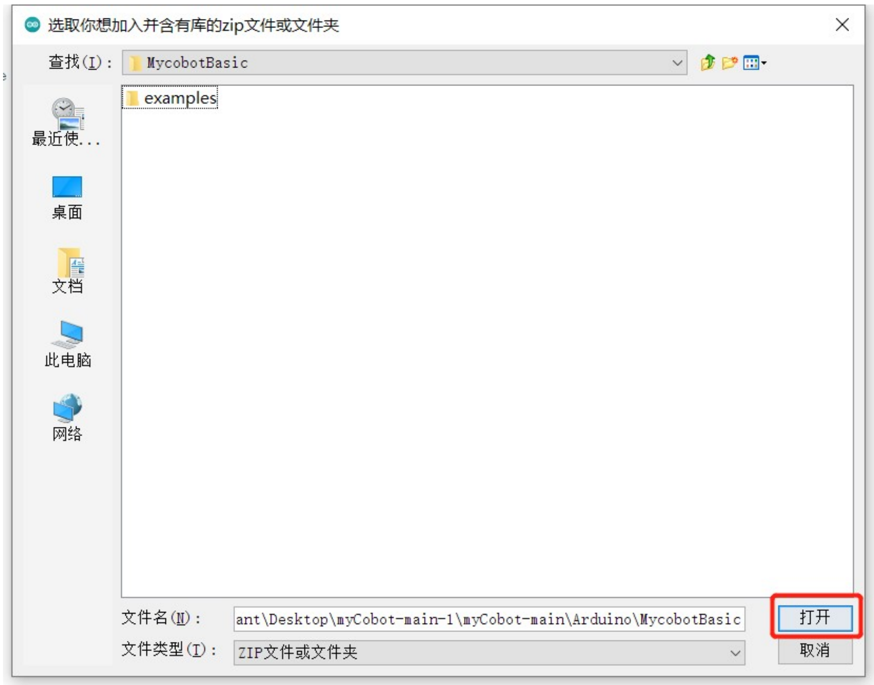
选取你想加入并含有库的zip文件或文件夹

查找(I): Arduino

MycobotBasic

文件名(N): MycobotBasic

文件类型(T): ZIP文件或文件夹



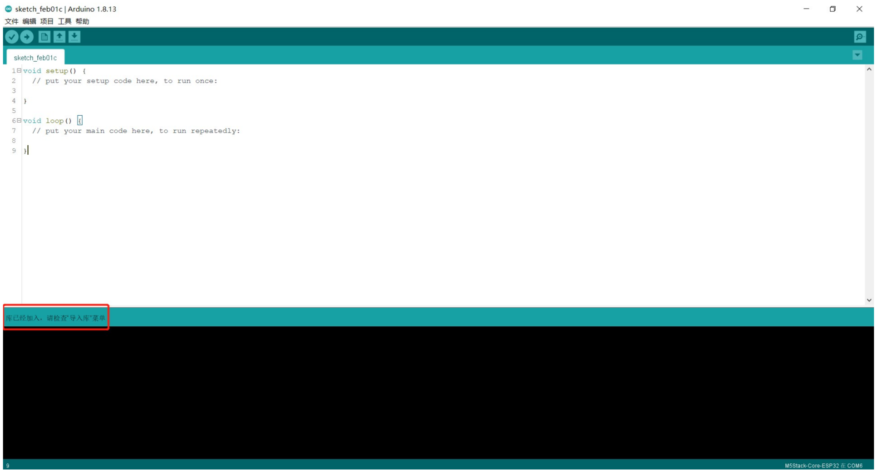
选取你想加入并含有库的zip文件或文件夹

查找(I)： MycobotBasic

examples

文件名(N)： ant\Desktop\myCobot-main-1\myCobot-main\Arduino\MycobotBasic

文件类型(T)： ZIP文件或文件夹



sketch_feb01c | Arduino 1.8.13

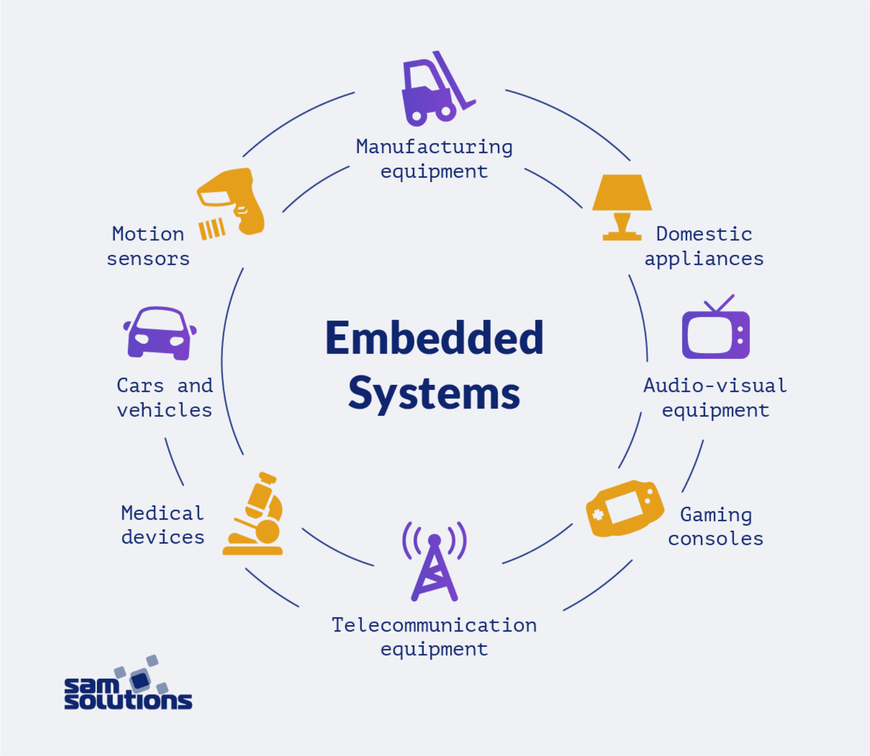
```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

库已经加入，请检查"导入库"菜单

M5Stack-Core-ESP32 在 COM6

Click open and "The library has been added. Please check the 'Import Library' menu.” is displayed in the lower right, then the environment configuration of Arduino is complete.

Electronics



Introduction

Single-Chip Microcomputer (Microcontrollers/SCM) is a kind of integrated circuit chip, which uses VLSI technology to integrate the central processing unit CPU with data processing capabilities, random access memory RAM, read-only memory ROM, various I/O ports, interrupt systems, and timer /Counter and other functions (may also include display drive circuit, pulse width modulation circuit, analog multiplexer, A/D converter and other circuits) integrated into a silicon chip to form a small and complete microcomputer system, widely used in the field of industrial control. From the 1980s, it developed from the 4-bit, 8-bit microcontroller to the current 300M high-speed microcontroller.

Basic Structure

- **Arithmetic Unit**

Arithmetic Unit is composed of arithmetic logic unit (ALU), accumulator and register. The function of ALU is to perform arithmetic or logical operations on the incoming data. The input comes from two 8-bit data sources, one from the accumulator and the other from the data register. ALU can perform various operations on the data, such as Addition, Subtraction, AND, OR, Comparison, etc., and finally store the results in the accumulator.

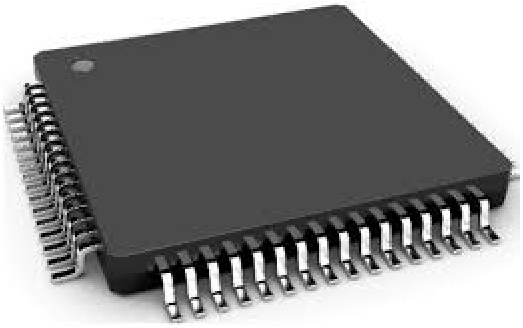
The arithmetic machine has two functions:

- Perform various arithmetic operations.
- Perform various logical operations and perform logical tests, such as a zero-value test or a comparison of two values.
- All operations performed by arithmetic unit are directed by the control signal issued by the controller, and an arithmetic operation produces an operation, and a logical operation produces a decision.
- **Controller**

Controller is composed of program counter, instruction register, instruction decoder, timing generator and operation controller, etc. It is the "decision-making body" of issuing commands, namely coordinating and directing the operation of the entire microcomputer system. Its main functions are：

- Extract an instruction from memory and indicates the location of the next instruction in memory.
- Decode and test the instructions,then generate the corresponding operation control signal to facilitate the execution of the specified actions.
- Direct and control the direction of data flow between CPU, memory, and input/output devices.

Microprocessor interconnects ALU, counter, register and control parts through an internal bus, and connects the external memory, input and output interface circuit through an external bus. External bus, also known as system bus, is divided into data bus DB, address bus AB and control bus CB. It can be connected with various peripheral devices through the input and output interface circuit.



1.4 Mechanics Background

Is being developed

Motor & Steering Gear

Motor

According to the type of working power supply, it can be divided into:

- 1). DC(Direct Current) motor
- 1.1). Brushless DC motor
    - 1.2). Brushed DC motor
- 1.2.1). Permanent magnet DC motor
- 1.2.1.1). Rare Earth permanent magnet DC motor
- 1.2.1.2). DC Ferrite permanent magnet DC motor
- 1.2.1.3). Aluminium Nickel-cobalt permanent magnet DC motor
- 1.2.2). Electromagnetic DC motor
- 1.2.2.1). DC series motor
- 1.2.2.2). Shunt DC motor
- 1.2.2.3). Separately Excited DC motor
- 2). AC (Alternating Current) motor
- 2.1). Single-phase motor
- 2.2). Three-phase motor

According to the application, it can be divided into:

- **Drive motor:** Electric tools (including drilling, buffing, polishing, grooving, cutting, reaming and other tools) motor, household appliances(including washing machine, electric fan, refrigerator, air conditioner, tape recorder, video recorder, DCD, vacuum cleaner, camera, hair dryer, electric shaver, etc.) motor, and other general small mechanical equipment (including all kinds of small machine tools, small machinery, medical equipment, electronic instruments, etc.) motor
- **Control motor:**
- Stepping motor
- Servo motor

Servo Motor

The steering gear is actually a servo motor, just like the rudder shaft control in the model aircraft and other equipment, so these lightweight servo motor is called steering gear.

Servo Motor

Servo Motor refers to the engine that controls the operation of mechanical components in the servo system, it is a kind of indirect speed change device for auxiliary motor.

Hardware of myCobot

The hardware of myCobot is composed of **electronic parts** and **mechanical parts**. Electronic parts include PCBA, controller, steering gear, charger, etc.. Structural parts include solid plastic casing, LEGO-mounted ports, flanges, fastener bearings, etc.

Basic & Atom

The microcontroller of myCobot is mainly composed of **BASIC installed at the base** and **Atom installed at the end**. The reason for the design is to separate the motion program from the application program, so that users can program and control myCobot by himself while making it has some real - time performance.

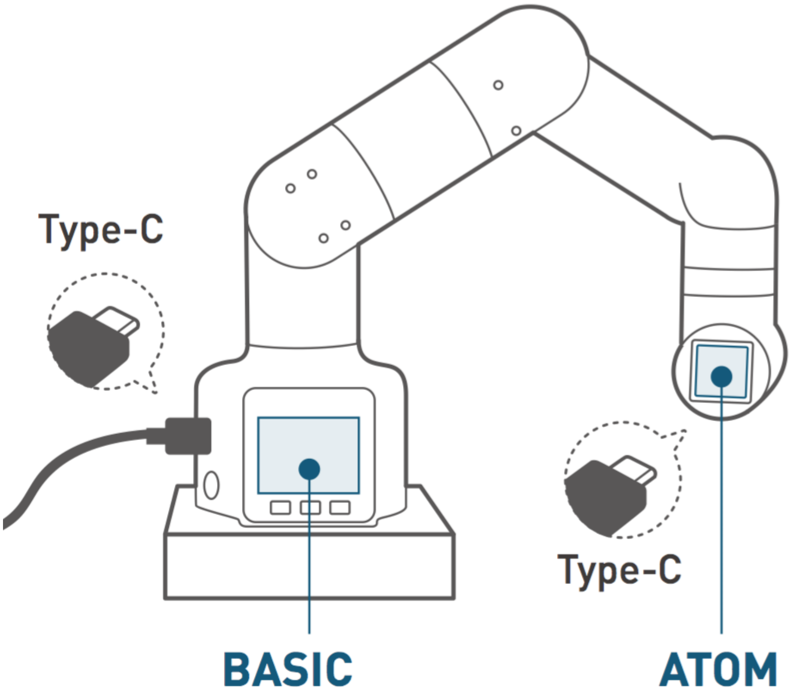
Basic - M5Stack Basic

- It is mainly used for the application side of myCobot, which can carry out Arduino programming, UIFlow programming, communication or various software defined by users themselves. Basic related programs are open source for everyone. Check out our GitHub for more information.

ATOM - M5Stack Atom

- It is mainly used for the kinematics algorithm control of myCobot, including forward and inverse kinematics, solution selection, acceleration and deceleration, speed synchronization, multiple square interpolation, coordinate transformation, real-time control and multi-threading, etc.. Atom related programs are not open source yet.

Note: Basic and Atom can communicate with each other through a **communication protocol** that is open to all users simultaneously.



myCobotElectronic Components

myCobot Electronics

- **Base**（gray shell）
  - Charging Board PCBA: charging and voltage protection function
  - PCBA Substrate: control signal conversion function
  - M5 Basic: main controller
  - Servo Motor No. 1
- **Body** (white shell)
  - Servo Motor No. 2-6
  - Pinboard of Servo Motor
- **End**
  - M5 Atom：2nd controller
- **Peripheral** -Charger

Structural Parts

- **Base**
  - Through-Hole
  - Lego Interface
- **Body**
  - White plastic shell
  - Fixed parts, fasteners, bearings, etc
- **End**

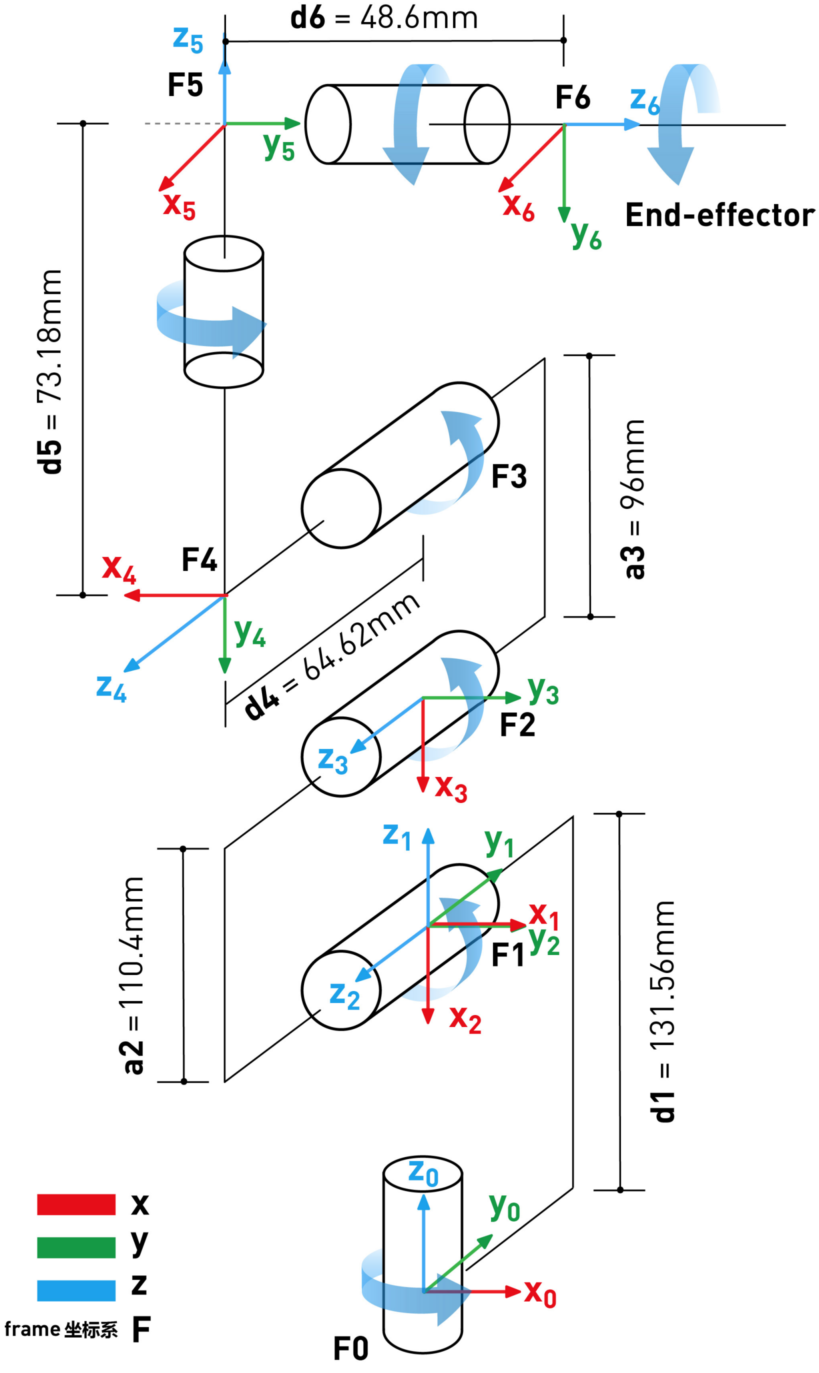
  - Output Flange (silver)

DH Parameter & Coordinate System

DH parameter of myCobot is the modified DH parameter, and the specific parameter values are as follows.

Joint	alpha	a	d	theta	offset
1	0	0	131.56	theta_1	0
2	PI/2	0	0	theta_2	-PI/2
3	0	-110.4	0	theta_3	0
4	0	-96	66.39	theta_4	-PI/2
5	PI/2	0	73.18	theta_5	PI/2
6	-PI/2	0	43.6	theta_6	0

The coordinate system represented by DH parameter of myCobot is as follows.



Maintenance

If there are some problems with your myCobot, you can contact our customer service for maintenance.

BASIC

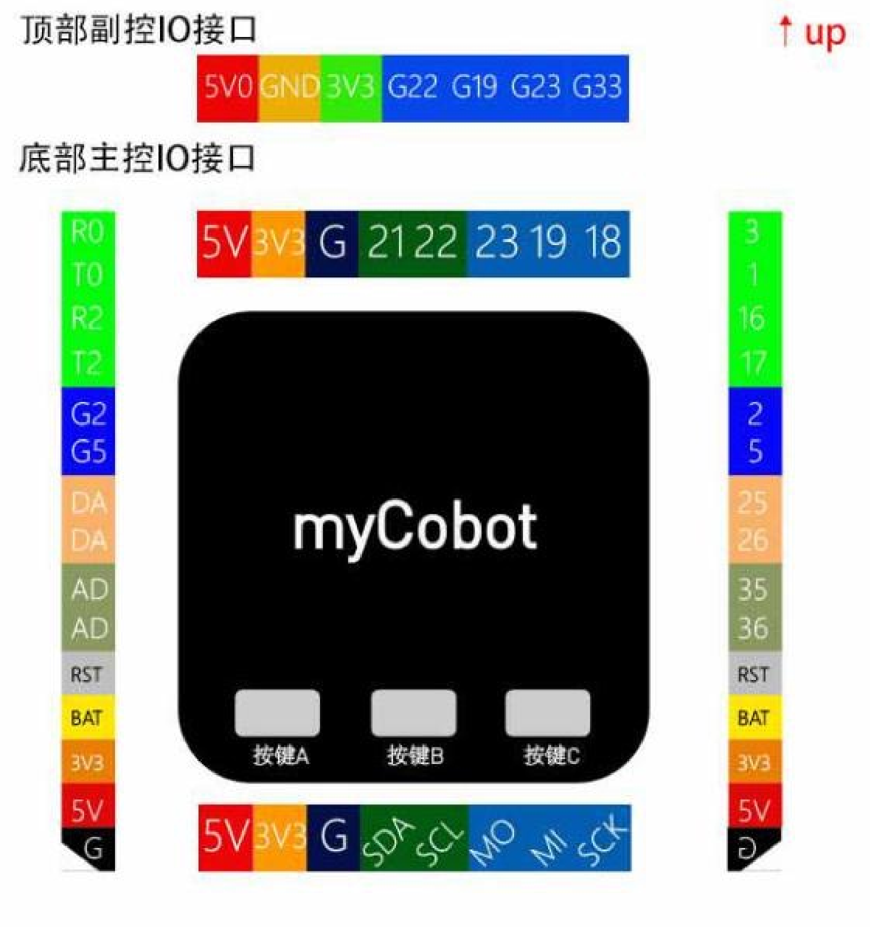
Description

M5Stack BASIC Kit, like its namesake, is a starter kit among the M5Stack development kit series. It’s a modular, stackable, scalable, and portable device which is powered with an ESP-32 core, which makes it open source, low cost, full-function, and easy for developers to handle new product development on all stages including circuit design, PCB design, software, mold design and production. This Basic kit provides a friendly price and full-featured resources which makes it a good starter kit for you to explore IoT.

If you want to explore the fastest way of IoT prototyping, M5Stack development board is the perfect solution. Not like others, M5Stack development board is highly efficient, covered with industrial grade case and ESP32-based development board. It integrates with Wi-Fi & Bluetooth modules and contains a dual-core and 16MB of SPI Flash . Together with 30+ M5Stack stackable modules , 40+ extendable units and different levels of program language, you can create and verify your IoT product in a very short time.

Supportive development platforms and programming languages: Arduino, Blockly language with UIFlow, Micropython. Regardless of what level programming skill you have, M5Stack would guide you in every step of the way to realize your idea as well as to the final productlization. e If you ever played with ESP8266, you would realize that ESP32 is a perfect upgrade out of ESP8266. In comparison, ESP32 has more GPIOs, more analog inputs and two analog outputs, multiple extra peripherals(like a spare UART). Official developing platform ESP-IDF has transplanted with FreeRTOS. With dual-core and real time OS you can get more organized code and much high speed processor.

Pin description of myCobot


顶部副控IO接口
↑ up
5V0 GND 3V3 G22 G19 G23 G33
底部主控IO接口
5V 3V3 G 21 22 23 19 18
R0 T0 R2 T2 G2 G5 DA DA AD AD RST BAT 3V3 5V G
3 1 16 17 2 5 25 26 35 36 RST BAT 3V3 5V G
myCobot
按键A
按键B
按键C
5V 3V3 G SDA SCL MO MI SCK

Features

- ESP32-based
- Built-in Speaker, Buttons,Color LCD, Power/Reset button
- TF card slot (16G Maximum size)
- Magnetic suction at back
- Extendable Pins & Holes
- M-Bus Socket & Pins
- Program Platform: UIFlow, MicroPython, Arduino

Applications

- Internet of things terminal controller
- Stem education product
- DIY creation
- Smart home equipment

Parameters

Resources	Parameter
ESP32	240MHz dual core, 600 DMIPS, 520KB SRAM, Wi-Fi, dual mode Bluetooth
Flash Memory	16MB
Power Input	5V\@ 500mA
Port	TypeC x 1, GROVE(I2C+I/0+UART) x 1
Core Bottom Port	PIN (G1，G2，G3，G16, G17, G18, G19, G21, G22, G23, G25, G26, G35, G36)
IPS Screen	2 inch, 320x240 Colorful TFT LCD, ILI9342C, max brightness 853nit
Button	Custom button x 3
Battery	110mAh\@ 3.7V
Antenna	2.4G 3D Antenna
Operating Temperature	32°F to 104°F (0°C to 40°C)
Net weight	47.2g
Gross weight	93g
Product Size	54 x 54 x 18mm
Package Size	95 x 65 x 25mm
Case Material	Plastic (PC)

LCD & TF card

LCD： 320x240 TF card Maximum size 16GB

ESP32 Chip	GPIO23	GPIO19	GPIO18	GPIO14	GPIO27
ILI9342C	MOSI/MISO	/	CLK	CS	DC
TF	MOSI	MISO	CLK		

Button

ESP32 Chip	GPIO39	GPIO38	GPIO37	GPIO25
Button Pin	BUTTON A	BUTTON B	BUTTON C	/

GROVE Port A & IP5306

ESP32 Chip	GPIO22	GPIO21	5V	GND
GROVE A	SCL	SDA	5V	GND
IP5306	SCL	SDA	5V	GND

IP5306 charging/discharging，Voltage parameter

charging	discharging
0.00~ 3.40V -> 0%	4.20~ 4.07V -> 100%
3.40~ 3.61V -> 25%	4.07~ 3.81V -> 75%
3.61~ 3.88V -> 50%	3.81~ 3.55V -> 50%
3.88~ 4.12V -> 75%	3.55~ 3.33V -> 25%
4.12~ / -> 100%	3.33~ 0.00V -> 0%

ESP32 ADC/DAC

ADC1	ADC2	DAC1	DAC2
8 channels	10 channels	2 channels	2 channels
G32-39	G0/2/4/12-15/25-27	G25	G26

Charging current measured value

charging current	Fully charged current(Power OFF)	Fully charged current(Power ON）
0.55A	-	0.066A

RelatedLink

Datasheet

- ESP32
- IP5306

API

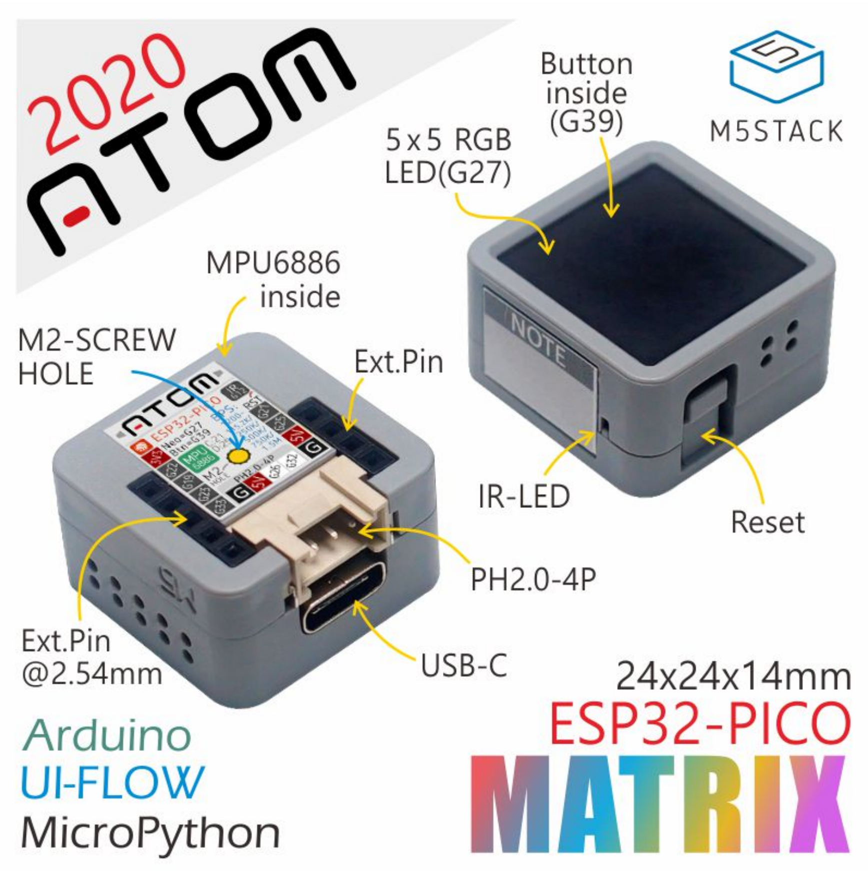
- Arduino API

pcba

- pcba.pdf)

M5 Stack Atom

Description



ATOM Matrix , which has a size of only 24 * 24mm, is the most compact development board in the M5Stack development kit series. It provides more GPIO pins and is very suitable for handy and miniature embedded device development. The main control adopts the ESP32-PICO-D4 chip, which comes integrated with Wi-Fi and Bluetooth technologies and has 4MB of integrated SPI flash memory. The Atom board provides an Infra-Red LED along with the 5 * 5 RGB LED matrix on the panel, a built-in IMU sensor (MPU6886), and a HY2.0 interface. A general purpose programmable button is provied below the RGB Led matrix to enable users to add input support to their various projects. The on-board USB interface (Type-C) enables rapid program uploading and execution. One M2 screw hole is provided on the back for mounting the board.

Note: When using FastLED lib, the recommended brightness of RGB LED is 20. Please do not set it to a high brightness value to avoid damage to the LED and acrylic screen. (In ATOM lib, we have mapped its appropriate brightness range to 0~100)

Product Features

- ESP32 PICO-based
- Programmable button
- 5 * 5 RGB LED matrix panel(WS2812C)
- Buitl-in Infra-red LED
- Built-in MPU6886 Inertial Sensor
- Extendable Pins & Holes
- Program Platform:Arduino,UIFlow

Applications

- Internet of things terminal controller
- IoT node
- Wearable peripherals

Specification

Resources	Parameter
ESP32	240MHz dual core, 600 DMIPS, 520KB SRAM, Wi-Fi, dual mode Bluetooth
Flash	4MB
Power Input	5V\@ 500mA
Port	TypeC x 1, GROVE(I2C+I/0+UART) x 1
PIN Port	G19, G21，G22，G23，G25, G33
RGB LED	WS2812C 2020 x 25
MEMS	MPU6886
IR	Infrared transmission
Button	Custom button x 1
Antenna	2.4G 3D Antenna
Operating Temperature	32°F to 104°F (0°C to 40°C)
Net weight	3g
Gross weight	14g
Product Size	24*24*14mm
Package Size	43*43*20mm
Case Material	Plastic (PC)

3.4 Motors and Servos

Specification

Specification	Parameter
Dimension	45.2X24.7X35mm
Locked-rotor torque	19.5 kg·cm\@ 7.4V
Locked-rotor speed	52RPM\@7.4V
Feedback	Load/position/speed/voltage/current/temperature
Electronic protection	Overheat/overcurrent/overvoltage/overload protection

Structure Feature

- The shell adopts engineering plastic shell with higher strength, which optimizes the center point alignment distance and makes the overall structure more compact.
- Steering gear adopts 1:345 copper tooth combination,which makes greater torque.
- Under the condition of the same torque, it will appear shorter (5mm) than the size of the standard steering gear.
- The body adopts double-axis structure design, round lining three-dimensional structure characteristics with metal main and auxiliary steering wheel and double wire wiring, suitable for quadruped robot, snake robot, desktop robot, humanoid robot, mechanical arm applications.

Electric Control Function

- **Acceleration Start & Stop**Speed and acceleration can be set, the movement is better and softer.

- **High Precision**

> 360-degree absolute position 4096 bit accuracy, the highest position resolves 0.088 degrees. If you control 90 degrees, input 4096/36090=1024; if you control 180 degrees, input 4096/360180=2048; the rest can be done in the same manner .

- **Working Modes**

- **Mode 0:** Location mode, which is the default one. 360 degree absolute Angle control can be achieved in this mode and. Support acceleration motion.

- **Mode 1:** Speed closed loop. In the programming interface, the operation mode is set to 1, you can switch to speed closed loop mode, and enter the corresponding speed under the speed bar to run.

- **Mode 2:** Speed open loop, in the programming interface, the operation mode is set to 2, you can switch to speed open loop mode, and enter the corresponding time under the time bar to run.

A key to Calibrate

> Install in any position with 360°Angle, {enter 128 (decimal) at address number 40 (decimal)} calibrate the current position to the center with one

Learn Structure and fixation of myCobot

How to fix myCobot

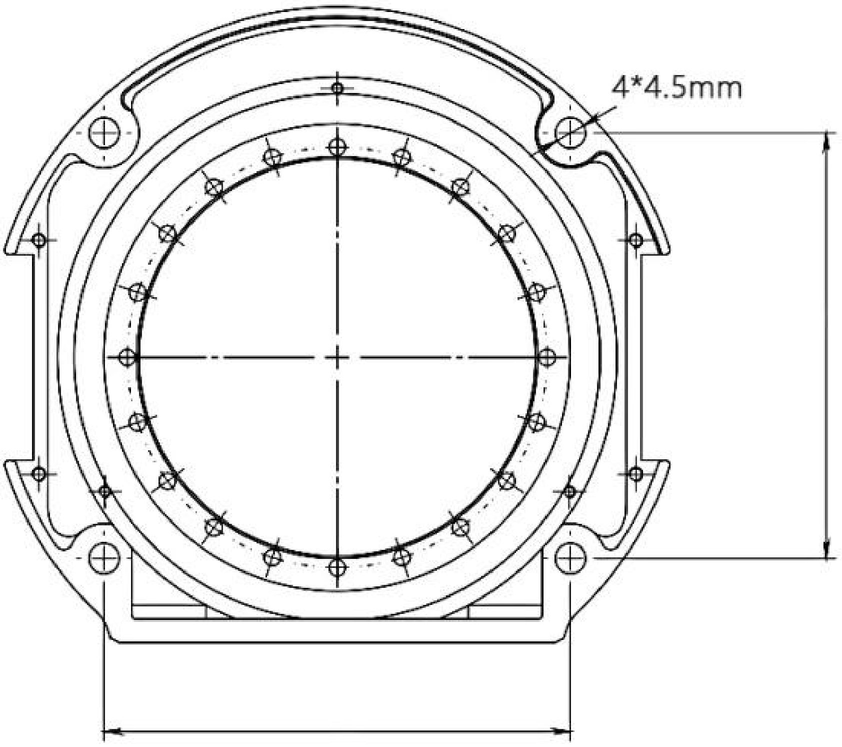
The actual weight of myCobot collaborative robot is 850g. Considering the movement of the robot, the center of gravity will move as the robot moves. Therefore, the robot **needs to be fixed** on a solid base to be used normally.

Common Fix: myCobot There are **two common ways** to fix myCobot:

- 1 Secure the **lego connector** on the base with lego interface.
  types of bases: Flap Base and G Base, you can find them in myCobot's peripheral base. myCobot Support
- Use **4 screws** to pass through myCobot base, secure to a threaded base. The position of the 4 screw holes on the base can be referenced in the myCobot screw position diagram below.

Interface size of robot base

The pedestal fixing hole is the interface that fixes the robot to other bases or planes. The specific hole size is shown as following. It is 4 through holes with a diameter of 4.5mm, which can be fixed with M4 bolts.



Make sure that there is a corresponding threaded hole on the fixed base before installing. **Before you officially install, please confirm:**

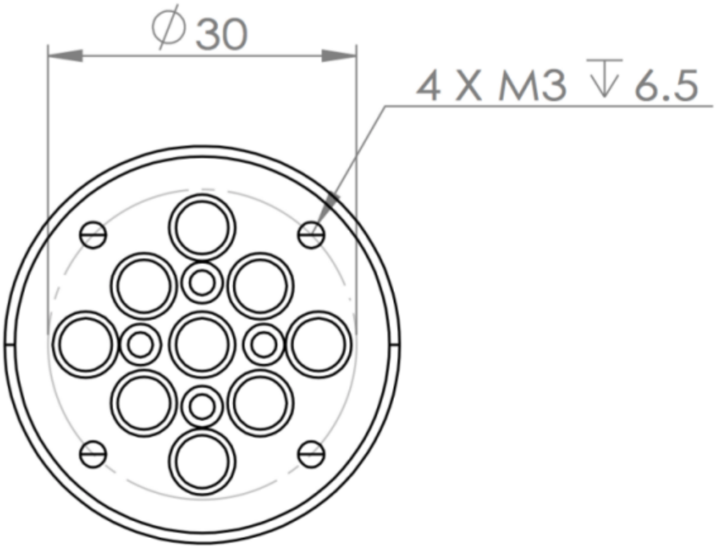
- The environment to be installed complies with the requirements above.
- The installation position is not less than the working range of the robot, and there is enough space for installation, use, maintenance and repair.
- Place the stand in the proper position.
- Installation related tools are ready, such as screws, wrenches, etc.

After confirming the above, move the robot to the mounting surface of the base, adjust the position of the robot, and align the fixing hole of the robot base with the hole on the mounting surface of the base.

Note: When adjusting the position of the robot on the mounting base, please avoid pushing the robot directly on the mounting surface of the base to avoid scratches. When manually moving the robot, please try to avoid applying external force to the weak part of the robot body to avoid unnecessary damage to the robot.

End assembly diagram

The end of the robot arm is compatible with both lego connector holes and screw threaded holes.



MyCobot Urdf Mode

myStudio

The design of myStudio

- myStudio is a one-stop platform for robots of myRobot/myCobot.
- It is convenient for users to choose different firmware and download it according to their own usage scenarios, and learn relevant textbooks and videos.

Supported platforms and versions

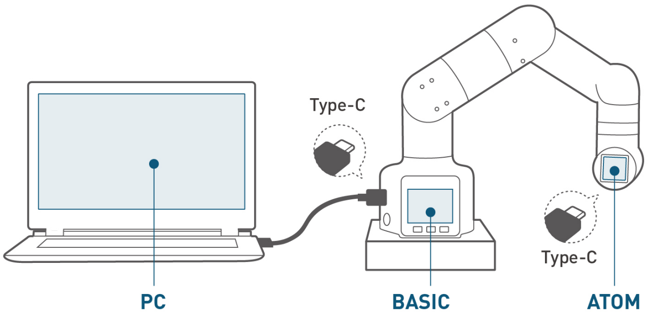
- Version 1.2
- windows, mac, linux

The main functions

- Update the firmware;
- Provide video tutorials on how to use the robot;
- Provide maintenance and repair information (such as video tutorials, Q&A, etc.).

Select your use purpose

Please download different **PC, BASIC** firmware and **Atom** firmware depending on your development environment.



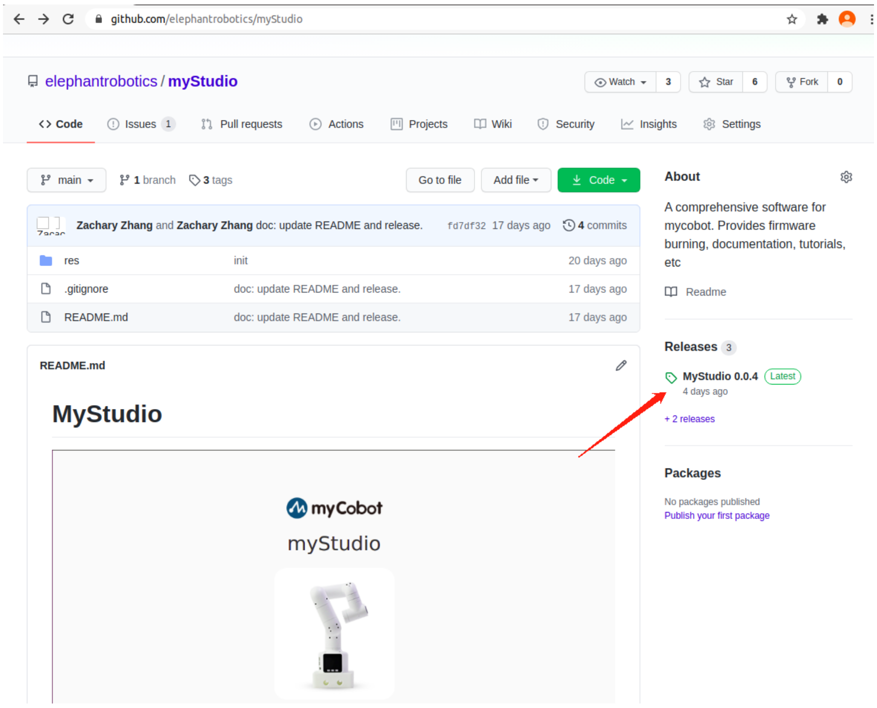
开发使用环境	PC所需Library	Basic所需固件	Atom所需固件
默认程序 Default Program	N/a	mainControl	atomMain
UI flow可视化编程 Visual Programming	UIFLOW	UIFLOW烧录器 由M5提供	atomMain
RoboFlow 工业级可视化编程软件 Industrial Programming Software	RoboFlow库	Transponder文件	atomMain
Arduino 创客！ Maker!	Arduino IDE + M5Stack Lib 库 + MycobotBasic Lib 库	各类程序自定义 All Exapmles	atomMain
API 开发软件接口 on Desktop	Python/ C+	Transponder文件	atomMain
ROS Development ROS 开发	ROS库	Transponder文件	atomMain
通信协议 – USB/TxRx0(G1/G3)	通信协议阅读 Read Protocol	Transponder文件	atomMain
蓝牙或无线连接 BlueTooth 蓝牙直接连接Basic	通信协议阅读 Read Protocol	BT_Transponder文件	atomMain
手机控制器 phoneApp	myCobot 手机 (安卓/苹果) Android/iPhone	BT_Transponder文件	atomMain

Download and Loading myStudio

Download myStudio

myStudio https://github.com/elephantrobotics/myStudio

Select the latest version



Zachary Zhang and Zachary Zhang doc: update README and release. fd7df32 17 days ago 4 commits

res	init	20 days ago
.gitignore	doc: update README and release.	17 days ago
README.md	doc: update README and release.	17 days ago

About

A comprehensive software for mycobot. Provides firmware burning, documentation, tutorials, etc

Readme

Releases 3

MyStudio 0.0.4 Latest

4 days ago

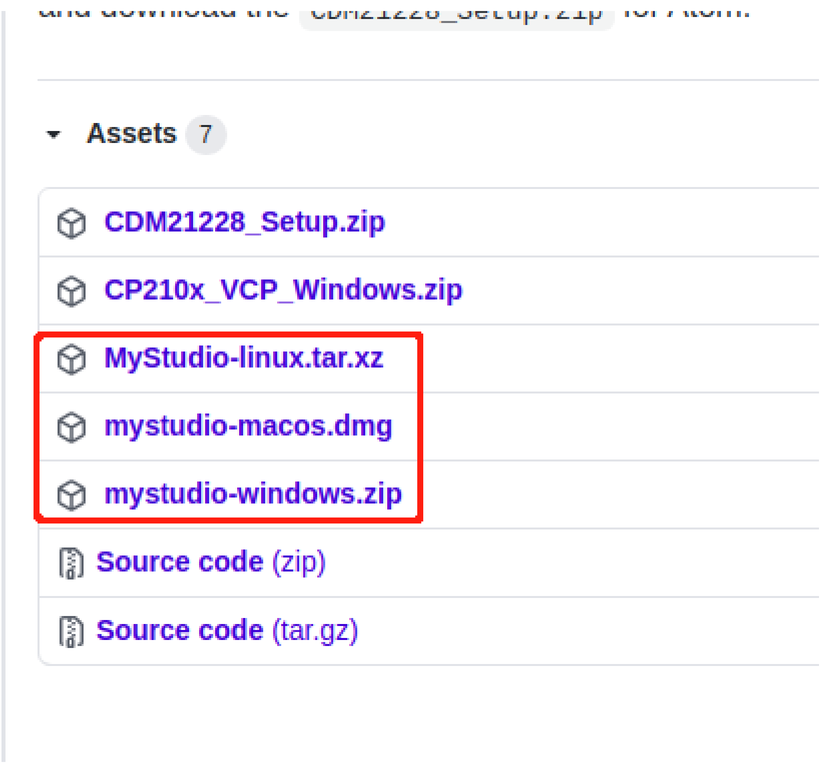
+ 2 releases

Packages

No packages published

Publish your first package

Then select different versions for different systems



and download the `CDM21228_Setup.zip` for Atom.

Assets 7

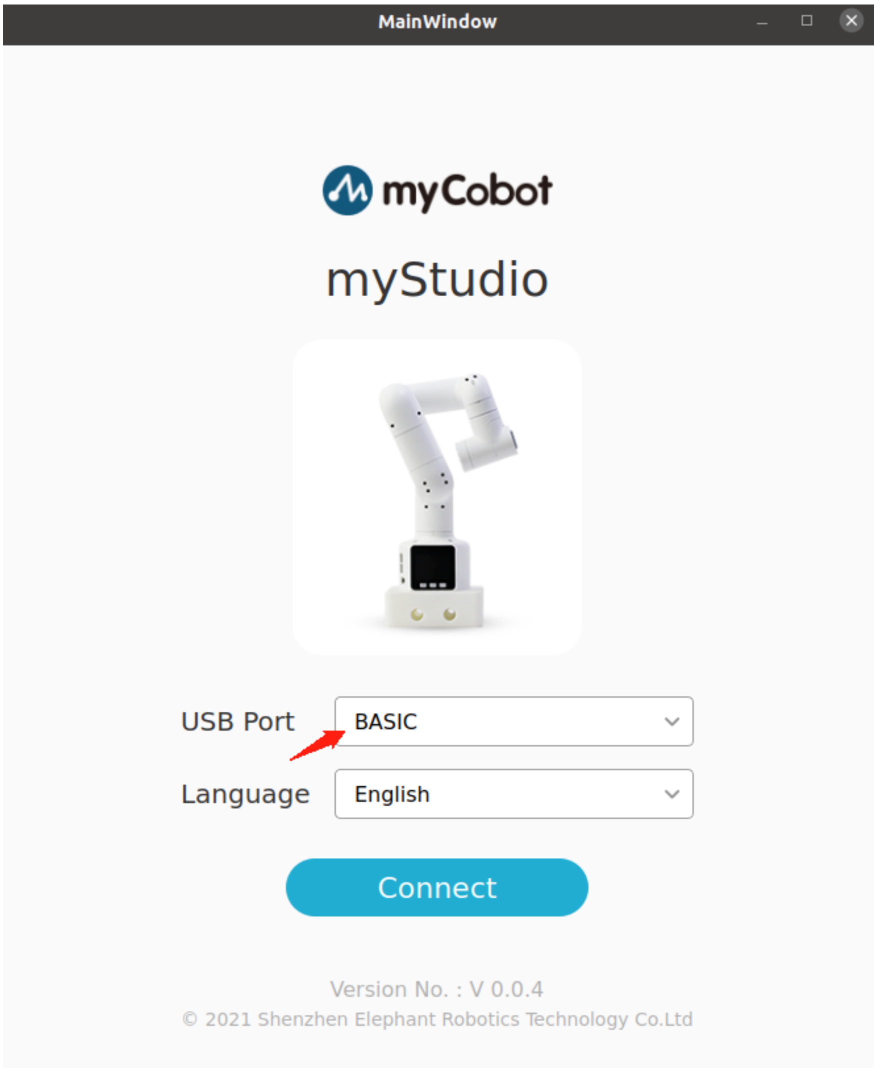
- CDM21228_Setup.zip
- CP210x_VCP_Windows.zip
- MyStudio-linux.tar.xz
- mystudio-macos.dmg
- mystudio-windows.zip
- Source code (zip)
- Source code (tar.gz)

Note: Don’ t install in a folder with a space directory.

Download Basic & Atom

Burn Basic

- First, connect the BASIC development board with USB, the connection window of myStudio will display the connected development board, select and click “Connect”



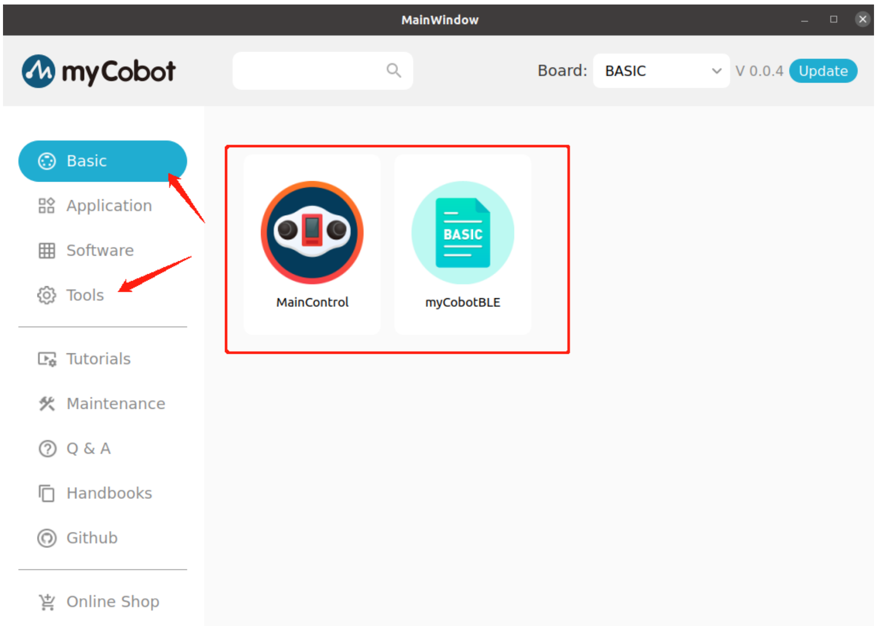
USB Port: BASIC

Language: English

Version No. : V 0.0.4

© 2021 Shenzhen Elephant Robotics Technology Co.Ltd

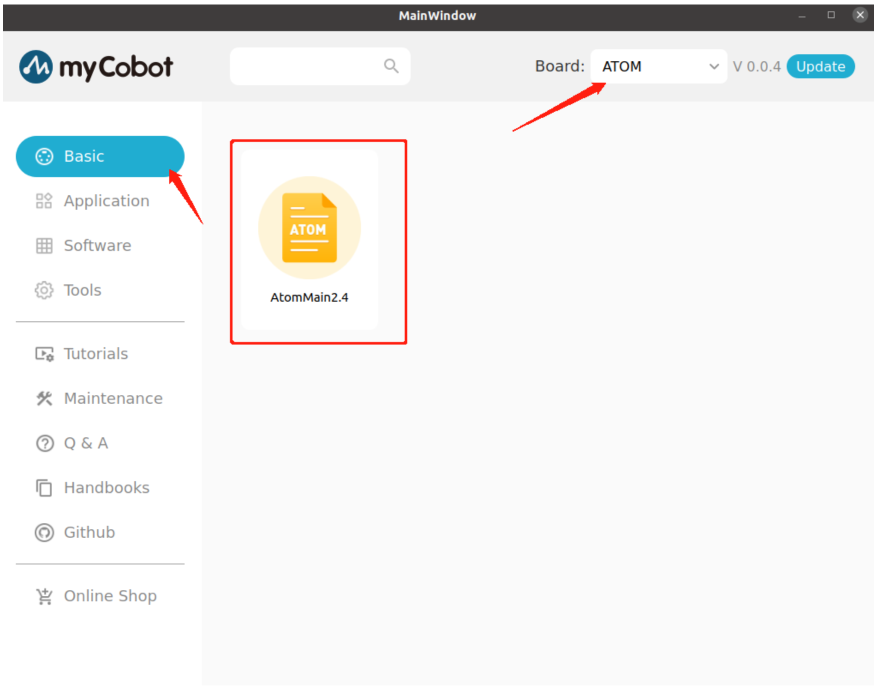
- Then there are the basic-related firmware in the Basic and tools. Select the firmware you want to burn, and click to burn



Board: BASIC V 0.0.4

Burn Atom

- Burn Atom is same as Burn Basic, withUSB connection at the end of the Atom
- ATOM can be selected in the Board, firmware of Atom will appear
- There is only one firmware of Atom, click to burn



Usage of myStudio

https://www.bilibili.com/video/BV1Qr4y1N7B5/

Q&A

Q: When you click on the Tools it will be stuck in the side bar for the first time?

A: Please make sure your network status is good.

Software Platform & API

Before developing, make sure that the Basic and Atom in your myCobot are using the latest firmware and suitable for your environment.

You can update the firmware through **MyStudio**

We currently support the development of the following API

Arduino

- Suitable for maker development, you can use all kinds of Arduino program library.

uiFlow

- Suitable for beginners to develop, based on Visual programming, can be adapted to the full M5 education kit.

python

- Suitable for users with a certain level of Python development.Based on python3.

ROS&MoveIt

- Suitable for professional development, can undertake the simulation and calculation of mechanical arm, trajectory planning, etc.

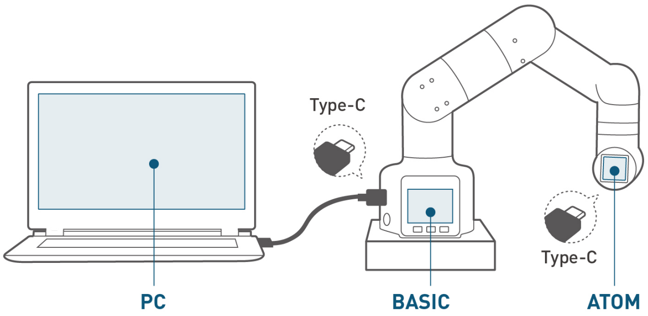
roboFlow

- Suitable for industrial development, It is an industrial oriented operating system launched by Elephant Robot, adapted all series of robotic arms of Elephant Robot

Communication Protocols and Messages

- Suitable for serial port development, can directly communicate by sending message

> In addition, we are also developing C# and other software interface API for development.



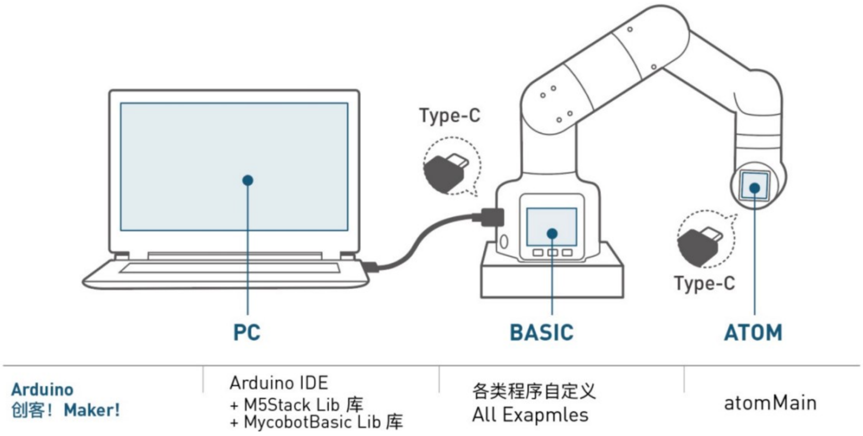
开发使用环境	PC所需Library	Basic所需固件	Atom所需固件
默认程序 Default Program	N/a	mainControl	atomMain
UI flow可视化编程 Visual Programming	UIFLOW	UIFLOW烧录器 由M5提供	atomMain
RoboFlow 工业级可视化编程软件 Industrial Programming Software	RoboFlow库	Transponder文件	atomMain
Arduino 创客！ Maker!	Arduino IDE + M5Stack Lib 库 + MycobotBasic Lib 库	各类程序自定义 All Exapmles	atomMain
API 开发软件接口 on Desktop	Python/ C+	Transponder文件	atomMain
ROS Development ROS 开发	ROS库	Transponder文件	atomMain
通信协议 – USB/TxRx0(G1/G3)	通信协议阅读 Read Protocol	Transponder文件	atomMain
蓝牙或无线连接 BlueTooth 蓝牙直接连接Basic	通信协议阅读 Read Protocol	BT_Transponder文件	atomMain
手机控制器 phoneApp	myCobot 手机 (安卓/苹果) Android/iPhone	BT_Transponder文件	atomMain

Arduino

1.Preparation before development

1.1 Connected Device

- Connect the Basic on the base of myCobot with PC terminal by Tyep-C data line



1.2 Requirement

- ATOM：Burn the latest version of ATOMMain(At least 2.7 Version)
- Basic：None

1.3 Check for Connection

- After connecetion with computer, open device manager to check if there is a device
- If the device is not detected, please replace the USB cable. If it indicates that the device cannot be used, please install and download CP210X . After downloading, unzip it and install the required version of driver.

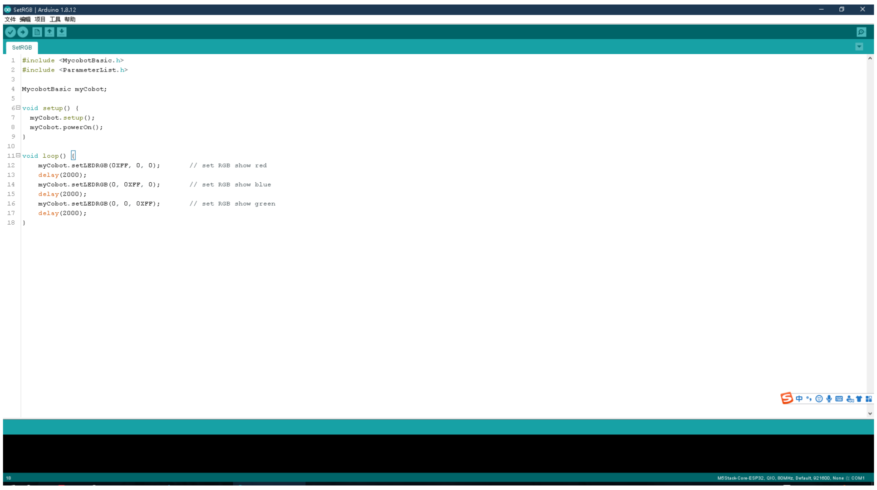
- open Arduino IDE -> tools -> port to check if there is a device. If the device is not detected, please replace the USB cable to test, or test whether the driver has been installed successfully

2.Start development

2.1 Burn an offcial demo

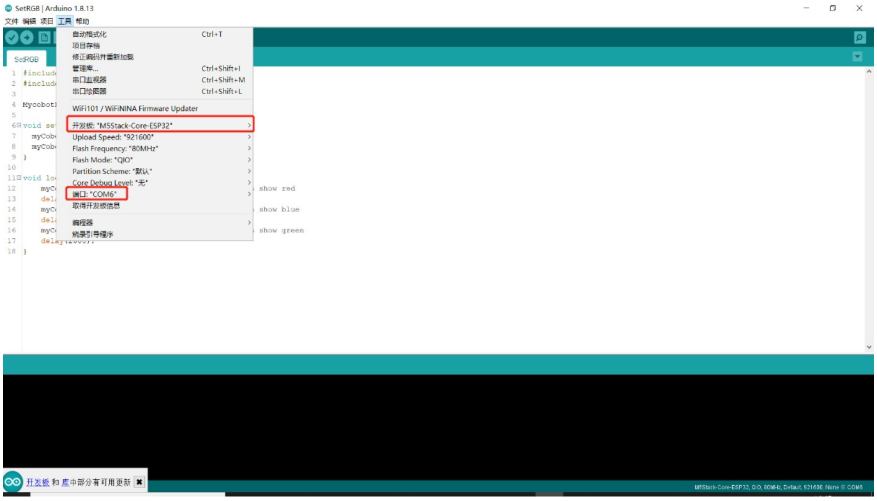
- Open Arduino IDE,select ->file-> Examples-> mycobotbasic, then you can see all Examples about myCobot
- Burn an demo->SetRGB.ino.

Open SetRGB from Examples



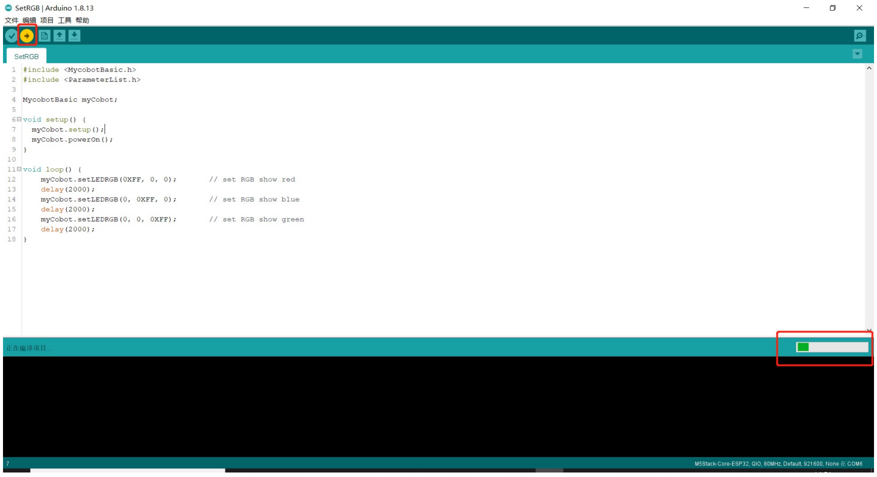
```
#include <MycobotBasic.h>
#include <ParameterList.h>

MycobotBasic myCobot;

void setup() {
  myCobot.setup();
  myCobot.powerOn();
}

void loop() {
    myCobot.setLEDRGB(0XFF, 0, 0);        // set RGB show red
    delay(2000);
    myCobot.setLEDRGB(0, 0XFF, 0);        // set RGB show blue
    delay(2000);
    myCobot.setLEDRGB(0, 0, 0XFF);        // set RGB show green
    delay(2000);
}
```

Select the development board: M5Stack-Core-ESP32and COM



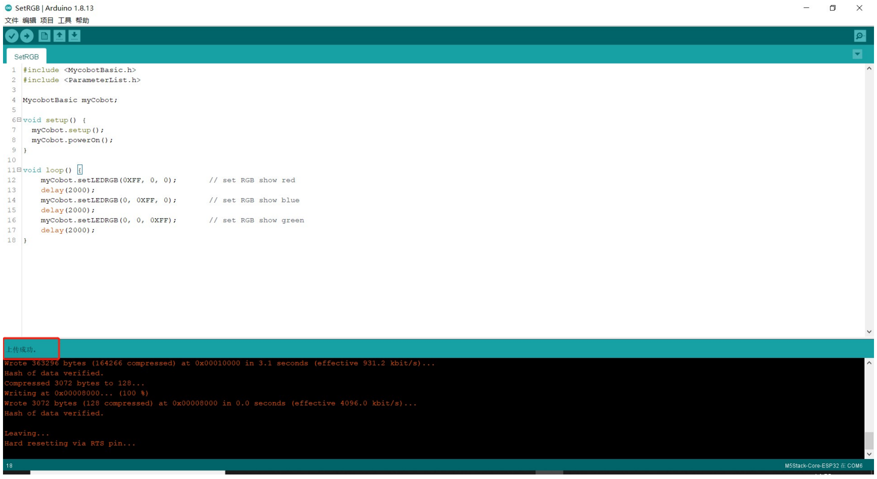
开发板: "M5Stack-Core-ESP32"
Upload Speed: "921600"
Flash Frequency: "80MHz"
Flash Mode: "QIO"
Partition Scheme: "默认"
Core Debug Level: "无"
端口: "COM6"

Click to download



```
#include <MycobotBasic.h>
#include <ParameterList.h>

MycobotBasic myCobot;

void setup() {
  myCobot.setup();
  myCobot.powerOn();
}

void loop() {
    myCobot.setLEDRGB(0XFF, 0, 0);        // set RGB show red
    delay(2000);
    myCobot.setLEDRGB(0, 0XFF, 0);        // set RGB show blue
    delay(2000);
    myCobot.setLEDRGB(0, 0, 0XFF);        // set RGB show green
    delay(2000);
}
```

Wait until the bottom right shows upload success, which means that the application has been downloaded



```
#include <MycobotBasic.h>
#include <ParameterList.h>

MycobotBasic myCobot;

void setup() {
  myCobot.setup();
  myCobot.powerOn();
}

void loop() {
    myCobot.setLEDRGB(0XFF, 0, 0);       // set RGB show red
    delay(2000);
    myCobot.setLEDRGB(0, 0XFF, 0);       // set RGB show blue
    delay(2000);
    myCobot.setLEDRGB(0, 0, 0XFF);       // set RGB show green
    delay(2000);
}
```

上传成功.

```
Wrote 363296 bytes (164266 compressed) at 0x00010000 in 3.1 seconds (effective 931.2 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 128...
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (128 compressed) at 0x00008000 in 0.0 seconds (effective 4096.0 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

Then you'll see the Atom screen loop with red, green and blue lights

Arduino API

1. Overall Status

`powerOn();`

- Function: atom power on（open by default）
- Return Value: none

`powerOff();`

- Function: atom power off
- Return Value: none

`isPoweredOn();`

- Function: atom status inquiry, return Atom link status
- Return Value: power on is TRUE, power off is FALSE

`setFreeMove();`

- Function: All joints close torsion output
- Return Value: none

2. MDI Mode and Robot Control (Manual Data Input)

`getAngles();`

- Function: Read all joint angles, when used one Angles should be defined to receive data that was read. Angles are defined in terms of variables or functions built into library functions. We can define a memory space that is 6 angles to store Angle variables, it is used in the same way as arrays.
- Return Value: Angles type of array

`writeAngle(int joint, float value, int speed);`

- Function: Send a single joint Angle
- Parameter Specification：
  Joint Number= joint, range from 1 to 6 Specified Angle Value= value, range approximately from -160°~ + 160° Specified Speed= value, range from 0~100
- Return Value: none

`writeAngles(Angles angles, int speed);`

- Function: Synchronize joint angles, send joint angles at the same time. Specified Angles is a container with a capacity of 6 data, can be viewed as an array. Use a for loop to assign values, or assign values separately.
- Angles[0] = Specified Angle, Angles[2] = Specify Angle,range from 0 – 90
  (the value range should be the same as writeAngle) unit°
  Movement Speed = speed, range from 0 – 100 unit°
- Return Value: none

`getCoords();`

- Function: Read x,y,z,rx,ry,rz of the end of myCobot, a Coords tempcoords should be defined when used to received angles that was read. Coords are defined in terms of variables or functions built into library functions. We can define a memory space that is 6 tempcoords to store Angle variables, it is used in the same way as arrays.
- Return Value: An array of type Coords. You need to define variables of type Coords.

`isInPosition (Coords coord,bool is_linear);`

- Function：Read x, y, z, rx, ry, rz at the end of the current robot arm to test whether the specified point has been reached, you should define a Coords tempcoords to receive the read angle when you use it. Coords is a variable number or function definition of a library function that defines a storage space of 6 memory, tempcoords, which is used in the same way as an array.
- Return value：An array under the Coords type that needs to define variables of the Coords type

`writeCoord(Axis axis, float value, int speed);`

- Function: Send the specific value of the individual coordinate parameters x/y/z, the ends are going to move in a single direction.
- Parameter Specification:

Value of Moving Path Coordinate = value range from -300 – 300 (The position coordinates of axis=Axis::X,aixs=Axis::Y and axis=Axis::Z are respectively X,Y,Z, the units would be mm. Position coordinate value range is not uniform, axis=Axis::RX, aixs=Axis::RY and axis=Axis::RZ are respectively RX,RY,RZ ranging from-180°~180°, if the value is beyond the range it will return the clue "inverse kinematics no solution")
Specified Speed = speed range from 0~100 unit %

- Return Value: none

`writeCoords(Coords coords, int speed);`

- Function: Send the specified coordinate parameter, the type of parameter is Coords, need to declares a variable of type COORDS, it is used in the same way as arrays.
- Parameter Specification
  coords[0] = X, coords[1] = Y, coords[2] = Z,
  X,Y,Z range from -300-300.00 (Value range is not defined. If the value is beyond the range, the clue "inverse kinematics no solution" will be given) unit mm
  RX,RY,RZ range from -180~180
  Specified Speed = speed, range from 0~100unit %
- Return Value: none

`checkRunning();`

- Function: Check whether the equipment is in motion
- Return Value: In motion is TRUE，on the contrary it's FALSE

`setEncoder(int joint, int encoder);`

- Function: Set a single joint to rotate to a specified potential vaule
- Parameter Specification:
  Joint Number = joint, range from 1-6
  Potential Vaule of Servo Motor = encoder, range from 0-4096 (The range should be positively related to the range of each joint)
- Return Value: none

`getEncoder(int joint);`

- Function: Get the specified joint potential vaule
- Parameter Specification: Servo Motor Number = joint range from 1-6
- Return Value: int type, range from 0-4096

`setEncoders(Angles angleEncoders, int speed);`

- Function: Set the six joints to run synchronously to the specified position
- Parameter Specification: Need to define a variableof type Angles: angleEncoders, it is used in the same way as arrays. Assign a value to the array angleEncoders, values range from 0 to 4096 (The range should be positively related to the range of each joint) , the length range of the array is 6. Specified Speed = speed, range from 0~100unit %
- Return Value: none

3. JOG Mode

`jogAngle(int joint, int direction, int speed);`

- Function: Control the movement of a single joint in one direction
- Parameter Specification:
- Joint/Servo Motor
  - Number = joint, range from 1-6
  - Direction of Joint Motion = Direction, range from -1/1
  - Specified Speed = speed, range from 0~100unit %
- Return Value: none

`jogCoord(Axis axis, int direction, int speed);`

- Function: Control myCobot moves in one direction in Cartesian space
- Parameter Specification:
  - Direction Selection = axis, range from X,Y,Z,RX,RY,RZ
  - Direction of Joint Motion = Direction, range from -1/1
  - Specified Speed = speed, range from 0~100unit %
- Return Value: none

`jogStop();`

- Function: Stops the specified direction of motion that has started
- Return Value: none

`pause();`

- Function: Program pause
- Return Value:none

`resume();`

- Function: Program continues to run
- Return Value: none

`stop();`

- Function: Program stop
- Return Value: none

4. Running Status and Settings

`getSpeed();`

- Function: read the current running speed
- Return Value: int tape, range from 0-100, unit %

`setSpeed(int percentage);`

- Function: set the running speed
- Parameter Specification: percentage, range from 0-100, unit %

`getFeedOverride();` (Not open at this time)

- Function: read FeedOverride
- Return Value: float type of the array

`sendFeedOverride(float feedOverride);` （Not open at this time）

- Function: send FeedOverride

`getAcceleration();` （Not open at this time）

- Function: read acceleration
- Return Value: floattype of the array

`setAcceleration(float acceleration);` （Not open at this time）

- Function: set acceleration
- Parameter Specification: acceleration range from 0-100 (Value range is not defined)

`getJointMin(int joint);`

- Function: read the joint minimal limit Angle
- Parameter Specification: Joint Number = joint, range from 1-6
- Return Value: float type of the array

`getJointMax(int joint);`

- Function: read the joint maximal limit Angle
- Parameter Specification: Joint Number = joint, range from 1-6
- Return Value: float type of the array

`setJointMin(int joint, float angle);`

- Function: set the joint minimal limit Angle
- Parameter Specification: joint/servo motor number = joint, range from 1-6
- Return Value: none

`setJointMax(int joint, float angle);`

- Function: set the joint maximal limit Angle
- Parameter Specification: joint/servo motor number = joint, range from 1-6
- Return Value: none

5. Joint Servo Control

`isServoEnabled(int joint);`

- Function: Check whether the joint is properly connected
- Parameter Specification: Joint Number = joint, range from 1-6
- Return Value: Normal links return TRUE, on the contrary return FALSE

`isAllServoEnabled();`

- Function: Check whether all joins are properly connected
- Return Value: Normal links return TRUE, on the contrary return FALSE

`getServoData(int joint, byte data_id);`

- Function: Read the data of the specified address of joint
- Parameter Specification:
  - Joint Number = joint, range from 1-6
  - Data Address = data_id. , refer to the following Figure for address
- Return Value: refer to the following Figure for the value range

`setServoCalibration(int joint);`

- Function: Calibrate the current position of the joint to zero Angle, the corresponding potential value is 2048
- Parameter Specification: joint number = joint, range from 1 – 6

`jointBrake();`

- Function: brake a single stand-alone
- Parameter Specification: joint number = joint, range from 1 – 6

`setPinMode(byte pin\_no, byte pin\_mode);`

- Function: set atom specified pin mode
- Parameter Specification:
  - Pin Number = pin_no, range from:19, 22, 23, 26, 32, 33
  - Pin mode = pin_mode, range from:0, 1
- Return Value: none

6. Atom IO Control

`setLEDRGB(byte r, byte g, byte b);`

- Function:set the color of the RGB lights of atom
- Parameter Specification:
  - Parameter of Red Light = r, range from 0x00 – 0xFF;
  - Parameter of Green Light = g, range from 0x00 – 0xFF;
  - Parameter of Blue Light = b, range from 0x00 – 0xFF;
- Return Value: none

`setGripper(int data);`

- Function: Set the jaw opening and closing
- Parameter Specification: data 0 is open, 1 is close`

7. Coordinate Control Mode

`setToolReference(Coords coords);`

- Function: set coordinate system of tool
- Parameter Specification:
  - X,Y,Z range from -300-300.00 (Value range is not defined. If the value is beyond the range, the clue ”inverse kinematics no solution” will be given) unit mm
  - RX,RY,RZ range from -180~180
- Value: none

`setWorldReference(Coords coords);`

- Function: set coordinate system of world
- Parameter Specification:
  - X,Y,Z range from -300-300.00 (Value range is not defined. If the value is beyond the range, the clue ”inverse kinematics no solution” will be given) unit mm
  - RX,RY,RZ range from -180~180
- Return Value: none

`getToolReference();`

- Function: get coordinate system of tool
- Parameter Specification: none
- Return Value:
  - X,Y,Z range from -300-300.00 (Value range is not defined. If the value is beyond the range, the clue ”inverse kinematics no solution” will be given) unit mm
  - RX,RY,RZ range from -180~180

`getWorldReference();`

- Function: get coordinate system of world
- Parameter Specification: none
- Return Value:
  - X,Y,Z range from -300-300.00 (Value range is not defined. If the value is beyond the range, the clue ”inverse kinematics no solution” will be given) unit mm
  - RX,RY,RZ range from -180~180

`setReferenceFrame(RFType rftype);`

- Function: set coordinate system of frame
- Parameter Specification: RFType::BASE takes the robot base as the base coordinate, RFType::WORLD takes the world/Youdao/Dict/8.9.6.0/resultui/html/index.html#/javascript:;) coordinate/Youdao/Dict/8.9.6.0/resultui/html/index.html#/javascript:;)

system/Youdao/Dict/8.9.6.0/resultui/html/index.html#/javascript:;) as the base coordinate
- Return Value: none

`getReferenceFrame();`

- Function：get coordinate system of flange
- Parameter Specification: none
- Return Value:
- X,Y,Z range from -300-300.00 (Value range is not defined. If the value is beyond the range, the clue ”inverse kinematics no solution” will be given) unit mm

RX,RY,RZ range from -180~180

`setEndType(EndType end\_type)`

- Function: set coordinate system of the end
- Parameter Specification: EndType::FLANGE set the end as flange, EndType::TOOL set the end to the tool end
- Return Value: none

`getEndType();`

- Function: get coordinate system of the end
- Parameter Specification: none
- Return Value:
    - X,Y,Z range from -300-300.00 (Value range is not defined. If the value is beyond the range, the clue ”inverse kinematics no solution” will be given) unit mm
    - RX,RY,RZ range from -180~180

`setGripperValue();`

- Function: Set the electric potential of the gripper. Get the current electric potential of the gripper before use
- Parameter Specification: Input Value (0-4095)
- Return Value: none

`setGripperIni();`

- Function: Set the gripperbe .zero
- Parameter Specification: none
- Return Value: none

`getGripperValue();`

- Function: get the value of the current electric potential
- Parameter Specification: none
- Return Value: return the value of the current electric potential, range from0-4085

`setGripperState;`

- Function: Set the opening and closing of the gripper
- Parameter Specification: 0 represents open,1 represents clsoe
- Return Value: none

`setDigitalOutput;`

- Function: Set the working state of IO pins
- Parameter Specification: 0 input; 1 output; 2 pull_up_input
- Return Value: none

`getDitialInput;`

- Function: Read input
- Parameter Specification: none
- Return Value: none

`setPWMOutput(byte pin_no, int freq, byte pin_write);`

- Function：Set the PWM signal of the ATOM end IO output that specifies the duty-through ratio
- Parameter description： pin_no：IO serial number freq： clock frequency pin_write：ratio: 0 ~ 256;128 indicates 50%.
- Return value：none

Test Program

Functional Specification

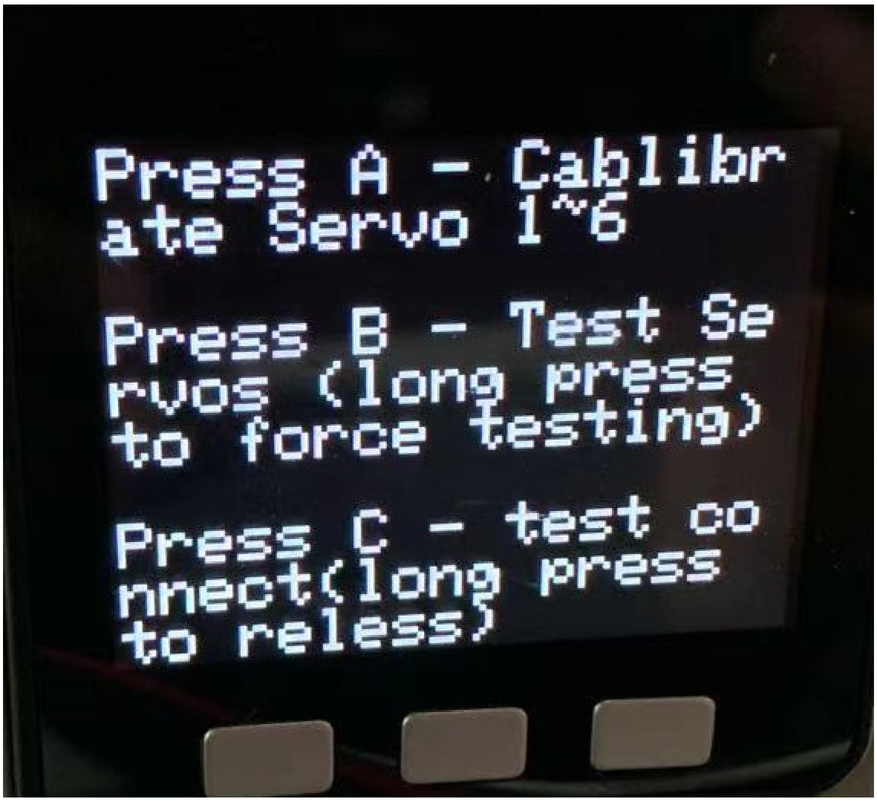
Burn the connect-test firmware, which can detect the connection of the device, and the Basic screen will show the linked device

Requirement

- Use MyStudio Basic -> connect-test
- Use MyStudio Atom -> AtomMain

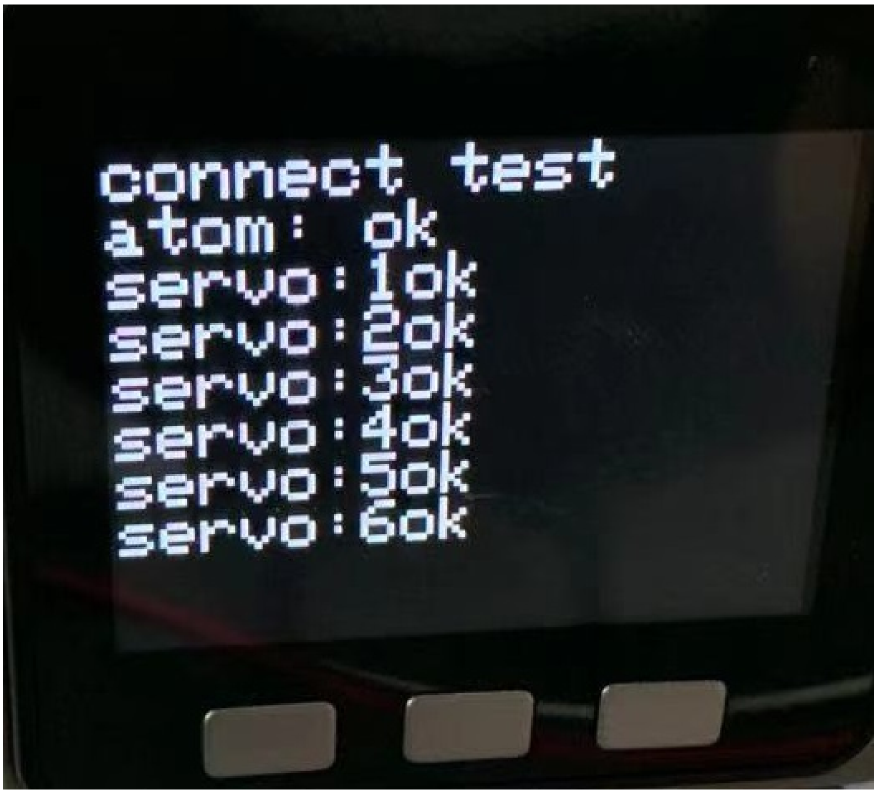
Flow Chart

- Burn AtomMain for Atom
- Burn connect-test for Basic



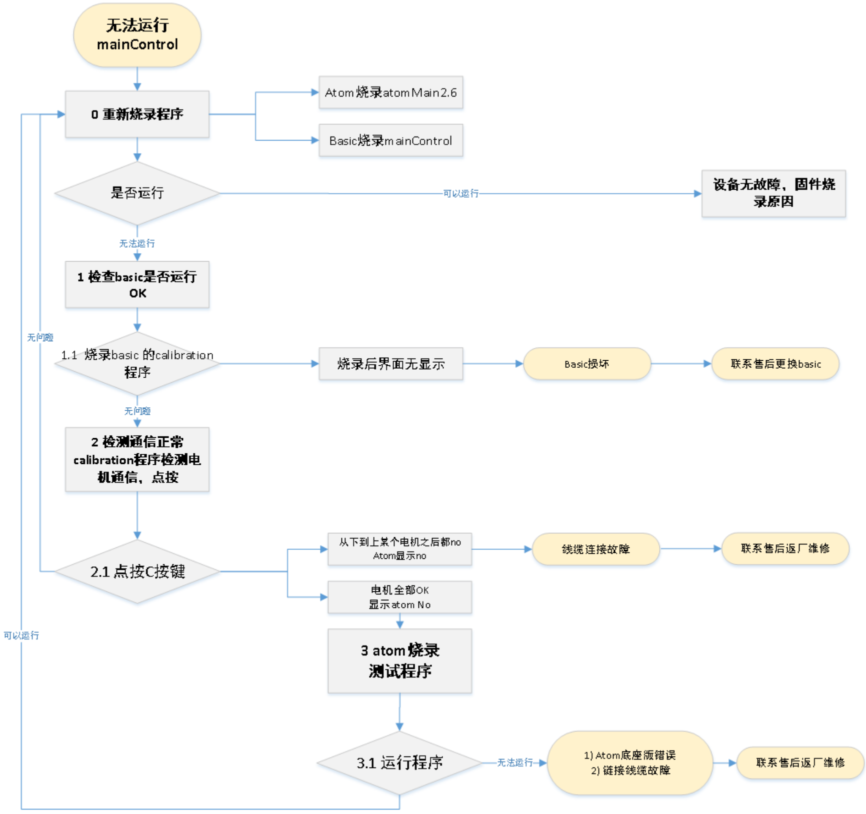
```
Press A - Cablibr
ate Servo 1~6

Press B - Test Se
rvos (long press
to force testing)

Press C - test co
nnect(long press
to reless)
```

- Press Button C：



```
connect test
atom:  ok
servo:1ok
servo:2ok
servo:3ok
servo:4ok
servo:5ok
servo:6ok
```

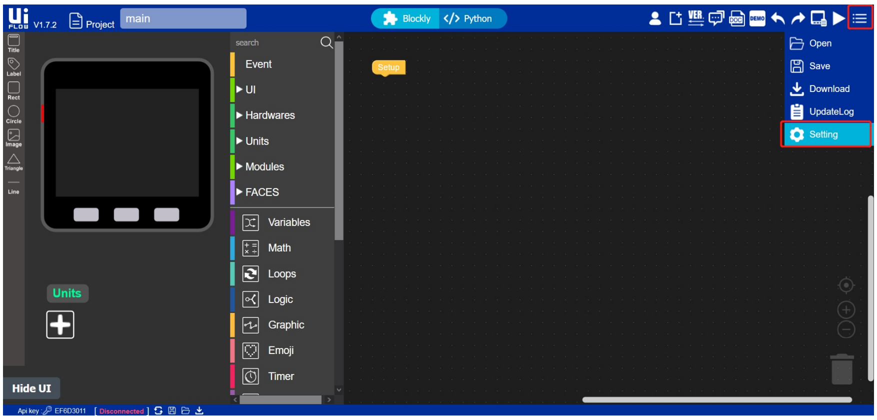
- If there is any problem：

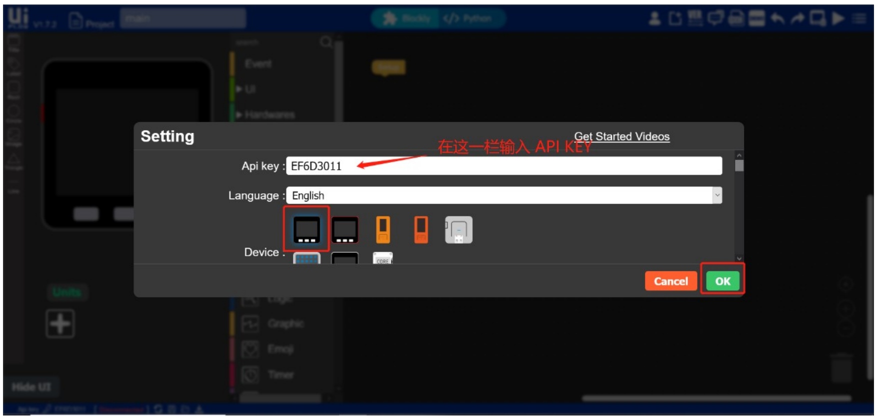


5.2 UIFlow

Press the web URL of uiFlow :https://flow.m5stack.com/https://flow.m5stack.com/

Operate as shown below/

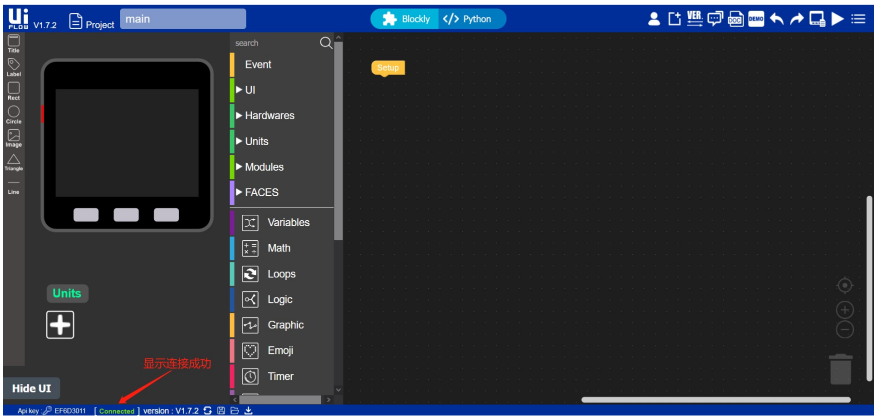



Setting

Api key : EF6D3011

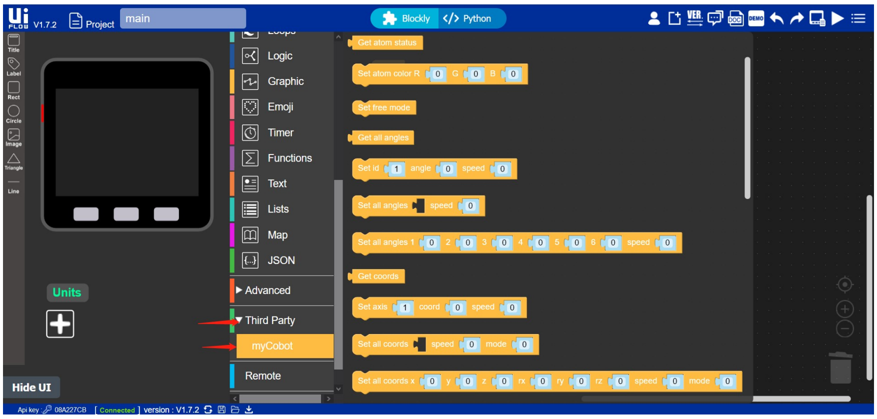
Language : English

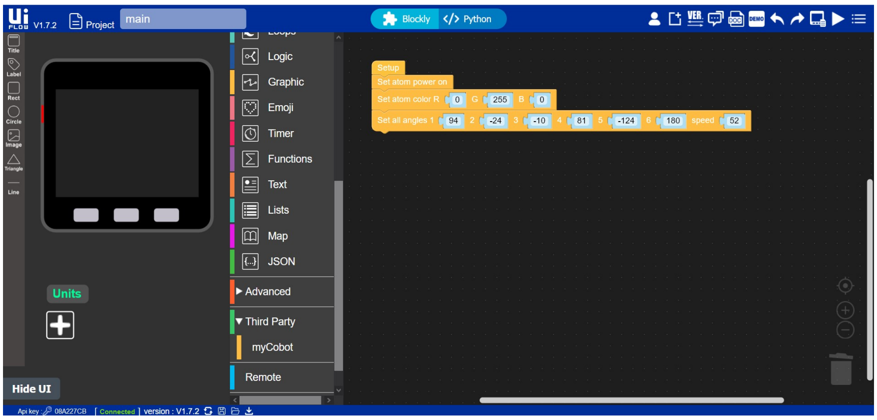
Device :



Then you can start programming

Make the Atom screen green and adjust the angles of the six joints as shown in the figure





```
Setup
Set atom power on
Set atom color R 0 G 255 B 0
Set all angles 1 94 2 -24 3 -10 4 81 5 -124 6 180 speed 52
```

5.3 Python

> Programming in Python Environment

Identify development goals

PyMycobot is a Python package that communicates with myCobot on a serial port. It supports Python2, Python3.5 and later versions..

If you want to control myCobot by programming in Python, that is your choice.

How to install and use

Before using PyMycobot, make sure you have a myCobot and have it ready.

Pre-preparation:

> Make sure the top Atom burns into Atom and the bottom BASIC burns into Transponder.
>
> The firmware Atom and Transponder download address: https://github.com/elephantrobotics/myCobot/tree/main/Software

Pip

```
pip install pymycobot --upgrade --user
```

Notes:

> Only Atom2.6 and later versions are currently supported; if you are using previous versions, please install PyMycobot 1.0.7.

```
pip install pymycobot==1.0.7 --user
```

Installing from Source

```
git clone https://github.com/elephantrobotics/pymycobot.git <your-path>
cd <your-path>/pymycobot
# Install
python2 setup.py install
# or
python3 setup.py install
```

API serves as a secondary directory

If you don't want to install it, you can use it.First, you need to go to GitHub and download it locally.

> Download Path：https://github.com/elephantrobotics/pymycobot

Then, put the Pymycobot file in your project so that you can import it and use it.

pymycobot API

Class:

- MyCobot
    - Overall status
    - MDI mode and operation
    - JOG mode and operation
    - Running status and Settings
    - Servo control
    - Atom IO
- Angle
- Coord

MyCobot

```
from pymycobot.mycobot import MyCobot
```

> Note: If no parameter is given, there is no parameter; if no return value is given, there is no return value

Overall status

power_on()

- **Description**: Robot arm power up.

power_off()

- **Description**: Robot arm power down.

is_power_on()

- **Description**: Adjust robot arm whether power on.
- **Returns**
    - `1` : power on
    - `0` : power off
    - `-1` : error

set_free_mode()

- **Description**: Set robot arm into free moving mode.

MDI mode and operation

get_angles()

- **Description**: Get the degree of all joints.
- **Returns**: `list` : A float list of all degree.

send_angle()

- **Description**: Send one degree of joint to robot arm.
- **Parameters**
    - `id` : Joint id(`genre.Angle`)
    - `degree` : degree value(`float`)
    - `speed` : (`int`) 0 ~ 100
- **Example**

```python
from pymycobot.mycobot import MyCobot
from pymycobot.genre import Angle

mycobot = MyCobot('/dev/ttyUSB0')
mycobot.send_angle(Angle.J2.value, 10, 50)
```

send_angles()

- **Description**: Send the degrees of all joints to robot arm.
- **Parameters**

    `degrees` : a list of degree value(`List[float]`)

    `speed` : (`int`)

- **Example**

```python
from pymycobot.mycobot import MyCobot
from pymycobot.genre import Angle

mycobot = MyCobot('/dev/ttyUSB0')
mycobot.send_angles([0,0,0,0,0,0], 80)
```

get_radians()

- **Description**: Get the radians of all joints.
- **Returns**: `list` : A float list of radian.

send_radians()

- **Description**: Send the radians of all joint to robot arm.
- **Parameters**:

    - `degrees` : a list of radian value(`List[float]`)
    - `speed` : (`int`) 0 ~ 100
- **Example**

```
from pymycobot.mycobot import MyCobot
from pymycobot.genre import Angle

mycobot = MyCobot('/dev/ttyUSB0')
mycobot.send_angles_by_radian([1,1,1,1,1,1], 70)
```

get_coords()

- **Description**: Get the Coords from robot arm, coordinate system based on base.
- **Returns**: `list` : A float list of coord - `[x, y, z, rx, ry, rz]`

send_coord()

- **Description**: Send one coord to robot arm.
- **Parameters**
    - `id` : coord id(`genre.Coord`)
    - `coord` : coord value(`float`)
    - `speed` : (`int`) 0 ~ 100
- **Example**

```
from pymycobot.mycobot import MyCobot
from pymycobot.genre import Coord

mycobot = MyCobot('/dev/ttyUSB0')
mycobot.send_coord(Coord.X.value, -40, 70)
```

send_coords()

- **Description**: Send all coords to robot arm.
- **Parameters**
    - `coords` : a list of coords value(`List[float]`)
    - `speed` : (`int`) 0 ~ 100
    - `mode` : (`int`): `0` - angluar, `1` - linear
- **Example**

```
from pymycobot.mycobot import MyCobot
from pymycobot.genre import Coord

mycobot = MyCobot('/dev/ttyUSB0')
mycobot.send_coords([160, 160, 160, 0, 0, 0], 70, 0)
```

sync_send_angles()

- **Description**: Send the angle in synchronous state and return when the target point is reached
- **Parameters**
  - `id` : Joint id(`genre.Angle`)
  - `degree` : degree value(`float`)
  - `speed` : (`int`) 0 ~ 100

sync_send_coords()

- **Description**: Send the coord in synchronous state and return when the target point is reached
- **Parameters**
  - `coords` : a list of coords value(`List[float]`)
  - `speed` : (`int`) 0 ~ 100
  - `mode` : (`int`): `0` - angluar, `1` - linear

pause()

- **Description**: Pause movement.

resume()

- **Description**: Recovery movement.

stop()

- **Description**: Stop moving.

is_paused()

- **Description**: Judge whether the manipulator pauses or not.
- **Returns** :
  - `1` - paused
  - `0` - not paused
  - `-1` - error

is_in_position()

- **Description**: Judge whether in the position.
- **Parameters**
  - `data` : A data list, angles or coords.
  - `flag` : Tag the data type, `0` - angles, `1` - coords.
- **Returns**
  - `1` - true
  - `0` - false

    - `-1` - error

JOG mode and operation

jog_angle()

- **Description**: Jog control angle
- **Parameters**
    - `joint_id` : (`int`) 1 ~ 6
    - `direction` : `0` - decrease, `1` - increase
    - `speed` : 0 ~ 100

jog_coord()

- **Description**: Jog control coord.
- **Parameters**
    - `coord_id` : (`int`) 1 ~ 6
    - `direction` : `0` - decrease, `1` - increase
    - `speed` : 0 ~ 100

jog_stop()

- **Description**: Stop jog moving.

Running status and Settings

get_speed()

- **Description**: Get speed.
- **Returns**: speed: (`int`)

set_speed()

- **Description**: Set speed.
- **Parameters**: speed: (`int`) 0 ~ 100

get_joint_min_angle()

- **Description**: Gets the minimum movement angle of the specified joint
- **Parameters**: `joint_id` : (`int`)
- **Returns**: angle value (`float`)

get_joint_max_angle()

- **Description**: Gets the maximum movement angle of the specified joint
- **Parameters**: `joint_id` : (`int`)
- **Returns**: angle value (`float`)

Servo control

is_servo_enable()

- **Description**: Determine whether all steering gears are connected
- **Parameters**: `servo_id` (`int`) 1 ~ 6
- **Returns**
  - `0` : disable
  - `1` : enbale
  - `-1` : error

is_all_servo_enable()

- **Description**: Determine whether the specified steering gear is connected
- **Returns**
  - `0` : disable
  - `1` : enbale
  - `-1` : error

release_servo()

- **Description**: Power off designated servo
- **Parameters**: `servo_id` : 1 ~ 6

focus_servo()

- **Description**: Power on designated servo
- **Parameters**: `servo_id` : 1 ~ 6

Atom IO

set_color()

- **Description**: Set the color of the light on the top of the robot arm.
- **Parameters**
  - `r` : 0 ~ 255
  - `g` : 0 ~ 255
  - `b` : 0 ~ 255

set_pin_mode()

- **Parameters**
  - `pin_no` (int):
  - `pin_mode` (int): 0 - input, 1 - output, 2 - input_pullup

set_digital_output()

- **Parameters**
  - `pin_no` (int):
  - `pin_signal` (int): 0 / 1

get_digital_input()

- **Parameters**: `pin_no` (int)
- **Return**: signal value

get_gripper_value()

- **Description**: Get gripper value
- **Return**: gripper value (int)

set_gripper_state()

- **Description**: Set gripper switch state
- **Parameters**
  - `flag` (`int`): 0 - open, 1 - close
  - `speed` (`int`): 0 ~ 100

set_gripper_value()

- **Description**: Set gripper value
- **Parameters**
  - `value` (int): 0 ~ 4096
  - `speed` (int): 0 ~ 100

set_gripper_ini()

- **Description**: Set the current position to zero, set current position value is `2048` .

is_gripper_moving()

- **Description**: Judge whether the gripper is moving or not
- **Returns**

  - `0` : not moving
  - `1` : is moving
  - `-1` : error data

Angle

```
from pymycobot.genre import Angle
```

Description

Instance class of joint. It's recommended to use this class to select joint.

Coord

```
from pymycobot.genre import Coord
```

Description

Instance class of coord. It's recommended to use this class to select coord.

ROS Introduction

ROS is abbreviation of Robot Operating System. ROS is a highly flexible software architecture for writing robotic software programs.

ROS icon：

ROS is open source, a post-operating system, or secondary operating system for robot control. It provides features similar to those provided by the operating system, including hardware abstract descriptions, underlying driver management, execution of common functionality, inter-program messaging, and package management, as well as tool programs and libraries for acquiring, building, writing, and running multi-machine integrated programs.

ROS's primary design goal is to increase code reuse in the field of robotics development. ROS is a framework of distributed processes (or "nodes") that are encapsulated in packages and feature packs that are easy to share and publish. ROS also supports a joint system similar to a code repository, which can also enable engineering collaboration and release. This design enables the development of an engineering implementation to make completely independent decisions (without ROS restrictions) from the file system to the user interface. At the same time, all projects can be integrated with the basic tools of ROS.

MoveIt Introduction

Moveit! is the most advanced software for robotic arm movement operations and is used on more than 100 robots. It combines the latest results in motion planning, control, 3D perception, operations control, control and navigation, provides an easy-to-use platform for developing advanced robotics applications, and provides an integrated software platform for the design and evaluation of new robot products in industrial, commercial and research and development fields.

Moveit icon：

Basic development environments require the installation of robotic operating systems ROS, MoveIt, and git version managers, which are described below.

ROS Installation

1.Choose version

ROS has a one-to-one relationship with ubuntu, and different versions of ubuntu correspond to different versions of ROS, the website below:http://wiki.ros.org/Distributions

If the version is different, the download will fail. The system we selected here is Ubuntu 16.04, corresponding to ros version ROS Kinetic Kame.

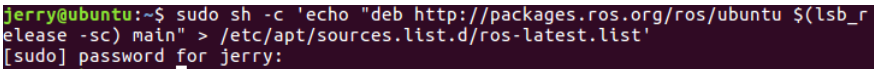
2.Begin installation

2.1.Add source

There are no ROS software sources in the list of Ubuntu's own software sources, so you need to configure the ROS software source into the software list repository first, then you can download ROS and open a console terminalCtrl+Alt+T),Enter the following instructions：

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /e
```

The results are as follows (the user password is required here, just enter the user password set when Ubuntu is installed)



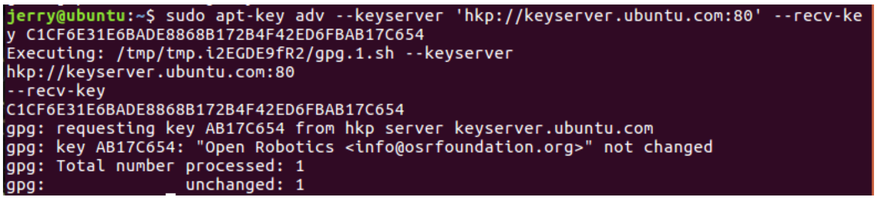
```
jerry@ubuntu:~$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_r
elease -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
[sudo] password for jerry:
```

2.2.Set the key

Configure the public network key, this step is to let the system confirm that our path is safe, so that download file is no problem, otherwise the download will be deleted immediately:

```
sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key C1CF6E31E6BADE
```

The results of the execution are as follows:



```
jerry@ubuntu:~$ sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-ke
y C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
Executing: /tmp/tmp.i2EGDE9fR2/gpg.1.sh --keyserver
hkp://keyserver.ubuntu.com:80
--recv-key
C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
gpg: requesting key AB17C654 from hkp server keyserver.ubuntu.com
gpg: key AB17C654: "Open Robotics <info@osrfoundation.org>" not changed
gpg: Total number processed: 1
gpg:              unchanged: 1
```

2.3.installation

After adding a new software source, update the list of software sources:

```
sudo apt-get update
```

install ROS：

```
sudo apt-get install ros-kinetic-desktop-full
```

It is recommended to install a complete ROS to prevent the loss of libraries and dependencies.

The installation process takes a long time and requires patience

2.4.Configure the ROS environment to the system

rosdep allows you to easily install the source code that you want to compile, or be dependent on some system that the ROS core components,need, and execute the following commands in turn at the terminal. Initialize rosdep：

```
sudo rosdep init
```

```
rosdep update
```

Once initialization is complete, in order to avoid having to re-take effect on the ROS functional path after each shutdown of the terminal window, we can configure the path into the environment variable so that the ROS functional path automatically takes effect each time a new terminal is opened. At the terminal, the following commands are executed in turn:

Bash

```
echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc
source ~/.bashrc
```

Zsh

```
echo "source /opt/ros/kinetic/setup.bash" >> ~/.zshrc
source ~/.zshrc
```

2.5.Install the ROS extra dependence

Enter the following command at the terminal：

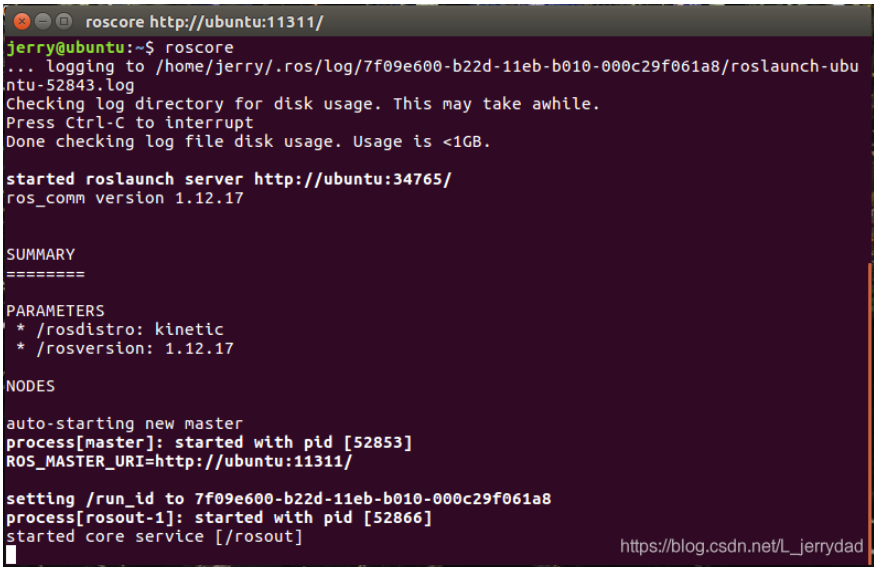
```
sudo apt-get install python-rosinstall python-rosinstall-generator python-wstool buil
```

3.Verify and installation

The start-up of the ROS system requires a ROS Master, the node manager, and we can enter the roscore instruction at the terminal to start the ROS Maste, to verify that the ROS was installed successfully, the following commands are executed at the terminal:

```
roscore
```

When the following interface is displayed, means the ROS installation is successful


roscore http://ubuntu:11311/
https://blog.csdn.net/L_jerrydad

```
jerry@ubuntu:~$ roscore
... logging to /home/jerry/.ros/log/7f09e600-b22d-11eb-b010-000c29f061a8/roslaunch-ubu
ntu-52843.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://ubuntu:34765/
ros_comm version 1.12.17


SUMMARY
========

PARAMETERS
 * /rosdistro: kinetic
 * /rosversion: 1.12.17

NODES

auto-starting new master
process[master]: started with pid [52853]
ROS_MASTER_URI=http://ubuntu:11311/

setting /run_id to 7f09e600-b22d-11eb-b010-000c29f061a8
process[rosout-1]: started with pid [52866]
started core service [/rosout]
```

For more detailed installation instructions, you can refer to the official installation instructions，website: http://wiki.ros.org/ROS/Installation

MoveIt Installation

MoveIt is the component of a series of mobile operations in ros, including motion planning, collision detection, kinematics, 3D awareness, operation control and other functions.

1.Update the list of software sources

Enter the following command in the terminal window to update the list of software sources:

```
sudo apt-get update
```

2.Install MoveIt

Enter the following command in the terminal window to perform the installation of MoveIt:

```
sudo apt-get install ros-kinetic-moveit
```

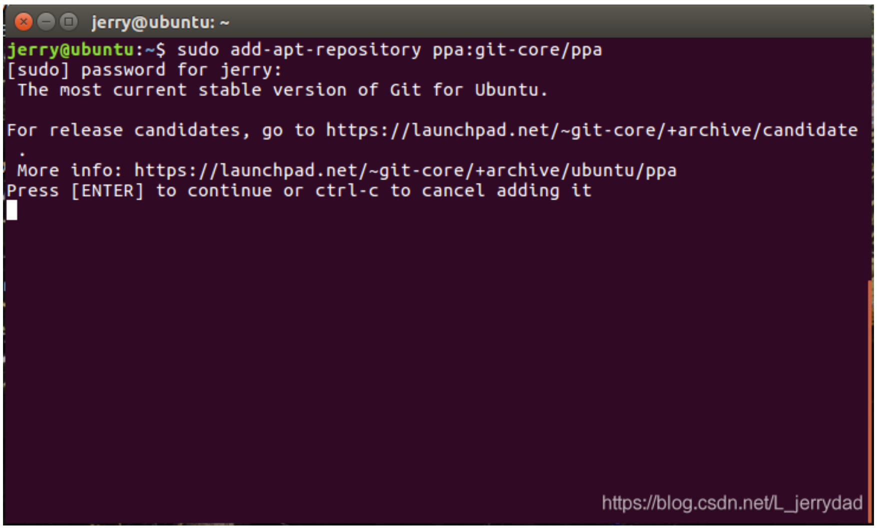
git installation

1.Add a software source

Add the git-installed software source to the list of ubuntu's software sources and enter the following command in the terminal window:

```
sudo add-apt-repository ppa:git-core/ppa
```

You need click`Enter` manually to continue.



```
jerry@ubuntu:~$ sudo add-apt-repository ppa:git-core/ppa
[sudo] password for jerry:
 The most current stable version of Git for Ubuntu.

For release candidates, go to https://launchpad.net/~git-core/+archive/candidate
 .
 More info: https://launchpad.net/~git-core/+archive/ubuntu/ppa
Press [ENTER] to continue or ctrl-c to cancel adding it
```

2.Update the list of software sources

Enter the following command in the terminal window to update the list of software sources:

```
sudo apt-get update
```

3.Install git

Enter the following command in the terminal window to perform the installation of git:

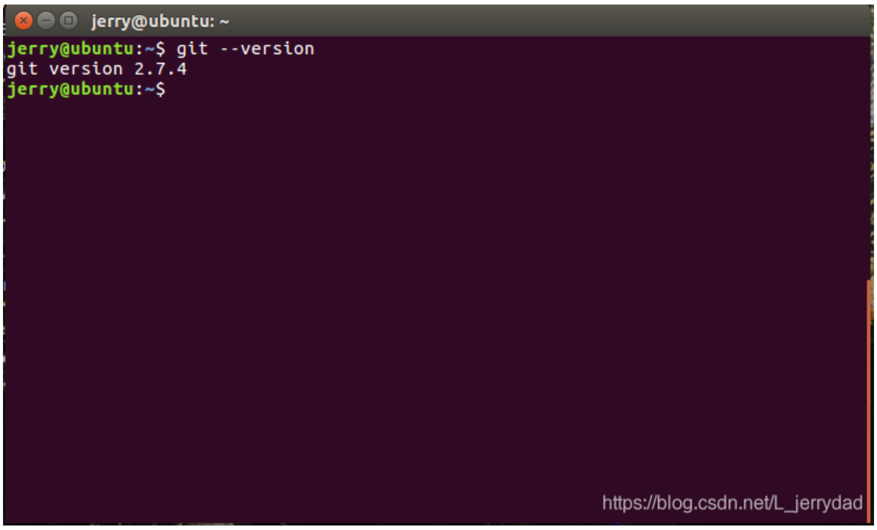
```
sudo apt-get install git
```

4.Verify and installation

Read git version, Enter the following command in the terminal window:

```
git --version
```

The git version number can be displayed in the terminal, as follows, for a successful installation.



```
jerry@ubuntu:~$ git --version
git version 2.7.4
jerry@ubuntu:~$
```

Introduction

`mycobot_ros` is from ElephantRobotics, fits its desktop six-axis robotic arm mycobot.

The project address：http://github.com/elephantrobotics/mycobot_ros

Installation

NOTE：

- This package relies on ROS and MoveIT. Ensure a successful installation of ROS and MoveIT before use.
- The interaction of this package with the real robot arm is depend on PythonApi - `pymycobot`
- The project site of Api is：https://github.com/elephantrobotics/pymycobot
- Fast installation： `pip install pymycobot --upgrade`
- The way you install it depends on Git, make sure Git is installed on your computer.

The following text will refer to the ROS workspace path in your computer with `<ros-workspace>` , and make sure that you replace `<ros-workspace>` with your native's real path when you execute the following command.

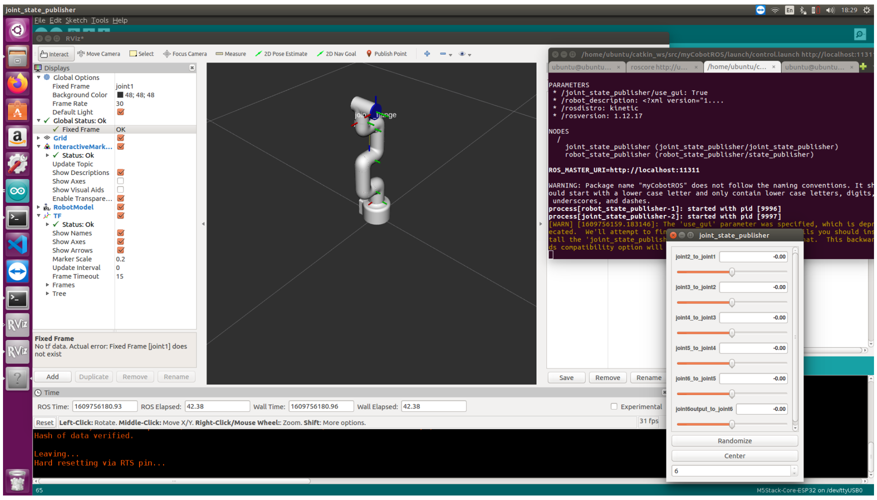
```
cd <ros-workspace>/src
git clone https://github.com/elephantrobotics/mycobot_ros.git
cd ..
catkin_make
source <ros-workspace>/devel/setup.bash
```

Slider control

Open a command line and run：

```
roslaunch mycobot_ros mycobot_slider.launch
```

It will open the rviz and a slider assembly, and you'll see as the following:



Global Options
Fixed Frame joint1
Background Color 48; 48; 48
Frame Rate 30
Default Light ☑
✓ Global Status: Ok
✓ Fixed Frame OK
Grid ☑
InteractiveMark... ☑
✓ Status: Ok
Update Topic
Show Descriptions ☑
Show Axes □
Show Visual Aids □
Enable Transpare... ☑
RobotModel ☑
TF ☑
✓ Status: Ok
Show Names ☑
Show Axes ☑
Show Arrows ☑
Marker Scale 0.2
Update Interval 0
Frame Timeout 15
Frames
Tree

Fixed Frame
No tf data. Actual error: Fixed Frame [joint1] does not exist

ROS Time: 1609756180.93 ROS Elapsed: 42.38 Wall Time: 1609756180.96 Wall Elapsed: 42.38 □ Experimental

Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options.

```
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

/home/ubuntu/catkin_ws/src/myCobotROS/launch/control.launch http://localhost:1131

```
PARAMETERS
 * /joint_state_publisher/use_gui: True
 * /robot_description: <?xml version="1....
 * /rosdistro: kinetic
 * /rosversion: 1.12.17

NODES
  /
    joint_state_publisher (joint_state_publisher/joint_state_publisher)
    robot_state_publisher (robot_state_publisher/state_publisher)

ROS_MASTER_URI=http://localhost:11311

WARNING: Package name "myCobotROS" does not follow the naming conventions. It sh
ould start with a lower case letter and only contain lower case letters, digits,
 underscores, and dashes.
process[robot_state_publisher-1]: started with pid [9996]
process[joint_state_publisher-2]: started with pid [9997]
[WARN] [1609756159.183146]: The 'use_gui' parameter was specified, which is depr
ecated.  We'll attempt to fin                              ils you should ins
tall the 'joint_state_publish                              hat.  This backwar
ds compatibility option will
```

joint_state_publisher

joint2_to_joint1	-0.00
joint3_to_joint2	-0.00
joint4_to_joint3	-0.00
joint5_to_joint4	-0.00
joint6_to_joint5	-0.00
joint6output_to_joint6	-0.00

6

You can then control the movement of the model in the rviz by dragging the slider. If you want the real mycobot to move with you, you need to open another command line and run:

```
rosrun mycobot_ros slider_control.py
#or
rosrun mycobot_ros slider_control.py _port:=/dev/ttyUSB0 _baud:=115200
```

It publishes the angle to mycobot in real time. The script supports the setting of port numbers and baud rates, with defaults of `"/dev/ttyUSB0"` and `115200` 。

Model follow

In addition to the controls above, we can also allow the model to follow the real robot arm movement. Open a command line to run:

```
rosrun mycobot_ros follow_display.py
#or
rosrun mycobot_ros follow_display.py _port:=/dev/ttyUSB0 _baud:=115200
```

It will release the angle of the real robotic arm. The script supports setting port numbers and baud rates, which are defaulted to `"/dev/ttyUSB0"` 和 `115200` 。

Then open another command line and run:

```
roslaunch mycobot_ros mycobot_follow.launch
```

It opens the rviz display model to follow the effect.

Keyboard control

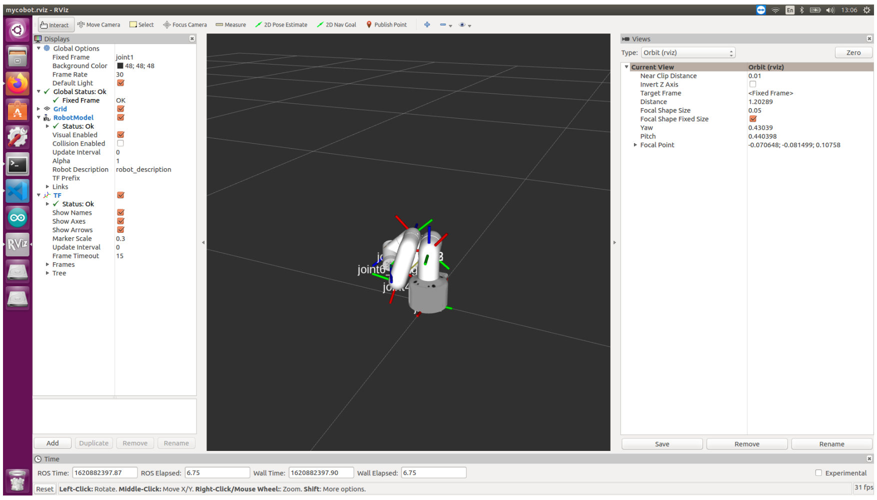
Keyboard control was added to the `mycobot_ros` package and synchronized in real time in rviz. This function relies on pythonApi, so make sure to connect to the real robot arm.

Open a command line and run：

```
roslaunch mycobot_ros mycobot_teleop_keyboard.launch
#or
roslaunch mycobot_ros mycobot_teleop_keyboard.launch port:=/dev/ttyUSB0 baud:=115200
```

Due to the need for real robotic arm communication, the launch supports the setting of port numbers and Baud rates, which default to `"/dev/ttyUSB0"` and `115200` .

The running resullts are as follows:



Global Options
Fixed Frame joint1
Background Color ■ 48; 48; 48
Frame Rate 30
Default Light ☑
✓ Global Status: Ok
✓ Fixed Frame OK
Grid ☑
RobotModel ☑
✓ Status: Ok
Visual Enabled ☑
Collision Enabled □
Update Interval 0
Alpha 1
Robot Description robot_description
TF Prefix
Links
TF ☑
✓ Status: Ok
Show Names ☑
Show Axes ☑
Show Arrows ☑
Marker Scale 0.3
Update Interval 0
Frame Timeout 15
Frames
Tree

Type: Orbit (rviz) Zero
Current View Orbit (rviz)
Near Clip Distance 0.01
Invert Z Axis □
Target Frame <Fixed Frame>
Distance 1.20289
Focal Shape Size 0.05
Focal Shape Fixed Size ☑
Yaw 0.43039
Pitch 0.440398
Focal Point -0.070648; -0.081499; 0.10758

ROS Time: 1620882397.87 ROS Elapsed: 6.75 Wall Time: 1620882397.90 Wall Elapsed: 6.75 □ Experimental

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift**: More options. 31 fps

The information of myCobot is output from the command line as follows:

```
SUMMARY
========

PARAMETERS
 * /mycobot_services/baud: 115200
 * /mycobot_services/port: /dev/ttyUSB0
 * /robot_description: <?xml version="1....
 * /rosdistro: kinetic
 * /rosversion: 1.12.17

NODES
  /
    mycobot_services (mycobot_ros/mycobot_services.py)
    real_listener (mycobot_ros/listen_real.py)
    robot_state_publisher (robot_state_publisher/state_publisher)
    rviz (rviz/rviz)

auto-starting new master
process[master]: started with pid [1333]
ROS_MASTER_URI=http://localhost:11311

setting /run_id to f977b3f4-b3a9-11eb-b0c8-d0c63728b379
process[rosout-1]: started with pid [1349]
started core service [/rosout]
process[robot_state_publisher-2]: started with pid [1357]
process[rviz-3]: started with pid [1367]
process[mycobot_services-4]: started with pid [1380]
process[real_listener-5]: started with pid [1395]
[INFO] [1620882819.196217]: start ...
[INFO] [1620882819.205050]: /dev/ttyUSB0,115200

MyCobot Status
--------------------------------
Joint Limit:
    joint 1: -170 ~ +170
    joint 2: -170 ~ +170
    joint 3: -170 ~ +170
    joint 4: -170 ~ +170
    joint 5: -170 ~ +170
    joint 6: -180 ~ +180

Connect Status: True

Servo Infomation: all connected

Servo Temperature: unknown

Atom Version: unknown

[INFO] [1620882819.435778]: ready
```

Next, open another command line and run:

```
rosrun mycobot_ros teleop_keyboard.py
#or
rosrun mycobot_ros teleop_keyboard.py _speed:=70
```

You'll see the following output on the command line:

```
Mycobot Teleop Keyboard Controller
---------------------------
Movimg options(control coordinations [x,y,z,rx,ry,rz]):
              w(x+)

    a(y-)     s(x-)     d(y+)

    z(z-) x(z+)

u(rx+)   i(ry+)   o(rz+)
j(rx-)   k(ry-)   l(rz-)

Gripper control:
    g - open
    h - close

Other:
    1 - Go to init pose
    2 - Go to home pose
    3 - Resave home pose
    q - Quit

currently:      speed: 50       change percent 5
```

The parameters supported by the script：

- `_speed` ： The movement speed of the robot arm
- `_change_percent` ： The percentage of the movement distance.

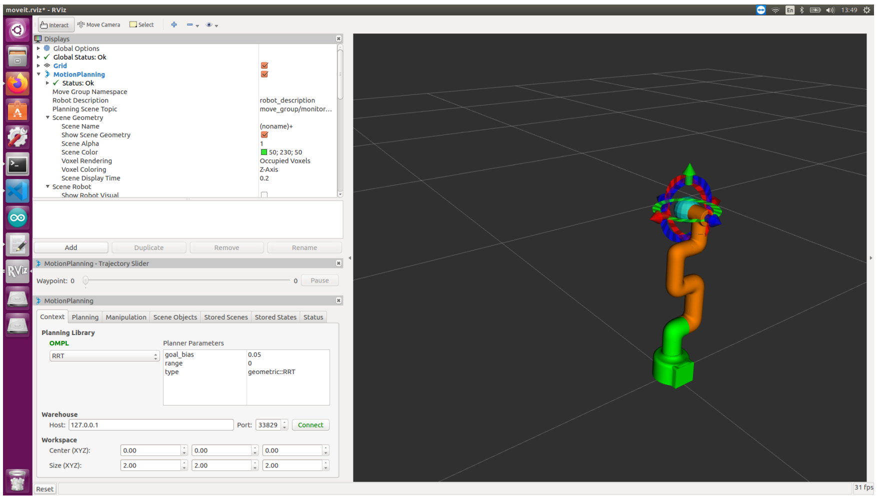
MoveIt

`mycobot_ros` is now integrated into the MoveIt section

Open the command line to run:

```
roslaunch mycobot_ros mycobot_moveit.launch
```

The running results are as follows:



Global Options
Global Status: Ok
Grid ☑
MotionPlanning ☑
Status: Ok
Move Group Namespace
Robot Description robot_description
Planning Scene Topic move_group/monitor...
Scene Geometry
Scene Name (noname)+
Show Scene Geometry ☑
Scene Alpha 1
Scene Color 🟩 50; 230; 50
Voxel Rendering Occupied Voxels
Voxel Coloring Z-Axis
Scene Display Time 0.2
Scene Robot
Show Robot Visual □

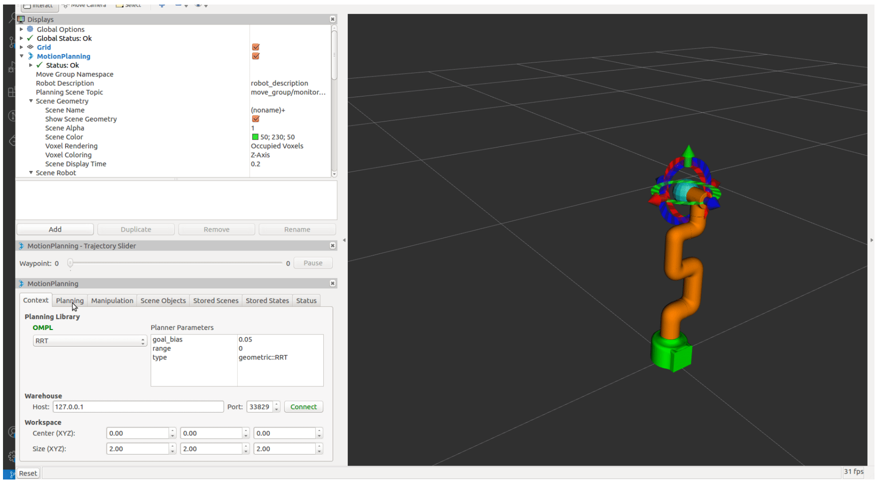
Waypoint: 0 0

Planning Library
OMPL
RRT
Planner Parameters
goal_bias 0.05
range 0
type geometric::RRT

Warehouse
Host: 127.0.0.1 Port: 33829

Workspace
Center (XYZ): 0.00 0.00 0.00
Size (XYZ): 2.00 2.00 2.00

You can plan and execute to demonstrate the effect:



Global Options
Global Status: Ok
Grid ☑
MotionPlanning ☑
Status: Ok
Move Group Namespace
Robot Description robot_description
Planning Scene Topic move_group/monitor...
Scene Geometry
Scene Name (noname)+
Show Scene Geometry ☑
Scene Alpha 1
Scene Color 🟩 50; 230; 50
Voxel Rendering Occupied Voxels
Voxel Coloring Z-Axis
Scene Display Time 0.2
Scene Robot

Waypoint: 0 0

Planning Library
OMPL
RRT
Planner Parameters
goal_bias 0.05
range 0
type geometric::RRT

Warehouse
Host: 127.0.0.1 Port: 33829

Workspace
Center (XYZ): 0.00 0.00 0.00
Size (XYZ): 2.00 2.00 2.00

If you need to synchronize the plan with the real robot arm, you need to open another command line and run:

```
rosrun mycobot_ros sync_plan.py
#or
rosrun mycobot_ros sync_plan.py _port:=/dev/ttyUSB0
```

The script supports setting port numbers and baud rates, which are defaulted to `"/dev/ttyUSB0"` and `115200` 。

> Please look forward to more...

RoboFlow

Download RoboFlow

1. Introduction to the RoboFlow

The RoboFlow is the operating system of the Elephant Collaborative Robot. It provides a human-computer interaction interface, which is convenient for operators to interact with the elephant robot and use the elephant robot correctly. That is to say, when the user uses the robot, most of the time is achieved by using the RoboFlow operating system.

For example, since the RoboFlow operating system runs in the teach pendant, the user can use the carrier of the teach pendant to perform manual robotics, programming, and other operations. The operating system OS can also be used to communicate with other robots or devices. All in all, with the advantages of friendly interface and rich functions, the appearance of the RoboFlow operating system makes it easier for users to start using the elephant robot. It makes everyone a commander of robots.

2 Main interface introduction

2.1User Login Interface

When the controller is powered up and the system startup button on the teach pendant is pressed, the login page is entered. Figure 2-1 shows the login page of the OS3 operating system.

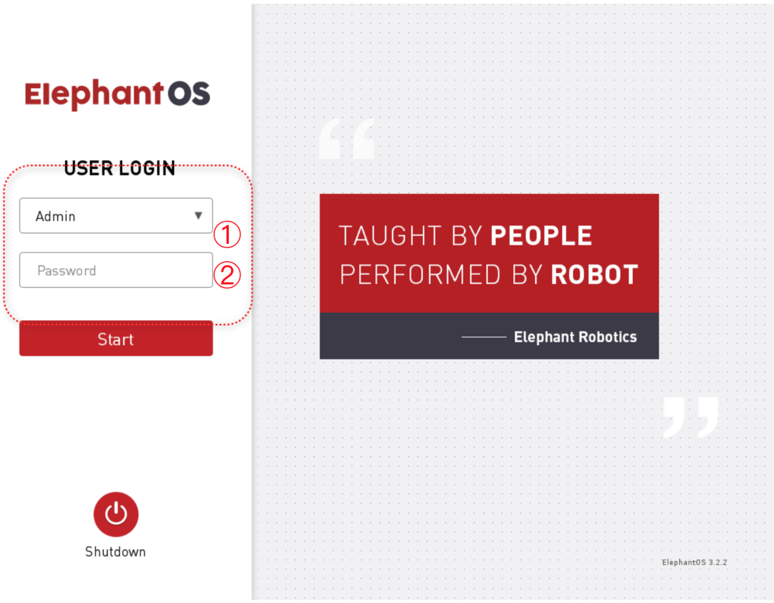


Figure 2- 1 login interface

As shown in the login page, "TAUGHT BY PEOPLE, PERFORMED BY ROBOT", this is the concept that the elephant robotics has always insisted on making the operator become the commander of the robot. Let robots replace people with simple but repetitive tasks, work in harsh working conditions, and work that people can't do well (such as scenes with very high operational accuracy).

There are two types of login users for the OS3 operating system, one is the administrator and the other is the operator. The administrator has the highest authority to perform all operations, programming and setup. The operator can only load and run existing programs and check the statistical data information.

Administrators can add and modify multiple accounts in the settings, including operator accounts.

By clicking on the "Shutdown" button, the OS3 operating system can be turned off, and then the power supply can be turned off, thus the robot system can be shut down.

2.2Main Menu

When the login is successful, it will go to the main menu page. The main menu of the OS3 operating system is shown in Figure 2-2.

Figure 2- 2 Main menu

On the left side of the main menu, there are four different options available:

- **Run Program**
  - Load an existing program directly and control the program to run. In this window, the user is not allowed to edit the program, but can only control the program running (such as control program running, pausing, stopping). At the same time, you can view the log and other related information during the running process of the program.
- **Edit Program**

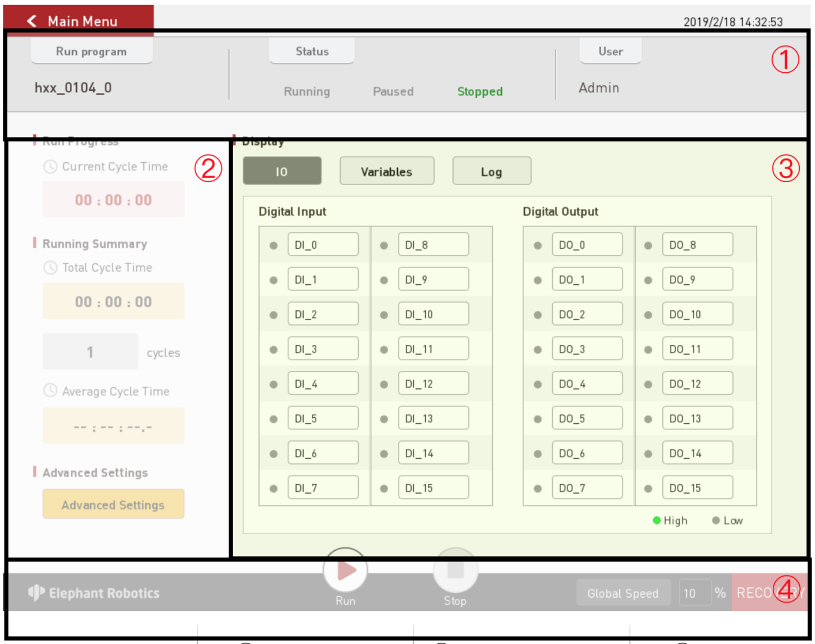
    - Users can choose to load an existing program in this window for modification, or they can choose to create a new blank program for editing.This window is the most frequently used function window for users. Besides programming, it can also perform other operations, such as manual manipulation of robots with "fast moving" function, forced control of IO signals, new variables, etc.
- **Statistics**
    - In this window, users can not only view the existing running data of the system, but also view related information saved before.
- **Settings**
    - In this window, the user can make basic settings for the robot. Such as robot open, robot off, account management, default program settings, etc.

In addition to these four main options, in the right window of the main menu, the user can see and open the most recently run program files. It is convenient for users to quickly find the most recently run program and control the program to run.

Click the "Shutdown" button to close the OS3 operating system; click the "Logout" button to log out.

2.3 Run Program

If the user selects "Run Program" in the main menu, it will enter the Run Program window. The running program window of the OS3 operating system is shown in Figure 2-3.



①- Running program basic information | ②- Operational statistics | ③- Display window | ④- Program run control bar

Figure 2- 3 Program editing window option

Users can enter the program window by loading the program they need to run. In this window, users can:

1.Get the basic information of the current (ready) running program, including program name, running status, user type.

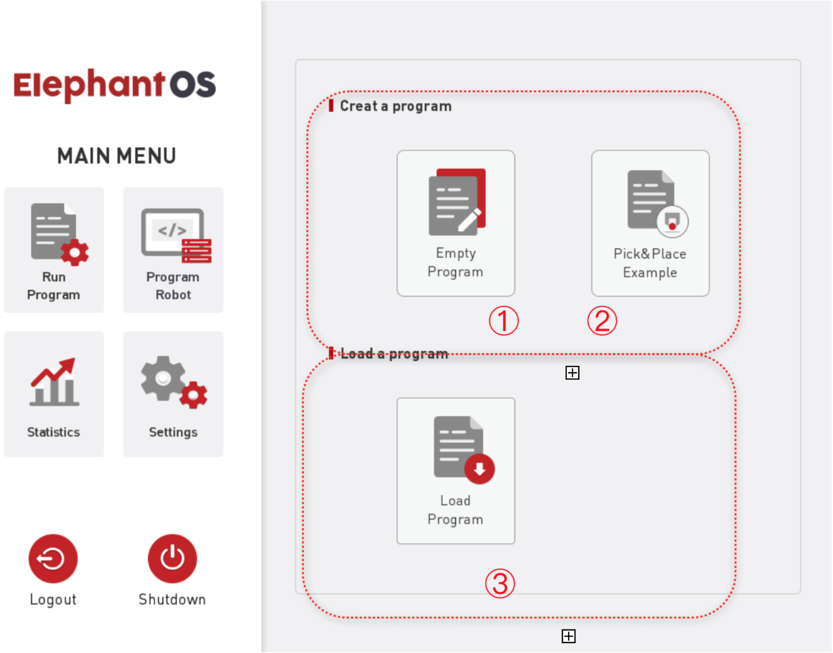
2.Understand the statistical information of the current running program, such as the total number of runs and the rhythm, etc.

3.Read the relevant information of the current running program through the display window, such as IO, variables, logs, etc.

4.The most important thing is that the running program window is the channel for the user to load and run the program that has been debugged.

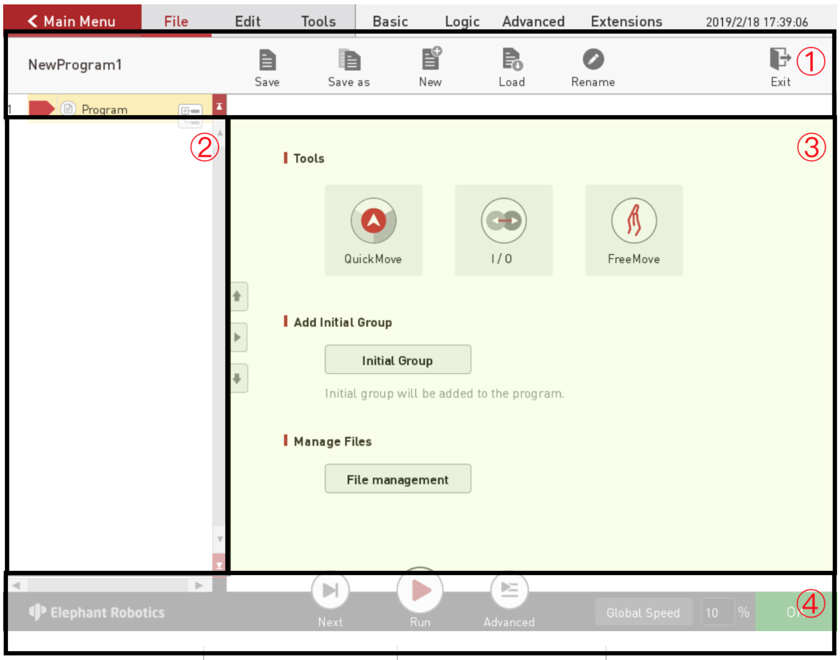
2.4Edit Program

As shown in Figure 2-4, if the user selects "Write Program" in the main menu, two options will appear in the right window. The first is to create a program (optional blank or template) and the second is to load the program.



①-Blank program | ②-Pick and place | ③-Load program

Figure 2- 4 Program Editing Window Options



Initial group will be added to the program.

①- Function bar | ②- Program Display Window | ③- Functional Editing Window | ④- Program Running Control Bar

Figure 2- 5 Program Editing Interface

When first entering the program edit page, the user sees the initial page as shown in figure 2-5. In this page, common tools, initialization group and file management functions are provided. The role of the initialization group is to make it easy for the user to set the program content to run at the beginning of the program and run only once. For example, set the initial point, Io State, and so on before the robot starts formal work. File management provides users with a way to manage files. Users can manage program files here, and can copy them to the U disk, or from the U disk to the system memory. If the user wants to go back to the initial page during the programming process, click "Back".

Function bar As shown in Figure 2-6, the function bar has seven sub-options, which are divided into two categories, one is the program editing toolbar, and the other is the function editing column.

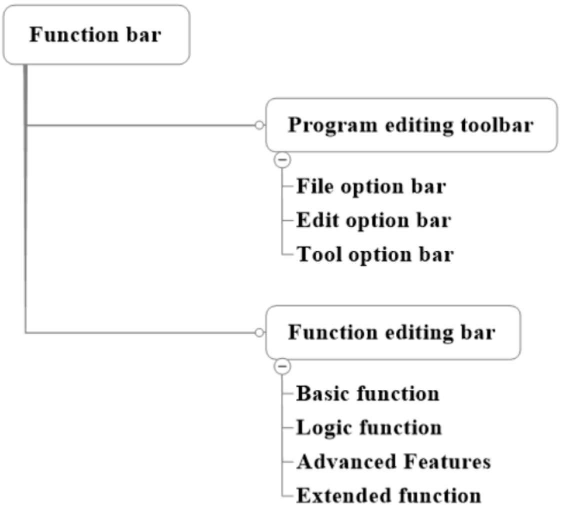


Figure 2- 6 Function bar

Figure 2- 6 Function bar

Program editing toolbar: Includes file option bar, edit option bar, and toll options bar. File：As shown in Figure 2-7, you can edit the program file. There are several operation options: Save, Save As, New, Load, Rename, and Exit.

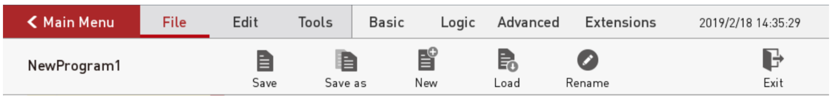


Figure 2- 7 File option bar

Edit：As shown in Figure 2-8, you can edit the specific command content in the program file. There are cut, copy, paste, delete, disable, delete all, redo, undo options.

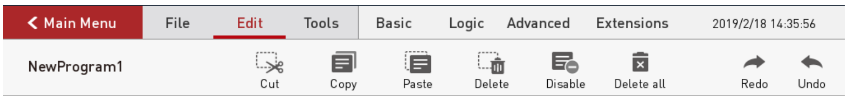


Figure 2- 8 Edit option bar

A. Tool options bar：As shown in Figure 2-9, it is a shortcut toolbar. When editing a robot program, the user often uses other tools to operate the robot. The tool options bar provides tools commonly used in program editing. Tools provided include: Quickmove, install, input and output, variables, logs, basic settings. For example, when editing a motion command, the user needs to manually operate the robot to a working position and teach the point. Then, the “Quickmove” tool in the toolbar can be selected to manually operate the robot to move to the position.

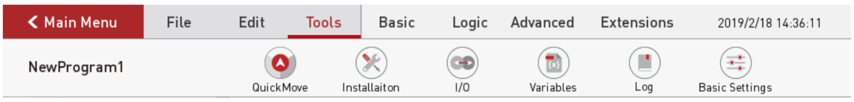


Figure 2- 9 Tool options bar

Functional Editing Window The OS3 operating system provides a rich set of features that allow users to perform complex functions with simple operations. Simple, but not simple functions, thus reducing the time workers to learn programming, efficient accomplish their goals. The function editing bar includes basic functions, logic functions, advanced functions, and extended functions.

Basic functions： As shown in Figure 2-10, the basic functions include Waypoint, Gripper, Wait, Set, and Group, which are some basic functions commonly used by users.

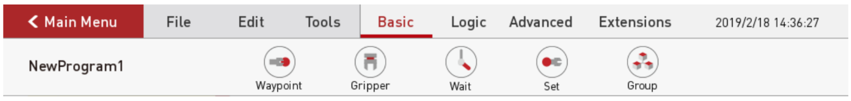


Figure 2- 10 Basic functions

- **Waypoint**: “Create new waypoints → Manually operate the robot to move the robot to the target point → Save current point → Running program”. With this series of operations, the user completes the goal of controlling the movement of the robot to the target point. If you create multiple waypoints, the motion of the robot will form a trajectory when you run the program.
- **Gripper**: The user can use this function to set the end effector. For example, it holds the workpiece or releases the workpiece.
- **Wait**: Users can use this function to delay, or wait for signals, conditions, and so on.
- **Set**: Users can use this function to set the input and output signals and custom conditions.
- **Group**: Users can use this function to edit the programs in the group.
- **Logic function**： As shown in Figure 2-11, the logic functions include Loop, If/Else, Subprogram, Thread, Halt, Switch, to complete the program running process control.

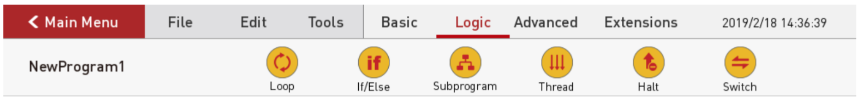


Figure 2- 11 Logic function

- **Loop**: The user can use this function to set a block to run cyclically multiple times.
- **If/Else**: The user can use this function to make conditional judgments, such as the determination of an input signal.
- **Subprogram**: The user can use this function to call a subroutine.
- **Thread**: Users can use this function to achieve robot multi-thread control.
- **Halt**: The user can use this function to control the program to pause, stop, restart, and pop up the window to display the corresponding prompt information.
- **Switch**: The user can use this function to make a condition selection and determine the content to be executed according to the value of the selected object.
- **Advanced function**: As shown in Figure 2-12, advanced functions include Pallet, Assign to Var, Script, Popup, and Sender, all of which perform more complex operations.

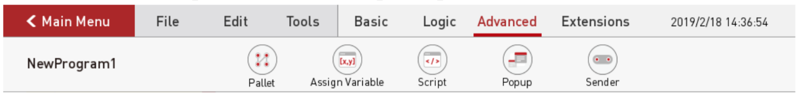


Figure 2- 12 Advanced function

- **Pallet**: Users can use this function to realize the robot to perform regular point movements. For example, the handling of workpieces in pallets, palletizing, etc. It is also possible to implement the fixed but irregular rendezvous motion of the robot in sequence.
- **Assign to Var**: Users can use this function to implement the assignment of a variable.
- **Script**: With the scripting feature, users can use the other common functions to achieve simple tasks while using the elephant robot, and can also use script programming to complete more complex tasks.
- **Popup**: Users can use this function to customize the pop-up window to display related information. This helps the operator to analyze the status of the current robot running program.
- **Sender**: Users can use this function to achieve TCP/IP communication between the elephant robot and other devices.
- **Extended function**: To adapt to different application scenarios, the OS3 operating system provides some extension functions, and even customizes functions according to important application scenarios proposed by users.

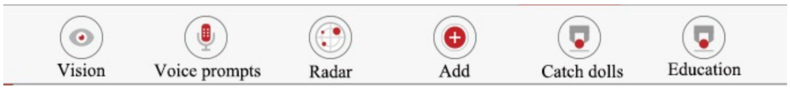
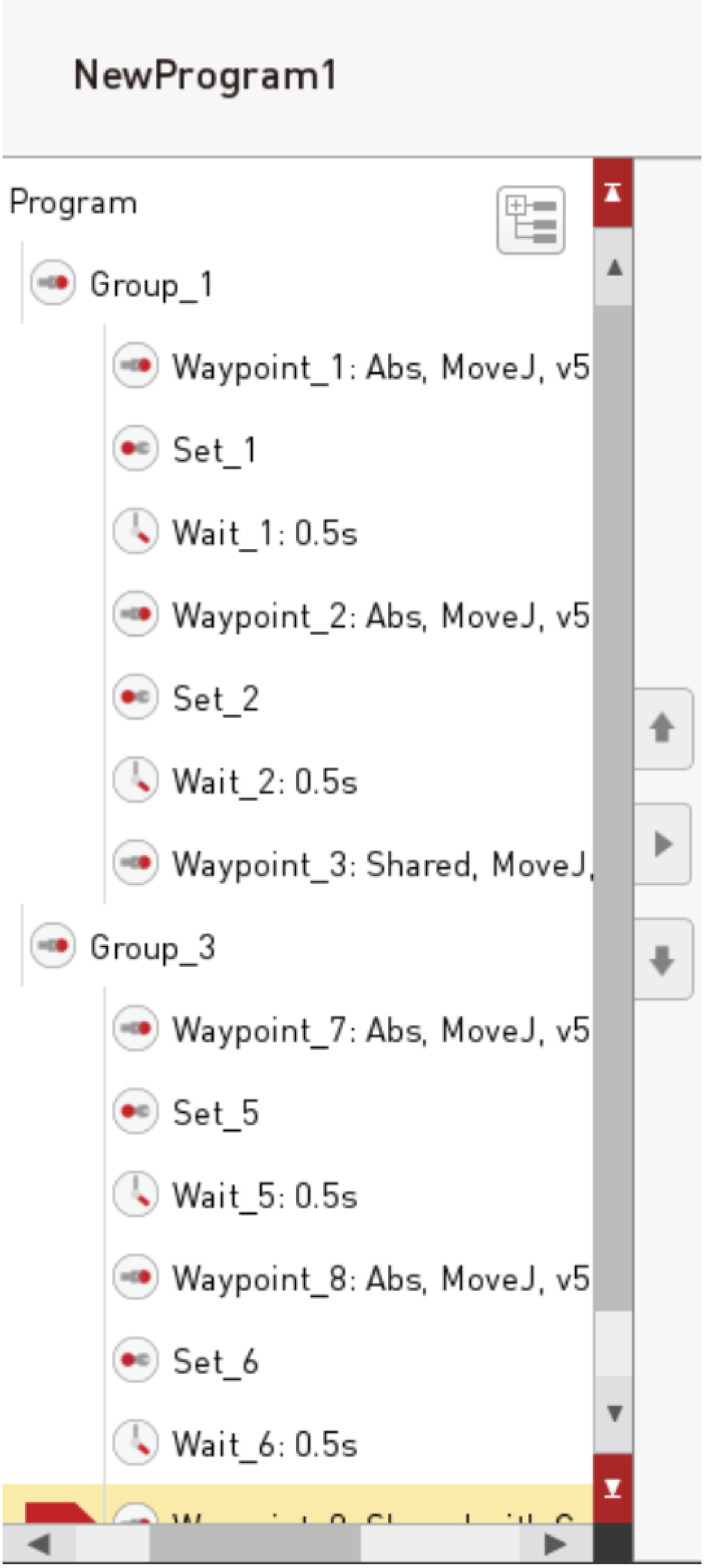


Figure 2- 13 Extended function

Program Display Window On the left side of the program editing page, there is a program display window as shown in Figure 2-14. The upper part is the name of the currently open program file, and the lower part is the program tree, which records the specific instructions and related information.



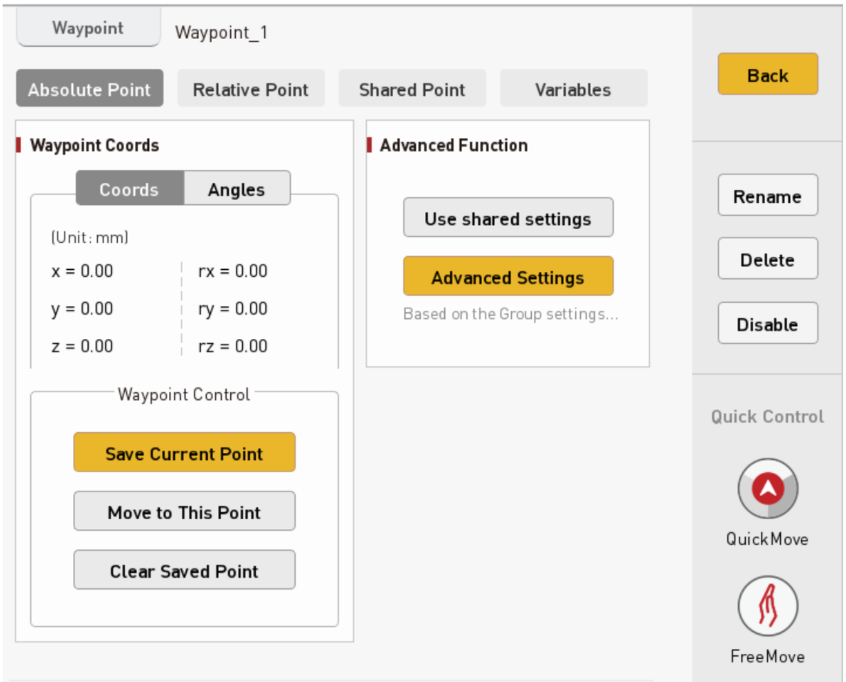
NewProgram1

Program

- Group_1
  - Waypoint_1: Abs, MoveJ, v5
  - Set_1
  - Wait_1: 0.5s
  - Waypoint_2: Abs, MoveJ, v5
  - Set_2
  - Wait_2: 0.5s
  - Waypoint_3: Shared, MoveJ,
- Group_3
  - Waypoint_7: Abs, MoveJ, v5
  - Set_5
  - Wait_5: 0.5s
  - Waypoint_8: Abs, MoveJ, v5
  - Set_6
  - Wait_6: 0.5s

Figure 2- 14 Program Display Window

On the right side of the program editing page, there is a function editing window as shown in Figure 2-15, which shows the specific contents of the function instructions.


Waypoint: Waypoint_1

(Unit: mm)

x = 0.00 | rx = 0.00
y = 0.00 | ry = 0.00
z = 0.00 | rz = 0.00

Based on the Group settings...

Figure 2- 15 Functional Editing Window

The user can make specific settings for the function instructions in this window. Quick control and current command renaming, deletion, and disabling are also provided here.

At the bottom of the program editing page, there is a program running control bar as shown in Figure 2-16. When debugging a program, users can use it to run, pause, stop and limit the running speed of the program.

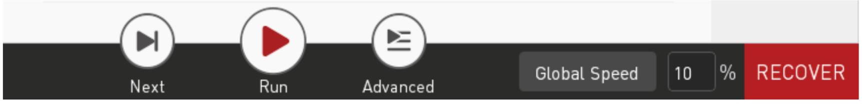

Global Speed 10 %

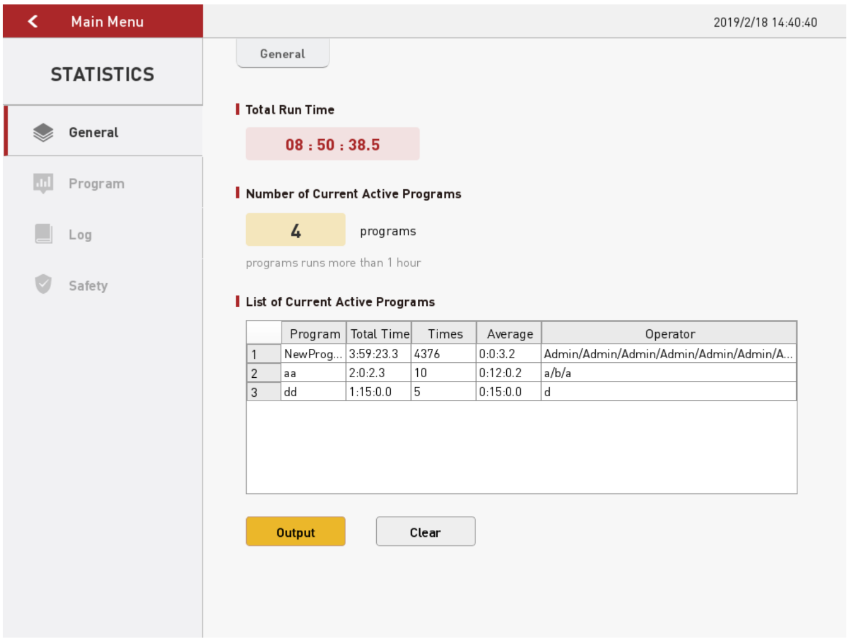
Figure 2- 16 Program run control bar

2.5Statistics

When users use elephant robots, they can not only program and control the robot to complete the corresponding tasks, but also get some valuable statistical data in the statistical report window for analysis and statistics.

The statistical report window is divided into four sub-windows.

As shown in Figure 2-17, the general class counts the total running time, the number of active programs, and the specific information of active programs.



Total Run Time

08 : 50 : 38.5

Number of Current Active Programs

4 programs

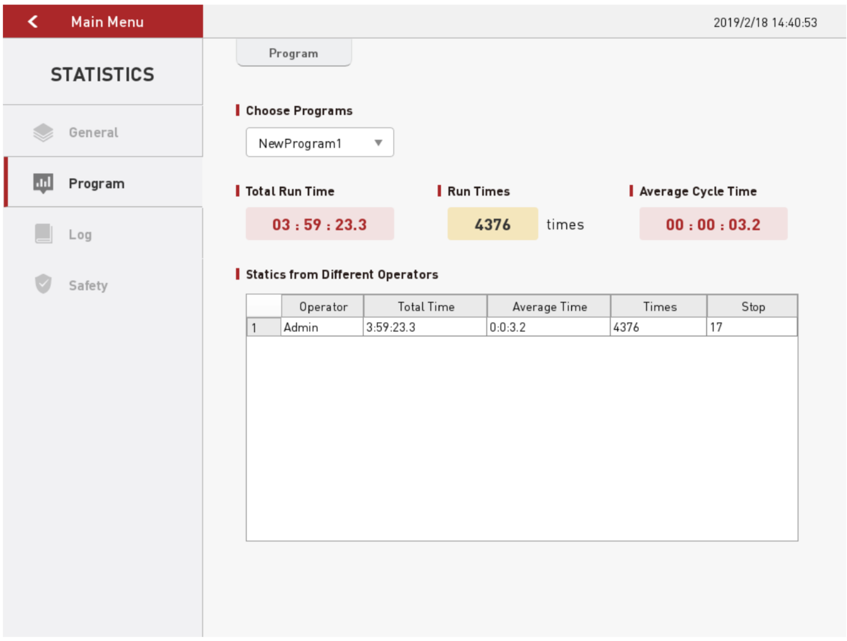
programs runs more than 1 hour

List of Current Active Programs

	Program	Total Time	Times	Average	Operator
1	NewProg...	3:59:23.3	4376	0:0:3.2	Admin/Admin/Admin/Admin/Admin/Admin/A...
2	aa	2:0:2.3	10	0:12:0.2	a/b/a
3	dd	1:15:0.0	5	0:15:0.0	d

Figure 2- 17 Conventional statistics

As shown in Figure 2-18, the program class counts the total running time and times of different programs.



Choose Programs

NewProgram1 ▼

Total Run Time

03 : 59 : 23.3

Run Times

4376 times

Average Cycle Time

00 : 00 : 03.2

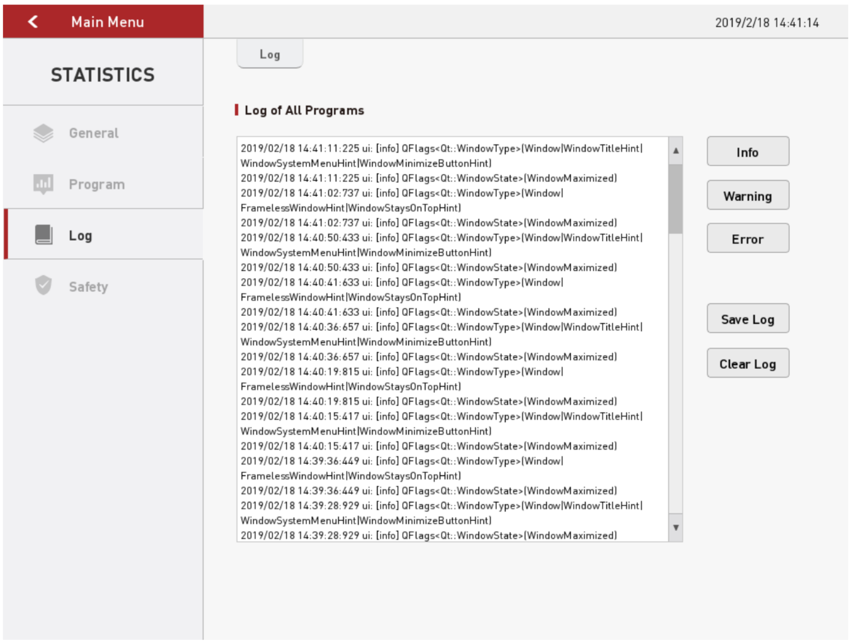
Statics from Different Operators

	Operator	Total Time	Average Time	Times	Stop
1	Admin	3:59:23.3	0:0:3.2	4376	17

Figure 2- 18 Procedural statistics

As shown in Figure 2-19, the log lists the general information, warning information and error information recorded by the system during the user's use of OS3 operating system. This information helps users to determine what changes and feedback the system has made during the operation of the OS3 operating system.

In particular, error information can help users quickly locate the possible causes of errors, so as to solve problems according to error information and resume normal use.

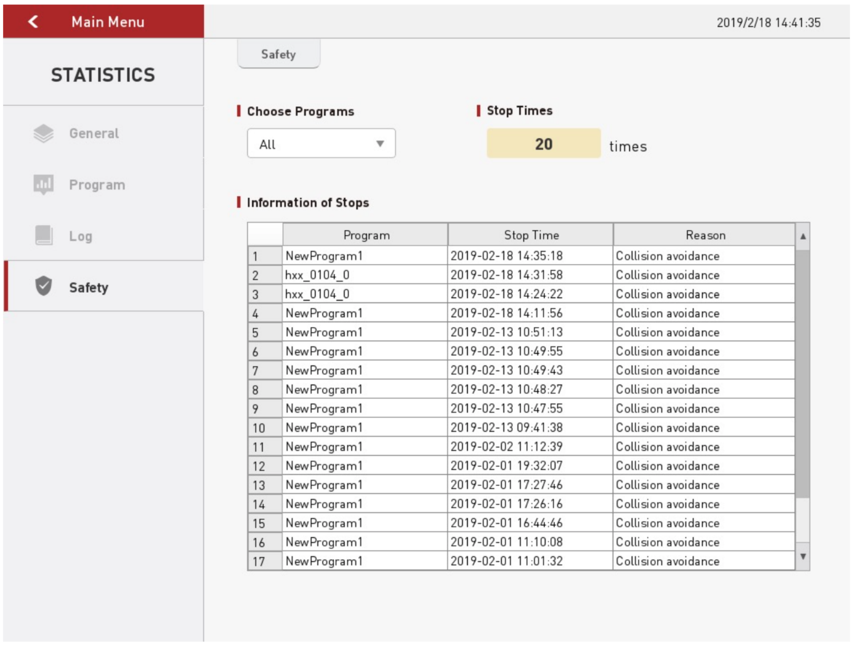


Log of All Programs

```
2019/02/18 14:41:11:225 ui: [info] QFlags<Qt::WindowType>(Window|WindowTitleHint|
WindowSystemMenuHint|WindowMinimizeButtonHint)
2019/02/18 14:41:11:225 ui: [info] QFlags<Qt::WindowState>(WindowMaximized)
2019/02/18 14:41:02:737 ui: [info] QFlags<Qt::WindowType>(Window|
FramelessWindowHint|WindowStaysOnTopHint)
2019/02/18 14:41:02:737 ui: [info] QFlags<Qt::WindowState>(WindowMaximized)
2019/02/18 14:40:50:433 ui: [info] QFlags<Qt::WindowType>(Window|WindowTitleHint|
WindowSystemMenuHint|WindowMinimizeButtonHint)
2019/02/18 14:40:50:433 ui: [info] QFlags<Qt::WindowState>(WindowMaximized)
2019/02/18 14:40:41:633 ui: [info] QFlags<Qt::WindowType>(Window|
FramelessWindowHint|WindowStaysOnTopHint)
2019/02/18 14:40:41:633 ui: [info] QFlags<Qt::WindowState>(WindowMaximized)
2019/02/18 14:40:36:657 ui: [info] QFlags<Qt::WindowType>(Window|WindowTitleHint|
WindowSystemMenuHint|WindowMinimizeButtonHint)
2019/02/18 14:40:36:657 ui: [info] QFlags<Qt::WindowState>(WindowMaximized)
2019/02/18 14:40:19:815 ui: [info] QFlags<Qt::WindowType>(Window|
FramelessWindowHint|WindowStaysOnTopHint)
2019/02/18 14:40:19:815 ui: [info] QFlags<Qt::WindowState>(WindowMaximized)
2019/02/18 14:40:15:417 ui: [info] QFlags<Qt::WindowType>(Window|WindowTitleHint|
WindowSystemMenuHint|WindowMinimizeButtonHint)
2019/02/18 14:40:15:417 ui: [info] QFlags<Qt::WindowState>(WindowMaximized)
2019/02/18 14:39:36:449 ui: [info] QFlags<Qt::WindowType>(Window|
FramelessWindowHint|WindowStaysOnTopHint)
2019/02/18 14:39:36:449 ui: [info] QFlags<Qt::WindowState>(WindowMaximized)
2019/02/18 14:39:28:929 ui: [info] QFlags<Qt::WindowType>(Window|WindowTitleHint|
WindowSystemMenuHint|WindowMinimizeButtonHint)
2019/02/18 14:39:28:929 ui: [info] QFlags<Qt::WindowState>(WindowMaximized)
```

Figure 2- 19 Log statistics

As shown in Figure 2-20, security statistics can help users to count security-related information, such as collision information, number of stops, etc.



Choose Programs: All

Stop Times: 20 times

Information of Stops

	Program	Stop Time	Reason
1	NewProgram1	2019-02-18 14:35:18	Collision avoidance
2	hxx_0104_0	2019-02-18 14:31:58	Collision avoidance
3	hxx_0104_0	2019-02-18 14:24:22	Collision avoidance
4	NewProgram1	2019-02-18 14:11:56	Collision avoidance
5	NewProgram1	2019-02-13 10:51:13	Collision avoidance
6	NewProgram1	2019-02-13 10:49:55	Collision avoidance
7	NewProgram1	2019-02-13 10:49:43	Collision avoidance
8	NewProgram1	2019-02-13 10:48:27	Collision avoidance
9	NewProgram1	2019-02-13 10:47:55	Collision avoidance
10	NewProgram1	2019-02-13 09:41:38	Collision avoidance
11	NewProgram1	2019-02-02 11:12:39	Collision avoidance
12	NewProgram1	2019-02-01 19:32:07	Collision avoidance
13	NewProgram1	2019-02-01 17:27:46	Collision avoidance
14	NewProgram1	2019-02-01 17:26:16	Collision avoidance
15	NewProgram1	2019-02-01 16:44:46	Collision avoidance
16	NewProgram1	2019-02-01 11:10:08	Collision avoidance
17	NewProgram1	2019-02-01 11:01:32	Collision avoidance

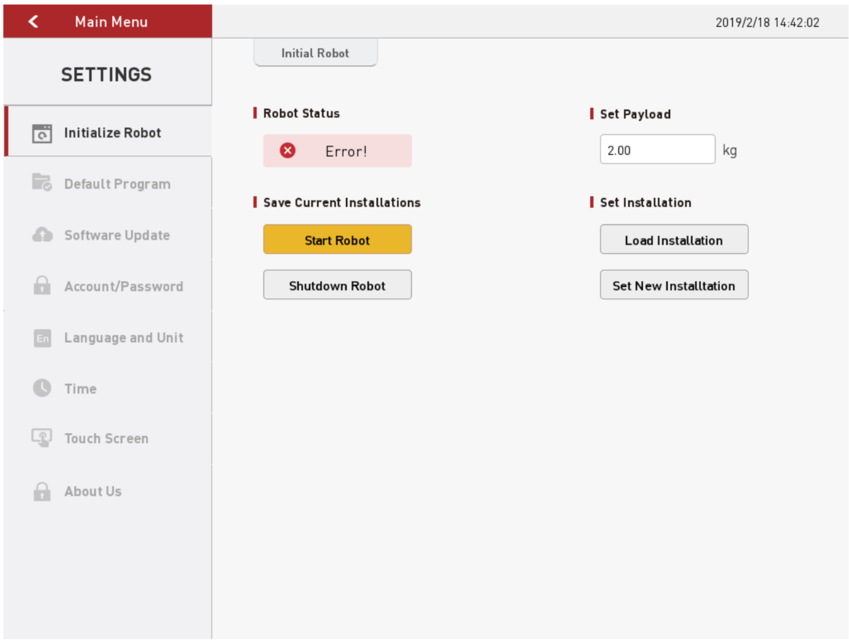
Figure 2- 20 Security statistics

2.6Settings

In the configuration center, users can configure the robot. For example, power the robot, turn off the robot, set the load, time, network and so on. Initialization

The initialization configuration page is shown in Figure 2-21.

When robot movement is required, the user needs to enter the configuration center → initialize the robot, or shut down the robot. In the initialization page, you can also set the load and installation, these two are important configuration content before other operations, such as configuration errors may cause unexpected situations.



2019/2/18 14:42:02

Robot Status: Error!

Set Payload: 2.00 kg

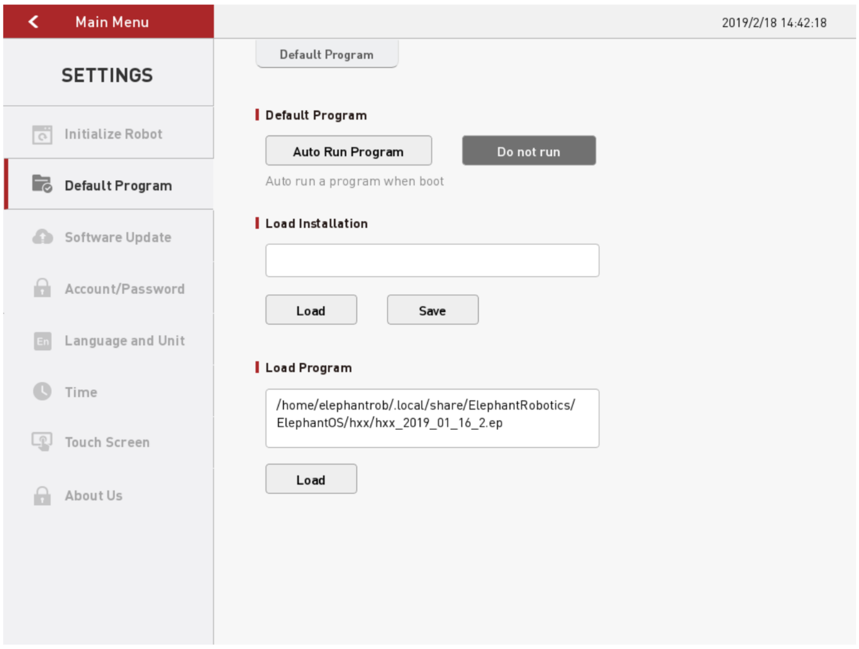
Save Current Installations

Set Installation

Figure 2- 21 Initialization

Default program

Figure 2-22 shows the default program settings page.



2019/2/18 14:42:18

Default Program

Auto run a program when boot

Load Installation

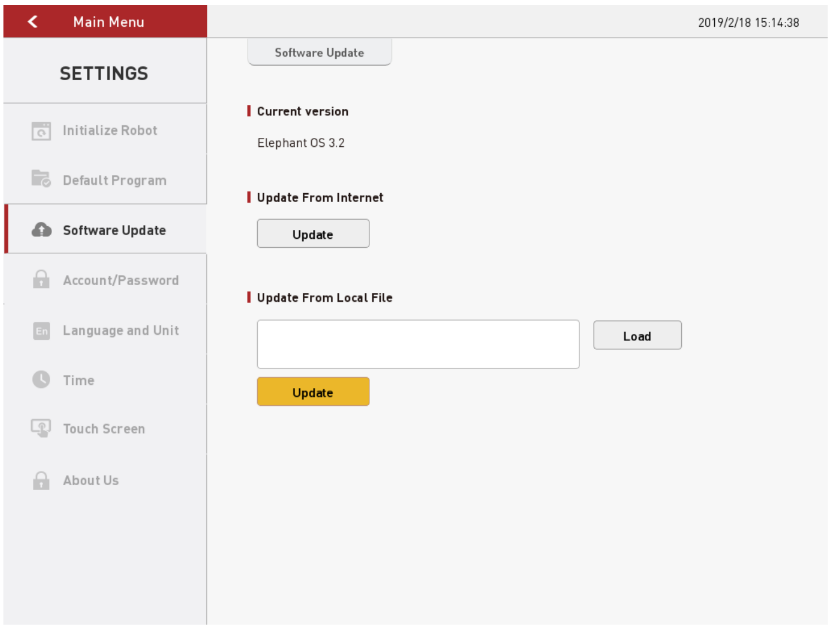
Load Program: /home/elephantrob/.local/share/ElephantRobotics/ElephantOS/hxx/hxx_2019_01_16_2.ep

Figure 2- 22 Default program

This function allows the user to set a default running program. As long as the system starts, the robot directly enters the running program window, and can start running the program and perform corresponding actions to complete the specified task.

If the user does not want the system to start and the startup program starts running, you can choose not to run.

Version update Figure 2-23 shows the version update settings page.

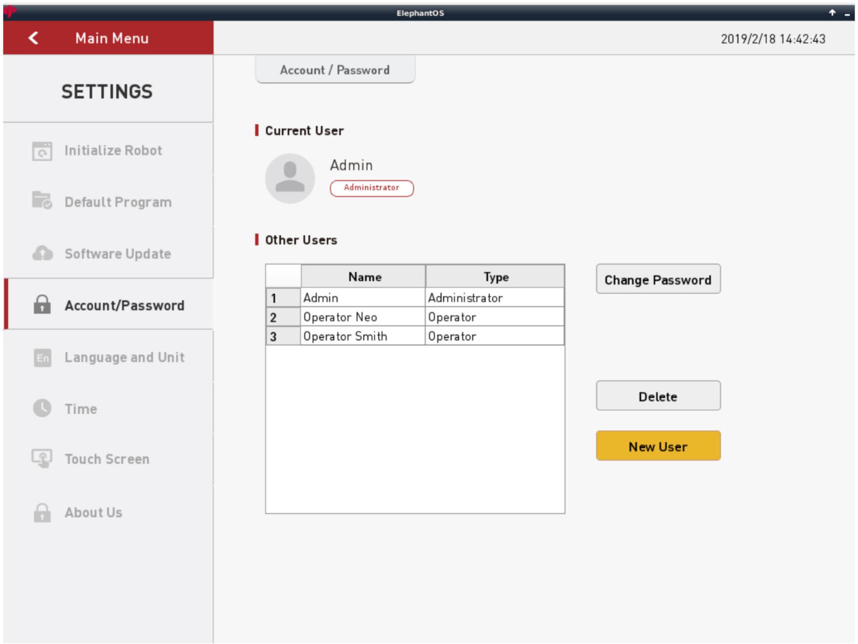


Current version

Elephant OS 3.2

Figure 2- 23 Version update

This page allows users to update the OS3 operating system in two ways, one for local file updates and one for network updates. Figure 2-24 shows the account management page.



Current User

Admin
Administrator

Other Users

	Name	Type
1	Admin	Administrator
2	Operator Neo	Operator
3	Operator Smith	Operator

Figure 2- 24 Account management

Users can add new users, delete expired users, or change passwords on this page. On this page, the user can get all the account information. Language and unit The language and unit settings page are shown in Figure 2-25. At present, the OS3 operating system supports Chinese and English and metric units. Other languages and units are increasing, so stay tuned!

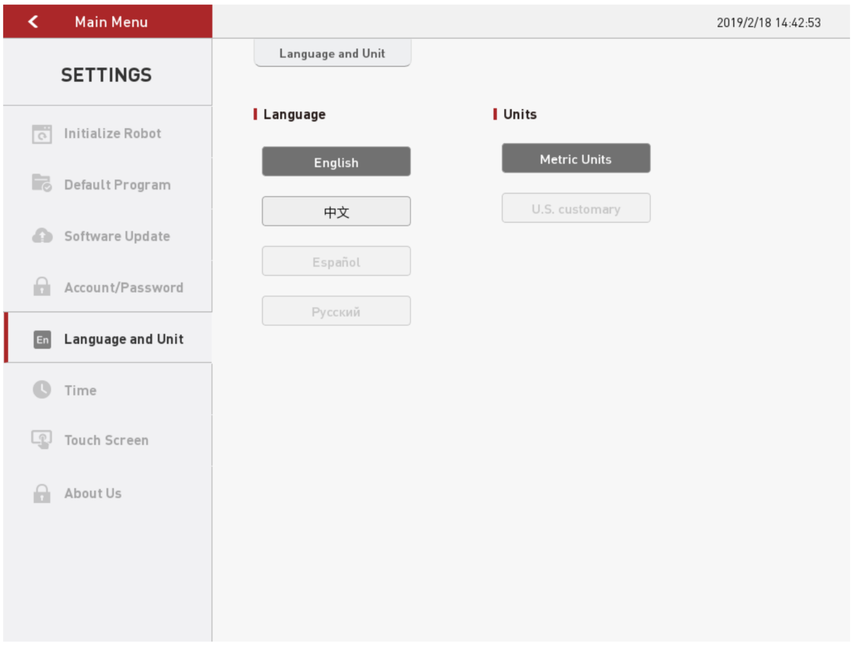


Figure 2- 25 Language and unit

Time Figure 2-26 shows the time setting page.

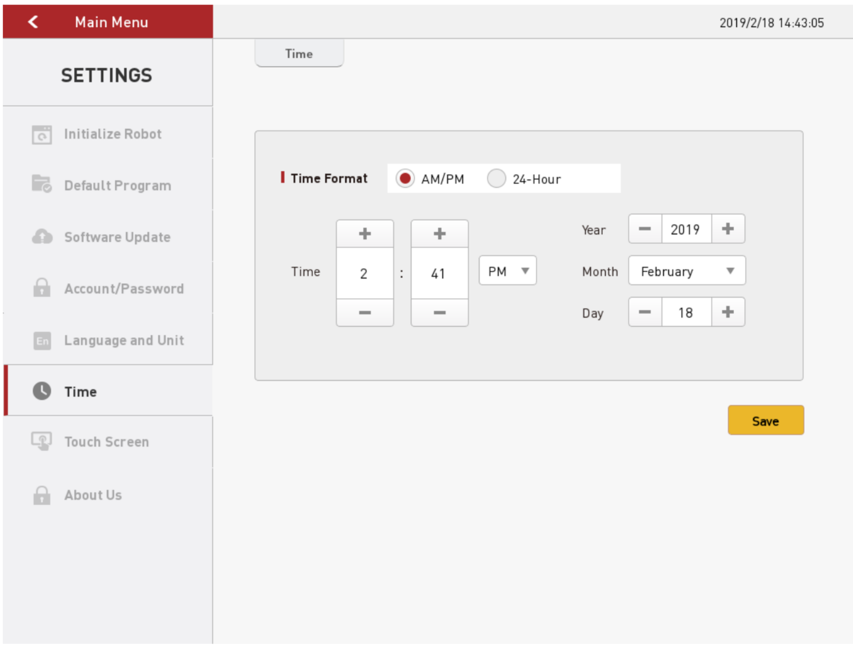
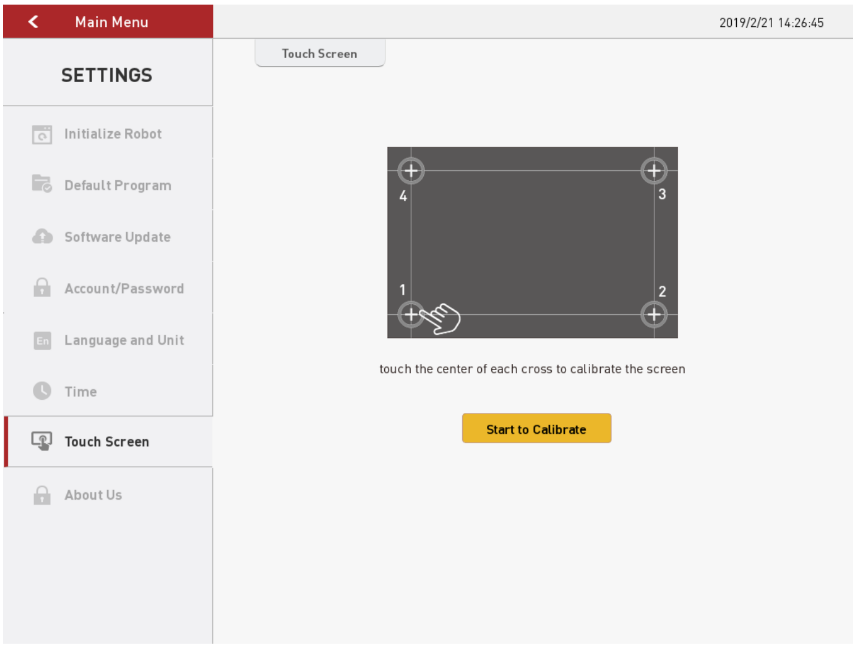


Figure 2- 26 Time

The user can set the system time on the current page. If the "24-hour system" is not checked, the time display format defaults to 12-hour system. Touch screen calibration

Figure 2-27 shows the touch screen calibration instructions. The user clicks on "Start to Calibrate" to enter the calibration interface. The calibration interface will appear in sequence with four circles, as shown in the figure. The user needs to click the center of the circle with a touch pen, and each time the button is clicked, the next circle will appear until all four circles appear. A pop-up window will appear indicating that the calibration is complete, and you can exit the calibration screen after confirming the pop-up.

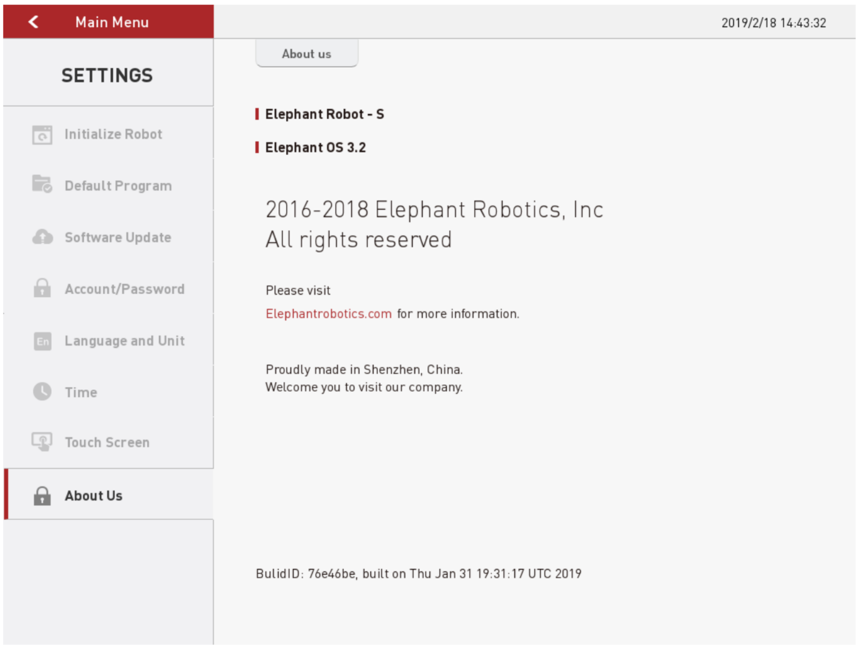
If the calibration times out or the steps are wrong, a pop-up prompts the calibration failure. At this point, you can confirm to exit the calibration interface and return to the page in Figure 2-27 to recalibrate.



touch the center of each cross to calibrate the screen

图2- 27 Touch Screen

About us As shown in Figure 2-28, it is about our page.



Elephant Robot - S

Elephant OS 3.2

2016-2018 Elephant Robotics, Inc
All rights reserved

Please visit
Elephantrobotics.com for more information.

Proudly made in Shenzhen, China.
Welcome you to visit our company.

BulidID: 76e46be, built on Thu Jan 31 19:31:17 UTC 2019

Figure 2- 28 About us

This page shows basic information about the operating system of the OS3 operating system. For example, the model of the robot used is the Elephant series, version information, and so on.

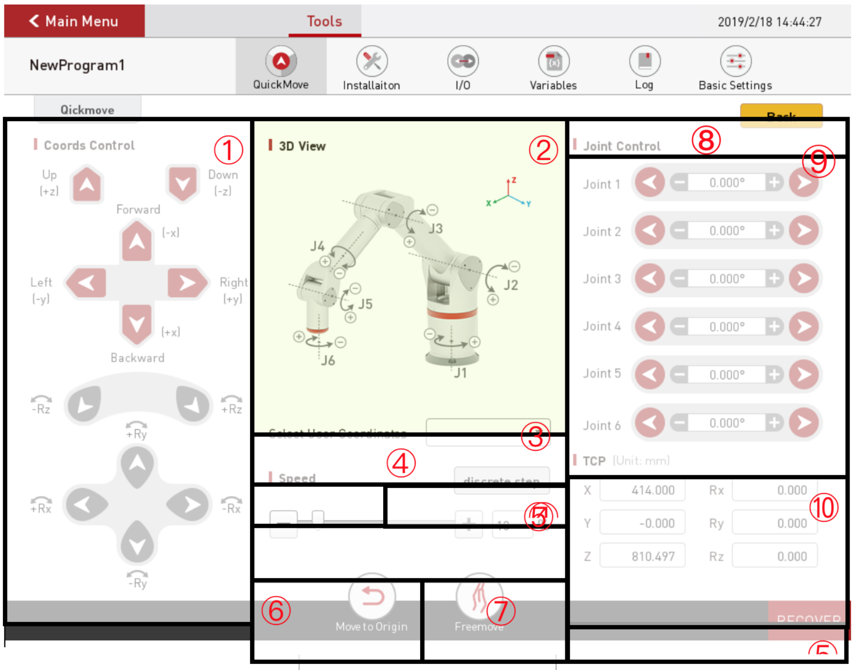
For more information, please visit the official website https://www.elephantrobotics.cn。

3 Introduction to common tools

3.1Quickmove

Quickmove is a tool that users use frequently when they operate the robot quickly and manually. Therefore, every user must be very familiar with the use of Quickmove using methods. The wrong operation may result in damage to the robot and its peripheral equipment, and even injuries to personnel.

As shown in Figure 3-1, Quickmove are mainly composed of 11 parts, which are described below.



①- Coordinate control	②- 3D View	③- User coordinate system
④- Stepping motion	⑤- Speed	⑥- Move to origin
⑦- Freemove	⑧-Return	⑨- Joint control
⑩- Coordinate position	○11- Status display button	

Figure 3- 1 Quickmove

Motion Control Mode in Cartesian Coordinate System

As shown in Figure 3-2, over-fixed-point O, three axes perpendicular to each other, all with O as the origin and generally with the same unit of length. These three axes are called x-axis (horizontal axis), y-axis (vertical axis), and z-axis (vertical axis), which are collectively referred to as coordinate axes. The x-axis and y-axis are usually arranged on a horizontal plane, while the z-axis is a plumb line. Their positive direction is in accordance with the right-hand rule, that is, holding the z-axis with the right hand. When the four fingers of the right hand turn from the positive x-axis to the positive y-axis from the π/2 angle, the thumb is pointed to the positive direction of the z-axis. Such three axes form a spatial Cartesian coordinate system, and point O is called the coordinate origin. This constitutes a Cartesian coordinate.

There are three planes in the three-dimensional Cartesian coordinate system, XY-plane, YZ-plane, and XZ-plane. These three planes divide the three-dimensional space into eight parts, called octant spaces. The three coordinates of each point of the first limit are positive values.

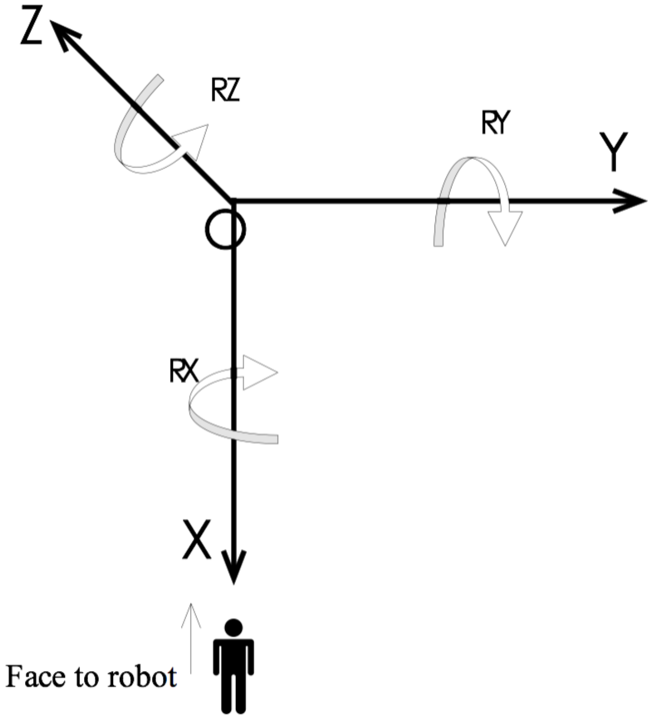


Figure 3- 2 Cartesian coordinate system Direction callout diagram

As shown in Figure 3-3, the robot can be controlled to move in the direction of the Cartesian coordinate system by clicking the key corresponding to the direction of the Cartesian coordinate system.

Figure 3- 3 Cartesian coordinate system motion control mode button

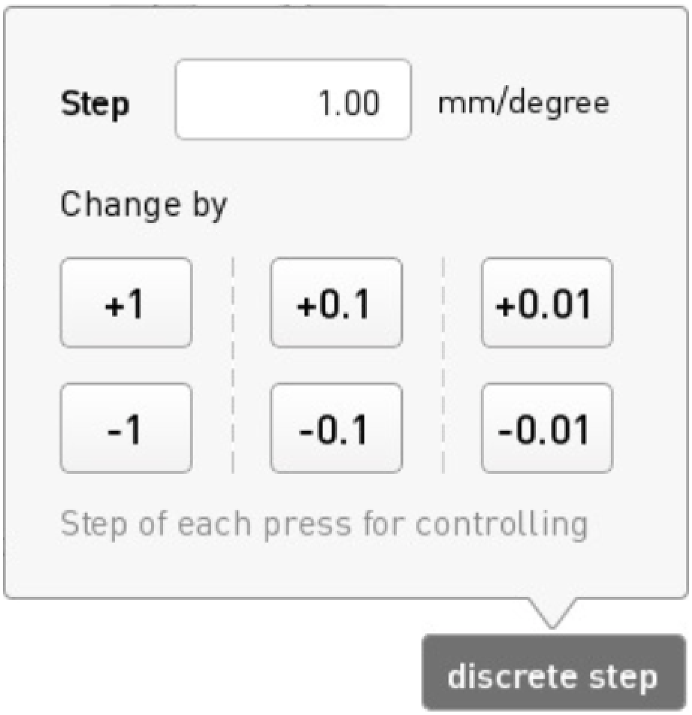
3D View This window marks the direction of movement of the six joints of the robot.

- User coordinate system
- Motion mode switching

There are two main motion modes for manual manipulation robots.

- **Continuous motion mode**: The user presses the motion control button and allows the robot to move until the user releases the button and the robot stops. For example, if you press the + X direction motion control button, you need to hold the button all the time. The time of pressing the motion control key determines the distance of the robot in the + X direction.
- **Stepping motion mode**: Manual manipulation robot step motion, click "step motion" and open the step setting window as shown in Figure 3-4. Then the user chooses the step in this window and clicks the key of the target control

direction. Every time he clicks, the robot takes a step. For example, choose a 1 mm step, click the X-direction movement control button, and every time you click the button, the robot will move 1 mm in the + X direction.



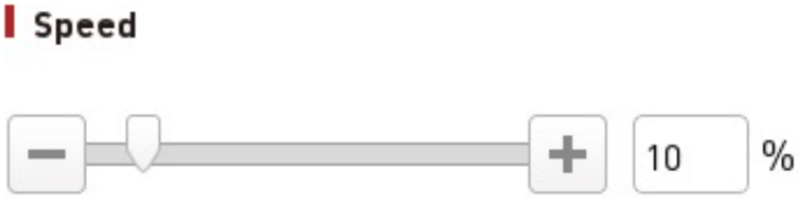
Step 1.00 mm/degree

Change by

Step of each press for controlling

Figure 3- 4 Step-by-step motion step setting window

Speed As shown in Figure 3-5, the control speed of the manual manipulator can be set here. Speed can be set from 0 to 100%.



Speed 10 %

Figure 3- 5 Speed setting window

Move to origin By selecting the icon shown in Figure 3-6, the robot can be controlled to return to its original position and posture.

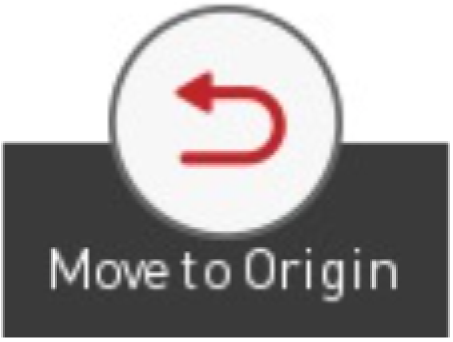


Figure 3- 6 Move to origin

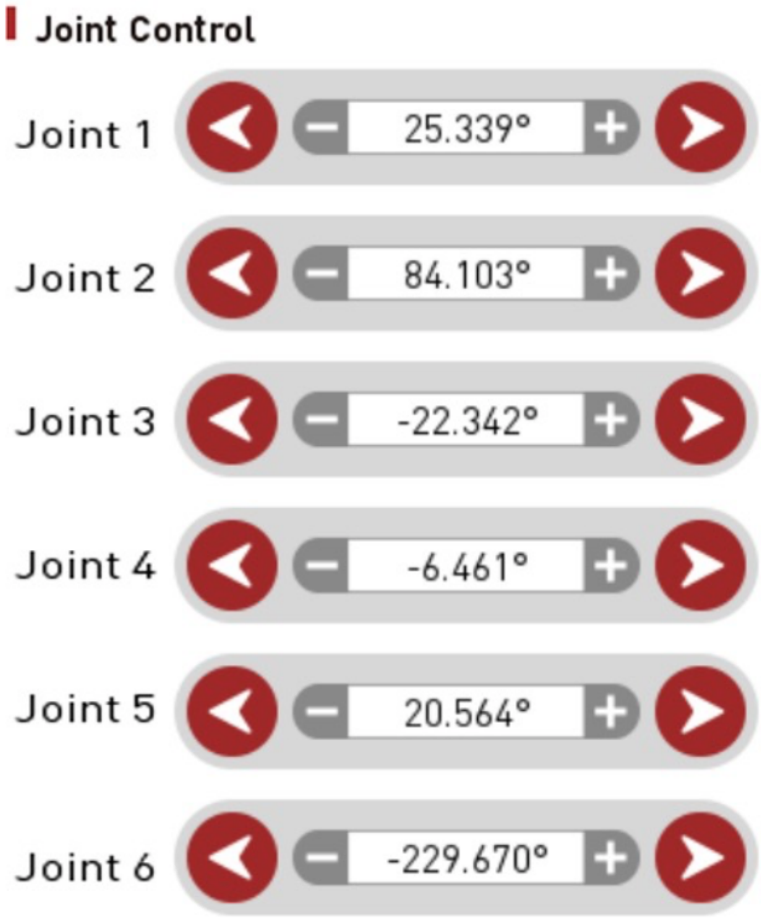
Freemove Select the icon shown in Figure 3-7 to switch to the drag mode.

Figure 3- 7 Freemove

Return Click on the icon shown in Figure 3-8 to return to the programming operation window.

Figure 3- 8 Return

Joint control Serial robot is an open kinematic chain of the robot. It is formed by a series of connecting rods connected in series with a rotating joint or a moving joint. The elephant cooperative robot belongs to a 6-axis serial robot. It drives the relative motion of the connecting rod by using motor drivers to drive the movement of 6 joints, allowing the end operator to reach the right posture. The Joint control window shown in Figure 3-9 provides the keys used by the operator to manually manipulate the robot and control the robot for joint movement using the instructor. The control buttons for each joint are divided into 2 directions, and the angle data of each axis can be seen.

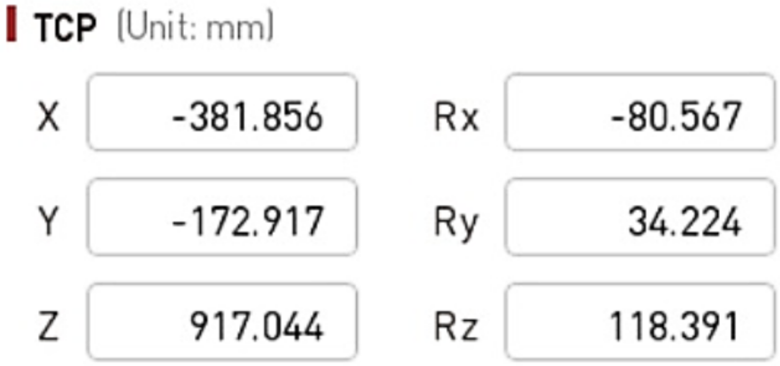


Joint Control

Joint 1	25.339°
Joint 2	84.103°
Joint 3	-22.342°
Joint 4	-6.461°
Joint 5	20.564°
Joint 6	-229.670°

Figure 3- 9 Joint motion mode control window

Coordinate position As shown in Figure 3-10, this window displays the coordinate position corresponding to the coordinate control.



TCP (Unit: mm)

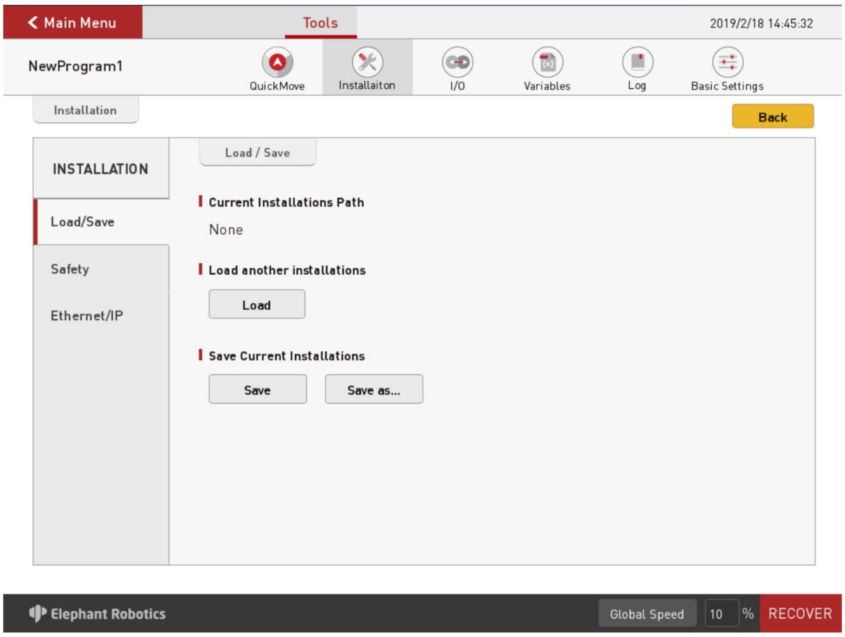
X	-381.856	Rx	-80.567
Y	-172.917	Ry	34.224
Z	917.044	Rz	118.391

Figure 3- 10 Coordinate position display window

Status display button: The button has two states, "OK" (displays green) and "Reset" (displays red). When the display is normal, it indicates that the robot is working properly, and when the reset is displayed, the robot is abnormal, the anomaly needs to be lifted and the key is clicked for reset.

3.2Installation

As shown in Figure 3-11, there are three submenus inside the installation tool. It is used to implement the loading/saving installation configuration, security configuration, and network configuration of the elephant robot.



Current Installations Path

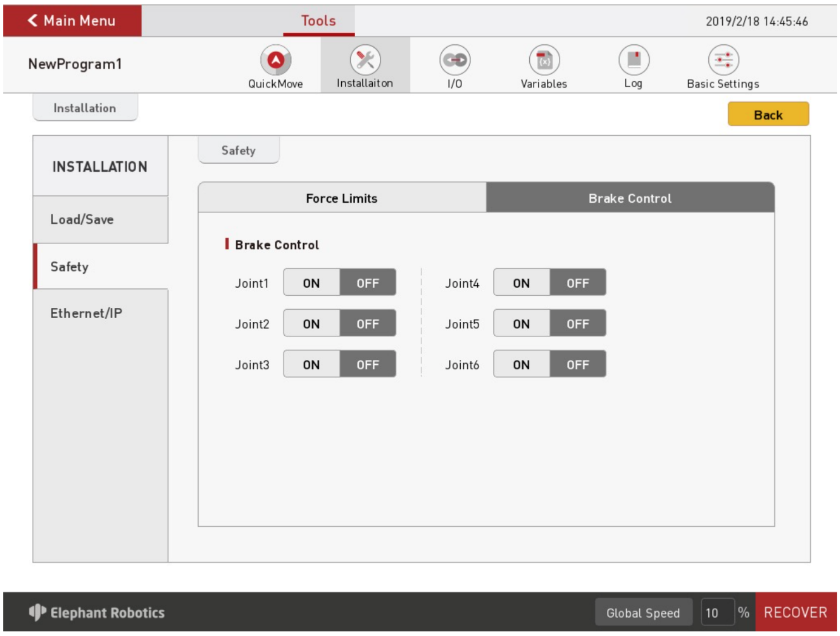
None

Load another installations

Save Current Installations

Figure 3- 11 Load/save installation

Security configuration: As shown in Figure 3-12, set the torque limit and brake control of the elephant robot.

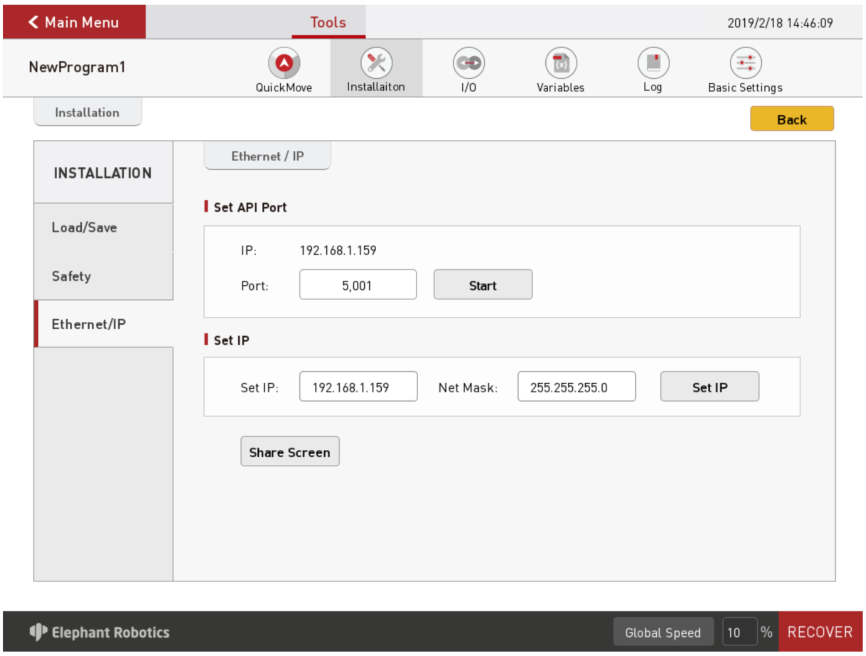


Brake Control

Joint1	ON OFF	Joint4	ON OFF
Joint2	ON OFF	Joint5	ON OFF
Joint3	ON OFF	Joint6	ON OFF

Figure 3- 12 Security configuration

Network Configuration: As shown in Figure 3-13, configure the IP address and port number of the Ethernet communication here.



IP: 192.168.1.159

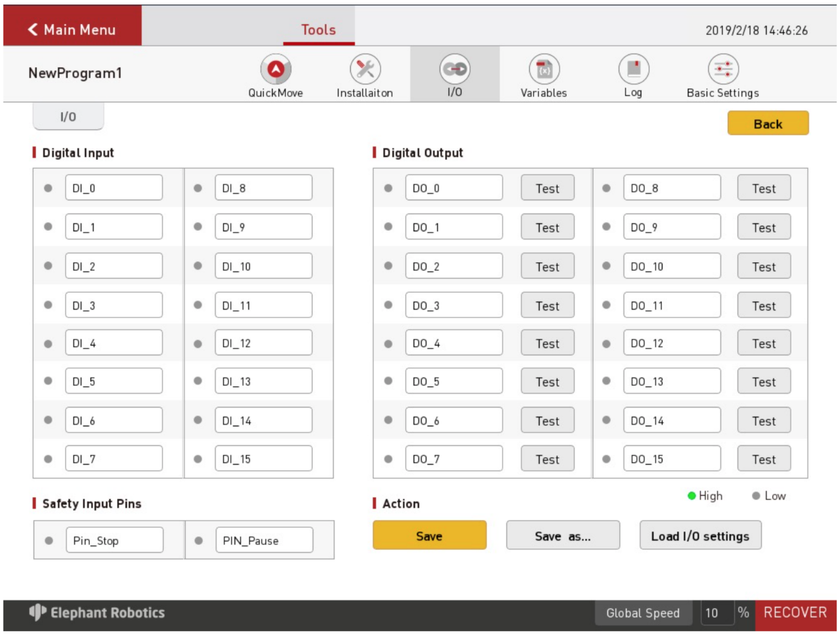
Port: 5,001

Set IP: 192.168.1.159 Net Mask: 255.255.255.0

Figure 3- 13 Network settings

3.3 Input and output configuration

The robot system has a total of 16 digital input signals and 16 digital output signals. As shown in Figure 3-14, the input and output signals can be configured and monitored in this window, and the output signals can be forcibly output. IO configuration files can also be saved and loaded on this page. As shown in Figure 3-15, it is an input/output interface description corresponding to the page shown in Figure 3-14.



Digital Input

● DI_0	● DI_8
● DI_1	● DI_9
● DI_2	● DI_10
● DI_3	● DI_11
● DI_4	● DI_12
● DI_5	● DI_13
● DI_6	● DI_14
● DI_7	● DI_15

Digital Output

● DO_0	Test	● DO_8	Test
● DO_1	Test	● DO_9	Test
● DO_2	Test	● DO_10	Test
● DO_3	Test	● DO_11	Test
● DO_4	Test	● DO_12	Test
● DO_5	Test	● DO_13	Test
● DO_6	Test	● DO_14	Test
● DO_7	Test	● DO_15	Test

Safety Input Pins

● Pin_Stop ● PIN_Pause

Figure 3- 14 Input and output configuration

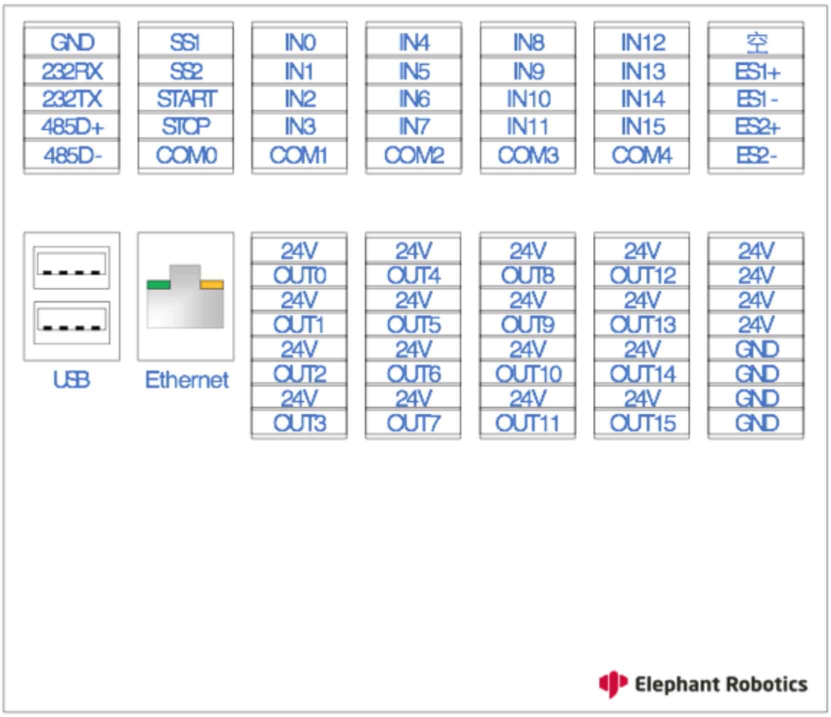


Figure 3- 15 Input and output interface description

It should be noted that the input public terminal needs to be connected to 24V power supply. It can be determined whether the input is active high or active low according to the common configuration (hardware connection determines 24V or 0V). As shown in Figure 3-16, when the common terminal is connected to 24V, once an external device inputs 0V, the input signal is in the “High” state, otherwise it is in the “Low” state; vice versa.

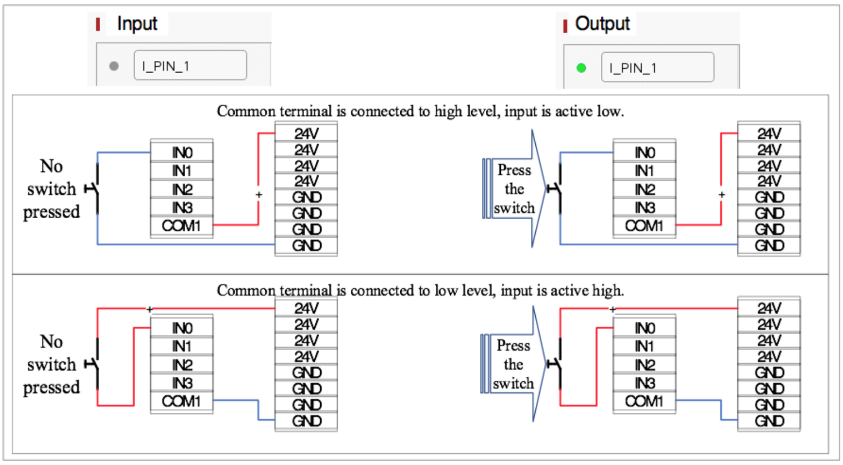


Figure 3- 16 Input signal application diagram

As shown in Figure 3-17, the output is 24V when there is no output. Once the output is turned on (that is, the output is High), the output is 0V.

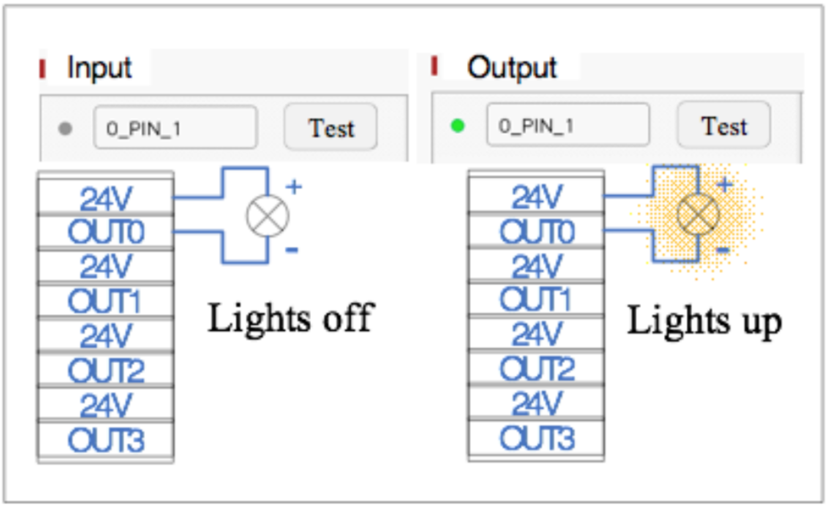
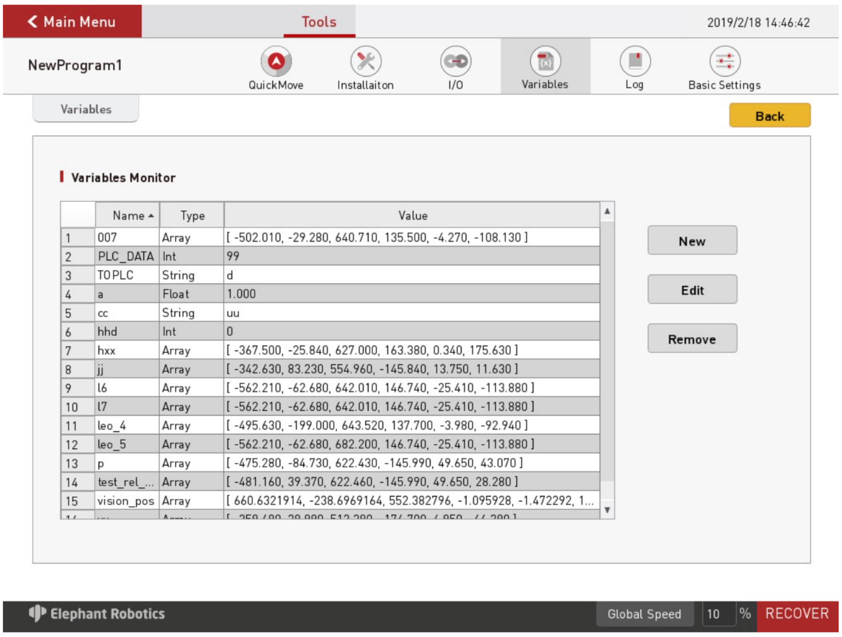


Figure 3- 17 Output signal application diagram

3.4Variable

As shown in Figure 3-18, in the variable editing window, you can add, edit, and delete variables.



	Name	Type	Value
1	007	Array	[-502.010, -29.280, 640.710, 135.500, -4.270, -108.130]
2	PLC_DATA	Int	99
3	TOPLC	String	d
4	a	Float	1.000
5	cc	String	uu
6	hhd	Int	0
7	hxx	Array	[-367.500, -25.840, 627.000, 163.380, 0.340, 175.630]
8	jj	Array	[-342.630, 83.230, 554.960, -145.840, 13.750, 11.630]
9	l6	Array	[-562.210, -62.680, 642.010, 146.740, -25.410, -113.880]
10	l7	Array	[-562.210, -62.680, 642.010, 146.740, -25.410, -113.880]
11	leo_4	Array	[-495.630, -199.000, 643.520, 137.700, -3.980, -92.940]
12	leo_5	Array	[-562.210, -62.680, 682.200, 146.740, -25.410, -113.880]
13	p	Array	[-475.280, -84.730, 622.430, -145.990, 49.650, 43.070]
14	test_rel_...	Array	[-481.160, 39.370, 622.460, -145.990, 49.650, 28.280]
15	vision_pos	Array	[660.6321914, -238.6969164, 552.382796, -1.095928, -1.472292, 1...

Figure 3- 18 Variable editing

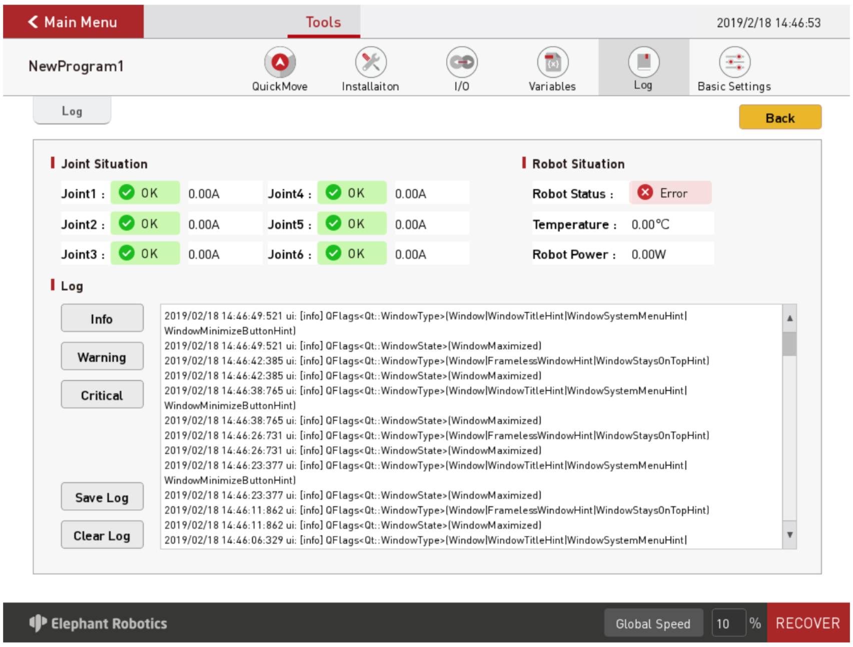
As shown in Figure 3-19, there are 5 types of editable variable types. They are string variables, pose variables, floating point variables, integer variables, and Boolean variables. On this page, you can edit the variable name and initial value.

Figure 3- 19 New variable interface

3.5Log

As shown in Figure 3-20, you can view information about the robot running status, error information, and alarm information in the running log window. Click the "Information", "Warning" and "Error" buttons to sort the corresponding logs.

Users can save logs to a local folder. Log files are a record of how the system is performing, helping users to have a clearer understanding of the system and also helping to troubleshoot errors.



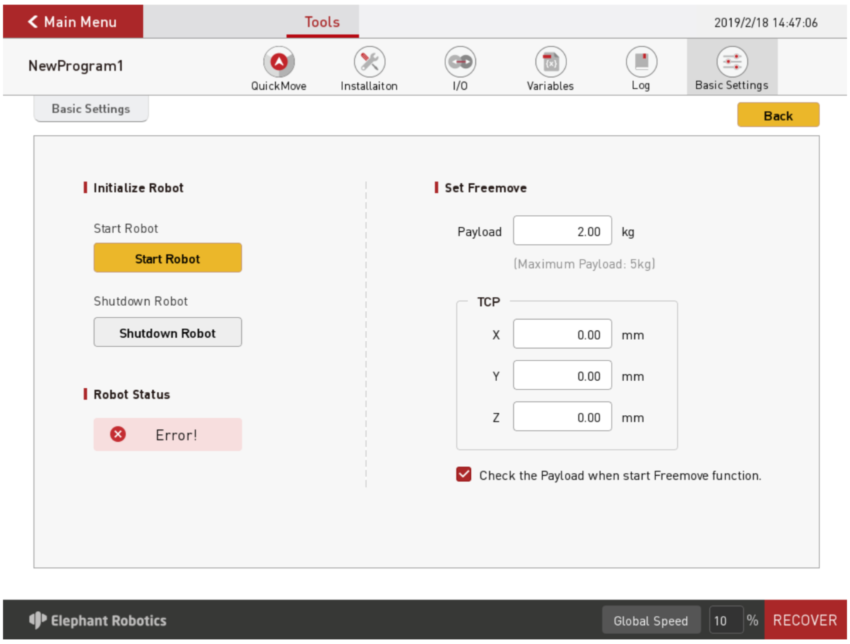
Joint1 :	✓ OK	0.00A	Joint4 :	✓ OK	0.00A	Robot Status :	✗ Error
Joint2 :	✓ OK	0.00A	Joint5 :	✓ OK	0.00A	Temperature :	0.00℃
Joint3 :	✓ OK	0.00A	Joint6 :	✓ OK	0.00A	Robot Power :	0.00W

```
2019/02/18 14:46:49:521 ui: [info] QFlags<Qt::WindowType>(Window|WindowTitleHint|WindowSystemMenuHint|
WindowMinimizeButtonHint)
2019/02/18 14:46:49:521 ui: [info] QFlags<Qt::WindowState>(WindowMaximized)
2019/02/18 14:46:42:385 ui: [info] QFlags<Qt::WindowType>(Window|FramelessWindowHint|WindowStaysOnTopHint)
2019/02/18 14:46:42:385 ui: [info] QFlags<Qt::WindowState>(WindowMaximized)
2019/02/18 14:46:38:765 ui: [info] QFlags<Qt::WindowType>(Window|WindowTitleHint|WindowSystemMenuHint|
WindowMinimizeButtonHint)
2019/02/18 14:46:38:765 ui: [info] QFlags<Qt::WindowState>(WindowMaximized)
2019/02/18 14:46:26:731 ui: [info] QFlags<Qt::WindowType>(Window|FramelessWindowHint|WindowStaysOnTopHint)
2019/02/18 14:46:26:731 ui: [info] QFlags<Qt::WindowState>(WindowMaximized)
2019/02/18 14:46:23:377 ui: [info] QFlags<Qt::WindowType>(Window|WindowTitleHint|WindowSystemMenuHint|
WindowMinimizeButtonHint)
2019/02/18 14:46:23:377 ui: [info] QFlags<Qt::WindowState>(WindowMaximized)
2019/02/18 14:46:11:862 ui: [info] QFlags<Qt::WindowType>(Window|FramelessWindowHint|WindowStaysOnTopHint)
2019/02/18 14:46:11:862 ui: [info] QFlags<Qt::WindowState>(WindowMaximized)
2019/02/18 14:46:06:329 ui: [info] QFlags<Qt::WindowType>(Window|WindowTitleHint|WindowSystemMenuHint|
```

Figure 3- 20 Running log

3.6Basic Settings

As shown in Figure 3-21, the basic settings page provides a common setting channel, allowing the user to quickly set up some functions, such as free movement related parameter settings, even when leaving the programming window while writing the program.



Error!

Payload 2.00 kg

(Maximum Payload: 5kg)

X 0.00 mm

Y 0.00 mm

Z 0.00 mm

☑ Check the Payload when start Freemove function.

Global Speed 10 %

Figure 3- 21 Basic Settings

4 Function instruction

4.1 Basic function

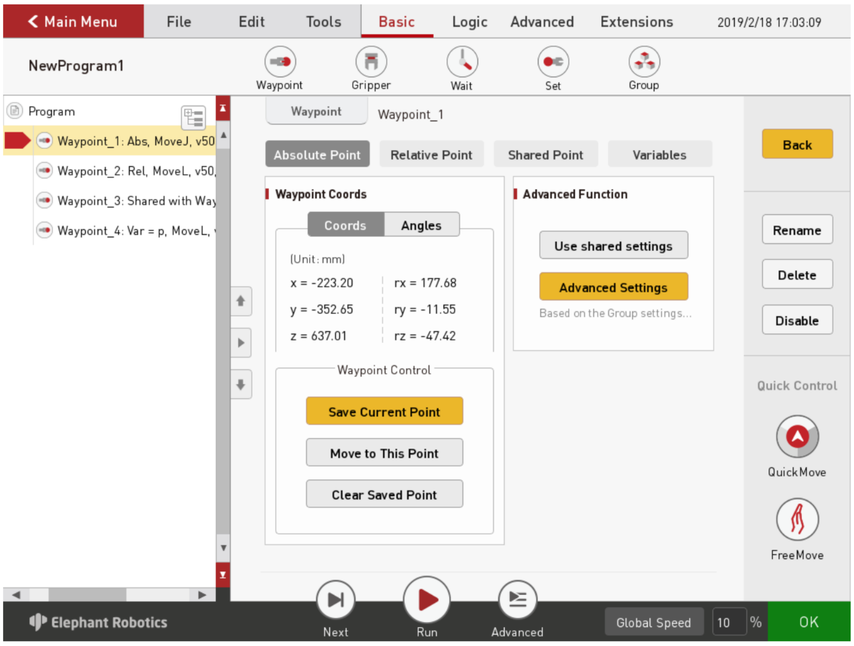
4.1.1Waypoint

There are four types of waypoints: Absolute points, Relative points, Shared points, and Variables. These four types are side-by-side. Under one waypoint command, you can only choose one.

Absolute point: The absolute point is a description of the actual pose of the robot.

- That is, as long as the robot records the absolute point, the next time the instruction is executed, regardless of the position of the robot (other settings unchanged), will reproduce the original teaching of the absolute point of posture.

The specific configuration page for absolute points is shown in Figure 4-1.



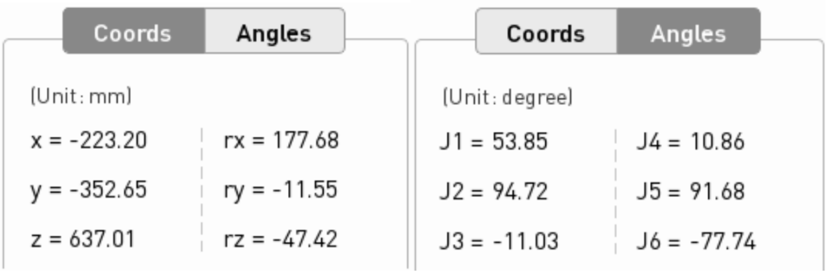
(Unit: mm)

x = -223.20 | rx = 177.68
y = -352.65 | ry = -11.55
z = 637.01 | rz = -47.42

Figure 4- 1 Absolute point

Road Point coordinates

- As shown in Figure 4-2, there are two formats for the representation of absolute points, namely Cartesian coordinate system coordinate values and joint angles. Among them, the Cartesian coordinate system coordinate value records the position and attitude of the robot TCP relative to the base coordinate system (in mm),.The joint angle is a direct record of the actual angle of each axis (in degree, degrees).



(Unit: mm)

x = -223.20 | rx = 177.68
y = -352.65 | ry = -11.55
z = 637.01 | rz = -47.42

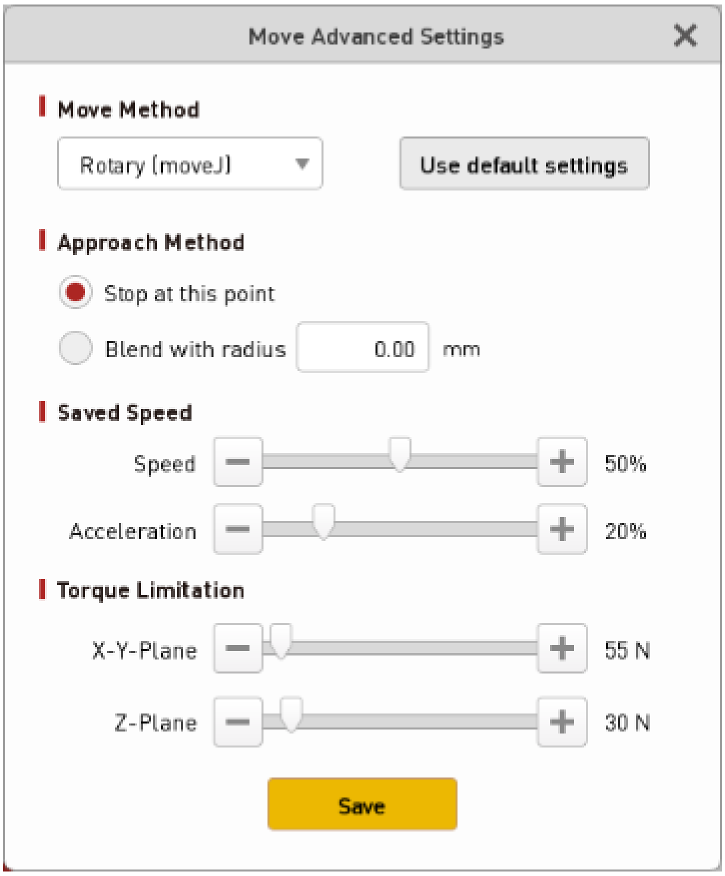
(Unit: degree)

J1 = 53.85 | J4 = 10.86
J2 = 94.72 | J5 = 91.68
J3 = -11.03 | J6 = -77.74

①- Coordinate ②- Angle

Figure 4- 2 Absolute point Position data

Waypoint control Save current point This button is used to save the current pose data of the robot. Move to this point If you need to verify the teaching point or move to the teaching point for some operations, press and hold the button until the robot moves to the current teaching point. If the current teaching point is no longer needed, this button is used to clear the current teaching point. Advanced Features Shared configuration: This feature is being debugged, so stay tuned! Advanced configuration As shown in Figure 4-3, in the advanced configuration page, the user can set the movement mode, proximity mode, command speed, and torque limit.



Move Advanced Settings

Move Method

Rotary (moveJ)

Approach Method

● Stop at this point

○ Blend with radius 0.00 mm

Saved Speed

Speed 50%

Acceleration 20%

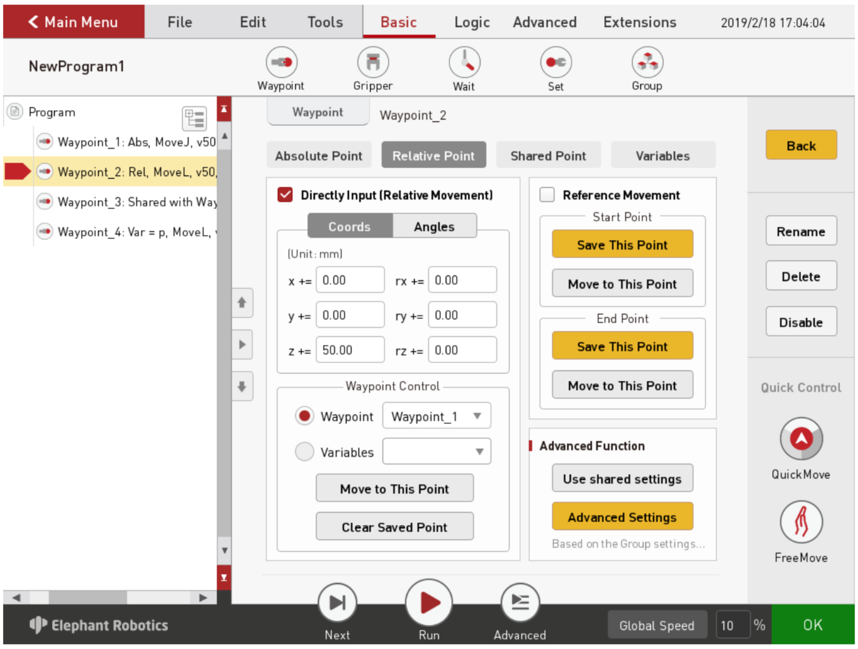
Torque Limitation

X-Y-Plane 55 N

Z-Plane 30 N

Figure 4- 3 Advanced configuration

Relative point: The relative point is used in a situation where a certain displacement is required based on a corresponding point of the movement instruction on the robot/an absolute point/variable point offset. The displacement can be a distance in a single direction, or a superposition of displacements in multiple directions, and can also teach a segment to offset. Figure 4-4 shows the specific configuration page of the relative point.



Program
Waypoint_1: Abs, MoveJ, v50
Waypoint_2: Rel, MoveL, v50,
Waypoint_3: Shared with Way
Waypoint_4: Var = p, MoveL,

Waypoint: Waypoint_2

Absolute Point | Relative Point | Shared Point | Variables

☑ Directly Input (Relative Movement)

Coords | Angles

(Unit: mm)

x +=	0.00	rx +=	0.00
y +=	0.00	ry +=	0.00
z +=	50.00	rz +=	0.00

Waypoint Control
● Waypoint: Waypoint_1
○ Variables
Move to This Point
Clear Saved Point

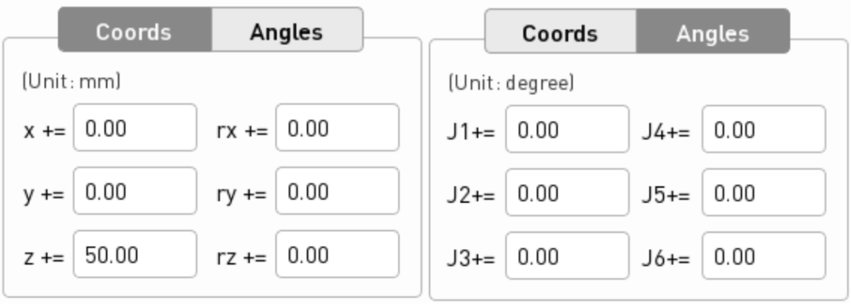
□ Reference Movement
Start Point
Save This Point
Move to This Point
End Point
Save This Point
Move to This Point

Advanced Function
Use shared settings
Advanced Settings
Based on the Group settings...

Global Speed 10 %

Figure 4- 4 Relative point

Direct input(relative movement) As shown in Figure 4-5, you can directly enter the coordinate value / joint angle.



Coords | Angles

(Unit: mm)

x +=	0.00	rx +=	0.00
y +=	0.00	ry +=	0.00
z +=	50.00	rz +=	0.00

Coords | Angles

(Unit: degree)

J1+=	0.00	J4+=	0.00
J2+=	0.00	J5+=	0.00
J3+=	0.00	J6+=	0.00

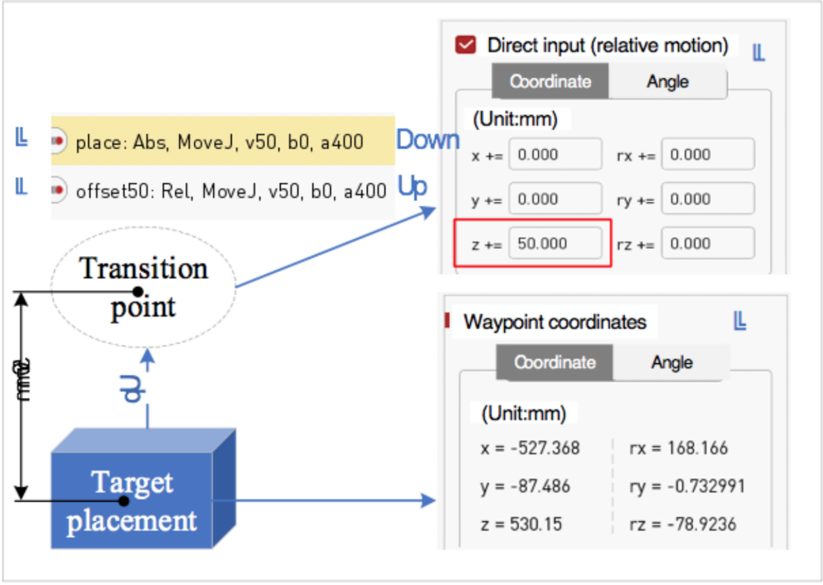
①- Directly enter coordinate values

②- Directly input joint angle

Figure 4- 5 Two forms of direct input

Regardless of whether the coordinate value or the joint angle is input, one or more of the six values are selected according to the offset requirement, and not every value is required to be input.

For example, as shown in Figure 4-6, in the actual pickup and placement process, it is necessary to set a transition point above the target placement position. At this time, we can set a path command as absolute point, control the robot (at this time the robot should be holding the state of the workpiece) to move to the placement point, click to save the current point, which generates the command line 2 shown in Figure 4-6. Then click on the basic function - waypoint: select the relative point, set the z-direction of the icon to increase the relative point of 50mm, then the robot will move to the position of the transition point after running the last sentence. In the actual pickup and placement process, other instructions, such as setting instructions, may be added between the two instructions to open the gripper.



place: Abs, MoveJ, v50, b0, a400

offset50: Rel, MoveJ, v50, b0, a400

(Unit:mm)

x += 0.000 rx += 0.000

y += 0.000 ry += 0.000

z += 50.000 rz += 0.000

(Unit:mm)

x = -527.368 rx = 168.166

y = -87.486 ry = -0.732991

z = 530.15 rz = -78.9236

Figure 4- 6 Application examples of direct input coordinate values

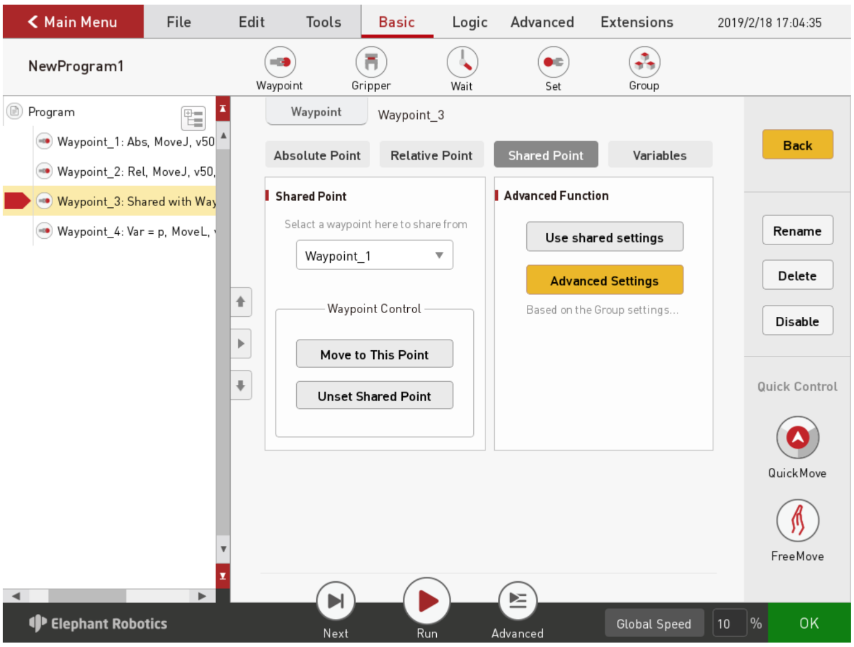
In addition to offset based on the position of the last motion instruction, relative point instruction can also be offset based on a Waypoint or Variable point.

The "Move to this" button verifies the offset motion, and "Clear Saved Points" clears the currently entered content.

Reference move: By teaching two points, a path is generated, based on the current point, and the track is reproduced.

Advanced Features: The advanced configuration of the absolute point is not repeated here.

Shared point：The share point can use the location of other waypoints. Figure 4-7 shows the specific configuration page of the share point.



2019/2/18 17:04:35

NewProgram1

Program
- Waypoint_1: Abs, MoveJ, v50
- Waypoint_2: Rel, MoveJ, v50,
- Waypoint_3: Shared with Way
- Waypoint_4: Var = p, MoveL,

Waypoint: Waypoint_3

Shared Point

Select a waypoint here to share from: Waypoint_1

Waypoint Control

Advanced Function

Based on the Group settings...

Global Speed 10 %

Figure 4- 7 Shared point

Shared point: Select the point you want to share in the box, you can keep pressing the "Move to this point" button to control the robot to move to that point. If you click "Clear Saved Points" to clear the current share point.

Advanced Features: The advanced configuration of the absolute point is not repeated here.

Variable: The waypoint can be assigned by a variable. The user can use the communication method to obtain the waypoint location from other devices.

Figure 4-8 shows the specific configuration page of the variable point.

Variable assignment: The user can select the associated pose variable, and "Move to this point" can check whether the pose is the target pose.

Advanced Features: The advanced configuration of the absolute point is not repeated here.

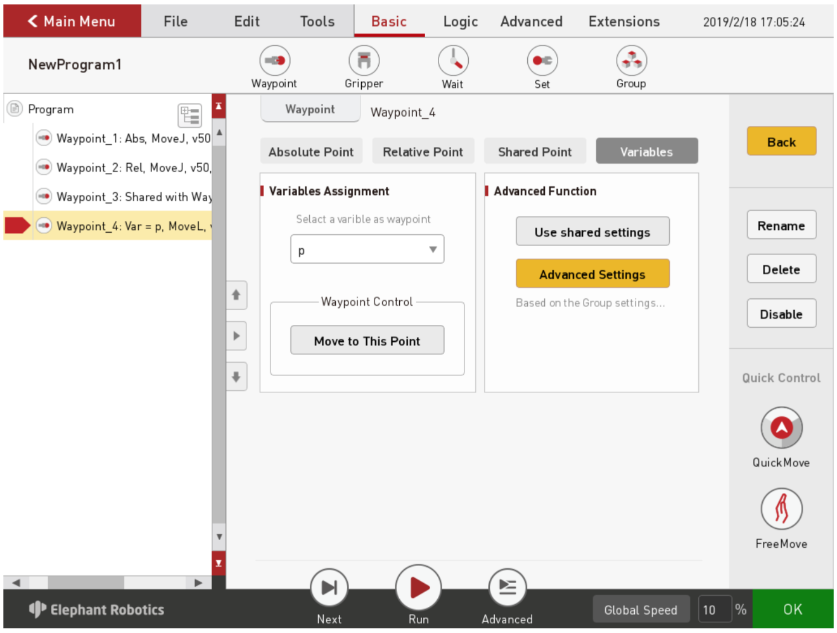
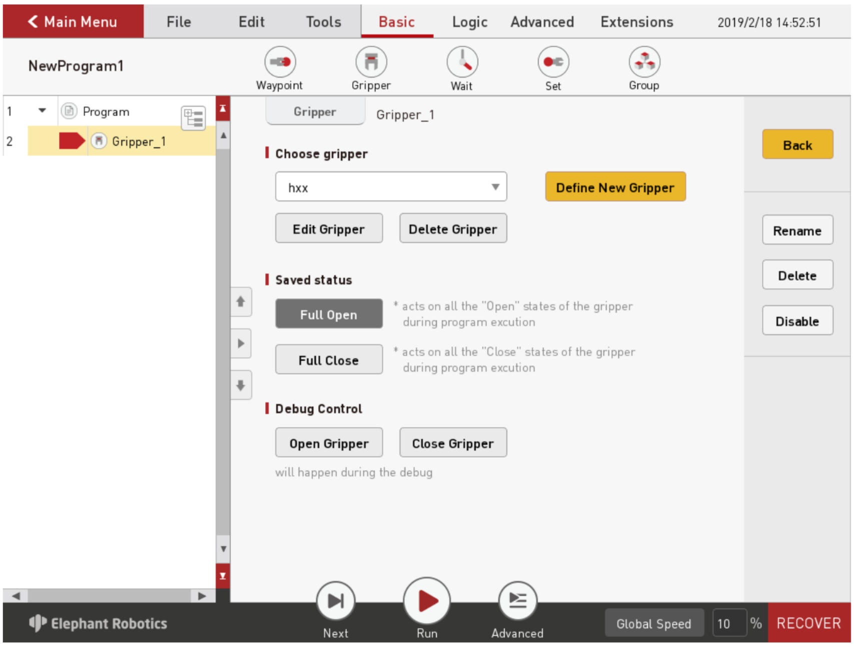


Figure 4- 8 Variable

4.1.2 Gripper Figure 4-9 shows the specific configuration page of the gripper.



Choose gripper: hxx

Saved status

* acts on all the "Open" states of the gripper during program excution

* acts on all the "Close" states of the gripper during program excution

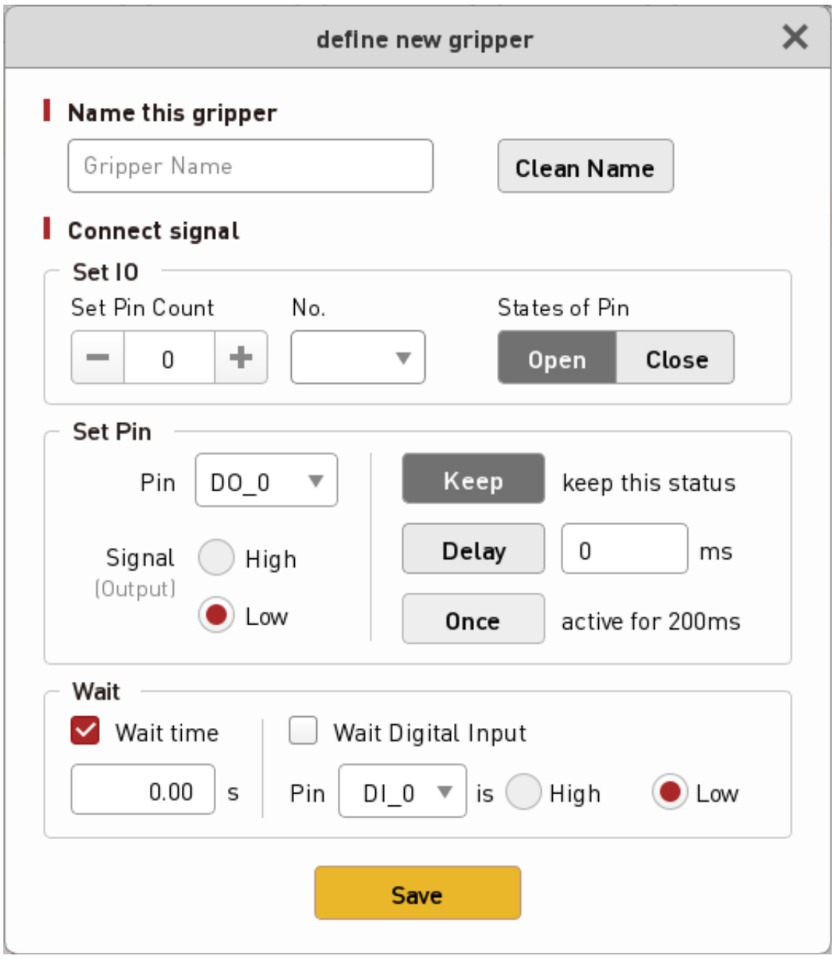
Debug Control

will happen during the debug

Figure 4- 9 Gripper

The user defines and controls the gripper through a simple function.

- 1 Select gripper
- 2 Set existing grippers
- 3 Select the gripper, you can edit or delete the existing gripper.
- 4 Define new grippers 如图4-10所示，可以命名夹爪，同时控制多个输入信号：设置需要控制的输出信号的数量、在“设置”中选择设置第几个信号、设置状态（关系到具体执行时对应“打开”或“关闭”功能）、设置对应输出信号。在设置完成后，还可以选择等待条件。



define new gripper

Name this gripper

Gripper Name

Clean Name

Connect signal

Set IO

Set Pin Count: 0

No.

States of Pin: Open | Close

Set Pin

Pin: DO_0

Signal (Output): ○ High ● Low

Keep: keep this status

Delay: 0 ms

Once: active for 200ms

Wait

☑ Wait time: 0.00 s

□ Wait Digital Input: Pin DI_0 is ○ High ● Low

Save

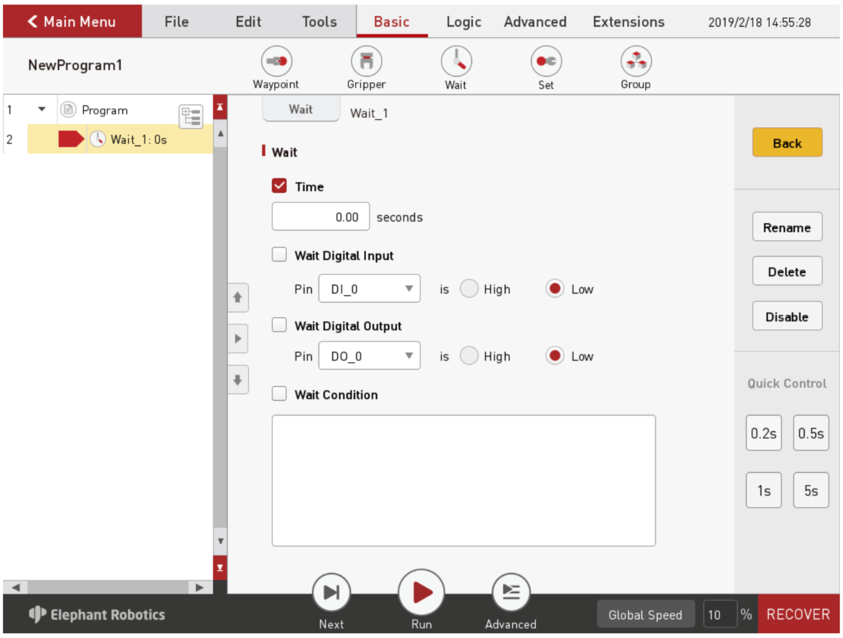
Figure 4- 10 Define new grippers

Set the saved state

- Fully open: The option in the execution gripper definition is the"open" state.
- Completely off: The option in the execution gripper definition is "off" status. Debug control
- Open the gripper: Manual operation performs the option of the "Open" state in the gripper definition.
- Close the gripper: Manual operation performs the option of the "Close" state in the gripper definition.

4.1.3 Wait As shown in Figure 4-11, there are four modes for waiting for instructions.

- Waiting time: The delay time can be set in seconds.
- Waiting for the input signal: The state of the input signal is judged and waits until it meets the set input signal state condition.
- Waiting for the output signal: The state of the output signal is judged and waits until it meets the set output signal state condition.
- Waiting conditions: You can customize the wait condition and wait until the wait condition is met.



1 Program
2 Wait_1: 0s

Wait Wait_1

Wait

☑ Time

0.00 seconds

□ Wait Digital Input

Pin DI_0 is ○ High ● Low

□ Wait Digital Output

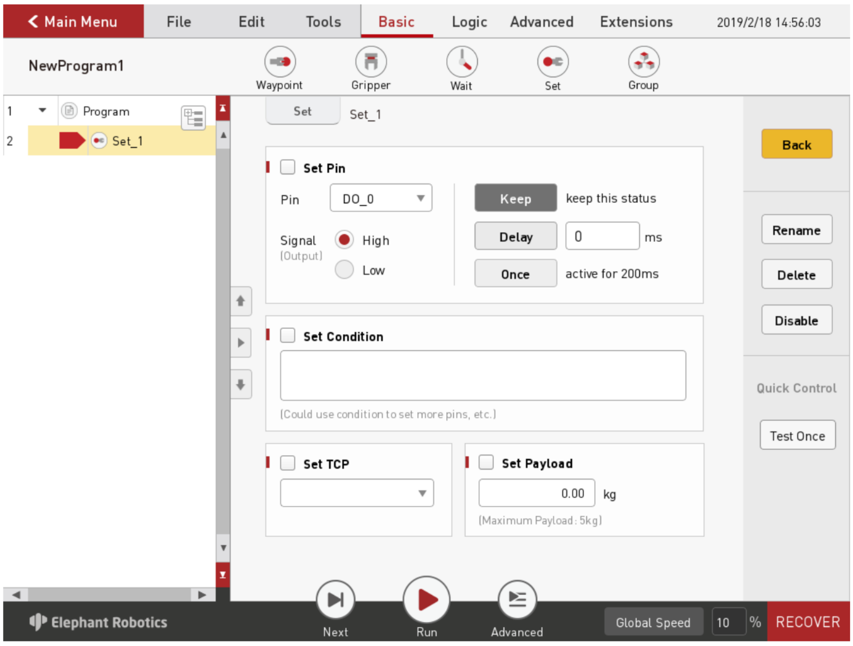
Pin DO_0 is ○ High ● Low

□ Wait Condition

Figure 4- 11Wait

4.1.4 Set As shown in Figure 4-12, the setup command has four modes of selection.

- Set IO: Set the state of the output signal. In addition to selecting the set output signal to determine whether it is on or off, you can also set the time that the signal is held.
- Set conditions: Customize the content of the settings.
- Set TCP (i.e. tool center point).
- Set the load.



1 Program
2 Set_1

Set Set_1

□ Set Pin

Pin DO_0

Signal (Output) ● High ○ Low

Keep keep this status

Delay 0 ms

Once active for 200ms

□ Set Condition

(Could use condition to set more pins, etc.)

□ Set TCP

□ Set Payload

0.00 kg

(Maximum Payload: 5kg)

Figure 4- 12 Set

4.1.5Group As shown in Figure 4-13, the group instructions provide common combination templates, such as grabbing and placing combinations.

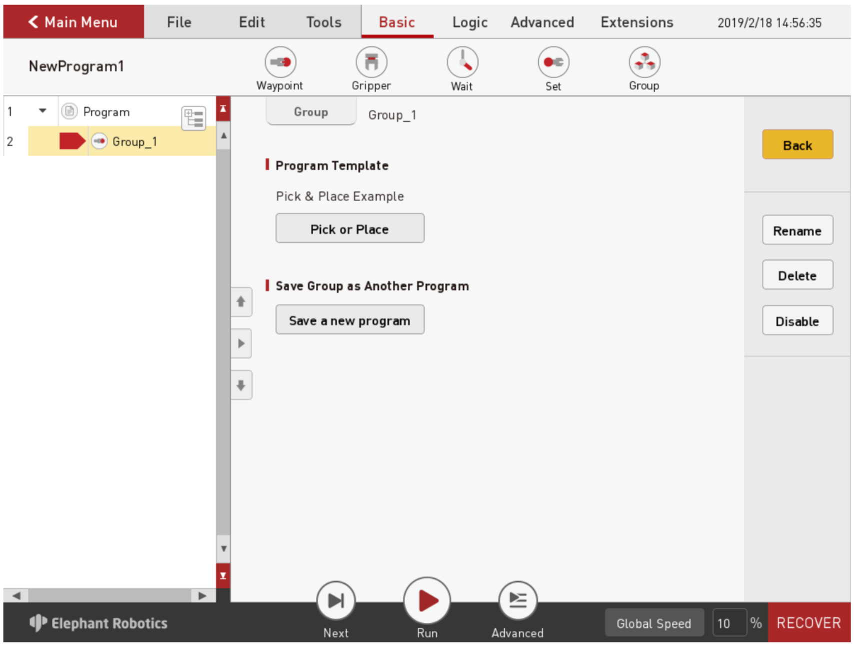


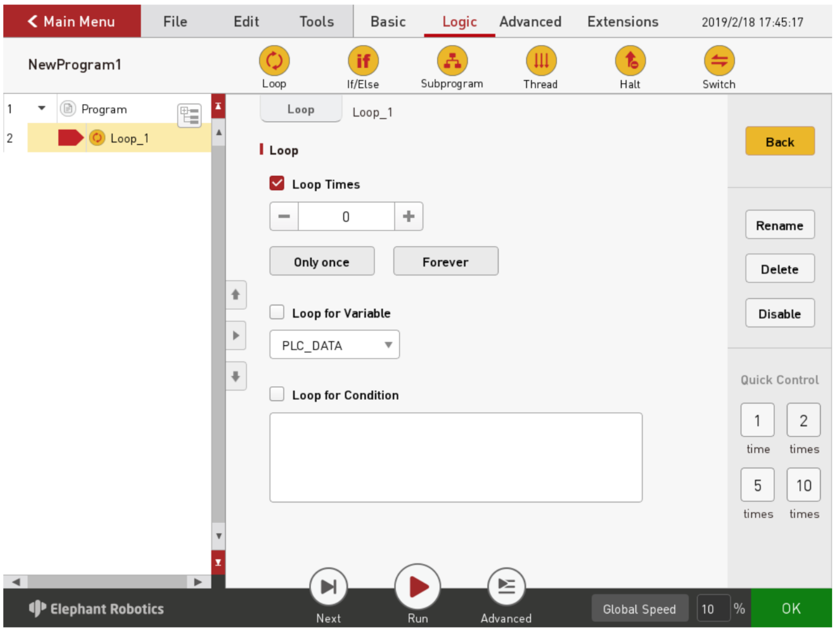
Figure 4- 13 Group

When users use group instruction, such as grabbing and placing combinations, they can modify parameters and teach Waypoints directly on the basis of template programs, or they can add or delete instructions freely according to their needs.

The user can simplify the process of finding instructions by using the group instruction. And it is more convenient and quicker to complete the programming of the corresponding project.

4.2 Logic function

4.2.1Loop Loop instructions can repeat all instructions within a loop for a certain number of times. As shown in Figure 4-14, the number of loops can be represented by a constant or a variable or an expression.



Loop

☑ Loop Times

0

Only once

Forever

☐ Loop for Variable

PLC_DATA

☐ Loop for Condition

Figure 4- 14 Loop

4.2.2 If/Else

Judging the set conditions allows the program to read the data, determine and determine what to do next.

If/Else can be used to determine the I/O signal and can also be used to determine other conditions.

If/Else consists of three parts: If, Else If, and Else. The relationship between these three parts is as follows:

Except that If is an integral part, the remaining two are optional parts;

If both If, Else If, and Else exist, the program will first read If, then read Else If, Else If ... Else. The relationship between the three is shown in Figure 4-15:

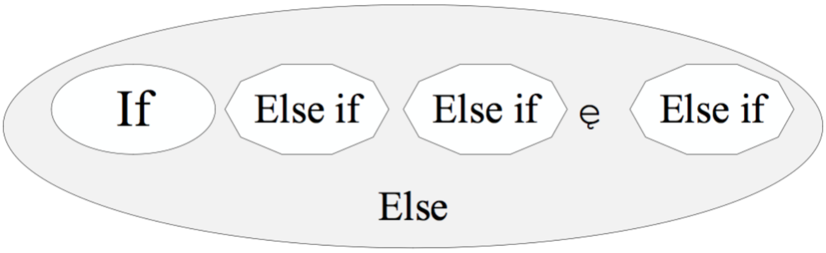
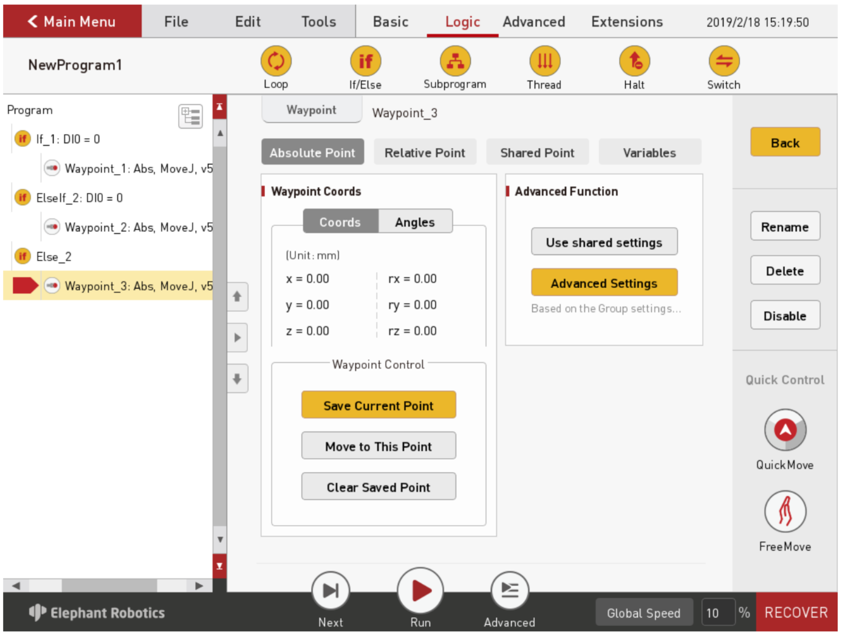


Figure 4- 15 Relationship between If, Else If, and Else

There can be more than one Else If, but there is only one If, and if you choose to add Else, you can only have one Else.

You can delete Else If or Else, but if you delete If, delete all Else If and Else.

Figure 4-16 shows the setting page of the conditional judgment command.



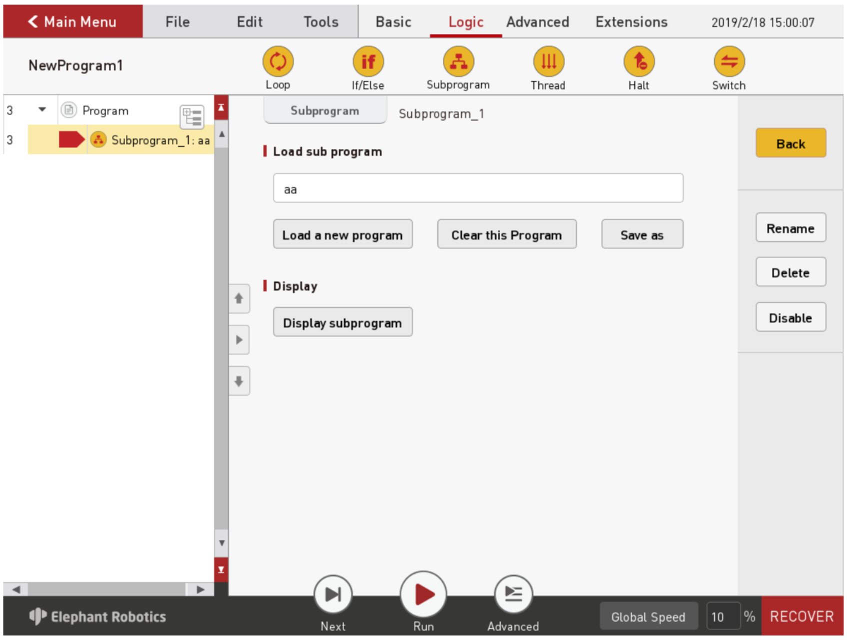
Program
If_1: DI0 = 0
Waypoint_1: Abs, MoveJ, v5
ElseIf_2: DI0 = 0
Waypoint_2: Abs, MoveJ, v5
Else_2
Waypoint_3: Abs, MoveJ, v5

(Unit: mm)
x = 0.00　rx = 0.00
y = 0.00　ry = 0.00
z = 0.00　rz = 0.00

Figure 4- 16 If/Else

As shown in the figure above, if the condition following "If" is met, the robot will move to waypoint 1; if it meets the condition followed by "Else if", it will move to waypoint 2; if both conditions are not met, the "Else" corresponding block will be executed, that is, the robot will move to the waypoint 3.

4.2.3 Subprogram As shown in Figure 4-17, other subroutines can be called using this instruction. The main program can use multiple subroutines, but there are no subroutines in the subprogram.



3 Program
3 Subprogram_1: aa

aa

Figure 4- 17 Subprogram

As shown in Figure 4-18, you can view and edit subroutines in the main program. If you edit the subroutine, please note that it will not take effect until it is saved.

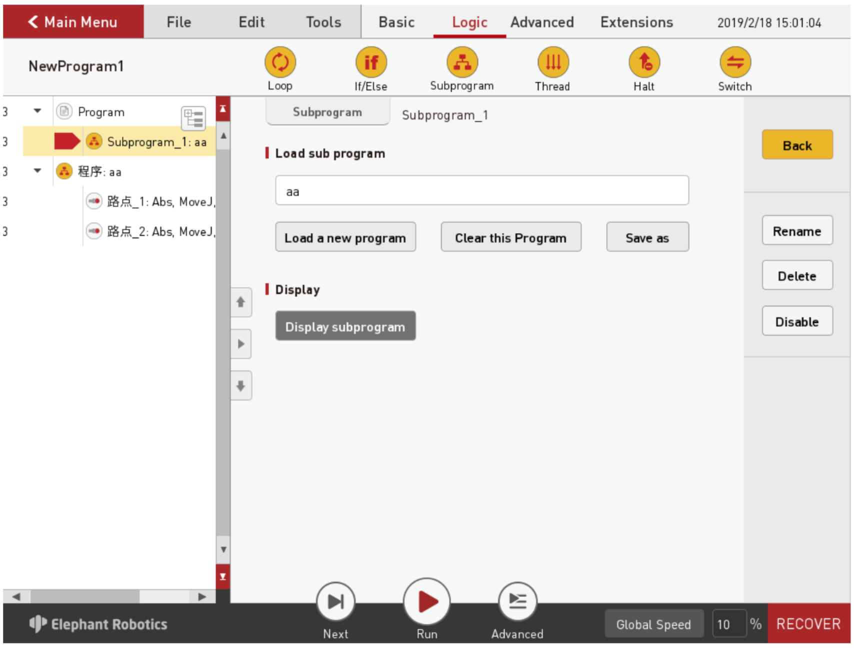
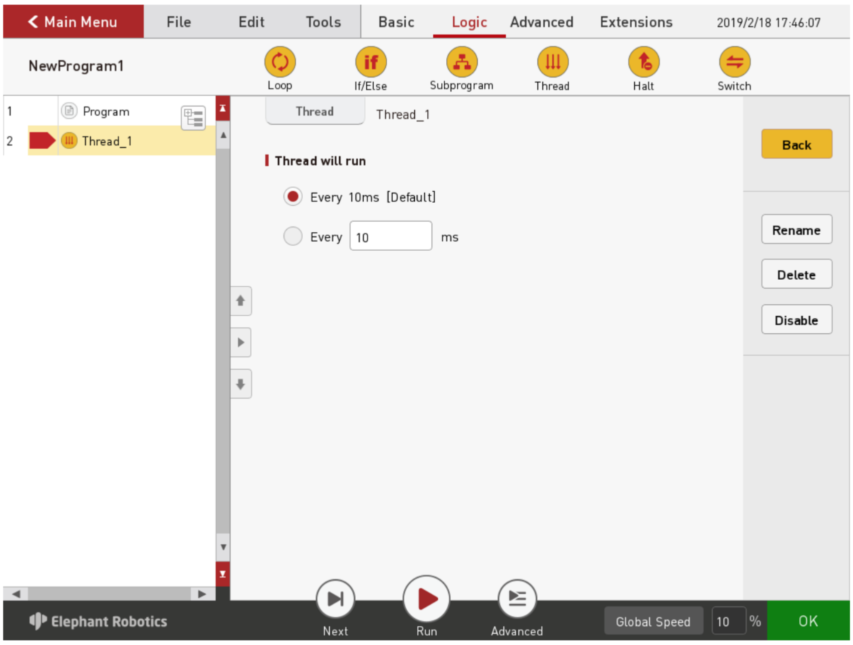


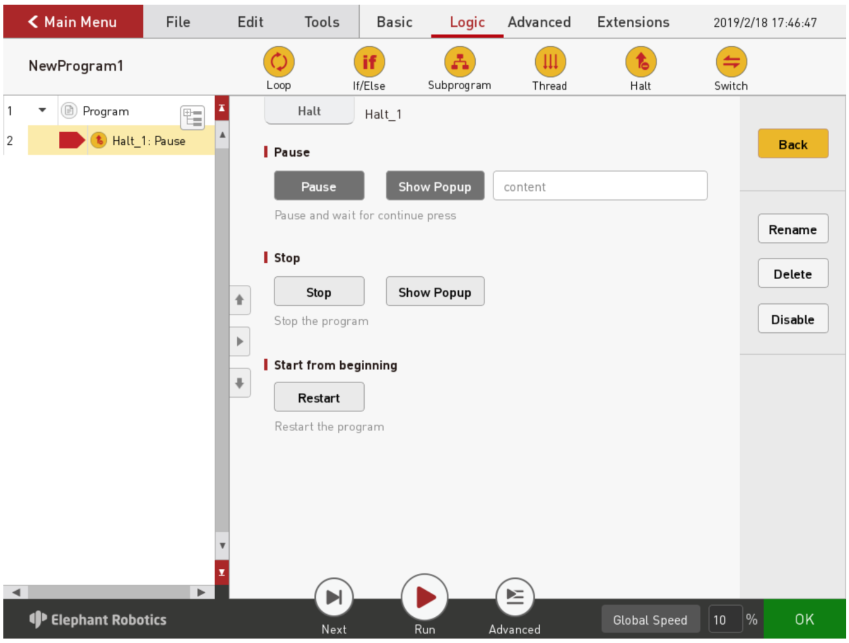
Figure 4- 18 Display subroutine

4.2.4Thread The thread runs along the main program. It is used to check signals such as emergency buttons or safety light curtains. As shown in Figure 4-19, you can set the running interval of threads.



Note that motion instructions are not allowed in threads. **4.2.5Halt** The pause command is used to control the robot to pause, stop, and resume. Figure 4-20 shows the specific configuration page for the pause command.

- When setting the pause and stop status, you can also select “Show Popup” to customize the contents displayed by the popup.
- Set the restart state. When the program runs to this instruction, it will start running again from the first instruction at the beginning.



Halt Halt_1

Pause and wait for continue press

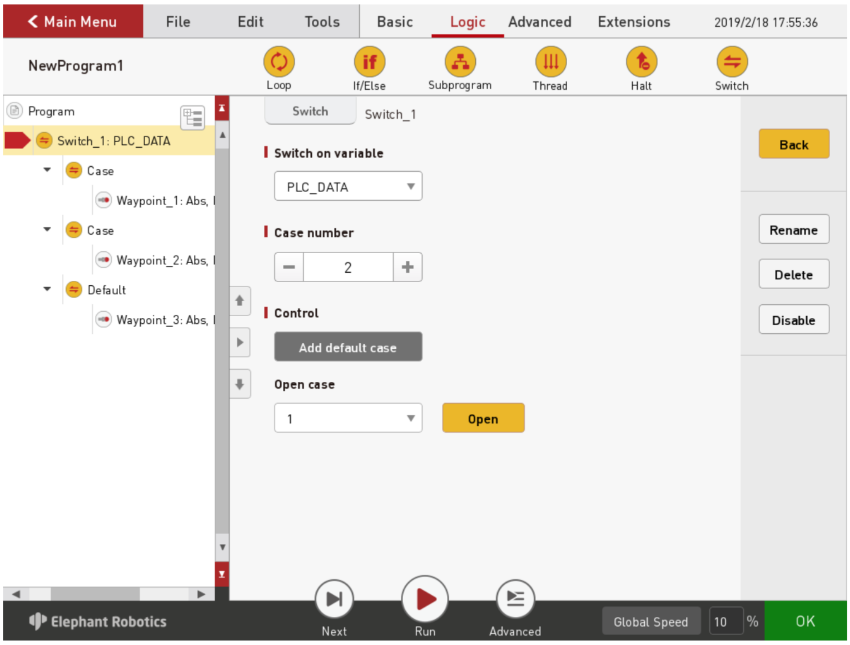
Stop the program

Restart the program

Global Speed 10 %

Figure 4- 20 Halt

4.2.6 Switch As shown in Figure 4-21, the conditional selection instruction is used to judge the value of a variable.



Switch Switch_1

Switch on variable: PLC_DATA

Case number: 2

Open case: 1

Global Speed 10 %

Figure 4- 21 Switch

Corresponding to different conditional values, how many conditional values need to be judged to add how many cases, you can open each case, increase the corresponding execution instructions. For example, to judge the integer variable A, set two cases, if A is 1, execute the first route instruction, if A is 2, execute the second route instruction.

If only a few variables are judged, and other cases are handled uniformly, we need to select "default" and add corresponding instructions to the switch.

4.3 Advanced function

4.3.1Pallet The pallet instruction allows the user to teach only a few points, through which the position of the other points can be calculated by the robotic system. Running this instruction can control the movement of the robot to these points. As shown in Figure 4-22, you can select a line, plane, cube, discrete point.

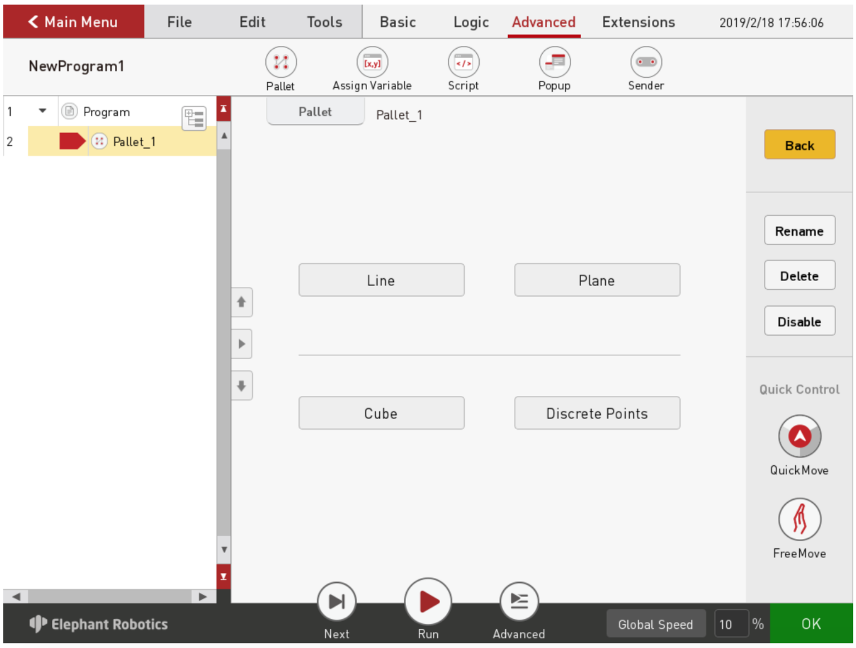


Figure 4- 22 Pallet type Selection

As shown in Figure 4-23, after you select line, select the number of points, and the line will be split evenly based on the number of points. These points are the split point. The user determines this line by teaching two points.

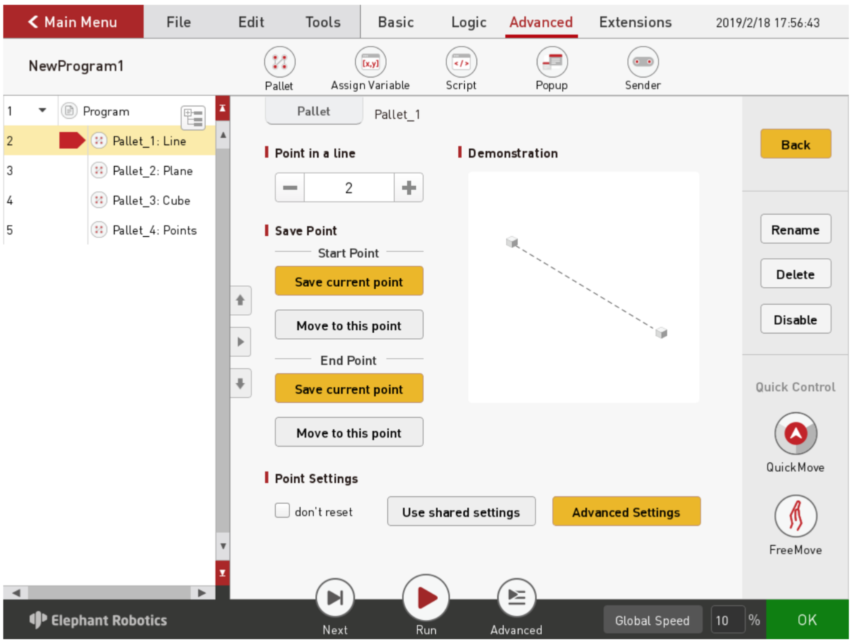
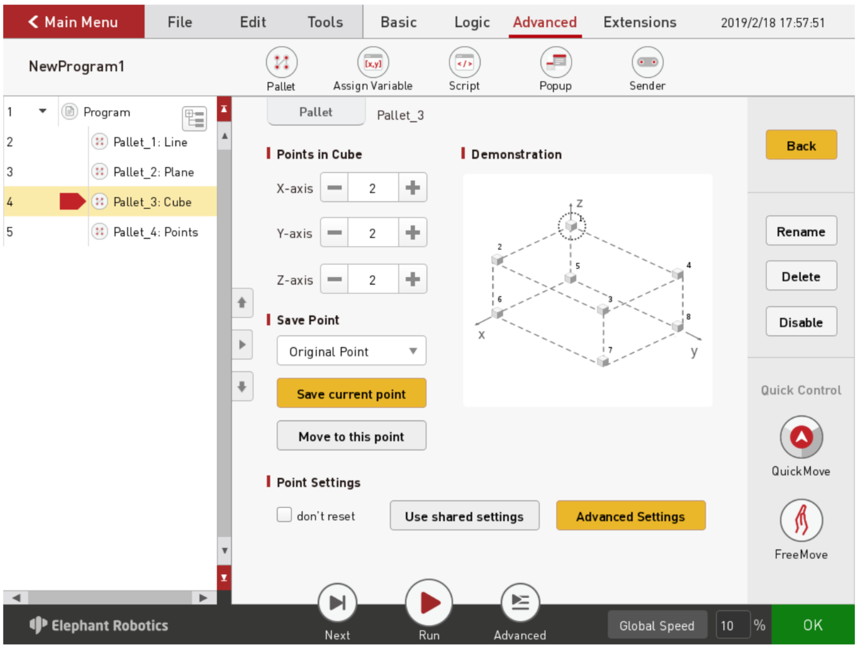


Figure 4- 23 Line

As shown in Figure 4-24, after selecting “Plane”, select the number of points of the two axes, and the plane is divided equally. These points are the dividing points. This plane is determined by teaching four points.

Figure 4- 24 Plane

As shown in Figure 4-25, after selecting "Cube", select the number of points of the three axes, and the cube is divided equally. These points are the dividing points. Determine this cube by teaching eight points.



1 Program
2 Pallet_1: Line
3 Pallet_2: Plane
4 Pallet_3: Cube
5 Pallet_4: Points

Pallet: Pallet_3

Points in Cube

X-axis 2

Y-axis 2

Z-axis 2

Save Point: Original Point

Save current point

Move to this point

Point Settings

□ don't reset

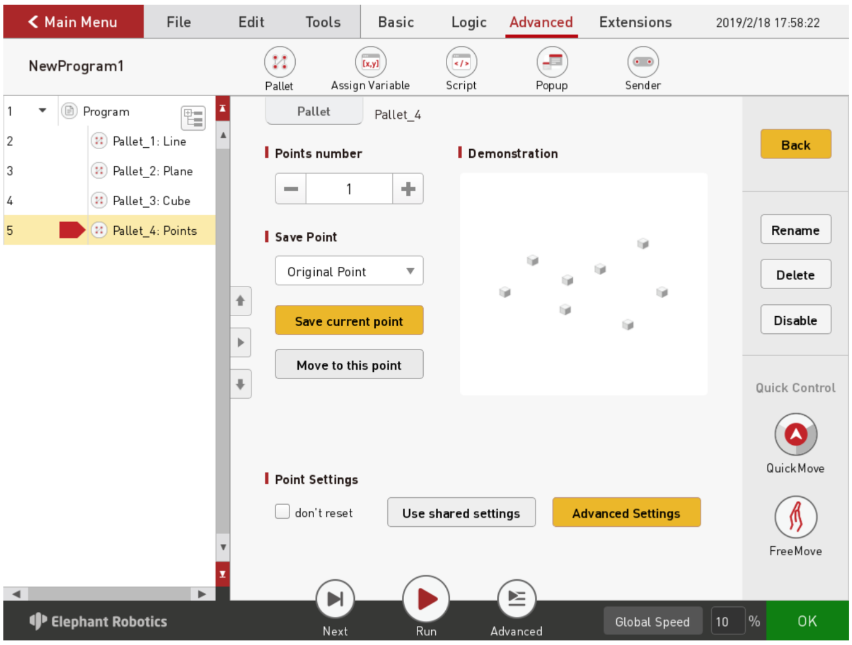
Use shared settings

Advanced Settings

Global Speed 10 %

Figure 4- 25 Cube

As shown in Figure 4-26, when “Discrete Point” is selected, the number of points is selected to teach different points. That is, a discrete point is a collection of multiple points.



2019/2/18 17:58:22

NewProgram1

1 Program
2 Pallet_1: Line
3 Pallet_2: Plane
4 Pallet_3: Cube
5 Pallet_4: Points

Pallet: Pallet_4

Points number: 1

Demonstration

Save Point: Original Point

Save current point

Move to this point

Point Settings

□ don't reset

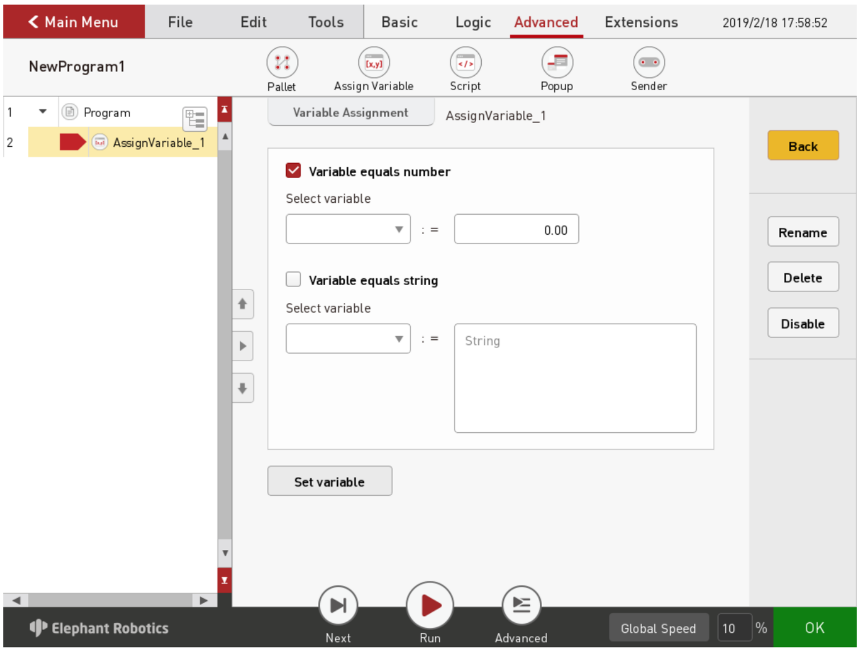
Use shared settings

Advanced Settings

Global Speed 10 %

Figure 4- 26 Discrete point

4.3.2Assign to var As shown in Figure 4-27, this command can assign values to integer variables and string variables. You can also use the "set variables" to directly set the value of the variable according to the instruction.



2019/2/18 17:58:52

NewProgram1

1 Program
2 AssignVariable_1

Variable Assignment: AssignVariable_1

☑ Variable equals number

Select variable : = 0.00

□ Variable equals string

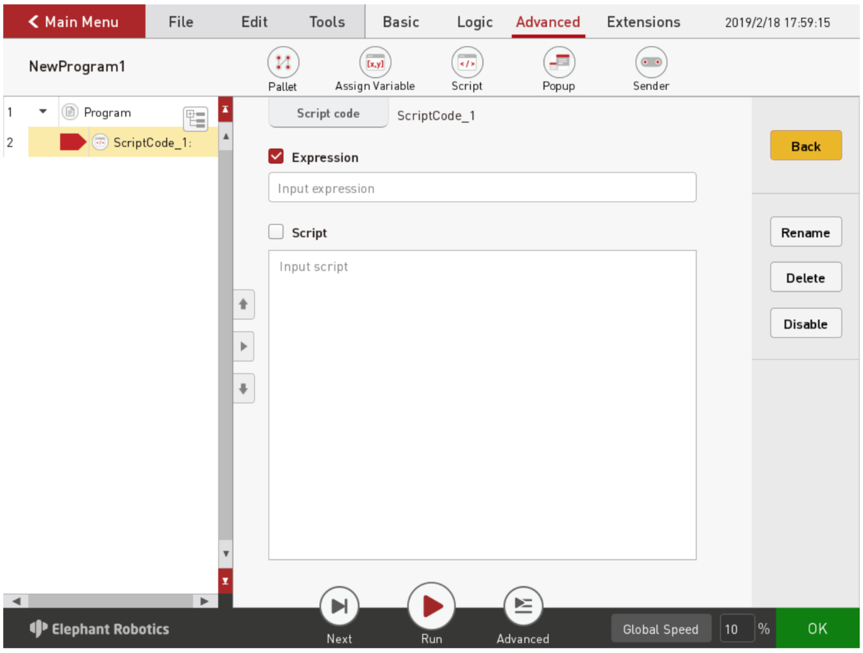
Select variable : = String

Set variable

Global Speed 10 %

Figure 4- 27 Assign to var

4.3.3Script Script instructions can be used to edit complex instructions, providing a richer set of functional instructions. Figure 4-28 shows the specific configuration page of the script command. There are two types of setup scripts, one is a single-line expression and the other is a multi-line script.



☑ Expression

Input expression

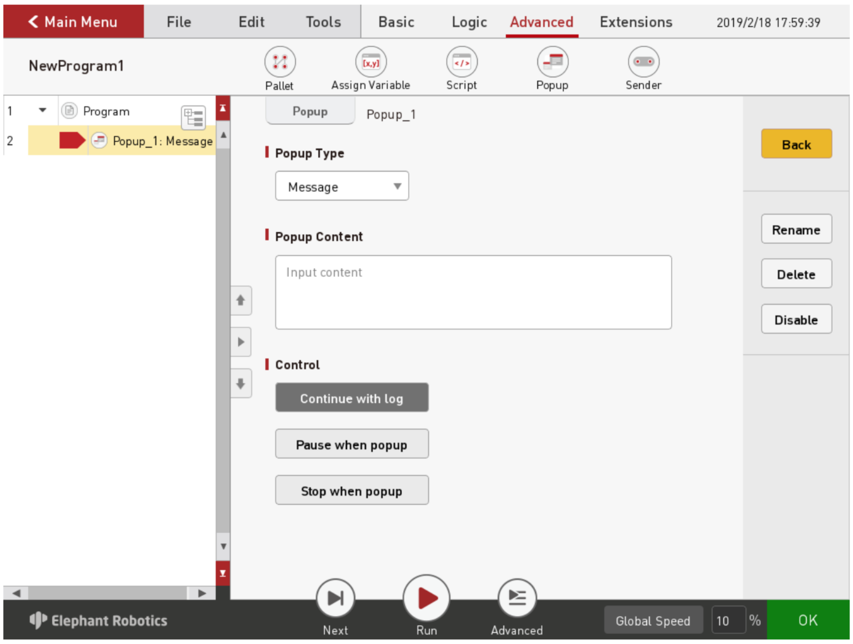
□ Script

Input script

Figure 4- 28 Script

4.3.4Popup The pop-up command allows the user to customize the pop-up window. In other words, when this command is executed, a pop-up window appears, and the pop-up content is user-defined content. As shown in Figure 4-29, there are three types of pop-up windows, information, warnings, and errors. The user selects one and customizes the pop-up content.

There are also three kinds of pop-up window control: continue the program (logging), that is, do not pop the window, just display the contents of the pop-up window to the log, and the program continues to run; When the window is popped, the program is paused, that is, the pop-up window appears, and the program is suspended; When the window is popped, the program stops, and the pop-up window appears, and the program stops running.



Popup Type: Message

Popup Content

Input content

Control

Figure 4- 29 Popup

4.3.5Sender

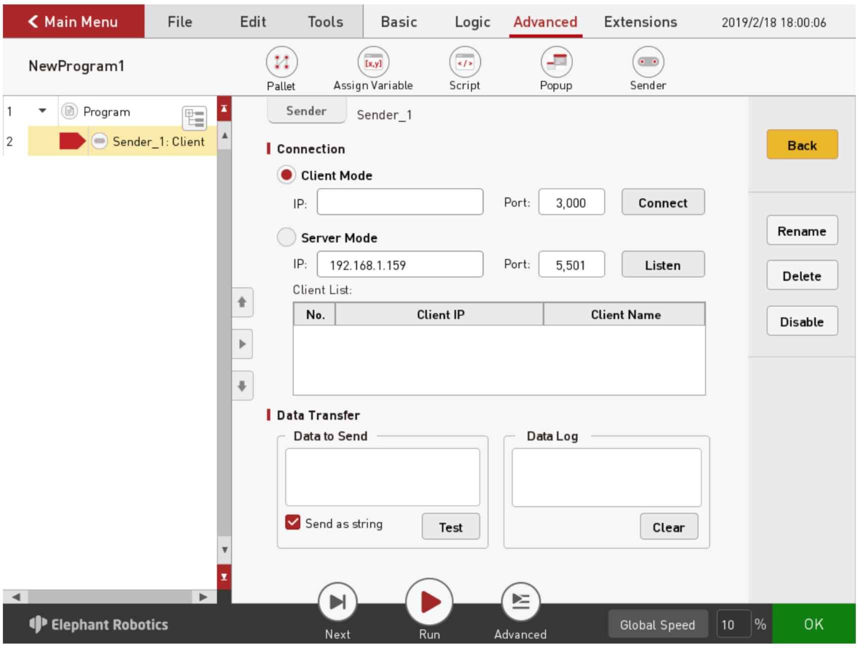
If TCP/IP communication is to be performed, the robot system must set the IP and port number as a client or server to communicate with other devices.

The sender allows the user to set up a TCP/IP connection. Figure 4-30 shows the specific configuration page of the Sender instruction.

If the robot system acts as a client, the IP address filled in is the IP address of the external device of the server, and the port number corresponds to the port number assigned to the robot system by the server. When the server is in the state of monitoring, it can communicate with the server by clicking the "connection" button.

If the robot system serves as a server, the IP address filled in is the local IP address, and the port number corresponds to the port number assigned to the client device. Click on the "monitor" button, at which point the client device can connect to the robot system. In the client list, you can view the IP addresses and port numbers of all clients.

After establishing communication, data can be sent and received.



Sender_1

● Client Mode
IP: Port: 3,000

○ Server Mode
IP: 192.168.1.159 Port: 5,501

☑ Send ass string

Figure 4- 30 Sender

5 Quickly create a new runnable project

5.1 Flow Description

5.1.1 Ready to work

Precondition

- Complete robot system
- No problem **Prepare content**

- Plug the power cord into the board that provides the AC 220V.
- Turn on the power switch. **Press the start button on the teach pendant.**

5.1.2 Flow chart As shown in Figure 5-1, it is the program editing flowchart.

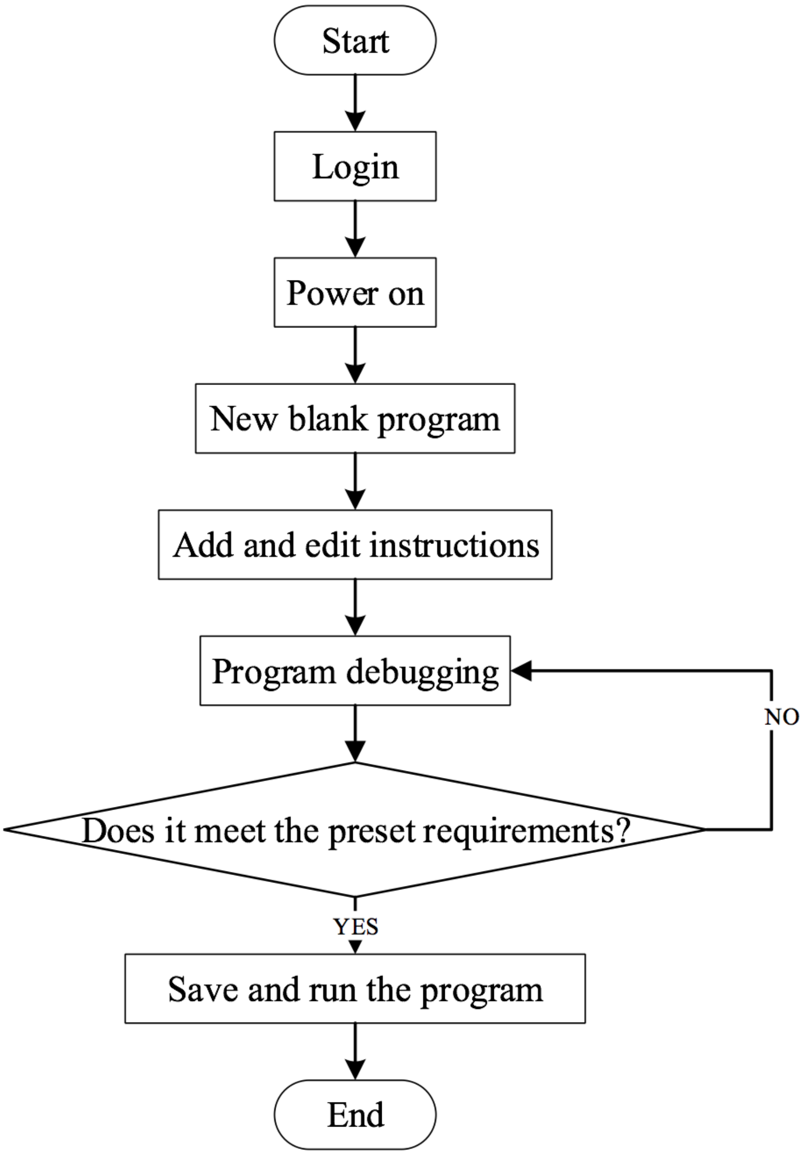


Figure 5- 1 Program editing flow chart

5.2 Specific steps

5.2.1 Login After the system is successfully started, it will enter the login interface of the OS3 operating system as shown in Figure 5-2.

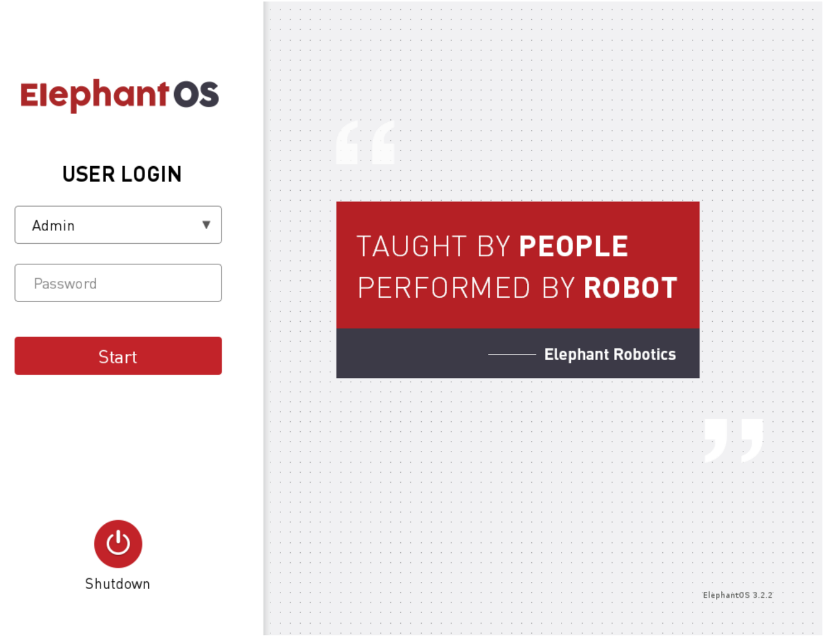


Figure 5- 2 Login interface

Select the login user name "Admin" or other administrator user name (only administrator permissions are allowed to edit and debug the program), click on the password box will pop up as shown in Figure 5-3.

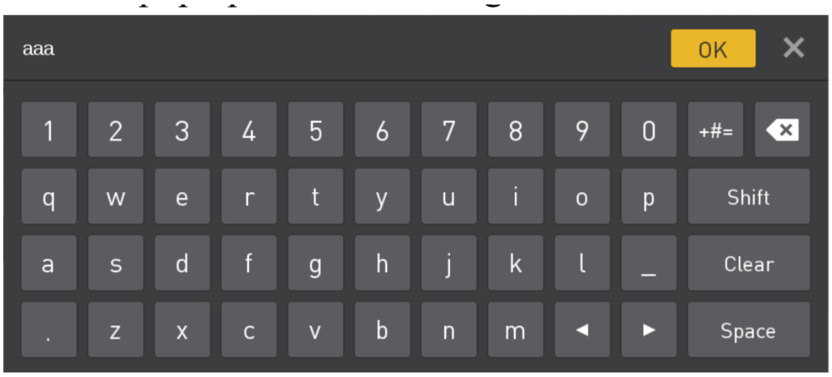
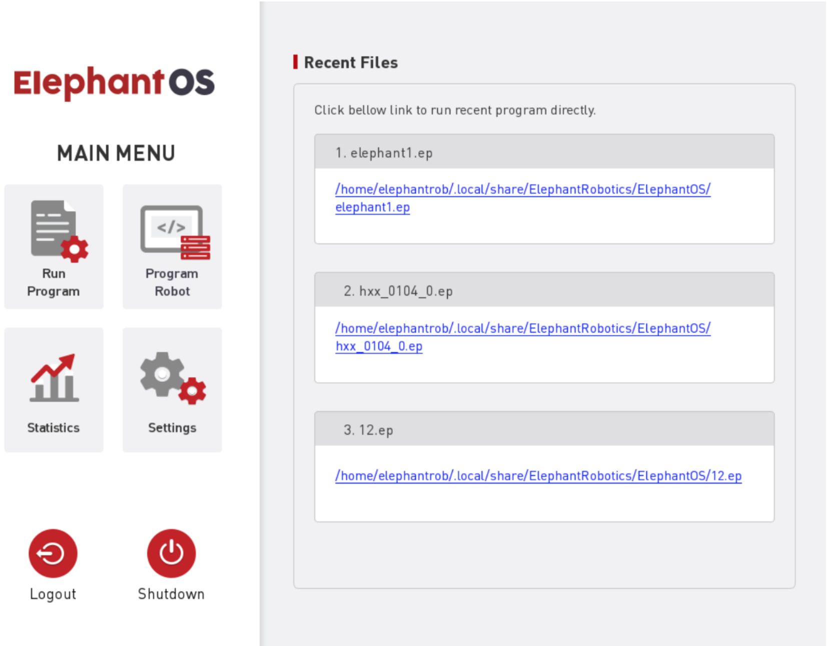

aaa
OK
✕
1 2 3 4 5 6 7 8 9 0 +#=
q w e r t y u i o p Shift
a s d f g h j k l _ Clear
. z x c v b n m ◄ ► Space

Figure 5- 3 Input keyboard

The login password corresponding to the default administrator user “Admin” is “aaa” (if the other administrator user name is selected, enter the corresponding login password), enter the password and click “OK”, and return to the interface of Figure 5-2. Then click "Login" to log in successfully. **5.2.2 Power on** After the login is successful, the main menu interface shown in Figure 5-4 will be displayed.



Recent Files

Click bellow link to run recent program directly.

1. elephant1.ep
/home/elephantrob/.local/share/ElephantRobotics/ElephantOS/elephant1.ep

2. hxx_0104_0.ep
/home/elephantrob/.local/share/ElephantRobotics/ElephantOS/hxx_0104_0.ep

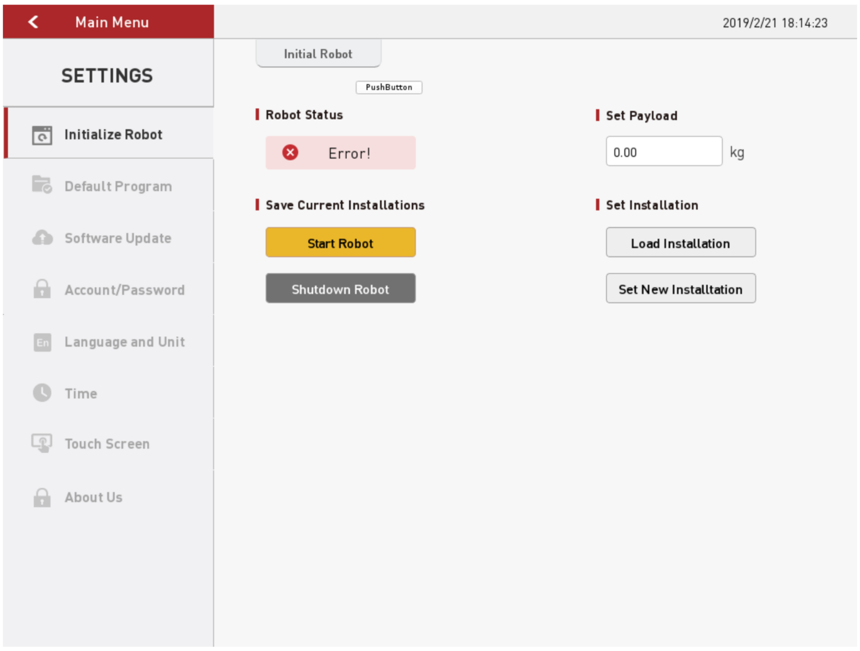
3. 12.ep
/home/elephantrob/.local/share/ElephantRobotics/ElephantOS/12.ep

Figure 5- 4 Main menu

In the main menu interface, select “Settings”, it will enter the interface as shown in Figure 5-5 (this time has not been powered).

To ensure that the emergency stop knob is not pressed, click on the “Start Robot” button as shown in Figure 5-5. The interface will change and the “Powering On” icon as shown in Figure 5-6 will appear. If the power is turned on successfully, the “I’m OK!” status shown in Figure 5-7 will appear. If it fails, check if you are missing any steps.

After completing the previous step, return to the main menu by pressing the motor “<Main Menu” button in the configuration center.

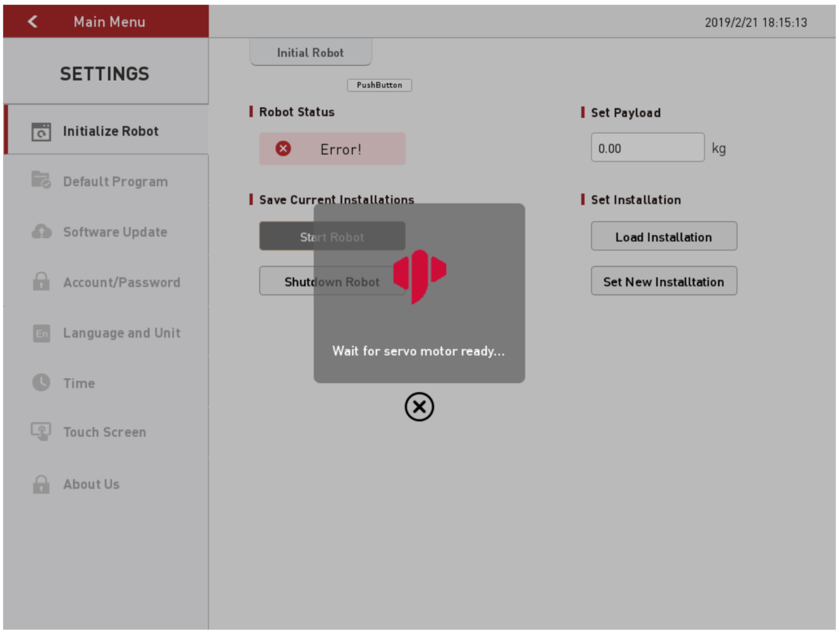


2019/2/21 18:14:23

Robot Status: Error!

Set Payload: 0.00 kg

Figure 5- 5 Unpowered state

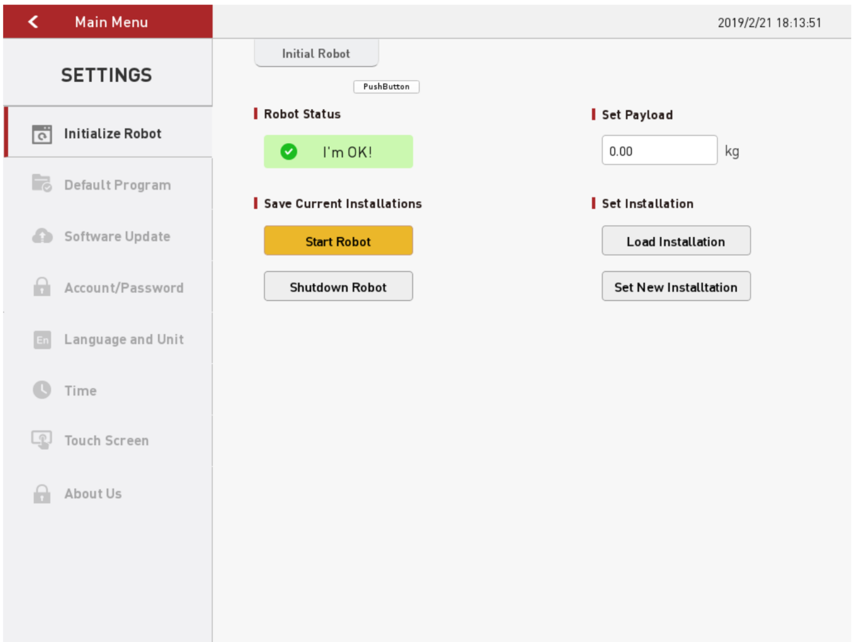


Robot Status: Error!

Set Payload: 0.00 kg

Wait for servo motor ready...

Figure 5- 6 Powering up



Robot Status: I'm OK!

Set Payload: 0.00 kg

Figure 5- 7 Power on

5.2.3 New blank program As shown in Figure 5-8, click “Program Robot” and then select “Empty Program”.

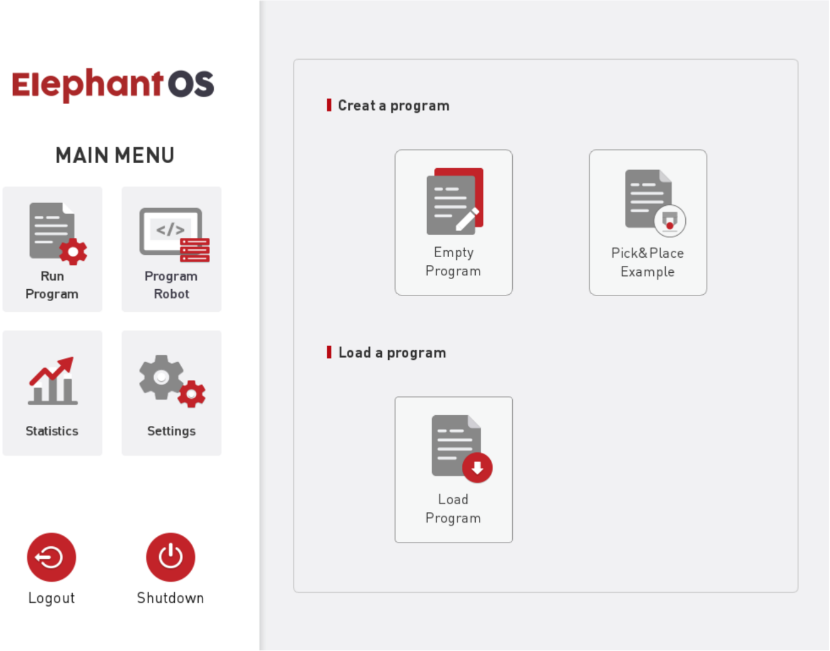


Figure 5- 8 Select "Empty program"

After performing the previous step, enter the program editing interface as shown in Figure 5-9.

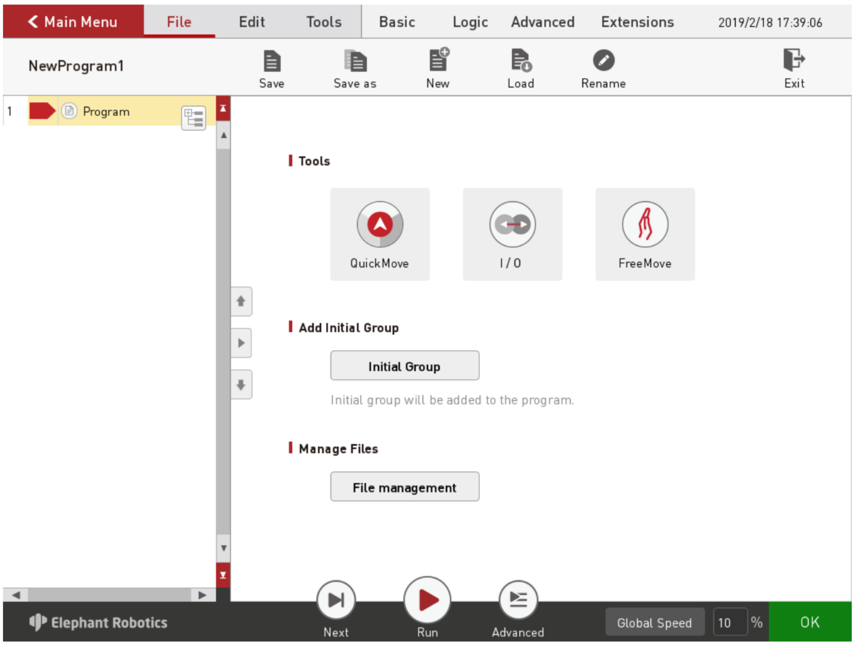


Figure 5- 9 Enter the program editing interface

5.2.4 Add and edit instructions

As shown in Figure 5-10, add two waypoints: absolute point, and teach two points (that is, use the Quickmove to manually operate the robot, control the robot to move to a certain pose, return, click "Save Current Point" The teaching steps of the two points are the same. To verify the save point, press and hold the “Move to this point” button to manually control the robot to move to the teaching point.).

After editing is complete, please note that the program file is saved. Click “Save” in the file option bar as shown in Figure 5-10, and the window shown in Figure 5-11 will pop up. Click on "File Name" and the input keyboard shown in Figure 5-12 will appear. After entering the file name, click "OK". Then go back to the save interface, click "OK", the program file is saved successfully. After the save is successful, as shown in Figure 5-13, the program name in the upper left corner of the program editing interface will be changed.

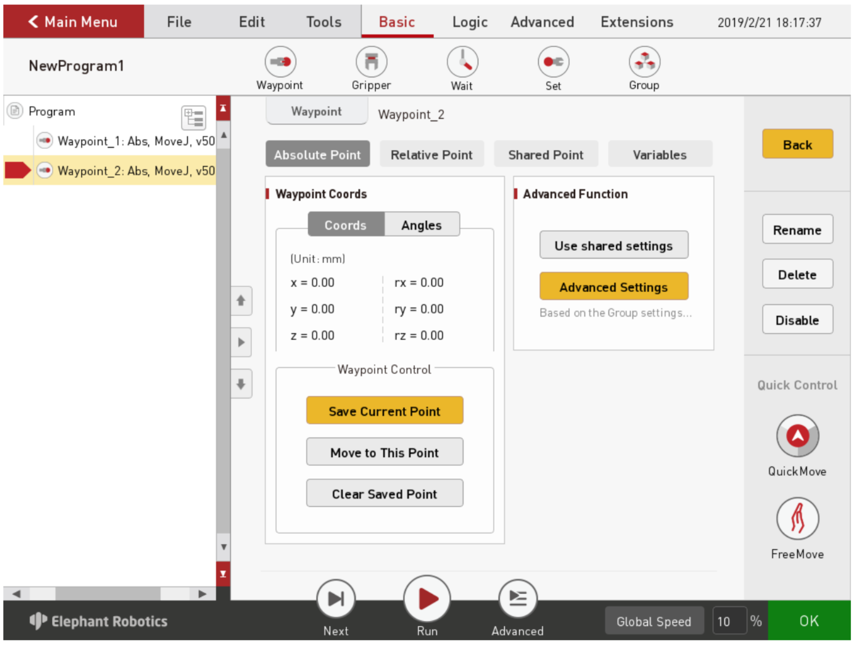
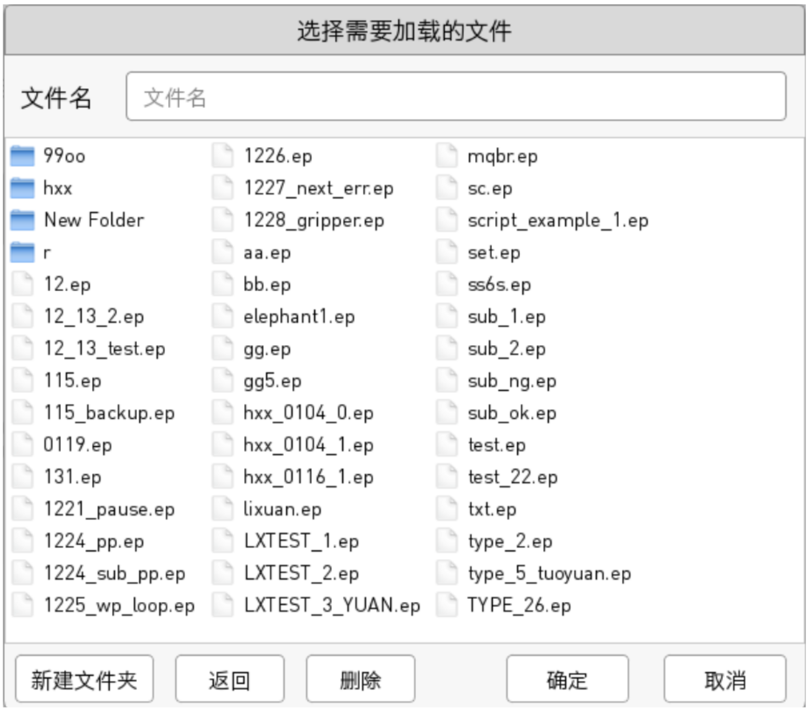


Figure 5- 10 Program editing

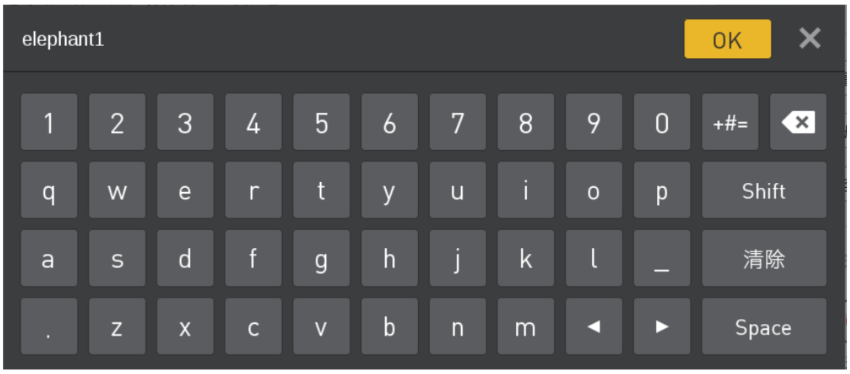


选择需要加载的文件

文件名 文件名

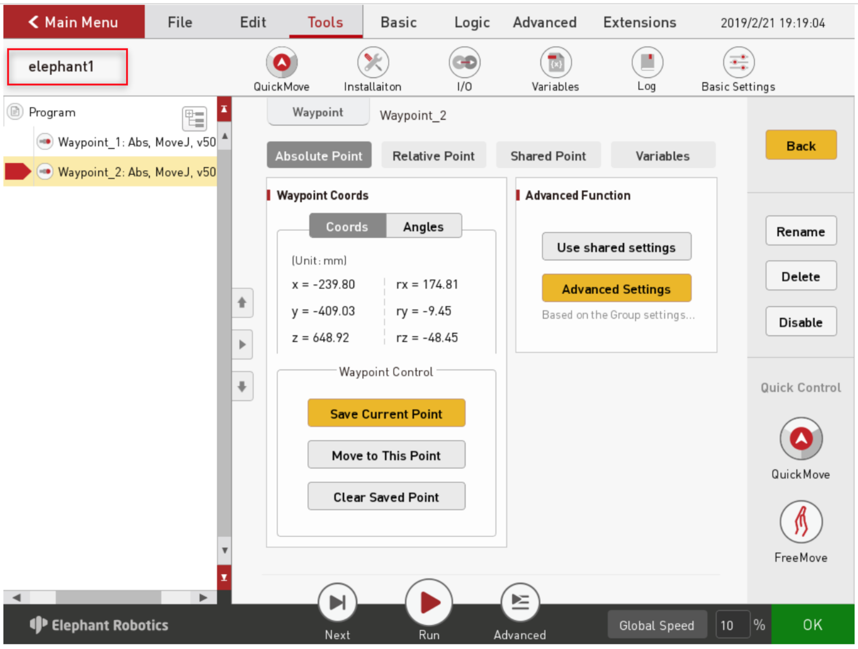
9900
hxx
New Folder
r
12.ep
12_13_2.ep
12_13_test.ep
115.ep
115_backup.ep
0119.ep
131.ep
1221_pause.ep
1224_pp.ep
1224_sub_pp.ep
1225_wp_loop.ep
1226.ep
1227_next_err.ep
1228_gripper.ep
aa.ep
bb.ep
elephant1.ep
gg.ep
gg5.ep
hxx_0104_0.ep
hxx_0104_1.ep
hxx_0116_1.ep
lixuan.ep
LXTEST_1.ep
LXTEST_2.ep
LXTEST_3_YUAN.ep
mqbr.ep
sc.ep
script_example_1.ep
set.ep
ss6s.ep
sub_1.ep
sub_2.ep
sub_ng.ep
sub_ok.ep
test.ep
test_22.ep
txt.ep
type_2.ep
type_5_tuoyuan.ep
TYPE_26.ep

Figure 5- 11



elephant1

Figure 5- 12 Enter the program name



2019/2/21 19:19:04

elephant1

Program
Waypoint_1: Abs, MoveJ, v50
Waypoint_2: Abs, MoveJ, v50

Waypoint_2

(Unit: mm)

x = -239.80 rx = 174.81
y = -409.03 ry = -9.45
z = 648.92 rz = -48.45

Global Speed 10 %

Figure 5- 13 File saved successfully

5.2.5 Program Debugging As shown in Figure 5-14, in addition to the "Next" and "Run" functions provided in the program run control bar, click "Advanced" to enter the more settings interface.

Among them, the "Next" function corresponds to step by step execution of the program, click to run only one step at a time, if you need to continue to run, continue to click "Next." The "Run" function corresponds to automatically running the program once.

In "Advanced ", you can set the number of cycles to run, or you can run in an infinite loop. You can also control whether the program runs in automatic or manual mode. In the automatic mode, you can use "Next", "Run" and cycle operation. In the interface shown in Figure 5-14, select "Manual Run Mode" and then select "Run" or "Infinite Loop" in the loop run. You can enter the running interface in manual operation mode as shown in Figure 5-15.

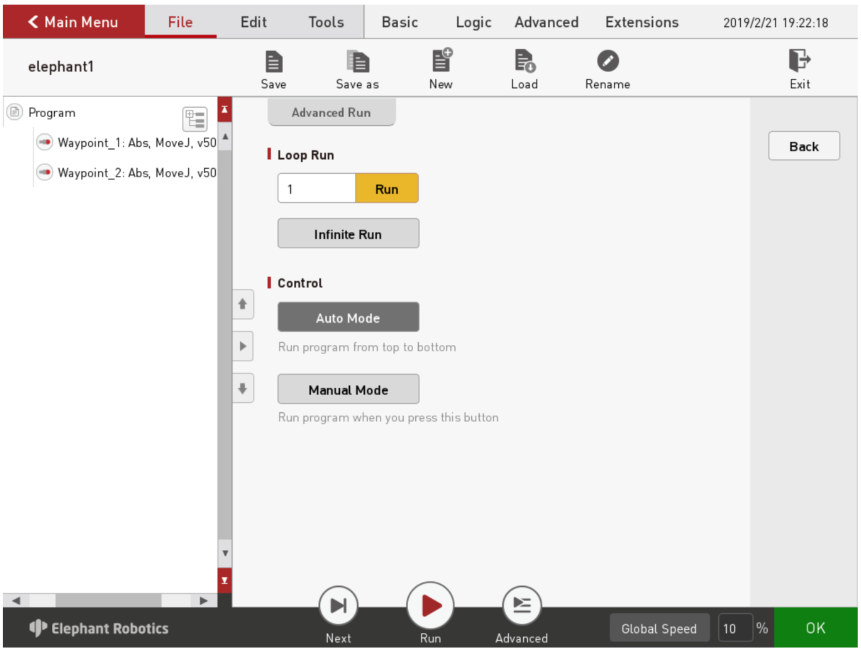


Figure 5- 14 Program debugging

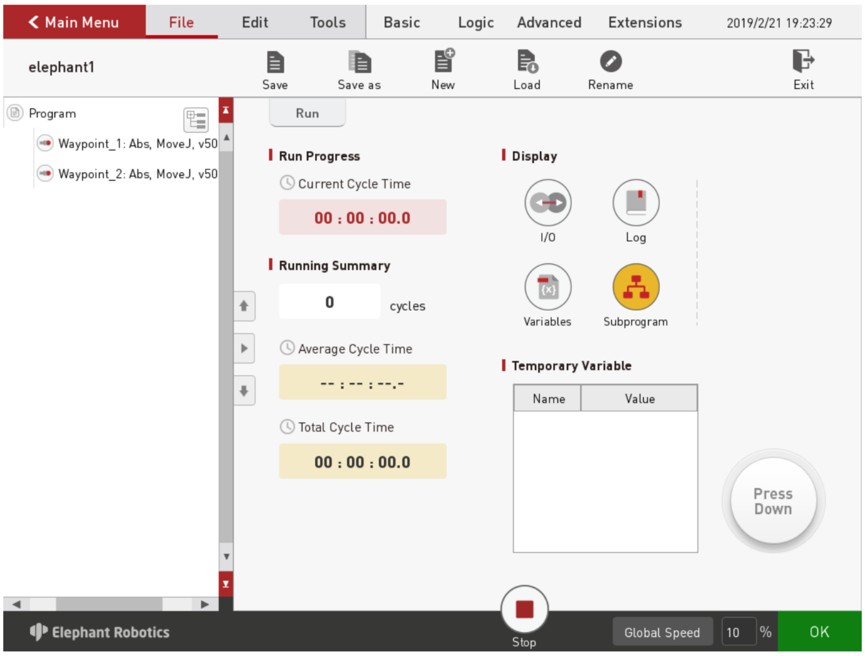
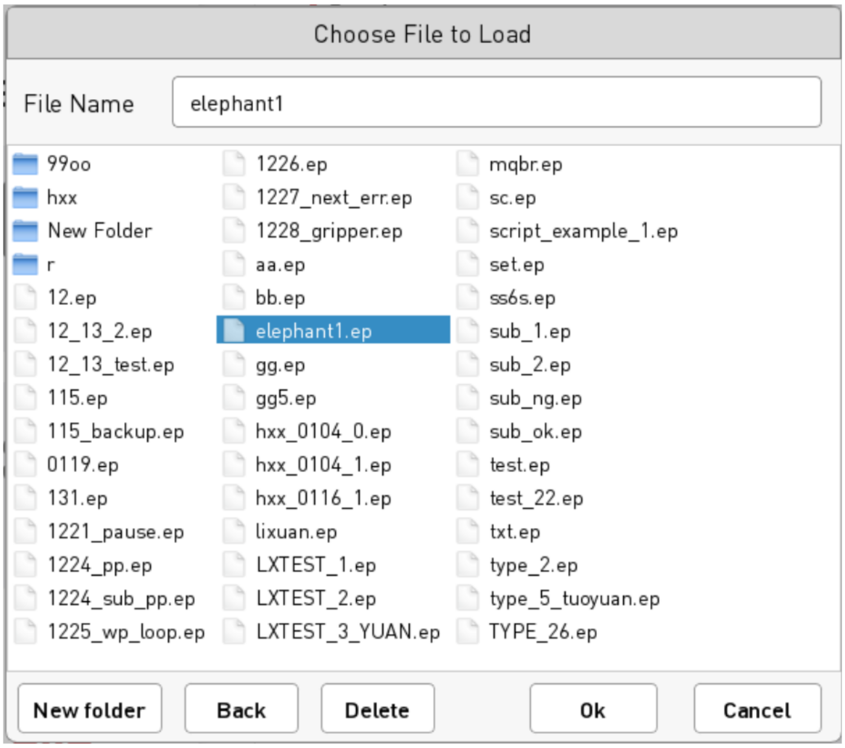


Figure 5- 15 Manual mode to debug the program

If you use manual mode to debug the program, you need to keep pressing the "Press Down" button to continue running. If you release the button, the program pauses and presses again to continue.

5.2.6 **Save and run the program**

If debugging is complete, make sure you have saved the debugged program. After returning to the main menu, select "Run Program". The pop-up window shown in Figure 5-16 will appear. Select the program to complete the debugging and click “OK”.



Choose File to Load

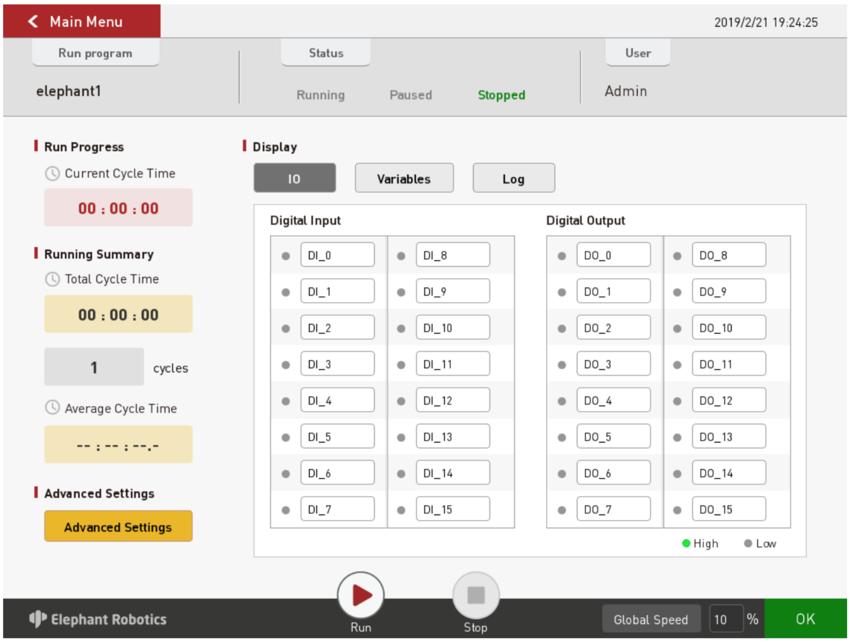
File Name
elephant1

9900
hxx
New Folder
r
12.ep
12_13_2.ep
12_13_test.ep
115.ep
115_backup.ep
0119.ep
131.ep
1221_pause.ep
1224_pp.ep
1224_sub_pp.ep
1225_wp_loop.ep
1226.ep
1227_next_err.ep
1228_gripper.ep
aa.ep
bb.ep
elephant1.ep
gg.ep
gg5.ep
hxx_0104_0.ep
hxx_0104_1.ep
hxx_0116_1.ep
lixuan.ep
LXTEST_1.ep
LXTEST_2.ep
LXTEST_3_YUAN.ep
mqbr.ep
sc.ep
script_example_1.ep
set.ep
ss6s.ep
sub_1.ep
sub_2.ep
sub_ng.ep
sub_ok.ep
test.ep
test_22.ep
txt.ep
type_2.ep
type_5_tuoyuan.ep
TYPE_26.ep

Figure 5- 16 Selection procedure

After selecting the program, it will enter the running program interface as shown in Figure 5-17. In this interface, you can run the program to view the program running information.

If you are sure that the program will continue to run in the near future, you can also select it in the Settings - Default Program. In this way, as long as the system is started, it will automatically jump to the “Run Program” interface. After the power is turned on successfully, click “Run” to run the program.



2019/2/21 19:24:25

Run program
elephant1

Status
Running Paused Stopped

User
Admin

Run Progress

Current Cycle Time
00 : 00 : 00

Running Summary

Total Cycle Time
00 : 00 : 00

1 cycles

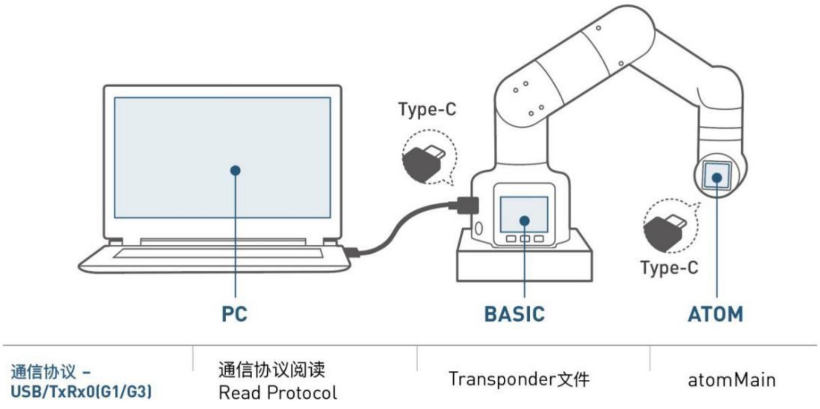
Average Cycle Time
-- : -- : --.-

Global Speed 10 %

Figure 5- 17 Running programs

6 Communications and messages

Note: When use a communication protocol to communicate directly you need to burn **Transponder** in Basic and the latest version of **AtomMain** in Atom.



6.1Communication Settings

Make sure your communication Settings are as follows

- Bus Interface: USB Type-C
- Baud Rate: 115200
- Date Bits: 8
- Parity Check:none
- Stop Bit: 1

6.2 Command Frame Description & Single Instruction Parsing

The main BASIC sends data to the Atom, and the Atom parses the data after receiving it. For example, an instruction containing Return Value will be returned to the BASIC within 500ms.

6.3Format for Sending and Receiving Command Frames

All commands are hexadecimal, and the format of send and receive is the same.

Each communication command must contain the following five parts,part 3 and 4 of which can be null.

- ***1 Command Frame Head: 0xFE 0xFE**
  - Fixed
  - Included
- **Effective Command Length : 0x02 ~ 0x10**
  - The length of all the following commands
  - Included

- **Command Sequence Number: 00 ~ 8F**
  - Various commands have been developed
  - Null
- **Command Content: some**
  - Null
- **End of the Command: 0XFA**
  - Fixed
  - Included

6.4 Single Instruction Parsing

The main BASIC sends data to the Atom, and the Atom parses the data after receiving it. For example, an instruction containing Return Value will be returned to the BASIC within 500ms.

Type	Data Description	Data Length	Description
Command frame	First byte 0	1	The frame head identification, 0xfe
	First byte1	1	The frame head identification, 0xfe
	Data length byte	1	Different instructions correspond to different lengths of data
	Command byte	1	Depends on different commands
Data frame	Data	0-16	Command attached data, depends on different commands
End frame	End the byte	1	Stop bit, 0xfa

1). Atom Power On

Data field	Description	Data
Data[0]	Identify the frame	0xfe
Data[1]	Identify the frame	0xfe
Data[2]	Data length frame	0x02
Data[3]	Instruction frame	0x10
Data[4]	End frame	0xfa

Serial port sending example: FE FE 02 10 FA

No Return Value

2). Atom Power Off

Data field	Description	Data
Data[0]	Identify the frame	0xfe
Data[1]	Identify the frame	0xfe
Data[2]	Data length frame	0x02
Data[3]	Instruction frame	0x11
Data[4]	End frame	0xfa

Serial port sending example: FE FE 02 11 FA

NO Return Value

3). Atom Status Inquiry

Data field	Description	Data
Data[0]	Identify the frame	0xfe
Data[1]	Identify the frame	0xfe
Data[2]	Data length frame	0x02
Data[3]	Instruction frame	0x12
Data[4]	End frame	0xfe

Serial port sending example: FE FE 02 12 FA

Return Value

Data field	Description	Data
Data[0]	Return frame header	0XFE
Data[1]	Return frame header	0XFE
Data[2]	Return data length frame	0X02
Data[3]	Return instruction frame	0X12
Data[4]	Power on/off	0X01/0X00
Data[5]	End frame	0XFA

Serial port sending example: FE FE 02 12 00 FA

4).Read Angle (read position information)

Data field	Description	Data
Data[0]	Identify the frame	0xfe
Data[1]	Identify the frame	0xfe
Data[2]	Data length frame	0x02
Data[3]	Instruction frame	0x20
Data[4]	End frame	0xfa

Serial port sending example: FE FE 02 20 FA

Returns a data structure from Atom

Data field	Description	Data
Data[0]	Return frame header	0XFE
Data[1]	Return frame header	0XFE
Data[2]	Return data length frame	0X0E
Data[3]	Return instruction frame	0X20
Data[4]	Servo 1 high Angle	Angle1_low
Data[5]	Servo 1 low Angle	Angle1_high
Data[6]	Servo 1 high Angle	Angle2_low
Data[7]	Servo 1 low Angle	Angle2_high
Data[8]	Servo 1 high Angle	Angle3_low
Data[9]	Servo 1 low Angle	Angle3_high
Data[10]	Servo 1 high Angle	Angle4_low
Data[11]	Servo 1 low Angle	Angle4_high
Data[12]	Servo 1 high Angle	Angle5_low
Data[13]	Servo 1 low Angle	Angle5_high
Data[14]	Servo 1 high Angle	Angle6_low
Data[15]	Servo 1 low Angle	Angle6_high
Data[16]	End frame	0XFA

Serial port sending example: FE FE 0E 20 06 E6 EA 4E C4 81 0B BD EA C0 02 B6 FA

How to get the angle of joint 1 ?

temp = angle1_low+angle1_high*256

Angle1=(temp > 33000 ? (temp–65536) : temp)/100

(The rest are the same)

5). Send Individual Angles

Data field	Description	Data
Data[0]	Identify the frame	0xfe
Data[1]	Identify the frame	0xfe
Data[2]	Data length frame	0x06
Data[3]	Instruction frame	0x21
Data[4]	Joint number	Joint_no
Data[5]	Angle of rotation	Angle
Data[6]	high Angle	Angle_high
Data[7]	low Angle	Angle_low
Data[8]	End frame	0xfa

Serial port sending example: FE FE 06 21 00 00 00 20 FA

The value of joint NO ranges from 0 to 5

Angle High: Data type Byte

Calculation: The Angle value is multiplied by 100 first to int and then to take the high hexadecimal byte

Angle Low: Data type Byte

Calculation: The Angle value is multiplied by 100 first to int and then to take the high hexadecimal byte

No Return Value

6). Send All Angles

Data field	Description	Data
Data[0]	Identify the frame	0xfe
Data[1]	Identify the frame	0xfe
Data[2]	Data length frame	0x0f
Data[3]	Instruction frame	0x22
Data[4]	Angle 1 is in low_byte	Angle1_low
Data[5]	Angle 1 is in high_byte	Angle1_high
Data[6]	Angle 2 is in low_byte	Angle2_low
Data[7]	Angle 2 is in high_byte	Angle2_high
Data[8]	Angle 3 is in low_byte	Angle3_low
Data[9]	Angle 3 is in high_byte	Angle3_high
Data[10]	Angle 4 is in low_byte	Angle4_low
Data[11]	Angle 4 is in high_byte	Angle4_high
Data[12]	Angle 5 is in low_byte	Angle5_low
Data[13]	Angle 5 is in high_byte	Angle5_high
Data[14]	Angle 6 is in low_byte	Angle6_low
Data[15]	Angle 6 is in high_byte	Angle6_high
Data[16]	Specified speed	Sp
Data[17]	End frame	0xfa

Serial port sending example: FE FE 0F 22 06 E6 EA 4E C4 81 0B BD EA C0 02 B6 FA

Angle High: Data type Byte

Calculation: The Angle value of Joint 1 is multiplied by 100 first to int and then to take the high hexadecimal byte

Angle Low: Data type Byte

Calculation: The Angle value of Joint 1 is multiplied by 100 first to int and then to take the high hexadecimal byte

(The rest are the same)

No Return Value

7). Read All Coordinates

Data field	Description	Data
Data[0]	Identify the frame	0xfe
Data[1]	Identify the frame	0xfe
Data[2]	Data length frame	0x02
Data[3]	Instruction frame	0x23
Data[4]	End frame	0xfa

Serial port sending example: FE FE 02 23 FA

Returns a data structure from Atom

Data field	Description	Data
Data[0]	Return frame header	0XFE
Data[1]	Return frame header	0XFE
Data[2]	Return data length frame	0X0E
Data[3]	Return instruction frame	0X23
Data[4]	Specified x is in a low coordinate	x_high
Data[5]	Specified x is in a high coordinate	x_low
Data[6]	Specified y is in a low coordinate	y_ high
Data[7]	Specified y in a high coordinate	y_ low
Data[8]	Specified z is in a low coordinate	z_ high
Data[9]	Specified z in a high coordinate	z_low
Data[10]	Specified rx is in a low coordinate	rx_high
Data[11]	Specified rx is in a high coordinate	rx_low
Data[12]	Specified ry is in a low coordinate	ry_high
Data[13]	Specified ry in a high coordinate	ry_low
Data[14]	Specified rz is in a low coordinate	rz_high
Data[15]	Specified rz is in a high coordinate	rz_low
Data[16]	End frame	0XFA

How to get x-coordinate ?

temp = x_low + x_high*256

x-coordinate =(temp > 33000 ?(temp – 65536) : temp)/10

(same as the y/z-coordinate)

How to get rx-coordinate ?

temp = rx_low + rx_high*256

rx-coordinate =(temp > 33000 ?(temp – 65536) : temp) /10

(same as the ry/rz-coordinate)

8).Send Individual Coordinates

Data field	Description	Data
Data[1]	Identify the frame	0xfe
Data[2]	Data length frame	0x06
Data[3]	Instruction frame	0x24
Data[4]	Specified coordinate	X/y/z/rx/ry/rz
Data[5]	Specified xyz/rxryrz is in a low parameter	Xyz/ rxryrz_low
Data[6]	Specified xyz/rxryrz is in a high parameter	Xyz/rxryrz_high
Data[7]	Specified speed	Sp
Data[8]	End frame	0xfa

Set the x-coordinate to be 100 and the target speed to be 20

Serial port sending example: FE FE 06 24 00 00 64 20 FA

Specify coordinate axis: data type Byte

Value range: 0~5

xyz_high: Data type Byte

Calculation: The x/y/z coordinate value is multiplied by 10 and then to take the high hexadecimal byte

xyz_low: Data type Byte

Calculation: The x/y/z coordinate value is multiplied by 10 and then to take the low hexadecimal byte

rxyz_high: Data type Byte

Calculation: The rx/ry/rz coordinate value is multiplied by 10 and then to take the high hexadecimal byte

rxyz_low: Data type Byte

Calculation: The rx/ry/rz coordinate value is multiplied by 10 and then to take the low hexadecimal byte

No Return Value

9).Send All Coordinates

Data field	Description	Data
Data[0]	Identify the frame	0xfe
Data[1]	Identify the frame	0xfe
Data[2]	Data length frame	0x10
Data[3]	Instruction frame	0x25
Data[4]	Specified x is in a low coordinate	X_low
Data[5]	Specified x is in a high coordinate	X_high
Data[6]	Specified y is in a low coordinate	Y_low
Data[7]	Specified y in a high coordinate	Y_high
Data[8]	Specified z is in a low coordinate	Z_low
Data[9]	Specified z in a high coordinate	Z_high
Data[10]	Specified rx is in a low coordinate	Rx_low
Data[11]	Specified rx is in a high coordinate	Rx_high
Data[12]	Specified ry is in a low coordinate	Ry_low
Data[13]	Specified ry in a high coordinate	Ry_high
Data[14]	Specified rz is in a low coordinate	Rz_low
Data[15]	Specified rz is in a high coordinate	Rz_high
Data[16]	Specified speed	Sp
Data[17]	Mode	Mode
Data[18]	End frame	0xfa

Set the target point at the end of the robot (-14，-27，275，-89.5, 0.7，-90.7)

set the target speed to be 20

Serial port sending example: FE FE 10 25 FF 74 FE EE 0A C1 DD 05 00 48 DC 95 32 01 FA

x_high: Data type Byte

Calculation: The x coordinate value is multiplied by 10 and then to take the high hexadecimal byte

x_low：Data type Byte

Calculation: The x coordinate value is multiplied by 10 and then to take the low hexadecimal byte

(same as the y/z-coordinate)

rx_high: Data type Byte

Calculation: The rx coordinate value is multiplied by 10 and then to take the high hexadecimal byte rx_low: Data type Byte

Calculation: The rx coordinate value is multiplied by 10 and then to take the low hexadecimal byte

(same as the ry/rz-coordinate)

No Return Value

10).Specified Point Arrival Detection (under development)

Data field	Description	Data
Data[0]	Identify the frame	0xfe
Data[1]	Identify the frame	0xfe
Data[2]	Data length frame	0x10
Data[3]	Instruction frame	0x25
Data[4]	x is in a low coordinate	X_low
Data[5]	x is in a high coordinate	X_high
Data[6]	y is in a low coordinate	Y_low
Data[7]	y in a high coordinate	Y_high
Data[8]	z is in a low coordinate	Z_low
Data[9]	z in a high coordinate	Z_high
Data[10]	rx is in a low coordinate	Rx_low
Data[11]	rx is in a high coordinate	Rx_high
Data[12]	ry is in a low coordinate	Ry_low
Data[13]	ry in a high coordinate	Ry_high
Data[14]	rz is in a low coordinate	Rz_low
Data[15]	rz is in a high coordinate	Rz_high
Data[16]	Is_linear	type
Data[17]	End frame	0xfa

Determine whether the manipulator has reached the specified point

Serial port sending example: FE FE 10 25 FF 74 FE EE 0A C1 DD 05 00 48 DC 95 32 01 FA

x_high: Data type Byte

Calculation: The x coordinate value is multiplied by 10 and then to take the high hexadecimal byte

x_low：Data type Byte

Calculation: The x coordinate value is multiplied by 10 and then to take the low hexadecimal byte

(same as the y/z-coordinate)

rx_high: Data type Byte

Calculation: The rx coordinate value is multiplied by 10 and then to take the high hexadecimal byte

rx_low: Data type Byte

Calculation: The rx coordinate value is multiplied by 10 and then to take the low hexadecimal byte

(same as the ry/rz-coordinate)

Type: Data type Byte (Not yet in use)

Returns a data structure

Data field	Description	Data
Data[0]	Return frame	0XFE
Data[1]	Return frame	0XFE
Data[2]	Return data length frame	0X03
Data[3]	Return instruction frame	0X2A
Data[4]	InPosition/noInPosition	0X01/0X00
Data[5]	End frame	0XFA

It has reached a point;

Serial port sending example: FE FE 03 2A 00 FA

11).Motion Detection

Data field	Description	Data
Data[0]	Identify the frame	0XFE
Data[1]	Identify the frame	0XFE
Data[2]	Data length frame	0X02
Data[3]	Instruction frame	0X2B
Data[4]	End frame	0XFA

Check whether the robot is moving

Serial port sending example: FE FE 02 2B FA

Returns a data structure

Data field	Description	Data
Data[0]	Return frame	0XFE
Data[1]	Return frame	0XFE
Data[2]	Return data length frame	0X02
Data[3]	Return instruction frame	0X2B
Data[4]	Not running/no data - running	0X00/0X01
Data[5]	End frame	0XFA

Serial port sending example: FE FE 02 2B 00 FA

12).Jog-Direction Motion

Data field	Description	Data
Data[0]	Identify the frame	0xfe
Data[1]	Identify the frame	0xfe
Data[2]	Data length frame	0x05
Data[3]	Instruction frame	0x30
Data[4]	Joint number	Joint
Data[5]	Joint direction	Direction
Data[6]	Specified speed	Sp
Data[7]	End frame	0xfa

Set Joint 1 to rotate clockwise at 50% speed

Serial port sending example: FE FE 05 30 01 01 32 FA

Joint No. ranges from 1~6

di: Data type Byte, ranges from 0~1

sp: Data type Byte, ranges from 0-100%

No Return Value

13).Jog-Coordinate Motion

Data field	Description	Data
Data[0]	Identify the frame	0xfe
Data[1]	Identify the frame	0xfe
Data[2]	Data length frame	0x05
Data[3]	Instruction frame	0x33
Data[4]	Specified coordinates	X1 y2 z3 rx4 ry5 rz6
Data[5]	Joint direction	Direction
Data[6]	Specified speed	Sp
Data[7]	End frame	0xfa

Set the end to move at a speed of 50% counterclockwise toward the X-axis

Serial port sending example: FE FE 05 32 01 00 32 FA

Axis_number: Data type Byte(x = 0 ,y,z,rx,ry,rz) Range from 1~6

di:Data type Byte, ranges from 0~1

sp: Data type Byte, ranges from 0-100%

No Return Value

14).jog stop

Data field	Description	Data
Data[0]	Identify the frame	0xfe
Data[1]	Identify the frame	0xfe
Data[2]	Data length frame	0x02
Data[3]	Instruction frame	0x34
Data[4]	End frame	0xfa

Jog stop to move

Serial port sending example: FE FE 02 34 FA

No Return Value

15).send potential value

Data field	Description	Data
Data[0]	Identify the frame	0xfe
Data[1]	Identify the frame	0xfe
Data[2]	Data length frame	0x05
Data[3]	Instruction frame	0x3a
Data[4]	Joint number	Joint
Data[5]	Encoder is in a Low position	Encoder_low
Data[6]	Encoder is in a High position	Encoder_high
Data[7]	End frame	0xfa

1. Return data structure

```
| Data domain | Description | Data |
| - | - | - |
| Data[0] | recognition frame | 0XFE |
| Data[1] | recognition frame | 0XFE |
| Data[2] | data length frames | 0X05 |
| Data[3] | instruction frame | 0X3A |
| Data[4] | Joint serial number | Joint |
| Data[5] | High value of steering gear potential | Encoder\_high |
| Data[6] | Low potential of steering gear | Encoder\_low |
| Data[7] | End frame | 0XFA |
```

Example, set joint 2 to 2249 potential.

Example of serial port sending: FE FE 05 3A 01 08 C9 FA

Range of joint serial numbers: 0~5

byte Joint： data types

byte Encoder_high： data types

Calculation method: high bit value (hexadecimal) is taken

byte Encoder_low： data types

Calculation method: take potential value (hexadecimal) low position

Return Value： No

16.Gets the potential value;)

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X03
Data[3]	Instruction frame	0X3B
Data[4]	Joint serial number	joint
Data[5]	End frame	0XFA

Get the potential value of steering gear 1

Example of serial port sending: FE FE 03 3B 00 FA

Range of joint serial numbers: 1~6

Return data structure

Data domain	Description	Data
Data[0]	Return recognition frame	0XFE
Data[1]	Return recognition frame	0XFE
Data[2]	Returns data length frames	0X04
Data[3]	Returns the instruction frame	0X3B
Data[4]	High value of steering gear potential	Encoder_high
Data[5]	Low potential of steering gear	Encoders_low
Data[6]	End frame	0XFA

Example of serial port return: FE FE 04 3B 08 C9 FA

Potential =2249

How to calculate the potential value

Potential value = low potential value + high potential value *256

---

1. Transmission of potential values for six steering gear

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X15
Data[3]	Instruction frame	0X3C
Data[4]	1 steering gear with high potential	encoder_1_high
Data[5]	Low byte potential of steering gear 1	encoder_1_low
Data[6]	2 steering gear with high potential	encoder_2_high
Data[7]	Low byte potential of steering gear 2	encoder_2_low
Data[8]	3 steering gear with high potential	encoder_3_high
Data[9]	Low byte potential of steering gear 3	encoder_3_low
Data[10]	4 steering gear with high potential bytes	encoder_4_high
Data[11]	Low byte potential of steering gear 4	encoder_4_low
Data[12]	5 steering gear with high potential	encoder_5_high
Data[13]	Low byte potential of steering gear 5	encoder_5_low
Data[14]	6 steering gear with high potential	encoder_6_high
Data[15]	Low byte potential of steering gear 6	encoder_6_low
Data[16]	Specify speed	Sp
Data[17]	End frame	0XFA

Send potential values for all motors

Example of serial port sending: FE FE 15 3C 00 00 00 00 00 00 00 00 00 00 00 00 20FA

(separate potential values sent over reference)

byte encoder_1_high： data types

Calculation method :1 steering gear potential value converted to int type and then hexadecimal high byte

byte encoder_1_low： data types

Calculation method :1 steering gear potential value converted to int type and then hexadecimal low byte

(Other Same)

Sp： data type byte range 0~100

Return Value： No

1. Read speed

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X02
Data[3]	Instruction frame	0X40
Data[4]	End frame	0XFA

Example of serial port sending: FE FE 02 40 FA

Return data structure

Data domain	Description	Data
Data[0]	Return recognition frame	0XFE
Data[1]	Return recognition frame	0XFE
Data[2]	Returns data length frames	0X03
Data[3]	Returns the instruction frame	0X40
Data[4]	Specify speed	Sp
Data[5]	End frame	0XFA

Example of serial port return: FE FE 03 40 32 FA

---

1. Set speed

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X02
Data[3]	Instruction frame	0X41
Data[4]	Specify speed	sp
Data[5]	End frame	0XFA

Example of serial port sending: FE FE 02 41 32 FA

Return Value： No

---

1. Read FeedOverride(not open yet)

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X02
Data[3]	Instruction frame	0X42
Data[4]	End frame	0XFA

Return Value： No

---

1. Read acceleration (not yet open)

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X02
Data[3]	Instruction frame	0X44
Data[4]	End frame	0XFA

---

1. Read the minimum angle of the joint

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X03
Data[3]	Instruction frame	0X4A
Data[4]	Joint steering gear serial number	Joint_number
Data[5]	End frame	0XFA

Read

Example of serial port sending: FE FE 03 4A 00 FA

joint_no range: 0~5

Return data structure

Data domain	Description	Data
Data[0]	Return recognition frame	0XFE
Data[1]	Return recognition frame	0XFE
Data[2]	Returns data length frames	0X04
Data[3]	Returns the instruction frame	0X4A
Data[4]	High angle of steering gear	Angle_high
Data[5]	Low angle of steering gear	Angle_low
Data[6]	End frame	0XFA

Example of serial port return: FE FE 04 4A 01 44 FA

Angle =90

How to get the minimum angle

temp =angle1_low+angle1_high*256

Angle1= temp \33000? (temp –65536): temp)/10

Calculation method: low angle value + high angle value multiplied by 256 to determine whether it is greater than 33000 if it is greater than 33000 then subtract 65536 and divide 10 if directly divided by 10

---

1. Read the maximum angle of the joint

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X03
Data[3]	Instruction frame	0X4B
Data[4]	Joint steering gear serial number	joint_number
Data[5]	End frame	0XFA

Example of serial port sending: FE FE 03 4B 01 FA

Return data structure

Data domain	Desccription	Data
Data[0]	Return recognition frame	0XFE
Data[1]	Return recognition frame	0XFE
Data[2]	Returns data length frames	0X04
Data[3]	Returns the instruction frame	0X4B
Data[4]	High angle of steering gear	Angle_high
Data[5]	Low angle of steering gear	Angle_low
Data[6]	End frame	0XFA

Example of serial port return: FE FE 04 4B 01 44 FA

joint_no range 0~5

How to get the maximum angle of the joint

temp =angle1_low+angle1_high*256

Angle1= temp \33000? (temp –65536): temp)/10

Calculation method: low angle value + high angle value multiplied by 256 to determine whether it is greater than 33000 if it is greater than 33000 then subtract 65536 and divide 10 if directly divided by 10

---

1. View connection

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X03
Data[3]	Instruction frame	0X50
Data[4]	End frame	0XFA

Example of serial port sending: FE FE 03 50 FA

Return data structure

Data domain	Description	Data
Data[0]	Return recognition frame	0XFE
Data[1]	Return recognition frame	0XFE
Data[2]	Returns data length frames	0X03
Data[3]	Instruction frame	0X50
Data[4]	Connected/unconnected	0X01/0X00
Data[5]	End frame	0XFA

Example of serial port return: FE FE 03 50 00 FA

---

1. Check to see if the steering gear is all powered up

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X02
Data[3]	Instruction frame	0X51
Data[4]	End frame	0XFA

Example of serial port sending: FE FE 02 51 FA

Return data structure

Data domain	Description	Data
Data[0]	Return recognition frame	0XFE
Data[1]	Return recognition frame	0XFE
Data[2]	Returns data length frames	0X03
Data[3]	Instruction frame	0X51
Data[4]	Electricity on/off	0X01/0X00
Data[5]	End frame	0XFA

Example of serial port return: FE FE 03 51 00 FA

---

1. Read steering gear status

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X04
Data[3]	Instruction frame	0X53
Data[4]	Joint steering gear serial number	joint_number
Data[5]	Data	data_id
Data[6]	End frame	0XFA

Read Parameter of position P ratio of steering gear 1

Example of serial port sending: FE FE 04 53 00 21 FA

joint_no range 0~5

byte Data_id： data types

The values are taken in the table below

Address	Function	value range	initial value	value analysis
20	LED alarm	0-254	0	1\0 = open or close LED alarm
21	Speed loop P	0-254	123joint8,456joint5	control the proportional coefficient of the motor
22	Position ring I	0-254	123joint20,456joint13	control the differential coefficient of the motor
23	Position ring D	0-254	0	control the integral coefficient of the motor
24	minimum starting force	0-1000	0	set the minimum output torque1000 = 100%

Return data structure

Data domain	Description	Data
Data[0]	Return recognition frame	0XFE
Data[1]	Return recognition frame	0XFE
Data[2]	Returns data length frames	0X03
Data[3]	Returns the instruction frame	0X53
Data[4]	Return data	data_state
Data[5]	End frame	0XFA

Example of serial port return: FE FE 03 53 00 FA

1. Set steering gear zero

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X03
Data[3]	Instruction frame	0X55
Data[4]	Joint steering gear serial number	joint_number
Data[5]	End frame	0XFA

Example of serial port sending: FE FE 03 55 00 FA

Return Value： No

1. Brake single motor (not yet open)

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X03
Data[3]	Instruction frame	0X56
Data[4]	Joint steering gear serial number	joint_number
Data[5]	End frame	0XFA

Example of serial port sending: FE FE 03 56 00 FA

Return Value： No

---

1. Setting atom Pin Mode

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X04
Data[3]	Instruction frame	0X60
Data[4]	Number of pins	pin_no
Data[5]	Pin mode	pin_mode
Data[6]	End frame	0XFA

atom pin16 set to output mode 0

Example of serial port sending: FE FE 04 60 19 00 FA

byte Pin_no： data types

byte Pin_mode： data types

Return Value： No

---

1. The program paused

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X02
Data[3]	Instruction frame	0X26
Data[4]	End frame	0XFA

Example of serial port sending: FE FE 02 26 FA

Return Value： No

---

1. The program continues to run

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X02
Data[3]	Instruction frame	0X27
Data[6]	End frame	0XFA

Example of serial port sending: FE FE 02 27 FA

Return Value： No

1. The program stops running

Data	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X02
Data[3]	Instruction frame	0X28
Data[4]	End frame	0XFA

Example of serial port sending: FE FE 02 28 FA

Return Value： No

---

1. Set steering gear status

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X05
Data[3]	Instruction frame	0X52
Data[4]	Joint steering gear serial number	servo_no
Data[5]	Steering gear status	servo_state
Data[6]	Data	servo_data
Data[7]	End frame	0XFA

Set position P ratio Parameter 1

Example of serial port sending: FE FE 05 52 00 21 01 FA

Return Value： No

---

1. Robot free mode (turn off all torque output)

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X02
Data[3]	Instruction frame	0X13
Data[4]	End frame	0XFA

Example of serial port sending: FE FE 02 13 FA

Return Value： No

1. Set atom screen RGB light color

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X05
Data[3]	Instruction frame	0X6A
Data[4]	R	R
Data[5]	G	G
Data[6]	B	B
Data[7]	End frame	0XFA

RGB set to blue

Example of serial port sending: FE FE 05 6A 00 00 FF FA

Return Value： No

1. Setting Claw Angle

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X03
Data[3]	Instruction frame	0X66
Data[4]	Claw Data	Gripper_data
Data[5]	End frame	0XFA

Example of serial port sending: FE FE 03 66 00 FA

Gripper_data： data type byte range 0-1

Return Value： No

---

1. Set FeedOverride (not open yet)

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X04
Data[3]	Instruction frame	0X43
Data[4]	Feed_override high	Feed_override_high
Data[5]	Feed_override low	Feed_override_low
Data[6]	End frame	0XFA

1. Set acceleration (not open yet)

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X04
Data[3]	Instruction frame	0X45
Data[4]	High acceleration values	acceleration_high
Data[5]	Acceleration low	acceleration_low
Data[6]	End frame	0XFA

byte acceleration_high： data types

Calculation method: the acceleration value multiplied by 10 is converted to int format and then hexadecimal high byte

byte acceleration_low： data types

Calculation method: the acceleration value multiplied by 10 is converted to int format and then hexadecimal low byte

1. Set minimum angle of joint

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X05
Data[3]	Instruction frame	0X4C
Data[4]	Joint steering gear serial number	Joint
Data[5]	High angle	Angle_high
Data[6]	Low angle	Angle_low
Data[7]	End frame	0XFA

Set the minimum angle to 90

Example of serial port sending: FE FE 05 4C 00 01 44 FA

Joint range 0~5

byte angle_high： data types

Calculation: angle value multiplied by 10 converted to int form and then hexadecimal high byte

byte angle_low： data types

Calculation: angle value multiplied by 10 converted to int form and then hexadecimal low byte

Return Value： No

---

1. Set the maximum angle of the joint

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X05
Data[3]	Instruction frame	0X4D
Data[4]	Joint steering gear serial number	Joint
Data[5]	High angle	Angle_high
Data[6]	Low angle	Angle_low
Data[7]	End frame	0XFA

<<<<<<< HEAD set maximum angle to 90

Example of serial port sending: FE FE 05 4D 00 01 44 **FA**

Example of serial port sending: FE FE 02 12 FA

Return Value： Yes

Joint range 0~5

byte of <<<<<<<HEAD angle_high： data types

Calculation method: angle value multiplied by 10 converted to int form and then hexadecimal high byte

Example of serial port sending: FE FE 02 20 FA

Return Value： Yes

Example of serial port return: FE FE 0E 20 06 E6 EA 4E C4 81 0B BD EA C0 02 B6 FA

byte angle_low： data types

Calculation: angle value multiplied by 10 converted to int form and then hexadecimal low byte

Return Value： No

---

1. Set the Tool Coordinate System

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X14
Data[3]	Instruction frame	0X81
Data[4]	X coordinates high byte	x_high
Data[5]	X coordinates low byte	x_low
Data[6]	Y coordinates high byte	y_high
Data[7]	Y coordinates low byte	y_low
Data[8]	Z coordinates high byte	z_high
Data[9]	Z coordinates low byte	z_low
Data[10]	RX coordinates high byte	rx_high
Data[11]	RX coordinates low byte	rx_low
Data[12]	RY coordinates high byte	ry_high
Data[13]	RY coordinates low byte	ry_low
Data[14]	RZ coordinates high byte	rz_high
Data[15]	RZ coordinates low byte	rz_low
Data[16]	End frame	0XFA

Set tool coordinate system (-14,-27,275,-89.5,0.7,-90.7),

Example of serial port sending: FE FE 14 81FF 74FE EE 0A C1DD 05 00 48DC 95FA

byte x_high： data types

Calculation: x coordinates multiplied by 10 and then high bytes in hexadecimal

byte x_low： data types

Calculation: x coordinates multiplied by 10 and then hexadecimal low bytes

(y axis coordinates z axis coordinates are the same)

byte rx_high： data types

Calculation: rx coordinate value multiplied by 100 and then hexadecimal high byte

byte rx_low： data types

Calculation: rx coordinate value multiplied by 100 and then hexadecimal low byte

(ry axis coordinates rz axis coordinates are the same)

Return Value： No

---

1. Setting the world coordinate system

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X14
Data[3]	Instruction frame	0X83
Data[4]	X coordinates high byte	x_high
Data[5]	X coordinates low byte	x_low
Data[6]	Y coordinates high byte	y_high
Data[7]	Y coordinates low byte	y_low
Data[8]	Z coordinates high byte	z_high
Data[9]	Z coordinates low byte	z_low
Data[10]	RX coordinates high byte	rx_high
Data[11]	RX coordinates low byte	rx_low
Data[12]	RY coordinates high byte	ry_high
Data[13]	RY coordinates low byte	ry_low
Data[14]	RZ coordinates high byte	rz_high
Data[15]	RZ coordinates low byte	rz_low
Data[16]	End frame	0XFA

Set the world coordinate system (-14,-27,275,-89.5,0.7,-90.7),

Example of serial port sending: FE FE 14 83 FF 74 FE EE 0A C1 DD 05 00 48 DC 95 FA

byte x_high： data types

Calculation: x coordinates multiplied by 10 and then high bytes in hexadecimal

byte x_low： data types

Calculation: x coordinates multiplied by 10 and then hexadecimal low bytes

(y axis coordinates z axis coordinates are the same)

byte rx_high： data types

Calculation: rx coordinate value multiplied by 100 and then hexadecimal high byte

<<<<<<<HEAD

<<<<<<< HEAD

byte rx_low： data types

Example of serial port sending: FE FE 05 30 01 01 32 FA Range of joint serial numbers 1~6 di：data type byte value range 0 and 1 sp：data type range 0-100%

no Return Value

f8f31e0282f58b3f27f944c8d7a6ac99
dc0185de

Calculation: rx coordinate value multiplied by 100 and then hexadecimal low byte

(ry axis coordinates rz axis coordinates are the same)

Return Value： No

---

1. Gets the tool coordinate

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X02
Data[3]	Instruction frame	0X82
Data[4]	End frame	0XFA

Example of serial port sending: FE FE 02 82 FA

Return data structure

Data domain	Description	Data
Data[0]	Return recognition frame	0XFE
Data[1]	Return recognition frame	0XFE
Data[2]	Returns data length frames	0X14
Data[3]	Returns the instruction frame	0X82
Data[4]	X coordinates high byte	x_high
Data[5]	X coordinates low byte	x_low
Data[6]	Y coordinates high byte	y_high
Data[7]	Y coordinates low byte	y_low
Data[8]	Z coordinates high byte	z_high
Data[9]	Z coordinates low byte	z_low
Data[10]	RX coordinates high byte	rx_high
Data[11]	RX coordinates low byte	rx_low
Data[12]	RY coordinates high byte	ry_high
Data[13]	RY coordinates low byte	ry_low
Data[14]	RZ coordinates high byte	rz_high
Data[15]	RZ coordinates low byte	rz_low
Data[16]	End frame	0XFA

Example of serial port return: FE FE 14 82 FF 74 FE EE 0A C1DD 05 00 48 DC 95 FA

byte x_high： data types

Mode of calculation: x coordinates multiplied by 10 are converted to int type and then hexadecimal high bytes

byte x_low： data types

Mode of calculation: x coordinates multiplied by 10 are converted to int type and then hexadecimal low bytes

(y axis coordinates z axis coordinates are the same)

byte rx_high： data types

Mode of calculation: rx coordinates multiplied by 100 are converted to int type and then hexadecimal high bytes

byte rx_low： data types

Mode of calculation: rx coordinates multiplied by 100 are converted to int type and then hexadecimal low bytes

(ry axis coordinates rz axis coordinates are the same)

1. Get the world coordinate system

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X02
Data[3]	Instruction frame	0X84
Data[4]	End frame	0XFA

Example of serial port sending: FE FE 02 84 FA

Return data structure

Data domain	Description	Data
Data[0]	Return recognition frame	0XFE
Data[1]	Return recognition frame	0XFE
Data[2]	Returns data length frames	0X14
Data[3]	Returns the instruction frame	0X84
Data[4]	X coordinates high byte	x_high
Data[5]	X coordinates low byte	x_low
Data[6]	Y coordinates high byte	y_high
Data[7]	Y coordinates low byte	y_low
Data[8]	Z coordinates high byte	z_high
Data[9]	Z coordinates low byte	z_low
Data[10]	RX coordinates high byte	rx_high
Data[11]	RX coordinates low byte	rx_low
Data[12]	RY coordinates high byte	ry_high
Data[13]	RY coordinates low byte	ry_low
Data[14]	RZ coordinates high byte	rz_high
Data[15]	RZ coordinates low byte	rz_low
Data[16]	End frame	0XFA

Example of serial port return: FE FE 14 84 FF 74 FE EE 0A C1 DD 05 00 48 DC 95 FA

---

1. Set up flange base coordinate system

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X03
Data[3]	Instruction frame	0X85
Data[4]	RFType	0x00/0x01
Data[5]	End frame	0XFA

Example of serial port sending: FE FE 03 85 00FA

byte RFType： data types

Value range :0~1 BASE =0; WORLD BASE =1;

The RFType：：BASE is to take the robot base as the base coordinate, RFType：：WORLD the world coordinate system as the base coordinate.

---

1. Get the flange base coordinate system

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X02
Data[3]	Instruction frame	0X86
Data[4]	End frame	0XFA

Example of serial port sending: FE FE 02 86 FA

Return data structure

Data domain	Description	Data
Data[0]	Return recognition frame	0XFE
Data[1]	Return recognition frame	0XFE
Data[2]	Returns data length frames	0X03
Data[3]	Returns the instruction frame	0X86
Data[4]	RFType	0x00/0x01
Data[5]	End frame	0XFA

Example of serial port return: FE FE 03 86 00 FA

---

1. Set the terminal coordinate system

Data domain	Description	Data
Data[0]	Return recognition frame	0XFE
Data[1]	Return recognition frame	0XFE
Data[2]	Returns data length frames	0X03
Data[3]	Returns the instruction frame	0X89
Data[4]	EndType	0x00/0x01
Data[5]	End frame	0XFA

Example of serial port sending: FE FE 03 89 00 FA

Return Value： No

byte EndType： data types

Value range :0~1 FLANGE =0; TOOL FLANGE =1;

EndType：：FLANGE to set the end to flange, EndType：：TOOL to set the end to tool end.

1. Get the terminal coordinate system

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X02
Data[3]	Instruction frame	0X8A
Data[4]	End frame	0XFA

Example of serial port sending: FE FE 02 8A FA

Return data structure

Data domain	Description	Data
Data[0]	Return recognition frame	0XFE
Data[1]	Return recognition frame	0XFE
Data[2]	Returns data length frames	0X03
Data[3]	Returns the instruction frame	0X8A
Data[4]	EndType	0x00/0x01
Data[5]	End frame	0XFA

Example of serial port return: FE FE 03 8A 00 FA

---

1. Single motor shutdown

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X03
Data[3]	Instruction frame	0X56
Data[4]	Steering gear serial number	Servo_no
Data[5]	End frame	0XFA

Turn down the first steering gear

Example of serial port sending: FE FE 03 56 01 FA

Return Value： No

1. Single motor powered on

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X02
Data[3]	Instruction frame	0X57
Data[4]	Steering gear serial number	Servo_no
Data[5]	End frame	0XFA

Power the steering gear one

Example of serial port sending: FE FE 03 57 01 FA

Return Value： No

---

1. Setting Atom IO port level state

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X04
Data[3]	Instruction frame	0X61
Data[4]	Number of pins	Pin_no
Data[5]	Level signal	0X00/0X01
Data[6]	End frame	0XFA

set pin P22 to high level

Example of serial port sending: FE FE 04 61 22 01 FA

Return Value： No

---

1. Read Atom IO port level state

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X03
Data[3]	Instruction frame	0X62
Data[4]	Number of pins	pin_no
Data[5]	End frame	0XFA

Read pin P22 level state

Example of serial port sending: FE FE 03 62 22 FA

Return data structure

Data domain	Description	Data
Data[0]	Return recognition frame	0XFE
Data[1]	Return recognition frame	0XFE
Data[2]	Returns data length frames	0X03
Data[3]	Returns the instruction frame	0X62
Data[4]	Level state	0X00/0X01
Data[5]	End frame	0XFA

---

1. Set Atom pin PWM mode

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X04
Data[3]	Instruction frame	0X63
Data[4]	Number of pins	pin_no
Data[5]	Channels	channel
Data[6]	End frame	0XFA

Example of serial port sending: FE FE 04 63 22 01 FA

Return Value： No

1. Set Atom pin PWM output

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X04
Data[3]	Instruction frame	0X64
Data[4]	Channels	channel
Data[5]	Duty cycle	Pin_val
Data[6]	End frame	0XFA

Example of serial port sending: FE FE 04 64 01 20 FA

Return Value： No

1. Reading Claw Angle

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X02
Data[3]	Instruction frame	0X65
Data[4]	End frame	0XFA

Example of serial port sending: FE FE 02 65 FA

Return data structure

Data domain	Description	Data
Data[0]	Return recognition frame	0XFE
Data[1]	Return recognition frame	0XFE
Data[2]	Returns data length frames	0X04
Data[3]	Instruction frame	0X65
Data[4]	High angle	Value_high
Data[5]	Low angle	Value_low
Data[6]	End frame	0XFA

---

1. Set Claw Mode

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X04
Data[3]	Instruction frame	0X66
Data[4]	Claw opening and closing	0X00/0X01
Data[5]	Speed	Sp
Data[6]	End frame	0XFA

Set the claw to open at 20

Example of serial port sending: FE FE 04 66 00 20 FA

Return Value： No

---

1. Set Claw Angle

Data domain	Decription	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X05
Data[3]	Instruction frame	0X67
Data[4]	High angle	Angle_high
Data[5]	Low angle	Angle_low
Data[6]	Speed	Sp
Data[7]	End frame	0XFA

Sample serial port sending:

Return Value： No

---

1. Claw setting zero

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X02
Data[3]	Instruction frame	0X68
Data[4]	End frame	0XFA

Set the current position of the claw to zero

Example of serial port sending: FE FE 02 68 FA

---

1. Check the movement of the claw

Data domain	Description	Data
Data[0]	Recognition frame	0XFE
Data[1]	Recognition frame	0XFE
Data[2]	Data length frame	0X02
Data[3]	Instruction frame	0X69
Data[4]	End frame	0XFA

Set the current position of the claw to zero

Example of serial port sending: FE FE 02 69 FA

Return data structure

Data domain	Description	Data
Data[0]	Return recognition frame	0XFE
Data[1]	Return recognition frame	0XFE
Data[2]	Returns data length frames	0X03
Data[3]	Instruction frame	0X69
Data[4]		0X00/0X01
Data[5]	End frame	0XFA

Appendix:

A coordinate transformation program is added to the ATOM library and the kinematics library, which is implemented as follows:

1. change the terminal coordinate system
2. can set the end coordinate system by setEndType and getEndType functions, EndType：：FLANGE to set the end to flange, EndType：：TOOL to set the end to tool end.
3. can set the coordinate information of the reading tool by setToolReference and getToolReference functions. The flange coordinate system is set as the relative coordinate system, and the tool end information is relative to the flange coordinate system.
4. set the EndType to FLANGE, both the GetCoords and the WriteCoords methods are calculated by the flange position.
5. set the EndType to TOOL, both the GetCoords and the WriteCoords methods are calculated at the end position of the tool.

6. Change the base coordinate system
7. base coordinate system can be set by setReferenceFrame function, RFType：：BASE the robot base as the base coordinate and the world coordinate system as the base coordinate. getReferenceFrame function is to read the current base coordinate system type.
8. read base coordinate system information can be set by setWorldReference and getWorldReference functions. When set, the world coordinate system is used as the relative coordinate system, and the position information of the base of the robot relative to the world coordinate system is input.
9. when the base coordinate system is the base, the GetCoords and WriteCoords methods take the base as the reference coordinate system.
10. When the base coordinate system is the world coordinate system, both the GetCoords and WriteCoords methods use the world coordinate system as the reference coordinate system.

Communications related changes (temporary)

The setting and reading of the terminal coordinate system, the setting and reading of the world coordinate system, the setting and reading of the current reference coordinate system, the setting and reading of the terminal type, the setting and reading of the moving mode, and the sending and receiving of the manipulator information are added.

These communications are temporarily set to 0x80 to 0x8A

The new roboticMessages space in the ParameterList.h file is used to add the manipulator communication information.

MOVEL function simple design idea is as follows:

The Euclidean distance between the initial point and the target point is obtained, and an interpolation point is inserted every 10 mm based on the Euclidean distance. If the interpolation point has no inverse solution, search position invariant three directions attitude positive and negative PI/30 adjacent space whether there are inverse solutions, mainly to avoid singular values and some special positions that can not be solved.

The point transmission interval between MOVEL and JOG is changed to dynamic time. The moving time is calculated according to the maximum joint moving distance between two points, and then the moving time minus the specific time is taken as the time interval.

7 Accessories

myCobot accessories includes

End Effectors

- Parallel Gripper
- Adaptive Gripper
- Opening Angle Gripper
- Suction Pump

Bases

- G Base
- Flat Base

Accessories

- JoyStick
- Battery Box

7.1 End Effectors

End effectors of myCobot now in mass production

- **Parallel Gripper**: Use of Clamping Objects
- **Suction Pump**: Use of adsorption objects

7.1.1 Gripper

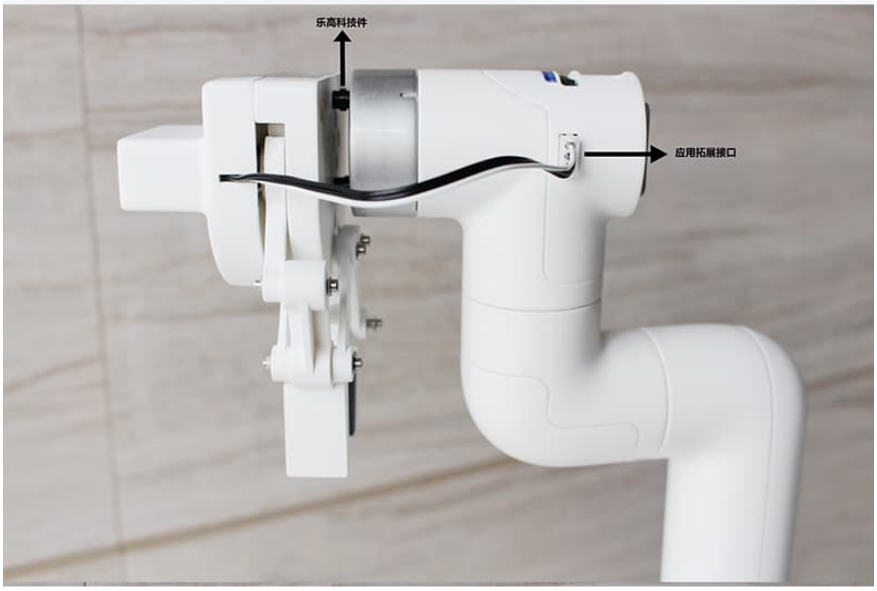
Gripper

- 1.fix the gripper to the top of the robotic arm with a lego tech and connect to the M5STACK Basic end-effector extension interface (described in the unboxing video)
- 2.grippers can be used in all development environments, such as ROS、Arduino、UIFlow and RoboFlow.

Applicable objects

- small box
- small ball
- long strip

Tips you can paste rubber at the fingertips of the gripper for better friction.



API control you can control gripper by downloading the latest myCobot API.

1. Control the opening and closing degree of the claw：please read the claw position first, then set the range. The value range is 0-4096; the actual range is near 2000.

2. Set the initial point of the claw：the initial point corresponds to a closing degree of 2048.

setGripperIni() // set gripper encode initial center point(to be 2048) **3. Read gripper position**

getGripperValue() // return gripper encdoer value (from 0 to 4096) **4 设置夹爪状态：**现阶段只支持张开闭合两个状态，如果需要设置精准位置，可以设置夹爪闭合程度；

setGripperState // only used for adaptive gripper, 0 or 1 for open and close

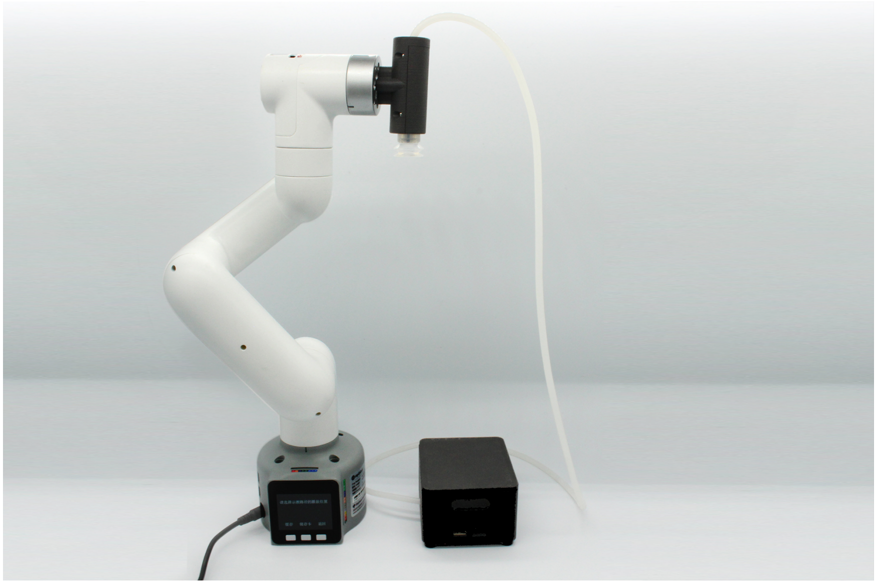
7.1.2 Suction pump

Applicable object

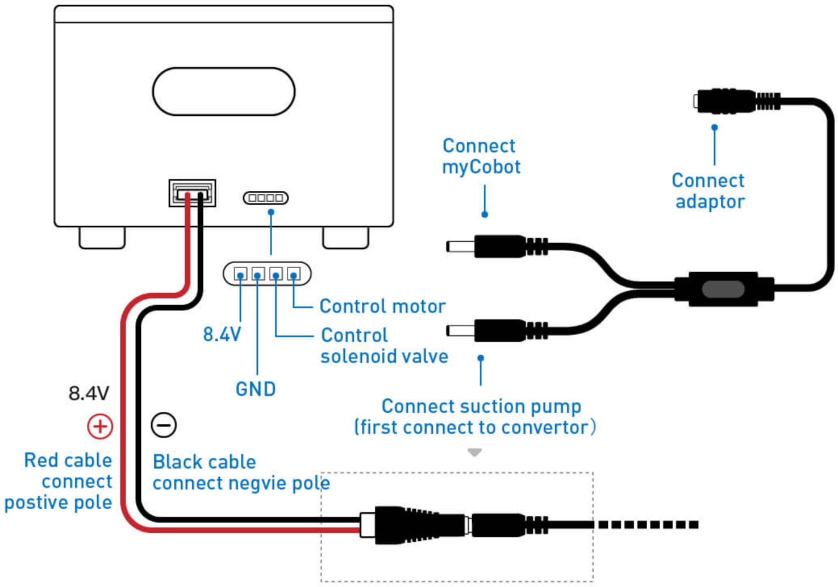
- Paper/plastics
- Smooth planar object
- Cards, etc

Product Description

Installation schematic



wiring diagram



Notes

- Please ensure that the product is connected successfully according to the instructions
- Make sure the product is powered by attached adapter
- Please ensure the access direction of positive and negative electrodes

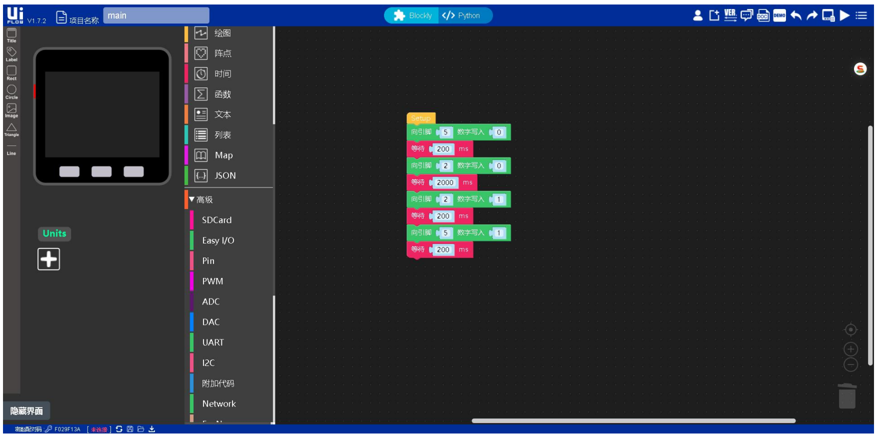
Use of suction pump in Arduino

1. connect the suction pin to the Basic pin
   - solenoid valve control pin link Basic -G2 pin
   - Pump control pin link Basic -G5 pin
2. create a new Arduino program to copy the following:

```
set pin 2 to high level and close solenoid valve//
// please confirm solenoid valve connection G2 pin, pump link G5 pin
// connection completed, high level off, low level open
//

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);//open serial port, baud rate 9600
  pinMode(2,OUTPUT); //set pin G2 to output state
  pinMode(5,OUTPUT); //set pin G5 to output state
  delay(100);
  digitalWrite(2,1);//set pin 2 to high level and close solenoid valve
  digitalWrite(5,1);//set pin 5 to high level and shut down pump
}

void loop() {
  // 使用时按照需求控制电磁阀与泵机
  digitalWrite(5,0);//将引脚5设为低电平，打开泵机
  delay(200);//延时200ms
  digitalWrite(2,0);//将引脚2设为低电平，打开电磁阀
  delay(2000);//延时2000ms,松开吸住的物体
  digitalWrite(2,1);//将引脚2设为高电平，关闭电磁阀
  delay(200);//延时200ms
  digitalWrite(5,1);//将引脚5设为高电平，关闭泵机
  delay(200);//延时200ms
}
```

Use of suction pumps in UIflow

1. connect the suction pin to the Baisc pin
   - Solenoid valve control pin link Basic -G2 pin
   - Pump control pin link Basic -G5 pin
2. Create a new UIFlow program to add the following content:



```
Setup
向引脚 5 数字写入 0
等待 200 ms
向引脚 2 数字写入 0
等待 2000 ms
向引脚 2 数字写入 1
等待 200 ms
向引脚 5 数字写入 1
等待 200 ms
```

7.1.3 Pen holder

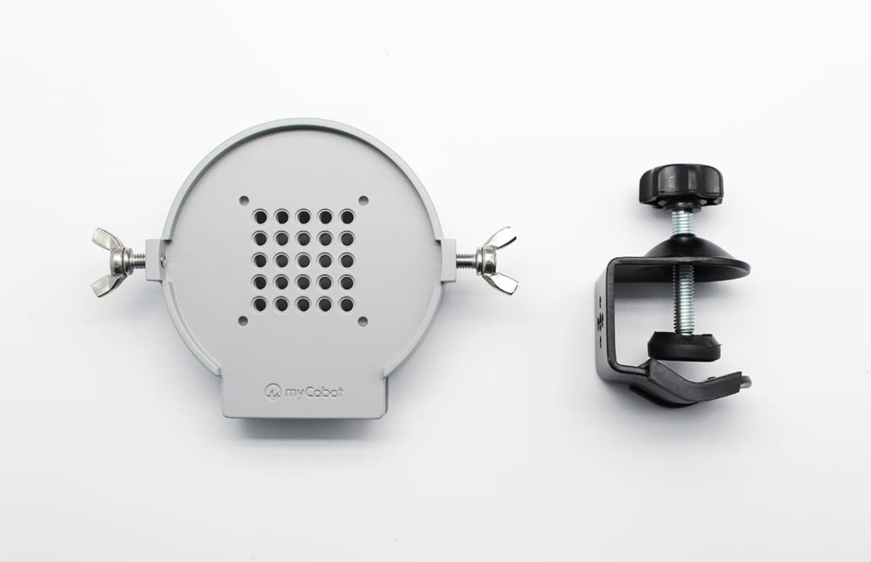
Is being developed and written...ss

7.2 Base

myCobot current support base:

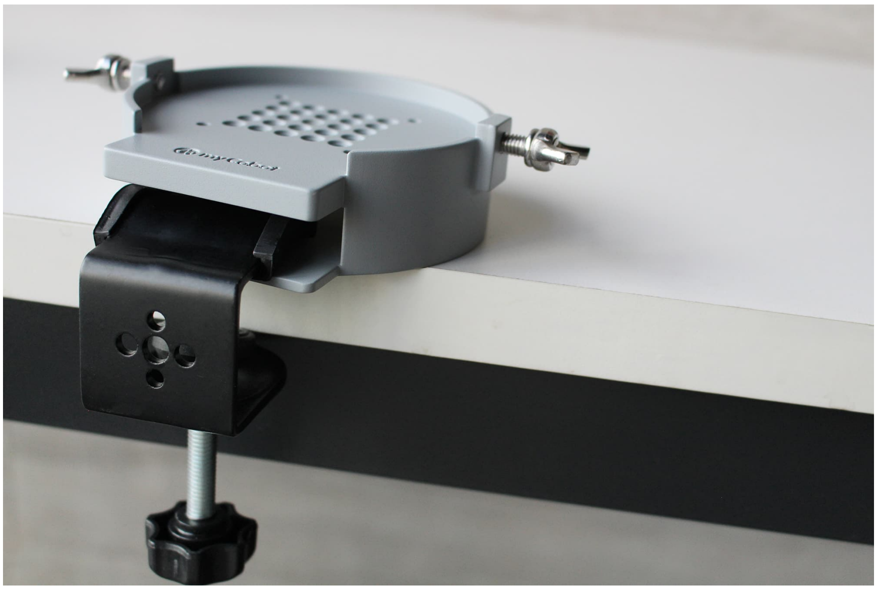
- **G Base**: Conveniently secured to table
- **Sucking Base**: Easy to fix directly on smooth table

7.2.1 G Base



G Base - fixed to the edge of the table

- 1.Tighten the horn screws at both ends and insert a rubber sleeve into the he
- 2.Fix the base on the edge of the table with a G clip
- 3.The base and the bottom of the arm with the attached Lego tech
- 4.Make sure it’s stable before use



7.2.2 Sucking Base

Suitable for smooth surfaces

- 1.Install the sucker at the four corners of the base and tighten them
- 2.Use the attached Lego piece to connect the sucking base and the bottom of the manipulator
- 3.Fix the four suckers to a smooth and flat surface before use

Tips
A small amount of **non-conductive liquid**s can be added under the sucker to fill the gap between the sucker and the desktop to obtain the best adsorption effect.

7.3 Accessories

Is being developed and written...

8 Machine Vision Development

What is image recognition?

1. **principle of image recognition**

    - Image recognition refers to the technology of using computers to process, analyze and understand images in order to identify targets and objects of various modes. It is a practical application of deep learning algorithms.

2. **Application scenarios of image recognition**

    - At the present stage, image recognition technology is generally divided into face recognition and commodity recognition. Face recognition is mainly used in security check, identity verification and mobile payment. Product identification is mainly used in the process of commodity circulation, especially in unmanned retail areas such as unmanned shelves and intelligent retail cabinets.

3. **Artificial intelligence application of image recognition**

    - Image recognition is an important field of artificial intelligence. Different image recognition models have been proposed in order to make computer programs simulating human image recognition activities. An example is the template matching model. This model holds that to recognize an image, one must have a memory pattern of the image in the past experience, also known as the template. If the current stimulus matches the template in the brain, the image is recognized. For example, if there is a letter “A”, the letter “A” is recognized if the size, orientation, and shape of the letter “A” are exactly the same as the template of "A" in the mind. The model is simple and straightforward, and easy to be applied in practice. However, this model emphasizes that the image must be completely consistent with the template in the brain before it can be recognized. In fact, people can not only recognize the image that is completely consistent with the template in the brain, but also recognize the image that is not completely consistent with the template. For example, people can identify not only a specific letter “A”, but also printed, handwritten, misaligned, and different-sized letters “A”. At the same time, people can recognize a large number of images, if the recognition of each image has a corresponding template in the brain, it is impossible.
    - In order to solve the problem of template matching model, Gestalt psychologists put forward a prototype matching model. According to this model, what is stored in long-term memory is not the innumerable templates to be recognized, but some "similarity" of the images. The "similarity" abstracted from the image can be used as a prototype to test the image to be recognized .If a similar prototype can be found, the image is identified. This model is better than template-matching models,

    both in the process of neural and memory exploration, and it can also explain the recognition of images that are irregular, but in some ways similar to the prototype. However, this model does not explain how people can identify and process similar stimuli, and it is difficult to implement in a computer program. Therefore, a more complex model is proposed, that is, the "pan-demonic" recognition model.
    - In industrial applications, pictures are usually taken by industrial cameras, and then processed by software according to the grayscale difference of the picture to identify useful information. The representative of the image recognition software is Connex.
4. **development of image recognition**
    - The development of image recognition has experienced three stages: character recognition, digital image processing and recognition, and object recognition. The research of character recognition began in 1950. It is generally used to recognize letters, numbers and symbols. It is widely used from printed character recognition to handwritten character recognition.
    - The research of digital image processing and recognition began in 1965. Compared with analog images, digital images have great advantages such as storage, convenient transmission and compression, not easy distortion in transmission and convenient processing, which provide a powerful impetus for the development of image recognition technology. Object recognition mainly refers to the perception and understanding of the object and environment of the three-dimensional world, which belongs to advanced computer vision. It is based on the digital image processing and recognition of the combination of artificial intelligence, systems science and other disciplines research, its research results have been widely used in a variety of industrial and exploring robots. One of the shortcomings of modern image recognition technology is the poor adaptive performance. Once the target image is polluted by strong noise or the target image has large imperfection, there will not be an ideal identification result.
    - The mathematical nature of image recognition is a mapping problem from pattern space to category space. At present, there are mainly three recognition methods: statistical pattern recognition, structural pattern recognition and fuzzy pattern recognition in the development of image recognition.Image segmentation is a key technology in image processing. Since the 1970s, its research has a history of several decades and has been highly valued by people. Up to now, thousands of segmentation algorithms have been proposed with the help of various theories, and the research in this field is still being actively carried out.
    - There are many kinds of existing image segmentation methods, including threshold segmentation, edge detection, region extraction, and the segmentation method combined with specific theoretical tools. From the type of image to include: gray image segmentation, color image segmentation and texture image segmentation. As early as 1965, someone proposed the edge detection operator, which resulted in many classical edge detection algorithms. However, in the past 20 years, with

the rapid development of image segmentation based on histogram and wavelet transform, computing technology and VLSI technology, the research on image processing has made great progress. Image segmentation methods combine some specific theories, methods and tools, such as image segmentation based on mathematical morphology, segmentation based on wavelet transform, and segmentation based on genetic algorithm.

Use StivckV + Maixpy - IDE to image recognition development

Development Platform

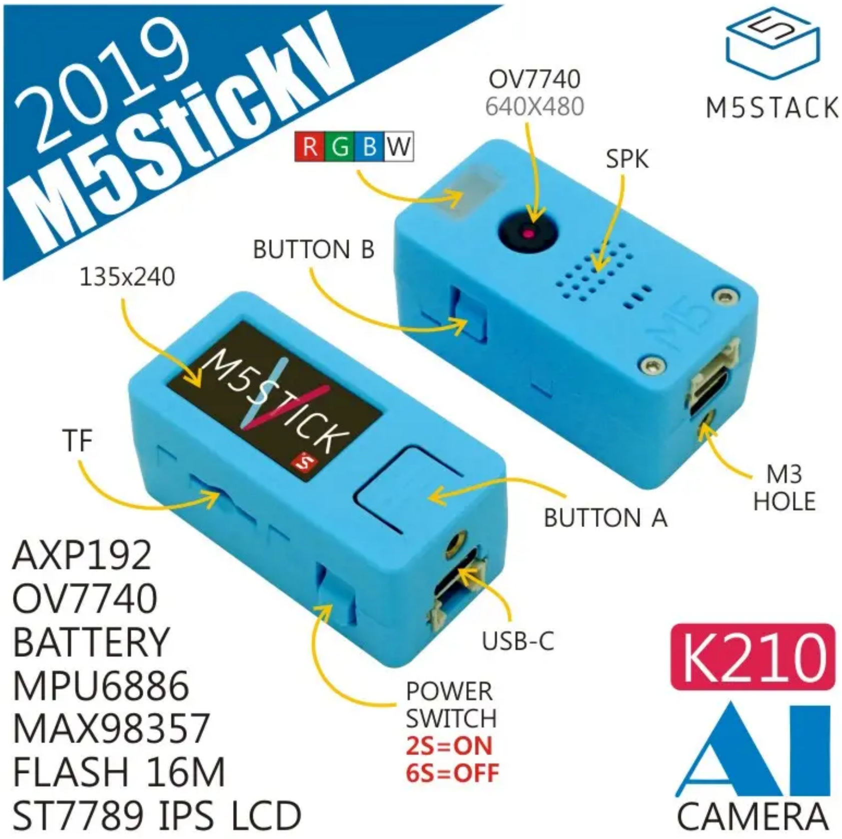
- Maixpy - IDE

Development Environment

- Windows
- Linux

Developer Components

- M5Stack - StickV



Description

M5Stick-V RISC-V AI Camera

M5Stack recently launched the new AIoT(AI+IoT) Camera powered by Kendryte K210 -an edge computing system-on-chip(SoC) with dual-core 64bit RISC-V CPU and advanced neural network processor..

M5StickV AI Camera possesses machine vision capabilities, equips OmniVision OV7740 image sensor, adopts OmniPixel®3-HS technology, provides optimum low light sensitivity, supports various vision identification capabilities. (e.g. Real-time acquisition of the size, type and coordinates of the detected target) In addition to an OV7740 sensor, M5StickV features more hardware resources such as a speaker with built-in I2S Class-D DAC, IPS screen, 6-axis IMU, 200mAh Li-po battery, and more.

It is able to perform convolutional neural network calculations at low power consumption, so M5StickV will be a good zero-threshold machine vision embedded solution. It is in support with MicroPython, which makes your code to be more concise when you use M5stick-V for programming.

Product Features

- Dual-Core 64-bit RISC-V RV64IMAFDC (RV64GC) CPU / 400Mhz(Normal)
- Dual Independent Double Precision FPU
- Neural Network Processor(KPU) / 0.8Tops

- Field-Programmable IO Array (FPIOA)
- Dual hardware 512-point 16bit Complex FFT
- SPI, I2C, UART, I2S, RTC, PWM, Timer Support
- AES, SHA256 Accelerator
- Direct Memory Access Controller (DMAC)
- Micropython Support
- Firmware encryption support
- Case Material: PC + ABS

Applications

- Face recognition/detection
- Object detection/classification
- Obtaining size and coordinates of the target in real-time
- Obtaining the type of detected target in real-time
- Shape recognition
- Video/Display
- Game simulator

USB Drive problems

- M5StickV may not work without driver in some systems. Users can manually installFTDI to fix this problem.

Resources	Parameter
Kendryte K210	Dual core 64-bit RISC-V RV64IMAFDC（RV64GC）CPU / 400Mhz（Normal）
SRAM	8MiB
Flash	16M
Power input	5V @ 500mA
KPU parameter size of neural network	5.5MiB-5.9MiB
Port	TypeC x 1，GROVE（I2C + I / 0 + UART）x 1
RGB LED	RGBW x 1
Button	Custom buttonx 2
IPS screen	1.14 TFT，135 * 240，ST7789
Camera	OV7740(30w pixels)
FOV	55deg
PMU	AXP192
Battery	200mAh
External storage	TF-card（microSD）
MEMS	MPU6886
Net weight	23g
Gross weight	82g
Product Size	48 *24* 22mm
Package Size	144 *44* 43mm
Case Material	Plastic（PC）

TF-card（microSD）test

- M5StickV not currently recognize all types of TF-card(microSD). We have tested some common TF-card(microSD). The test results are as follows.

Brand	Storage	Type	Class	Format	Test Results
Kingston	8G	HC	Class4	FAT32	ok
Kingston	16G	HC	Class10	FAT32	ok
Kingston	32G	HC	Class10	FAT32	no
Kingston	64G	XC	Class10	FAT文件	ok
SanDisk	16G	HC	Class10	FAT32	ok
SanDisk	32G	HC	Class10	FAT32	ok
SanDisk	64G	XC	Class10	/	no
SanDisk	128G	XC	Class10	/	no
XIAKE	16G	HC	Class10	FAT32	ok（purple）
XIAKE	32G	HC	Class10	FAT32	ok
XIAKE	64G	XC	Class10	/	no
XIAKE	32G	HC	Class10	/	no

Functional Description

Kendryte K210

The Kendryte K210 is a system-on-chip (SoC) that integrates machine vision. Using TSMC’s ultra-low-power 28-nm advanced process with dualcore 64-bit processors for better power efficiency, stability and reliability. The SoC strives for ” zero threshold” development and to be deployable in the user’s products in the shortest possible time, giving the product artificial intelligence

- Machine Vision
- Better low power vision processing speed and accuracy
- KPU high performance Convolutional Neural Network (CNN) hardware accelerator
- Advanced TSMC 28nm process, temperature range -40°C to 125°C
- Firmware encryption support
- Unique programmable IO array maximises design flexibility
- Low voltage, reduced power consumption compared to other systems with the same processing power
- 3.3V/1.8V dual voltage IO support eliminates need for level shifters

CPU

The chip contains a high-performance, low power RISC-V ISA-based dual core 64-bit CPU with the following features:

- Core Count： Dual-core processor
- Bit Width: 64-bit CPU 400MHz
- Frequency: 400MHz
- ISA extensions: IMAFDC
- FPU: Double Precision
- Platform Interrupts: PLIC
- Local Interrupts: CLINT
- I-Cache: 32KiB x 2
- D-Cache: 32KiB x 2
- On-Chip SRAM: 8MiB

OV7740

- support for output formats: RAW RGB and YUV
- support for image sizes: VGA, QVGA, CIF and any size smaller
- support for black sun cancellation
- support for internal and external frame synchronization
- standard SCCB serial interface
- digital video port (DVP) parallel output interface
- embedded one-time programmable (OTP) memory
- on-chip phase lock loop (PLL)
- embedded 1.5 V regulator for core
- Sophisticated Edge Rate Control Enables Filterless Class D Outputs
- 77dB PSRR at 1kHz
- Low RF Susceptibility Rejects TDMA Noise from GSM Radios
- Extensive Click-and-Pop Reduction Circuitry
- array size: 656 x 488
- power supply: – core: 1.5VDC ± 5% – analog: 3.3V ± 5% – I/O: 1.7 ~ 3.47V
- temperature range: – operating: -30° C to 70°C – stable image: 0° C to 50° C
- output format: – 8-/10-bit raw RGB data – 8-bit YUV
- lens size: 1/5"
- input clock frequency: 6 ~ 27 MHz
- max image transfer rate: VGA (640x480): 60 fps – QVGA (320 x 240): 120 fp
- sensitivity: 6800 mV/(Lux-sec)
- maximum exposure interval: 502 x tROW
- pixel size: 4.2 µm x 4.2 µm
- image area: 2755.2 µm x 2049.6 µm
- package/die dimensions: – CSP3: 4185 µm x 4345 µm – COB: 4200 µm x 4360 µm

MAX98357

- Single-Supply Operation (2.5V to 5.5V).

- 3.2W Output Power into 4Ω at 5V
- 2.4mA Quiescent Current
- 92% Efficiency (RL = 8Ω, POUT = 1W)
- 22.8μVRMS Output Noise (AV = 15dB)
- Low 0.013% THD+N at 1kHz
- No MCLK Required
- Sample Rates of 8kHz to 96kHz
- Supports Left, Right, or (Left/2 + Right/2) Output
- Sophisticated Edge Rate Control Enables Filterless Class D Outputs
- 77dB PSRR at 1kHz
- Low RF Susceptibility Rejects TDMA Noise from GSM Radios
- Extensive Click-and-Pop Reduction Circuitry

AXP192

- Operation Voltage: 2.9V - 6.3V(AMR：-0.3V~15V)
- Configurable Intelligent Power Select system
- Current and voltage limit of adaptive USB or AC adapter input
- The resistance of internal ideal diode lower than 100mΩ

MPU6886

Gyroscope features

- Digital-output X-, Y-, and Z-axis angular rate sensors (gyroscopes) with a user-programmable full-scale range of ±250 dps, ±500 dps, ±1000 dps, and ±2000 dps and integrated 16-bit ADCs
- Digitally-programmable low-pass filter
- Low-power gyroscope operation
- Factory calibrated sensitivity scale factor
- lens size: 1/5"
- Self-test

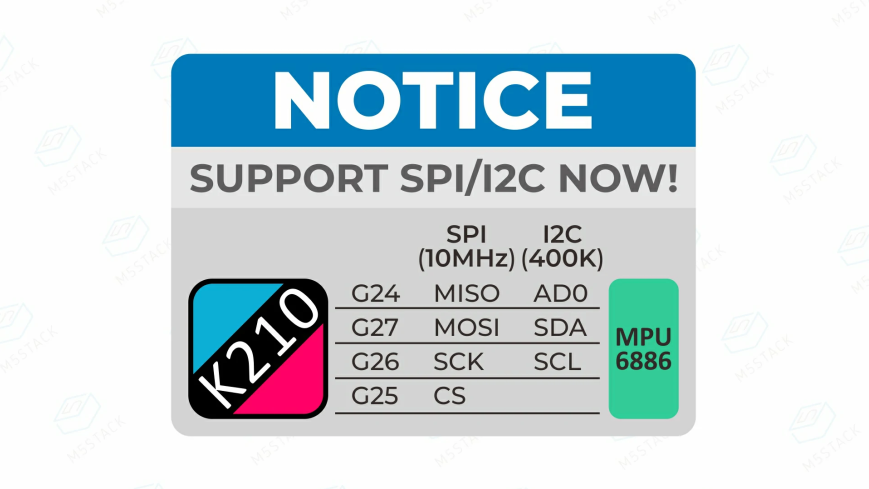
Accelerometer features

- Digital-output X-, Y-, and Z-axis accelerometer with a programmable full scale range of ±2g, ±4g, ±8g and ±16g and integrated 16-bit ADCs
- User-programmable interrupts
- Wake-on-motion interrupt for low power operation of applications processor
- Self-test

SPI / I2C dual communication mode

> Note: There are two versions of M5StickV currently released by M5Stack. When programming, users need to configure differently according to their corresponding pin mapping. The specific differences are as follows.

- In the M2StickV circuit design of the I2C single-mode (blue PCB) version, MPU6886 only supports the user to configure its communication mode to I2C, and its pin mapping is SCL-28, SDA-29.
- In the SPI/I2C dual mode (black PCB) version of the M5StickV circuit design, MPU6886 supports the user to configure its communication mode to SPI or I2C, and its pin mapping is SCL-26, SDA-27., when using, you can switch CS Pin level to switch modes (high level 1 is I2C mode, low level 0 is SPI mode)
- The specific pin mapping is shown below:



NOTICE

SUPPORT SPI/I2C NOW!

K210	SPI (10MHz)	I2C (400K)	MPU 6886
G24	MISO	AD0	
G27	MOSI	SDA	
G26	SCK	SCL	
G25	CS		

Links

datasheet

- MPU6688
- SH200Q

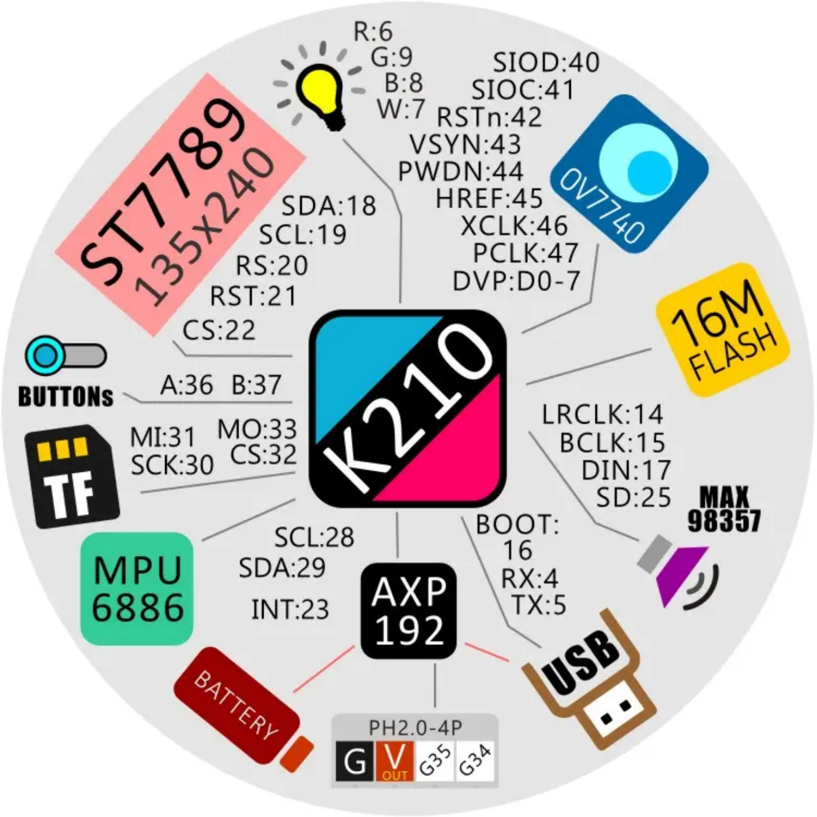
Web page

- sipeed

GITHUB

- API

schematic



K210-CAM

Procedure

- Maixpy referenceexample

8.1 Set up MaixPy environment

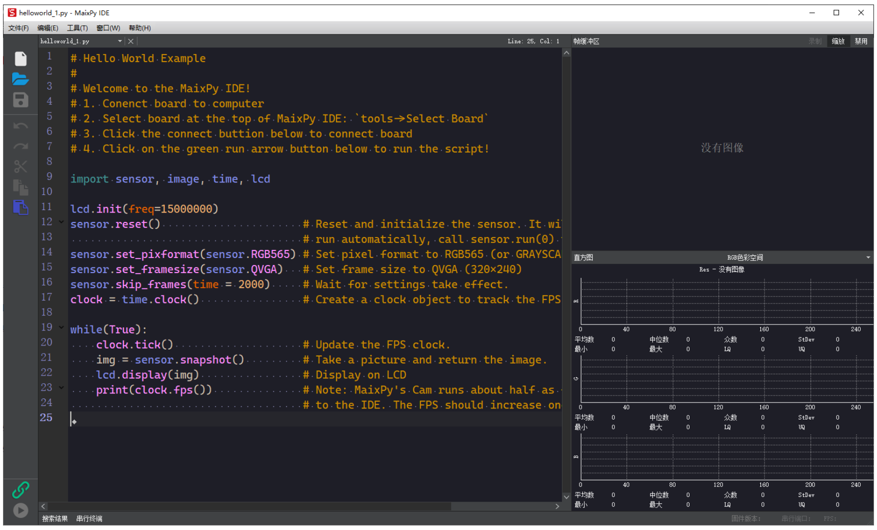
1.1 What is MaixPy?

MaixPy is a project transplanted Micropython into K210 (a 64 - bit dual-core with hardware FPU, convolution accelerator, FFT, Sha256 of RISC-V CPU), supporting MCU regular operations, it also incorporates hardware acceleration AI machine vision and a microphone array, 1TOPS computing power core module is less than ￥50, which quickly develops very low cost and volume of practical AIOT field of smart applications.

The first thing you need to know is Maixpy uses MicroPython scripting syntax, so it doesn't require compilation like C language, it can be used without an IDE: using the serial terminal tool which has been installed previously.

Using IDE, you can edit scripts in real time on the computer and upload them to the development board，which is convenient to execute scripts directly on the development board, view camera images in real time on the computer, save files to the development board and so on.

However, using an IDE will compress and transport some resources, so the performance will be reduced, and if MaixPy goes down, it’s hard to find problems like serial terminal.



```
# Hello World Example
#
# Welcome to the MaixPy IDE!
# 1. Conenct board to computer
# 2. Select board at the top of MaixPy IDE: `tools->Select Board`
# 3. Click the connect buttion below to connect board
# 4. Click on the green run arrow button below to run the script!

import sensor, image, time, lcd

lcd.init(freq=15000000)
sensor.reset()                      # Reset and initialize the sensor. It wi
                                    # run automatically, call sensor.run(0)
sensor.set_pixformat(sensor.RGB565) # Set pixel format to RGB565 (or GRAYSCA
sensor.set_framesize(sensor.QVGA)   # Set frame size to QVGA (320x240)
sensor.skip_frames(time = 2000)     # Wait for settings take effect.
clock = time.clock()                # Create a clock object to track the FPS

while(True):
    clock.tick()                    # Update the FPS clock.
    img = sensor.snapshot()         # Take a picture and return the image.
    lcd.display(img)                # Display on LCD
    print(clock.fps())              # Note: MaixPy's Cam runs about half as
                                    # to the IDE. The FPS should increase on
```

For more information, please click official websiteto view the official instructions.

1.2 What can MaixPy be used for?

- Face Recognition
- Object Recognition
- Color Identification
- Emotion Recognition
- License Plate Recognition
- Sorting System

For more information, please clickofficial websiteto view the official instructions.

2.How to set up the usage environment?

2.1 Preparation of the development environment

2.1.1 Install Driver

- Before using MaixPy, we need to install the serial driver before we can proceed to the next step of development and use. The board is connected to the computer via a USB-to-serial device (K210 does support USB hardware), please install the driver according to the USB to serial port chip model of the board.
- Click to download and installthe serial driver
- When operating a serial port on Linux or Mac, if you don't want to use “sudo” every time, perform “sudo usermod -a -G dialout $(whoami)” to add yourself to the “dialout” user group, which may require a logout or reboot.

2.1.2 Burn Firmware

- The firmware must be V0.3.1 or above to use the MaixPy IDE, otherwise the MaixPy IDE will not be connected. Try to check the firmware version and IDE version before use, and update it to the latest version to ensure normal use.
- Click to download the firmware burner *Kflsh* , a compressed package after downloading.

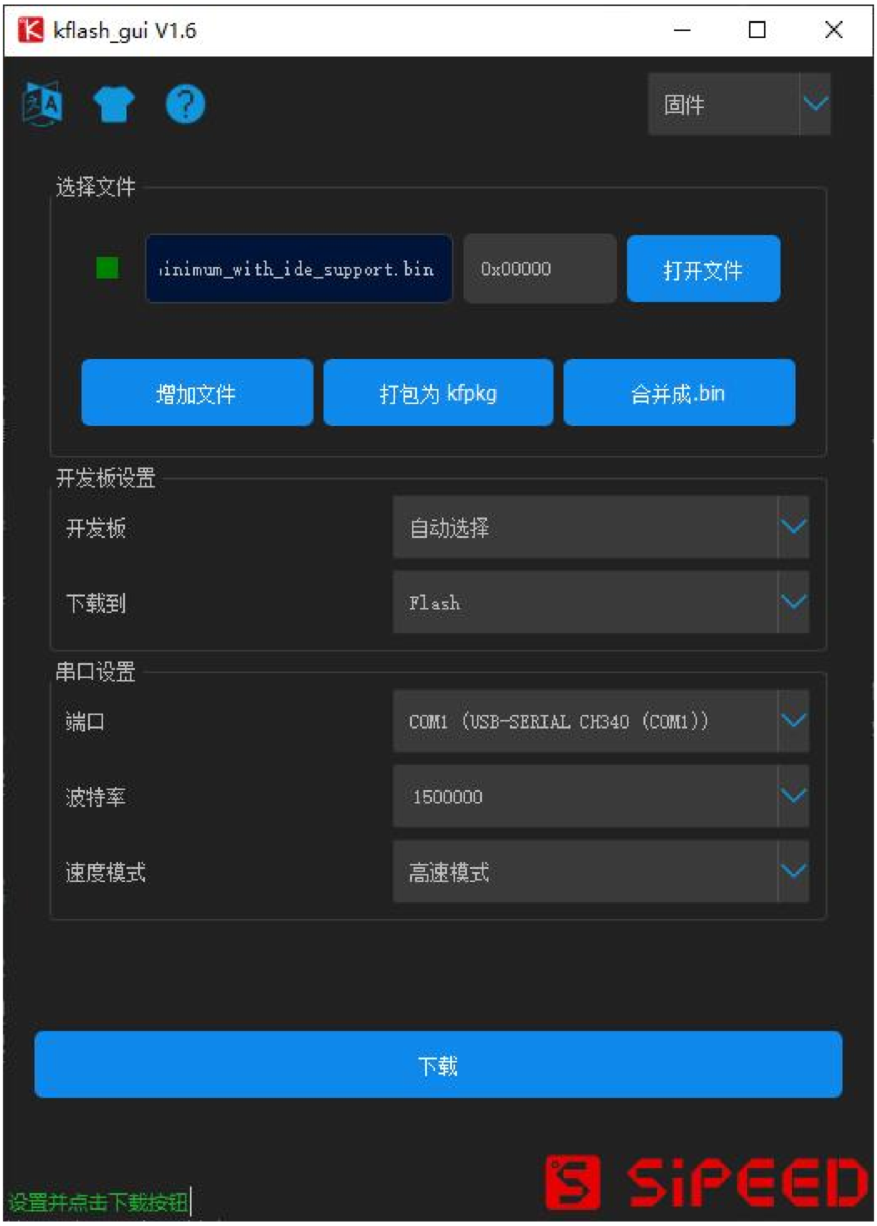
Kflash_gui is cross-platform and can be used on multiple systems (including Windows, Linux, MacOS, and even Raspberry Pi).

Windows version of Kendryte may be unable to download. Please use the software kflash_gui to download.

- Click to download and installfirmware

- Burn firmware

After downloading, use the firmware burner to burn the firmware to M5StickV:



kflash_gui V1.6

固件

选择文件

🟩 ıinimum_with_ide_support.bin 0x00000

开发板设置

开发板 自动选择

下载到 Flash

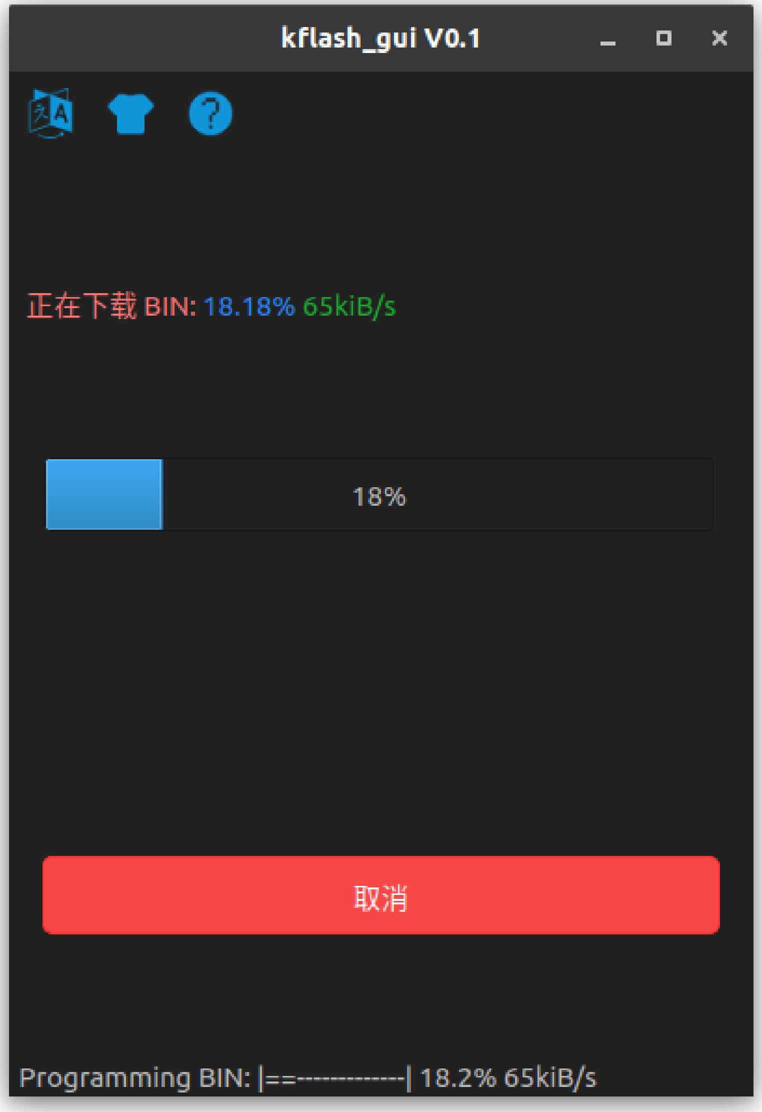
串口设置

端口 COM1 (USB-SERIAL CH340 (COM1))

波特率 1500000

速度模式 高速模式

设置并点击下载按钮



```
Programming BIN: |==-------------| 18.2% 65kiB/s
```

2.2 Download and install software

2.2.1 Click to download MaixPy-IDE

Note: Please refer to the readme.txt file in the latest version folder for the list of files. If the download speed is slow, please use the cdn link to download.

Install software

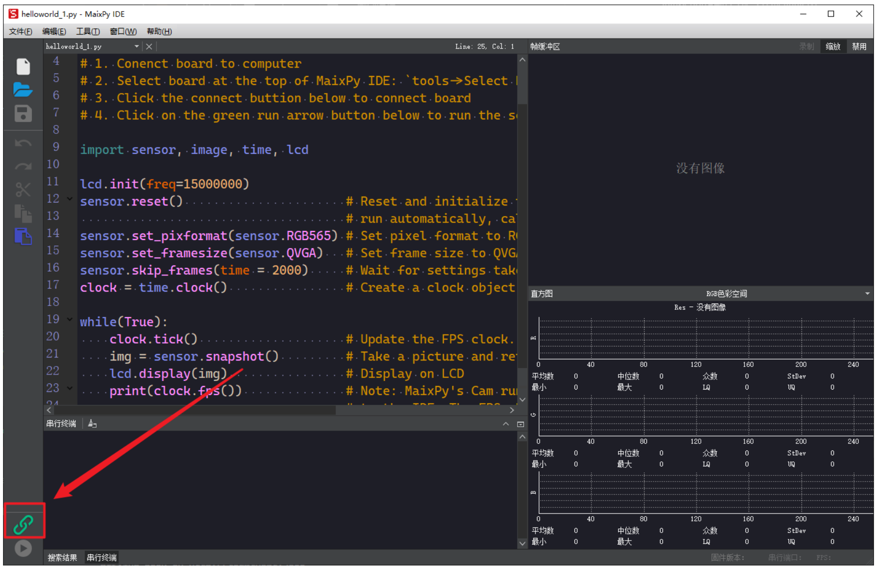
Download the file, directly double-click the exe file to run the installation program in Windows

Test connection

Open MaixPy IDE and select the model of the development board from the top toolbar. Select M5Stickv to connect.

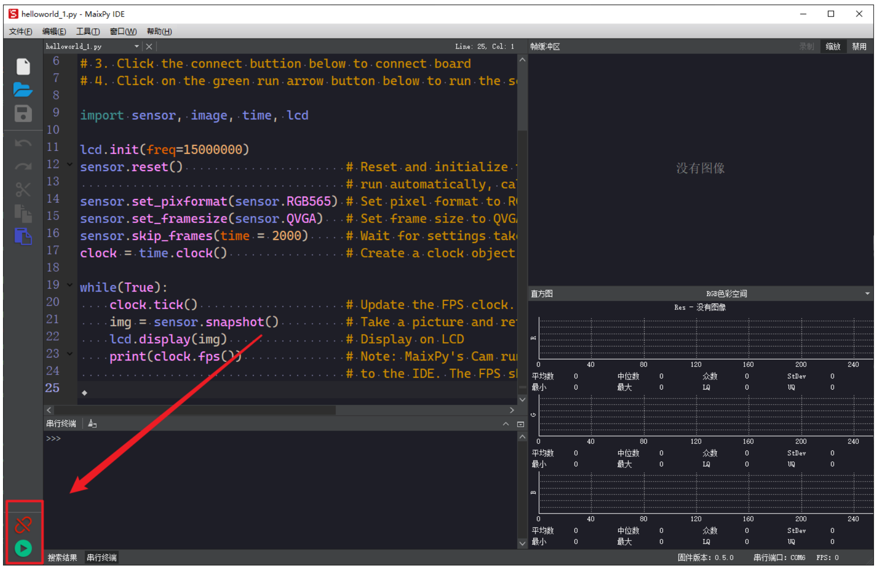
Tool-> Select Board (Tools -> Select Development Board)

Click “connect” to connect MaixPy IDE



```
# 1. Conenct board to computer
# 2. Select board at the top of MaixPy IDE: `tools->Select 
# 3. Click the connect buttion below to connect board
# 4. Click on the green run arrow button below to run the s

import sensor, image, time, lcd

lcd.init(freq=15000000)
sensor.reset()                      # Reset and initialize 
                                    # run automatically, ca
sensor.set_pixformat(sensor.RGB565) # Set pixel format to R
sensor.set_framesize(sensor.QVGA)   # Set frame size to QVG
sensor.skip_frames(time = 2000)     # Wait for settings tak
clock = time.clock()                # Create a clock object

while(True):
    clock.tick()                    # Update the FPS clock.
    img = sensor.snapshot()         # Take a picture and re
    lcd.display(img)                # Display on LCD
    print(clock.fps())              # Note: MaixPy's Cam ru
```

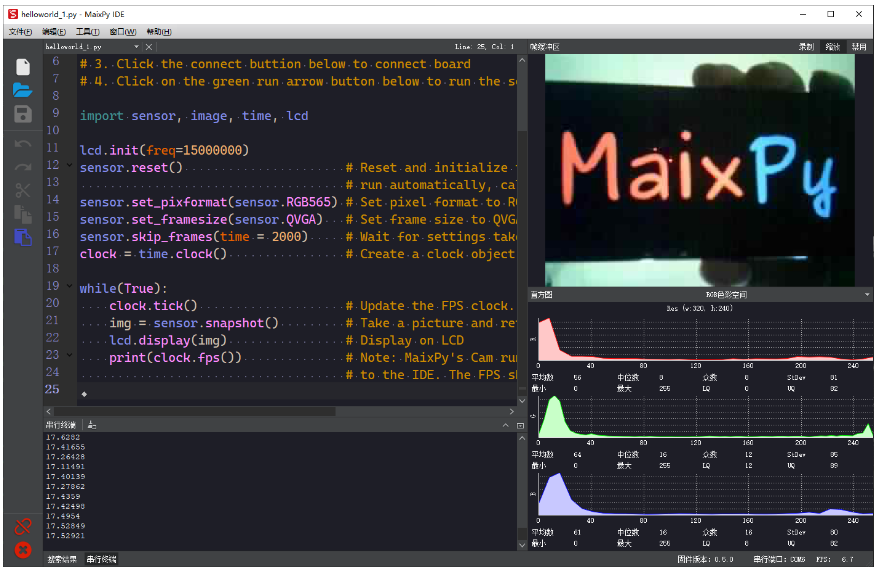
When the connection is successful, it will change from green to red.



```
# 3. Click the connect buttion below to connect board
# 4. Click on the green run arrow button below to run the s

import sensor, image, time, lcd

lcd.init(freq=15000000)
sensor.reset()                      # Reset and initialize 
                                    # run automatically, ca
sensor.set_pixformat(sensor.RGB565) # Set pixel format to R
sensor.set_framesize(sensor.QVGA)   # Set frame size to QVG
sensor.skip_frames(time = 2000)     # Wait for settings tak
clock = time.clock()                # Create a clock object

while(True):
    clock.tick()                    # Update the FPS clock.
    img = sensor.snapshot()         # Take a picture and re
    lcd.display(img)                # Display on LCD
    print(clock.fps())              # Note: MaixPy's Cam ru
                                    # to the IDE. The FPS s
```

```
>>>
```

Below the connect button is the run button, which executes the py file in the current edit area.



```
# 3. Click the connect buttion below to connect board
# 4. Click on the green run arrow button below to run the s

import sensor, image, time, lcd

lcd.init(freq=15000000)
sensor.reset()                      # Reset and initialize 
                                    # run automatically, ca
sensor.set_pixformat(sensor.RGB565) # Set pixel format to R
sensor.set_framesize(sensor.QVGA)   # Set frame size to QVG
sensor.skip_frames(time = 2000)     # Wait for settings tak
clock = time.clock()                # Create a clock object

while(True):
    clock.tick()                    # Update the FPS clock.
    img = sensor.snapshot()         # Take a picture and re
    lcd.display(img)                # Display on LCD
    print(clock.fps())              # Note: MaixPy's Cam ru
                                    # to the IDE. The FPS sl
```

```
17.6282
17.41655
17.26428
17.11491
17.40139
17.27862
17.4359
17.42498
17.4954
17.52849
17.52921
```

```
平均数  56   中位数  8    众数  8    StDev  81
最小    0    最大    255  LQ    0    UQ     82
平均数  64   中位数  16   众数  12   StDev  85
最小    0    最大    255  LQ    12   UQ     89
平均数  61   中位数  16   众数  16   StDev  80
最小    0    最大    255  LQ    8    UQ     82
```

固件版本: 0.5.0 串行端口: COM6 FPS: 6.7

Click the Run button again (red) to stop running the current code.

3. Troubleshooting Guide of USB Serial Port

If you do not see a serial port, check for hardware problems in the following order.

- Insert the computer, whether there is a ding-dong sound, such as the sound of the USB drive loading when inserting the U disk. If there isn’t, it means that there is a problem with the serial chip on the hardware.
- Replace the cable and try again. Replace the USB port of the computer and try again. It still won't load out, change the computer to confirm.

If you cannot burn firmware, check for hardware problems in the following order.

- Use the serial port tool to see if the MAIXPY firmware exists in the hardware.
- Set 115200 baud rate to connect the serial port, press the reset key (RST) to receive the data of the chip. Whatever it happens means the serial port chip is working normally, but if there is nothing, it means the hardware is abnormal.
- Based on the above, burn the firmware again. Before burning, press the BOOT key of the hardware and press reset, and then release the BOOT key, which means the burning is processing normally. If not, it means that Flash is damaged, you can try to burn to SRAM. If the burning fails, it means that the serial port chip is abnormal.
- If you get to this step and still can't fix the problem, then the hardware does have a defect.

3.1 Burning mechanism introduction of K210

We often call this a key download circuit, can easily complete the control of BOOT and RST pin and enter the burning mode through the control of the serial port RST and DTR. As described above, the hardware circuit is expected to replace human to automatically perform the operation of pressing RST and BOOT, which is strongly dependent on hardware implementation. Only based on this can data transmission of TX and RX be carried out, so we need to use functional pins of UART serial port.

Kflash can be divided into a variety of versions of a variety of burning triggers. We can simply divide it into several categories, 115200 baud rate at low speed and 1500000 baud rate at high speed, taking the matching burning mode of these two types of baud rate as the difference point.

If it is found that the download process fails, the baud rate can be reduced appropriately, because the serial port chip is not stable. The selection of the type in the tool will only affect the trigger of the first burning mode, and after that the burning firmware will be burned at the configured baud rate, usually not exceed the communication burning speed with flash, commonly seen in 50~60 KB/S.

If you find that no matter how to replace the burning mode can not enter, either the burning version does not match, or the serial chip DTR RST pin problem (physical).

8.2 Color Recognition

Preparation for development

- Complete the device link
- Complete firmware burning
- Complete software environment

Example code

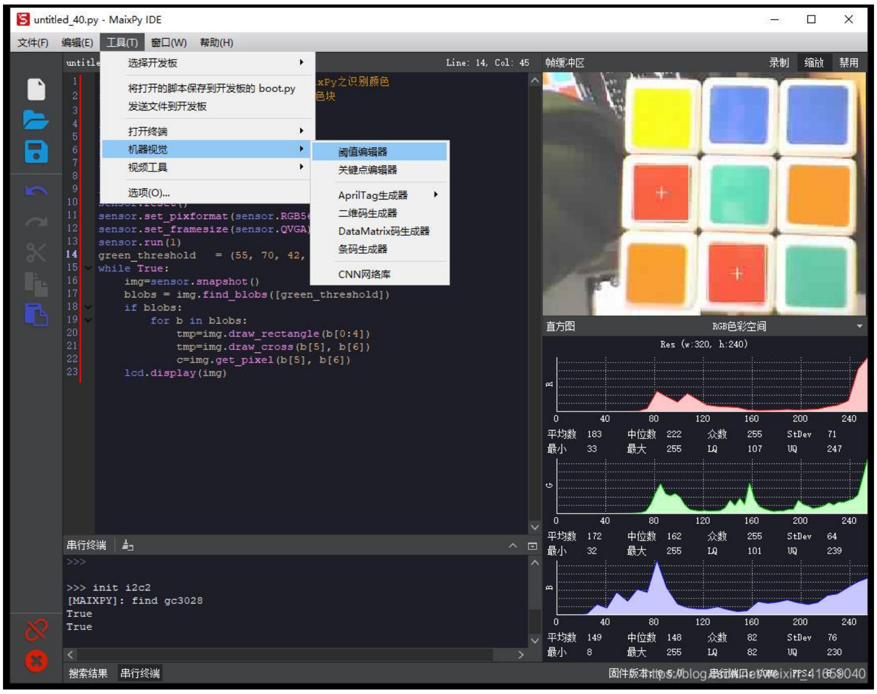
```
import sensor
import image
import lcd
import time

from Maix import GPIO
from fpioa_manager import fm, board_info
from machine import UART

clock = time.clock()

fm.register(34, fm.fpioa.UART2_TX, force=True)
fm.register(35, fm.fpioa.UART2_RX, force=True)
uart_Port = UART(UART.UART2, 115200,8,0,0, timeout=1000, read_buf_len= 4096)

lcd.init()
sensor.reset()
sensor.set_pixformat(sensor.RGB565)
sensor.set_framesize(sensor.QVGA)
sensor.run(1)
lcd.rotation(2)

#blue,red,green,yellow

colour = ['blue','red','green','yellow']

colour_threshold =([27, 55, -18, 4, -39, -20],
                   [55, 64, 14, 52, -9, 3],
                   [29, 65, -109, -18, 36, 59],
                   [30, 68, -25, -10, 52, 87])
blobs = [0,0,0,0]

def blobs_output(blobs):
    for b in blobs:
        tmp=img.draw_rectangle(b[0:4])
        tmp=img.draw_cross(b[5], b[6])
        img.draw_string(b[5], b[6], colour[i],color=(255,0,0), scale=2)
        c=img.get_pixel(b[5], b[6])
        _colour = {}
        _colour['colour'] = colour[i]
        data_ = []
        data_.append(_colour)
        data_.append(blobs[0])
    uart_Port.write(str(blobs))

def show_fps():
    fps =clock.fps()
    img.draw_string(200, 1, ("%2.2ffps" %(fps)),color=(255,0,0), scale=2)

while True:
    clock.tick()
    img=sensor.snapshot()
    show_fps()

    for i in range(4):
        blobs[i] = img.find_blobs([colour_threshold[i]],area_threshold=100,pixels_thre
        if blobs[i]:
            blobs_output(blobs[i])

    lcd.display(img)
```

Functional Assignment

- Identify four colors
- The serial port outputs data
- The display screen displays the recognized color blocks

Define the color you want to recognize

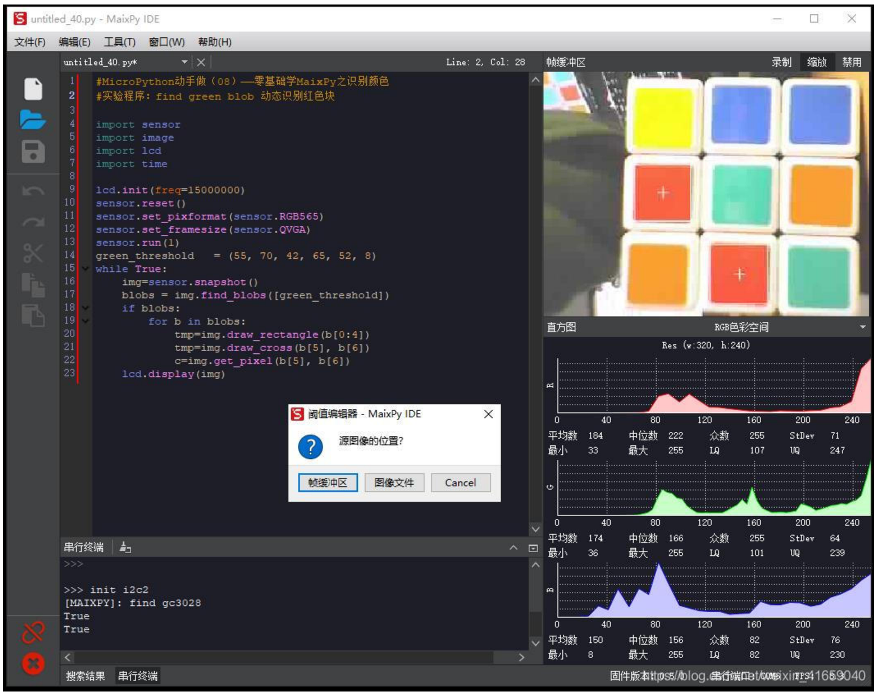
Open the MaixPy IDE and select Tool -- Machine Vision -- Curve Editor



```
xPy之识别颜色
色块

sensor.set_pixformat(sensor.RGB5
sensor.set_framesize(sensor.QVGA)
sensor.run(1)
green_threshold   = (55, 70, 42,
while True:
    img=sensor.snapshot()
    blobs = img.find_blobs([green_threshold])
    if blobs:
        for b in blobs:
            tmp=img.draw_rectangle(b[0:4])
            tmp=img.draw_cross(b[5], b[6])
            c=img.get_pixel(b[5], b[6])
    lcd.display(img)
```

```
>>>

>>> init i2c2
[MAIXPY]: find gc3028
True
True
```

Open source image location and select frame buffer.



```
#MicroPython动手做（08）——零基础学MaixPy之识别颜色
#实验程序: find green blob 动态识别红色块

import sensor
import image
import lcd
import time

lcd.init(freq=15000000)
sensor.reset()
sensor.set_pixformat(sensor.RGB565)
sensor.set_framesize(sensor.QVGA)
sensor.run(1)
green_threshold   = (55, 70, 42, 65, 52, 8)
while True:
    img=sensor.snapshot()
    blobs = img.find_blobs([green_threshold])
    if blobs:
        for b in blobs:
            tmp=img.draw_rectangle(b[0:4])
            tmp=img.draw_cross(b[5], b[6])
            c=img.get_pixel(b[5], b[6])
    lcd.display(img)
```

```
>>>

>>> init i2c2
[MAIXPY]: find gc3028
True
True
```

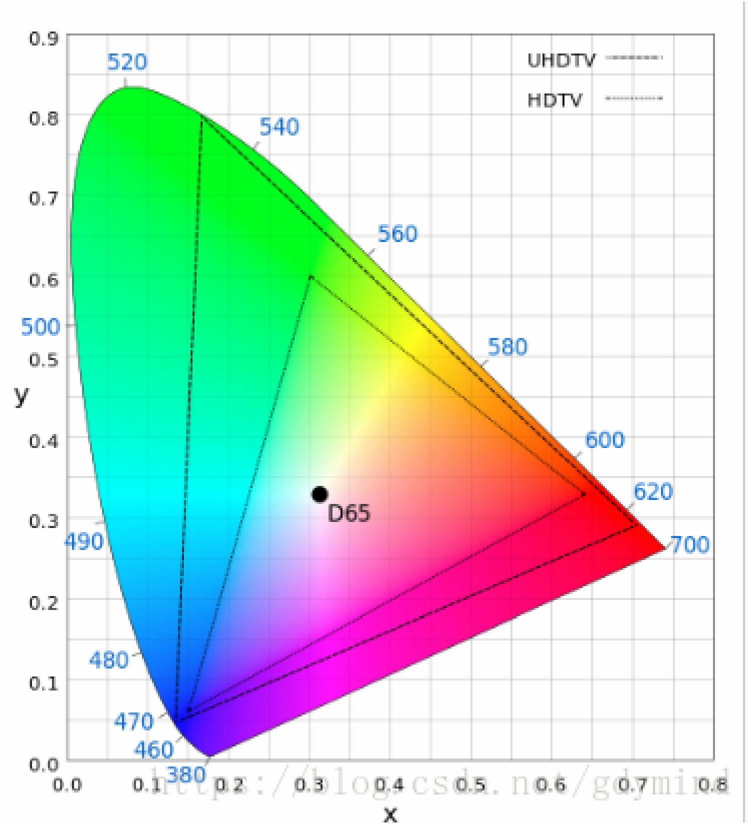
Adjust the Lab tie down value mainly in the binary image field, and white pixel is the tracked pixel

Related knowledge

MaixPy 机械视觉 API

Full Knowledge of Lab color space

- Name Before we begin, let's clarify the name of the Lab Color Space: The full name of Lab is CIELAB, sometimes also written CIE Lab*. CIE stands for the International Commission on Illumination, an International authority on lighting, colour, etc..
- Channel Lab is composed of a brightness channel and two color channels. In Lab color space, each color is represented by three numbers, L, a, b, and the meaning of each component is as follows: L stands for brightness a is the component from green to red b is the component from blue to yellow
- Perceptual uniform Lab is designed based on people's perception of colors. More specifically, it is perceptual uniform. Perceptual uniform means that if the number (L, a, b) changes equally, then it brings about a similar degree of visual change. Lab is more consistent with human vision than RGB and CMYK color space, and is easier to adjust. If you want to adjust the brightness (regardless of the Helmholtz -- Kohlrausch effect, refer to the note below), adjust the L channel, and if you want to adjust only the color balance, adjust a and b respectively. Note: Helmholtz–Kohlrausch effect is an illusion in the human eyes -- that colors appear brighter when they are saturated.
- **Device-independent** Lab has a nice feature -- Device-independent. That is, given a white point in the color space (the figure below represents a white spot in a color space), the color space can clearly determine how each color is created and displayed, regardless of the display medium used.



For example, when you want to convert an RGB image on the screen to a CMYK image for printing, you can first convert it from RGB to Lab and then convert the Lab image to CMYK mode. Because gamut of Lab is larger than RGB and CMYK (The gamut of Lab is so large that a large part of it is beyond the range of human vision that it cannot be called "color"). It is important to note that Lab defines the color relative to the white point. We will not know the other colors until we define the color of the white point (e.g. it is defined as CIE Standard Illuminant D50).

- **Range of value** In theory, L, a, and b are all real numbers, but in practice they are confined to a range of integers: The larger the L is, the higher the brightness is. When L is 0, it represents black, and when L is 100, it represents white. a and b are both gray when they are 0. When a goes from a negative number to a positive number, the corresponding color will change from green to red. When b goes from a negative number to a positive number, the corresponding color will change from blue to yellow. In practical application, we often use the range of color channel between -100~+100 or -128127.
- **Visualize** Lab* has three components in total, so it can be presented in three dimensional space. In two-dimensional space, the Chromaticity Diagram is often used to visualize it, that is, to fix the brightness L, to see the change of a and b. Note that these visualizations are not accurate, it’s just helpful to understand.
- CIELUV There is a color space similar to Cielab, called Cie 1976 (L, U, V), also known as Cieluv. The L of the color space is the same as the CIELAB, but the color component is different.
- **Conversion between LAB and RGB and CMYK** Since RGB and CMYK are both device related, they cannot be directly converted to Lab. So before the

conversion, you must define an absolute color space, such as sRGB or Adobe RGB. Conversion from RGB to SrGB is device independent, but the subsequent conversion are device independent.

8.3 Shape Recognition

Preparation for development

- Complete the device link
- Complete firmware burning
- Complete software environment

Example Code

```
# identify the rectangle
import sensor
import image
import lcd
import time
from Maix import GPIO
from fpioa_manager import fm, board_info
from machine import UART
clock = time.clock()
fm.register(34, fm.fpioa.UART2_TX, force=True)
fm.register(35, fm.fpioa.UART2_RX, force=True)
uart_Port = UART(UART.UART2, 115200,8,0,0, timeout=1000, read_buf_len= 4096)
lcd.init()
sensor.reset()
sensor.set_pixformat(sensor.RGB565)
sensor.set_framesize(sensor.QQVGA)
sensor.run(1)
lcd.rotation(2)
def uart_write(i):
    data_ = []
    data_.append(blobs[i])
    uart_Port.write(str(data_))
while True:
    img=sensor.snapshot()
    RECT = img.find_rects(threshould = 10000)
    if RECT:
        for b in RECT:
            img.draw_rectangle(b.rect(),color =(255,0,0))
            for p in b.corners():
                img.draw_circle(p[0],p[1],3,color = (0,255,0))
            c=img.get_pixel(b[0], b[1])
    lcd.display(img)
```

```
# identify the circle
import sensor, image, time, lcd
sensor.reset()
sensor.set_pixformat(sensor.RGB565)
sensor.set_framesize(sensor.QVGA)
sensor.skip_frames(time = 300)
sensor.run(1)
lcd.init()
lcd.rotation(2)
clock = time.clock()
while(True):
    clock.tick()
    img = sensor.snapshot()
    for c in img.find_circles(threshold = 1800, x_margin = 20, y_margin = 20, r_margir
            r_min = 5, r_max = 45, r_step = 20):
        img.draw_circle(c.x(), c.y(), c.r(), color = (255, 0, 0))#识别到的红色圆形用红色
    print("r %f" % c.r())
    print("FPS %f" % clock.fps())
    lcd.display(img)
```

Functional Assignment

- Identify the specified shape
- Return the corresponding data

Relate knowledge

MaixPy 机械视觉 API

8.4 Face Recognition

Preparation for development

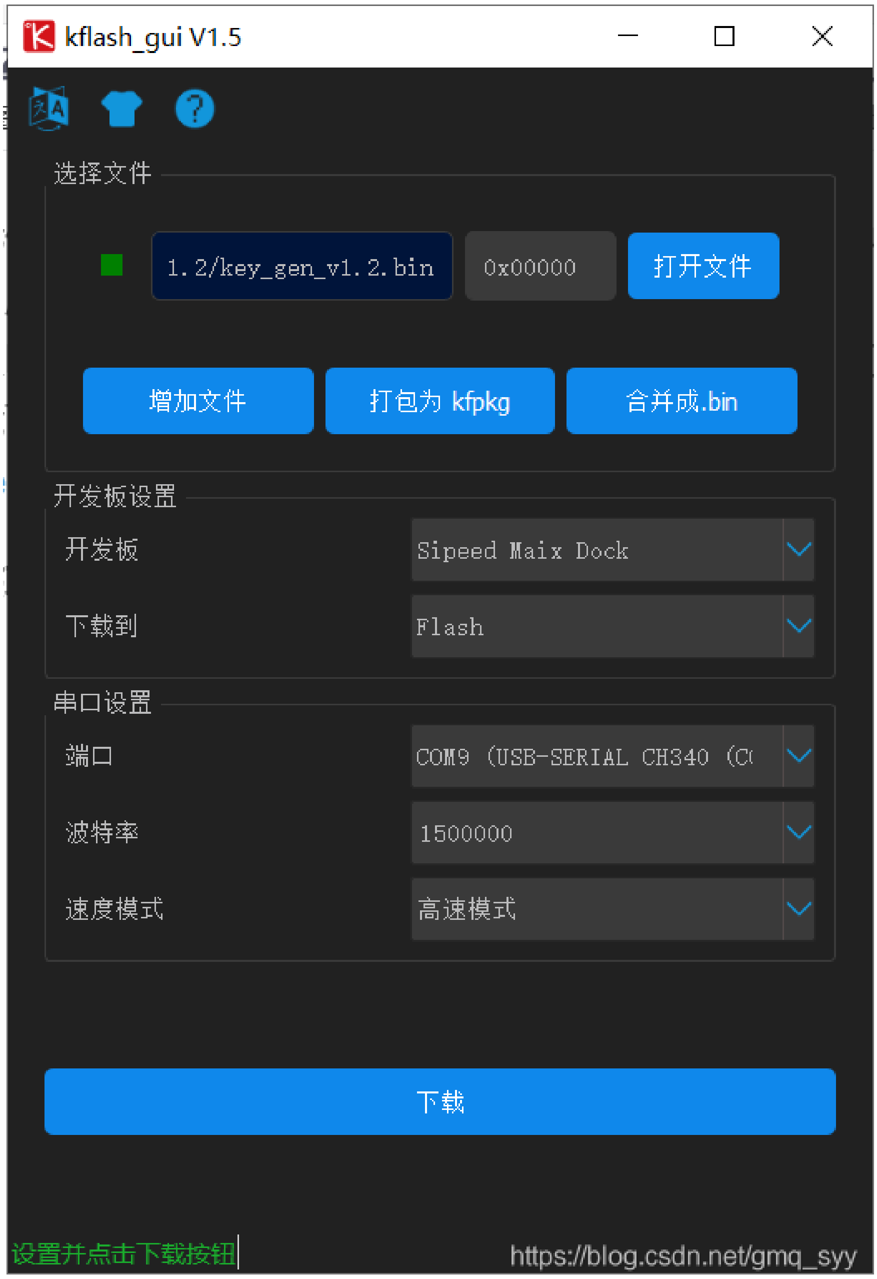
- Complete the device link
- Complete firmware burning
- Complete software environment

Burn specified firmware

We use the MaixPy IDE to run this routine. First, we need to download the model we need from the MaixHub, which need us to provide the machine code, so we download the bin file that can get the machine code, then burn it token_gen bin file address

Get Machine code

Use kfalsh_gui to burn ken_gen.bin file，click it to download.


kflash_gui V1.5

选择文件

🟩 1.2/key_gen_v1.2.bin 0x00000 打开文件

增加文件 打包为 kfpkg 合并成.bin

开发板设置

开发板 Sipeed Maix Dock

下载到 Flash

串口设置

端口 COM9 (USB-SERIAL CH340 (C(

波特率 1500000

速度模式 高速模式

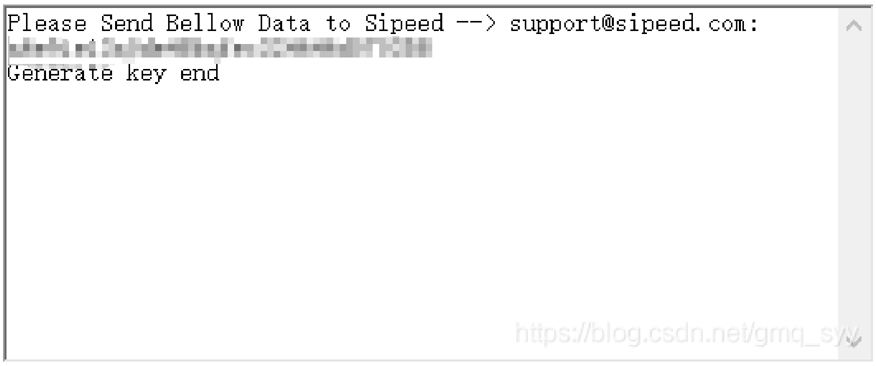
下载

设置并点击下载按钮

https://blog.csdn.net/gmq_syy

View the machine code in the serial output

After the firmware is burned, connect the computer, open the serial assistant, then open the serial port.



```
Please Send Bellow Data to Sipeed --> support@sipeed.com:
[redacted]
Generate key end
```

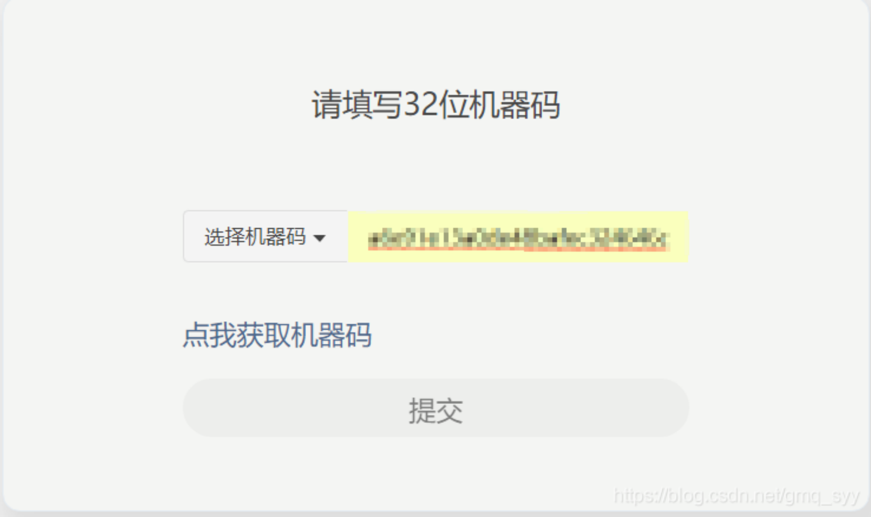
https://blog.csdn.net/gmq_syy

Download model

With the machine code we can start to download the corresponding model files.

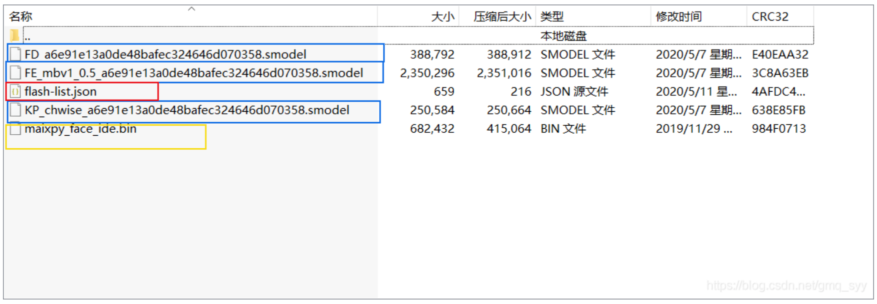
Model download address

model download address



[redacted]

List of files to extract :



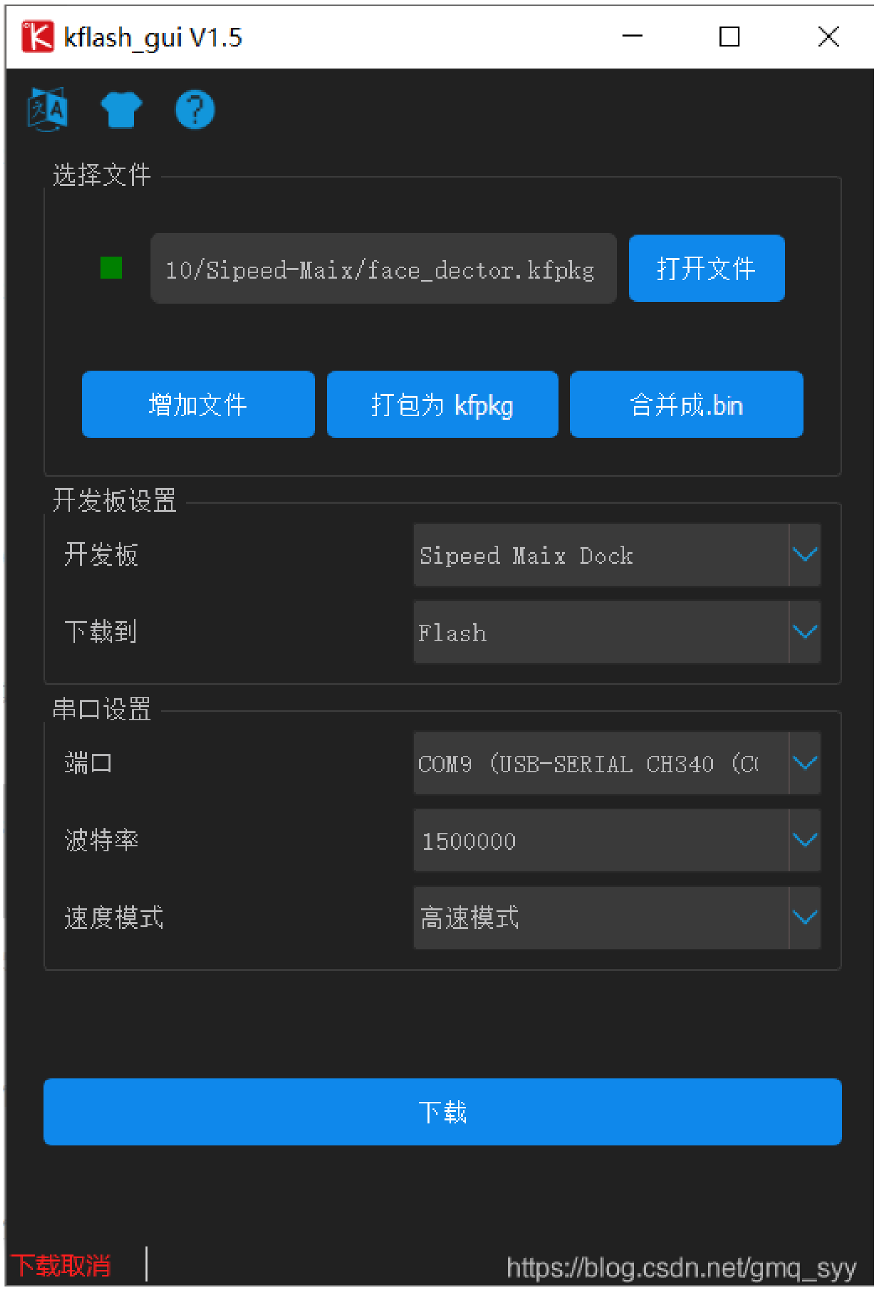
名称	大小	压缩后大小	类型	修改时间	CRC32
..			本地磁盘		
FD a6e91e13a0de48bafec324646d070358.smodel	388,792	388,912	SMODEL 文件	2020/5/7 星期...	E40EAA32
FE_mbv1_0.5_a6e91e13a0de48bafec324646d070358.smodel	2,350,296	2,351,016	SMODEL 文件	2020/5/7 星期...	3C8A63EB
flash-list.json	659	216	JSON 源文件	2020/5/11 星...	4AFDC4...
KP_chwise_a6e91e13a0de48bafec324646d070358.smodel	250,584	250,664	SMODEL 文件	2020/5/7 星期...	638E85FB
maixpy_face_ide.bin	682,432	415,064	BIN 文件	2019/11/29 ...	984F0713

The blue part is the model file, and the yellow part is the bin file of MaixPy.This is the compact version, and the size is relatively small, just more than 600 KB. Inside the json file is the configuration, which is about where these files should be downloaded to Flash and whether they need to be checked.

```
{
    "version": "0.1.0",
    "files": [
        {
            "address": 0,
            "bin": "maixpy_face_ide.bin",
            "sha256Prefix": true
        },
        {
            "address": 5242880,
            "bin": "FD_a6e91e13a0de48bafec324646d070358.smodel",
            "sha256Prefix": false
        },
        {
            "address": 6291456,
            "bin": "KP_chwise_a6e91e13a0de48bafec324646d070358.smodel",
            "sha256Prefix": false
        },
        {
            "address": 7340032,
            "bin": "FE_mbv1_0.5_a6e91e13a0de48bafec324646d070358.smodel",
            "sha256Prefix": false
        }
    ]
}
```

Burn model files

Download the Kkfpkg file to our development board using kfalsh_gui and run it



选择文件

🟩 10/Sipeed-Maix/face_dector.kfpkg

开发板设置

开发板: Sipeed Maix Dock

下载到: Flash

串口设置

端口: COM9 (USB-SERIAL CH340 (C

波特率: 1500000

速度模式: 高速模式

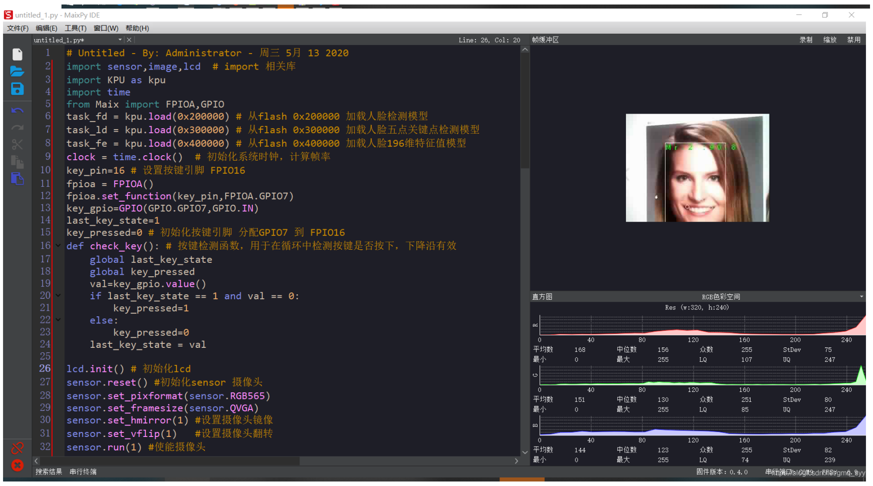
Run example code

code address

```
import sensor,image,lcd  # import 相关库
import KPU as kpu
import time
from Maix import FPIOA,GPIO
task_fd = kpu.load(0x200000) # 从flash 0x200000 加载人脸检测模型
task_ld = kpu.load(0x300000) # 从flash 0x300000 加载人脸五点关键点检测模型
task_fe = kpu.load(0x400000) # 从flash 0x400000 加载人脸196维特征值模型
clock = time.clock()  # 初始化系统时钟，计算帧率
key_pin=16 # 设置按键引脚 FPIO16
fpioa = FPIOA()
fpioa.set_function(key_pin,FPIOA.GPIO36)
key_gpio=GPIO(GPIO.GPIO36,GPIO.IN)
last_key_state=1
key_pressed=0 # 初始化按键引脚 分配GPIO7 到 FPIO16
def check_key(): # 按键检测函数，用于在循环中检测按键是否按下，下降沿有效
    global last_key_state
    global key_pressed
    val=key_gpio.value()
    if last_key_state == 1 and val == 0:
        key_pressed=1
    else:
        key_pressed=0
    last_key_state = val

lcd.init() # 初始化lcd
sensor.reset() #初始化sensor 摄像头
sensor.set_pixformat(sensor.RGB565)
sensor.set_framesize(sensor.QVGA)
sensor.set_hmirror(1) #设置摄像头镜像
sensor.set_vflip(1)   #设置摄像头翻转
sensor.run(1) #使能摄像头
anchor = (1.889, 2.5245, 2.9465, 3.94056, 3.99987, 5.3658, 5.155437, 6.92275, 6.718375
dst_point = [(44,59),(84,59),(64,82),(47,105),(81,105)] #standard face key point posit
a = kpu.init_yolo2(task_fd, 0.5, 0.3, 5, anchor) #初始化人脸检测模型
img_lcd=image.Image() # 设置显示buf
img_face=image.Image(size=(128,128)) #设置 128 * 128 人脸图片buf
a=img_face.pix_to_ai() # 将图片转为kpu接受的格式
record_ftr=[] #空列表 用于存储当前196维特征
record_ftrs=[] #空列表 用于存储按键记录下人脸特征， 可以将特征以txt等文件形式保存到sd卡后，
names = ['Mr.1', 'Mr.2', 'Mr.3', 'Mr.4', 'Mr.5', 'Mr.6', 'Mr.7', 'Mr.8', 'Mr.9' , 'Mr.
while(1): # 主循环
    check_key() #按键检测
    img = sensor.snapshot() #从摄像头获取一张图片
    clock.tick() #记录时刻，用于计算帧率
    code = kpu.run_yolo2(task_fd, img) # 运行人脸检测模型，获取人脸坐标位置
    if code: # 如果检测到人脸
        for i in code: # 迭代坐标框
            # Cut face and resize to 128x128
            a = img.draw_rectangle(i.rect()) # 在屏幕显示人脸方框
            face_cut=img.cut(i.x(),i.y(),i.w(),i.h()) # 裁剪人脸部分图片到 face_cut
            face_cut_128=face_cut.resize(128,128) # 将裁出的人脸图片 缩放到128 * 128像素
            a=face_cut_128.pix_to_ai() # 将猜出图片转换为kpu接受的格式
            #a = img.draw_image(face_cut_128, (0,0))
            # Landmark for face 5 points
            fmap = kpu.forward(task_ld, face_cut_128) # 运行人脸5点关键点检测模型
            plist=fmap[:] # 获取关键点预测结果
            le=(i.x()+int(plist[0]*i.w() - 10), i.y()+int(plist[1]*i.h())) # 计算左眼位
            re=(i.x()+int(plist[2]*i.w()), i.y()+int(plist[3]*i.h())) # 计算右眼位置
            nose=(i.x()+int(plist[4]*i.w()), i.y()+int(plist[5]*i.h())) #计算鼻子位置
            lm=(i.x()+int(plist[6]*i.w()), i.y()+int(plist[7]*i.h())) #计算左嘴角位置
            rm=(i.x()+int(plist[8]*i.w()), i.y()+int(plist[9]*i.h())) #右嘴角位置
            a = img.draw_circle(le[0], le[1], 4)
            a = img.draw_circle(re[0], re[1], 4)
            a = img.draw_circle(nose[0], nose[1], 4)
            a = img.draw_circle(lm[0], lm[1], 4)
            a = img.draw_circle(rm[0], rm[1], 4) # 在相应位置处画小圆圈
            # align face to standard position
            src_point = [le, re, nose, lm, rm] # 图片中 5 坐标的位置
            T=image.get_affine_transform(src_point, dst_point) # 根据获得的5点坐标与标准
            a=image.warp_affine_ai(img, img_face, T) #对原始图片人脸图片进行仿射变换，变拮
            a=img_face.ai_to_pix() # 将正脸图像转为kpu格式
            #a = img.draw_image(img_face, (128,0))
            del(face_cut_128) # 释放裁剪人脸部分图片
            # calculate face feature vector
            fmap = kpu.forward(task_fe, img_face) # 计算正脸图片的196维特征值
            feature=kpu.face_encode(fmap[:]) #获取计算结果
            reg_flag = False
            scores = [] # 存储特征比对分数
            for j in range(len(record_ftrs)): #迭代已存特征值
                score = kpu.face_compare(record_ftrs[j], feature) #计算当前人脸特征值与
                scores.append(score) #添加分数总表
            max_score = 0
            index = 0
            for k in range(len(scores)): #迭代所有比对分数，找到最大分数和索引值
                if max_score < scores[k]:
                    max_score = scores[k]
                    index = k
            if max_score > 85: # 如果最大分数大于85， 可以被认定为同一个人
                a = img.draw_string(i.x(),i.y(), ("%s :%2.1f" % (names[index], max_sco
            else:
                a = img.draw_string(i.x(),i.y(), ("X :%2.1f" % (max_score)), color=(25
            if key_pressed == 1: #如果检测到按键
                key_pressed = 0 #重置按键状态
                record_ftr = feature
                record_ftrs.append(record_ftr) #将当前特征添加到已知特征列表
            break
    fps =clock.fps() #计算帧率
    print("%2.1f fps"%fps) #打印帧率
    a = lcd.display(img) #刷屏显示
    #kpu.memtest()

#a = kpu.deinit(task_fe)
#a = kpu.deinit(task_ld)
#a = kpu.deinit(task_fd)
```

Run the program with MaixPy IDE

Program running as shown in figure



```
# Untitled - By: Administrator - 周三 5月 13 2020
import sensor,image,lcd  # import 相关库
import KPU as kpu
import time
from Maix import FPIOA,GPIO
task_fd = kpu.load(0x200000) # 从flash 0x200000 加载人脸检测模型
task_ld = kpu.load(0x300000) # 从flash 0x300000 加载人脸五点关键点检测模型
task_fe = kpu.load(0x400000) # 从flash 0x400000 加载人脸196维特征值模型
clock = time.clock()  # 初始化系统时钟，计算帧率
key_pin=16 # 设置按键引脚 FPIO16
fpioa = FPIOA()
fpioa.set_function(key_pin,FPIOA.GPIO7)
key_gpio=GPIO(GPIO.GPIO7,GPIO.IN)
last_key_state=1
key_pressed=0 # 初始化按键引脚 分配GPIO7 到 FPIO16
def check_key(): # 按键检测函数，用于在循环中检测按键是否按下，下降沿有效
    global last_key_state
    global key_pressed
    val=key_gpio.value()
    if last_key_state == 1 and val == 0:
        key_pressed=1
    else:
        key_pressed=0
    last_key_state = val

lcd.init() # 初始化lcd
sensor.reset() #初始化sensor 摄像头
sensor.set_pixformat(sensor.RGB565)
sensor.set_framesize(sensor.QVGA)
sensor.set_hmirror(1) #设置摄像头镜像
sensor.set_vflip(1)   #设置摄像头翻转
sensor.run(1) #使能摄像头
```

Press BUTTON A to record the face. After the face is recorded, the name will be assigned in order and displayed when the face is recognized.

Related knowledges

8.5 QR code Identification

Preparation for development

- Complete the device link
- Complete firmware burning
- Complete software environment
- Use MaixPy to generate recognizable QR codes Select -> Tools -> Machine Vision -> AprilTag ->TAG36H11 to generate the recognizable code

example code

```
# AprilTags3D定位例程
#
# 这个例子展示了OpenMV Cam的功能，可以检测OpenMV Cam M7/H7 上的April标签。OpenMV2 M4版本无

import sensor, image, time, math, lcd

sensor.reset()
sensor.set_pixformat(sensor.RGB565)
sensor.set_framesize(sensor.QQVGA) # 如果分辨率太高，内存可能溢出
sensor.skip_frames(time = 2000)
#sensor.set_auto_gain(False)  # 必须关闭此功能，以防止图像冲刷…
#sensor.set_auto_whitebal(False)  # 必须关闭此功能，以防止图像冲刷…
clock = time.clock()
lcd.init()
lcd.rotation(2)

# 注意！与find_qrcodes不同，find_apriltags方法不需要对镜像进行镜头校正。

#标签系列有什么区别？ 那么，例如，TAG16H5家族实际上是一个4x4的方形标签。
#所以，这意味着可以看到比6x6的TAG36H11标签更长的距离。 然而，较低的H值（H5对H11）
#意味着4x4标签的假阳性率远高于6x6标签。 所以，除非你有理由使用其他标签系列，
#否则使用默认族TAG36H11。


# AprilTags库输出标签的姿势信息。 这是x / y / z平移和x / y / z旋转。
# x / y / z旋转以弧度表示，可以转换为度数。 至于翻译单位是无量纲的，
# 你必须应用一个转换函数。

# f_x是相机的x焦距。它应该等于以mm为单位的镜头焦距除以x传感器尺寸（以mm为单位）乘以图像中的像
# 以下数值适用于配备2.8毫米镜头的OV7725相机。

# f_y是相机的y焦距。它应该等于以mm为单位的镜头焦距除以y传感器尺寸（以mm为单位）乘以图像中的像
# 以下数值适用于配备2.8毫米镜头的OV7725相机。

# c_x是以像素为单位的图像x中心位置
# c_x是以像素为单位的图像x中心位置

f_x = (2.8 / 3.984) * 160 # find_apriltags 如果没有设置，则默认为这个
f_y = (2.8 / 2.952) * 120 # find_apriltags 如果没有设置，则默认为这个
c_x = 160 * 0.5 # find_apriltags 如果没有设置，则默认为这个 (the image.w * 0.5)
c_y = 120 * 0.5 # find_apriltags 如果没有设置，则默认为这个 (the image.h * 0.5)

def degrees(radians):
    return (180 * radians) / math.pi

while(True):
    clock.tick()
    img = sensor.snapshot()
    for tag in img.find_apriltags(fx=f_x, fy=f_y, cx=c_x, cy=c_y): # 默认为 TAG36H11
        tmp = img.draw_rectangle(tag.rect(), color = (255, 0, 0))
        tmp = img.draw_cross(tag.cx(), tag.cy(), color = (0, 255, 0))
        print_args = (tag.x_translation(), tag.y_translation(), tag.z_translation(), \
            degrees(tag.x_rotation()), degrees(tag.y_rotation()), degrees(tag.z_rotati
        # 变换单位不详。旋转单位是度数。
        print("Tx: %f, Ty %f, Tz %f, Rx %f, Ry %f, Rz %f" % print_args)
    print(clock.fps())
    lcd.display(img)
```

功能解析

- 识别指定的QR码
- 串口输出数据
- 显示屏显示识别到的QR码

扩展知识

MaixPy 机械视觉 API

april官网

1 Maintenance

After-sales Service

Return service is limited to goods not opened within 7 days after the receipt date of logistics of the products. The freight or other risks incurred in return shall be borne by the customer.
Customers should provide the purchasing invoice and warranty card as the warranty certification when a warranty is being asked.
Elephant Robotics will be responsible for the hardware faults of products caused by the normal using during the warranty period.

The warranty period starts from the date of purchase or the receipt date of the logistics.
The faulty parts from the products will be owned by Elephant Robotics, and the appropriate cost will be charged if necessary.

If you need to apply for warranty service, please contact our customer service first to confirm the detailed information. The following is warranty terms of detailed components:

Note: If there is a conflict with the Product Brochure, the User Manual shall prevail.

Servos

Warranty Period	Warranty Services
≤1 months	Elephant Robotics offers a free new sever motor and bear the freight.
1-3 months	Elephant Robotics offers a free new sever motor, customs shall bear the freight.
≥3 months	Customers need to buy it themselves.

Electrical Parts（M5 Hardware）

Warranty Period	Warranty Services
≤3 months	Customers need to send it back after disassembly, Elephant Robotics shall send a new one for free and bear the freight out and home.
3-6 months	Customers need to send it back after disassembly and bear the freight out and home, Elephant Robotics shall send a new one for free.
≥6 months	Customers need to buy it themselves.

Structure Parts, including Shell Parts

保修期限	保修服务
≤1 years	Elephant Robotics offers free new components once, customs shall bear the freight.
≥1 years	Customers need to buy it themselves.

During the warranty period of the delivered product, the company only repairs the malfunctions that occur during normal use of the robot for free. However, in the following cases, the customer will be charged for repairs (even during the warranty period):

- Damage or malfunction caused by incorrect use and improper use different from the manual.
- Failure caused by unauthorized disassembly by the customer.
- Damage caused by improper adjustment or unauthorized repairs.
- Damage caused by natural disasters such as earthquakes and floods.

please strictly follow the instructions in this manual and related manual to operate the robot.

Limit

Joint	Range of Motion/Angle
Joint 1	-165 ~ +165
Joint 2	-165 ~ +165
Joint 3	-165 ~ +165
Joint 4	-165 ~ +165
Joint 5	-165 ~ +165
Joint 6	-175 ~ +175

- After we receive myCobot, we should pay attention to the limit of each joint of myCobot, and the rotation angle of each axis cannot exceed its own maximum physical limit.
- It should be turned at a small angle and gently. After reaching the limit, you should not continue turning.
- Be careful not to touch the robot arm itself when the joint is running so as not to knock it out.

- ## Connectivity issues (November 2020 batch)

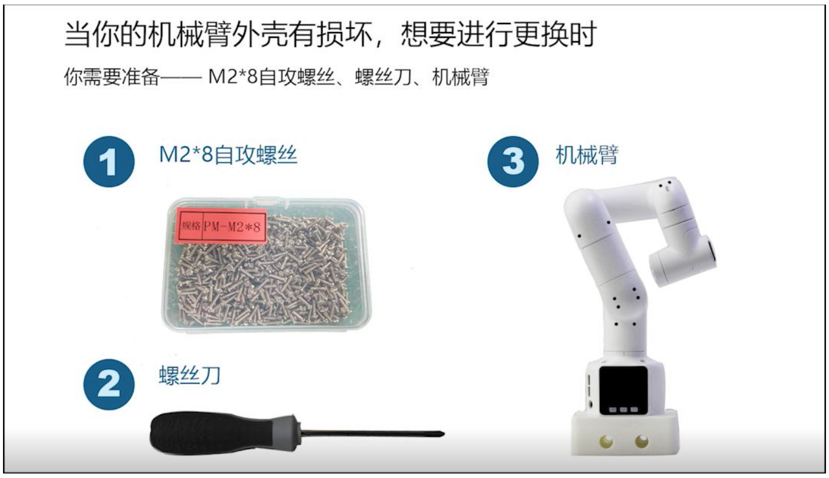
Problem description: when the arm suddenly loses force and can not be driven, open the J5 joint shell to check whether the connection line is off, then need to cut off the longer pin, attach insulation tape and then put back to the steering gear, and cover the cover plate. ****Replacement steps**

- Need to remove back cover housing
- Cable adjustment and adhesive tape Youtube-video link: https://youtu.be/1wq0kTJVqw4

Shell Disassembly

Youtube-[Video Link: https://youtu.be/wHzFsExkYrE] **Steps**

- Prepare the items as shown, loosen the screws of the corresponding shell, force properly, and gently remove the shell from the left and right.
- After removing the shell, check that there is no damage inside, you can install the new shell.
- I Fasten the new shell, the shell and the manipulator have a clear touch, after fastening, the screws are placed and tightened to complete the shell removal.

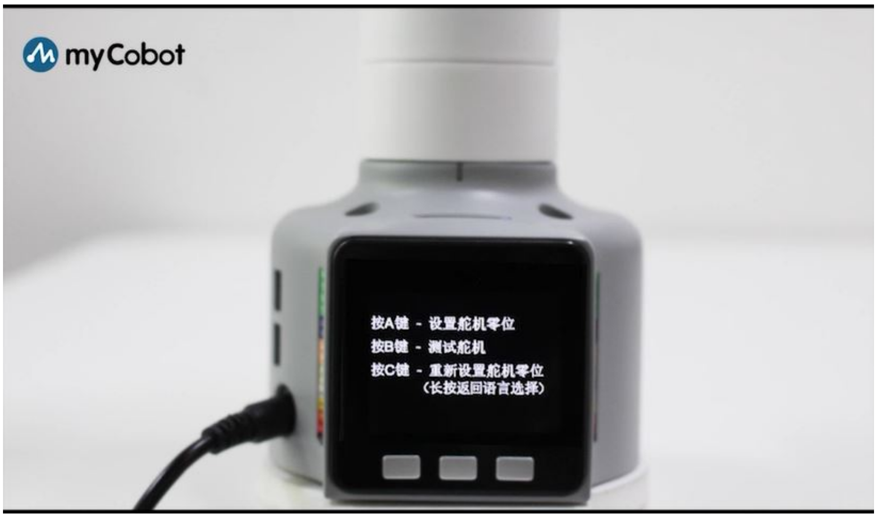


当你的机械臂外壳有损坏，想要进行更换时

你需要准备—— M2*8自攻螺丝、螺丝刀、机械臂

calibration

Youtube-video Link：https://youtu.be/vGznxW4OF10



steps

- Burning calibration on myStudio
- When the zero position of the manipulator is aligned with the zero position calibration groove of each joint
- Adjust the zero position calibration groove manually and press the A key (leftmost, set the steering gear zero) to calibrate each joint in turn
- After calibration, the basic screen will show that all steering gear bits have been set.
- Press the B key (middle key) to test the steering gear, the test is finished and the correction is over.
- If not, press the C key (right-click) to reset the steering gear zero and repeat the above steps.

2 常见问题FAQ

faq

Hardware

Q： why it appear small jitter when I use the vertical state, but not when running?

A：please check whether it is vertical, vertical state is not affected by gravity, mechanical voids lead to small jitter, after the use of this state will not occur. The recommended speed in this state is 400-500.

Q: Will ROS system charge later ?

A：ROS is open source and will update on our github. There is no charge for firmware upgrades.

Q：myCobot joint hard limit

A： The first axis and fifth axis have limit, the first axis clockwise about 160°, counterclockwise about 160°. the fifth axis clockwise and counterclockwise rotation about 160° Note: when turning the robot arm, it should be small angle and gently, after reaching the limit. you can not continue to rotate.

Q： what is the unit of velocity of the robotic arm?

A： speed 180 degrees per second

Q：Is it just not support M5STACK's steering gear, will it support other M5 sensors?

A：M5STACK Basic currently supports all sensors.

Q：Do Atom support sensors in the future?

A：atom support needs to further open the interface, within our plan, need to wait

Software

Q： why can't my compiler find the corresponding device?

A：You need to build the development environment and install the corresponding project library to develop the equipment.

Q： why my compiler can't compile the sample program properly?

A： the required project library is not installed or the project inventory is in conflict, it is recommended to check that the project library is installed correctly. If the installation is correct and still can not be compiled, please reinstall the arduino development environment.

Q： why did I burn the firmware to the ATOM terminal and the device didn't work properly?

A：ATOM terminal firmware needs to use our factory firmware, you can not change other unofficial firmware in use, If the device burn other firmware accidentally. You can use "myCobot firmware burner" to select ATOM terminal-select serial port-select ATOMMAIN firmware to burn the ATOM terminal.

l **UIFlow**

Q：UI Flow don't support with the latest firmware? A： Need M5STACK update, we have asked M5 to help us update, the time cycle is unknown.

l **RoboFlow**

Q： Can I use robotStudio software programming A： Our own industrial programming software RoboFlow can be used, robotStudio is ABB company, it can not communicate with us.

l **ROS**

Q：ROS version A：/ rosdistro：kinetic /rosversion：1.12.17

Q：Does the ROS system operate with a computer attached to the robot arm? A： Yes, connect with the computer is ok.

Q：Ros folder downloaded from github runs only display controls but not display myCobot 3D models. A： You need to open rviz,rosrun rviz rviz manually

Q：Range out or error[101] occurs when in runtime A： Check that the serial port is correct, basic and atom firmware are correct.

Q： operation mode A： It need to open 3 terminal for current operation

l **myStudio**

Q：Does myStudio replace the UIFlow? A： No, it insteads of the current downloader.

l **myCobot phone controller**

Q：I have a few questions to ask about APP: 1) Bluetooth has two controls on the right, one is back to zero, the other is? 2) The sixth axis has no mechanical zero, how to get the robot back to zero? Rz the Euler angle is 0. 3) Press the control button of the joint and end shaft, then loose, will it keep moving? A：1. Another free model.

2.When the robot arm is in calibration correction, the current position is set to zero by default.

3.After release, you should stop. APP use the jog control. However, due to the instability of Bluetooth connection, there may be a stop instruction but arm do not received.

l **Firmware Burner**

Q： Motor don't work A： Use the latest version 1.3 firmware burner, burn atommain to the atom. Only if computer link to the end of the robotic arm typeC interface can you update the firmware to atom.

Q： After burn, how to return to the factory settings? A： Download firmware burner or arduino program, burn main firmware.

l **Others**

Q：What is the difference between MainControl and Transponder?

A：MainControl is the factory with its own, transponder is the transfer program, after burning can be directly sent packet protocol control.

Q： default software - Drag to teach, action can only be saved to ram can not be saved to flash A： old version will be released with the Chinese version mainControl after the bug adjustment

Q： each Data is a hexadecimal number, right, and then converted to hex characters to send to the machine A： right, serial communication sends and receives data in HEX form

Others

Q：Can I get real-time data while executing instructions

A：It is not allowed for the time being, the bus can not be disturbed when working

Q： The end interface rudder model has a limited end of the control card, what is the relationship with the M5stack? Is it independent?

A： End of the device needs to be able to adapt to our controller, it can support PWM mode control of the steering gear.

Q：what does the speed parameter 0-100 in the WrigeAngle mean

A： speed parameter 0-100 is the percentage of speed

Q： Can I use GetAngles to get the angle information of 6 axes?

The return value of the

A：GetAngles is a float array with a length of 6. The index starts at 0, followed by 6 joints

Q： Is servo or stepper motor?

A：myCobot carry six servo motors.

Q： repeated positioning accuracy

A：+/-0.5mm

Q： what is the reducer structure?

A： reducer is gear set.

Q： what is the gear?

A： metal tooth structure

Q： what is this programmed with?

A：myCobot is open ROS, now supports most platforms on the market including UIFLOW,Arduino,microPython,FreeROTS

Q：How is the Real Time

A：Uart communications, maximum 50 ms delay, normal 5 ms

Q:where can the detailed documentation, sample code of the Arduino library and API interface be seen?

A： check our github: https://github.com/elephantrobotics/myCobot

Q： How much time continuous operation / motor life

A：300-500 hours, rest 15-30 minutes before use will prolong the life of the arm;

Q： shell material

A： plastic; photosensitive resin; SLA

Q：Do the robot arm have torque sensors

A： most collaborative robots don't have torque sensors

Q：Brand of motor

A： self-made motor, external processing

Q： Turn the joint tape encoder?

A： yes

Q： not harmonics?

A： notss

Q： I don't understand servo steering gear , is it a steering gear like a model?

A：it is a steering gear like a model, but we customize and modify a lot, is more suitable for the manipulator. Six-axis linkage, three-power interpolation, not as uniform as industrial robots, the final price here.

Q：motor parameters

A：large steering gear: rated load 10 kgcm, maximum speed 60 rpm, voltage 7.4 V, encoder 12 bit magnetic knitting.

small steering gear: rated load 2 kgcm, maximum speed 80 rpm, voltage 7.4 V, encoder 12 bit magnetic knitting.

Q：ROS version

l rosdistro：kinetic //

l rosversion：1.12.17//

Q： Angle Accuracy

l 360/4096=0.0879°

Q： power supply range

I 6~9V

3~5A

Q：IO input, output, voltage, how to debug; why marked 5V?

A：3.3 V,atom of IO control is gradually open; there are 5V of power; debugging needs to see which platform to develop, arduino,uiFlow,python is different

Q：IO port connection problem

A： IO according to the position, it is divided into the IO, on the basic of the manipulator base and the IO;IO at the end of the manipulator, which is divided into digital signal input and output DIO, analog signal, communication I2C,SPI, power grounding.

Q：Are encoders absolute or incremental?

A： absolute value (steering gear with high precision encoder)

Q：asic software brush uiflow, later with your burning software brush back maincontrol, why the robot arm track recording software can not be used

A：atom firmware version is not compatible, can only use the same version of firmware; update the atom version can

Q： why did it remind TimeOutwhen burning?

A： power up again or press the reset button when burning

Q： why did lights go out when I burn Bluetooth or Transponder atom

A： need command control to reopen LED, close by default

Q： why my equipment can't be used after burning the demo program

A： need to check your version number, corresponding to the version number of firmware can, different versions of the program is incompatible

Q： the accuracy range deviation of the manipulator is a little large

A： the current accuracy, we need more precision to view our industrial arm

Q： I can't drag to teach after burning

A： First check whether the basic and atom are burned; whether the burned firmware corresponds to the requirements to be achieved; and whether the burning is the latest firmware bottom Basic burning atommain at the top

Q：can't control robotic arm when burn transponder

A：Atom also need to burn, burn firmware atom2.1alpha; Basic burn transponder

Q： by which of the six steering gear

A：Atom control at top

Q：python API sample tutorial

A： currently have demo, test code under github Test folder

Q：Do you need to start again after drag the instruction

A： because when re-recording joint motor lock position can not be cut off, click on the release can make the arm joint loose, can be the next drag teaching; power failure can also start again

Q：Does instruction be achieved by reading the set potential value (drag instruction principle)

A： principle is like this

Q：What information does the send to the joint motor through the ROS? Can feedback encoder information

A： potential value; read potential value

Q： show six values in Get coords

A：x,y,z,rx,ry,rz, translation plus rotation; z-y-x compliance

Q：AC flash red light, robotic arm screen not on

A： if the motor doesn't move, may be Basic is broken. Ac frequently flash red lights mean power supply, then power off and power supply. It could be some pcba broken.

3 Resources

Website

Official Website

https://www.elephantrobotics.com/en/

myCobot github-software

https://github.com/elephantrobotics/myCobot

M5 UI Flow

https://docs.m5stack.com/#/zh_CN/quick_start/m5core/m5stack_core_get_started_MicroPython

Videos

youtube

- maintenance
  - connecting line https://youtu.be/1wq0kTJVqw4
  - disassembly https://youtu.be/wHzFsExkYrE
  - limit https://youtu.be/PUeU-mynljw
  - calibration https://youtu.be/vGznxW4OF10
- Tutorials unboxing https://youtu.be/Lwi8UoihzNc
  - free move https://youtu.be/WzrbOrdQop0
  - Maincontrol https://youtu.be/VKd8b989M8g
  - ROS https://youtu.be/-Jo_IJ8RaXc
  - Arduino https://youtu.be/pkQIApDRJpo
  - myStudio https://youtu.be/Kr9i62ZPf4w
- others
  - 2 display screens-traffic signals https://youtu.be/9ej0tEwhXuE
  - promotional video https://youtu.be/uSw5rsymjVY
  - User cases(1) https://youtu.be/0AI1MN50RS0
  - User cases(2) https://youtu.be/eoR2-MId_-I