Jyothy Institute Of Technology

Off Kanakpura Road, Tataguni, Bangalore – 560 062

DEPARTMENT OF COMPUTER SCIENCE & ENGG.

&

DEPARTMENT OF INFORMATION SCIENCE & ENGG.

Internet Of Things Lab Manual

Faculty Incharge

Vadiraja Acharya, Asst. Prof., Dept of ISE

Adithya B, Asst. Prof., Dept of CSE

Student Team

Shruthi Sagar, 6th Sem ISE

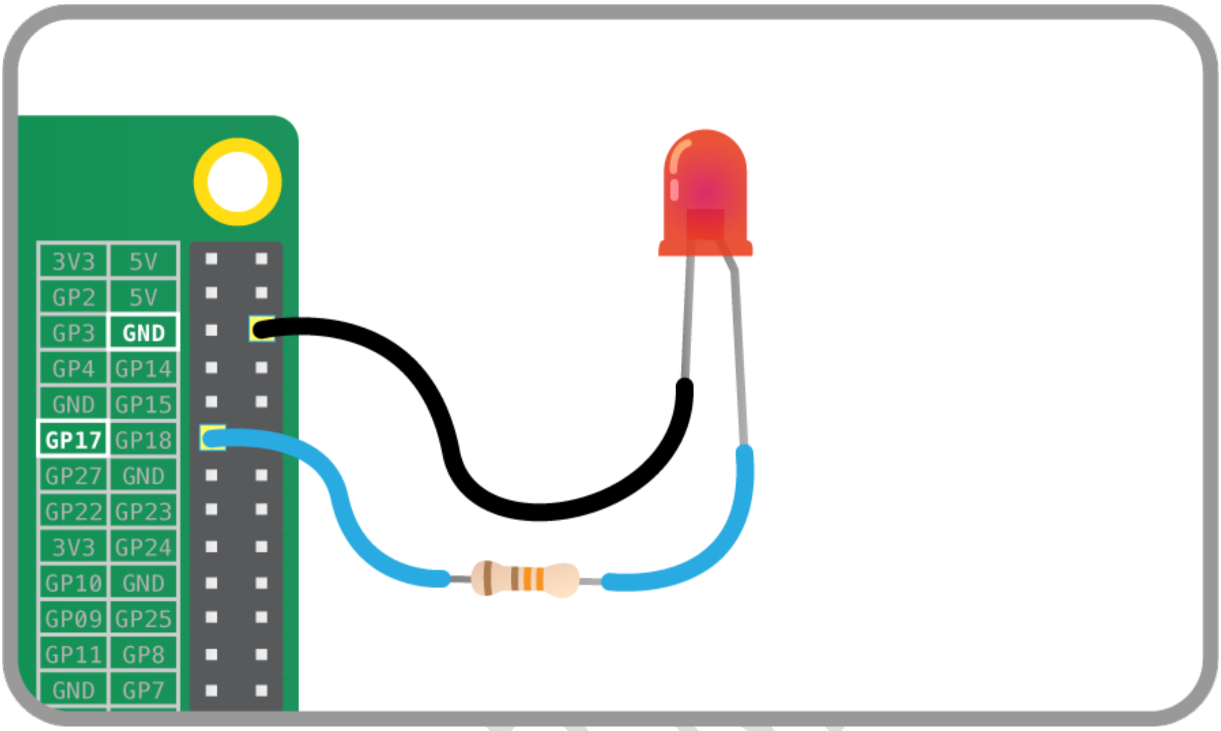
Perumal, 6th Sem CSE

Vinay Bhatt, 6th Sem CSE

Vinay Kiran, 6th Sem CSE

LED ON/OFF THROUGH SHELL

1. Make the circuit connection as shown.



2. Open the terminal and type **python.**
3. You can switch an LED on and off by typing commands directly into the Python interpretor window (also known as the Python shell). Let's do this by first, importing the GPIO library.

```
Type:

from RPi import GPIO (press ENTER)

GPIO.setmode(GPIO.BCM) (press ENTER)

GPIO.setup(17,GPIO.OUT) (press ENTER)
```

4. To make the LED switch on, type the following and press enter:

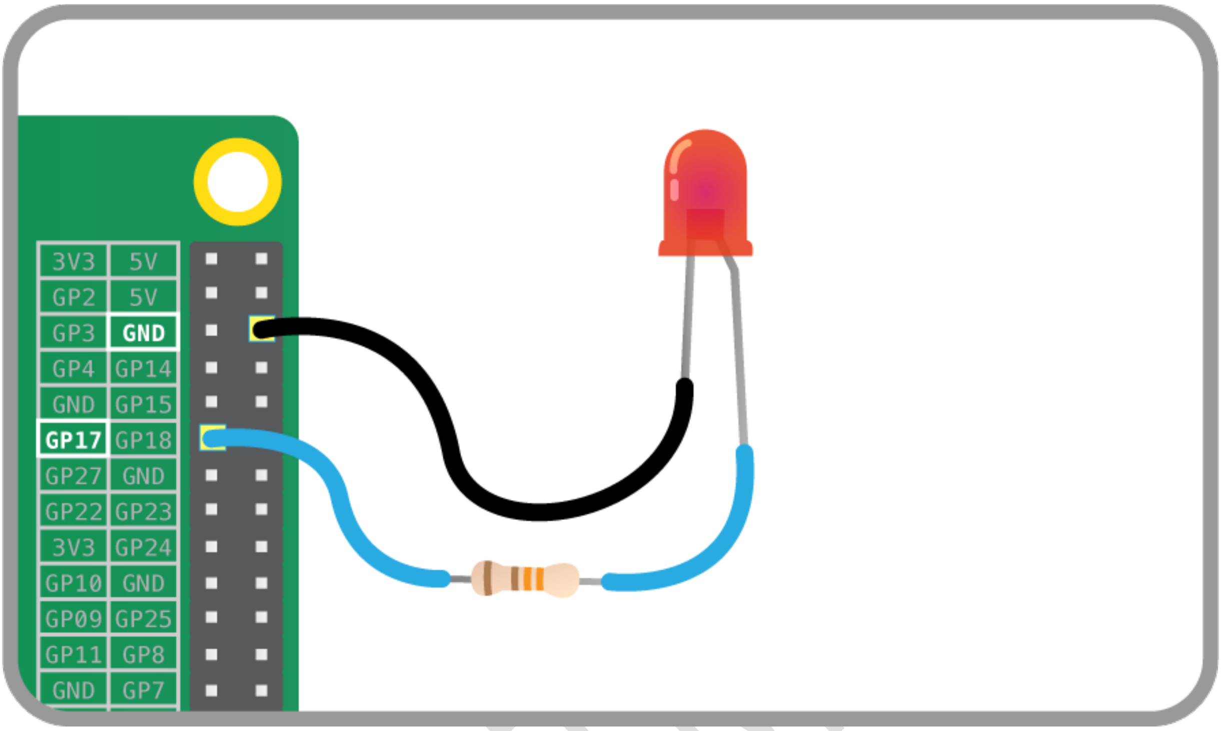
```
Type:

GPIO.output(17,1)
```

5. To make it switch off you can type:

```
Type:

GPIO.output(17,0)
```

LED BLINK

1. Make the circuit connection as shown.



2.Create a new file by clicking **File > New file.**

3. Save the file as `gpio_led.py` .

4. Enter the following code to get started:

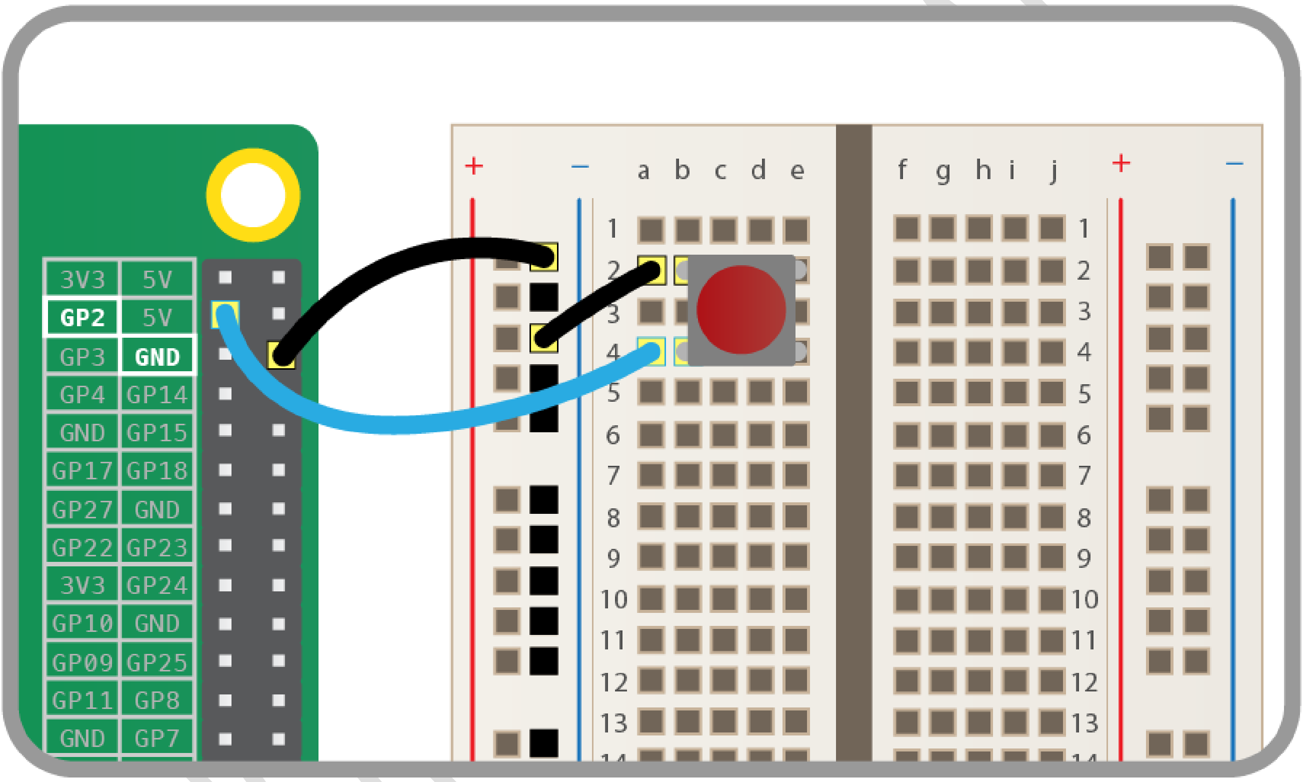
```
    from RPi import GPIO

from time import sleep

GPIO.setmode(GPIO.BCM)

GPIO.setup(25,GPIO.OUT)

while True:

        GPIO.output(25,1)

        sleep(1)

        GPIO.output(25,0)

        sleep(1)
```

5. Save with **Ctrl + S** and run the code with `sudo python gpio_led.py`

6. The LED should be flashing on and off.

Push Button

1. Make the circuit connection as shown.



2.Create a new file by clicking **File** > **New file.**

3**.** Save the file as `gpio_button.py`.

4. Enter the following code to get started:

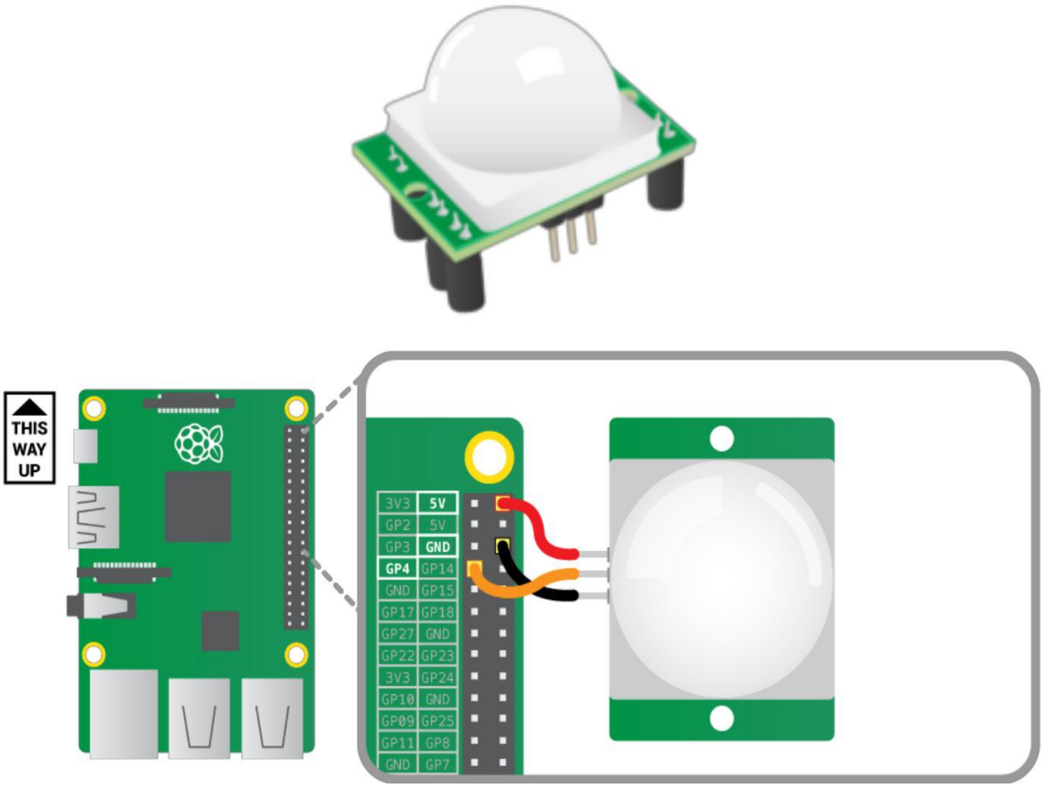
```
    import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BCM)



GPIO.setup(25,GPIO.IN,pull_up_down=GPIO.PUD_UP)
```

```
while True:
        input_state = GPIO.input(25)
        if input_state == False:
                print "Button Pressed"
                time.sleep(0.2)
```

5. Save with **Ctrl + S** and run the code with `sudo python gpio_button.py.`

Motion Detection Using Passive Infrared Motion Sensor

1.Set up the following circuit on Raspberry Pi.



2.Create a python file with the following code We have *`motion.py`*

```
import RPi.GPIO as GPIO import time

sensor = 4

GPIO.setmode(GPIO.BCM)

GPIO.setup(sensor, GPIO.IN, GPIO.PUD_DOWN)

previous_state = False current_state = False

while True: time.sleep(0.1)

previous_state = current_state current_state =
GPIO.input(sensor) if current_state != previous_state:
```

3. Run the following command in terminal. *`sudo python motion.py`*

Server-Client Systems on using two RASPBERRY PI

1. Set the static IP for both the Raspberry Pi's to form a network.

```
sudo ipconfig eth0 192.168.1.1
```

//Assume, this to be our server

```
sudo ipconfig eth0 192.168.1.2
```

//Assume, this to be our client

2. To confirm that both the pi are in a network we shall ping from each of the device ping

`192.168.1.1` //Do this on Server

`ping 192.168.1.2` //Do this on Client

3. On Server:

create a python file with

We have **thing-server.py** here.

```
import RPi.GPIO as GPIO import time

import network SWITCH = 10

GPIO.setmode(GPIO.BCM) GPIO.setup(SWITCH, GPIO.IN) def
heard(phrase):

print "heard:" + phrase for a in phrase:

if a == "\r" or a == "\n": pass # strip it

else:

if (GPIO.input(SWITCH)): network.say("1")

else:

network.say("0") while True:

print "waiting for connection" network.wait(whenHearCall=heard)
print "connected"

while network.isConnected(): print "server is running"
time.sleep(1)

print "connection closed"
```

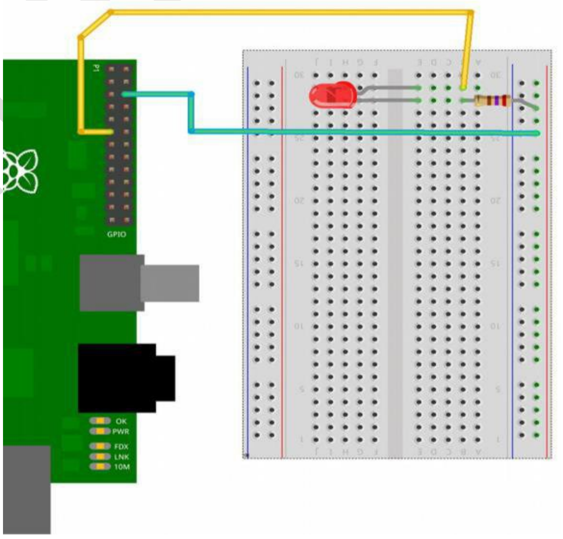
4. On Client

Create a python file again here

We have **thing-client.py** here

```
import RPi.GPIO as GPIO import time

import sys import network

SERVER_IP = sys.argv[1]

LED = 11 GPIO.setmode(GPIO.BCM) GPIO.setup(LED, GPIO.OUT)
gotResponse = False

def heard(phrase): global gotResponse print "heard:" + phrase

for a in phrase:

if a == "\r" or a == "\n": pass # skip it

elif a == "0": GPIO.output(LED, False)

else:

GPIO.output(LED, True) gotResponse = True

while True: while True:

try:

print "connecting to switch server" network.call(SERVER_IP,
whenHearCall=heard) break

except:

print "refused" time.sleep(1)

print "connected"
```

5. Set up the following circuit on client Raspberry Pi.



6. It’s DONE!!

in server run *sudo python thing-server.py*

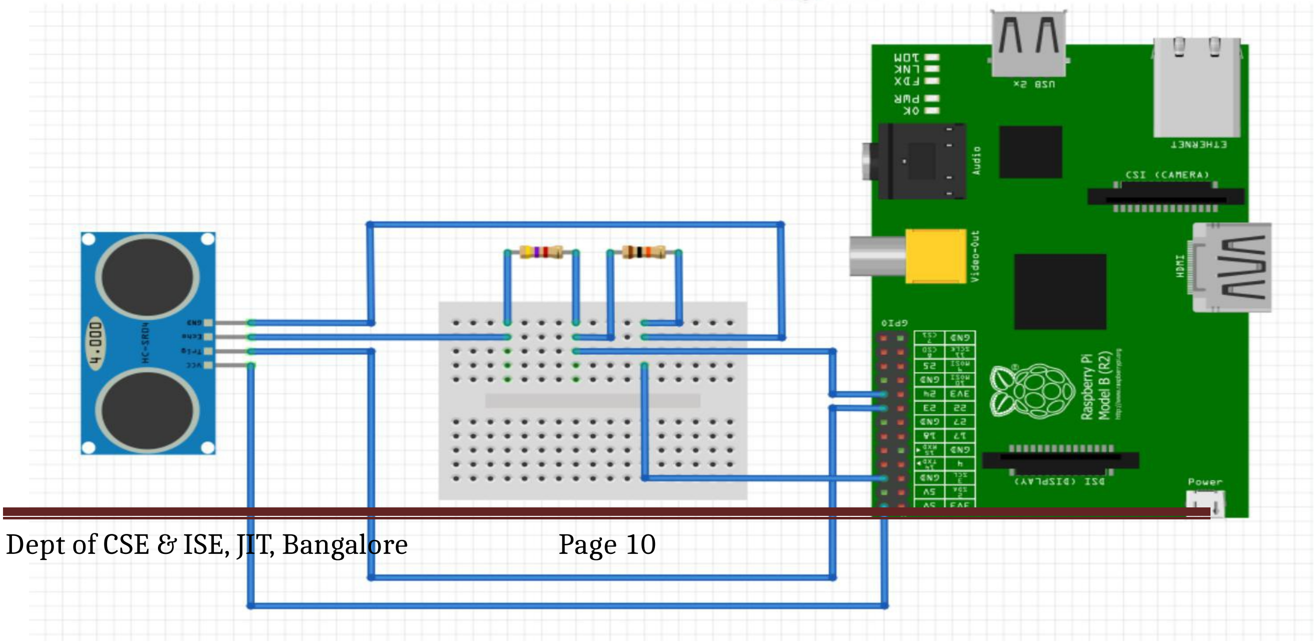
in client run *sudo python thing-client.py 192.168.1.1*

Distance Measurement using HC-SR04 Ultrasonic range sensor

Things Needed:

HC-SR04 1kΩ
Resistor
2kΩ Resistor
Jumper Wires

1. Set up the following circuit on Raspberry Pi.



2.

Create a python file

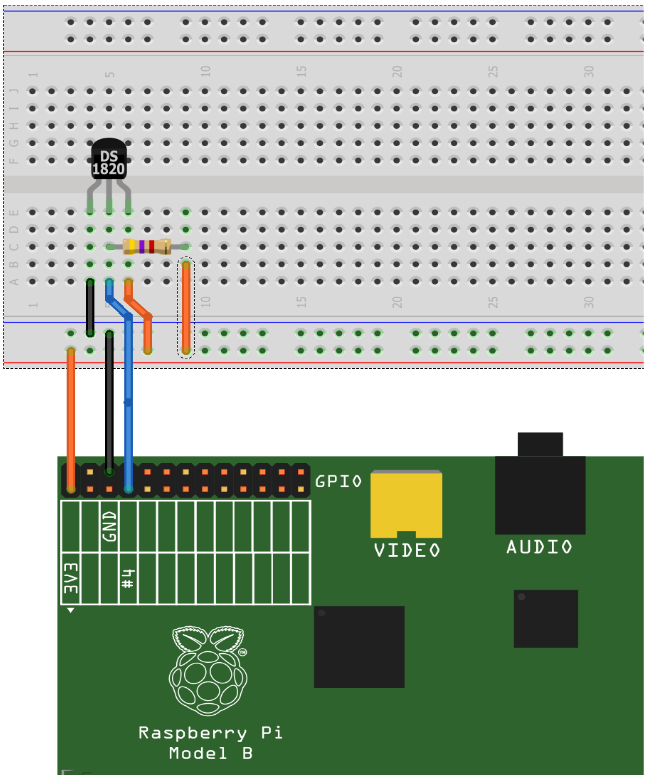
We have *ultrasonicsensor.py*

```
import RPi.GPIO as GPIO                    #Import GPIO library
import time                   #Import time library

GPIO.setmode(GPIO.BCM)                     #Set GPIO pin numbering
GPIO.setwarnings(False)

TRIG = 38                     #Associate pin 38 to TRIG

ECHO = 37                     #Associate pin 37 to ECHO

print "Distance measurement in progress"
GPIO.setup(TRIG,GPIO.OUT)                  #Set pin as GPIO out

GPIO.setup(ECHO,GPIO.IN)                   #Set pin as GPIO in

while True:
GPIO.output(TRIG, False)                   #Set TRIG as LOW
print "Waitng For Sensor To Settle"
time.sleep(2)                 #Delay of 2 seconds
```

```
GPIO.output(TRIG, True)                    #Set TRIG as HIGH
time.sleep(0.00001)                       #Delay of 0.00001 seconds
GPIO.output(TRIG, False)                   #Set TRIG as LOW
while GPIO.input(ECHO)==0:                 #Check whether the ECHO is LOW
pulse_start = time.time()                 #Saves the last known time of LOW pulse


while GPIO.input(ECHO)==1:                 #Check whether the ECHO is HIGH
pulse_end = time.time()                    #Saves the last known time of HIGH pulse
pulse_duration = pulse_end - pulse_start #Get pulse duration to a variable


distance = pulse_duration * 17150          #Multiply pulse duration by 17150 to get distance
distance = round(distance, 2)              #Round to two decimal points
if distance > 2 and distance < 400:        #Check whether the distance is within range
print "Distance:",distance - 0.5,"cm"  #Print distance with 0.5 cm calibration
else:
print "Out Of Range"                      #display out of range
```

Temperature sensor

Pin Diagram



Building the circuit

- Put the resistor between pins 2 and 3 as in the diagram.
- Connect pin 3 of temperature gauge to 3.3v GPIO pin.
- Connect pin 2 of temperature gauge to GPI04(pin 7).
- Connect pin 1 of temperature gauge to GND(pin 6)

Steps to sense the temperature

At the prompt pi@raspberrypi /$ Type: `sudo modprobe w1-gpio`

```
sudo modprobe w1-therm
```

Now you need to enter the following at the command prompt:

```
sudo nano /boot/config.txt
```

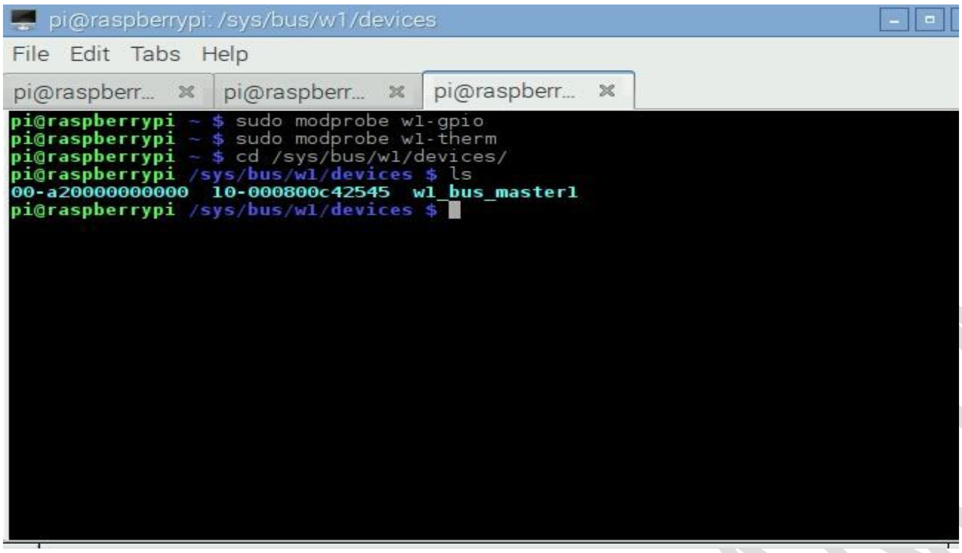
to open the */boot/config.txt* file for editing. Then scroll down to the bottom of the file, and add the line:

```
dtoverlay=w1-gpio
```

```
cd /sys/bus/w1/devices/
 ls
```

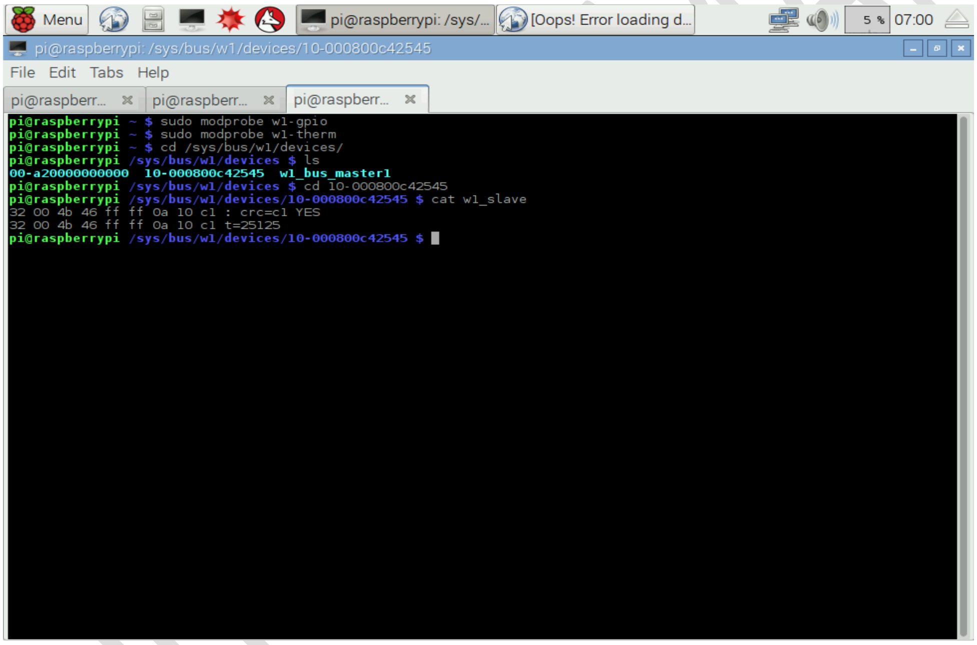
This gives you serial number of a directory. The serial number of our thermometer is

10-000800c42545.



```
pi@raspberrypi ~ $ sudo modprobe w1-gpio
pi@raspberrypi ~ $ sudo modprobe w1-therm
pi@raspberrypi ~ $ cd /sys/bus/w1/devices/
pi@raspberrypi /sys/bus/w1/devices $ ls
00-a20000000000  10-000800c42545  w1_bus_master1
pi@raspberrypi /sys/bus/w1/devices $ 
```

Type: **`cd 10-000800c42545 cat w1_slave`**

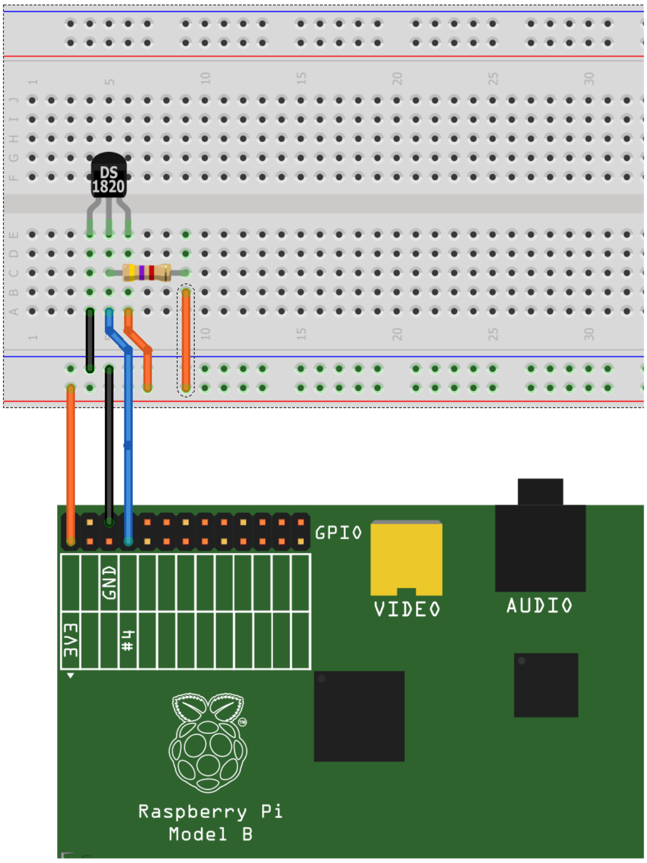


```
pi@raspberrypi ~ $ sudo modprobe w1-gpio
pi@raspberrypi ~ $ sudo modprobe w1-therm
pi@raspberrypi ~ $ cd /sys/bus/w1/devices/
pi@raspberrypi /sys/bus/w1/devices $ ls
00-a20000000000  10-000800c42545  w1_bus_master1
pi@raspberrypi /sys/bus/w1/devices $ cd 10-000800c42545
pi@raspberrypi /sys/bus/w1/devices/10-000800c42545 $ cat w1_slave
32 00 4b 46 ff ff 0a 10 c1 : crc=c1 YES
32 00 4b 46 ff ff 0a 10 c1 t=25125
pi@raspberrypi /sys/bus/w1/devices/10-000800c42545 $ 
```

```
temp.py

import time

while 1:
     tempfile = open("/sys/devices/w1_bus_master1/10-000800c4dc4a/w1_slave")
     thetext = tempfile.read()
     tempfile.close()

     tempdata = thetext.split("\n") [1].split(" ")[9]
     temperature = float(tempdata[2:])
     temperature = temperature / 1000

     print temperature
     time.sleep(1)
```

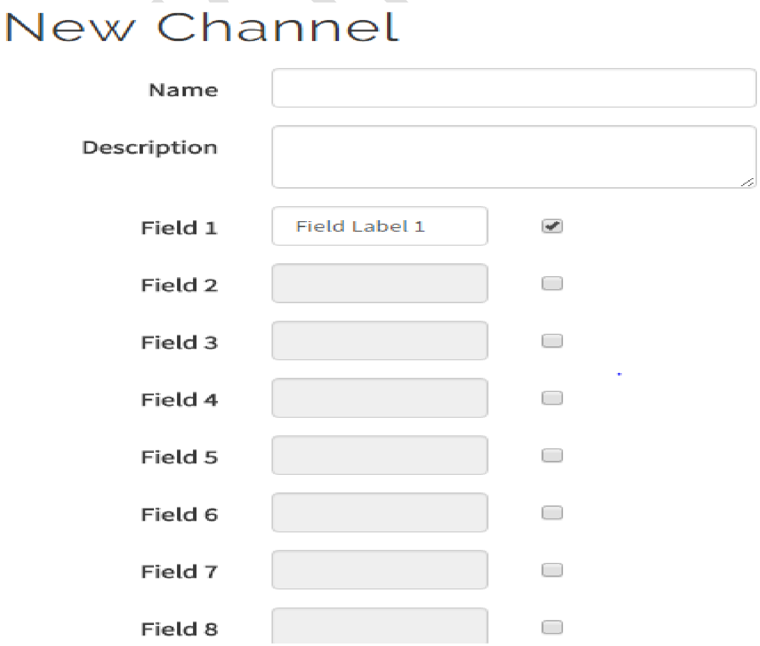
Temperature sensor data to cloud

Pin Diagram



Create an account in ThingSpeak

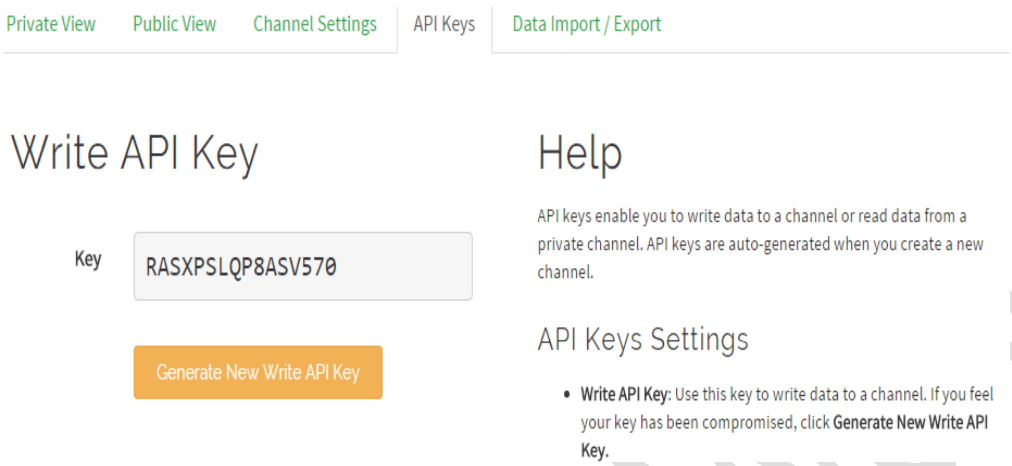
1. Sign in
2. Go to channels and create a new channel



New Channel

Name		
Description		
Field 1	Field Label 1	☑
Field 2		□
Field 3		□
Field 4		□
Field 5		□
Field 6		□
Field 7		□
Field 8		□

3. Give the name as RaspberryPi , description as Temperature sensor and field 1 as temperature
4. Press on save channel ,A new channel will be created with a field to show graph.
5. Copy your API key , To get API key selct your channe and click on APIKeys TAB



Write API Key

Key RASXPSLQP8ASV570

Help

API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.

API Keys Settings

- **Write API Key:** Use this key to write data to a channel. If you feel your key has been compromised, click **Generate New Write API Key.**

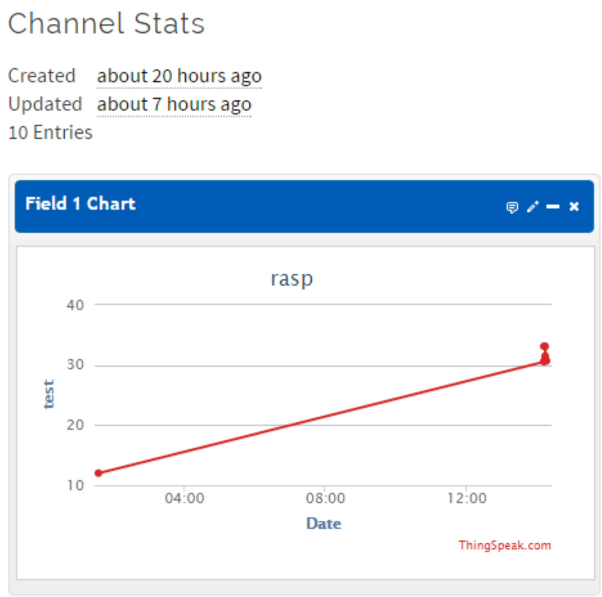
Copy the above API key and place in your program.

temp-cloud.py

```
import time
import httplib, urllib

def doit(temp):
    params = urllib.urlencode({'field1': temp,
'key':'RASXPSLQP8ASV570'})
    headers = {"Content-type": "application/x-www-
form-urlencoded","Accept": "text/plain"}
    conn =
httplib.HTTPConnection("api.thingspeak.com:80")

    conn.request("POST", "/update", params, headers)
    response = conn.getresponse()
    print response.status, response.reason
    data = response.read()
    conn.close()

while 1:
     tempfile = open("/sys/devices/w1_bus_master1/10-
000800c4dc4a/w1_slave")
     thetext = tempfile.read()
     tempfile.close()

     tempdata = thetext.split("\n") [1].split(" ")[9]
     temperature = float(tempdata[2:])
     temperature = temperature / 1000
```

```
        print temperature
        doit(temperature)
        time.sleep(16)
```

Run the above program chek the graph in thing speak.
6.



Channel Stats

Created about 20 hours ago
Updated about 7 hours ago
10 Entries

Field 1 Chart

For more information CONTACT US :

Adithya B

adithya.b@jyothyit.org

Vadiraja Acharya

vadiraja.a@jyothyit.org

College Website:

http://jyothyit.ac.in/