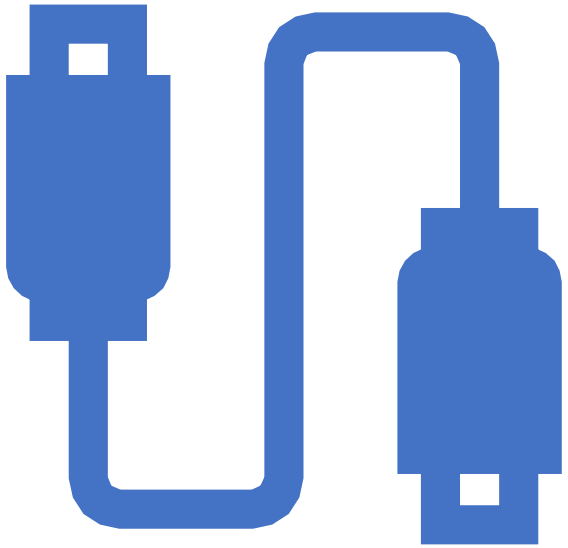
# Internet of Things

Rpi – Usage & GPIO Demo



# Getting Started

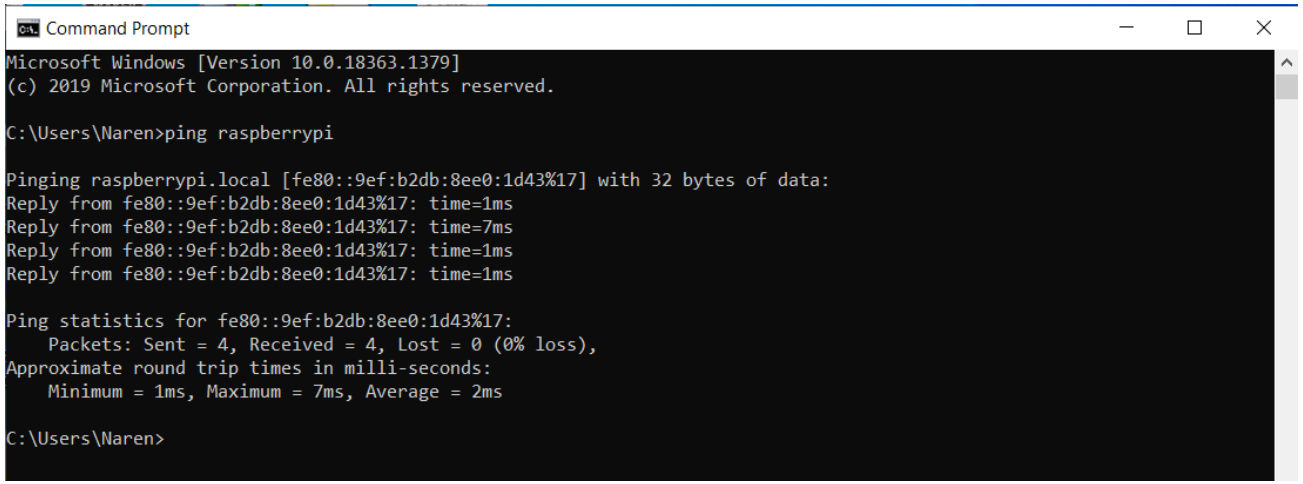
- Hardware Requirements:
  - 1 x Raspberry Pi Board(Any variant ).
  - 1 x USB Cable (Power Adapter/ Laptop USB Port)
  - 1 x 16 or 32 Gb SD card.

  - 1 Keyboard.
  - 1 Mouse.
  - 1 Monitor screen.

# Detect RPi on the network

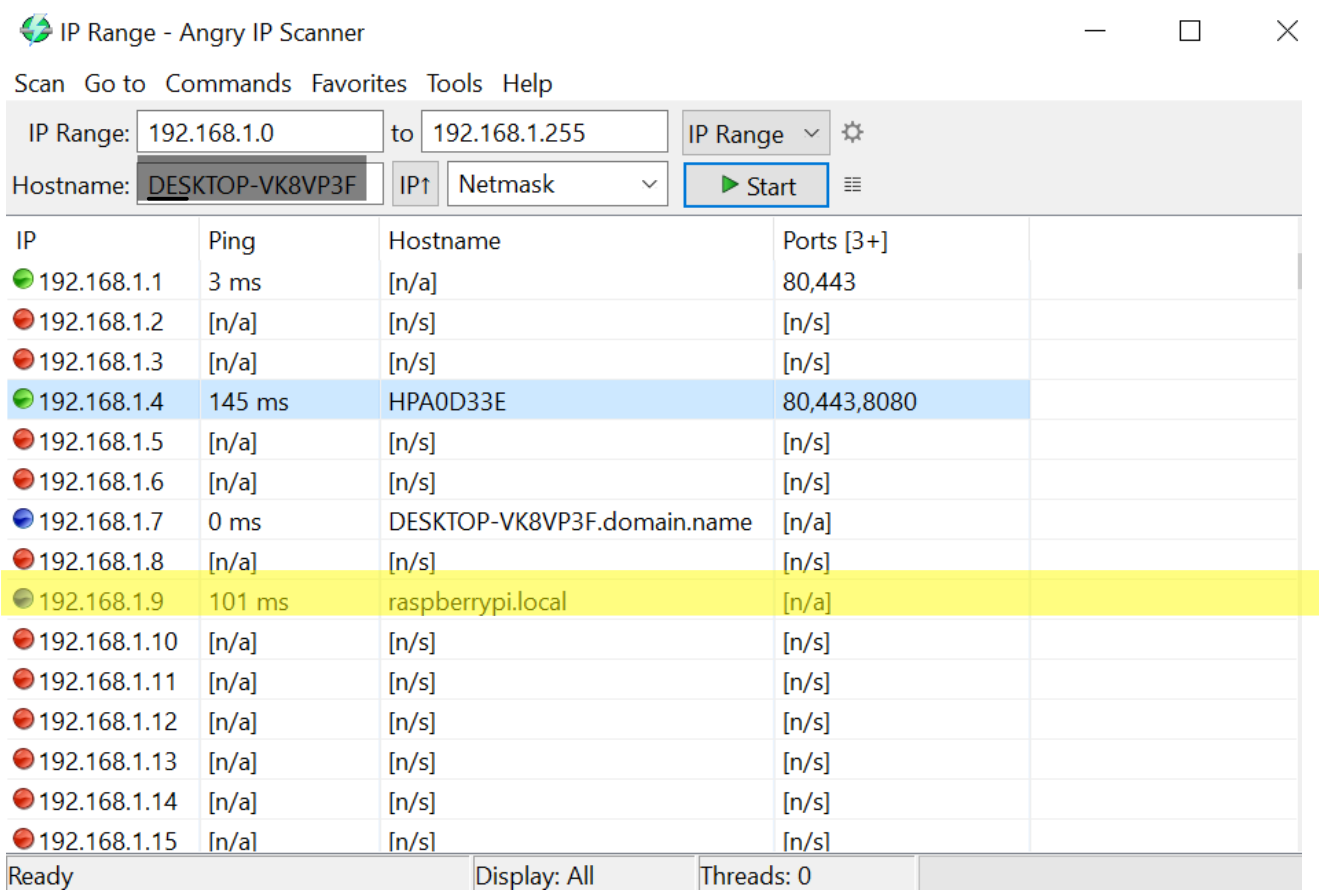
Open **Command Prompt terminal** and type “**ping raspberrypi**”.

**A valid response** it indicates that the raspberry pi and the Laptop are present in the same network.



# Finding IP Address – Angry IP Scanner

- After receiving conformation about the availability of the device, the IP address of the device must be determined to establish the virtual connection.
- For this purpose we can use **“Angry IP Scanner”** software , which scans the entire subnet to determine various devices which are connected to the network.
- The highlighted portion in yellow is the IP address of the Raspberry-Pi device.

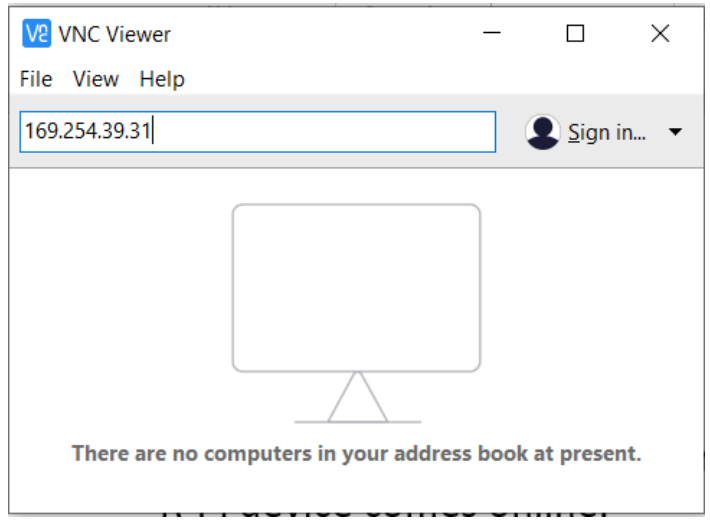


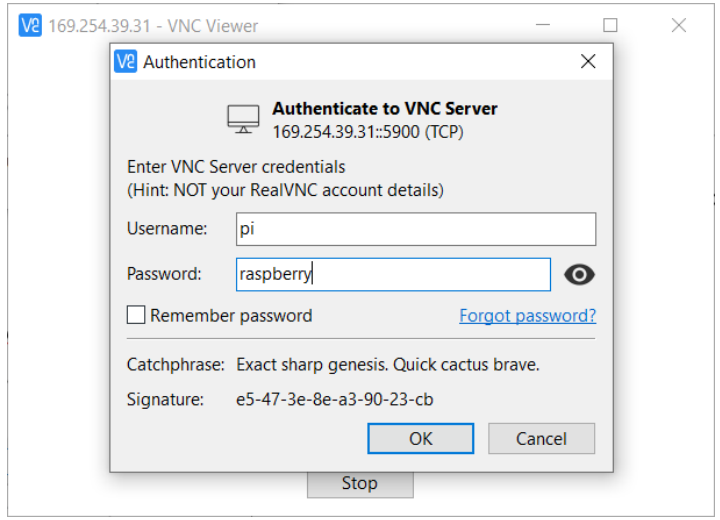
# Connecting through VNC

- Open the **VNC Viewer** on your computer, enter the IP address of the RPi and press Enter on your keyboard.
- Once it connects, enter the correct login credentials to start using the RPi

Username: ***pi***

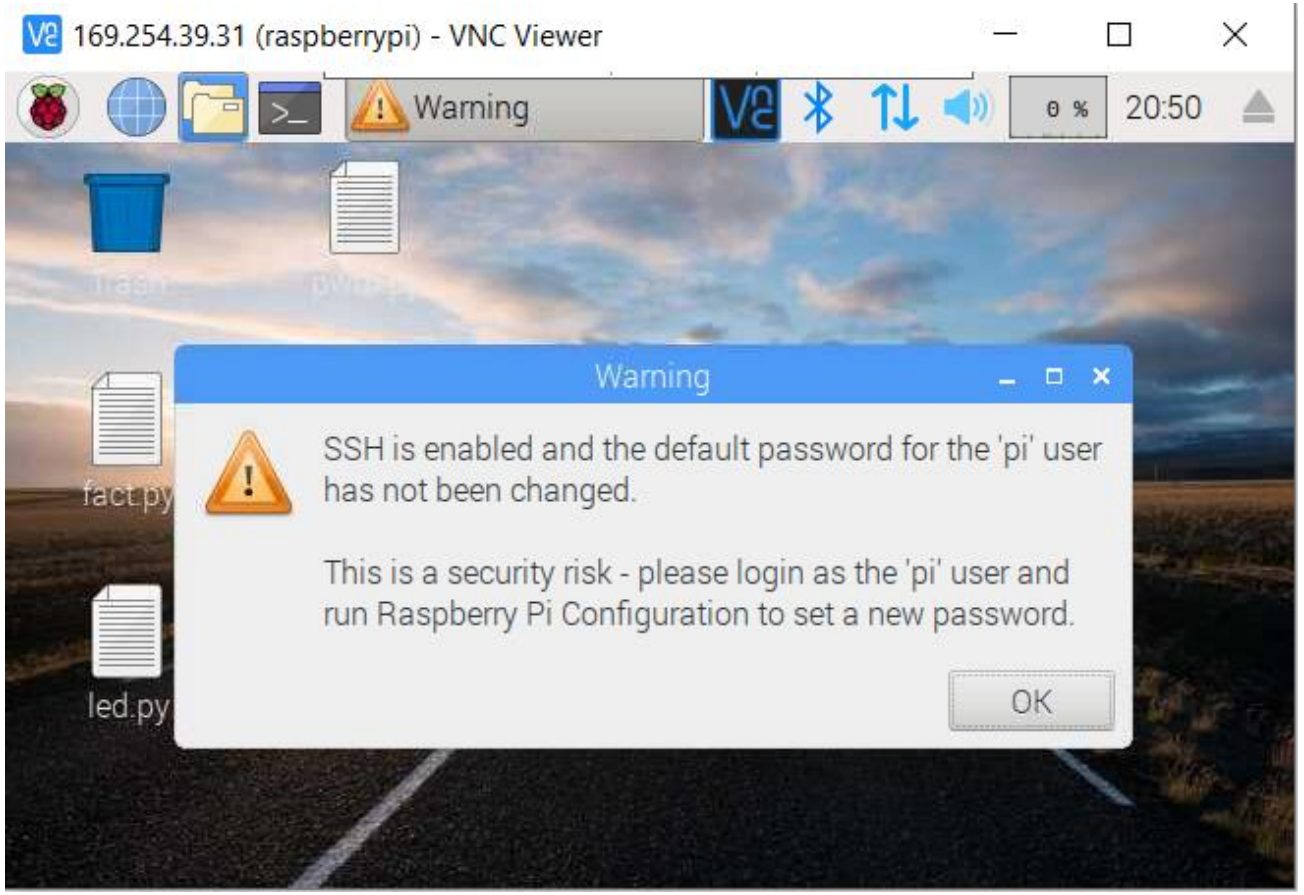
Password: ***raspberrypi***



# Accessing R-Pi

- After Successful login, the R-Pi device comes online.

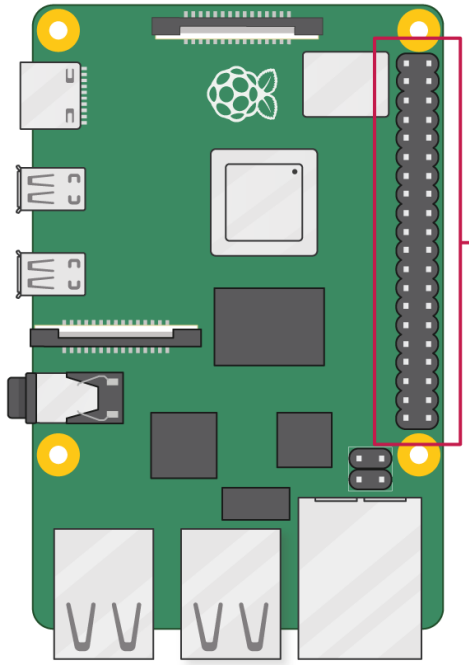


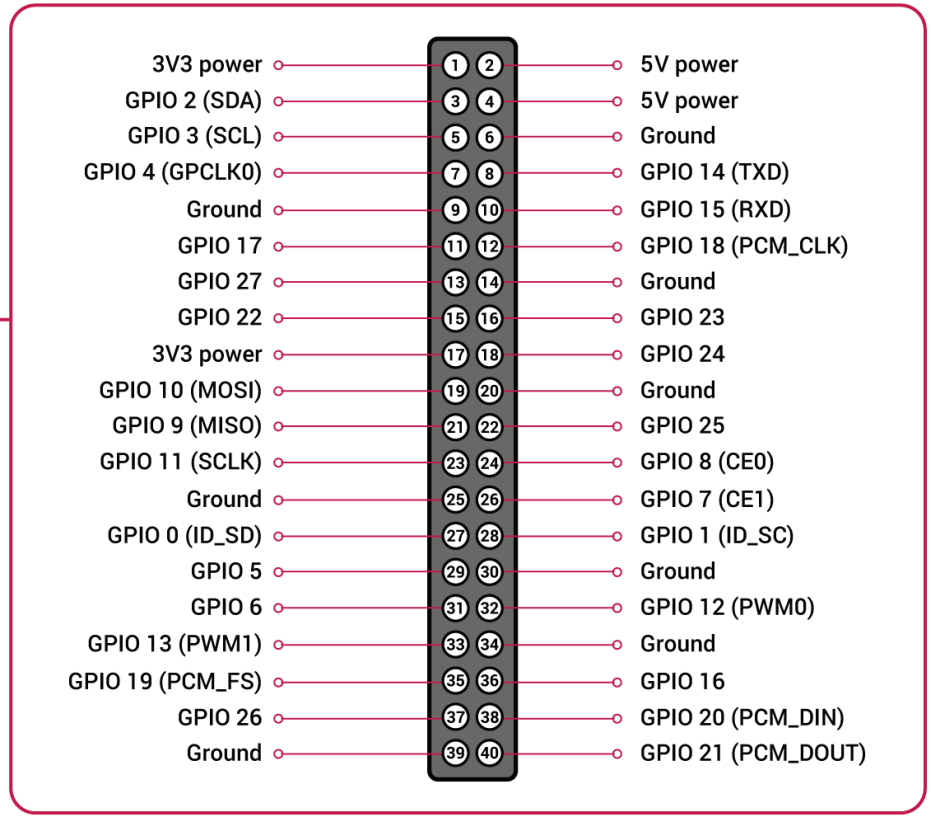
# Demo-I

Configuring GPIO and Blinking LED

# Connections

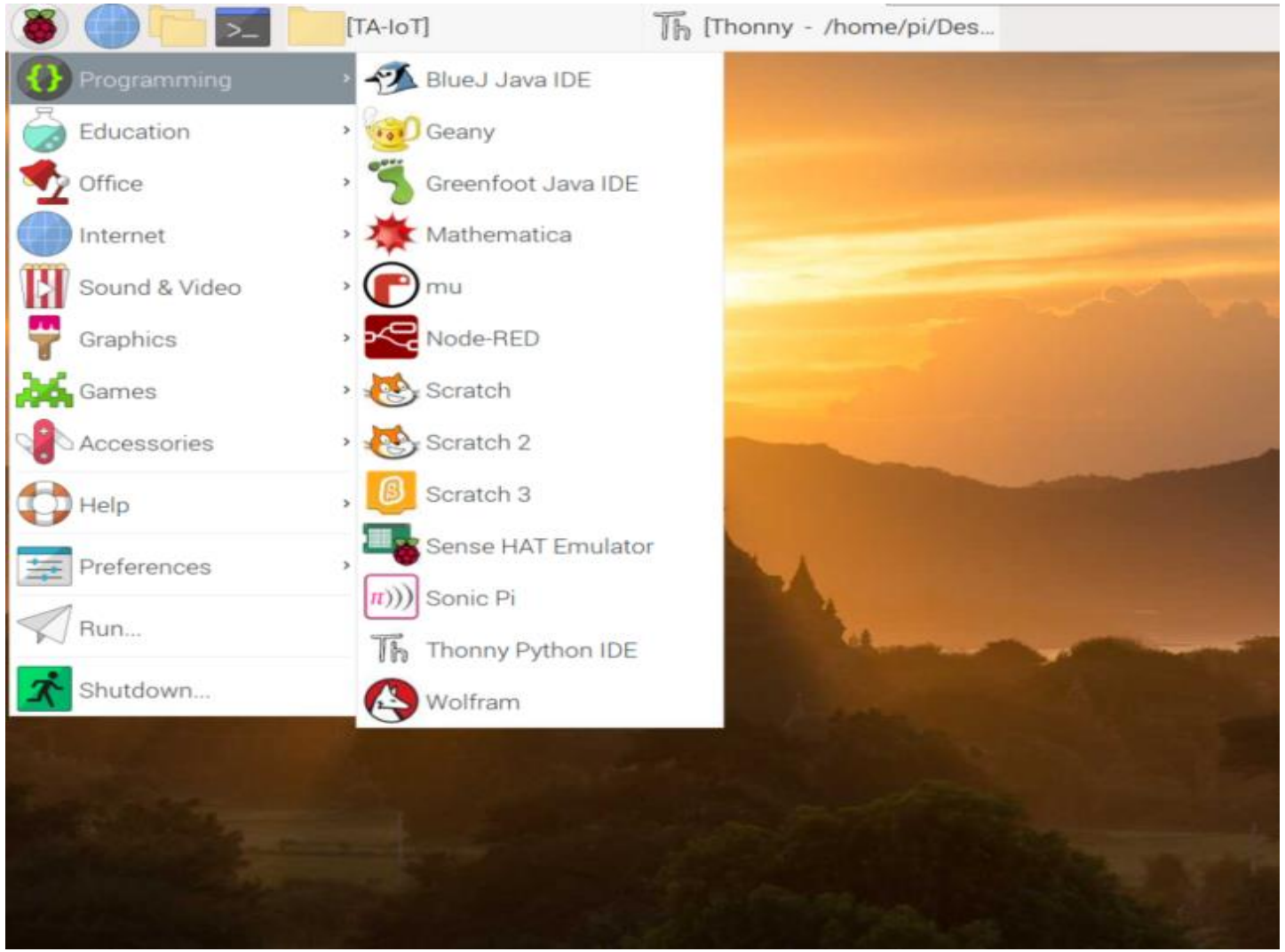
- LED Pos -> Pin 11 (GPIO17)
- Led Neg -> Ground (Pin 6)





# DEMO -1 : Blinking LED with GPIO

Open Thonny Python IDE from start menu.



# Code: Blinking LED with GPIO

LED_simple_new.py

```
1  import RPi.GPIO as GPIO
2  import time
3  LedPin = 11 # Corresponds to GPIO 17
4  GPIO.setmode(GPIO.BOARD) #Numbers GPIO by Physical Location
5  GPIO.setup(LedPin ,GPIO.OUT) #Set LedPin's to Output mode
6  GPIO.output(LedPin, GPIO.HIGH) #Set LedPin HIGH (3.3V) to turn on LED
7  try:
8      while True:
9          GPIO.output(LedPin, GPIO.HIGH)
10         time.sleep(0.5)
11         GPIO.output(LedPin, GPIO.LOW)
12         time.sleep(0.5)
13 except KeyboardInterrupt:
14     GPIO.output(LedPin, GPIO.LOW)
15     GPIO.cleanup()  #Release Resource
```

# Code: Varying LED Brightness with PWM

GPIO_PWM.py

```
1  import RPi.GPIO as GPIO
2  import time
3  GPIO.setmode(GPIO.BOARD)
4  GPIO.setup(12,GPIO.OUT)
5  pwm12 = GPIO.PWM(12,50)
6  pwm12.start(0)
7  try:
8      while True:
9          for dutyCycle in range (0,100,5):
10             pwm12.ChangeDutyCycle(dutyCycle)
11             time.sleep(0.05)
12         for dutyCycle in range(100,0,-5):
13             pwm12.ChangeDutyCycle(dutyCycle)
14             time.sleep(0.05)
15 except KeyboardInterrupt:
16     pwm12.stop()
17 GPIO.cleanup()
```

# Internet of Things

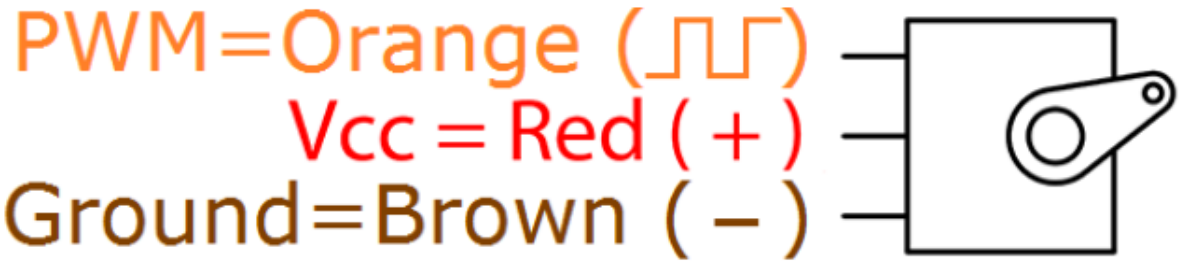
Servo Motor

# Servo Motor

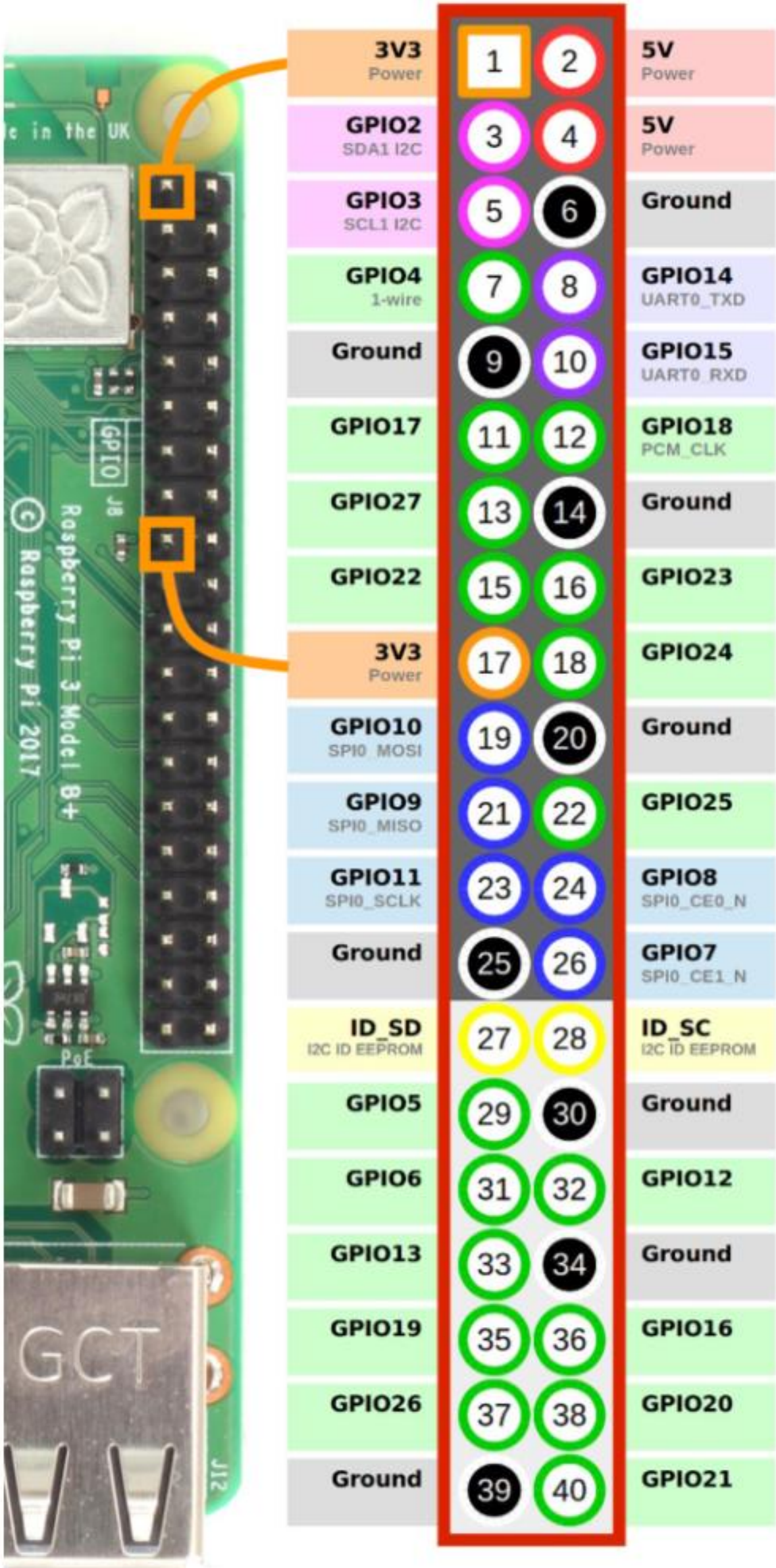
- A servo motor is a type of a DC motor that, upon receiving a signal of certain frequency , can rotate itself to any angle from 0- 180 degrees.

- Its 90 degree position is generally referred to as neutral position because it can rotate equally from that position.

# Servo Motor Interfacing




- The servo motor has 3 pins :
  1. Power Pin (5V)
  2. Data Pin(3.3V)
  3. Ground Pin
- The power Pin is connected to the 5V power supply.
- The Data pin is interfaced with the GPIO pin of the R-Pi.
- The Gnd pin of the servo motor is connected with the Gnd pin of the R-Pi.
- The Data pin of the Servo motor is provided with PWM modulated voltage to control the rotation of the R-Pi.


3V3 Power
1
2
5V Power
GPIO2 SDA1 I2C
3
4
5V Power
GPIO3 SCL1 I2C
5
6
Ground
GPIO4 1-wire
7
8
GPIO14 UART0_TXD
Ground
9
10
GPIO15 UART0_RXD
GPIO17
11
12
GPIO18 PCM_CLK
GPIO27
13
14
Ground
GPIO22
15
16
GPIO23
3V3 Power
17
18
GPIO24
GPIO10 SPI0_MOSI
19
20
Ground
GPIO9 SPI0_MISO
21
22
GPIO25
GPIO11 SPI0_SCLK
23
24
GPIO8 SPI0_CE0_N
Ground
25
26
GPIO7 SPI0_CE1_N
ID_SD I2C ID EEPROM
27
28
ID_SC I2C ID EEPROM
GPIO5
29
30
Ground
GPIO6
31
32
GPIO12
GPIO13
33
34
Ground
GPIO19
35
36
GPIO16
GPIO26
37
38
GPIO20
Ground
39
40
GPIO21
GPIO
JB
Raspberry Pi 3 Model B+
© Raspberry Pi 2017
PoE
GCT
J12

# PWM Calculation – Using Datasheet Information

- Rotational Range: 180°
- Pulse Cycle: ca. 20 ms
- Pulse Width: 500-2400 µs

- 500 µs = 0.5 ms = 2.5% of 20ms
- 2400 µs = 2.4 ms = 12% of 20ms

# Code: Servo Motor Control

Servo.py

```python
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BOARD)
servoPin = 12
GPIO.setup(servoPin,GPIO.OUT)
pwm_servo = GPIO.PWM(servoPin,50)
pwm_servo.start(0)

try:
    while True:
        pwm_servo.ChangeDutyCycle(2.5)
        time.sleep(3)
        pwm_servo.ChangeDutyCycle(5)
        time.sleep(3)
        pwm_servo.ChangeDutyCycle(7.5)
        time.sleep(3)
        pwm_servo.ChangeDutyCycle(10)
        time.sleep(3)
        pwm_servo.ChangeDutyCycle(12)
        time.sleep(3)
        pwm_servo.ChangeDutyCycle(10)
        time.sleep(3)
        pwm_servo.ChangeDutyCycle(7.5)
        time.sleep(3)
        pwm_servo.ChangeDutyCycle(5)
        time.sleep(3)
        pwm_servo.ChangeDutyCycle(2.5)
        time.sleep(3)
except KeyboardInterrupt:
    pwm_servo.stop()
GPIO.cleanup()
```

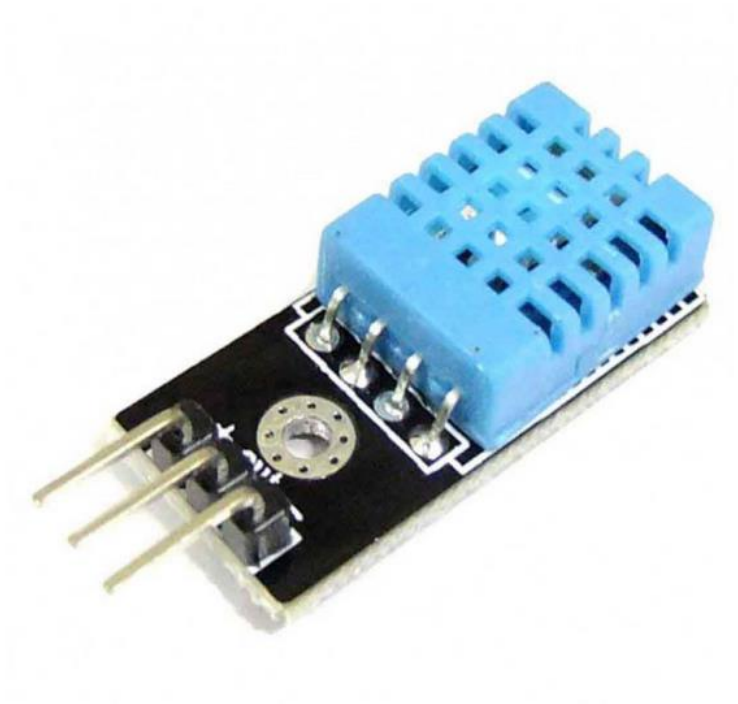
# Internet of Things

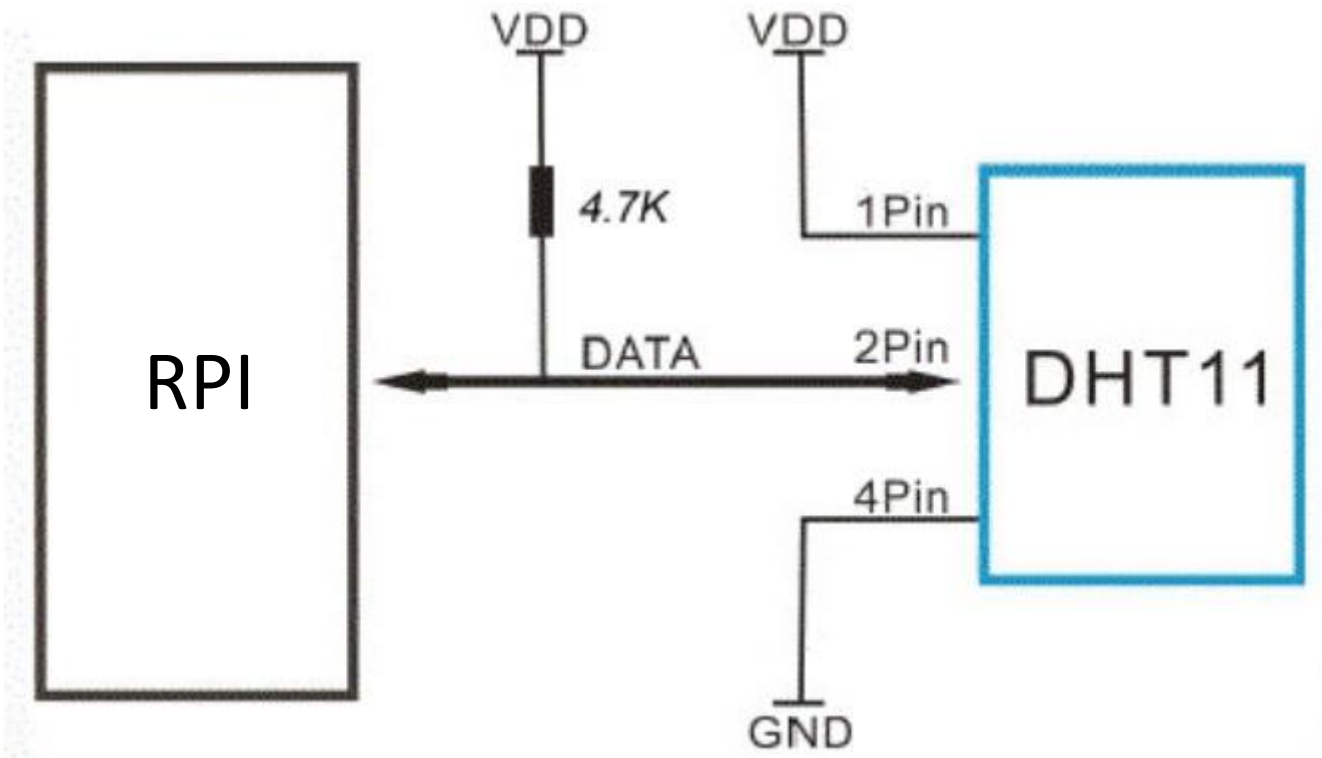
## Interfacing DHT 11 (Temp & Humidity Sensor)

# Temperature and Humidity Sensor

- The DHT 11 is a temperature and a Humidity sensor that provides **digital temperature and humidity readings.**
- Popular for use in **remote weather stations, soil monitors and home automation systems**.

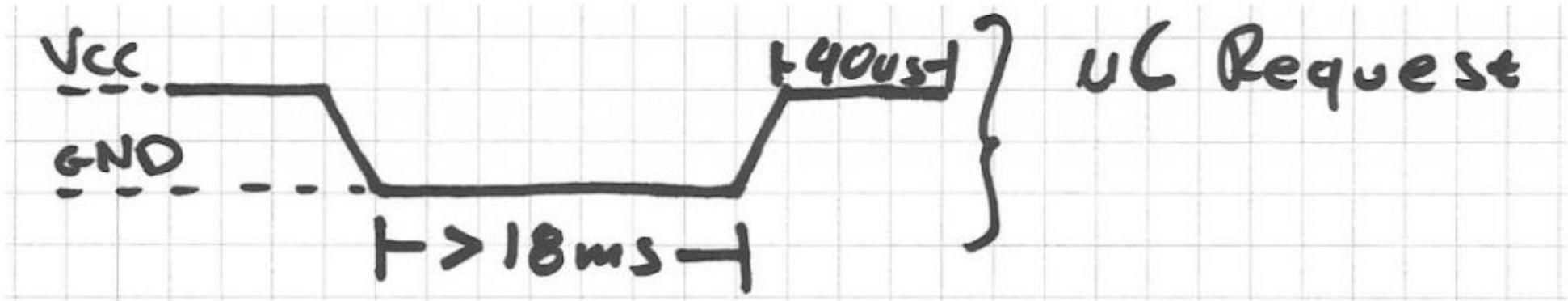
# Connections





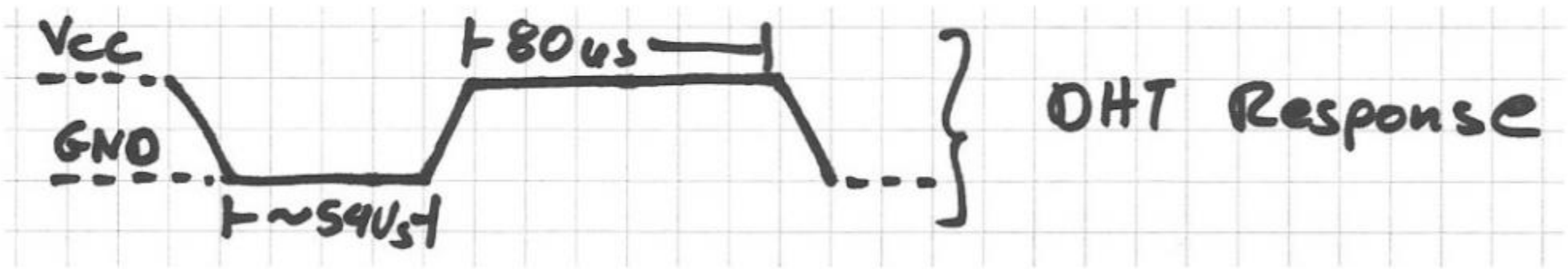
# 1. Request Data

1) **Request:** To make the DHT-11 to send you the sensor readings you have to send it a request. The request is, to pull down the bus for more than **18ms** in order to give DHT time to understand it and then pull it up for **40uS**.



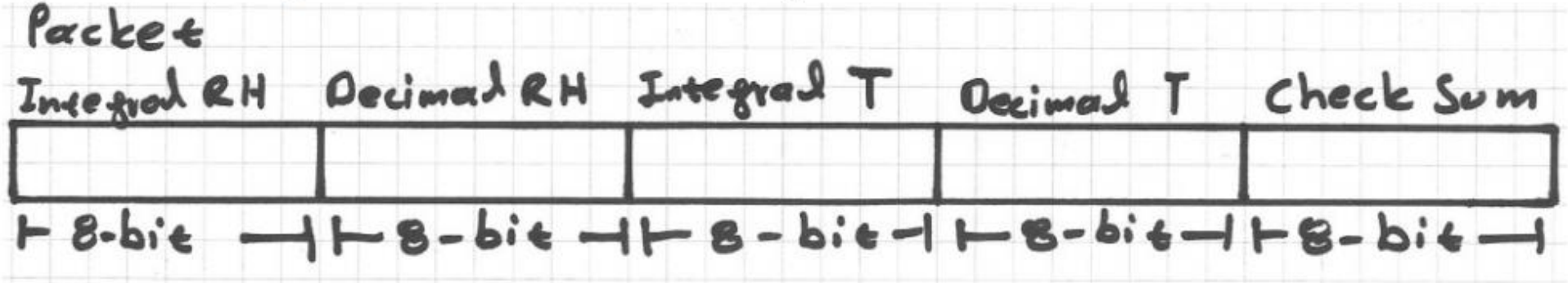
# Response from DHT11

2) **Response:** What comes after the request is the DHT-11 response. This is an automatic reply from DHT which indicates that DHT received your request. The response is ~54uS low and 80uS high.



# 3. Data Reading

3) **Data Reading**: What will come after the response is the sensor data. The data will be packed in a packet of 5 segments of 8-bits each. Totally 5×8 =40bits.



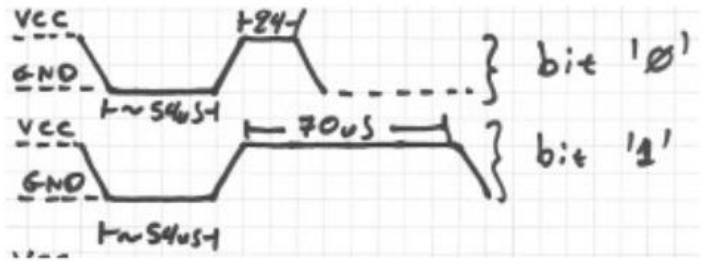

First two segments are Humidity read, integral & decimal. Following two are Temperature read in Celsius, integral & decimal and the last segment is the Check Sum which is the sum of the 4 first segments. If Check Sum's value isn't the same as the sum of the first 4 segments that means that data received isn't correct.

# Identifying Data Bits

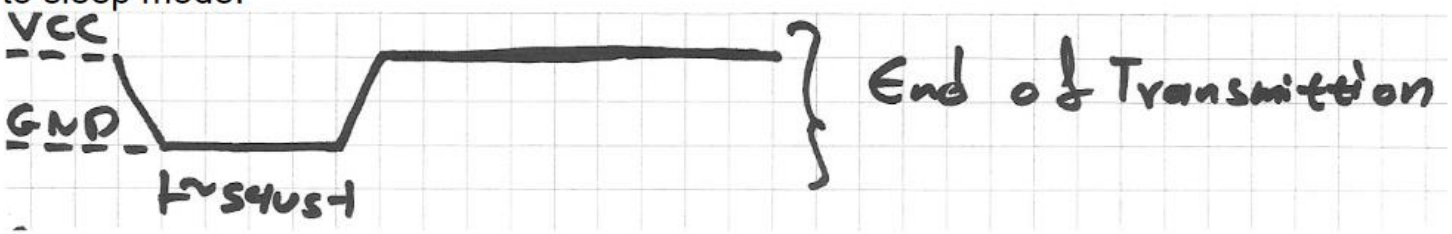
**How to Identify Bits:** Each bit sent is a follow of ~54uS Low in the bus and ~24uS to 70uS High depending on the value of the bit.

Bit '0' : ~54uS Low and ~24uS High

Bit '1' : ~54uS Low and ~70uS High




**End Of Frame:** At the end of packet DHT sends a ~54uS Low level, pulls the bus to High and goes to sleep mode.



# Installing a DHT Library

## Software Setup

To start with update your package lists and install a few Python libraries :

```
sudo apt-get update
sudo apt-get install build-essential python-dev
```

Then clone the Adafruit library from their repository :

```
git clone https://github.com/adafruit/Adafruit_Python_DHT.git
cd Adafruit_Python_DHT
```

Then install the library for Python 2 and Python 3 :

```
sudo python setup.py install
sudo python3 setup.py install
```

Hopefully at this point the library is installed and ready to be used within a Python script.

# Code: Interfacing DHT11 Sensor

DHT_11.py

```python
import Adafruit_DHT
DHT11_sensor = Adafruit_DHT.DHT11 #DHT11 Instance

# Data Pin Connected to GPIO 17 (Pin 11 on Raspberry Pi)
gpio=17

# Use read_retry method. This will retry up to 15 times to
# get a sensor reading (waiting 2 seconds between each retry).
humidity, temperature = Adafruit_DHT.read_retry(DHT11_sensor, gpio)
# Reading the DHT11 is very sensitive to timings and occasionally
# the Pi might fail to get a valid reading. So check if readings are valid.
if humidity is not None and temperature is not None:
  print('Temp={0:0.1f}*C  Humidity={1:0.1f}%'.format(temperature, humidity))
else:
  print('Failed to get reading. Try again!')

```