Lesson 6 Drawing Square

1. Preparation

1.1 Software

Please connect MaxArm to Python editor according to the tutorial in folder

“4. Underlying Program Learning/Python Development/Lesson 1 Set Development Environment”.

2. Working Principle

Firstly, import the kinematics encapsulation library “espmax” and instantiate the library function.

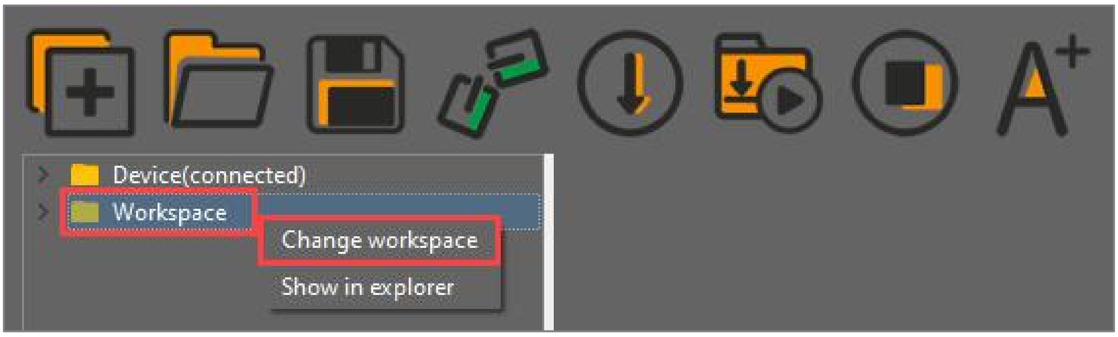
The path to the kinematics encapsulation library is “Appendix/8. Underlying File/Python Development/espmax.py”

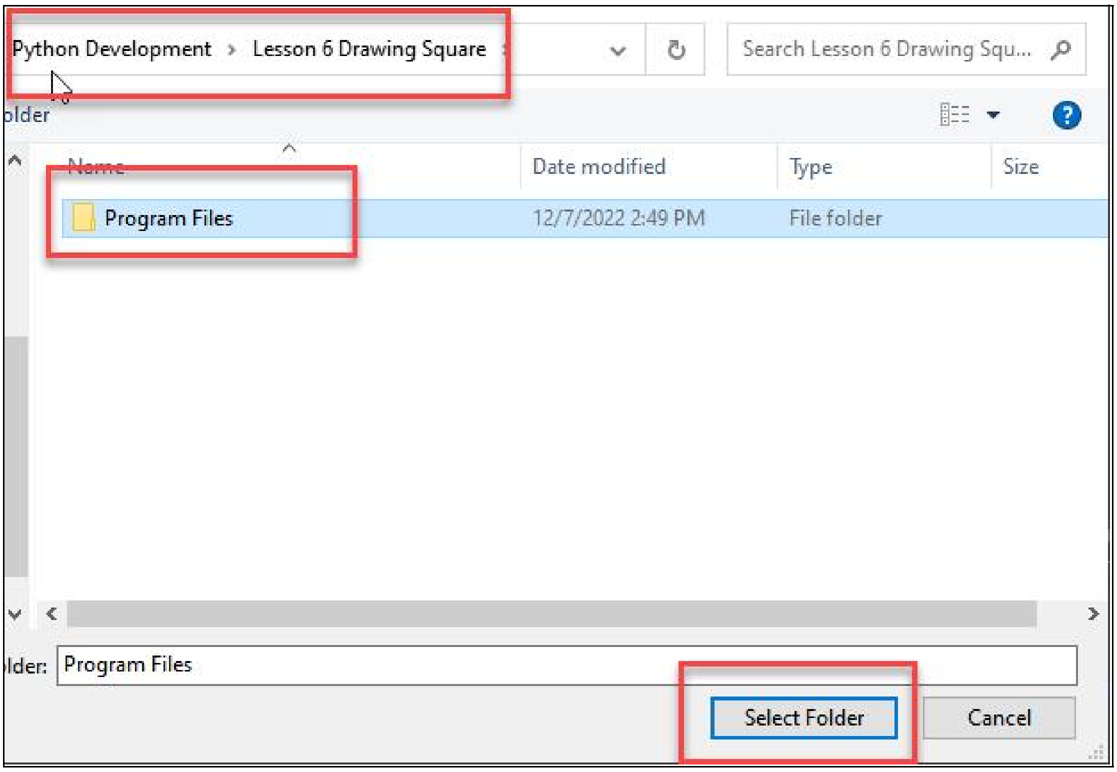
Then, specify the initial position of the robot arm and do the path planning for drawing “square”.

Finally, call the arm.set_position() function. According to the position of xyz axes in space and kinematics, get the servo angle and complete the path planning for drawing “square”.

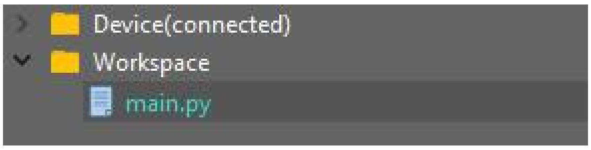
3. Program Download

1) After connecting, change the path of Workspace to “8. Inverse Kinematics Basic and Application/Python Development/Lesson 5 Drawing Square/Program File”.

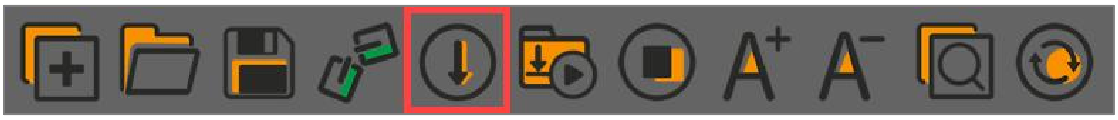




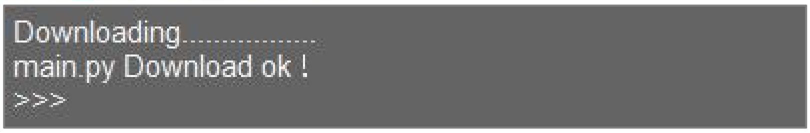
2) Double click “Workshop” folder, and then double click “main.py” to open the program.



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

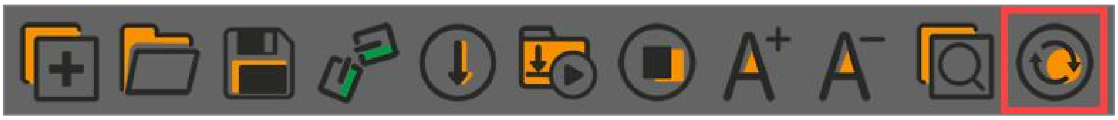


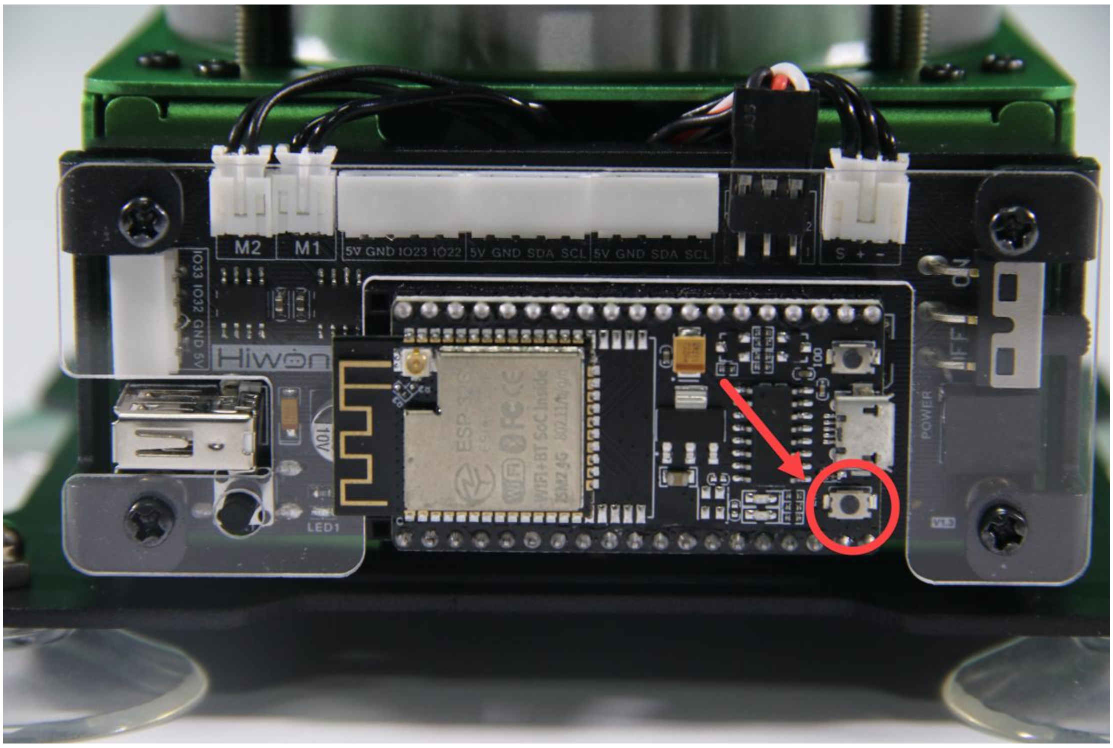
4) When the terminal prints the prompt as shown in the image below, it means download completed.



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

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





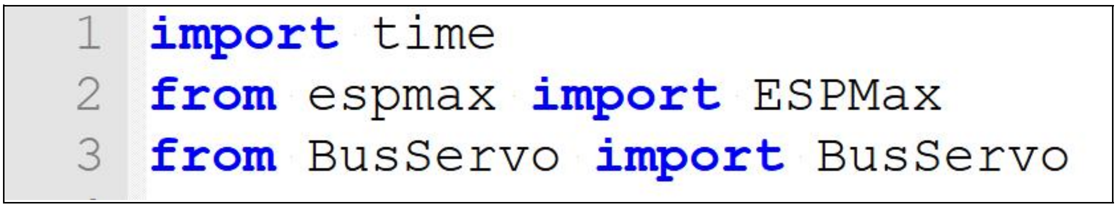
4. Project Outcome

When running program, the robotic arm will be controlled to draw “square”. After stopping, exit the program automatically.

5. Program Instruction

5.1 Import function file

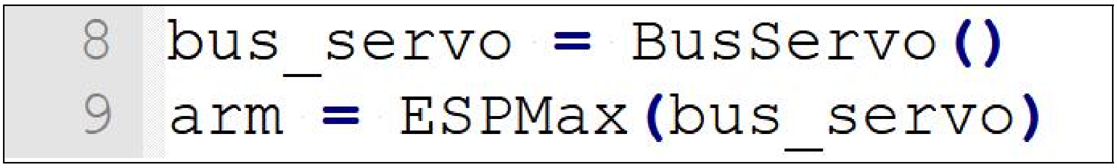
Before the robotic arm starts to move, the function libraries espmax and BusSerb need to be imported.



```
import time
from espmax import ESPMax
from BusServo import BusServo
```

5.2 Instantiate library function

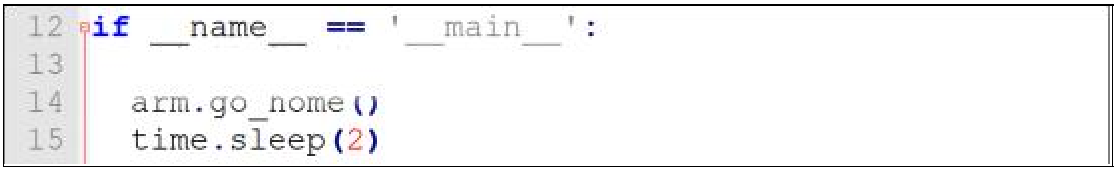
Instantiate the functions importing to the function library for later debugging.



```
bus_servo = BusServo()
arm = ESPMax(bus_servo)
```

5.3 Return to the initial position

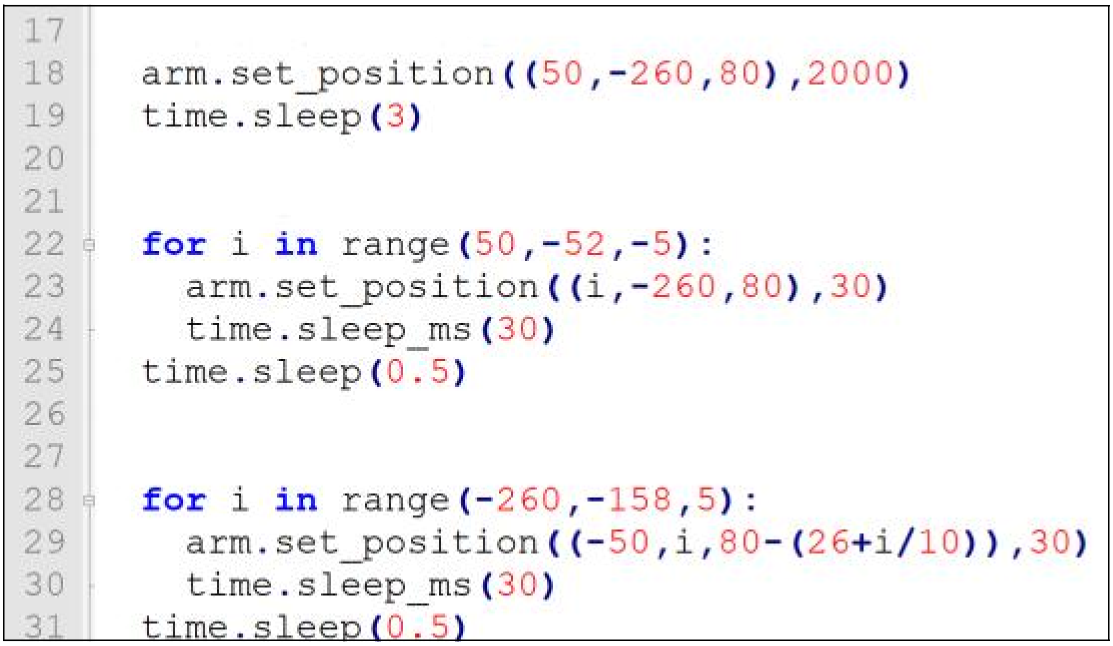
Firstly, use go_home() function in function library to control the robotic arm to move to the initial position. The specific program is as follow:



```
if __name__ == '__main__':

    arm.go_home()
    time.sleep(2)
```

5.4 Control Robotic Arm



```

arm.set_position((50,-260,80),2000)
time.sleep(3)


for i in range(50,-52,-5):
  arm.set_position((i,-260,80),30)
  time.sleep_ms(30)
time.sleep(0.5)


for i in range(-260,-158,5):
  arm.set_position((-50,i,80-(26+i/10)),30)
  time.sleep_ms(30)
time.sleep(0.5)
```

Use arm.set_position() function to control robotic arm to draw cross in space. Take the code “arm.set_position((50,-260,80),2000)” as example.

The first parameter “(50,-260,80)” is the position of the suction nozzle on xyz axes.

The second parameter “2000” is the running time and the unit is ms.