


# Today! Internet of Things (IoT) Hands-on Activities

- Introduction
  - Download Folders
    - Github Repository (Recommended)
    - [https://github.com/muratkuzlu/ODU\\_BLAST2022](https://github.com/muratkuzlu/ODU_BLAST2022)
  - WiFi Information (SSID and Password)  

  - Presentation (Summary)
  - Hands-on Activities
- 
- Q&A



# Building Leaders for Advancing Science and Technology (BLAST) (Summary)

## **Internet of Things (IoT) Hands-on Activities with Arduino**

**Dr. Murat Kuzlu**

**Department of Engineering Technology**

# Outline

- Internet
- Internet of Things (IoT)
- Arduino
- ThingSpeak
- Hands-on Activities



<https://medium.com/@otavioguastamacchia/creating-a-simple-iot-case-8102f22908a7>

# What is the Internet?



<https://www.connexusuk.com/high-speed-internet/>

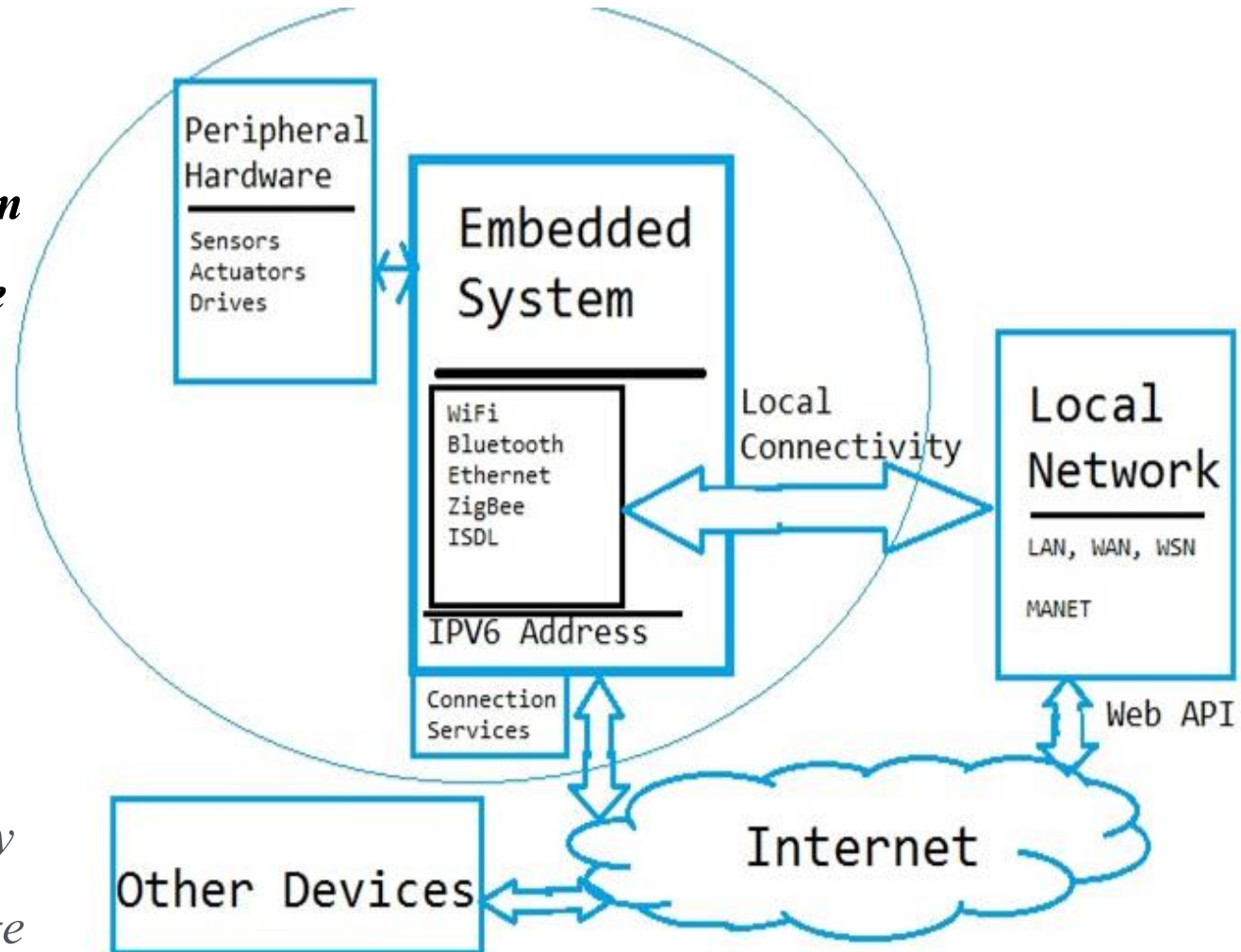


- The Internet is a **global web of computers connected to each by communication lines (wired or wireless).**
- If you look at a map of big cities, smaller towns, and scattered houses, each is **connected together with roads, railways, etc.** This is similar to the Internet, except with the Internet, wires connect computers. The Internet is a superhighway.

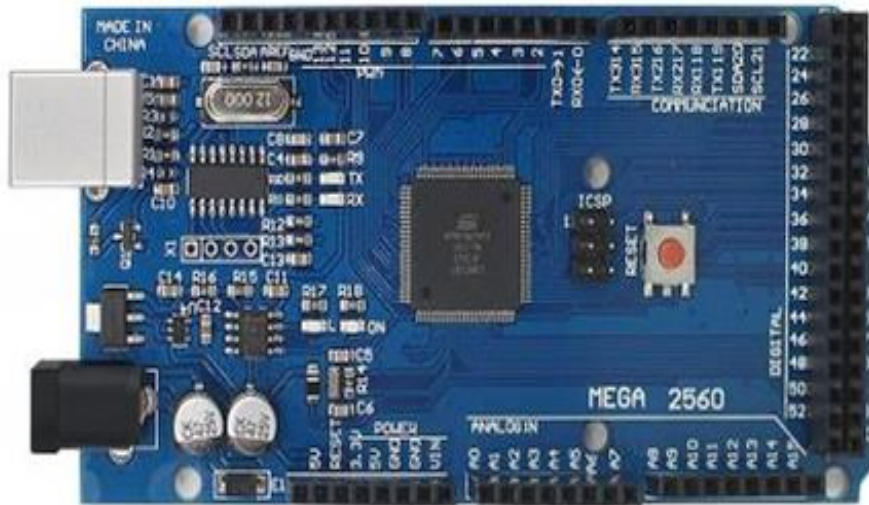
# The Internet of Things (IoT)

***IoT: You can access anything (devices) from anywhere (through the INTERNET)!***

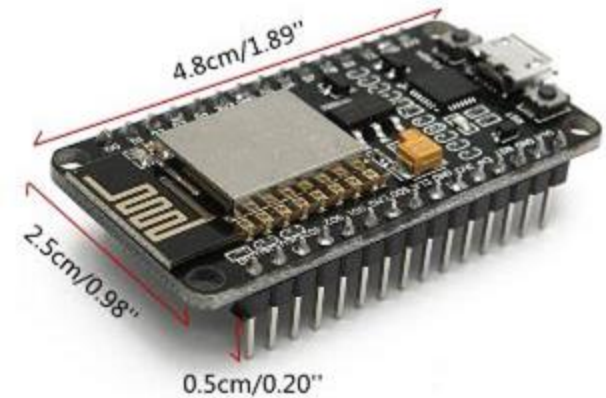
- *Network of physical objects*
- *Embedded System*
- *Network connectivity*
- *Collect and exchange data*



# Arduino



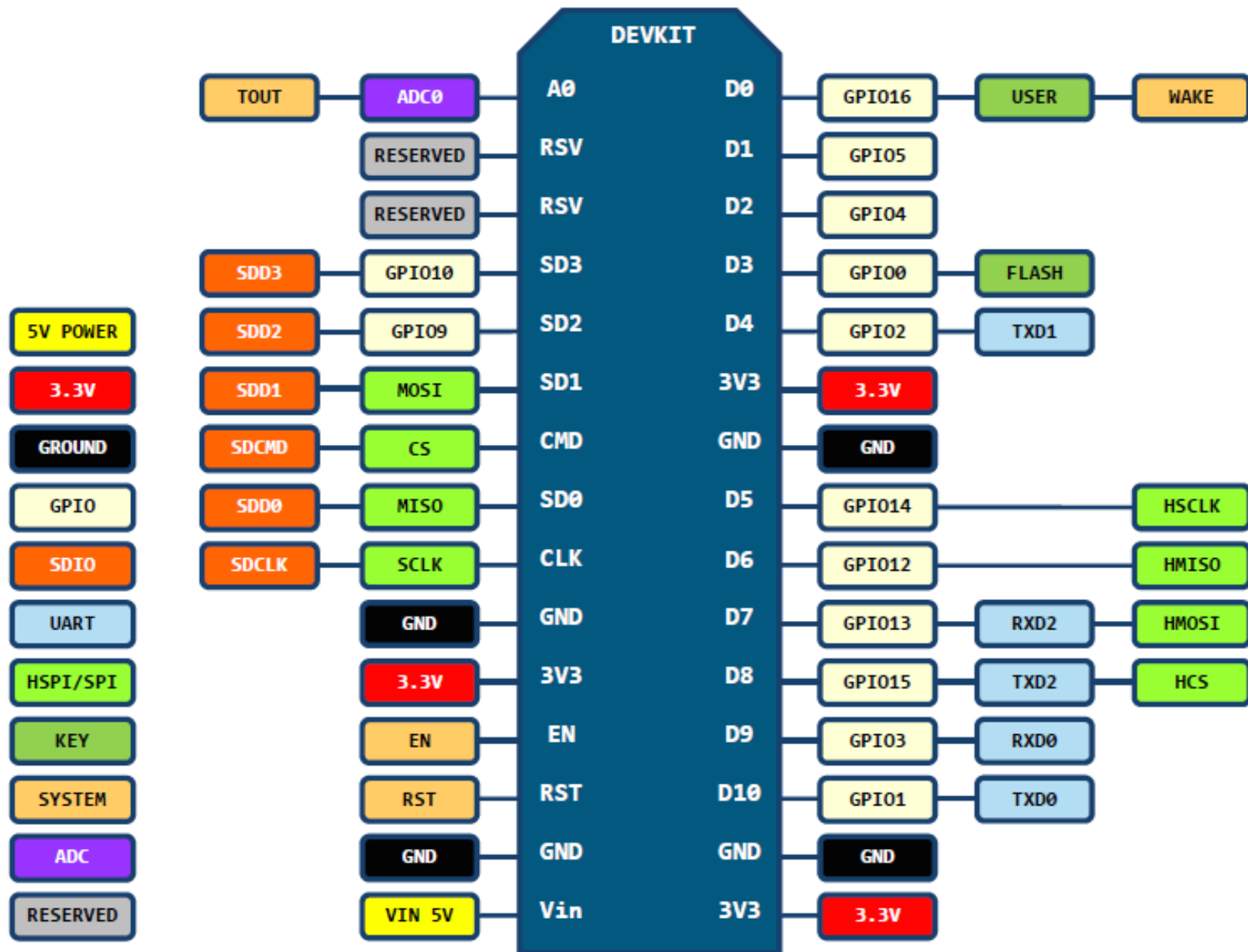
Mega2560 R3 ATmega2560-16AU  
CH340 Development Board



ESP8266 **NodeMCU** LUA  
CP2102 ESP-12E



# NodeMCU ESP8266 ESP-12E Pinout



# What is ThingSpeak?

- **What is ThingSpeak:**  
ThingSpeak is an IoT analytics platform service that lets you collect and store sensor data in the cloud and develop Internet of Things applications.
- <https://thingspeak.com>
- **ThingSpeak Channel:**  
**‘ThingSpeak Channel’ is the core element of the ThingSpeak platform.** This channel is used to store the real-time data, or the data transferred through various sensors and embedded systems. Data stored at the channel is further used for analysis and visualization.
- Software Requirement: Internet
- Hardware Requirement: Arduino.





# Creating A Channel

- <https://thingspeak.com>
- Before creating a channel, you need to sign into things speak. You can easily sign in either using your either ThingSpeak account or mathwork account, or create a new mathwork account via following link:
- [https://thingspeak.com/users/sign\\_up](https://thingspeak.com/users/sign_up)
- Login Page
- Email: [mkuzlu@hotmail.com](mailto:mkuzlu@hotmail.com)
- Password: ODU\_Blast2000

## Create MathWorks Account

Email Address

Missing required information

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

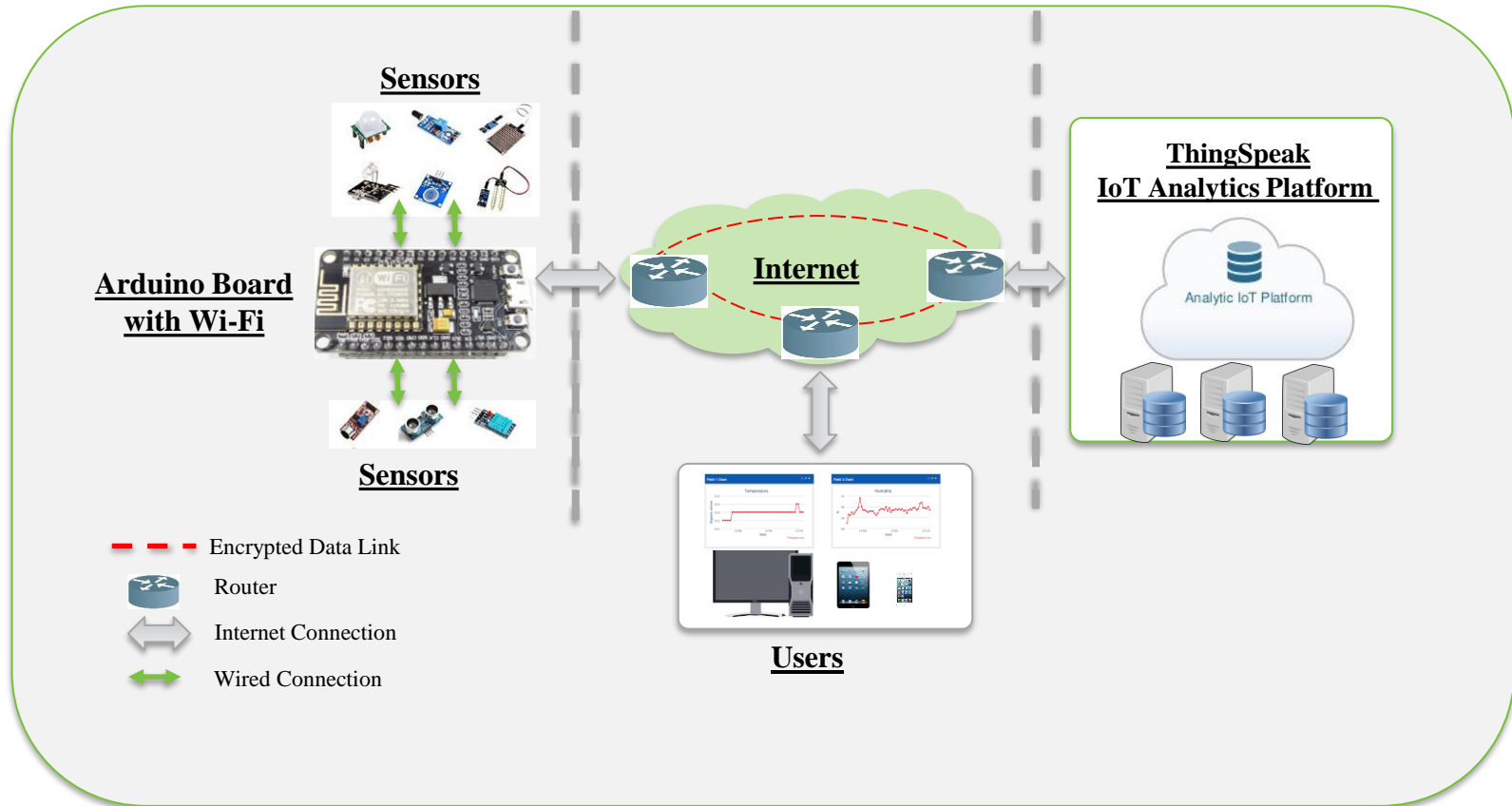
Location

United States

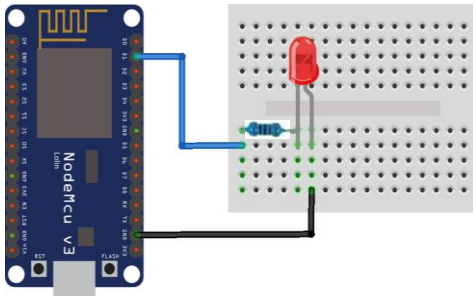
First Name

Last Name

# Architecture of an IoT-based System



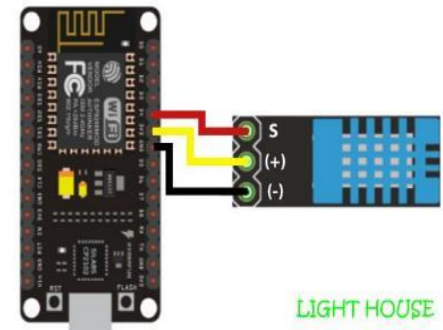
# Hands-on Activities



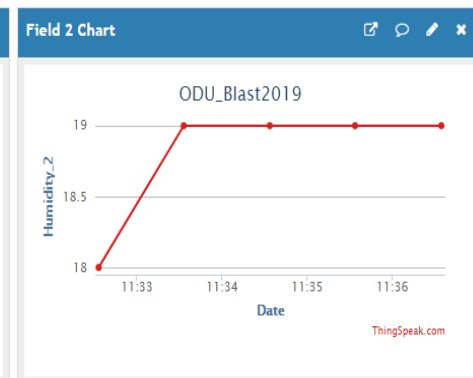
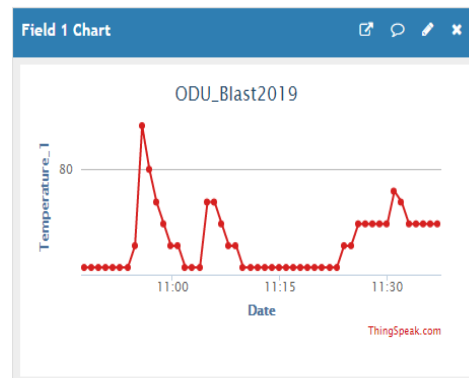
```
#define LED D1 // Led in NodeMCU at pin GPIO5 (D1).

void setup()
{
  pinMode(LED, OUTPUT); // set the digital pin as output.
}

void loop()
{
  digitalWrite(LED, HIGH); // turn the LED off.
  delay(1000); // wait for 1 second.
  digitalWrite(LED, LOW); // turn the LED on.
  delay(1000); // wait for 1 second.
}
```



LIGHT HOUSE



---

# Preparation!

---

# Step 1 - Installing IDE

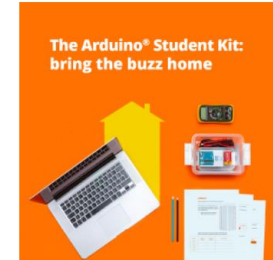
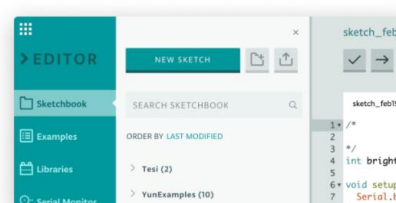
# Installing IDE

- The Arduino Integrated Development Environment (IDE) is the software side of the Arduino platform.
- Go to <https://www.arduino.cc/en/software>

Arduino Web Editor

Start coding online and save your sketches in the cloud. The most up-to-date version of the IDE includes all libraries and also supports new Arduino boards.

**CODE ONLINE** **GETTING STARTED**



## Downloads



### Arduino IDE 1.8.15

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

#### SOURCE CODE

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

#### DOWNLOAD OPTIONS

**Windows** Win 7 and newer

**Windows** ZIP file

**Windows app** Win 8.1 or 10 

**Linux** 32 bits

**Linux** 64 bits

**Linux** ARM 32 bits

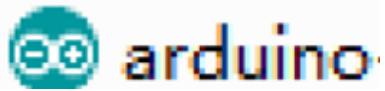
**Linux** ARM 64 bits

**Mac OS X** 10.10 or newer

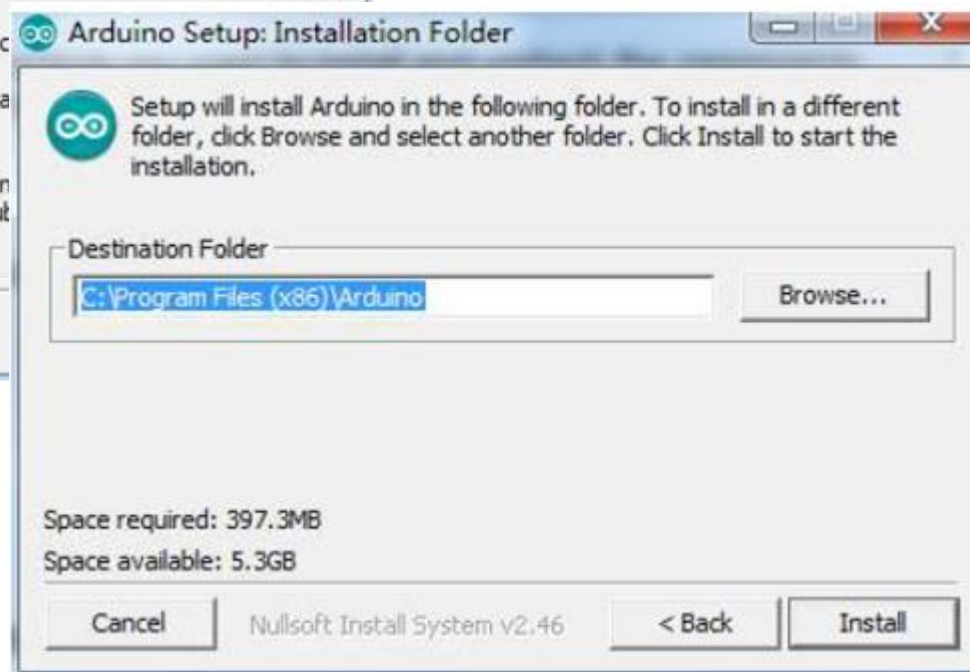
[Release Notes](#) [Checksums \(sha512\)](#)



# Installing Arduino (Windows) — It can be different based on Windows version!



Click Agree and Install...

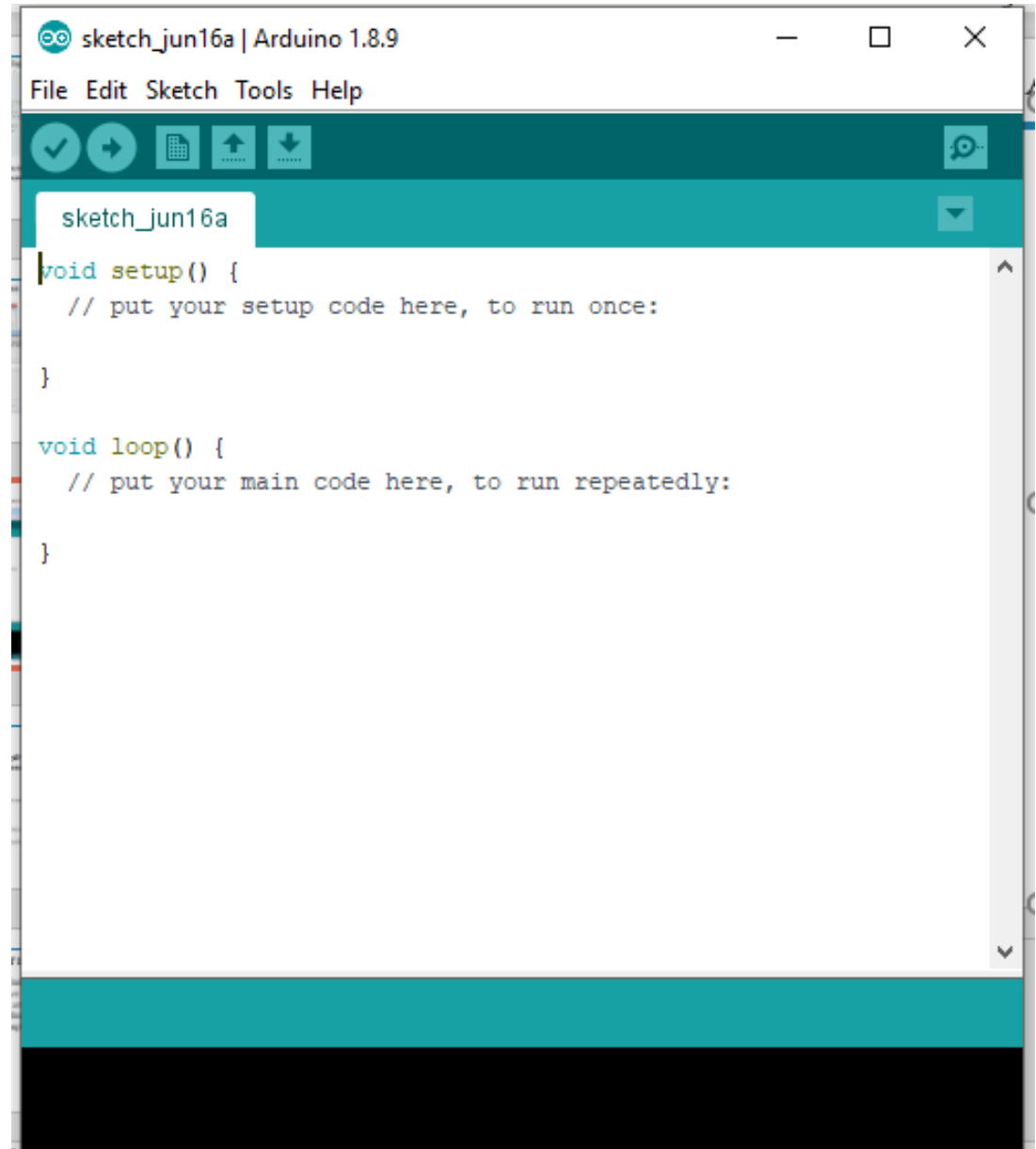


The following Icon appears on the desktop



# Arduino IDE

Double-click to enter the  
desired development  
environment



---

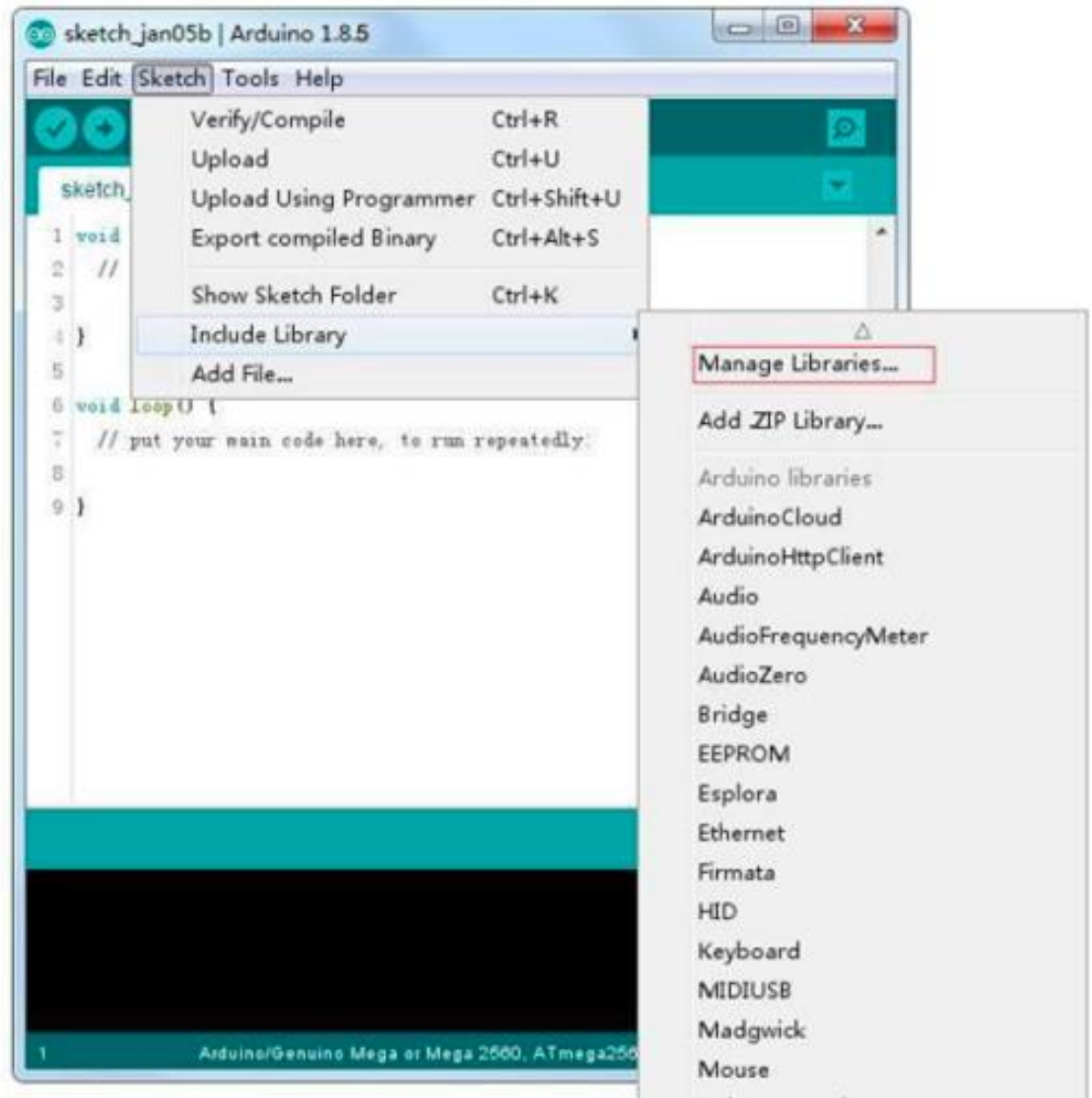
## Step 2 - Add Libraries

# Add Libraries

## How to Install a Library?

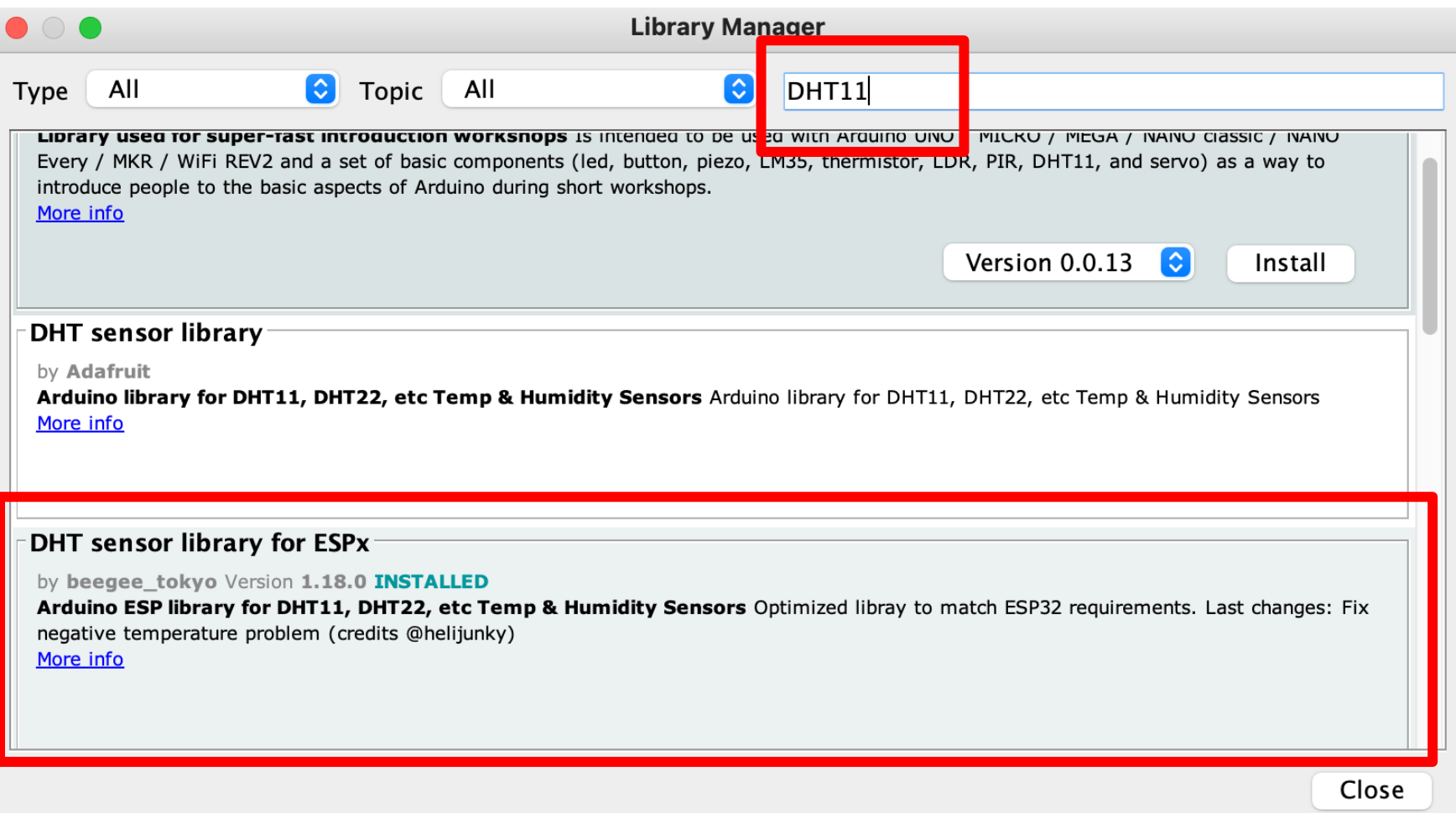
Using the Library Manager  
To install a new library into your Arduino IDE you can use the Library Manager (available from IDE version 1.8.0).

Open the IDE and click to the "Sketch" menu and then Include Library > Manage Libraries.



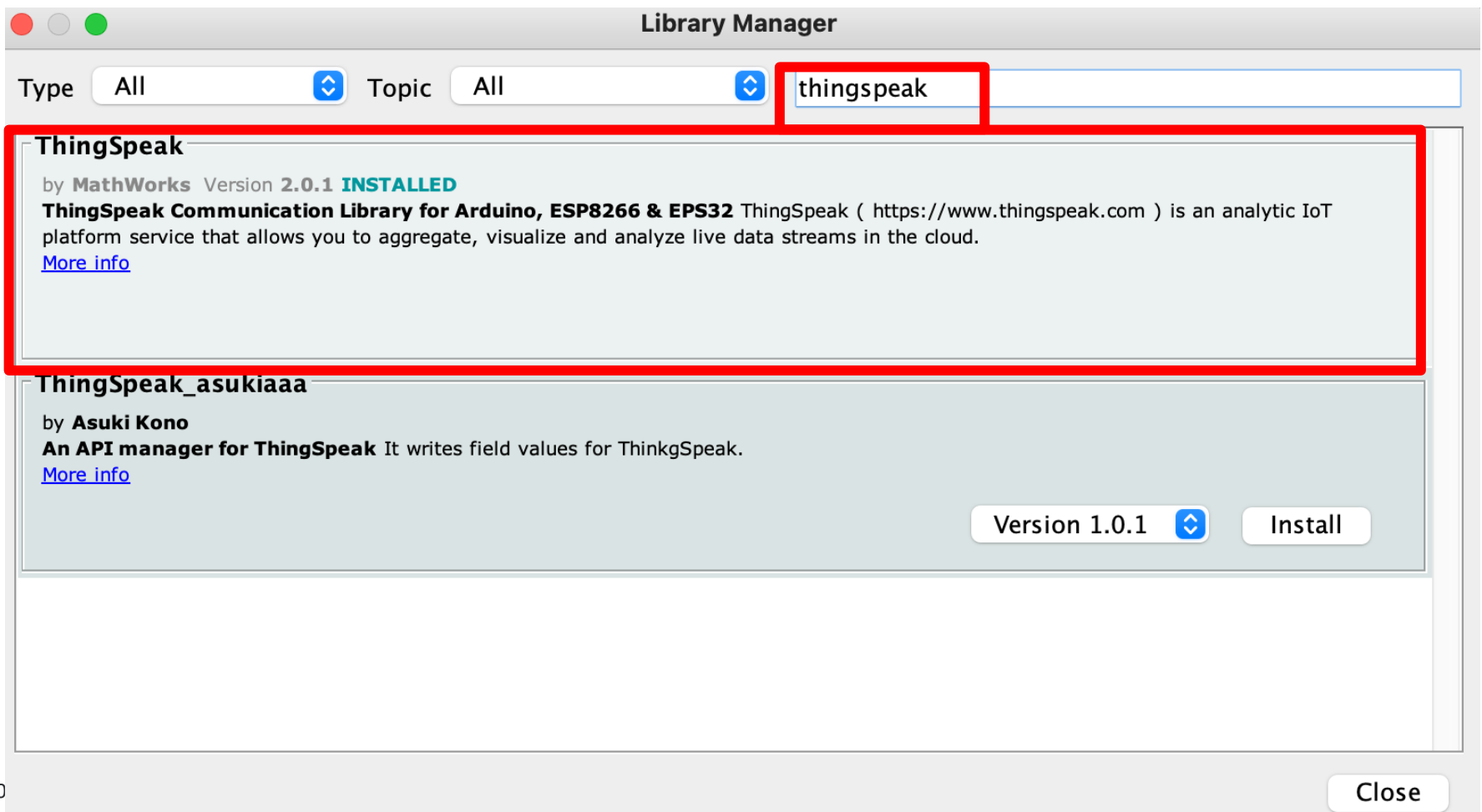
# Add Libraries- DHT11

Open the IDE and click to the "Sketch" menu and then Include Library > Manage Libraries > Type "DHT11" > Install "DHT sensor library" and "DHT sensor library for ESPx"



# Add Libraries- Arduino

- The Arduino IDE needs to have the ThingSpeak library installed in order for your devices to know how to send data to ThingSpeak. In the Arduino IDE, choose Sketch, Include Library, and Manage Libraries. Search for “thingspeak” and click Install.





---

## Step 3 - NodeMCU Setup

# NodeMCU Setup

- *Additional Board Manager URL:*
  - *`http://arduino.esp8266.com/stable/package_esp8266com_index.json`*

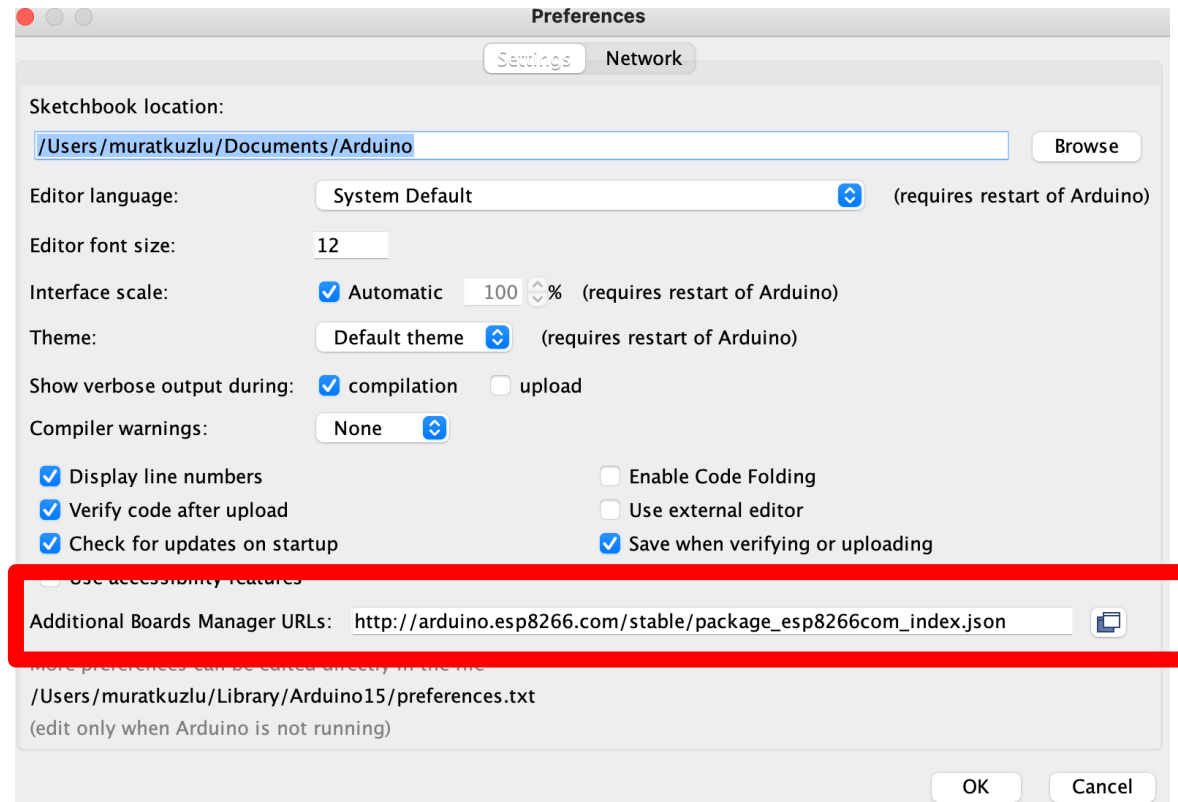
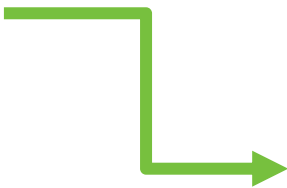
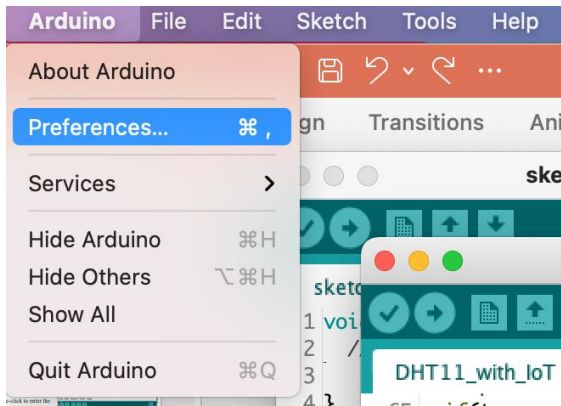
## Installing with Boards Manager

Starting with 1.6.4, Arduino allows installation of third-party platform packages using Boards Manager. We have packages available for Windows, Mac OS, and Linux (32 and 64 bit).

- Install Arduino 1.6.8 from the [Arduino website](http://arduino.cc/en/Main/Software).
- Start Arduino and open Preferences window.
- Enter `http://arduino.esp8266.com/stable/package_esp8266com_index.json` into *Additional Board Manager URLs* field. You can add multiple URLs, separating them with commas.
- Open Boards Manager from Tools > Board menu and install *esp8266* platform (and don't forget to select your ESP8266 board from Tools > Board menu after installation).

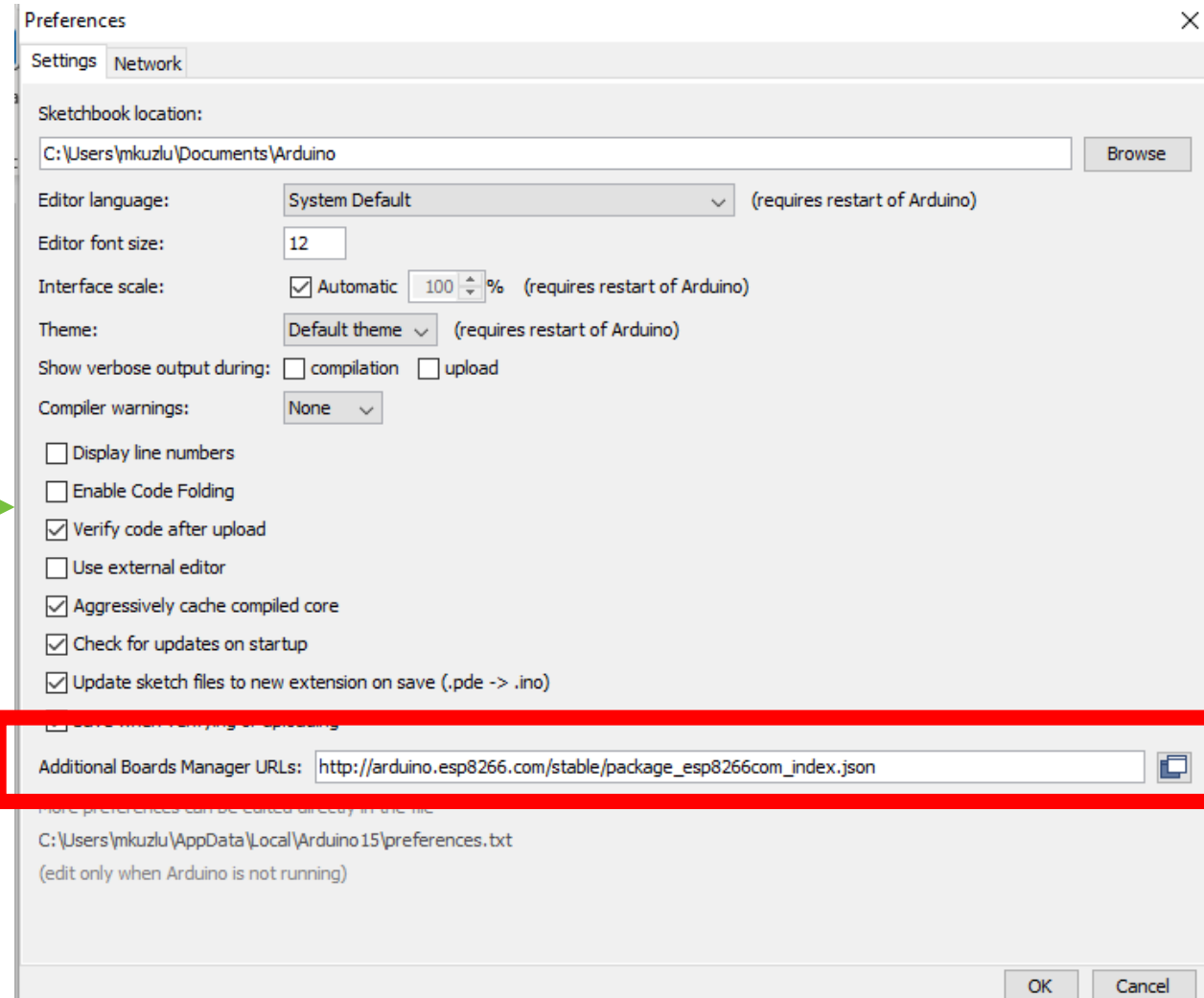
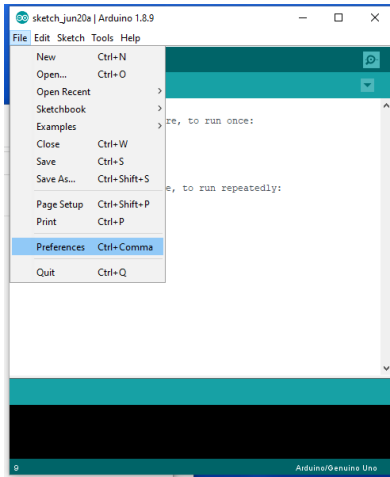
# NodeMCU Setup - macOS

- Add the link to the *Additional Board Manager URLs*



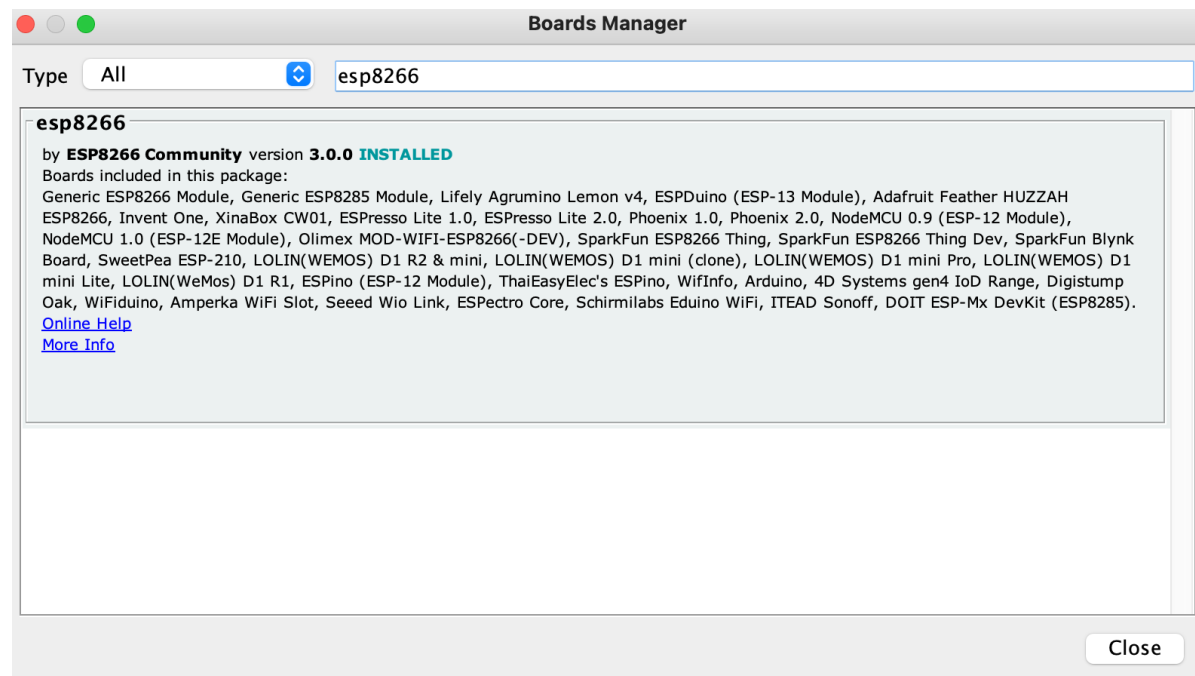
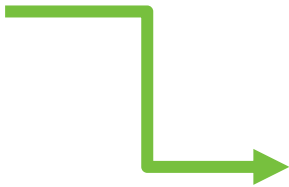
# NodeMCU Setup - Windows

- Add the link to the *Additional Board Manager URLs*



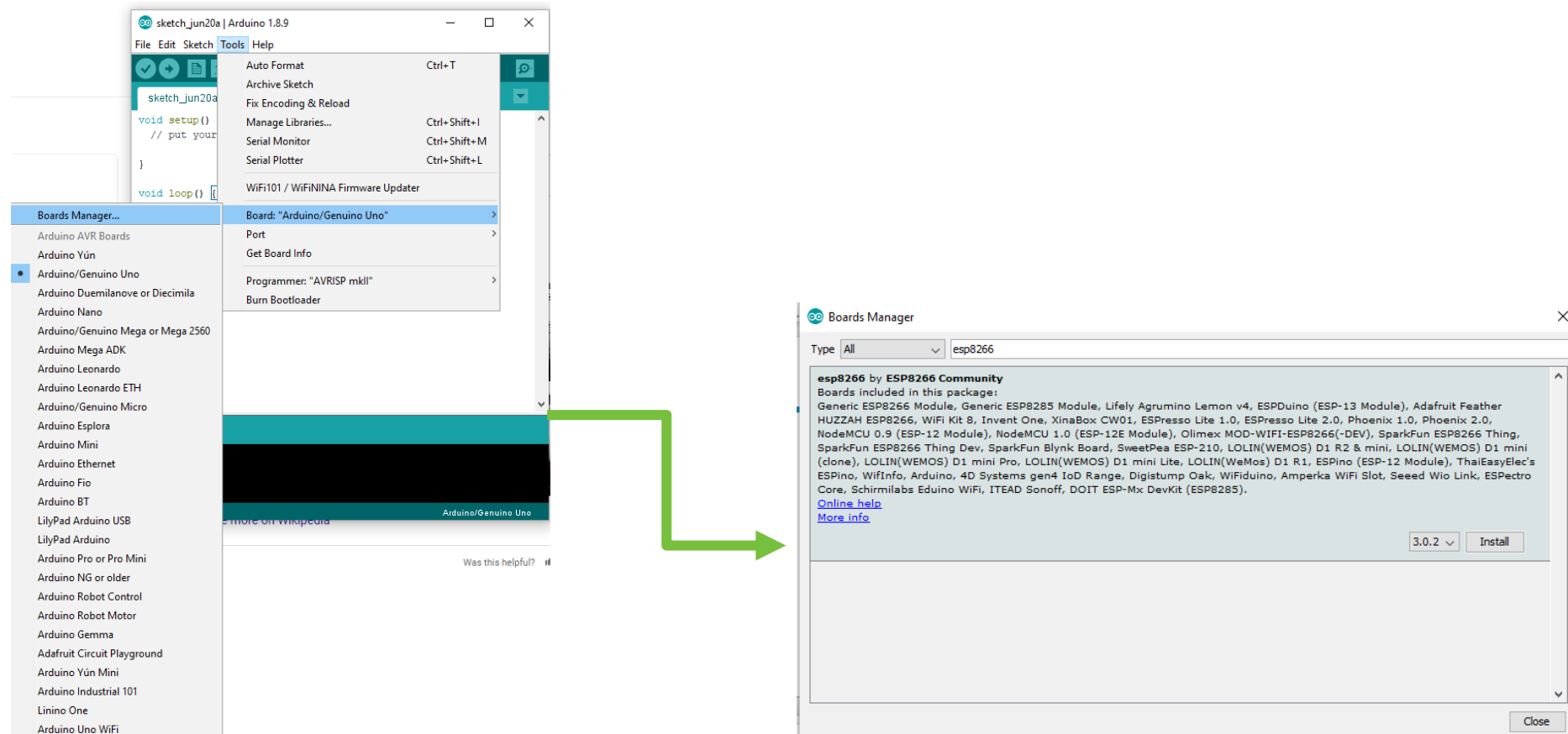
# NodeMCU Setup - macOS

- Search the board manager for *esp8266*
- Install the ESP8266 library



# NodeMCU Setup - Windows

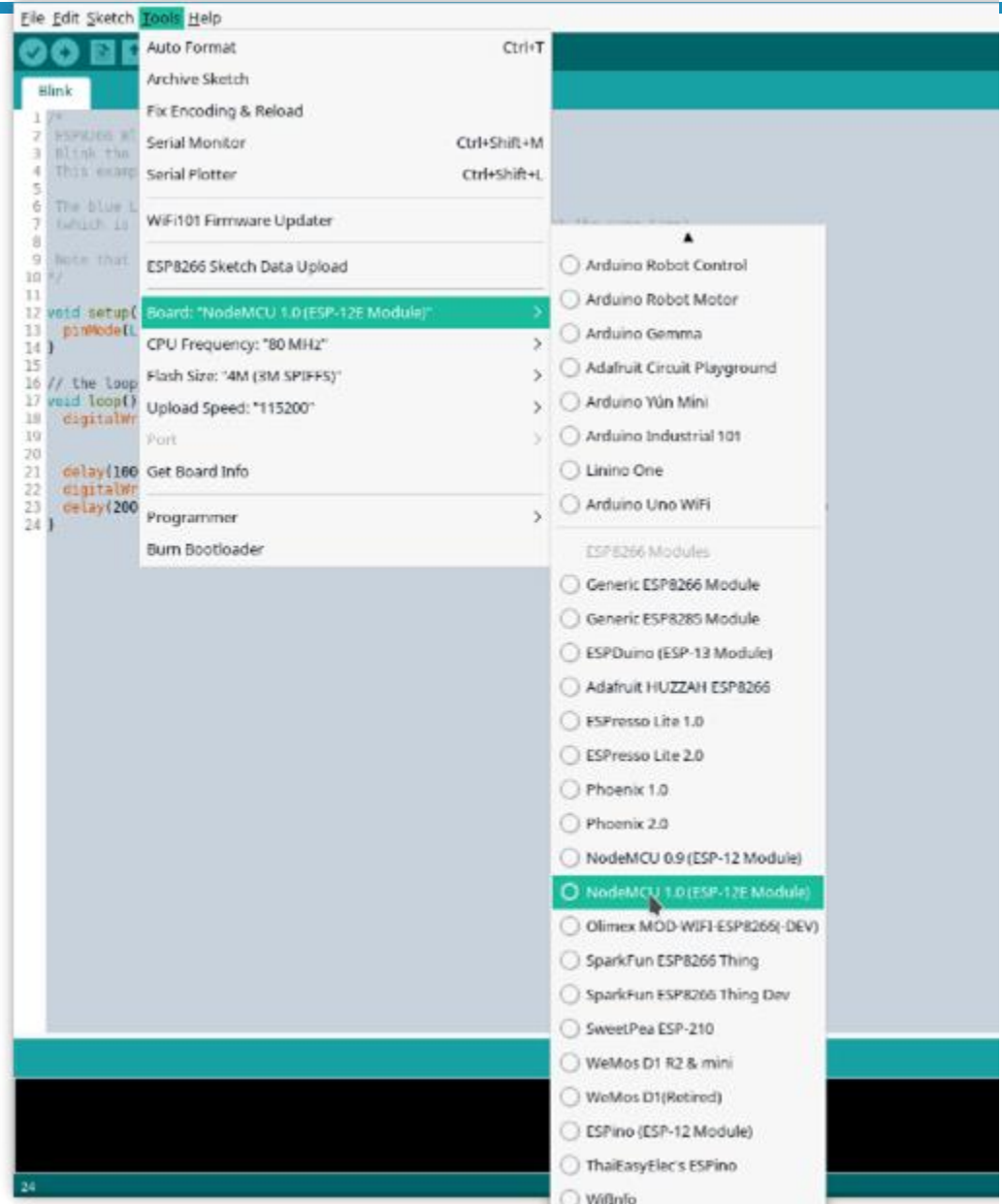
- Search the board manager for *esp8266*
- Install the ESP8266 library





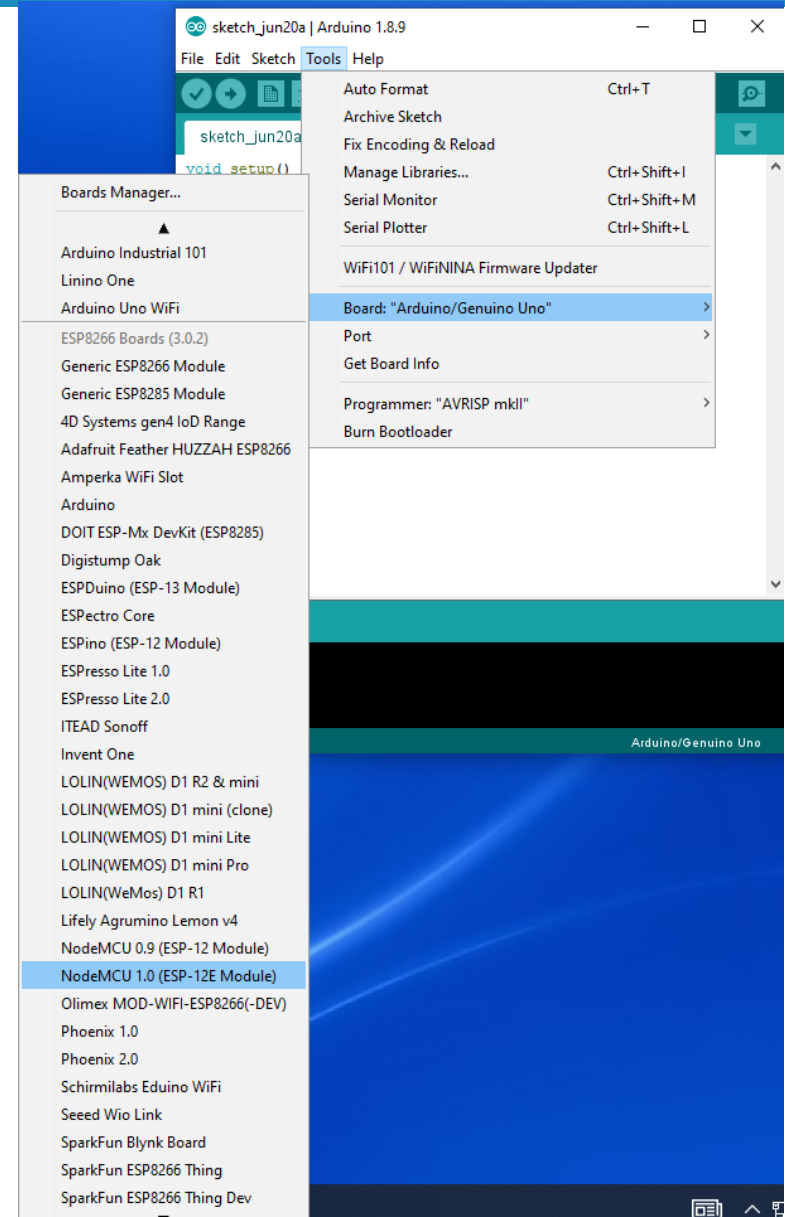
# NodeMCU Setup - macOS

- Select the NodeMCU 2.0 as your board
- Additional settings appear under the board menu
- They can be left as they are
- Higher Upload Speed reduces your upload times



# NodeMCU Setup - Windows

- Select the NodeMCU 2.0 as your board
- Additional settings appear under the board menu
- They can be left as they are
- Higher Upload Speed reduces your upload times



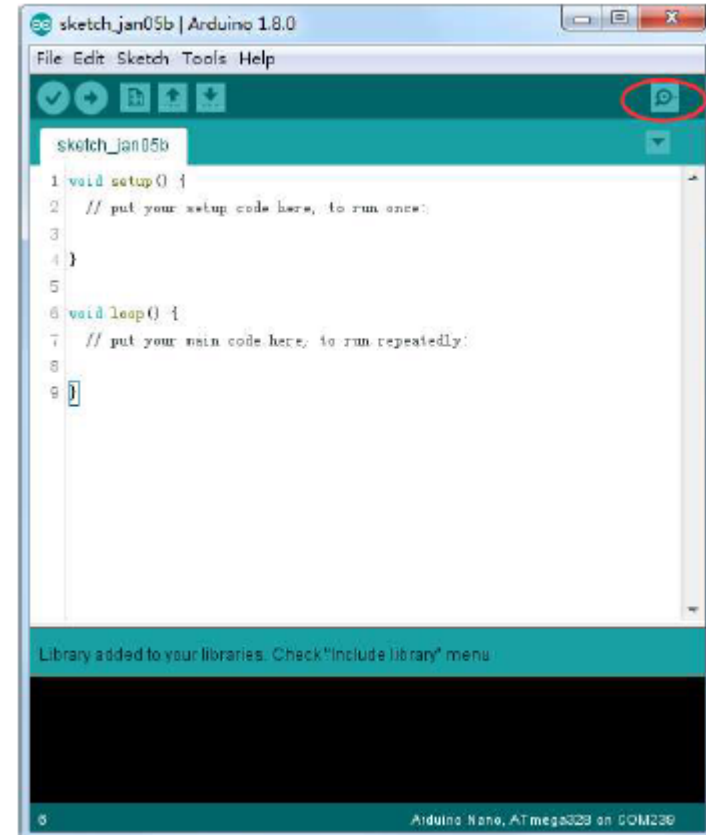
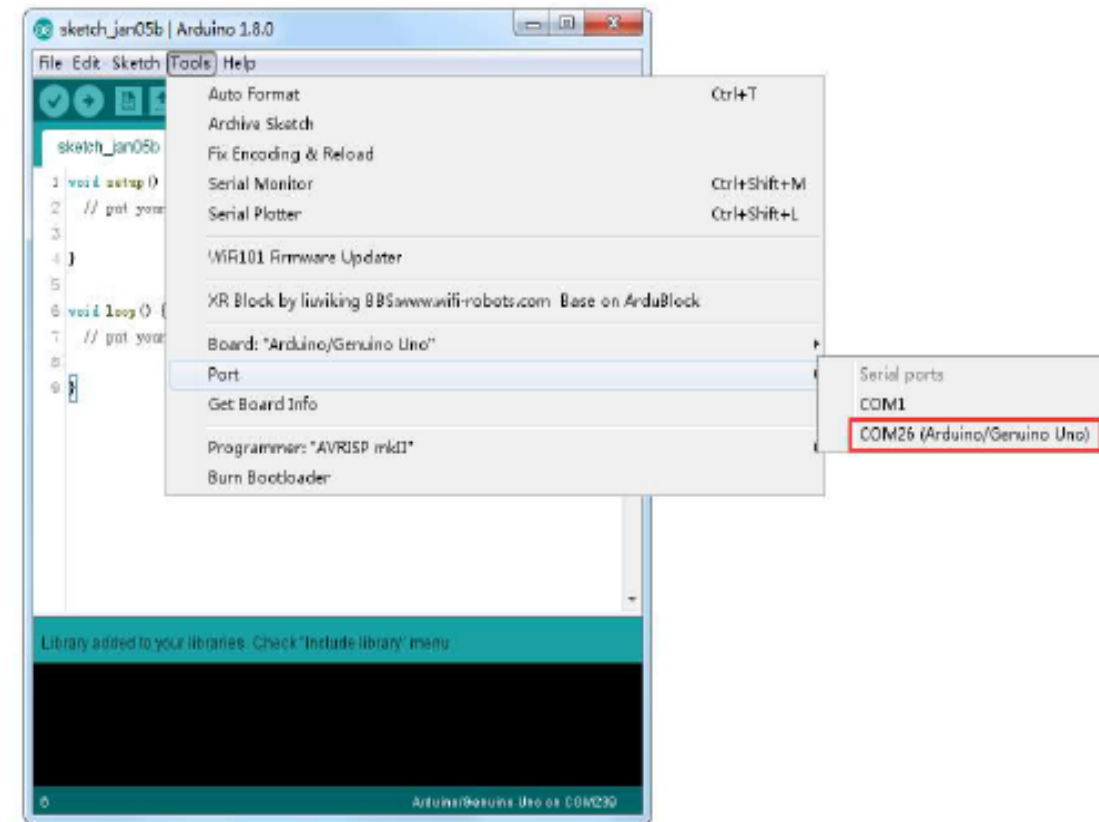
---

# Step 4 - Open Serial Monitor

# Open Serial Monitor

## Arduino Serial Monitor (Windows, Mac, Linux) and Make a Serial Connection

The Arduino Integrated Development Environment (IDE) is the software side of the Arduino platform. And, because using a terminal is such a big part of working with Arduinos and other microcontrollers, they decided to include a serial terminal with the software. Within the Arduino environment, this is called the Serial Monitor.



# Check Serial Monitor and Baud Rate

sketch\_jun22e | Arduino 1.8.13

✓ ↻ ⌂ ⬆ ⬇

sketch\_jun22e

```
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```

Serial Monitor

Rate

/dev/cu.usbserial-0001

Send

16:44:45.826 -> Temperature Value is :77.90F  
16:44:45.826 -> Humidity Value is :48.00%  
16:44:50.817 -> Temperature Value is :78.08F  
16:44:50.855 -> Humidity Value is :48.00%  
16:44:55.857 -> ??????(??A? L ??????!!??=??FB ? A???? ?!(??C? ?BTemperature Value is :78.08F  
16:45:05.889 -> Humidity Value is :48.00%  
16:45:10.914 -> Temperature Value is :78.08F  
16:45:10.914 -> Humidity Value is :48.00%  
16:45:15.943 -> Temperature Value is :78.08F  
16:45:15.943 -> Humidity Value is :48.00%

☒ Autoscroll ☒ Show timestamp

Newline

115200 baud

Clear output

Baud Rate is NOT CORRECT!

---

# Step 5 - Connect ThingSpeak



# Connect ThingSpeak

<https://thingspeak.com/>

- Login Page
- Email: [mkuzlu@hotmail.com](mailto:mkuzlu@hotmail.com)
- Password: ODU\_Blast2000

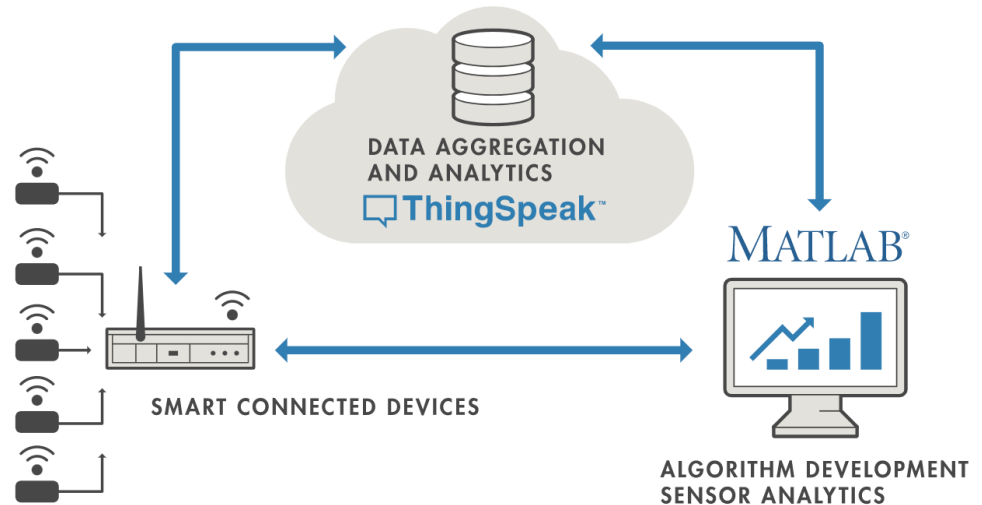


Email

No account? [Create one!](#)

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

Next





# Creating A Channel

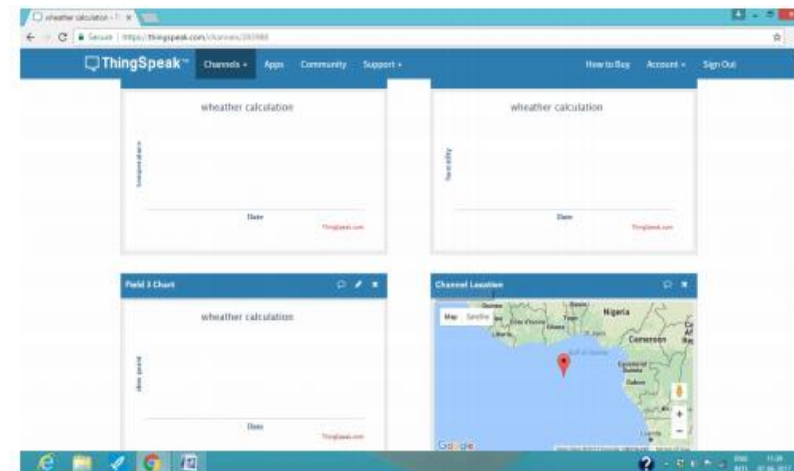
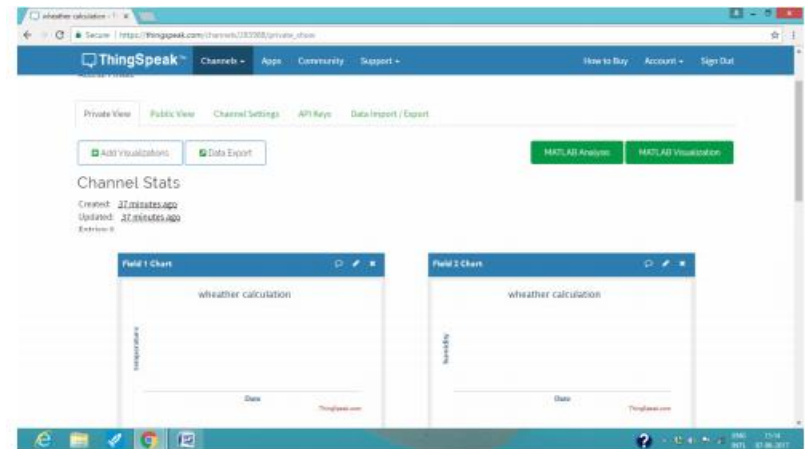
## See My Channels

**Private View**-- It displays the information about your channel that is only visible to you (Recommended)

**Public View**- if you have chosen to make your channel publicly visible then it will display the selected fields and information

## My Channels

New Channel		Search by tag	Q
Name	Created	Updated	
 ODU_Blast2000	2019-06-17	2021-06-03 22:36	
<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>			
 Channel_Test	2021-06-22	2021-06-22 20:14	
<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>			



---

# Step 6 - Hands on Activities and Source Code

# Hands on Activities and Source Code

---

**Copy&Paste or Download** examples, from GitHub:

- Full: [https://github.com/muratkuzlu/ODU\\_BLAST2022](https://github.com/muratkuzlu/ODU_BLAST2022)
- DHT11 (Recommended):  
[https://github.com/muratkuzlu/ODU\\_BLAST2022/tree/main/Projects/4\\_DHT11](https://github.com/muratkuzlu/ODU_BLAST2022/tree/main/Projects/4_DHT11)

**Focus on end-to-end results, not details**

---

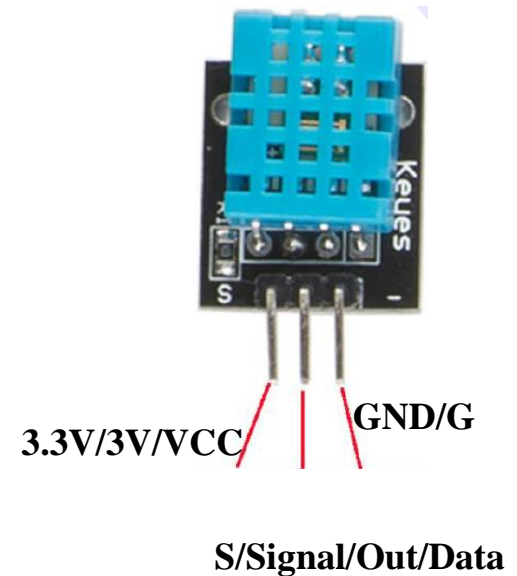
Start!  
Hands-on!

# Temperature & Humidity Sensor DHT11

The DHT11 sensor can detect temperature (C and F) & humidity.

The DHT11 detects water vapor by measuring the electrical resistance between two electrodes. The humidity sensing component is a moisture holding substrate with electrodes applied to the surface.

It has everything it requires built into it, so it will work very well with the NodeMCU. This sensor is used in conjunction with the DHT11 Library.



# Wiring a switch with Arduino

---

We will learn how to set up the DHT11 Humidity and Temperature sensor on your NodeMCU. And learn about how the Humidity sensor works, and how to check output readings from the Serial monitor.

## Hardware

- NodeMCU
- DHT11 Humidity and Temperature sensor
- Breadboard
- Jumper Wires (Optional)
- Micro USB Cable

## Software

- [Arduino IDE\(version 1.6.4+\)](#)
- [https://github.com/muratkuzlu/ODU\\_BLAST2021/tree/main/Projects/4\\_DHT11](https://github.com/muratkuzlu/ODU_BLAST2021/tree/main/Projects/4_DHT11)

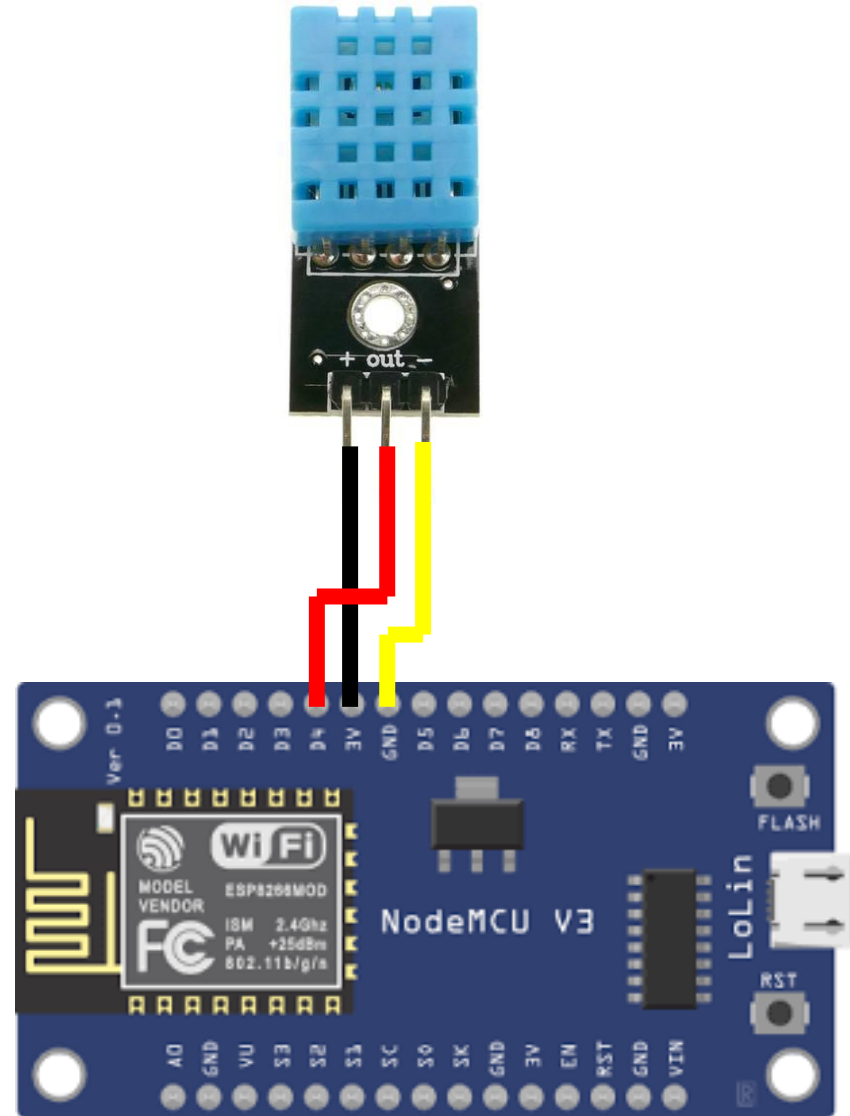
# Digital input with Arduino

## Set up

Wiring the **DHT11** to the NodeMCU is really easy, but the connections are different depending on which type you have either 3-pins or 4-pins

The **wiring connections** are made as follows:

- Pin (**S** or **SIG** or **Out** or **Data**) of the DHT11 goes into Digital Pin **D4** of the NodeMCU.
- Pin (**VCC** or **+** or **VC**) of the DHT11 goes into **+3.3v** or **3V** of the NodeMCU.
- Pin (**GND** or **-**) of the DHT11 goes into Ground Pin (**GND**) of the NodeMCU.





---

# Without IoT

# Digital input with Arduino

## Code

→ Copy the following code to the IDE

Add the DHTesp.h library

Define Variables

Initialize variables. Runs once

Used to actively control the Arduino board. Run repeatedly

```
#include "DHTesp.h"
```

```
int temperature, humidity, k=0, l=0;
```

```
#define DHTPIN D4 //pin where the dht11 is connected  
DHTesp dht;
```

```
void setup()
```

```
{  
  Serial.begin(115200);  
  delay(10);  
  //dht.begin();  
  dht.setup(DHTPIN, DHTesp::DHT11); // data pin 4  
}
```

```
void loop()
```

```
{  
  static boolean data_state = false;  
  float humidity = dht.getHumidity();  
  float temperature = dht.getTemperature();  
  temperature = CelsiusToFahrenheit(temperature);  
  Serial.print("Temperature Value is :");  
  Serial.print(temperature);  
  Serial.println("F");  
  Serial.print("Humidity Value is :");  
  Serial.print(humidity);  
  Serial.println("%");  
  delay(5000);  
}
```

# Functions

## //Functions

```
float FahrenheitToCelsius(float fahrenheit)
{
    float celsius;
    celsius = (fahrenheit - 32.0) * 5.0 / 9.0;
    return celsius;
}
```

```
float CelsiusToFahrenheit(float celsius)
{
    float fahrenheit;
    fahrenheit = (celsius * 9.0) / 5.0 + 32;
    return fahrenheit;
}
```

NOTE:

When you check the serial monitor make sure the baud rate and the serial begin number in your code is the same.

→ Upload

---

With IoT

# Digital input with Arduino and IoT - I

Now we are going to connect to IoT

→ Copy the following code to a new IDE sketch

```
#include "DHTesp.h"  
#include "ThingSpeak.h"  
#include <ESP8266WiFi.h>
```

```
const char *ssid = "ODU_Blast1"; // replace with your wifi ssid and wpa2 key  
const char *pass = " ODU_Blast1 ";  
const char * server = "api.thingspeak.com";  
const char * myWriteAPIKey = " ABY92BDPLDTEJJP"; // Enter your Write API key from ThingSpeak  
unsigned long myChannelNumber = 803487;  
uint8_t temperature, humidity, k=0, l=0;
```

```
#define DHTPIN D4 //pin where the dht11 is connected  
DHTesp dht;
```

```
WiFiClient client;
```

The screenshot shows the ThingSpeak website interface. At the top, there are tabs for 'Private View', 'Public View', 'Channel Settings', 'Sharing', 'API Keys', and 'Data Import / Export'. The 'API Keys' tab is selected, showing a 'Write API Key' section with a key value of '3M0SBN71PI6UD1A4' and a 'Generate New Write API Key' button. Below this, the 'Channel Settings' tab is selected, showing a 'Channel Settings' section with a 'Percentage complete' of 50%. The 'Channel ID' is 803487, the 'Name' is 'ODU\_Blast2019', and the 'Description' is 'Hands-on IOT Activities'. There are three fields: 'Field 1' (Temperature), 'Field 2' (Humidity), and 'Field 3' (LED\_Status), each with a checkbox.

# Digital input with Arduino and IoT - II

```
void setup()
{
  Serial.begin(115200);
  delay(10);
  //dht.begin();
  dht.setup(DHTPIN, DHTesp::DHT11); // data pin 4
  Serial.println("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  // Print the IP address
  Serial.println(WiFi.localIP());
  ThingSpeak.begin(client);
}
```

# Digital input with Arduino and IoT - III

```
void loop()
{
  static boolean data_state = false;
  float humidity = dht.getHumidity();
  float temperature = dht.getTemperature();
  temperature = CelsiusToFahrenheit(temperature);
  Serial.print("Temperature Value is :");
  Serial.print(temperature);
  Serial.println("F");
  Serial.print("Humidity Value is :");
  Serial.print(humidity);
  Serial.println("%");
  // Write to ThingSpeak. There are up to 8 fields in a channel, allowing you to store up to 8 different pieces of information in a channel.
  Here, we write to field 1.
  if(temperature < 255)
  {
    k=temperature;
  }
  if(humidity < 255)
  {
    l=humidity;
  }
  if( data_state )
  {
    ThingSpeak.writeField(myChannelNumber, 1, k, myWriteAPIKey);
    data_state = false;
  }
  else
  {
    ThingSpeak.writeField(myChannelNumber, 2, l, myWriteAPIKey);
    data_state = true;
  }
  47 delay(30000); // ThingSpeak will only accept updates every 15 seconds.
}
```

→ Upload

# Digital output with Arduino and IoT -IV

→ Check ThingSpeak

## ODU\_Blast2000

Channel ID: 803487

Author: mkuzlu123

Access: Public

Hands-on IOT Activities

Private View

Public View

Channel Settings

Sharing

API Keys

Data Import / Export

+ Add Visualizations

+ Add Widgets

Export recent data

MATLAB Analysis

MATLAB Visualization

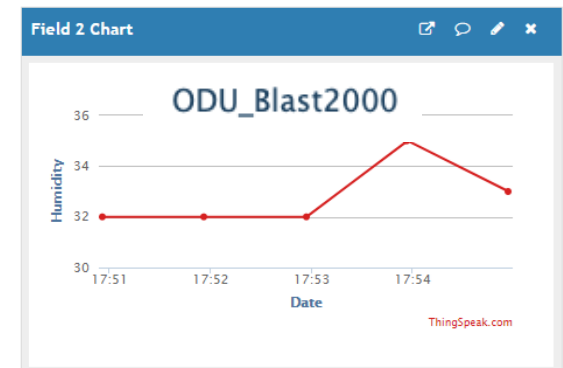
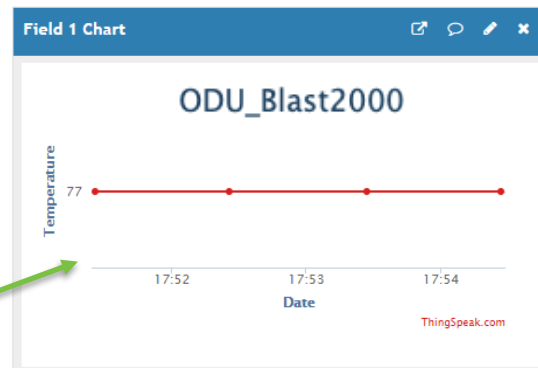
### Channel Stats

Created: about 24 hours ago

Last entry: 5 minutes ago

Entries: 64

**Changes in  
Temperature and  
Humidity**







**Dr. Murat Kuzlu**  
**[mkuzlu@odu.edu](mailto:mkuzlu@odu.edu)**