

# **Internet of Things Master Class Day 6**

**M.K.Jeevarajan**  
**[www.pantechsolutions.net](http://www.pantechsolutions.net)**

# What you will learn Today



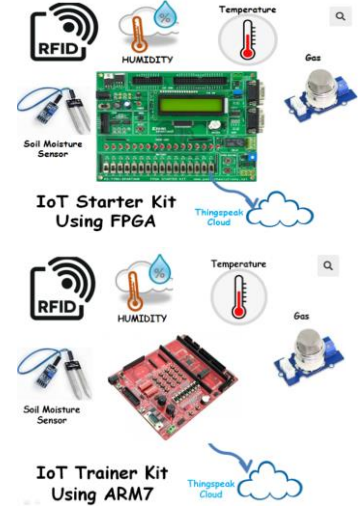
Sending Data to eps32 -Thingspeak -Arduino+Humidity+Temperature(Weather monitoring system)

- ✓ **How to test the ESP32 Board**
- ✓ **Programming DHT22**
- ✓ **Programming Thingspeak**
- ✓ **Integrating DHT22 and Thingspeak**
- ✓ **Visualizing data on Thingspeak**



# About Pantech Prolabs India Pvt Ltd

- ✓ Started in the Year 2004
- ✓ Lab equipments and Sensor Interface
- ✓ Manufacturer of Brainsense EEG Headset
- ✓ Reconfigurable Algorithms on AI
- ✓ Manufacture of AI development Boards
- ✓ Power electronics, Fuel cell and Renewable Energy trainers



## Vision

To help 10 Million students around the globe to learn technology in a easy way

[www.pantechsolutions.net](http://www.pantechsolutions.net)

# About me



## Education



College of Engineering, Guindy

Masters of Engineering, Applied Electronics

2002 – 2004



Govt College of Engg, Bargur

Bachelor of Engineering (B.E.), Electrical, Electronics and Communications Engineering, A

1998 – 2002

## My Primary Expertise

Microcontroller Architecture: 8051, PIC, AVR, ARM, MSP430, PSOC3

DSP Architecture: Blackfin, C2000, C6000, 21065L Sharc

FPGA: Spartan, Virtex, Cyclone

Image Processing Algorithms: Image/Scene Recognition, Machine Learning, Computer Vision, Deep Learning, Pattern Recognition, Object Classification, Image Retrieval, Image enhancement and denoising.

Neural Networks : SVM, RBF, BPN

Cryptography : RSA, DES, 3DES, Elliptic curve, Blowfish, Diffie Hellman

Compilers: Keil, Visual DSP++, CCS, Xilinx Platform studio, ISE, Matlab, Open CV

<https://www.linkedin.com/in/jeevarajan/>

[www.pantechsolutions.net](http://www.pantechsolutions.net)

# Announcement

- Attendance Link at 9 pm
- Minimum attendance required for an E-Certificate is 27 Days. Attendance link will be valid for 1 hrs. after the event.
- For Internship Candidates no attendance required ,it will be accessed from the LMS Portal. ([learn.pantechsolutions.net](http://learn.pantechsolutions.net))
- Recorded Video Streaming for LAB classes to improve Learning Experience
- PPT in facebook group
- Source code and projects available download only for Internship candidates

# Mindset Lesson for the Day

Focus is the Most Valuable asset not the Time.

Learn How to Focus.

Practice Focus.

# **Weather Monitoring System using ESP32 and Thingspeak**

# Hardware & Software Required

- ESP32
- DHT22(Humidity+Temperature)
- 10K Resistor -1 Qty
- Breadboard
- Jumper wires
- USB cable for Programming(ESP32)
- Arduino IDE
- Thingspeak account



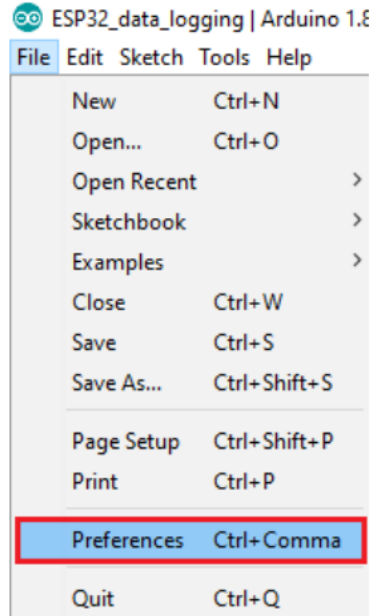
## Prerequisites: Arduino IDE Installed

- Before starting this installation procedure, make sure you have the latest version of the Arduino IDE installed in your computer. If you don't, uninstall it and install it again. Otherwise, it may not work.
- Having the latest Arduino IDE software installed from [arduino.cc/en/Main/Software](https://arduino.cc/en/Main/Software), continue with this tutorial.

# TESTING THE ESP32 BOARD

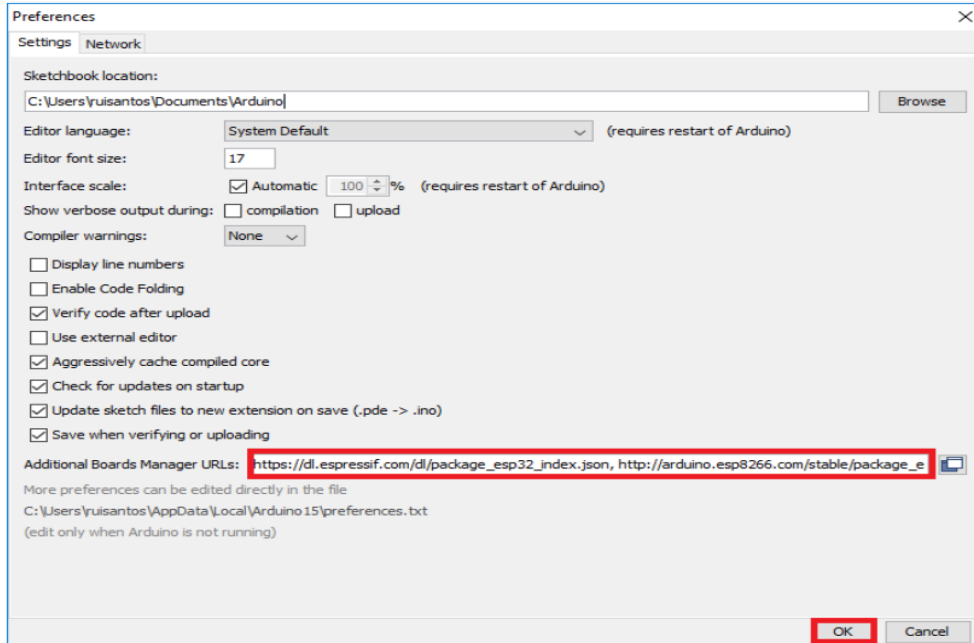
# Step 1- Installing ESP32 Add-on in Arduino IDE

1. In your Arduino IDE, go to **File> Preferences**



# Step 2

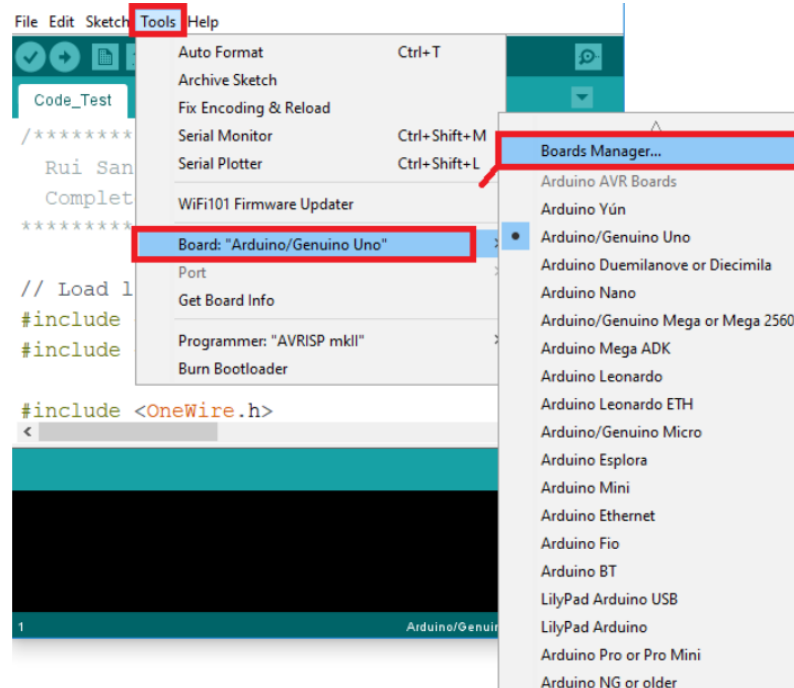
- Enter [https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json) into the “Additional Board Manager URLs” field as shown in the figure below. Then, click the “OK” button:



**Note:** if you already have the ESP8266 boards URL, you can separate the URLs with a comma as follows:  
[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json),  
[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

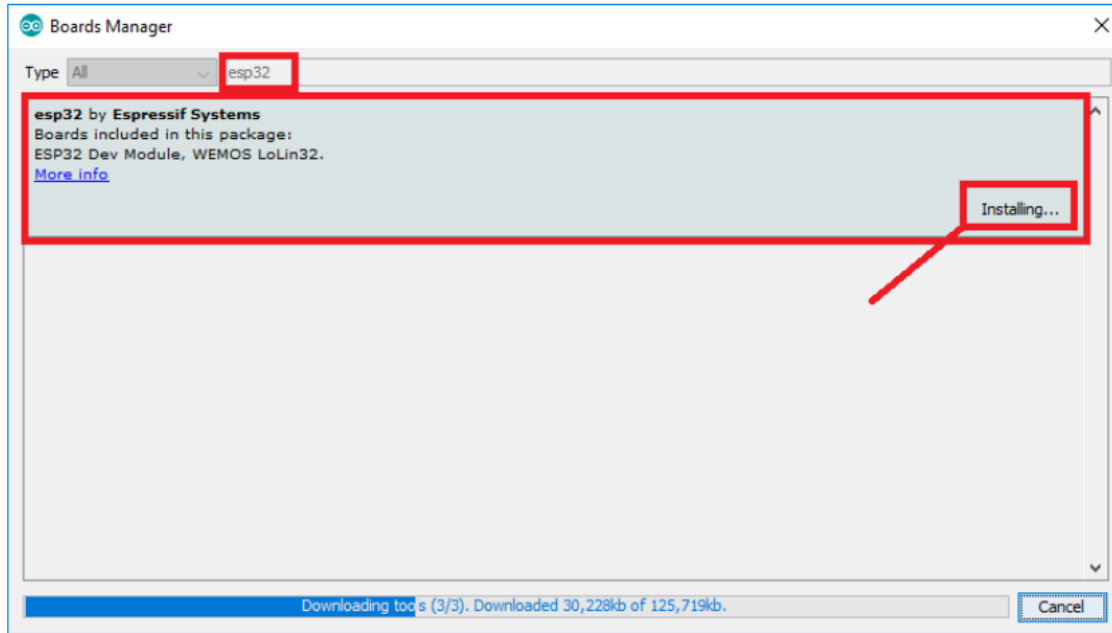
# Step 3

- Open the Boards Manager. Go to Tools > Board > Boards Manager...



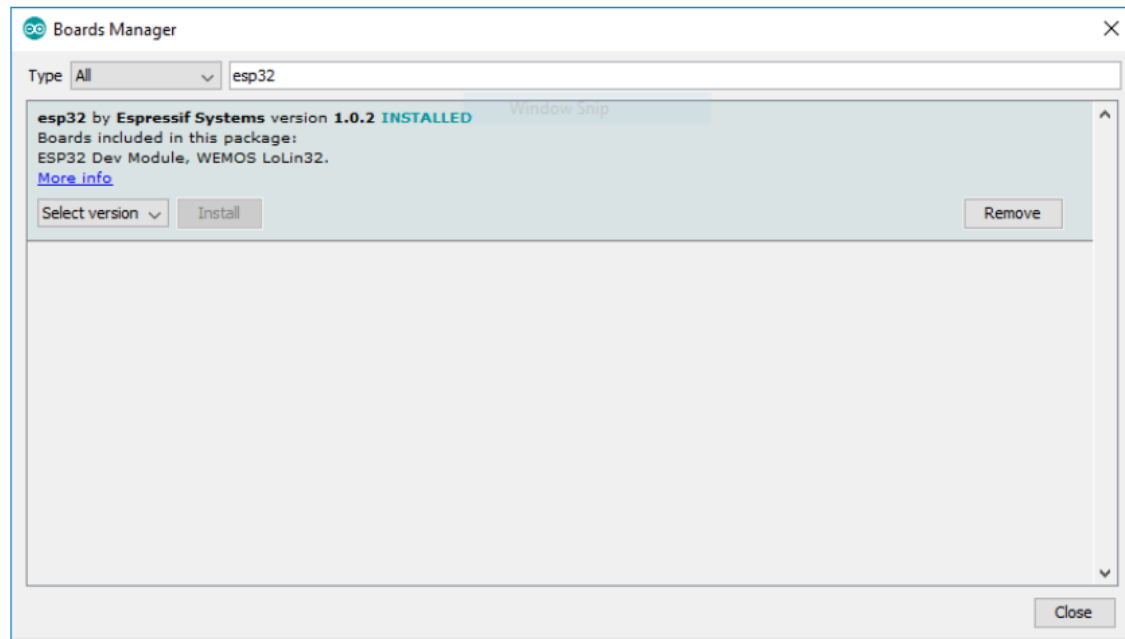
# Step 4

4. Search for **ESP32** and press install button for the “**ESP32 by Espressif Systems**”:



# Step 5

5. That's it. It should be installed after a few seconds.

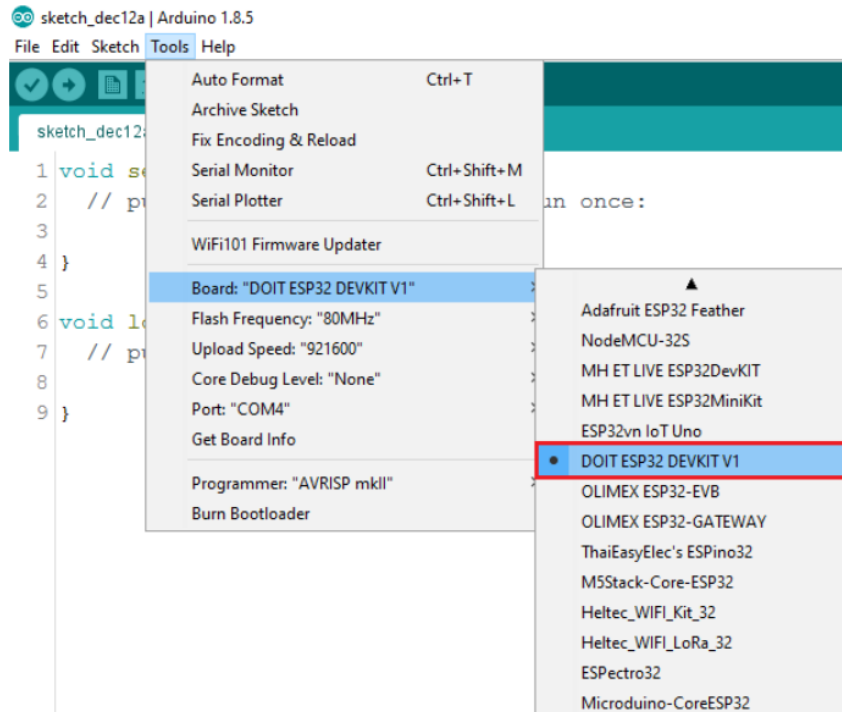


# Testing the Installation

- Plug the ESP32 board to your computer. With your Arduino IDE open, follow these steps:
- 1. Select your Board in **Tools > Board** menu (in my case it's the **DOIT ESP32 DEVKIT V1**)

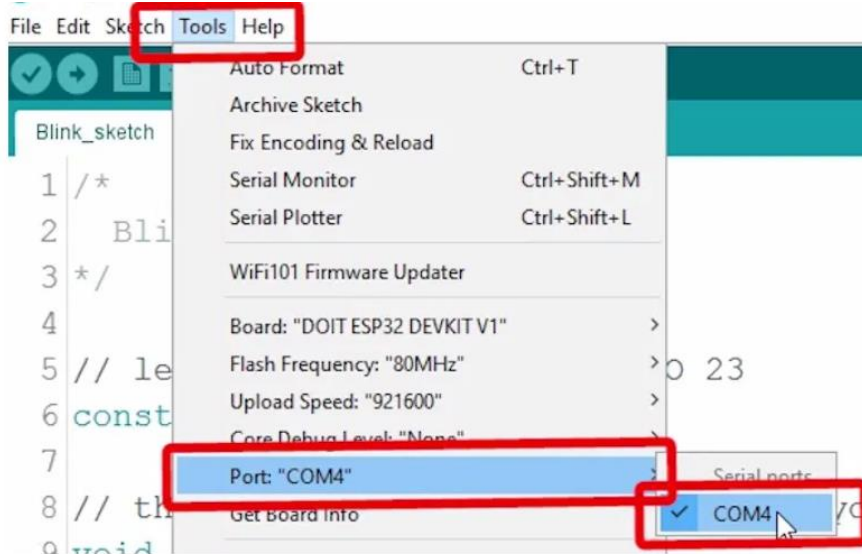


# Step 1



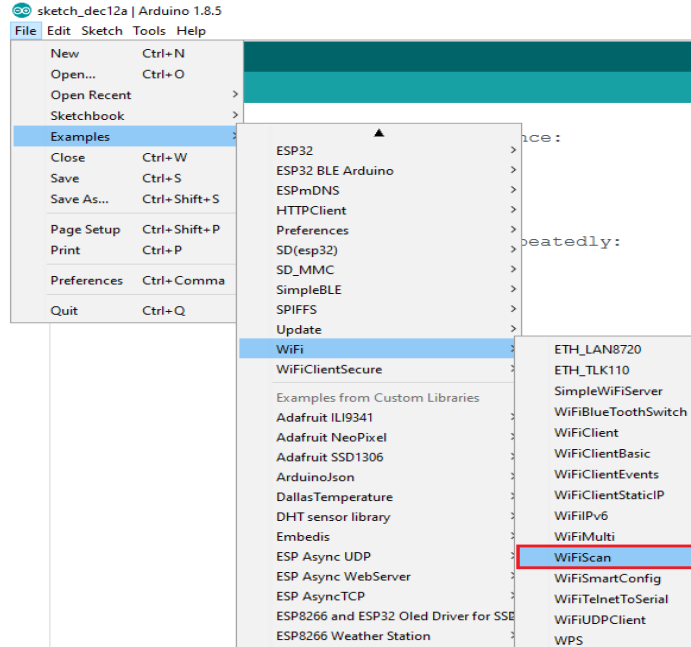
# Step 2

- 2. Select the Port (if you don't see the COM Port in your Arduino IDE, you need to install the CP210x USB to UART Bridge VCP Drivers):



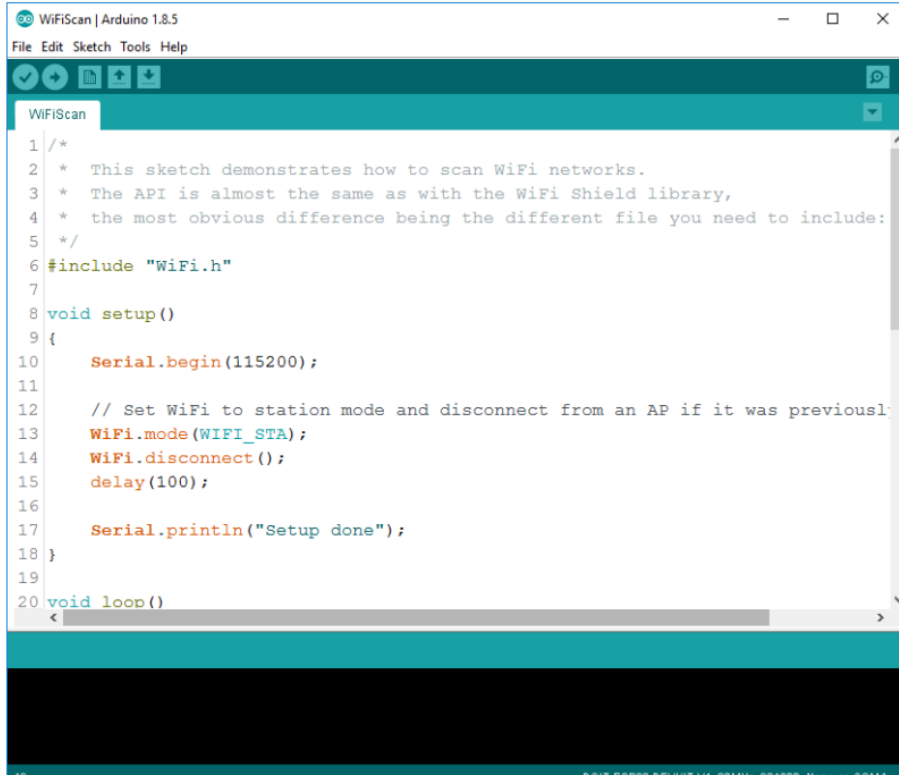
# Step 3

- 3. Open the following example under File > Examples > WiFi (ESP32) > WiFiScan



# Step 4

- 4. A new sketch opens in your Arduino IDE:

A screenshot of the Arduino IDE interface. The title bar reads 'WiFiScan | Arduino 1.8.5'. The menu bar includes 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. Below the menu bar is a toolbar with icons for opening, saving, and running. The main text area shows the following code:

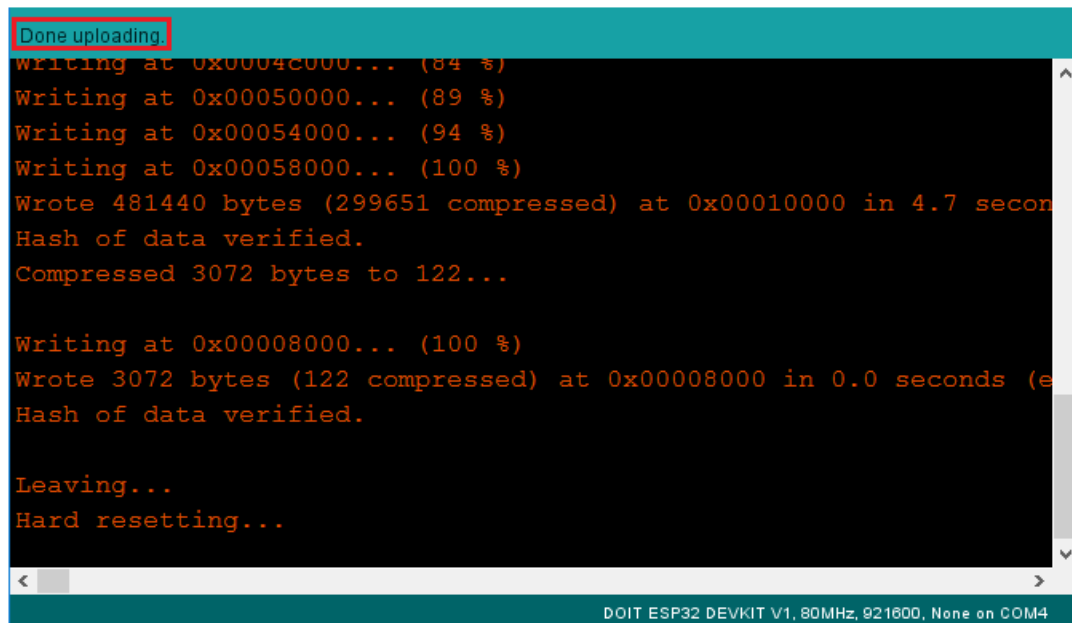
```
1 /*
2  * This sketch demonstrates how to scan WiFi networks.
3  * The API is almost the same as with the WiFi Shield library,
4  * the most obvious difference being the different file you need to include:
5  */
6 #include "WiFi.h"
7
8 void setup()
9 {
10     Serial.begin(115200);
11
12     // Set WiFi to station mode and disconnect from an AP if it was previously
13     WiFi.mode(WIFI_STA);
14     WiFi.disconnect();
15     delay(100);
16
17     Serial.println("Setup done");
18 }
19
20 void loop()
```

# Step 5

- 5. Press the **Upload** button in the Arduino IDE. Wait a few seconds while the code compiles and uploads to your board.

# Step 6

6. If everything went as expected, you should see a “**Done uploading.**” message.



```
Done uploading.
Writing at 0x0004c000... (84 %)
Writing at 0x00050000... (89 %)
Writing at 0x00054000... (94 %)
Writing at 0x00058000... (100 %)
Wrote 481440 bytes (299651 compressed) at 0x00010000 in 4.7 seconds
Hash of data verified.
Compressed 3072 bytes to 122...

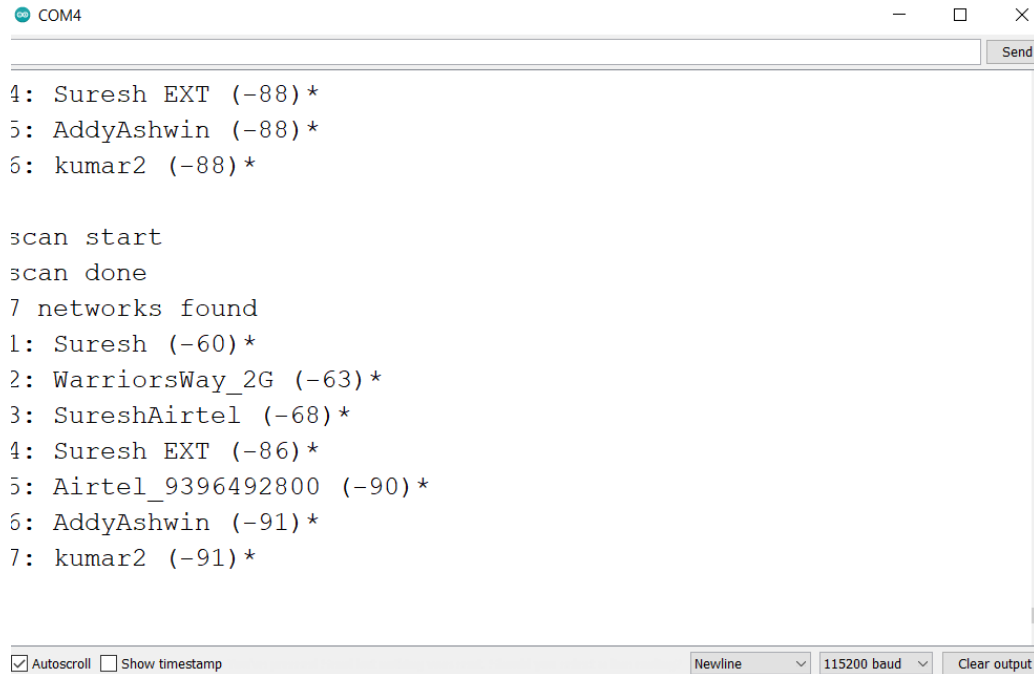
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (122 compressed) at 0x00008000 in 0.0 seconds (effective 12.5 KiB/s)
Hash of data verified.

Leaving...
Hard resetting...
```

DOIT ESP32 DEVKIT V1, 80MHz, 921600, None on COM4

# Step 7

- 7. Open the Arduino IDE Serial Monitor at a baud rate of 115200:



The screenshot shows the Arduino IDE Serial Monitor window for COM4. The window title is "COM4" with standard window controls. The main text area displays the output of a network scan. The output is as follows:

```
4: Suresh EXT (-88)*
5: AddyAshwin (-88)*
6: kumar2 (-88)*

scan start
scan done
7 networks found
1: Suresh (-60)*
2: WarriorsWay_2G (-63)*
3: SureshAirtel (-68)*
4: Suresh EXT (-86)*
5: Airtel_9396492800 (-90)*
6: AddyAshwin (-91)*
7: kumar2 (-91)*
```

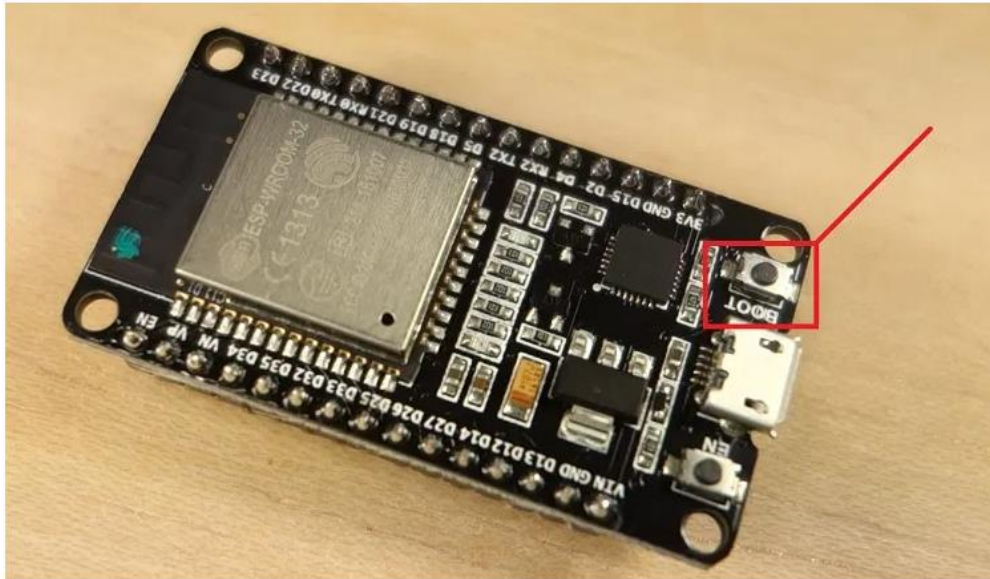
At the bottom of the window, there are controls: a checked "Autoscroll" checkbox, an unchecked "Show timestamp" checkbox, a "Newline" dropdown menu, a "115200 baud" dropdown menu, and a "Clear output" button.

# Troubleshooting -tips

If you try to upload a new sketch to your ESP32 and you get this error message “*A fatal error occurred: Failed to connect to ESP32: Timed out... Connecting...*”. It means that your ESP32 is not in flashing/uploading mode.

Having the right board name and COM port selected, follow these steps:

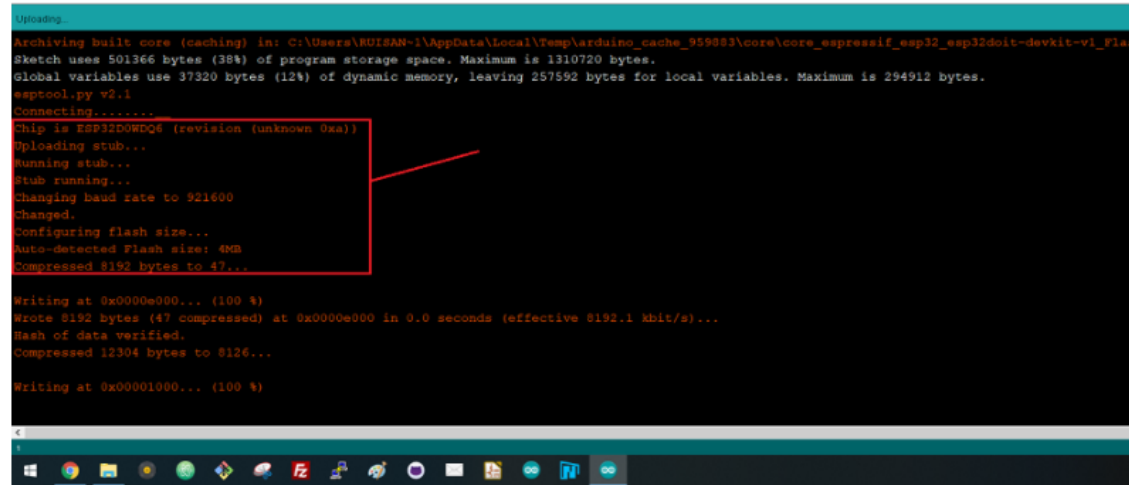
- Hold-down the “**BOOT**” button in your ESP32 board





# Troubleshooting -tips

- Press the “Upload” button in the Arduino IDE to upload your sketch:
  - After you see the “Connecting....” message in your Arduino IDE, release the finger from the “BOOT” button:



```
Uploading...
Archiving built core (caching) in: C:\Users\BUTSMN-1\AppData\Local\Temp\arduino_cache_959883\core\core_espressif_esp32_esp32doit-devkit-v1_Flash
Sketch uses 501366 bytes (38%) of program storage space. Maximum is 1310720 bytes.
Global variables use 37320 bytes (12%) of dynamic memory, leaving 257592 bytes for local variables. Maximum is 294912 bytes.
esptool.py v2.1
Connecting.....
Chip is ESP32D0WDQ6 (revision (unknown 0aa))
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 921600
Changed.
Configuring flash size...
Auto-detected Flash size: 4MB
Compressed 8192 bytes to 47...

Writing at 0x00000000... (100 %)
Wrote 8192 bytes (47 compressed) at 0x00000000 in 0.0 seconds (effective 8192.1 kbit/s)...
Hash of data verified.
Compressed 12304 bytes to 8126...



Writing at 0x00001000... (100 %)
```

- After that, you should see the “Done uploading” message

# TESTING THE DHT22

# DHT11 VS DHT22

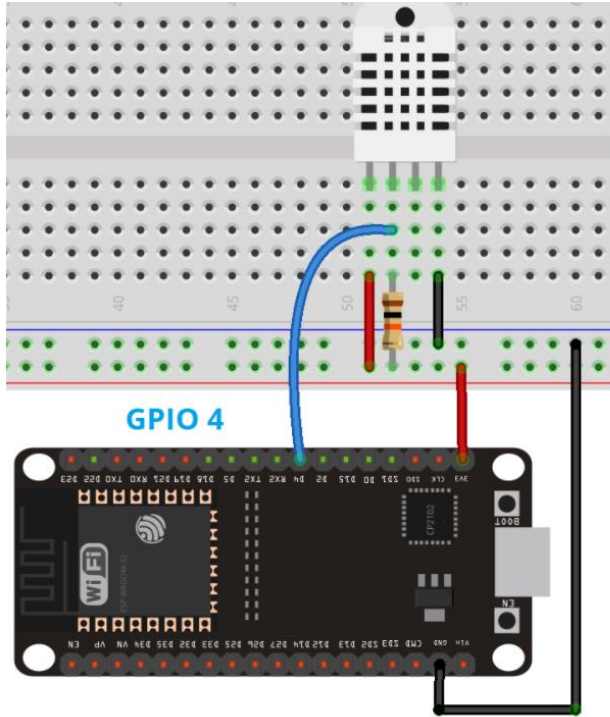
- Digital-output relative humidity & temperature sensor/module.

	DHT11	DHT22
		
Temperature range	0 to 50 °C $\pm 2$ °C	-40 to 80 °C $\pm 0.5$ °C
Humidity range	20 to 90% $\pm 5$ %	0 to 100% $\pm 2$ %
Resolution	Humidity: 1% Temperature: 1°C	Humidity: 0.1% Temperature: 0.1°C
Operating voltage	3 – 5.5 V DC	3 – 6 V DC
Current supply	0.5 – 2.5 mA	1 – 1.5 mA
Sampling period	1 second	2 seconds

# PIN DETAILS

DHT pin	Connect to
1	3.3V
2	Any digital GPIO; also connect a 10k Ohm pull-up resistor
3	Don't connect
4	GND

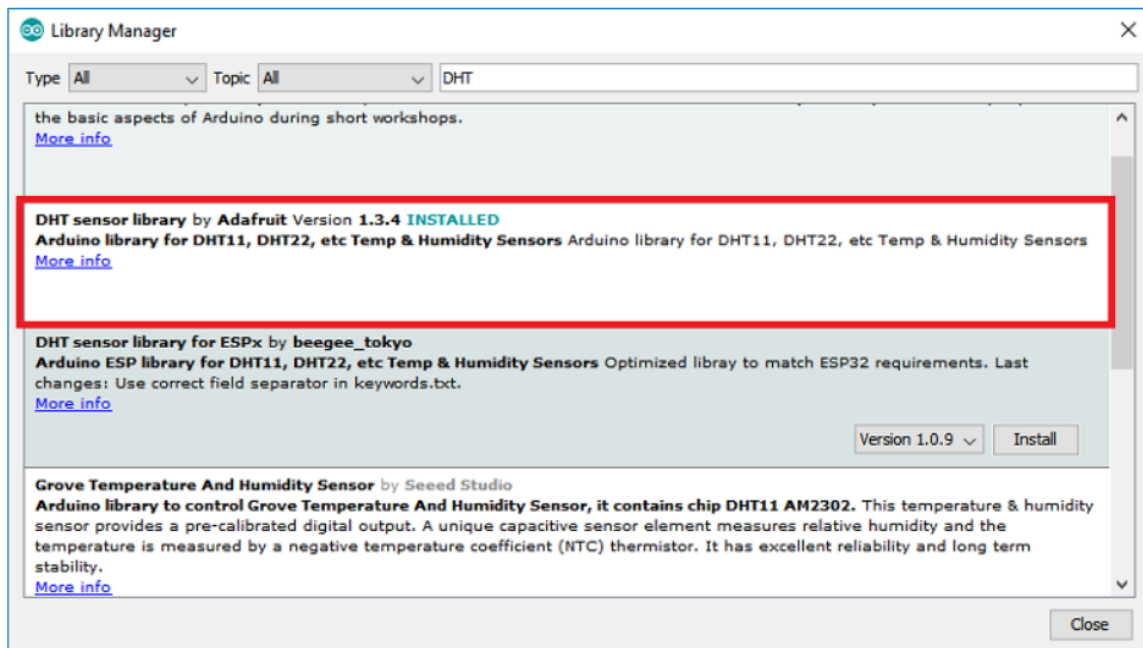
# Schematic Diagram



In this example, we're connecting the DHT data pin to `2`. However, you can use any other suitable digital pin.

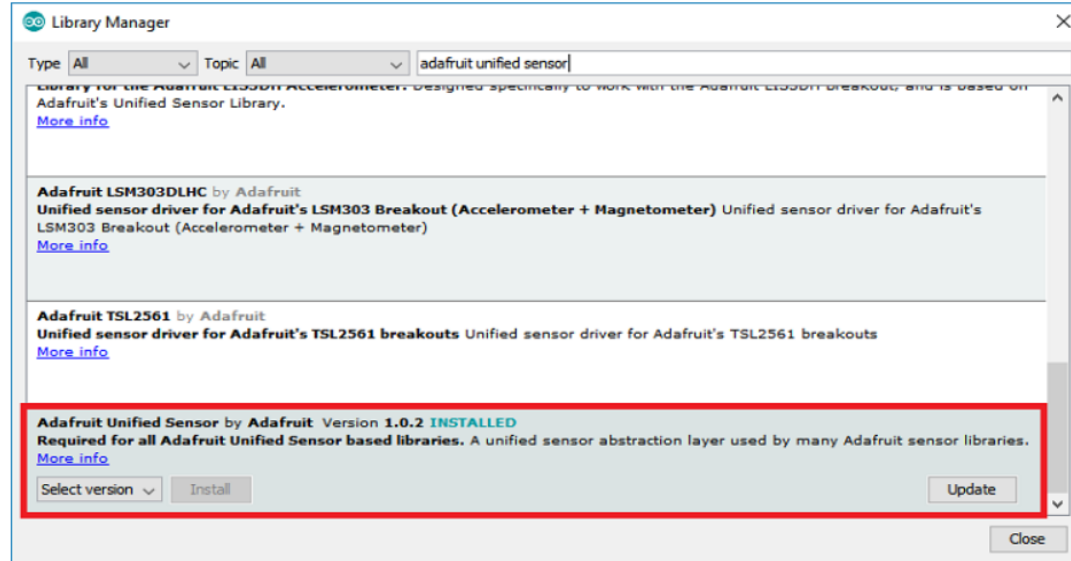
# Library required

Search for “DHT” on the Search box and install the DHT library from Adafruit.



# Library required

After installing the DHT library from Adafruit, type “**Adafruit Unified Sensor**” in the search box. Scroll all the way down to find the library and install it.



- After installing the libraries, restart your Arduino IDE.

**Download TEST Code**

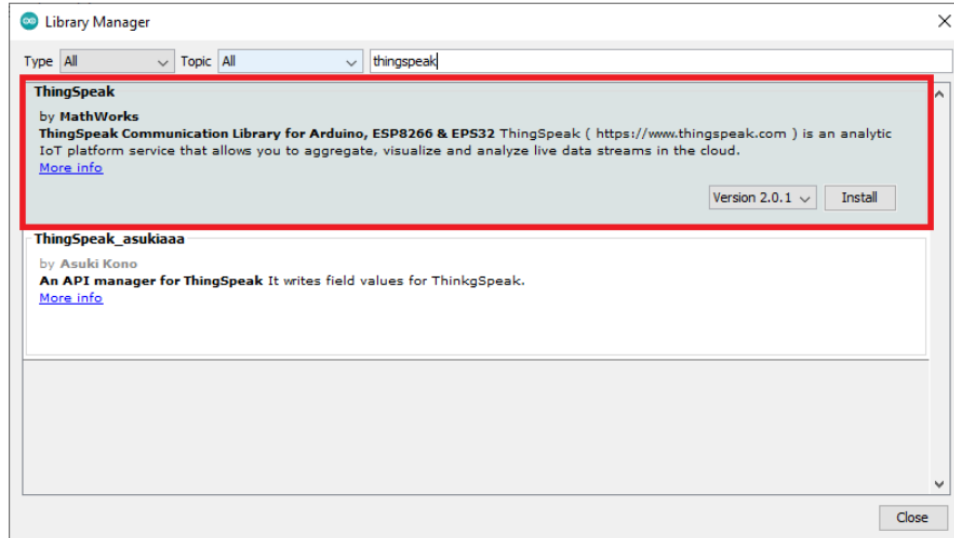


# PUBLISH DATA TO THINGSPEAK

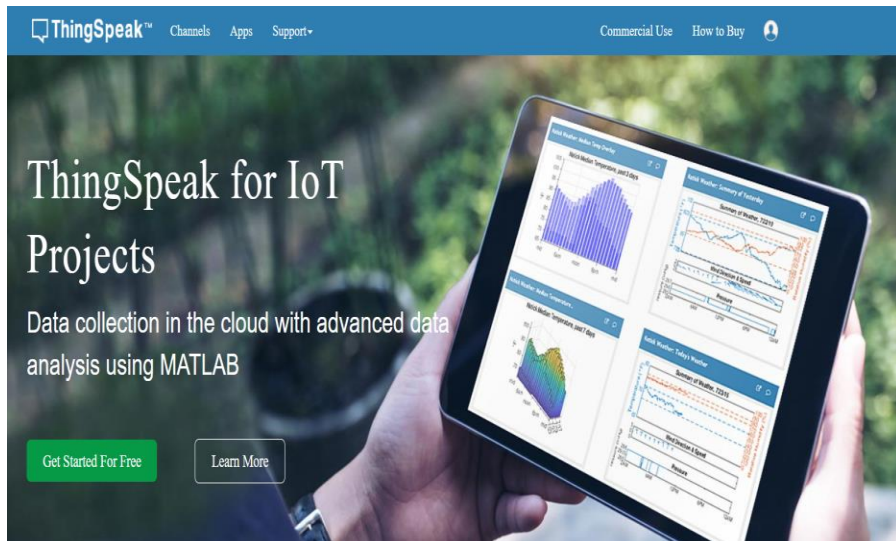


# Installing the ThingSpeak Library

To send sensor readings to ThingSpeak, we'll use the thingspeak-arduino library. You can install this library through the Arduino Library Manager. Go to **Sketch > Include Library > Manage Libraries...** and search for “**ThingSpeak**” in the Library Manager. Install the ThingSpeak library by MathWorks.



# Create an Account



To use ThingSpeak, you must sign in with your existing MathWorks account or create a new one.

Non-commercial users may use ThingSpeak for free. Free accounts offer limits on certain functionality. Commercial users are eligible for a time-limited free evaluation. To get full access to the MATLAB analysis features on ThingSpeak, log in to ThingSpeak using the email address associated with your university or organization.

To send data faster to ThingSpeak or to send more data from more devices, consider the [paid license options](#) for commercial, academic, home and student usage.

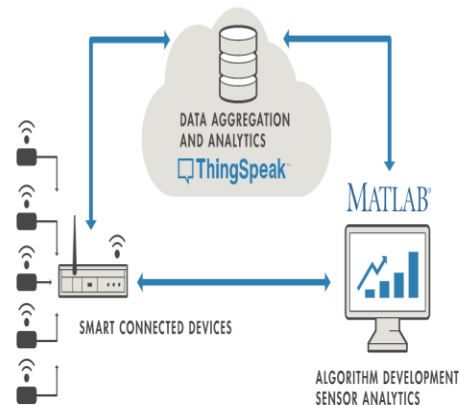


Email

No account? [Create one!](#)

By signing in you agree to our [privacy policy](#).

Next



<https://thingspeak.com/>

# Home Page

[Channels ▾](#)[Apps ▾](#)[Support ▾](#)[Commercial Use](#)[How to Buy](#)[JR](#)

Signed in successfully.



## My Channels

[New Channel](#)

Name	Created	Updated
<div> Test</div> <div><a href="#">Private</a> <a href="#">Public</a> <a href="#">Settings</a> <a href="#">Sharing</a> <a href="#">API Keys</a> <a href="#">Data Import / Export</a></div>	2020-05-25	2020-05-25 10:25

## Help

Collect data in a ThingSpeak channel from a device, from another channel, or from the web.

Click **New Channel** to create a new ThingSpeak channel.

Click on the column headers of the table to sort by the entries in that column or click on a tag to show channels with that tag.

Learn to [create channels](#), explore and transform data.

Learn more about [ThingSpeak Channels](#).

## Examples

- [Arduino](#)
- [Arduino MKR1000](#)
- [ESP8266](#)
- [Raspberry Pi](#)
- [Netduino Plus](#)

## Upgrade

Need to send more data faster?

Need to use ThingSpeak for a commercial project?

## New Channel

Name

Description

Field 1

Field Label 1

☒

Field 2

☐

Field 3

☐

Field 4

☐

Field 5

☐

Field 6

☐

Field 7

☐

Field 8

☐

Metadata

## 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

- **Percentage complete:** Calculated based on data entered into the various fields of a channel. Enter the name, description, location, URL, video, and tags to complete your channel.
- **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

# Create fields

## New Channel

Name	<input type="text" value="Arduino-IoT-Matlab"/>	
Description	<input type="text"/>	
Field 1	<input type="text" value="MQ2"/>	<input checked="" type="checkbox"/>
Field 2	<input type="text" value="TEMP"/>	<input checked="" type="checkbox"/>
Field 3	<input type="text" value="DISTANCE"/>	<input checked="" type="checkbox"/>
Field 4	<input type="text"/>	<input type="checkbox"/>
Field 5	<input type="text"/>	<input type="checkbox"/>
Field 6	<input type="text"/>	<input type="checkbox"/>
Field 7	<input type="text"/>	<input type="checkbox"/>
Field 8	<input type="text"/>	<input type="checkbox"/>
Metadata	<input type="text"/>	
-	<input type="text"/>	

## 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

- **Percentage complete:** Calculated based on data entered into the various fields of a channel. Enter the name, description, location, URL, video, and tags to complete your channel.
- **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

# Channel Created

## Arduino-IoT-Matlab

Channel ID: **1161425**

Author: [mailtojeeva](#)

Access: Private

Private View

Public View

Channel Settings

Sharing

API Keys

Data Import / Export

+ Add Visualizations

+ Add Widgets

Export recent data

MATLAB Analysis

MATLAB Visualization

Channel 2 of 2 < >

## Channel Stats

Created: less than a minute ago

Entries: 0

Field 1 Chart



Arduino-IoT-Matlab

MO2

Field 2 Chart



Arduino-IoT-Matlab

TEMP

This website uses cookies to improve your user experience, personalize content and ads, and analyze website usage. By continuing to use this website, you consent to our use of cookies. Learn more about our cookies policy.

# Create a template

## Templates:

- ☐ Custom (no starter code)
- ☐ Get data from a private channel
- ☐ Get data from a public channel
- ☐ Get data from a webpage

## Examples: Sample code to analyze and transform data

- ☐ Calculate and display average humidity
- ☐ Calculate wind chill and update channel
- ☐ Remove outliers from wind speed data
- ☐ Convert temperature units
- ☐ Calculate high and low temperatures
- ☒ Read Channel to Trigger Email
- ☐ Replace missing values in data
- ☐ Analyze text for the most common color
- ☐ Read live web data for vessels at the port of Boston
- ☐ Scrape web temperature data

Create

## Templates

MATLAB Analysis templates provide sample MATLAB code for analyzing data and writing it to a ThingSpeak channel. If you are new to MATLAB, you can learn interactively at [MATLAB Academy](#).

## Examples

To see MATLAB Analysis in action, select the example and click **Create**.

These examples read data from public ThingSpeak channels:

- [Calculate and display average humidity](#) over the last 60 minutes, and write the results to a new channel.
- [Calculate wind chill and update channel](#) by writing to a new channel at regular intervals.
- [Remove outliers from wind speed data](#) over the past six hours, and write data to a new channel.
- [Convert temperature units](#) from Fahrenheit to Celsius, and write the results to a new channel.
- [Calculate high and low temperatures](#) over the past 24 hours, and write the data with timestamps to a new channel.
- [Trigger Email](#) by analyzing daily soil moisture values.
- [Replace missing values in data](#) of a weather channel, and clean the data using a list-wise deletion algorithm. Then display the missing values, or write data to a new channel.
- [Analyze text for the most common color](#) requested on the public Cheerlights channel, and write the data to a new channel.

These examples scrape data from websites:

- [Read live web data for vessels at the port of Boston](#) from the MarineTraffic website, count the number of vessels, and write the data to a new channel.



# Read and Write API Keys

Author: [mailtojeeva](#)

Access: Private

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

## Write API Key

Key 4GPW0Q6KLJ5M0UQB

Generate New Write API Key

## Read API Keys

Key JW5HGV6UDD9DF0UX

Note

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.

## API Requests

### Write a Channel Feed

```
GET https://api.thingspeak.com/update?api_key=4GPW0Q6KLJ5M0UQB&field=
```

### Read a Channel Feed

```
GET https://api.thingspeak.com/channels/1161425/feeds.json?api_key=
```

## Arduino-IoT-Matlab

Channel ID: **1161425**

Author: [mailtojeeva](#)

Access: Private

**DEMO**

# Questions and Answers



?



**Thank you**