Lesson 2 Touch Detection and Placement

1. Working Principle

This lesson uses touch sensor based on the principle of capacitive sensing. After supplying the touch sensor power, the touching will be sensed when our fingers or metal touch the metal sensing plate. In the meantime, the signal terminal OUT will output low level signal, vice verse. According to this characteristics, the robotic arm can be controlled to perform the corresponding action.

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

```
import time
from Buzzer import Buzzer
from espmax import ESPMax
from PWMServo import PWMServo
from BusServo import BusServo
from Touch_sensor import TOUCH
from RobotControl import RobotControl
from SuctionNozzle import SuctionNozzle


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

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

Next, create the functions of buzzer and touch control. Set the buzzer to make sound feedback and the robotic arm to perform the corresponding action when short press the touch sensor.

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

detected object to the side and place it.

2. Preparation

2.1 Hardware

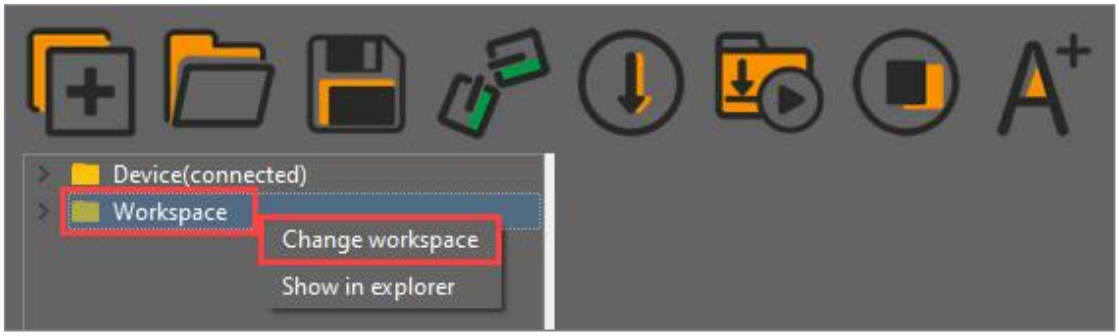
Please assemble the touch sensor to the corresponding position on MaxArm according to the tutorial in folder “Lesson 1 Sensor Assembly” under the same directory.

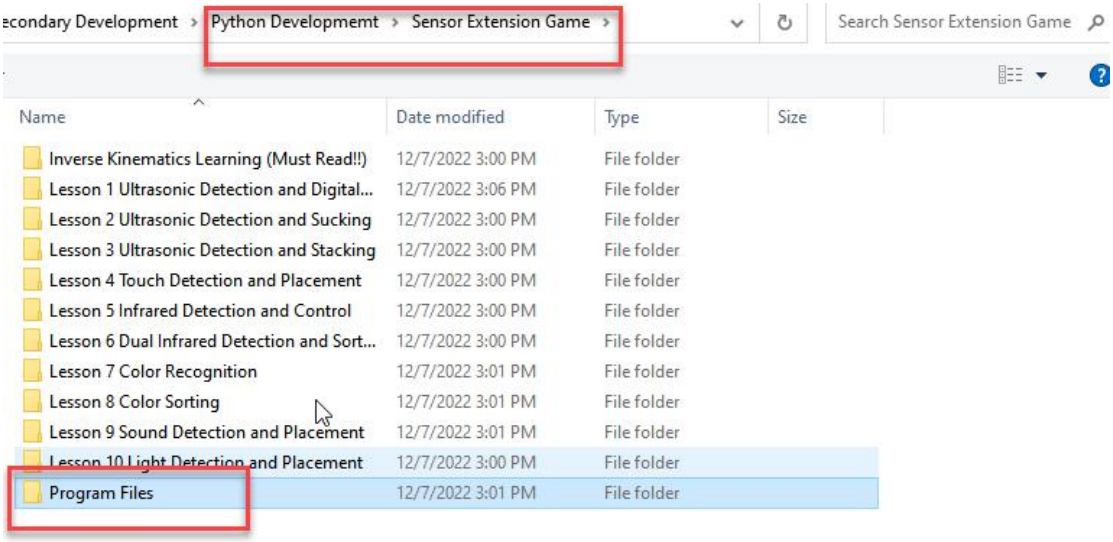
2.2 Software

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

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





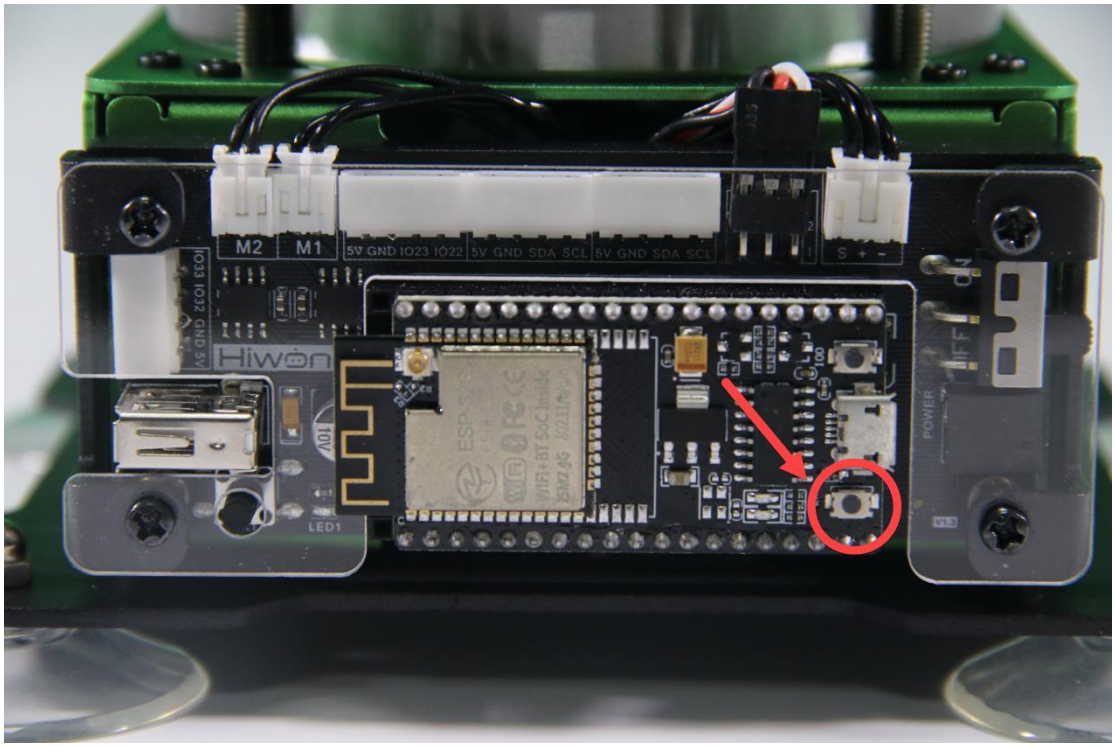
2) Click the folder “Touch Detection and Placement”, and then select all the program files in the folder.

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.

```
>>>
Downloading files
Start downloading main.py.....
main.py Download ok !
>>>
Start downloading Touch_sensor.py.....
Touch_sensor.py Download ok !
>>>
```

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 beeps once, the robotic arm will suck and place the blocks in a line. When three blocks are placed completely, the buzzer will make “DiDi” sound as a feedback, and then a round of placement is finished.

5. Program Instruction

5.1 Import Function library

The path to the program of the source code is “6. Secondary Development Application/Python Development/Sensor Related Game/Game Programs/Touch Detection and Placement/main.py”.

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

```
import time
from Buzzer import Buzzer
from espmax import ESPMax
from PWMServo import PWMServo
from BusServo import BusServo
from Touch_sensor import TOUCH
from RobotControl import RobotControl
from SuctionNozzle import SuctionNozzle
```

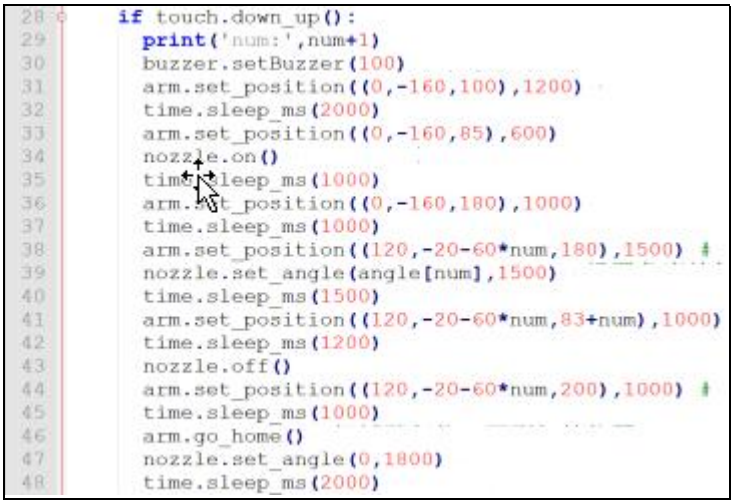
5.2 Touch detection

Use the touch.run_loop() function to detect whether the touch is sensed.

```
if touch.down_up():
    print('num:',num+1)
    buzzer.setBuzzer(100)
    arm.set_position((0,-160,100),1200)
    time.sleep_ms(2000)
    arm.set_position((0,-160,85),600)
    nozzle.on()
    time.sleep_ms(1000)
    arm.set_position((0,-160,180),1000)
    time.sleep_ms(1000)
    arm.set_position((120,-20-60*num,180),1500) #
    nozzle.set_angle(angle[num],1500)
    time.sleep_ms(1500)
    arm.set_position((120,-20-60*num,83+num),1000)
    time.sleep_ms(1200)
    nozzle.off()
    arm.set_position((120,-20-60*num,200),1000) #
    time.sleep_ms(1000)
    arm.go_home()
    nozzle.set_angle(0,1800)
    time.sleep_ms(2000)
```

5.3 Control robotic arm

After sensing the touch, MaxArm will execute the corresponding action.



```
if touch.down_up():
    print('num:',num+1)
    buzzer.setBuzzer(100)
    arm.set_position((0,-160,100),1200)
    time.sleep_ms(2000)
    arm.set_position((0,-160,85),600)
    nozzle.on()
    time.sleep_ms(1000)
    arm.set_position((0,-160,180),1000)
    time.sleep_ms(1000)
    arm.set_position((120,-20-60*num,180),1500) #
    nozzle.set_angle(angle[num],1500)
    time.sleep_ms(1500)
    arm.set_position((120,-20-60*num,83+num),1000)
    time.sleep_ms(1200)
    nozzle.off()
    arm.set_position((120,-20-60*num,200),1000) #
    time.sleep_ms(1000)
    arm.go_home()
    nozzle.set_angle(0,1800)
    time.sleep_ms(2000)
```

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

The first parameter “100” represents the sounding time 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),1200)” as example.

The first parameter “(0，-160，100)” is the position of the suction nozzle on x, y and z axes.

The second parameter “1200” 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(0,1800)” as example.

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

The second parameter “1800” represents the running time and the unit is ms.