

Internet de las Cosas

Prácticas PIR



Rogelio Ferreira Escutia

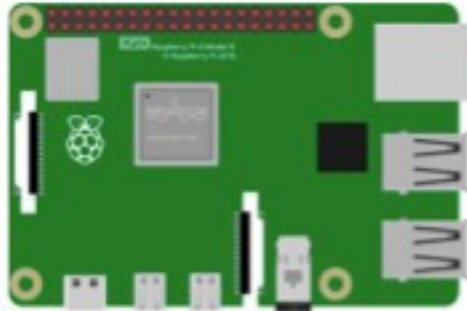
Profesor / Investigador
Tecnológico Nacional de México
Campus Morelia



PIR
(human body
infrared pyroelectric)

Material

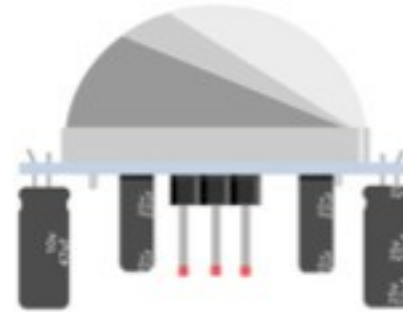
1 * Raspberry Pi



1 * T-Extension Board



1 * PIR Sensor Module



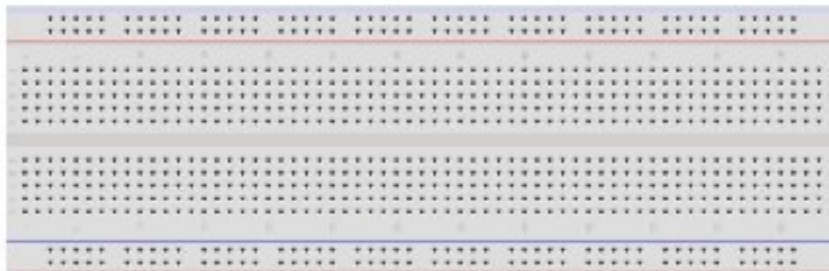
1 * 40-pin Cable



Several Jumper Wires



1 * Breadboard



3 * 220 Resistor



1 * RGB LED



Sensor

- PIR

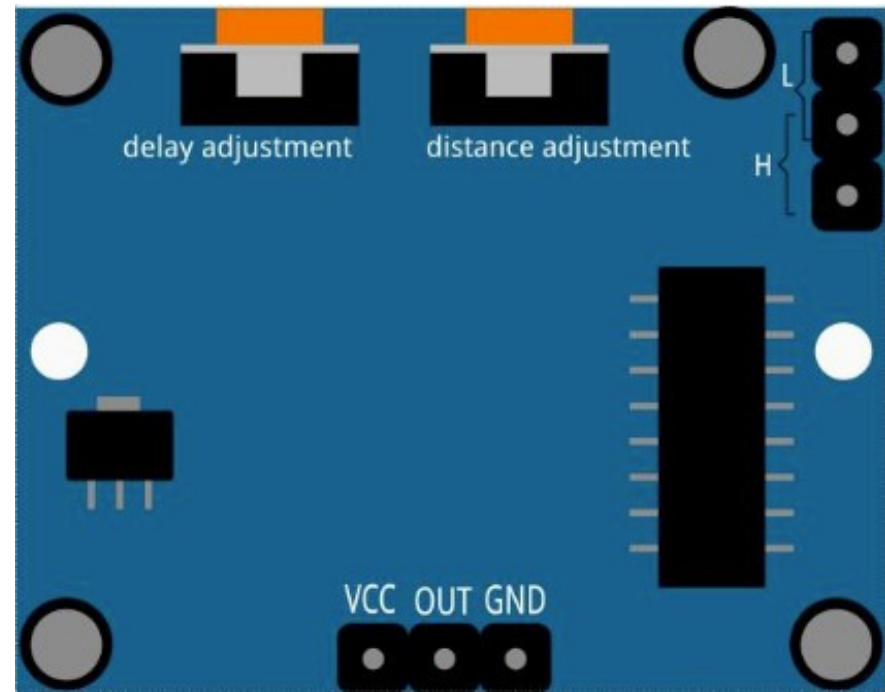
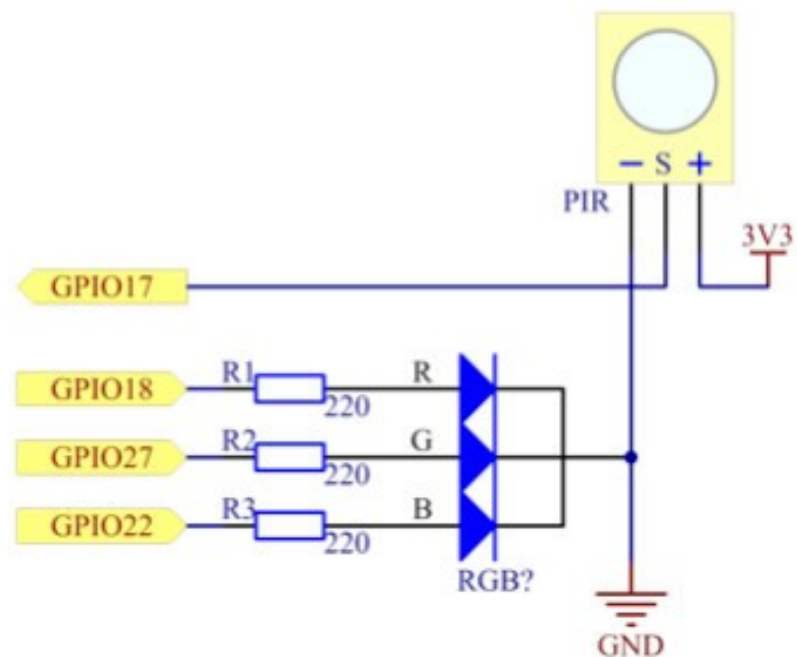


Diagrama eléctrico

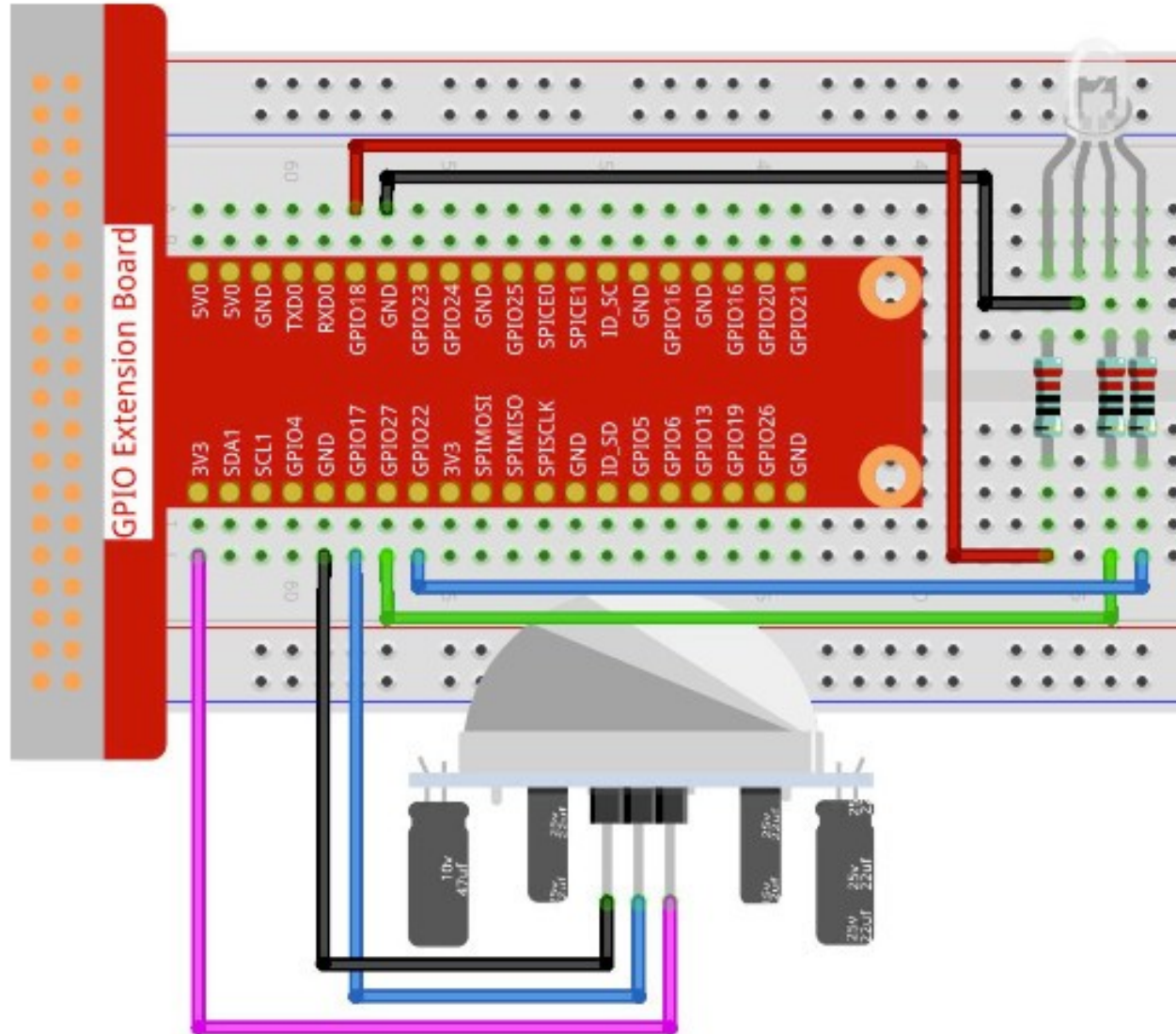
- PIR

T-Board Name	physical	wiringPi	BCM
GPIO17	Pin 11	0	17
GPIO18	Pin12	1	18
GPIO27	Pin13	2	27
GPIO22	Pin15	3	22



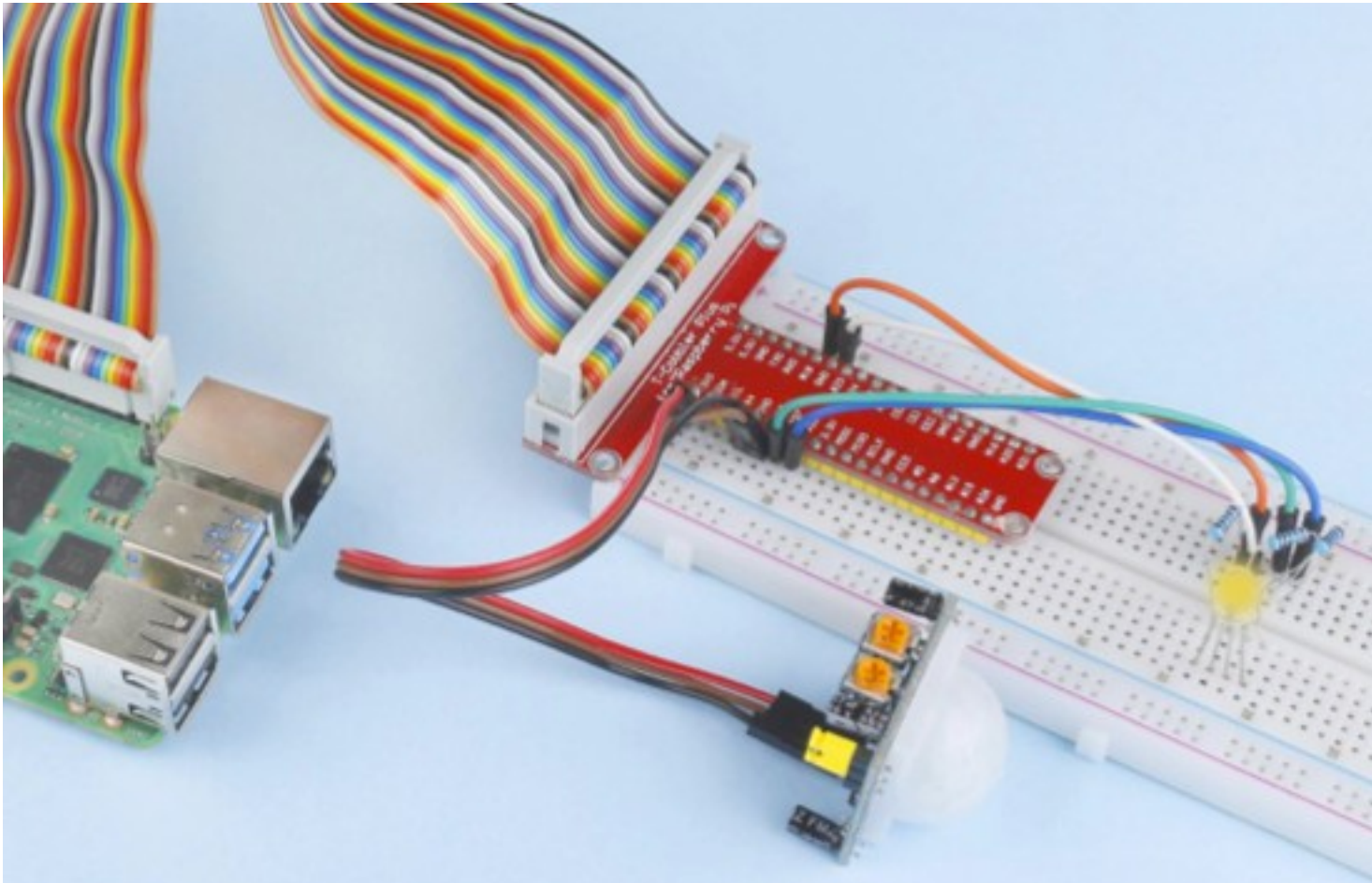
Conexión

- PIR



Conexión

- PIR



Código

- PIR

```
import RPi.GPIO as GPIO
import time

rgbPins = {'Red':18, 'Green':27, 'Blue':22}
pirPin = 17    # the pir connect to pin17
```



Código

■ PIR

```
def setup():
    global p_R, p_G, p_B
    GPIO.setmode(GPIO.BCM)      # Set the GPIO modes to BCM Numbering
    GPIO.setup(pirPin, GPIO.IN)  # Set pirPin to input
    # Set all LedPin's mode to output and initial level to High(3.3v)
    for i in rgbPins:
        GPIO.setup(rgbPins[i], GPIO.OUT, initial=GPIO.HIGH)

    # Set all led as pwm channel and frequece to 2KHz
    p_R = GPIO.PWM(rgbPins['Red'], 2000)
    p_G = GPIO.PWM(rgbPins['Green'], 2000)
    p_B = GPIO.PWM(rgbPins['Blue'], 2000)

    # Set all begin with value 0
    p_R.start(0)
    p_G.start(0)
    p_B.start(0)
```



Código

- PIR

```
# Define a MAP function for mapping values. Like from 0~255 to 0~100
def MAP(x, in_min, in_max, out_min, out_max):
    return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min
```



Código

- **PIR**

```
# Define a function to set up colors
def setColor(color):
    # configures the three LEDs' luminance with the inputted color value .
    # Devide colors from 'color' variable
    R_val = (color & 0xFF0000) >> 16
    G_val = (color & 0x00FF00) >> 8
    B_val = (color & 0x0000FF) >> 0
    # Map color value from 0~255 to 0~100
    R_val = MAP(R_val, 0, 255, 0, 100)
    G_val = MAP(G_val, 0, 255, 0, 100)
    B_val = MAP(B_val, 0, 255, 0, 100)

    p_R.ChangeDutyCycle(R_val)
    p_G.ChangeDutyCycle(G_val)
    p_B.ChangeDutyCycle(B_val)
```



Código

- PIR

```
def loop():  
    while True:  
        pir_val = GPIO.input(pirPin)  
        if pir_val==GPIO.HIGH:  
            setColor(0xFFFF00)  
        else :  
            setColor(0x0000FF)
```



Código

- PIR

```
def destroy():  
    p_R.stop()  
    p_G.stop()  
    p_B.stop()  
    GPIO.cleanup()
```

Código

- PIR

```
if __name__ == '__main__':    # Program start from here
    setup()
    try:
        loop()
    except KeyboardInterrupt:  # When 'Ctrl+C' is pressed, the child program destroy()
will be executed.
        destroy()
```





Rogelio Ferreira Escutia

Profesor / Investigador
Tecnológico Nacional de México
Campus Morelia



rogelio.fe@morelia.tecnm.mx



rogeplus@gmail.com



xumarhu.net



[@rogeplus](https://twitter.com/rogeplus)



[https://www.youtube.com/
channel/UC0on88n3LwTKxJb8T09sGjg](https://www.youtube.com/channel/UC0on88n3LwTKxJb8T09sGjg)



[rogelioferreiraescutia](https://www.linkedin.com/in/rogelioferreiraescutia)

