

Table of Contents

Introduction

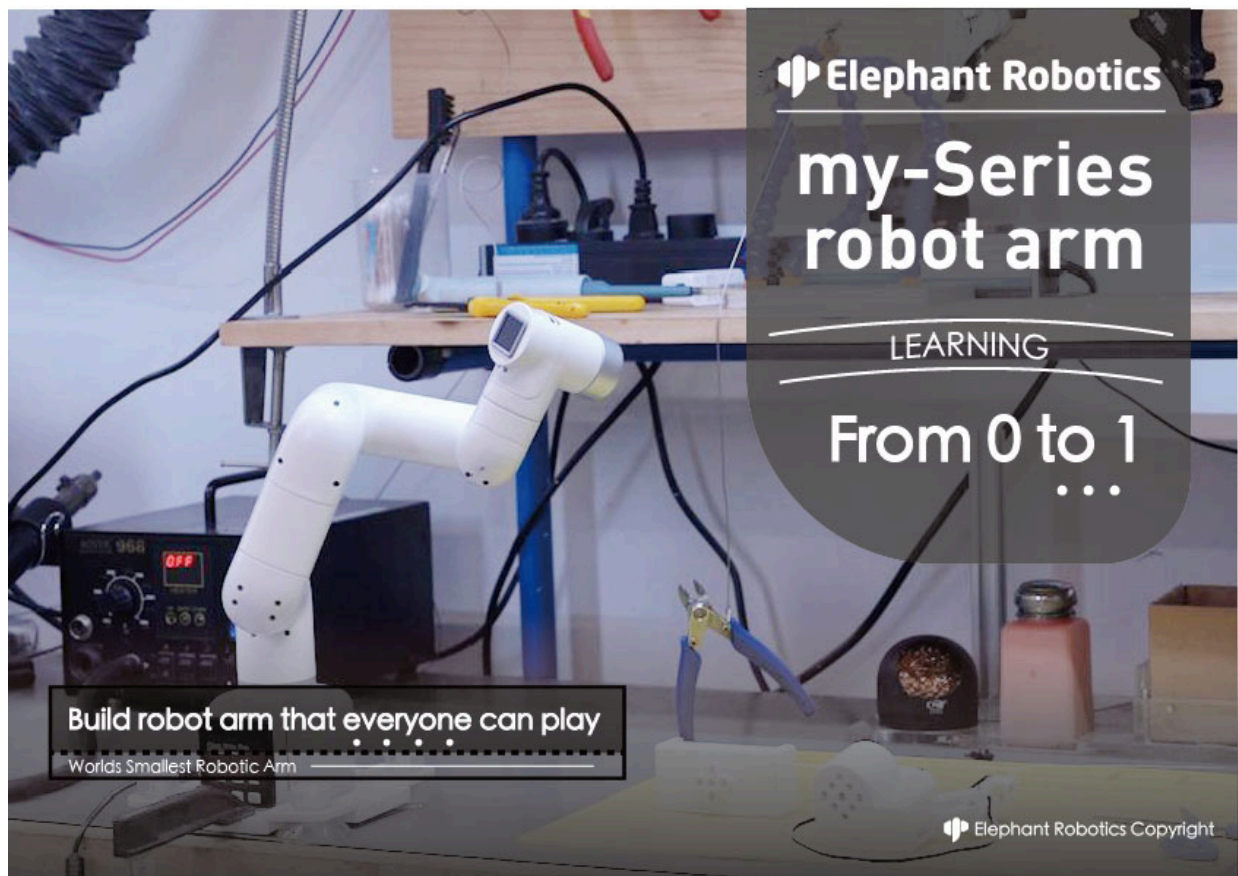
1 Elephant Robotics	1.1
1 Company Introduction	1.1.1
2 Development History	1.1.2
3 How to Read	1.1.3
2 Robot Kits	1.2
2.1 Artificial Intelligence Kit 2023	1.2.1
1 Adaptive robot	1.2.1.1
2 SetUp	1.2.1.2
Suit build	1.2.1.2.1
Environment build	1.2.1.2.2
3 Knowledge preparation	1.2.1.3
4 Visualization software	1.2.1.4
5 Color recognition	1.2.1.5
myCobot 280	1.2.1.5.1
pure Python	1.2.1.5.1.1
Combined with ROS	1.2.1.5.1.2
mechArm 270	1.2.1.5.2
myPalletizer 260	1.2.1.5.3
6 Shape recognition	1.2.1.6
myCobot 280	1.2.1.6.1
pure Python	1.2.1.6.1.1
Combined with ROS	1.2.1.6.1.2
mechArm 270	1.2.1.6.2
myPalletizer 260	1.2.1.6.3
7 AR QR code recognition	1.2.1.7
myCobot 280	1.2.1.7.1
pure Python	1.2.1.7.1.1
Combined with ROS	1.2.1.7.1.2
mechArm 270	1.2.1.7.2
myPalletizer 260	1.2.1.7.3
8 Feature point recognition	1.2.1.8
myCobot 280	1.2.1.8.1
pure Python	1.2.1.8.1.1
Combined with ROS	1.2.1.8.1.2
mechArm 270	1.2.1.8.2
myPalletizer 260	1.2.1.8.3

9 yolov5 recognition	1.2.1.9
mycobot 280	1.2.1.9.1
mechArm 270	1.2.1.9.2
myPalletizer 260	1.2.1.9.3
3 FAQ	1.3
1 How to Ask Questions Gracefully	1.3.1
2 Driver-related	1.3.2
3 Software	1.3.3
4 Hardware	1.3.4

Preparations

4 Background Knowledge	2.1
4.1 Robot Arm	2.1.1
4.2 Electronic Background Knowledge	2.1.2
4.3 Knowledge of Motor and Servo	2.1.3
4.4 Robot Kinematics	2.1.4
1 myCobot Series 6-Axis Collaborative Robotic Arm	2.1.4.1
5 Basic Usage	2.2
5.1 myStudio	2.2.1
1 Environment Building	2.2.1.1
2 Burning and Updating Firmwares	2.2.1.2
6 Comment & Feedback	2.3

myCobot: From 0 to 1



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-axis series 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 as bases, grippers, suction pumps and so on
- **Machine Vision** -- learn to control myCobot under the guidance of machine vision
- **Robot Modification** -- learn how to modify 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>
 - [Download PDF](#)
-

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
-

Elephant Robotics



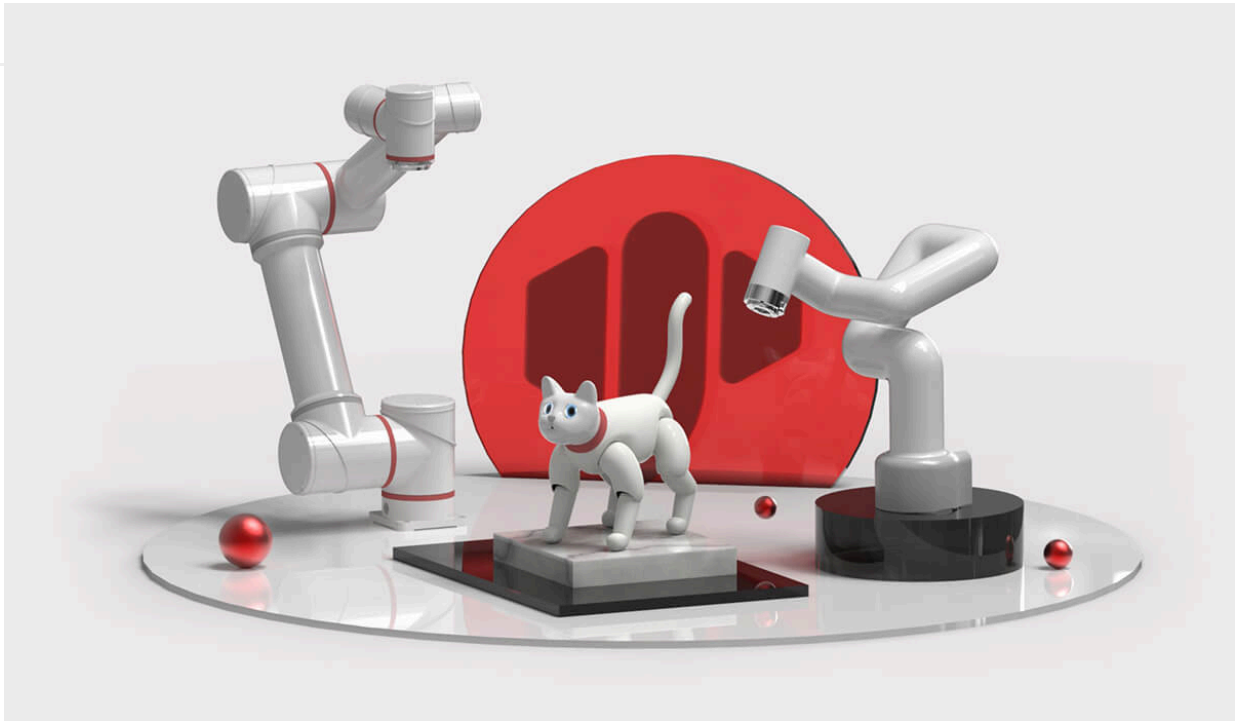
1 Company Introduction

Elephant Robotics is based in Shenzhen, China, a high-tech company which focuses on the design, research and development of robots and automation solutions.

We are devoted to providing highly flexible robots, easy-to-learn operating systems and intelligent automation solutions for robot education, scientific research institutions, business situations and industrial production. Our product quality and intelligent solutions have obtained unanimous acceptance and favorable comments from a number of world top 500 enterprises & factories in South Korea, Japan, America, Germany, Italy, Greece, etc.

Abiding by the vision of "enjoy robots world, Elephant Robotics initiates collaborative work between human and robots to make robots be good life and work helpers for human so as to free people from simple, repeated and dull jobs and give full play to the advantages of human-robot coordination, thus improving work efficiency and helping human to create a nice, new life.

In the future, Elephant Robotics hopes to promote the development of the robot industry through a new generation of cutting-edge technology, and starts a new era of automation and intelligence with its customers hand in hand.



2 Development History

In August 2016, Elephant Robotics was established.

In August 2016, entered HAX Incubator and obtained SOSV seed round investment.

In July 2017, the two founders were included in Forbes Asia's "30 Business Elites under Age 30".

In October 2017, published the fifth generation of single-arm industrial cobot called Elephant S.

In April 2018, obtained angel round investment from Cloud Angel Fund.

In June 2018, was awarded "Intelligent Manufacture Entrepreneurship MBA Award" by CKGSB.

In June 2018, was awarded "Startup Accelerator X-elerator Award" operated by Tsinghua University.

In November 2018, won the second place in the Asian Smart Hardware Competition in Shenzhen Division

In November 2018, obtained the "Most Invested Company Award" in GaogongGold Globe Award.

In March 2019, obtained the "Leading Person Award" in Gaogong Gold Globe Award.

In April 2019, obtained Catbot "Industrial Robot Innovation Award".

In September 2019, attended Huawei European Eco-Conference (HCE) and became a member of Huawei eco-partners.

In November 2019, Elephant Robotics attended the IROS International Conference on Intelligent Robots and Systems jointly with Harbin Institute of Technology.

In December 2019, obtained "Gaogong 2019 Innovation Technology Award".

In December 2019, was awarded as one of the Gaogong 2019 Top 10 Fast Growing Enterprises.

In December 2019, was awarded the "Emerging Enterprise Award" in the industrial robotics segment field of Shenzhen equipment industry.

In December 2019, launched the first type of bionic robotic cat called MarsCat in the world.

In May 2020, the founders obtained \"Shenzhen Robot Emerging Talent Award\" in 2019.

In October 2020, launched the smallest six-axis cobot named myCobot in the world.

In March 2020, launched the smallest cobot named myCobotPro 320 for scientific research in the world.

In May 2021, the Mars bionic cat named MarsCat was reported by several media such as Xinhua Finance, China Daily, Nanjing Daily, Harbin Daily, etc.

In July 2021, published the seat for the smallest hybrid robot, a baby elephant moving robot called myAGV.

In September 2021, launched the world's first type of fully wrapped four-axis robot arm, a tiny elephant palletizing robot arm called myPalletizer.

3 Related Links

- Official website: <https://www.elephantrobotics.com>
- Purchase link
 - shopify: <https://shop.elephantrobotics.com/>
- Video
 - bilibili: <https://space.bilibili.com/2126215657>
 - youtube: <https://www.youtube.com/c/Elephantrobotics>

4 Contact Us

If you have any other problems, contact us via the ways below. + Email: We will give a reply within 1-2 business days; **Email + WeChat**: We provide one-to-one service only for those users who have purchased myCobot via WeChat.

Development history of my series of products

Development History

In October 2020, we launched the smallest six-axis cobot named **myCobot 280-M5** in the world.

In December 2020, **myCobot 280** was put on the market.

In April 2021, **myCobot 320** products was put on the market.

In May 2021, we published the Raspberry Pi version of **myCobot 280**.

In June 2021, we published the Raspberry Pi version of **myCobot 320**.

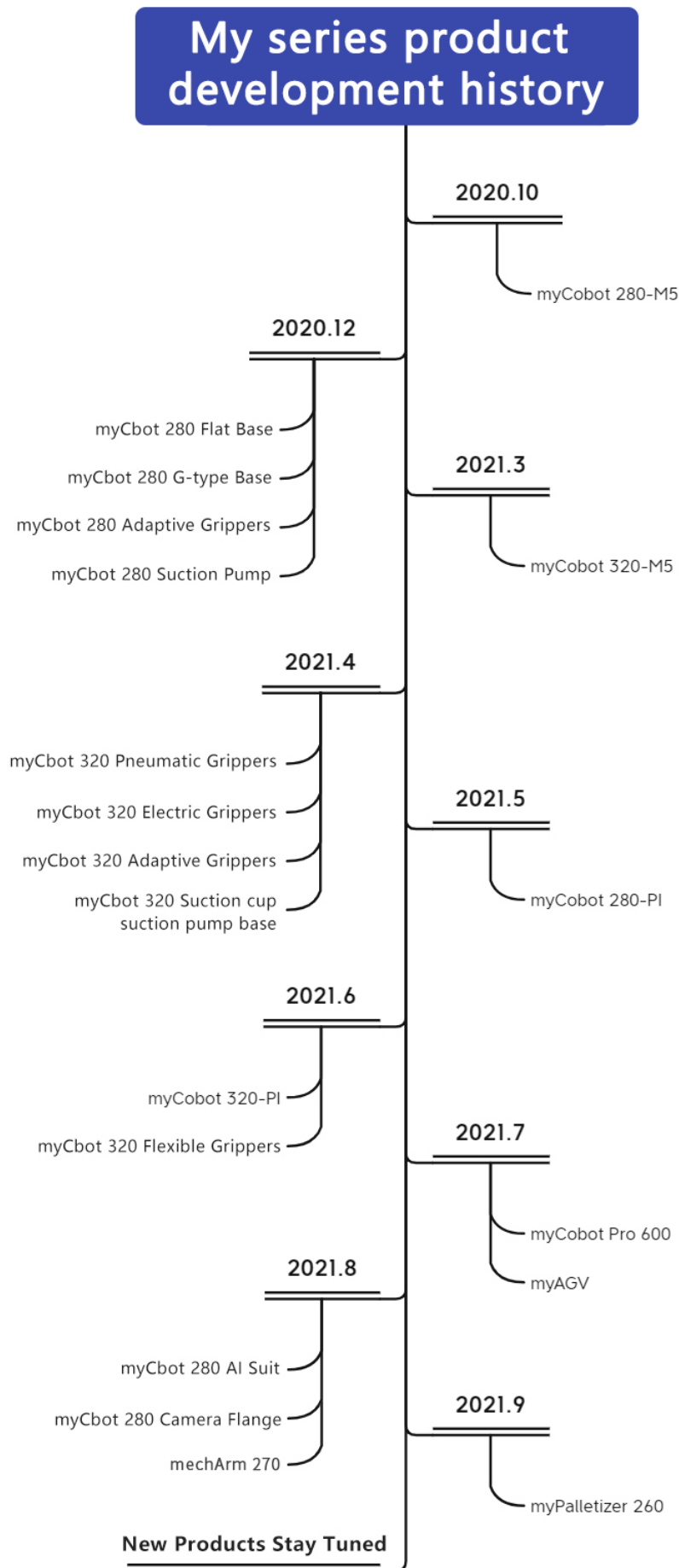
In July 2021, we published the commercial version of a babyelephant coordinative robot arm called **myCobot Pro 600**.

In July 2021, we published the smallest hybrid robot, a baby elephant moving robot called **myAGV**.

In August 2021, **AI kits** and the computer vision enabled for visual machines were put on the market.

In August 2021, The world's most compact and portable small six-axis mechanical **mechArm 270** series came out

In September 2021, launched the world's first type of fully wrapped four-axis robot arm, a tiny palletizing robot arm called **myPalletizer**.





The book is designed to help you reach the following targets.

- Understand the basic use of mechanisms, electronics and software related to robot arms.
- Understand the basic principle, joints, coordinates, terms, control and other information of the robot arm. Able to do simple forward and inverse kinematics calculations.
- Understand the basic operations of API-controlled robot arms and the robot arms controlled by myBlockly graphical programming language.

- Understand the image recognition algorithms related to machine vision.
- Understand the building of robot vision scenes and the methods and strategies of coordination between vision and robot arm.
- Grasp all skills of artificial intelligence (AI) packages.

You should read the book based on your background. We divide the background and related knowledge into three degrees:

Degree	Background	Skills	Estimated time of learning	Recommended platform
Beginner	Specialty related to information, electronics and automation	Understand a kind of programming language and the basic knowledge related to electronics	100 hours	myBlockly
Advanced	Understand Arduino or similar hardwares, servo and programming, IO interface, etc	Able to debug API and interfaces, and understand communication	50 hours	Arduino
Professional	The readers used at least one industrial or consumer robot arms, and have the ability to develop hardware and software	Understand the Cartesian coordinate system, joint control, and basic use of robots	30 hours	Random

3 Learning Steps and Time

No.	Target knowledge points	Theory	Practice	Estimated hours of learning
1	Quick unpacking	1 Drag teaching	1. The accessories for myCobot 2. drive the robot to perform drag teaching	1 hour
2	Background knowledge learning	1. Use background of industrial robots; 2. coordinate and space learning, and Cartesian 3D coordinate and rotation, xyz; 3. joint and coordinate control of industrial robots	1. Joint control and recurrence of robot joints 2. speed control of robots 3. control and cycle of robot coordinate points	5 hours
3	Hardware learning	1. Principles and operations of embedded electronics 2. the principle and knowledge of servo and motor 3. actuator learning	1. Basic/atom control and driving 2. driving and motion of servo 3. robot accessory learning	5 hours
4	Software and firmware and their updating	1. Identify different software platforms and their use 2. firmware loading and adaptation principle	1. Select the developing platform suitable for you 2. load and update the corresponding firmware.	2 hours

No.	Target knowledge points	Theory	Practice	Estimated hours of learning
5	Building of software development environment	1. Build Arduino platform 2. load and update of Arduino library file 3. understand serial communication	1. Familiar with Arduino platform 2. load a library 3. operate and run the first line of codes	2 hours
6	Learning and development of robot library	1. Basic communication and operation types of robots 2. common operating methods of robots 3. control of direction and coordinate modes	1. Communicate with the robot 2. control the robot to move 3. operate the IO interface, gripper, etc. of the robot;	5 hours
7	myBlockly operation robot arm	1. Understand the basic architecture and relation of graphical programming language interfaces: sensor, actuator and procedure 2. variable, cycle and judgment 3. control method of robot arm	1. Display different fonts in the basic 2. make the robot arm move to different positions using three buttons of the basic 3. control the robot arm to make it move to several positions circularly	10 hours

No.	Target knowledge points	Theory	Practice	Estimated hours of learning
8	The use of roboFlow	1. Learn the industrial operating systems commonly used for robots 2. learn the common modules for roboFlow: point, quick movement, IO control and output 3. learn the advanced modules of roboFlow: cycle, judgment, and pallet program	1. Control the movement of the robot arm 2. basic control of IO input and output 3. cycle control and judgment	5 hours
9	Algorithms related to image recognition	1. Common color recognition methods and strategies 2. common shape recognition methods and strategies 3. common area recognition methods and strategies	1. Building of a ROS environment 2. reading of different colors 3. recognition of different shapes	20 hours
10	Vision and the joint debugging of the robot	1. Connect the world with a camera coordinate system 2. QR code image calibration 3. movement and correction	1. Operate the robot arm to the camera coordinate system 2. the robot arm moves in the camera coordinate system 3. recalibrate and set	10 hours

No.	Target knowledge points	Theory	Practice	Estimated hours of learning
11	Artificial intelligence (AI) package	1. Flow chart learning and making 2. electrical connection diagram learning and making 3. operation strategies such as image recognition and classification, etc	1. Sensor connection 2. gripper actuator connection and driving 3. robot arm driving and visual joint debugging	20 hours

4 Additional problems

If the above learning contents cannot meet your actual use needs, you can contact **Elephant Robotics helper** for further communication. We provide customized services for software and hardware, and the service fee is based on the **actual cost**.

Advanced Package

Take lead in the frontier of science and technology, and create practical packages, including core packages such as machine vision, composite robots, etc., and enlightens the user's creative thinking to make him understand the open source innovation culture.

With good expansibility and high openness, the package can be used for multiple applications. It can be easily used for the practice platforms of colleges, robotics building, robot laboratories and personal study.

Package Type:

AI Kit 2023

- Artificial intelligence Kit 2023 version is a new upgrade, learn the control principle of four-axis and six-axis collaborative robots, and realize fully automated operations from identification to sorting.

myCobot 320 AI Kit 2023

- The myCobot 320 AI Kit 2023 Edition is a Python-based platform that allows control of the robotic arm through software development. It enables fully automated operations, from recognition to sorting, making it suitable for a wide range of applications.

AI Kit 3D

- The AI Kit 3D Edition is equipped with a 3D camera for three-dimensional recognition and positioning. It eliminates the need for complex position calibration, making the setup process hassle-free. The kit offers four recognition algorithms and two major applications. The open-source Python-based code allows developers to control the robotic arm through the development software, enabling full automation from recognition to sorting tasks. With this kit, you can achieve seamless automation in your operations, from object recognition to automated sorting.

AI Kit

- A robot is guided through machine vision, and intelligent gripper can be realized at the same time.

Industry 4.0 Package.

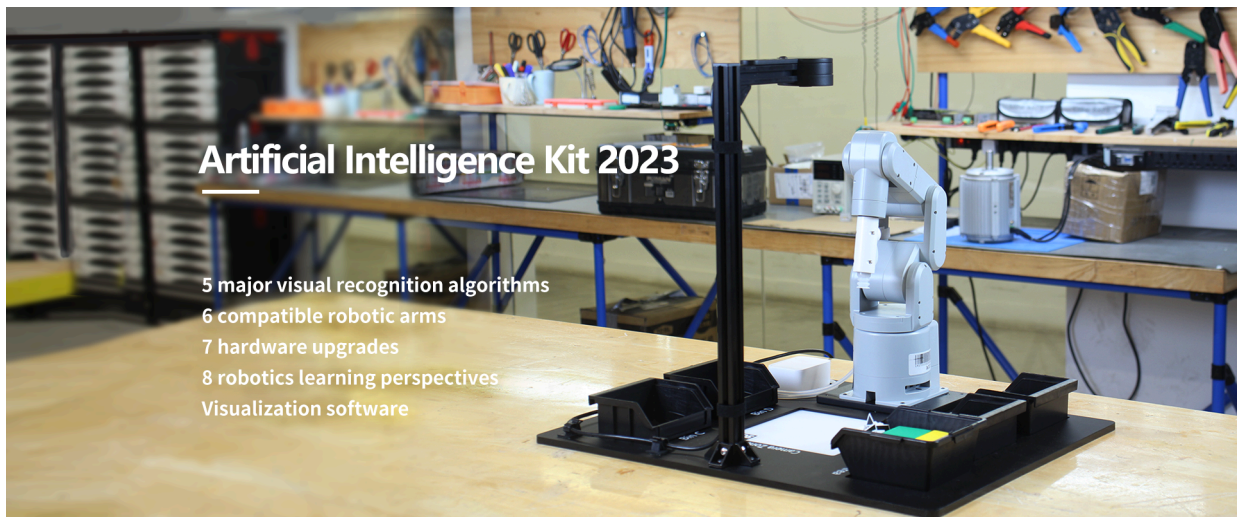
- A robot is guided through machine vision, and intelligent gripper can be realized at the same time. (Upgraded version of AI package)

Mobile Hybrid Robot Package

- The mobile hybrid robot allows your workspace to be expanded sufficiently to enable it to complete more tasks.

Smart Warehousing Palletizing Package. It is going to be put on the market soon.

Artificial Intelligence Kit 2023

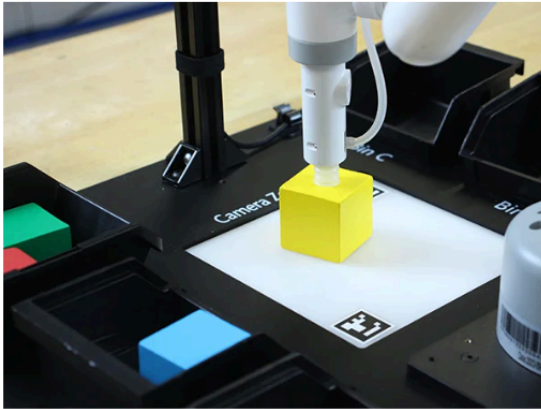


1 Product introduction

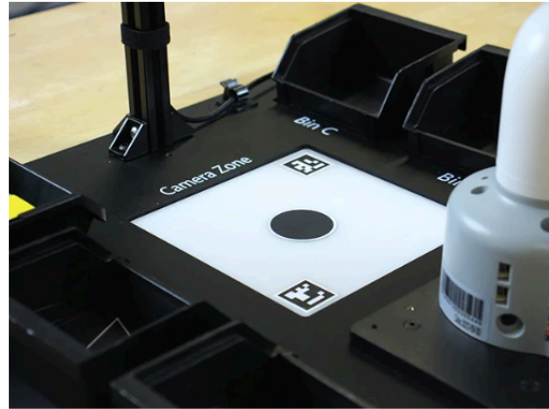
- The Artificial Intelligence Kit 2023 is newly upgraded with 5 major visual recognition algorithms and 7 hardware updates, supporting 6 robotic arms of Elephant Robotics. Users can learn major robotics knowledge from 8 perspectives and use visualization software with this kit. It can achieve positioning, grabbing, and automatic sorting. Based on the Python platform, the control of the robotic arm can be realized through the software. Users can learn the basics of artificial intelligence, inspire innovative creation, and enjoy the open source creative culture with AI Kit 2023.
- The Artificial Intelligence Kit 2023 has good expandability and high openness, and can be used for multiple purposes. It can be easily used for higher education training platforms, robotics study and research, or personal learning and project development.

2 Product performance:

- 5 major visual recognition algorithms: AI kit has 5 built-in main visual recognition algorithms and users can use different colored wooden blocks and different shaped cards for the recognition function. Also, users can use 4 types of QR CODE recognition to learn the relationship between 2D and 3D. Feature point recognition helps users learn to understand image segmentation and features and learn YOLOv5 algorithm quickly.



Color recognition



Shape recognition



Feature point recognition



YOLOv5 recognition

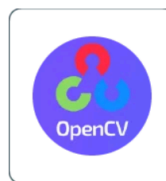


AR code recognition

- 6 compatible robotic arms: It is compatible with 6 robotic arms of Elephant Robotics, supports M5Stack and Raspberry Pi versions.
- 8 robotics learning perspectives:

Vision education kit What can you learn?

- OpenCV
- Python
- ROS
- Inverse kinematics
- Hand-eye calibration
- Robot vision
- Robot control principle
- End-effector use



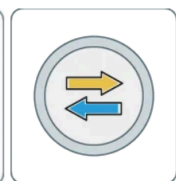
OpenCV



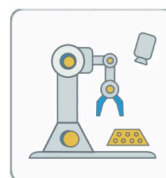
Python



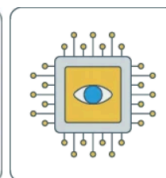
Ros



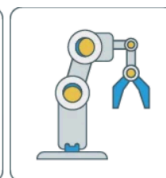
Inverse kinematics



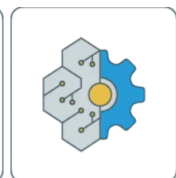
Hand-eye calibration



Robot vision



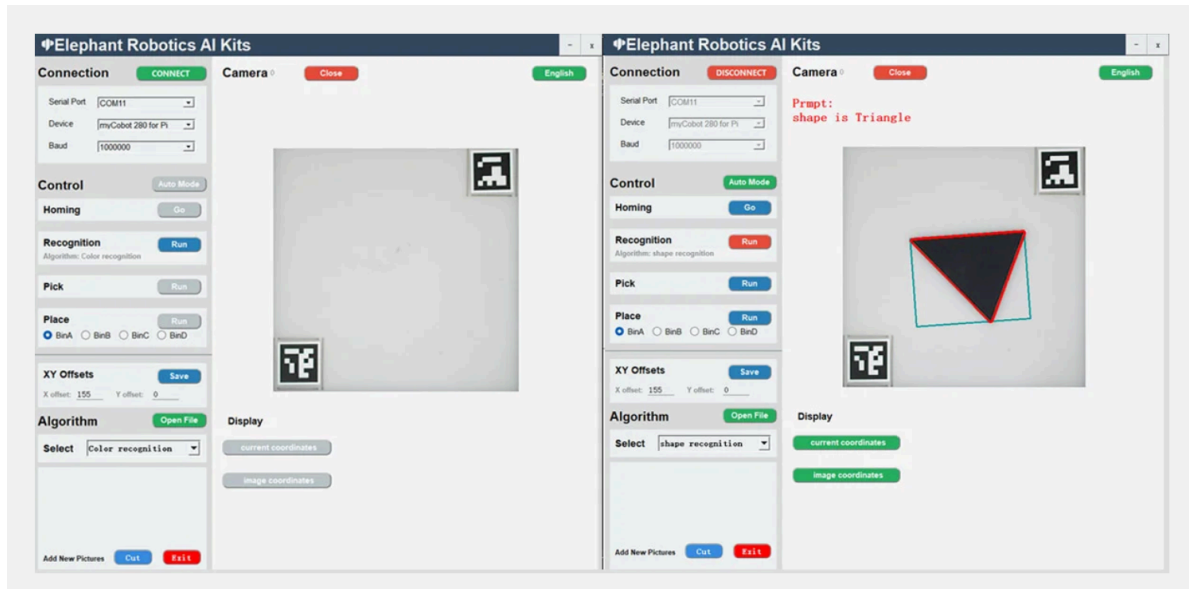
Robot control principle



End-effector use

1 Adaptive robot

- Visualization software: The Artificial Intelligence Kit 2023 supports visual software operation, provides customized and simple operation methods. Users can quickly start the artificial intelligence learning journey.



3 Product parameters

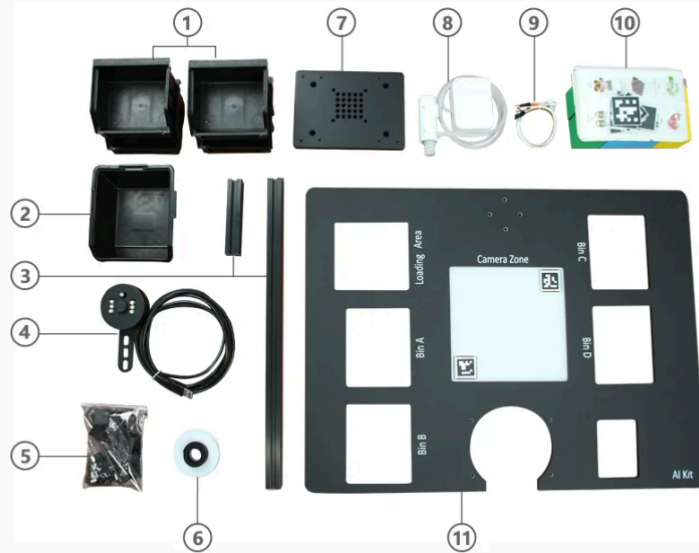
Index	Parameter
Product Name	AI Kit 2023
Total measurement	500X400X405mm
Recognition speed	600msColor/shape/QR code: 300ms Feature points: 600ms
Recognition accuracy	3mm
Camera name	USB distortion free camera
Image pixels	2M 1080p
Supported image formats	MJPEG/YUY2
Pixel size	3.0umx3.0um
Maximum frame rate	MJPEG: 1920 1080@30fps YUV: 1920 1080@30fps
USB protocol	USB2.0 HS/FS
Supported Resolutions	1280X720,640X480,320X240
Power supply	DC5V 90mA
Field of view	110° no distortion
Supported system	Windows XP/7/8/10/VISTA/SEVEN/MAC LINUX(include uvc)/Raspberry Pi/Android 4.2+

4 Product System Composition Structure

- Hardware composition

KIT COMPONENTS

- ★ 1. Small bin x4
- ★ 2. Big bin
- ★ 3. Camera extrusion stand x2
- ★ 4. Camera
- ☆ 5. Accessories pack
- ☆ 6. Black and white velcro
- ★ 7. Arm base
- ★ 8. Suction pump
- ☆ 9. Suction pump connection wire
- ☆ 10. Image cards & Wooden blocks
- ★ 11. Acrylic plate
Physical Star: Represents a hardware upgrade



Adaptive Robots

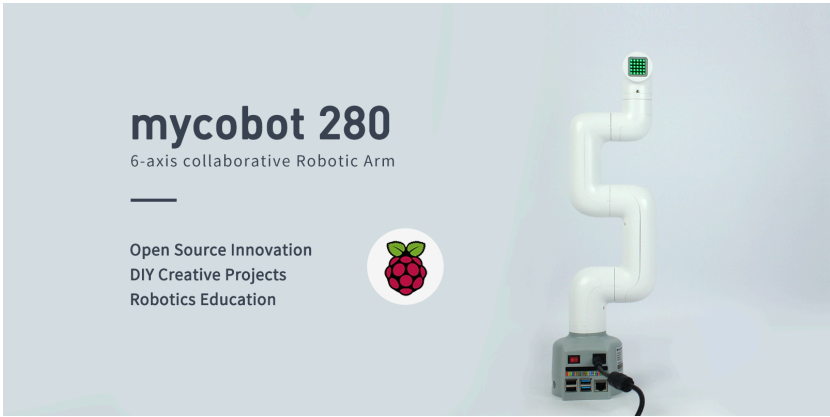
myPallitizer 260

Product	Image	Link
myPalletizer 260 M5	 The image shows the myPalletizer 260 M5 robotic arm, a white industrial robot with a blue logo and text 'myPalletizer Elephant palletizing robotic arm' on the left. The robot is positioned on a grey base with a small screen.	Introduction
myPalletizer 260 PI	 The image shows the myPalletizer 260 PI robotic arm, a white industrial robot with a blue logo and text 'myPalletizer Raspberry Pi Ecology' on the left. The robot is positioned on a grey base with a small screen.	Introduction

mechArm 270

Product	Image	Link
mechArm 270 M5	 The image shows the mechArm 270-M5, a compact, grey, 6-axis collaborative robot arm. It is mounted on a dark blue base which features a small screen displaying the robot's status. The text "mechArm 270-M5" and "The World's Smallest Collaborative Robot" is overlaid on the image.	Introduction
mechArm 270 PI	 The image shows the mechArm 270-Pi, a compact, white, 6-axis collaborative robot arm. It is mounted on a grey base. A Raspberry Pi logo is visible in the bottom left corner of the image. The text "mechArm 270-Pi" and "The most compact 6-axis robot arm" is overlaid on the image.	Introduction

myCobot 280

Product	Image	Link
myCobot 280 M5	 mycobot 280 6-axis collaborative Robotic Arm Open Source Innovation DIY Creative Projects Robotics Education The image shows a white 6-axis collaborative robotic arm (myCobot 280 M5) standing on a base. To the left of the arm, there is a circular logo with a blue cube icon. The background is a light blue gradient.	Introduction
myCobot 280 PI	 mycobot 280 6-axis collaborative Robotic Arm Open Source Innovation DIY Creative Projects Robotics Education The image shows a white 6-axis collaborative robotic arm (myCobot 280 PI) standing on a base. To the left of the arm, there is a circular logo with a red Raspberry Pi icon. The background is a light blue gradient.	Introduction

myArm 300 Pi

Product	Image	Link
myArm 300 PI	 myArm 300 PI 2023 如手臂般灵巧的7轴机器人 7 轴入门首选 100+ 编程控制接口 支持ROS1/2开发环境 The image shows a white 7-axis collaborative robotic arm (myArm 300 PI 2023) standing on a base. The background is a brick wall with some greenery on the sides.	Introduction

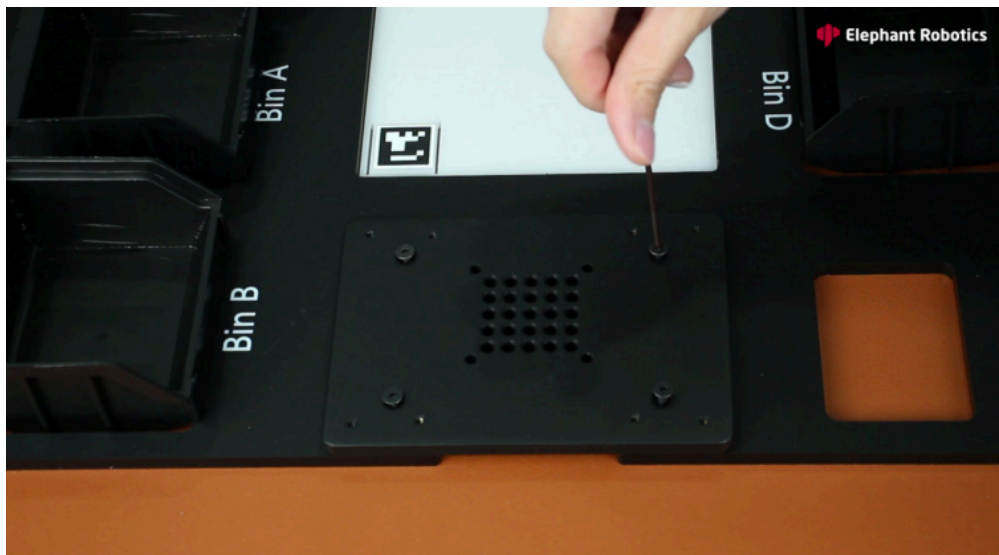
Suit building

Installation video



1 Install the base

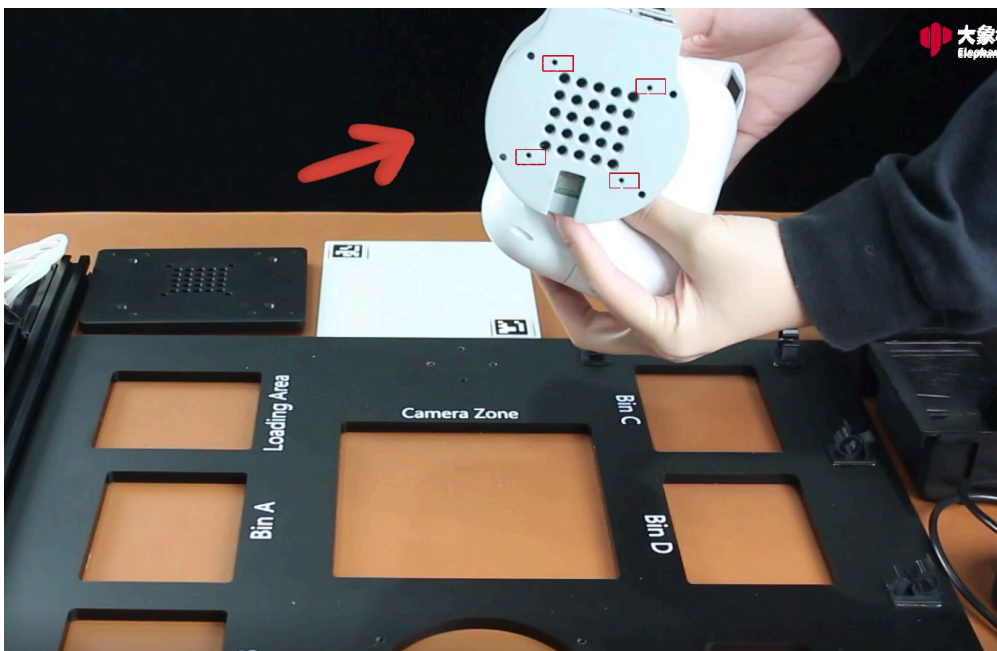
1.1 For the mechArm M5/Pi version of the robotic arm, first install the base of the robotic arm with M4X14 countersunk head screws.



1.2 Align the robotic arm with the holes and fix it with M3X12 Phillips screws.

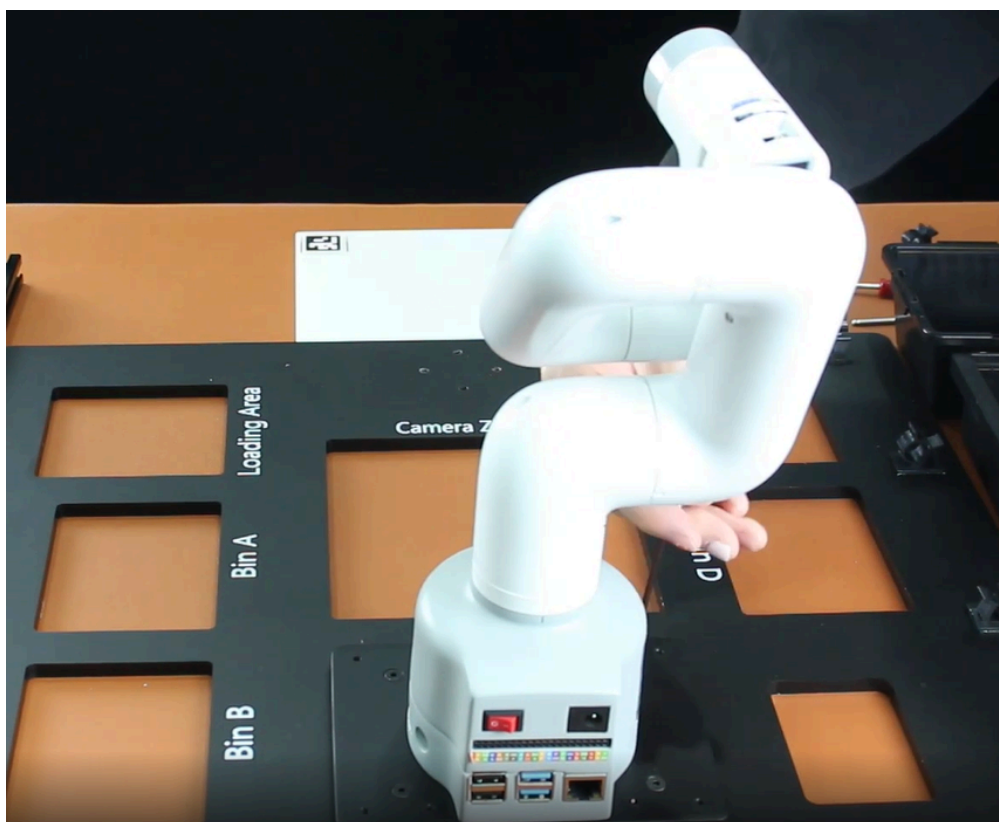


1.3 Other robotic arm installation tutorials: align the base of the robotic arm with the hole at the bottom of myCobot Pi, and fix it with M3X12 Phillips screws, or install Lego keys on the base of the robotic arm.



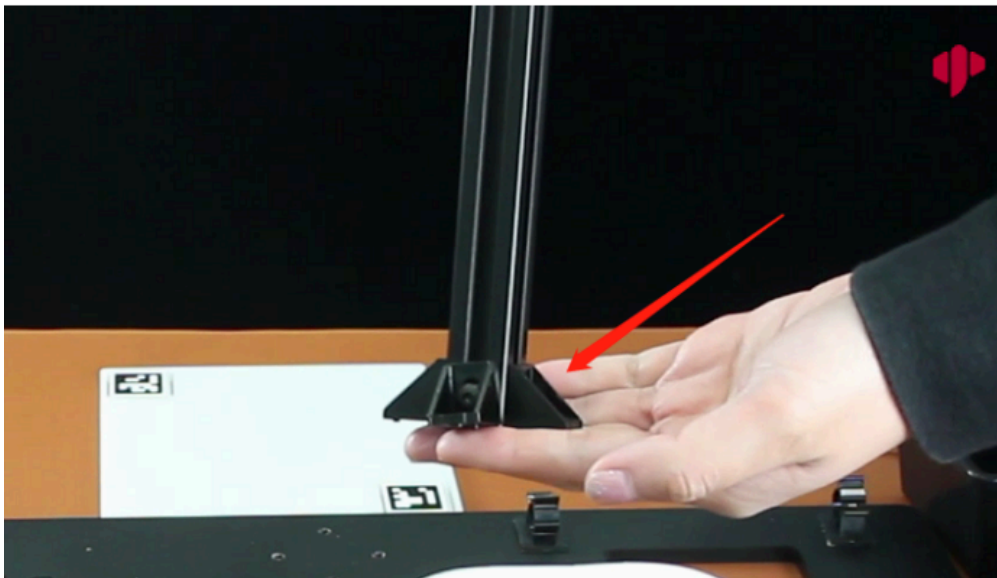
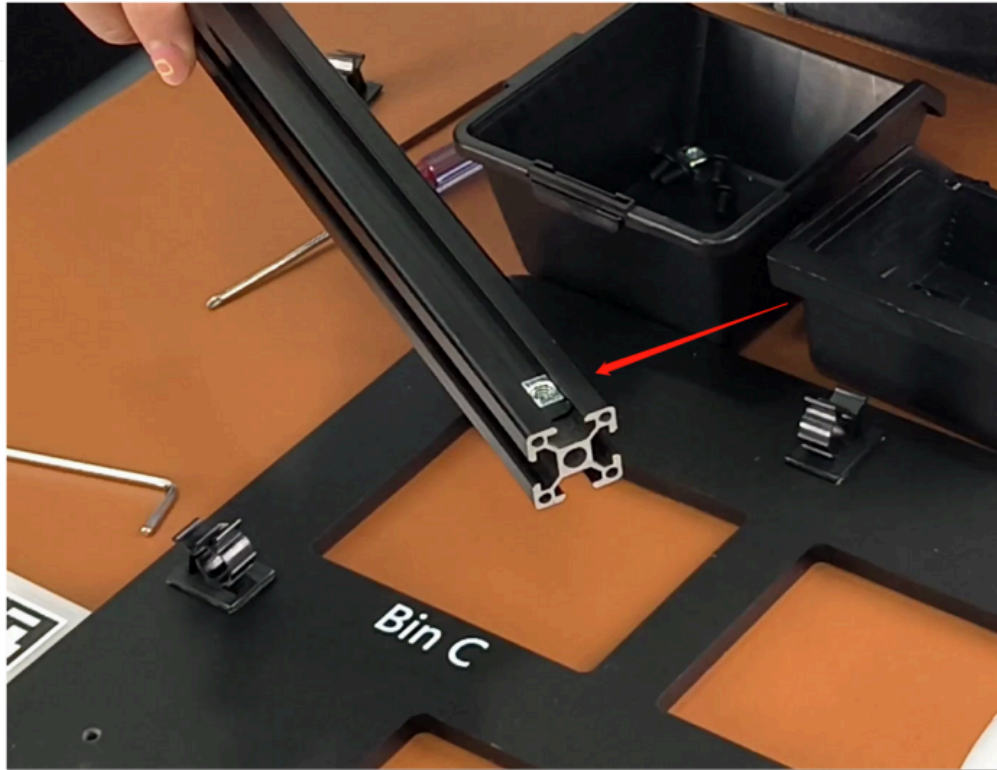


1.4 Put the installed machine on the acrylic bottom plate and fix it with M4X14 countersunk head hexagon socket head screws. Pay attention to simply fix the 4 screws first, and then tighten the screws one by one to avoid deviation of the hole position.



2 Install the camera bracket

2.1 Put the T-nut into the groove of the long profile, use the M5x8 pan head hexagon socket screw to install the corner code, and install 3 pieces in sequence.



2.2 Repeat the above to install the short profile corner code.

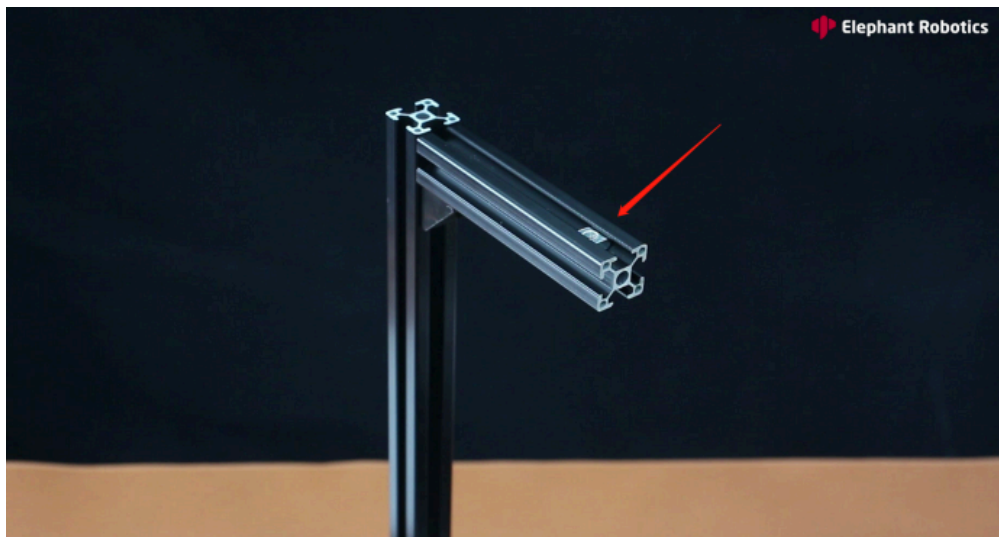
2.3 Put the T-nut into the groove of the profile, and install the short profile on the long profile with M5x8 pan head hexagon socket screws.

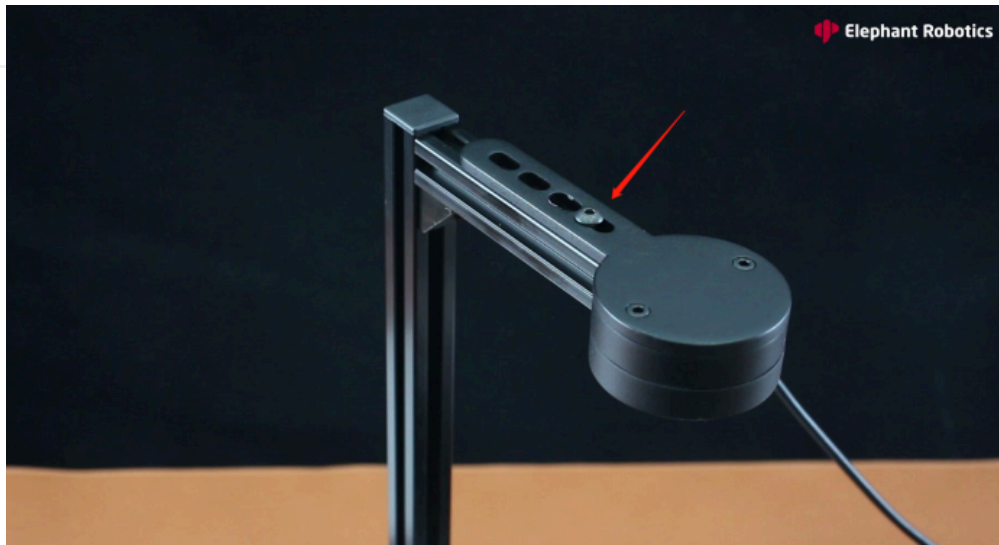


2.4 Align the assembled camera bracket with the holes on the acrylic board, and fix it with M5x10 pan head hexagon socket head screws. Pay attention to fix the three screws first, and then tighten the screws in sequence to avoid deviation of the holes.



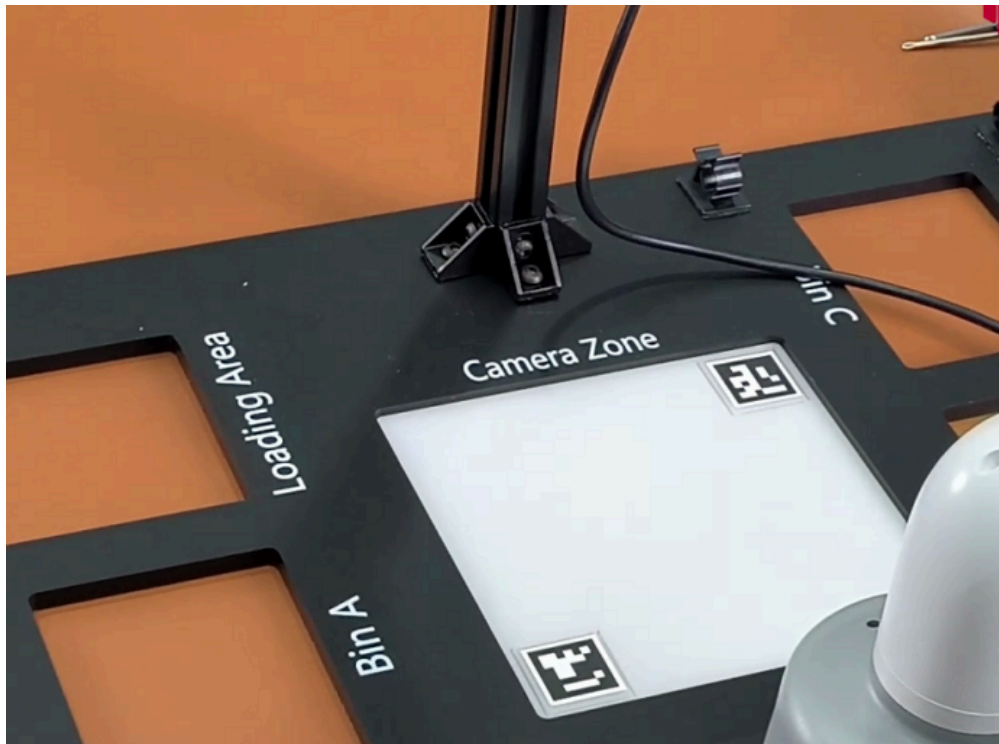
2.5 Put the T-nut first, and use the M5x10 pan head hexagon socket head screw to fix the camera, so it is installed.



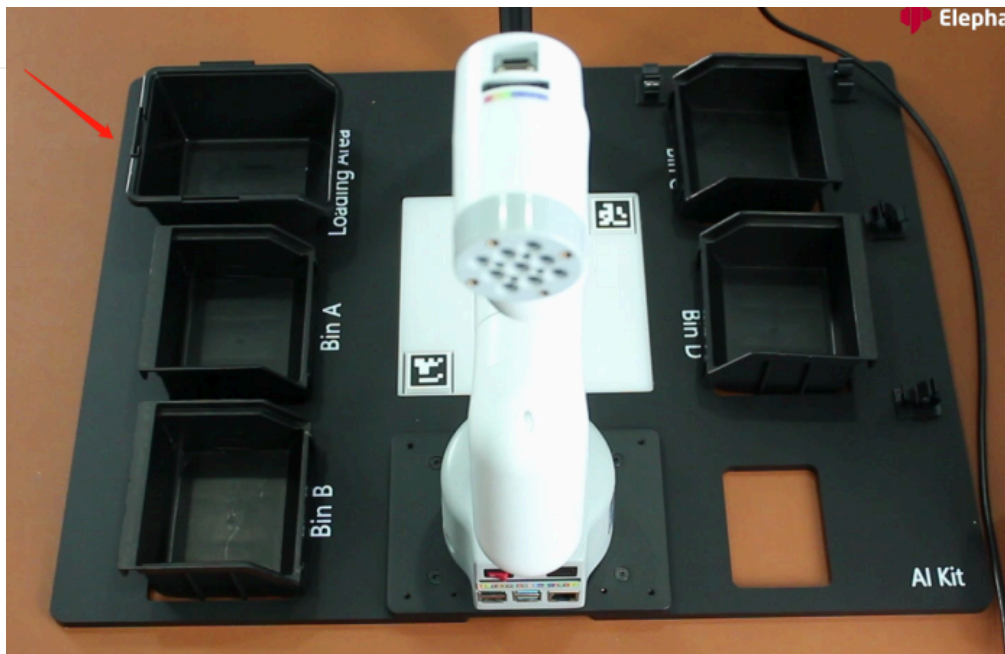


3 place parts

3.1 First tear off the yellow protective film, paste the QR code according to the video and place it.



3.2 Arrange large and small parts boxes.



4 Install the suction pump

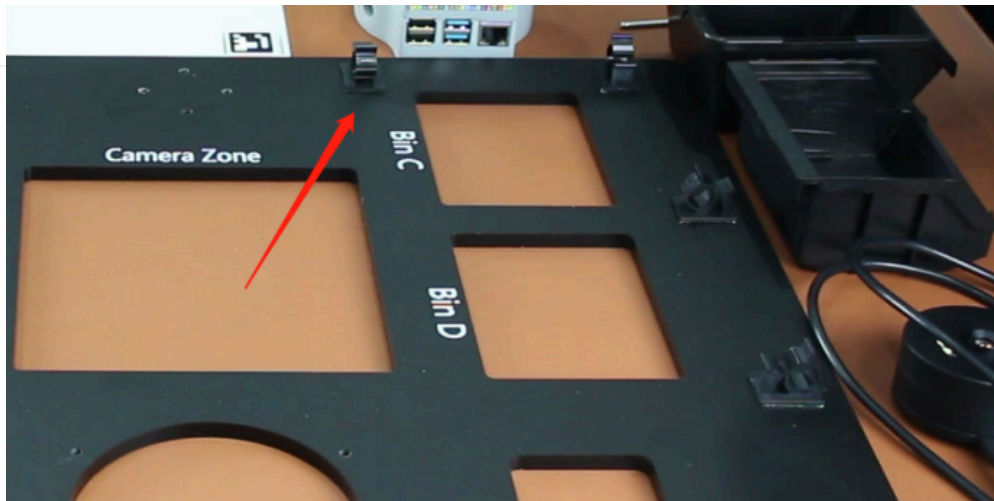
4.1 The suction pump is filled with Lego keys and installed at the end of the robotic arm.

4.2 The Grove port is connected to the suction pump. According to the pin definition, the other end is connected to GND, 5V, 21, 22 of the Raspberry Pi in turn; the M5 version is connected to GND, 5V, 2, 5.



5 fixed routing

5.1 The self-adhesive cable clip is pasted on the acrylic base plate according to the position of the camera cable.



5.2 Use the black velcro to fix the cable of the camera.

5.3 Put the wire into the wire clamp to fix it.

5.4 Use the white Velcro to fix the air tube of the suction pump.

5.5 Finally put the wooden blocks, the overall assembly is complete, and you can start using it.



1 Windows Environment Build

First of all, it needs to be clear that all case experiments are implemented based on python+opencv+pymycobot, so when using the M5 robotic arm, you need to use Mystudio to burn ATOM firmware and M5Stack-basic firmware on the head and bottom of the robotic arm respectively.

- **Usage environment:** Windows 10 or Windows 11, Python3+OpenCV+pymycobot, Git
- **Applicable devices:** mypalletizer 260 M5

1.1 Install Python+pymycobot

pymycobot is a Python package for serial communication with myCobot, supporting Python2, Python3.5 and later versions.

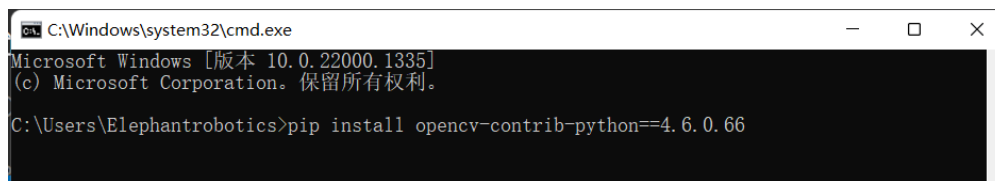
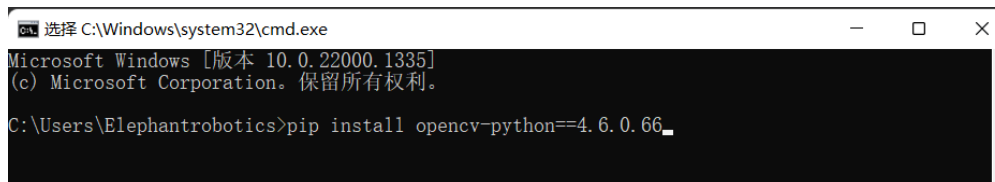
Before using pymycobot to control the robotic arm, you need to build a Python environment. For the specific building process, please refer to [Pyhton Environment Build](#) 章节。

1.2 Install OpenCV

OpenCV-Python is a Python-based library designed to solve computer vision problems. Before installation, make sure that the Python environment has been set up.

Open a console terminal (shortcut key Win+R, enter cmd to enter the terminal), enter the following command:

```
# The version numbers of the two need to be consistent, and version 4.6.0.66 is installed here
pip install opencv-python==4.6.0.66
pip install opencv-contrib-python==4.6.0.66
```



Install successfully, can check lib version and location with the command below

```
# to check python libs, format: pip show lib-name
pip show opencv-python
pip show opencv-contrib-python

# install pyqt5
pip install pyqy5
```

1 Adaptive robot

```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.22000.1335]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\Elephantrobotics>pip show opencv-python
Name: opencv-python
Version: 4.6.0.66
Summary: wrapper package for OpenCV python bindings.
Home-page: https://github.com/skvark/opencv-python
Author:
Author-email:
License: MIT
Location: c:\users\elephantrobotics\appdata\local\programs\python\python39\lib\site-package
Requires: numpy, numpy, numpy
Required-by:
```

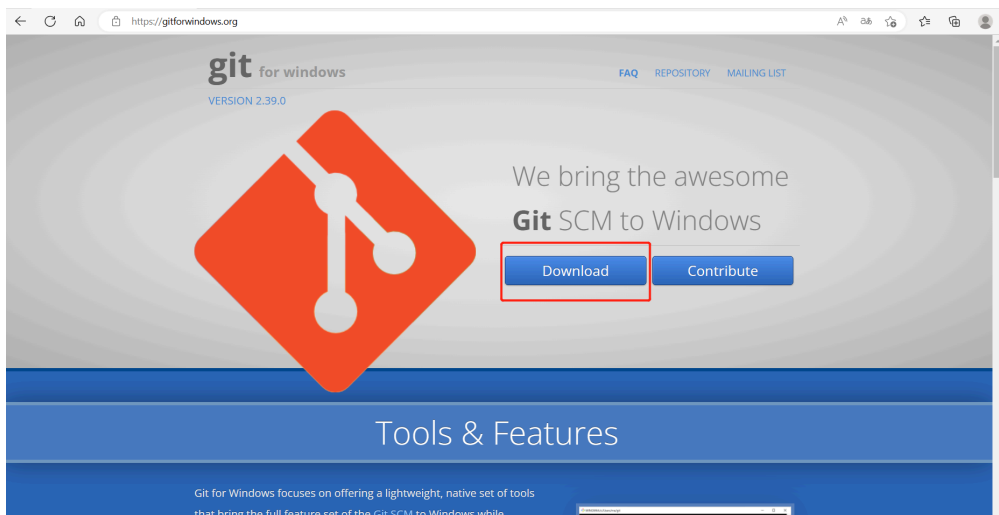
```
C:\Windows\system32\cmd.exe

C:\Users\Elephantrobotics>pip show opencv-contrib-python
Name: opencv-contrib-python
Version: 4.6.0.66
Summary: wrapper package for OpenCV python bindings.
Home-page: https://github.com/skvark/opencv-python
Author:
Author-email:
License: MIT
Location: c:\users\elephantrobotics\appdata\local\programs\python\python39\lib\site-package
Requires: numpy, numpy, numpy
Required-by:
```

1.3 Install Git

Git is an open source distributed version control system for effective and high-speed processing of project version management from small to very large.

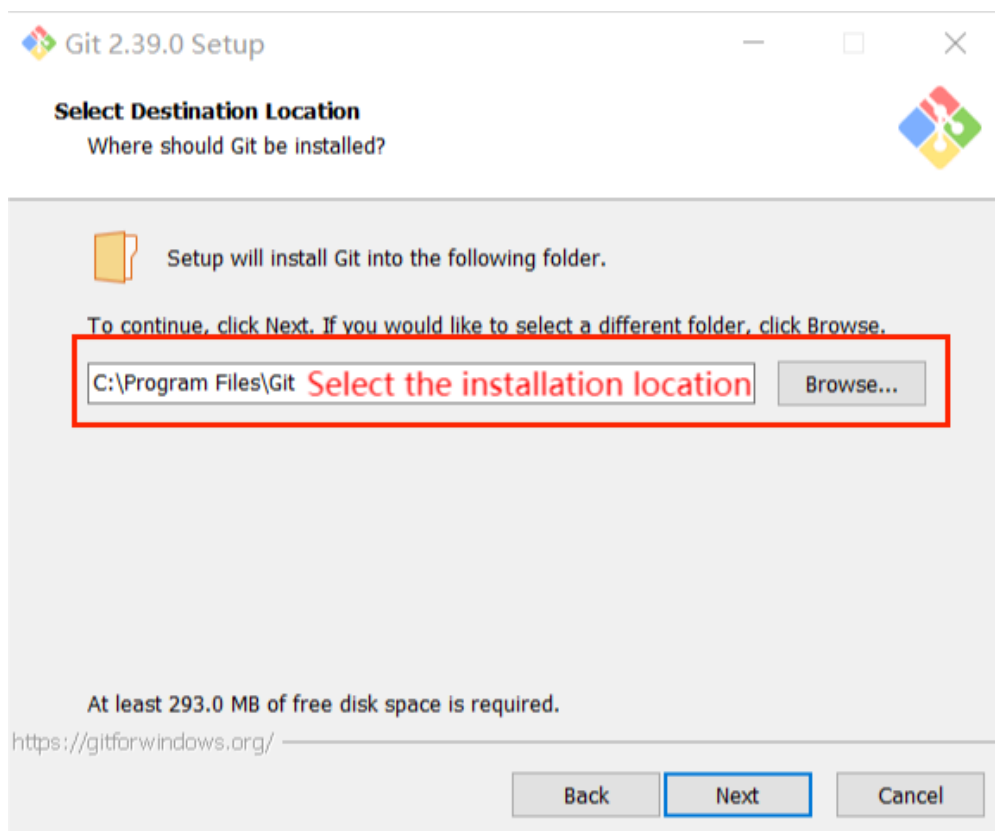
- **Git official download address:** <https://gitforwindows.org/>
- **Click the `Download` option to start downloading Git**



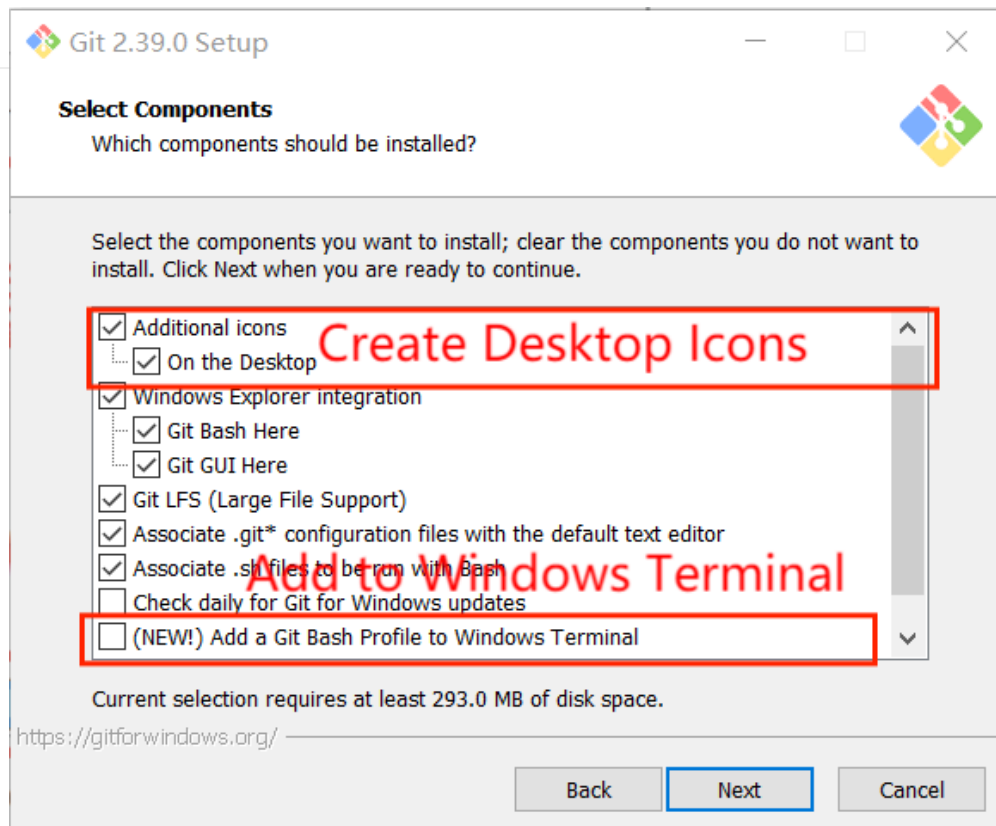
- **Double-click the downloaded exe file to install**



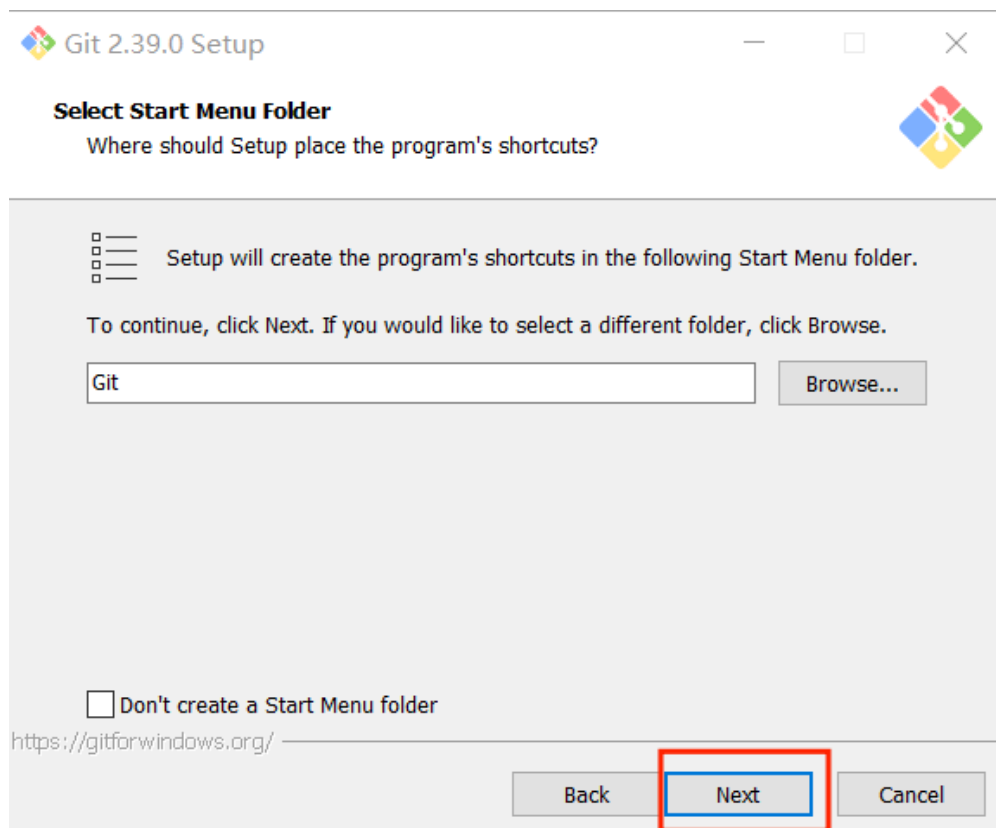
- Select the installation location, the default is under the C drive, if you don't want to put it under the C drive, you can change the location. After selecting the installation location, click next.



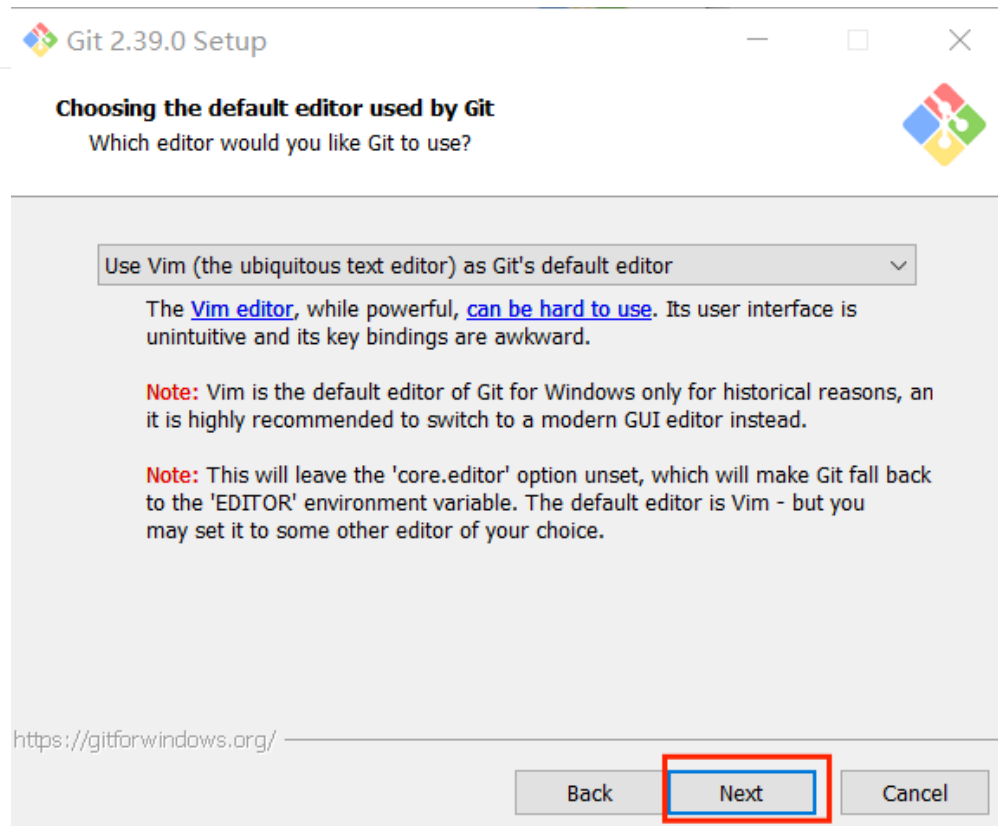
- By default, no icon will be created on the desktop, and a new feature has been added, adding GitBash to the Windows terminal, it is recommended to check it.



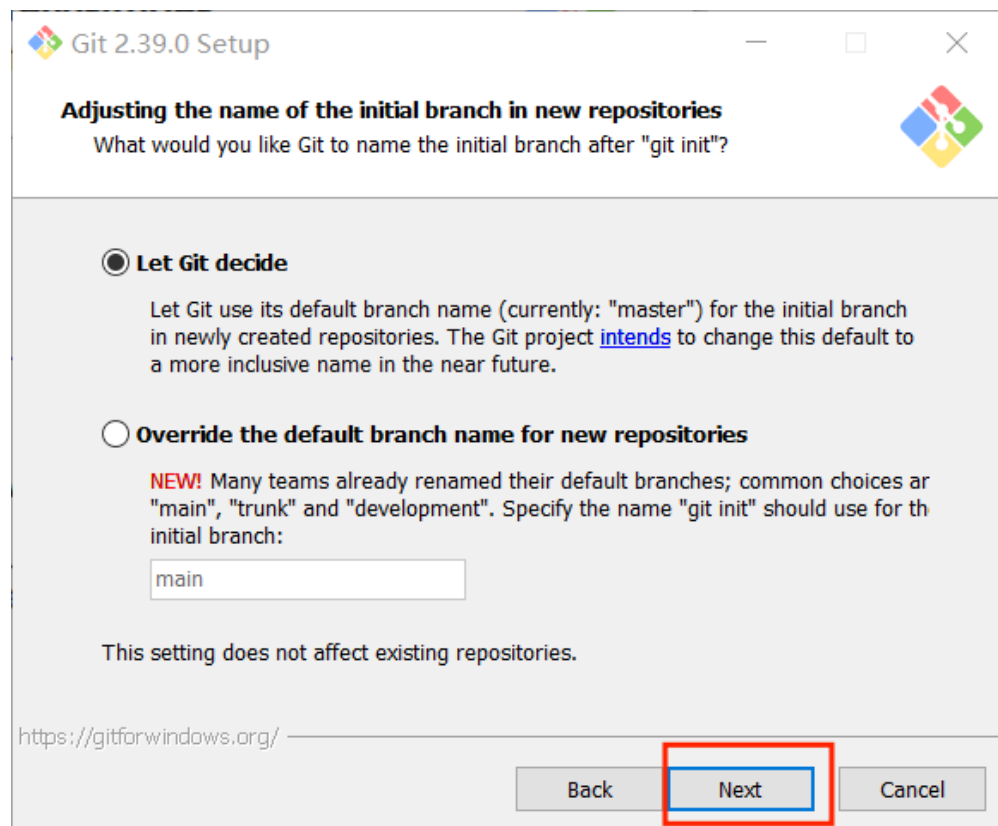
- The next step is the default, just select Next.



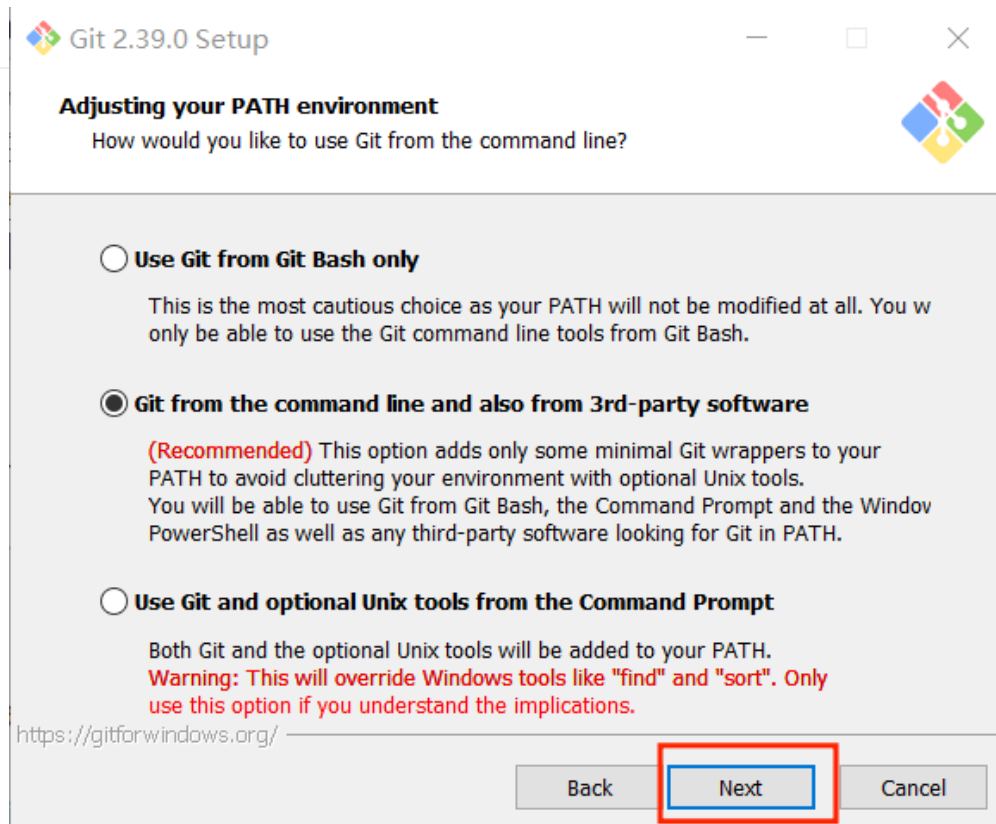
- Choose a default editor, the default is Vim (Linux), and there is Notepad inside, I choose to use Vim



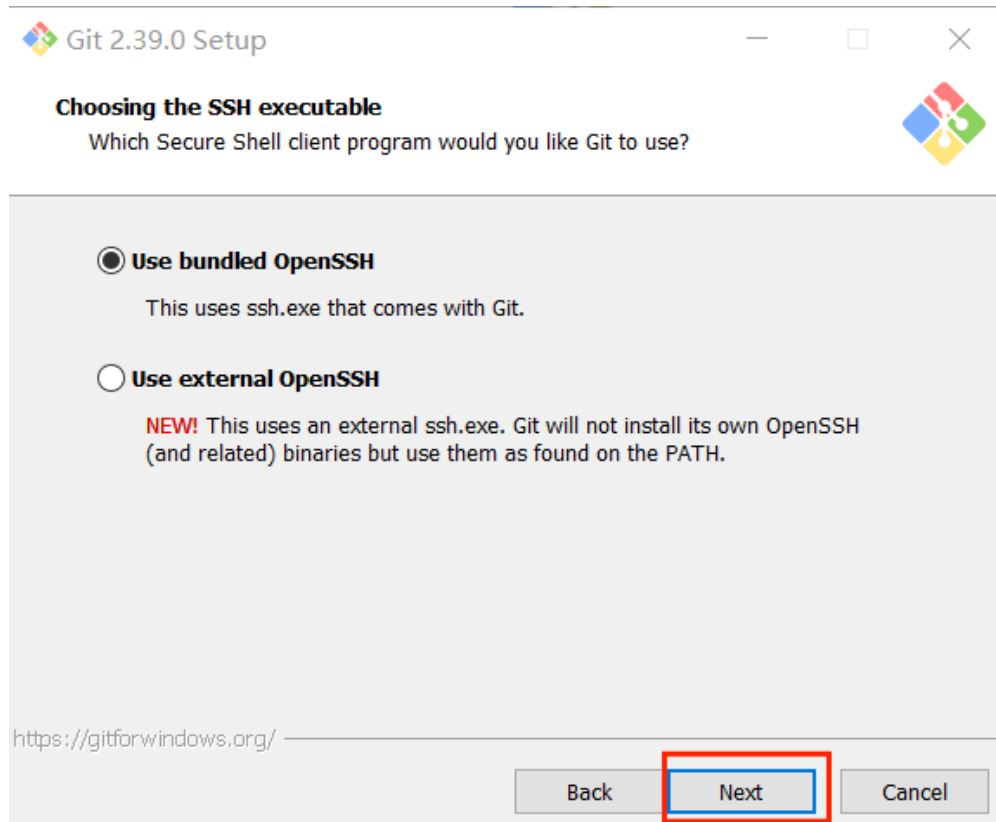
- The default name (master) after Git creates a branch. If there is no special default setting, click next.



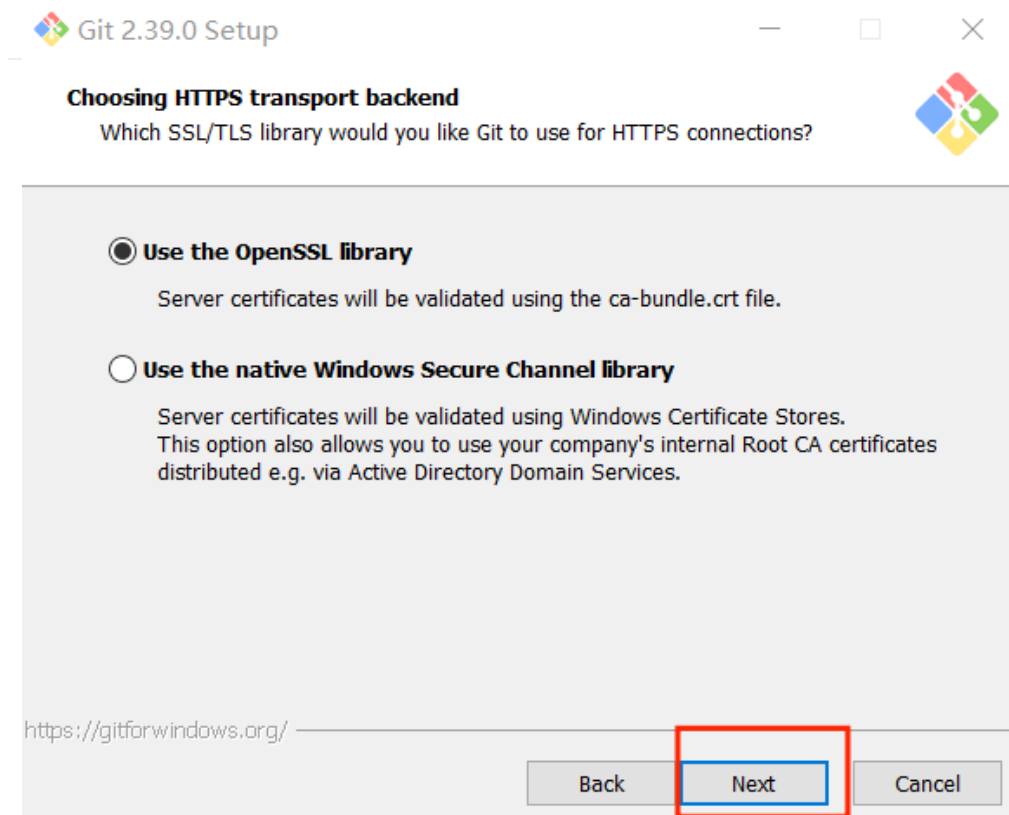
- The next step is the default, just select Next.



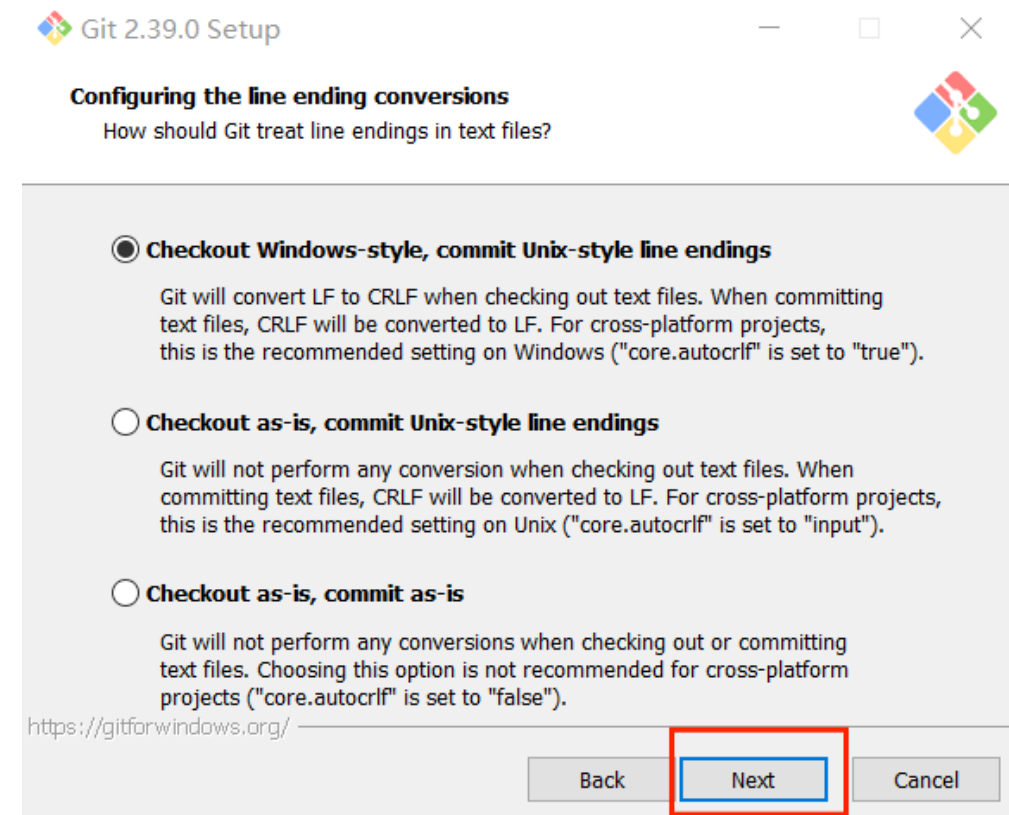
- The next step is the default, just select Next.



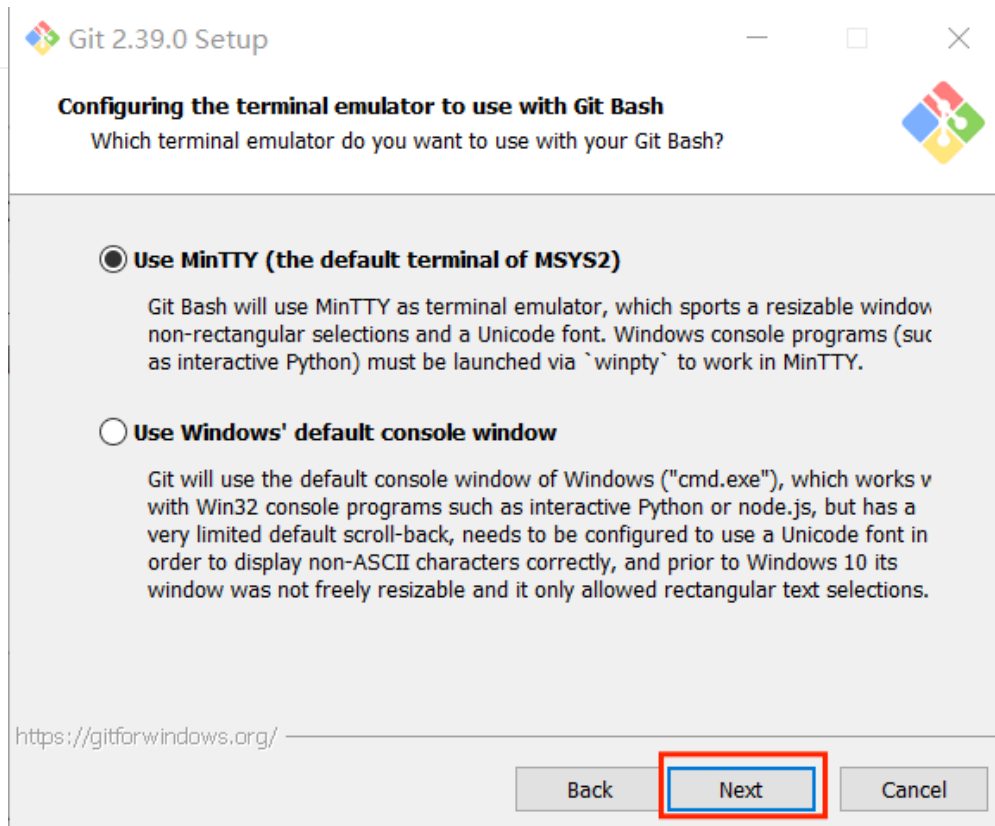
- The next step is the default, just select Next.



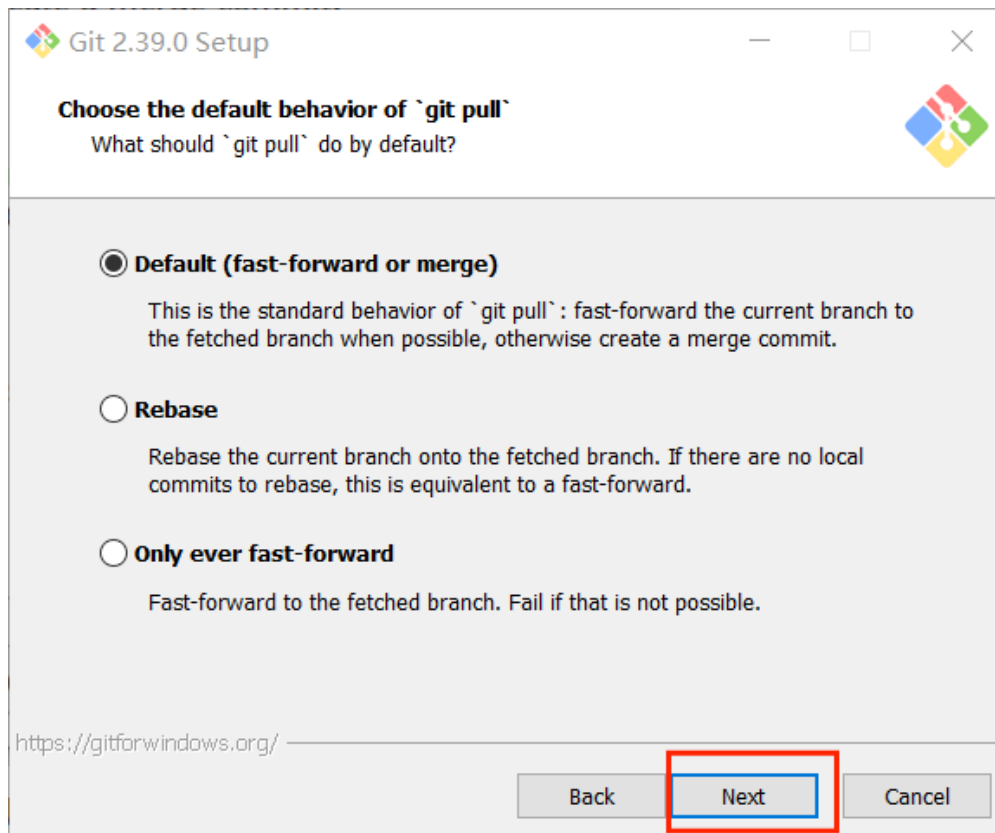
- The next step is the default, just select Next.



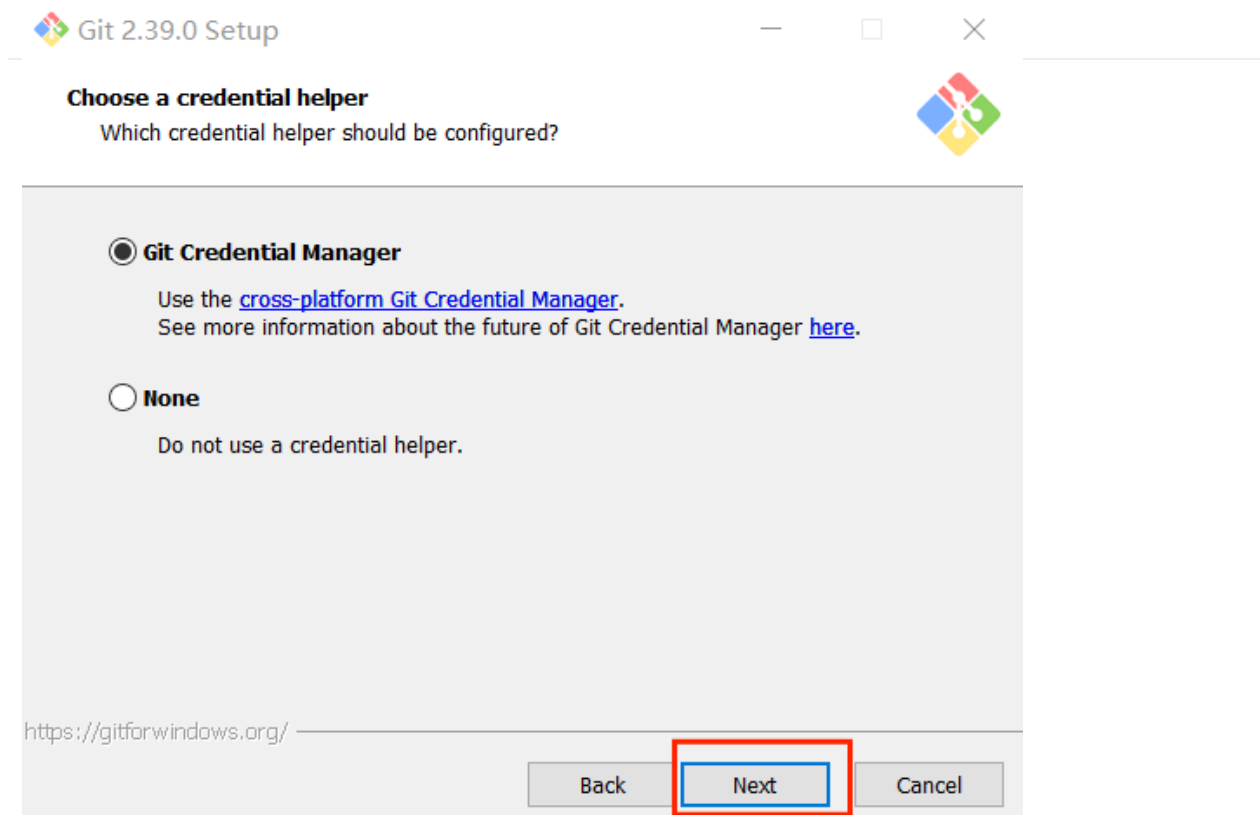
- The next step is the default, just select Next.



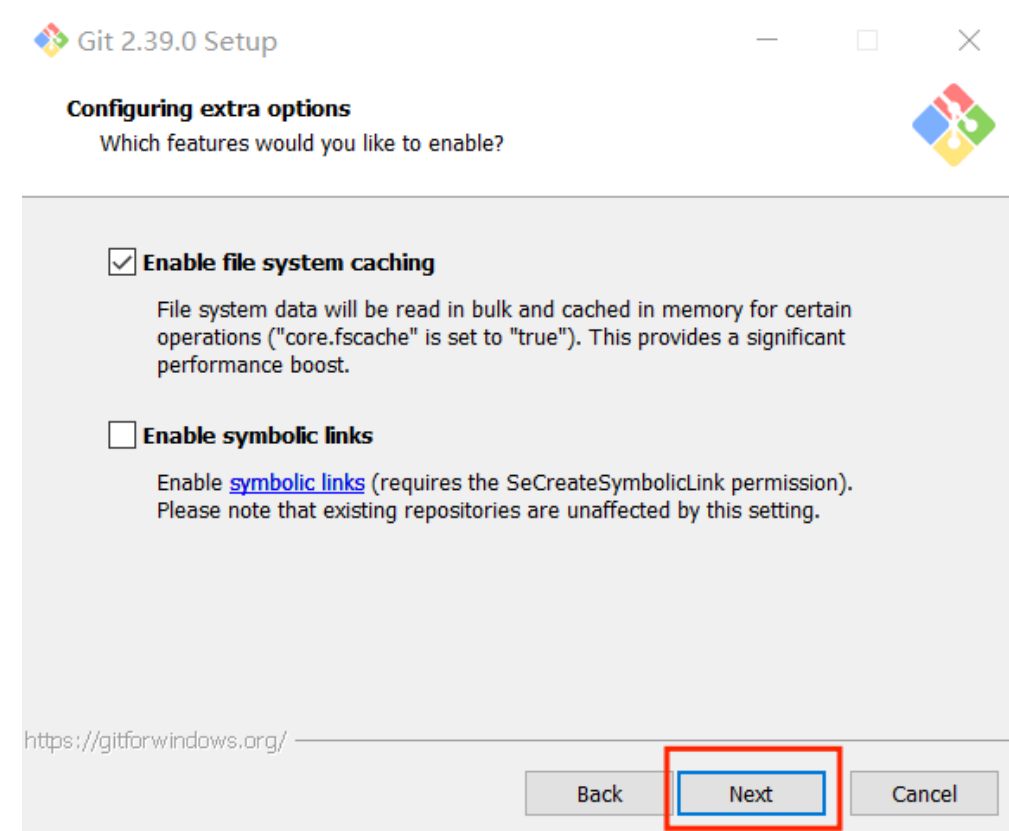
- The next step is the default, just select Next.



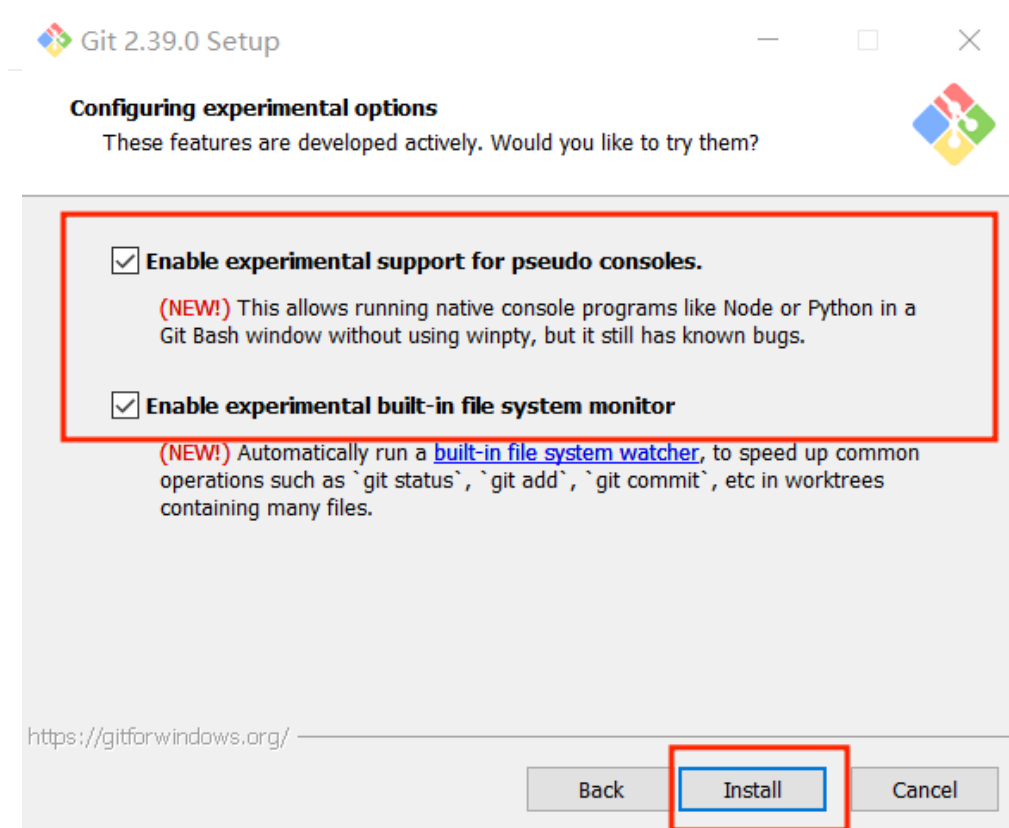
- The next step is the default, just select Next.



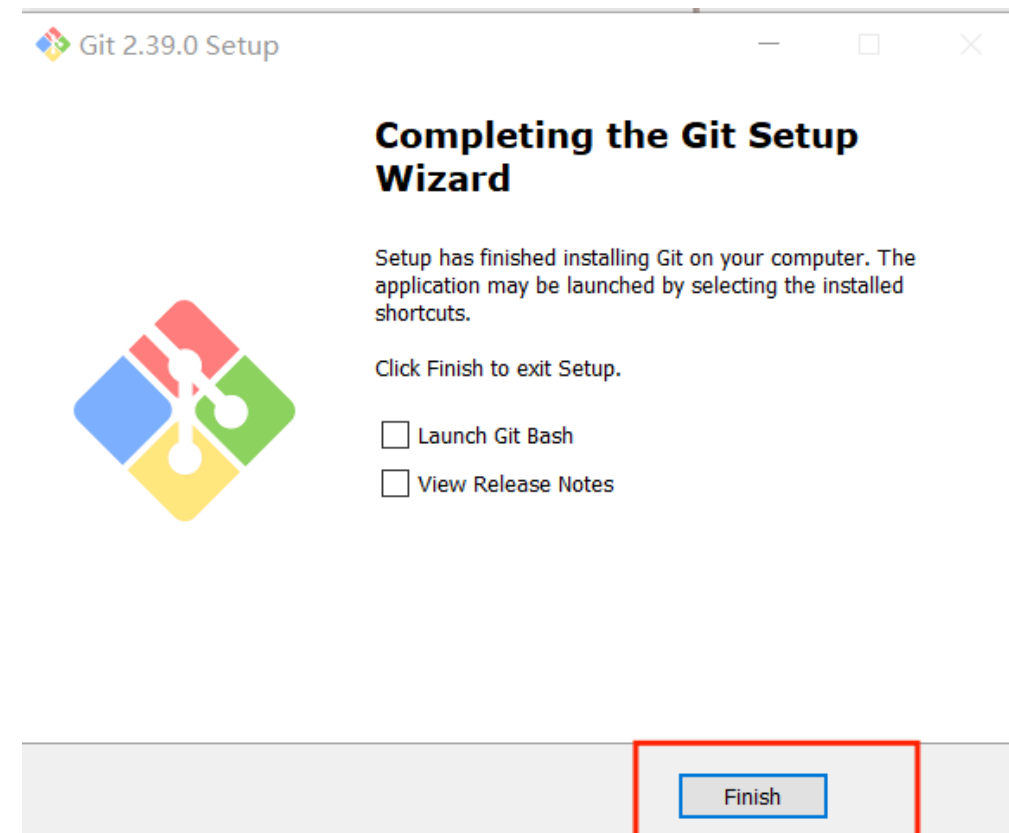
- The next step is the default, just select Next.



- Additional configuration options are not selected by default, you can check these two boxes, support (node, python although there are still bugs) and then choose install.



- After the installation is complete, click Finish to complete the installation.

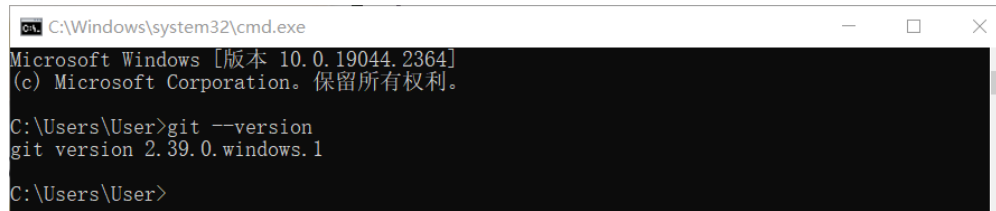


checking

Open a console terminal (shortcut key Win+R, enter cmd to enter the terminal), and enter the following command to view the Git version information:

1 Adaptive robot

```
git --version
```



```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.19044.2364]
(c) Microsoft Corporation。保留所有权利。

C:\Users\User>git --version
git version 2.39.0.windows.1

C:\Users\User>
```

You need to use git to download the aikit_V2 package in the future. For the use of git, please refer to the link below:

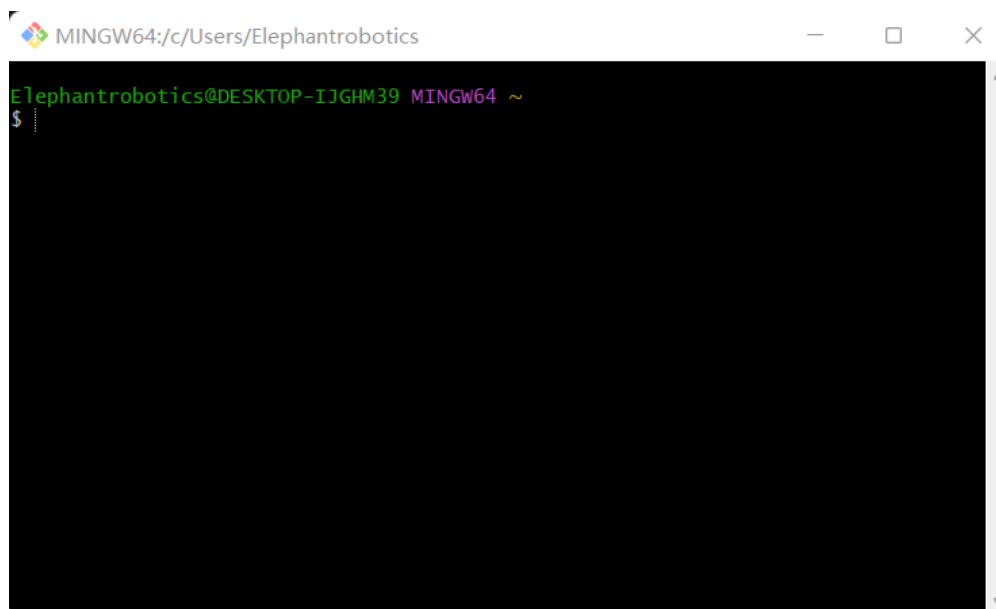
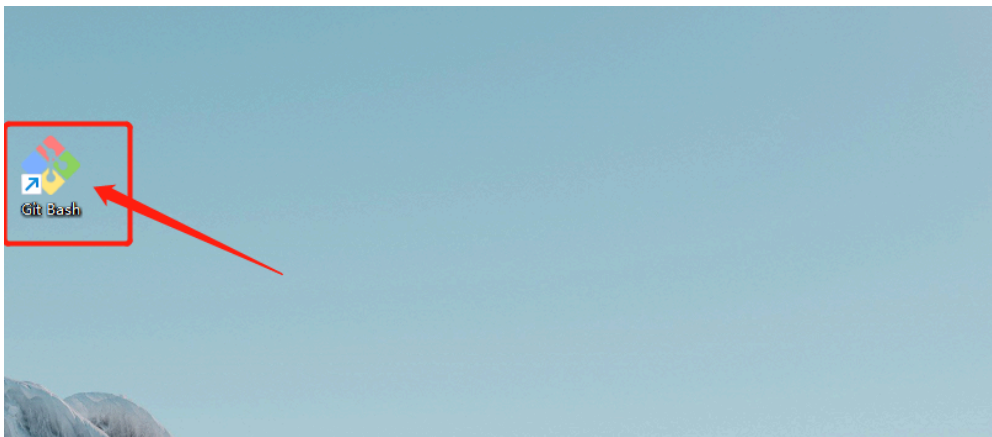
- <https://git-scm.com/book/zh/v2>
- <https://www.runoob.com/git/git-tutorial.html>

1.4 aikit_V2 Install

aikit_V2 is launched by ElephantRobotics, which is an artificial intelligence kit V2.0 package adapted to its desktop robotic arm.

project address: https://github.com/elephantrobotics/aikit_V2

- 1 Double-click the `Git Bash` icon on the desktop to enter the Git terminal.

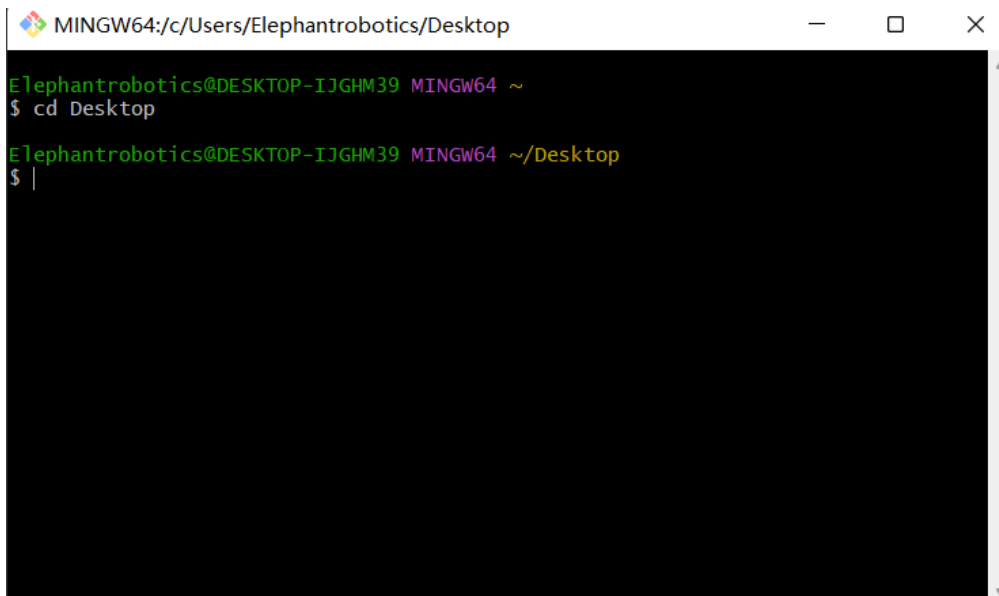


- 2 Enter the following command in the Git terminal:

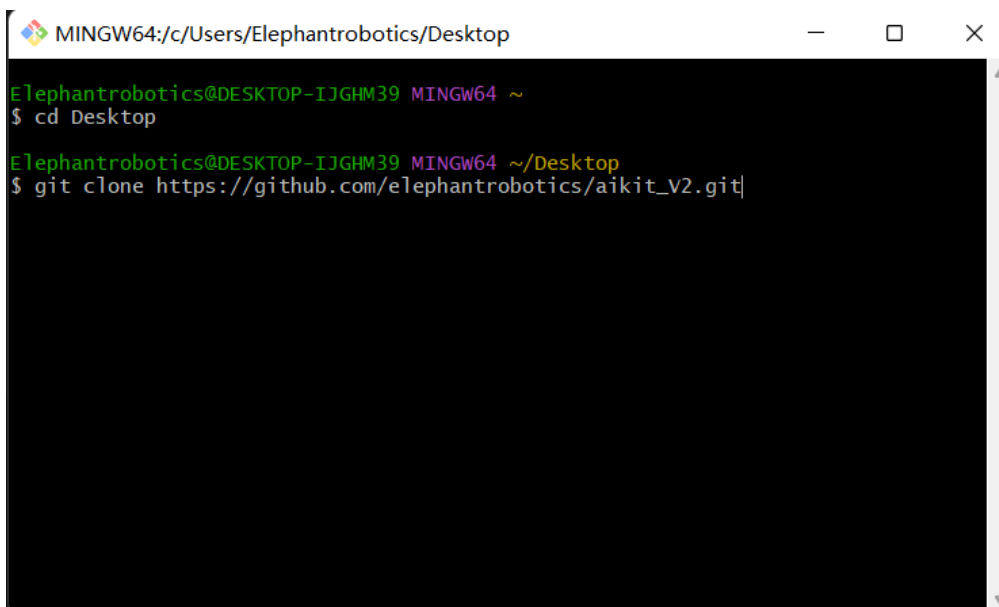
1 Adaptive robot

```
cd Desktop
```

```
git clone https://github.com/elephantrobotics/aikit_v2.git
```

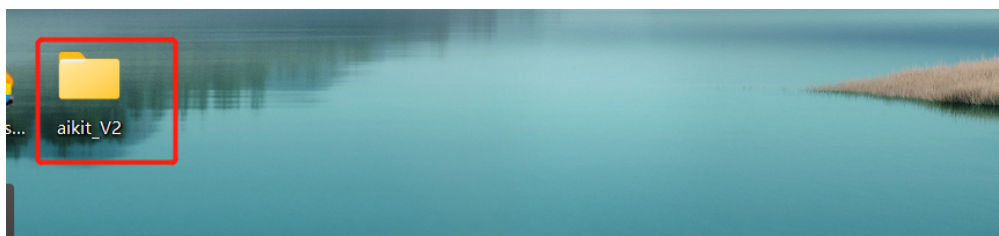


```
MINGW64:/c/Users/Elephantrobotics/Desktop  
Elephantrobotics@DESKTOP-IJGHM39 MINGW64 ~  
$ cd Desktop  
Elephantrobotics@DESKTOP-IJGHM39 MINGW64 ~/Desktop  
$ |
```



```
MINGW64:/c/Users/Elephantrobotics/Desktop  
Elephantrobotics@DESKTOP-IJGHM39 MINGW64 ~  
$ cd Desktop  
Elephantrobotics@DESKTOP-IJGHM39 MINGW64 ~/Desktop  
$ git clone https://github.com/elephantrobotics/aikit_v2.git
```

Then the `aikit_v2` folder will appear on the desktop.



At this point, the Windows environment is built.

2 Raspberry Pi Environment Construction

Since the Raspberry Pi version has a built-in Ubuntu 20.04 system, there is no need to build an environment.

- **Applicable devices:** myCobot 280 Pi、mechArm 270 Pi、myArm 300 Pi

2.1 Install OpenCV

Open a console terminal (shortcut key Ctrl+Alt+T), enter the command:

```
# The version numbers of the two need to be consistent, and version 4.6.0.66 is installed here
pip install opencv-python==4.6.0.66
pip install opencv-contrib-python==4.6.0.66
```

2.2 aikit_V2 Install

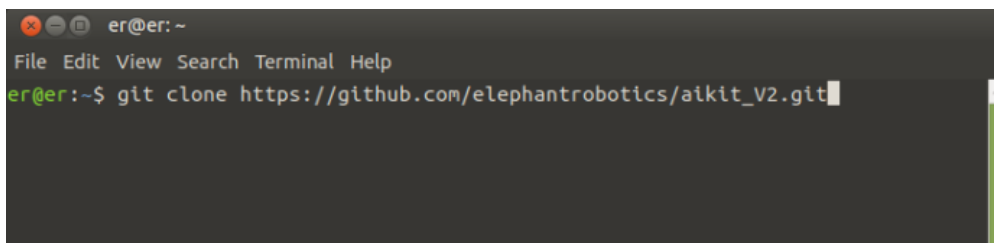
`aikit_v2` is launched by ElephantRobotics, which is an artificial intelligence kit V2.0 package adapted to its desktop robotic arm.

Applicable devices: myCobot 280 Pi、mechArm 270 Pi、myArm 300 Pi

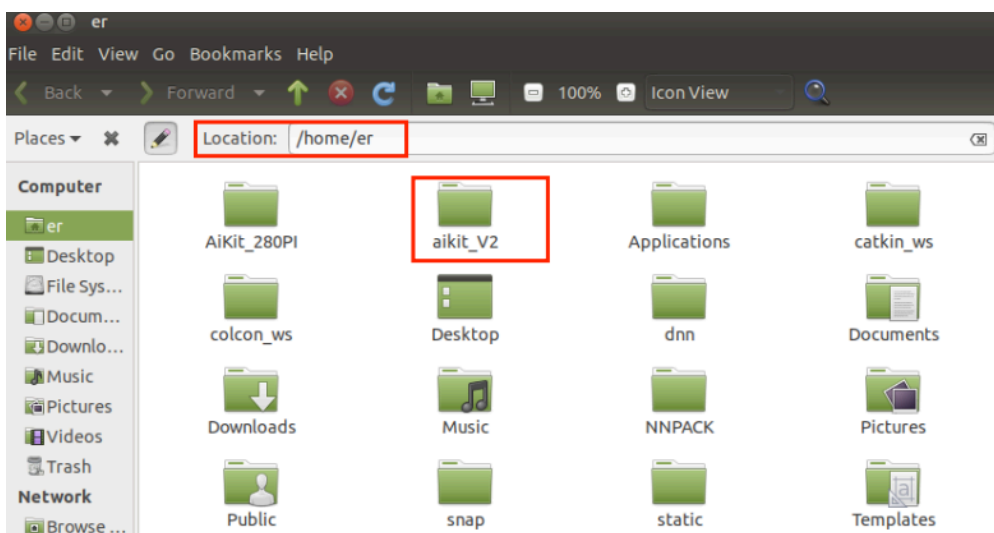
project address: https://github.com/elephantrobotics/aikit_V2

- 1 Open a console terminal (shortcut key Ctrl+Alt+T), enter the command:

```
git clone https://github.com/elephantrobotics/aikit_V2.git
```



- 2 Then under the main directory `/home/er`, there will be a folder named `aikit_V2`.



So far, `aikit_V2` has completed the installation.

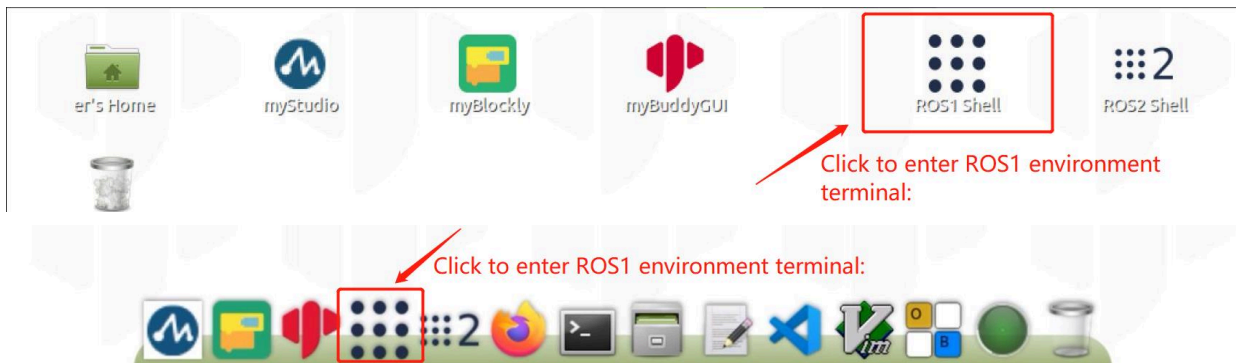
2.3 mycobot_ros Install and update

`mycobot_ros` is a ROS package launched by ElephantRobotics that is suitable for all types of desktop robotic arms. It also integrates the artificial intelligence suit V2.0 program package.

Applicable devices: myCobot 280 Pi

project address: https://github.com/elephantrobotics/mycobot_ros

Click the `ROS1 Shell` icon on the desktop or the corresponding icon in the lower bar of the desktop to open the ROS1 environment terminal:



Then enter the following command:

```
cd ~/catkin_ws/src # Go to the src folder in the workspace
# Clone the code on github
git clone https://github.com/elephantrobotics/mycobot_ros.git
cd .. # return to workspace
catkin_make # Build the code in the workspace
source devel/setup.bash # Add environment variables
```

Note: If the `mycobot_ros` folder already exists in the `/home/er/catkin_ws/src` (equivalent to `~/catkin_ws/src`) directory, you need to delete the original `mycobot_ros` before executing above command.

1 OpenCV simple use

1.1 What is OpenCV

The full name of openCV is Open Source Computer Vision Library. It is an open source computer vision library of Intel (Intel), which consists of a series of C functions and C++ classes, and realizes many general algorithms in image processing and computer vision.

1.2 Installation and use of OpenCV

Windows

Open a console terminal (shortcut key Win+R, enter cmd, enter the terminal), enter the following command:

```
# The version numbers of the two need to be consistent, and version 4.6.0.66 is installed here
pip install opencv-python==4.6.0.66
pip install opencv-contrib-python==4.6.0.66
```

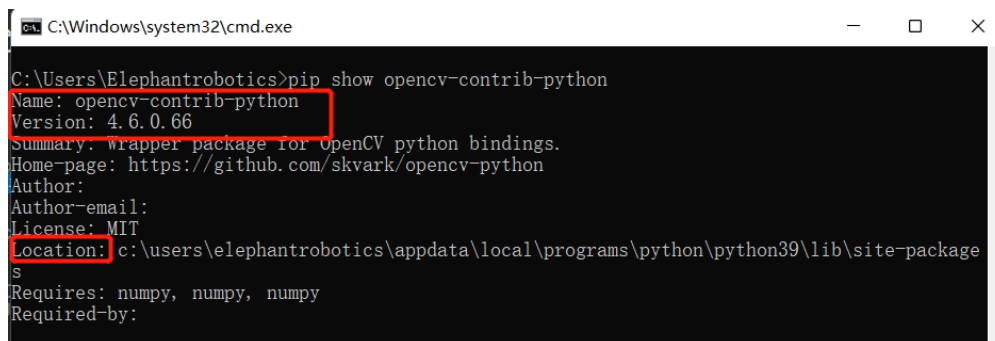
- Installation Verification

```
# You can also view other python libraries format: pip show library name
pip show opencv-python
pip show opencv-contrib-python
```



```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.22000.1335]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\Elephantrobotics>pip show opencv-python
Name: opencv-python
Version: 4.6.0.66
Summary: wrapper package for OpenCV python bindings.
Home-page: https://github.com/skvark/opencv-python
Author:
Author-email:
License: MIT
Location: c:\users\elephantrobotics\appdata\local\programs\python\python39\lib\site-package
s
Requires: numpy, numpy, numpy
Required-by:
```



```
C:\Windows\system32\cmd.exe

C:\Users\Elephantrobotics>pip show opencv-contrib-python
Name: opencv-contrib-python
Version: 4.6.0.66
Summary: wrapper package for OpenCV python bindings.
Home-page: https://github.com/skvark/opencv-python
Author:
Author-email:
License: MIT
Location: c:\users\elephantrobotics\appdata\local\programs\python\python39\lib\site-package
s
Requires: numpy, numpy, numpy
Required-by:
```

Linux

Open a console terminal (shortcut key Ctrl+Alt+T), enter the command:

1 Adaptive robot

```
# The version numbers of the two need to be consistent, and version 4.6.0.66 is installed here
pip install opencv-python==4.6.0.66
pip install opencv-contrib-python==4.6.0.66
```

simple use

```
import cv2
def openCamera():
    print('opencv version',cv2.__version__)
    print("Keyboard ESC to exit")

    cap = cv2.VideoCapture(0) # Linux
    # cap = cv2.VideoCapture(1) # Windows
    while cap.isOpened():
        sucess,frame = cap.read()
        if sucess:
            cv2.imshow("frame",frame)
            if cv2.waitKey(10)==27:
                break
    cap.release()
    cv2.destroyAllWindows()

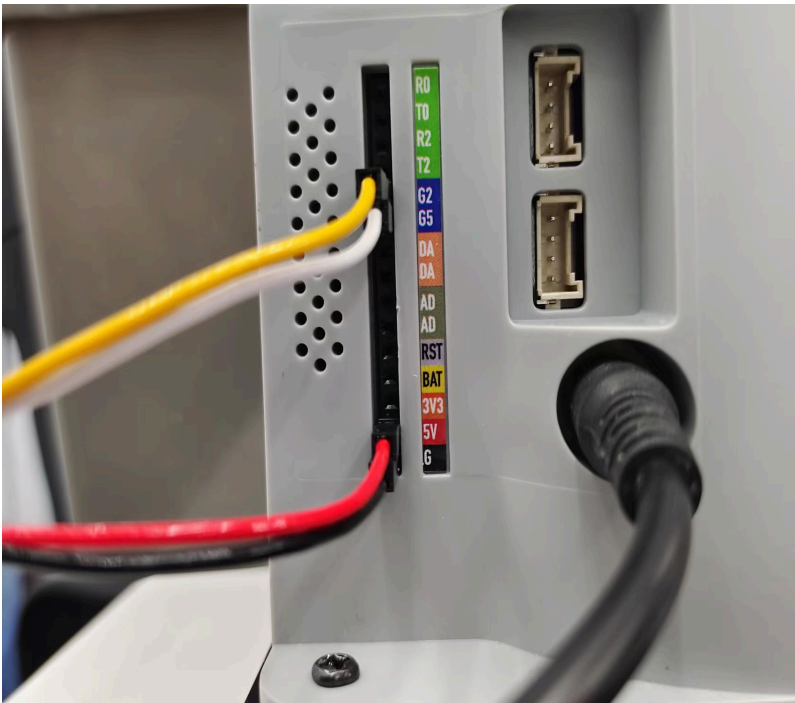
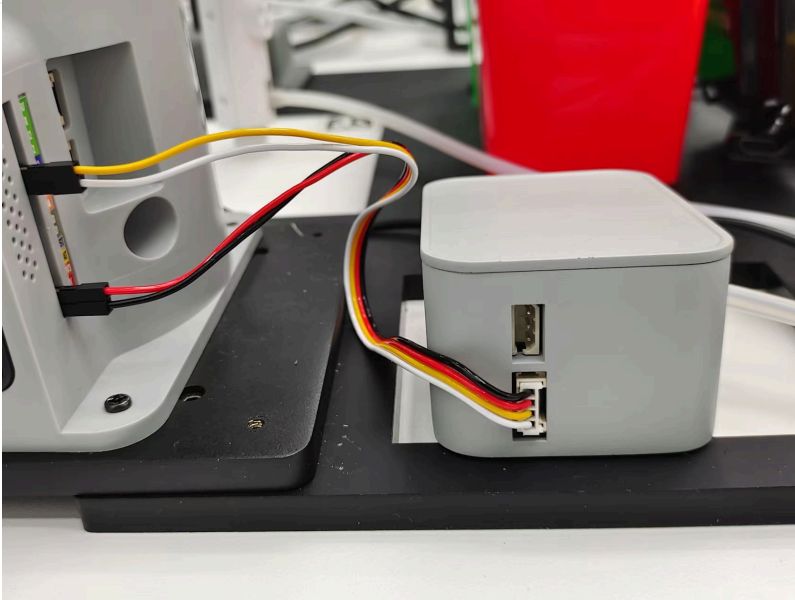
if __name__=="__main__":
    openCamera()
```

2 End effector



- pump:

- **M5 version:** Connect the pin interface of G2, G5, 5V and GND on the left side of the robot arm.



- **Raspberry Pi version:** The red, black, yellow, and white wires in the DuPont line are respectively connected to the 5V, GND, 20, and 21 pins in the first row of pins.



Precautions

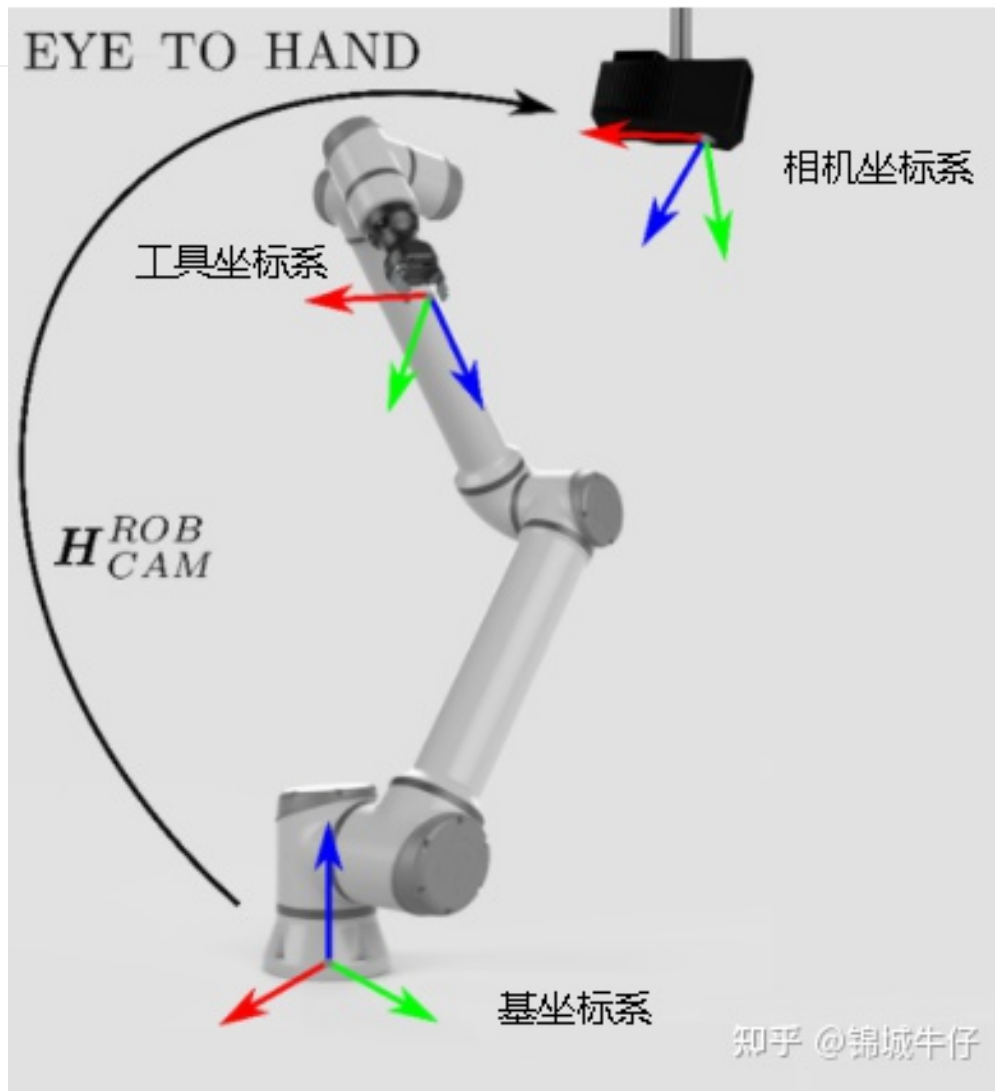
- Please make sure that the product has been successfully connected according to the instructions
- Please make sure the product is powered by the included adapter
- Please make sure the power adapter is working properly
- Please ensure the connection direction of the positive and negative poles

3 Coordinate Transformation and Calibration

3.1 Image coordinate transformation and calibration

3.1.1 four coordinate systems

In the development of robotic arm vision, an important preparatory work is to understand the meaning of the four coordinate systems, namely the camera coordinate system, the world coordinate system, the tool coordinate system and the base coordinate system.



- World coordinate system: The world coordinate system (world coordinate), also known as the measurement coordinate system, is a three-dimensional Cartesian coordinate system, based on which the spatial position of the camera and the object to be measured can be described. The position of the world coordinate system can be freely determined according to the actual situation. It can be understood as the earth coordinate system.
- Camera coordinate system: The camera coordinate system (camera coordinate) is a three-dimensional rectangular coordinate system. The origin is located at the optical center of the lens. The x and y axes are respectively parallel to the two sides of the phase surface. The z axis is the lens optical axis and is perpendicular to the image plane.
- Tool coordinate system: The tool coordinate system of the 6-axis or 7-axis collaborative arm. The origin is generally the center of the end of the robot arm, also called TCP (ToolCenterPoint). The Z axis coincides with the axis of the end and points to the direction of the end.
- Base coordinate system: Robots have a base coordinate system, and the default origin is the center point of the base of the collaborative arm.

3.1.2 camera calibration

The conversion of one coordinate system to another coordinate system mainly includes two aspects: position (origin) and attitude (XYZ axis). The transformation of the origin position is called translation, and the transformation of attitude is called rotation

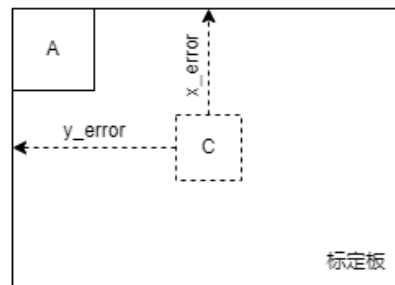
3.1.3 Convert pixel coordinate system to base coordinate system

- Taking the visual recognition of a dual-arm robotic arm as an example

Obtain the value of the pixel center point in the image through the camera, and make a difference with the pixel center point of the starting point to obtain the pixel difference value. Transform the calculated pixel difference according to the scale set by the calibration. Converted to the difference in the world coordinate system, after the robot arm obtains the world coordinate of the starting point, it can obtain the actual world coordinate to be grasped according to the calculated difference of the world coordinate. Send the coordinates to the robotic arm to realize positioning and grabbing.

双臂机器人二维码检测原理图

相机坐标系下的检测区域



- ①发送角度让机械臂运动到A二维码上方。获取区域A的世界坐标

使用api:
`.send_angles()`
`.get_base_coord()`

- ② 读取二维码C的像素坐标，与区域a的像素坐标做差值

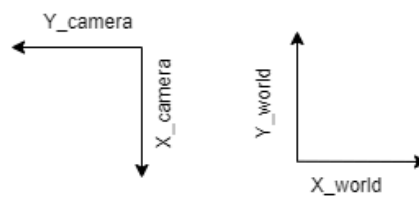
$x_error = C_x - A_x$
 $y_error = C_y - A_y$

- ③计算像素差值与实际距离差值的比例

$ratio_x = \text{实际}x\text{差值} / \text{像素差值}$
 $ratio_y = \text{实际}y\text{差值} / \text{像素差值}$
 $x_world = x_error * ratio_x$
 $y_world = y_error * ratio_y$
注意：计算时注意单位转换

- ④像素坐标系转换到机器人的基坐标系

在实验测试阶段
 相机坐标系，基坐标系如下图所示



- ⑤在获得世界坐标后，根据二维码A的世界坐标做偏移

$move_x_world = x_a + x_world$
 $move_y_world = y_a + y_world$

- ⑥发送世界坐标给机器人

使用api:
`.write_base_coords()`

3.2 Coordinate Transformation and Calibration of Robotic Arm

- The pixel distance M1 between the two arucos can be calculated by measuring the pixel points of the two arucos in the capture area, and the actual distance M2 of the two arucos can be measured, so that we can obtain the ratio of the pixel to the actual distance = $M2/M1$. We can calculate the pixel difference between the block and the center of the grabbing area from the picture, so that we can calculate the relative coordinates (x1, y1) of the block from the center of the grabbing area. Add the relative coordinates (x1, y1) to the relative coordinates (x2, y2) from the center of the grasping area to the robotic arm to obtain the relative coordinates (x3, y3) of the object to the robotic arm.

```
# Grab the coordinates of the center point relative to the robot arm
self.camera_x, self.camera_y = camera_x, camera_y

# The coordinates of the object relative to the robot arm can be set as:
self.c_x = self.c_y = 0, 0

# Pixel to Actual Ratio
self.ratio = 0

# Set the parameter function used to calculate the coordinates between the cube and the arm
def set_params(self, c_x, c_y, ratio):
    self.c_x = c_x
    self.c_y = c_y
    self.ratio = 220.0/ratio

# Get the position function, used to calculate the coordinates between the cube and the arm
def get_position(self, x, y):
    return ((y - self.c_y)*self.ratio + self.camera_x), ((x - self.c_x)*self.ratio + self.camera_y)
```

- Next, the program will call the setting parameter function **set_params()**, which is used to calculate and set the parameters for calculating the real coordinates between the object and the robot arm

```
detect.set_params(
    (detect.sum_x1+detect.sum_x2)/20.0,
    (detect.sum_y1+detect.sum_y2)/20.0,
    abs(detect.sum_x1-detect.sum_x2)/10.0 +
    abs(detect.sum_y1-detect.sum_y2)/10.0
)
```

- Then the function **get_position()** will be called to calculate the real coordinates between the cube and mycobot.

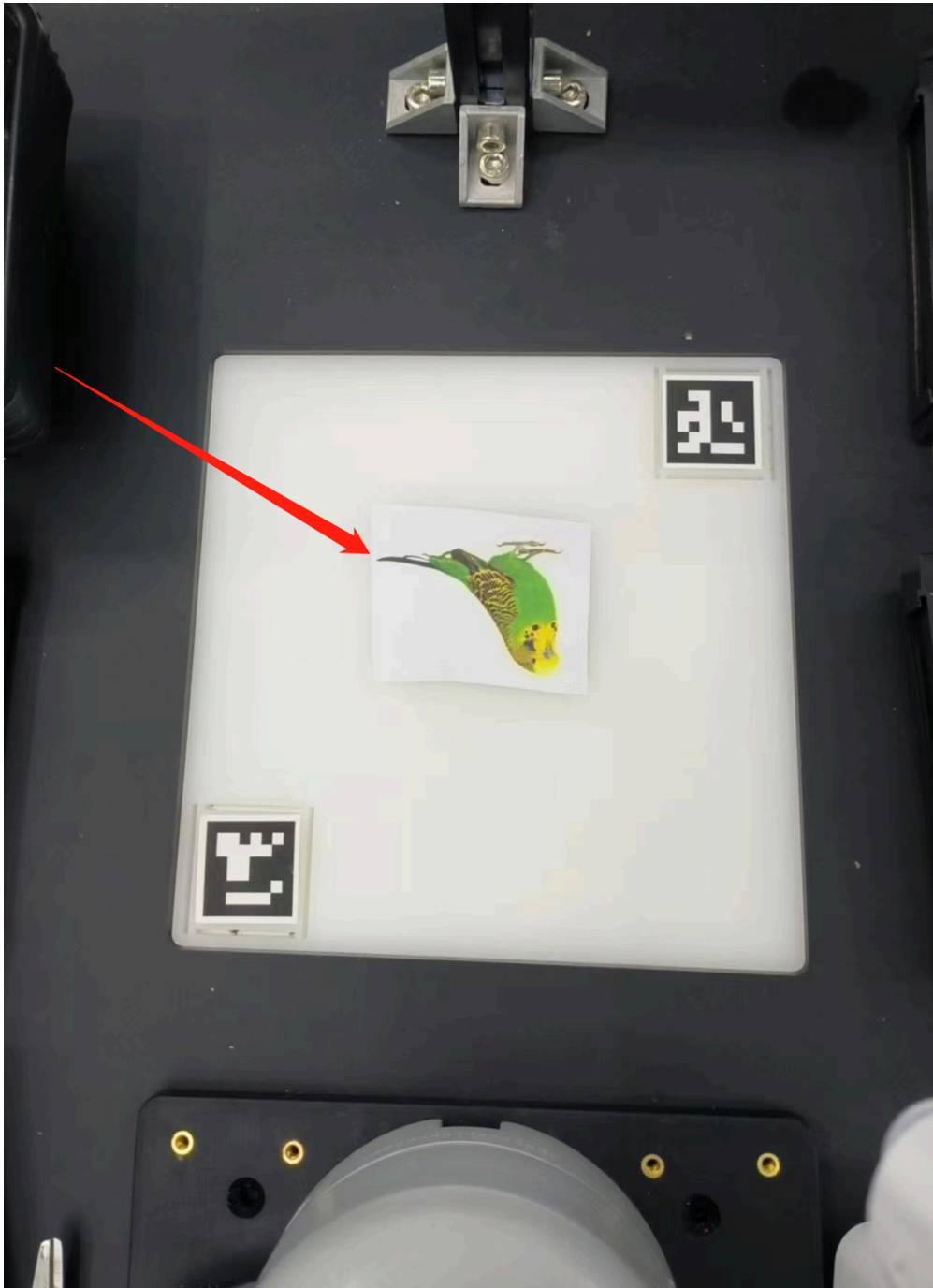
```
real_x, real_y = detect.get_position(x, y)
```

- Finally, the real coordinates obtained will be slightly processed, and passed into the **decide_move()** function for subsequent robotic arm grabbing.

```
detect.decide_move(real_sx/20.0, real_sy/20.0, detect.color)
```

Precautions

1. **opencv-python** and **opencv-contrib-python** version libraries need to be consistent, and the version number is greater than **4.5.5**.
2. When the camera does not automatically frame the recognition area correctly, you need to close the program, adjust the position of the camera, and move the camera to the left or right.
3. If the command terminal does not show ok, and the color cannot be recognized, you need to move the camera slightly backward or forward, and the program can run normally when the command terminal shows ok.
4. OpenCV color recognition will be affected by the environment, and the recognition effect will be greatly reduced in a darker environment.
5. When performing yolov5 image recognition, in order to better recognize the picture, it is necessary to reverse the object in the recognition area**.



Ai Kit UI Instructions

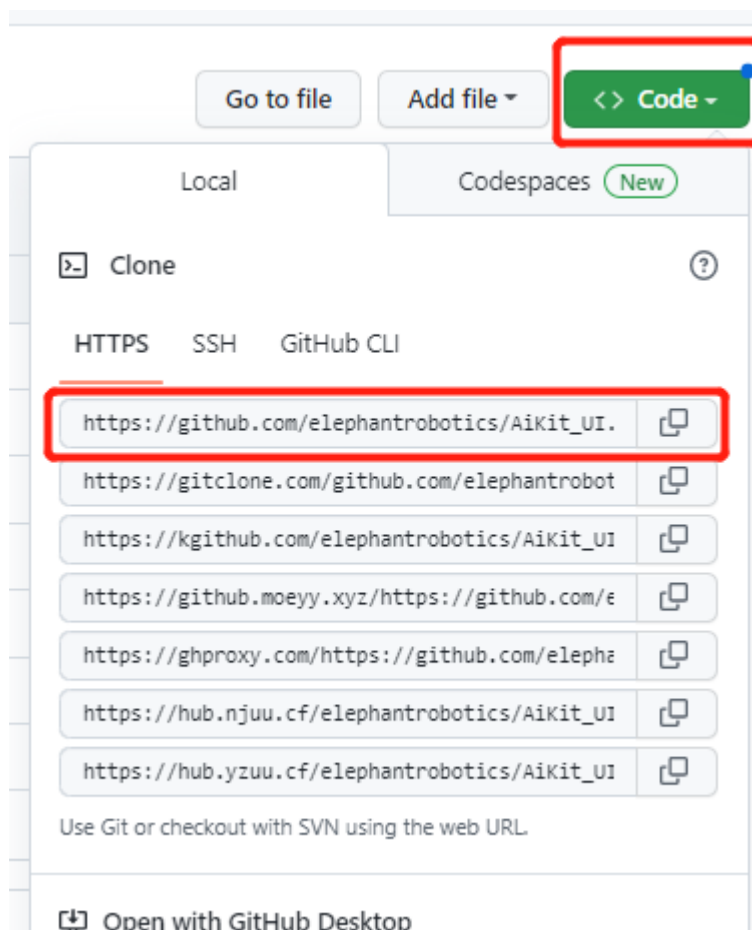
AiKit UI Download

NOTE: myArm 300 Pi devices should use the `myArm-300` branch, That is https://github.com/elephantrobotics/AiKit_UI/tree/myArm-300

1. Enter URL: https://github.com/elephantrobotics/AiKit_UI
2. download item

download using git

Use git clone, under the 'Code' button, take the first address as an example



Right click in the folder to open the git window

Git Bash Here

Enter the command:

1 Adaptive robot

```
git clone https://github.com/elephantrobotics/AiKit_UI.git

# If using a myArm 300 Pi device, enter:
git clone -b myArm-300 https://github.com/elephantrobotics/AiKit_UI.git
```

Download the ZIP file directly, under the 'Code' button.

 Download ZIP

start method

path

```
python <path>/main.py
```

After the startup is successful, as shown in the figure below:

Elephant Robotics AI Kits

Connection **CONNECT**

Serial Port:

Device:

Baud:

Control **Auto Mode**

Homing **Go**

Recognition **Run**
Algorithm: Color recognition

Pick **Run**

Place **Run**
☒ BinA ☐ BinB ☐ BinC ☐ BinD

XY Offsets **Save**
X offset: Y offset:

Algorithm **Open File**
Select

Display
current coordinates
image coordinates

Add New Pictures **Add** **Exit**

Camera **Open** **English**

Features

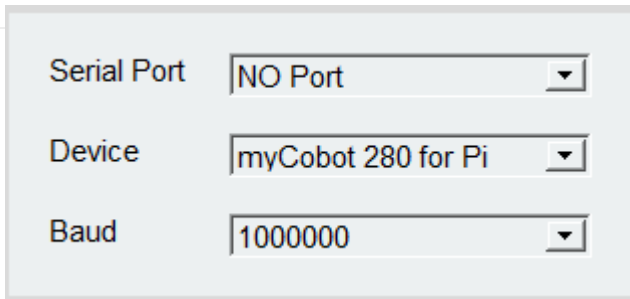
language switch

Click the button in the upper right corner of the window to switch between languages (Chinese, English).

Connection **CONNECT** **Camera** **Open** **English**

device connection

1. Select serial port, device, baud rate

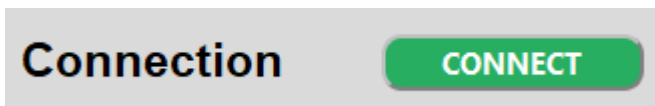


Serial Port NO Port

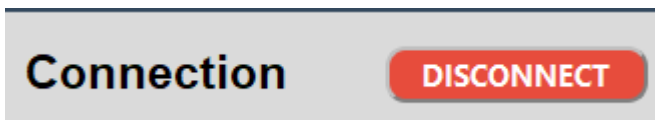
Device myCobot 280 for Pi

Baud 1000000

2. Click the 'CONNECT' button to connect, after the connection is successful, the 'CONNECT' button will change to 'DISCONNECT'



3. Clicking the 'DISCONNECT' button will disconnect the robot arm



4. After the robotic arm is successfully connected, the gray button will be lit and become clickable.

Control

Auto Mode

Homing

Go

Recognition

Run

Algorithm: Color recognition

Pick

Run

Place

Run

☒ BinA
 ☐ BinB
 ☐ BinC
 ☐ BinD

XYZ Offsets

Save

X: 260

Y: 5

Z: 142

Algorithm

Open File

Select

Color recognition

Add New Pictures

Add

Exit

Display

current coordinates

image coordinates

Turn on the camera

1. Set the camera serial number, the default serial number is 0, when Windows is used, the serial number is usually 1, and when Linux is used, the serial number is usually 0.

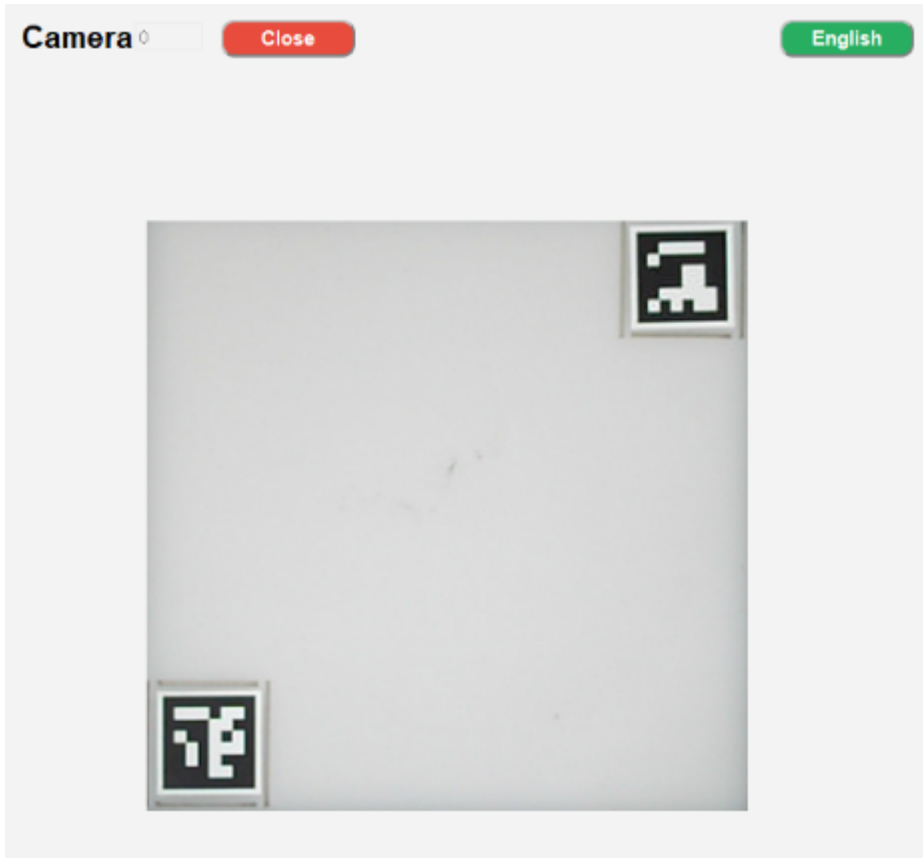
Camera

0

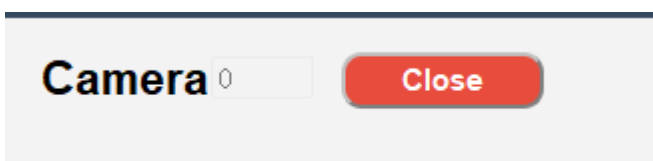
Open

1 Adaptive robot

2. Click the 'Open' button to try to open the camera. If the opening fails, you should try to change the camera serial number; the camera is successfully opened as shown in the figure below: Note: Before use, the camera should be adjusted to be just above the QR code whiteboard, and there is a line The straight line is facing the mechanical arm.



3. After successfully opening the camera, click the 'Close' button to close the camera



algorithm control

1. Fully automatic mode, after clicking the 'Auto Mode' button, the recognition, grabbing, and placing will always be on; click the 'Auto Mode' button again to turn off the fully automatic mode.

The screenshot shows a 'Control' panel with a red 'Auto Mode' button at the top right. Below it are four sections: 'Homing' with a blue 'Go' button; 'Recognition' with a red 'Run' button and the text 'Algorithm: Color recognition'; 'Pick' with a red 'Run' button; and 'Place' with a red 'Run' button and four radio buttons labeled 'BinA' (selected), 'BinB', 'BinC', and 'BinD'.

2. Go back to the initial point of grabbing, click the 'Go' button, it will stop the current operation and return to the initial point.

A close-up of the 'Homing' section, showing the label 'Homing' and a blue 'Go' button.

3. Step-by-step Recognition recognition: click the 'Run' button to start the recognition, Algorithm is the current algorithm used.

A close-up of the 'Recognition' section, showing the label 'Recognition', the text 'Algorithm: Color recognition', and a blue 'Run' button.

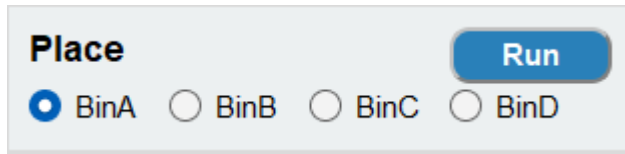
Pick: Click the 'Run' button to start the capture.

After the capture is successful, the recognition and capture will be automatically closed, and you need to click it again for the next use.

A close-up of the 'Pick' section, showing the label 'Pick' and a blue 'Run' button.

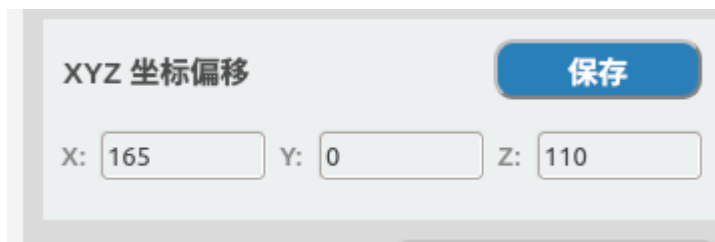
Placement: Click the 'Run' button to start placing.

The BinA, BinB, BinC, and BinD selection boxes correspond to BinA, BinB, BinC, and BinD 4 storage boxes, respectively, and will be placed in the designated storage box after selection.

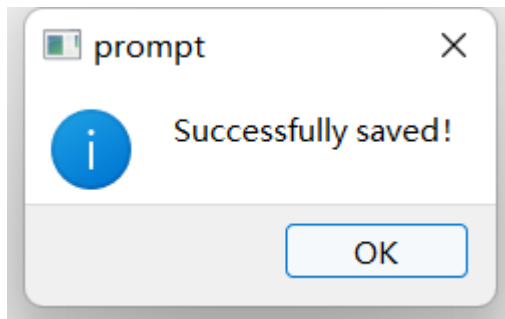


A control panel titled "Place" with a blue "Run" button. Below the title are four radio buttons labeled "BinA", "BinB", "BinC", and "BinD". The "BinA" radio button is selected, indicated by a blue dot.

4. Grasping point adjustment, X offset, Y offset, Z offset respectively represent the position of the X-axis, Y-axis, and Z-axis of the robot arm coordinates. They can be modified according to actual needs. Click the 'Save' button Save it. After successful saving, it will be captured according to the latest point.



A control panel titled "XYZ 坐标偏移" (XYZ Coordinate Offset) with a blue "保存" (Save) button. Below the title are three input fields labeled "X:", "Y:", and "Z:". The "X:" field contains the value "165", the "Y:" field contains "0", and the "Z:" field contains "110".



5. Open the file location, our code is open source, you can modify it according to your needs, click the 'Open File' button to open the file location.



A section header titled "Algorithm" with a green "Open File" button to its right.

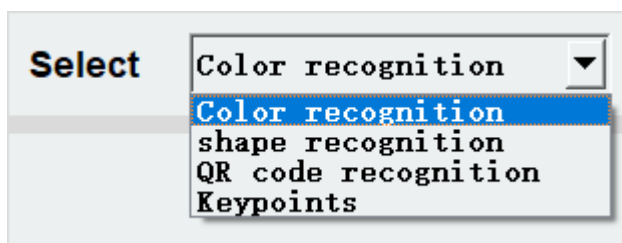
Open the 'main.py' file and modify it

1 Adaptive robot

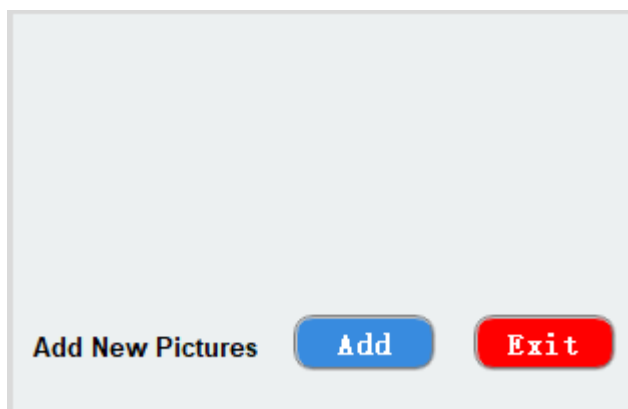
📁 .git	2023/1/6 15:46	文件夹	
📁 .idea	2023/1/6 17:12	文件夹	
📁 __pycache__	2022/12/16 10:39	文件夹	
📁 log	2023/1/6 17:12	文件夹	
📁 offset	2023/1/4 18:25	文件夹	
📁 pyqtFile	2023/1/6 15:56	文件夹	
📁 res	2022/12/20 17:20	文件夹	
📄 main.py	2023/1/6 15:33	Python 源文件	97 KB
📄 main.py.bak	2023/1/4 16:18	BAK 文件	97 KB
📄 README.md	2022/12/5 11:16	Markdown File	7 KB

Note: The 'main.py.bak' file is the backup of the 'main.py' file, delete the 'main.py' file when you need to use it, and re-modify the 'main.py'. The suffix of the bak' file is 'main.py'; then re-backup the 'main.py' file and name it 'main.py.bak'; you can also choose to re-download the project.

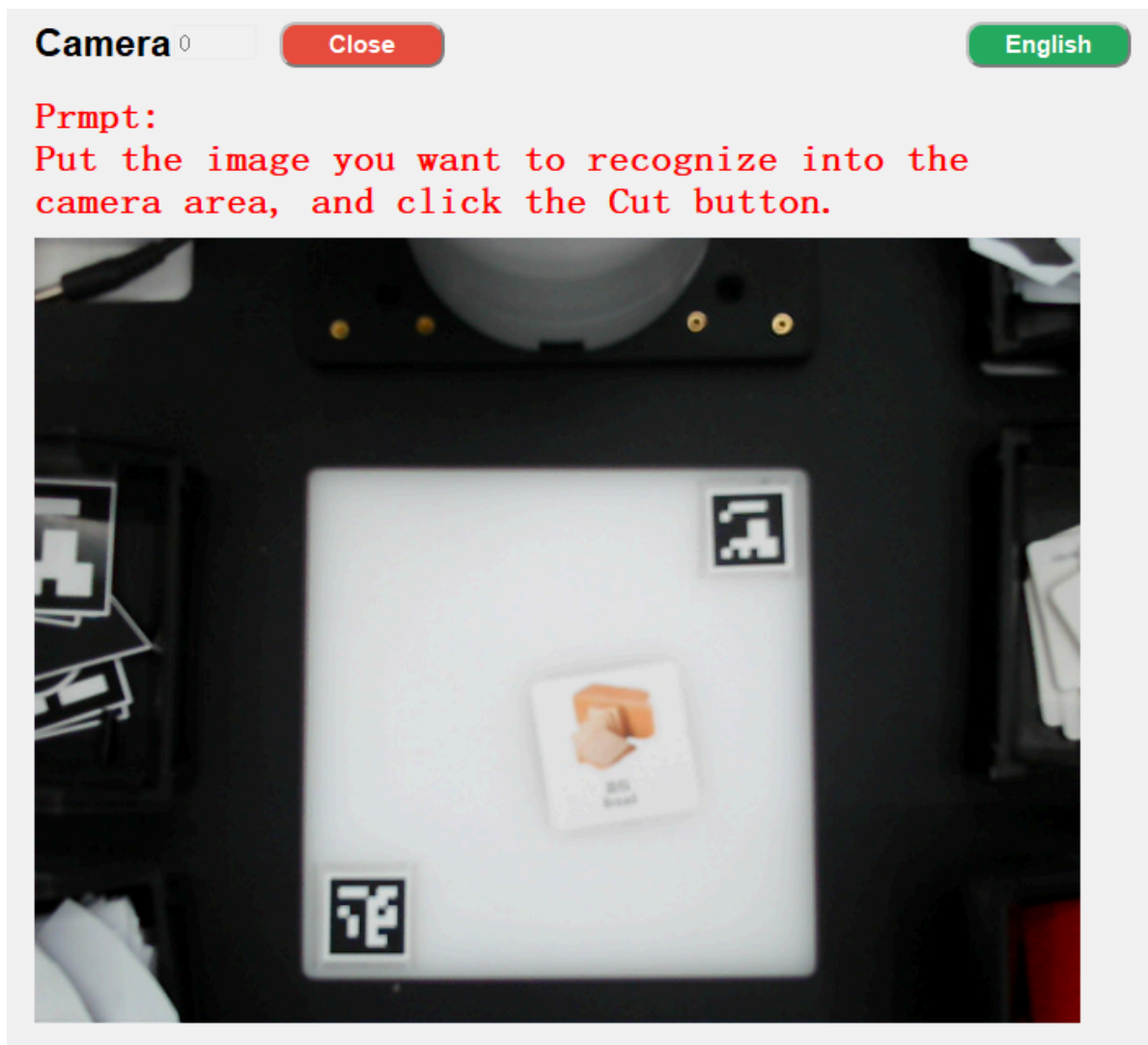
6. Algorithm selection includes color recognition, shape recognition, two-dimensional code recognition, and Keypoints recognition. Selecting the corresponding algorithm will perform corresponding recognition.



7. Add a picture for 'Keypoints'

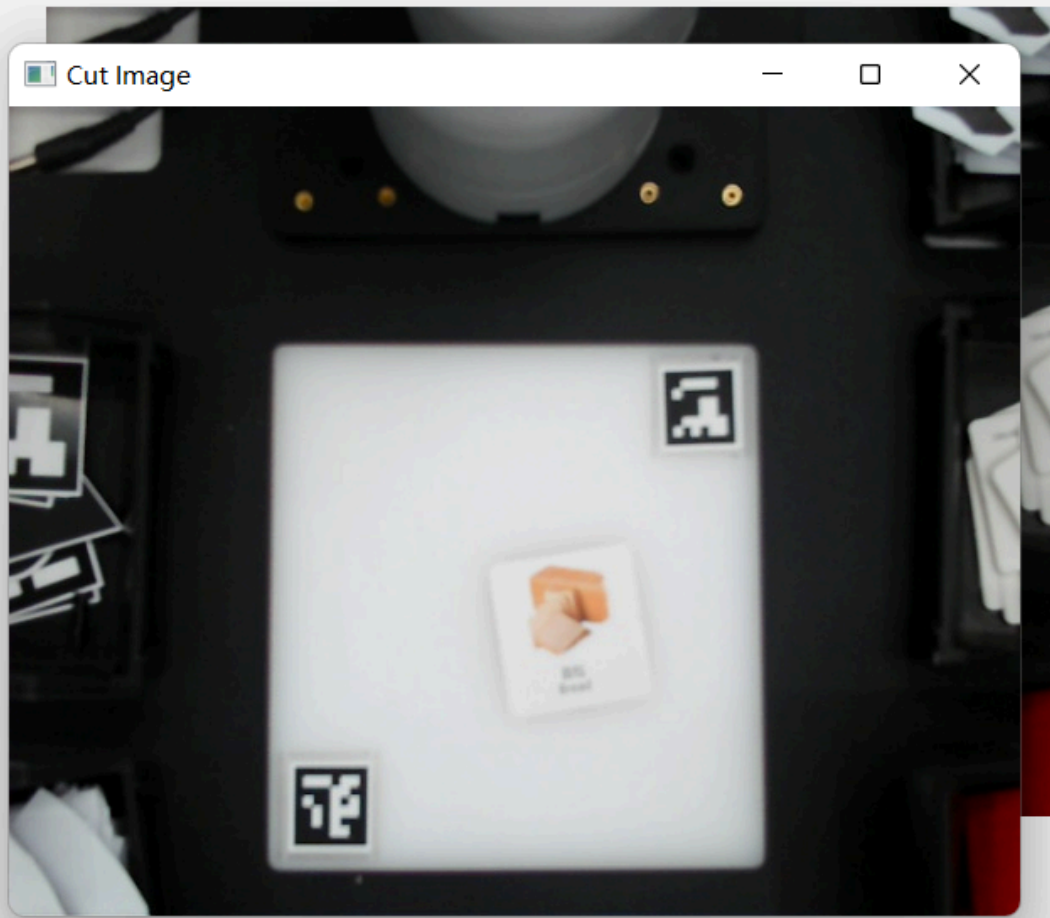


Click the 'Add' button, the camera will open and a prompt will appear.

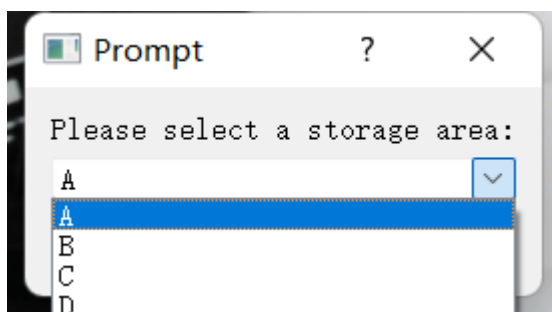


Click the 'Cut' button, the current camera content will be intercepted, and a prompt will be given to 'press the ENTER key after the content needs to be saved'

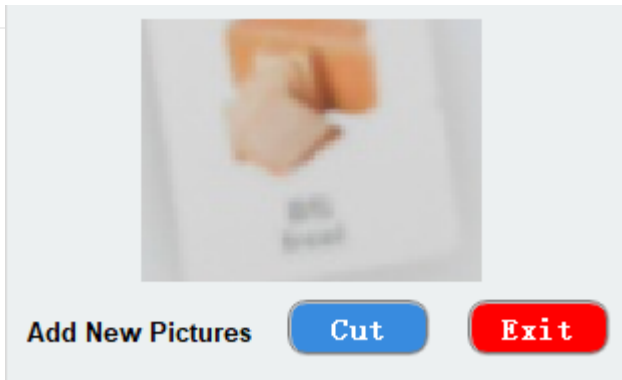
Prmpt:
Please intercept the part to be recognized, and
then press the enter key



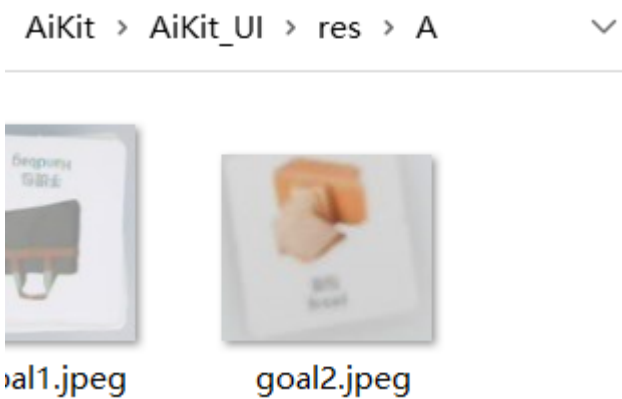
Frame the content to be saved and press the ENTER key to start selecting the saved area, corresponding to BinA, BinB, BinC, BinD 4 storage boxes.



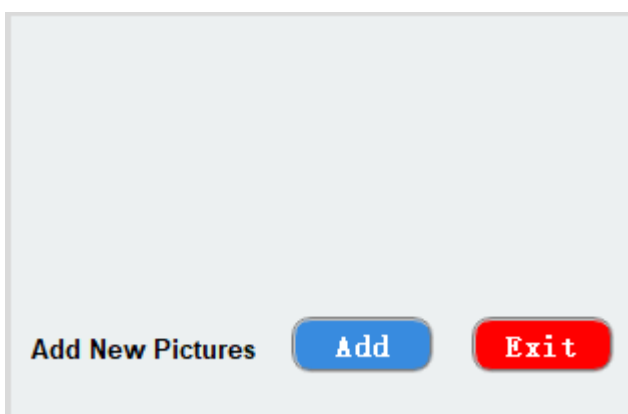
The intercepted content will be displayed here



You can enter the following path to view the saved pictures

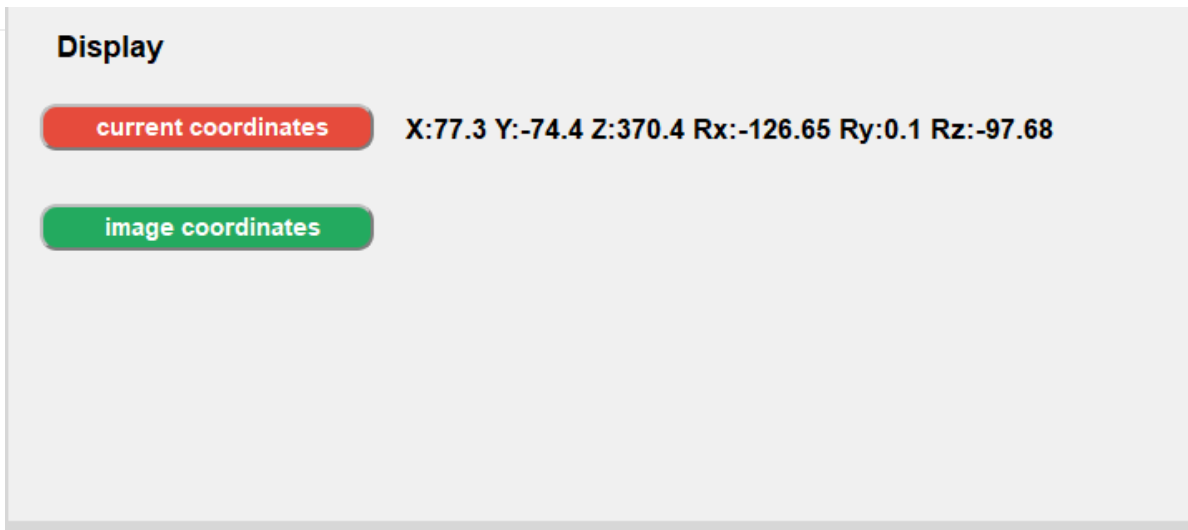


8. Click the 'Exit' button to exit adding pictures. Note: If you start capturing, please exit after capturing. You can choose not to save the captured pictures.

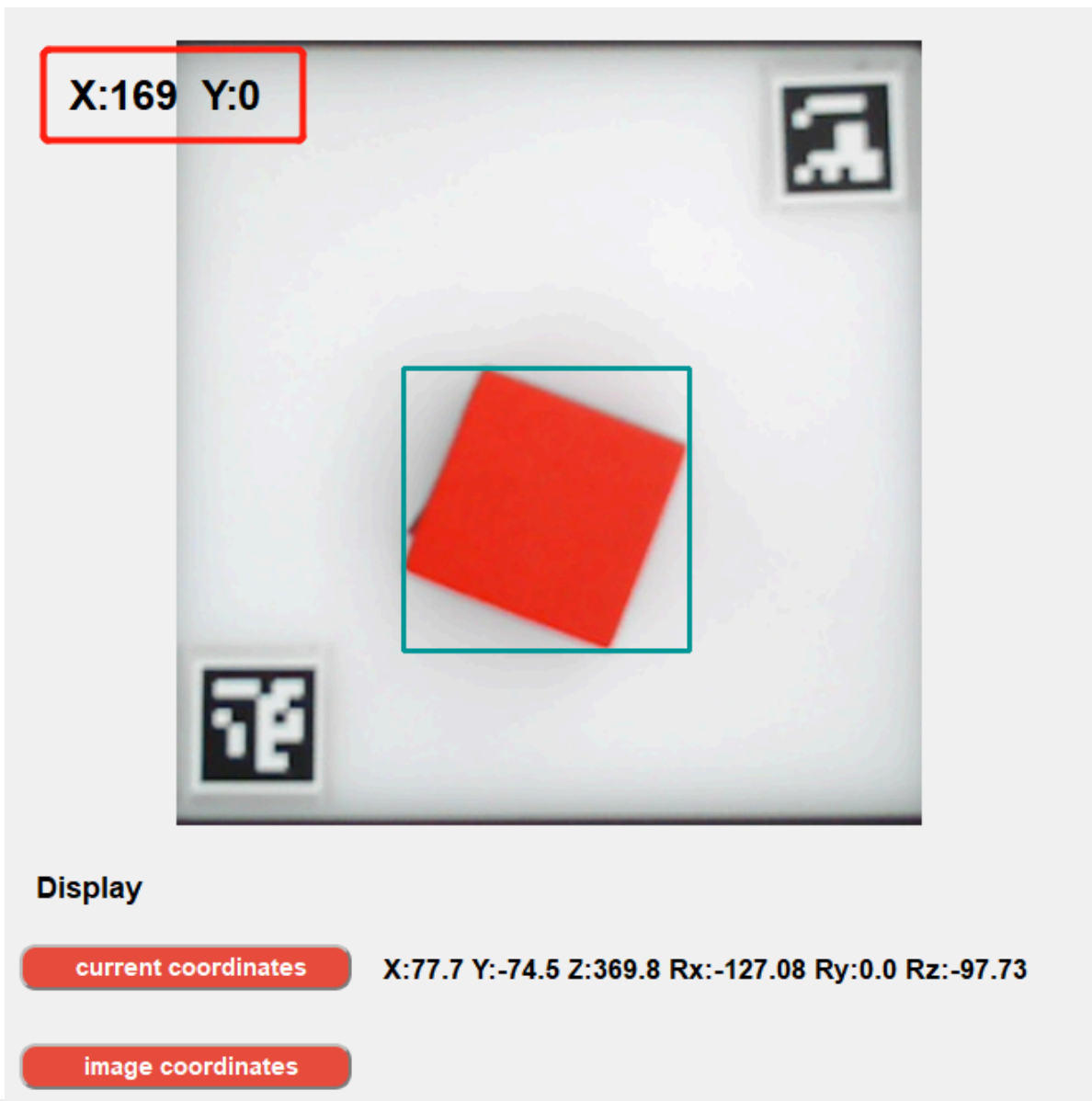


coordinate display

1. Real-time coordinate display of the robotic arm: click the 'current coordinates' button to open



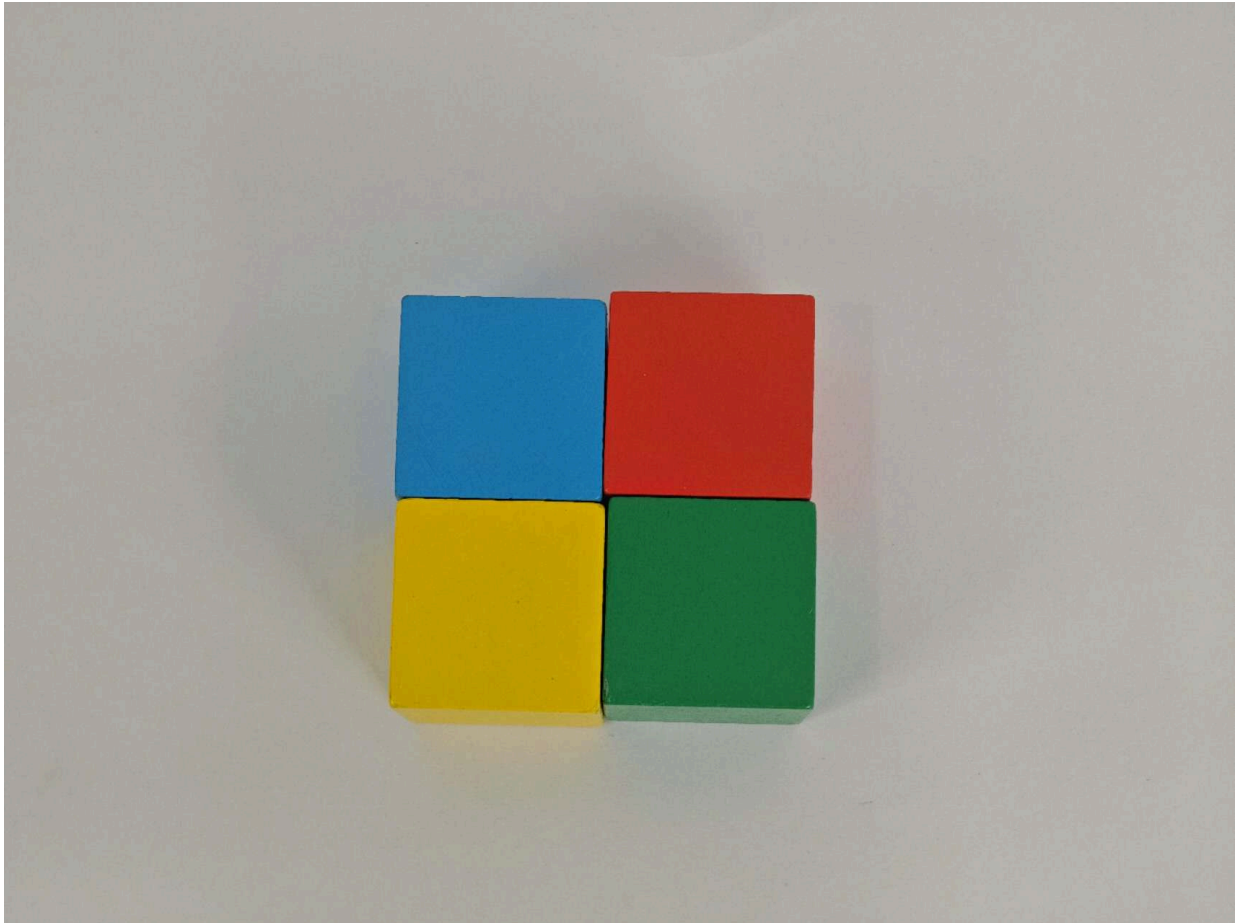
2. Recognition coordinate display: click the "image coordinates" button to open



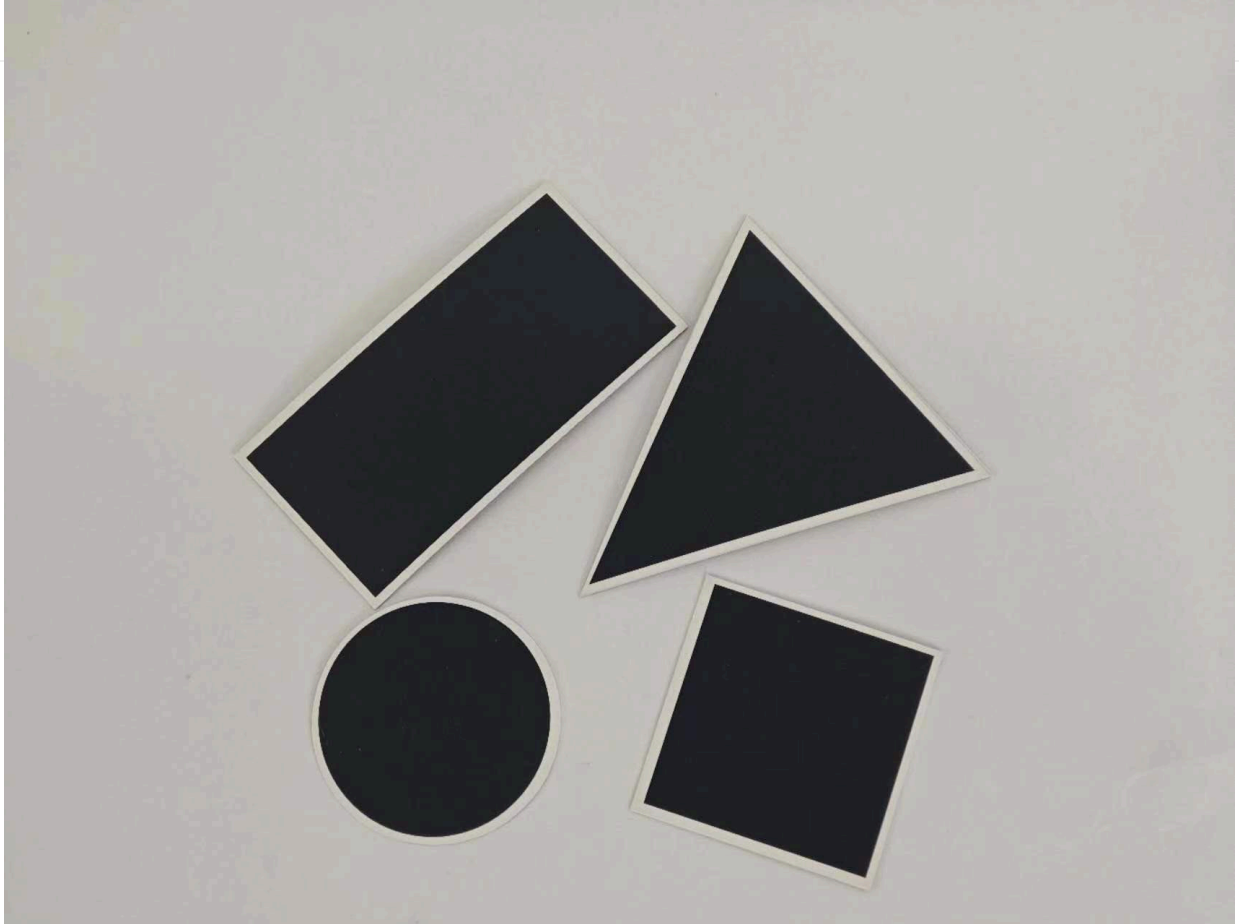
Algorithm material corresponding description

Here is a brief explanation of the material usage corresponding to various algorithms:

- Color recognition:



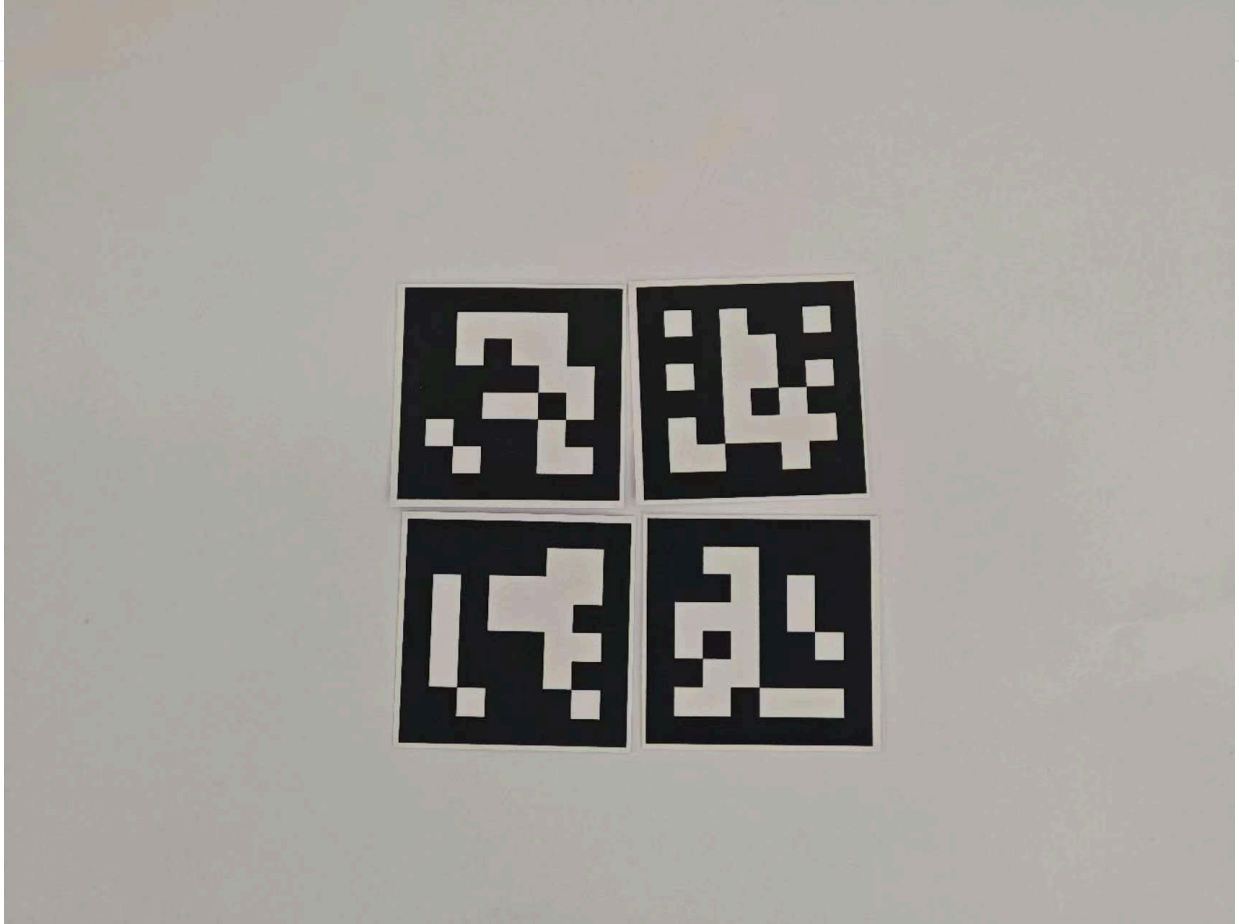
- Shape recognition:



- Feature point recognition:



- QR code recognition:



- yolov5 recogniton:



Recognize Color Blocks

This case uses the eye-to-hand mode, uses the camera, combines **Python+OpenCV**, performs color positioning through OpenCV, and frames the color blocks that meet the situation, and calculates the relative position of the blocks through the relevant points. at the spatial coordinate position of the manipulator. Set up a set of related actions for the robotic arm, and place them in different areas according to the color of the identified objects.

1 Camera Adjustment

First, you need to use python to run OpenVideo.py under the aikit_V2 package. If the enabled camera is a computer camera, cap_num needs to be modified. For details, please refer to: **Precautions. Make sure that the camera completely covers the entire recognition area, and the recognition area is square** in the video, as shown in the figure below. If the recognition area does not meet the requirements in the video, the camera position needs to be adjusted.

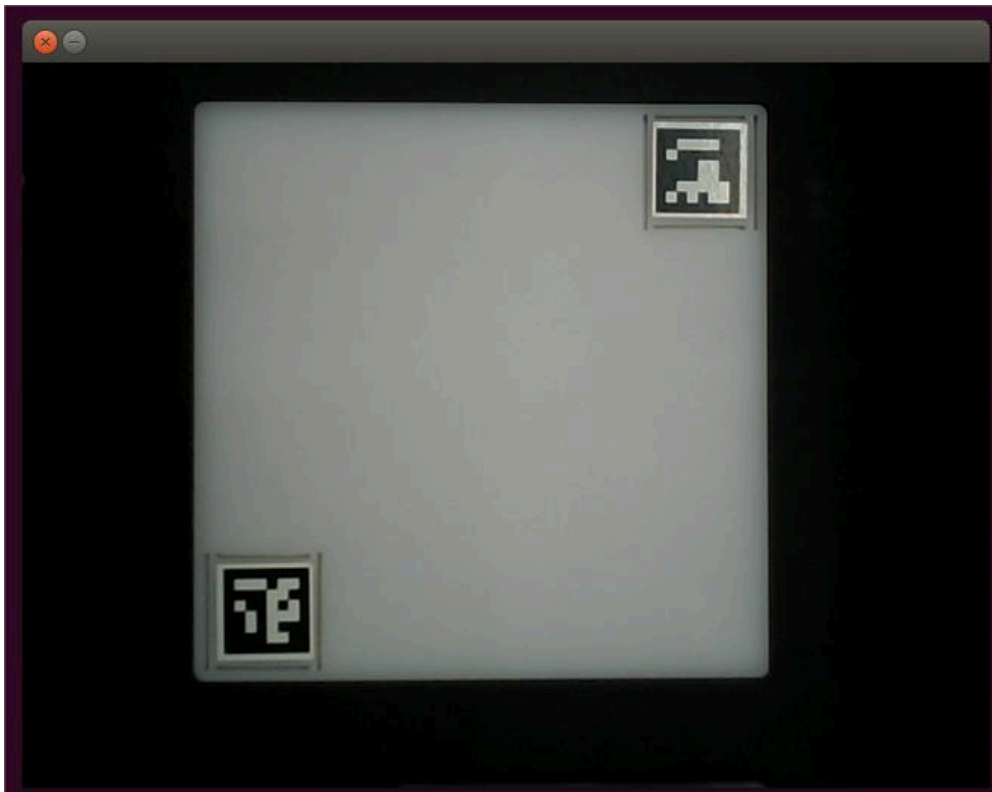
M5 version:

- Open a command prompt terminal (shortcut Win+R, enter "cmd" to access the terminal). Switch to the directory path where "aikit_V2" is located, then enter the target folder.

```
cd aikit_V2/AiKit_280M5/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```



Raspberry Pi version:

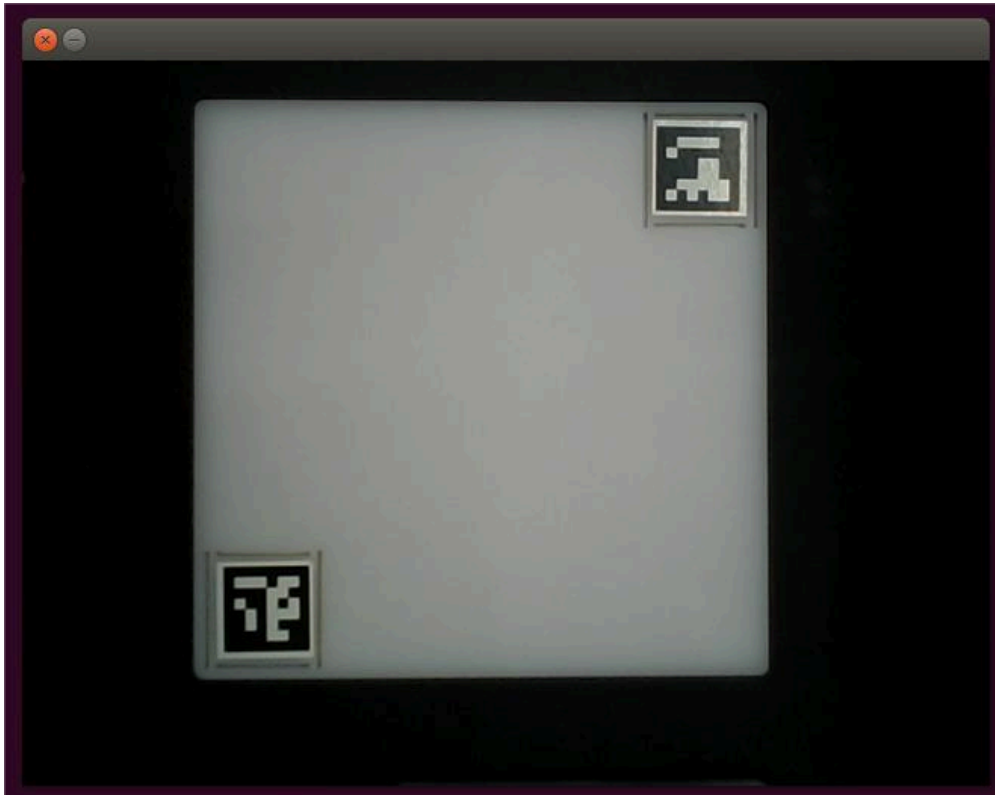
- Open a console terminal (shortcut key Ctrl+Alt+T), enter the target folder

1 Adaptive robot

```
cd ~/aikit_V2/AiKit_280PI/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```



2 Case Reproduction

M5 version:

- Open a command prompt terminal (shortcut Win+R, enter "cmd" to access the terminal). Switch to the directory path where "aikit_V2" is located, then enter the target folder.

```
cd aikit_V2/AiKit_280M5/
```

- Then enter the following command to start the color recognition program.

```
python scripts/aikit_color.py
```

Raspberry Pi version:



- Open a console terminal (shortcut key `Ctrl+Alt+T`), enter the following command to enter the target folder:

```
cd ~/aikit_V2/AiKit_280PI
```

- Then enter the following command to start the color recognition program.

```
python scripts/aikit_color.py
```

- When the command terminal shows `ok` and the camera window can be opened normally, it means that the program runs successfully. At this time, the identifiable objects can be placed in the recognition area for recognition and capture.

Precautions

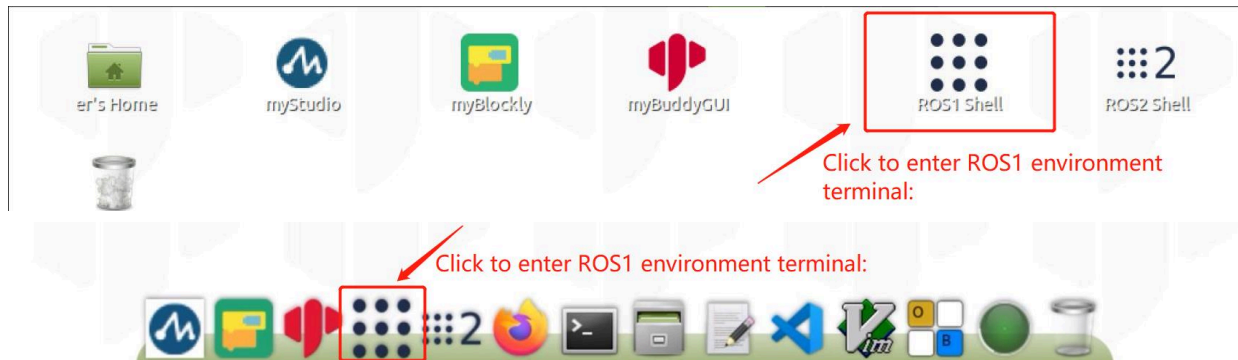
1. When the camera does not automatically frame the recognition area correctly, you need to close the program, adjust the position of the camera, and move the camera to the left or right.
2. If the command terminal does not show `ok`, and the color cannot be recognized, you need to move the camera slightly backward or forward, and the program can run normally when the command terminal shows `ok`.
3. OpenCV color recognition will be affected by the environment, and the recognition effect will be greatly reduced in a darker environment.

Recognize Color Blocks

This case uses the eye-to-hand mode, uses the camera, combines **Python+OpenCV**, performs color positioning through OpenCV, and frames the color blocks that meet the situation, and calculates the relative position of the blocks through the relevant points. at the spatial coordinate position of the manipulator. Set up a set of related actions for the robotic arm, and place them in different areas according to the color of the identified objects.

1 program compilation

Click the `ROS1 Shell` icon on the desktop or the corresponding icon in the lower bar of the desktop to open the ROS1 environment terminal:



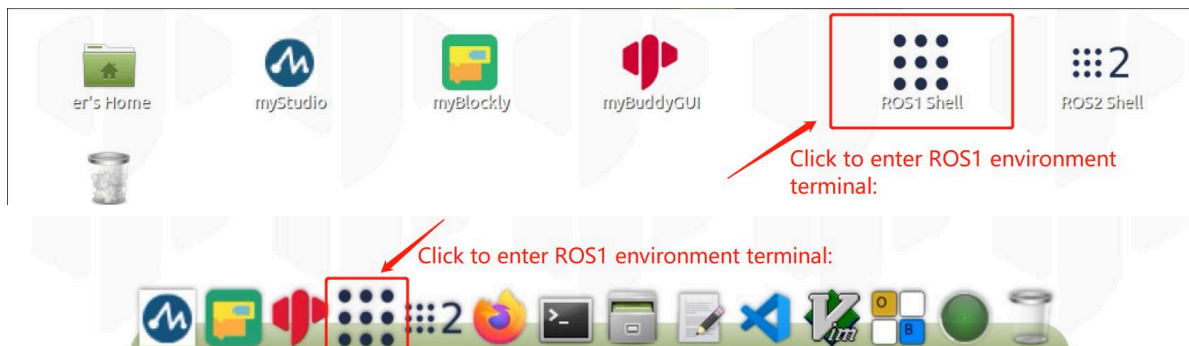
Then enter the following command:

```
cd ~/catkin_ws # Go to the src folder in the workspace
catkin_make # Build the code in the workspace
source devel/setup.bash # Add environment variables
```

2 Camera Adjustment

First, you need to use python to run `OpenVideo.py` under the `aikit_V2` package. If the enabled camera is a computer camera, `cap_num` needs to be modified. For details, please refer to: **Precautions. Make sure that the camera completely covers the entire recognition area, and the recognition area is square** in the video, as shown in the figure below. If the recognition area does not meet the requirements in the video, the camera position needs to be adjusted.

- Click the `ROS1 Shell` icon on the desktop or the corresponding icon in the lower bar of the desktop to open the ROS1 environment terminal, and enter the following command to enter the target folder:

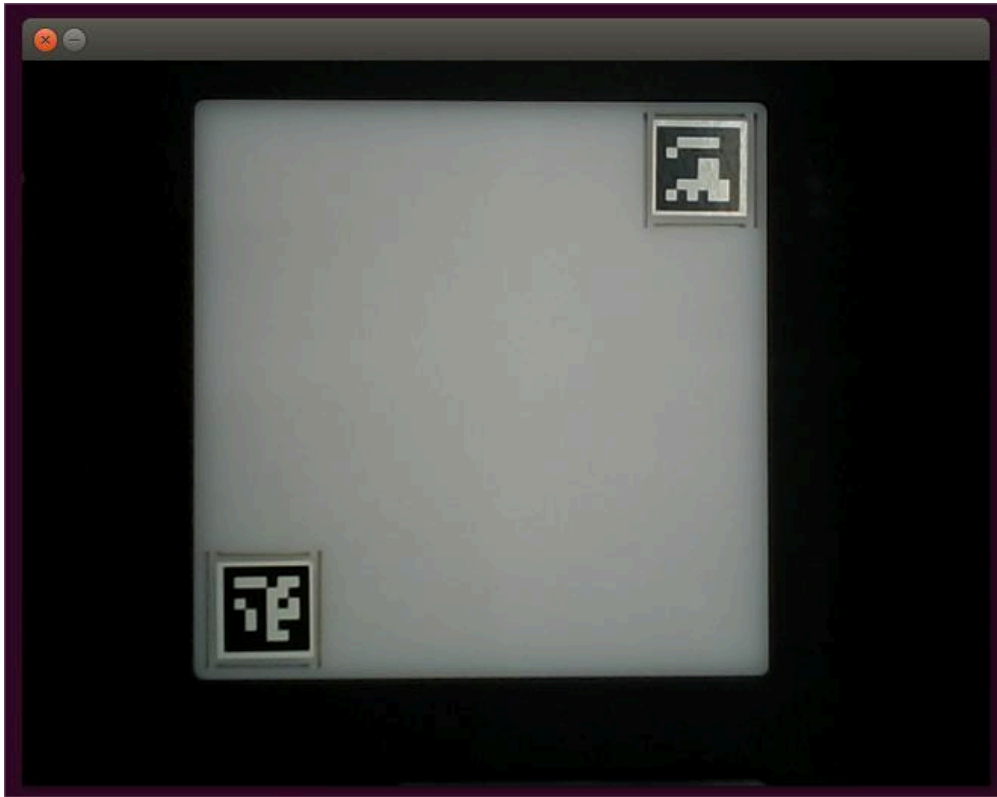


1 Adaptive robot

```
cd ~/catkin_ws/src/mycobot_ros/mycobot_ai/aikit_280_pi/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```



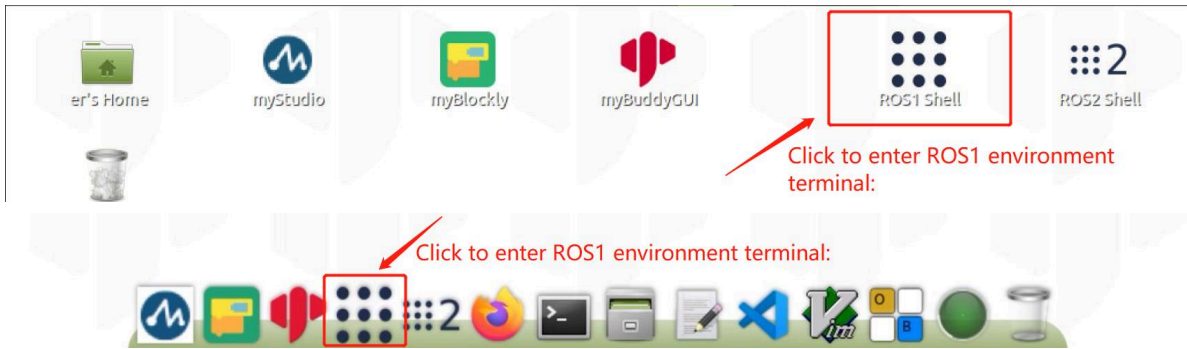
3 Case Reproduction

Raspberry Pi version:



1 Adaptive robot

- Click the ROS1 Shell icon on the desktop or the corresponding icon in the lower bar of the desktop to open the ROS1 environment terminal, and enter the following command to start the launch file:



```
roslaunch aikit_280_pi vision_pi.launch
```

- Open another ROS1 environment terminal (the opening method is the same as above), enter the following command to enter the target folder:

```
cd ~/catkin_ws/src/mycobot_ros/mycobot_ai/aikit_280_pi/
```

- Then enter the following command to start the color recognition program:

```
python scripts/aikit_color.py
```

- When the command terminal shows ok and the camera window can be opened normally, it means that the program runs successfully. At this time, the identifiable objects can be placed in the recognition area for recognition and capture.

Precautions

- When the camera does not automatically frame the recognition area correctly, you need to close the program, adjust the position of the camera, and move the camera to the left or right.
- If the command terminal does not show ok, and the color cannot be recognized, you need to move the camera slightly backward or forward, and the program can run normally when the command terminal shows ok.
- OpenCV color recognition will be affected by the environment, and the recognition effect will be greatly reduced in a darker environment.

Recognize Color Blocks

This case uses the eye-to-hand mode, uses the camera, combines **Python+OpenCV**, performs color positioning through OpenCV, and frames the color blocks that meet the situation, and calculates the relative position of the blocks through the relevant points. at the spatial coordinate position of the manipulator. Set up a set of related actions for the robotic arm, and place them in different areas according to the color of the identified objects.

1 Camera Adjustment

First, you need to use python to run OpenVideo.py under the aikit_V2 package. If the enabled camera is a computer camera, cap_num needs to be modified. For details, please refer to: **Precautions. Make sure that the camera completely covers the entire recognition area, and the recognition area is square** in the video, as shown in the figure below. If the recognition area does not meet the requirements in the video, the camera position needs to be adjusted.

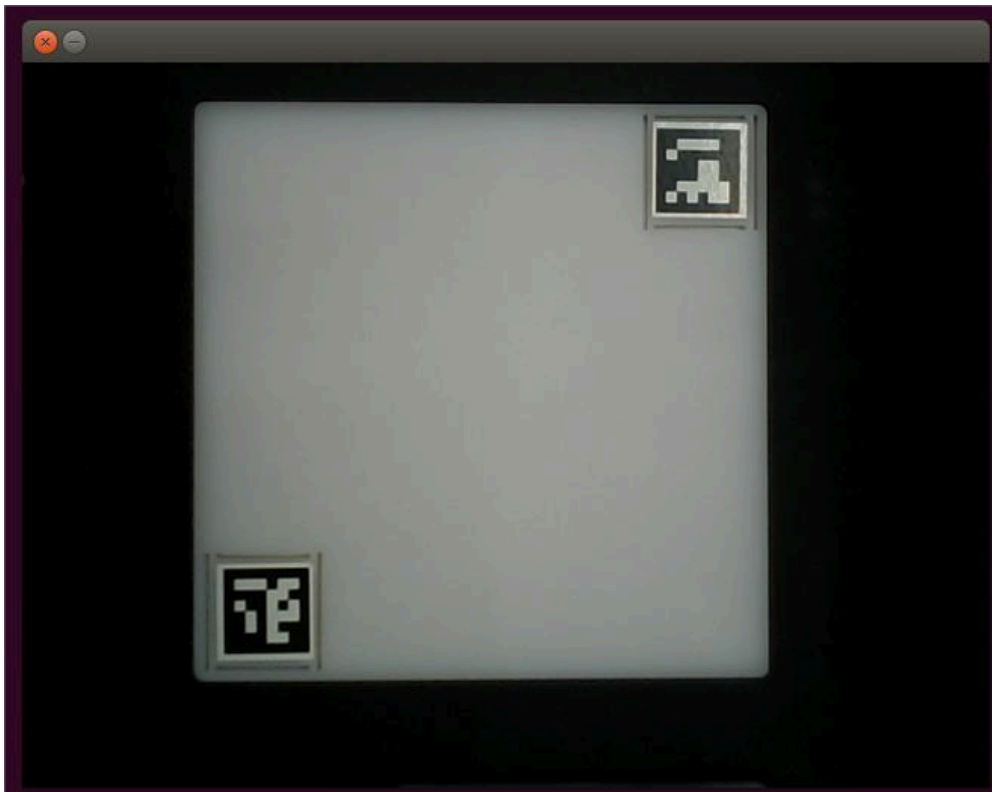
M5 version:

- Open a command prompt terminal (shortcut Win+R, enter "cmd" to access the terminal). Switch to the directory path where "aikit_V2" is located, then enter the target folder.

```
cd aikit_V2/AiKit_270M5/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```



Raspberry Pi version:

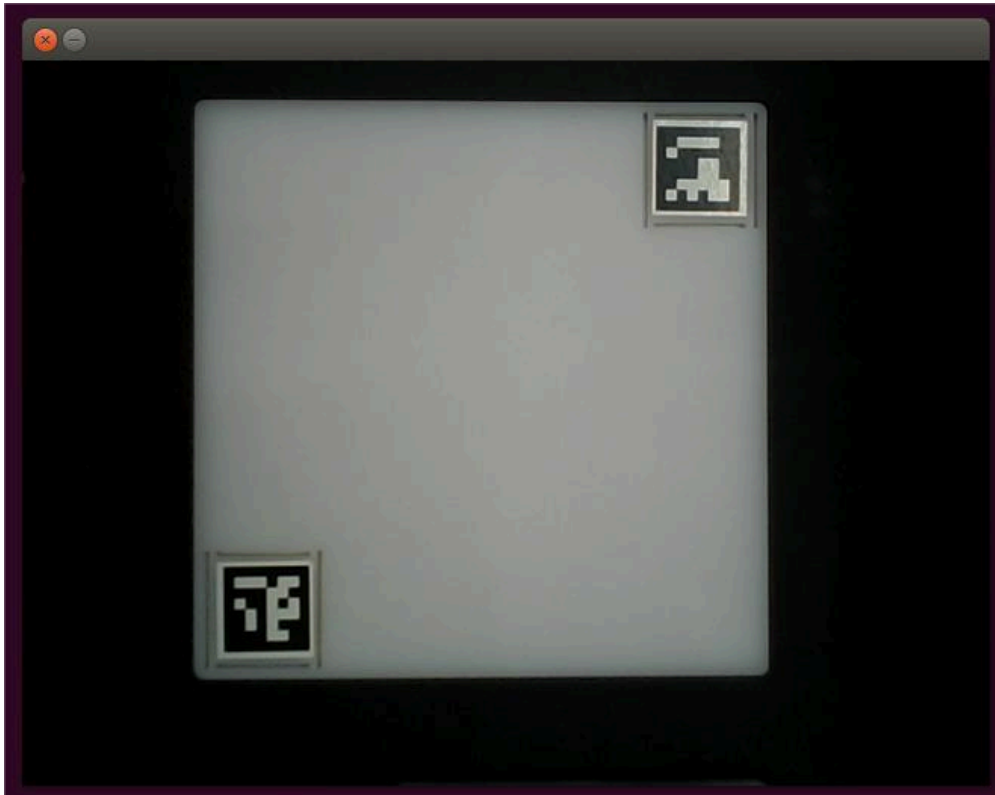
- Open a console terminal (shortcut key Ctrl+Alt+T), enter the target folder

1 Adaptive robot

```
cd ~/aikit_V2/AiKit_270Pi/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```



2 Case Reproduction

M5 version:

- Open a command prompt terminal (shortcut Win+R, enter "cmd" to access the terminal). Switch to the directory path where "aikit_V2" is located, then enter the target folder.

```
cd aikit_V2/AiKit_270M5/
```

- Then enter the following command to start the color recognition program.

```
python scripts/aikit_color.py
```

Raspberry Pi version:



- Open a console terminal (shortcut key `Ctrl+Alt+T`), enter the following command to enter the target folder:

```
cd ~/aikit_V2/AiKit_270Pi
```

- Then enter the following command to start the color recognition program.

```
python scripts/aikit_color.py
```

- When the command terminal shows `ok` and the camera window can be opened normally, it means that the program runs successfully. At this time, the identifiable objects can be placed in the recognition area for recognition and capture.

Precautions

1. When the camera does not automatically frame the recognition area correctly, you need to close the program, adjust the position of the camera, and move the camera to the left or right.
2. If the command terminal does not show `ok`, and the color cannot be recognized, you need to move the camera slightly backward or forward, and the program can run normally when the command terminal shows `ok`.
3. OpenCV color recognition will be affected by the environment, and the recognition effect will be greatly reduced in a darker environment.

Recognize Color Blocks

This case uses the eye-to-hand mode, uses the camera, combines **Python+OpenCV**, performs color positioning through OpenCV, and frames the color blocks that meet the situation, and calculates the relative position of the blocks through the relevant points. at the spatial coordinate position of the manipulator. Set up a set of related actions for the robotic arm, and place them in different areas according to the color of the identified objects.

1 Camera Adjustment

First, you need to use python to run OpenVideo.py under the aikit_V2 package. If the enabled camera is a computer camera, cap_num needs to be modified. For details, please refer to: **Precautions. Make sure that the camera completely covers the entire recognition area, and the recognition area is square** in the video, as shown in the figure below. If the recognition area does not meet the requirements in the video, the camera position needs to be adjusted.

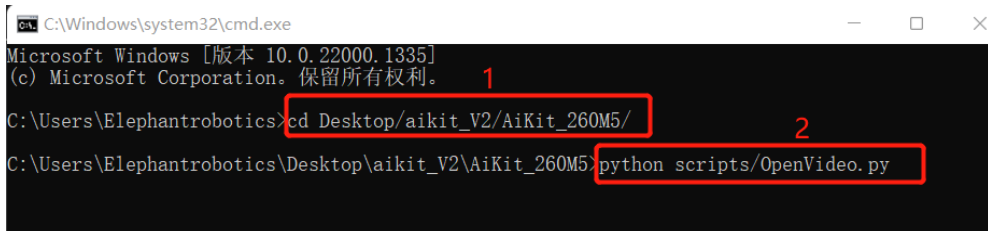
M5 version:

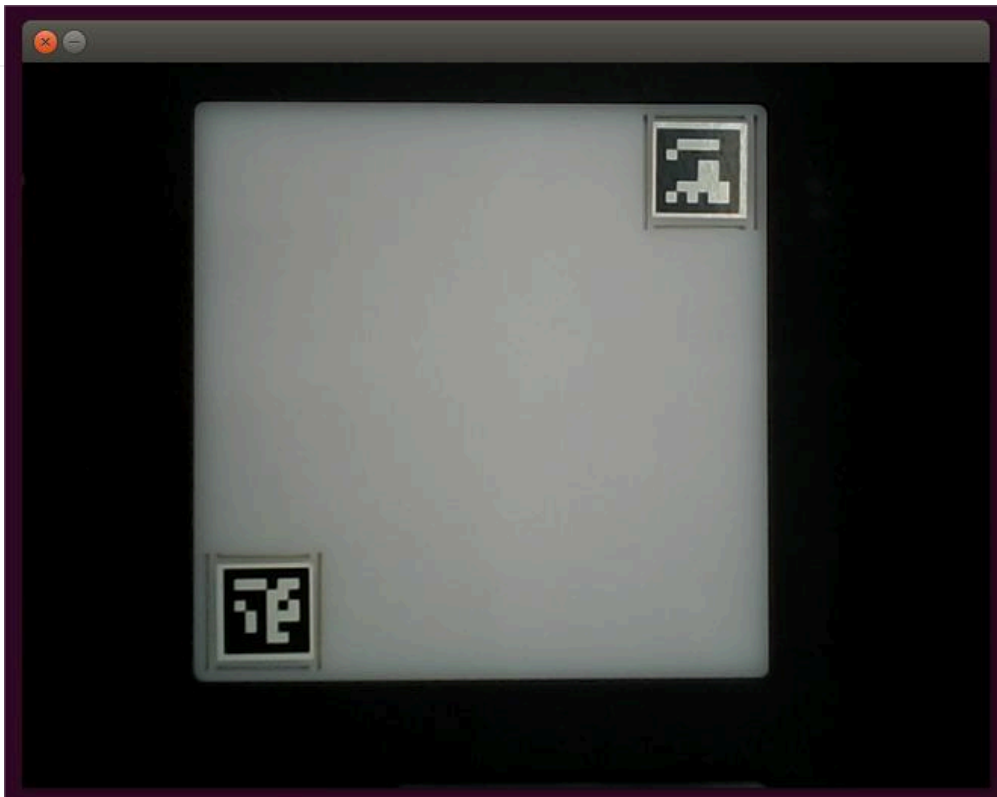
- Open a command prompt terminal (shortcut Win+R, enter "cmd" to access the terminal). Switch to the directory path where "aikit_V2" is located, then enter the target folder.

```
cd Desktop/aikit_V2/AiKit_260M5/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```





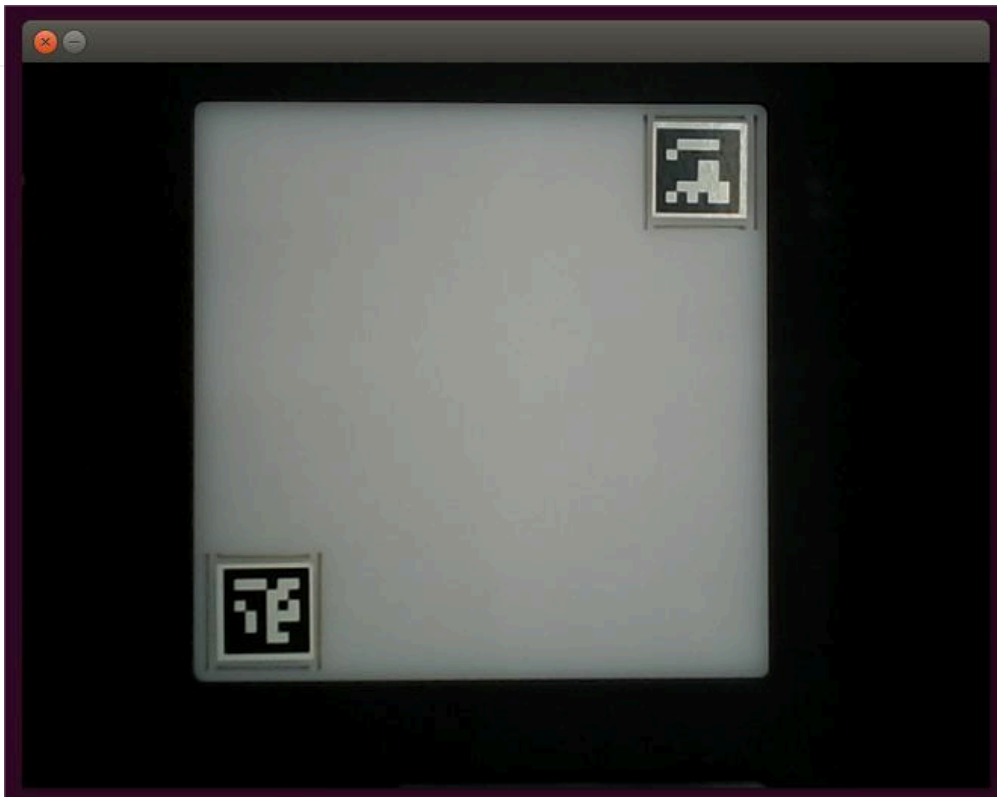
Raspberry Pi version:

- Open a console terminal (shortcut key `Ctrl+Alt+T`), enter the following command to enter the target folder:

```
cd ~/aikit_V2/AiKit_260PI/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```

2 Case Reproduction

M5 version:

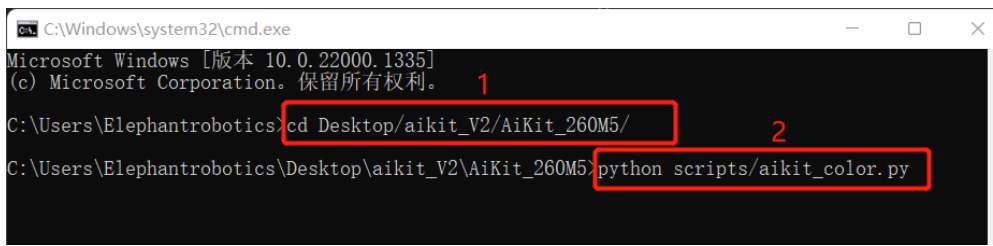


- Open a console terminal (shortcut key Win+R, enter cmd to enter the terminal), enter the following command to enter the target folder:

```
cd Desktop/aikit_V2/AiKit_260M5/
```

- Then enter the following command to start the color recognition program.

```
python scripts/aikit_color.py
```



Raspberry Pi version:

- Open a console terminal (shortcut key `Ctrl+Alt+T`), enter the following command to enter the target folder:

```
cd ~/aikit_V2/AiKit_260PI/
```

- Then enter the following command to start the color recognition program.

```
python scripts/aikit_color.py
```

- When the command terminal shows `ok` and the camera window can be opened normally, it means that the program runs successfully. At this time, the identifiable objects can be placed in the recognition area for recognition and capture.

Precautions

1. When the camera does not automatically frame the recognition area correctly, you need to close the program, adjust the position of the camera, and move the camera to the left or right.
2. If the command terminal does not show `ok`, and the color cannot be recognized, you need to move the camera slightly backward or forward, and the program can run normally when the command terminal shows `ok`.
3. OpenCV color recognition will be affected by the environment, and the recognition effect will be greatly reduced in a darker environment.

Recognize Shape Cards

In this case, the eye-to-hand mode is used, the camera is combined with **Python+OpenCV**, the shape is positioned through OpenCV, and the shape that meets the situation is framed, and the relative position of the object to the machine is calculated through the relevant points. The spatial coordinate position of the arm. Set a set of related actions for the robot arm, and place them in different areas according to the recognized shape.

1 Camera Adjustment

First, you need to use python to run OpenVideo.py under the aikit_V2 package. If the enabled camera is a computer camera, cap_num needs to be modified. For details, please refer to: **Precautions. Make sure that the camera completely covers the entire recognition area, and the recognition area is square** in the video, as shown in the figure below. If the recognition area does not meet the requirements in the video, the camera position needs to be adjusted.

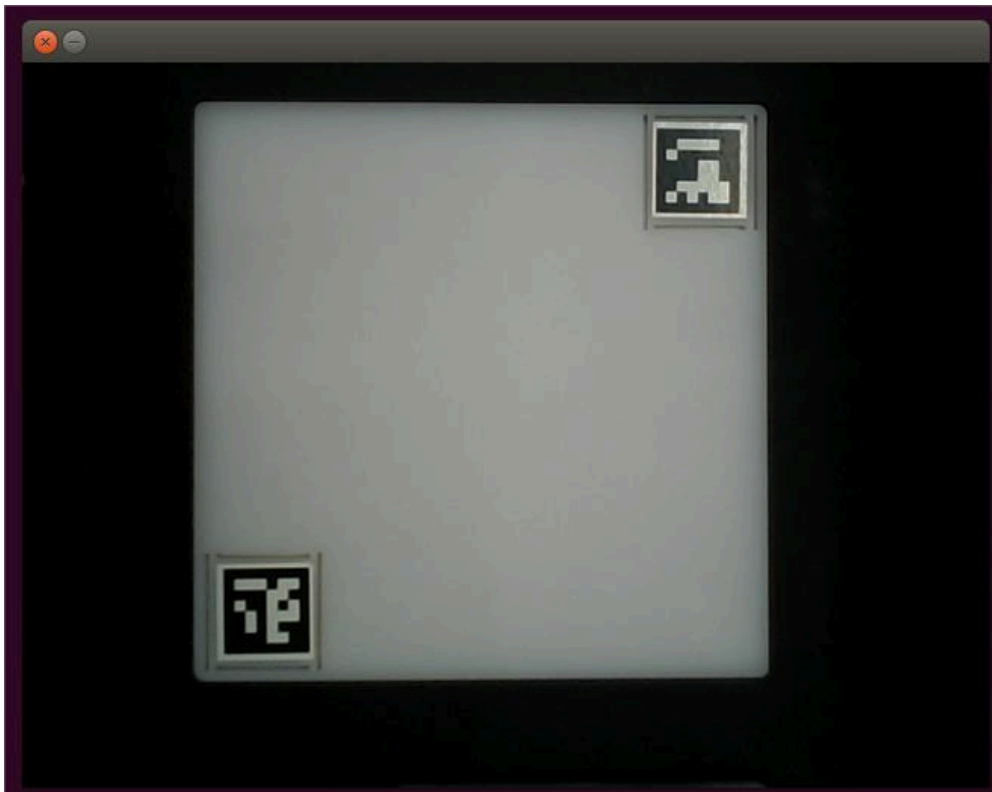
M5 version:

- Open a command prompt terminal (shortcut Win+R, enter "cmd" to access the terminal). Switch to the directory path where "aikit_V2" is located, then enter the target folder.

```
cd Desktop/aikit_V2/AiKit_280M5/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```



Raspberry Pi version:

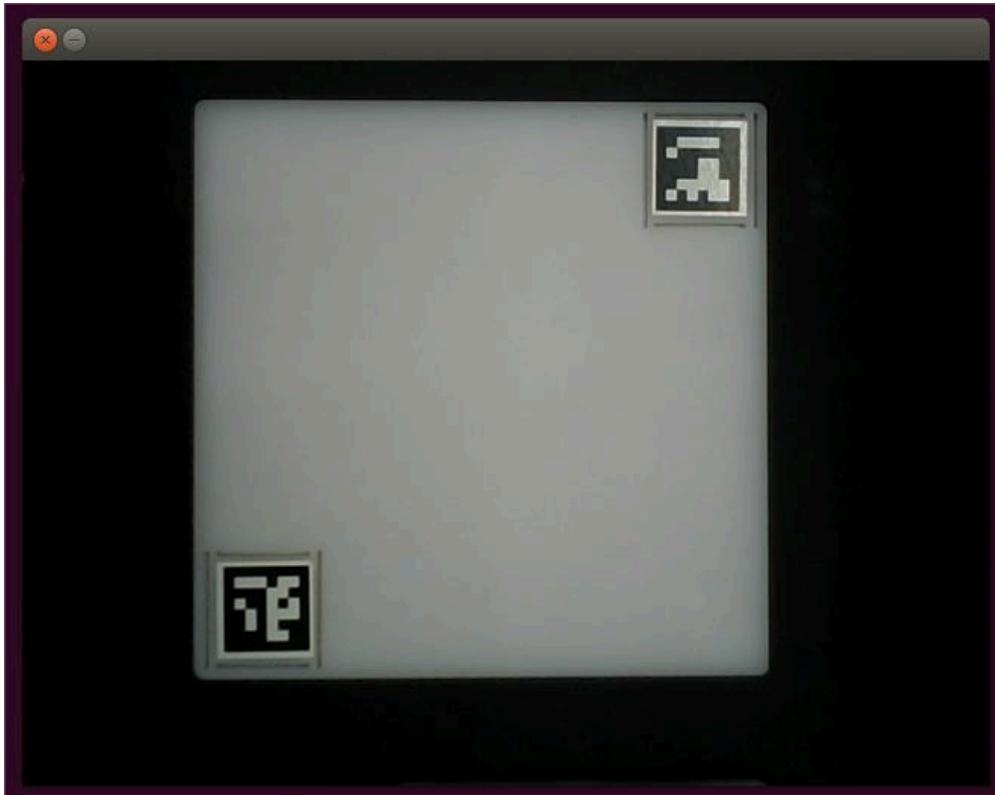
- Open a console terminal (shortcut key Ctrl+Alt+T), enter the target folder

1 Adaptive robot

```
cd ~/aikit_V2/AiKit_280PI/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```



2 Case Reproduction

M5 version:

- Open a command prompt terminal (shortcut Win+R, enter "cmd" to access the terminal). Switch to the directory path where "aikit_V2" is located, then enter the target folder.

```
cd Desktop/aikit_V2/AiKit_280M5/
```

- Then enter the following command to start the shape recognition program.

```
python scripts/aikit_shape.py
```

Raspberry Pi version:



- Open a console terminal (shortcut key `Ctrl+Alt+T`), enter the following command to enter the target folder:

```
cd ~/aikit_V2/AiKit_280PI
```

- Then enter the following command to start the shape recognition program.

```
python scripts/aikit_shape.py
```

- When the command terminal shows `ok` and the camera window can be opened normally, it means that the program runs successfully. At this time, the identifiable objects can be placed in the recognition area for recognition and capture.

Precautions

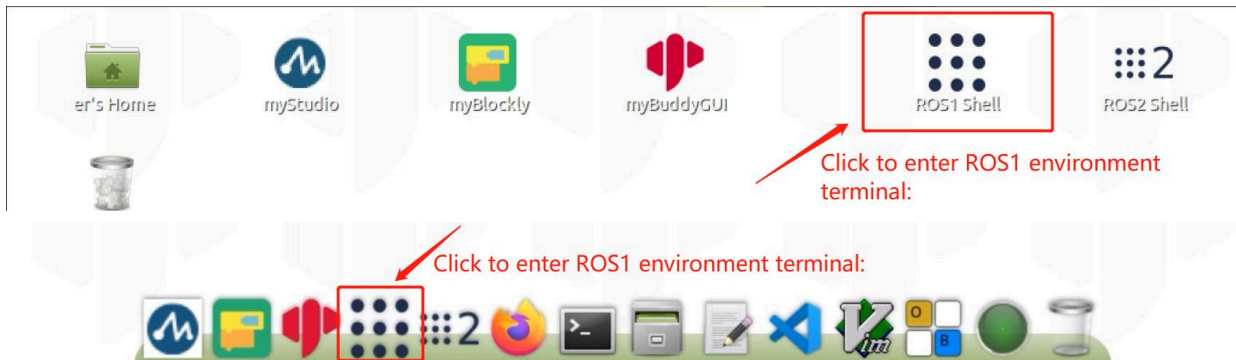
1. When the camera does not automatically frame the recognition area correctly, you need to close the program, adjust the position of the camera, and move the camera to the left or right.
2. If the command terminal does not show `ok`, and the color cannot be recognized, you need to move the camera slightly backward or forward, and the program can run normally when the command terminal shows `ok`.
3. OpenCV color recognition will be affected by the environment, and the recognition effect will be greatly reduced in a darker environment.

Recognize Shape Cards

This case uses the eye-to-hand mode, uses the camera, combines **Python+OpenCV+ROS**, performs shape positioning through OpenCV, and frames the shape that meets the situation, and calculates the relative position of the object through the relevant points. at the spatial coordinate position of the manipulator. Set a set of related actions for the robot arm, and place them in different areas according to the recognized shape.

1 program compilation

Click the `ROS1 Shell` icon on the desktop or the corresponding icon in the lower bar of the desktop to open the ROS1 environment terminal:



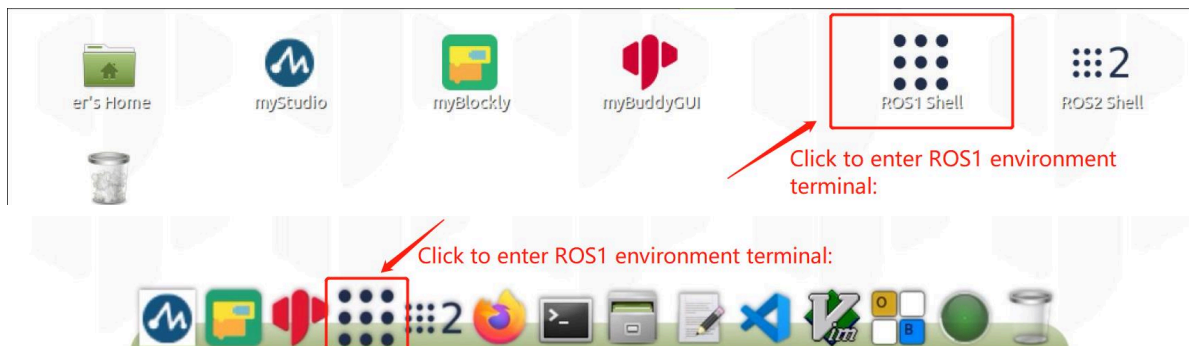
Then enter the following command:

```
cd ~/catkin_ws # Go to the src folder in the workspace
catkin_make # Build the code in the workspace
source devel/setup.bash # Add environment variables
```

2 Camera Adjustment

First, you need to use python to run `OpenVideo.py` under the `aikit_V2` package. If the enabled camera is a computer camera, `cap_num` needs to be modified. For details, please refer to: **Precautions. Make sure that the camera completely covers the entire recognition area, and the recognition area is square** in the video, as shown in the figure below. If the recognition area does not meet the requirements in the video, the camera position needs to be adjusted.

- Click the `ROS1 Shell` icon on the desktop or the corresponding icon in the lower bar of the desktop to open the ROS1 environment terminal, and enter the following command to enter the target folder:

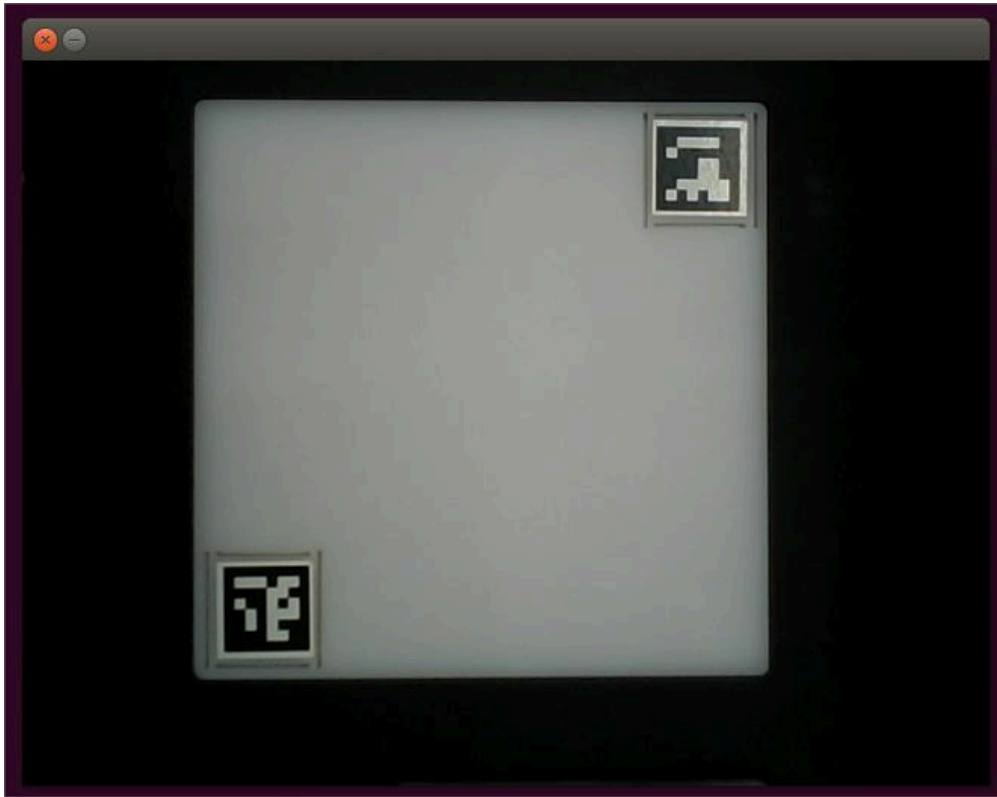


1 Adaptive robot

```
cd ~/catkin_ws/src/mycobot_ros/mycobot_ai/aikit_280_pi/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```



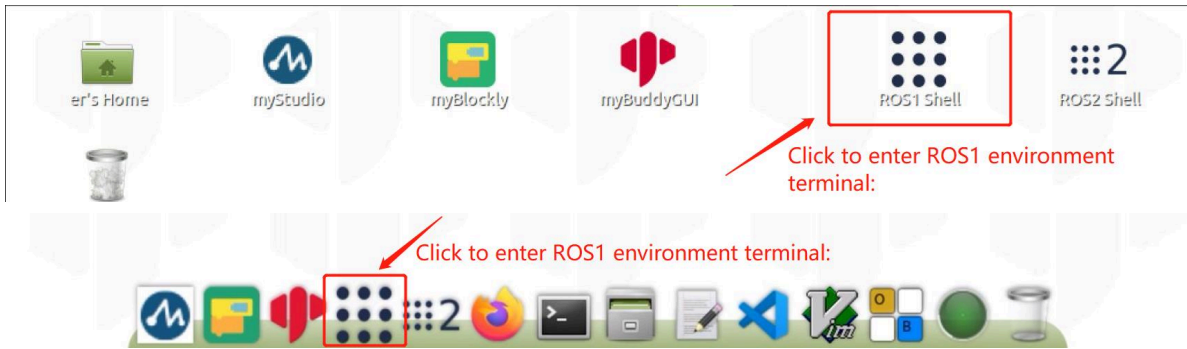
3 Case Reproduction

Raspberry Pi version:



1 Adaptive robot

- Click the ROS1 Shell icon on the desktop or the corresponding icon in the lower bar of the desktop to open the ROS1 environment terminal, and enter the following command to start the launch file:



```
roslaunch aikit_280_pi vision_pi.launch
```

- Open another ROS1 environment terminal (the opening method is the same as above), enter the following command to enter the target folder:

```
cd ~/catkin_ws/src/mycobot_ros/mycobot_ai/aikit_280_pi/
```

- Then enter the following command to start the color recognition program:

```
python scripts/aikit_shape.py
```

- When the command terminal shows ok and the camera window can be opened normally, it means that the program runs successfully. At this time, the identifiable objects can be placed in the recognition area for recognition and capture.

Precautions

- When the camera does not automatically frame the recognition area correctly, you need to close the program, adjust the position of the camera, and move the camera to the left or right.
- If the command terminal does not show ok, and the color cannot be recognized, you need to move the camera slightly backward or forward, and the program can run normally when the command terminal shows ok.
- OpenCV color recognition will be affected by the environment, and the recognition effect will be greatly reduced in a darker environment.

Recognize Shape Cards

In this case, the eye-to-hand mode is used, the camera is combined with **Python+OpenCV**, the shape is positioned through OpenCV, and the shape that meets the situation is framed, and the relative position of the object to the machine is calculated through the relevant points. The spatial coordinate position of the arm. Set a set of related actions for the robot arm, and place them in different areas according to the recognized shape.

1 Camera Adjustment

First, you need to use python to run OpenVideo.py under the aikit_V2 package. If the enabled camera is a computer camera, cap_num needs to be modified. For details, please refer to: **Precautions. Make sure that the camera completely covers the entire recognition area, and the recognition area is square** in the video, as shown in the figure below. If the recognition area does not meet the requirements in the video, the camera position needs to be adjusted.

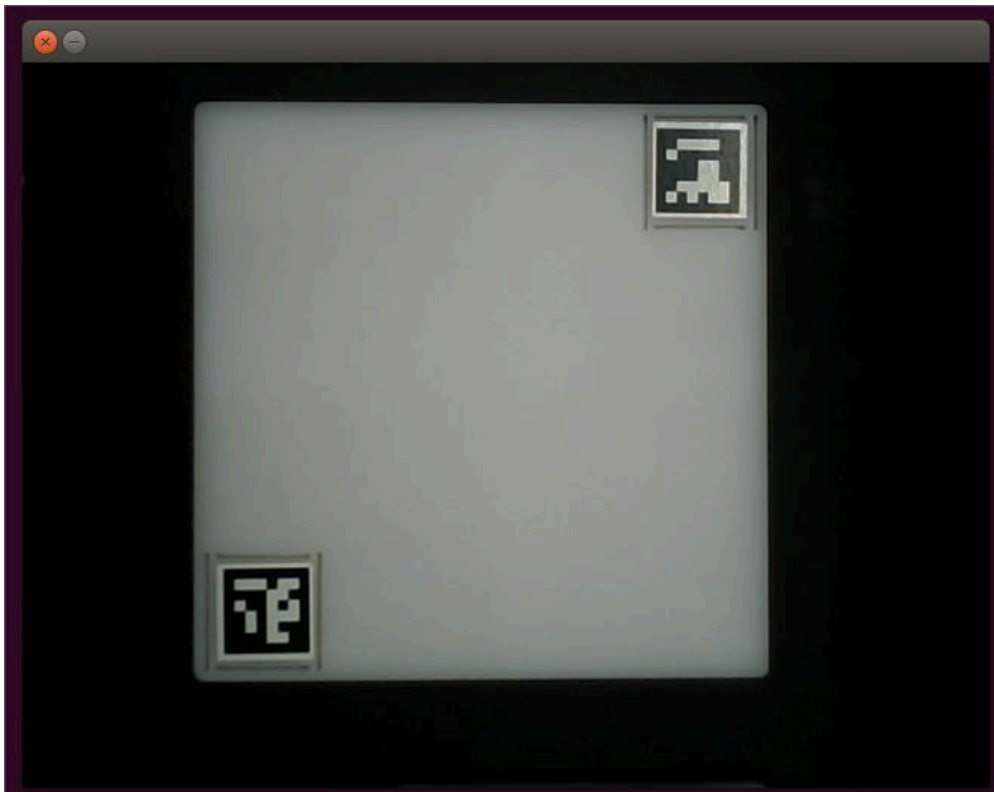
M5 version:

- Open a command prompt terminal (shortcut Win+R, enter "cmd" to access the terminal). Switch to the directory path where "aikit_V2" is located, then enter the target folder.

```
cd Desktop/aikit_V2/AiKit_270M5/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```



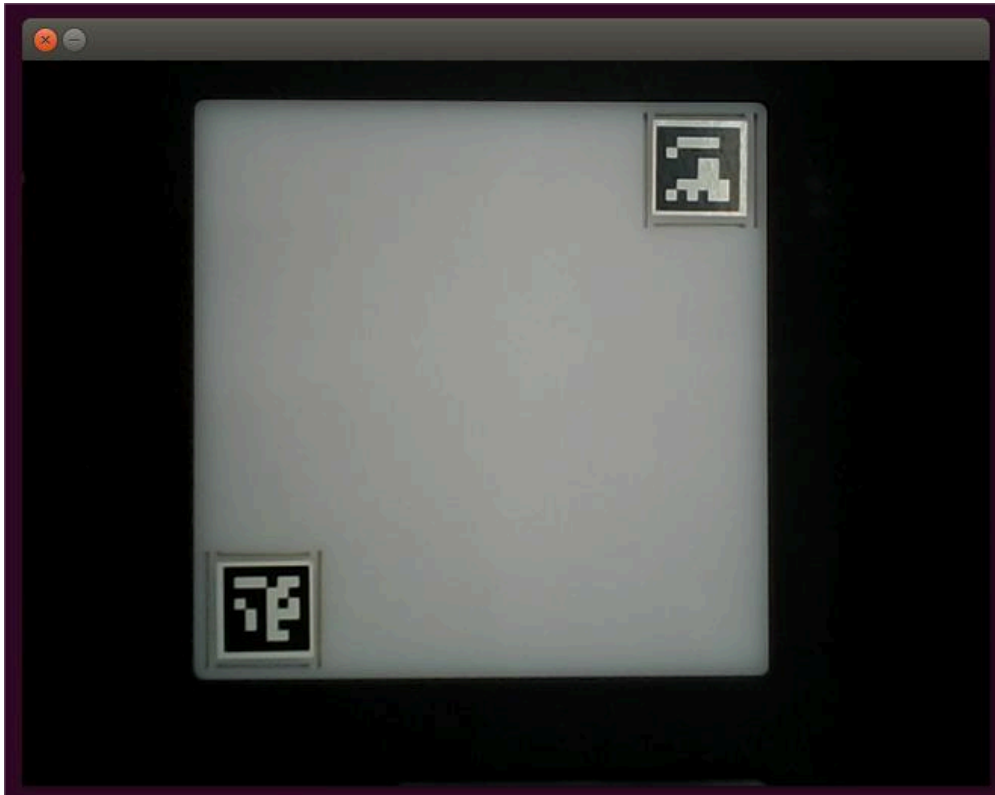
- Open a console terminal (shortcut key Ctrl+Alt+T), enter the target folder

1 Adaptive robot

```
cd ~/aikit_V2/AiKit_270Pi/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```



2 Case Reproduction

M5 version:

- Open a command prompt terminal (shortcut Win+R, enter "cmd" to access the terminal). Switch to the directory path where "aikit_V2" is located, then enter the target folder.

```
cd Desktop/aikit_V2/AiKit_270M5/
```

- Then enter the following command to start the color recognition program.

```
python scripts/aikit_shape.py
```

Raspberry Pi version:



- Open a console terminal (shortcut key `Ctrl+Alt+T`), enter the following command to enter the target folder:

```
cd ~/aikit_V2/AiKit_270Pi
```

- Then enter the following command to start the color recognition program.

```
python scripts/aikit_shape.py
```

- When the command terminal shows `ok` and the camera window can be opened normally, it means that the program runs successfully. At this time, the identifiable objects can be placed in the recognition area for recognition and capture.

Precautions

1. When the camera does not automatically frame the recognition area correctly, you need to close the program, adjust the position of the camera, and move the camera to the left or right.
2. If the command terminal does not show `ok`, and the color cannot be recognized, you need to move the camera slightly backward or forward, and the program can run normally when the command terminal shows `ok`.
3. OpenCV color recognition will be affected by the environment, and the recognition effect will be greatly reduced in a darker environment.

Recognize Shape Cards

In this case, the eye-to-hand mode is used, the camera is combined with **Python+OpenCV**, the shape is positioned through OpenCV, and the shape that meets the situation is framed, and the relative position of the object to the machine is calculated through the relevant points. The spatial coordinate position of the arm. Set a set of related actions for the robot arm, and place them in different areas according to the recognized shape.

1 Camera Adjustment

First, you need to use python to run OpenVideo.py under the aikit_V2 package. If the enabled camera is a computer camera, cap_num needs to be modified. For details, please refer to: **Precautions. Make sure that the camera completely covers the entire recognition area, and the recognition area is square** in the video, as shown in the figure below. If the recognition area does not meet the requirements in the video, the camera position needs to be adjusted.

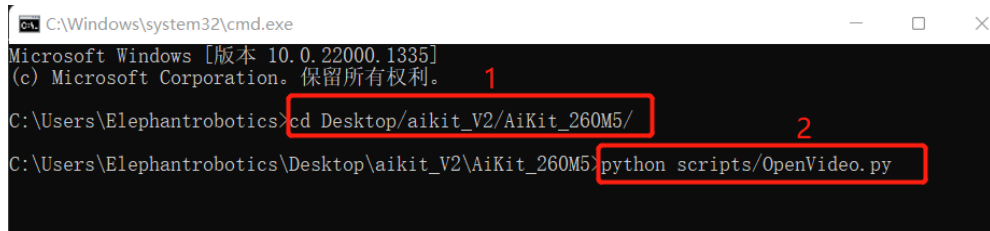
M5 version:

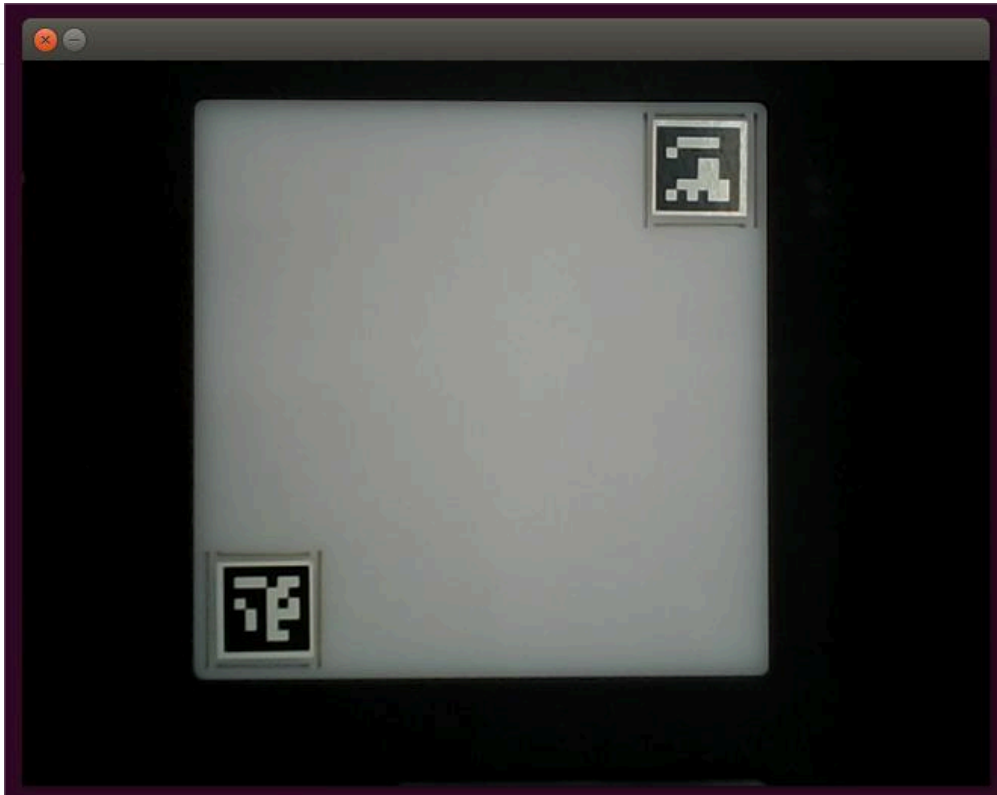
- Open a console terminal (shortcut key Win+R, enter cmd to enter the terminal), enter the following command to enter the target folder:

```
cd Desktop/aikit_V2/AiKit_260M5/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```





2 Case Reproduction

M5 version:

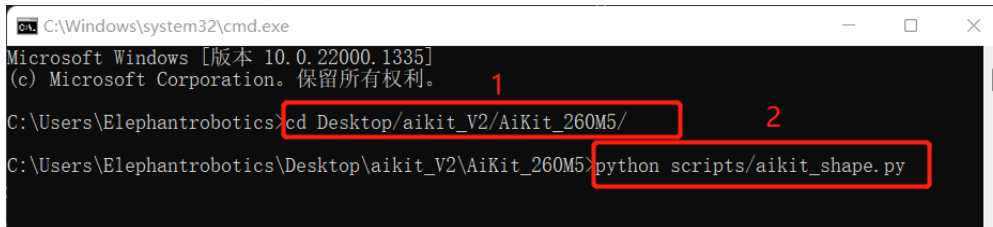


- Open a console terminal (shortcut key Win+R, enter cmd to enter the terminal), enter the following command to enter the target folder:

```
cd Desktop/aikit_V2/AiKit_260M5/
```

- Then enter the following command to start the color recognition program.

```
python scripts/aikit_shape.py
```



Raspberry Pi version:

- Open a console terminal (shortcut key `Ctrl+Alt+T`), enter the following command to enter the target folder:

```
cd ~/aiKit_V2/AiKit_260PI/
```

- Then enter the following command to start the color recognition program.

```
python scripts/aikit_shape.py
```

- When the command terminal shows `ok` and the camera window can be opened normally, it means that the program runs successfully. At this time, the identifiable objects can be placed in the recognition area for recognition and capture.

Precautions

1. When the camera does not automatically frame the recognition area correctly, you need to close the program, adjust the position of the camera, and move the camera to the left or right.
2. If the command terminal does not show `ok`, and the color cannot be recognized, you need to move the camera slightly backward or forward, and the program can run normally when the command terminal shows `ok`.
3. OpenCV color recognition will be affected by the environment, and the recognition effect will be greatly reduced in a darker environment.

Identify AR code

This case uses the eye-to-hand mode, uses the camera, combines **Python+OpenCV**, performs attitude estimation through the ArUco marker in OpenCV, and frames the AR code that meets the situation, and calculates it through the relevant points. The spatial coordinate position of the AR code relative to the robotic arm. Set a set of related actions for the robotic arm, and place them in different areas according to the recognized AR code.

1 Camera Adjustment

First, you need to use python to run OpenVideo.py under the aikit_V2 package. If the enabled camera is a computer camera, cap_num needs to be modified. For details, please refer to: **Precautions. Make sure that the camera completely covers the entire recognition area, and the recognition area is square** in the video, as shown in the figure below. If the recognition area does not meet the requirements in the video, the camera position needs to be adjusted.

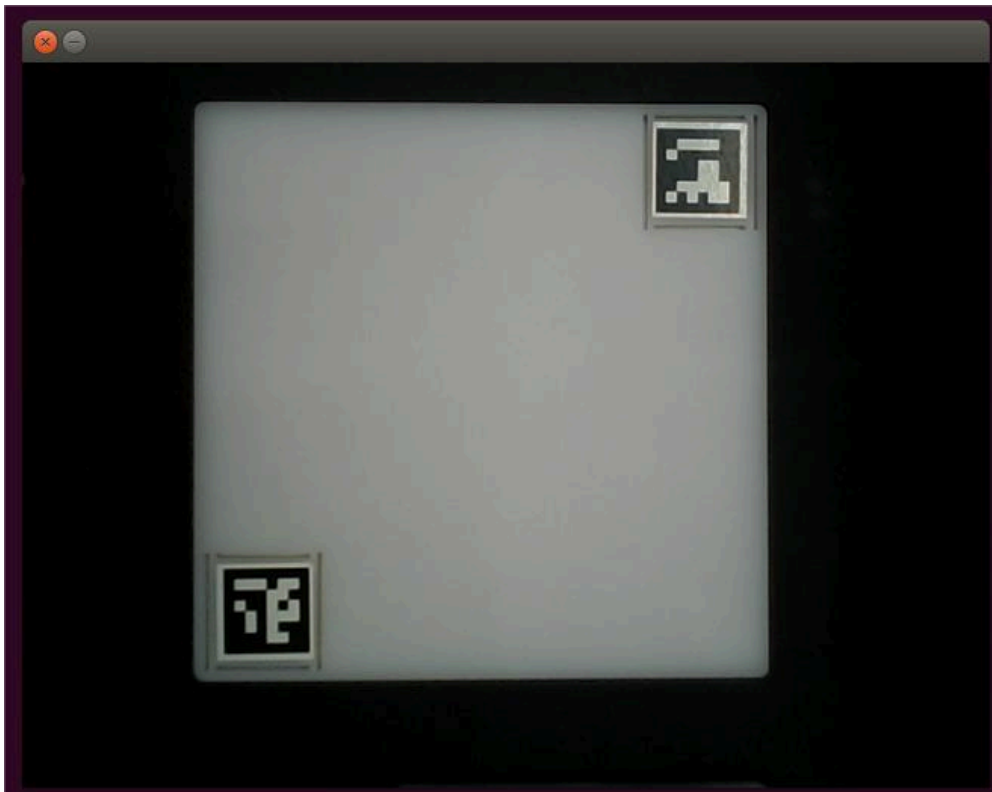
M5 version:

- Open a command prompt terminal (shortcut Win+R, enter "cmd" to access the terminal). Switch to the directory path where "aikit_V2" is located, then enter the target folder.

```
cd Desktop/aikit_V2/AiKit_280M5/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```



Raspberry Pi version:

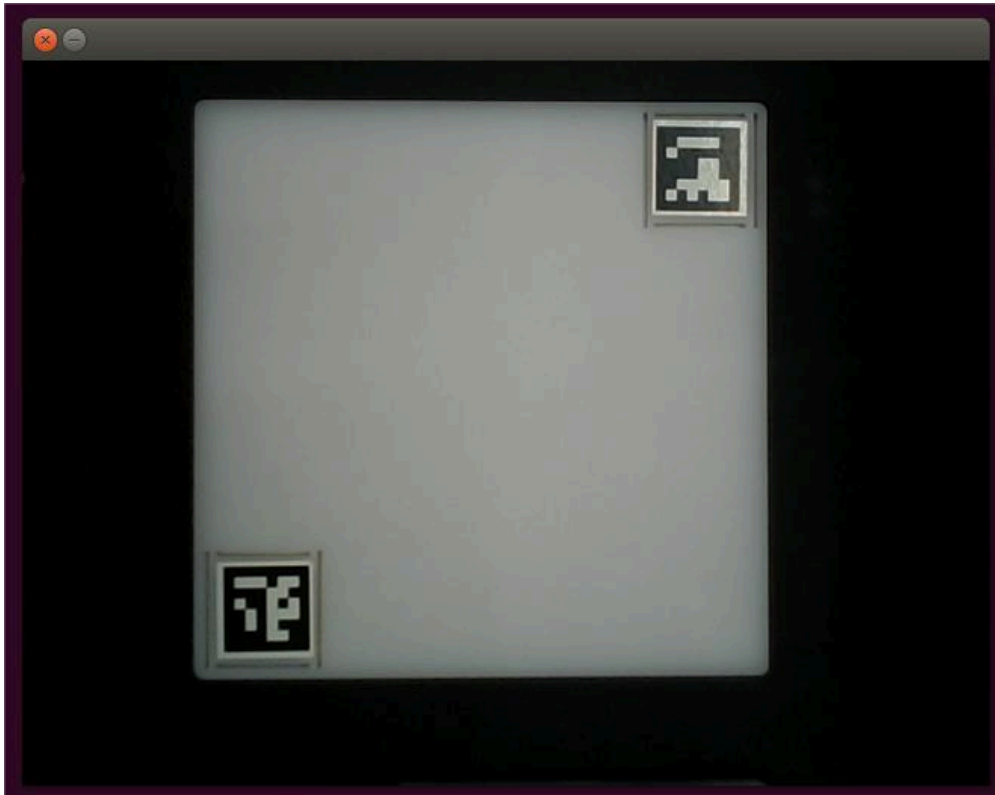
- Open a console terminal (shortcut key Ctrl+Alt+T), enter the target folder

1 Adaptive robot

```
cd ~/aikit_V2/AiKit_280PI/
```

- Enter the following command to open the camera for adjustment

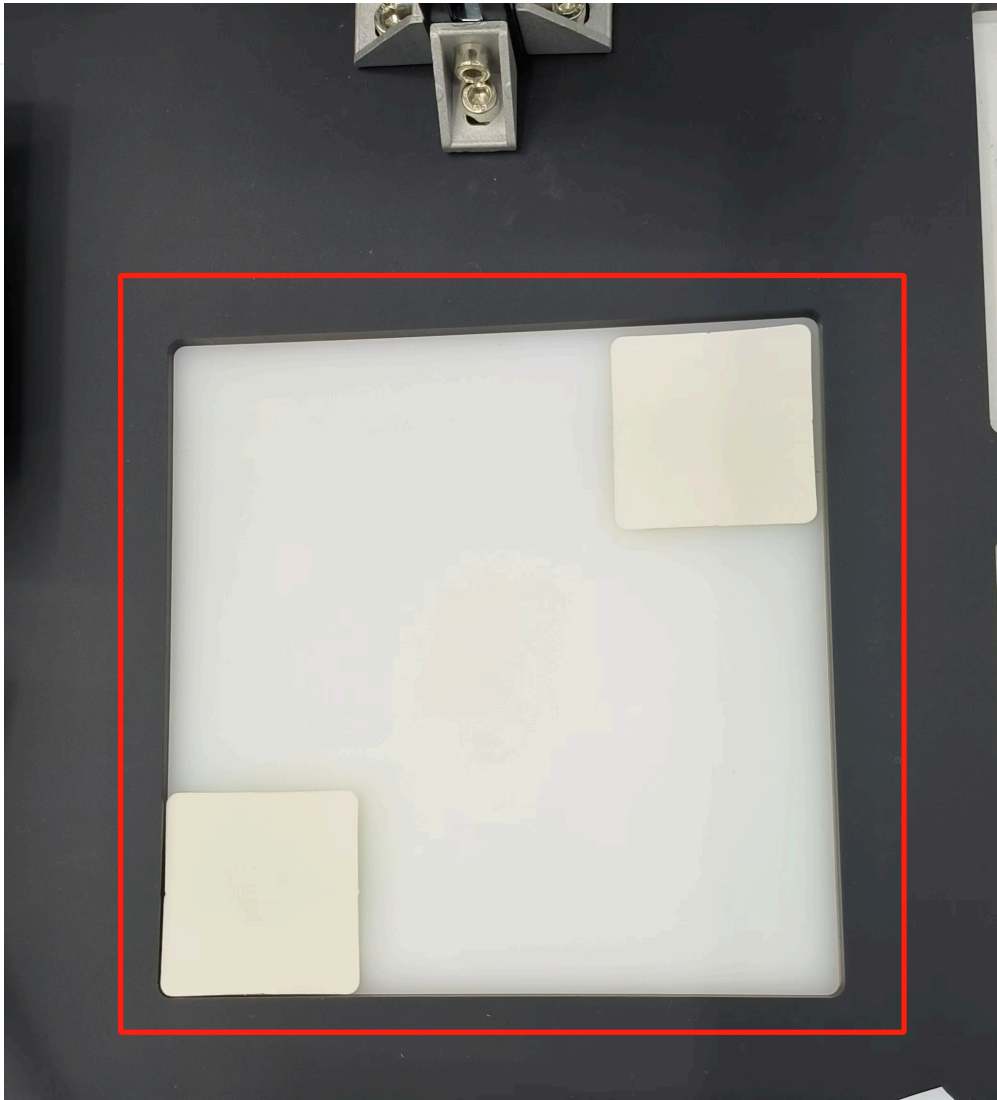
```
python scripts/OpenVideo.py
```



2 Case Reproduction

Prerequisites

Flip the QR code board to the back and place it, or take two cards to cover the two QR codes on the board to prevent misidentification when starting the AR code program.



M5 version:

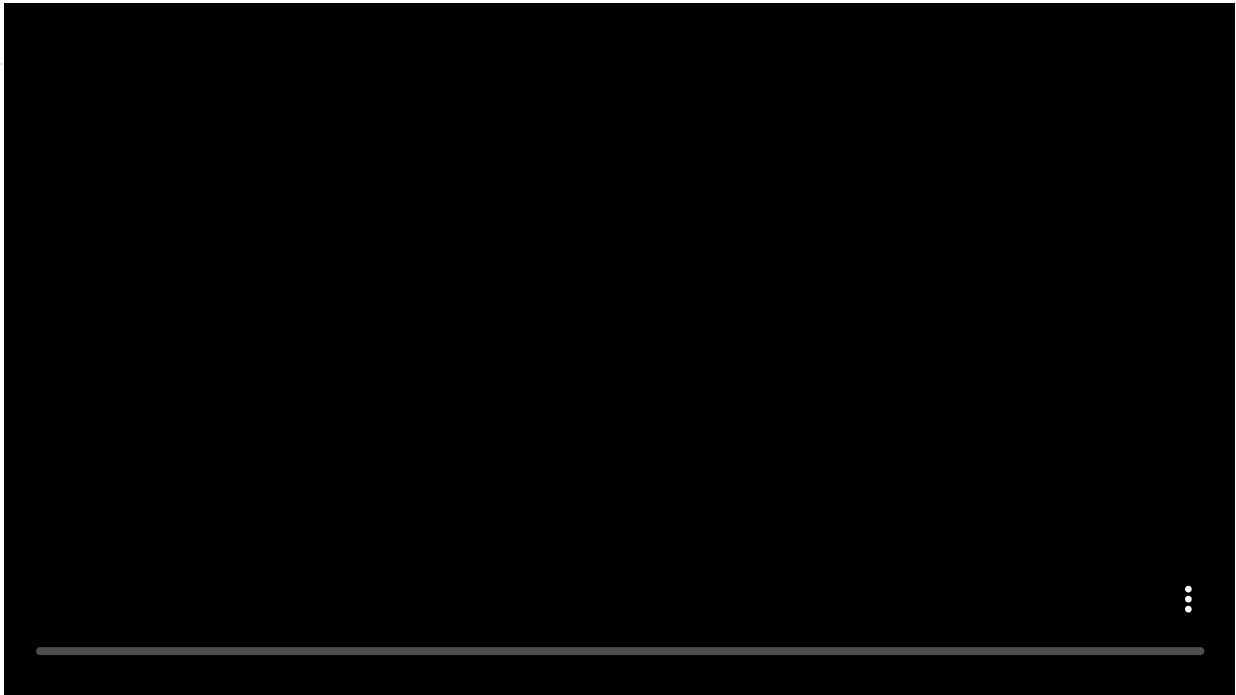
- Open a command prompt terminal (shortcut Win+R, enter "cmd" to access the terminal). Switch to the directory path where "aikit_V2" is located, then enter the target folder.

```
cd Desktop/aikit_V2/AiKit_280M5/
```

- Then enter the following command to start the AR code recognition program.

```
python scripts/aikit_encode.py
```

Raspberry Pi version:



- Open a console terminal (shortcut key `Ctrl+Alt+T`), enter the following command to enter the target folder:

```
cd ~/aikit_V2/AiKit_280PI
```

- Then enter the following command to start the AR code recognition program.

```
python scripts/aikit_encode.py
```

- When the command terminal shows `ok` and the camera window can be opened normally, it means that the program runs successfully. At this time, the identifiable objects can be placed in the recognition area for recognition and capture.

Precautions

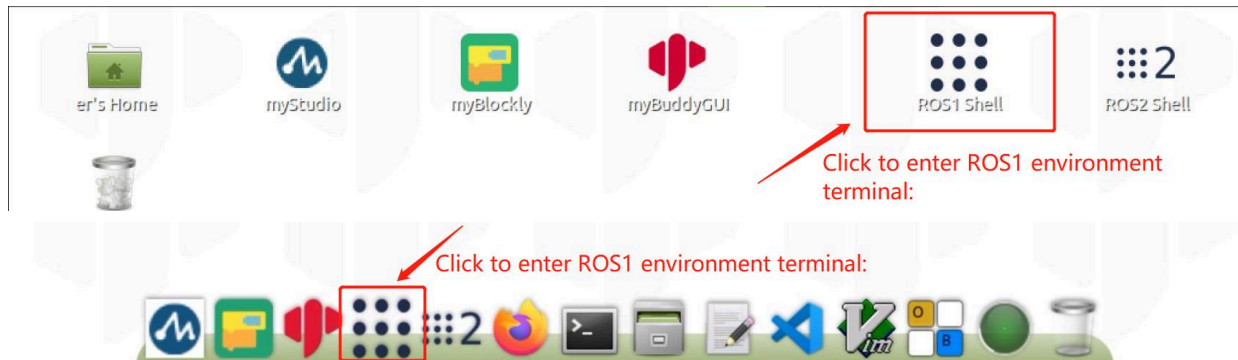
1. When the camera does not automatically frame the recognition area correctly, you need to close the program, adjust the position of the camera, and move the camera to the left or right.
2. If the command terminal does not show `ok`, and the color cannot be recognized, you need to move the camera slightly backward or forward, and the program can run normally when the command terminal shows `ok`.
3. OpenCV color recognition will be affected by the environment, and the recognition effect will be greatly reduced in a darker environment.

Identify AR code

This case uses the eye-to-hand mode, uses the camera, combines **Python+OpenCV**, performs attitude estimation through the ArUco marker in OpenCV, and frames the AR code that meets the situation, and calculates it through the relevant points. The spatial coordinate position of the AR code relative to the robotic arm. Set a set of related actions for the robotic arm, and place them in different areas according to the recognized AR code.

1 program compilation

Click the `ROS1 Shell` icon on the desktop or the corresponding icon in the lower bar of the desktop to open the ROS1 environment terminal:



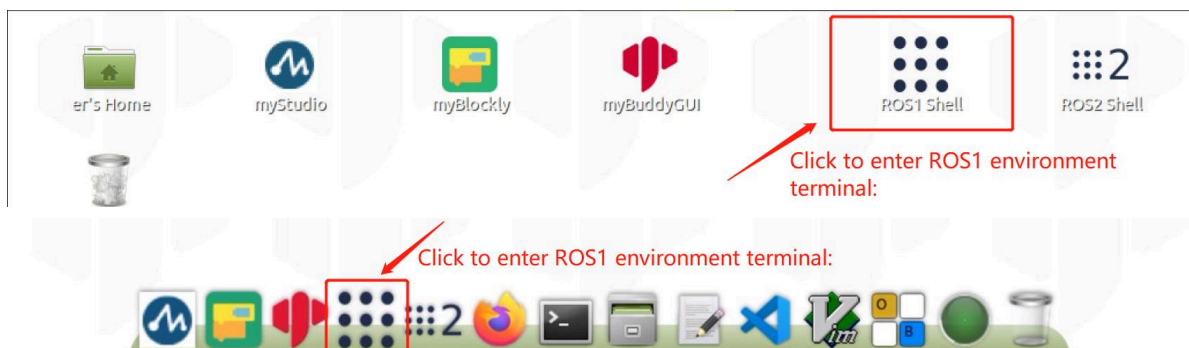
Then enter the following command:

```
cd ~/catkin_ws # Go to the src folder in the workspace
catkin_make # Build the code in the workspace
source devel/setup.bash # Add environment variables
```

2 Camera Adjustment

First, you need to use python to run `OpenVideo.py` under the `aikit_V2` package. If the enabled camera is a computer camera, `cap_num` needs to be modified. For details, please refer to: **Precautions. Make sure that the camera completely covers the entire recognition area, and the recognition area is square** in the video, as shown in the figure below. If the recognition area does not meet the requirements in the video, the camera position needs to be adjusted.

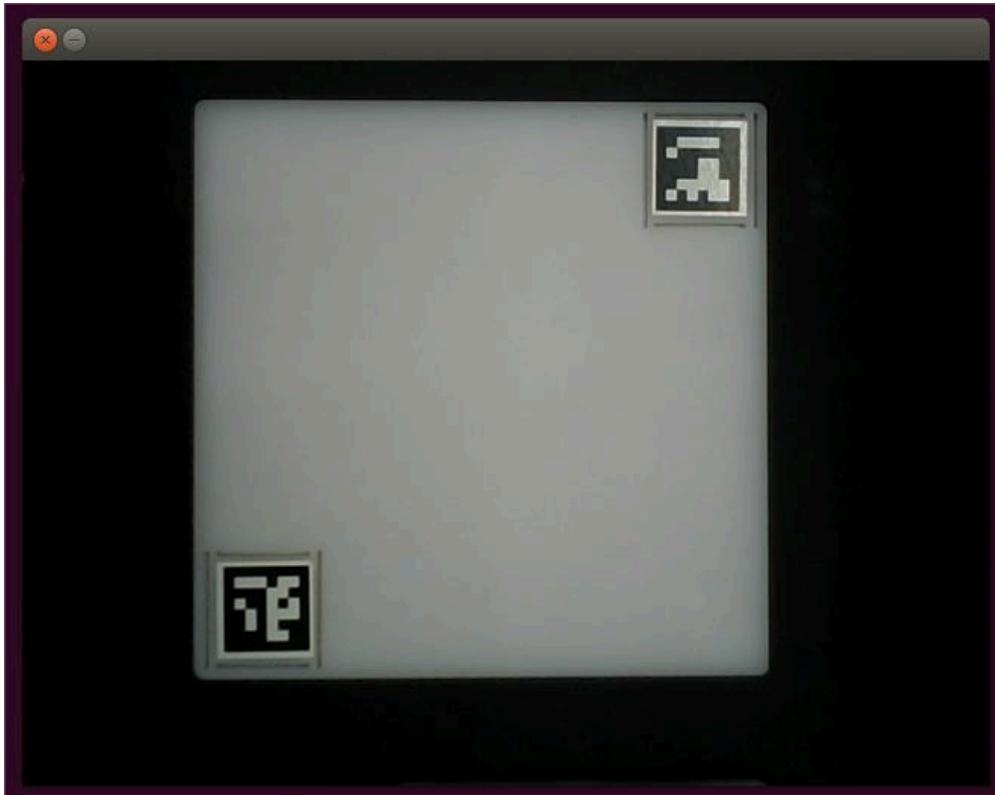
- Click the `ROS1 Shell` icon on the desktop or the corresponding icon in the lower bar of the desktop to open the ROS1 environment terminal, and enter the following command to enter the target folder:



```
cd ~/catkin_ws/src/mycobot_ros/mycobot_ai/aikit_280_pi/
```

- Enter the following command to open the camera for adjustment

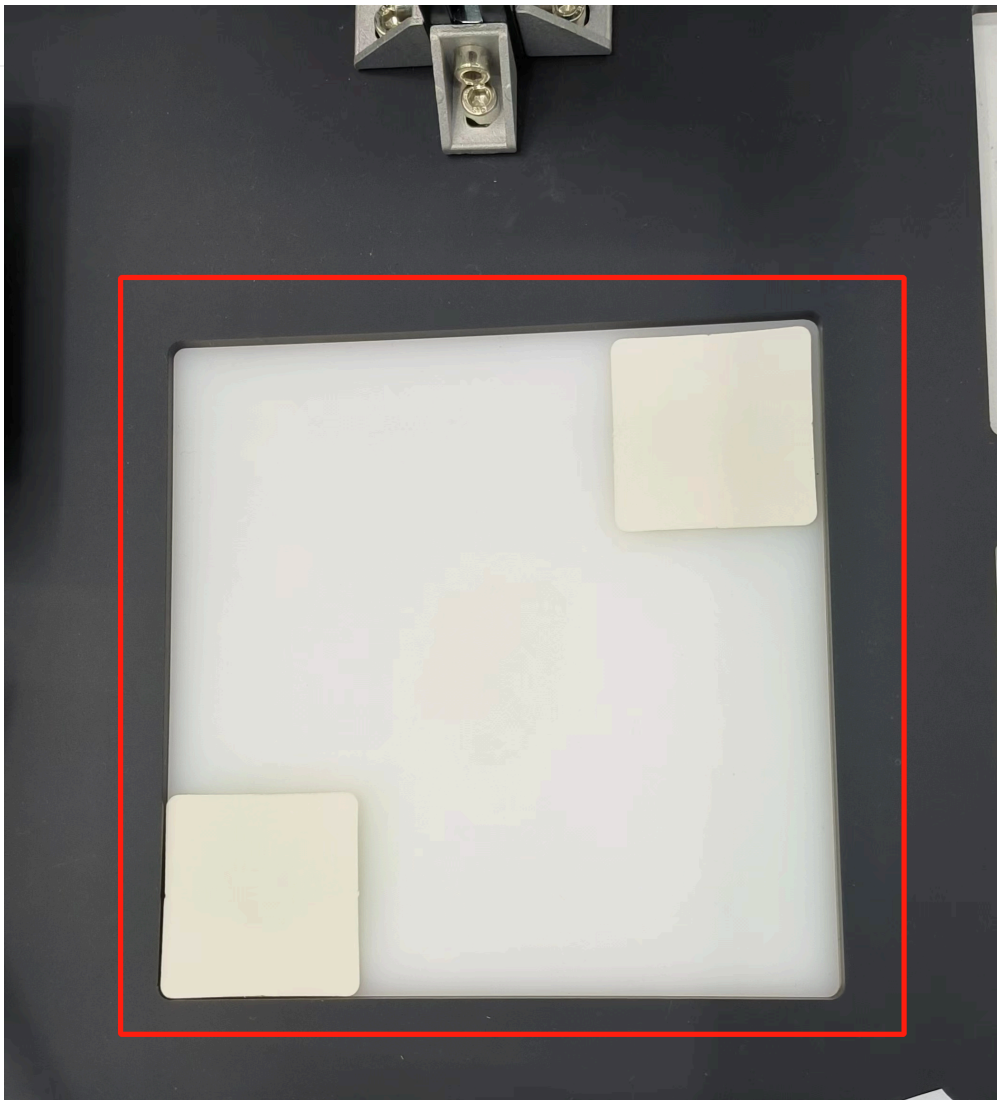
```
python scripts/OpenVideo.py
```



3 Case Reproduction

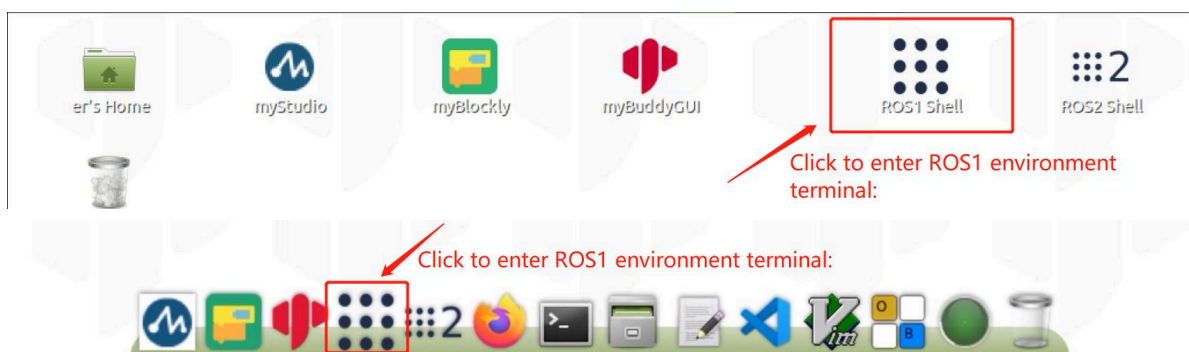
Prerequisites

Flip the QR code board to the back and place it, or take two cards to cover the two QR codes on the board to prevent misidentification when starting the AR code program.



Raspberry Pi version:

- Click the ROS1 Shell icon on the desktop or the corresponding icon in the lower bar of the desktop to open the ROS1 environment terminal, and enter the following command to start the launch file:



```
roslaunch aikit_280_pi vision_pi.launch
```

- Open another ROS1 environment terminal (the opening method is the same as above), enter the following command to enter the target folder:

```
cd ~/catkin_ws/src/mycobot_ros/mycobot_ai/aikit_280_pi/
```

- Then enter the following command to start the AR code recognition program.

```
python scripts/aikit_encode.py
```

- When the command terminal shows ok and the camera window can be opened normally, it means that the program runs successfully. At this time, the identifiable objects can be placed in the recognition area for recognition and capture.

Precautions

1. When the camera does not automatically frame the recognition area correctly, you need to close the program, adjust the position of the camera, and move the camera to the left or right.
2. If the command terminal does not show ok, and the color cannot be recognized, you need to move the camera slightly backward or forward, and the program can run normally when the command terminal shows ok.
3. OpenCV color recognition will be affected by the environment, and the recognition effect will be greatly reduced in a darker environment.

Identify AR code

This case uses the eye-to-hand mode, uses the camera, combines **Python+OpenCV**, performs attitude estimation through the ArUco marker in OpenCV, and frames the AR code that meets the situation, and calculates it through the relevant points. The spatial coordinate position of the AR code relative to the robotic arm. Set a set of related actions for the robotic arm, and place them in different areas according to the recognized AR code.

1 Camera Adjustment

First, you need to use python to run OpenVideo.py under the aikit_V2 package. If the enabled camera is a computer camera, cap_num needs to be modified. For details, please refer to: **Precautions. Make sure that the camera completely covers the entire recognition area, and the recognition area is square** in the video, as shown in the figure below. If the recognition area does not meet the requirements in the video, the camera position needs to be adjusted.

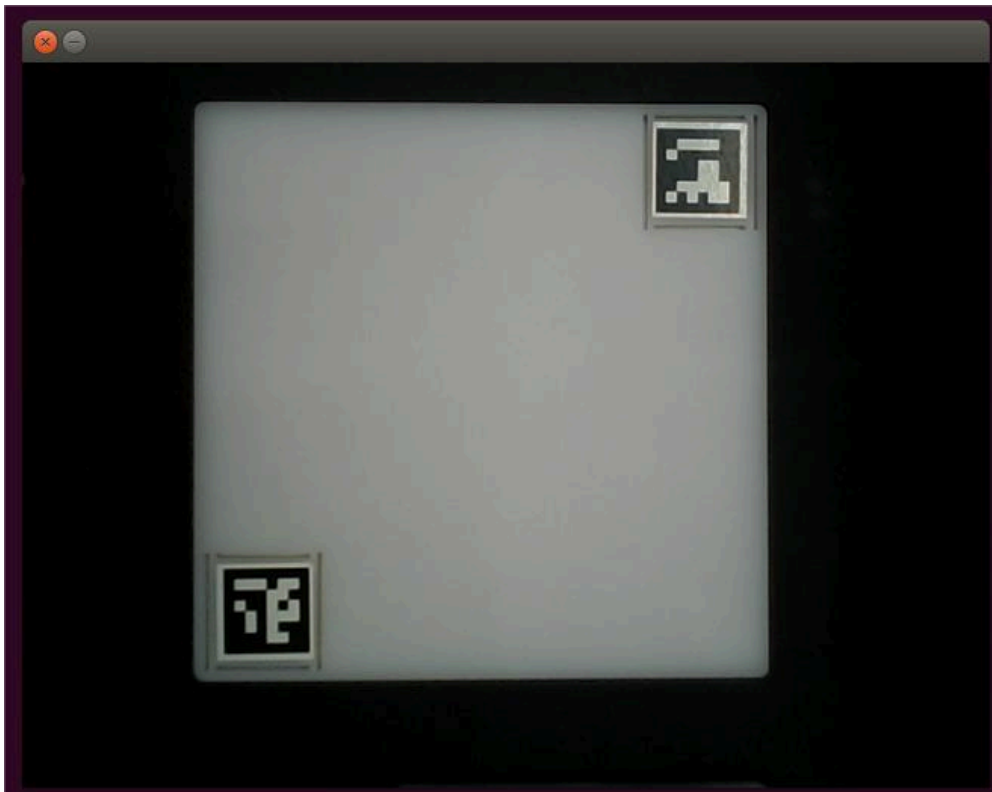
M5 version:

- Open a command prompt terminal (shortcut Win+R, enter "cmd" to access the terminal). Switch to the directory path where "aikit_V2" is located, then enter the target folder.

```
cd Desktop/aikit_V2/AiKit_270M5/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```



Raspberry Pi version:

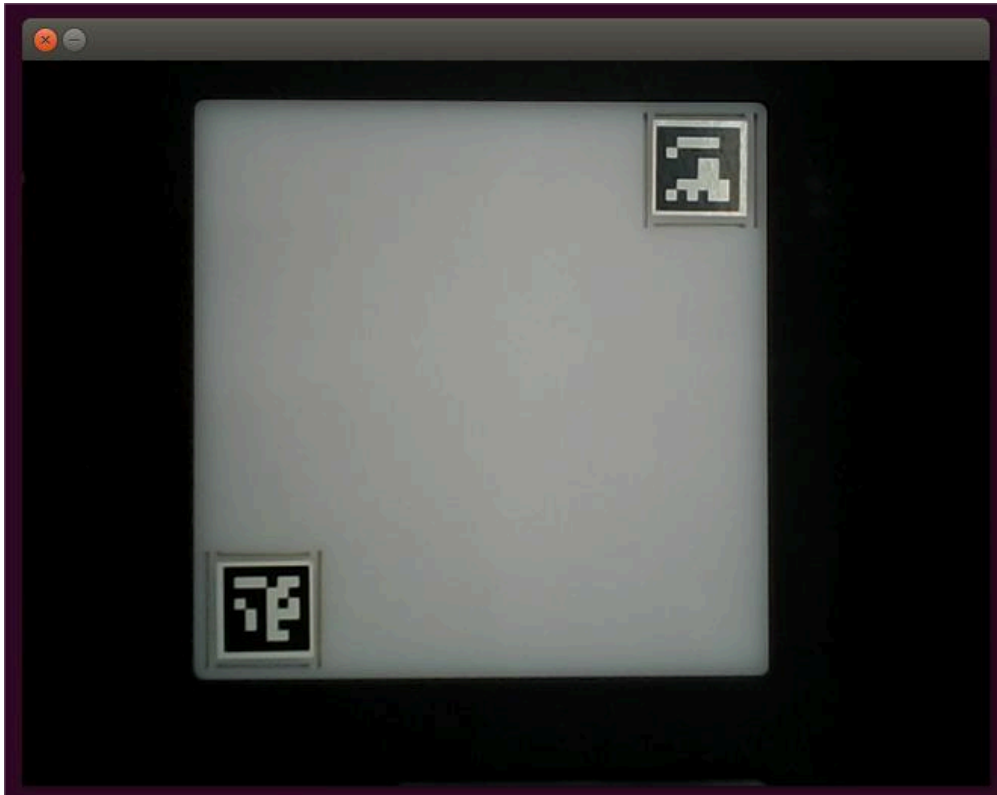
- Open a console terminal (shortcut key Ctrl+Alt+T), enter the target folder

1 Adaptive robot

```
cd ~/aikit_V2/AiKit_270Pi/
```

- Enter the following command to open the camera for adjustment

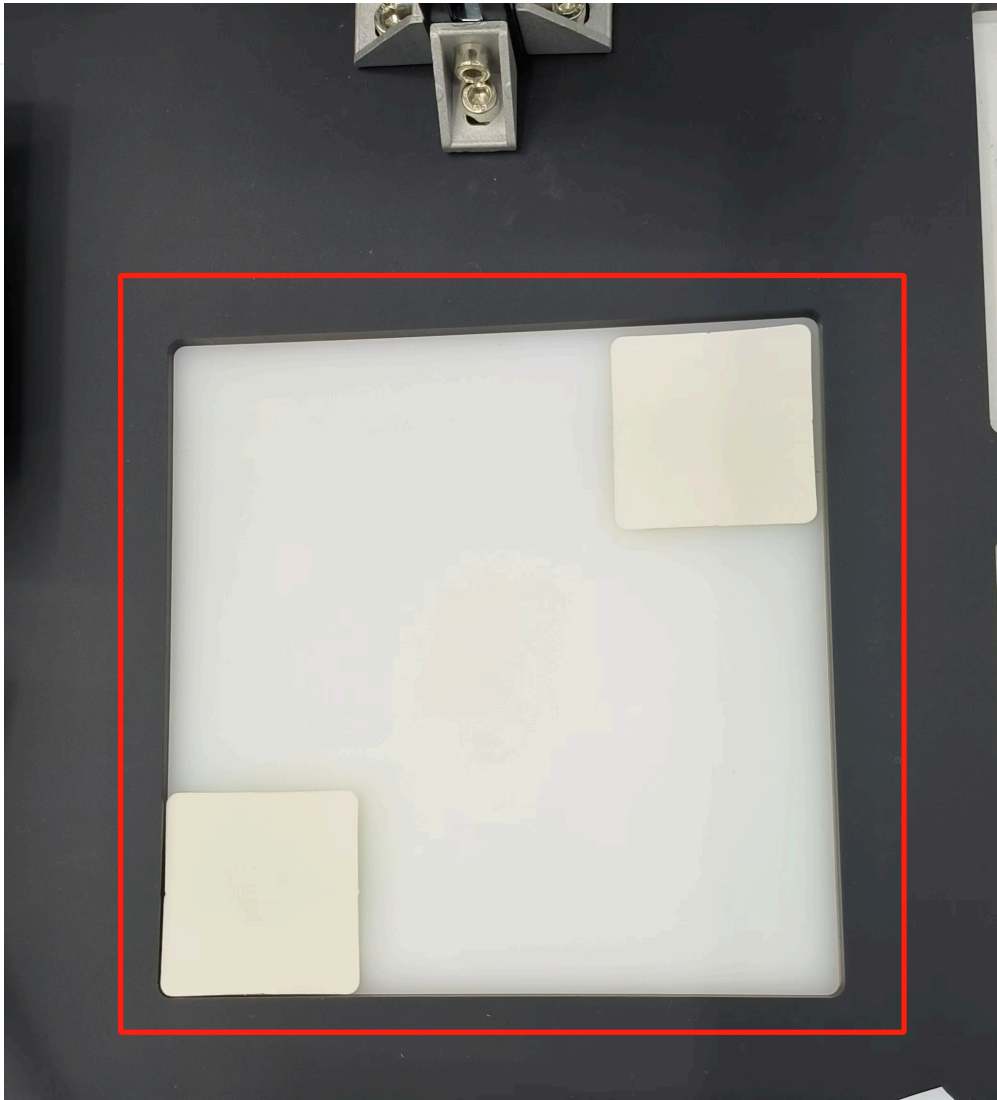
```
python scripts/OpenVideo.py
```



2 Case Reproduction

Prerequisites

Flip the QR code board to the back and place it, or take two cards to cover the two QR codes on the board to prevent misidentification when starting the AR code program.



M5 version:

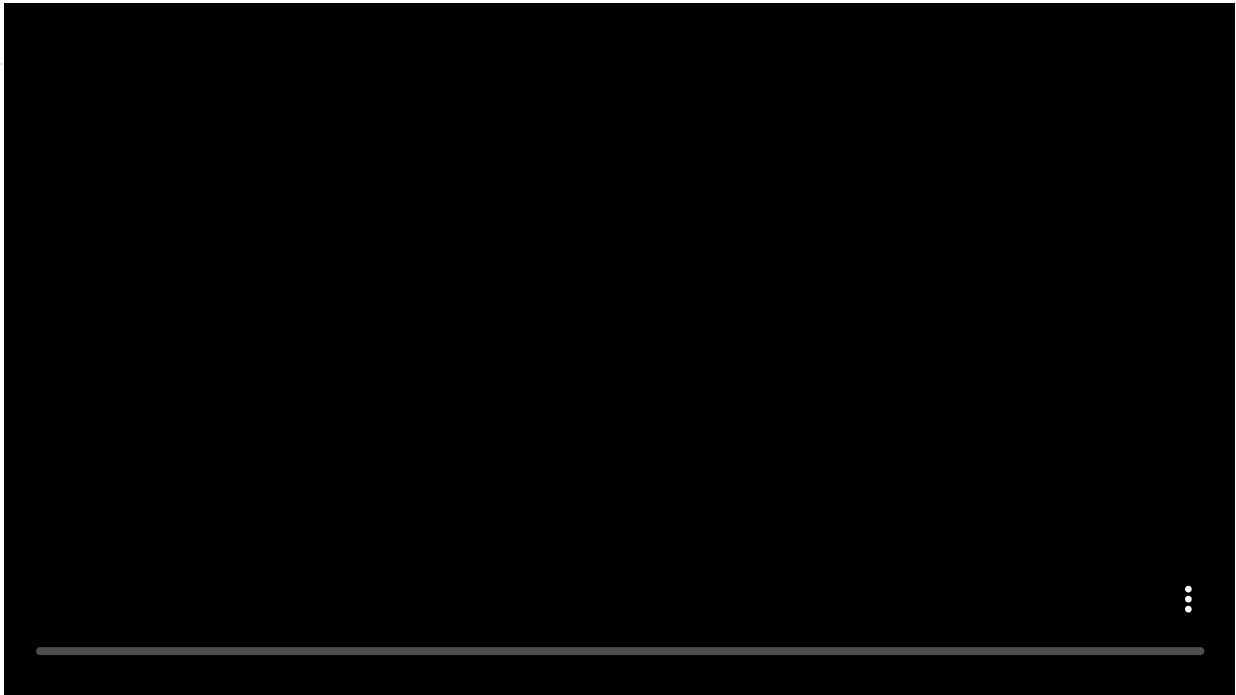
- Open a command prompt terminal (shortcut Win+R, enter "cmd" to access the terminal). Switch to the directory path where "aikit_V2" is located, then enter the target folder.

```
cd Desktop/aikit_V2/AiKit_270M5/
```

- Then enter the following command to start the AR code recognition program.

```
python scripts/aikit_encode.py
```

Raspberry Pi version :



- Open a console terminal (shortcut key `Ctrl+Alt+T`), enter the following command to enter the target folder:

```
cd ~/aikit_V2/AiKit_270Pi
```

- Then enter the following command to start the AR code recognition program.

```
python scripts/aikit_encode.py
```

- When the command terminal shows `ok` and the camera window can be opened normally, it means that the program runs successfully. At this time, the identifiable objects can be placed in the recognition area for recognition and capture.

Precautions

1. When the camera does not automatically frame the recognition area correctly, you need to close the program, adjust the position of the camera, and move the camera to the left or right.
2. If the command terminal does not show `ok`, and the color cannot be recognized, you need to move the camera slightly backward or forward, and the program can run normally when the command terminal shows `ok`.
3. OpenCV color recognition will be affected by the environment, and the recognition effect will be greatly reduced in a darker environment.

Identify AR code

This case uses the eye-to-hand mode, uses the camera, combines **Python+OpenCV**, performs attitude estimation through the ArUco marker in OpenCV, and frames the AR code that meets the situation, and calculates it through the relevant points. The spatial coordinate position of the AR code relative to the robotic arm. Set a set of related actions for the robotic arm, and place them in different areas according to the recognized AR code.

1 Camera Adjustment

First, you need to use python to run OpenVideo.py under the aikit_V2 package. If the enabled camera is a computer camera, cap_num needs to be modified. For details, please refer to: **Precautions. Make sure that the camera completely covers the entire recognition area, and the recognition area is square** in the video, as shown in the figure below. If the recognition area does not meet the requirements in the video, the camera position needs to be adjusted.

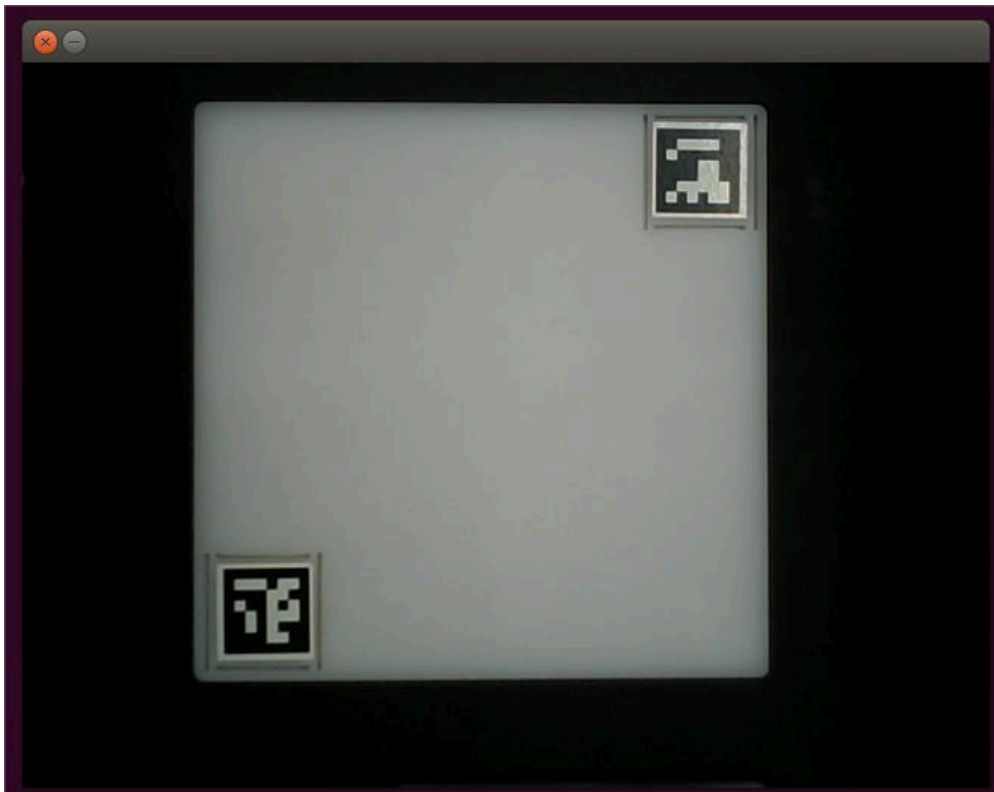
M5 version:

- Open a console terminal (shortcut key Win+R, enter cmd to enter the terminal), enter the following command to enter the target folder:

```
cd Desktop/aikit_V2/AiKit_260M5/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```



Raspberry Pi version:

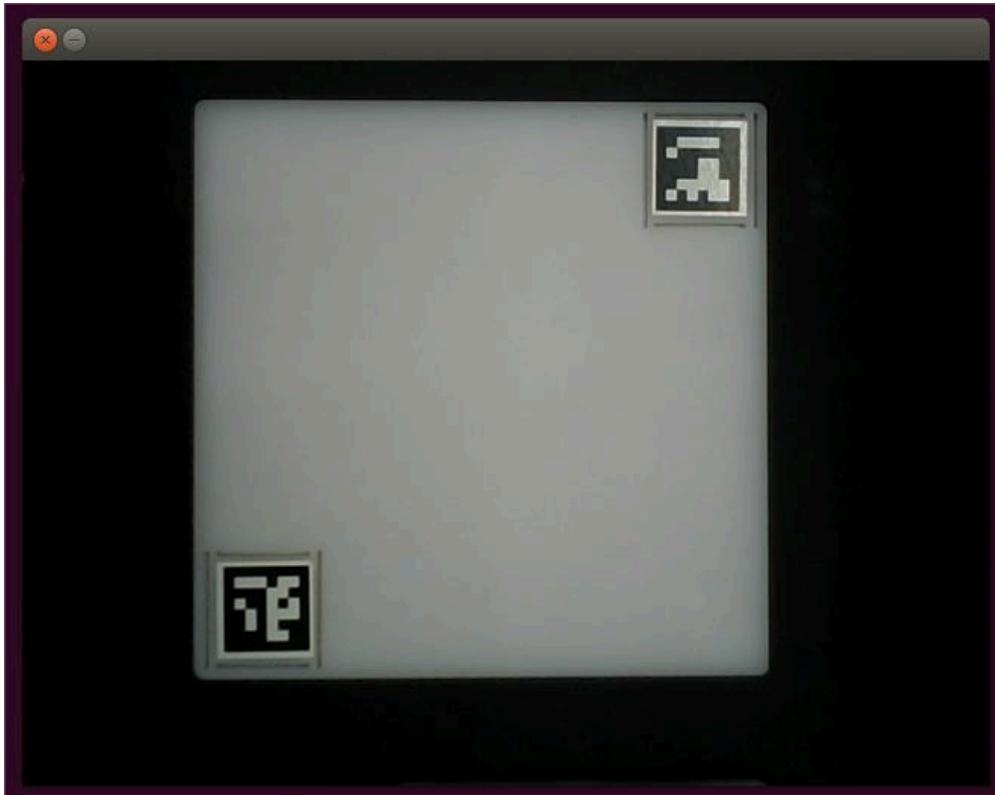
- Open a console terminal (shortcut key Ctrl+Alt+T), enter the target folder

1 Adaptive robot

```
cd ~/aikit_V2/AiKit_260PI/
```

- Enter the following command to open the camera for adjustment

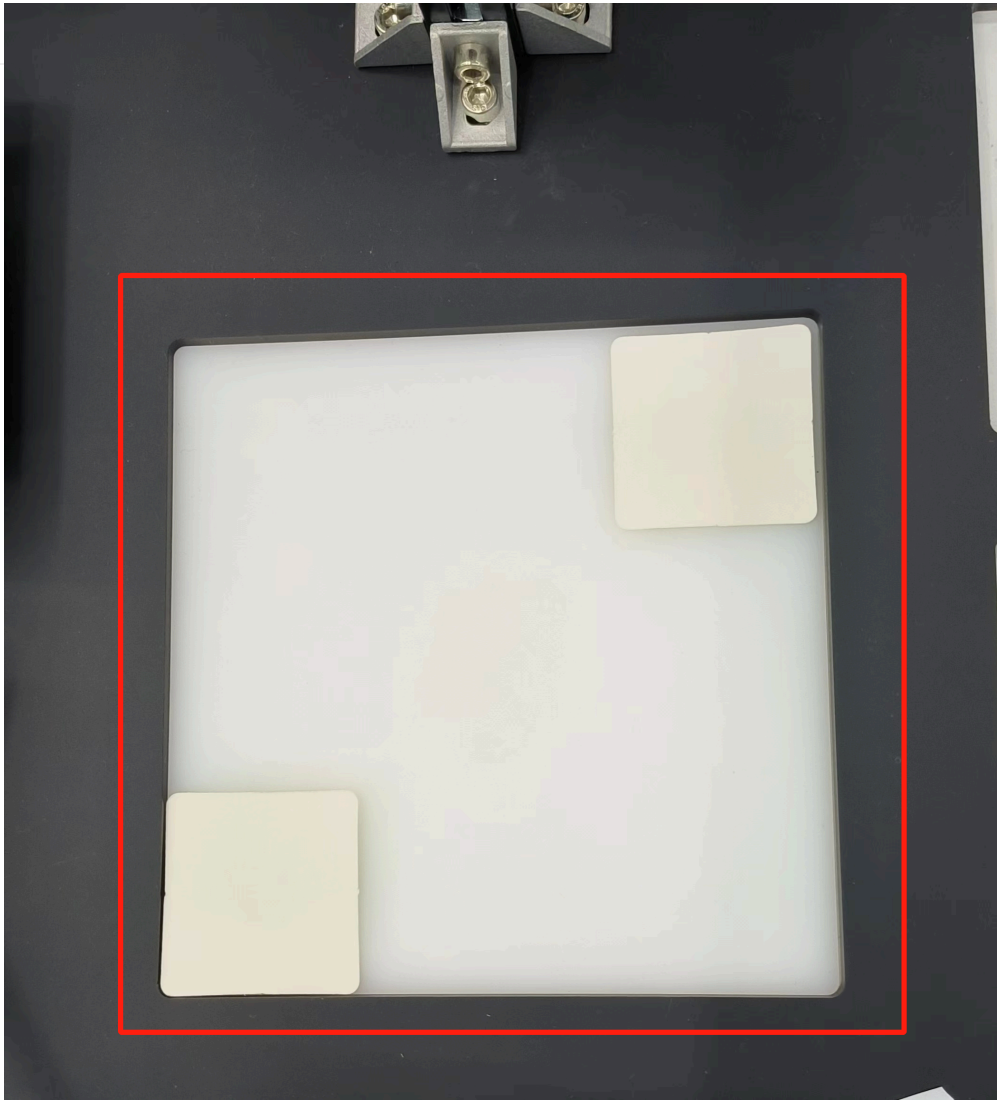
```
python scripts/OpenVideo.py
```



2 Case Reproduction

Prerequisites

Flip the QR code board to the back and place it, or take two cards to cover the two QR codes on the board to prevent misidentification when starting the AR code program.



M5 version:



- Open a console terminal (shortcut key Win+R, enter cmd to enter the terminal), enter the following command to enter the target folder:

```
cd Desktop/aikit_V2/AiKit_260M5
```

- Then enter the following command to start the AR code recognition program.

```
python scripts/aikit_encode.py
```

Raspberry Pi version:

- Open a console terminal (shortcut key Ctrl+Alt+T), enter the target folder

```
cd ~/aikit_V2/AiKit_260PI/
```

- Then enter the following command to start the AR code recognition program.

```
python scripts/aikit_encode.py
```

- When the command terminal shows `ok` and the camera window can be opened normally, it means that the program runs successfully. At this time, the identifiable objects can be placed in the recognition area for recognition and capture.

Precautions

1. When the camera does not automatically frame the recognition area correctly, you need to close the program, adjust the position of the camera, and move the camera to the left or right.
2. If the command terminal does not show `ok`, and the color cannot be recognized, you need to move the camera slightly backward or forward, and the program can run normally when the command terminal shows `ok`.
3. OpenCV color recognition will be affected by the environment, and the recognition effect will be greatly reduced in a darker environment.

Recognize image cards

This case uses the eye-to-hand mode, uses the camera, loads the model data trained by Tensorflow through OpenCV, recognizes the image block and locates the position of the image block in the video. Through the relevant points, the space coordinate position of the block relative to the robot arm is calculated. Set up a set of related actions for the robot arm, and put the identified object into the bucket. In the following chapters, the code implementation process of the entire case will be introduced in detail.

1 Camera Adjustment

First, you need to use python to run OpenVideo.py under the aikit_V2 package. If the enabled camera is a computer camera, cap_num needs to be modified. For details, please refer to: **Precautions. Make sure that the camera completely covers the entire recognition area, and the recognition area is square** in the video, as shown in the figure below. If the recognition area does not meet the requirements in the video, the camera position needs to be adjusted.

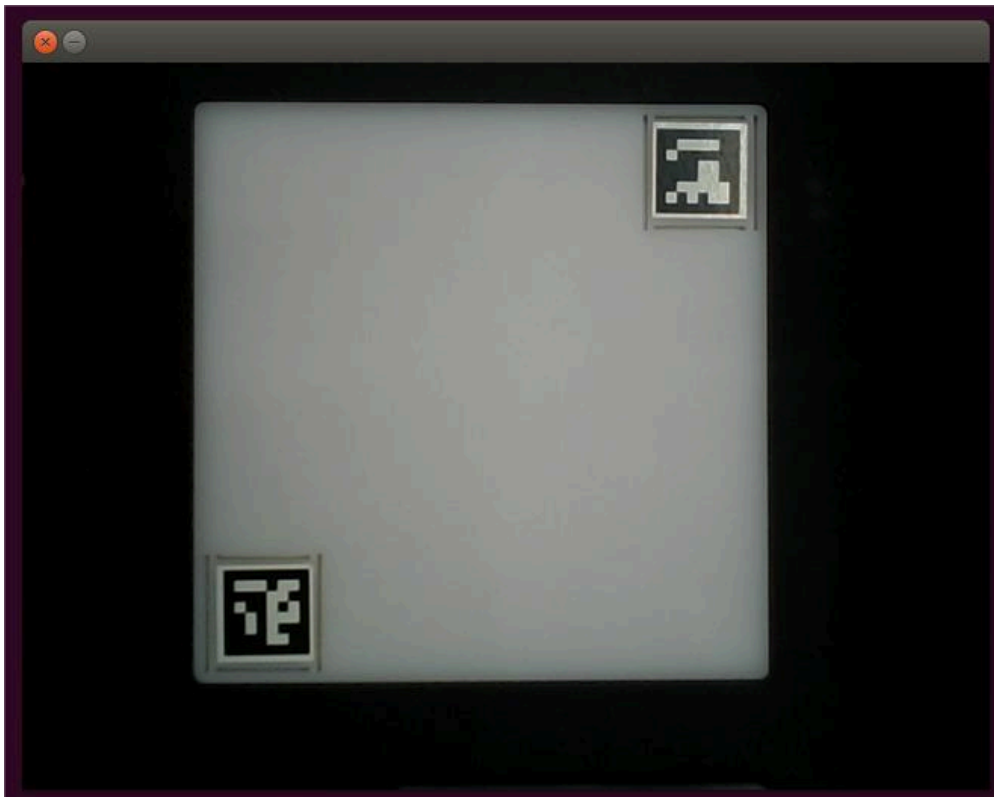
M5 version:

- Open a command prompt terminal (shortcut Win+R, enter "cmd" to access the terminal). Switch to the directory path where "aikit_V2" is located, then enter the target folder.

```
cd Desktop/aikit_V2/AiKit_280M5/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```



Raspberry Pi version:

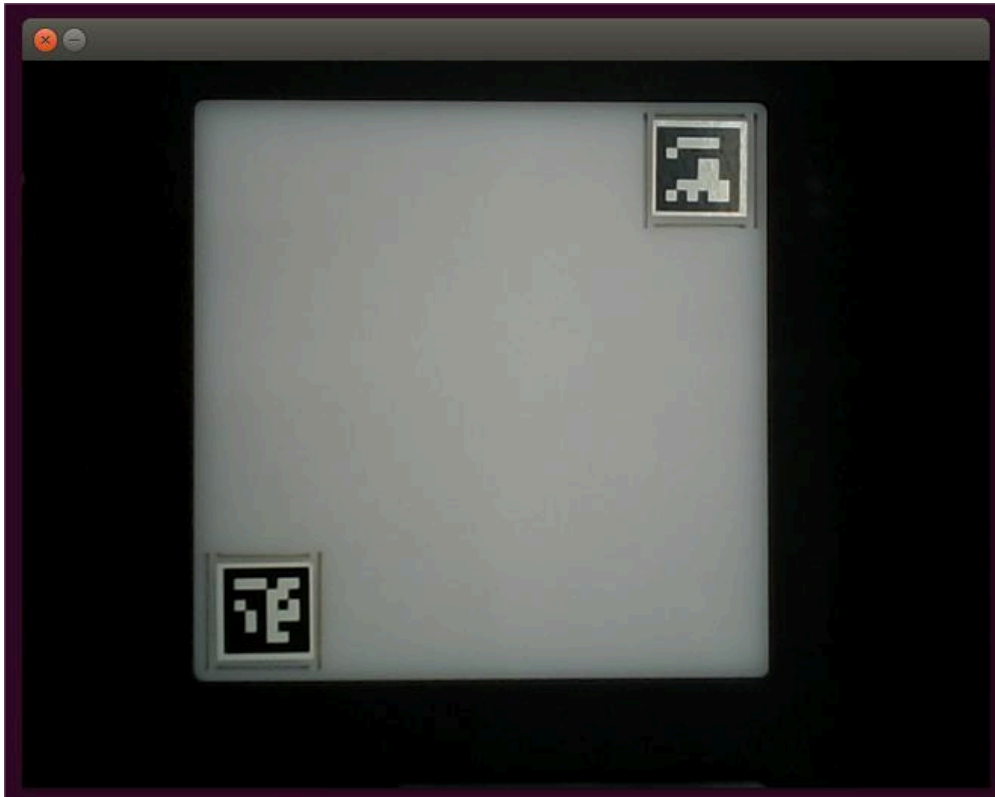
1 Adaptive robot

- Open a console terminal (shortcut key Ctrl+Alt+T), enter the target folder

```
cd ~/aikit_V2/AiKit_280PI/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```



2 Add new image

M5 version:

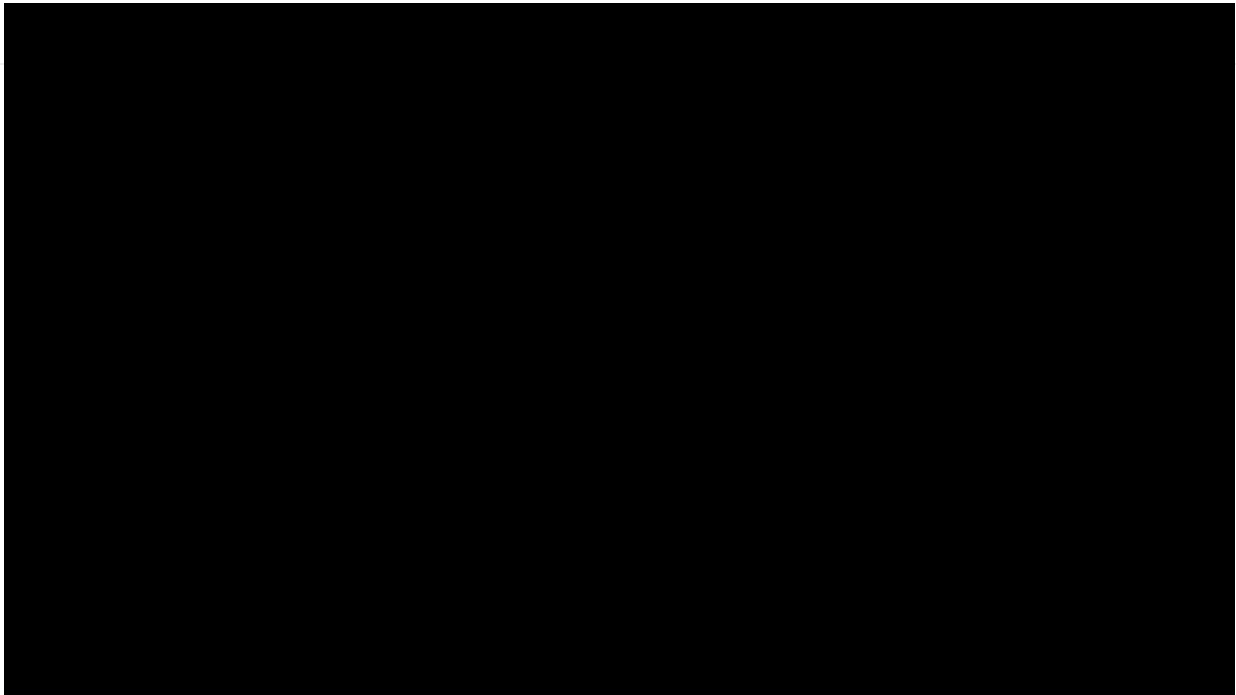
- Open a command prompt terminal (shortcut Win+R, enter "cmd" to access the terminal). Switch to the directory path where "aikit_V2" is located, then enter the target folder.

```
cd Desktop/aikit_V2/AiKit_280M5/
```

- Enter the following command to start the process of adding images.

```
python scripts/add_img.py
```

Raspberry Pi version:



- Open a console terminal (shortcut key Ctrl+Alt+T), enter the target folder

```
cd ~/aikit_V2/AiKit_280PI/
```

- Enter the following command to start the process of adding images.

```
python scripts/add_img.py
```

- Operate according to the prompts entered by the terminal, and capture the image in the second image box that pops up.
- After the image capture area is completed, press the Enter key, and according to the terminal prompt, enter a number (1~4) to save to the folder corresponding to the image, and press the Enter key to save it to the corresponding folder.

3 Case Reproduction

M5 version:

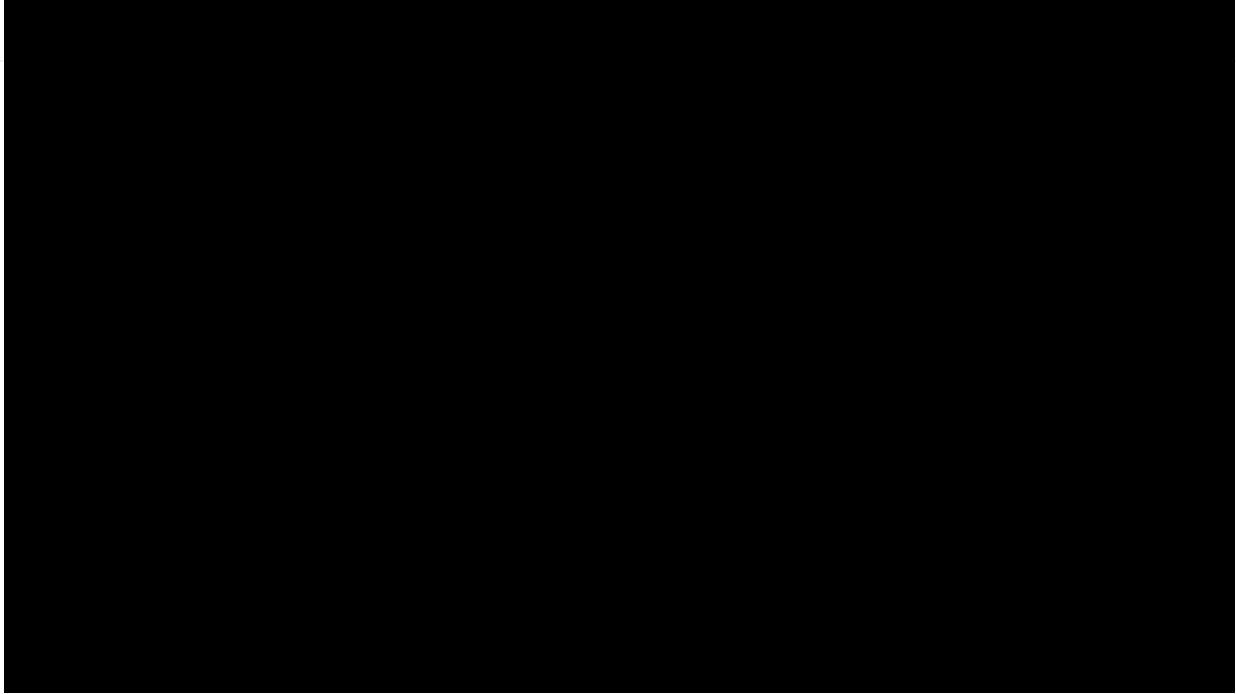
- Open a command prompt terminal (shortcut Win+R, enter "cmd" to access the terminal). Switch to the directory path where "aikit_V2" is located, then enter the target folder.

```
cd Desktop/aikit_V2/AiKit_280M5/
```

- Then enter the following command to start the image recognition program.

```
python scripts/aikit_img.py
```

Raspberry Pi version:



- Open a console terminal (shortcut key `Ctrl+Alt+T`), enter the following command to enter the target folder:

```
cd ~/aikit_V2/AiKit_280PI
```

- Then enter the following command to start the image recognition program.

```
python scripts/aikit_img.py
```

- When the command terminal shows ok and the camera window can be opened normally, it means that the program runs successfully. At this time, the identifiable objects can be placed in the recognition area for recognition and capture.

Precautions

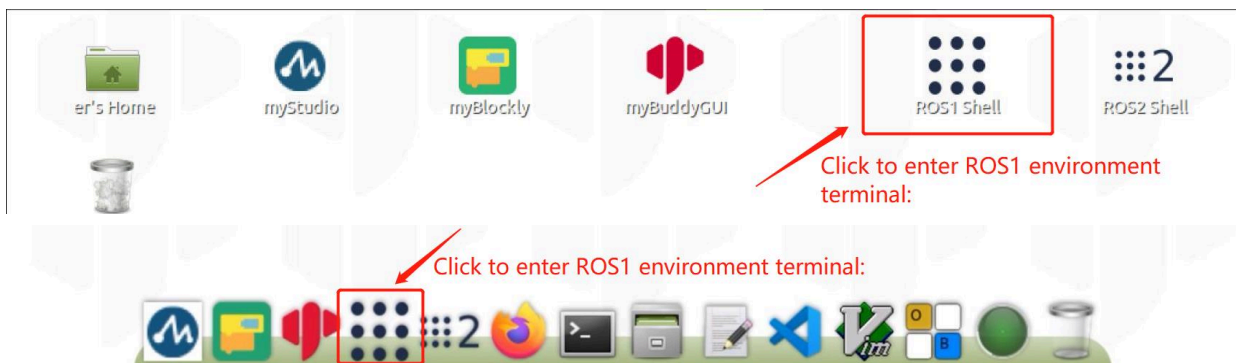
1. When the camera does not automatically frame the recognition area correctly, you need to close the program, adjust the position of the camera, and move the camera to the left or right.
2. If the command terminal does not display ok, and the picture cannot be recognized, you need to move the camera slightly backward or forward, and the program can run normally when the command terminal displays ok.
3. Since the recognition method has been changed, please make sure that there is a folder for storing pictures and pictures that need to be recognized under the code path. If not, you can use `python scripts/add_img.py` to add pictures.
4. OpenCV color recognition will be affected by the environment, and the recognition effect will be greatly reduced in a darker environment.

Recognize image cards

This case uses the eye-to-hand mode, uses the camera, loads the model data trained by Tensorflow through OpenCV, recognizes the image block and locates the position of the image block in the video. Through the relevant points, the space coordinate position of the block relative to the robot arm is calculated. Set up a set of related actions for the robot arm, and put the identified object into the bucket. In the following chapters, the code implementation process of the entire case will be introduced in detail.

1 program compilation

Click the ROS1 Shell icon on the desktop or the corresponding icon in the lower bar of the desktop to open the ROS1 environment terminal:



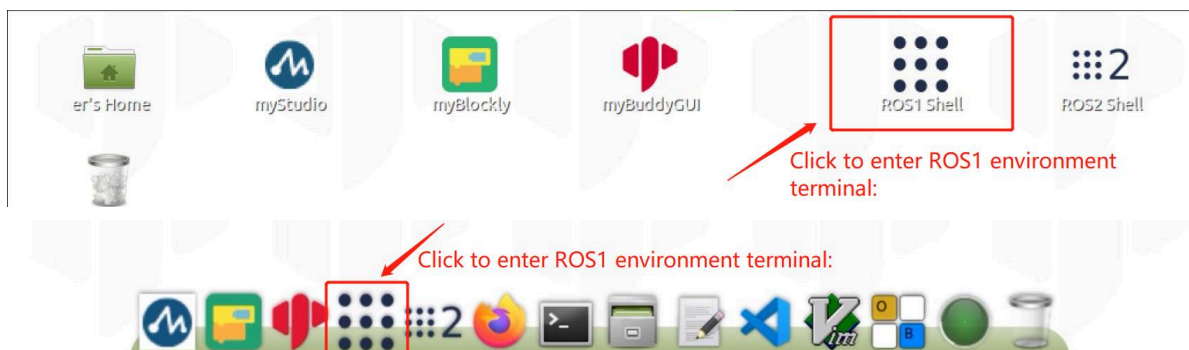
Then enter the following command:

```
cd ~/catkin_ws # Go to the src folder in the workspace
catkin_make # Build the code in the workspace
source devel/setup.bash # Add environment variables
```

2 Camera Adjustment

First, you need to use python to run OpenVideo.py under the aikit_V2 package. If the enabled camera is a computer camera, cap_num needs to be modified. For details, please refer to: **Precautions. Make sure that the camera completely covers the entire recognition area, and the recognition area is square** in the video, as shown in the figure below. If the recognition area does not meet the requirements in the video, the camera position needs to be adjusted.

- Click the ROS1 Shell icon on the desktop or the corresponding icon in the lower bar of the desktop to open the ROS1 environment terminal, and enter the following command to enter the target folder:

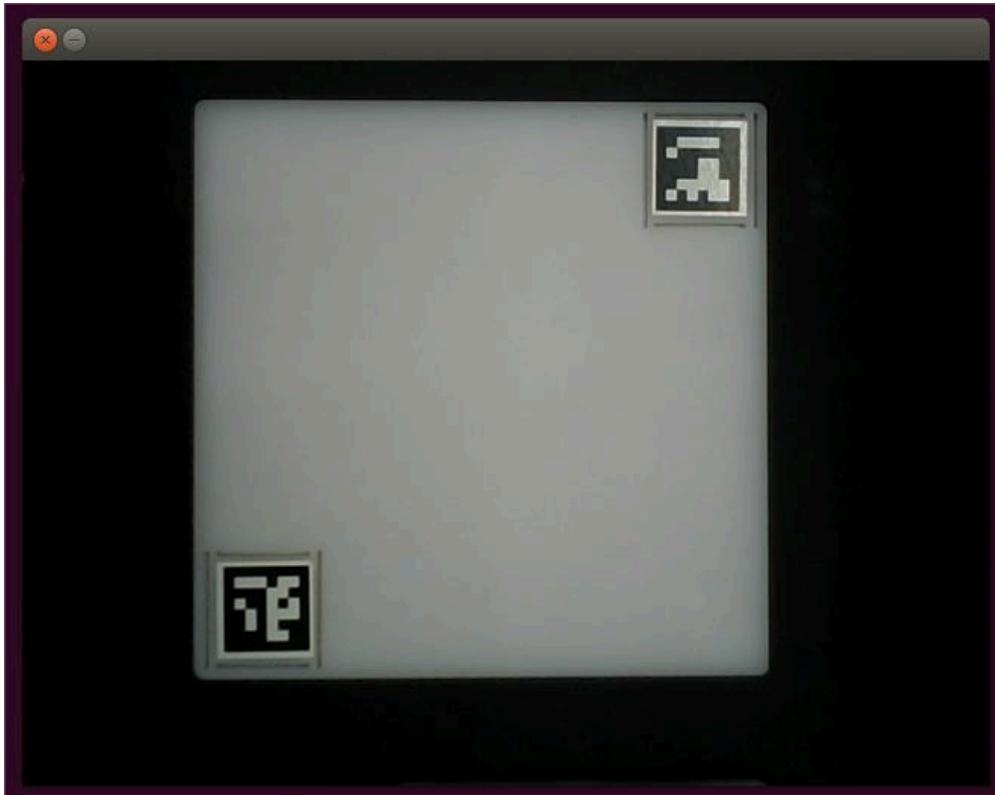


1 Adaptive robot

```
cd ~/catkin_ws/src/mycobot_ros/mycobot_ai/aikit_280_pi/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```



2 Add new image



- Open a console terminal (shortcut key Ctrl+Alt+T), enter the target folder

```
cd ~/catkin_ws/src/mycobot_ros/mycobot_ai/aikit_280_pi/
```

- Enter the following command to start the process of adding images.

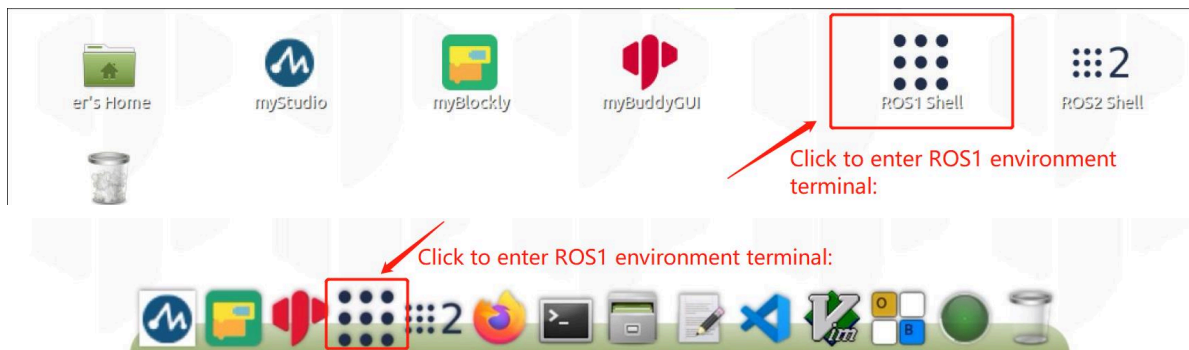
```
python scripts/add_img.py
```

- Operate according to the prompts entered by the terminal, and capture the image in the second image box that pops up.
- After the image capture area is completed, press the Enter key, and according to the terminal prompt, enter a number (1~4) to save to the folder corresponding to the image, and press the Enter key to save it to the corresponding folder.

3 Case Reproduction

Raspberry Pi version:

- Click the ROS1 Shell icon on the desktop or the corresponding icon in the lower bar of the desktop to open the ROS1 environment terminal, and enter the following command to start the launch file:



```
roslaunch aikit_280_pi vision_pi.launch
```

- Open another ROS1 environment terminal (the opening method is the same as above), and enter the following command to enter the target folder:

```
cd ~/catkin_ws/src/mycobot_ros/mycobot_ai/aikit_280_pi/
```

- Then enter the following command to start the image recognition program.

```
python scripts/aikit_img.py
```

- When the command terminal shows ok and the camera window can be opened normally, it means that the program runs successfully. At this time, the identifiable objects can be placed in the recognition area for recognition and capture.

Precautions

1. When the camera does not automatically frame the recognition area correctly, you need to close the program, adjust the position of the camera, and move the camera to the left or right.
2. If the command terminal does not display ok, and the picture cannot be recognized, you need to move the camera slightly backward or forward, and the program can run normally when the command terminal displays

ok.

- 3. Since the recognition method has been changed, please make sure that there is a folder for storing pictures and pictures that need to be recognized under the code path. If not, you can use `python scripts/add_img.py` to add pictures.
- 4. OpenCV color recognition will be affected by the environment, and the recognition effect will be greatly reduced in a darker environment.

Recognize image cards

This case uses the eye-to-hand mode, uses the camera, loads the model data trained by Tensorflow through OpenCV, recognizes the image block and locates the position of the image block in the video. Through the relevant points, the space coordinate position of the block relative to the robot arm is calculated. Set up a set of related actions for the robot arm, and put the identified object into the bucket. In the following chapters, the code implementation process of the entire case will be introduced in detail.

1 Camera Adjustment

First, you need to use python to run OpenVideo.py under the aikit_V2 package. If the enabled camera is a computer camera, cap_num needs to be modified. For details, please refer to: **Precautions. Make sure that the camera completely covers the entire recognition area, and the recognition area is square** in the video, as shown in the figure below. If the recognition area does not meet the requirements in the video, the camera position needs to be adjusted.

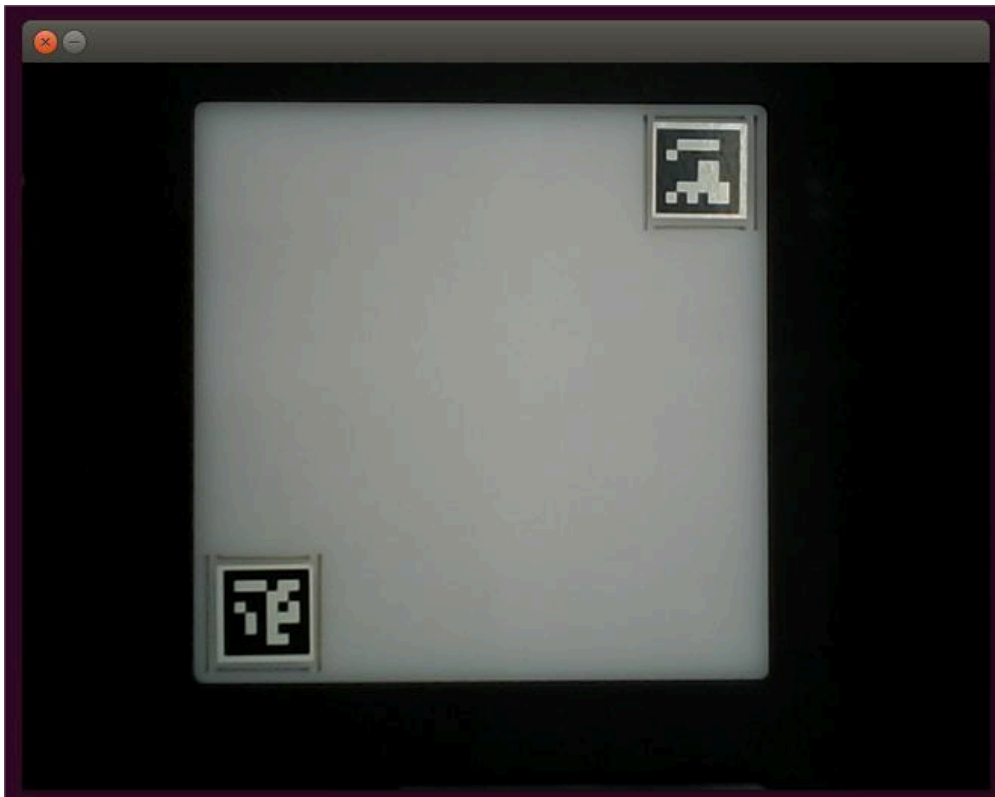
M5 version:

- Open a command prompt terminal (shortcut Win+R, enter "cmd" to access the terminal). Switch to the directory path where "aikit_V2" is located, then enter the target folder.

```
cd Desktop/aikit_V2/AiKit_270M5/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```



Raspberry Pi version:

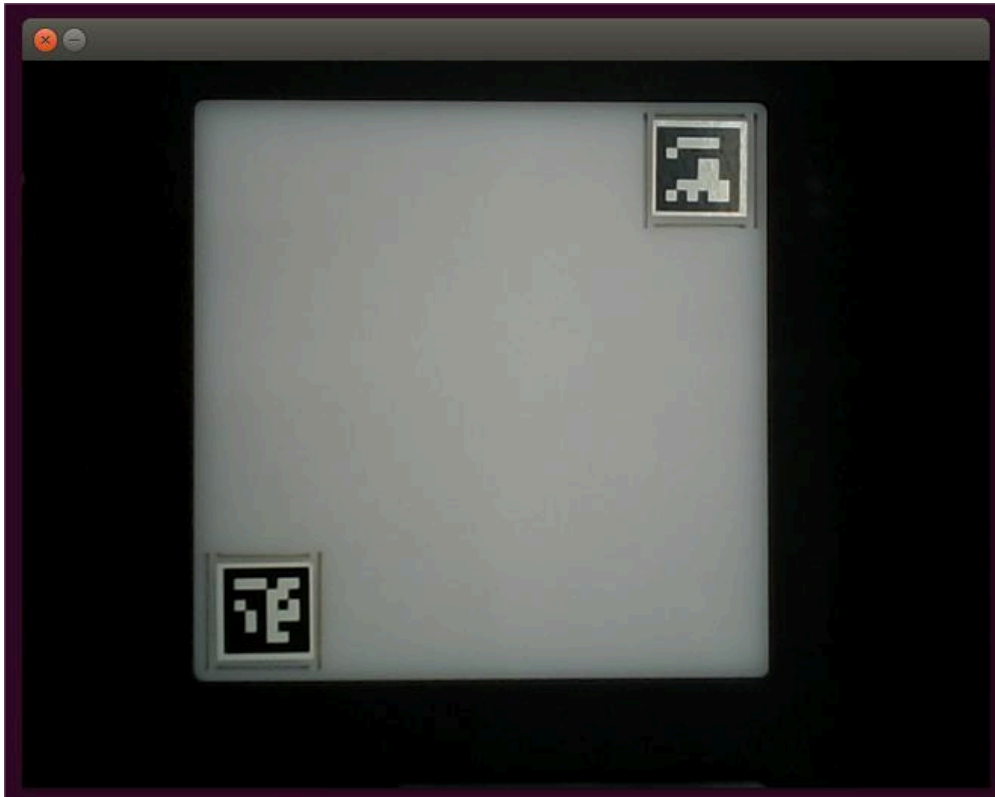
1 Adaptive robot

- Open a console terminal (shortcut key Ctrl+Alt+T), enter the target folder

```
cd ~/aikit_V2/AiKit_270Pi/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```



2 Add new image

M5 version:

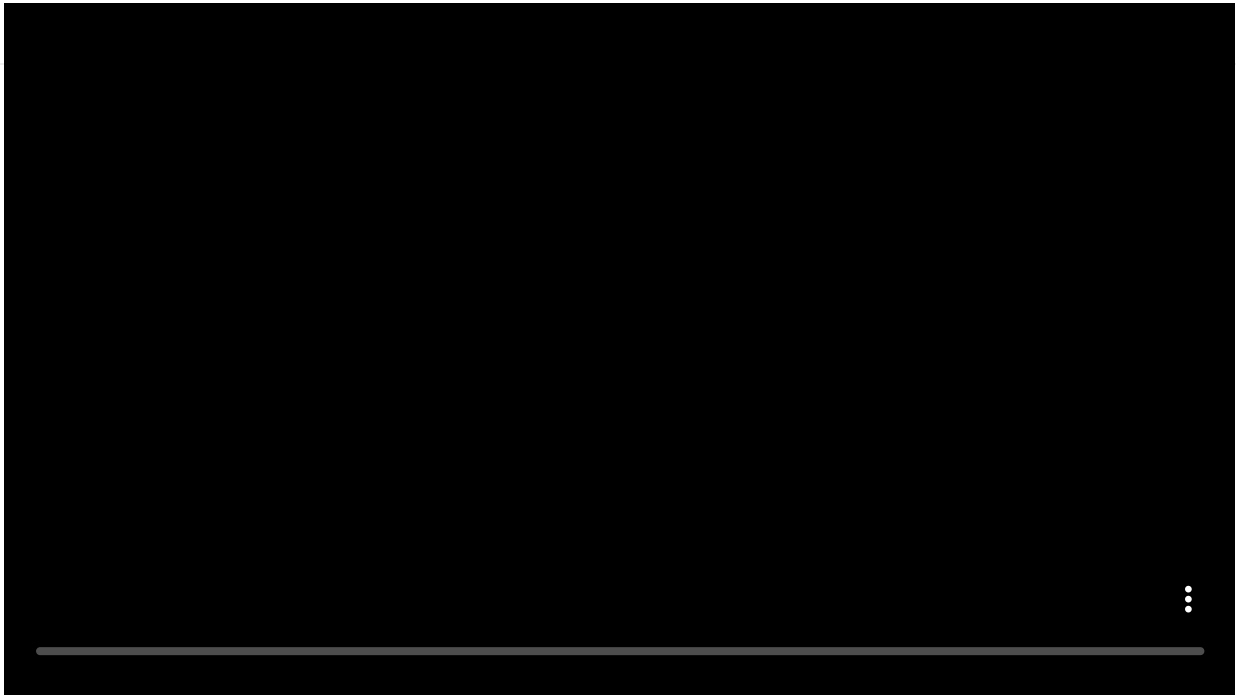
- Open a command prompt terminal (shortcut Win+R, enter "cmd" to access the terminal). Switch to the directory path where "aikit_V2" is located, then enter the target folder.

```
cd Desktop/aikit_V2/AiKit_270M5/
```

- Enter the following command to start the process of adding images.

```
python scripts/add_img.py
```

Raspberry Pi version:



- Open a console terminal (shortcut key Ctrl+Alt+T), enter the target folder

```
cd ~/aikit_V2/AiKit_270Pi/
```

- Enter the following command to start the process of adding images.

```
python scripts/add_img.py
```

- Operate according to the prompts entered by the terminal, and capture the image in the second image box that pops up.
- After the image capture area is completed, press the Enter key, and according to the terminal prompt, enter a number (1~4) to save to the folder corresponding to the image, and press the Enter key to save it to the corresponding folder.

3 Case Reproduction

M5 version:

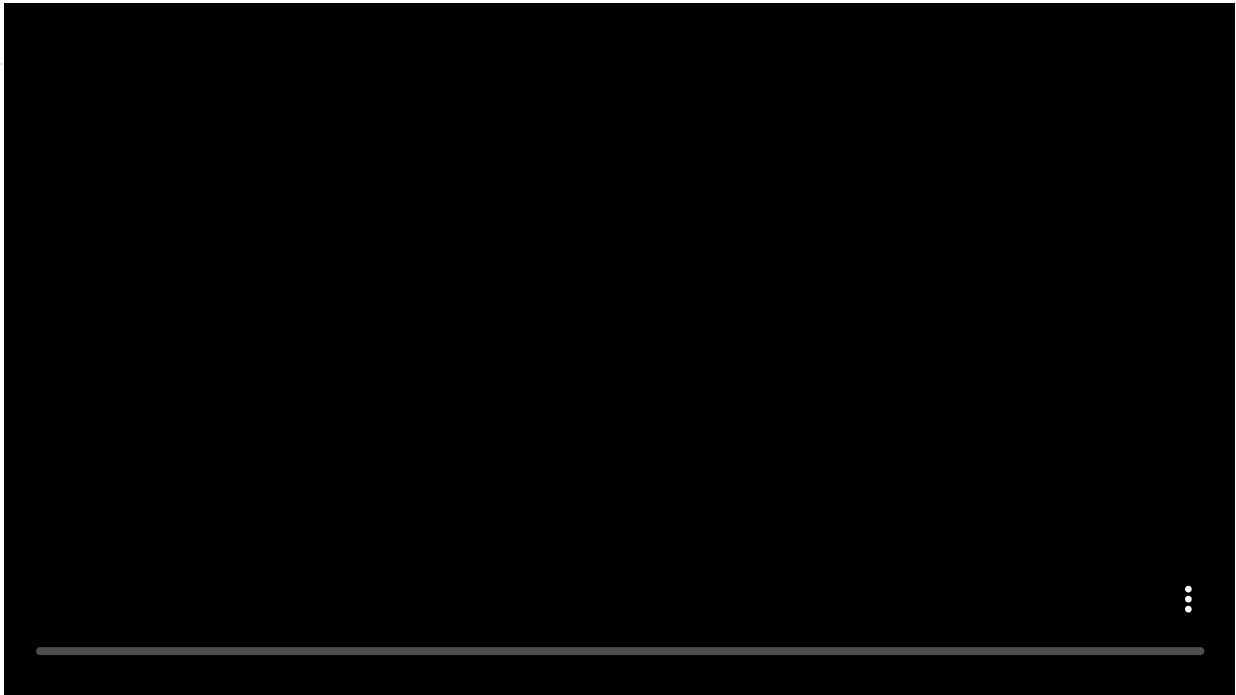
- Open a command prompt terminal (shortcut Win+R, enter "cmd" to access the terminal). Switch to the directory path where "aikit_V2" is located, then enter the target folder.

```
cd Desktop/aikit_V2/AiKit_270M5/
```

- Then enter the following command to start the image recognition program.

```
python scripts/aikit_img.py
```

Raspberry Pi version:



- Open a console terminal (shortcut key `Ctrl+Alt+T`), enter the following command to enter the target folder:

```
cd ~/aikit_V2/AiKit_270Pi
```

- Then enter the following command to start the image recognition program.

```
python scripts/aikit_img.py
```

- When the command terminal shows ok and the camera window can be opened normally, it means that the program runs successfully. At this time, the identifiable objects can be placed in the recognition area for recognition and capture.

Precautions

1. When the camera does not automatically frame the recognition area correctly, you need to close the program, adjust the position of the camera, and move the camera to the left or right.
2. If the command terminal does not display ok, and the picture cannot be recognized, you need to move the camera slightly backward or forward, and the program can run normally when the command terminal displays ok.
3. Since the recognition method has been changed, please make sure that there is a folder for storing pictures and pictures that need to be recognized under the code path. If not, you can use `python scripts/add_img.py` to add pictures.
4. OpenCV color recognition will be affected by the environment, and the recognition effect will be greatly reduced in a darker environment.

Recognize image cards

This case uses the eye-to-hand mode, uses the camera, loads the model data trained by Tensorflow through OpenCV, recognizes the image block and locates the position of the image block in the video. Through the relevant points, the space coordinate position of the block relative to the robot arm is calculated. Set up a set of related actions for the robot arm, and put the identified object into the bucket. In the following chapters, the code implementation process of the entire case will be introduced in detail.

1 Camera Adjustment

First, you need to use python to run OpenVideo.py under the aikit_V2 package. If the enabled camera is a computer camera, cap_num needs to be modified. For details, please refer to: **Precautions. Make sure that the camera completely covers the entire recognition area, and the recognition area is square** in the video, as shown in the figure below. If the recognition area does not meet the requirements in the video, the camera position needs to be adjusted.

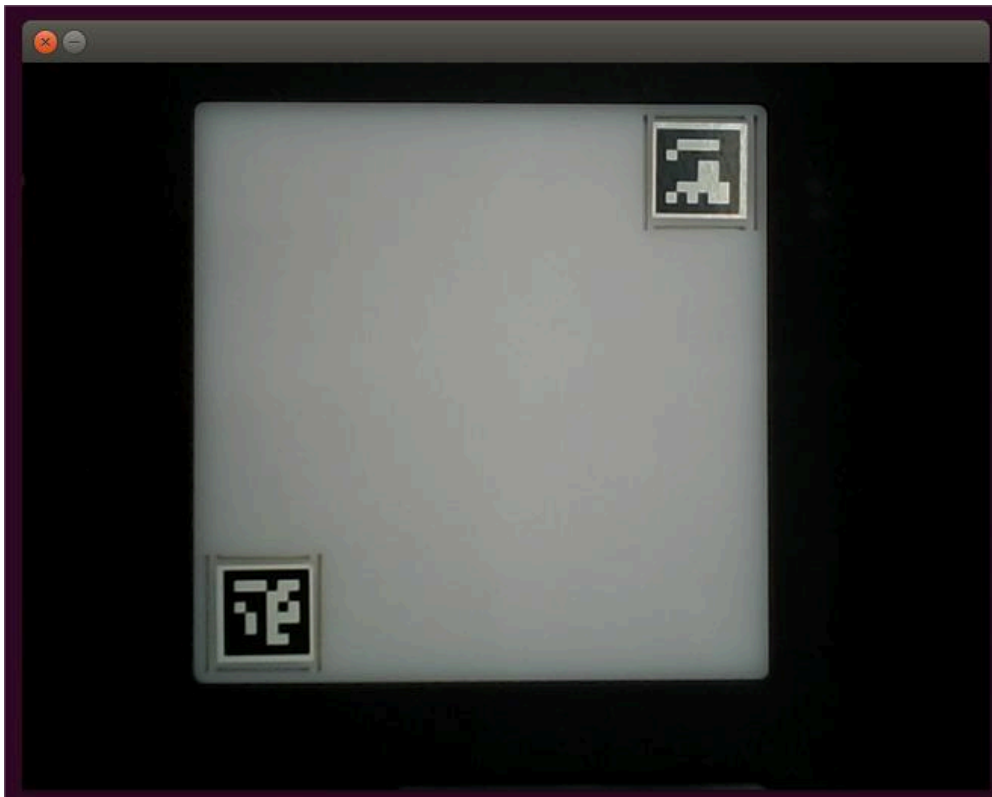
M5 version:

- Open a console terminal (shortcut key Win+R, enter cmd to enter the terminal), enter the following command to enter the target folder:

```
cd Desktop/aikit_V2/AiKit_260M5/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```



Raspberry Pi version:

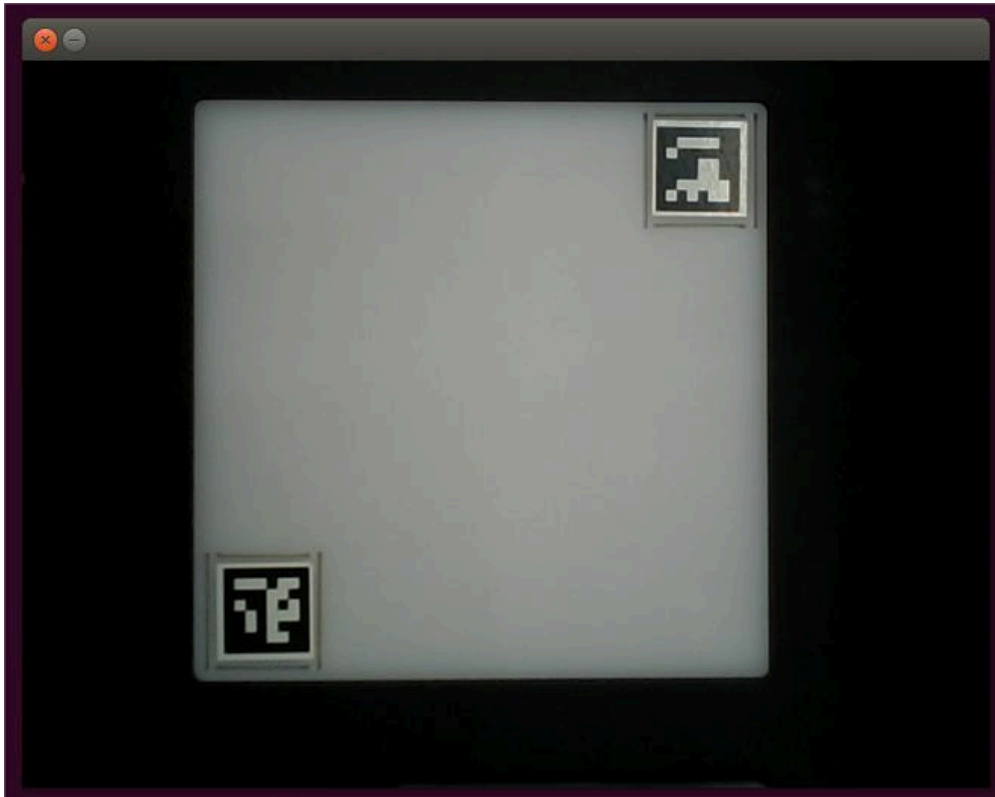
1 Adaptive robot

- Open a console terminal (shortcut key Ctrl+Alt+T), enter the target folder

```
cd ~/aikit_V2/AiKit_260PI/
```

- Enter the following command to open the camera for adjustment

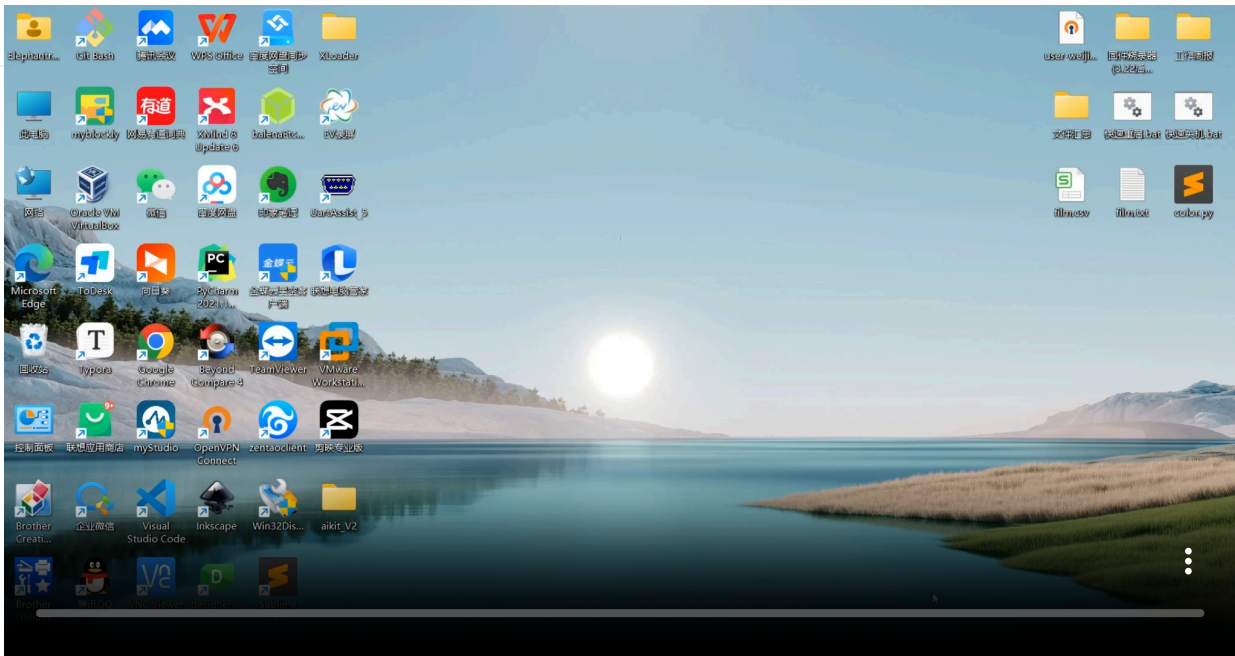
```
python scripts/OpenVideo.py
```



2 Add new image

M5 version:

1 Adaptive robot



- Open a console terminal (shortcut key Win+R, enter cmd to enter the terminal), enter the following command to enter the target folder:

```
cd Desktop/aikit_V2/AiKit_260M5/
```

- Enter the following command to start the process of adding images.

```
python scripts/add_img.py
```

Raspberry Pi version:

- Open a console terminal (shortcut key Ctrl+Alt+T), enter the target folder

```
cd ~/aikit_V2/AiKit_260PI/
```

- Enter the following command to start the process of adding images.

```
python scripts/add_img.py
```

- Operate according to the prompts entered by the terminal, and capture the image in the second image box that pops up.
- After the image capture area is completed, press the Enter key, and according to the terminal prompt, enter a number (1~4) to save to the folder corresponding to the image, and press the Enter key to save it to the corresponding folder.

3 Case Reproduction

M5 version:

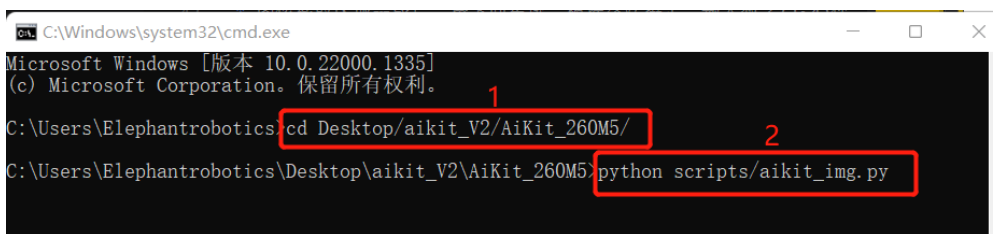


- Open a console terminal (shortcut key Win+R, enter cmd to enter the terminal), enter the following command to enter the target folder:

```
cd Desktop/aikit_V2/AiKit_260M5/
```

- Then enter the following command to start the image recognition program.

```
python scripts/aikit_img.py
```



Raspberry Pi version:

- Open a console terminal (shortcut key Ctrl+Alt+T), enter the target folder

```
cd ~/aikit_V2/AiKit_260PI/
```

- Then enter the following command to start the image recognition program.

```
python scripts/aikit_img.py
```

- When the command terminal shows ok and the camera window can be opened normally, it means that the program runs successfully. At this time, the identifiable objects can be placed in the recognition area for recognition and capture.

Precautions

1. When the camera does not automatically frame the recognition area correctly, you need to close the program, adjust the position of the camera, and move the camera to the left or right.
2. If the command terminal does not display ok, and the picture cannot be recognized, you need to move the camera slightly backward or forward, and the program can run normally when the command terminal displays ok.
3. Since the recognition method has been changed, please make sure that there is a folder for storing pictures and pictures that need to be recognized under the code path. If not, you can use `python scripts/add_img.py` to add pictures.
4. OpenCV color recognition will be affected by the environment, and the recognition effect will be greatly reduced in a darker environment.

yolov5 image recognition

This case uses the eye-to-hand mode, uses the camera to take pictures, loads the yolov5 model data through OpenCV, recognizes the image block and locates the position of the image block in the recognition area. Through the relevant points, the space coordinate position of the block relative to the robot arm is calculated. Set a set of related actions for the robot arm, and put the recognized object into the corresponding area.

1 Camera Adjustment

First, you need to use python to run OpenVideo.py under the aikit_V2 package. If the enabled camera is a computer camera, cap_num needs to be modified. For details, please refer to: **Precautions. Make sure that the camera completely covers the entire recognition area, and the recognition area is square** in the video, as shown in the figure below. If the recognition area does not meet the requirements in the video, the camera position needs to be adjusted.

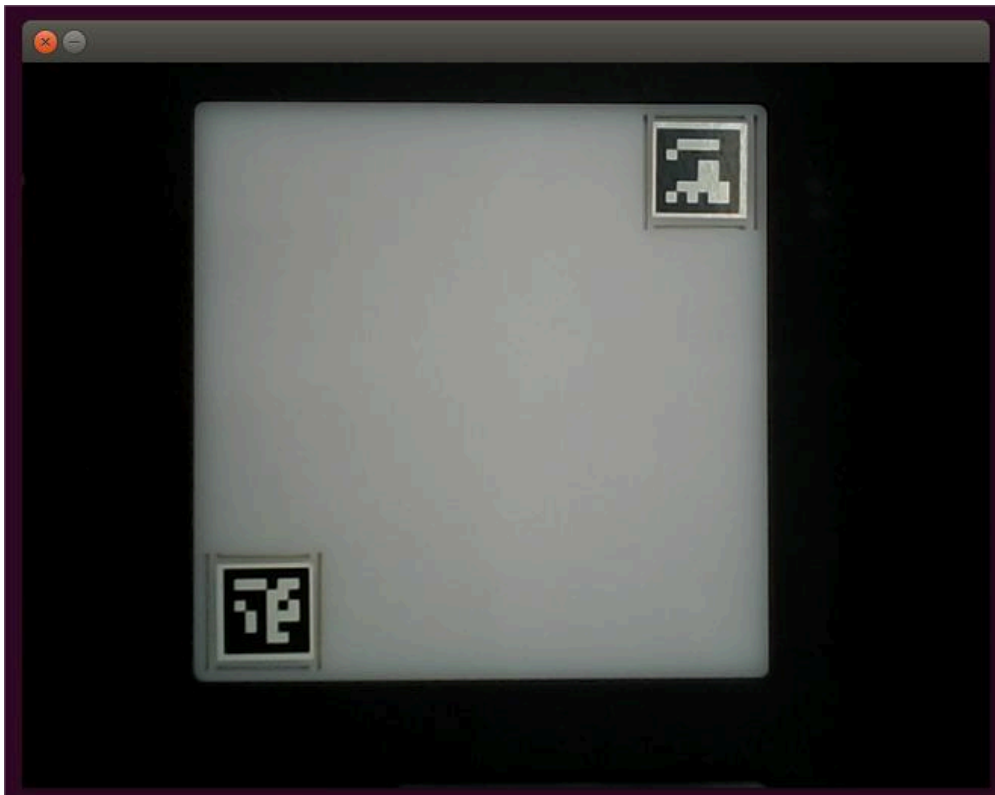
M5 version:

- Open a command prompt terminal (shortcut Win+R, enter "cmd" to access the terminal). Switch to the directory path where "aikit_V2" is located, then enter the target folder.

```
cd Desktop/aikit_V2/AiKit_280M5/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```



Raspberry Pi version:

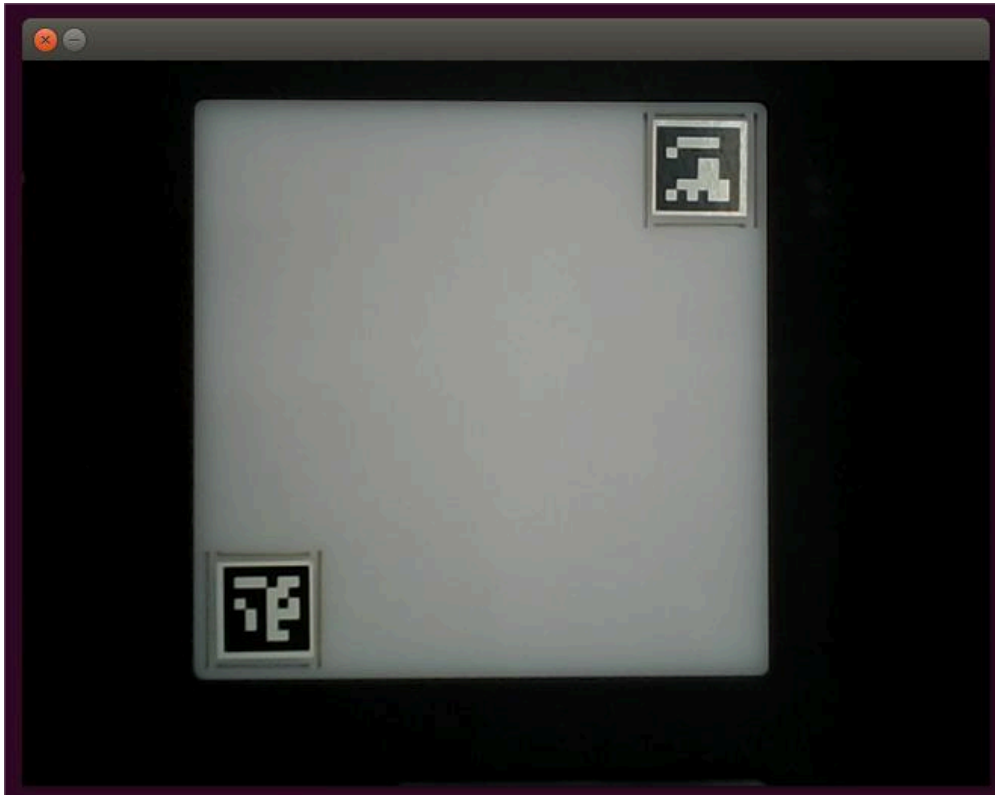
- Open a console terminal (shortcut key Ctrl+Alt+T), enter the target folder

1 Adaptive robot

```
cd ~/aikit_V2/AiKit_280PI/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```



2 Case Reproduction

M5 version:

- Open a command prompt terminal (shortcut Win+R, enter "cmd" to access the terminal). Switch to the directory path where "aikit_V2" is located, then enter the target folder.

```
cd Desktop/aikit_V2/AiKit_280M5/
```

- Then enter the following command to start the yolov5 recognition program.

```
python scripts/yolov5_img.py
```

Raspberry Pi version:

1 Adaptive robot



- Open a console terminal (shortcut key `Ctrl+Alt+T`), enter the following command to enter the target folder:

```
cd ~/aikit_V2/AiKit_280PI
```

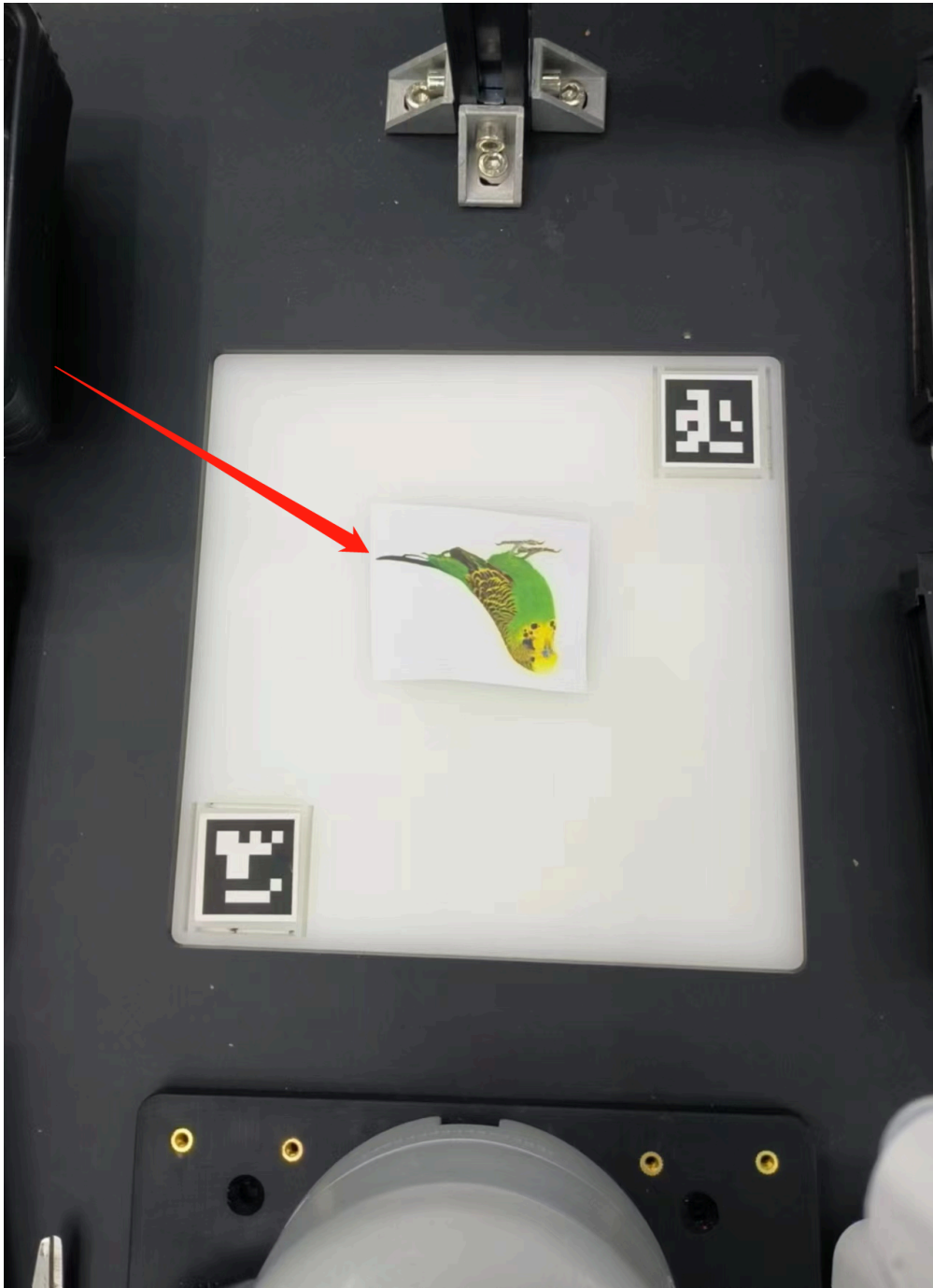
- Then enter the following command to start the yolov5 recognition program.

```
python scripts/yolov5_img.py
```

- When the camera window can be opened normally, it means that the program is running successfully. At this time, you can place the identifiable object in the recognition area, and take pictures and intercept according to the terminal prompts. When the command terminal shows ok, it means that yolov5 is recognizing and grasping, and the robot arm is completed. After a single capture, press the space bar for the next image recognition.

Precautions

1. If the command terminal does not display ok, and the picture cannot be recognized, you need to move the camera slightly backward or forward, and the program can run normally when the command terminal displays ok.
2. In order to better recognize the picture, you need to **reverse the object in the recognition area**.



yolov5 image recognition

This case uses the eye-to-hand mode, uses the camera to take pictures, loads the yolov5 model data through OpenCV, recognizes the image block and locates the position of the image block in the recognition area. Through the relevant points, the space coordinate position of the block relative to the robot arm is calculated. Set a set of related actions for the robot arm, and put the recognized object into the corresponding area.

1 Camera Adjustment

First, you need to use python to run OpenVideo.py under the aikit_V2 package. If the enabled camera is a computer camera, cap_num needs to be modified. For details, please refer to: **Precautions. Make sure that the camera completely covers the entire recognition area, and the recognition area is square** in the video, as shown in the figure below. If the recognition area does not meet the requirements in the video, the camera position needs to be adjusted.

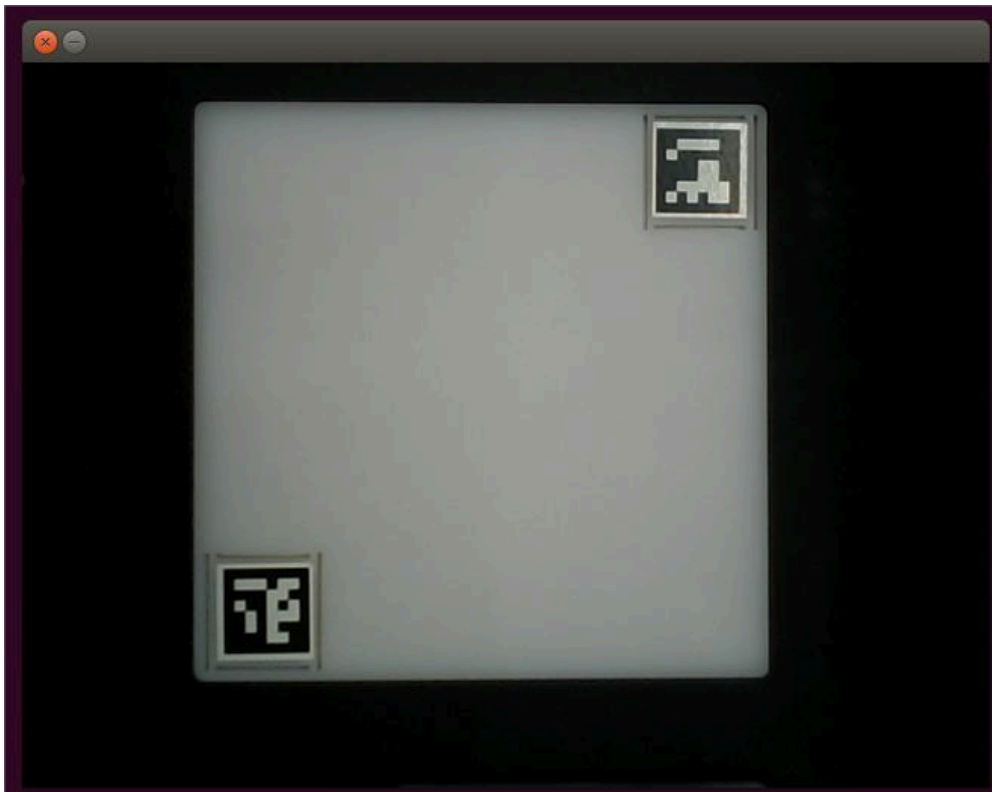
M5 version:

- Open a command prompt terminal (shortcut Win+R, enter "cmd" to access the terminal). Switch to the directory path where "aikit_V2" is located, then enter the target folder.

```
cd Desktop/aikit_V2/AiKit_270M5/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```



Raspberry Pi version:

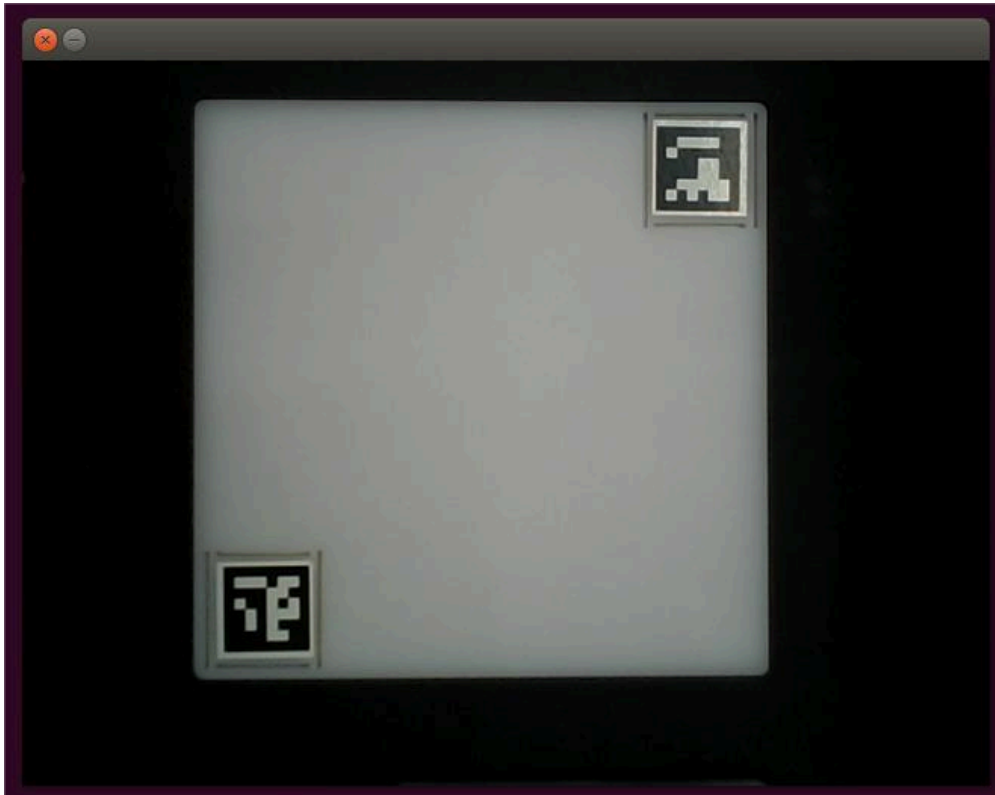
- Open a console terminal (shortcut key Ctrl+Alt+T), enter the target folder

1 Adaptive robot

```
cd ~/aikit_V2/AiKit_270Pi/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```



2 Case Reproduction

M5 version:

- Open a command prompt terminal (shortcut Win+R, enter "cmd" to access the terminal). Switch to the directory path where "aikit_V2" is located, then enter the target folder.

```
cd Desktop/aikit_V2/AiKit_270M5/
```

- Then enter the following command to start the yolov5 recognition program.

```
python scripts/yolov5_img.py
```

Raspberry Pi version:

1 Adaptive robot



- Open a console terminal (shortcut key `Ctrl+Alt+T`), enter the following command to enter the target folder:

```
cd ~/aikit_V2/AiKit_270Pi
```

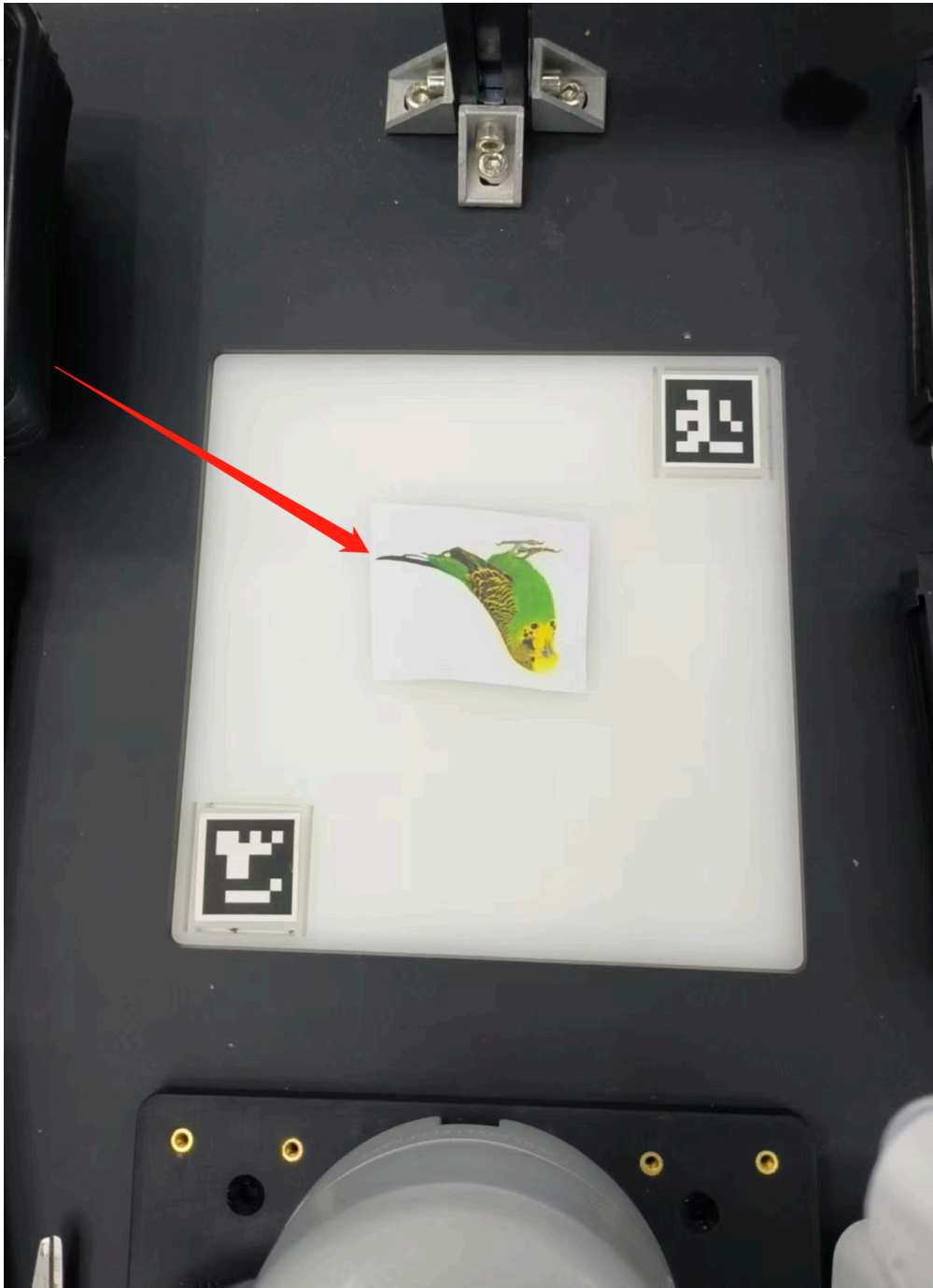
- Then enter the following command to start the yolov5 recognition program.

```
python scripts/yolov5_img.py
```

- When the camera window can be opened normally, it means that the program is running successfully. At this time, you can place the identifiable object in the recognition area, and take pictures and intercept according to the terminal prompts. When the command terminal shows ok, it means that yolov5 is recognizing and grasping, and the robot arm is completed. After a single capture, press the space bar for the next image recognition.

Precautions

1. If the command terminal does not display ok, and the picture cannot be recognized, you need to move the camera slightly backward or forward, and the program can run normally when the command terminal displays ok.
2. In order to better recognize the picture, you need to **reverse the object in the recognition area**.



yolov5 image recognition

This case uses the eye-to-hand mode, uses the camera to take pictures, loads the yolov5 model data through OpenCV, recognizes the image block and locates the position of the image block in the recognition area. Through the relevant points, the space coordinate position of the block relative to the robot arm is calculated. Set a set of related actions for the robot arm, and put the recognized object into the corresponding area.

1 Camera Adjustment

First, you need to use python to run OpenVideo.py under the aikit_V2 package. If the enabled camera is a computer camera, cap_num needs to be modified. For details, please refer to: **Precautions. Make sure that the camera completely covers the entire recognition area, and the recognition area is square** in the video, as shown in the figure below. If the recognition area does not meet the requirements in the video, the camera position needs to be adjusted.

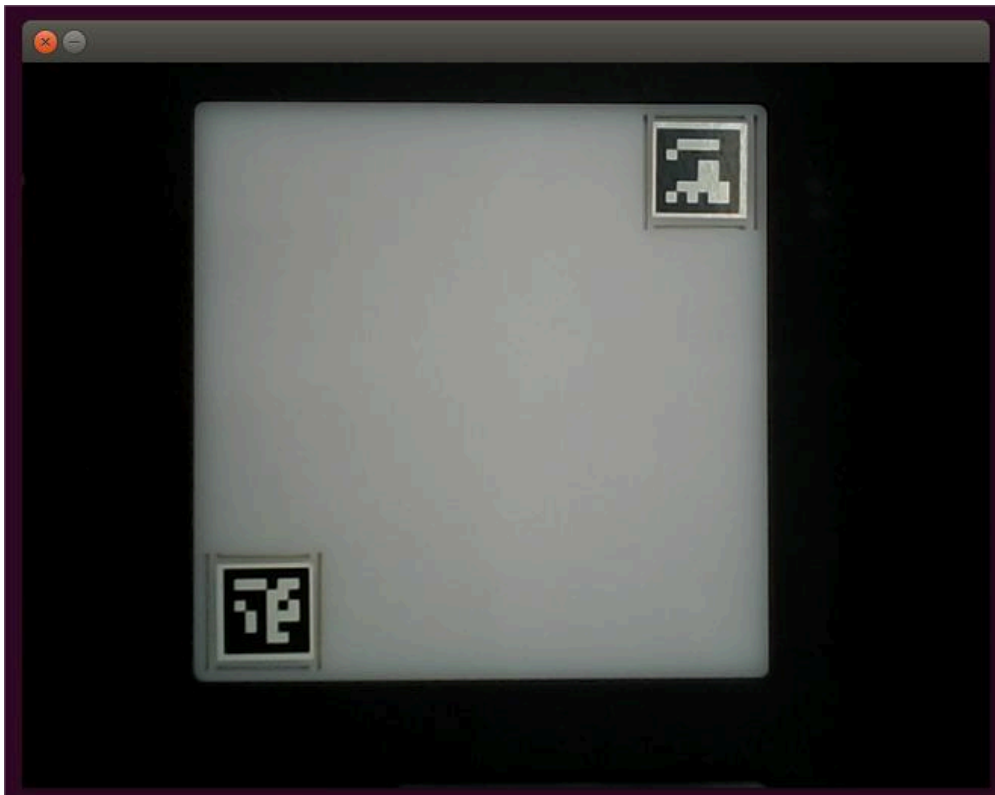
M5 version:

- Open a command prompt terminal (shortcut Win+R, enter "cmd" to access the terminal). Switch to the directory path where "aikit_V2" is located, then enter the target folder.

```
cd Desktop/aikit_V2/AiKit_260M5/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```



Raspberry Pi version:

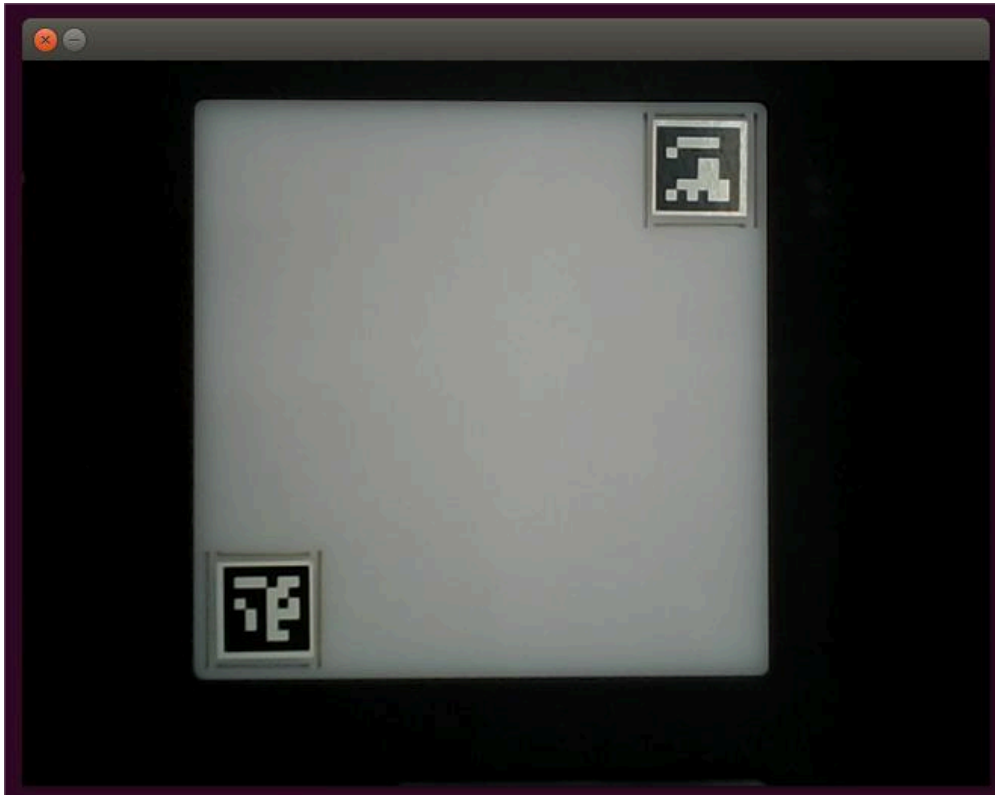
- Open a console terminal (shortcut key Ctrl+Alt+T), enter the target folder

1 Adaptive robot

```
cd ~/aikit_V2/AiKit_260PI/
```

- Enter the following command to open the camera for adjustment

```
python scripts/OpenVideo.py
```



2 Case Reproduction

M5 version:

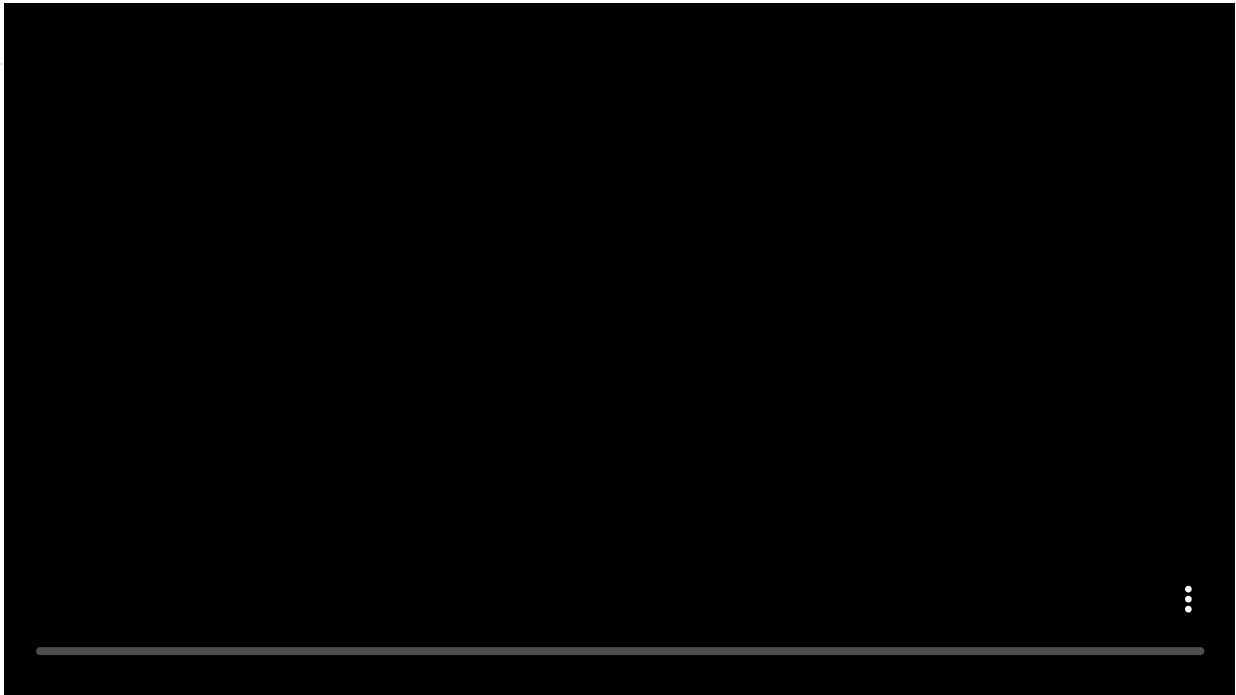
- Open a command prompt terminal (shortcut Win+R, enter "cmd" to access the terminal). Switch to the directory path where "aikit_V2" is located, then enter the target folder.

```
cd Desktop/aikit_V2/AiKit_260M5/
```

- Then enter the following command to start the yolov5 recognition program.

```
python scripts/yolov5_img.py
```

Raspberry Pi version:



- Open a console terminal (shortcut key `Ctrl+Alt+T`), enter the following command to enter the target folder:

```
cd ~/aikit_V2/AiKit_260PI
```

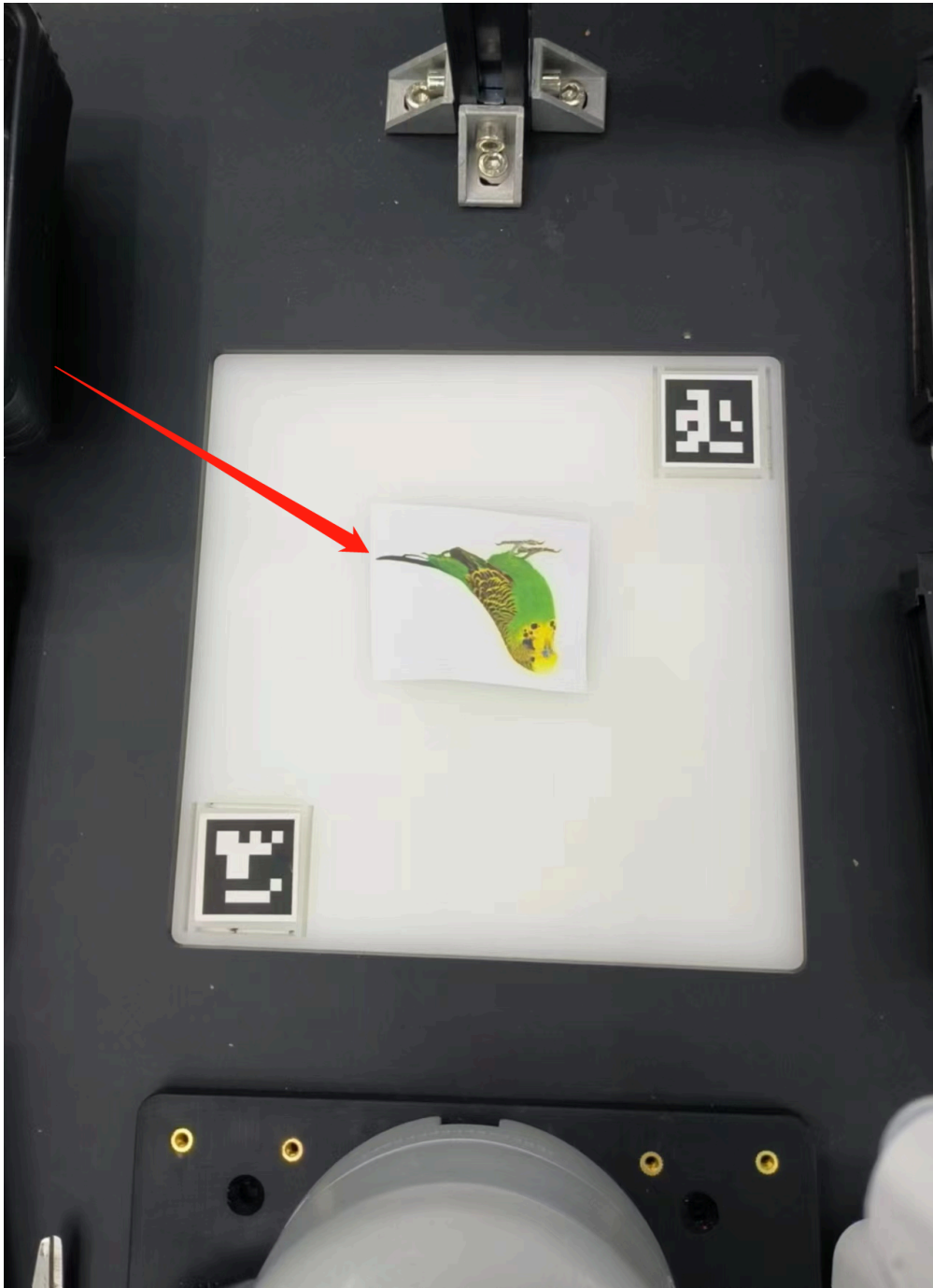
- Then enter the following command to start the yolov5 recognition program.

```
python scripts/yolov5_img.py
```

- When the camera window can be opened normally, it means that the program is running successfully. At this time, you can place the identifiable object in the recognition area, and take pictures and intercept according to the terminal prompts. When the command terminal shows ok, it means that yolov5 is recognizing and grasping, and the robot arm is completed. After a single capture, press the space bar for the next image recognition.

Precautions

1. If the command terminal does not display ok, and the picture cannot be recognized, you need to move the camera slightly backward or forward, and the program can run normally when the command terminal displays ok.
2. In order to better recognize the picture, you need to **reverse the object in the recognition area**.



Common Problem

In this part, some common driver-related problems, software-related problems and hardware-related problems are listed.

How To Ask Questions Gracefully [1 How To Ask Questions Gracefully](#)

Drive Related [2 Drive Related](#)

Software Problem [3 Software Problem](#)

Hardware Problem [4 Hardware Problem](#)

If you have purchase intention or any parameter questions, please send an email to this mailbox.
sales@elephantrobotics.com

If the listed problems can't help you solve and you have more after-sales questions, please send an email to this mailbox. support@elephantrobotics.com

How To Ask Questions Gracefully

1 When asking questions in various places, you will find several phenomena:

- No answer after asking question.
- It took a long time for the question to be answered.
- The other party always despise me and do nothing.

2 Before asking questions, make sure you have studied chapters three or four.

Chapters **3 and 4** of this document are the basis for using the **my series**. No matter whether you have development experience or not, be sure to read and operate from front to back.

Many problems will be solved during this process. Do not ask questions in QQ groups, forums, issues, or emails at the very beginning. Many problems explained in the document at the beginning may not be answered in a timely by the community. To save everyone's time, and for a better community environment, everyone can grow together better, please understand each other.

3 When asking questions, try to do the following, which will greatly increase the chances of a quick resolution:

To figure out what's going on and what I did, including:

- What effect and what function do I want to achieve?
- In order to achieve this effect, how do I do it and what is the detailed process?
- In the process of implementation, what error occurred and what is the phenomenon (for example, what is the error reported, what is the **complete** error content?)
- Have I read the error message carefully, and is there any indication of the cause and solution of the error in the error message?
- Based on these error messages, think carefully, can I solve the problem?
- Can I find a solution to the problem by searching the documentation, issues, and using a search engine?

4 If you really can't solve the problem yourself, you need to ask someone for help, and you need to consider:

- Who to ask, where to ask, and who has a better chance of answering my question? And how about real-time?
- What data and phenomena should I give him so that he is willing to help me solve the problem quickly?
 - Provide my purpose (to let the answerer know what you are doing).
 - Provide the complete implementation process and the phenomena that occur in the process (for the answerer to follow your process to do it again, that is, the problem recurs).
 - Give the wrong place, indicate where the phenomenon or result is different from what you expected! (Let the answerer know, where did not meet expectations).
 - For the error information that appears, it needs to be complete, as many screenshots as possible, more logs, don't be stingy to take a small picture, or give a part of the log (because the answerer may not have

done this for a long time, they forgot some details, and they need to rely on screenshots and complete logs to quickly recall. And according to the detailed logs, they can quickly locate where the problem is).

- How to ask questions with a more sincere attitude, even if I don't understand anything, everyone is willing to answer.

5 Question Template

Try to ask questions as elegantly as possible, without adding redundant modal particles, complaining vocabulary, considering every word and punctuation, thinking about the problem from the perspective of the answerer, and how to let the answerer help me solve the problem quickly. Too few words will make it difficult to describe the question, and too many words will make the answerer impatient.

6 Title

No matter where the question is asked (including QQ group), prepare a title of about 30 words for your question, clarifying the central idea of the question, including:

- It is necessary to clearly distinguish the type of problem, whether it is a question to ask, a bug submission, or an experience sharing and so on. Let answerer instantly locate what you want to do on a screen full of text.
- One sentence to clarify the core of the problem, such as run the camera sample program, report an error reset fail, it may be a hardware problem.

So the title after synthesis can be like this:

- [Mycobot question] Running the camera sample program, the error reset fail is reported. Could it be a hardware problem?

Try to **don't** appear in such a title:

- Why is my board not working again?
- Why is my code not working?
- Why is my screen black?
- [Mycobot question] I received the development board, why is the development board screen red and has a small line of text?
- I ran a program and something went wrong.

You can ask this:

- [Mycobot question] My board can't start after I connected the power supply in reverse, how can I tell where the board is burnt, and if so, how can I save it?

7 Content

First stand on the answerer's point of view, if asked a question:

- First of all, I need to know what the other party wants to do and what the goal is to achieve.
- In order to achieve this goal, what steps did he refer to?
- In fact, what specific steps were used, and then at which step the problem occurred, so that I can try to reproduce the phenomenon according to his steps. If this problem seems to be difficult to solve and there are no steps to reproduce it, it may take a lot of time to reproduce, so let's put it aside and solve other problems first.
- What was the specific problem that arose, and if he only stated the problem, how would I know what was wrong with him, maybe a physical discomfort? So this is very important, I need to ask him to explain the phenomenon of the problem and indicate what is different from the expected, otherwise I have to guess what is the difference between the comparison and the expected, and the time to solve the problem has increased.

- If there is a problem, I may need his log file, so that I can analyze the source code according to the log, otherwise it may be difficult to solve the problem, then this problem can be looked at later.
-

In summary, you can ask the following questions:

- Elaborate on your goals, what you want to do, and what the phenomenon should look like.
- Is there any documentation, code, or tutorial I refer to?
- How to reproduce the error: how to do it in detail, write each step in detail until the problem occurs.
- Elaborate on what happened when the error occurred, and how it was different from what was expected, and needed to prove that the problem did occur.
- Attach log files, as well as screenshots, or even videos, logs and screenshots must be complete, not just a small part, the answerer may find some problems you did not notice from your full log and screenshots, this is very important.
- In addition, pay attention to the format when pasting the code. Do not display it in a mess after pasting, and it cannot be seen. Try to copy it and run it directly.
- Finally, you need to thank the community friends who answered the question.

Drive Related

1 About python

Q: `send_coords([x,y,z,rx,ry,rz], speed, 1)` What do the parameters in this API mean? What do rx, ry, and rz correspond to Euler angles? What is the rotation order of Euler angles? And what is the value range of each parameter?

- A: The parameters in the previous array are the coordinates of the end of the myCobot, speed is the speed, and the last parameter is the motion mode. rx, ry, and rz should correspond to rpy, that is, corresponding to roll, pitch, and yaw respectively. The order of Euler angles is zyx, and zyx is its own coordinate. The value range of x, y, z is -300~300.00 (the value range is undefined, if the range is exceeded, the inverse kinematics no solution prompt will be returned), and the value range of rx, ry, and rz is -180~180.

Q: Are sample tutorials for the python API provided?

- A: Yes, there is test code in the test folder of github, which can be executed with the terminal.
<https://github.com/elephantrobotics/pymycobot/tree/main/demo>

Q: How does the python drag teaching demo of mycobot280-Pi work?

- A: Run in the terminal, please enter 1000000 baud rate.

Q: Mycobot280-Pi uses python zero to calibrate the demo program, why is there an error?

- A: The atom firmware is not burned, please burn the atom firmware before running the program.

Q: Is the python API of different versions of myCobots the same?

- A: The API is the same.

2 About ROS

Q: How do a microcontroller-based myCobot and a microprocessor-based myCobot run ROS?

- A: The use of ROS for a microcontroller-based myCobot is currently on Ubuntu, and you can also develop your own ROS. A microprocessor-based myCobot have its own ROS environment and can be used directly.

Q: Can a microprocessor-based myCobot connect to a PC to use ROS and moveit?

- A: The current open source data is not controlled by direct communication. It can be implemented by modifying the existing node files through ros + socket.

Q: Can you provide files and programming examples of the rviz model?

- A: It is available on our github. https://github.com/elephantrobotics/mycobot_ros

Q: Can myAGV create maps remotely?

- A: You can create maps remotely. We are the control interface of open source ROS.

Q: Why is the error permission denied: '/dev/ttyUSB0' reported when using ROS to start the rviz model file?

- A: It is because the serial port permission is not given. You should type `sudo chmod 777 port name` in the terminal.

Q: Why is the error `init()` takes exactly 2 arguments (3 given) reported when running the slider control and model follow commands of ROS?

- A: The pymycobot library is not installed and started.

Q: Q: When using ROS, why is the myCobot angle inconsistent with the model angle after opening the rviz model?

- A: It is very likely that the zero position of the myCobot is not calibrated, and the zero position of the myCobot needs to be calibrated.

Software Problem

Q: Why can't my compiler find the corresponding device?

- A: You need to build a development environment and install the corresponding project library before you can develop the device.

1 About mystudio

Q: What is mystudio?

- A: It is the firmware burner.

Q: Why can't the device run normally after I burn the firmware to the ATOM terminal?

- A: The firmware of the ATOM terminal needs to use our factory firmware. Other unofficial firmware cannot be changed during use. If the device accidentally burns other firmware, you can use the "myCobot firmware burner" to select the ATOM terminal - select the serial port - select the ATOMMAIN firmware for ATOM terminal for burning.

Q: Can the drag teaching in the minirobot firmware control the gripper?

- A: It is achievable.

Q: Why can't drag teaching after the minirobot firmware is burned?

- A: First, check whether the M5Stack-basic firmware and atom firmware have been burned, whether the burned firmware corresponds to the requirements to be realized, and whether the burned firmware is the latest version. The bottom M5Stack-basic burns the minirobot, and the top atom burns the atommain.

Q: What should I do if myCobot's serial port is not recognized on mystudio?

- A: If your computer equipment does not prompt for the connected myCobot, please install the serial port driver first.

Q: Can the track recorded by drag teaching be saved to the card?

- A: Currently it cannot be saved to the memory card. And drag teaching can only save one path at a time, and the next recording will overwrite the previous action.

2 About Roboflow

Q: Can robotstudio software be used for programming?

- A: Our own industrial programming software roboflow can be used. RobotStudio is owned by ABB.

Q: What is the reason for the Quickmove of the roboflow software beyond the limit?

- A: It may be that a joint or multiple joints exceed the limit.

Q: How does roboflow load the already written program?

- A: After logging in, select program robot and click load program. Directly clicking run program cannot be used, only pro600 can.

Q: When the pro600 uses roboflow, the log shows that 456 joints are stopped. Is this normal?

- A: This is normal.

3 About mycobot phone controller

Q: What version of firmware should myCobot phone controller app be programmed with?

- A: You need to burn the atom firmware version atommain 2.5 in mystudio.

4 About myblockly

Q: Why does a pop-up box always appear when myblockly is running?

- A: Before running the myblockly program, close the serial port occupation.

5 About ROS1

Q: When a terminal switches to `~/catkin_ws/src` and uses git to install and update `mycobot_ros`, it appears that the target path "mycobot_ros" already exists. Why?

- A: A `mycobot_ros` package already exists in `~/catkin_ws/src`, so you need to delete it beforehand and run git again.

Q: When rosrn is running, a terminal error is reported saying `could not open port /dev/ttyUSB0: Permission: '/dev/ttyUSB0'`. Why?

- A: The serial port permission is insufficient. Enter `sudo chmod 777 /dev/ttyUSB0` to grant the permission.

Q: Why can't ros programs run in vscode?

- A: Since the vscode terminal cannot be loaded into the ros environment, it needs to run on the system terminal.

Q: Unable to register with master node [http://localhost:11311]: Unable to register with master node. master may not be running yet. Will he keep trying ?

- A: Before running the ros program, start the ros node and enter `roscore` on the terminal.

Q: When rosrn runs, the terminal error shows `could not open port /dev/ttyUSB0: No such file or directory: '/dev/ttyUSB1'`. Why?

- A: The serial port is incorrect. Check the actual serial port of the manipulator. To view the value, run `ls /dev/tty*`.

Q: Failure in Ubuntu18.04 `catkin_make` building code, terminal prompt `Project 'cv_bridge' specifies '/usr/include/opencv' as an include dir, which is not found`. Wait for an error message

- A: The opencv path in the configuration file is inconsistent with the actual path in the system. Need to use sudo to modify the configuration file (path is `/opt/ros/melodic/share/cv_bridge/cmake/cv_bridgeConfig.cmake`), the system actual opencv path in the `/usr/include/` directory.

Q: I cloned the mycobot_ros package and ran the rosrn program directly. package 'mycobot_280' not found or can't find the file?

A: The newly cloned mycobot_ros needs to build code to compile to the ros environment. The input terminal

```
cd ~/catkin_ws/  
catkin_make  
source devel/setup.bash
```

Q: After compiling, why does the following error occur when a new terminal runs launch command?

```
u20@u20-VirtualBox:~/catkin_ws$ roslaunch mybuddy_socket slider_control.launch  
RLException: [slider_control.launch] is neither a launch file in package [mybuddy_socket] nor is [mybuddy_socket] a launch file name  
The traceback for the exception was written to the log file
```

- A1: The system has not added ros environment variable, so every time a new terminal is started, the source is required:

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

- A2: ros environment variable is added to the system. There is no need to execute source after starting a new terminal:

```
# The noetic is Ubuntu20.04 system  
echo "source /opt/ros/noetic/setup.bash" >> ~/.bashrc  
source ~/.bashrc
```

- A3: The file name in the instruction may not be the same as the file name in the mycobot_ros package. Please carefully check whether the instruction is incorrect.

Questions about the Structure

Q1: What are the maximum angle of Joint 1 and Joint 5 of myCobot?

A1: The maximum angle of Joint 1 and Joint 5 are 160° both clockwise and anticlockwise. Remember to rotate the joints gently. Do not force a rotation when the joint reach its maximum angle.

Q2: What controls the six steering engine?

A2: They are controlled by Atom at the top of the robot.

Q3: What functions Atom performs?

A3: It serves to control the robot by algorithm, such as forward, inverse kinematics and coordinate switching. Atom is temporarily not open-sourced.

Questions about the Communication

Q1: Why the screen has no display although the robotic arm has been connected with HDMI cable? Does it need to download a port driver?

A1: Check whether the connection is correct and the power has been turned on. Try to use another interface and plunge into the interface stably. There is no need to download a port driver.

Q2: What are the versions of communication interface supported by different robotic arms?

A2: The robotic arms of micro-processor supports TCP communication, and the robotic arms of micro-controller supports USB-port communication.

Q3: How much is the frequency of the communication of robotic arm?

A3: 10-20Hz.

Questions about Parameters

Q1: What's the unit of measurement of the speed of robotic arm?

A2: 180°/s.

Solutions to Hardware Problems

1. How to deal with the shaking of robotic arms?

Step 1: Burn the latest version of ATOM via myStudio.

Step 2: Upgrade pymycobot. Click on Win+R, and type cmd to enter in the terminal. Type pip install pymycobot --upgrade --user and then click on Enter.

Step 3: Go to [GitHub](#) to download pid_read_write.py. Reset each parameter of the steering engine according to the prompt message. And then, operate the system again.

Notice: The parameters of each version of robot are different. Set the parameters according to the data given in the pictures below.

```

25  # 参数对应地址 corresponding parameter
26  data_id = [21, 22, 23, 24, 26, 27]
27  # 修改后的参数 reset parameters
28  data     = [32, 8, 0, 0, 3, 3]
29

```

Parameter	Connotation	myPalletizer 260			
		J1/J3	J1/J3	J2	J4
21	location loop P ratio coefficient	25	32	32	32
22	location loop D differentia coefficient	25	32	32	32
23	location loop I integral coefficient	0	0	0	0
24	minimun starting force	0	16	3	0
26	clockwise insensitive area	3	1	1	0
27	anticlockwise insensitive area	3	1	1	0

Parameter	Connotation	mechArm 270			
		J1/J3	J2	J4/J6	J5
21	location loop P ratio coefficient	32	32	25	25
22	location loop D differentia coefficient	8	8	25	25
23	location loop I integral coefficient	0	0	1	1
24	minimun starting force	0	0	0	0
26	clockwise insensitive area	3	1	3	3
27	anticlockwise insensitive area	3	1	3	3

Parameter	Connotation	myCobot 280		
		J1-J3	J4	J5-J6
21	location loop P ratio coefficient	32	25	25
22	location loop D differentia coefficient	8	25	25
23	location loop I integral coefficient	0	1	1
24	minimun starting force	0	0	0
26	clockwise insensitive area	3	3	3
27	anticlockwise insensitive area	3	3	3

Parameter	Connotation	myCobot pro 320		
		J1、J2、J3	J2	J5-J6
21	location loop P ratio coefficient	32	32	25
22	location loop D differentia coefficient	8	8	25
23	location loop I integral coefficient	0	0	1
24	minimun starting force	0	0	0
26	clockwise insensitive area	3	3	3
27	anticlockwise insensitive area	3	3	3

Questions about Gripper & End

Q1: Can the gripper close completely?

A1: Due to the thickness of the gripper, it cannot close completely. You resort to a shim to make it completely closed.

Q2: What is the kind of communication of adaptive gripper?

A2: It is TTL communication.

Q3: What is the communication of the end of myCobot 320?

A3: 485 communication interface.

Q4: How to fix a camera at the end of a robotic arm?

A4: A flange is required to fix a camera.

Other Questions

Q1: Why the electric engine powers off during usage?

A1: Because the engine is too hot to be used. Please wait for a few minutes and reuse again.

Q2: Does the robotic arms support the development based on Android?

A2: We doesn't support Android development. If you want to develop it by yourself, we can provide you with port protocol.

Q3:What is the function of the USB carried by the PI version?

A3: It serves to charge the robotic arm.

Background knowledge

Assessing Prior Knowledge



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](#)
- [Electric](#)
- [Motors](#)

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.

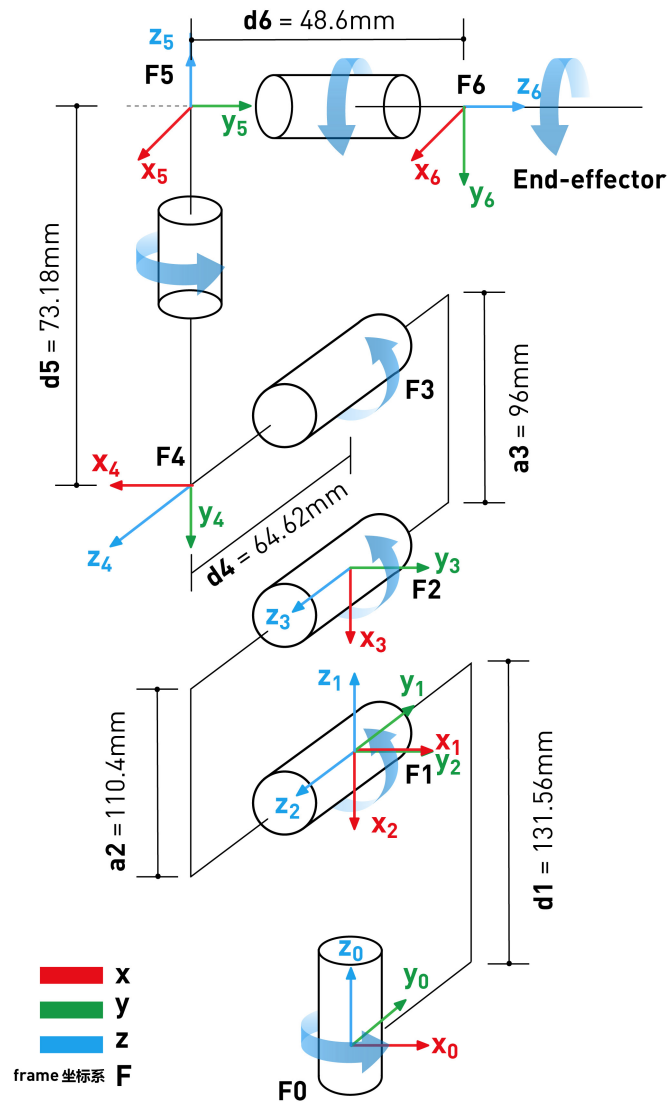
Inverse Kinematic

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 3×1 . Since many coordinates are often defined in the world coordinate system, a piece of information must be attached to the position vector indicating which coordinate system is defined. In this book, the position vector has a leading superscript to indicate the coordinate 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 $d=0.0$, the direction of X axis is the same as that of X_{n-1} axis, the origin position of coordinate system $\{n\}$ is selected to satisfy $d=0.0$. For prismatic joint n , the direction of axis z is set to meet $\theta=0.0$. When $d=0.0$, the origin of the coordinate system $\{n\}$ is selected to be located at the intersection of axis X_{n-1} and joint axis z_n .

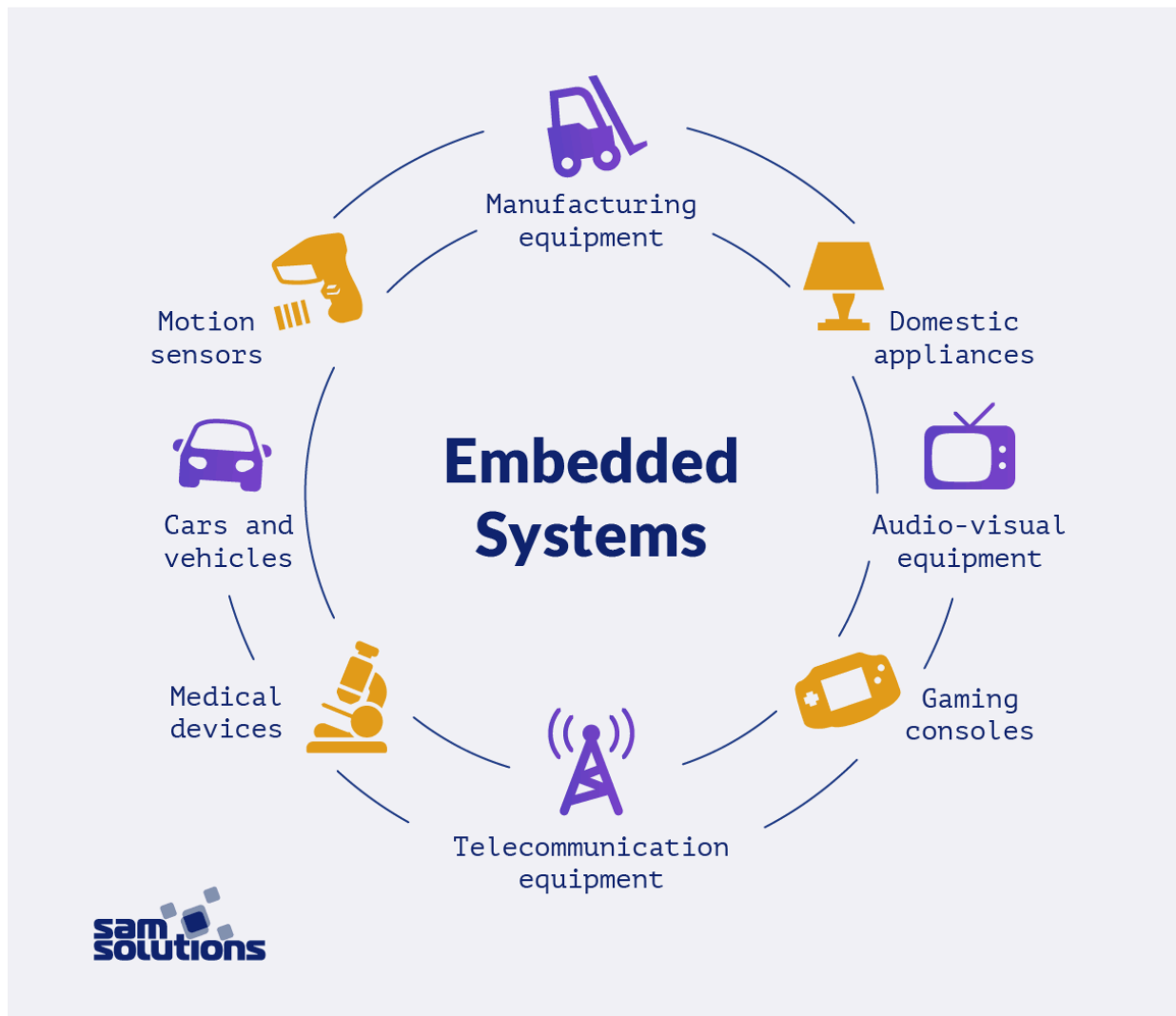
In the link coordinate system, if the link coordinate system is fixedly attached to the link as described above, the link parameters can be defined as follows:

- a_{i-1} : along x_{i-1} : move from the distance of z_{i-1} to z_i
- α_{i-1} : around x_{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
- θ_i : around z_i : rotate from the Angle of x_{i-1} to x_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

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.

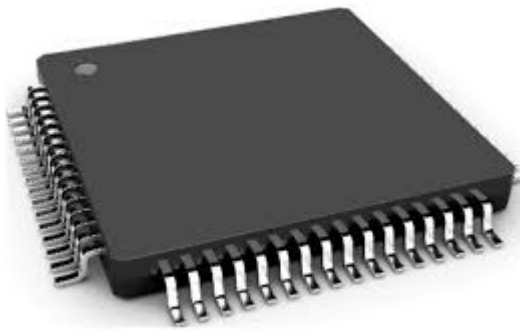
The arithmetic machine has two functions:

- 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.



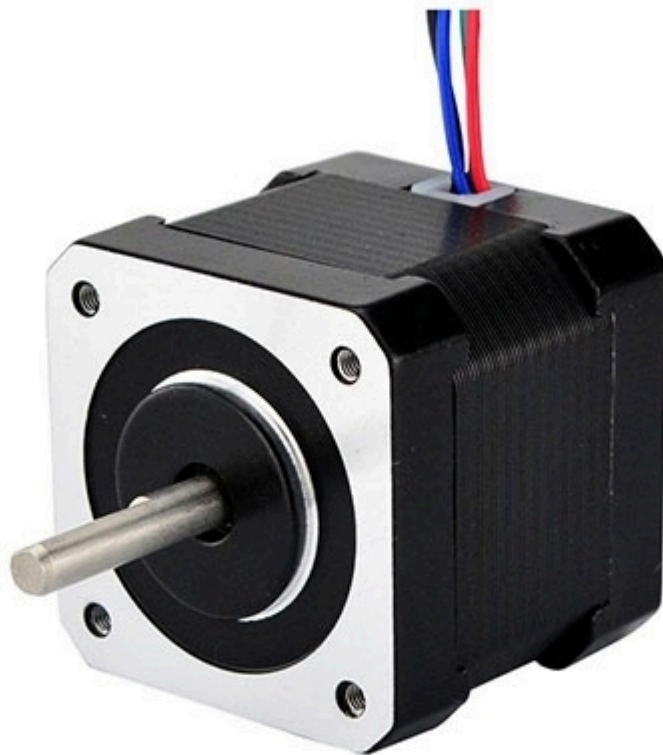
Motor & Steering Gear



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

What is Servo Motor



Electronic Basics #25: Servos and how to use them

Servo motor (servo motor) refers to the engine that controls the operation of mechanical components in the servo system, and is an auxiliary motor indirect transmission device. The servo motor can control the speed and position accuracy very accurately, and can convert the voltage signal into torque and speed to drive the control object. The rotor speed of the servo motor is controlled by the input signal and can respond quickly. In the automatic control system, it is used as an executive element, and has the characteristics of small electromechanical time constant and high linearity. It can convert the received electrical signal into the motor shaft. angular displacement or angular velocity output. It is divided into two categories: DC and AC servo motors. Its main feature is that when the signal voltage is zero, there is no rotation phenomenon, and the speed decreases uniformly with the increase of torque. 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.

How Servo Motors Work

1. The servo system is an automatic control system that enables the output controlled quantities such as the position, orientation, and state of the object to follow any changes in the input target (or given value). The servo mainly relies on pulses for positioning. Basically, it can be understood in this way that when the servo motor receives one pulse, it will rotate the angle corresponding to one pulse, thereby realizing the displacement. Because the servo motor itself has the function of sending out pulses, so every time the servo motor Rotating an angle, a corresponding number of pulses will be sent out, which echoes the pulses received by the servo motor, or is called a closed loop. In this way, the system will know how many pulses are sent to the servo motor and how many pulses are received at the same time. , in this way, the rotation of the motor can be precisely controlled, so as to achieve precise positioning, which can reach 0.001mm. DC servo motors are divided into brushed and brushless motors. Brushed motors have low cost, simple structure, large starting torque, wide speed regulation range, easy control, and require maintenance, but maintenance is inconvenient (carbon brush replacement), generates electromagnetic interference, and has environmental requirements. Therefore, it can be used in general industrial and civil occasions that are sensitive to cost. The brushless motor is small in size, light in weight, large in output, fast in response, high in speed, small in inertia, smooth in rotation and stable in torque. The control is complex, and it is easy to realize intelligentization. Its electronic commutation mode is

flexible, and it can be square wave commutation or sine wave commutation. The motor is maintenance-free, has high efficiency, low operating temperature, small electromagnetic radiation, long life, and can be used in various environments.

2. The AC servo motor is also a brushless motor, which is divided into synchronous and asynchronous motors. Synchronous motors are generally used in motion control. It has a large power range and can achieve great power. High inertia, low maximum rotational speed, and decreases rapidly as power increases. Therefore, it is suitable for applications that operate smoothly at low speeds.
3. The rotor inside the servo motor is a permanent magnet. The U/V/W three-phase electricity controlled by the driver forms an electromagnetic field. The rotor rotates under the action of this magnetic field. At the same time, the encoder that comes with the motor feeds back the signal to the driver, and the driver according to the feedback value Compared with the target value, the angle of rotation of the rotor is adjusted. The accuracy of the servo motor is determined by the accuracy (number of lines) of the encoder. The difference between AC servo motor and brushless DC servo motor in function: AC servo is better, because it is controlled by sine wave, and the torque ripple is small. DC servos are trapezoidal waves. But DC servos are simpler and cheaper.

Comparison of characteristics of servo motors

DC brushless servo motor features small moment of inertia, low starting voltage and low no-load current; abandoning the contact commutation system, greatly increasing the motor speed, the maximum speed is up to 100 000rpm; the brushless servo motor does not need an encoder when performing servo control It can also realize the control of speed, position, torque, etc.; there is no brush wear, and in addition to high speed, it also has the characteristics of long life, low noise, and no electromagnetic interference. DC brush servo motor features

4. Small size, quick action, quick response, large overload capacity, wide speed regulation range
5. Large torque at low speed, small fluctuation and stable operation
6. Low noise, high efficiency
7. The back-end encoder feedback (optional) constitutes the advantages of DC servo and so on
8. Large transformer range and adjustable frequency

The advantages of servo motors over ordinary motors

1. Accuracy: closed-loop control of position, speed and torque is realized; the problem of stepper motor out-of-step is overcome;
2. Speed: good high-speed performance, generally rated speed can reach 2000 ~ 3000 rpm;
3. Adaptability: strong anti-overload capability, able to withstand loads three times the rated torque, especially suitable for occasions with instantaneous load fluctuations and fast start requirements;
4. Stable: The low-speed operation is stable, and the stepping operation phenomenon similar to the stepping motor will not occur during low-speed operation. Applicable to occasions with high-speed response requirements;
5. Timeliness: The dynamic response time of motor acceleration and deceleration is short, generally within tens of milliseconds;
6. Comfort: heat and noise are significantly reduced. To put it simply: the ordinary motor that is usually seen will rotate for a while due to its own inertia after the power is turned off, and then stop. The servo motor and the stepper motor are to stop when they say they stop, and they can go when they say they are going, and the response is extremely fast. But the stepper motor has out-of-step phenomenon.

Application Scenarios of Servo Motors

There are too many applications for servo motors. As long as there is a power source, and there is a requirement for accuracy, it may generally involve a servo motor. Such as machine tools, printing equipment, packaging equipment, textile equipment, laser processing equipment, robots, automated production lines and other equipment that require relatively high process accuracy, processing efficiency and work reliability.

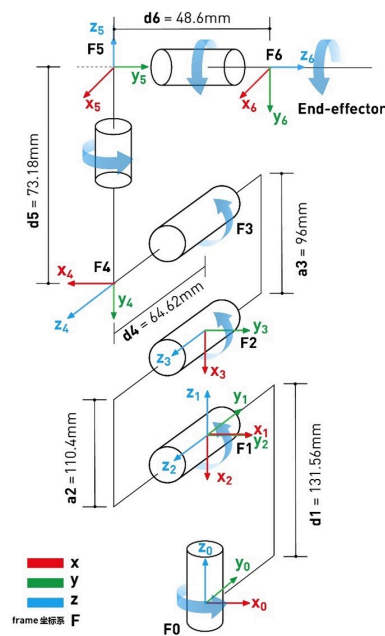
myCobot 280 algorithm

1 Structural Parameters

1.1 DH Parameters of Robotic Arm

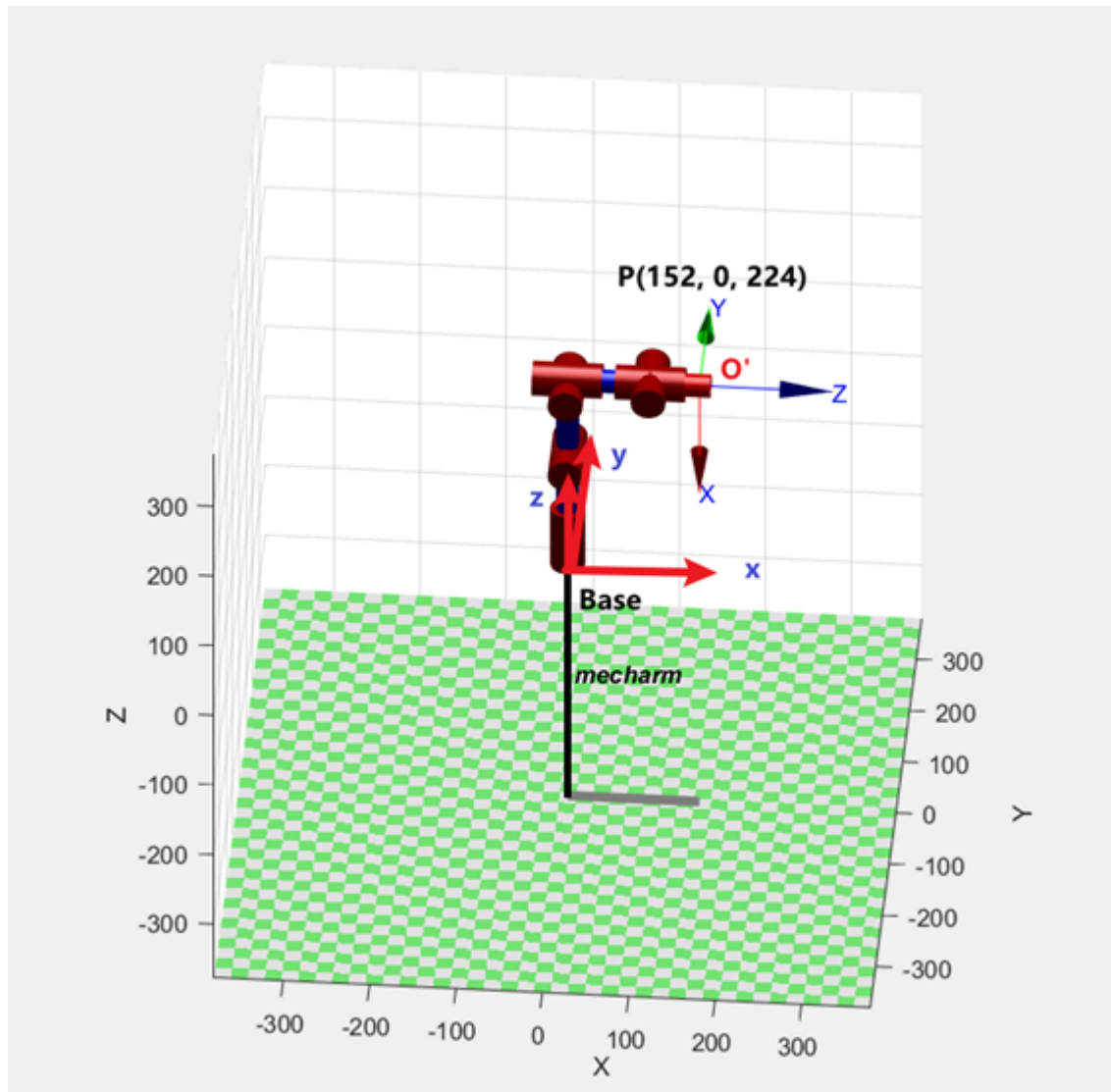
joint	theta	d	a	alpha	offset
1	q1	131.22	0	1.5708	0
2	q2	0	-110.4	0	-1.5708
3	q3	0	-96	0	0
4	q4	63.4	0	1.5708	-1.5708
5	q5	75.05	0	-1.5708	1.5708
6	q6	45.6	0	0	0

1.2 Kinematic Model



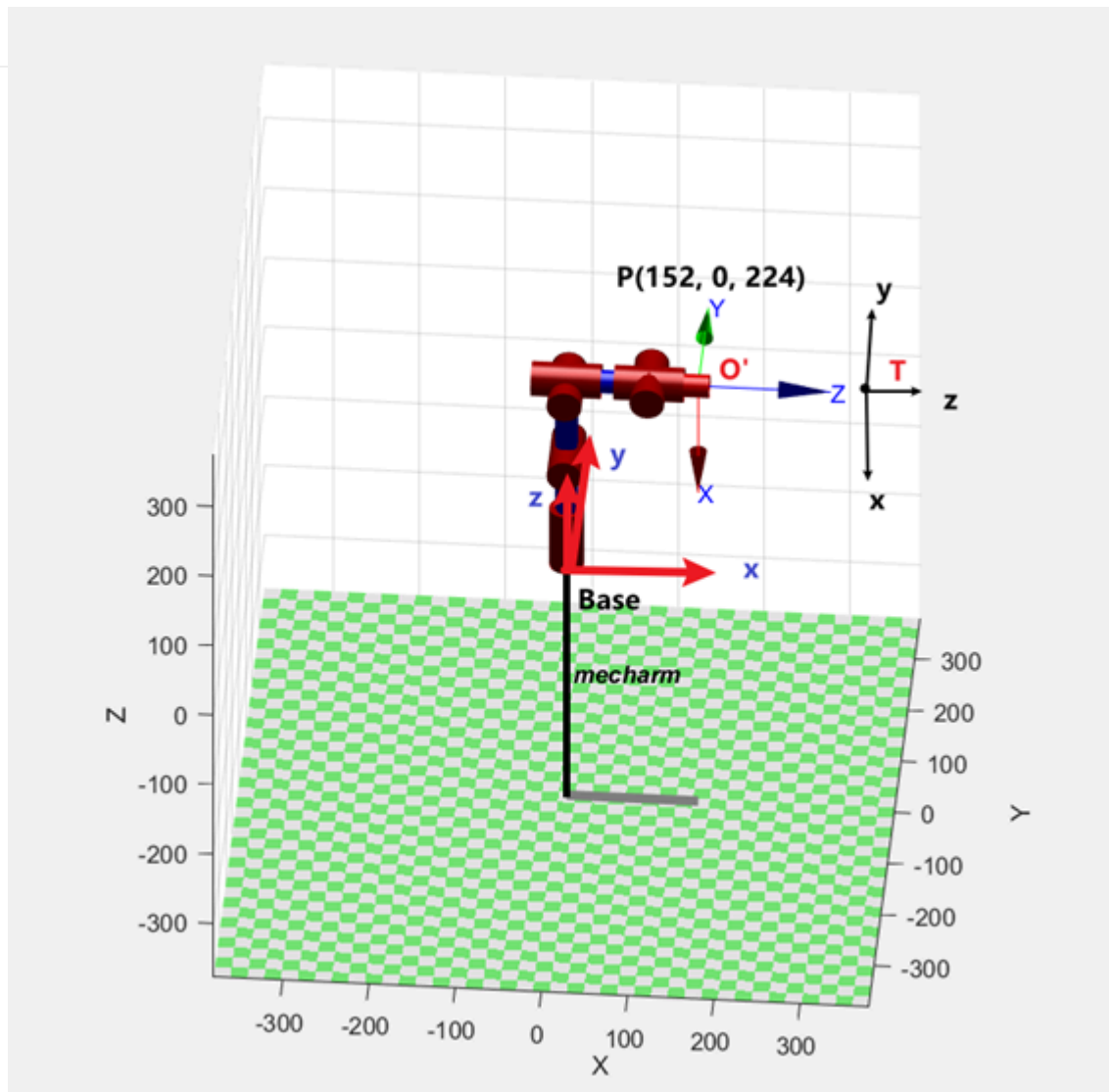
2 Coordinate System Introduction

2.1 Tool Coordinate System



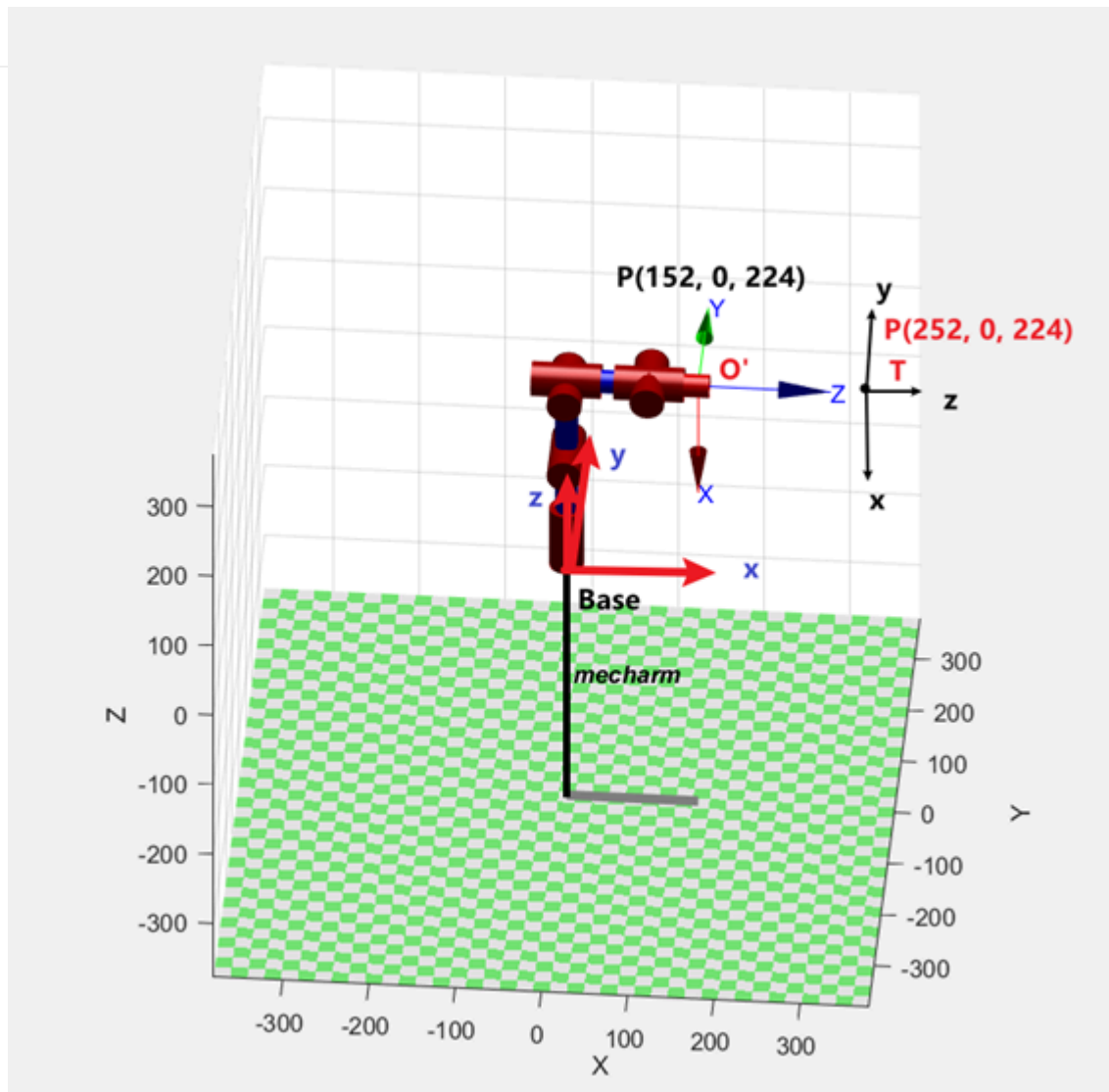
The figure shows the robot model of Mecharm270. Base in the figure represents the base coordinate system of the robot, O' represents the end flange coordinate system, and point P represents the position of the end of the manipulator relative to the base coordinate system ($x=152, y=0, z=224$)

Extend a certain pose on the basis of the end flange, and regard the set tool point as the end of the machine:



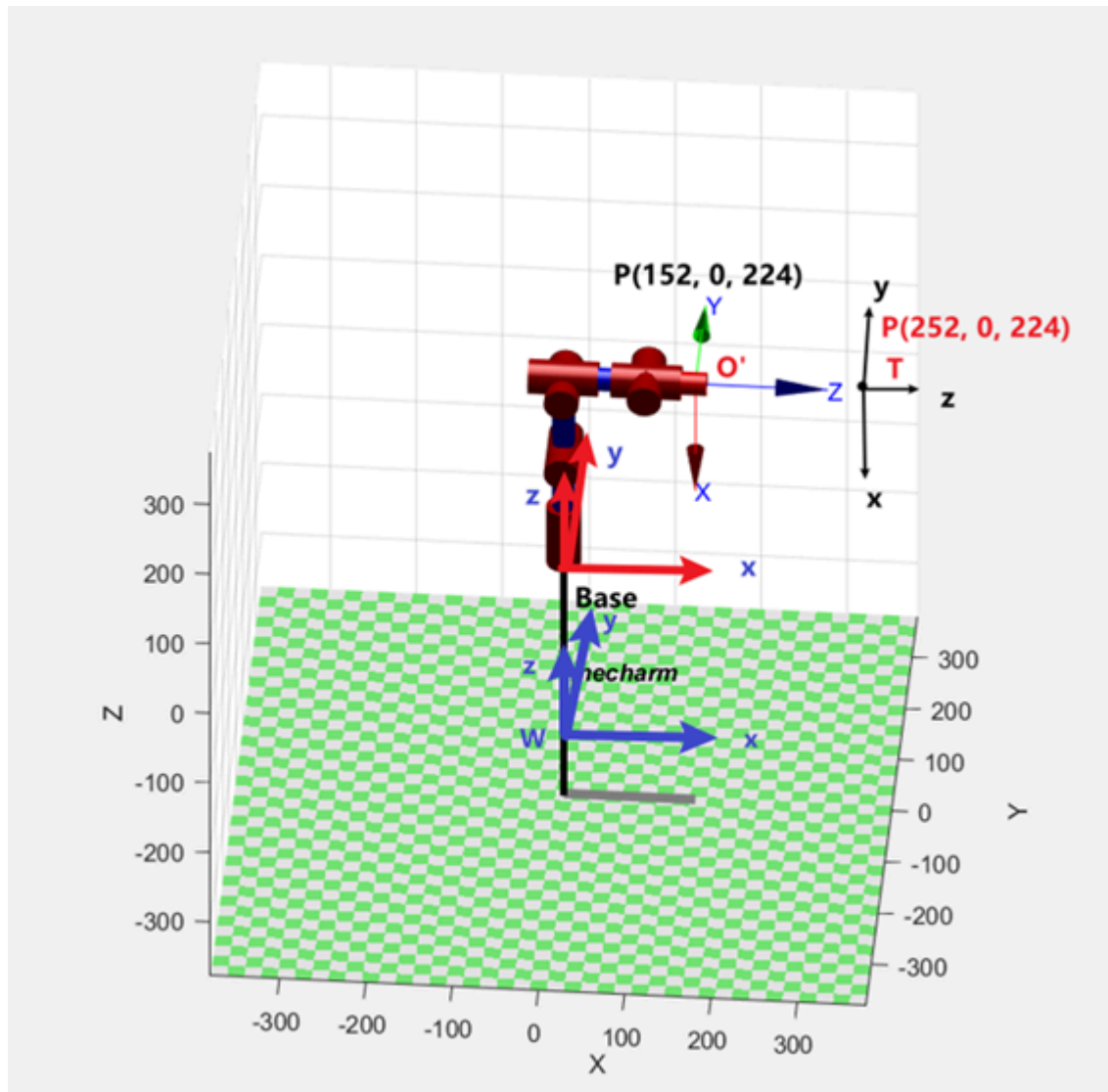
T in the figure is the set tool coordinate system. The posture of this coordinate system is consistent with O', and the relative displacement of the origin has occurred. Use the python function to set the tool coordinate system:

- `set_tool_reference([x, y, z, rx, ry, rz])` //Set tool coordinate system
- `set_end_type(1)` //Set the end coordinate system type as tool
- Assume that the tool coordinate system T is not rotated relative to O' ($rx = ry = rz = 0$)
- Assume that the origin of the tool coordinate system T is in the coordinate system O' at ($x = 0, y = 0, z = 100\text{mm}$)
- The final tool coordinate system parameter is `set_tool_reference(0, 0, 100, 0, 0, 0)`



Since the tool coordinate system is set, the end of the robot extends from O' to T at this time, and the coordinates of the end of the machine read at this time become $(152+100, 0, 224)$, and the coordinate posture movement will revolve around the tool point T rotate.

2.2 World Coordinate System



Section 2 introduces that by setting the tool coordinate system, the end coordinate system of the manipulator can be extended to a certain pose. We can also extend a certain pose on the basis of the base coordinate system of the manipulator by setting the world coordinate system. The set world coordinate system will replace the original Base coordinate system and become the new base coordinate system.

W in the figure is the set world coordinate system. The posture of this coordinate system is consistent with Base, and the relative displacement of the origin has occurred. Use the python function to set the world coordinate system:

- `set_world_reference([x, y, z, rx, ry, rz])` //Set the world coordinate system
- `set_reference_frame(1)` //Set the base coordinate system type to the world
- Assuming that the world coordinate system W has not rotated relative to Base($rx = ry = rz = 0$)
- Suppose the origin of the world coordinate system W is in the coordinate system Base ($x = 0, y = 0, z = -100\text{mm}$)
- The final world coordinate system parameter is `set_world_reference(0, 0, -100, 0, 0, 0)`

Since the world coordinate system is set, the origin of the robot extends from Base to W at this time, and the O' coordinate read at this time becomes (152, 0, 224+100).

Application of Basic Functions

1 myStudio

Before using the equipment, you have to **install the driver and update the firmware**.

1.1 Installing the driver

According to the operating system used by him, the user can click the button below to download the zip file of the corresponding **CP210X** or **CP34X** drivers. After decompressing the file, select and install the installation package corresponding to the operating system.

There are two kinds of driver chip versions: **CP210X** (suitable for CP2104 version)/**CP34X** (suitable for CH9102 version) driver zip files. If you are not sure of the USB chip used for your equipment, you can install two drivers at the same time. (During the installation of **CH9102_VCP_SER_MacOS**, an error may be reported; however, the installation has been done, and the error can be ignored.)

For Mac OS, ensure correct settings of the system "Preferred settings -> Security and privacy ->General" before installation, and allow the user to get it from App Store or an approved developer.

- Download the **Basic** serial port driver **CP210X**

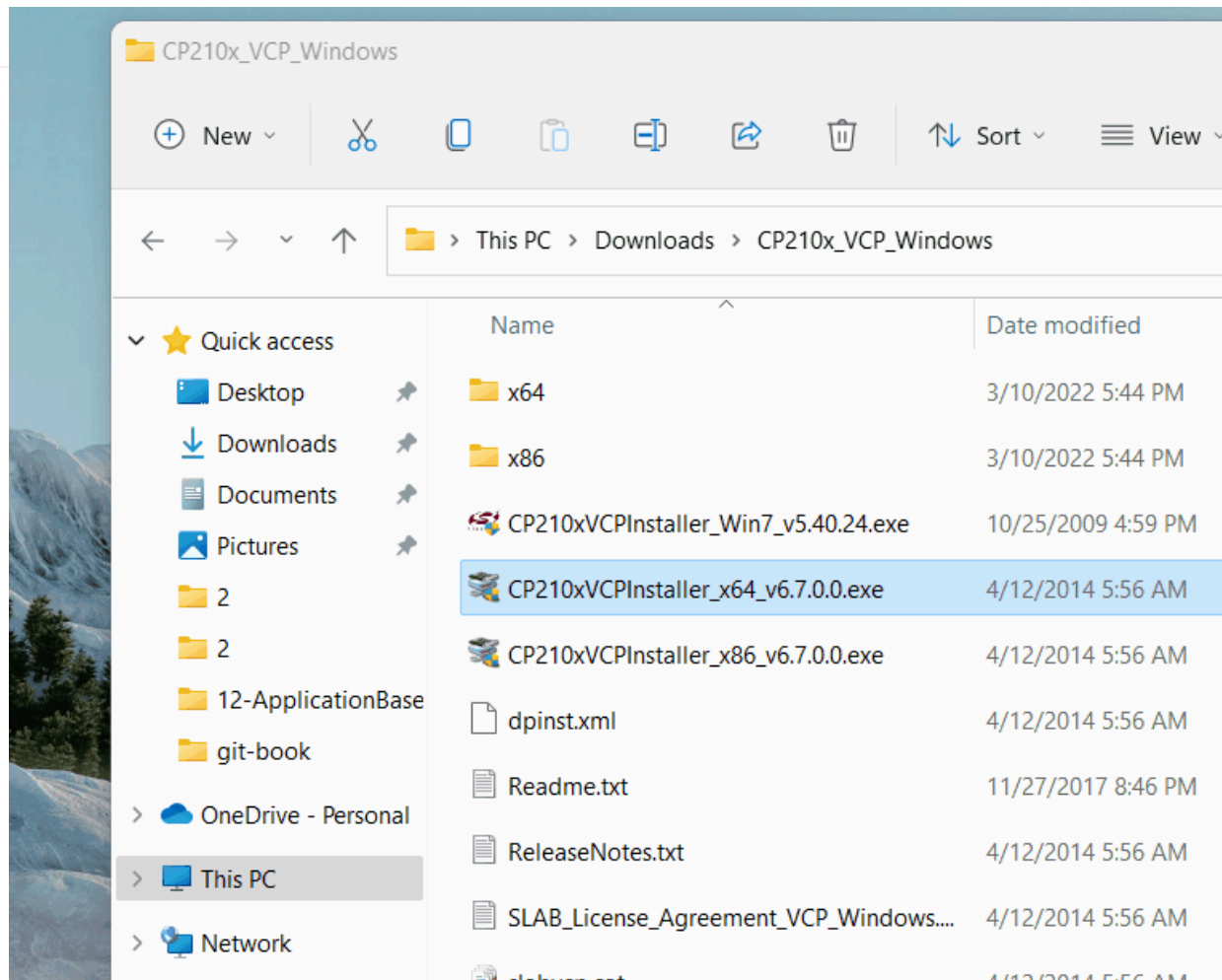
CP210X

- [Windows10](#)
- [MacOS](#)
- [Linux](#)

CP34X

- [Windows10](#)
- [MacOS](#)

- Download the **Atom** serial port driver for the end.
 - [Windows10](#)



1.2 Updating equipment firmware

Prior to development, the user is required to confirm whether his equipment firmware is the latest version so that he can use the equipment better during subsequent development.

The user can update the firmware through **myStudio**.

2 Factory firmware introduction

2.1 Drag teaching

Robot drag teaching is a process during which the operator can drag the joints of the robot directly to make them do ideal postures and then make records corresponding thereto. The cobot is a system that has this function earlier. This kind of teaching avoids various disadvantages of traditional teaching, so it is a prospective technology for robot applications.

2.2 Robot arm calibration

Calibrating the robot arm is the precondition for precise control of the robot arm, and setting joint zero and initializing the potential of the motor are basic jobs for subsequent advanced development.

2.3 Communication forwarding

Communication timeliness is vital to the micro-controller robot arm. For such arm, we often send control instructions to **M5Stack-basic** at the bottom. Through communication forwarding, the end effector analyzes the instructions and then implements target actions. At present, micro-controller devices have three methods of communication: **serial port, Bluetooth, and WIFI**. The user may choose an applicable method of communication for programming.

2.4 Connection detection

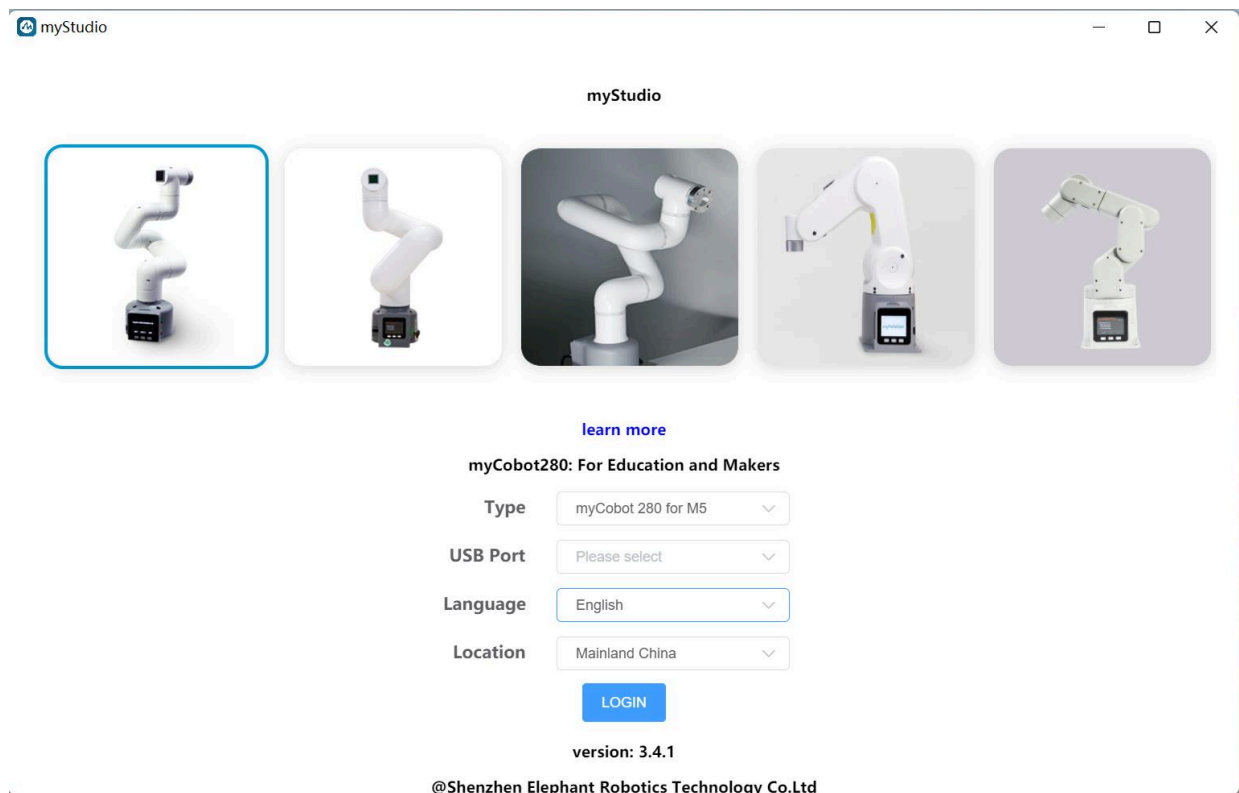
Link test is a detection function that uses the motor in the robot arm and the connection state of **Atom**. The function allows the user to remove equipment faults easily.

During the link test, the connection state of the equipment for the robot arm, including the **connection of the servo** and the **communication state of Atom** can be seen. In micro-controller devices, the versions of their current firmwares are shown on M5Stack-basic.

3 Use for the first time

After knowing the current function of the firmware, connect and fix the machine by following the steps in this chapter, and start applying the basic functions of the equipment.

myStudio



1 Reasons for Designing myStudio

- Because myStudio is a one-step application platform for a variety of robots.
- Because it is easy for users to select corresponding firmwares to meet demands and acquire related tutorial data online.

2 Latest Version and Supported Systems

- Version 3.4.1(updated on Oct 1, 2022)
- Supported Systems: Windows, Mac and Linux

3 Functions

- Burning and updating firmwares
- Providing tutorial data, such as user manuals and tutorial video
- Providing information about maintenance and repair

4 Matchable Robots

- myCobot 280
 - myCobot 280 M5

- myCobot 280 PI
 - myCobot 280 Jetson Nano
 - myCobot 280 for Arduino
- myCobot 320
 - myCobot 320 M5
 - myCobot 320 PI
- myPalletizer 260
 - myPalletizer 260 M5
 - myPalletizer 260 PI
- mechArm 270
 - mechArm 270-M5
 - mechArm 270 PI
- myCobot PRO 600
- myBuddy 280

5 Recommended Firmwares

The optimal firmwares varies with the type of robots in use. Recommended firmwares for different types of robots are listed as follows.

myCobot 280 series

myCobot 280 series have 4 versions: M5、PI、Arduino and Jetsonnano. With different core for signal connection, different firmwares are required to be burnt.

Robot Version	Core	Firmware to be Burnt	Recommended Firmware
M5	M5Stack-Basic	miniRobot	v2.1 is recommended for dragging teaching, wifi connection and bluetooth connection
	Atom	atomMain	v4.1 is recommended for robots labelled ER28001202200415 and before, or not labelled; v5.1 is recommended for robots labelled ER28001202200416 and after
PI	RaspberryPI 4B	ubuntu	v18.04.is recommended
	Atom	atomMain	v4.1 is recommended for robots labelled ER28001202200415 and before, or not labelled; v5.1 is recommended for robots labelled ER28001202200416 and after
Arduino	mega2560	transponder	v1.0 is recommended
	mkrwifi1010	transponder	v1.0 is recommended
	Atom	atomMain	v4.1 is recommended for robots labelled ER28001202200415 and before, or not labelled; v5.1 is recommended for robots labelled ER28001202200416 and after
Jetson nano	JestonNano	ubuntu	v18.04.is recommended
	Atom	atomMain	v4.1 is recommended for robots labelled ER28001202200415 and before, or not labelled; v5.1 is recommended for robots labelled ER28001202200416 and after

myCobot 320 (2022) series

myCobot 320 (2022) series have two versions: M5 and PI. With different core for signal connection, different firmwares are required to be burnt.

Robot Version	Core	Firmware to be Burnt	Recommended Firmware
M5	M5Stack-basic	miniRobot	v2.0 is recommended
	Atom	atomMain	v5.0 is recommended
	Pico	picoMain	v1.0 is recommended
PI	RaspberryPI 4B	ubuntu	v18.04. is recommended
	Atom	atomMain	v5.0 is recommended
	Pico	picoMain	v1.0 is recommended

****myCobot 320 (2020) series ****

Robot Version	Core	Firmware to be Burnt	Recommended Firmware
M5	M5Stack-basic	miniRobot	v1.0 is recommended
	Atom	atomMain	v4.2 is recommended
PI	RaspberryPI 4B	ubuntu	v18.04. is recommended
	Atom	atomMain	v4.2 is recommended

myPalletizer 260 series

myPalletizer 260 series have two versions: M5 and PI. With different core for signal connection, different firmwares are required to be burnt.

Robot Version	Core	Firmware to be Burnt	Recommended Firmware
M5	M5Stack-Basic	miniRobot	v1.1 is recommended
PI	Atom	atomMain	v1.1 is recommended

1 Environment Building

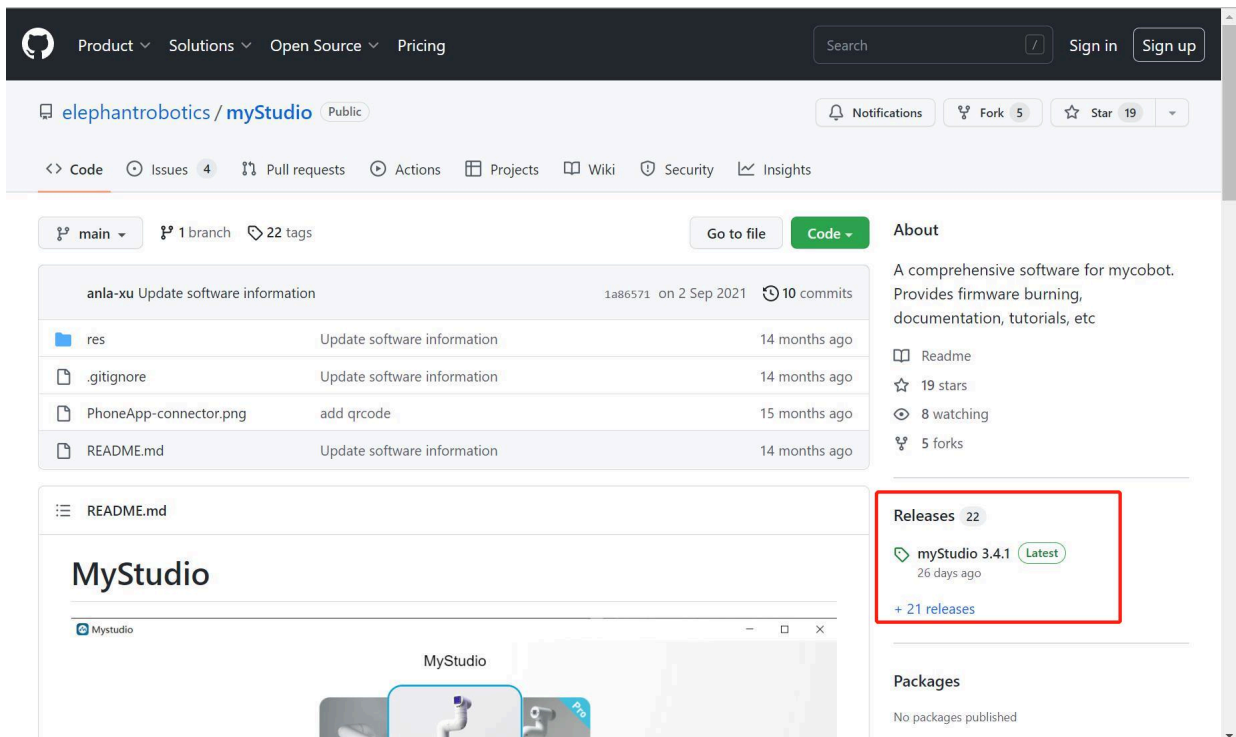
1.1 Download and Installation

Note: The installation path of myStudio installation cannot have any spaces

address for downloading:

1. GitHub

- Click on `myStudio` at the right side and download the version corresponding to your PC. **Do not install myStudio into files with space directory.**



The screenshot displays the GitHub repository for `elephantrobotics/myStudio`. The repository is public and has 19 stars and 5 forks. The main branch is `main` with 22 tags. The repository contains several files and folders, including `res`, `.gitignore`, `PhoneApp-connector.png`, and `README.md`. The `README.md` file is open, showing the MyStudio logo and a screenshot of the software interface. On the right side, the 'Releases' section is highlighted with a red box, showing the latest release 'myStudio 3.4.1' from 26 days ago, with a link to '+ 21 releases'.

- Different suffixes signifies versions applicable for different systems. -
 - `*.tra.xz` is applicable to Linux
 - `*.dmg` is applicable to Mac
 - `*.exe` is applicable to Window

▼ Assets

4

 [myStudio-3.4.1-arm64.ApplImage](#) [myStudio.Setup.3.4.1.exe](#) [Source code \(zip\)](#) [Source code \(tar.gz\)](#)

2. Elephant Robotics Official Website

Software Downloads



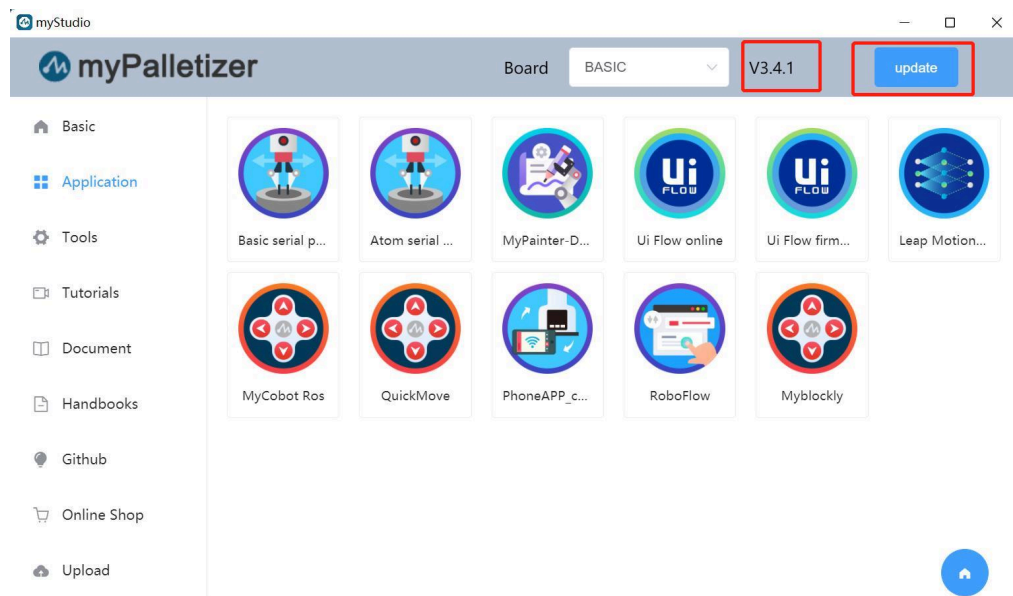
myStudio

myStudio is a one-stop platform for robots of myRobot/myCobot. The main functions of myStudio are: 1) Update the firmware; 2) Provide video tutorials on how to use the robot; 3) Provide maintenance and repair information (such as video tutorials, Q&A, etc.).

咨询 & 求助

Windows Linux Mac 

Notice: Download the latest version. You can log in to check the present version of myStudio and update to the latest.



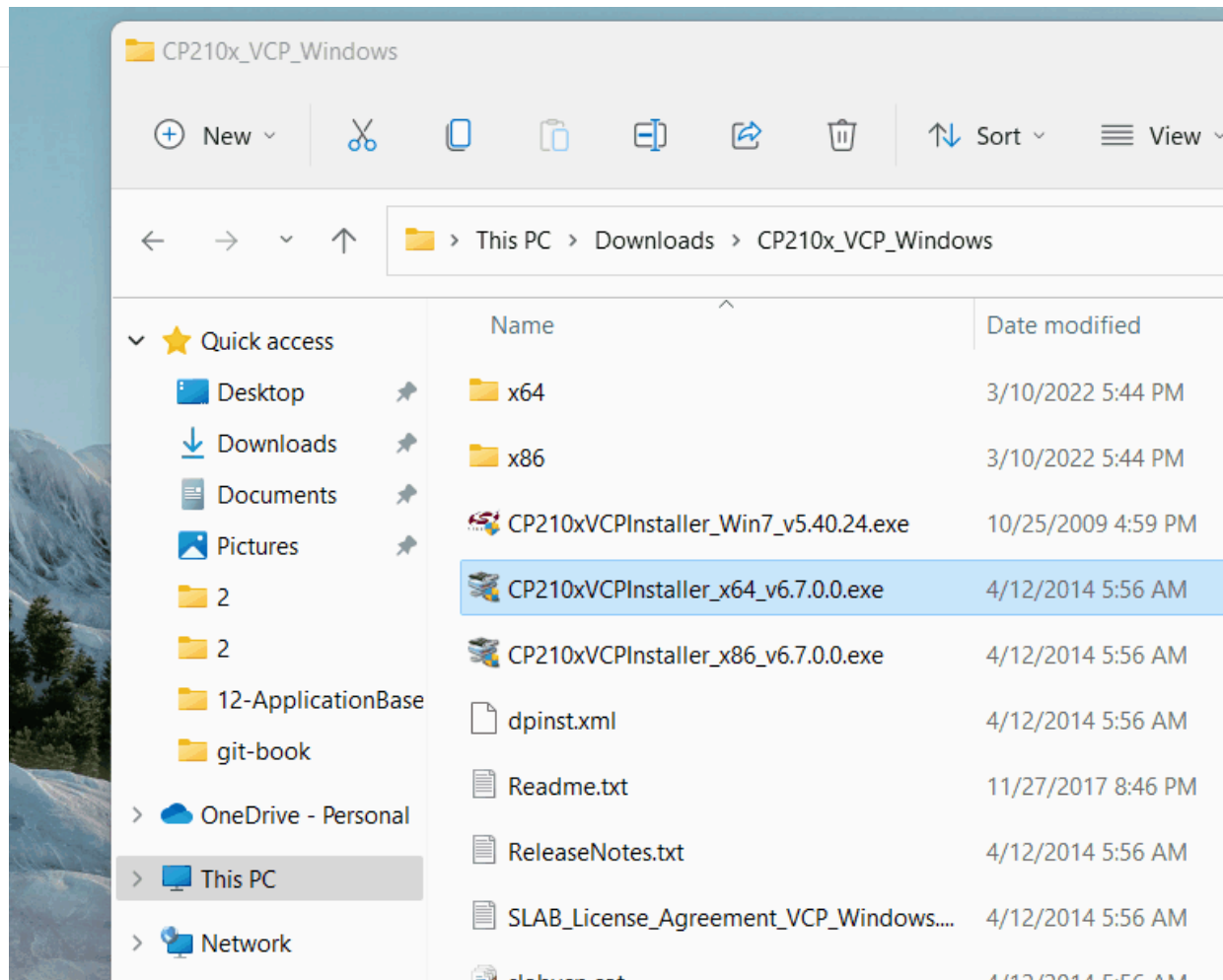
1.2 Serial Port Driver Installation

Download corresponding serial port driver according to the USB chip on your PC. CP210X is suitable for CP2104 version and CP34X is suitable for CH9102 version. You can install both of them if you are unable to confirm the type of USB chip. Go to the address below to download and install serial port driver.

- serial port driver for M5Stack Basic:
 - CP210X: [Windows 10](#), [MacOS](#), [Linux](#)
 - CP34X: [Windows 10](#), [MacOS](#)
- serial port driver for Atom:
 - [Windows 10](#)

Notice:

- An error may be reported during installation of CH9102_VCP_SER_MacOS. However, the installation already completes. Just ignore the error.
- Make sure that settings of the system \"Preferred settings -> Security and privacy -> General\" is correct and allow the user to get it from App Store or an approved developer.



1.3 How to distinguish between CP210X chip and CP34X chip

- As shown in the GIF below, go to `equipment manager` and check `ports (COM and LPT)` .



- If the ports (COM and LPT) show USB-Enhanced-SERIAL CH9102 , the chip is CP34X.



- If the ports (COM and LPT) show Silicon Labs CP210x USB to UART Bridge , the chip is CP210X.

- ▼  PROTEGO-7S2B7J5
 - >  USB 连接器管理器
 - >  安全设备
 - >  处理器
 - >  传感器
 - >  磁盘驱动器
 - >  存储控制器
 - >  打印队列
 - >  打印机
 - >  电池
 - ▼  端口 (COM 和 LPT)
 -  Silicon Labs CP210x USB to UART Bridge (COM6)
 - >  调制解调器
 - >  计算机
 - >  监视器
 - >  键盘
 - >  蓝牙
 - >  人机接口设备
 - >  软件设备
 - >  软件组件
 - >  生物识别设备
 - >  声音、视频和游戏控制器
 - >  鼠标和其他指针设备
 - >  通用串行总线控制器
 - >  网络适配器
 - >  系统设备
 - >  显示适配器
 - >  音频输入和输出
 - >  照相机

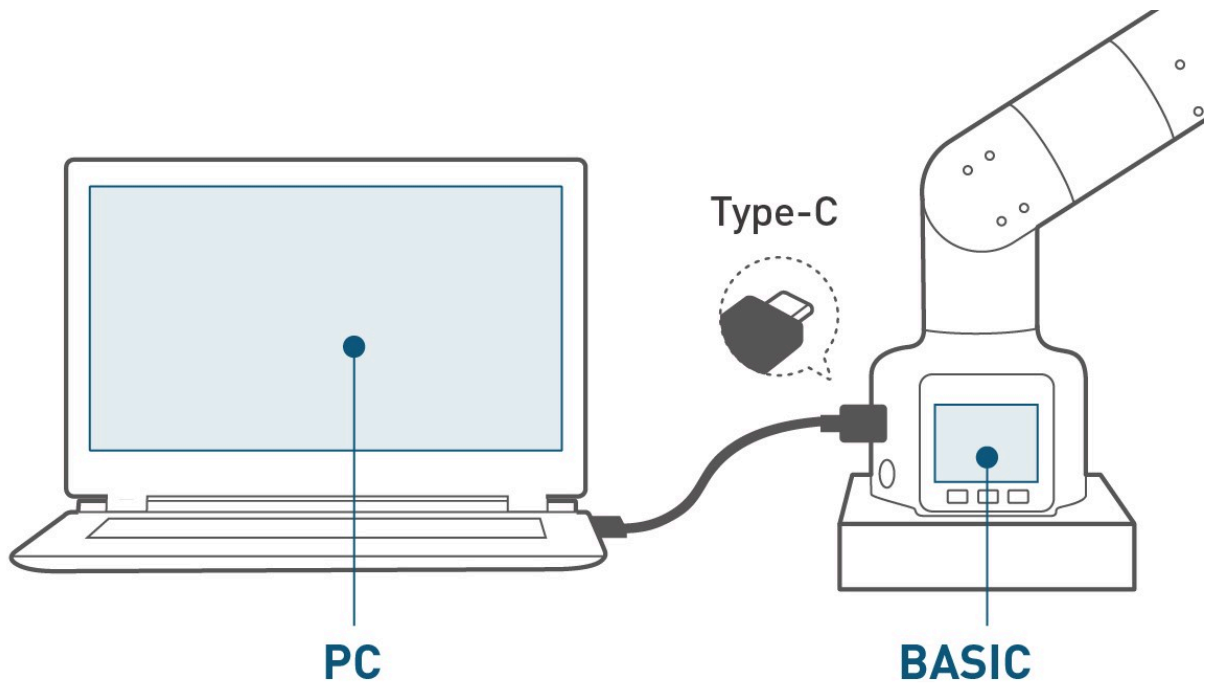
2 Burning and Updating Firmwares

Tutorial Video: <https://www.bilibili.com/video/BV1Qr4y1N7B5/>

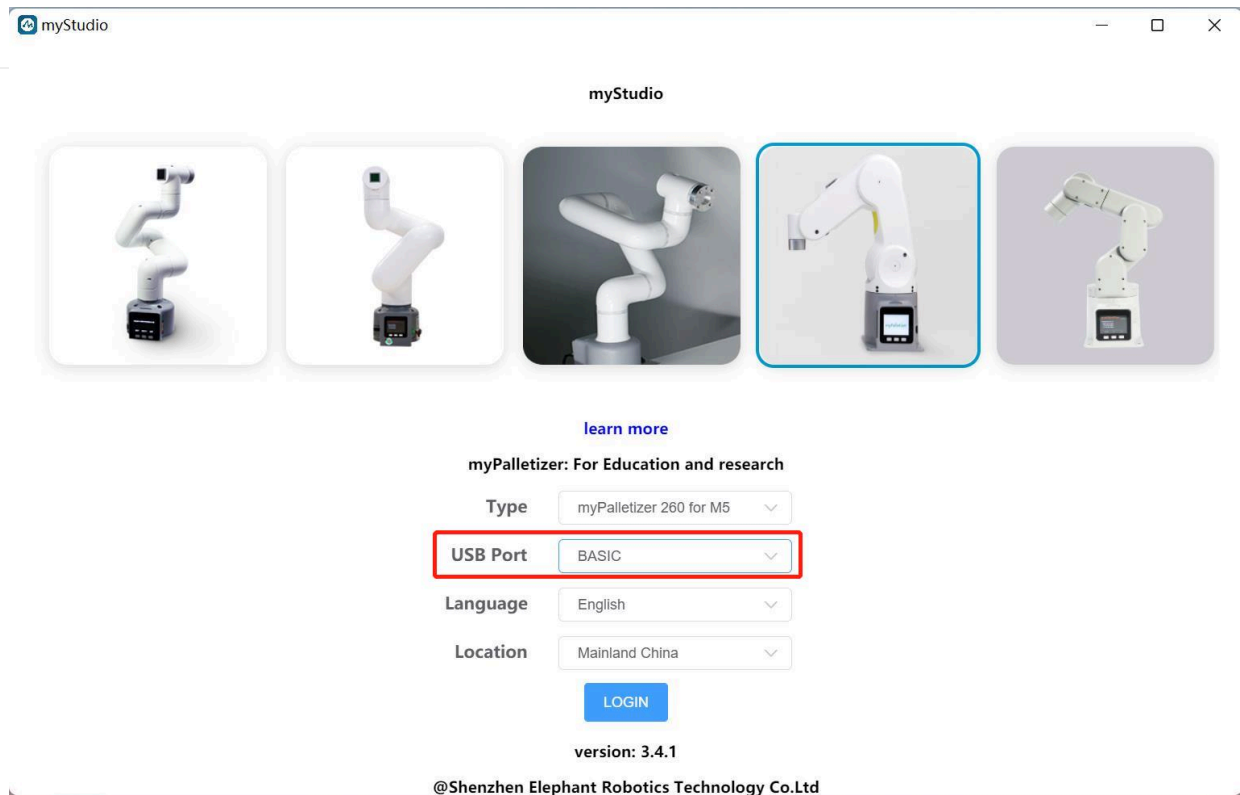
2.1 Burning M5Stack-basic Firmwares

Notice: PI version is not required to burn M5Stack-basic firmwares

Step 1: Connect M5Stack-basic with PC with USB.

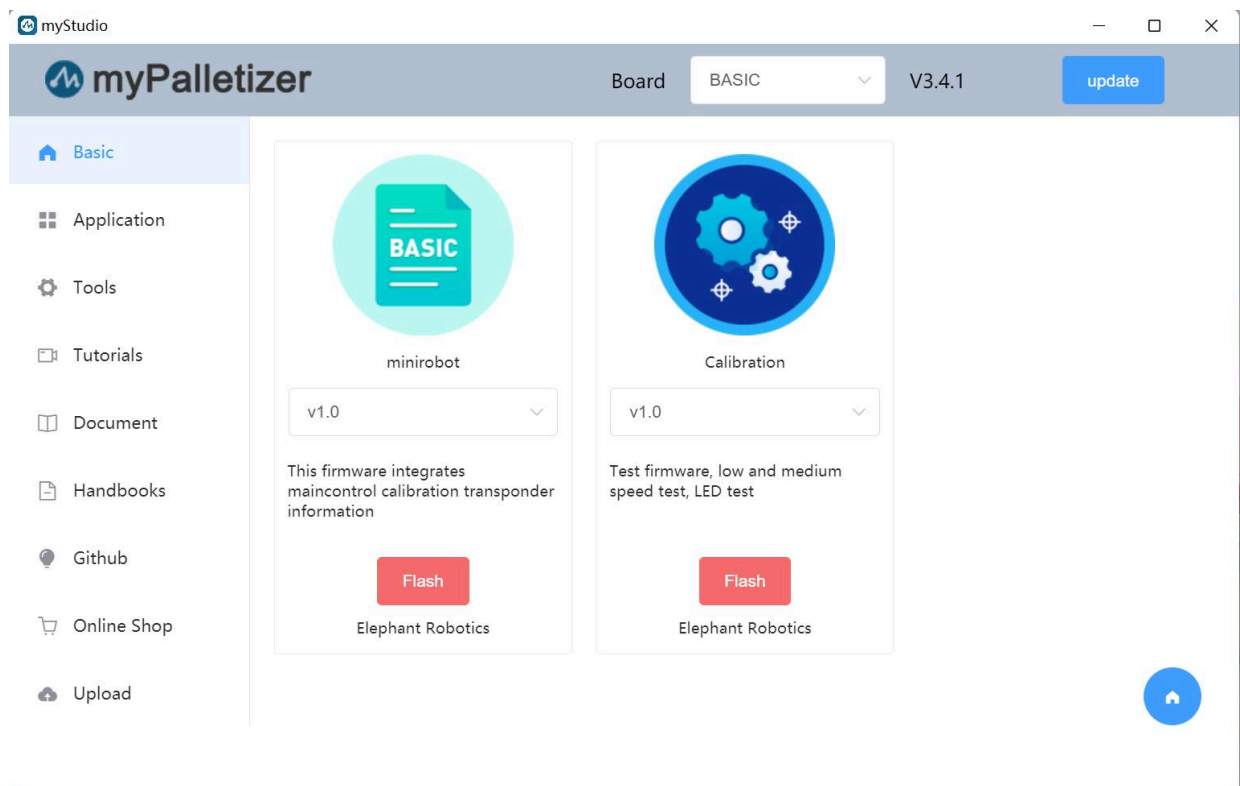


Step 2: Select `USB Port` . After connecting with PC, `USB Port` shows `M5Stack-basic` . The figure below takes myPalletizer 260 M5 as an example.



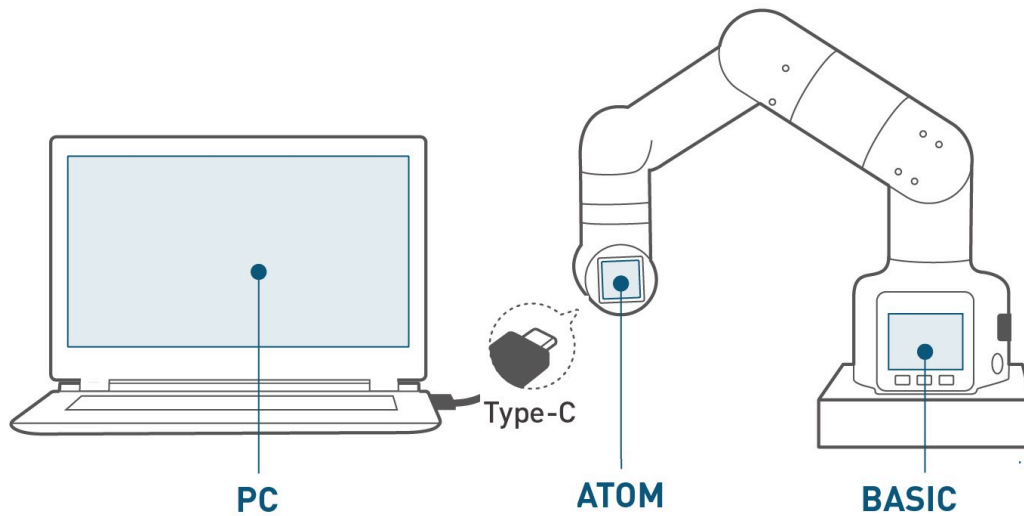
- Click on `LOGIN` -> `M5Stack-basic`, and then you can burn firmwares.

Notice: As 280 PI/Jetson nano/Arduino versions do not have M5Stack-basic, the `USB Port` cannot show anything after connection.

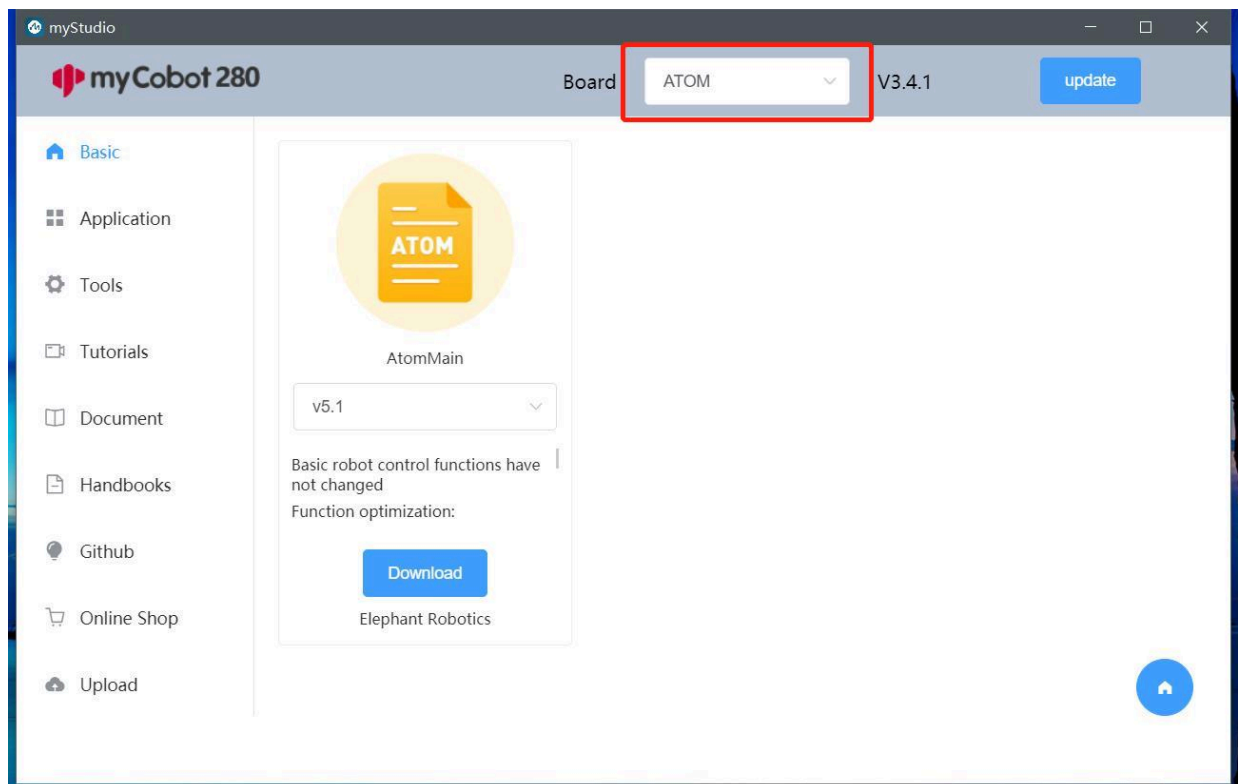


2.2 Burning Atom Firmware

Step 1: Connect Atom with PC with USB.



Step 2: Select `ATOM` at Board, and then burn AtomMain. The picture below takes myCobot 320 as an example.

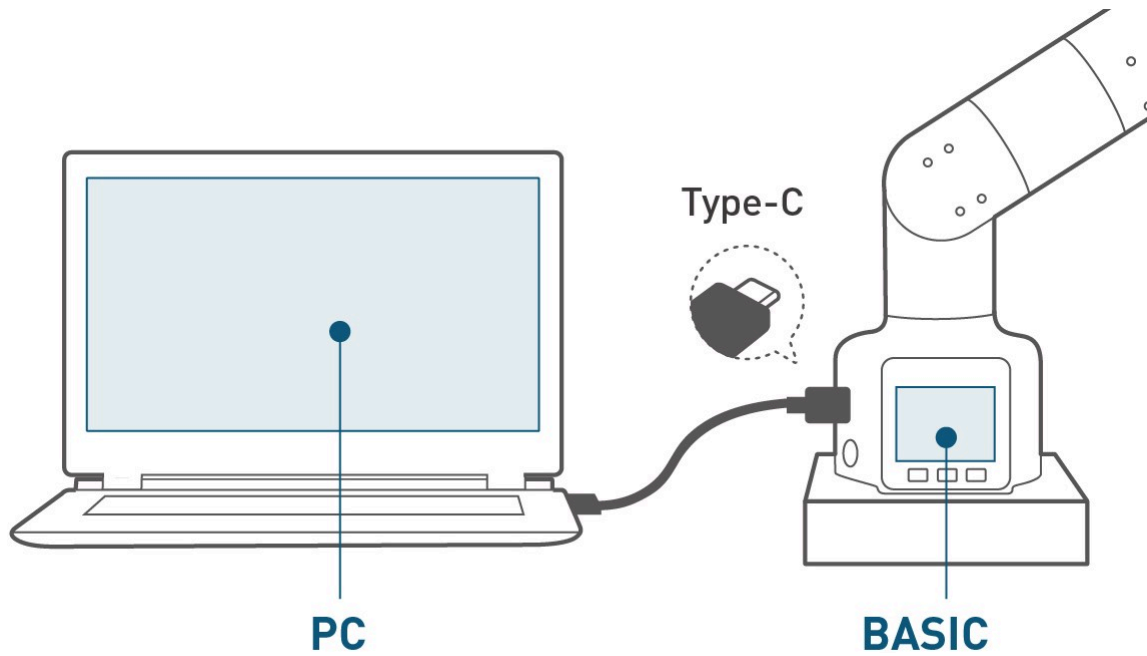


2.3 Firmware Burning for myCobot 320 series

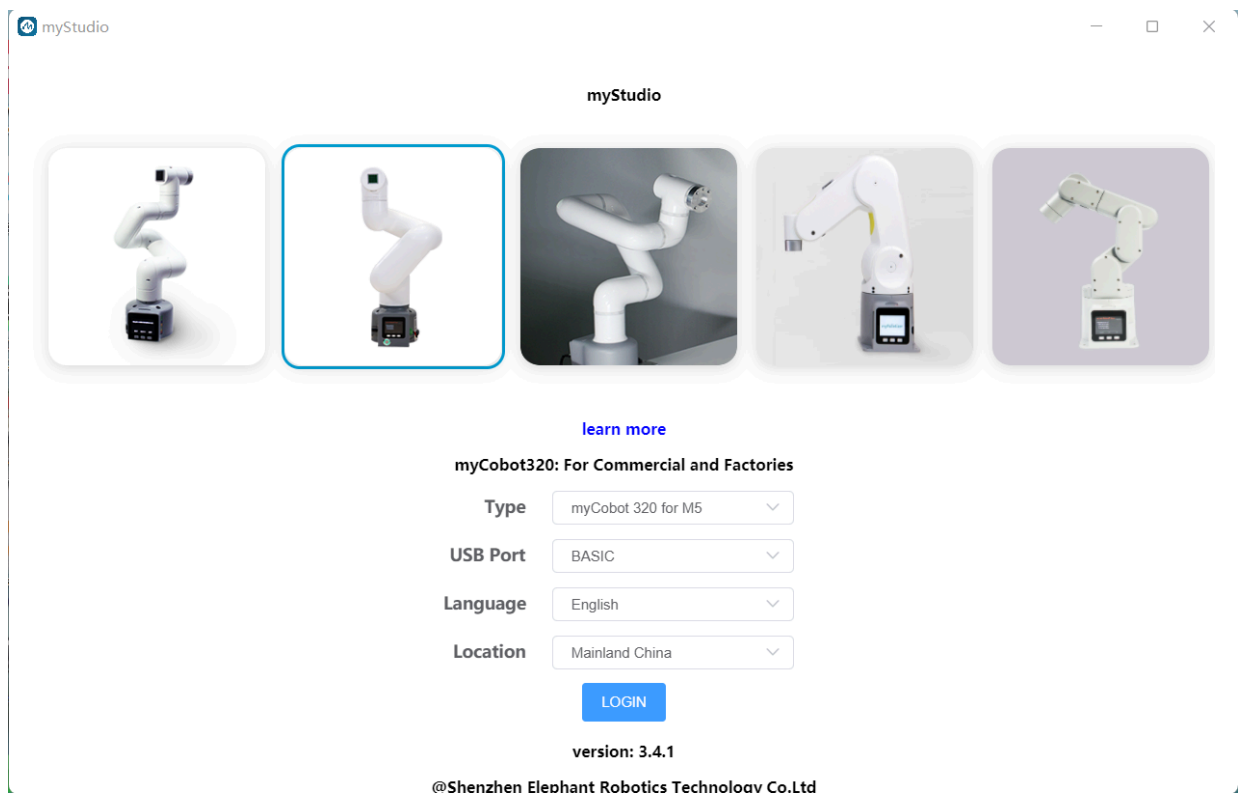
M5Stack-basic and picoMain are required to be burnt on myCobot 320 series. It should be noted that the two firmwares are burnt by different type-c interfaces.

1.Burning M5Stack-basic: miniRobot

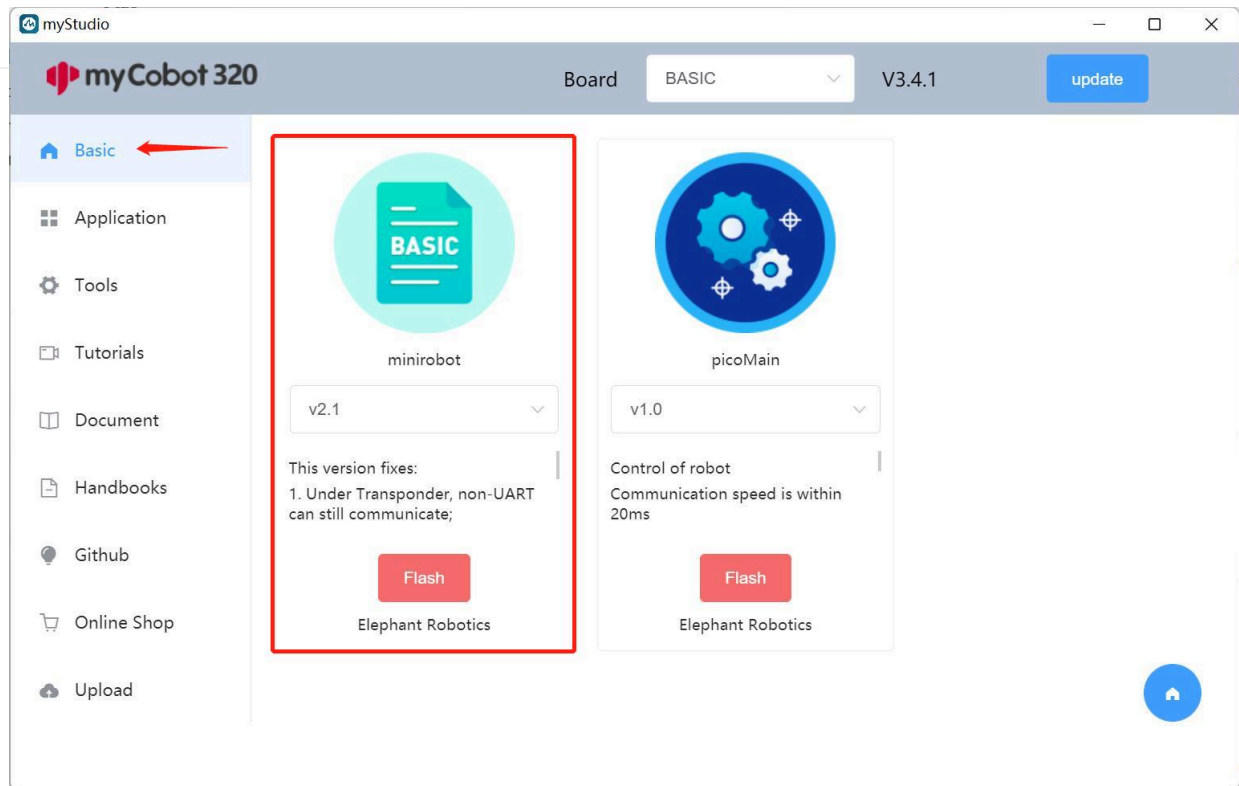
Step 1: Connect M5Stack-basic and PC.



Step 2: Select the 320 series, language and region, and then click on **LOGIN**.



Step 3: Click on **Basic**, and select minirobot to burn.



2. Burning Pico Firmware: picoMain

Step 1: Connect M5Stack-basic with PC.

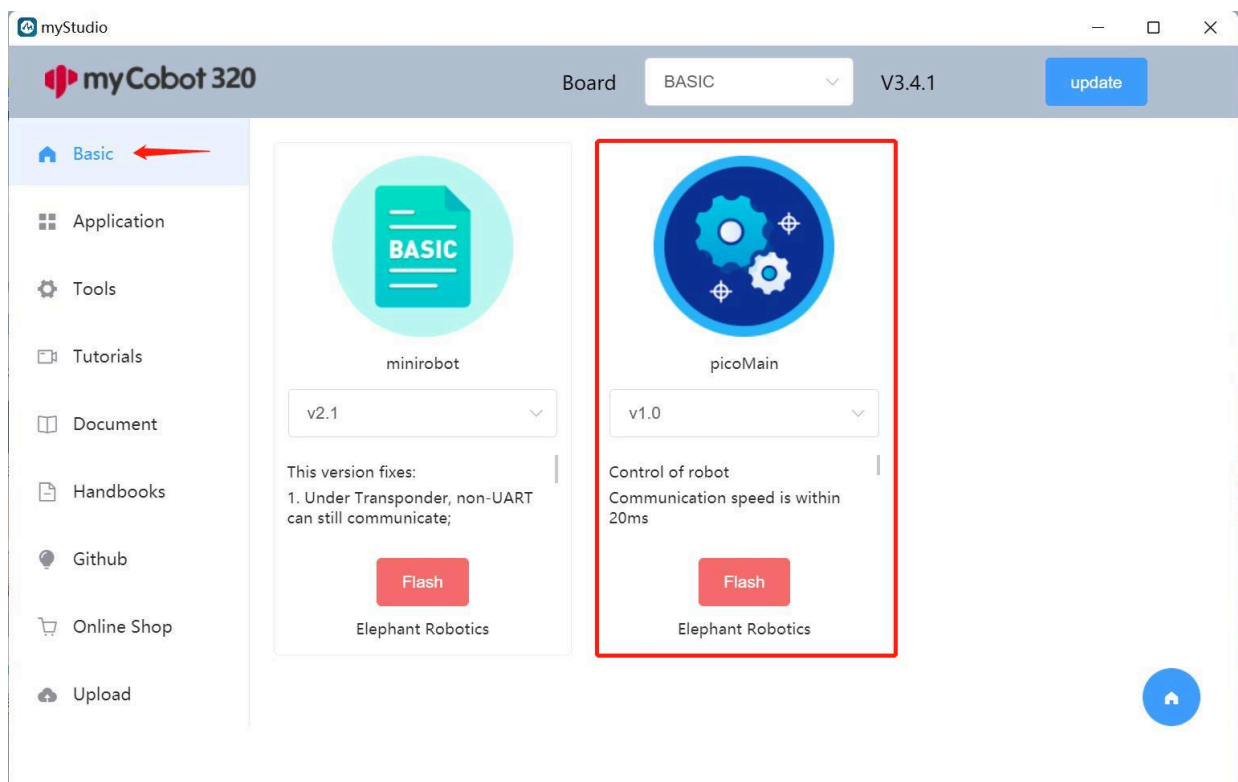


Step 2: Select **Flash** (press **UP** once or **DOWN** four times), and then the screen goes dark for approximately 30 seconds.



Step 3: select picoMain to burn via myStudio.

Notice: picoMain is required to be burnt during the sleep period of **Flash** mode. Otherwise it may fail to burn picoMain.



2.4 myAGV Firmware burning

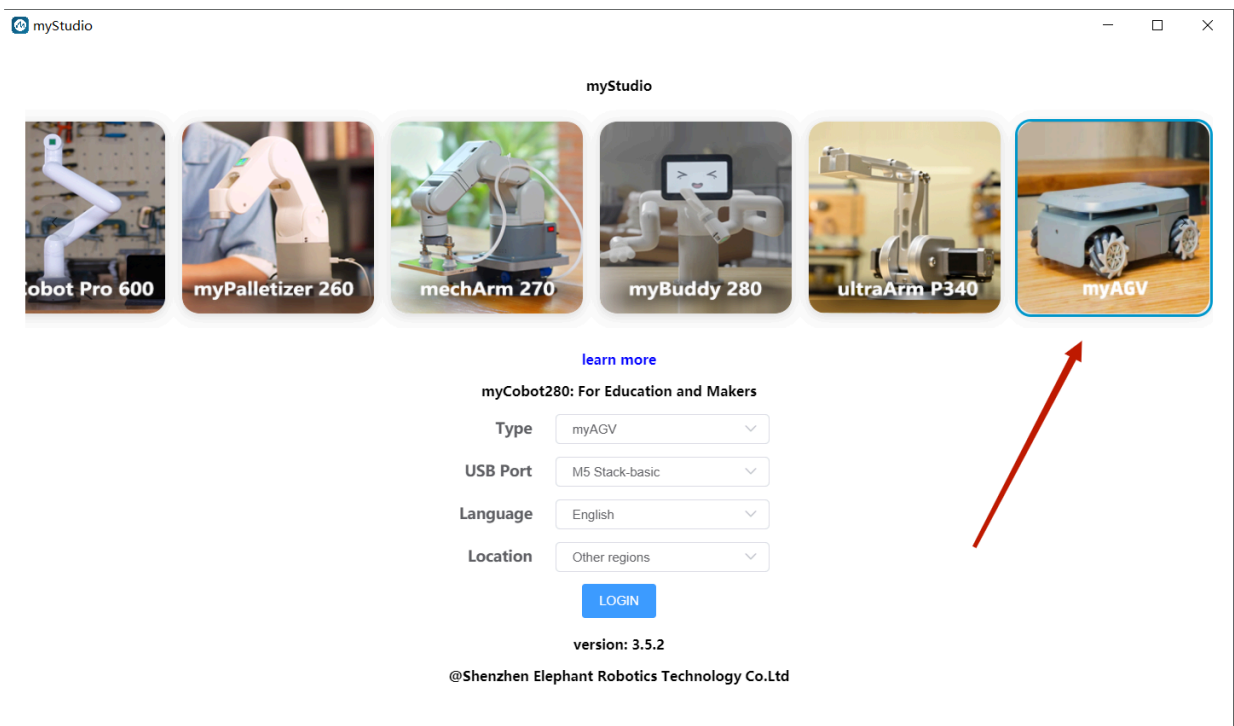
1. Burning Pico Firmware: picoMain

Step 1: Turn on the computer and connect the screen.

1 Adaptive robot

For detailed tutorials, please refer to: [2.5.1 First-time Use · GitBook \(elephantrobotics.com\)](#)

Step 2: Select the myAGV image, select the model language and region, and click to log in.



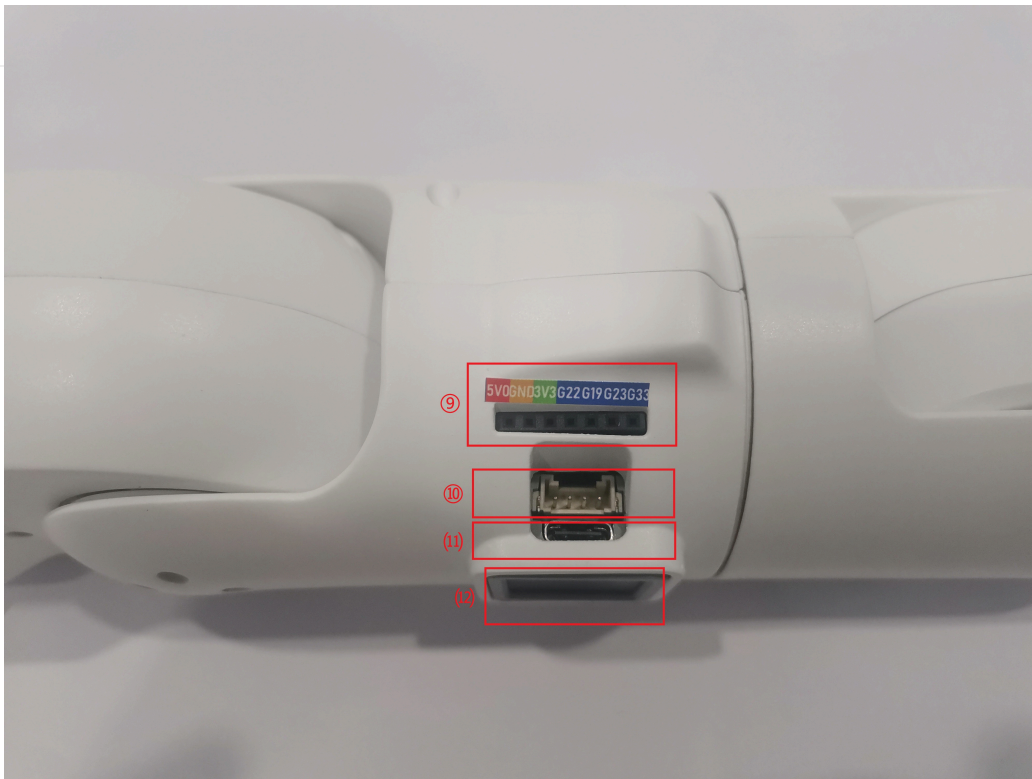
Step 3: After logging in, click 'Basic' and select picoMain to burn.



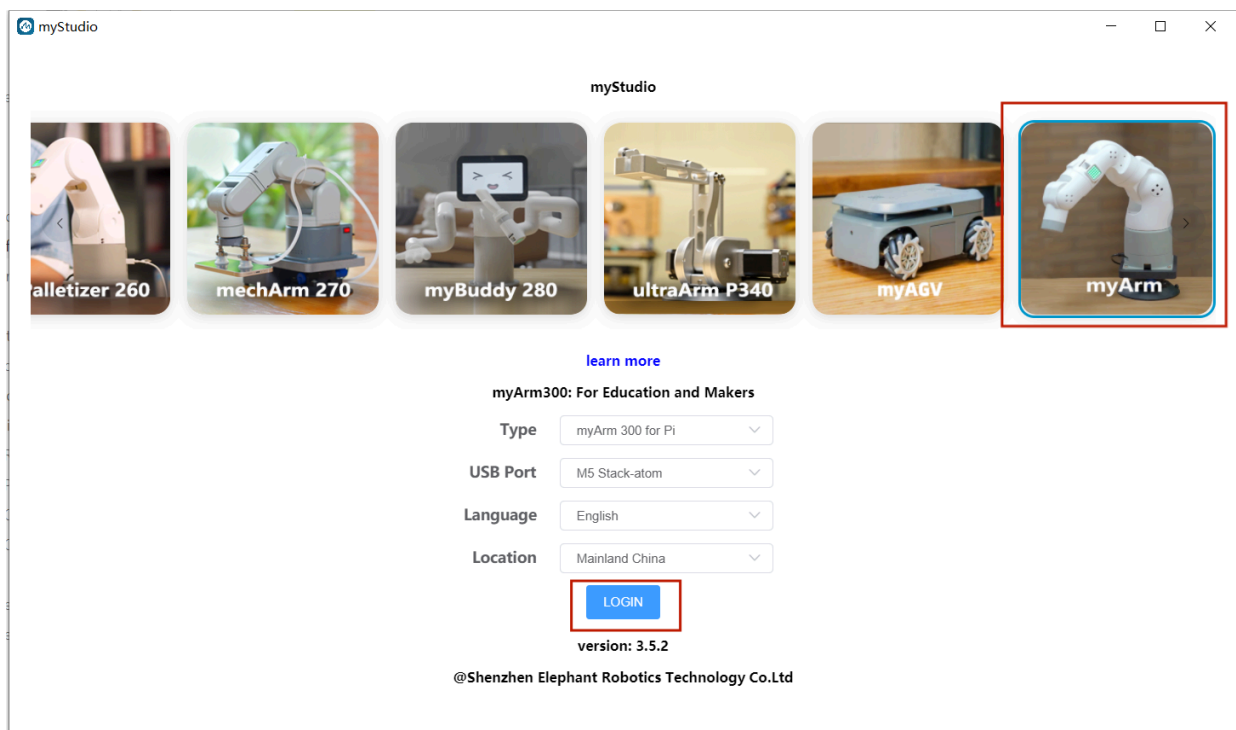
2.5 myArm Firmware burning

1. Burning Atom Firmware

Step 1: Connects to a PC. Use USB to connect the Atom interface (labeled 11 as Atom Type C interface).



Step 2: Select the myArm image, select the model language and region, and click to login.



Step 3: After login, click 'Basic', select AtomMain, click download, and then click the burn button to burn.

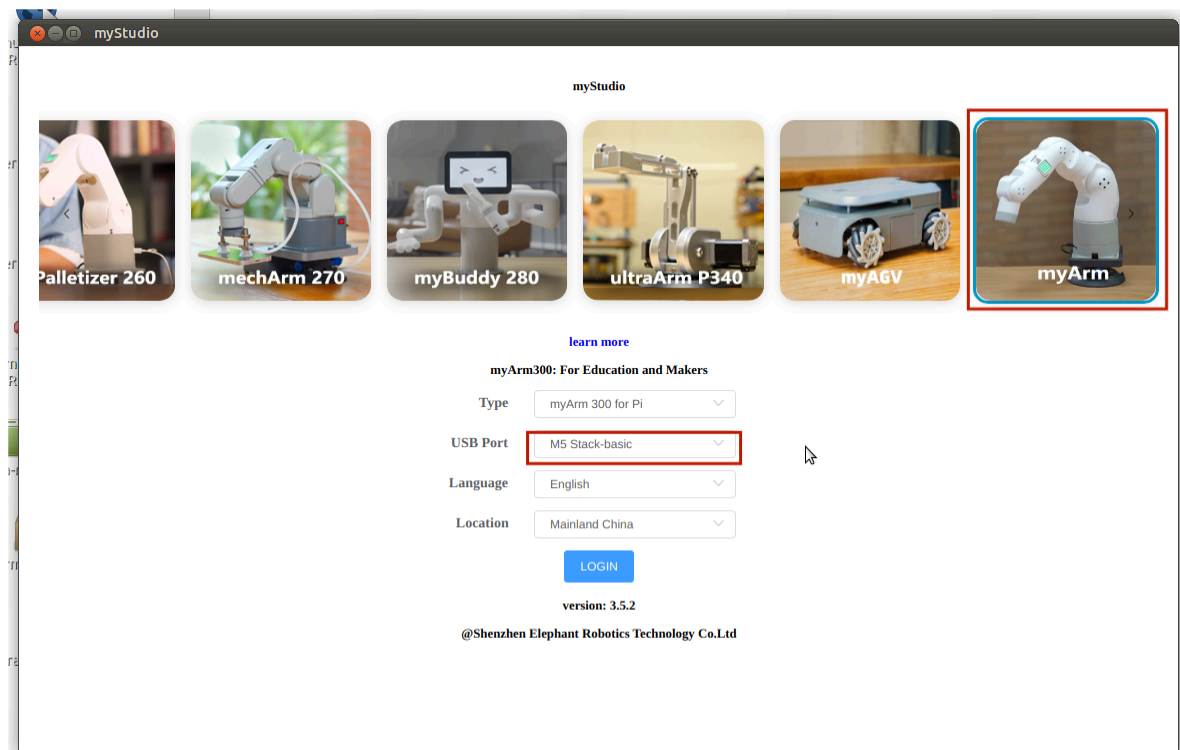


2. Burning Pico Firmware: picoMain

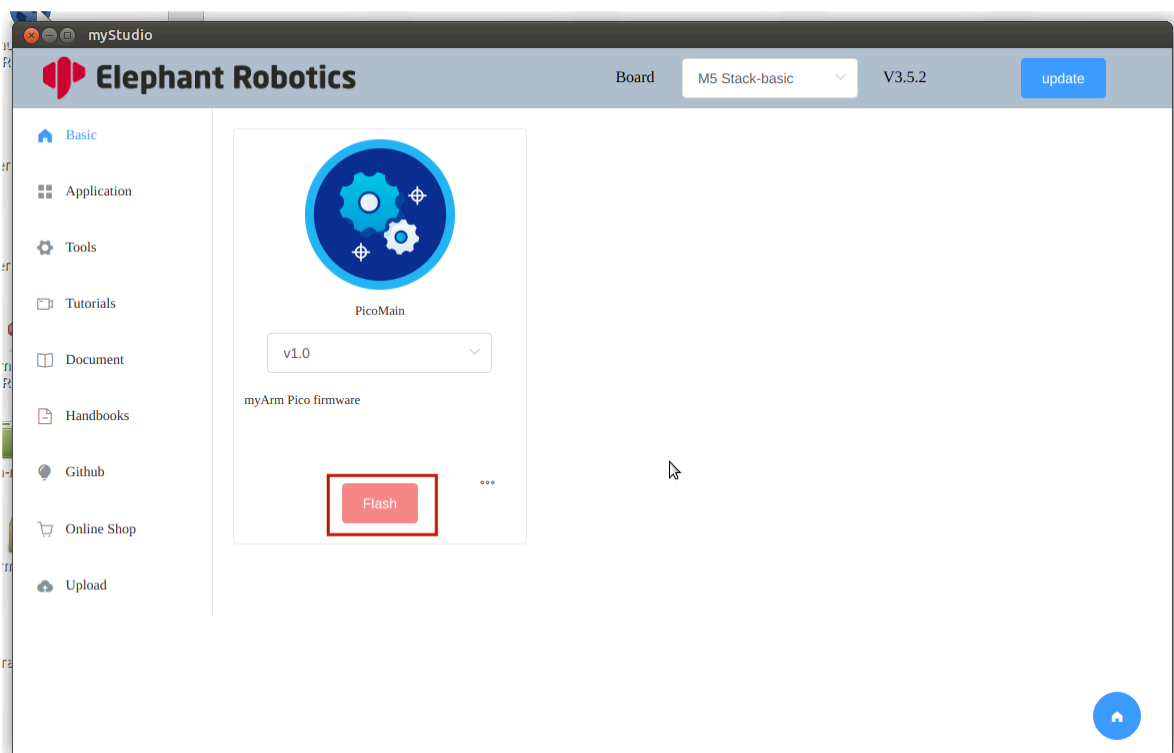
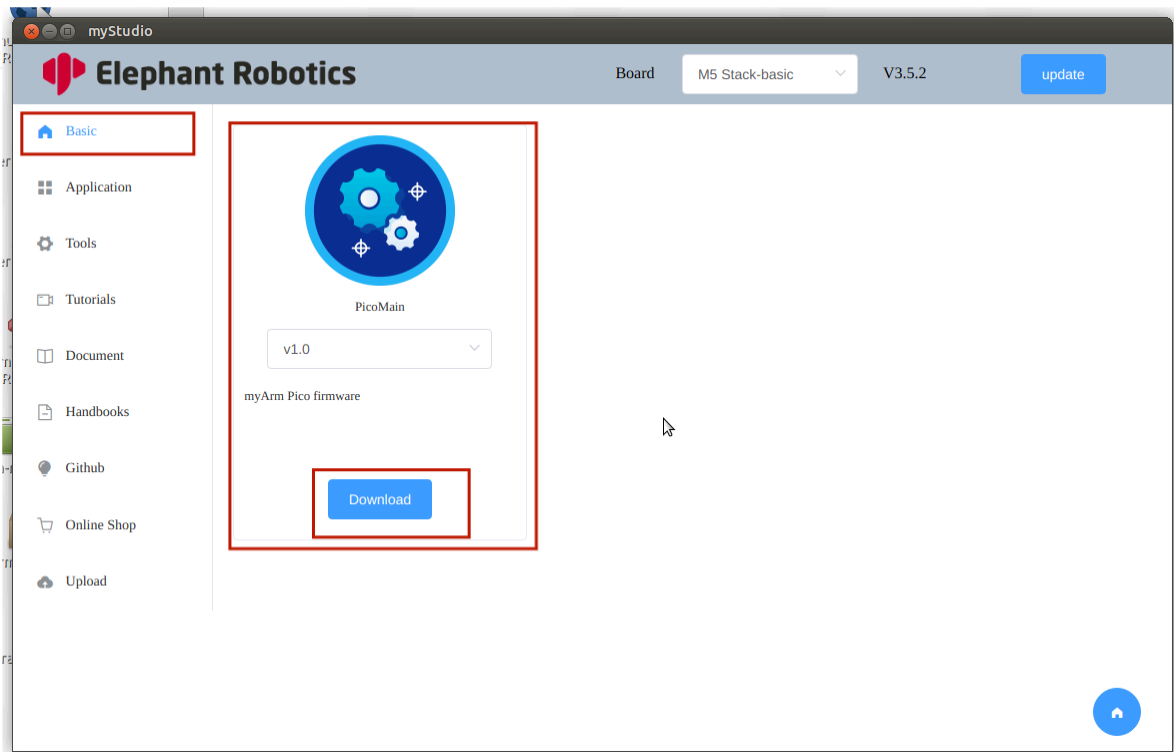
Step 1: Turn on the computer and connect the screen.

For detailed tutorials, please refer to: [initial-setup](#)

Step 2: Select the myArm image, select the model language and region, and click to login.



Step 3: After login, click 'Basic', select picoMain, click download, and then click the burn button to burn.



Message Board

💖 We're delighted to receive your comments, if you have any suggestions or questions about our gitbook, you can write here, we'll reply you as soon as possible, thanks ❤️

⚠️ If you can't see the message leaving place, it may be due to 🌐 network loading problems, please refresh the page, thanks !

