

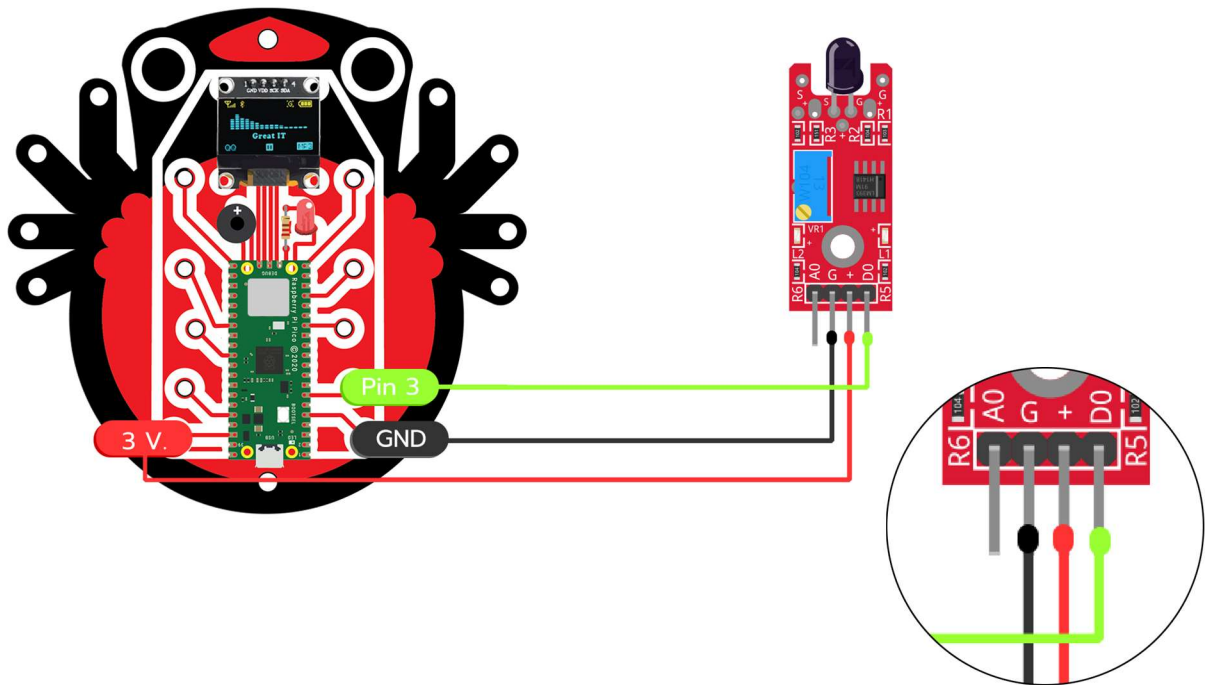
## 9.frame detector sensor show in OLED

เตรียมอุปกรณ์

- Spidey
- OLED (บนบอร์ด Spidey)
- frame detector sensor

การเชื่อมต่อ

-ตัวเซนเซอร์จะคล้ายกับ เซนเซอร์วัดฝน แต่เซนเซอร์วัดไฟ ตัวนี้เราจะใช้ Digital และปรับค่าความเข้มข้นของแสงที่ตัวเซนเซอร์แทน เพื่อให้ง่ายต่อการใช้งาน โดยให้ต่อเซนเซอร์เข้ากับพินที่ 3 ของสไปดี้



การโค้ด

```
from machine import Pin # frame detector sensor
from machine import I2C
import ssd1306
from time import sleep
```

```

import time

pIn3=Pin(3, Pin.IN, Pin.PULL_UP)

while True:
    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((''.join([str(x) for x in ['frame', ' : ',
pIn3.value()]])), 40, 47)
    oled.show()
    oled.fill(0)
    time.sleep(1)

```

## อธิบายโค้ด

- นำเข้าไลบรารีและคลาสที่ใช้ในโค้ด:
  - from machine import Pin:** นำเข้าคลาส **Pin** จากไลบรารี **machine** เพื่อใช้สำหรับควบคุมขา GPIO.
  - from machine import I2C:** นำเข้าคลาส **I2C** จากไลบรารี **machine** เพื่อใช้สำหรับการเชื่อมต่อแบบ I2C.
  - import ssd1306:** นำเข้าไลบรารี **ssd1306** เพื่อใช้สำหรับควบคุมหน้าจอ OLED.
  - from time import sleep:** นำเข้าฟังก์ชัน **sleep** จากไลบรารี **time** เพื่อใช้ในการหน่วงเวลา.
- กำหนดขาและอ็อบเจ็กต์สำหรับเซ็นเซอร์และหน้าจอ OLED:
  - pIn3 = Pin(3, Pin.IN, Pin.PULL\_UP):** สร้างอ็อบเจ็กต์ **pIn3** สำหรับขา GPIO 3 โดยกำหนดให้เป็นขาแสดงสถานะ (input) และตั้งค่าให้มีการเชื่อมต่อตัวยึด (pull-up) เพื่อใช้ในการตรวจจับเฟรม.
  - i2c = I2C(0, scl=Pin(1), sda=Pin(4)):** สร้างอ็อบเจ็กต์ **i2c** สำหรับการเชื่อมต่อแบบ I2C โดยกำหนดขา SCL เป็น GPIO 1 และขา SDA เป็น GPIO 4.
  - oled\_width = 128 และ oled\_height = 64:** กำหนดขนาดความกว้างและความสูงของหน้าจอ OLED.

- `oled = ssd1306.SSD1306_I2C(oled_width, oled_height, i2c)`: สร้างอ็อบเจกต์ `oled` สำหรับควบคุมหน้าจอ OLED ด้วยค่าความกว้างและความสูงที่กำหนด.
3. เริ่มลูปรการทำงานอย่างต่อเนื่อง:
    - `while True::` เริ่มลูปรการทำงานอย่างต่อเนื่อง.
  4. อ่านค่าสถานะของเฟรมและแสดงผลบนหน้าจอ OLED:
    - `oled.text(''.join([str(x) for x in ['frame', ' : ', pin3.value()])), 40, 47)`: สร้างข้อความที่จะแสดงบนหน้าจอ OLED โดยรวมข้อความ "frame : " และค่าสถานะของเฟรมจากการอ่านค่าจากขา GPIO 3.
    - `oled.show()`: แสดงผลบนหน้าจอ OLED.
    - `oled.fill(0)`: เคลียร์หน้าจอ OLED เพื่อเตรียมแสดงผลใหม่ในรอบถัดไป.
    - `time.sleep(1)`: หน่วงเวลา 1 วินาทีก่อนที่จะเริ่มการทำงานในรอบถัดไป.

โค้ดนี้จะทำงานอย่างต่อเนื่องเพื่อตรวจจับเฟรมและแสดงผลสถานะของเฟรมบนหน้าจอ OLED ด้วยการอ่านค่าจากขา GPIO 3 และควบคุมการแสดงผลบนหน้าจอ OLED ผ่าน I2C ตลอดเวลาที่โปรแกรมทำงานอยู่ในลูป `while True`.