Button control

1. Learning target

1.1 In this course, we will learn how to use pins of the Raspberry Pi Pico board.

1.2 How to read the state of the buttons, and to control the on board LED light by button module.

2. Preparation

Raspberry Pi Pico board *1

Pico sensor expansion board *1

PC *1

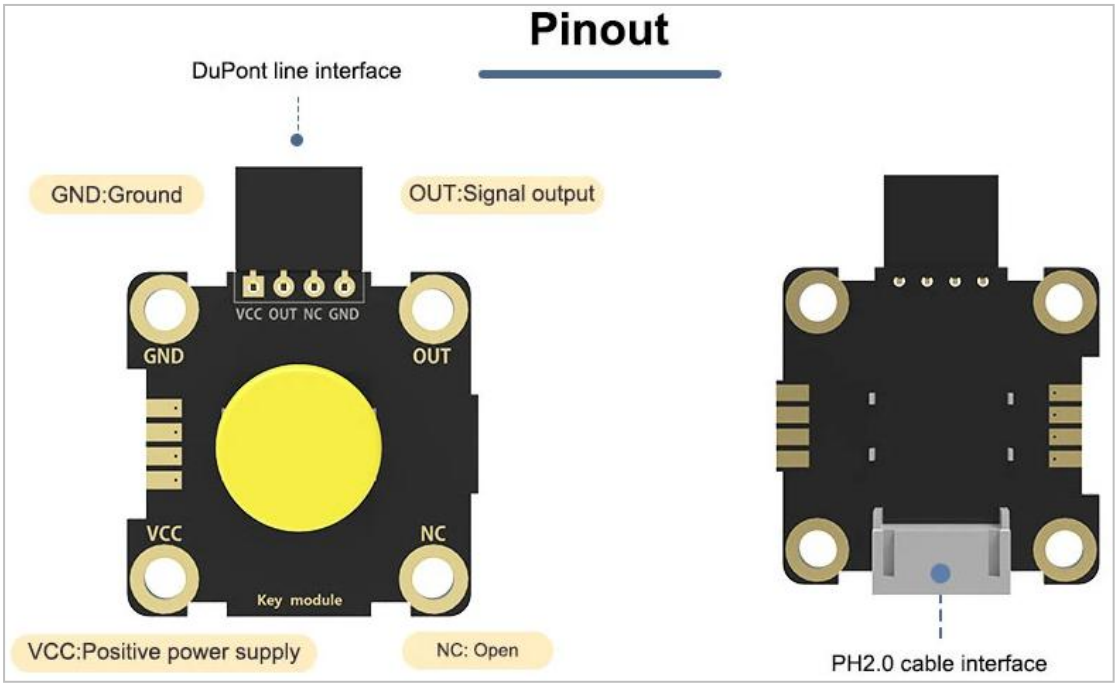
USB data cable *1

Button module *1

male-to-female DuPont line *3

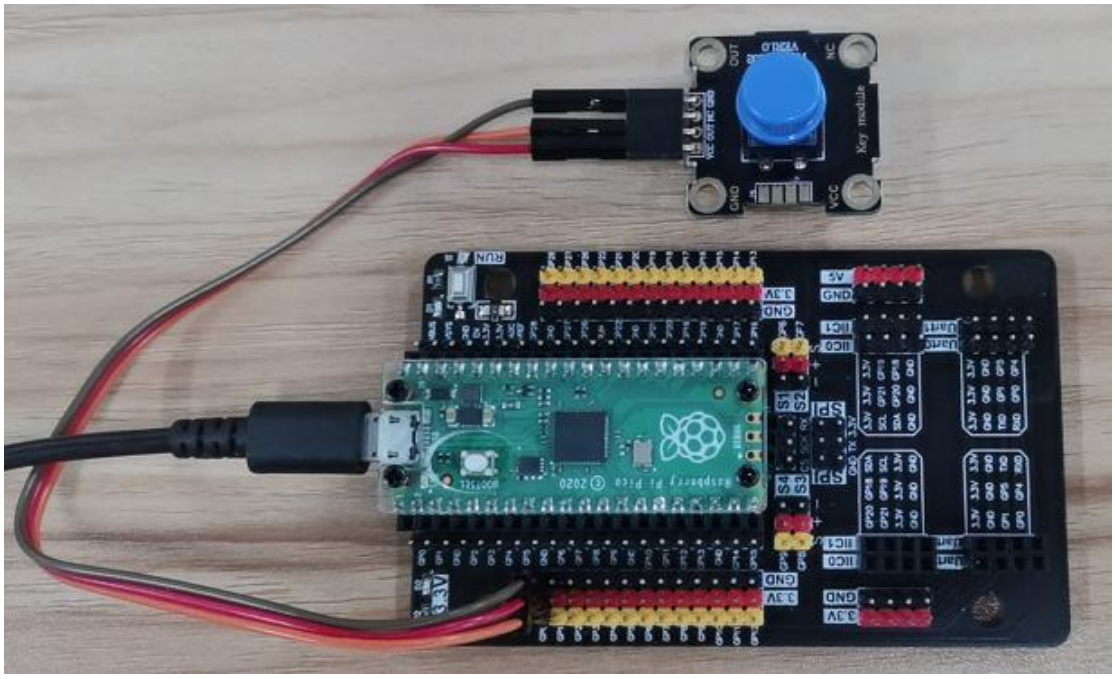
! Note:

The button module needs to pull up the input mode. When the button is pressed, the OUT pin outputs a low level, and when it is released, it outputs a high level.



3. About wiring

Button module	Pico sensor expansion board
IN	GP0
VCC	3.3V
GND	GND



4. About code

Thonny programming

About how to using ThonnyIDE, please check the tutorials in【2.Development environment】

```
from machine import Pin
import utime

# Initialize the LED lights and buttons
led = Pin(25, Pin.OUT)
key = Pin(0, Pin.IN, Pin.PULL_UP)


# Open the LED light that comes with the Pico board
def led_on():
    led.value(1)

# Close the LED light that comes with the Pico board
def led_off():
    led.value(0)

# Read the state of the button, press to return to True, release to return to False
def press_state():
    if key.value() == 0:
        return True
    return False
```

```
# Main loop, when the button is pressed, the LED is on, and “press” is printed every 100
milliseconds;
# when the button is released, the LED is off
while True:
    if press_state() == True:
        print("press")
        led_on()
        utime.sleep(.1)
    else:
        led_off()
```

5. Phenomenon

Click the green run button  of Thonny IDE to start running the program. Click the red stop button  to stop the program.

When the program is running, press button, the LED light on the Pico board will light up, and Shell will print the “press”.
Release the button, the LED light will be closed, and no more printing.