

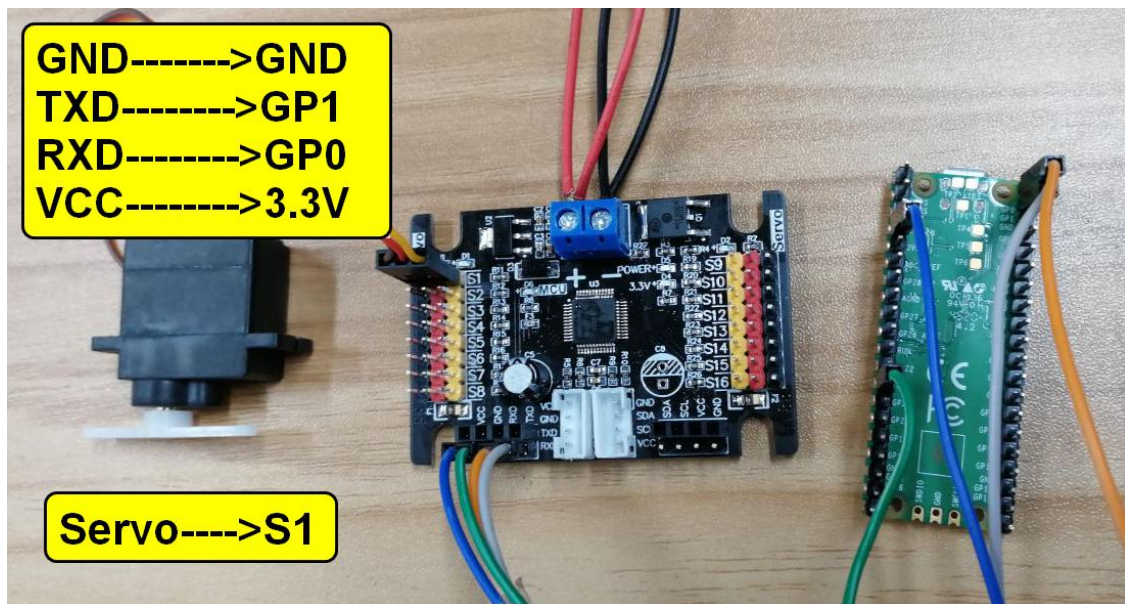
## Serial control servo

### 1. Learning goals

In this course, we mainly learn to use Pico and 16-channel servo debugging board to control the servo through IIC.

### 2. Preparation

Connect the TX and RX of the module to the GP1 and GP0 pins of the Pico board. VCC and GND are connected to 3.3V and GND of Pico board respectively. As shown below.



### 3. About code

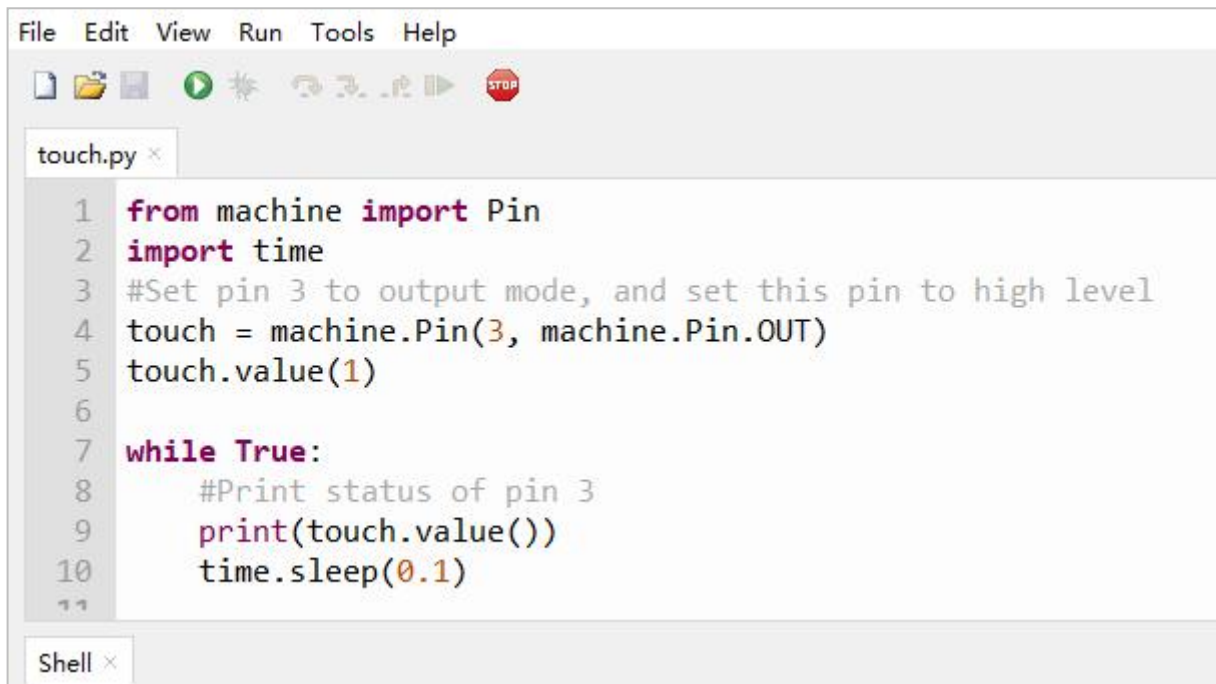
```

16CServo-uart.py
1  from machine import Pin, UART
2  import time
3  #Configure the serial port
4  uart = UART(0, 9600, bits=8, parity=None, stop=1, tx=Pin(0), rx=Pin(1))
5  #uart = UART(1, 9600, bits=8, parity=None, stop=1, tx=Pin(4), rx=Pin(5))
6  #Serial port control servo function
7  def UARTServo(servonum, angle):
8      ...servonum = 64 + servonum
9      ...date1 = int(angle/100 + 48)
10     ...date2 = int((angle%100)/10 + 48)
11     ...date3 = int(angle%10 + 48)
12     ...cmd=bytearray([36, servonum, date1, date2, date3, 35])
13     ...uart.write(cmd)
14     ...time.sleep(0.05)
15
16     UARTServo(1,0)
17     time.sleep(2)
18     UARTServo(1,180)
19     time.sleep(2)
20
21

```

#### 4. Compiling and downloading code

4.1 We can open the .py file by Thonny software.



```
File Edit View Run Tools Help
touch.py x
1 from machine import Pin
2 import time
3 #Set pin 3 to output mode, and set this pin to high level
4 touch = machine.Pin(3, machine.Pin.OUT)
5 touch.value(1)
6
7 while True:
8     #Print status of pin 3
9     print(touch.value())
10    time.sleep(0.1)
11
Shell x
```

4.2 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
>>>
```

#### 5. Phenomenon

After the program is downloaded successfully. The servo will rotate 0°, after 2s it will rotate 180°.