

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

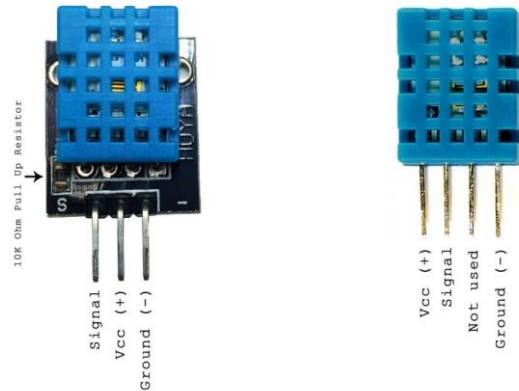
10K Resistor

Raspberry Pi

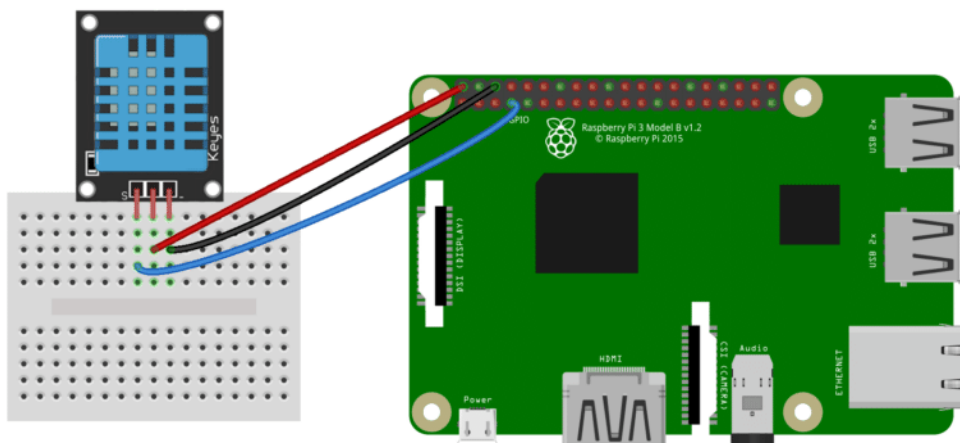
Jumper wires

2. Connecting DHT11 to Raspberry Pi

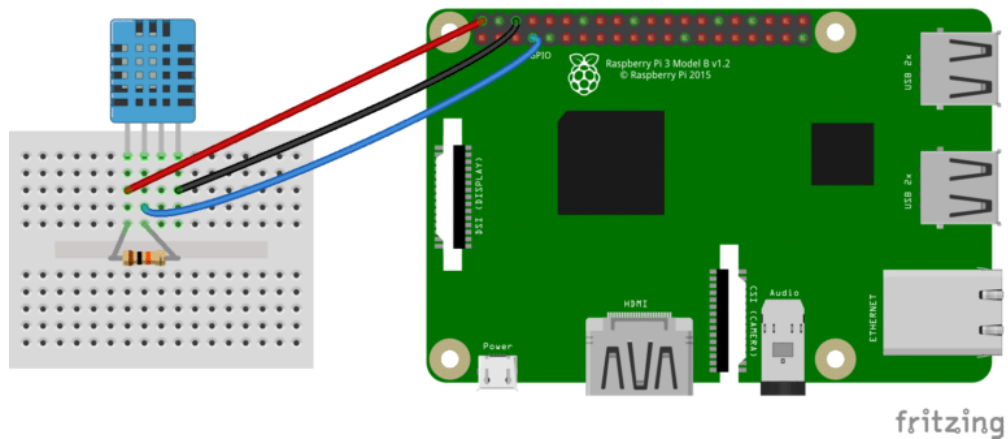
There are two variants of the DHT11 you're likely to come across.



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.

ThingSpeak™ Channels Apps Community Support ▾ Commercial Use How to Buy Sign In **Sign Up**

Sign up for ThingSpeak

It is free to sign up for ThingSpeak. Free accounts offer a fully functional experience on ThingSpeak with limits on certain functionality. Commercial users may sign up for a time-limited free evaluation. To send data faster to ThingSpeak or to send more data, consider our [paid license options](#) for commercial, academic, home and student usage. To start using ThingSpeak you must create a new MathWorks account, or, click cancel and log in using an existing MathWorks account.

Create MathWorks Account

Email Address

i To access your organization's MATLAB license, use your school or work email.

User ID ⓘ

Password ⓘ

India ▾

First Name

Last Name

☐ I accept the Online Services Agreement

[See our privacy policy for details.](#)

Already have a ThingSpeak account?
[Sign In](#)

DATA AGGREGATION AND ANALYTICS
ThingSpeak

SMART CONNECTED DEVICES

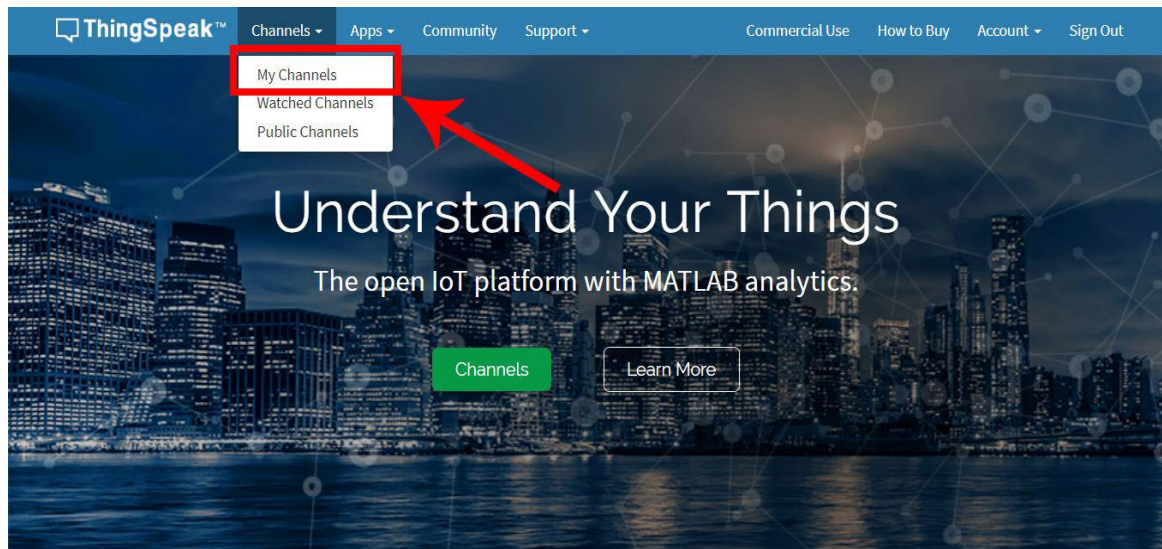
MATLAB®

ALGORITHM DEVELOPMENT
SENSOR ANALYTICS

Community | Documentation | Tutorials | Terms | Privacy Policy

© 2018 The MathWorks, Inc.

Step 2: Sign in and click on Channels



Step 3: Now create new Channel

The screenshot shows the 'My Channels' page in ThingSpeak. The 'New Channel' button is highlighted with a red box, and a red arrow points to it. Below the button is a search bar labeled 'Search by tag'. A table lists the user's channels:

Name	Created	Updated
Raspberrypi Private Public Settings Sharing API Keys Data Import / Export	2017-09-17	2018-02-02 22:35
Workshop Private Public Settings Sharing API Keys Data Import / Export	2018-02-03	2018-02-03 11:27
ESP8266 Wifi Private Public Settings Sharing API Keys Data Import / Export	2018-07-28	2018-07-29 12:56

On the right side of the page, there is a 'Help' section with instructions on how to create a new channel, an 'Examples' section with links to various IoT projects, and an 'Upgrade' section with an 'Upgrade' button.

Community | Documentation | Tutorials | Terms | Privacy Policy

© 2018 The MathWorks, Inc.

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

The screenshot shows the 'New Channel' form on the ThingSpeak website. The form includes fields for Name, Description, eight Fields, Metadata, Tags, Link to External Site, Link to GitHub, Elevation, Show Channel Location (with Latitude and Longitude), Show Video (with YouTube and Vimeo options and a Video URL), and Show Status. A red box highlights the 'Name' field (containing 'DHT11'), the 'Description' field (containing 'updating thingspeak with Temperature and humidity values using DHT11 and Raspberry Pi'), 'Field 1' (containing 'Temperature'), and 'Field 2' (containing 'Humidity'). A red arrow points from this box to the 'Save Channel' button, which is also highlighted with a red box. Another red arrow points from the 'Show Channel Location' section to the 'Save Channel' button. The 'Save Channel' button is a green button with white text.

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

New Channel

Name DHT11

Description updating thingspeak with Temperature and humidity values using DHT11 and Raspberry Pi

Field 1 Temperature ☒

Field 2 Humidity ☒

Field 3 ☐

Field 4 ☐

Field 5 ☐

Field 6 ☐

Field 7 ☐

Field 8 ☐

Metadata

Tags

Link to External Site

Link to GitHub

Elevation

Show Channel Location ☐

Latitude

Longitude

Show Video ☐

☒ YouTube ☐ Vimeo

Video URL

Show Status ☐

Save Channel

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.
- Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.
- Show Channel Location:**
 - Latitude:** Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5072.
 - Longitude:** Specify the longitude position in decimal degrees. For example, the longitude of the city of London is -0.1275.
 - Elevation:** Specify the elevation position meters. For example, the elevation of the city of London is 35.052.
- Video URL:** If you have a YouTube™ or Vimeo® video that displays your channel information, specify the full path of the video URL.
- Link to GitHub:** If you store your ThingSpeak code on GitHub®, specify the GitHub repository URL.

Using the Channel

You can get data into a channel from a device, website, or another ThingSpeak channel. You can then visualize data and transform it using [ThingSpeak Apps](#).

See [Tutorial: ThingSpeak and MATLAB](#) for an example of measuring dew point from a weather station that acquires data from an Arduino® device.

[Learn More](#)

Community | Documentation | Tutorials | Terms | Privacy Policy

© 2018 The MathWorks, Inc.

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.

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

DHT11

Channel ID: 573086
Author: ashishmarch12
Access: Private

updating thingspeak with Temperature and humidity values using DHT11 and Raspberry Pi

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

Write API Key

Key: 0GVK13PASHPE73AS

Generate New Write API Key

Read API Keys

Key: 1135FZD12JH7HP5P

Note:

Save Note Delete API Key

Generate New Read 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.

API Requests

Update Channel Feed

```
GET https://api.thingspeak.com/update?api_key=0GVK13PASHPE73AS&field=
```

Get a Channel Feed

```
GET https://api.thingspeak.com/channels/573086/feeds.json?api_key=1135FZD12JH7HP5P
```

Get a Channel Field

```
GET https://api.thingspeak.com/channels/573086/fields/1.json?api_key=1135FZD12JH7HP5P
```

Get Channel Status Updates

```
GET https://api.thingspeak.com/channels/573086/status.json?api_key=1135FZD12JH7HP5P
```

Learn More

Community | Documentation | Tutorials | Terms | Privacy Policy

© 2018 The MathWorks, Inc.

4. Installing necessary libraries

We'll be using the Adafruit DHT11 and putil 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 = "XXXXXXXXXXXXXXXXXXXX"

# 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.")
```