UPDATING THINGSPEAK CHANNEL USING DHT11 AND RASPBERRY PI

This manual is about creating and updating ThingSpeak channel using Temperature and Humidity values from DHT11 humidity sensor with Raspberry Pi.
To learn how to connect DHT11 with RaspberryPi download the project form this link:

https://github.com/ashishk7/raspberrydht11.git

1. Components Required

DHT11 Humidity sensor

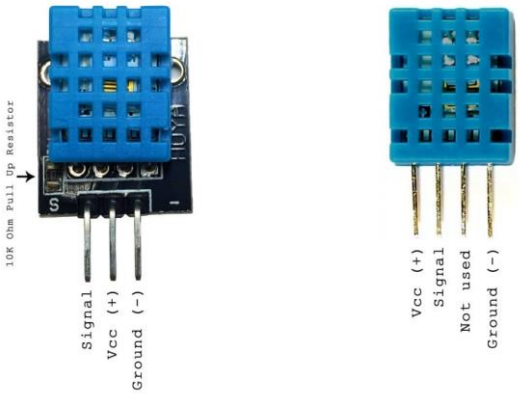
10K Resistor

Raspberry Pi

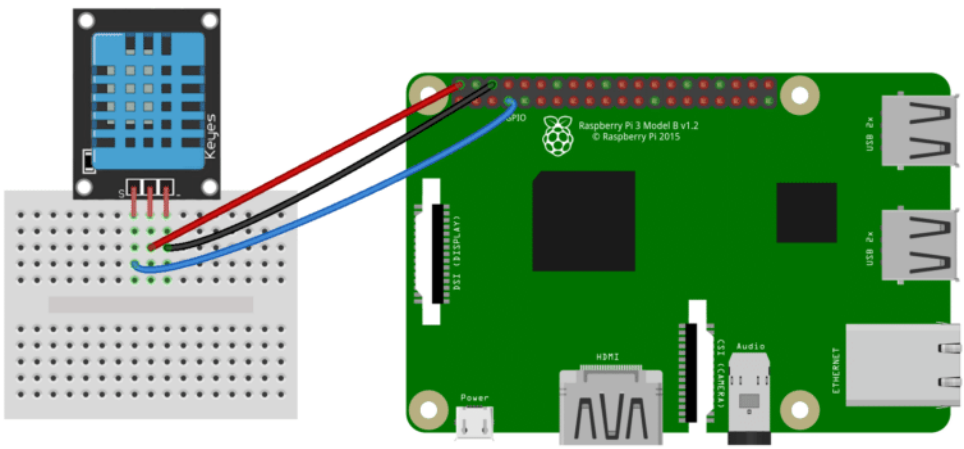
Jumper wires

2. Connecting DHT11 to Raspberry Pi

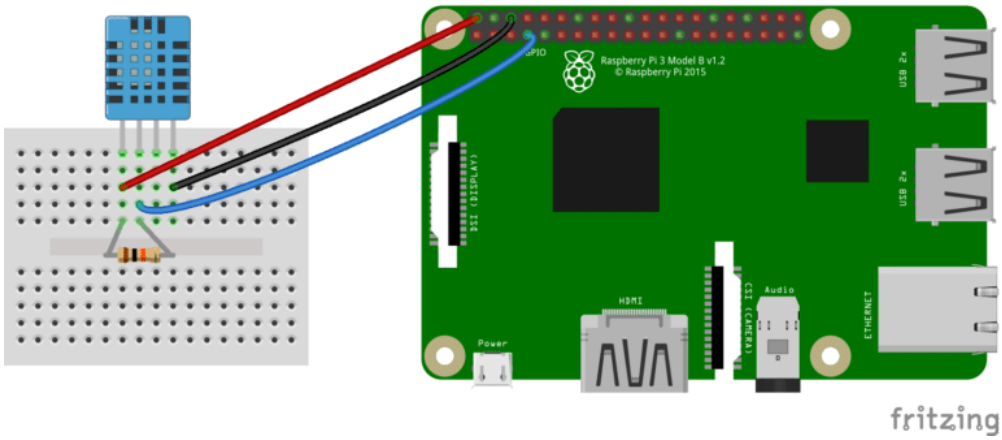
There are two variants of the DHT11 you're likely to come across.



Three pin DHt11 with SSH output:

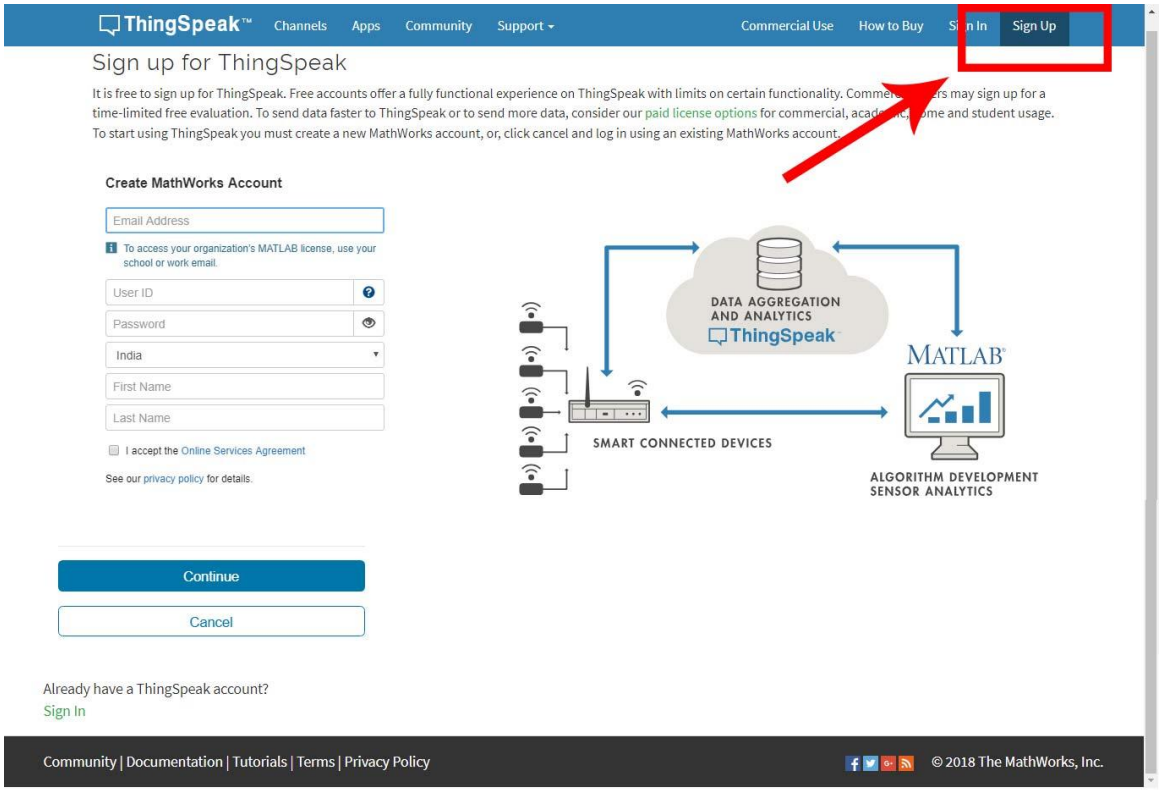


Four pin DHT11 with SSH output:

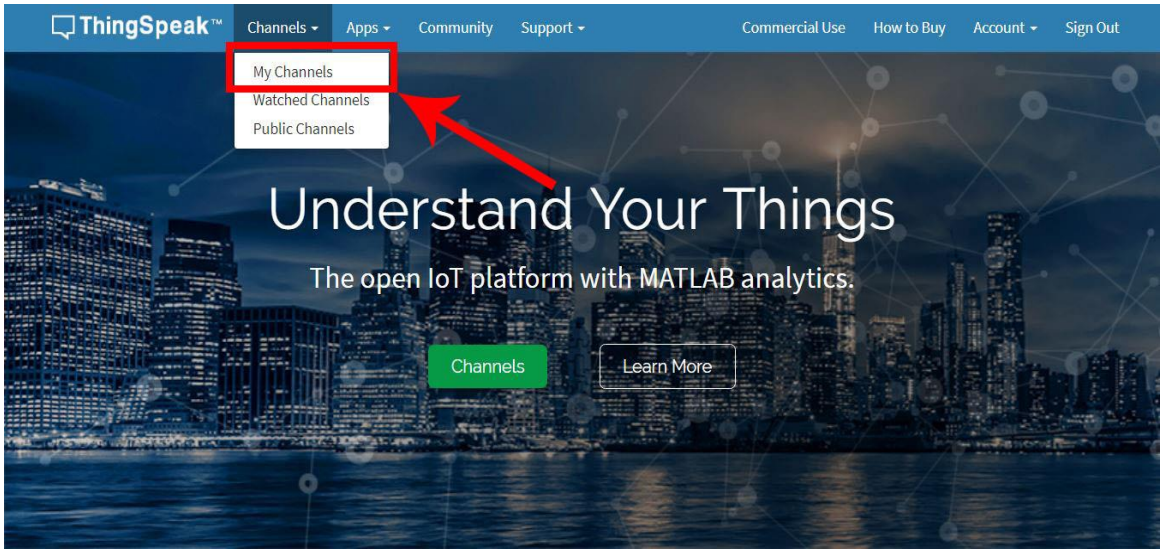


3. Creating Channel on ThingSpeak Server.

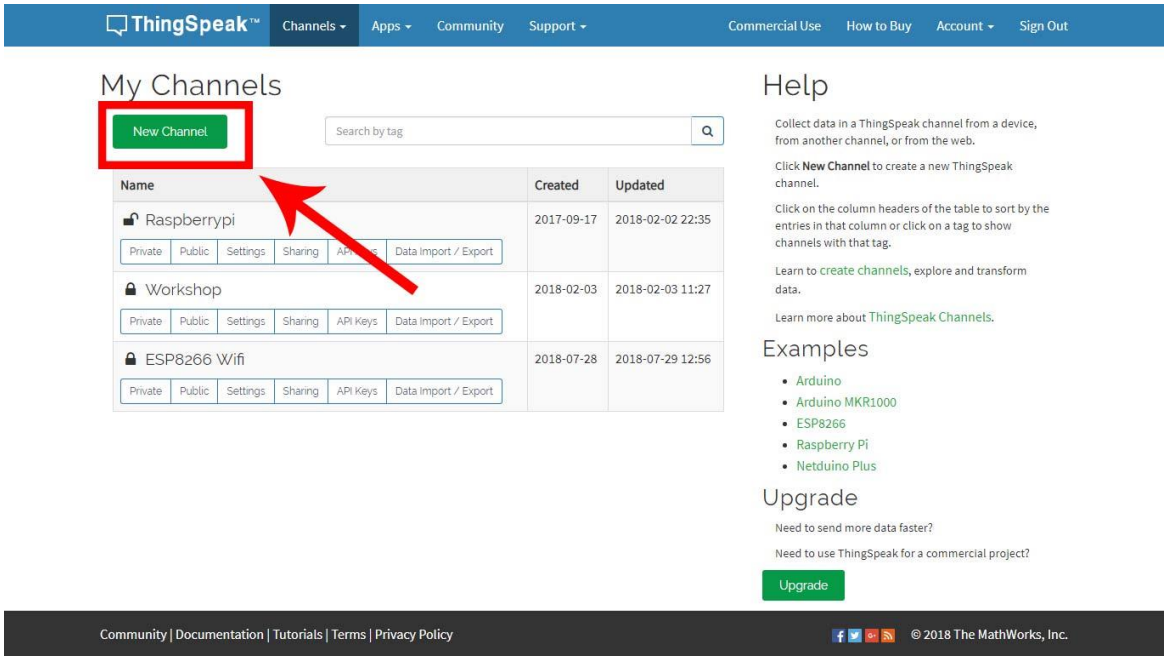
Step 1: SignUp and fill in your details.



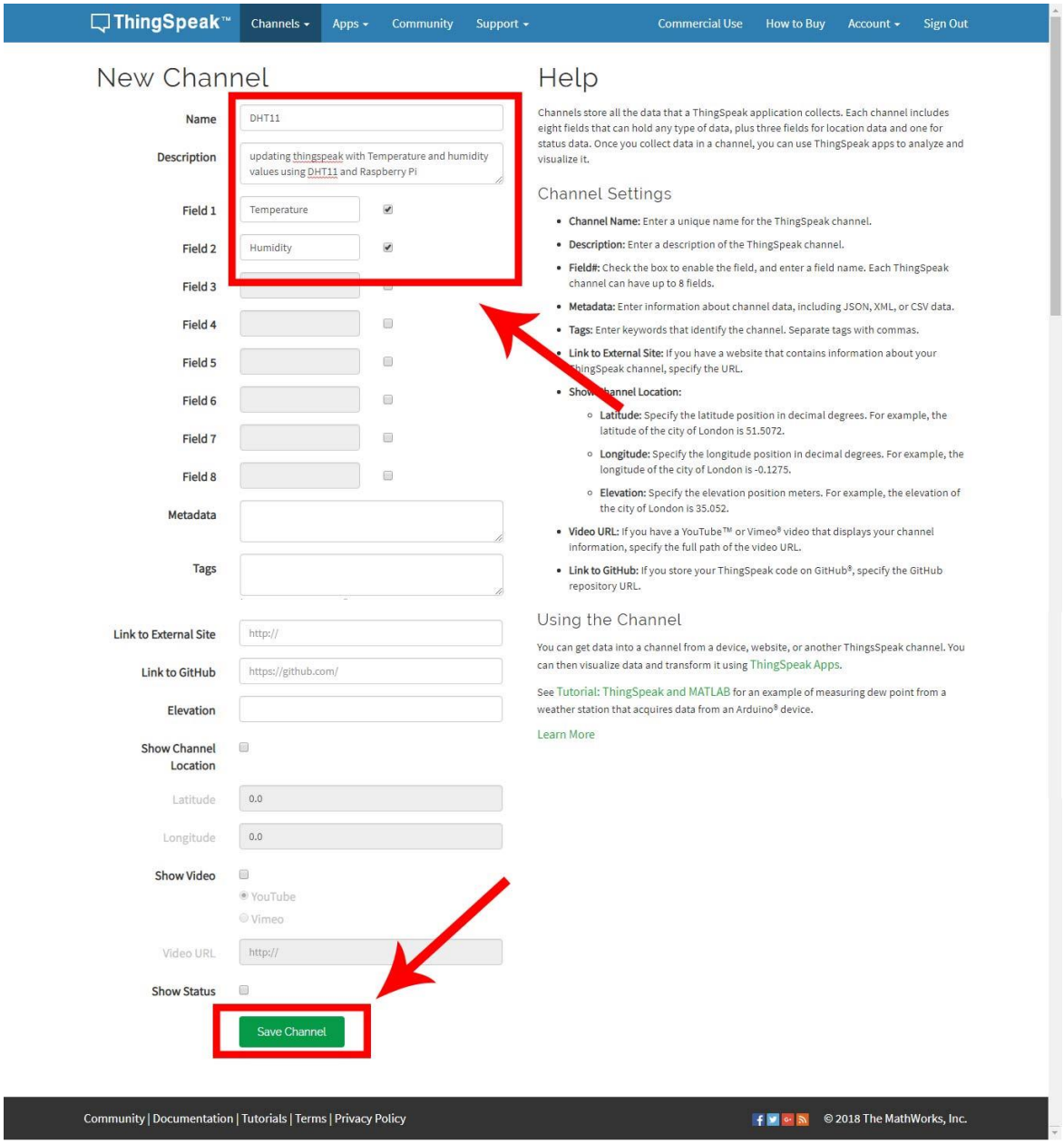
Step 2: Sign in and click on Channels



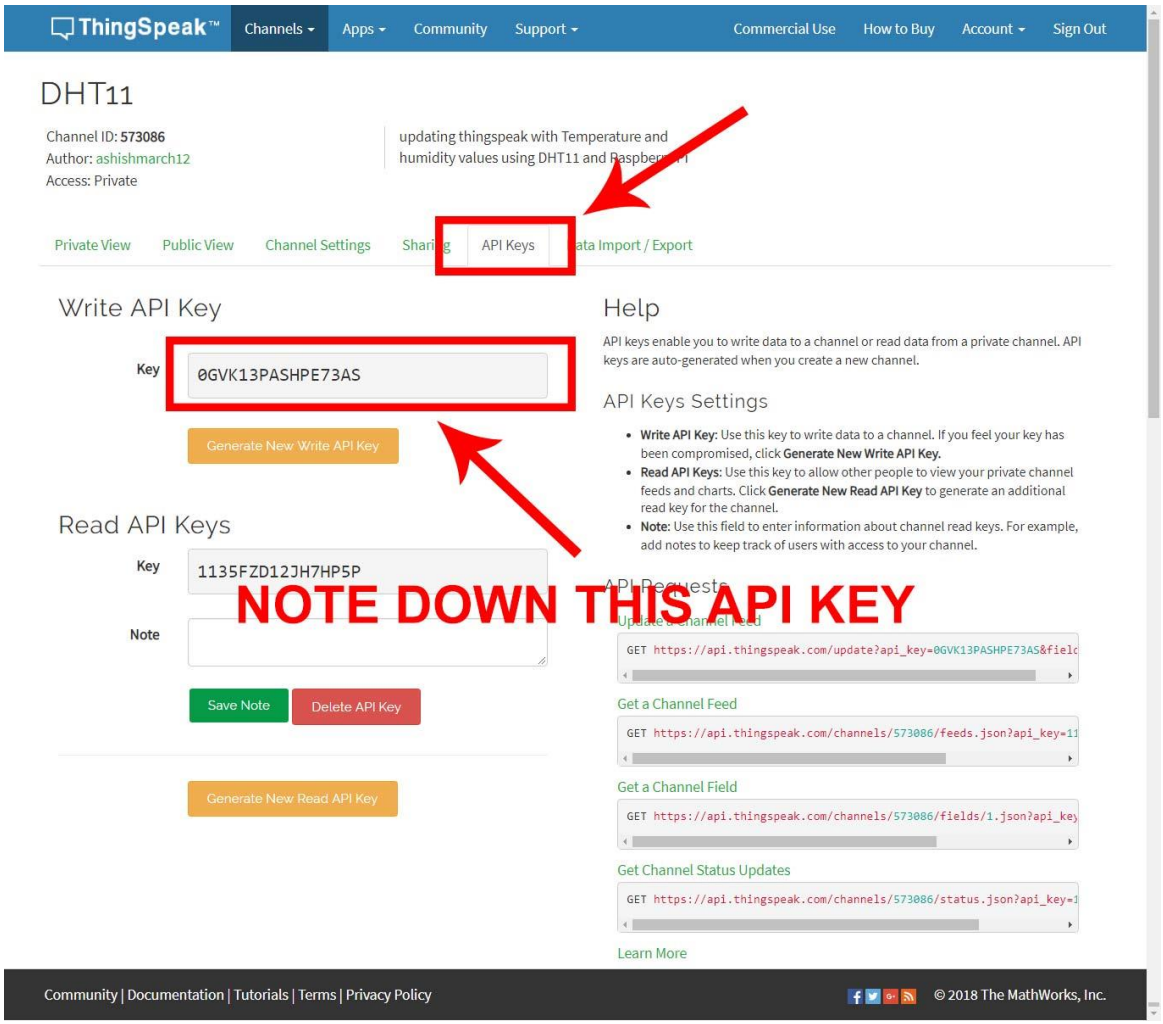
Step 3: Now create new Channel



Step 4: Fill the following details and click on Save Channel.



When channel is created, Select **“API keys”** in the tabs above and note the API key of your channel. See the picture on next page.



4. Installing necessary libraries

We’ll be using the Adafruit DHT11 and psutil Python libraries.

To install the Adafruit DHT11 library:

Enter this at the command prompt to download the library:

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

Change directories with

cd Adafruit_Python_DHT

Now enter this:

sudo apt-get install build-essential python-dev

Then install the library with:

sudo python setup.py install

To install psutil library:

Enter this at the command prompt to download the library:
https://github.com/giampaolo/psutil

Change directories with
cd psutil

Then install the library with:
sudo python setup.py install

To install Paho client library library:

Enter this at the command prompt to install the library:
sudo pip install paho-mqtt

5. Using MQTT on a Raspberry Pi

ThingSpeak has added MQTT as a method to accept channel updates. MQTT is different from HTTP/REST. It is specifically designed to be lightweight and intended for embedded devices with low RAM and CPU performance. Also, in most cases MQTT uses less bandwidth.

On the Raspberry Pi, an MQTT client library is required to send data to ThingSpeak. Paho is an open source MQTT client library that will be used for this examples. It has been adapted to many languages, including Python.

There are three MQTT connection methods demonstrated in this example: TCP, websockets and SSL.

To use a conventional TCP socket on port 1883, set

useUnsecuredTCP=True, This connection method is the simplest and requires the least system resources.

To use websockets on port 80, set

useUnsecuredWebsockets=True, Websockets are helpful when the default MQTT port is blocked on the local network.

Finally, if encryption is required, set

useSSLWebsockets=True , Be sure to only set one on the connection method variables to True and leave the other two False.

Also, be sure to change the **channel ID** and **API key** to your channel's settings.

6. Now write this code in python IDE or directly from terminal of Raspberry PI.

Code :

```
# ThingSpeak Update Using MQTT
# Compiled by Ashish Kumar (ashishmarch12@gmail.com)
# Date: 07 April 2018

from __future__ import print_function
import paho.mqtt.publish as publish
import psutil
import sys
import Adafruit_DHT
import time

# Replace this with your Channel ID
channelID = "XXXXXX"
# The Write API Key for the channel
apiKey = "XXXXXXXXXXXXXXXX"

#  MQTT Connection Methods
useUnsecuredTCP = True
useUnsecuredWebsockets = False
useSSLWebsockets = False

# The Hostname of the ThinSpeak MQTT service
mqttHost = "mqtt.thingspeak.com"

# MQTT connection parameters
if useUnsecuredTCP:
   tTransport = "tcp"
   tPort = 1883
   tTLS = None

if useUnsecuredWebsockets:
   tTransport = "websockets"
   tPort = 80
   tTLS = None

if useSSLWebsockets:
   import ssl
   tTransport = "websockets"
   tTLS = {'ca_certs':"/etc/ssl/certs/ca-certificates.crt",'tls_version':ssl.PROTOCOL_TLSv1}
   tPort = 443

# Create the topic string
topic = "channels/" + channelID + "/publish/" + apiKey

while(True):
   humidity, temperature = Adafruit_DHT.read_retry(11, 4)
   print("Temperature = %s     Humidity = %s" % (temperature, humidity))
   tPayload = "field1=" + str(temperature) + "&field2=" + str(humidity)
```

```
time.sleep(10)
try:
   publish.single(topic, payload=tPayload, hostname=mqttHost, port=tPort, tls=tTLS,
    transport=tTransport)

except (KeyboardInterrupt):
   break

except:
   print ("There was an error while publishing the data.")
```