Fan motor

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 the PWM function on Pico to drive a small motor fan.

2. Preparation

Raspberry Pi Pico board *1

Pico sensor expansion board *1

PC *1

USB data cable *1

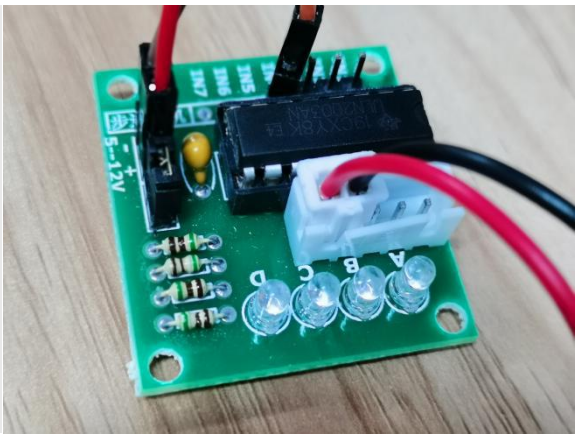
ULN2003 module*1

Motor fan *1

Male-to-male DuPont line *3

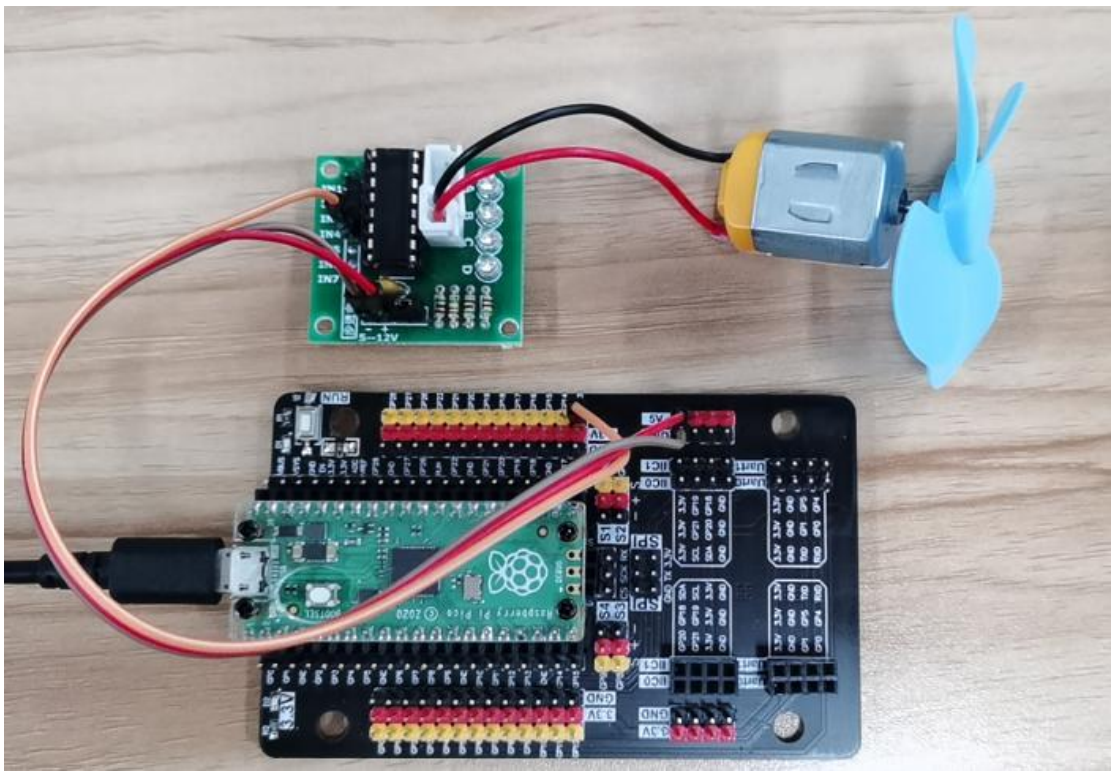
Note:

The small motor fan needs a relatively large current, so the ULN2003 drive module is needed to drive the small motor fan, and the speed of the small motor fan can be controlled by PWM.



3. About wiring

ULN2003 board	Pico sensor expansion board
IN1	GP16
IN2	GP15
IN3	GP14
IN4	GP13
Negative electrode(-)	GND
Positive electrode(-)	5V



4. About code

Thonny programming

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

```
from machine import Pin, PWM

# Initialize the motor fan
fan = PWM(Pin(13))
fan.freq(1000) # Set frequency

# 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)

# Set the fan speed, speed=[0, 100]
def pwm_motor(speed):
    if speed > 100 or speed < 0:
        print('Please enter a limited speed value of 0-100 ')
        return
    pulse = my_map(speed, 0, 100, 0, 65535)
    fan.duty_u16(pulse)

pwm_motor(50)
```

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 D red light of the ULN2003 drive module lights up, and the small motor fan starts to rotate at a speed of 50%.
If you need the fan to stop, you can press the RUN button on the expansion board to reset the Pico, or change the last line of code **pwm_motor(0)** to speed 0 and run it again.

Note: due to current supply problems, the speed value of the small motor fan should not be set too high, otherwise the Pico may be disconnected.