

IoT tutorial

This tutorial explains how to make IoT projects

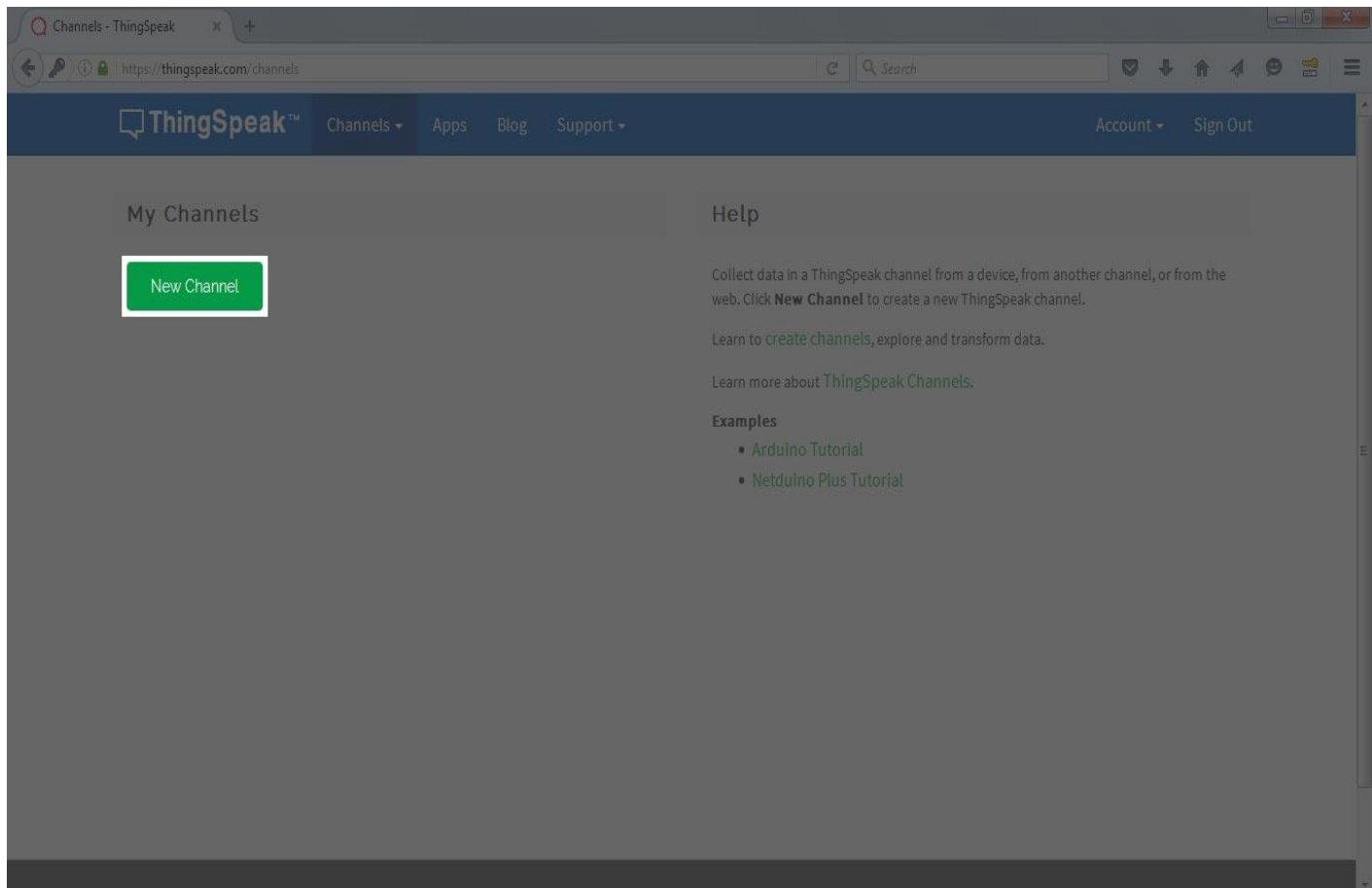
1. ThingSpeak Basics and account setup

ThingSpeak is an open cloud data platform where you can store and retrieve data.

URL : <https://thingspeak.com/>

If you do not have a ThingSpeak account create one. Once you have a ThingSpeak account login to your account.

Create a new channel by clicking on the button as shown in below image - A channel is the source for your data. Where you can store and retrieve data. A channel can have maximum 8 fields. It means you can store 8 different data to a channel.



2. Enter basic details of the channel

here we are creating channel to store data from LM35 temperature and humidity sensor so we need two fields.

ThingSpeak™ Channels ▾ Apps Community Support ▾ How to Buy Account ▾ Sign Out

Private View Public View Channel Settings API Keys Data Import / Export

Channel Settings

Percentage complete 50%

Channel ID 264765

Name

Description

Field 1

Field 2

Field 3

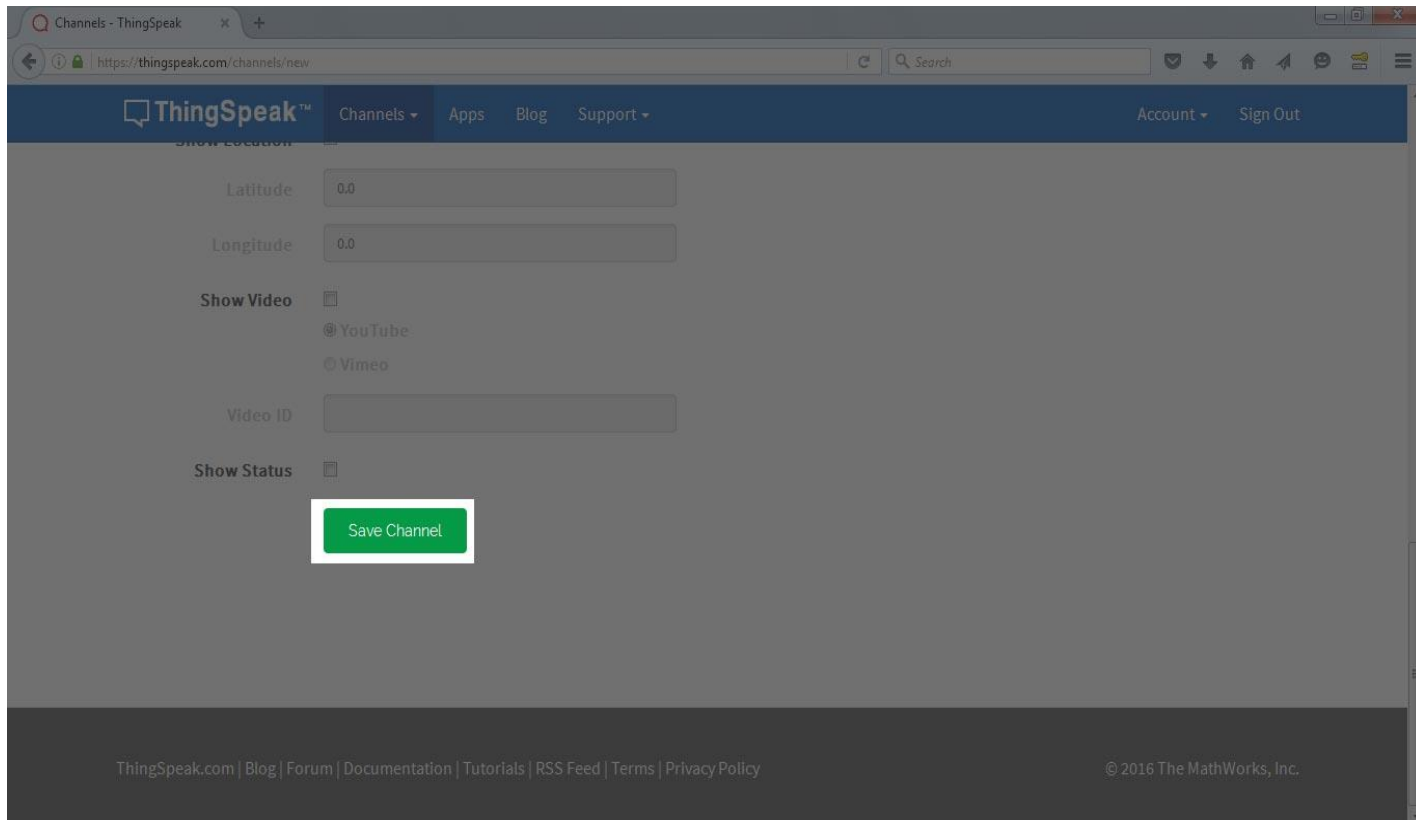
Help

Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

Channel Settings

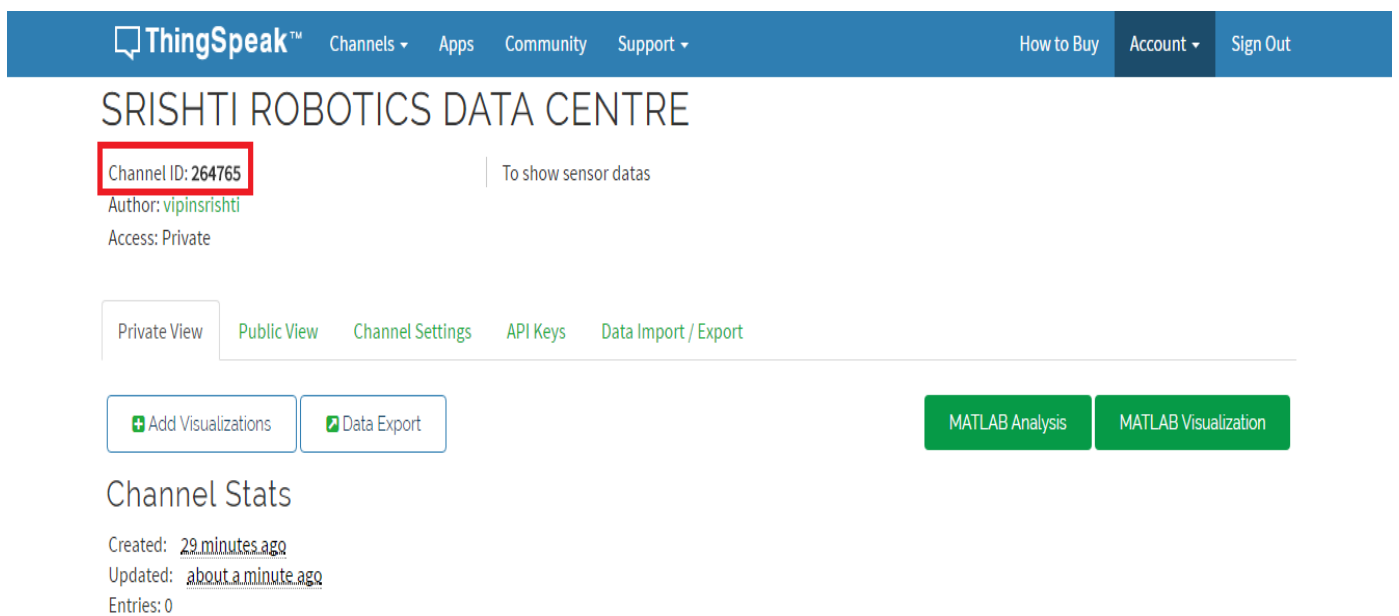
- **Channel Name:** Enter a unique name for the ThingSpeak channel.
- **Description:** Enter a description of the ThingSpeak channel.
- **Field#:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- **Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- **Tags:** Enter keywords that identify the channel. Separate tags with commas.
- **Latitude:** Specify the position of the sensor or thing that collects data in decimal degrees. For example, the latitude of the city of London is 51.5072.

3. Scroll down and save the channel



4. Channel ID

Channel Id is the identity of your channel. Note down this.



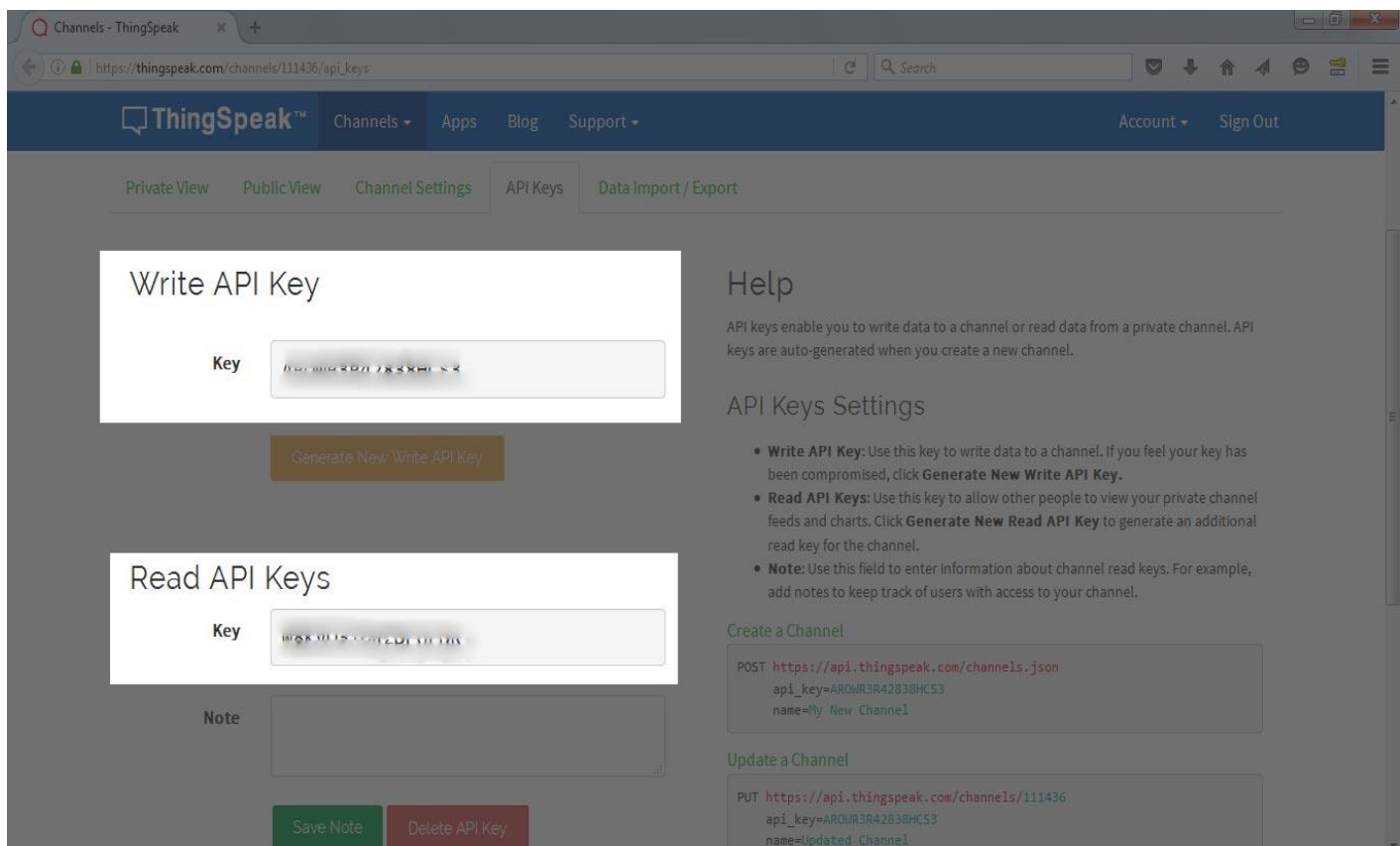
5.API Keys

API (Application Programming Interface) keys are the keys to access to your channel. In simple language you can understand that these are password to access your channel. You can access your channel in two ways-

1. To update channel / data logging : API Write Key will be used to access in this mode.

2. To retrieve data : API Read Key will be used to access in this mode.

Click on the API tab to know your API keys. We have blurred our API Keys for security reasons.



The screenshot shows the ThingSpeak API Keys page. The page has a dark blue header with the ThingSpeak logo and navigation links: Channels, Apps, Blog, Support, Account, and Sign Out. Below the header, there are tabs for Private View, Public View, Channel Settings, API Keys (selected), and Data Import / Export. The main content area is divided into two sections: 'Write API Key' and 'Read API Keys'. The 'Write API Key' section has a 'Key' field with a blurred value and a 'Generate New / Write API Key' button. The 'Read API Keys' section has a 'Key' field with a blurred value, a 'Note' field, and 'Save Note' and 'Delete API Key' buttons. On the right side, there is a 'Help' section explaining API keys and an 'API Keys Settings' section with a list of instructions. Below that, there are examples for 'Create a Channel' (POST request) and 'Update a Channel' (PUT request).

Write API Key

Key: [blurred]

Generate New / Write API Key

Read API Keys

Key: [blurred]

Note: [empty]

Save Note Delete API Key

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**.
- **Read API Keys:** Use this key to allow other people to view your private channel feeds and charts. Click **Generate New Read API Key** to generate an additional read key for the channel.
- **Note:** Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.

Create a Channel

```
POST https://api.thingspeak.com/channels.json
api_key=AR0WR3R42838HC53
name=My New Channel
```

Update a Channel

```
PUT https://api.thingspeak.com/channels/111436
api_key=AR0WR3R42838HC53
name=Updated Channel
```

6. Accessing Channel:

You may use following URLs to access your channel -

To Update channel / data uploading / data logging

URL: http://api.thingspeak.com/update?api_key=YOUR-API&field1=VAR-1&field2=VAR-2

make the following replacements in the above mentioned URL-

1. YOUR-API : Your API Write Key

2. VAR-1 : Temperature Data

3. VAR-2 : Humidity Data

Response : If you get a positive number that means the data has been uploaded to your channel. The number is index of the last entry you have made.

Retrieve channel / data reading

URL: http://api.thingspeak.com/channels/YOUR-CHANNEL-ID/fields/FIELD.json?results=NOS-OF-RESULTS&api_key=YOUR-API

make the following replacements in the above mentioned URL-

1. YOUR-CHANNEL-ID - Your channel ID

2. FIELD - Field you want to retrieve. Write 1 for Field1, 2 for Field2

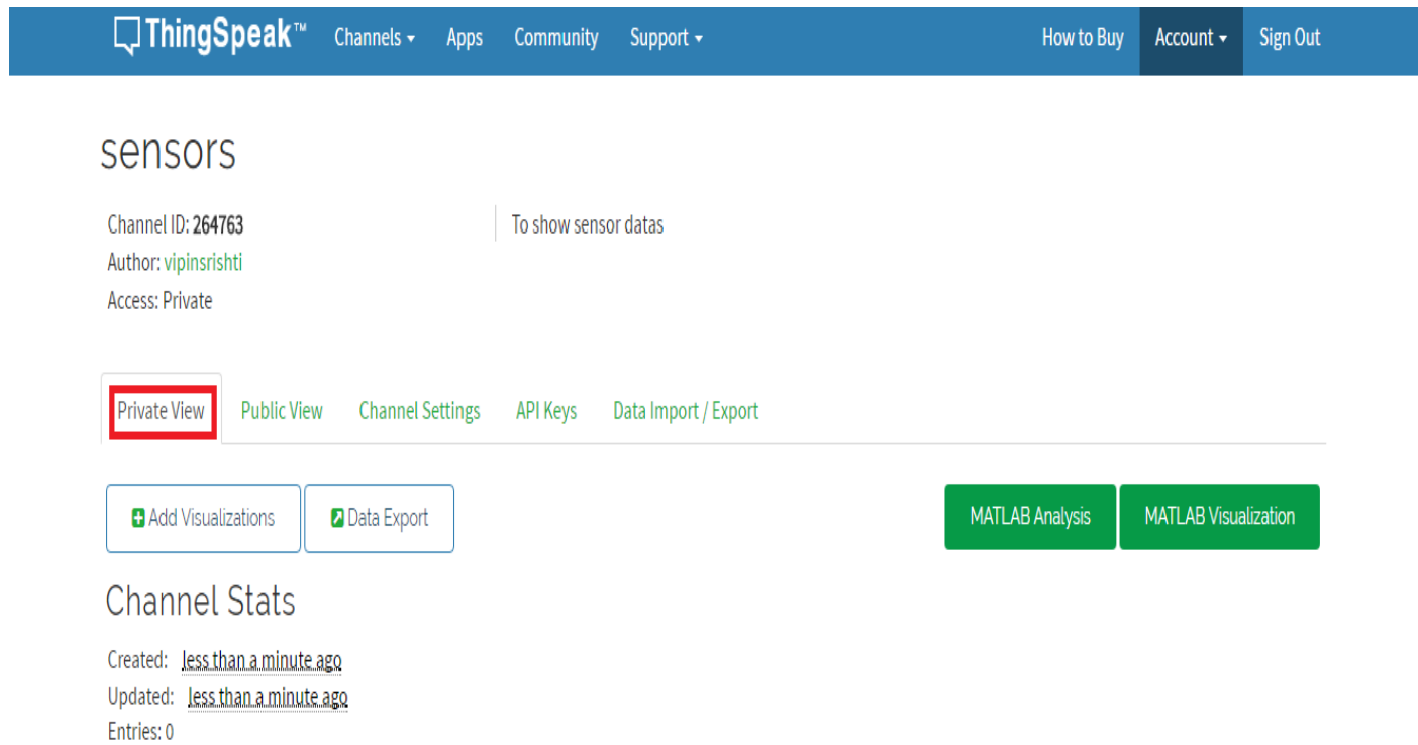
3. NOS-OF-RESULTS = The number of rows you want to retrieve.

4. YOUR-API : Your API Read Key

Response : You will get data as per your specifications in JSON format.

7. Reading data through ThingSpeak website.

Login to your account. Select your channel and click on the view as shown in the following image.



The screenshot shows the ThingSpeak website interface. At the top is a blue navigation bar with the ThingSpeak logo and links for Channels, Apps, Community, Support, How to Buy, Account, and Sign Out. Below the navigation bar, the page title is "sensors". The channel information is displayed: Channel ID: 264763, Author: vipinsrishti, and Access: Private. A message "To show sensor datas" is shown. Below this, there are navigation tabs: Private View (highlighted with a red box), Public View, Channel Settings, API Keys, and Data Import / Export. There are also buttons for Add Visualizations, Data Export, MATLAB Analysis, and MATLAB Visualization. The Channel Stats section shows: Created: less than a minute ago, Updated: less than a minute ago, and Entries: 0.

ThingSpeak™ Channels ▾ Apps Community Support ▾ How to Buy Account ▾ Sign Out

sensors

Channel ID: 264763 | To show sensor datas
Author: vipinsrishti
Access: Private

Private View Public View Channel Settings API Keys Data Import / Export

+ Add Visualizations Data Export

MATLAB Analysis MATLAB Visualization

Channel Stats

Created: [less than a minute ago](#)
Updated: [less than a minute ago](#)
Entries: 0

PROJECT 1: ONLINE WEATHER STATION

Hardware

1x LM35 temperature sensor with 4.7k resistor across vcc and data pin

1x NodeMCU

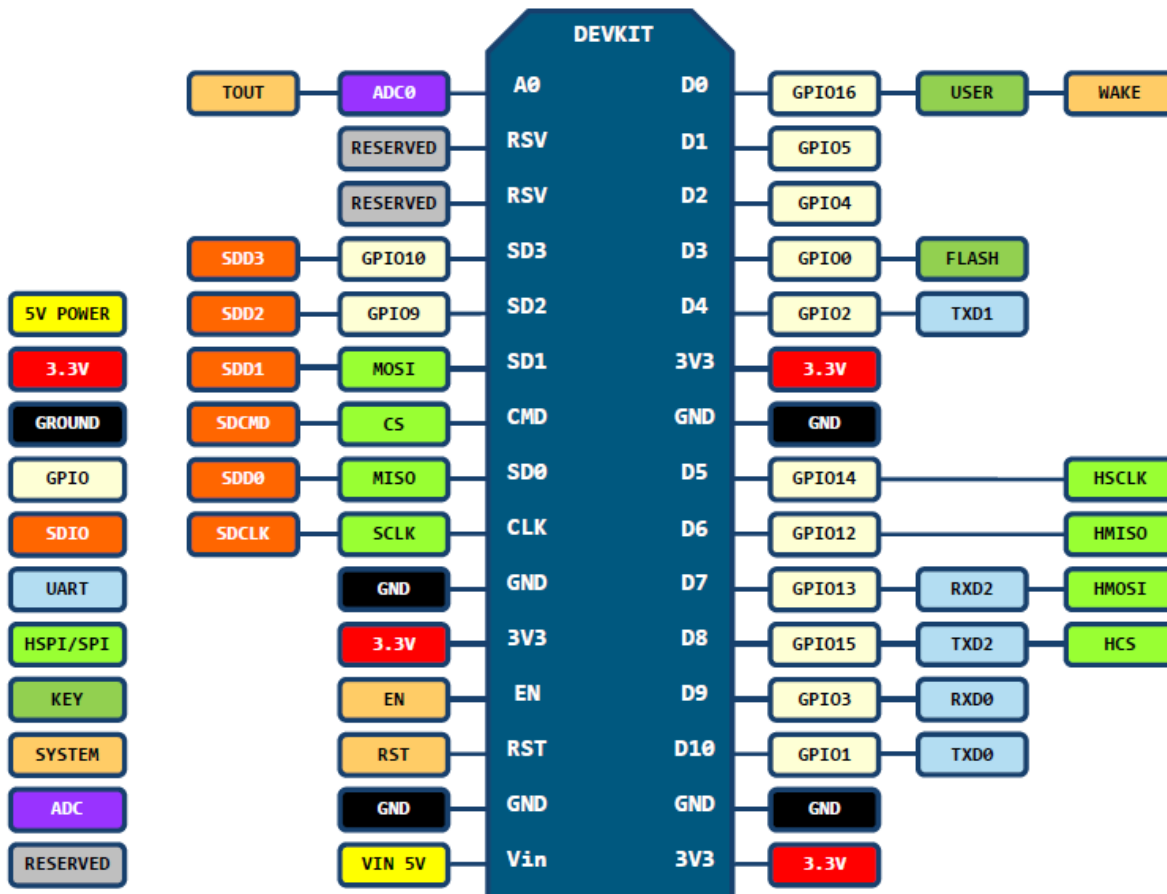
1x Access Point Connected to Internet(Wi-Fi)

Software:

Arduino IDE

The physical pins on NodeMCU v1.0 does not correspond to pins in Arduino IDE. Check out figure for pin mapping

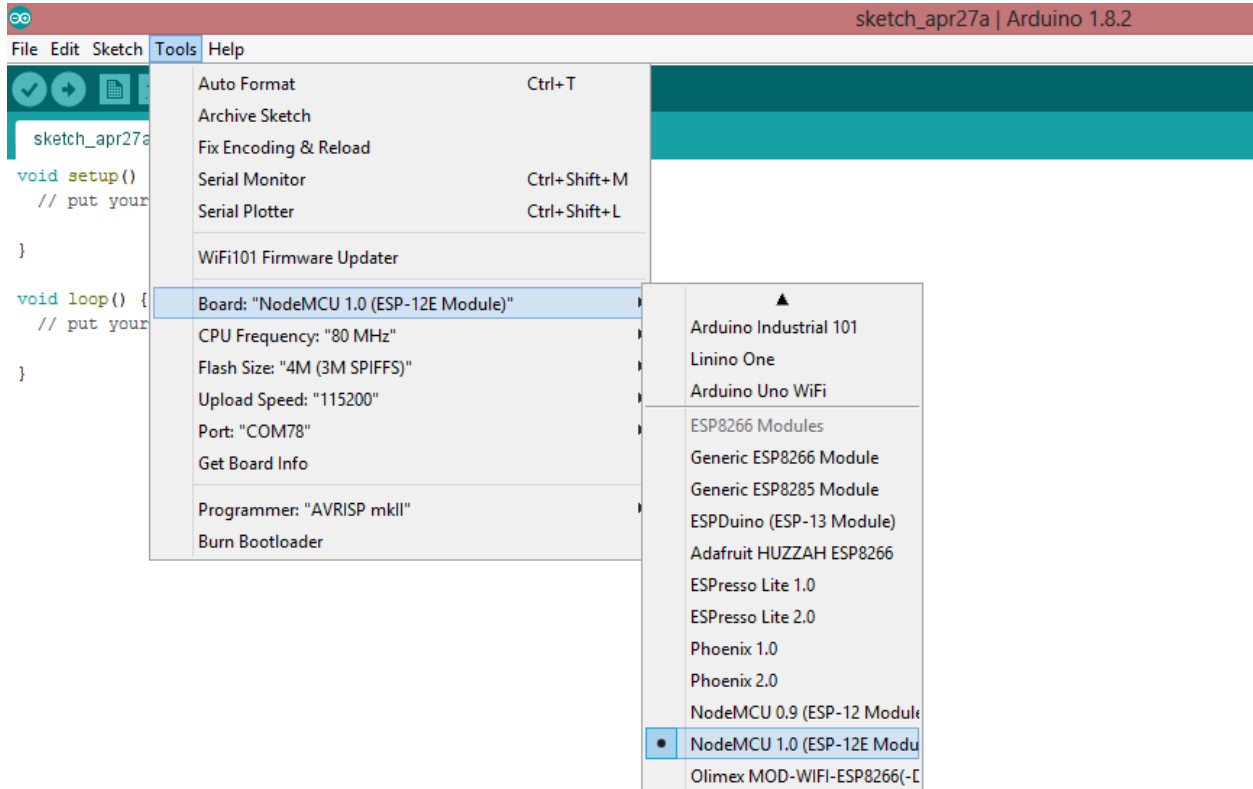
PIN DEFINITION



D0(GPI016) can only be used as gpio read/write, no interrupt supported, no pwm/i2c/ow supported.

1. Connect 3.3v and gnd from NodeMCU v1.0 to LM35 respectively.
2. Connect data from LM35 to A0 on NodeMCU v1.0

Connect microUSB to NodeMCU v1.0



Launch ESP8266 Arduino IDE, Select Tools -> Board -> NodeMCU v1.0; and ensure parameters are correct. Refer to screenshot.

Program the source code to read LM35 temperature sensor and the acquired data to be sent to thingspeak.


```
sketch_apr27b $
#include "ThingSpeak.h"
#include <ESP8266WiFi.h>

char ssid[] = "SRISHTI ROBOTICS";          // your network SSID (name)
char pass[] = "srishtirobotics123";      // your network password

int status = WL_IDLE_STATUS;
WiFiClient client;

unsigned long myChannelNumber = 234946; // modify this with your own Channel Number
const char * myWriteAPIKey = "G294VC763KY2LBLJ"; // modify this with your own API

void setup() {
  Serial.begin(9600); // for debugging reasons
  WiFi.begin(ssid, pass);
  ThingSpeak.begin(client);
}

void loop() {
  int sensorValue = analogRead(A0);
  Serial.println(sensorValue);
  ThingSpeak.writeField(myChannelNumber, 1, sensorValue, myWriteAPIKey);
  delay(20000);
}
```

Now observe data update of sensor on thingspeak.