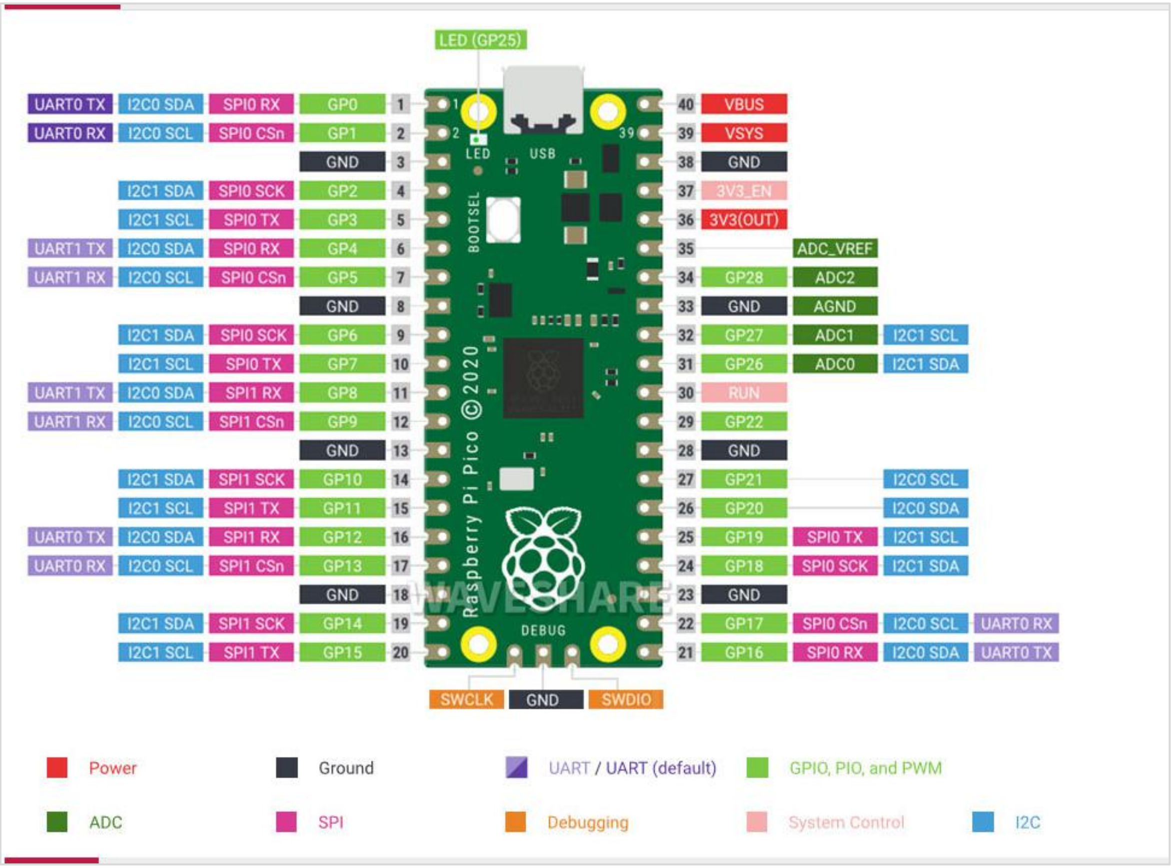
1. Download code

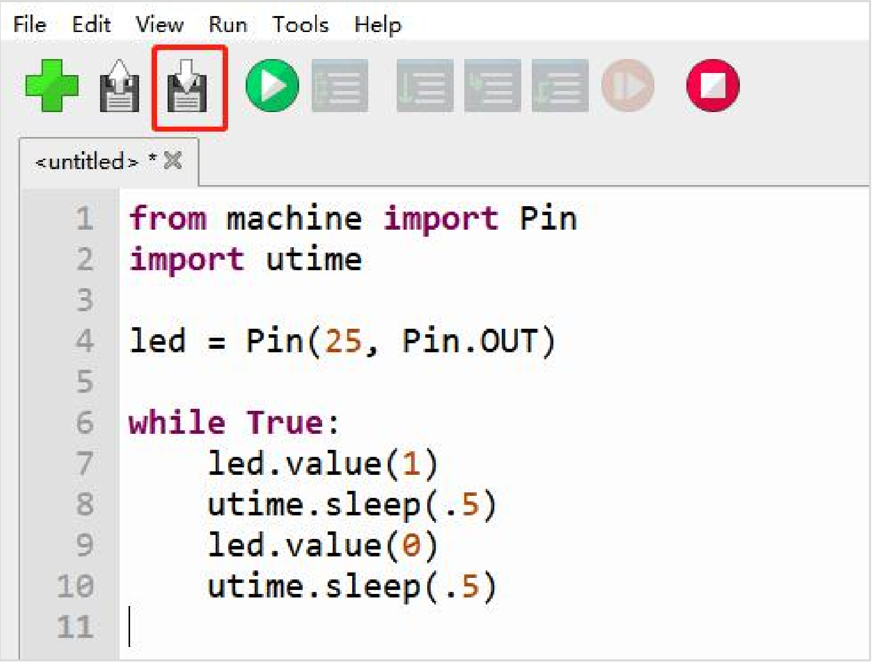
1.1 According to the pin diagram of Pico, we can know that the on board LED of Pico is connected to GPIO25, and we try to control the on board LED.



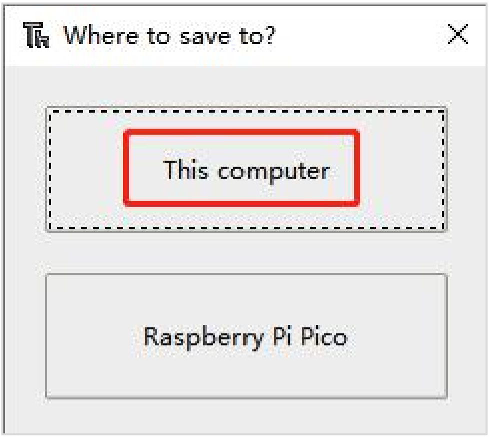
1.2 Code as shown below.

```
from machine import Pin
import utime

led = Pin(25, Pin.OUT)

while True:
        led.value(1)
        utime.sleep(.5)
        led.value(0)
        utime.sleep(.5)
```

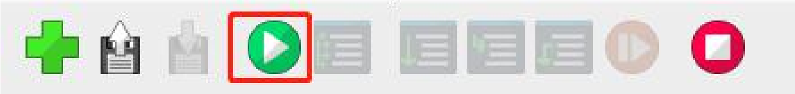
1.3 Enter the above code in the code edit box of Thonny IDE, and save it to local file (led.py).



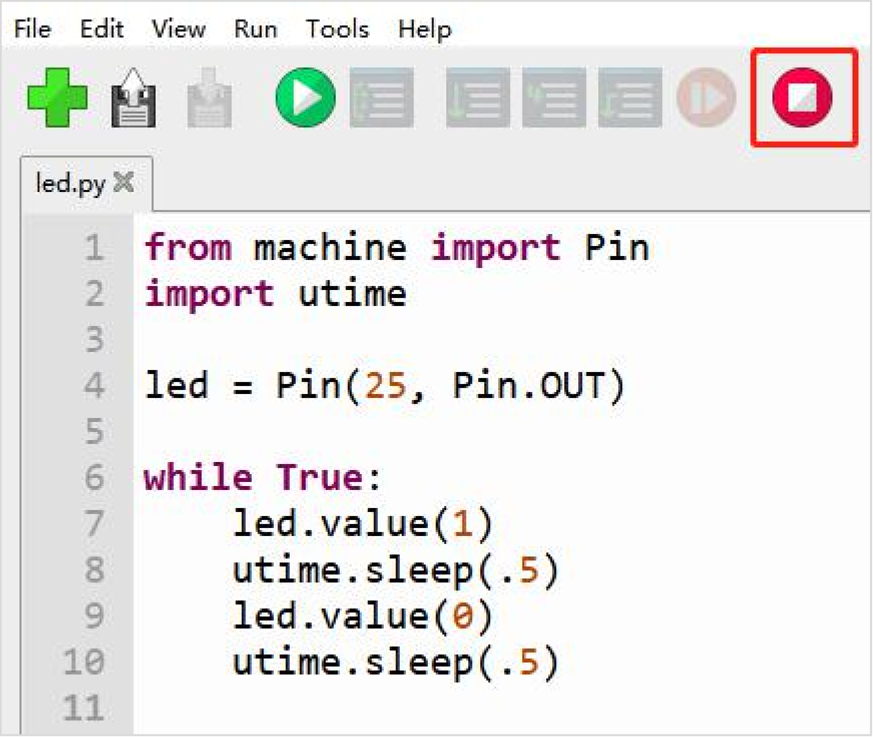
```
from machine import Pin
import utime

led = Pin(25, Pin.OUT)

while True:
    led.value(1)
    utime.sleep(.5)
    led.value(0)
    utime.sleep(.5)
```



1.4 Click the green run button to run this code.



1.5 If you need to end this code or re-connect to Raspberry Pi Pico board, please press the red stop button.



```
from machine import Pin
import utime

led = Pin(25, Pin.OUT)

while True:
    led.value(1)
    utime.sleep(.5)
    led.value(0)
    utime.sleep(.5)
```

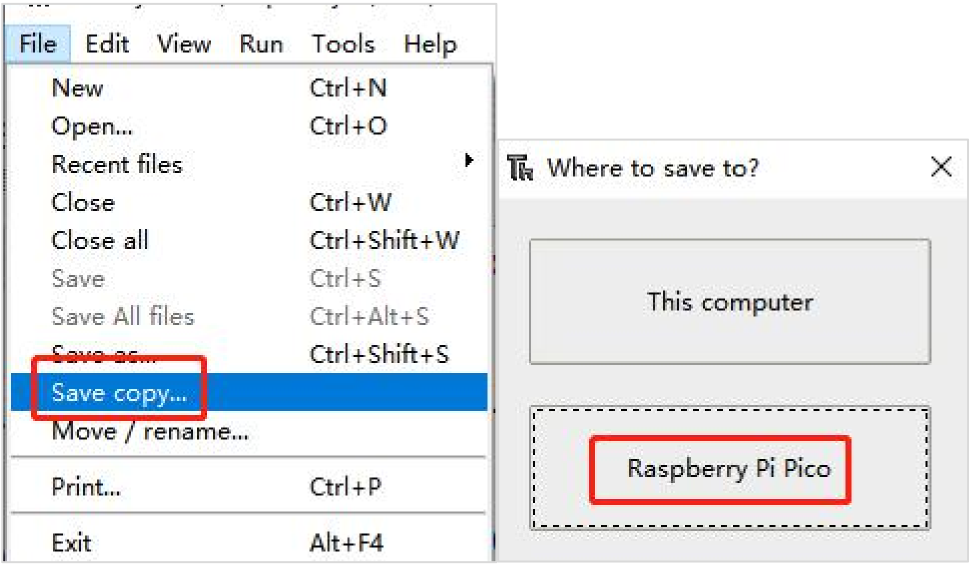
2. Set the boot to start automatically

2.1 After we run the program, the program will not be saved to the Pico motherboard after the Pico is powered off (or the reset button is pressed).

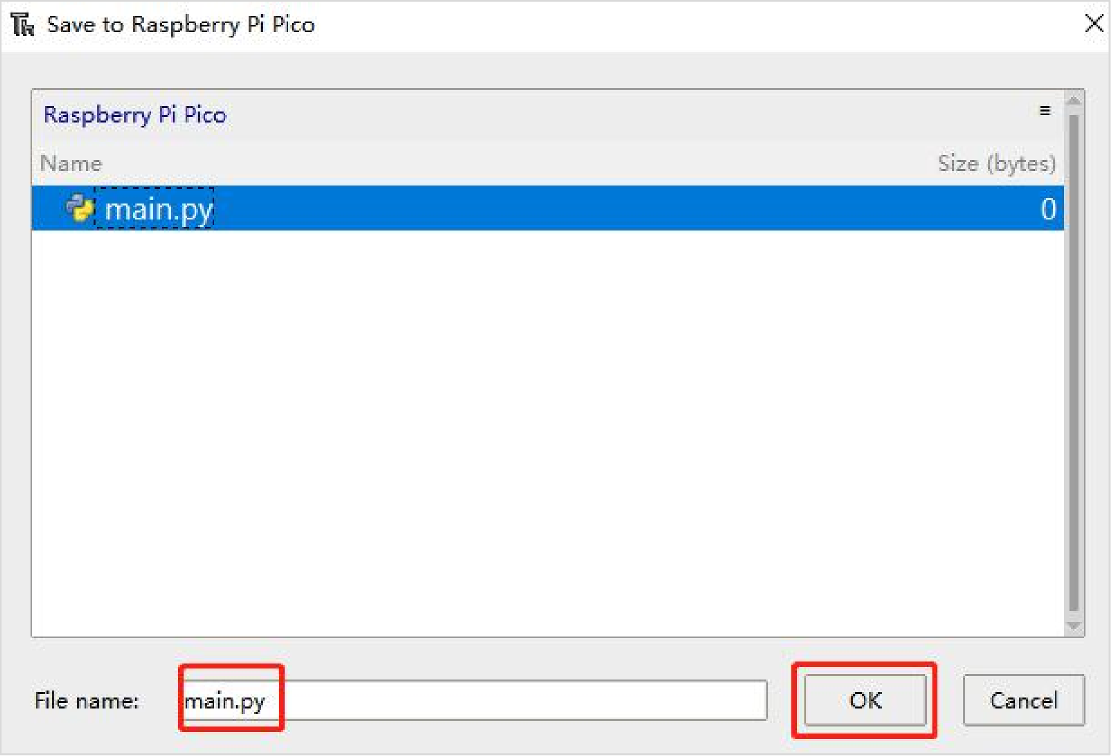
So you need to save the program to the main.py file in Pico to ensure that the entire program will run automatically after the Pico motherboard is reset.

2.2 Proceed as follows:

Click **[File]** -> **[Save copy...]** -> **[Raspberry Pi Pico]**, and the saved name is main.py.



2.3 If the main.py file already exists in Pico, the old main.py file will be overwritten.



Save to Raspberry Pi Pico

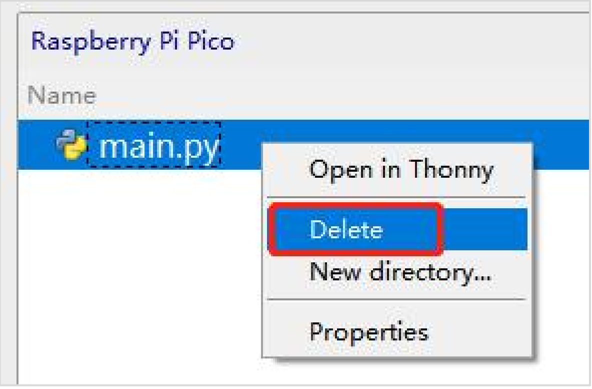
Raspberry Pi Pico

Name	Size (bytes)
main.py	0

File name: main.py

When we restart Pico, it will automatically run the LED flashing program.

When we don't need to run the modified program automatically, we can right-click and select **[Delete]** to delete the main.py file.



Raspberry Pi Pico

Name

main.py