

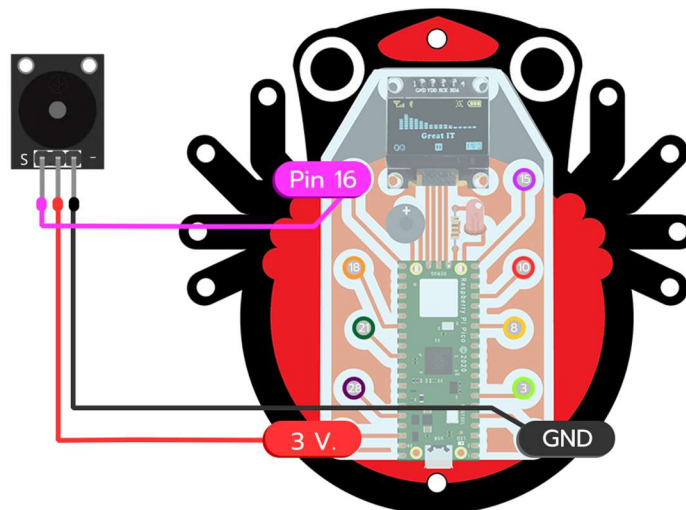
13. Passive buzzer

เตรียมอุปกรณ์

- Spidey
- buzzer passive

การเชื่อมต่อ

- เริ่มต้น ต่อสายขา GND หรือ ขากราวด์ ของ Buzzer กับสไปดี้ ที่ขา ลบ
- ต่อขา บวก ของ เซนเซอร์เข้ากับ สไปดี้ ที่ขา บวก
- ต่อขา S ที่ buzzer ที่ขา 16 ของสไปดี้



การโค้ด

```
from machine import Pin, PWM
from time import sleep

buzzerPIN=16
BuzzerObj=PWM(Pin(buzzerPIN))

def
buzzer(buzzerPinObject,frequency,sound_duration,silence_duration
):
    # Set duty cycle to a positive value to emit sound from
```

```

buzzer
    buzzerPinObject.duty_u16(int(65536*0.2))
    # Set frequency
    buzzerPinObject.freq(frequency)
    # wait for sound duration
    sleep(sound_duration)
    # Set duty cycle to zero to stop sound
    buzzerPinObject.duty_u16(int(65536*0))
    # Wait for sound interruption, if needed
    sleep(silence_duration)

# Play following notes by changing frequency:
#C (DO)
buzzer(BuzzerObj, 523, 0.5, 0.1)

#D (RE)
buzzer(BuzzerObj, 587, 0.5, 0.1)

#E (MI)
buzzer(BuzzerObj, 659, 0.5, 0.1)

#F (FA)
buzzer(BuzzerObj, 698, 0.5, 0.1)

#G (SOL)
buzzer(BuzzerObj, 784, 0.5, 0.1)

#A (LA)
buzzer(BuzzerObj, 880, 0.5, 0.1)

#B (SI)
buzzer(BuzzerObj, 987, 0.5, 0.1)

#Deactivates the buzzer
BuzzerObj.deinit()

```

โค้ดดังกล่าวเป็นตัวอย่างการใช้ MicroPython เพื่อเล่นเสียงบนบัชเซอร์ (Buzzer) โดยกำหนดค่าความถี่ของเสียงแต่ละโน้ตและระยะเวลาการเล่นเสียงและช่วงเวลาเงียบ

โค้ดทำงานดังนี้:

1. **buzzerPIN = 16** กำหนดขา GPIO ที่เชื่อมต่อกับบัชเซอร์
2. **BuzzerObj = PWM(Pin(buzzerPIN))** สร้างอ็อบเจกต์ PWM สำหรับการควบคุมบัชเซอร์

3. `def buzzer(buzzerPinObject, frequency, sound_duration, silence_duration):` สร้างฟังก์ชัน `buzzer` เพื่อกำหนดการเล่นเสียงบนบอร์ด
4. ในฟังก์ชัน `buzzer()`:
 - `buzzerPinObject.duty_u16(int(65536*0.2))` กำหนดค่าเกรดของเสียงให้เป็นเสียงเต็มกำลัง
 - `buzzerPinObject.freq(frequency)` กำหนดความถี่ของเสียง
 - `sleep(sound_duration)` หน่วงเวลาเล่นเสียงตามระยะเวลาที่กำหนด
 - `buzzerPinObject.duty_u16(int(65536*0))` หยุดเสียงโดยกำหนดเกรดเสียงเป็นศูนย์
 - `sleep(silence_duration)` หน่วงเวลาเงียบหลังจากเล่นเสียงตามระยะเวลาที่กำหนด
5. เรียกใช้ฟังก์ชัน `buzzer()` เพื่อเล่นโน้ตแต่ละโน้ตที่ต้องการ โดยกำหนดความถี่ของเสียงและระยะเวลาการเล่นเสียงและช่วงเวลาเงียบ
6. `BuzzerObj.deinit()` เลิกใช้งานบอร์ดโดยปิดอับเจกต์ PWM