Lesson 2 Infrared Detection and Control

1. Working Principle

Infrared obstacle avoidance is a photoelectric sensor integrating IR transmitter and IR receiver. Featuring long detection distance and low interference from visible light , it is widely used in robot and assembly line piecework, etc.

This sensor detects obstacle by transmitting and receiving infrared. When the infrared transmitted by the sensor meets the obstacle ahead, the infrared will be reflected to the receiving terminal. When the sensor detects this signal, it will send it to the microcontroller for processing.

The closer the obstacle is, the stronger the reflection intensity; the farther the obstacle is, the weaker the reflection intensity. Different surface color has different reflection intensity. White is the strongest and black is the weakest.

The path of the program file: “6. Secondary Development/Python Development/Sensor-extension Game/Program Files/Infrared Detection and Control/main.py”.

```
import time
from Buzzer import Buzzer
from espmax import ESPMax
from PWMServo import PWMServo
from BusServo import BusServo
from Infrared_sensor import INFRARED
from RobotControl import RobotControl
from SuctionNozzle import SuctionNozzle


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

```

Firstly, import the corresponding libraries and initialize buzzer, servo and action group.

Next, detect the object by the infrared sensor and buzzer will make sound. MaxArm will perform the corresponding action.

Then, execute the function for controlling action, buzzer, air pump to suck and place the object.

2. Preparation

2.1 Hardware

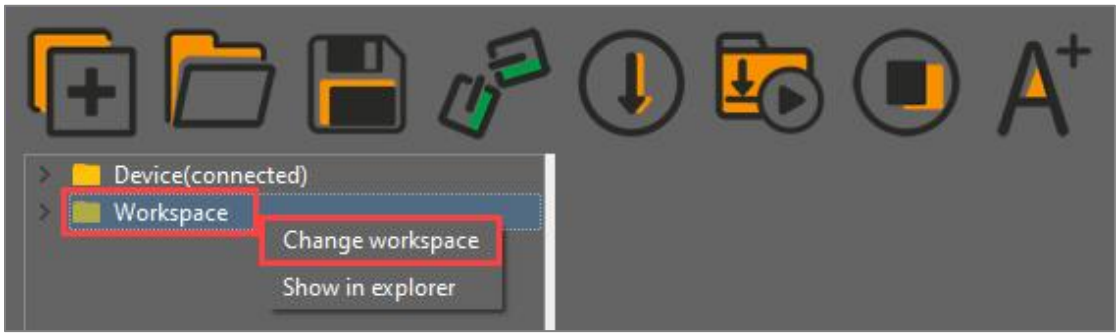
Please assemble the infrared sensor to the corresponding position on MaxArm according to the tutorial in folder “Lesson 1 Sensor Assembly” under the same directory.

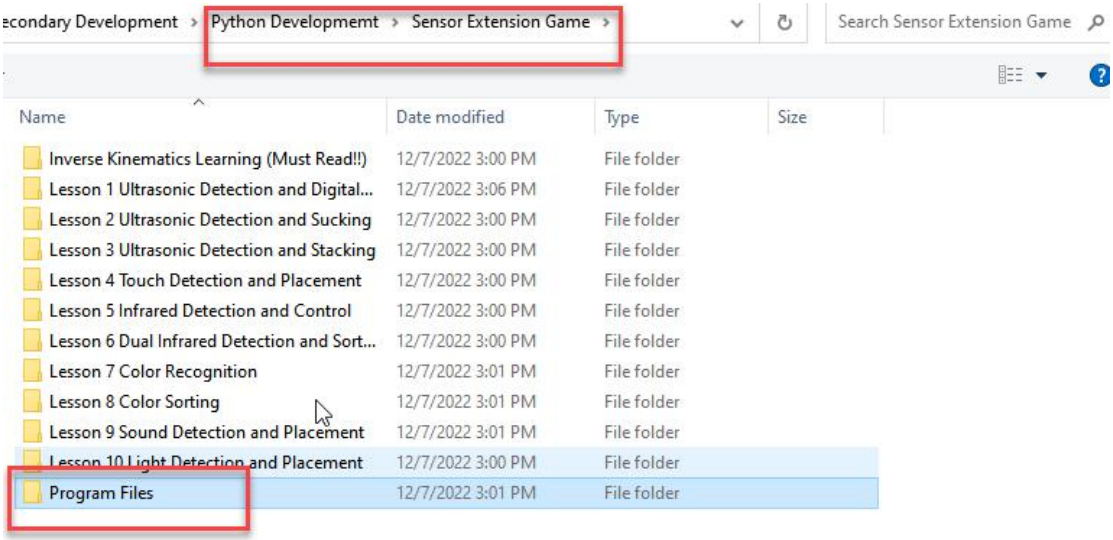
1.2 Software

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

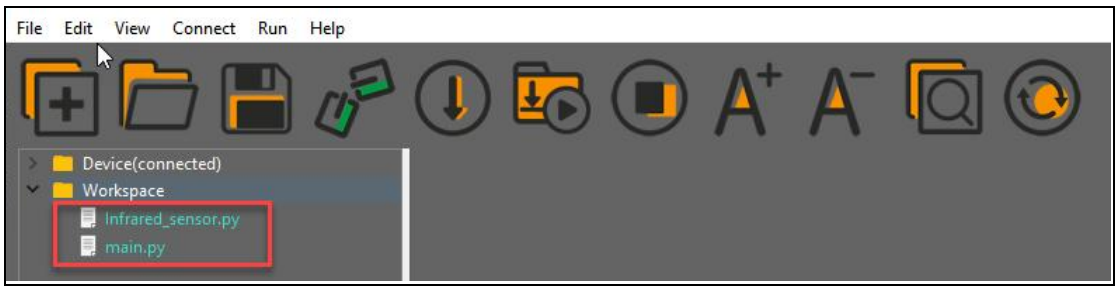
3. Program Download

1) After connecting, change the path of Workspace to “6. Secondary Development/Python Development/Sensor-extension Game”, and select “Program Files”.





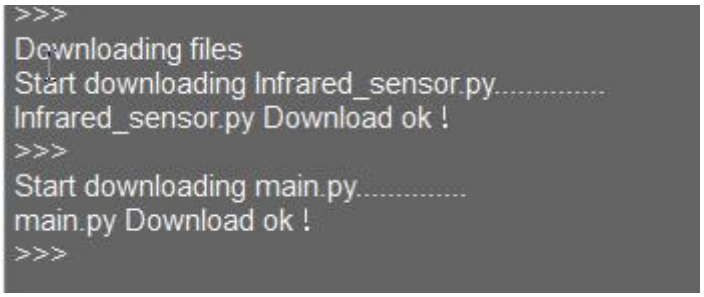
2) Click the folder “Infrared Detection and Control”, and then select all the program files in the folder.



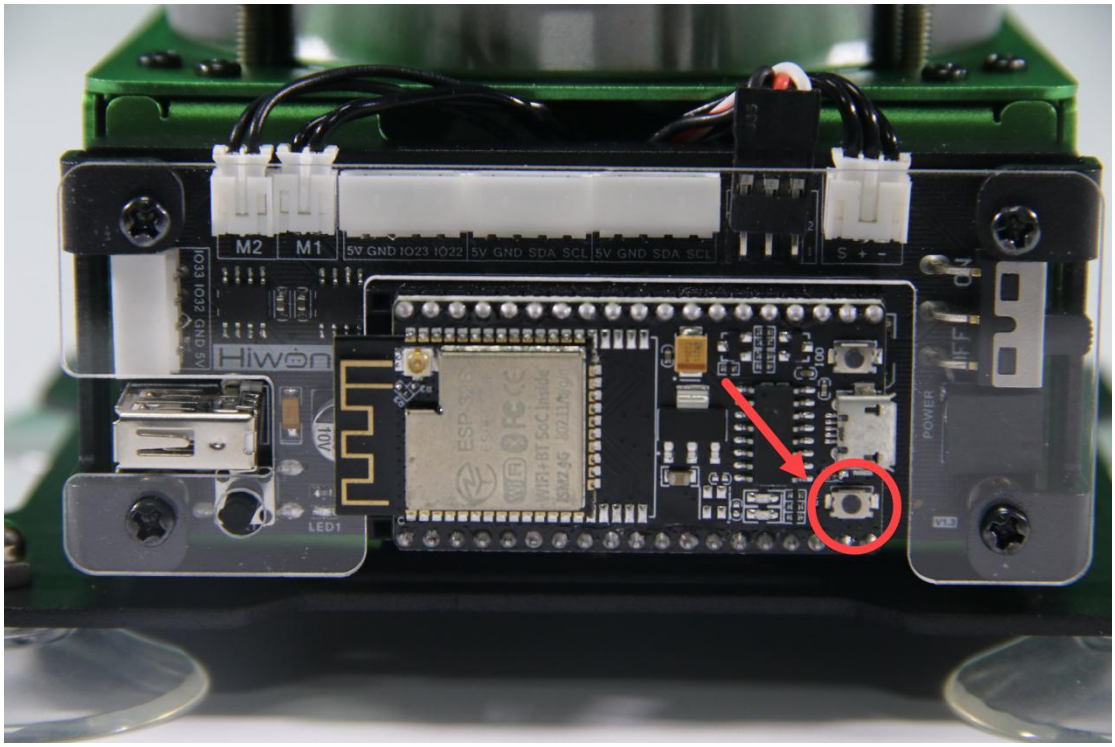
3) Then right click to download all the program files to the controller.

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 buzzer make “Di” sound, MaxArm will suck and move the block to the placement area. Finally, the air pump will be turned off and MaxArm will return to the initial posture

5. Program Instruction

5.1 Import Function Library

The path of the program file: “6. Secondary Development/Python Development/Sensor-extension Game/Program Files/Infrared Detection and Control/main.py”.

Before executing the program, the buzzer, PWM servo, bus servo, infrared sensor, air pump and other related Python function libraries need to be imported.

```
1 import time
2 from Buzzer import Buzzer
3 from espmax import ESPMax
4 from PWMServo import PWMServo
5 from BusServo import BusServo
6 from Infrared_sensor import INFRARED
7 from RobotControl import RobotControl
8 from SuctionNozzle import SuctionNozzle
```

5.2 Infrared Detection

Use the infrared.run_loop() function to detect whether there is a object.

```
27 while True:
28     infrared.run_loop()
```

5.3 Control robotic arm

After object is detected by the infrared sensor, MaxArm will execute the corresponding action.

```
    if infrared.close_long():
        buzzer.setBuzzer(100)
        arm.set_position((0,-160,100),1500)
        time.sleep_ms(1000)
        arm.set_position((0,-160,85),800)
        nozzle.on()
        time.sleep_ms(1000)
        arm.set_position((0,-160,200),1000)
        time.sleep_ms(1000)
        arm.set_position((70,-150,200),800)
        nozzle.set_angle(30,600)
        time.sleep_ms(1000)
        nozzle.set_angle(35,300)
        arm.set_position((70,-150,90),800)
        time.sleep_ms(800)
        arm.set_position((130,-150,88),500)
        time.sleep_ms(500)
        nozzle.off()
```

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

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

Use the arm.set_position() function to control the robotic arm. Take the code “arm.set_position((0,-160,100),1500)” as example.

The first parameter “(0，-160，100)” 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 the nozzle.set_angle() function to control the rotation of the suction nozzle. Take the code “nozzle.set_angle(30,600)” as example.

The first parameter “30” represents the angle of PWM servo.

The second parameter “600” represents the running time and the unit is ms.