Lesson 2 Color Sorting

1. Working Principle

The color sensor is a sensor with various functions such as recognizing the color of objects, detecting the brightness of the surrounding environment, realizing object proximity detection and non-contact gesture detection, etc.

This lesson will use the color sensor for recognizing and comparing the RGB value of object. These sensors generally determine whether the detected color is consistent with the set one by using RGB (red, green, and blue) LED light sources and the ratio of R, G, and B wavelengths in the reflected light of an object.

The path of the program file: “6. Secondary Development/Python Development/Sensor-extension Game/Color Sorting/Program Files/Color Sorting/main.py”.

```
import time
from machine import Pin, I2C
from Ultrasonic import ULTRASONIC
from Color_sensor import COLOR
from Buzzer import Buzzer
from espmax import ESPMax
from PWMServo import PWMServo
from BusServo import BusServo
from RobotControl import RobotControl
from SuctionNozzle import SuctionNozzle

# 颜色分拣

pwm = PWMServo()
buzzer = Buzzer()
bus_servo = BusServo()
arm = ESPMax(bus_servo)
robot = RobotControl()
nozzle = SuctionNozzle()

```

Firstly, import the corresponding libraries and initialize ultrasonic sensor and color sensor.

Then read and calculate the RGB value of the object, and print out the color result.

Finally, the ultrasonic sensor emits corresponding light, and the functions for controlling servo and air pump are executed. MaxArm will suck and place the object to the corresponding position according to the color.

2. Preparation

2.1 Hardware

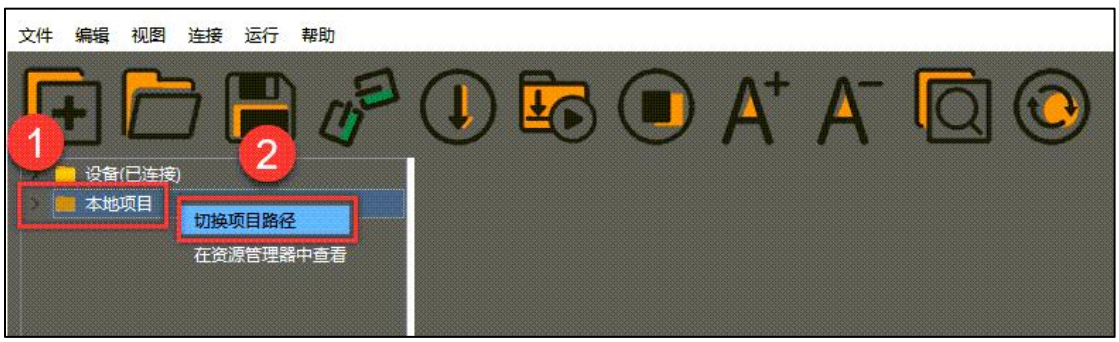
Please assemble the color sensor and the ultrasonic sensor to the corresponding position on MaxArm according to the tutorial in folder “Lesson 1 Sensor Assembly” under the same directory.

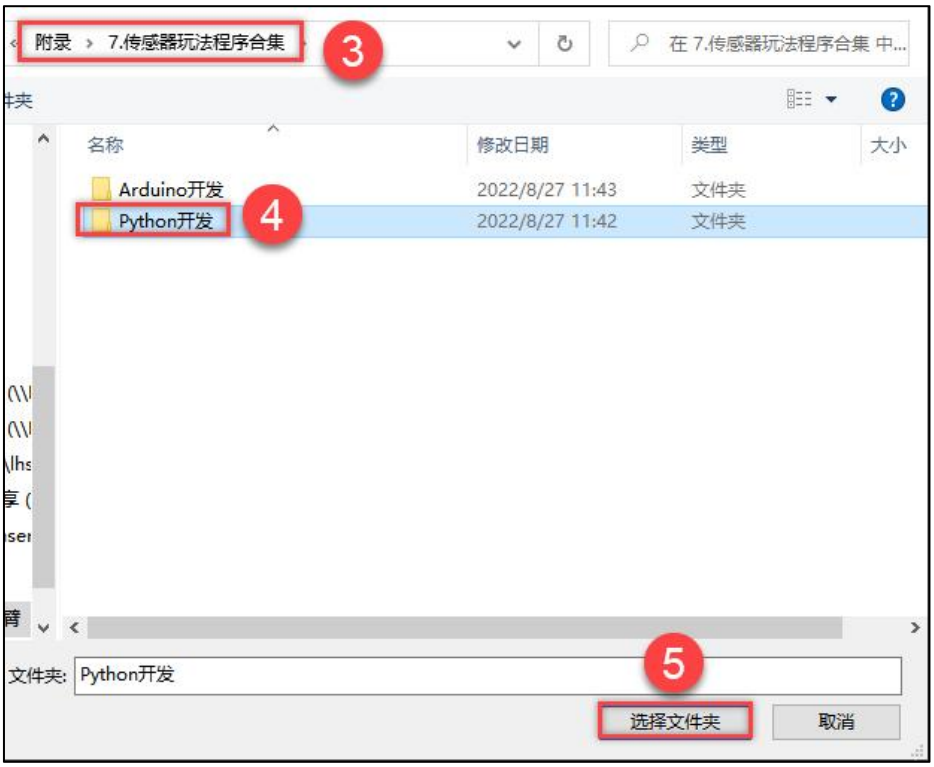
1.2 Software

Please connect MaxArm to Python editor according to the tutorial in folder “4. Underlying Program/Python Development/Lesson 1 Set Development Environment”.

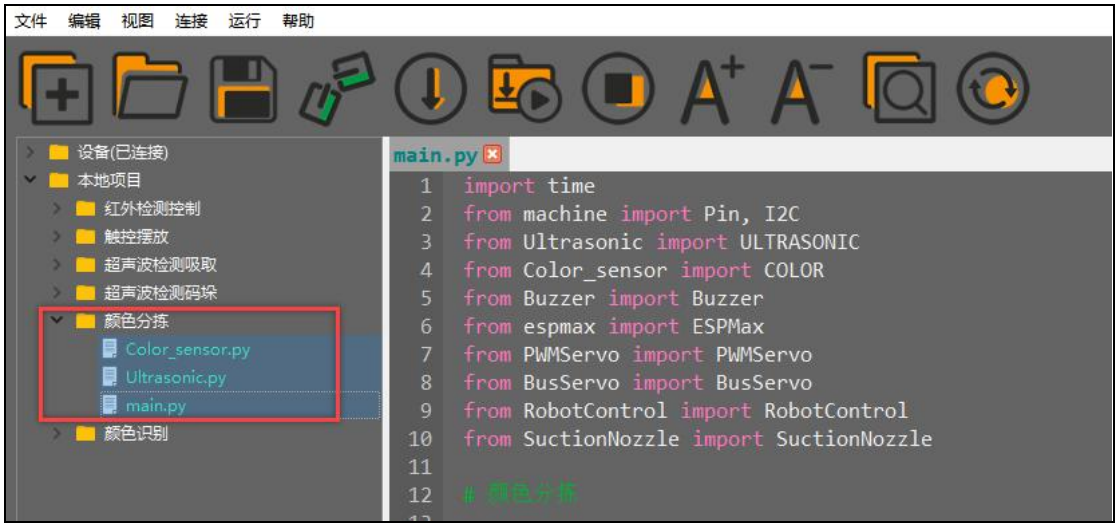
3. Program Download

1) After connecting, change the path of Workspace to “Appendix/ 7. Sensor-extension Game/Python Development” and select the folder “Program Files”.

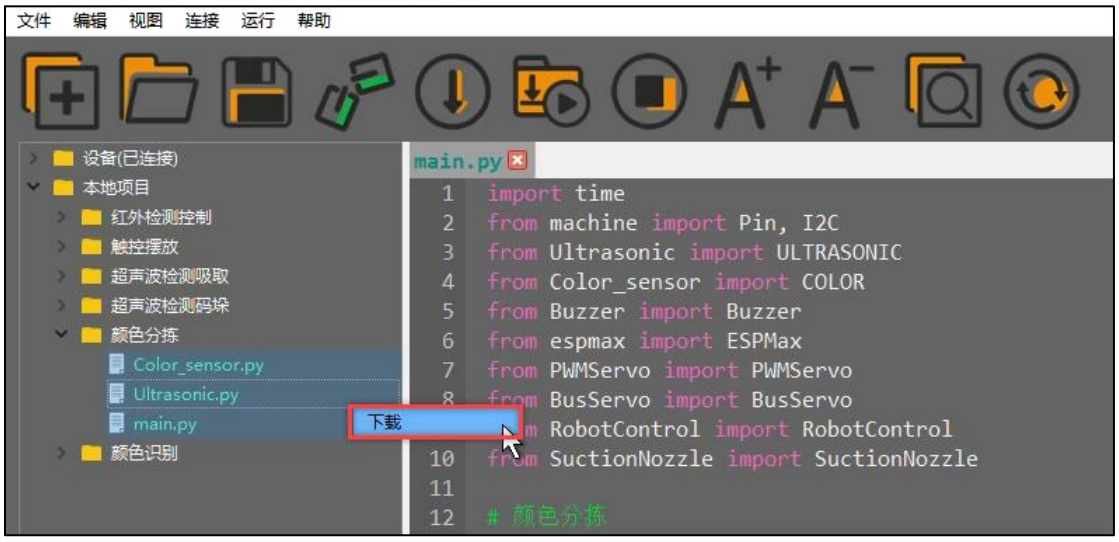




2) Click the folder “Color Sorting”, and then select all the program files in the folder.

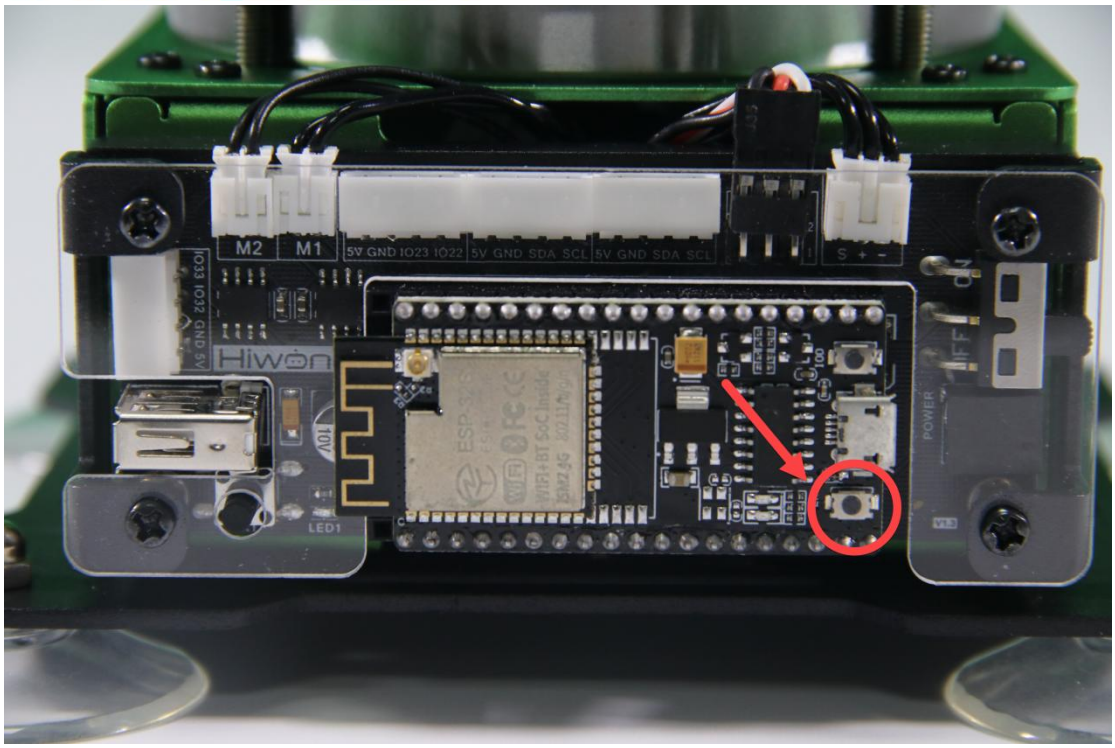


3) Then right click to download all the program files to the controller.



4) When the terminal prints the prompt as shown in the image below, it means download completed.

5) After downloading, click on the reset icon or press the reset button on ESP32 controller to run program.



4. Project Outcome

After the color sensor recognizes the color of the block and the ultrasonic sensor detects that the block is placed in the placement area, MaxArm will sort the block based on the recognized color, and suck and transport the block to the corresponding position.

5. Program Instruction

5.1 Import function library

The path of the program file: “6. Secondary Development/Sensor-extension Game/Python Development/Color Sorting/Program Files/Color Sorting/ main.py”.

Before executing the program, the I2C protocol, ultrasonic sensor, color sensor, buzzer, PWM servo, bus servo, infrared sensor, air pump and other related

Python function libraries need to be imported.

```
import time
from machine import Pin, I2C
from Ultrasonic import ULTRASONIC
from Color_sensor import COLOR
from Buzzer import Buzzer
from espmax import ESPMax
from PWMServo import PWMServo
from BusServo import BusServo
from RobotControl import RobotControl
from SuctionNozzle import SuctionNozzle
```

5.2 Color Detection

Use apds.readRedLight(), apds.readGreenLight() and apds.readBlueLight() functions of color sensor to detect the value of RGB channel of object, and then calculate the value.

```
    # 解析颜色传感器数据
    c = apds.readAmbientLight()
    r = apds.readRedLight()
    g = apds.readGreenLight()
    b = apds.readBlueLight()
    r = int(255 * (r - r_f) / (R_F - r_f))
    g = int(255 * (g - g_f) / (G_F - g_f))
    b = int(255 * (b - b_f) / (B_F - b_f))
```

5.3 Control the LED Light of Ultrasonic Sensor

After calculating the value of RGB channel of object color, determine the color of the object.

```
    if r > 25 and r > g and r > b:t = RED  # 输出颜色检测结果
    elif g > 25 and g > r and g > b:t = GREEN
    elif b > 25 and b > g and b > r:t = BLUE
    else:t = 0
```

And the block with different color is set to place in the different position.

```
    if t > 0:
        buzzer.setBuzzer(100)
        color = t # 得到颜色检测的结果
        print('color:',color)
        if color == 1: # 根据相应颜色设置超声波rgb灯的颜色
            angle = -45
            (x,y,z) = (120,-140,85)
```

Then, the ultrasonic sensor emits the corresponding color light.

```
        if color == 1: # 根据相应颜色设置超声波rgb灯的颜色
            angle = -45
            (x,y,z) = (120,-140,85)
            hwsr06.setRGBValue(bytes([255,0,0, 255,0,0]))
        elif color == 2:
            angle = -25
            (x,y,z) = (120,-80,85)
            hwsr06.setRGBValue(bytes([0,255,0, 0,255,0]))
        elif color == 3:
            angle = 0
            (x,y,z) = (120,-20,82)
            hwsr06.setRGBValue(bytes([0,0,255, 0,0,255]))
```

Use hwsr06.setRGBValue(bytes()) function to control the LED light of the ultrasonic sensor. Take the code “hwsr06.setRGBValue(bytes([255,0,0, 255,0,0]))” as example.

The first three parameters "255,0,0" are the color thresholds for the LED on the right side of the senor.

The last three parameters "255,0,0" are the color thresholds for the left LED of the sensor.

At this time, the LED emits red light.

5.4 Ultrasonic Detection

If the block is detected, use hwsr06.getDistance() function to measure distance. Then use print() function to print out the measured distance.

```
    if color > 0: # 检测到颜色
        Distance = hwsr06.getDistance() # 获取超声波检测距离
        print('distance:', Distance)
```

5.5 Control Robotic Arm

Determine whether there are objects at the distance between 70 and 80, and then execute the corresponding action.

```
85      if 70 < Distance < 80: # 色块距离满足吸取的条件，进行分拣
86        buzzer.setBuzzer(100)  #设置蜂鸣器响100ms
87        time.sleep_ms(1000)
88        arm.set_position((0,-160,85),1500) #吸取色块
89        nozzle.on() #打开气泵
90        time.sleep_ms(1600)
91        arm.set_position((0,-160,180),1000) #抬起来
92        time.sleep_ms(1000)
93        arm.set_position((x,y,180),1000) #到达相应颜色的放置区上方
94        time.sleep_ms(1000)
95        nozzle.set_angle(angle,800) # 设置角度补偿
96        arm.set_position((x,y,z),800) # 放置色块
97        time.sleep_ms(1000)
98        nozzle.off() #关闭气泵
99        arm.set_position((x,y,200),1000) #抬起来
100       time.sleep_ms(1000)
101       arm.go_home() #机械臂复位，回到初始位置
```

Use buzzer.setBuzze() function to control the buzzer. Take the code “buzzer.setBuzzer(100)” as example.

The first parameter “100” represents the sounding time of buzzer and the unit is ms.

Use arm.set_position() function to control robotic arm. Take the code “arm.set_position((0,-160,85),1500)” as example.

The first parameter “(0，-160，85)” represents is the position of the suction nozzle on x, y and z axes.

The second parameter “1500” is the running time and the unit is ms.

Use nozzle.set_angle() function to control the rotation of the suction nozzle. Take the code “nozzle.set_angle(0，800)” as example.

The first parameter “0” is the angle of PWM servo.

The second parameter “800” is the running time and the unit is ms.