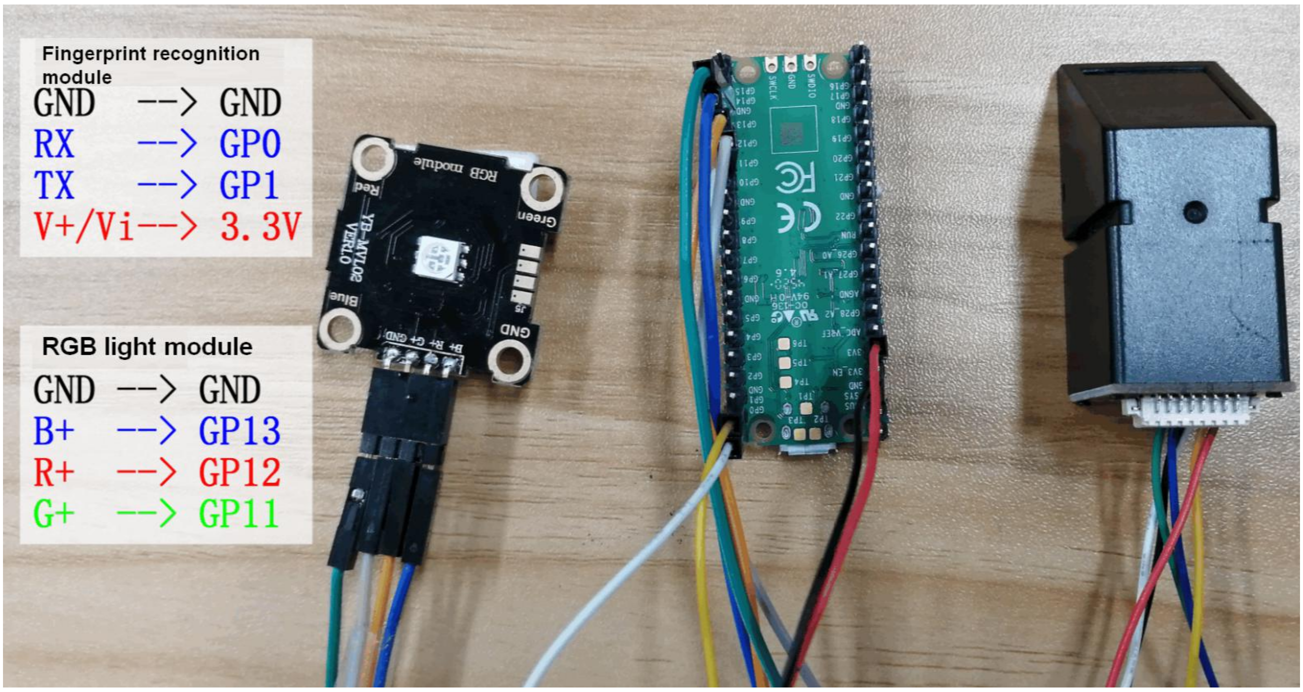
Fingerprint control light

1. Learning target

In this course, we will learn how to use Pico and fingerprint recognition module to achieve fingerprint delete function.

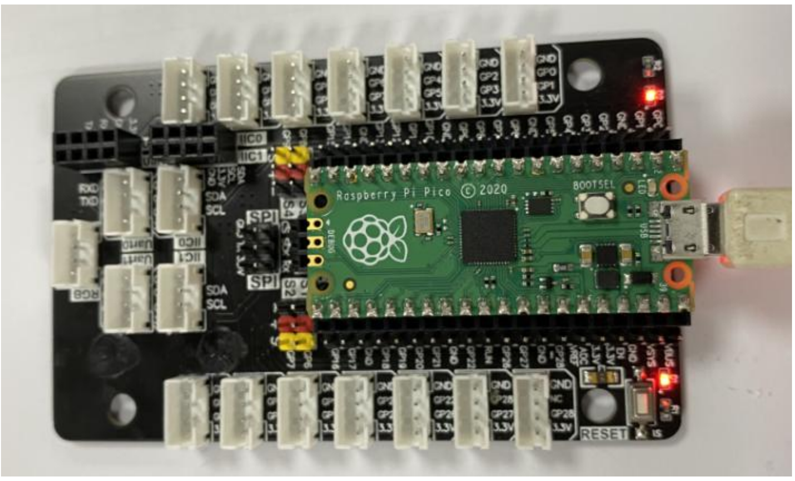
2. Preparation

The fingerprint recognition module uses UART communication, and the program uses a virtual serial port. Connect the TX and RX of the module to the GP1 and GP0 pins of the Pico board. V+/Vi and GND are connected to 3.3V and GND of Pico respectively.

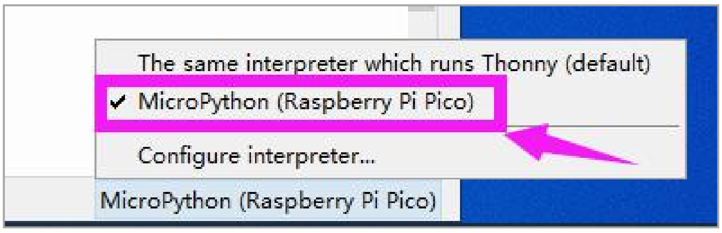


3. Import library file

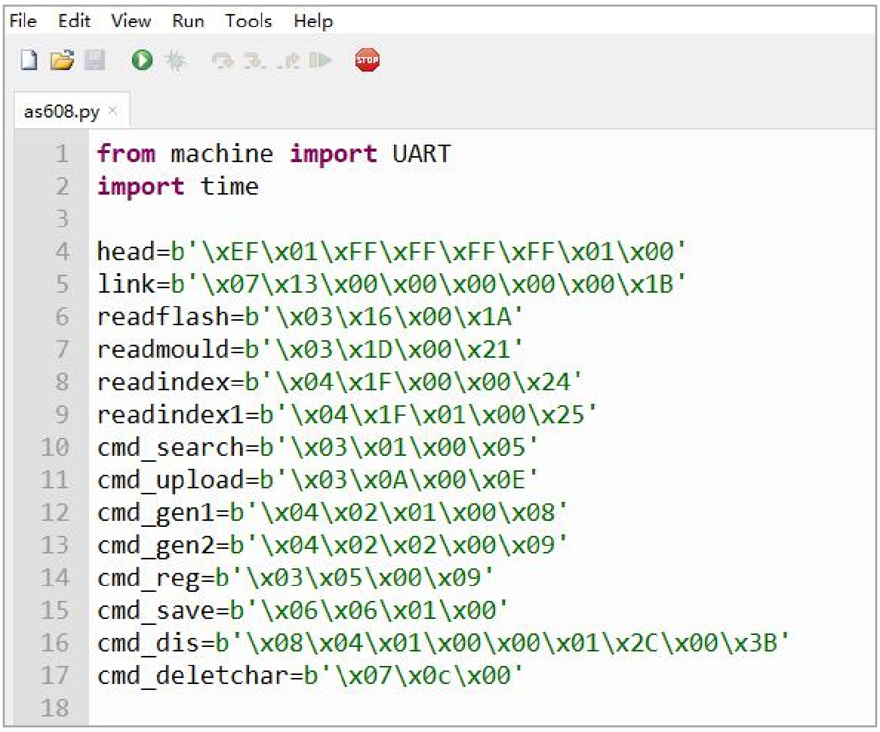
3.1 Connect Pico to your computer, as shown below.



3.2 Open the Thonny software, click the lower right corner to connect the Pico board.


The same interpreter which runs Thonny (default)
✔ MicroPython (Raspberry Pi Pico)
Configure interpreter...
MicroPython (Raspberry Pi Pico)

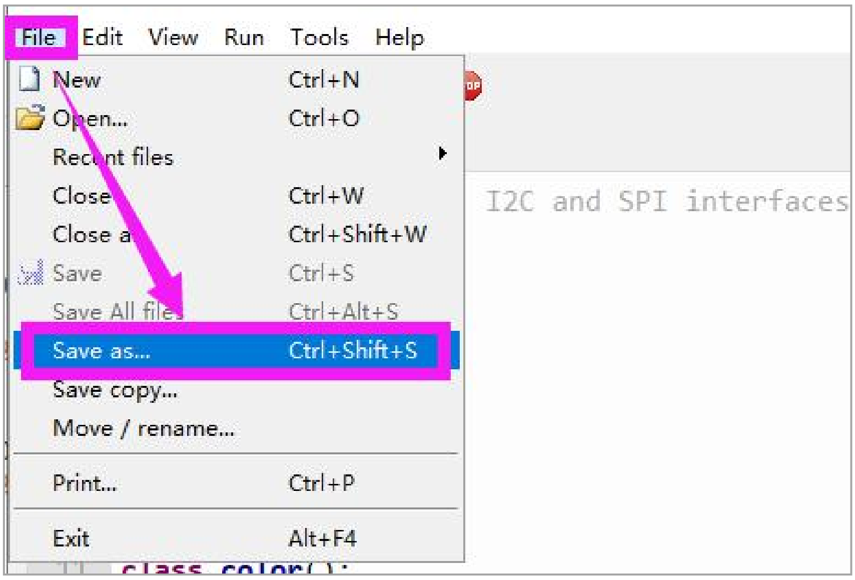
3.3 Open the as608.py in library folder by Thonny software.

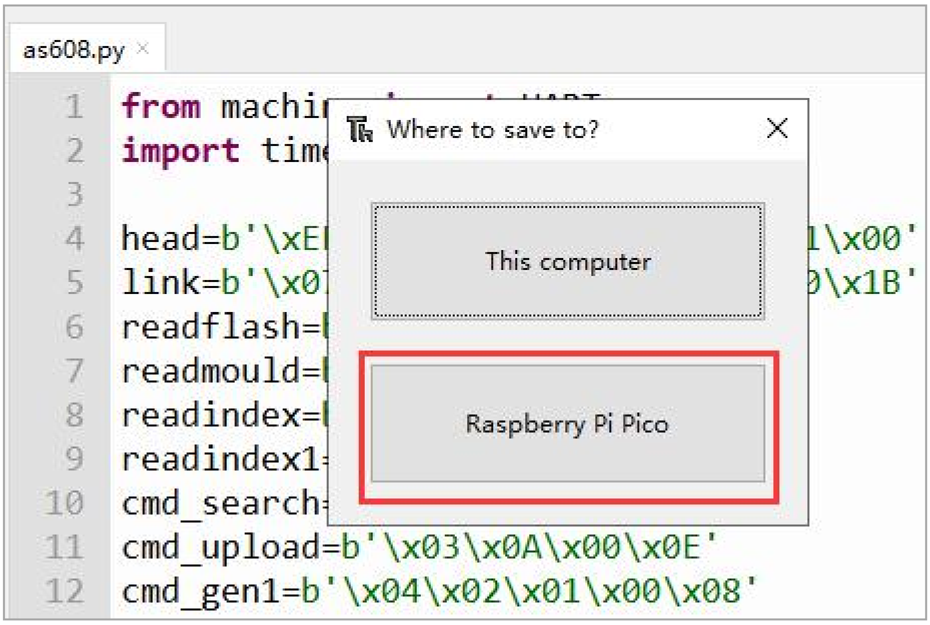


```
from machine import UART
import time

head=b'\xEF\x01\xFF\xFF\xFF\xFF\x01\x00'
link=b'\x07\x13\x00\x00\x00\x00\x00\x1B'
readflash=b'\x03\x16\x00\x1A'
readmould=b'\x03\x1D\x00\x21'
readindex=b'\x04\x1F\x00\x00\x24'
readindex1=b'\x04\x1F\x01\x00\x25'
cmd_search=b'\x03\x01\x00\x05'
cmd_upload=b'\x03\x0A\x00\x0E'
cmd_gen1=b'\x04\x02\x01\x00\x08'
cmd_gen2=b'\x04\x02\x02\x00\x09'
cmd_reg=b'\x03\x05\x00\x09'
cmd_save=b'\x06\x06\x01\x00'
cmd_dis=b'\x08\x04\x01\x00\x00\x01\x2C\x00\x3B'
cmd_deletchar=b'\x07\x0c\x00'
```

3.4 Save as this .py library file into Pico.

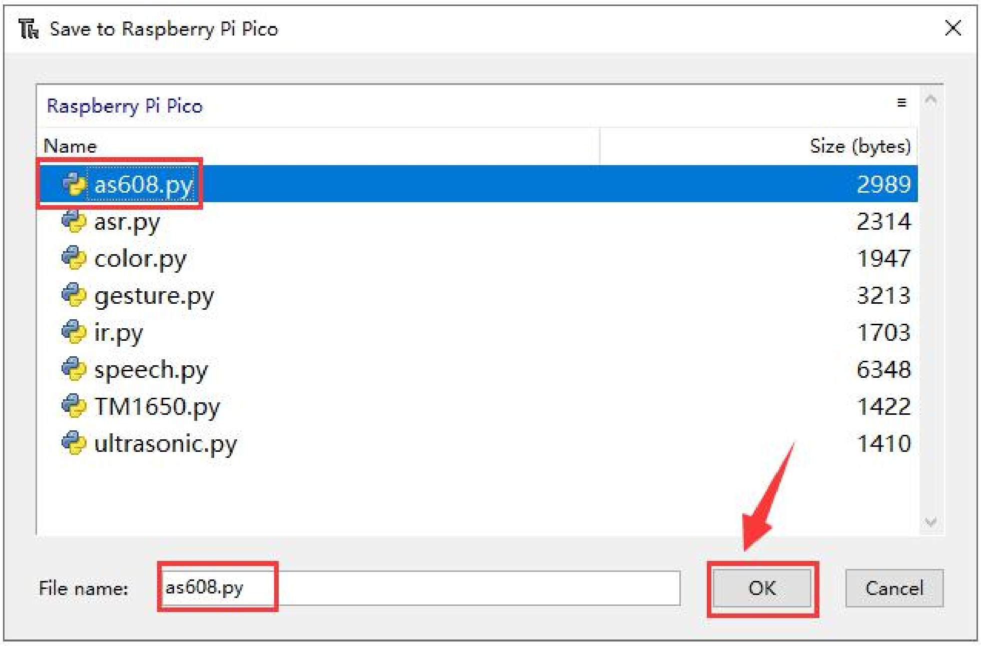




```
from machi
import tim

head=b'\xE                                    1\x00'
link=b'\x0                                    0\x1B'
readflash=
readmould=
readindex=
readindex1
cmd_search
cmd_upload=b'\x03\x0A\x00\x0E'
cmd_gen1=b'\x04\x02\x01\x00\x08'
```

3.5 Enter the same file name as the library file. Then, click “OK”.



Name	Size (bytes)
as608.py	2989
asr.py	2314
color.py	1947
gesture.py	3213
ir.py	1703
speech.py	6348
TM1650.py	1422
ultrasonic.py	1410

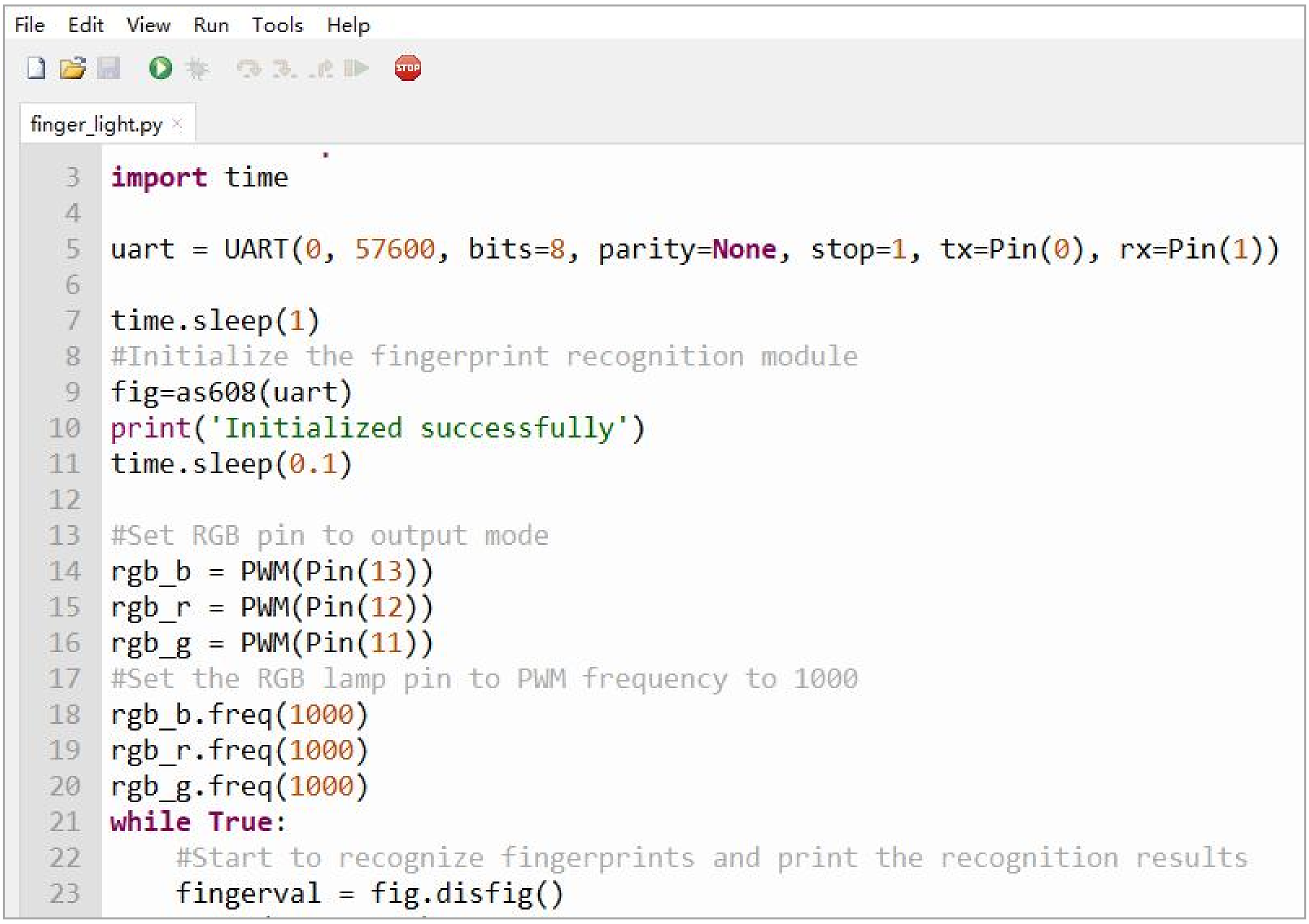
File name: as608.py

4. About code

Please view **finger_delet.py** we provided.

5. Compiling and running code

5.1 We can open the Gesture.py file by Thonny software.



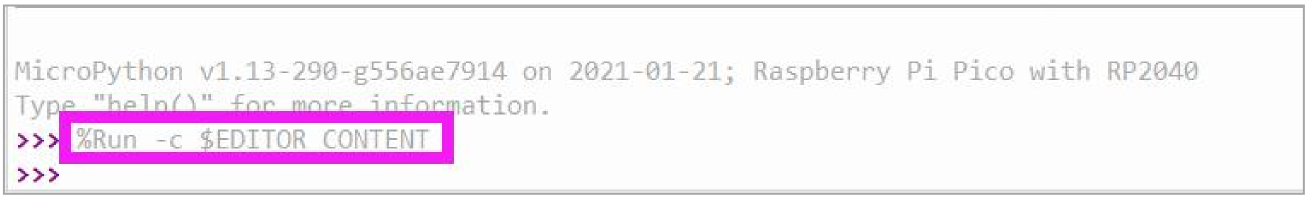
```python
import time

uart = UART(0, 57600, bits=8, parity=None, stop=1, tx=Pin(0), rx=Pin(1))

time.sleep(1)
#Initialize the fingerprint recognition module
fig=as608(uart)
print('Initialized successfully')
time.sleep(0.1)

#Set RGB pin to output mode
rgb_b = PWM(Pin(13))
rgb_r = PWM(Pin(12))
rgb_g = PWM(Pin(11))
#Set the RGB lamp pin to PWM frequency to 1000
rgb_b.freq(1000)
rgb_r.freq(1000)
rgb_g.freq(1000)
while True:
    #Start to recognize fingerprints and print the recognition results
    fingerval = fig.disfig()
```

5.2 In Thonny menu bar, we need to click run button to run this program.
"%Run -c $EDITOR_CONTENT" will be displayed. As shown below.



```
MicroPython v1.13-290-g556ae7914 on 2021-01-21; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
>>>
```

6. Phenomenon

After the program is run successfully. System will start to initialize the fingerprint recognition module.

If the initialize is successfully, it will display "Initialized successfully". Otherwise, please check the baud rate or wiring of the module.

When the Shell window shows "press finger", we need to put our finger on the module. If there is a corresponding fingerprint, the corresponding ID will be displayed. If the corresponding fingerprint is not found, it will display "No matching fingerprint found". If the ID of the current fingerprint is 1, the RGB light is on, otherwise the RGB light is off.