

## Rocker control RGB halo

### 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 RGB light halo and rocker module realize rocker control RGB halo module.

### 2. Preparation

Raspberry Pi Pico board \*1

Pico sensor expansion board \*1

PC \*1

USB data cable \*1

RGB light halo module \*1

Rocker module \*1

Male-to-male DuPont line \*5

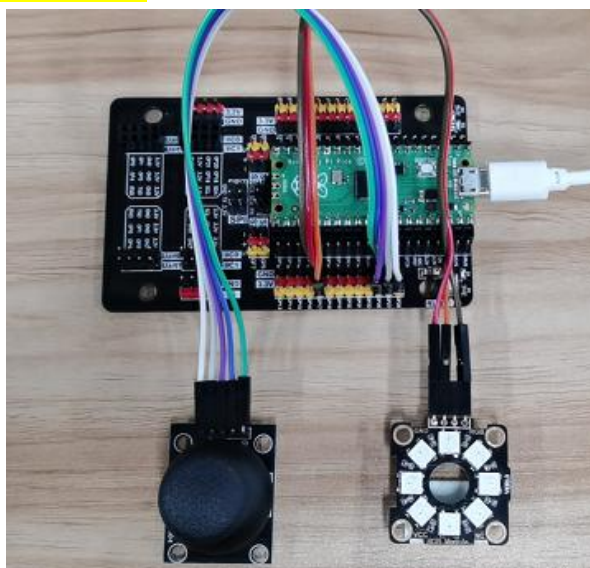
Female-to-male DuPont line \*3

### 3. About wiring

RGB light halo module	Pico sensor expansion board
RGB	GP17
VCC	3.3V
GND	GND

Rocker module	Pico sensor expansion board
+5V	3.3V
GND	GND
VRX	GP26
VRX	GP27
SW	GP28

Note: Since PICO is a 3.3V voltage, the +5V sign written in VCC on the rocker module must be connected to the 3.3V voltage of Pico.



#### 4. About code

##### Thonny programming

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

```
from machine import ADC, Pin
import ws2812b
import random
import utime
import math

ring_pin = 17 #RGB hal connect pin
numpix    = 8  #Number of RGB light

#Initialize the RGB light halo
strip = ws2812b.WS2812B(numpix, 0, ring_pin)
strip.fill(0,0,0)
strip.show()

#Initialize rocker module (ADC function)
rocker_x = ADC(26)
rocker_y = ADC(27)
button = Pin(28, Pin.IN, Pin.PULL_UP)

#Read the value of the X axis, range[0, 255]
def read_x():
    value = int(rocker_x.read_u16()) * 256 / 65536
    return value

#Read the value of the Y axis, range[0, 255]
def read_y():
    value = int(rocker_y.read_u16()) * 256 / 65536
    return value

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

#Control rocker function, val_x=[-128, 128], val_y = [-128, 128]
def rocker_rgb(val_x, val_y, color=[255, 255, 255]):
    index = 0
    offset = 20
```

```

brightness = 0
#Control lights in up and down
if val_x > 0 - offset and val_x < 0 + offset:
    brightness = int(math.fabs(val_y)*2)
    if val_y > 0:
        index = 0
    elif val_y < 0:
        index = 4
#Control lights in left and right
elif val_y > 0 - offset and val_y < 0 + offset:
    brightness = int(math.fabs(val_x)*2)
    if val_x > 0:
        index = 6
    elif val_x < 0:
        index = 2
#Control the lights in the upper left and lower left corners
elif val_x < 0 - offset and val_y > 0 + offset:
    index = 1
    brightness = int(math.sqrt(val_x*val_x + val_y*val_y))
elif val_x < 0 - offset and val_y < 0 - offset:
    index = 3
    brightness = int(math.sqrt(val_x*val_x + val_y*val_y))
#Control the lights in the upper right and lower right corners
elif val_x > 0 + offset and val_y > 0 + offset:
    index = 7
    brightness = int(math.sqrt(2*(val_x*val_x + val_y*val_y)))
elif val_x > 0 + offset and val_y < 0 - offset:
    index = 5
    brightness = int(math.sqrt(2*(val_x*val_x + val_y*val_y)))

strip.fill(0, 0, 0) #Clear the color buffer
strip.brightness(brightness) #Set brightness
strip.set_pixel(index, color[0], color[1], color[2])
strip.show()
return index, brightness

```

#Operate the joystick to change the color of the RGB light halo module.

while True:

    #Convert the X-axis and Y-axis data into a coordinate system, which will be the origin when the joystick is released.

    value\_x = read\_x() - 128 #Read the data of the X axis and convert it into a range from left to right [-128, 128]

    value\_y = 128 - read\_y() #Read the data of the Y axis and convert it into a range from up to

down [-128, 128]

```

state = btn_state() #Read the status of the installation
print("x:%d, y:%d, press:%s" % (value_x, value_y, state))
#Press the joystick button, RGB light become white
if state == True:
    strip.fill(255, 255, 255)
    strip.show()
else: #Control RGB light by rocker
    if value_x > -5 and value_x < 5 and value_y > -5 and value_y < 5:
        strip.fill(0, 0, 0)
        strip.show()
    else:
        #Generate random colors
        r = random.randint(0, 255)
        g = random.randint(0, 255)
        b = random.randint(0, 255)
        color = [r, g, b]
        #Rocker control RGB light halo
        a, b = rocker_rgb(value_x, value_y, color)
        print("---", a, b) #Print the current light and brightness
    utime.sleep(.1)

```

Before running this program, you need to load ws2812 library, please check the specific steps in **【2.Development environment】** .

## 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, we can control the rocker

module to move up and down, left and right, and the RGB light ring will light up the corresponding position light. When we press the rocker module, the RGB light modules all turn on white.