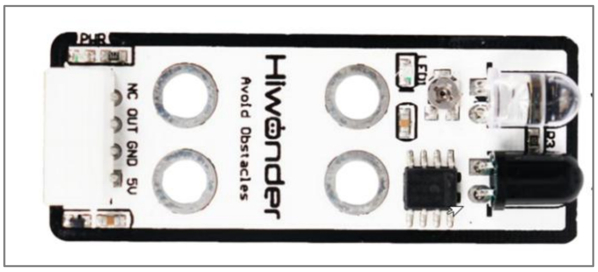
Lesson 2 Dual Infrared Detection and Sorting

1. Project Principle

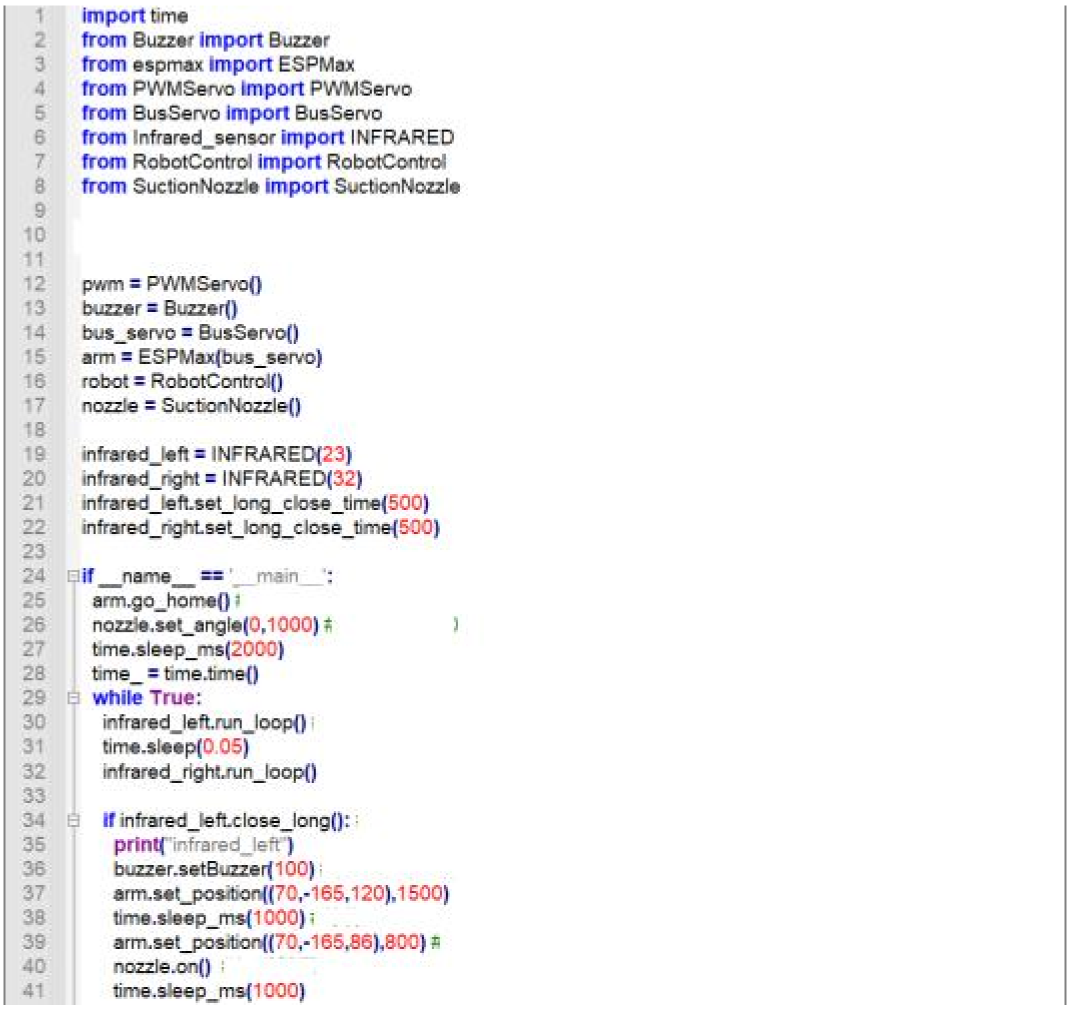
Infrared obstacle avoidance is a photoelectric sensor integrating IR transmitter and IR receiver. Featuring long detection distance and low interference from visible light , it is widely used in robot and assembly line piecework, etc.

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. After the signal is detected, it will travel to 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 program file: “6. Sensor-extension game/Python Development/ Program Files/ Dual Infrared Sorting/main.py”.



```python
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()
bus_servo = BusServo()
arm = ESPMax(bus_servo)
robot = RobotControl()
nozzle = SuctionNozzle()

infrared_left = INFRARED(23)
infrared_right = INFRARED(32)
infrared_left.set_long_close_time(500)
infrared_right.set_long_close_time(500)

if __name__ == '__main__':
  arm.go_home()
  nozzle.set_angle(0,1000) #
  time.sleep_ms(2000)
  time_ = time.time()
  while True:
    infrared_left.run_loop()
    time.sleep(0.05)
    infrared_right.run_loop()

    if infrared_left.close_long():
      print("infrared_left")
      buzzer.setBuzzer(100)
      arm.set_position((70,-165,120),1500)
      time.sleep_ms(1000)
      arm.set_position((70,-165,86),800) #
      nozzle.on()
      time.sleep_ms(1000)
```

Firstly, import the required libraries and reset robotic arm and suction nozzle.

Then call infrared detection function to detect object. When a object is detected, buzzer will sound for responding and robotic arm will suck the object and move it to the corresponding position.

2. Preparation

2.1 Hardware

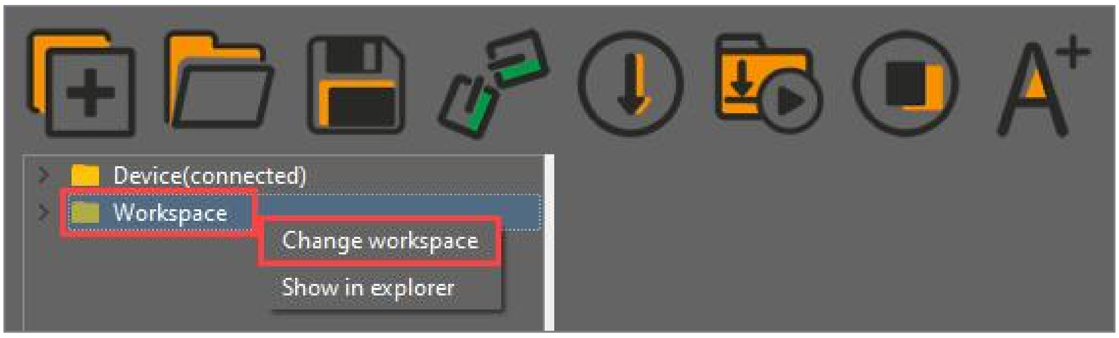
Please assemble infrared sensor to the corresponding position on MaxArm according to the tutorial in folder “Lesson 1 Sensor Assembly” under the same directory.

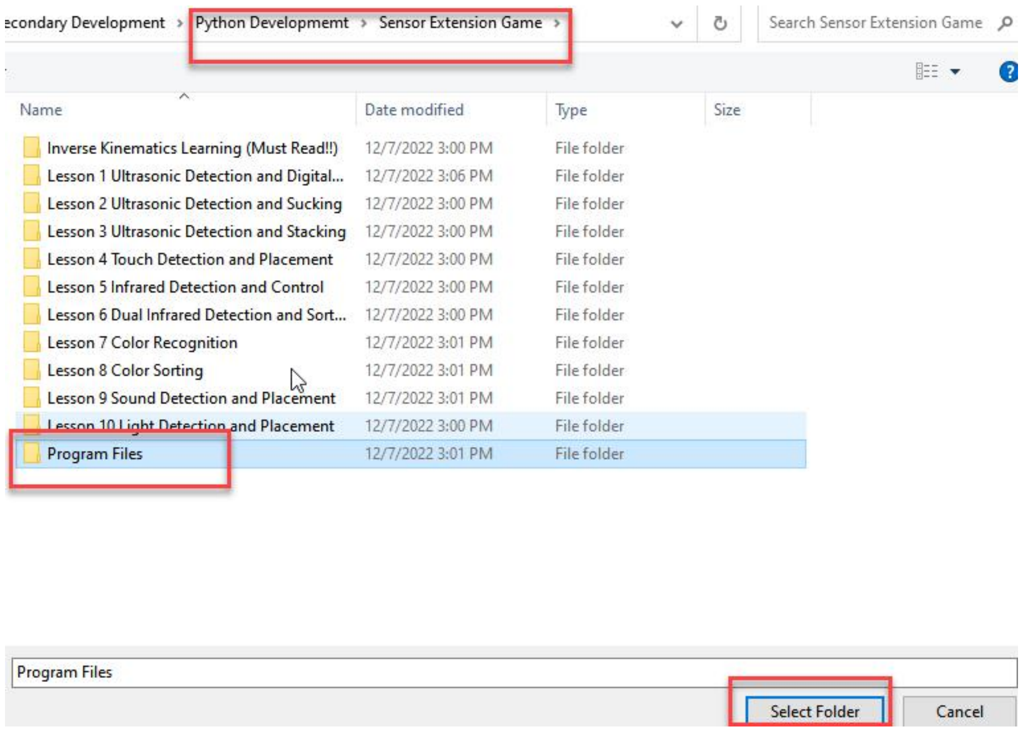
2.2 Software

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

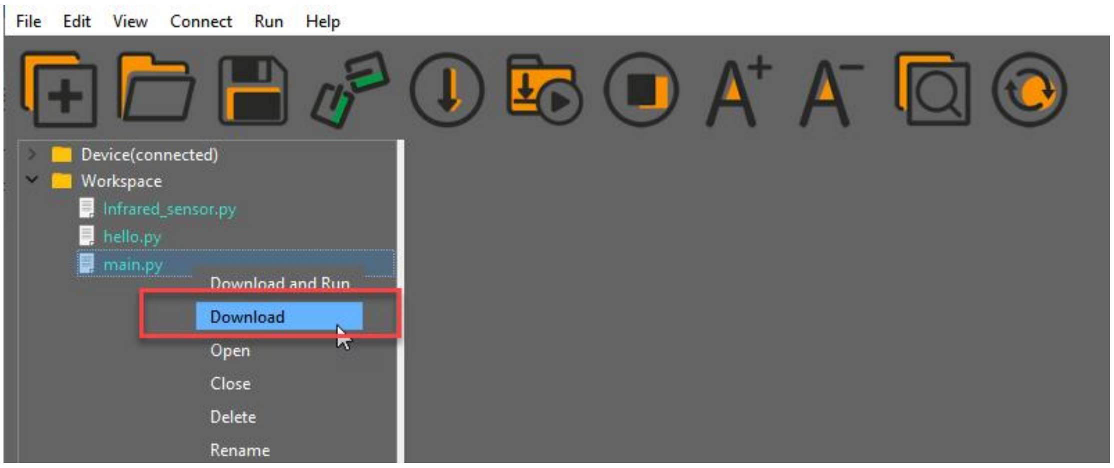
After connecting, change the path of Workspace to “6. Secondary Development/ Python Development/Sensor-extension Game”, and select “Program Files”.

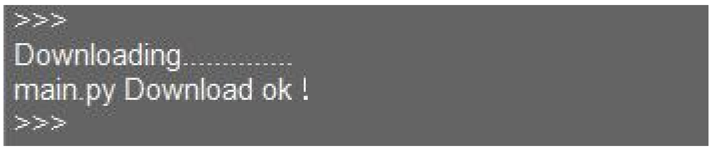



econdary Development › Python Developmemt › Sensor Extension Game ›
Search Sensor Extension Game

Name	Date modified	Type	Size
Inverse Kinematics Learning (Must Read!!)	12/7/2022 3:00 PM	File folder	
Lesson 1 Ultrasonic Detection and Digital...	12/7/2022 3:06 PM	File folder	
Lesson 2 Ultrasonic Detection and Sucking	12/7/2022 3:00 PM	File folder	
Lesson 3 Ultrasonic Detection and Stacking	12/7/2022 3:00 PM	File folder	
Lesson 4 Touch Detection and Placement	12/7/2022 3:00 PM	File folder	
Lesson 5 Infrared Detection and Control	12/7/2022 3:00 PM	File folder	
Lesson 6 Dual Infrared Detection and Sort...	12/7/2022 3:00 PM	File folder	
Lesson 7 Color Recognition	12/7/2022 3:01 PM	File folder	
Lesson 8 Color Sorting	12/7/2022 3:01 PM	File folder	
Lesson 9 Sound Detection and Placement	12/7/2022 3:01 PM	File folder	
Lesson 10 Light Detection and Placement	12/7/2022 3:00 PM	File folder	
Program Files	12/7/2022 3:01 PM	File folder	

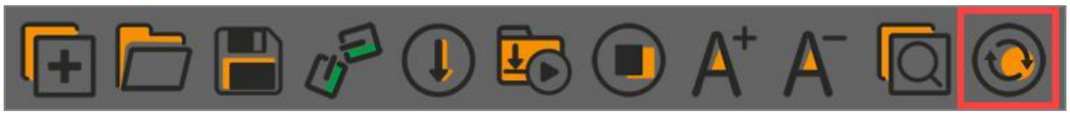
1) Click the folder “Dual Infrared Detection and Sorting”, and then select all the program files in the folder.Then right click to download all the program files to the controller.

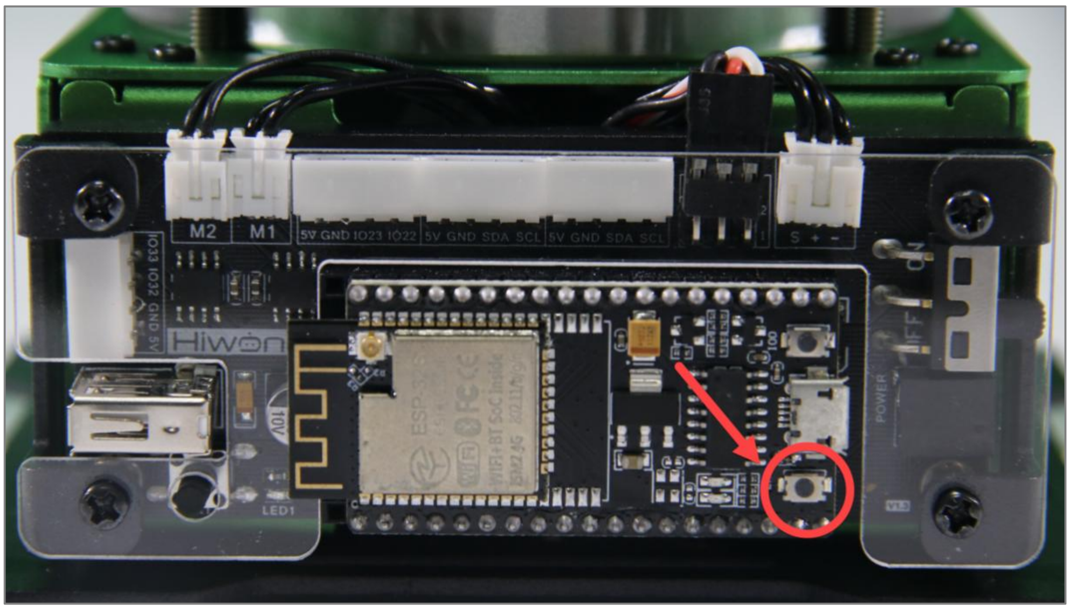




```
>>>
Downloading.............
main.py Download ok !
>>>
```

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





4. Project Outcome

Place a block in front of the right side of robot arm. When the block is detected by sensor, the buzzer will sound for a while and robotic arm will suck and place the block in the right side.

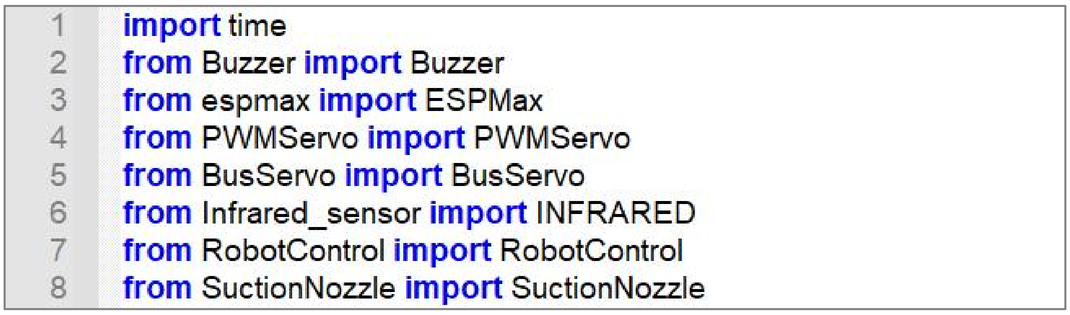
5. Program Instruction

The path of program file: “6. Secondary Development/ Python Development/ Program Files/ Dual Infrared Sorting/main.py”

5.1 Initialization Configuration

◆ Import function library

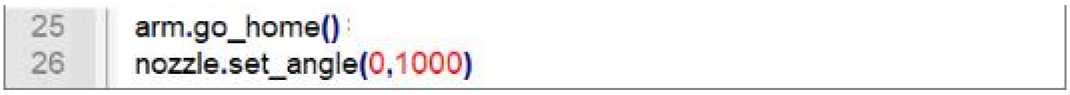
Firstly, import buzzer, kinematics, kinematics encapsulation library, PWM servo, suction nozzle and other related library files.



```
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
```

◆ Reset

Call the related function to reset robotic arm and suction nozzle.



```
arm.go_home()
nozzle.set_angle(0,1000)
```

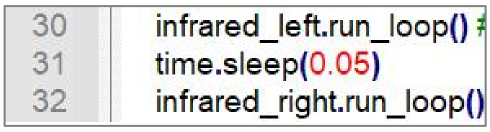
The nozzle.set_angle() function is used to control the rotation angle of suction nozzle, that is, control the corresponding PWM servo. The meaning of parameters in parentheses are as follow:

The first parameter “0” refers to the rotation angle of PWM servo.

The second parameter “1000” is the running time, that is, the time for the servo to rotate the specified angle, and the unit is milliseconds (ms).

5.2 Infrared Detection

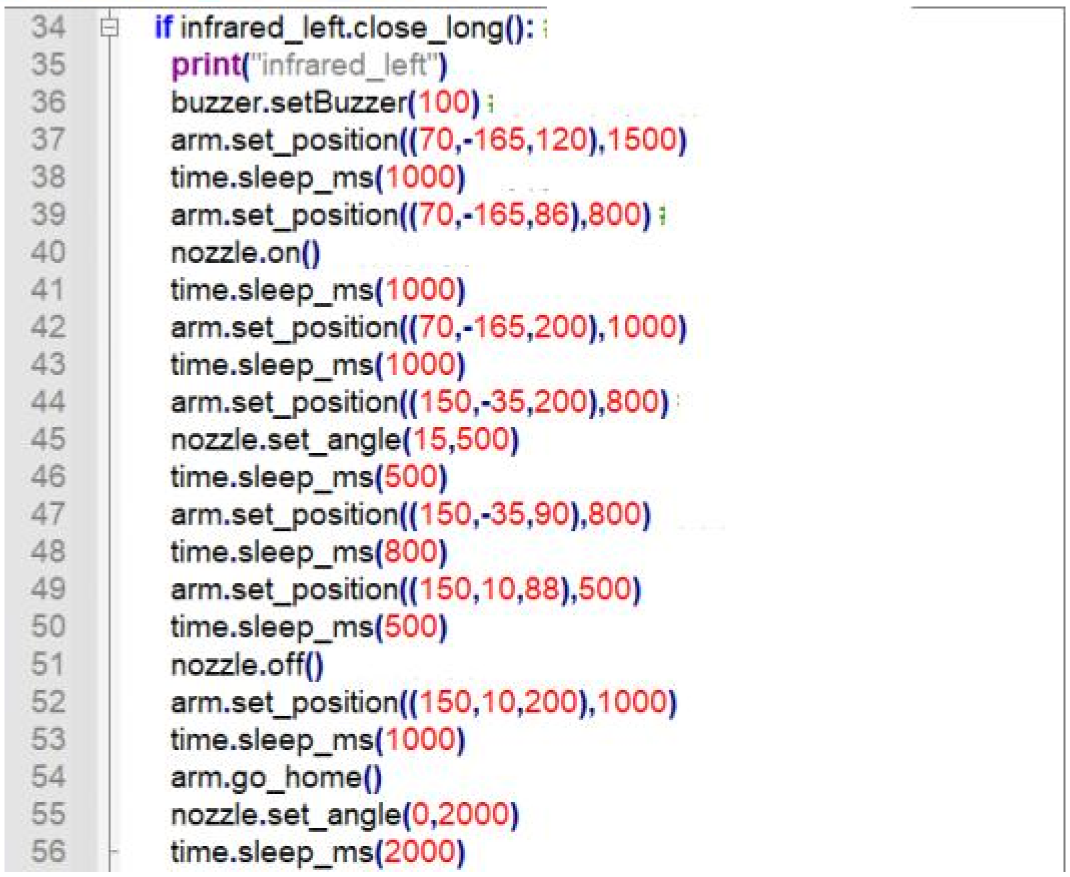
Call infrared sensor to detect object. The infrared_left.run_loop() function is the detection function corresponding to the left sensor, and the infrared_right.run_loop() function is the detection function corresponding to the right sensor.



```
infrared_left.run_loop() #
time.sleep(0.05)
infrared_right.run_loop()
```

5.3 Robotic Arm Control

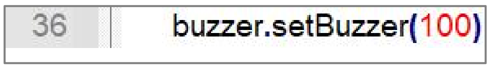
Take the left infrared sensor for example to analyze its source code. The realization principle is the same with the right infrared sensor.



```
if infrared_left.close_long():
    print("infrared_left")
    buzzer.setBuzzer(100)
    arm.set_position((70,-165,120),1500)
    time.sleep_ms(1000)
    arm.set_position((70,-165,86),800)
    nozzle.on()
    time.sleep_ms(1000)
    arm.set_position((70,-165,200),1000)
    time.sleep_ms(1000)
    arm.set_position((150,-35,200),800)
    nozzle.set_angle(15,500)
    time.sleep_ms(500)
    arm.set_position((150,-35,90),800)
    time.sleep_ms(800)
    arm.set_position((150,10,88),500)
    time.sleep_ms(500)
    nozzle.off()
    arm.set_position((150,10,200),1000)
    time.sleep_ms(1000)
    arm.go_home()
    nozzle.set_angle(0,2000)
    time.sleep_ms(2000)
```

◆ Buzzer Feedback

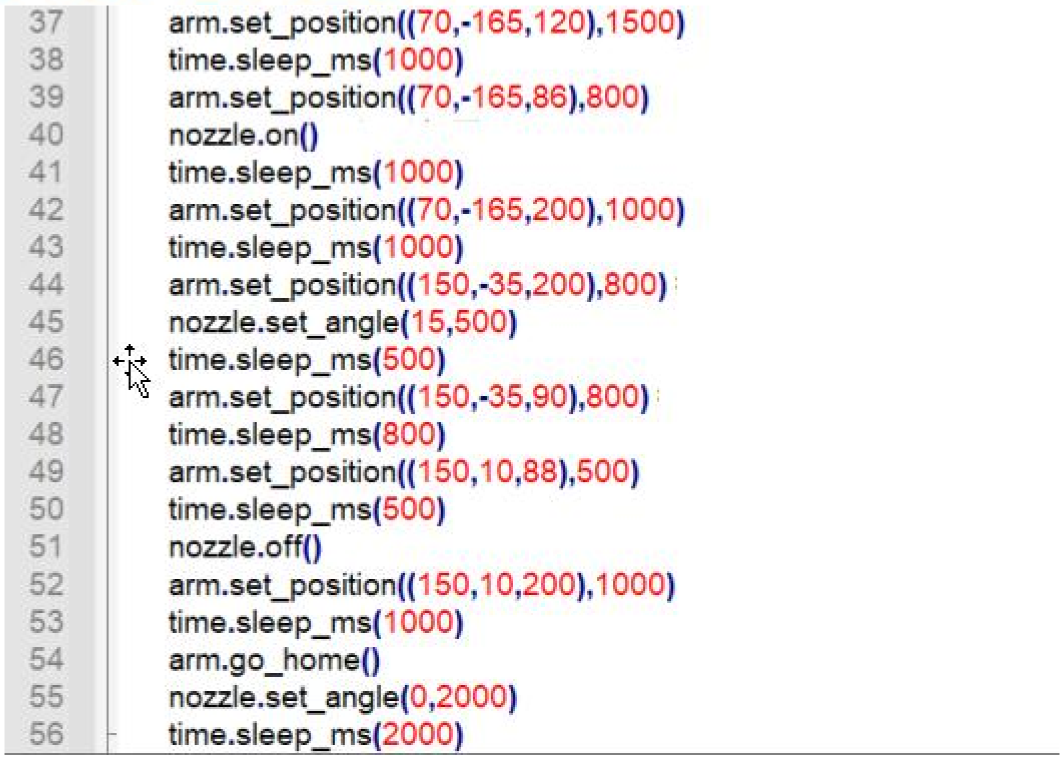
When a block is detected by infrared sensor, the buzzer alarms by calling buzzer.setBuzzer() function. The parameter in parenthesis refers to the duration of alarming and the unit is ms.



```
buzzer.setBuzzer(100)
```

◆ Suck & Place Block

By calling arm.set_position() function, robotic arm is controlled to rotate to the specific position. Combing this function and air pump control function to execute sorting function.



```
        arm.set_position((70,-165,120),1500)
        time.sleep_ms(1000)
        arm.set_position((70,-165,86),800)
        nozzle.on()
        time.sleep_ms(1000)
        arm.set_position((70,-165,200),1000)
        time.sleep_ms(1000)
        arm.set_position((150,-35,200),800)
        nozzle.set_angle(15,500)
        time.sleep_ms(500)
        arm.set_position((150,-35,90),800)
        time.sleep_ms(800)
        arm.set_position((150,10,88),500)
        time.sleep_ms(500)
        nozzle.off()
        arm.set_position((150,10,200),1000)
        time.sleep_ms(1000)
        arm.go_home()
        nozzle.set_angle(0,2000)
        time.sleep_ms(2000)
```

Take the code “arm.set_position((70,-165,120),1500)” for example. The meaning of the parameters in parenthesis is as follow:

The first parameter “(70,-165,120)” represents the target position, i.e., the target coordinate position of end effector. Three parameters are the value of x,y and z axes.

The second parameter “1500” is the running time, e.i., the time for the nozzle to move to the target position and the unit is ms.