10. PWM control servo

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 use drive servo by PWM.

2. Preparation

Raspberry Pi Pico board *1

Pico sensor expansion board *1

PC *1

USB data cable *1

Servo *1

The working principle of the servo:

The control signal enters the signal modulation chip from the channel of the receiver to obtain the bias voltage of the DC. It has a reference circuit inside, which generates a reference signal with a period of 20ms and a width of 1.5ms. It will compares the DC bias voltage with the voltage of the potentiometer to obtain a voltage difference and output. The positive and negative of the voltage difference is outputted to the motor drive chip to determine the forward and reverse of the motor. Servo rotation angle is by adjusting the duty ratios of PWM (pulse width modulation) signal. The standard PWM (pulse width modulation) signal has a fixed period of 20ms (50Hz). Theoretically, pulse width distribution should be between 1 ms to 2 ms, but in fact between pulse width can be 0.5 ms and 2.5 ms. Pulse width and the servo rotation angle 0°～180° corresponds, as shown below.

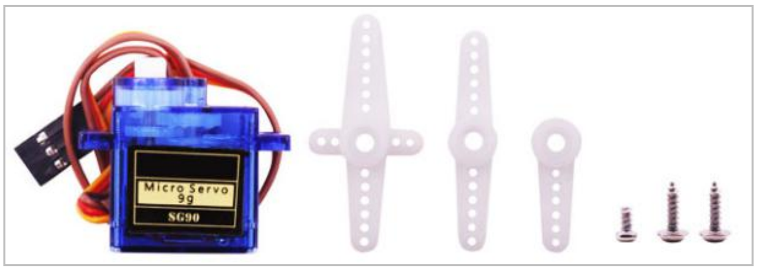
0.5ms-----------------0°

1.0ms-----------------45°

1.5ms-----------------90°

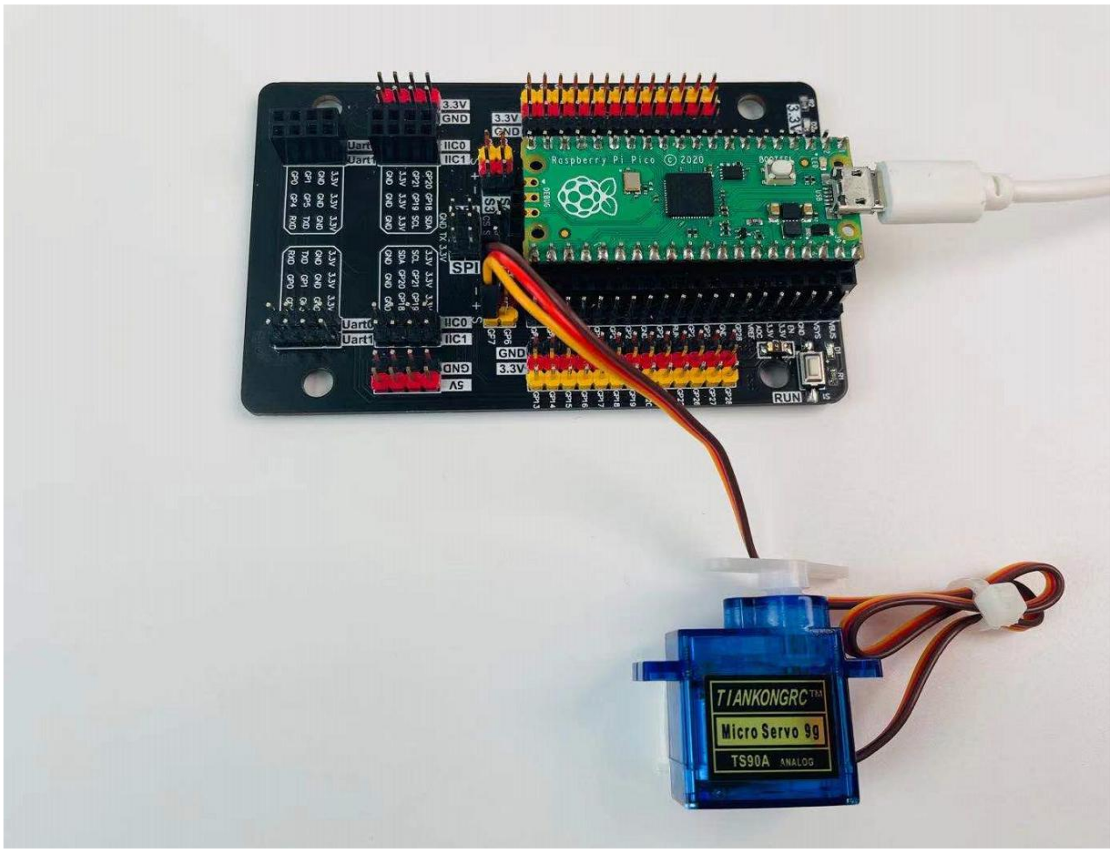
2.0ms-----------------135°

2.5ms-----------------180°



3. About wiring

Servo	Pico sensor expansion board
Yellow line	GP17
Brown line	GND
VCC line	3.3V



4. About code

Thonny programming

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

```
from machine import Pin, PWM
import utime

servo = PWM(Pin(7))
servo.freq(50)

# Numerical remapping
def my_map(x, in_min, in_max, out_min, out_max):
    return int((x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min)

# Control servo, value=[0, 180]
def servo_control(value):
    if value > 180 or value < 0:
        print('Please enter a limited speed value of 0-180 ')
        return
    duty = my_map(value, 0, 180, 500000, 2500000)
    servo.duty_ns(duty)
```

```
# Control servo
while True:
        servo_control(0)
        utime.sleep(1)
        servo_control(180)
        utime.sleep(1)
```

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, the servo will rotate 0°--->180°--->0°--->180°. And keep in loop with this status.