# 8. Rain drop sensor if detect buzzer on and show in OLED

เตรียมอุปกรณ์

- Spidey
- OLED (บนบอร์ด Spidey)
- Rain drop sensor
- Buzzer active

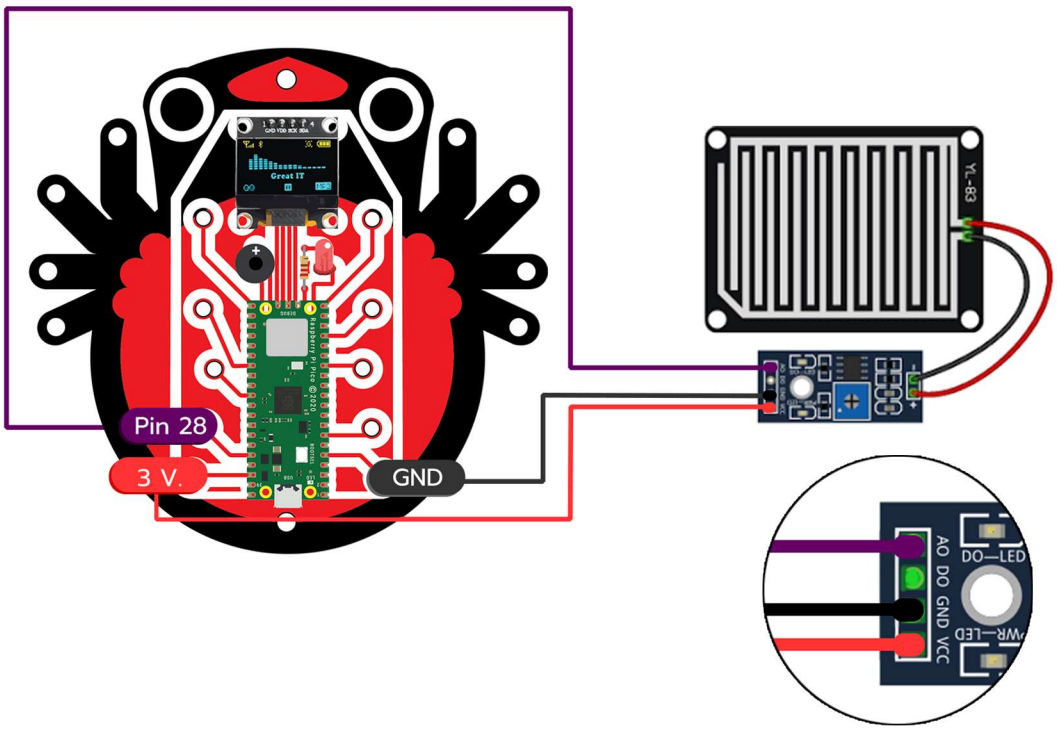
การเชื่อมต่อ

-เริ่มต้น ต่อสายขา GND หรือ ขากราวด์ ของ เซนเซอร์วัดฝน กับสไปดี้ ที่ขา ลบ

-ต่อขา บวก ของ เซนเซอร์เข้ากับ สไปดี้ ที่ขา บวก

-เซนเซอร์ วัดฝน สามารถรับข้อมูลได้ทั้ง Analog และ digital

การเชือมต่อโดยเลือกใช้ Analog เพราะ ให้ค่าข้อมูลที่ระเอียดกว่า Digital โดยต่อเซนเซอร์ไปที่พินที่ 28 ของสไปดี้



การโค้ด

```
from machine import ADC
from machine import Pin
from machine import I2C
import ssd1306
from time import sleep
import time

adc28=ADC(28) #GPIO28 rain drop sensor

def gpio_set(pin,value):
  if value >= 1:
    Pin(pin, Pin.OUT).on()
  else:
    Pin(pin, Pin.OUT).off()


while True:
  if (adc28.read_u16()) < 30000:
    gpio_set((6), True)
  else:
    gpio_set((6), False)
  i2c=I2C(0, scl=Pin(1), sda=Pin(4))
  oled_width = 128
  oled_height = 64
  oled = ssd1306.SSD1306_I2C(oled_width, oled_height, i2c)
  oled.text(('rain :' + str(adc28.read_u16())), 40, 40)
  oled.show()
  oled.fill(0)
  time.sleep(1)
```

อธิบายโค้ด

เป็นโค้ด Python ที่ใช้สำหรับอ่านค่าจากเซ็นเซอร์ตรวจจับการตกฝนโดยใช้ ADC และแสดงค่าบนหน้าจอ OLED นี่คือการแยกแต่ละบรรทัด:

1. **from machine import ADC, Pin, I2C**: นำเข้าโมดูล ADC, Pin, และ I2C จากไลบรารี machine เพื่อใช้ในการทำงานที่เกี่ยวข้องกับฮาร์ดแวร์
2. **import ssd1306**: นำเข้าโมดูล ssd1306 เพื่อควบคุมหน้าจอ OLED
3. **from time import sleep**: นำเข้าฟังก์ชัน sleep จากโมดูล time เพื่อสร้างความหน่วงในโค้ด

4. **adc28 = ADC(28)**: สร้างอ็อบเจกต์ ADC และกำหนดให้ตัวแปร adc28 เก็บค่านี้ไว้ นี่เป็นการตั้งค่า ADC เพื่ออ่านค่าจากขา GPIO หมายเลข 28 ซึ่งเชื่อมต่อกับเซ็นเซอร์ตรวจจับการตกฝน
5. **def gpio_set(pin, value):**: นิยามฟังก์ชันชื่อ gpio_set ที่รับพารามิเตอร์เป็น pin และ value เพื่อตั้งค่าขาที่ระบุให้เปิดหรือปิดตามค่าที่กำหนด
6. **if value >= 1:**: เช็คว่าค่าที่รับเข้ามาในฟังก์ชัน gpio_set มีค่ามากกว่าหรือเท่ากับ 1 หรือไม่
7. **Pin(pin, Pin.OUT).on()**: ถ้าค่ามากกว่าหรือเท่ากับ 1 ให้ตั้งขาที่ระบุให้เป็นโหมดเอาต์พุตและเปิด
8. **Pin(pin, Pin.OUT).off()**: ถ้าค่าน้อยกว่า 1 ให้ตั้งขาที่ระบุให้เป็นโหมดเอาต์พุตและปิด
9. **while True:**: เริ่มลูปไม่จำกัดจำนวนรอบ
10. **if (adc28.read_u16()) < 30000:**: อ่านค่าแอนะล็อกจากเซ็นเซอร์ตรวจจับการตกฝนโดยใช้ ADC และเช็คว่าค่าน้อยกว่า 30000 หรือไม่
11. **gpio_set((6), True)**: ถ้าค่าน้อยกว่า 30000 ให้เรียกใช้ฟังก์ชัน gpio_set เพื่อเปิดขาหมายเลข 6
12. **gpio_set((6), False)**: ถ้าค่ามากกว่าหรือเท่ากับ 30000 ให้เรียกใช้ฟังก์ชัน gpio_set เพื่อปิดขาหมายเลข 6
13. **i2c = I2C(0, scl=Pin(1), sda=Pin(4))**: สร้างอ็อบเจกต์ I2C เพื่อใช้สื่อสารกับหน้าจอ OLED โดยใช้ขา SCL และ SDA ที่ระบุ
14. **oled_width = 128**: กำหนดความกว้างของหน้าจอ OLED เป็น 128 พิกเซล
15. **oled_height = 64**: กำหนดความสูงของหน้าจอ OLED เป็น 64 พิกเซล
16. **oled = ssd1306.SSD1306_I2C(oled_width, oled_height, i2c)**: สร้างอ็อบเจกต์ SSD1306_I2C เพื่อควบคุมหน้าจอ OLED ด้วยความกว้างและความสูงที่ระบุ และอ็อบเจกต์ I2C
17. **oled.text(('rain :' + str(adc28.read_u16())), 40, 40)**: กำหนดข้อความที่จะแสดงบนหน้าจอ OLED เป็น 'rain :' ตามด้วยค่าที่อ่านจากเซ็นเซอร์ตรวจจับการตกฝน ข้อความนี้จะแสดงที่ตำแหน่งพิกเซล (40, 40) บนหน้าจอ
18. **oled.show()**: อัปเดตหน้าจอ OLED เพื่อแสดงข้อความ
19. **oled.fill(0)**: เคลียร์หน้าจอ OLED โดยเติมค่าศูนย์ในทุกๆ พิกเซล
20. **time.sleep(1)**: หยุดรันโปรแกรมเป็นเวลา 1 วินาที

โค้ดนี้จะวนลูปอ่านค่าจากเซ็นเซอร์ตรวจจับการตกฝนโดยใช้ ADC และแสดงค่าบนหน้าจอ OLED ซึ่งจะเรียกใช้ฟังก์ชัน gpio_set เพื่อเปิดหรือปิดขาแสดงสถานะตามค่าที่อ่านได้ และจะเริ่มลูปใหม่หลังจากนั้น