11. frame detector sensor if detect Buzzer Passive and RGB turn on show in OLED

เตรียมอุปกรณ์

- Spidey
- OLED (บนบอร์ด Spidey)
- frame detector sensor
- Buzzer Passive
- RGB

การเชื่อมต่อ

-เริ่มต้น ต่อสายขา GND หรือ ขากราวด์ ของ เซนเซอร์วัดไฟกับสไปดี้ ที่ขา ลบ

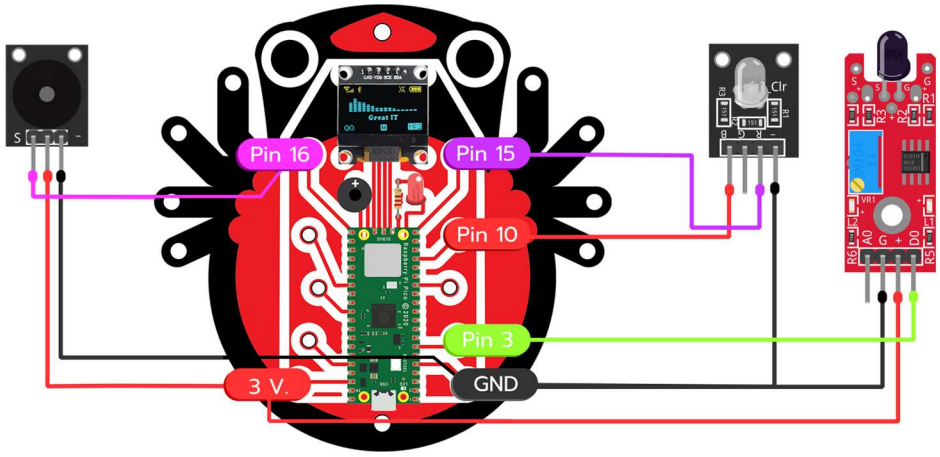
-ต่อขา บวก ของ เซนเซอร์เข้ากับ สไปดี้ ที่ขา บวก

-ตัวเซนเซอร์จะคล้ายกับ เซนเซอร์วัดฝน แต่เซนเซอร์ วัดไฟ ตัวนี้เราจะใช้ Digital และปรับค่าความเข้มข้นของแสงที่ตัวเซ้นเซอร์แทน เพื่อให้ง่ายต่อการใช้งาน โดยให้ต่อเซนเซอร์เข้ากับพินที่ 3 ของสไปดี้

- Buzzer แบบ Passive โดยต่อขาที่ 16 ของสไปดี้

-โดยโปรเจกต์นี้เราเพิ่ม การแจ้งเตือนให้ชัดเจนขึ้นอีกโดยการเพิ่ม RGB เข้ามา และ ขา R หรือ Red ของ RGB ต่อขา 15 และ B ต่อที่ขา 10 ของสไปดี้

*สามารถใช้ Traffic light แทน RGB ได้



การโค้ด

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

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

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


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()
  if (pIn3.value()) == 0:
    pwm15 = PWM(Pin(16)) #buzzer
    pwm15.freq(1000)
```

```
    pwm15.duty_u16(1000)
    gpio_set((10), True)  #RGB
    gpio_set((15), False) #RGB
    time.sleep(1)
  else:
    pwm15 = PWM(Pin(16)) #buzzer
    pwm15.freq(1000)
    pwm15.duty_u16(0)
    gpio_set((15), False) #RGB
    gpio_set((10), True)  #RGB
  oled.fill(0)
  time.sleep(1)
```

อธิบายโค้ด

โค้ดดังกล่าวเพิ่มการควบคุมสีของ LED RGB เพิ่มเติม

1. นำเข้าโมดูล **Pin**, **I2C**, **ssd1306**, **sleep**, **PWM** และ **time**.
2. กำหนดขา GPIO 3 เป็นขานำเข้าแบบ Pull-up ซึ่งเป็นขาที่ใช้ตรวจจับสถานะของเฟรม:
    - **pIn3 = Pin(3, Pin.IN, Pin.PULL_UP)**
3. กำหนดฟังก์ชัน **gpio_set()** เพื่อควบคุมสถานะของขา GPIO ให้เป็นสูงหรือต่ำ:
    - **def gpio_set(pin, value):**: ประกาศฟังก์ชัน **gpio_set()** รับพารามิเตอร์ **pin** และ **value**.
    - **if value >= 1:**: ตรวจสอบค่า **value** ว่ามากกว่าหรือเท่ากับ 1 หรือไม่.
        - **Pin(pin, Pin.OUT).on()**: กำหนดขา GPIO ให้เป็นสูง.
    - **else:**: กรณีค่า **value** น้อยกว่า 1.
        - **Pin(pin, Pin.OUT).off()**: กำหนดขา GPIO ให้เป็นต่ำ.
4. เริ่มต้นการทำงานในลูป **while True:**.
5. สร้างอ็อบเจ็กต์ I2C สำหรับการเชื่อมต่อกับหน้าจอ OLED:
    - **i2c = I2C(0, scl=Pin(1), sda=Pin(4))**
6. กำหนดขนาดของหน้าจอ OLED:
    - **oled_width = 128**
    - **oled_height = 64**
7. สร้างอ็อบเจ็กต์ SSD1306_I2C สำหรับการควบคุมหน้าจอ OLED:
    - **oled = ssd1306.SSD1306_I2C(oled_width, oled_height, i2c)**
8. แสดงผลสถานะของเฟรมบนหน้าจอ OLED:

- **oled.text((''.join([str(x) for x in ['frame', ' : ', pIn3.value()]])), 40, 47)**: สร้างข้อความที่แสดงสถานะของเฟรมโดยใช้ค่าจากการอ่านขา GPIO 3 และแสดงผลที่พิกัด (40, 47) บนหน้าจอ OLED.

9. แสดงหน้าจอ OLED ที่มีข้อความและสถานะของเฟรม:
    - **oled.show()**
10. ตรวจสอบสถานะของเฟรม:
    - **if (pIn3.value()) == 0:**: ตรวจสอบว่าค่าที่อ่านจากขา GPIO 3 เป็น 0 หรือไม่.
        - **pwm15 = PWM(Pin(15))**: สร้างอ็อบเจ็กต์ PWM สำหรับควบคุมขา GPIO 15 (Buzzer).
        - **pwm15.freq(1000)**: ตั้งค่าความถี่ให้เป็น 1000 Hz.
        - **pwm15.duty_u16(1000)**: ตั้งค่า Duty Cycle เป็น 1000.
        - **gpio_set((13), True)**: เปิด LED RGB โดยกำหนดขา GPIO 13 เป็นสูง (สีแดง).
        - **gpio_set((14), False)**: ปิด LED RGB โดยกำหนดขา GPIO 14 เป็นต่ำ (สีเขียว).
        - **time.sleep(1)**: หน่วงเวลา 1 วินาที.
- **else:**: กรณีค่าที่อ่านจากขา GPIO 3 ไม่ใช่ 0.
    - **pwm15 = PWM(Pin(15))**: สร้างอ็อบเจ็กต์ PWM สำหรับควบคุมขา GPIO 15 (Buzzer).
    - **pwm15.freq(1000)**: ตั้งค่าความถี่ให้เป็น 1000 Hz.
    - **pwm15.duty_u16(0)**: ตั้งค่า Duty Cycle เป็น 0.
    - **gpio_set((13), False)**: ปิด LED RGB โดยกำหนดขา GPIO 13 เป็นต่ำ (สีแดง).
    - **gpio_set((14), True)**: เปิด LED RGB โดยกำหนดขา GPIO 14 เป็นสูง (สีเขียว).

11. เตรียมหน้าจอ OLED สำหรับแสดงข้อมูลถัดไป:
    - **oled.fill(0)**: เตรียมหน้าจอ OLED โดยล้างสีดำทั้งหน้าจอ.
12. หน่วงเวลา 1 วินาที:
    - **time.sleep(1)**