

# Lesson 15 Serial Communication

**Note:** please prepare your own USB adapter and the female-to-female dupont line.

## 1. Project Overview

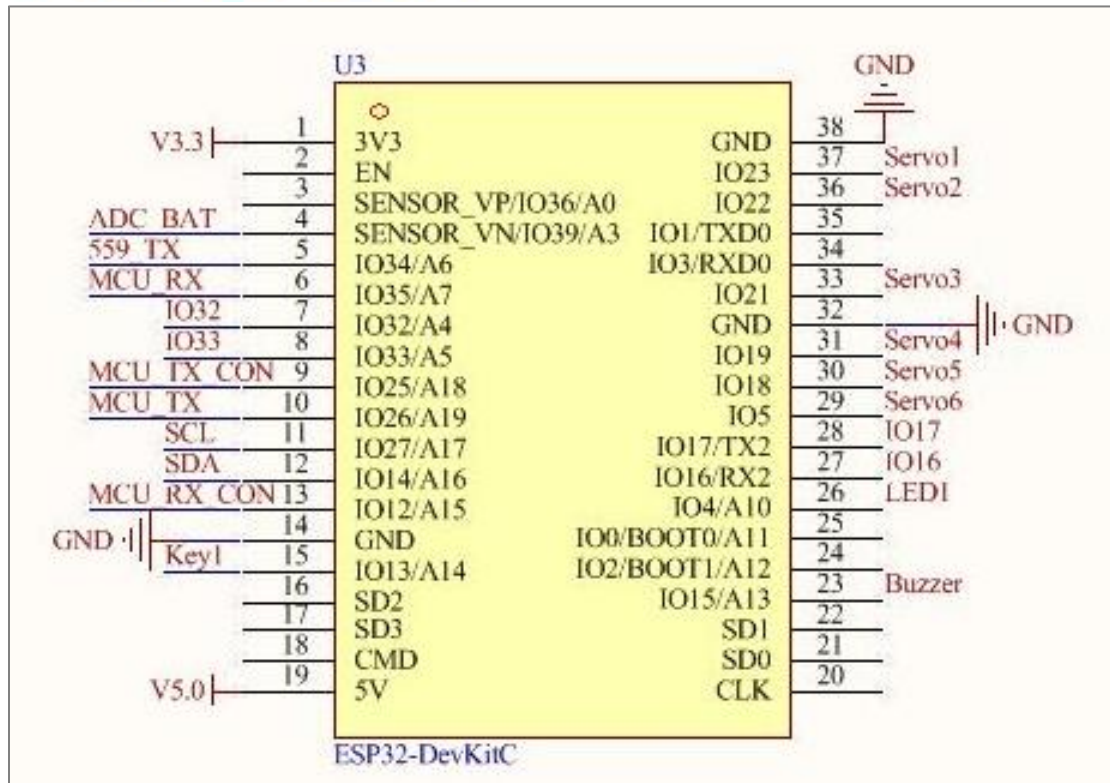
The serial port is a way to communicate. The data can be transmitted by serial communication between MCU and the print debugging can be realized by serial communication between MCU and computer.

## 2. Working Principle

The path to the source code of program is “Appendix/5.Hardware Basic Learning/ Python Development/Program Files/ESP32/UART/main.py”

```
1  from machine import UART
2  import time
3  # create message
4  sendbuffer = "Hiwonder\r\n"
5  # The serial port 0 is occupied by repel.
6  uart = UART(2,115200, tx=32, rx=33)
7  # Install your own USB to serial port module. The TX of module is connected to RX of ESP32.
  # The RX of module is connected to TX of ESP32. The pin of serial port2 is tx=32, rx=33 .
8  # If need to modify the parameters of the serial ports, please refer to the following parameter.
9  # UART(2, baudrate=115201, bits=8, parity=None, stop=1, tx=32, rx=33,
10 # rts=-1, cts=-1, txbuf=256, rxbuf=256, timeout=0, timeout_char=1)
11 # Print the parameters of the serial port
12 print(uart)
13 # In the infinite loop, a Hiwonder is output every 1000ms, and the data received by the serial port
  # is printed out in the form of bytes.
14 while True:
15     uart.write(sendbuffer)
16     readbuffer = uart.readline()
17     if readbuffer is not None:
18         print(readbuffer)
19     time.sleep_ms(1000)
```

According to the following circuit diagram, you can learn about the pin information of buttons on ESP32 expansion board.



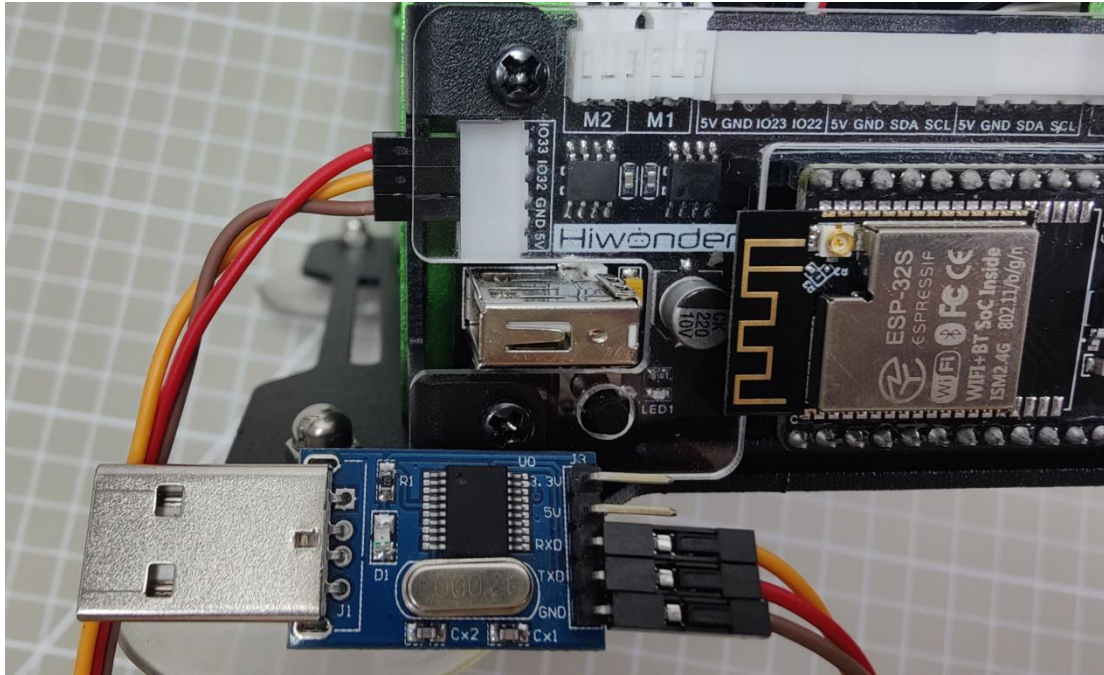
Among the pins, TxD is the output port of serial port, and RxD is the input port of serial port, i.e, the TxD of chip 1 should be connected to the RxD of chip 2, and the RxD of chip 1 should be connected to the TxD of chip 2, and both of them uses TTL communication according to the agreed data format.

### 3. Preparation

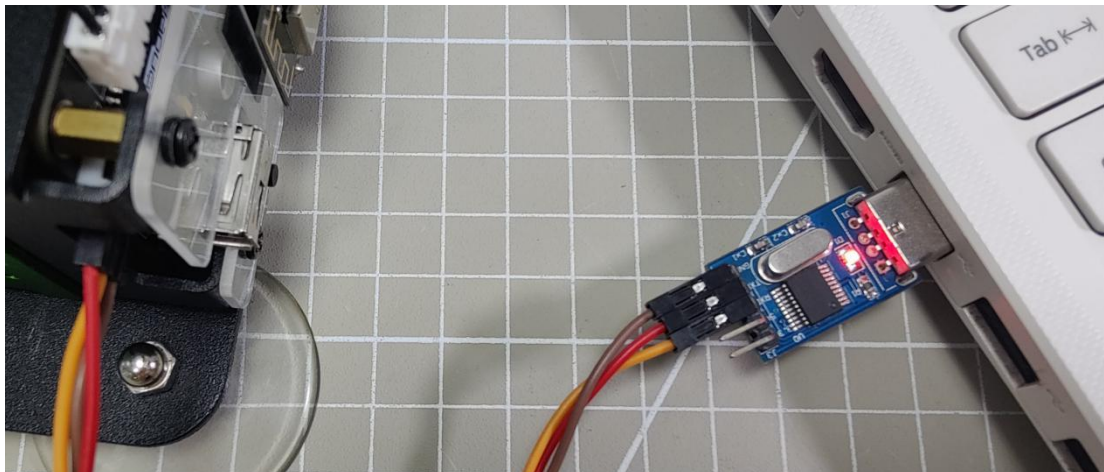
#### 3.1 Hardware

MaxArm robotic arm, power adapter , USB cable, USB adapter, four female-to-female dupond line.

- 1) Connect the RX, TX, and GND of USB adapter to IOS23, IO33, GND of ESP32 ports of expansion board correspondingly with dupond line.



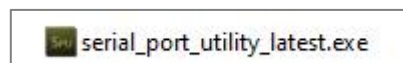
2) Connect the USB adapter to your computer.



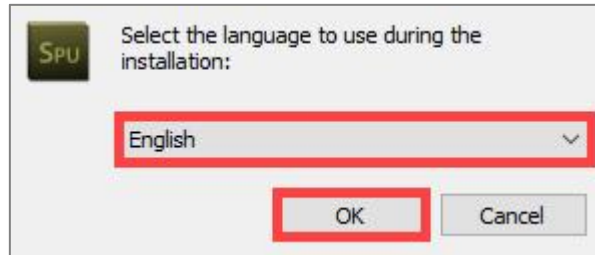
## 3.2 Software

### 3.2.1 Install and Start the Serial Port Debug Tool

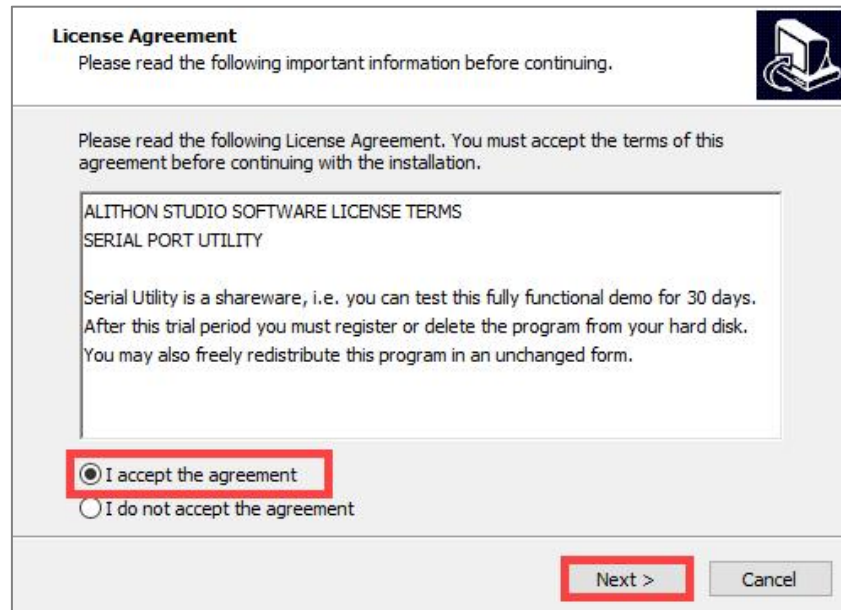
1) Double click the following executable file under the same directory.



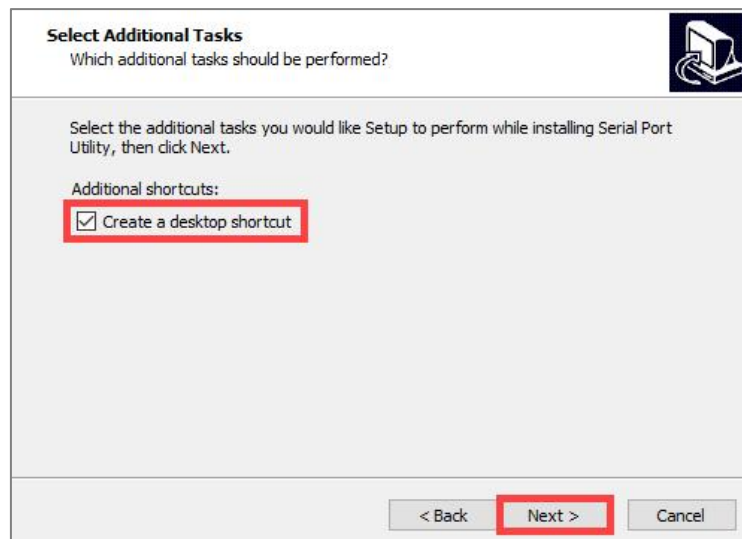
2) Select the installation language: English



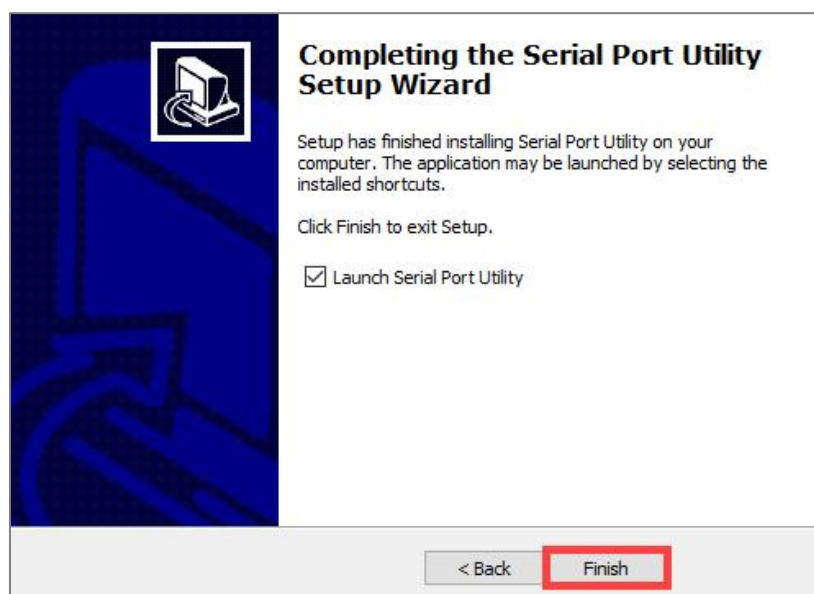
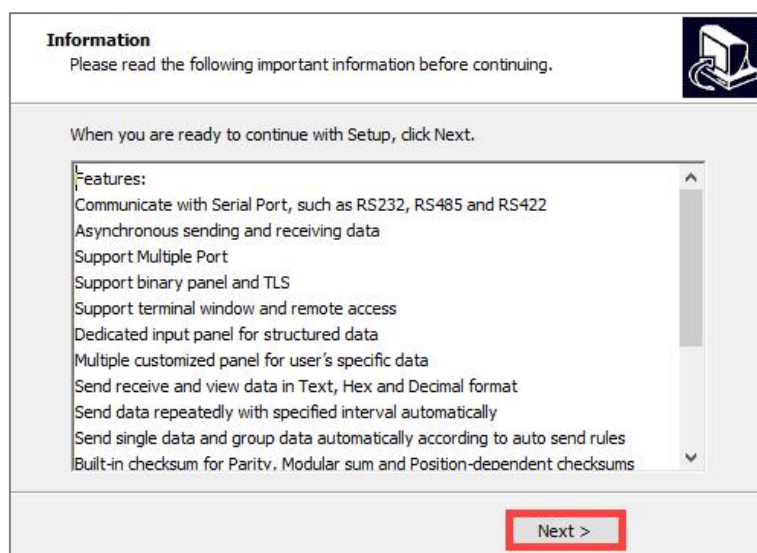
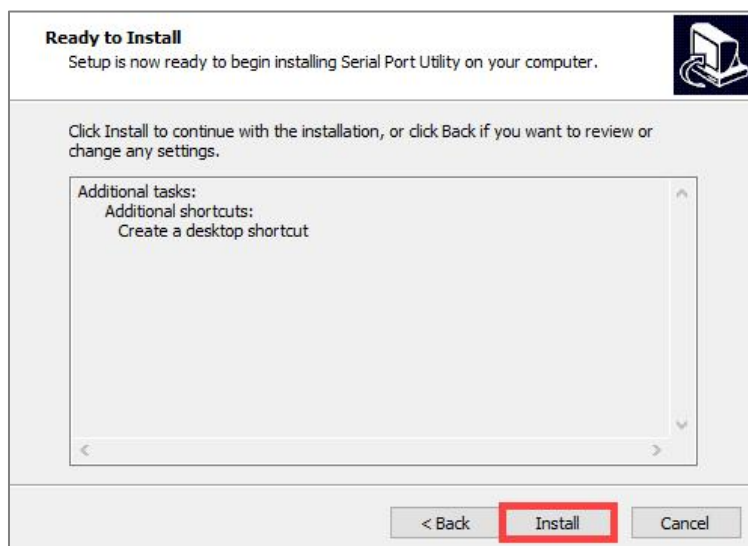
3) Select "I Agree", and click "Next".



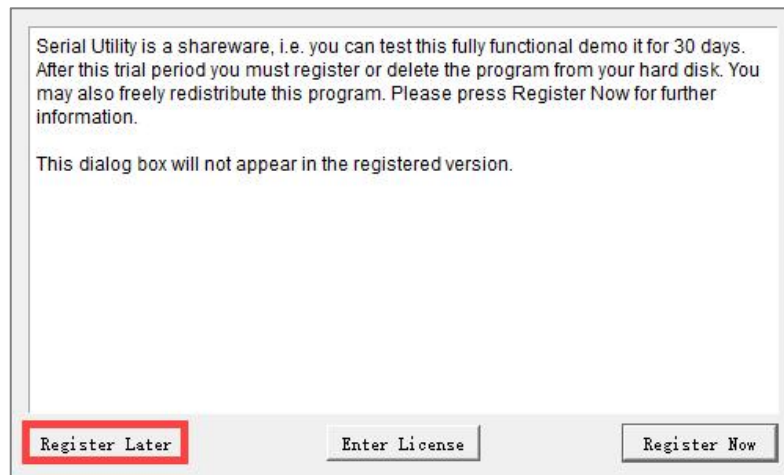
4) Select "Create a desktop shortcut".




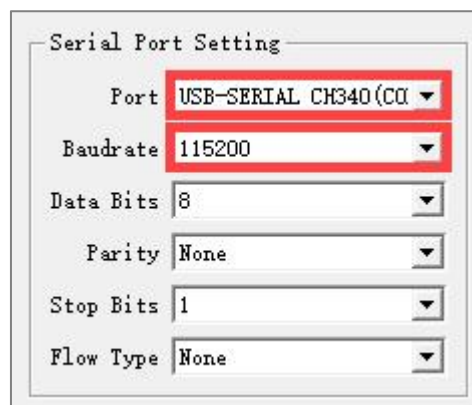




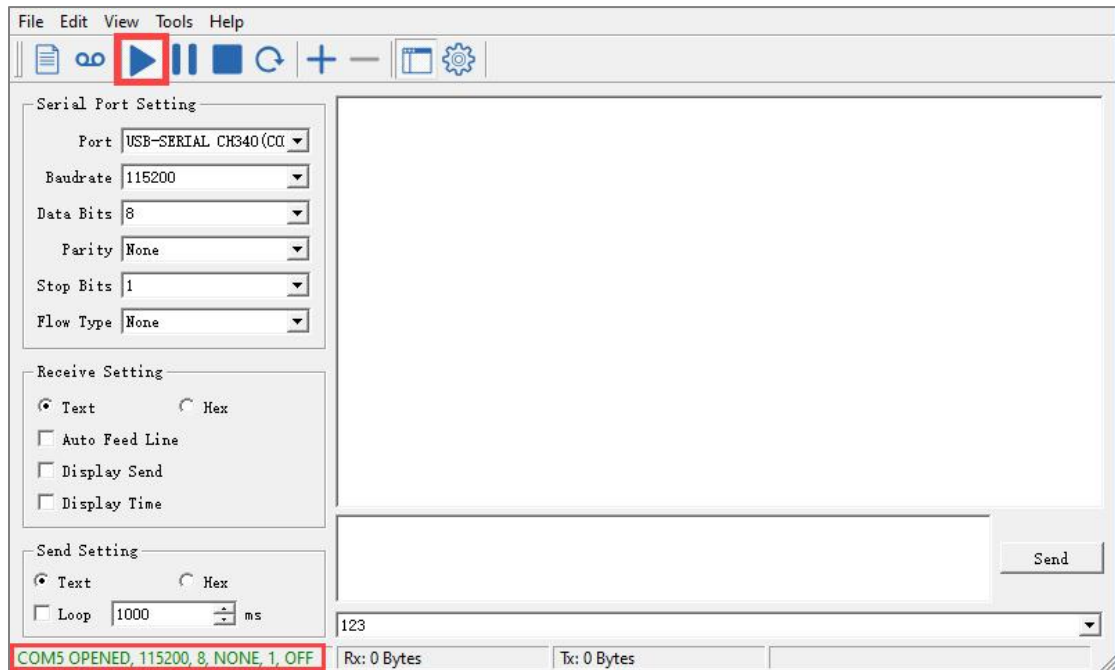
- 5) Click “Resister Later”.



- 6) After installing, click on  icon and select the COM port starting with “CH340”. And set the baud rate to 115200.



- 7) Click on the following button to start serial communication. At this time, the serial port number will be displayed in the lower left corner.



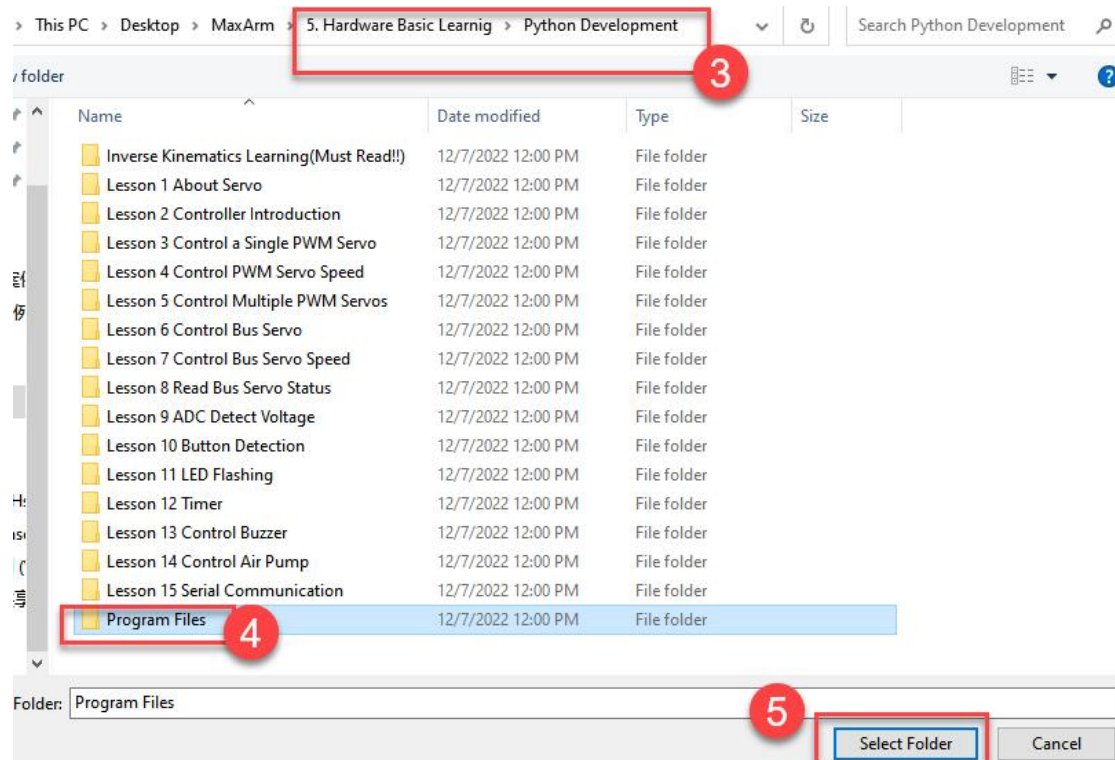
### 3.2.2 Connect Python Editor

Please refer to the material in folder “4.MaxArm Underlying Program Learning/Lesson 1 Set Development Environment” to connect ESP32 controller to Python Editor.

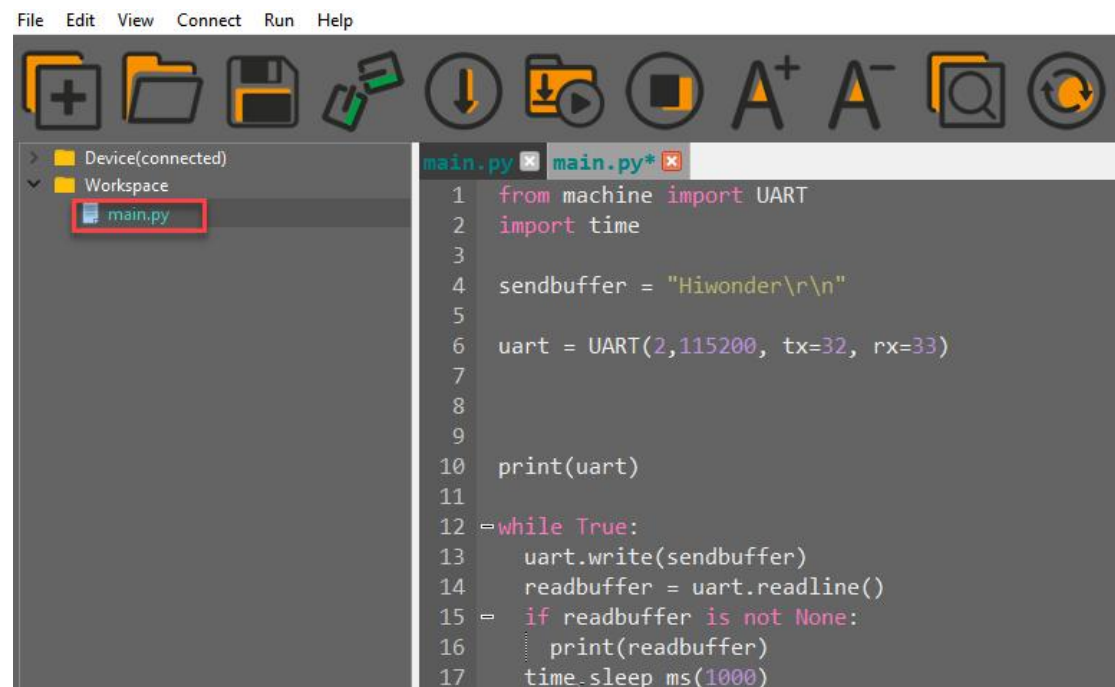
## 4. Program Download

- 1) After connecting, change the path of Workspace to “5.Hardware Basic Learning/Python Development” and select “Program Files”.





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



3) Click on the download icon to download program to ESP32 controller.

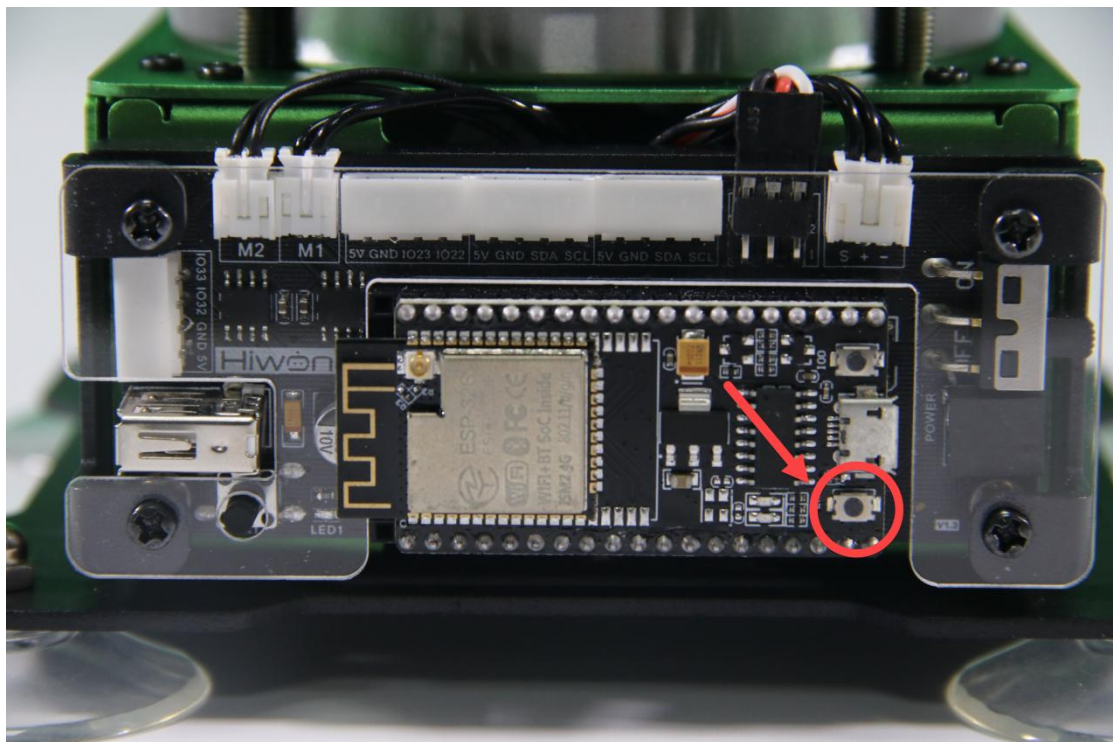




- 4) When the terminal prints the prompt, as shown in the image below, it means download completed.

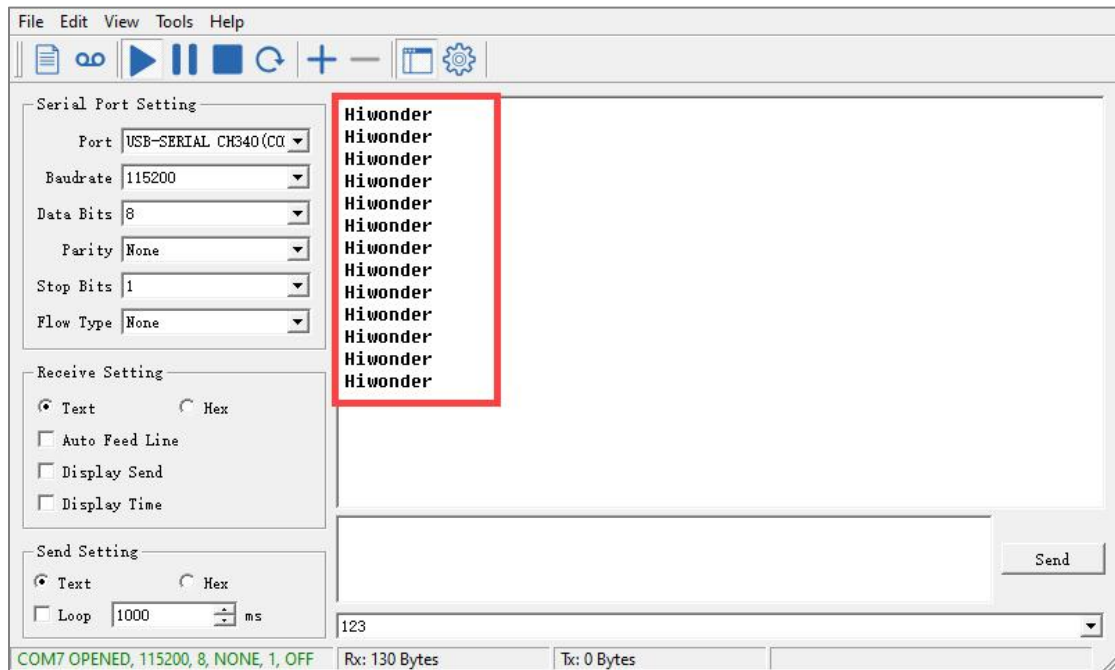
```
>>>
Downloading....
main.py Download ok!
>>>
```

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



## 5. Project Outcome

1) In the window of the serial port debug tool, “Hiwonder” sent by ESP32 controller can be viewed, as shown in the image below:



1) The tool can also send a paragraph of english or number characters to ESP32 controller. This section will send “123” as example.



2) At this time, the received characters can be viewed in the terminal of editor.

```
b'123'
```