

Interfacing with Sensors

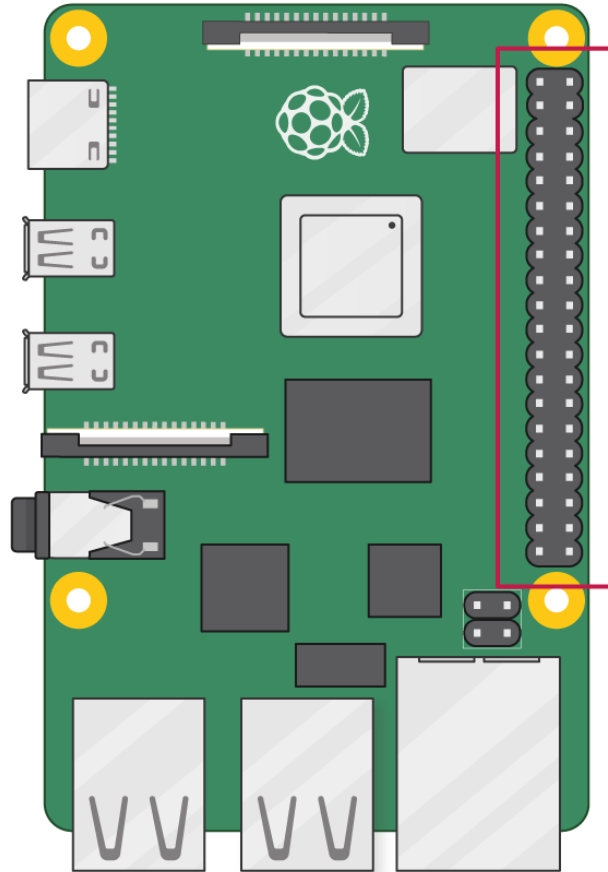
DAY 1 – SESSION 1

Software

Putty

VNC Server

Python 3.6



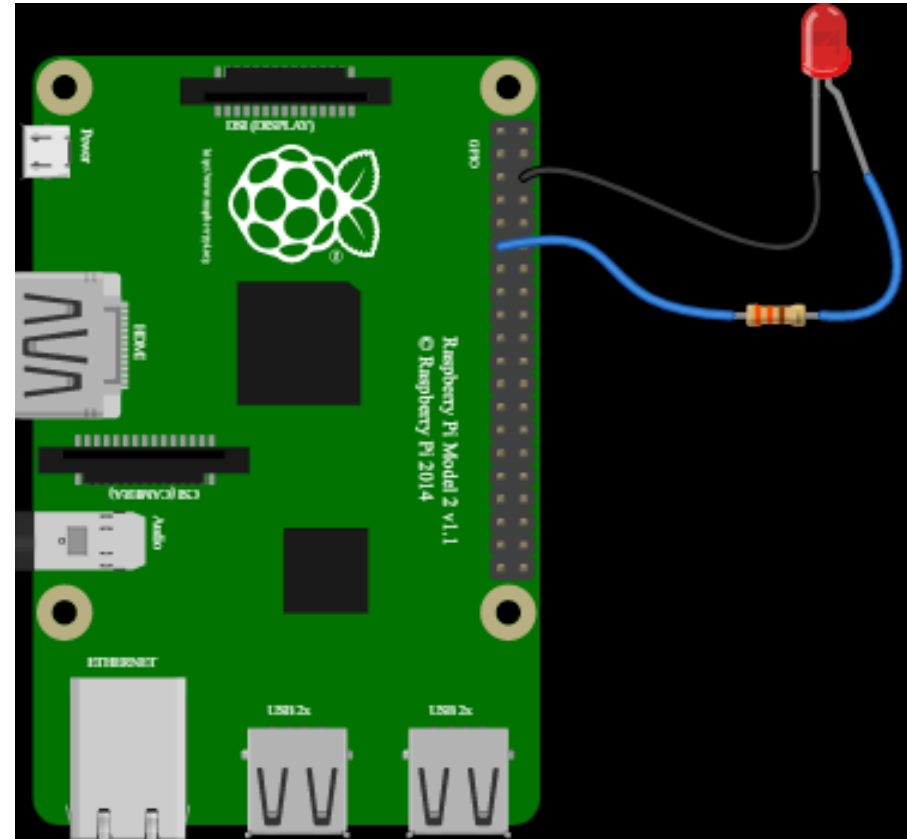
| | | | |
|------------------|----|----|--------------------|
| 3V3 power | 1 | 2 | 5V power |
| GPIO 2 (SDA) | 3 | 4 | 5V power |
| GPIO 3 (SCL) | 5 | 6 | Ground |
| GPIO 4 (GCLK0) | 7 | 8 | GPIO 14 (TXD) |
| Ground | 9 | 10 | GPIO 15 (RXD) |
| GPIO 17 | 11 | 12 | GPIO 18 (PCM_CLK) |
| GPIO 27 | 13 | 14 | Ground |
| GPIO 22 | 15 | 16 | GPIO 23 |
| 3V3 power | 17 | 18 | GPIO 24 |
| GPIO 10 (MOSI) | 19 | 20 | Ground |
| GPIO 9 (MISO) | 21 | 22 | GPIO 25 |
| GPIO 11 (SCLK) | 23 | 24 | GPIO 8 (CE0) |
| Ground | 25 | 26 | GPIO 7 (CE1) |
| GPIO 0 (ID_SD) | 27 | 28 | GPIO 1 (ID_SC) |
| GPIO 5 | 29 | 30 | Ground |
| GPIO 6 | 31 | 32 | GPIO 12 (PWM0) |
| GPIO 13 (PWM1) | 33 | 34 | Ground |
| GPIO 19 (PCM_FS) | 35 | 36 | GPIO 16 |
| GPIO 26 | 37 | 38 | GPIO 20 (PCM_DIN) |
| Ground | 39 | 40 | GPIO 21 (PCM_DOUT) |

LED

```
from gpiozero import LED
from time import sleep

led = LED(17)

while True:
    led.on()
    sleep(1)
    led.off()
    sleep(1)
```

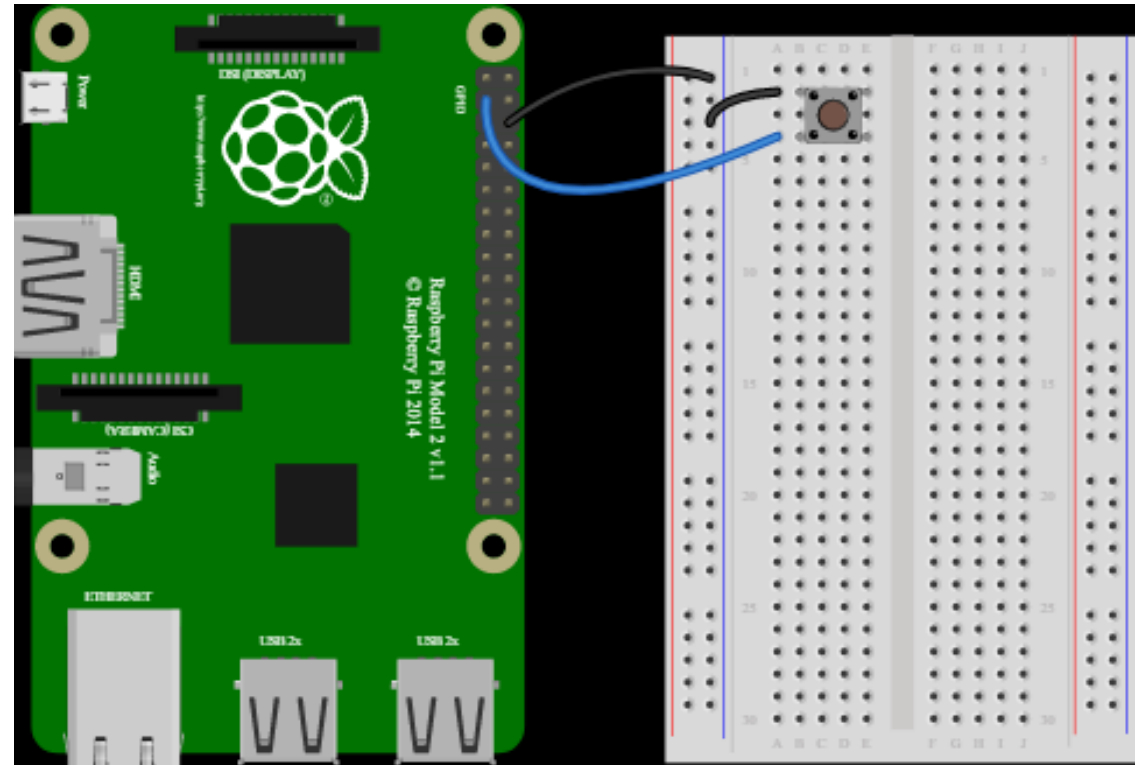


Button

```
from gpiozero import Button
from time import sleep

button = Button(2)

while True:
    if button.is_pressed:
        print("Pressed")
    else:
        print("Released")
    sleep(1)
```



Button + LED

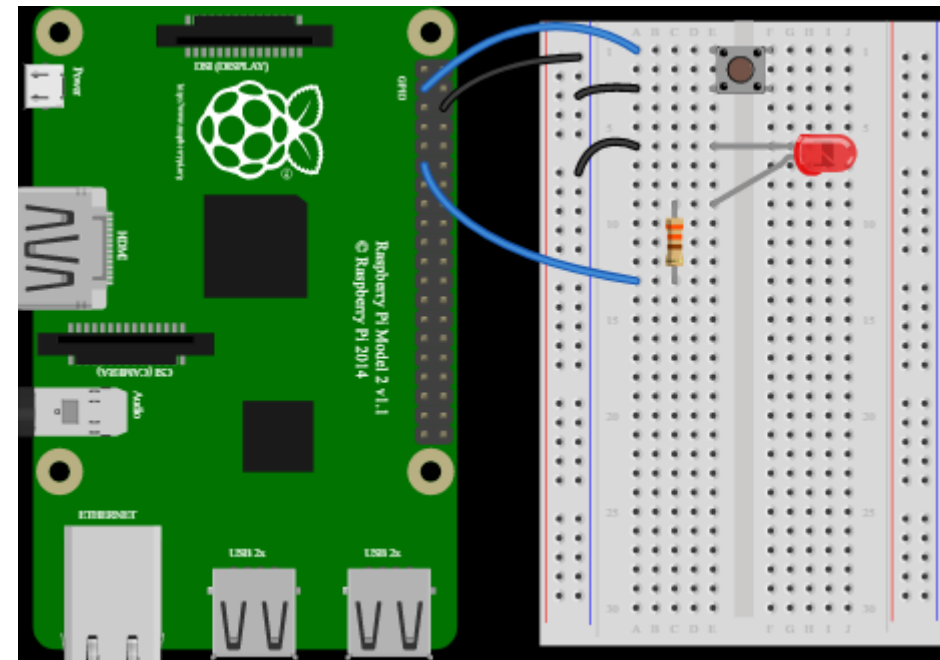
```
from gpiozero import LED, Button
```

```
led = LED(17)
```

```
button = Button(2)
```

```
button.when_pressed = led.on
```

```
button.when_released = led.off
```



Light Sensor

```
from gpiozero import LightSensor
```

```
sensor = LightSensor(21)
```

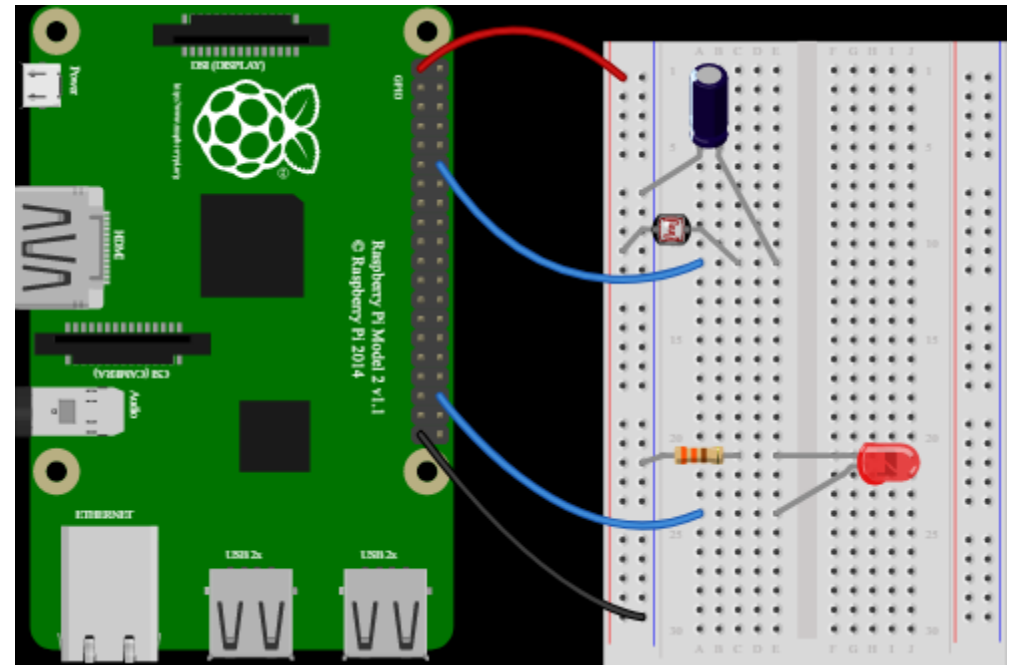
```
while True:
```

```
    sensor.wait_for_light()
```

```
    print("Its light")
```

```
    sensor.wait_for_dark()
```

```
    print("Its dark")
```



Connect one leg of the LDR to the 3V3 pin; connect positive leg of a 1µF capacitor to a ground pin; connect the other leg of the LDR and the negative leg of the capacitor to the same GPIO pin

LineSensor/IR Sensor

```
from gpiozero import LineSensor
```

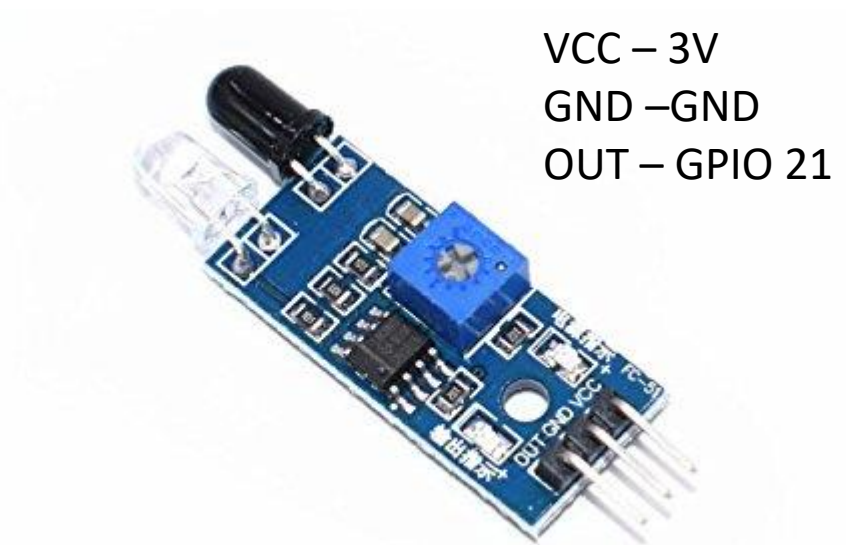
```
from signal import pause
```

```
sensor = LineSensor(21)
```

```
sensor.when_line = lambda: print('Object  
Detected')
```

```
sensor.when_no_line = lambda: print('Object  
Not Detected')
```

```
pause()
```



LineSensor + LED

```
from gpiozero import LineSensor,LED
```

```
from signal import pause
```

```
Led = LED(4)
```

```
sensor = LineSensor(21)
```

```
sensor.when_line = led.on
```

```
sensor.when_no_line = led.off
```

```
pause()
```



VCC – 3V
GND – GND
OUT – GPIO 21

+



GND – GND
OUT – GPIO 4

Distance Sensor

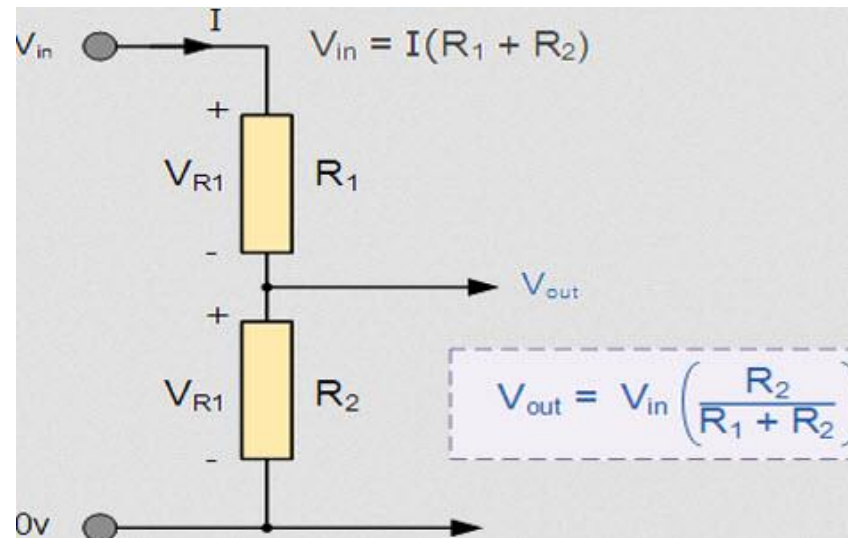
The distance sensor requires two GPIO pins: one for the *trigger* (marked TRIG on the sensor) and another for the *echo* (marked ECHO on the sensor).

However, a voltage divider is required to ensure the 5V from the ECHO pin doesn't damage the Pi.

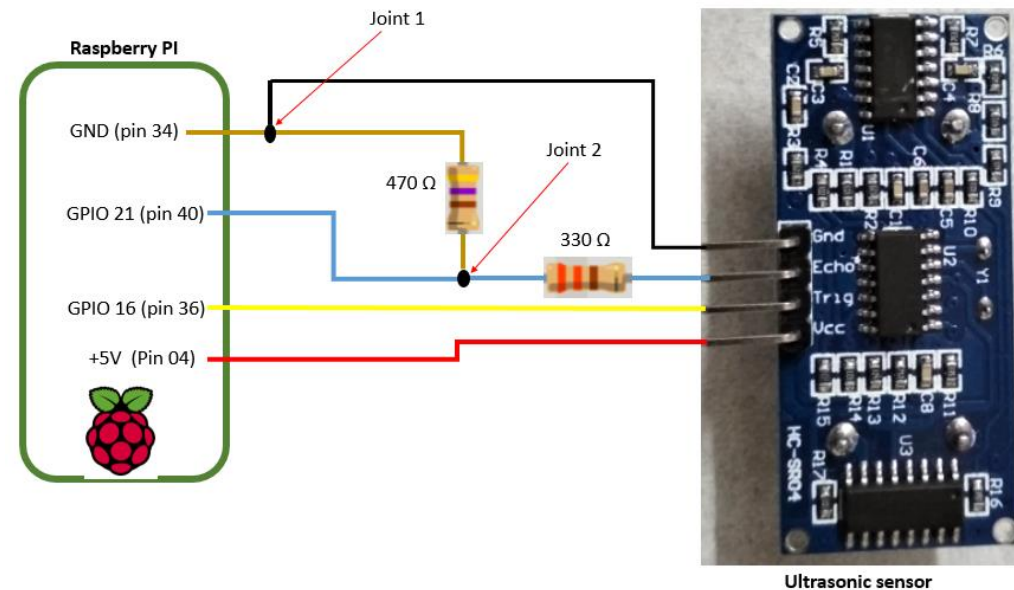
$$R_1 = 470 \, \Omega$$

$$R_2 = 330 \, \Omega$$

$$V = 5(470/(470+330)) = 2.93\text{v}$$



1. Connect the GND pin of the sensor to a ground pin on the Pi.
2. Connect the TRIG pin of the sensor a GPIO pin.
3. Connect one end of a 330Ω resistor to the ECHO pin of the sensor.
4. Connect one end of a 470Ω resistor to the GND pin of the sensor.
5. Connect the free ends of both resistors to another GPIO pin. This forms the required **voltage divider**.
6. Finally, connect the VCC pin of the sensor to a 5V pin on the Pi.



Distance Sensor

```
from gpiozero import DistanceSensor
```

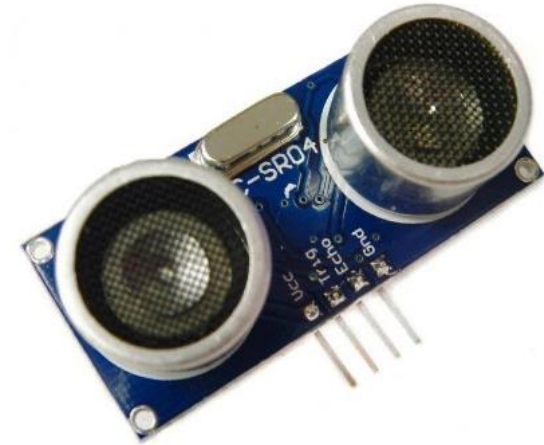
```
from time import sleep
```

```
sensor = DistanceSensor(echo=21, trigger=16)
```

```
while True:
```

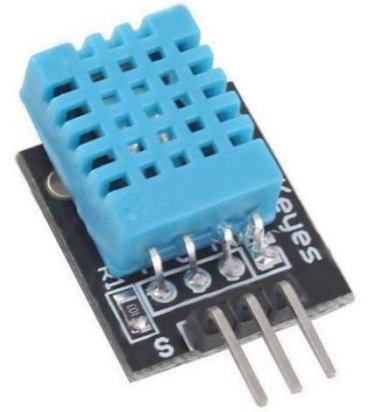
```
    print('Distance: ', sensor.distance * 100)
```

```
    sleep(1)
```



DHT11

pip3 install dht11



```
import RPi.GPIO as GPIO
import dht11

# initialize GPIO
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.cleanup()

# read data using pin 14
instance = dht11.DHT11(pin = 14)
result = instance.read()
```

```
if result.is_valid():
    print("Temperature: %-.3f C" %
          result.temperature)
    print("Humidity: %-.3f %" %
          result.humidity)
else:
    print("Error: %d" % result.error_code)
```

RFID

1. Enable SPI Interface

2. Check if spi_bcm2835 is loaded

`lsmod | grep spi`

3. `sudo pip3 install spidev`

4. `sudo pip3 install mfrc522`

RFID ---- Pi Physical Pins

SDA <--> 24

SCK <--> 23

MOSI <--> 19

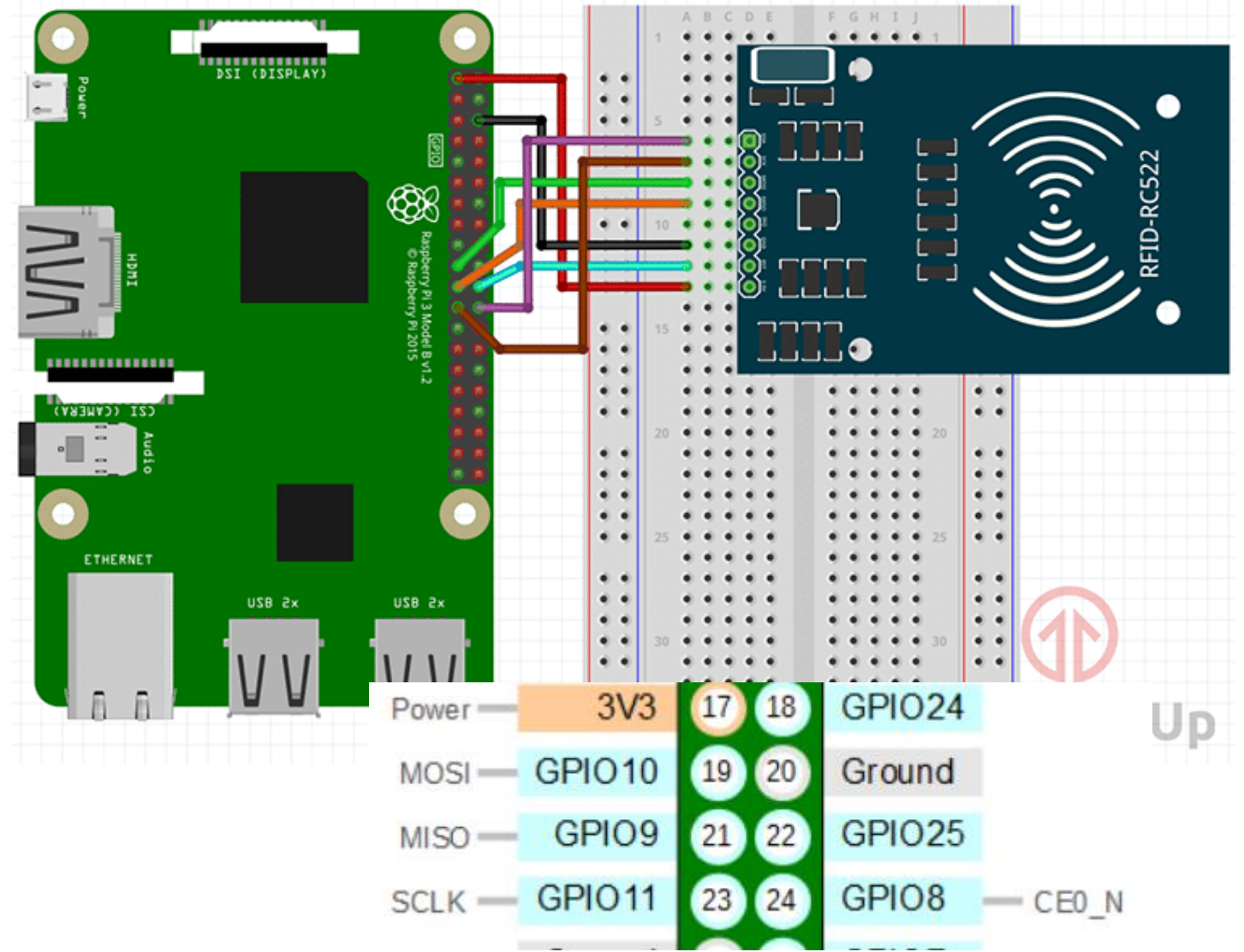
MISO <--> 21

IRQ <--> UNUSED

GND <--> 6

RST <--> 22

3.3V <--> 1



```
import RPi.GPIO as GPIO
from mfrc522 import SimpleMFRC522

reader = SimpleMFRC522()

try:
    text = input('New data:')
    print("Now place your tag to write")
    reader.write(text)
    print("Written")
finally:
    GPIO.cleanup()
```

Write a Tag

```
sudo python3 Write.py
```

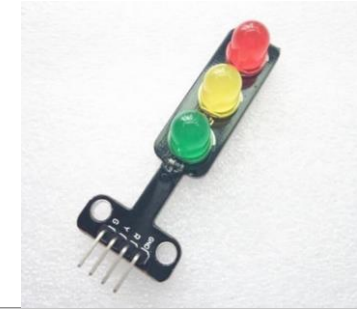

Read a Tag

```
import RPi.GPIO as GPIO
from mfrc522 import SimpleMFRC522
reader = SimpleMFRC522()
try:
    id, text = reader.read()
    print(id)
    print(text)
finally:
    GPIO.cleanup()
```

`sudo python3 Read.py`

<https://pimylifeup.com/raspberry-pi-rfid-rc522/>

Traffic Lights



```
from gpiozero import TrafficLights
```

```
from time import sleep
```

```
lights = TrafficLights(2, 3, 4)
```

```
lights.green.on()
```

```
while True:
```

```
    sleep(10)
```

```
    lights.green.off()
```

```
    lights.amber.on()
```

```
    sleep(1)
```

```
    lights.amber.off()
```

```
    lights.red.on()
```

```
    sleep(10)
```

```
    lights.amber.on()
```

```
    sleep(1)
```

```
    lights.green.on()
```

```
    lights.amber.off()
```

```
    lights.red.off()
```

Traffic LED's

```
from gpiozero import LED
```

```
from time import sleep
```

```
red = LED(2)
```

```
amber = LED(3)
```

```
green = LED(4)
```

```
green.on()
```

```
amber.off()
```

```
red.off()
```

Relay



```
from gpiozero import DigitalOutputDevice
```

```
relay = DigitalOutputDe
```

```
relay.on()
```

```
relay.off()
```

or

```
R.value = 1 # On
```

```
R.value = 0 # Off
```

