**ASSIGNMENT NO: 9**

AIM:

OBJECTIVE:

* To understand functionalities of various single board embedded platform fundamentals.
* To develop a temperature sensing application and sending this data to Thingspeak.

THEORY:

ThingSpeak is an open data platform for monitoring your data online. You can set the data as private or public depending on your choice. ThingSpeak takes minimum of 15 seconds to update your readings. Its a great platform for building your IOT projects.

 We will read the temperature and humidity from the DHT22 and then we will send it to the API of the ThingSpeak channel. We will get the API after creating the channel.

Temperature sensor:

It is a device ,a thermocouple or RTD,that provides temperature measurement through an electrical signal.

Thermocouple:

It is made from two dissimilar metals that generate electrical voltage in direct proportion to changes in temperature.The wires are joined together to form measuring junction and reference junction.

RTD:

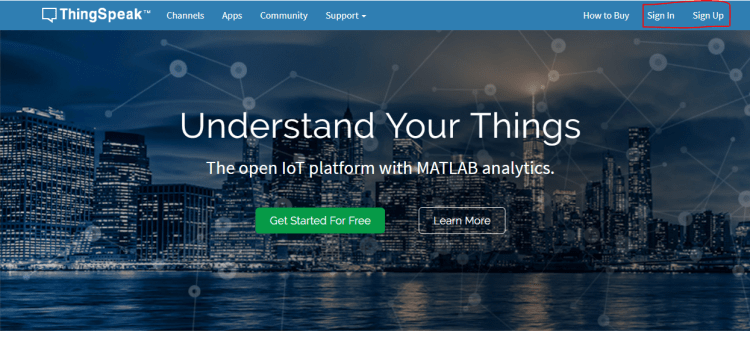
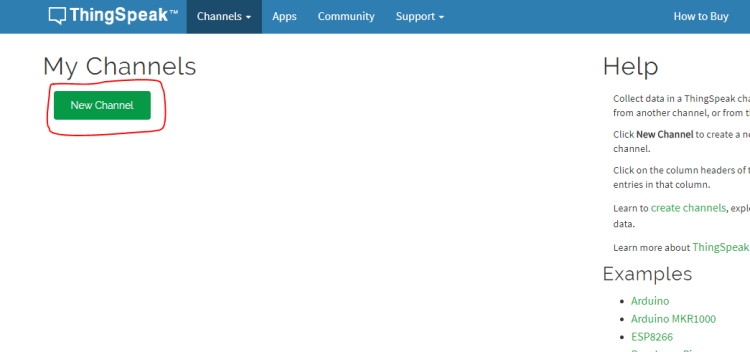
Resistor temperature detection is a variable resistor that will change its electrical resistance in direct proportion to chenges in temperature in a precise,repeatable and linear manner.

## ****Components Required****

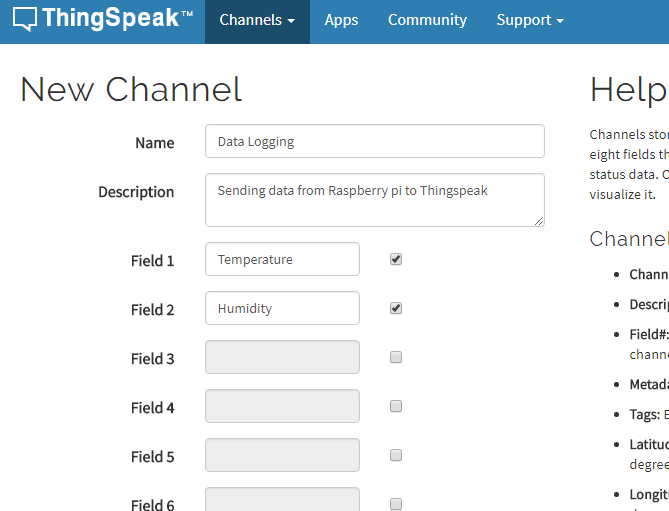
The components you will be required for this Raspberry Pi IOT project are as follows

* Raspberry Pi (I have used Raspberry pi 3)
* DHT22
* 10k resistor
* Jumper cables

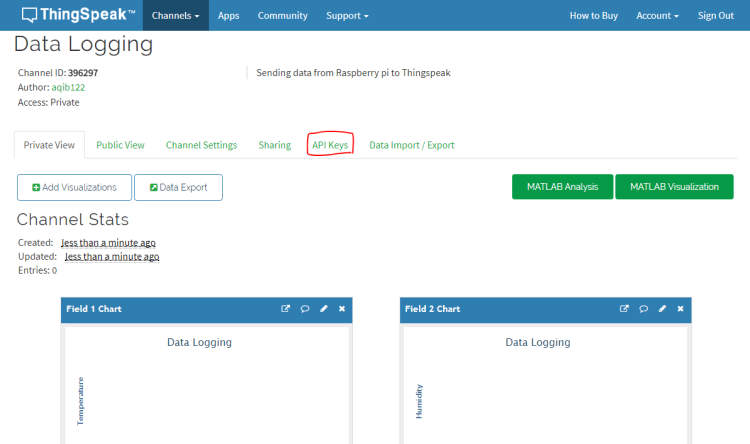
## ****Setting up the ThingSpeak Account****

1. First of all, go to the following link and sign up to ThingSpeak. If you already have an account, then sign in.  
   <https://thingspeak.com/>  
   
2. After creating the account or logging in, you will be see a page like shown below. Click on new channel.  
   

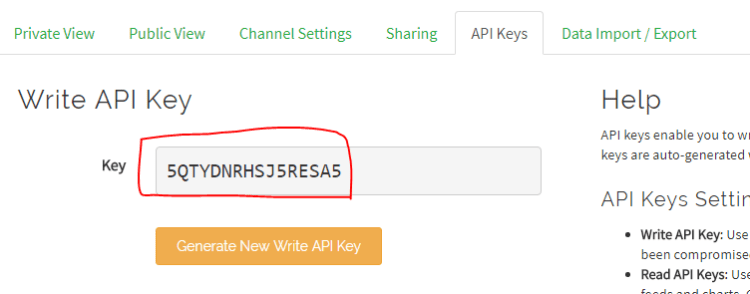
3.Now you will need to enter the information about the channel. Enter the information like shown below. Select two fields because we will be sending the data for the two fields from the raspberry pi. Leave the other information as it is and save the channel.



4.After saving the channel, you will see a page like shown below. From there, go to the API keys tab.



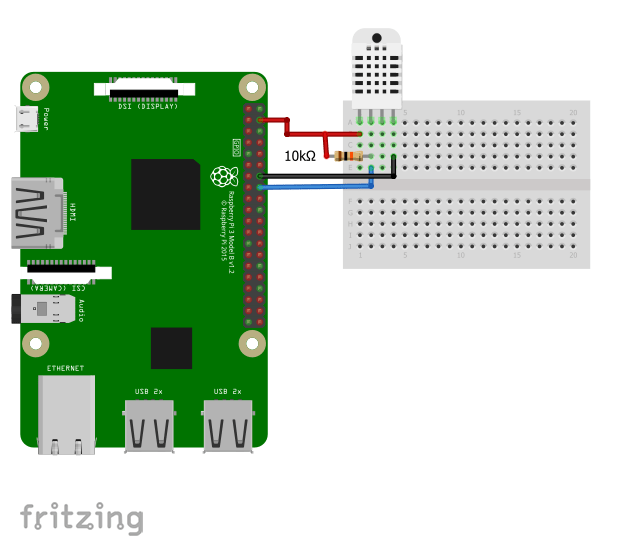
5.In the API keys tab, copy the write API key. This is the API key at which we will send the data from the Raspberry pi.



## ****Circuit Diagram and Explanation****

Make the connections of the DHT22 with the Raspberry pi as described below

|  |  |
| --- | --- |
| **DHT22** | **Raspberry pi** |
| VCC | 5v |
| Data pin | Connect to GPIO 23 and also connect to 5V through 10K resistor |
| GND | GND |

[](https://i0.wp.com/electronicshobbyists.com/wp-content/uploads/2017/11/d-articles-electronicshobbyists-raspberry-pi-dht2.png)

## ****Installing the DHT22 Library****

To install the library, Follow the below steps

1. Enter the below command to clone the library

git clone https://github.com/adafruit/Adafruit\_Python\_DHT.git

1. Then enter in to the installed directory using the below command

cd Adafruit\_Python\_DHT

1. Now download the required modules using the below command

sudo apt-get install build-essential python-dev

1. Then install the library using the below command

sudo python setup.py install

## ****Raspberry pi Code****

Enter the below code in your python script and run it.

import sys

import urllib2

from time import sleep

import Adafruit\_DHT as dht

# Enter Your API key here

myAPI = '5QTYDNRHSJ5RESA5'

# URL where we will send the data, Don't change it

baseURL = 'https://api.thingspeak.com/update?api\_key=%s' % myAPI

def DHT22\_data():

# Reading from DHT22 and storing the temperature and humidity

humi, temp = dht.read\_retry(dht.DHT22, 23)

**return** humi, temp

**while** True:

**try**:

humi, temp = DHT22\_data()

# If Reading is valid

**if** isinstance(humi, **float**) **and** isinstance(temp, **float**):

# Formatting to two decimal places

humi = '%.2f' % humi

temp = '%.2f' % temp

# Sending the data to thingspeak

conn = urllib2.urlopen(baseURL + '&field1=%s&field2=%s' % (temp, humi))

print conn.read()

# Closing the connection

conn.close()

**else**:

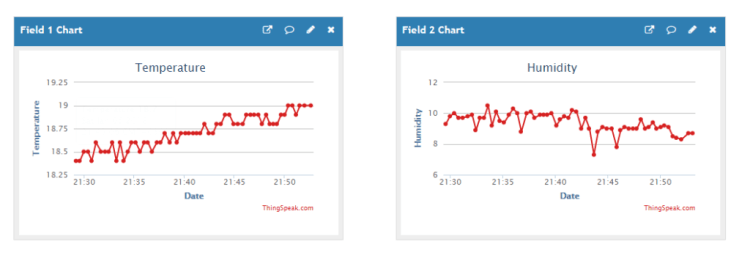
print 'Error'

# DHT22 requires 2 seconds to give a reading, so make sure to add delay of above 2 seconds.

sleep(20)

except:

**break**

After running it, output looks like below[](https://i0.wp.com/electronicshobbyists.com/wp-content/uploads/2018/01/ThingSpeak-output.png)

CONCLUSION:

After the study of this assignment we are familiar with the Thingspeak platform and how to send temperature and humidity data to thingspeak cloud .

**ASSIGNMENT NO: 0**

**AIM:** Design a web interface to control connected LEDs remotely using Raspberry-Pi/Beagle board/Arduino.

Theory:

One of the function of IOT is remote control of devices,being able to trigger action from aremote location.This is known as remote configuration,

Whether it is a home,office or an industrial site ,connected devices rely on some form of automation through sensors or machines and need a mechanism to control their operation remotely.

Here we will build a small IOT simulation using Rpi and showcase the most commeon operation switching on and off a remote device.

Hardware Requirements:

* RPi model with Raspbian OS
* 1 LED AND 1490OHM RESISTOR

Software Requirements:

* RPi GPIO,GPIO Python library for Raspbian OS.
* Pubnub Python SDK
* Pubnub Javascript SDK

We will create 2 subsystems.one will be controller, a basic web application in the form of a webpage which can display the current states of device and send control messages to it and second one is the actual device simulated as on LED and controlled via Raspberry Pi.

Web application

Th web interface looks like this:

Raspberry Pi LED controller

TOGGLE

LED

* This is a very simple webpage with a visual indicator for the device and a button to toggle ON/OFF state of LED.
* In the background,we have pubnub javascript API that performs 2 operations upon receiving certain events.

1)sends a request to toggle the state of device.

2)recives response with current state of the device.

* Button click event:

When the toggle button is clicked, the webpage sends a TOGGLE request message to the device via GPIO Raspberry control channel.

RPi and LED

Rpi GPIO python library is used to send the control messages to RPi GPIO Ports.

When a toggle button is clicked,the webpage sends a toggle request which is received by application and checks the current state of GPIO pin of LED toggles its state and send new state back to the web app as response.

**ASSIGNMENT NO: 0**

AIM: Install, configure APACHE server and deploy an application on Raspberry Pi/Beagle board/Arduino. Write client applications to get services from the server application.

Set up a LAMP (**L**inux, **A**pache, **M**ySQL, **P**HP) stack on your Raspberry Pi

and configure it to work as a web server. You’ll download and install

WordPress and set up a basic website which you can access on any device on

the same network as your Pi.

* Install software on your Raspberry Pi
* Install and configure Apache, PHP, and MySQL to create a LAMP web server
* Download WordPress and run it as a local website on your Raspberry Pi
* Configure WordPress and make your website accessible to other devices on your local network.

### 1)Hardware

* A Raspberry Pi computer connected to the internet

## 2)Set up an Apache web server

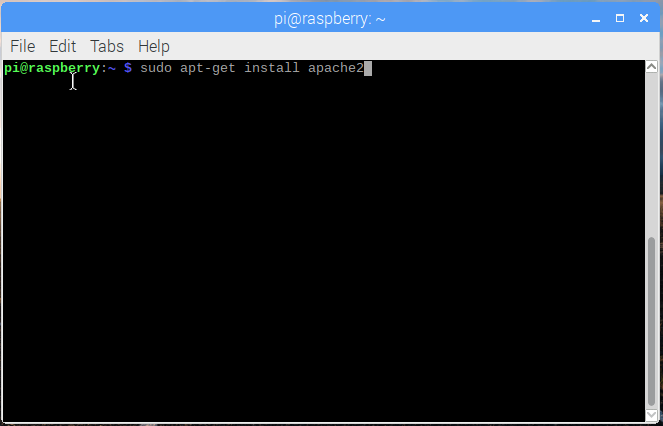
Apache is a popular web server application you can install on the Raspberry Pi to allow it to serve web pages.

On its own, Apache can serve HTML files over HTTP. With additional modules it can serve dynamic web pages using scripting languages such as PHP.

### Install Apache

* Open a terminal window by selecting **Accessories** > **Terminal** from the menu.
* Install the apache2 package by typing the following command into the terminal and pressing Enter:

sudo apt-get install apache2 -y



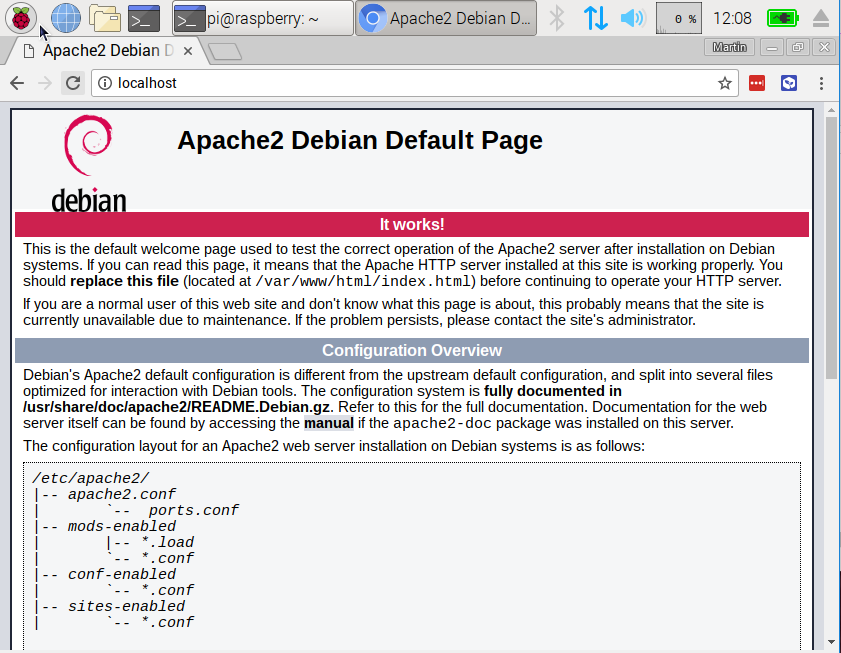
### Test the web server

By default, Apache puts a test HTML file in the web folder that you will be able to view from your Pi or another computer on your network.

Open the Apache default web page on your Raspberry Pi:

* Open Chromium by selecting **Internet** > **Chromium Web Browser** from the menu.
* Enter the address http://localhost.

You should see this in your browser window:



This means you have Apache working!

You will also be able to open this web page from any other computer on your network using the IP address of your Raspberry Pi, e.g. http://192.168.1.10.

To find out your Raspberry Pi’s IP address, type hostname -I into the terminal window. Your Raspberry Pi’s [IP address](https://www.raspberrypi.org/documentation/remote-access/ip-address.md) is a really useful and will allow you to remotely access it.

### Changing the default web page

This default web page is just a HTML file on the file system. It is located at /var/www/html/index.html.

* Navigate to this directory in the terminal and have a look at what’s inside:

cd /var/www/html

ls -al

You should see this in the window:

total 12

drwxr-xr-x 2 root root 4096 Jan 8 01:29 .

drwxr-xr-x 3 root root 4096 Jan 8 01:28 ..

-rw-r--r-- 1 root root 177 Jan 8 01:29 index.html

This shows that there is one file in /var/www/html/ called index.html. . refers to the directory itself /var/www/html, and .. refers to the parent directory /var/www/.

### What the columns mean

1. The permissions of the file or directory
2. The number of files in the directory (or 1 if it’s a file).
3. The user that owns the file or directory
4. The group that owns the file or directory
5. The size of the file or directory
6. The date and time of the last modification

As you can see, the html directory and index.html file are both owned by the root user, so you’ll need to use sudo to edit them.

You can edit this file using leafpad:

sudo leafpad index.html

If you make a change to the file, save it, and refresh the browser, you will see your change appear.

## 3)Install PHP

PHP is a **preprocessor**: it’s code that runs when the server receives a request for a web page via a web browser. It works out what needs to be shown on the page, and then sends that page to the browser. Unlike static HTML, PHP can show different content under different circumstances. Other languages are also capable of doing this, but since WordPress is written in PHP, that’s what we need to use this time. PHP is a very popular language on the web: huge projects like Facebook and Wikipedia are written in PHP.

* Install the PHP package with the following command:

sudo apt-get install php -y

### Test PHP

* Create the file index.php:

sudo leafpad index.php

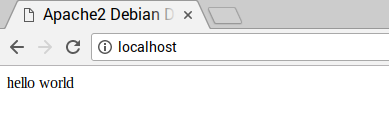
* Put some PHP content in it:

**<?php** echo "hello world"; **?>**

* Save the file.
* Delete index.html, because it takes precedence over index.php:

sudo rm index.html

Refresh your browser. You should see “hello world”. This page is not dynamic, but it is still served by PHP.



If you see the raw PHP above instead of “hello world”, reload and restart Apache like so:

sudo service apache2 restart

* Edit index.php to include some dynamic content, for example:

**<?php** echo date('Y-m-d H:i:s'); **?>**

Or show your PHP info:

**<?php** phpinfo(); **?>**

## 4)Install MySQL

MySQL (pronounced My Sequel or My S-Q-L) is a popular database engine. Like PHP, it’s widely used on web servers, which is why projects like WordPress use it, and why those projects are so popular.

Install the MySQL Server and PHP-MySQL packages by entering the following command into the terminal window:

sudo apt-get install mysql-server php-mysql -y

Now restart Apache:

sudo service apache2 restart

## 5)Download WordPress

You can download WordPress from [wordpress.org](http://wordpress.org/) using the wget command. Helpfully, a copy of the latest version of WordPress is always available at [wordpress.org/latest.tar.gz](https://wordpress.org/latest.tar.gz).At the time of writing, this is version 4.5.

* Change directory to /var/www/html/ and delete all the files in the folder.

cd /var/www/html/

sudo rm \*

* Download WordPress using wget.

sudo wget http://wordpress.org/latest.tar.gz

* Extract the WordPress tarball to get at the WordPress files.

sudo tar xzf latest.tar.gz

* Move the contents of the extracted wordpress directory to the current directory.

sudo mv wordpress/\* .

* Tidy up by removing the tarball and the now empty wordpress directory.

sudo rm -rf wordpress latest.tar.gz

* Running the ls or tree -L 1 command now will show you the contents of a WordPress project:

.

├── index.php

├── license.txt

├── readme.html

├── wp-activate.php

├── wp-admin

├── wp-blog-header.php

├── wp-comments-post.php

├── wp-config-sample.php

├── wp-content

├── wp-cron.php

├── wp-includes

├── wp-links-opml.php

├── wp-load.php

├── wp-login.php

├── wp-mail.php

├── wp-settings.php

├── wp-signup.php

├── wp-trackback.php

└── xmlrpc.php

3 directories, 16 files

This is the source of a default WordPress installation. The files you edit to customise your installation belong in the wp-content folder.

* You should now change the ownership of all these files to the Apache user:

sudo chown -R www-data: .

## Set up your WordPress Database

#### Set up MySQL/MariaDB

To get your WordPress site set up, you need a database. This is where MySQL and MariaDB come in!

* Run the MySQL secure installation command in the terminal window.

sudo mysql\_secure\_installation

* You will be asked Enter current password for root (enter for none): — press **Enter**.
* Type in **Y** and press **Enter** to Set root password?.
* Type in a password at the New password: prompt, and press **Enter**. **Important:**remember this root password, as you will need it later to set up WordPress.
* Type in **Y** to Remove anonymous users.
* Type in **Y** to Disallow root login remotely.
* Type in **Y** to Remove test database and access to it.
* Type in **Y** to Reload privilege tables now.

When complete, you will see the message All done! and Thanks for using MariaDB!.

#### 6)Create the WordPress database

* Run mysql in the terminal window:

sudo mysql -uroot -p

* Enter the root password you created.

You will be greeted by the message Welcome to the MariaDB monitor.

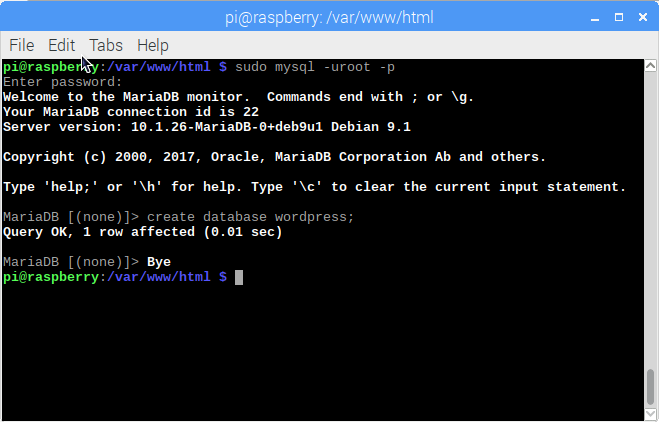
* Create the database for your WordPress installation at the MariaDB [(none)]> prompt using:

create database wordpress;

Note the semi-colon ending the statement.

If this has been successful, you should see this:

Query OK, 1 row affected (0.00 sec)



* Now grant database privileges to the root user. **Note:** you will need to enter your own password after IDENTIFIED BY.

GRANT ALL PRIVILEGES ON wordpress.\* TO 'root'@'localhost' IDENTIFIED BY 'YOURPASSWORD';

* For the changes to take effect, you will need to flush the database privileges:

FLUSH PRIVILEGES;

* Exit the MariaDB prompt with Ctrl + D.