PICO configure camera

Note: The esp32 camera needs to be burned with the factory firmware. If the esp32 camera has not been flashed with the firmware after it is received, it does not need to be burned. The factory default factory firmware, before using iic communication, you can use the serial port to configure the esp32 camera to the network, and iic is used for data reading

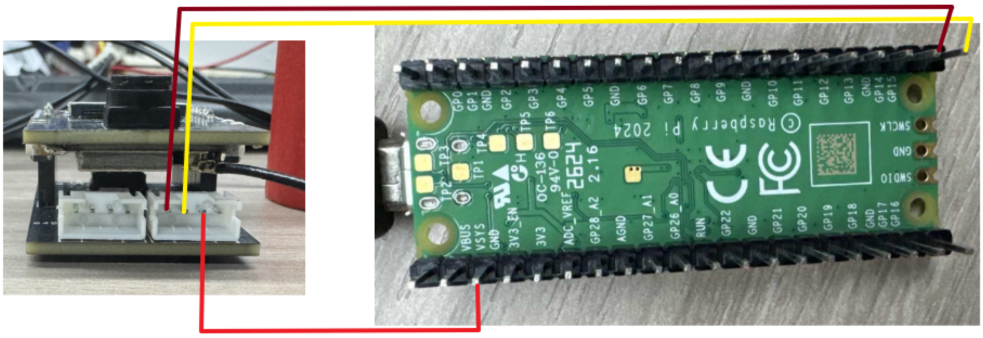
1. Experimental preparation

- A pico
- An esp32 camera

2. Wiring diagram

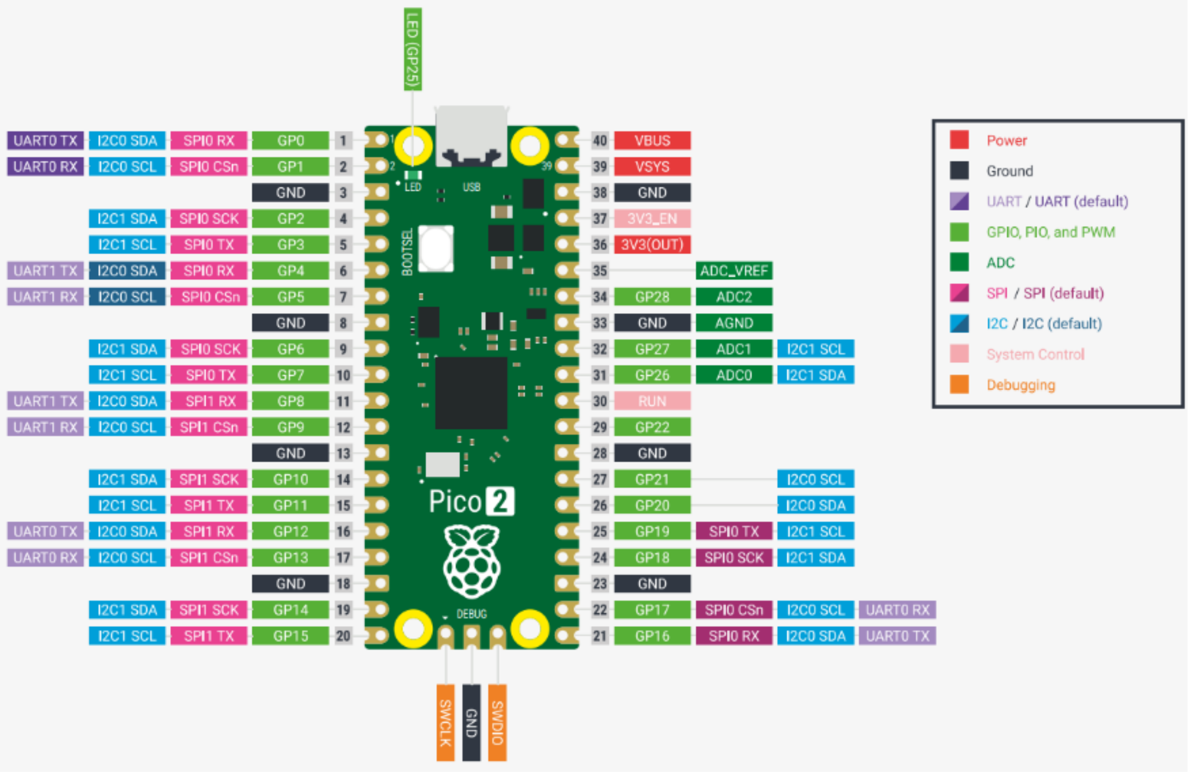
Note: Because the pico main control does not have a 5v interface, it is necessary to use type-c to power the esp32 camera

pico	esp32 camera
P14	SDA
P15	SCL
GND	GND
NC	5V



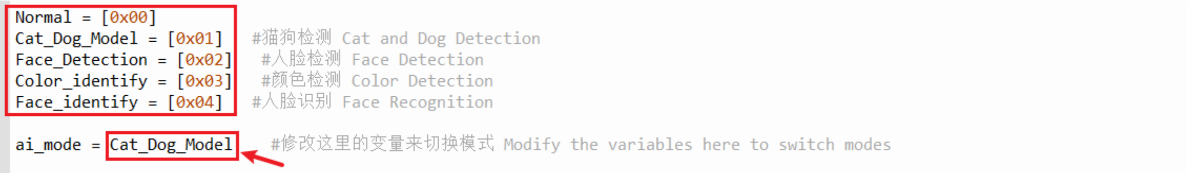
Physical connection diagram:

PICO pin diagram:


LED (GP25)
UART0 TX | I2C0 SDA | SPI0 RX | GP0 | 1
UART0 RX | I2C0 SCL | SPI0 CSn | GP1 | 2
GND | 3
I2C1 SDA | SPI0 SCK | GP2 | 4
I2C1 SCL | SPI0 TX | GP3 | 5
UART1 TX | I2C0 SDA | SPI0 RX | GP4 | 6
UART1 RX | I2C0 SCL | SPI0 CSn | GP5 | 7
GND | 8
I2C1 SDA | SPI0 SCK | GP6 | 9
I2C1 SCL | SPI0 TX | GP7 | 10
UART1 TX | I2C0 SDA | SPI1 RX | GP8 | 11
UART1 RX | I2C0 SCL | SPI1 CSn | GP9 | 12
GND | 13
I2C1 SDA | SPI1 SCK | GP10 | 14
I2C1 SCL | SPI1 TX | GP11 | 15
UART0 TX | I2C0 SDA | SPI1 RX | GP12 | 16
UART0 RX | I2C0 SCL | SPI1 CSn | GP13 | 17
GND | 18
I2C1 SDA | SPI1 SCK | GP14 | 19
I2C1 SCL | SPI1 TX | GP15 | 20
40 | VBUS
39 | VSYS
38 | GND
37 | 3V3_EN
36 | 3V3(OUT)
35 | ADC_VREF
34 | GP28 | ADC2
33 | GND | AGND
32 | GP27 | ADC1 | I2C1 SCL
31 | GP26 | ADC0 | I2C1 SDA
30 | RUN
29 | GP22
28 | GND
27 | GP21 | I2C0 SCL
26 | GP20 | I2C0 SDA
25 | GP19 | SPI0 TX | I2C1 SCL
24 | GP18 | SPI0 SCK | I2C1 SDA
23 | GND
22 | GP17 | SPI0 CSn | I2C0 SCL | UART0 RX
21 | GP16 | SPI0 RX | I2C0 SDA | UART0 TX
LED
USB
BOOTSEL
Pico 2
DEBUG
SWCLK
GND
SWDIO
Power
Ground
UART / UART (default)
GPIO, PIO, and PWM
ADC
SPI / SPI (default)
I2C / I2C (default)
System Control
Debugging

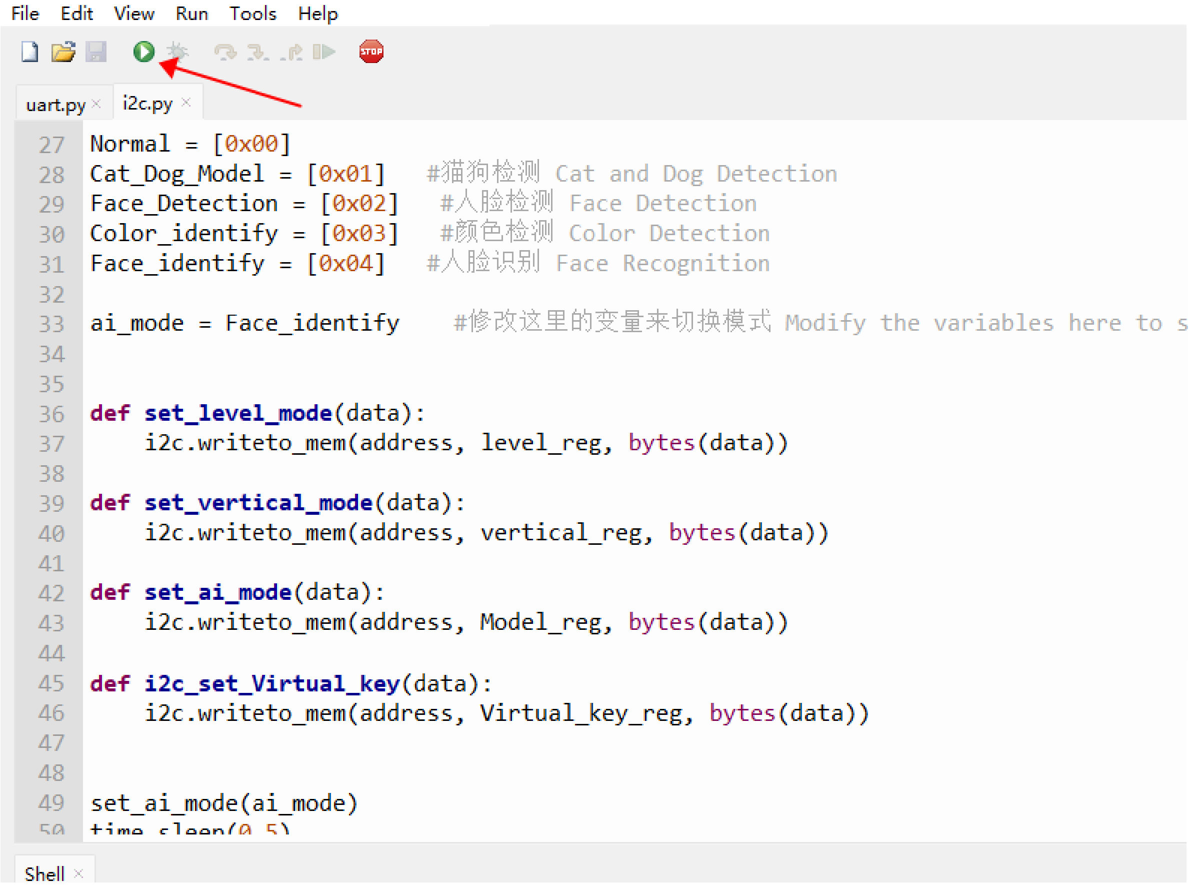
3. Experimental steps and experimental results

1. Open a thonny software
2. Open the code just uploaded, and change the position of the arrow in the figure to the corresponding ai mode.

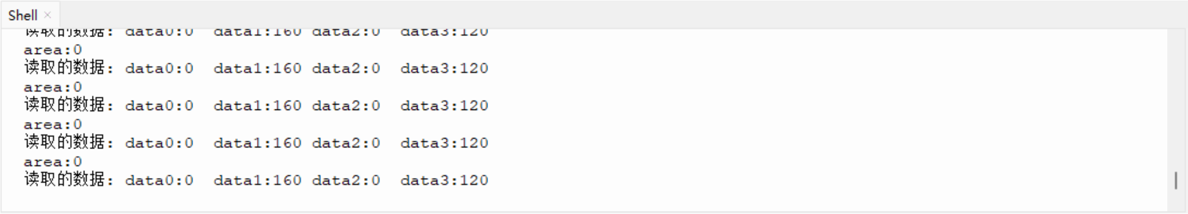


```
Normal = [0x00]
Cat_Dog_Model = [0x01]      #猫狗检测 Cat and Dog Detection
Face_Detection = [0x02]      #人脸检测 Face Detection
Color_identify = [0x03]      #颜色检测 Color Detection
Face_identify = [0x04]      #人脸识别 Face Recognition

ai_mode = Cat_Dog_Model       #修改这里的变量来切换模式 Modify the variables here to switch modes
```

3. Run the program, and the coordinates of the current mode will be printed below

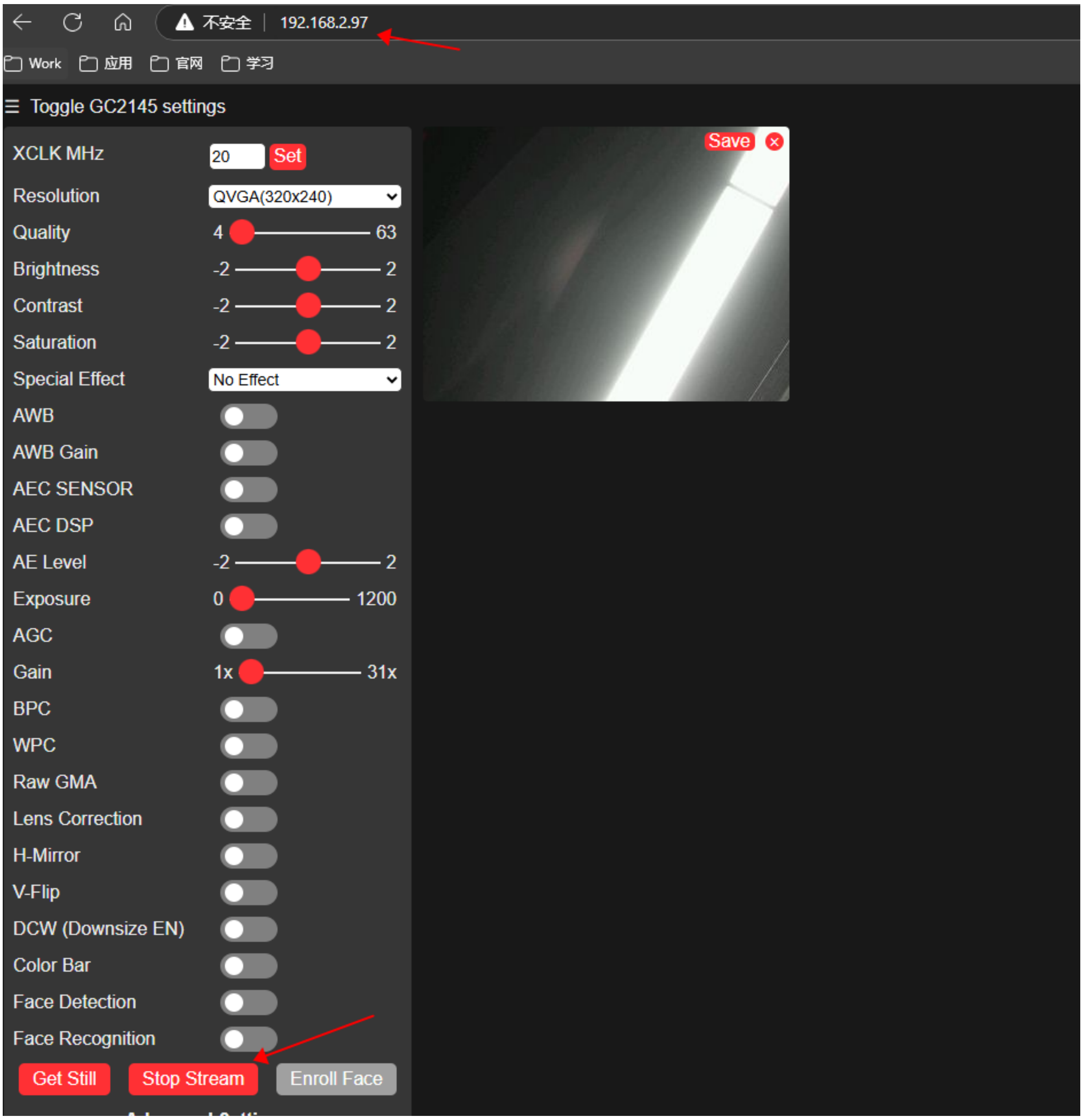


```
Normal = [0x00]
Cat_Dog_Model = [0x01]   #猫狗检测 Cat and Dog Detection
Face_Detection = [0x02]   #人脸检测 Face Detection
Color_identify = [0x03]   #颜色检测 Color Detection
Face_identify = [0x04]   #人脸识别 Face Recognition

ai_mode = Face_identify    #修改这里的变量来切换模式 Modify the variables here to s


def set_level_mode(data):
    i2c.writeto_mem(address, level_reg, bytes(data))

def set_vertical_mode(data):
    i2c.writeto_mem(address, vertical_reg, bytes(data))

def set_ai_mode(data):
    i2c.writeto_mem(address, Model_reg, bytes(data))

def i2c_set_Virtual_key(data):
    i2c.writeto_mem(address, Virtual_key_reg, bytes(data))


set_ai_mode(ai_mode)
time.sleep(0.5)
```



```
读取的数据: data0:0  data1:160 data2:0  data3:120
area:0
读取的数据: data0:0  data1:160 data2:0  data3:120
area:0
读取的数据: data0:0  data1:160 data2:0  data3:120
area:0
读取的数据: data0:0  data1:160 data2:0  data3:120
area:0
读取的数据: data0:0  data1:160 data2:0  data3:120
```

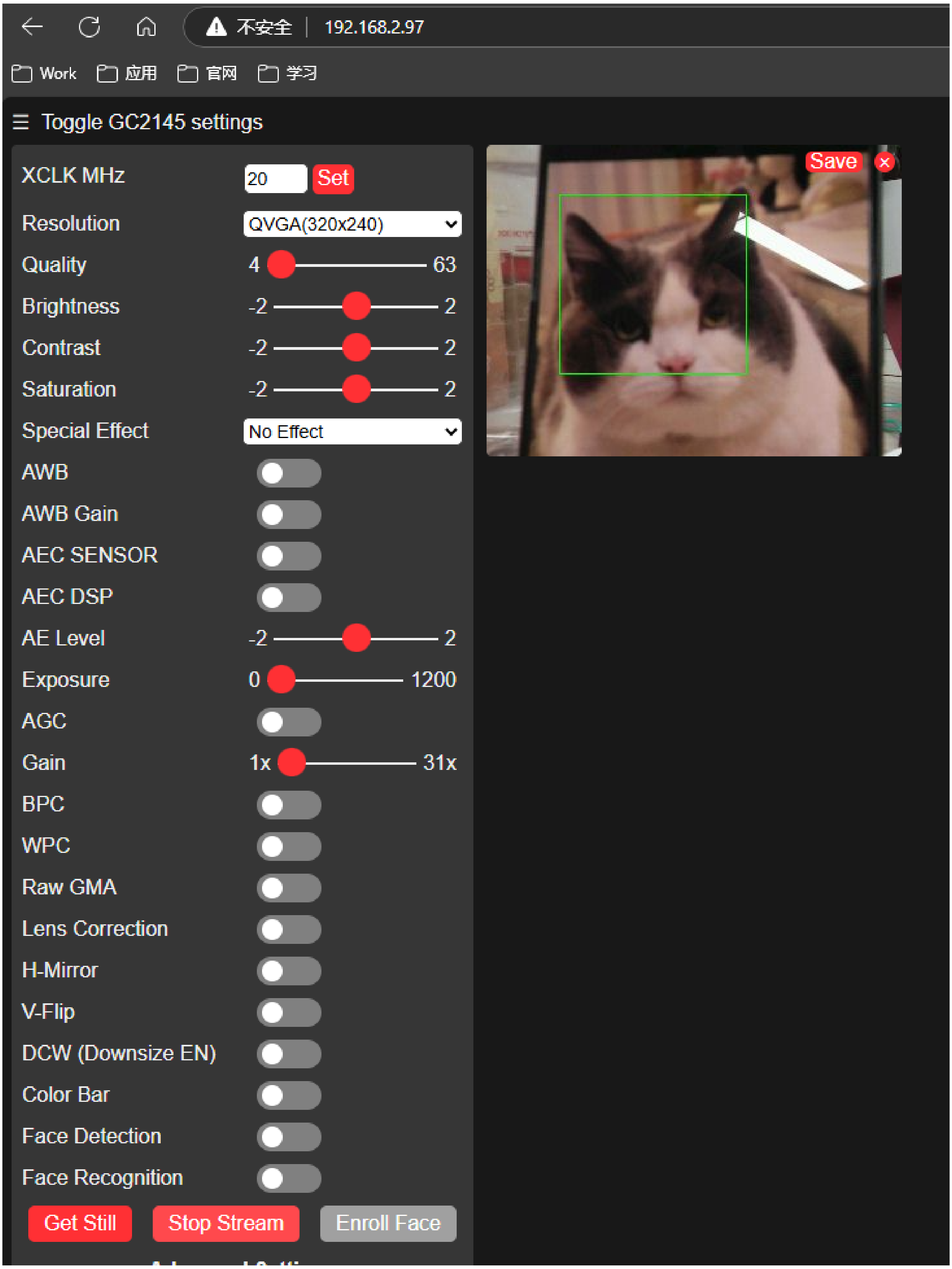
3. Log in through the wifi ip configured by the serial port, or through the hotspot connected to the camera. Because wifi has been configured before, the following figure directly uses the previously configured wifi ip address to log in to the webpage and open the camera



☰ Toggle GC2145 settings

XCLK MHz 20 Set
Resolution QVGA(320x240)
Quality 4 ● 63
Brightness -2 ● 2
Contrast -2 ● 2
Saturation -2 ● 2
Special Effect No Effect
AWB ○
AWB Gain ○
AEC SENSOR ○
AEC DSP ○
AE Level -2 ● 2
Exposure 0 ● 1200
AGC ○
Gain 1x ● 31x
BPC ○
WPC ○
Raw GMA ○
Lens Correction ○
H-Mirror ○
V-Flip ○
DCW (Downsize EN) ○
Color Bar ○
Face Detection ○
Face Recognition ○

4. Because we set it to cat and dog mode, we put the cat or dog in front of the camera and the cat or dog will be selected.



☰ Toggle GC2145 settings

XCLK MHz 20

Resolution QVGA(320x240)

Quality 4 — 63

Brightness -2 — 2

Contrast -2 — 2

Saturation -2 — 2

Special Effect No Effect

AWB ○

AWB Gain ○

AEC SENSOR ○

AEC DSP ○

AE Level -2 — 2

Exposure 0 — 1200

AGC ○

Gain 1x — 31x

BPC ○

WPC ○

Raw GMA ○

Lens Correction ○

H-Mirror ○

V-Flip ○

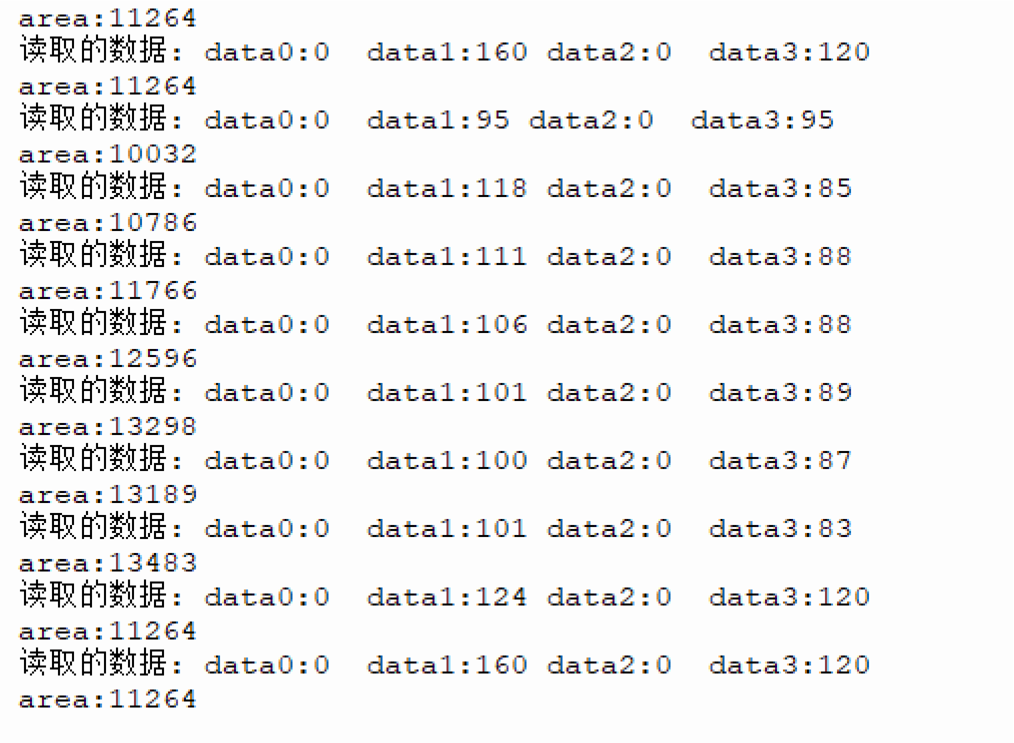
DCW (Downsize EN) ○

Color Bar ○

Face Detection ○

Face Recognition ○

At the same time, the terminal will print out the current center coordinates and the area has been selected.



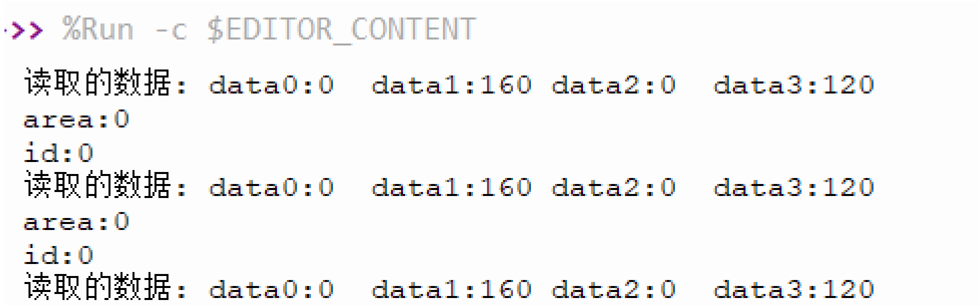
```
area:11264
读取的数据: data0:0  data1:160 data2:0  data3:120
area:11264
读取的数据: data0:0  data1:95 data2:0  data3:95
area:10032
读取的数据: data0:0  data1:118 data2:0  data3:85
area:10786
读取的数据: data0:0  data1:111 data2:0  data3:88
area:11766
读取的数据: data0:0  data1:106 data2:0  data3:88
area:12596
读取的数据: data0:0  data1:101 data2:0  data3:89
area:13298
读取的数据: data0:0  data1:100 data2:0  data3:87
area:13189
读取的数据: data0:0  data1:101 data2:0  data3:83
area:13483
读取的数据: data0:0  data1:124 data2:0  data3:120
area:11264
读取的数据: data0:0  data1:160 data2:0  data3:120
area:11264
```

Face recognition mode

When switching to face recognition mode, the terminal will additionally print the current face id

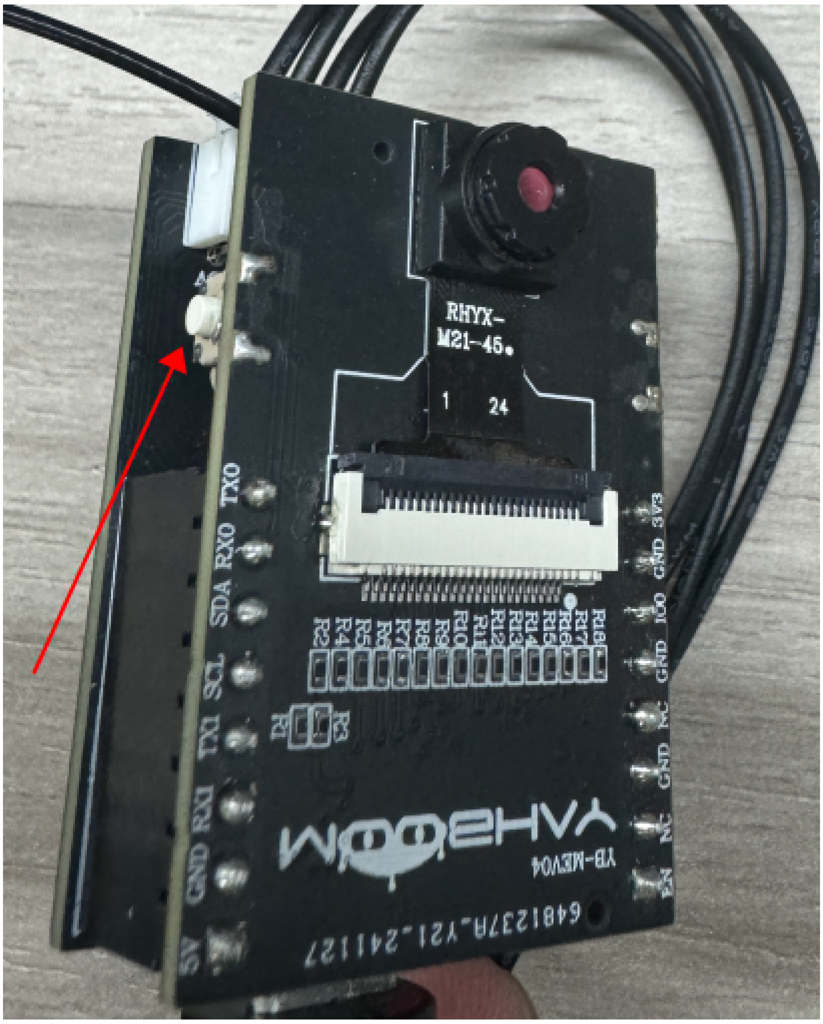
```
ai_mode = Face_identify    #修改这里的变量来切换模式 Modify the variables here to switch modes
```

```
python3 I2C_TEST.py
```

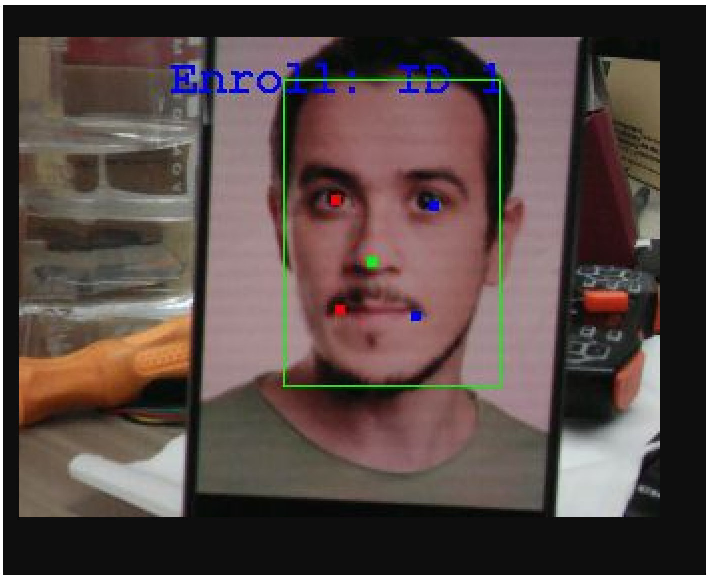


```
>> %Run -c $EDITOR_CONTENT
 读取的数据: data0:0  data1:160 data2:0  data3:120
 area:0
 id:0
 读取的数据: data0:0  data1:160 data2:0  data3:120
 area:0
 id:0
 读取的数据: data0:0  data1:160 data2:0  data3:120
```

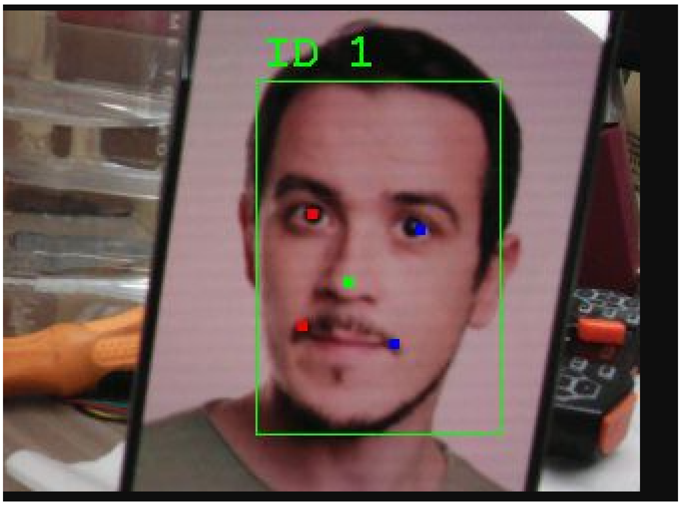
Recognize face. When you see a face, press the key button to record the face



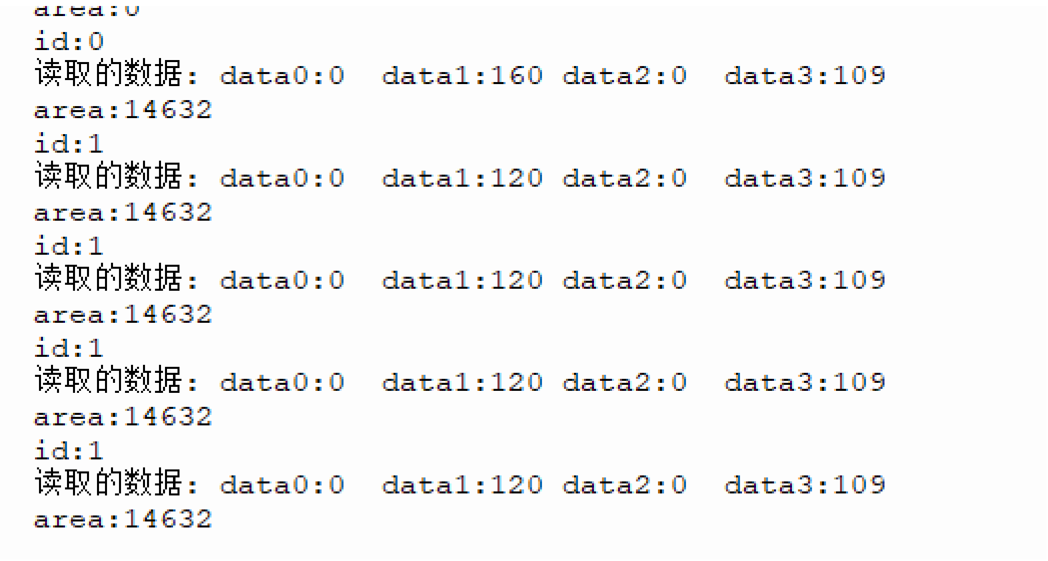
The following picture appears, which means the recording is successful, and the face 1 is recorded



At this time, you can press and hold the button for two seconds, then release it and press the button again to recognize the current face



At the same time, the terminal will print out the current center coordinates and the recognized face.



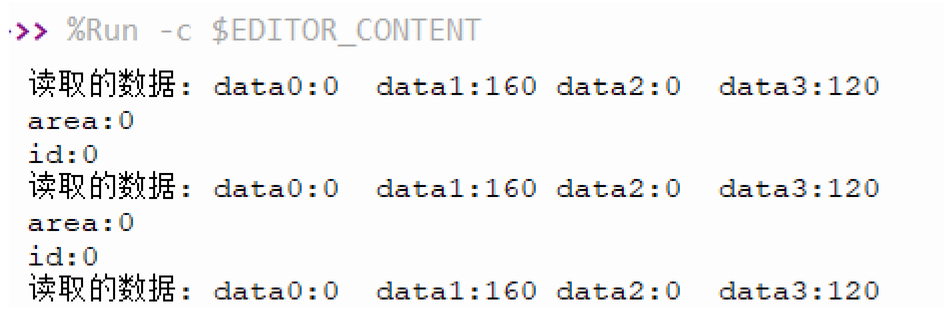
```
area:0
id:0
读取的数据: data0:0  data1:160 data2:0  data3:109
area:14632
id:1
读取的数据: data0:0  data1:120 data2:0  data3:109
area:14632
id:1
读取的数据: data0:0  data1:120 data2:0  data3:109
area:14632
id:1
读取的数据: data0:0  data1:120 data2:0  data3:109
area:14632
id:1
读取的数据: data0:0  data1:120 data2:0  data3:109
area:14632
```

Color detection mode

When we switch to color detection mode,

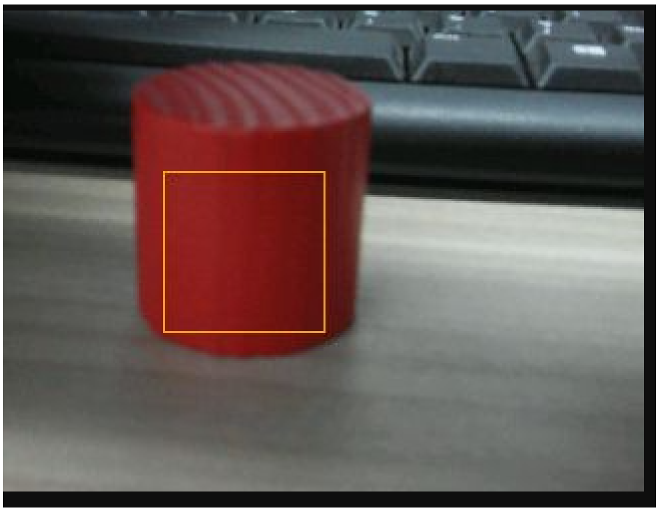
```
ai_mode = Color_identify    #修改这里的变量来切换模式 Modify the variables here to switch modes
```

```
python3 I2C_TEST.py
```

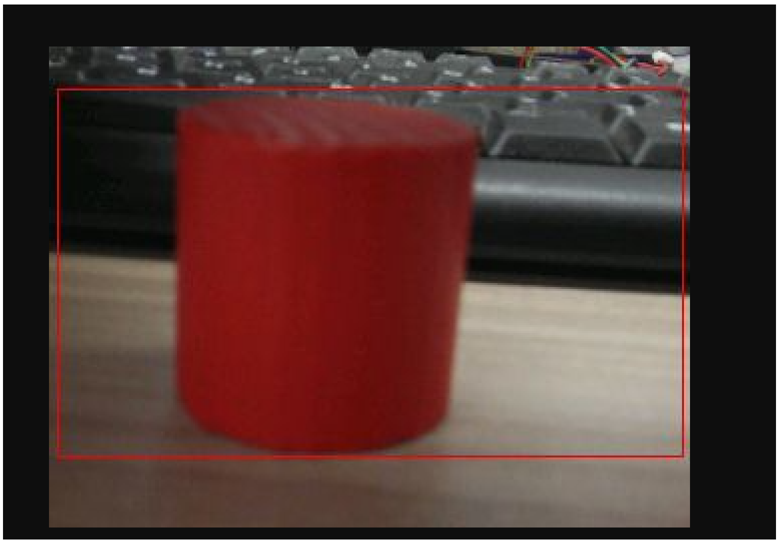


```
>> %Run -c $EDITOR_CONTENT
 读取的数据: data0:0  data1:160 data2:0  data3:120
 area:0
 id:0
 读取的数据: data0:0  data1:160 data2:0  data3:120
 area:0
 id:0
 读取的数据: data0:0  data1:160 data2:0  data3:120
```

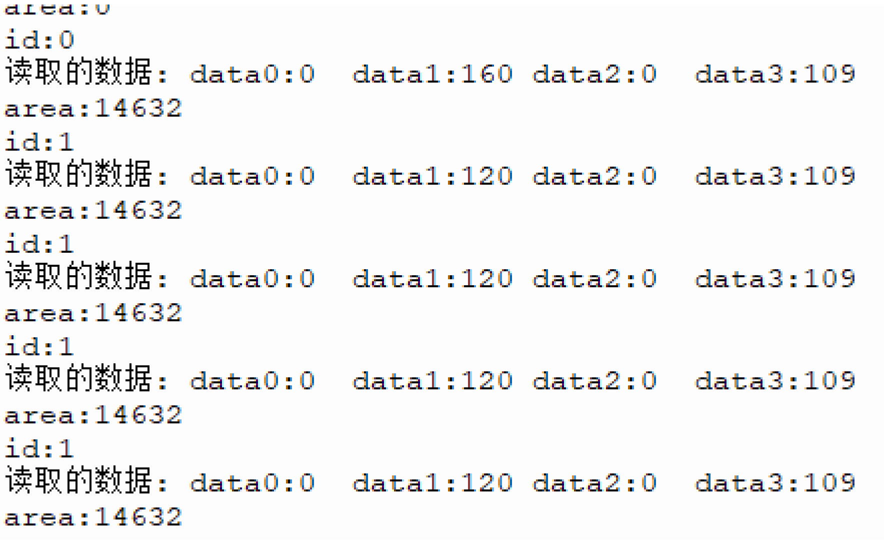
Recognize the color. Press the button and a box will appear. You can use this box to select the color you want to use.



Press and hold the button for two seconds, release it and press it again to identify the currently selected color, and a red frame will appear.



At the same time, the terminal will print out the current center coordinates.



```
area:0
id:0
读取的数据: data0:0  data1:160 data2:0  data3:109
area:14632
id:1
读取的数据: data0:0  data1:120 data2:0  data3:109
area:14632
id:1
读取的数据: data0:0  data1:120 data2:0  data3:109
area:14632
id:1
读取的数据: data0:0  data1:120 data2:0  data3:109
area:14632
id:1
读取的数据: data0:0  data1:120 data2:0  data3:109
area:14632
```