OLD DOMINION UNIVERSITY

Building Leaders for Advancing Science and Technology (BLAST) 2019

Internet of Things 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** (mostly phone lines).

http://mediatechnologyeducation.pbworks.com/w/page/20693030/The%20Information%20Superhighway

- 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.

Some ways to use the Internet

- Surfing
- E-mail
- Social media
- Shopping
- News
- Games
- **REMOTE MONITORING and CONTROL**

The Internet: https://www.uen.org

Why Internet is Important

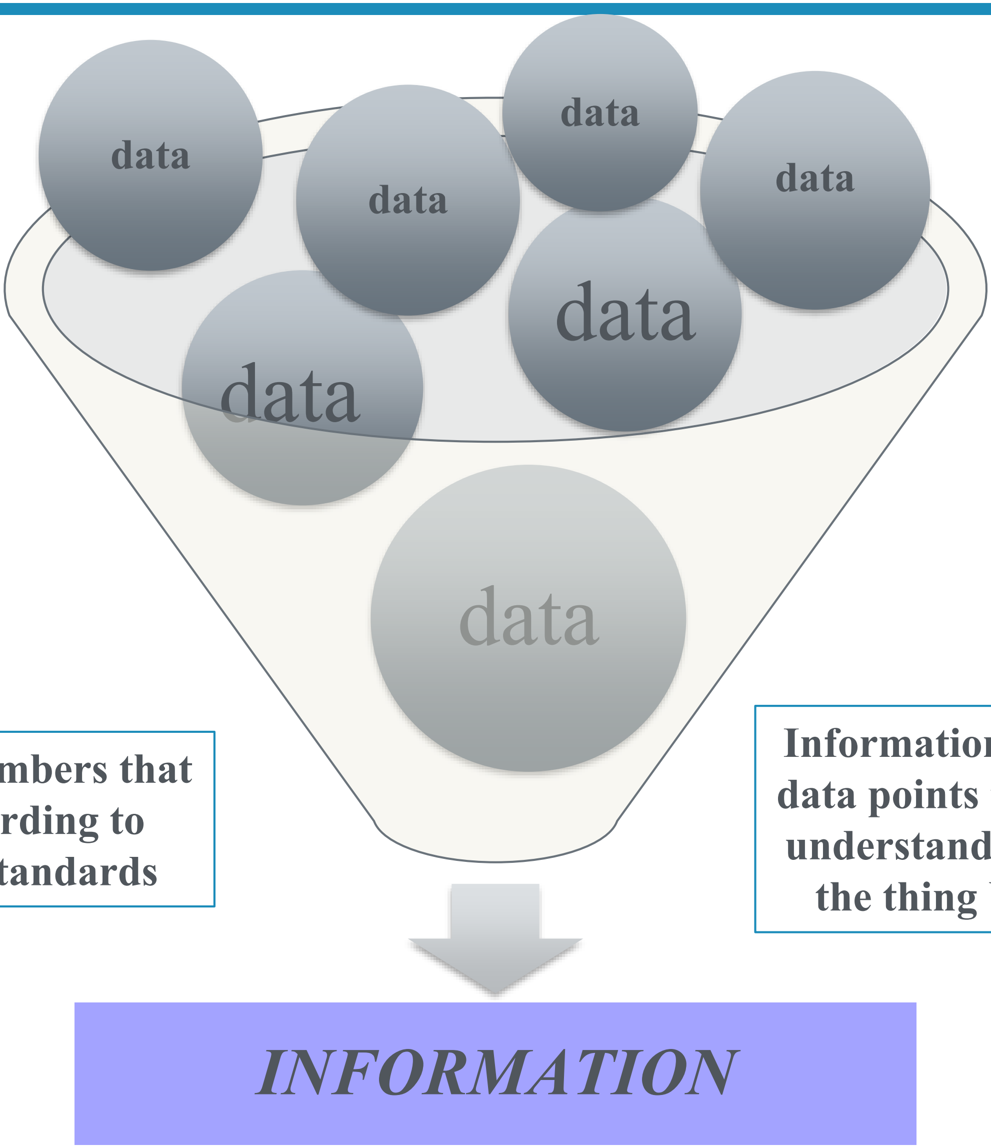
- Data, data, data!
- Modern organizations rely on the efficient transmission of data
- Enables distributed systems, **real-time communication,** electronic commerce, social media, and the Web

https://makeawebsitehub.com/social-media-sites/

https://www.edx.org/course/social-media-how-media-got-social

Ref: Taylor M. Wells: College of Business Administration, California State University, Sacramento

Data vs. Information



INFORMATION

Data is the raw numbers that we capture according to some agreed to standards

Information is a collection of data points that we can use to understand something about the thing being measured

Data are the facts or details from which info rmation is derived

Ref: Taylor M. Wells: College of Business Administration,
California State University, Sacramento

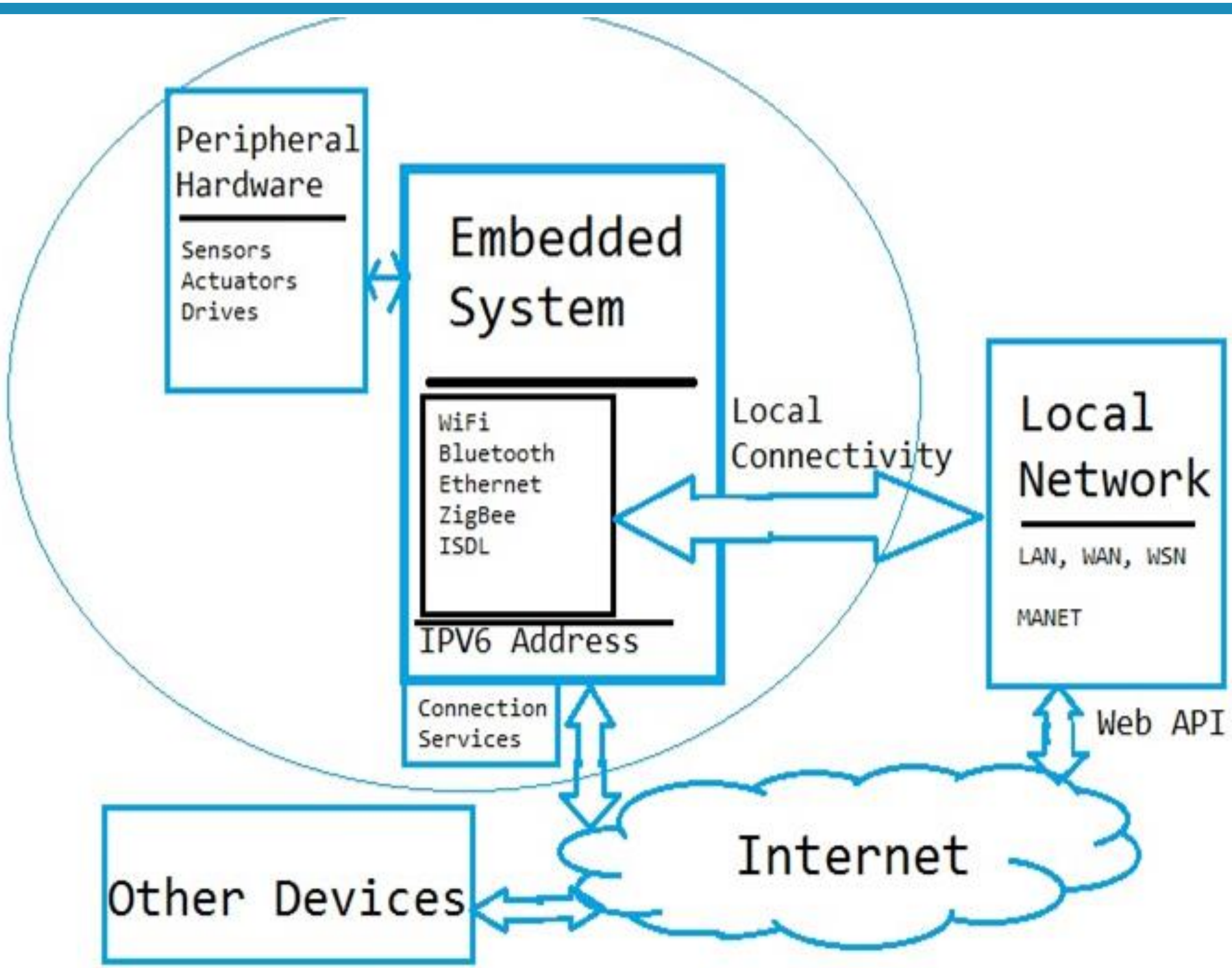
Trends – WoT and IoT

- The Web of Things
  - Everything connects to the network!
    - e.g., cars, refrigerators, thermostats, shoes, doors, etc.
  - Networks need to support the increased demands of these devices
- **The Internet of Thing (IoT)**
  - ***The network of physical objects*****—devices, vehicles, buildings and other items--** ***embedded with electronics, software, sensors, and network connectivity*****—that enables these objects** ***to collect and exchange data.*** **"**

The Internet of Thing (IoT)

IoT

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



IOT Examples

- Examples of objects that can fall into the scope of Internet of Things include connected security systems, sensors, thermostats, cars, electronic appliances, light in the household and commercial environments, alarm clocks, speaker systems, vending machines and more.

https://www.expressvpn.com/blog/what-is-the-internet-of-things-iot/

Internet-connected devices

John Romkey’s Toaster (1990, Ethernet)

Ambient Orb (2002, via pager network)

iPod (2001), iTunes Store (2003, via USB/PC)

Nike+ iPod (2006), Bracelet (2008 via USB/PC)

Rafi Haladjian’s Nabaztag (2006, Wifi)

Rob Faludi’s Botanicalls (2006, Ethernet)

Schulze&Webb Availabot (2006, via USB/PC)

iPhone (2007, GSM)

Amazon Kindle (2007, 3G)

Wafaa Bilal’s Shoot an Iraqi (2007, ?)

Withings BodyScale (2008, Wifi)

Vitality GlowCap (2008, Wifi; 2011, 3G)

BakerTweet (2009, 3G)

Adrian McEwen’s Bubblino (2009, Ethernet)

David Bowen’s Telepresent Water (2011, ?)

Nest Thermostat (2011, Wifi)

BERG’s Little Printer (2011, ?)

Supermechanical’s Twine (2012, Wifi)

Olly & Polly (2012, via USB/PC)

Koubachi Sensor (2012, Wifi)

Descriptive Camera (2012, Ethernet)

…….

The Future of IoT

- As far as the reach of the IoT, there are more than 12 billion devices that can currently connect to the Internet, and it is expected that by 2020 there will be 26 times more connected things than people.

https://www.networkworld.com/article/3198657/the-future-of-iot-where-its-heading-what-to-expect.html

Why Arduino ?

- It is an **open-source project**, software/hardware is extremely **accessible** and very flexible to be customized and extended

- It is **flexible**, offers a variety of digital and analog inputs, *SPI* and serial interface and digital and *PWM* outputs

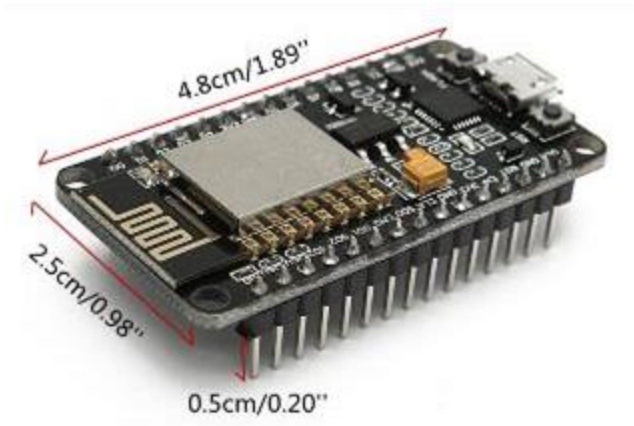
- It is **easy to use**, connects to computer via USB and communicates using standard serial protocol, runs in standalone mode and as interface connected to PC/Macintosh computers

- It is **inexpensive**, and comes with free authoring software

- Arduino is backed up by a growing **online community**, lots of source code is already available and we can share and post our examples for others to use, too.

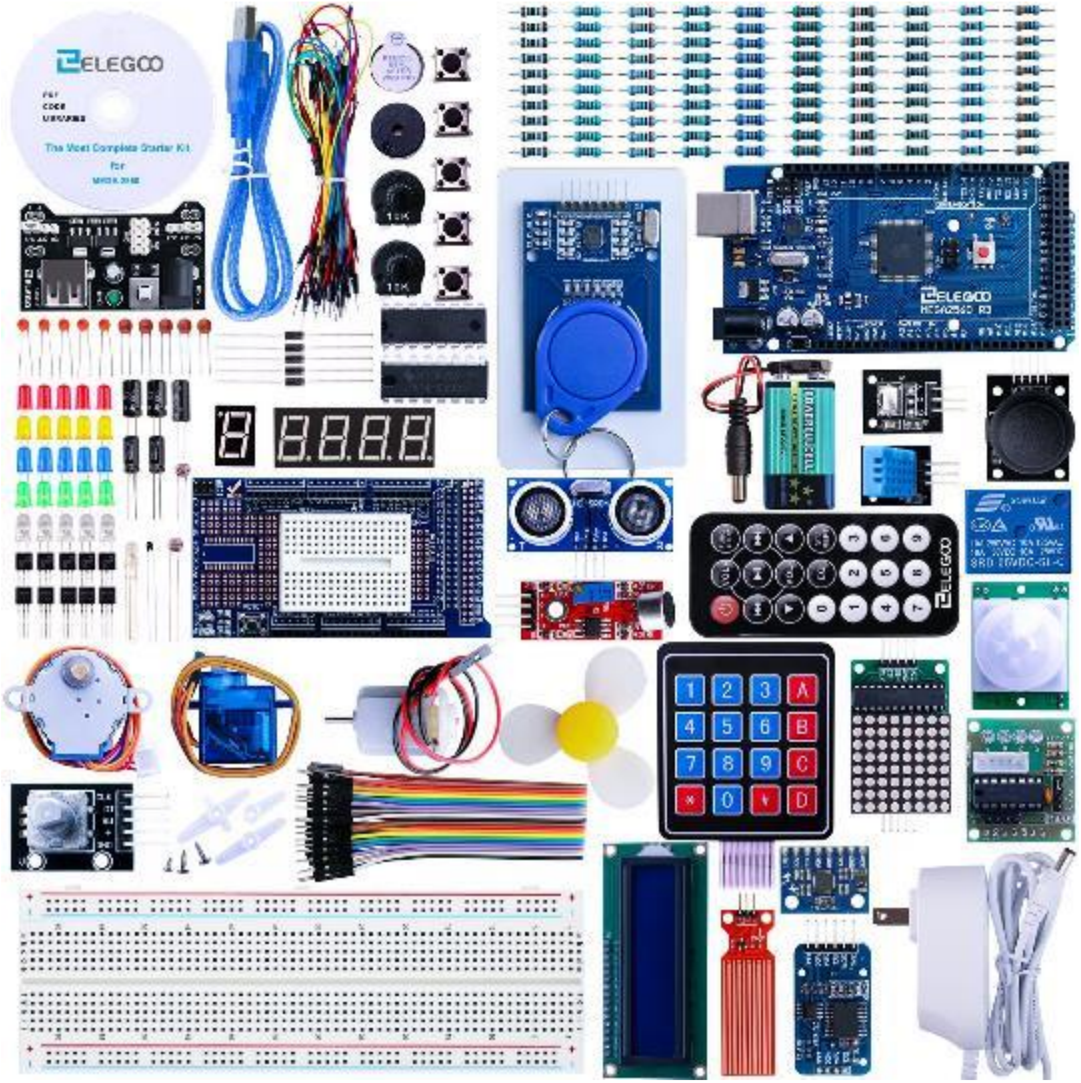
Arduino

Mega2560 R3 ATmega2560-16AU CH340 Development Board



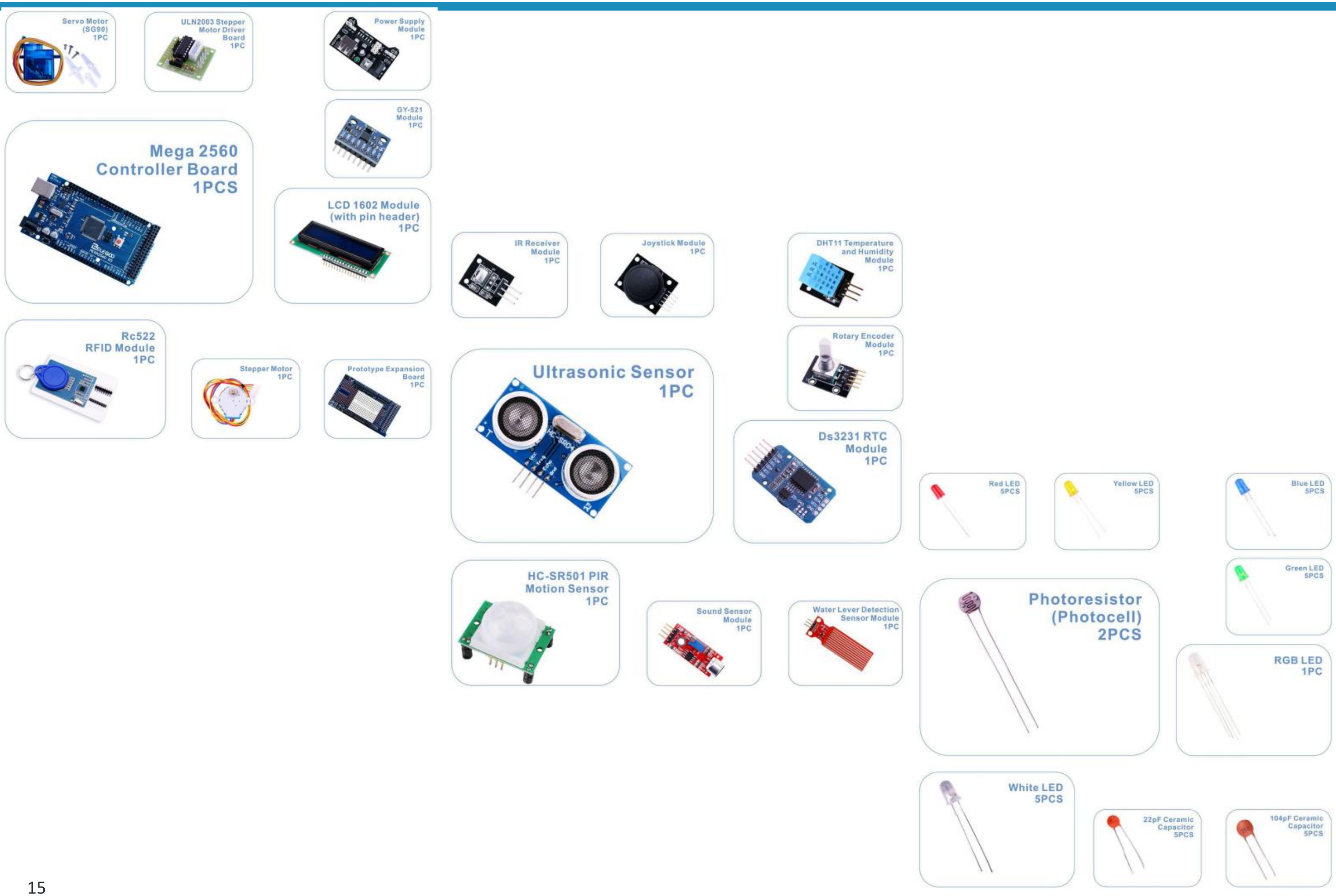
ESP8266 NodeMCU LUA CP2102 ESP-12E

Arduino



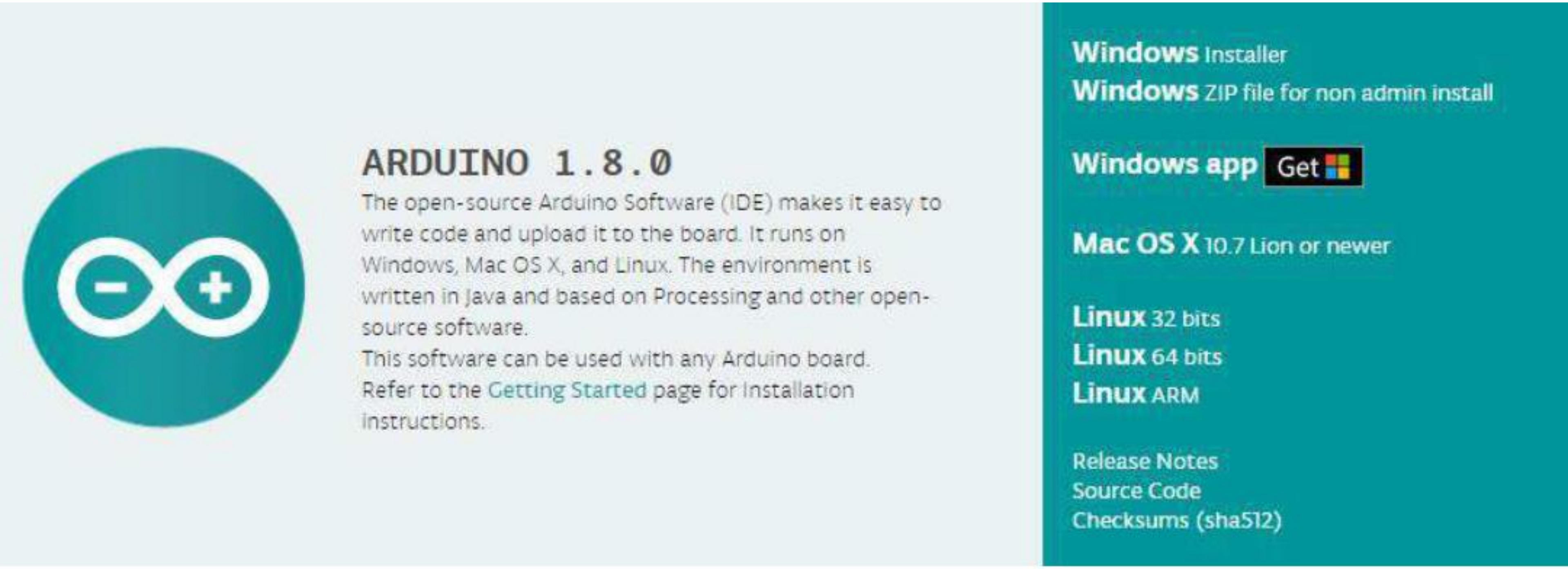
Elegoo EL-KIT-008 Mega 2560 Project
The Most Complete Ultimate Starter Kit

Arduino


Servo Motor (SG90) 1PC
ULN2003 Stepper Motor Driver Board 1PC
Power Supply Module 1PC
GY-521 Module 1PC
Mega 2560 Controller Board 1PCS
LCD 1602 Module (with pin header) 1PC
IR Receiver Module 1PC
Joystick Module 1PC
DHT11 Temperature and Humidity Module 1PC
Rc522 RFID Module 1PC
Stepper Motor 1PC
Prototype Expansion Board 1PC
Ultrasonic Sensor 1PC
Rotary Encoder Module 1PC
Ds3231 RTC Module 1PC
Red LED 5PCS
Yellow LED 5PCS
Blue LED 5PCS
Green LED 5PCS
HC-SR501 PIR Motion Sensor 1PC
Sound Sensor Module 1PC
Water Lever Detection Sensor Module 1PC
Photoresistor (Photocell) 2PCS
RGB LED 1PC
White LED 5PCS
22pF Ceramic Capacitor 5PCS
104pF Ceramic Capacitor 5PCS

Installing IDE

- The Arduino Integrated Development Environment



ARDUINO 1.8.0

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
This software can be used with any Arduino board.
Refer to the Getting Started page for Installation instructions.

Windows Installer
Windows ZIP file for non admin install

Windows app Get

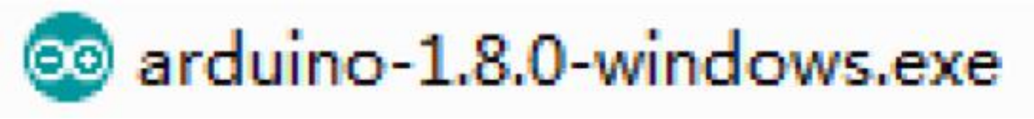
Mac OS X 10.7 Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM

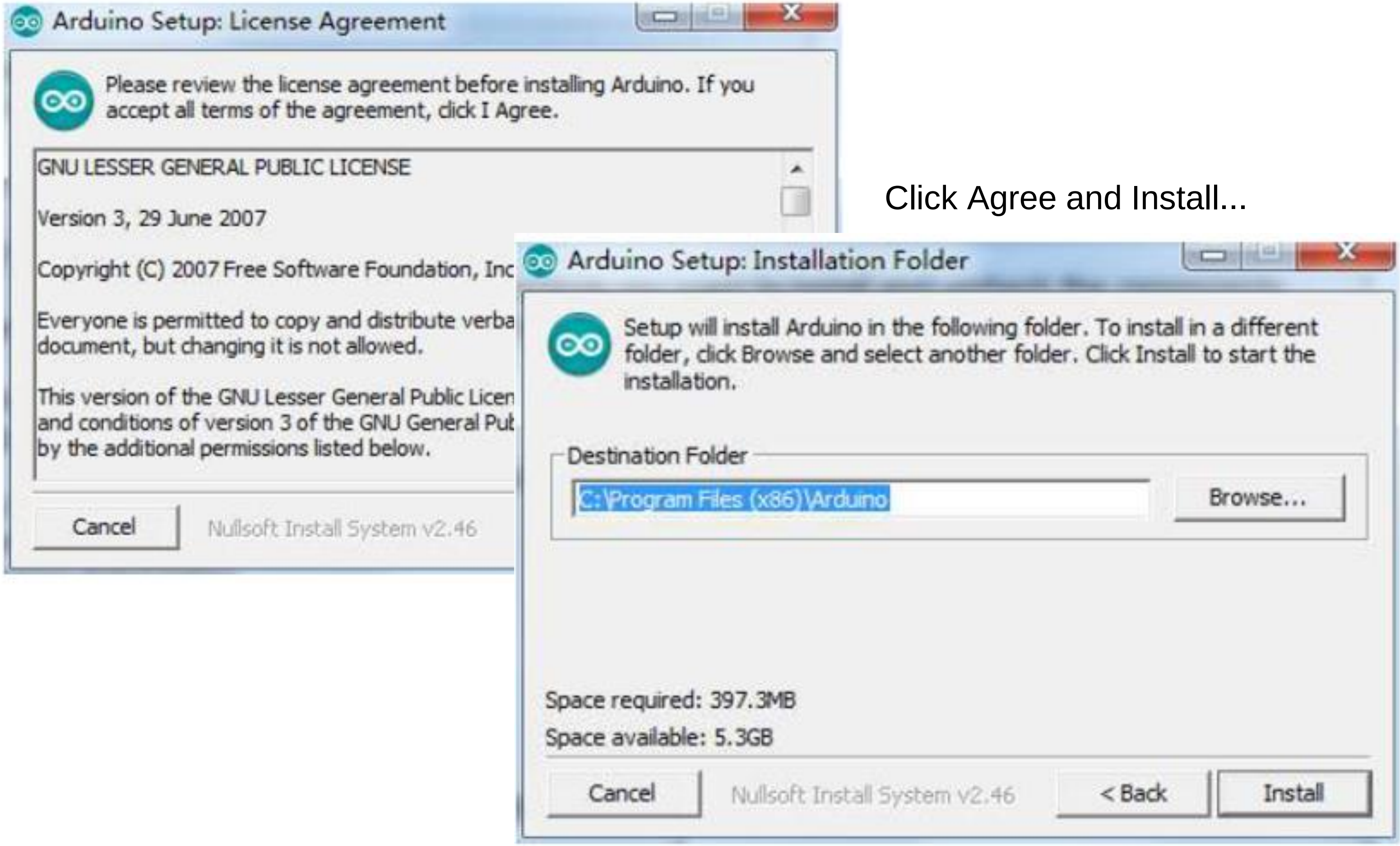
Release Notes
Source Code
Checksums (sha512)

The version available at this website is usually the latest version, and the actual version may be newer than the version in the picture.

Installing Arduino (Windows)



arduino-1.8.0-windows.exe



Arduino Setup: License Agreement

Please review the license agreement before installing Arduino. If you accept all terms of the agreement, click I Agree.

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc

Everyone is permitted to copy and distribute verba document, but changing it is not allowed.

This version of the GNU Lesser General Public Licen and conditions of version 3 of the GNU General Pub by the additional permissions listed below.

Arduino Setup: Installation Folder

Setup will install Arduino in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder: C:\Program Files (x86)\Arduino

Space required: 397.3MB

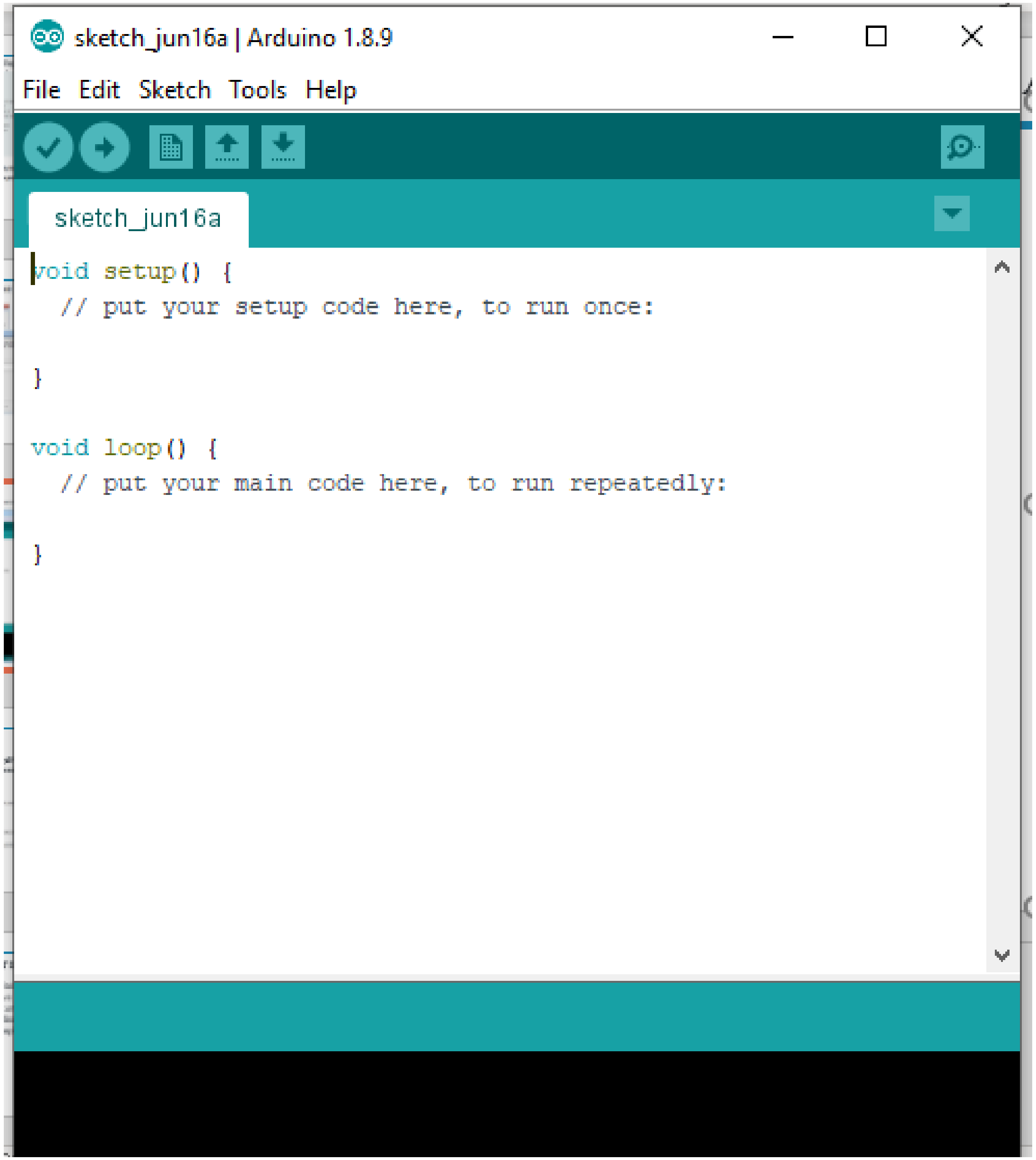
Space available: 5.3GB

Click Agree and Install...

The following Icon appears on the desktop

Arduino IDE

Double-click to enter the desired development environment



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Add Libraries and Open Serial Monitor

What are Libraries?

Libraries are a collection of code that makes it easy for you to connect to a sensor, display, module, etc. For example, the built-in LiquidCrystal library makes it easy to

talk to character LCD displays. There are hundreds of additional libraries available on the Internet for download. The built-in libraries and some of these additional libraries are listed in the reference. To use the additional libraries, you will need to install them.

Arduino Serial Monitor (Windows, Mac, Linux)

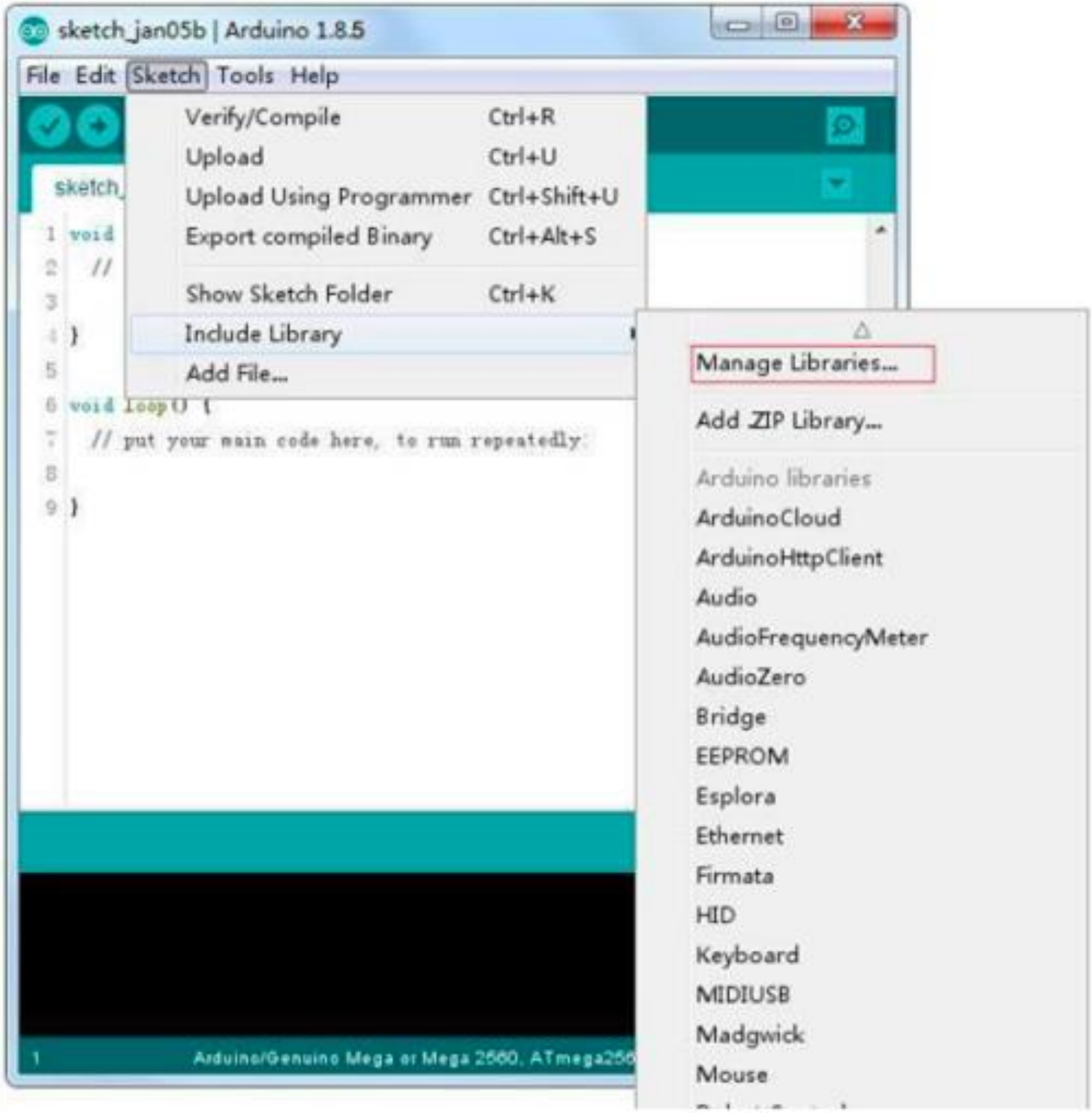
The Arduino Integrated Development Environment (IDE) is the software side of the Arduino platform. And, because using a terminal is such a big part of working with Arduinos and other microcontrollers, they decided to include a serial terminal with the software. Within the Arduino environment, this is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

Add Libraries and Open Serial Monitor

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.



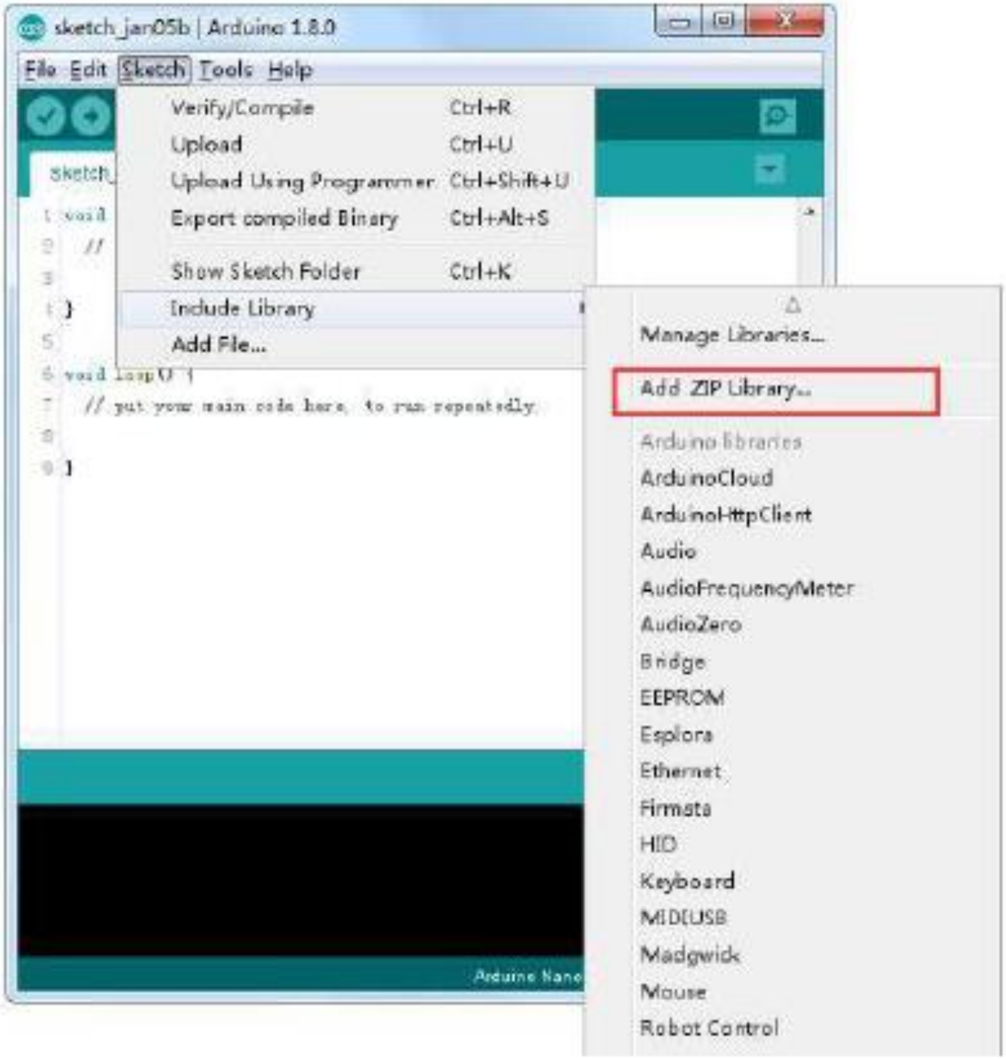
```
void
  //

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Add Libraries and Open Serial Monitor

How to Install a Library?

Importing a .zip Library.

In the Arduino IDE, navigate to Sketch > Include Library.
At the top of the drop down list, select the option to "Add .ZIP Library".

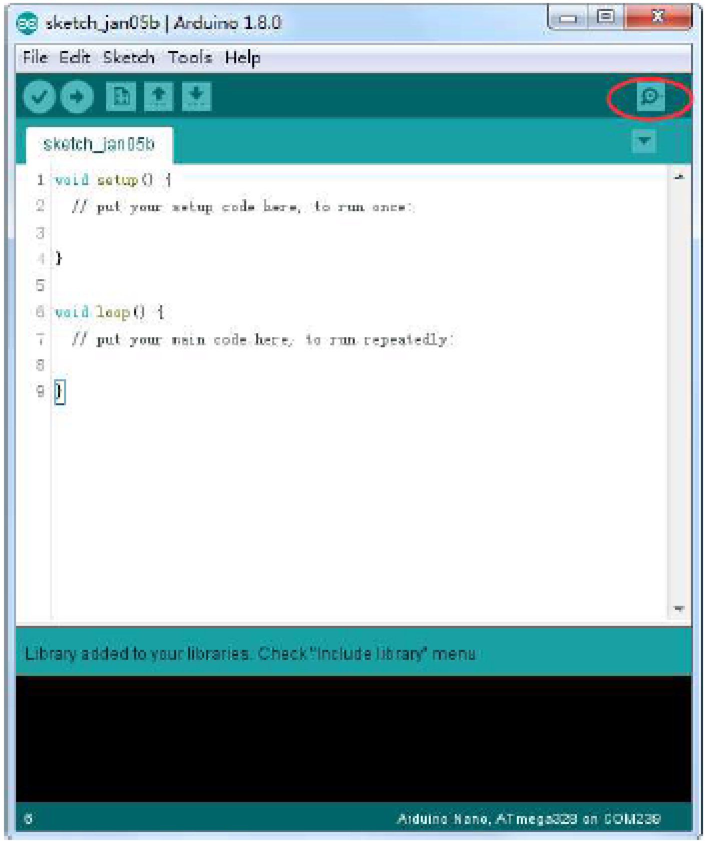


```
void
  //

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Add Libraries and Open Serial Monitor

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

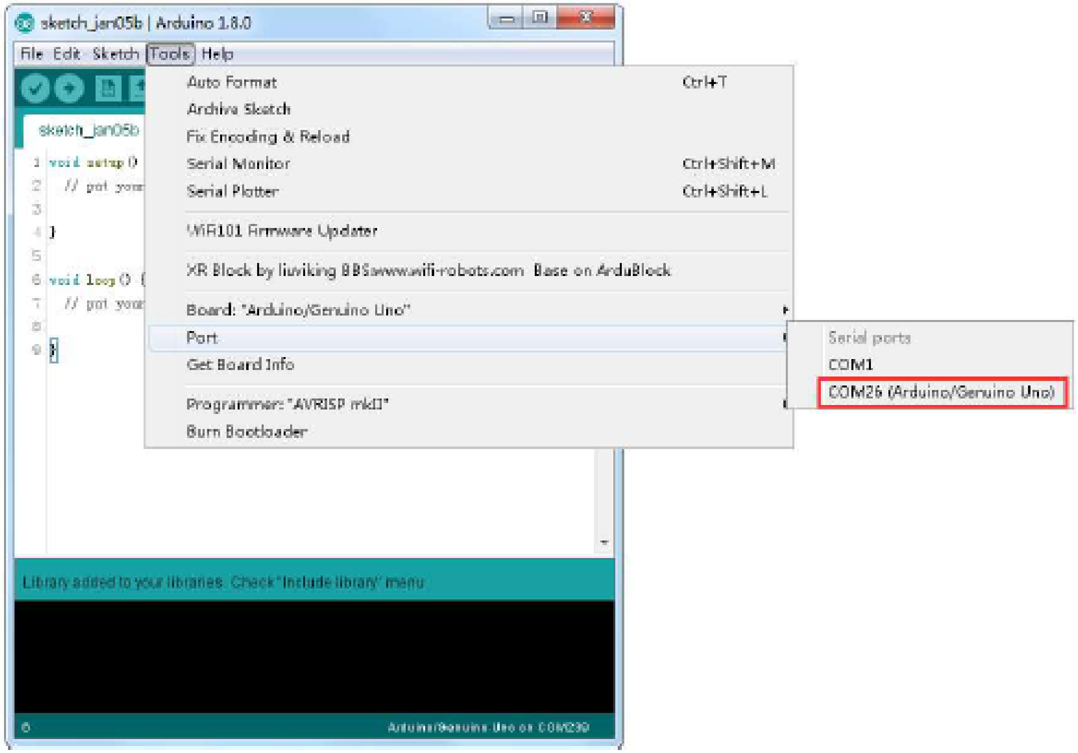
The Arduino Integrated Development Environment (IDE) is the software side of the Arduino platform. And, because using a terminal is such a big part of working with Arduinos and other microcontrollers, they decided to include a serial terminal with the software. Within the Arduino environment, this is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



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Library added to your libraries. Check "Include library" menu

Arduino Nano, ATmega328 on COM239



sketch_jan05b | Arduino 1.8.0
File Edit Sketch Tools Help
sketch_jan05b
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
XR Block by liuviking BBS:www.wifi-robots.com Base on ArduBlock
Board: "Arduino/Genuino Uno"
Port
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
COM1
COM26 (Arduino/Genuino Uno)

```
void setup()
  // put your

}

void loop() {
  // put your

}
```

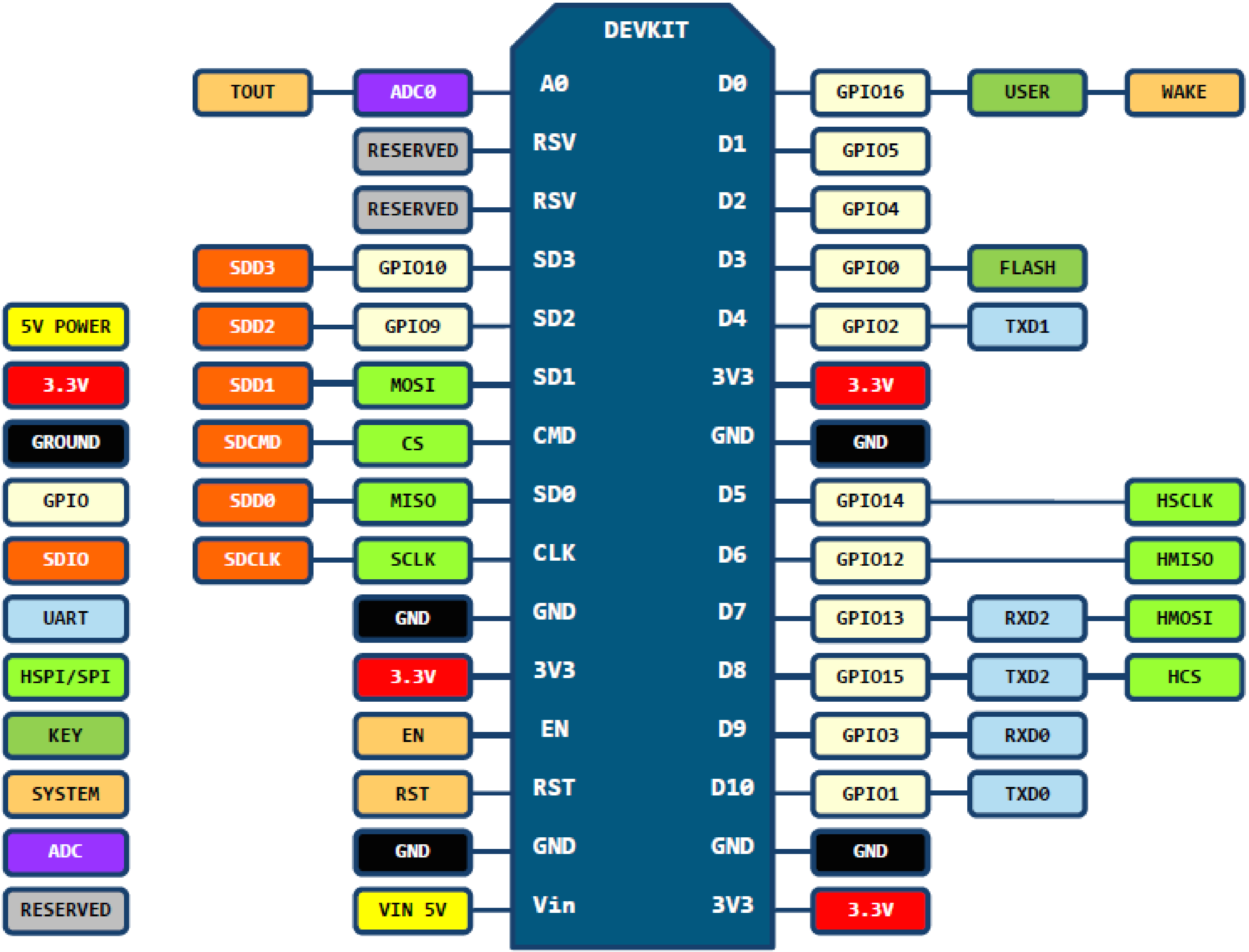
Library added to your libraries. Check "Include library" menu

Arduino/Genuino Uno on COM239

NodeMCU ESP8266 ESP-12E

- Voltage:3.3V.
- Wi-Fi Direct (P2P), soft-AP.
- Integrated TCP/IP protocol stack.
- GPIOs: 17 (multiplexed with other functions).
- Analog to Digital: 1 input with 1024 step resolution.
- 802.11 support: b/g/n.
- Maximum concurrent TCP connections: 5
- Good tutorial: https://www.handsontec.com/pdf_learn/esp8266-V10.pdf

NodeMCU ESP8266 ESP-12E Pinout



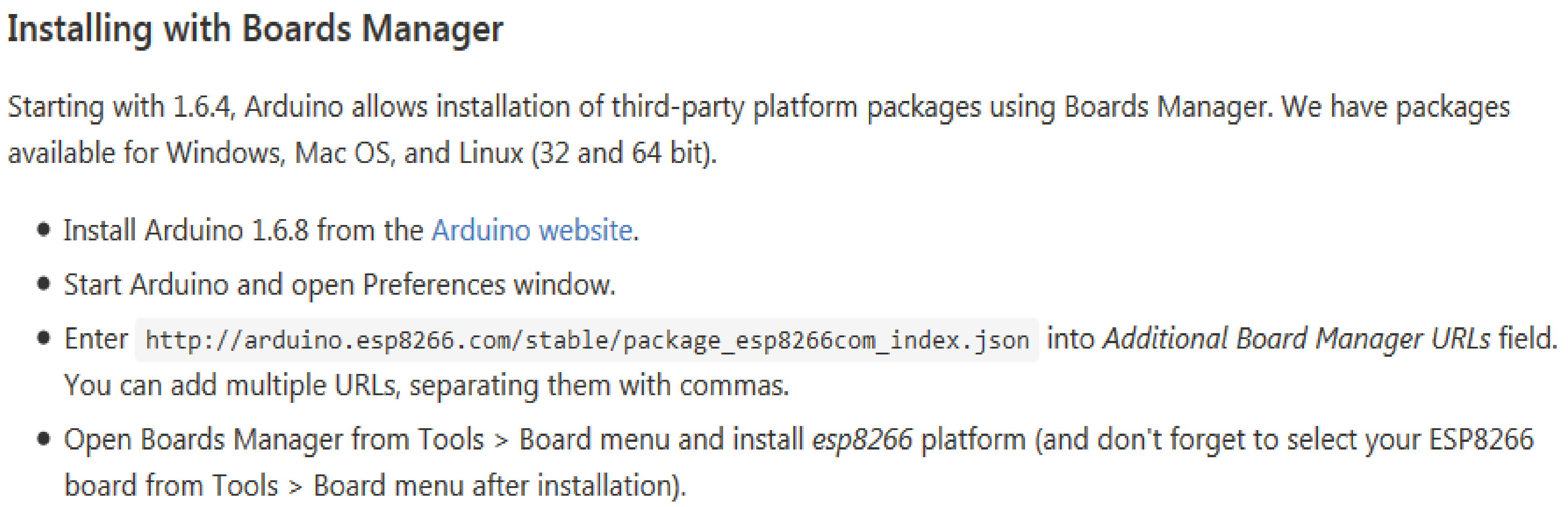
DEVKIT

TOUT	ADC0	A0	D0	GPIO16	USER	WAKE
	RESERVED	RSV	D1	GPIO5		
	RESERVED	RSV	D2	GPIO4		
SDD3	GPIO10	SD3	D3	GPIO0	FLASH	
SDD2	GPIO9	SD2	D4	GPIO2	TXD1	
SDD1	MOSI	SD1	3V3	3.3V		
SDCMD	CS	CMD	GND	GND		
SDD0	MISO	SD0	D5	GPIO14		HSCLK
SDCLK	SCLK	CLK	D6	GPIO12		HMISO
	GND	GND	D7	GPIO13	RXD2	HMOSI
	3.3V	3V3	D8	GPIO15	TXD2	HCS
	EN	EN	D9	GPIO3	RXD0	
	RST	RST	D10	GPIO1	TXD0	
	GND	GND	GND	GND		
	VIN 5V	Vin	3V3	3.3V		

- 5V POWER
- 3.3V
- GROUND
- GPIO
- SDIO
- UART
- HSPI/SPI
- KEY
- SYSTEM
- ADC
- RESERVED

NodeMCU setup

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



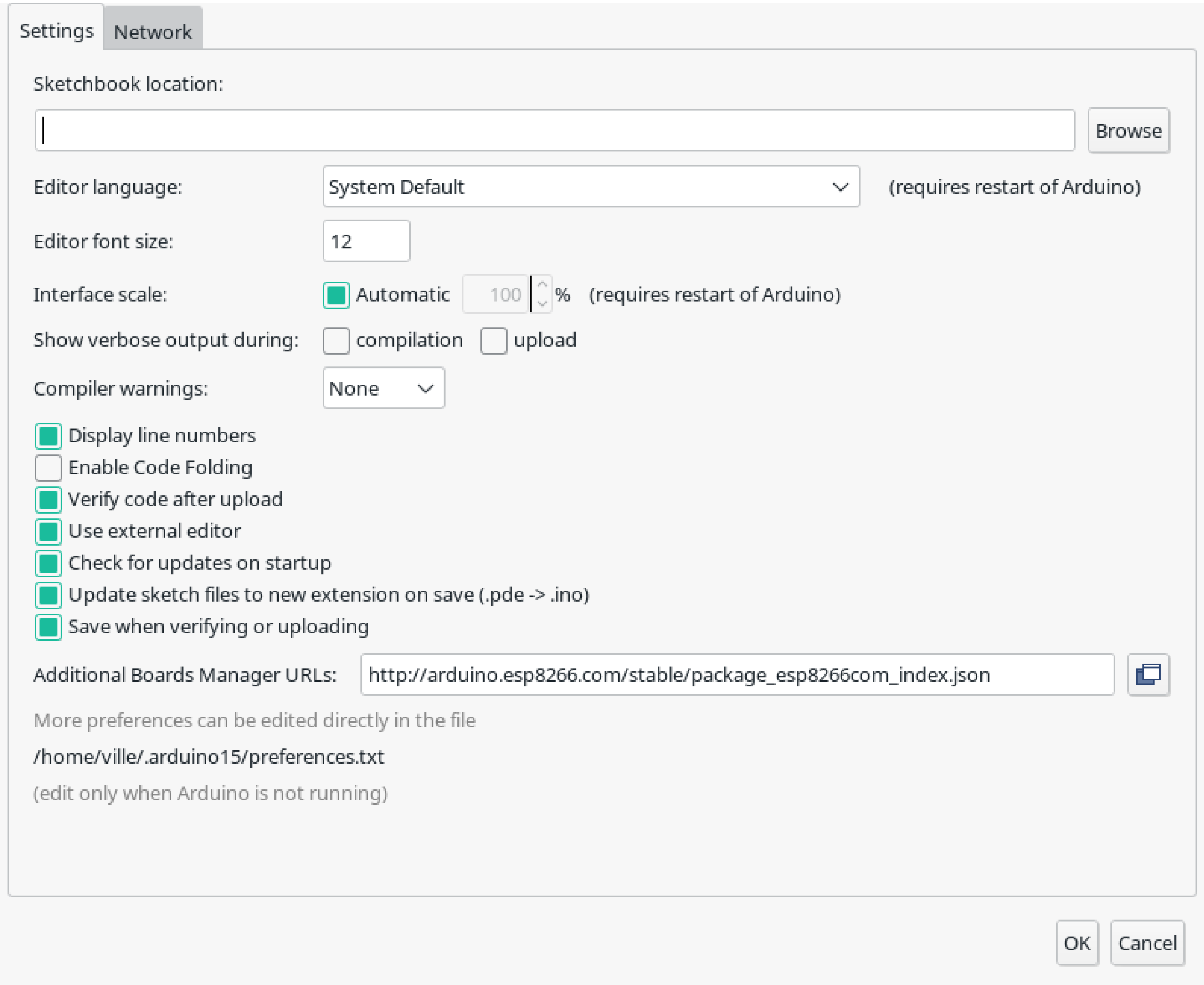
Installing with Boards Manager

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.
- 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

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



Sketchbook location:

Editor language: System Default (requires restart of Arduino)

Editor font size: 12

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Show verbose output during: □ compilation □ upload

Compiler warnings: None

☑ Display line numbers
□ Enable Code Folding
☑ Verify code after upload
☑ Use external editor
☑ Check for updates on startup
☑ Update sketch files to new extension on save (.pde -> .ino)
☑ Save when verifying or uploading

Additional Boards Manager URLs: http://arduino.esp8266.com/stable/package_esp8266com_index.json

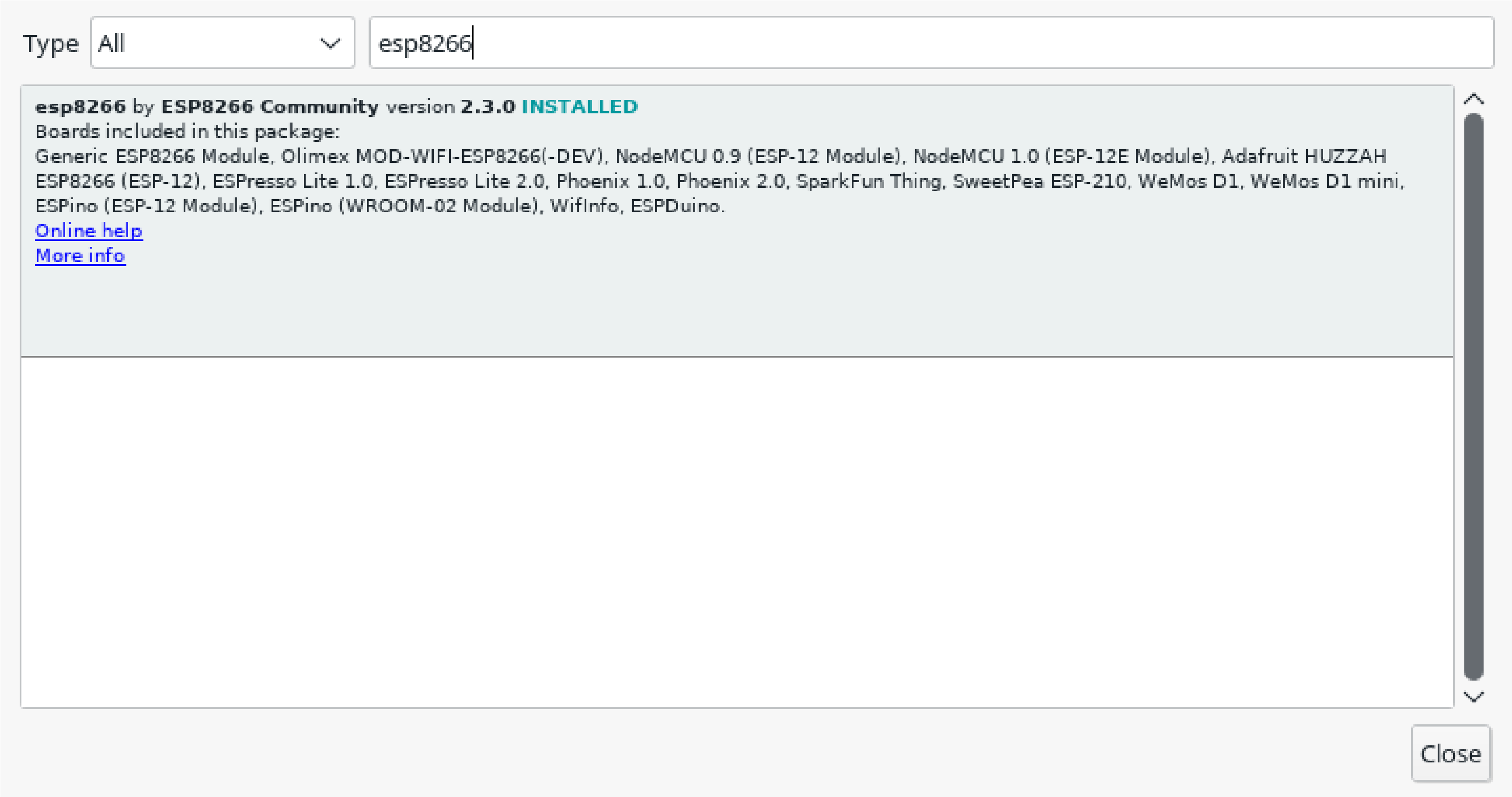
More preferences can be edited directly in the file

/home/ville/.arduino15/preferences.txt

(edit only when Arduino is not running)

NodeMCU setup

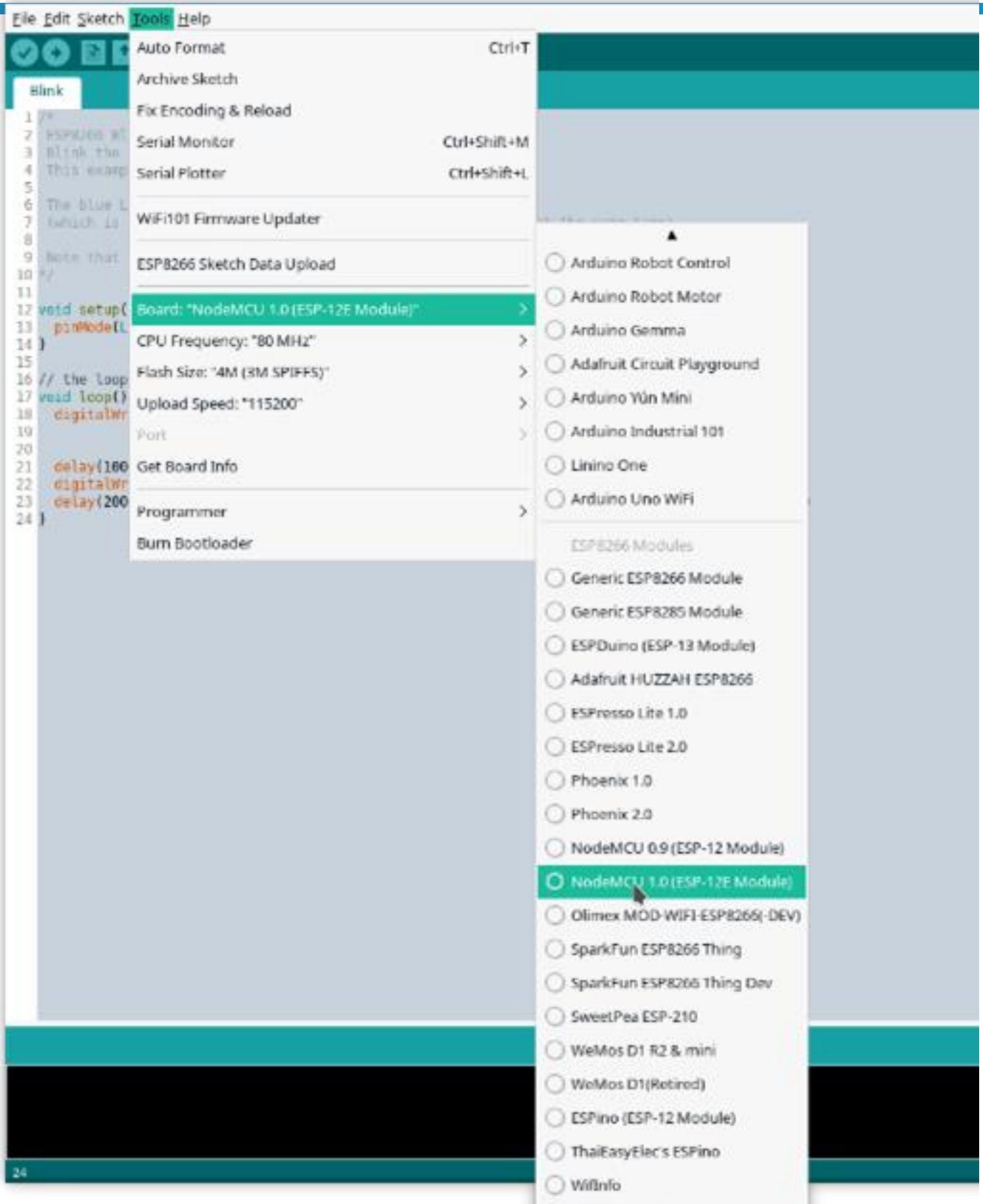
- Search the board manager for *esp8266*
- Install the library



esp8266 by **ESP8266 Community** version **2.3.0** **INSTALLED**
Boards included in this package:
Generic ESP8266 Module, Olimex MOD-WIFI-ESP8266(-DEV), NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Adafruit HUZZAH ESP8266 (ESP-12), ESPresso Lite 1.0, ESPresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, SparkFun Thing, SweetPea ESP-210, WeMos D1, WeMos D1 mini, ESPino (ESP-12 Module), ESPino (WROOM-02 Module), WifInfo, ESPDuino.
Online help
More info

NodeMCU setup

- 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



Board: "NodeMCU 1.0 (ESP-12E Module)"
CPU Frequency: "80 MHz"
Flash Size: "4M (3M SPIFFS)"
Upload Speed: "115200"

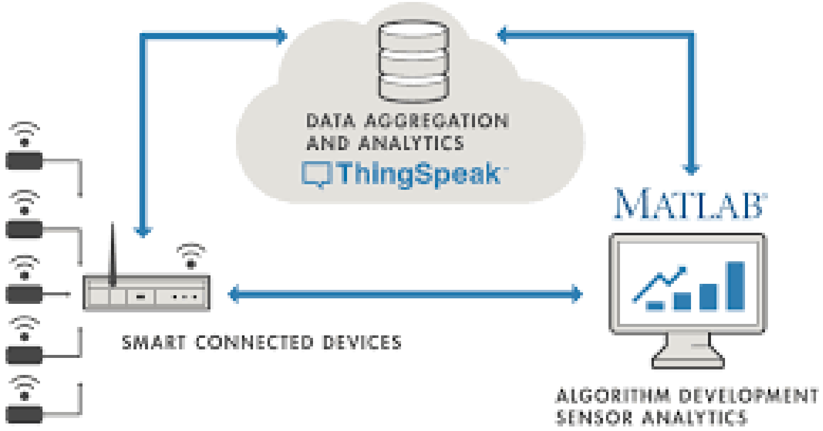
NodeMCU setup

- If the device doesn’t appear in the port menu after installing the library and connecting the board you can try installing the USB to Serial chip drivers.
- The chip is the ch340g
- Drivers can be found on the NodeMCU git: https://github.com/nodemcu/nodemcu-devkit/tree/master/Drivers

ThingSpeak

- ThingSpeak is a Web Service (REST API) that lets you collect and store sensor data in the cloud and develop Internet of Things
- applications.
- It works with Arduino, Raspberry Pi and MATLAB (premade libraries and APIs exists).
- But it should work with all kind of Programming Languages, since it uses a REST API and HTTP.

ThingSpeak.com



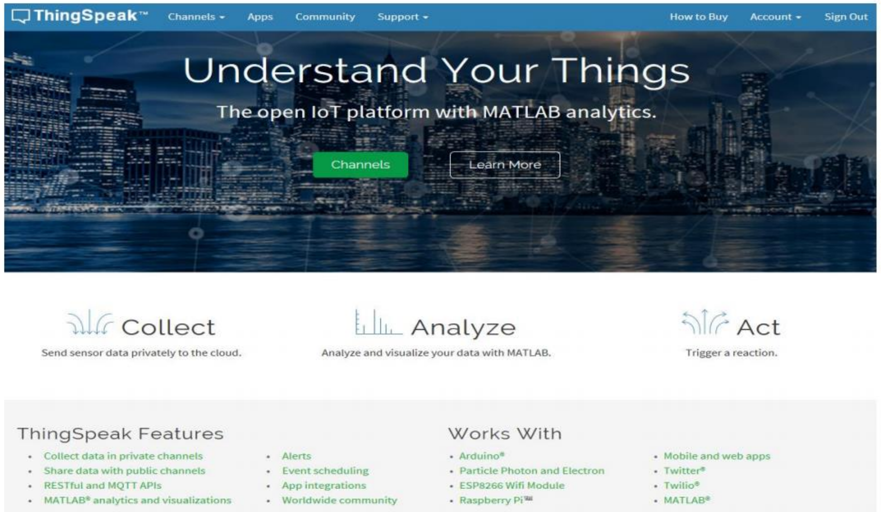
- https://thingspeak.com

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.

- **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.



Understand Your Things

The open IoT platform with MATLAB analytics.

Collect

Send sensor data privately to the cloud.

Analyze

Analyze and visualize your data with MATLAB.

Act

Trigger a reaction.

ThingSpeak Features

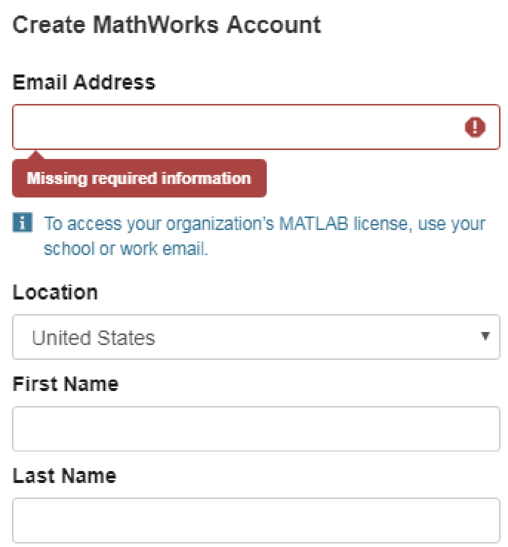
- Collect data in private channels
- Share data with public channels
- RESTful and MQTT APIs
- MATLAB® analytics and visualizations
- Alerts
- Event scheduling
- App integrations
- Worldwide community

Works With

- Arduino®
- Particle Photon and Electron
- ESP8266 Wifi Module
- Raspberry Pi™
- Mobile and web apps
- Twitter®
- Twilio®
- MATLAB®

Creating a channel :

- Before creating a channel you need to sign in to things speak. You can easily sign in either using your either thingspeak account or mathswork account, or create a new mathswork account via following link:
- https://thingspeak.com/users/sign_up



Create MathWorks Account

Email Address

Missing required information

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

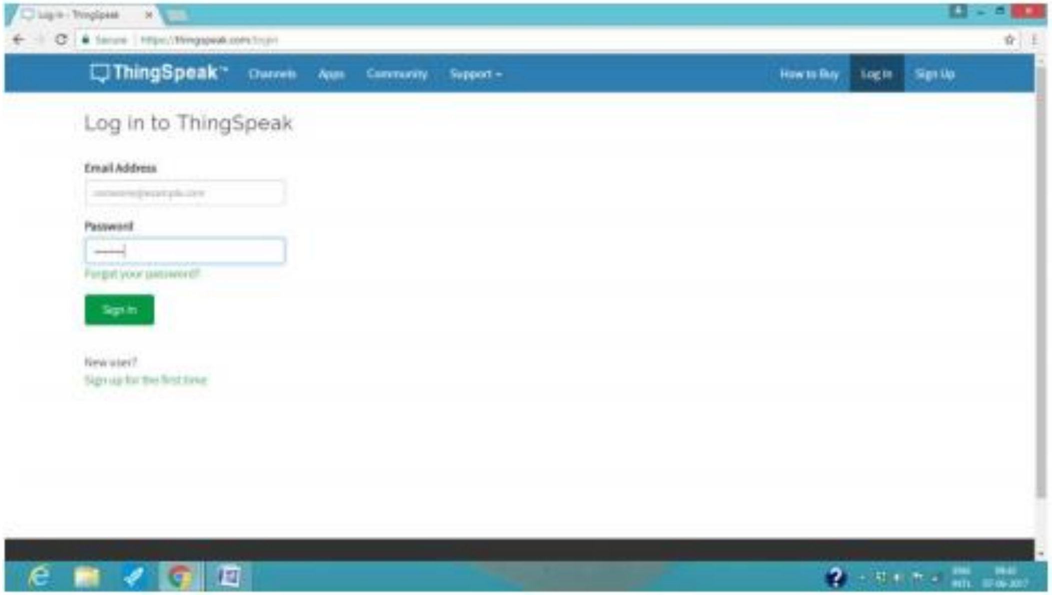
Location

United States

First Name

Last Name

- Login Page



Log in to ThingSpeak

Email Address

Password

Forgot your password?

Sign In

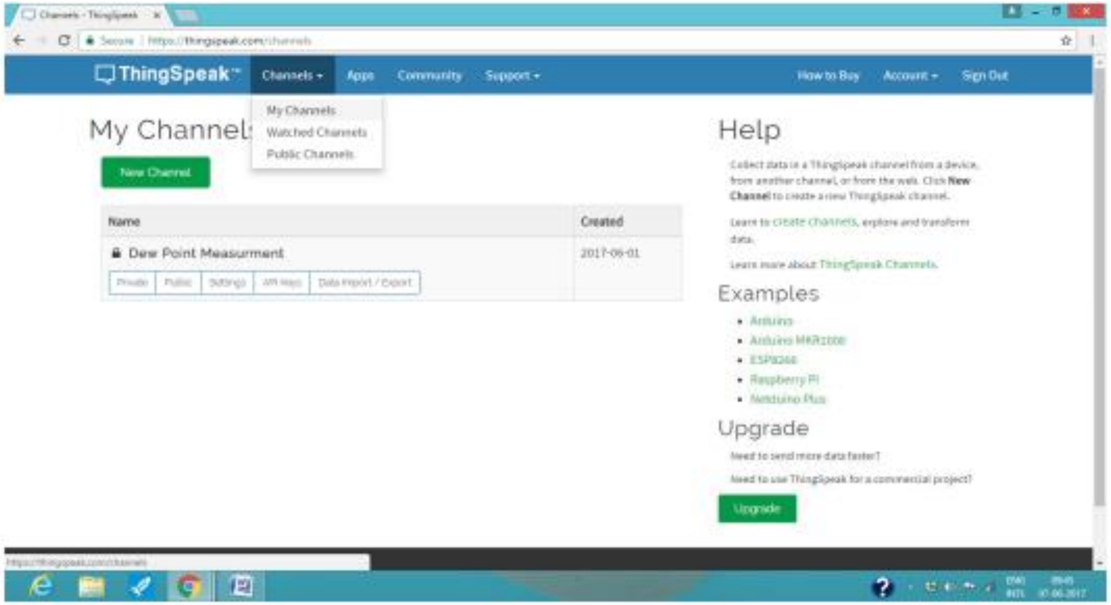
New user?

Sign up for the first time

- Email: mkuzlu@hotmail.com
- Password: ODU_Blast2019

Creating a channel :

1. Click on the menu bar **Channels> My Channels.**



Name	Created
🔒 Dew Point Measurment	2017-06-01

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.

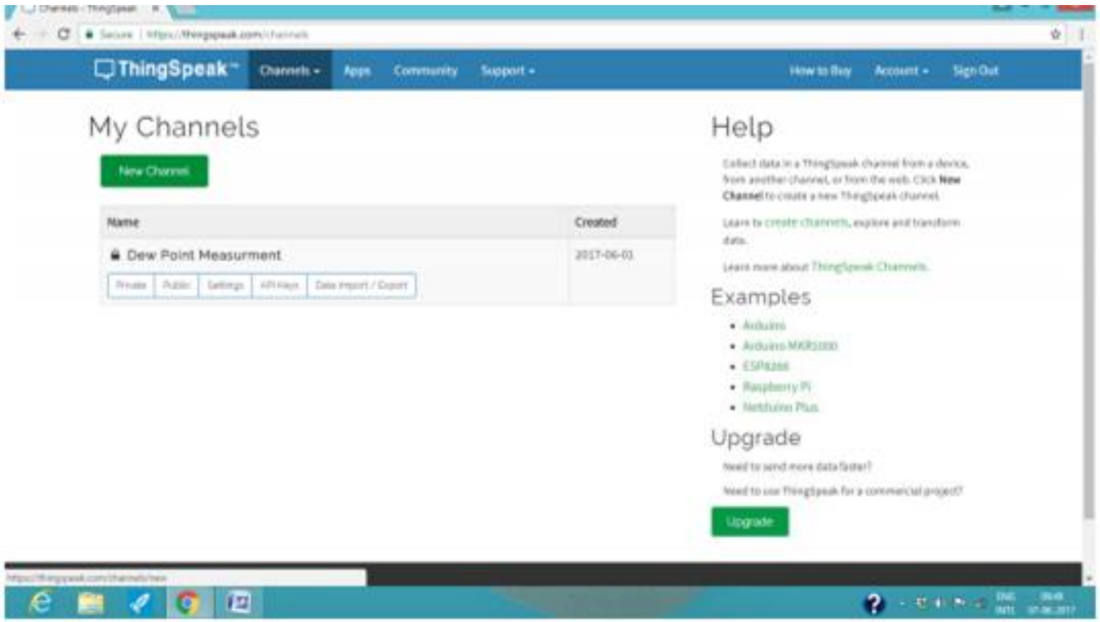
Learn to create channels, explore and transform data.

Learn more about ThingSpeak Channels.

Need to send more data faster?

Need to use ThingSpeak for a commercial project?

2. Now on the channels page click on the button **’New Channel.’**



Name	Created
🔒 Dew Point Measurment	2017-06-01

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.

Learn to create channels, explore and transform data.

Learn more about ThingSpeak Channels.

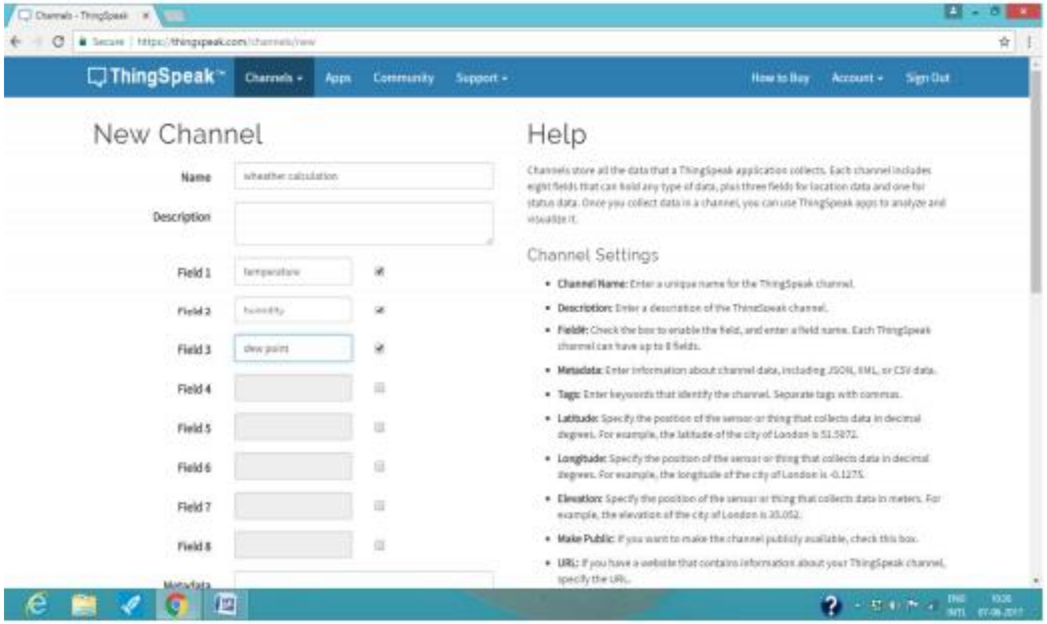
Need to send more data faster?

Need to use ThingSpeak for a commercial project?

Creating a channel :

3. New channel page have various text box fields showing the settings of the channel

- **a. Name --** provide a unique name to your channel.
- **b. Fields --** Click the check boxes next to the field and then enter the field name.
- c. To make your channel public check the **'Make Public'** check box
- d. Similarly, you can also add the location to your channel by clicking the **'Show Location'** check box
- e. Check the **'Show video'** check box to make the video visible uploaded by you.
- f. Now click the '**Save channel**' button to save your channel



New Channel

Name: wheather calculation

Description:

Field 1: temperature ☑

Field 2: humidity ☑

Field 3: dew point ☑

Field 4: ☐

Field 5: ☐

Field 6: ☐

Field 7: ☐

Field 8: ☐

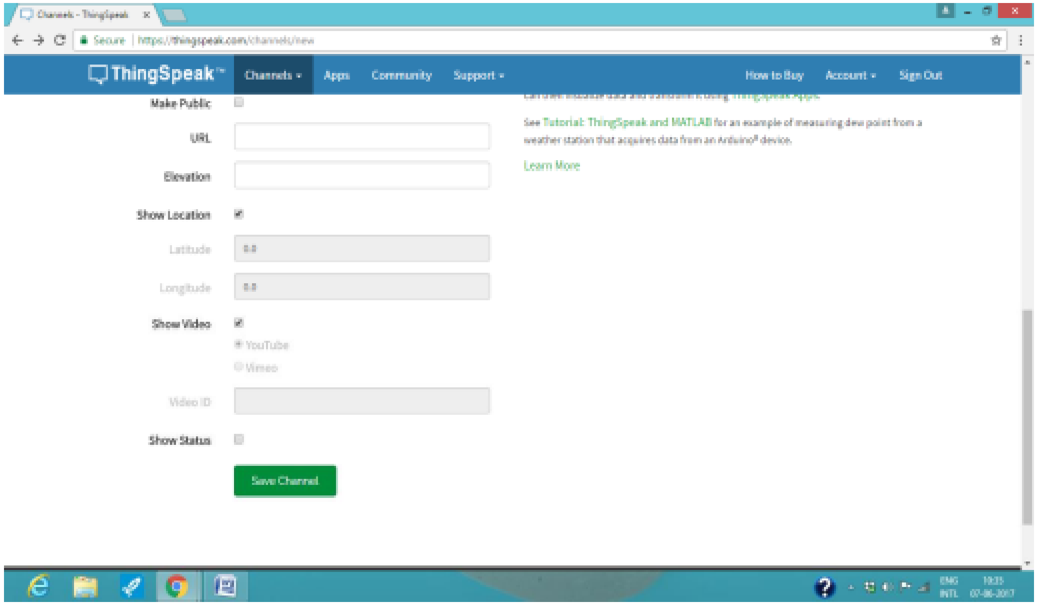
Metadata

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

- **Channel Name:** Enter a unique name for the ThingSpeak channel.
- **Description:** Enter a description of the ThingSpeak channel.
- **Fields:** 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.
- **Longitude:** Specify the position of the sensor or thing that collects data in decimal degrees. For example, the longitude of the city of London is -0.1275.
- **Elevation:** Specify the position of the sensor or thing that collects data in meters. For example, the elevation of the city of London is 35.052.
- **Make Public:** If you want to make the channel publicly available, check this box.
- **URL:** If you have a website that contains information about your ThingSpeak channel, specify the URL.



Make Public ☐

URL:

Elevation:

Show Location ☑

Latitude: 0.0

Longitude: 0.0

Show Video ☑

● YouTube

○ Vimeo

Video ID:

Show Status ☐

Save Channel

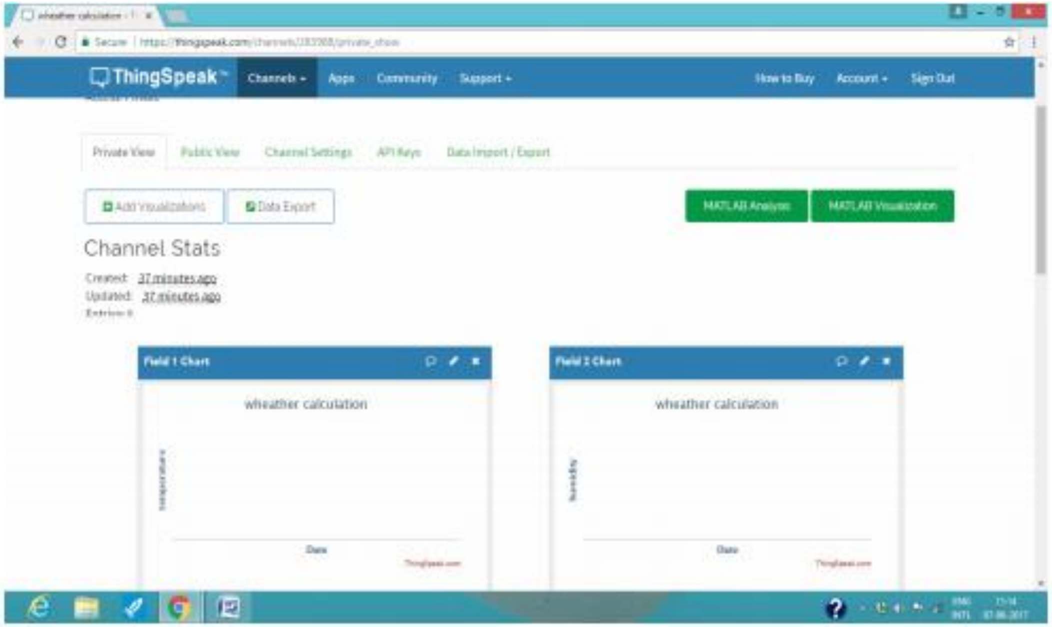
See Tutorial: ThingSpeak and MATLAB for an example of measuring dew point from a weather station that acquires data from an Arduino® device.

Learn More

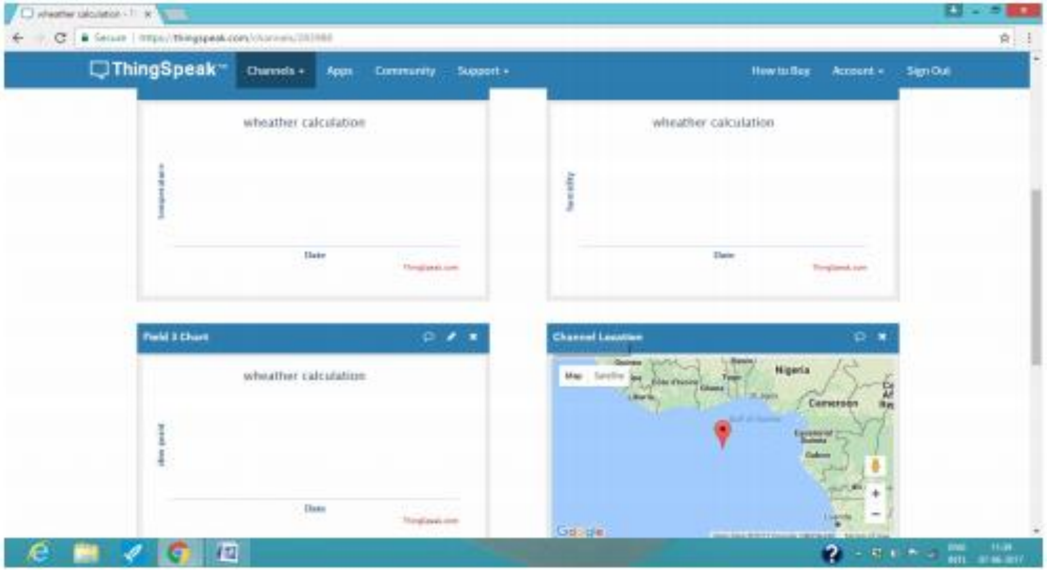
Creating a channel :

4. Now, the channel page opens with the following tabs:

a. Private View-- It displays the information about your channel that is only visible to you

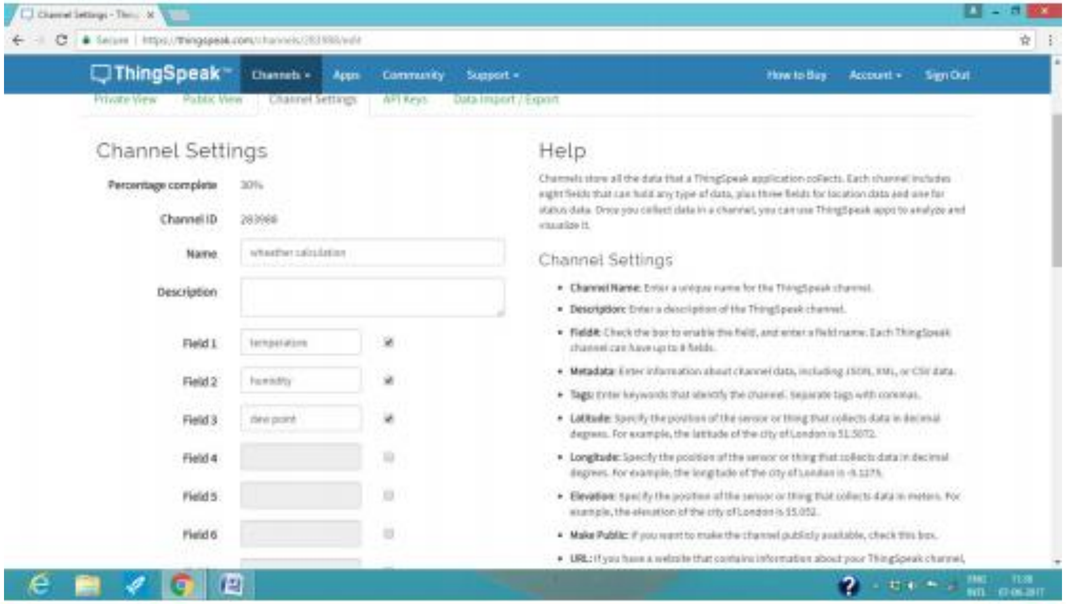


b. **Public View-** if you have chosen to make your channel publicly visible then it will display the selected fields and information



Creating a channel :

c. **Channel Settings-** it will show all the options that are available during the channel creation



Channel Settings - Thi... ×

Secure | https://thingspeak.com/channels/283988/edit

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 30%

Channel ID 283988

Name wheather calculation

Description

Field 1 temperature ☑

Field 2 humidity ☑

Field 3 dew point ☑

Field 4 □

Field 5 □

Field 6 □

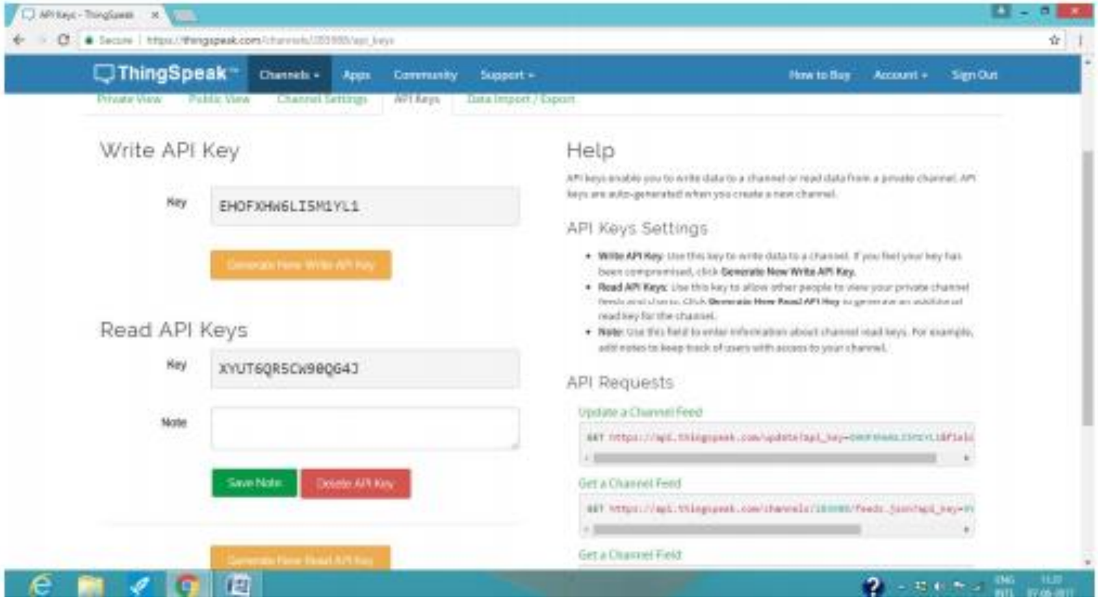
Help

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.
- **Longitude:** Specify the position of the sensor or thing that collects data in decimal degrees. For example, the longitude of the city of London is -0.1275.
- **Elevation:** Specify the position of the sensor or thing that collects data in meters. For example, the elevation of the city of London is 35.052.
- **Make Public:** If you want to make the channel publicly available, check this box.
- **URL:** If you have a website that contains information about your ThingSpeak channel,

d. **API Keys-** in this tab you will have two API Keys -- Read API Key (to read from your channel), write API Key (to write to your channel)



API Keys - ThingSpeak ×

Secure | https://thingspeak.com/channels/283988/api_keys

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

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

Write API Key

Key EHOFXHW6LI5M1YL1

Generate New Write API Key

Read API Keys

Key XYUT6QR5CW98QG4J

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 a Channel Feed

```
GET https://api.thingspeak.com/update?api_key=EHOFXHW6LI5M1YL1&field
```

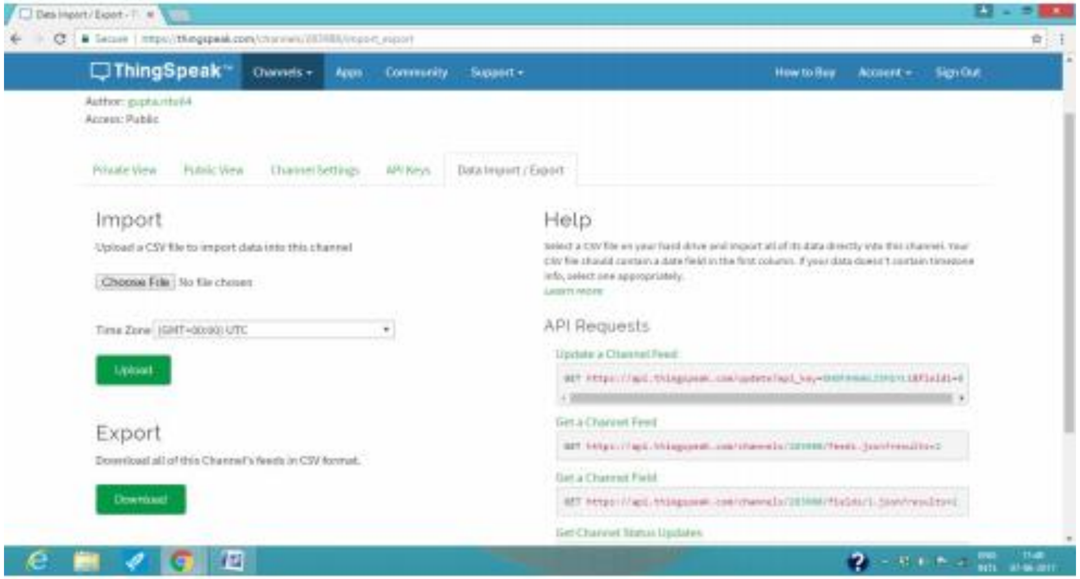
Get a Channel Feed

```
GET https://api.thingspeak.com/channels/283988/feeds.json?api_key=XY
```

Get a Channel Field

Creating a channel :

e. **Data import/export-** it enables you to import and export the channel data



Author: [illegible]
Access: Public

Import

Upload a CSV file to import data into this channel

No file chosen

Time Zone (GMT+00:00) UTC

Export

Download all of this Channel's feeds in CSV format.

Help

Select a CSV file on your hard drive and import all of its data directly into this channel. Your CSV file should contain a date field in the first column. If your data doesn't contain timezone info, select one appropriately.
Learn more

API Requests

Update a Channel Feed

```
GET https://api.thingspeak.com/update?api_key=[illegible]&field1=0
```

Get a Channel Feed

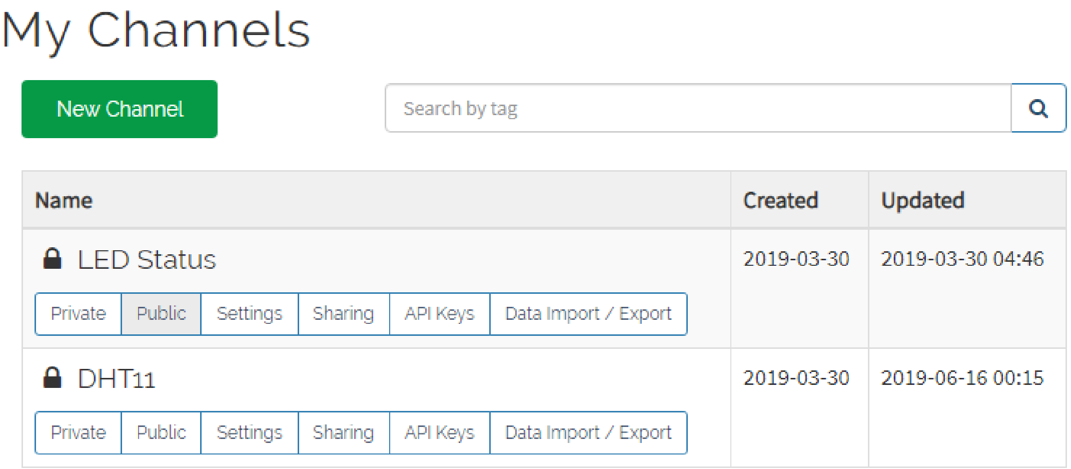
```
GET https://api.thingspeak.com/channels/[illegible]/feeds.json?results=2
```

Get a Channel Field

```
GET https://api.thingspeak.com/channels/[illegible]/fields/1.json?results=2
```

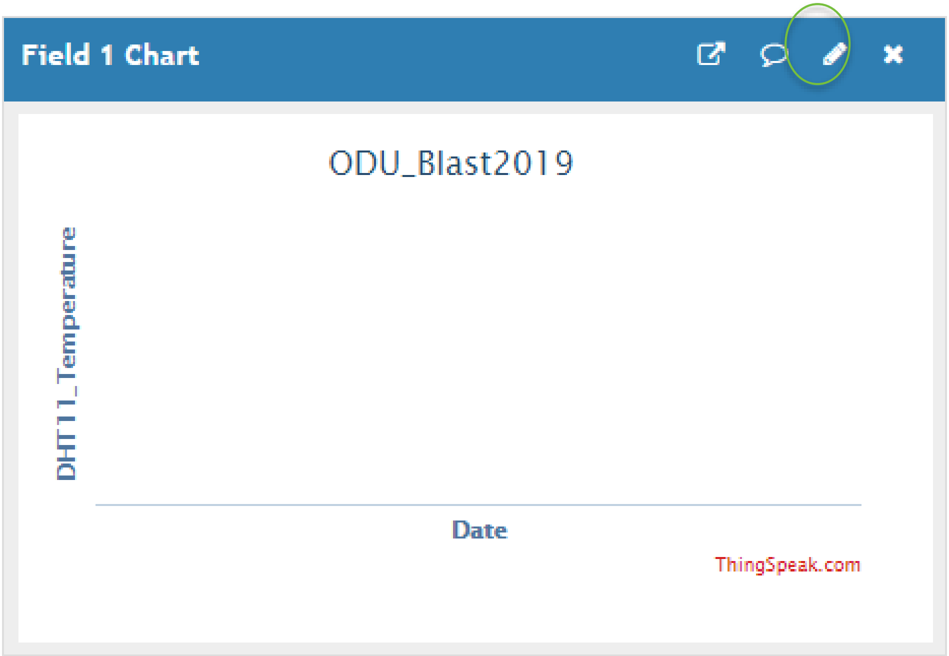
Get Channel Status Updates

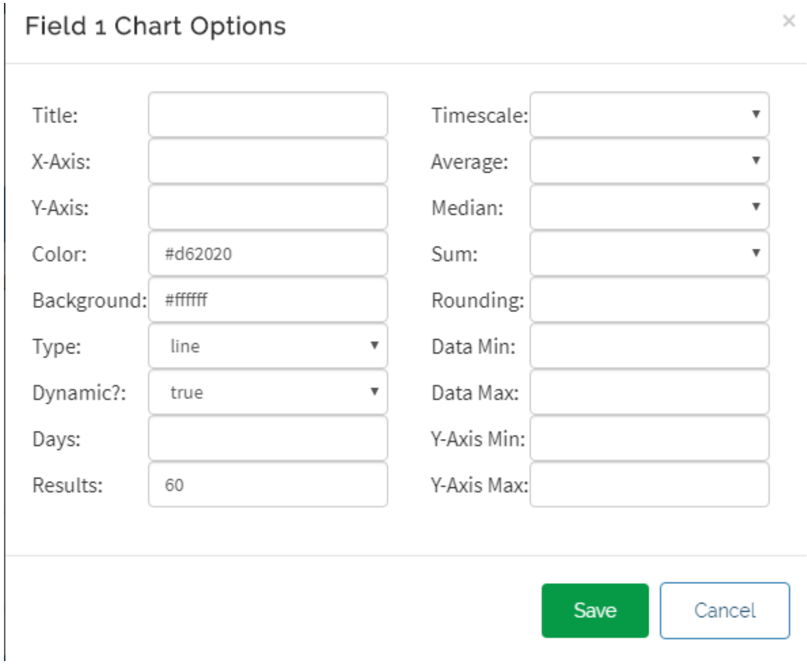
f. In future your channel will be available to you just by clicking **‘Channels> My Channels’**



Name	Created	Updated
LED Status Private · Public · Settings · Sharing · API Keys · Data Import / Export	2019-03-30	2019-03-30 04:46
DHT11 Private · Public · Settings · Sharing · API Keys · Data Import / Export	2019-03-30	2019-06-16 00:15

Chart Options:



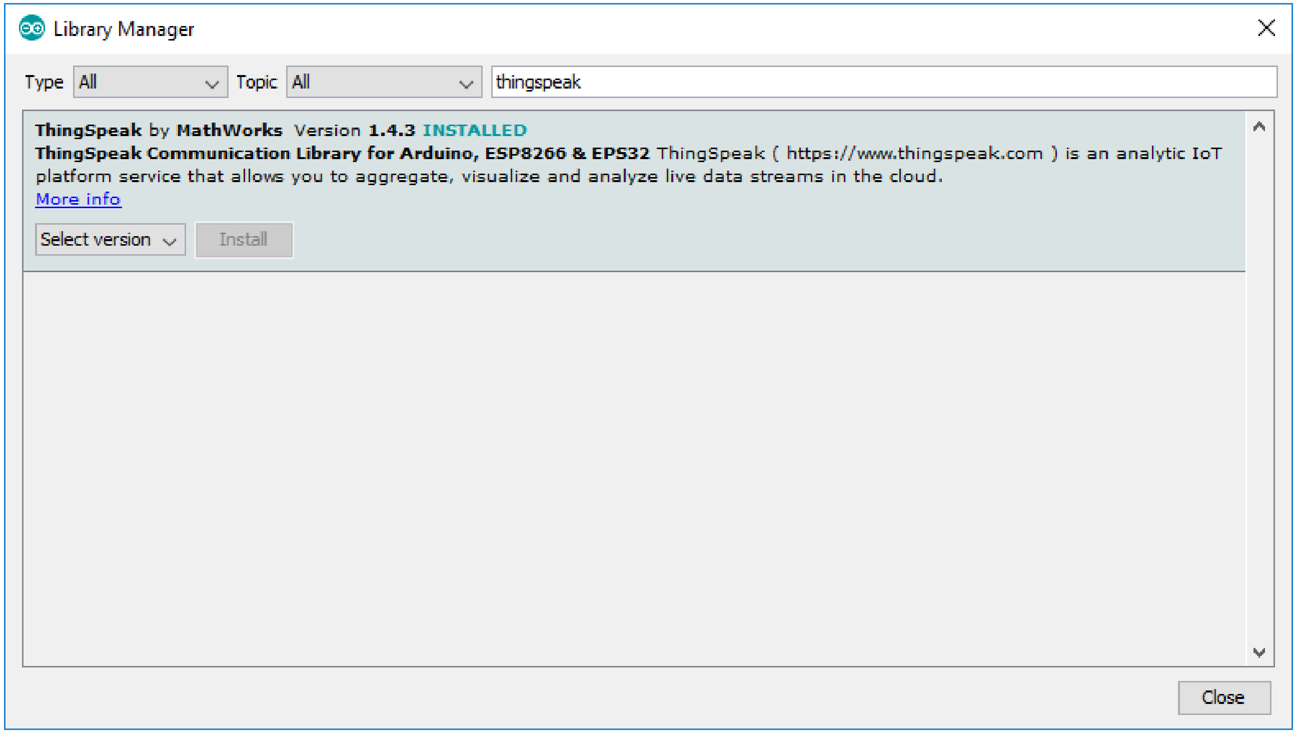


Field 1 Chart Options

Title:		Timescale:	
X-Axis:		Average:	
Y-Axis:		Median:	
Color:	#d62020	Sum:	
Background:	#ffffff	Rounding:	
Type:	line	Data Min:	
Dynamic?:	true	Data Max:	
Days:		Y-Axis Min:	
Results:	60	Y-Axis Max:	

ThingSpeak Library for 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.



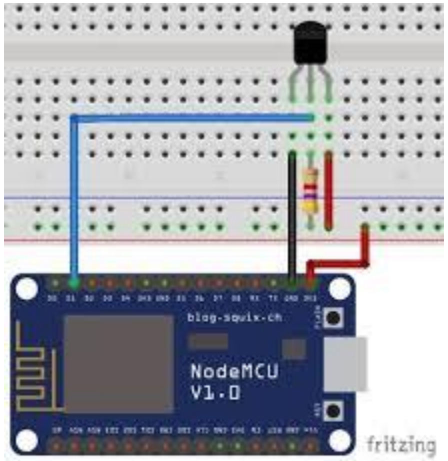
ThingSpeak by **MathWorks** Version **1.4.3** **INSTALLED**
ThingSpeak Communication Library for Arduino, ESP8266 & EPS32 ThingSpeak (https://www.thingspeak.com) is an analytic IoT platform service that allows you to aggregate, visualize and analyze live data streams in the cloud.
More info

Topics of this workshop

- **Getting started**
  (setup and programming of IoT hardware)

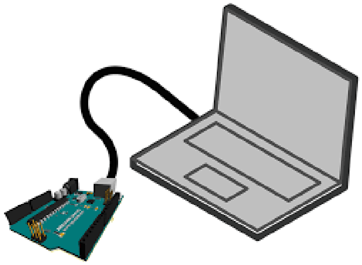
- **Reading and Writing**
  (physical computing: sensors and actuators)

- **Connecting** your device **to the Internet**
  (IoT: monitoring sensors, controlling actuators)



IoT hardware

Any Internet-connected computer with an **interface to the real world** (sensors, actuators)



Small => can be **embedded into things**

Small computer = **microcontroller** (or **board**), e.g. Arduino, Netduino Plus, BeagleBone, …

Hands on Activities and Source Code

Copy&Paste or Download examples, from github:

https://github.com/muratkuzlu/ODU_BLAST2019

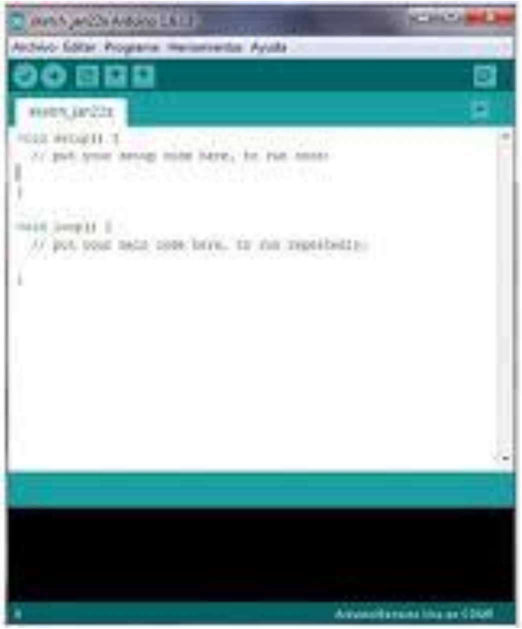
Focus on **end-to-end results, not details**

Getting started

The **IDE** (**I**ntegrated **D**evelopment **E**nvironment) allows you to **program** your board, i.e. “make it do something new”

You **edit** a program on your computer, then **upload** it to your board where it’s stored in the program memory (flash) and **executed** in RAM

Please Note: Once it has been programmed, your board can run on its own, without another computer


sketch_jan22a Arduino 1.6.1
Archivo Editar Programa Herramientas Ayuda
sketch_jan22a

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

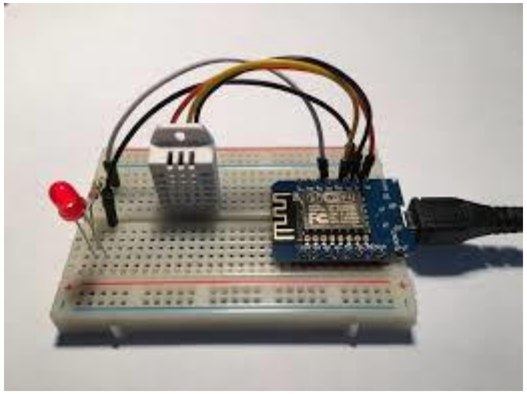
Reading and Writing

IoT hardware has an **interface to the real world**

GPIO (**G**eneral **P**urpose **I**nput/**O**utput) pins

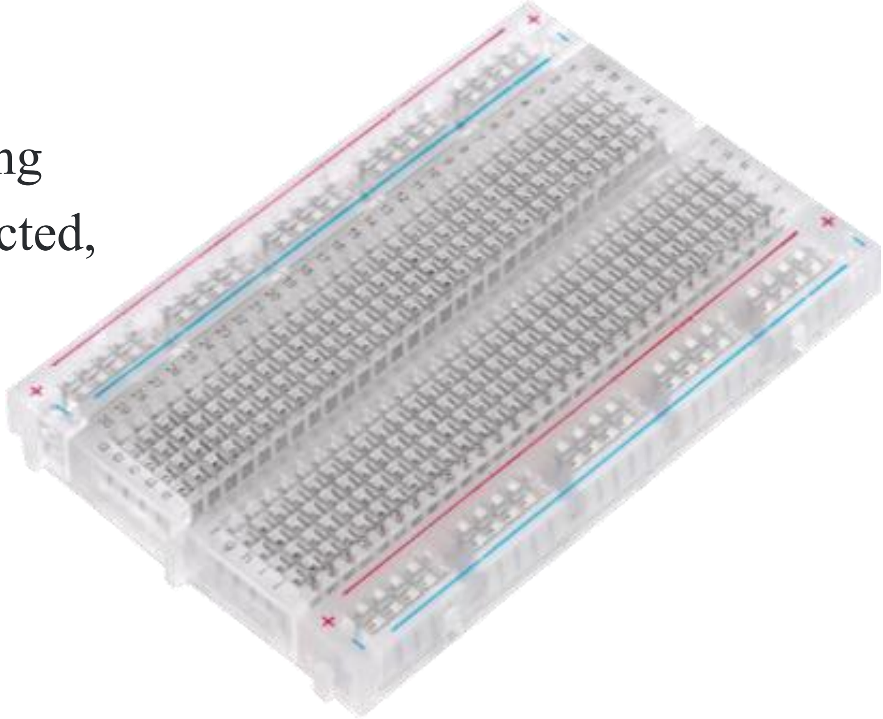
Measure: **read** sensor value from **input** pin
Manipulate: **write** actuator value to **output** pin

Inputs and outputs can be **digital or analog**



Prototyping Circuits Solderless Breadboard

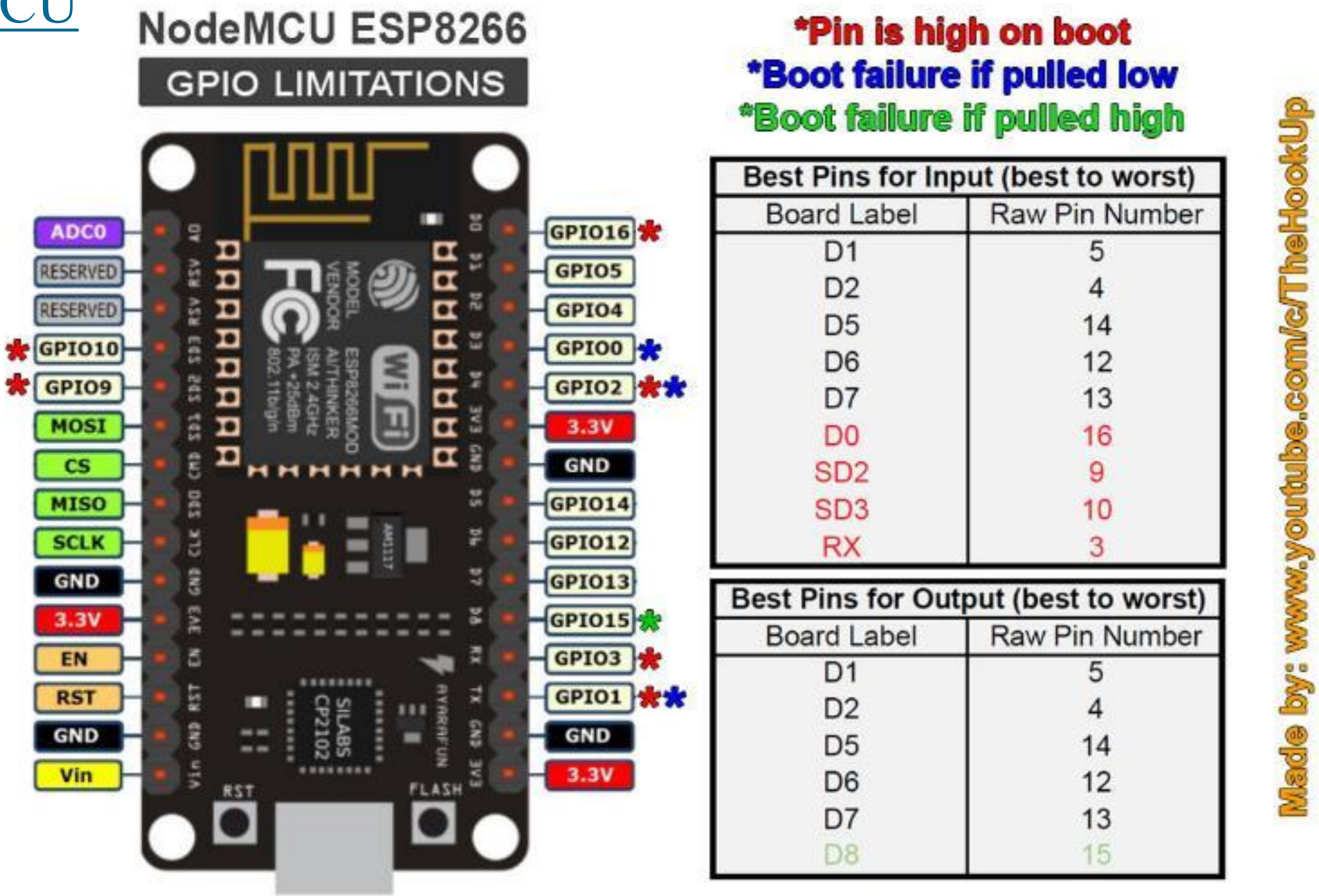
- One of the most useful tools in an engineer or Maker's toolkit.
- The three most important things:
  - A breadboard is easier than soldering
  - A lot of those little holes are connected, which ones?
  - Sometimes breadboards break



Wiring a LED with Arduino

Hardware

- NodeMCU



*Pin is high on boot

*Boot failure if pulled low

*Boot failure if pulled high

Best Pins for Input (best to worst)	
Board Label	Raw Pin Number
D1	5
D2	4
D5	14
D6	12
D7	13
D0	16
SD2	9
SD3	10
RX	3

Best Pins for Output (best to worst)	
Board Label	Raw Pin Number
D1	5
D2	4
D5	14
D6	12
D7	13
D8	15

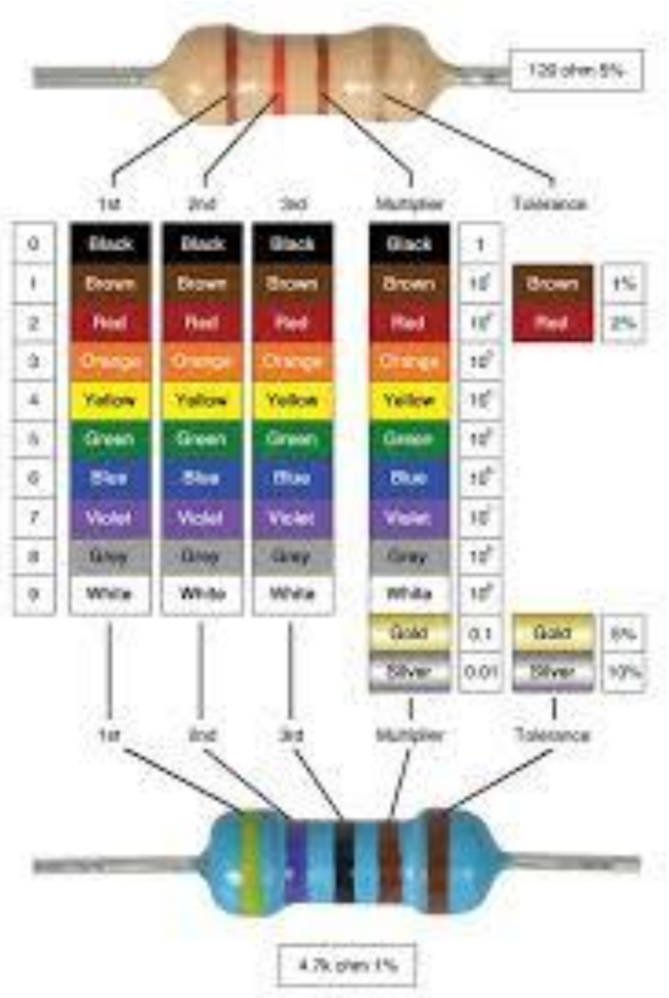
Made by: www.youtube.com/c/TheHookUp

The resistor

Resistors are the **workhorse of electronics**

Resistance is **measured in Ω** (Ohm) and adds up in series; a resistors orientation doesn't matter

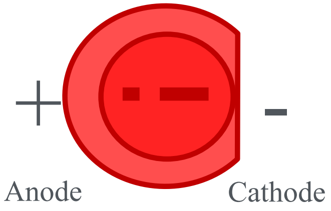
A resistors Ω value is **color-coded** right on it

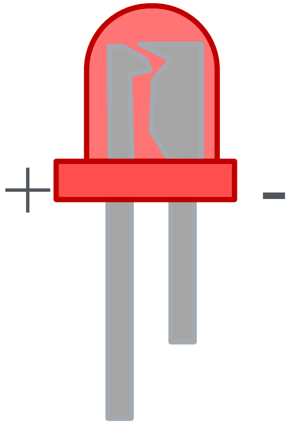


	1st	2nd	3rd	Multiplier		Tolerance	
0	Black	Black	Black	Black	1		
1	Brown	Brown	Brown	Brown	10^1	Brown	1%
2	Red	Red	Red	Red	10^2	Red	2%
3	Orange	Orange	Orange	Orange	10^3		
4	Yellow	Yellow	Yellow	Yellow	10^4		
5	Green	Green	Green	Green	10^5		
6	Blue	Blue	Blue	Blue	10^6		
7	Violet	Violet	Violet	Violet	10^7		
8	Gray	Gray	Gray	Gray	10^8		
9	White	White	White	White	10^9		
				Gold	0.1	Gold	5%
				Silver	0.01	Silver	10%

The LED

The **LED** (**L**ight **E**mitting **D**iode)
is a simple, digital **actuator**





LEDs have a **short leg** (-) and a **long leg** (+)
and it matters how they are oriented in a circuit

To prevent damage, LEDs are used together with a 1KΩ **resistor** (or anything from 300Ω to 2KΩ)

Hands-on Activity - I

NodeMCU

Wiring a LED with Arduino

We will introduce how to blink the on-board LED and how to blink a external LED.

Hardware

- NodeMCU x 1
- LED x 1
- 1K ohm resistor x 1
- Micro USB cable x 1
- PC x 1

Software

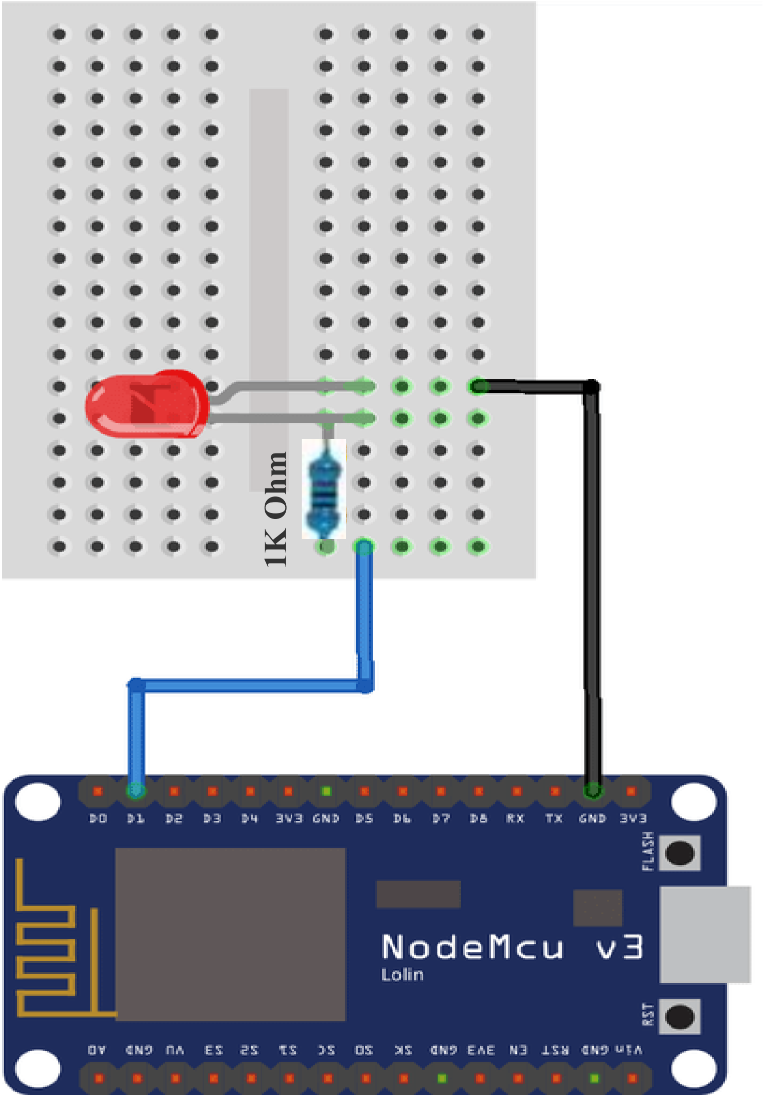
- Arduino IDE(version 1.6.4+)

Click in the link for reference

Wiring a LED with Arduino

Set up

- ➔ Connect the long leg of the LED (the positive leg, called the anode) to the other end of the resistor (1K ohm).
- ➔ Connect the short leg of the LED (the negative leg, called the cathode) to the GND.
- ➔ In the diagram below we show a NodeMCU that has D1 as the LED_BUILTIN value.



Digital output with Arduino

Code

HIGH = digital 1 (5V) means LED is **on**, **LOW** = digital 0 (0V) means LED is **off**

- ➔ Copy the following code to the IDE

```
#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.
}
```

Specifying what pin is going to be used

Initialize variables. Runs once

Used to actively control the Arduino board. Run repeatedly

Note that LOW is the voltage level but actually the LED is on; this is because it is active low on the ESP8266.

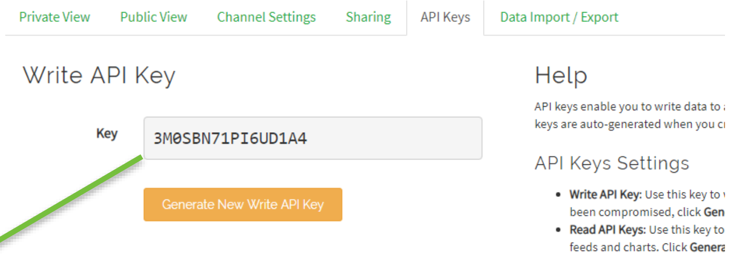
➔ Upload

Digital output with Arduino and IoT - I

Now we are going to connect to IoT

➔ Copy the following code to a new IDE sketch

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

const char *ssid =  “ssid";     // replace with your wifi ssid and wpa2 key
const char *pass =  “password";
const char* server = "api.thingspeak.com";
//  Enter your Write API key from ThingSpeak
 const char * myWriteAPIKey = "3M0SBN71PI6UD1A4";
unsigned long myChannelNumber = 803487;
uint8_t LED_Status, l=0;

WiFiClient client;

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



Write API Key

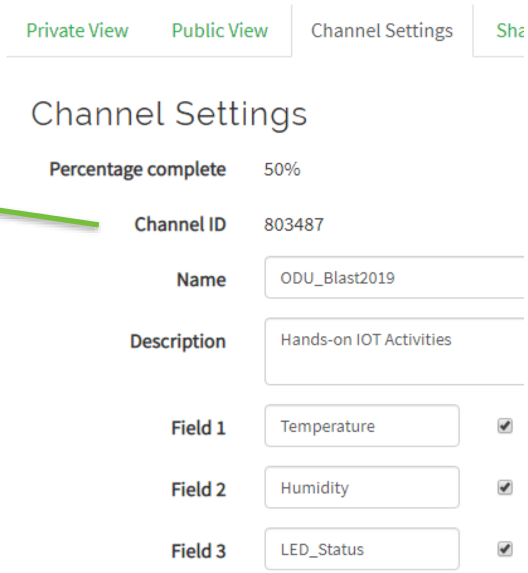
Key: 3M0SBN71PI6UD1A4

Help

API keys enable you to write data to
keys are auto-generated when you c

API Keys Settings

- **Write API Key:** Use this key to
  been compromised, click **Gen**
- **Read API Keys:** Use this key to
  feeds and charts. Click **Genera**



Channel Settings

Percentage complete	50%
Channel ID	803487
Name	ODU_Blast2019
Description	Hands-on IOT Activities
Field 1	Temperature ☑
Field 2	Humidity ☑
Field 3	LED_Status ☑

Digital output with Arduino and IoT - II

```
void setup()
{
     Serial.begin(115200);
     delay(10);

     pinMode(LED, OUTPUT);    // LED D1 pin as output.

     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);
}
```

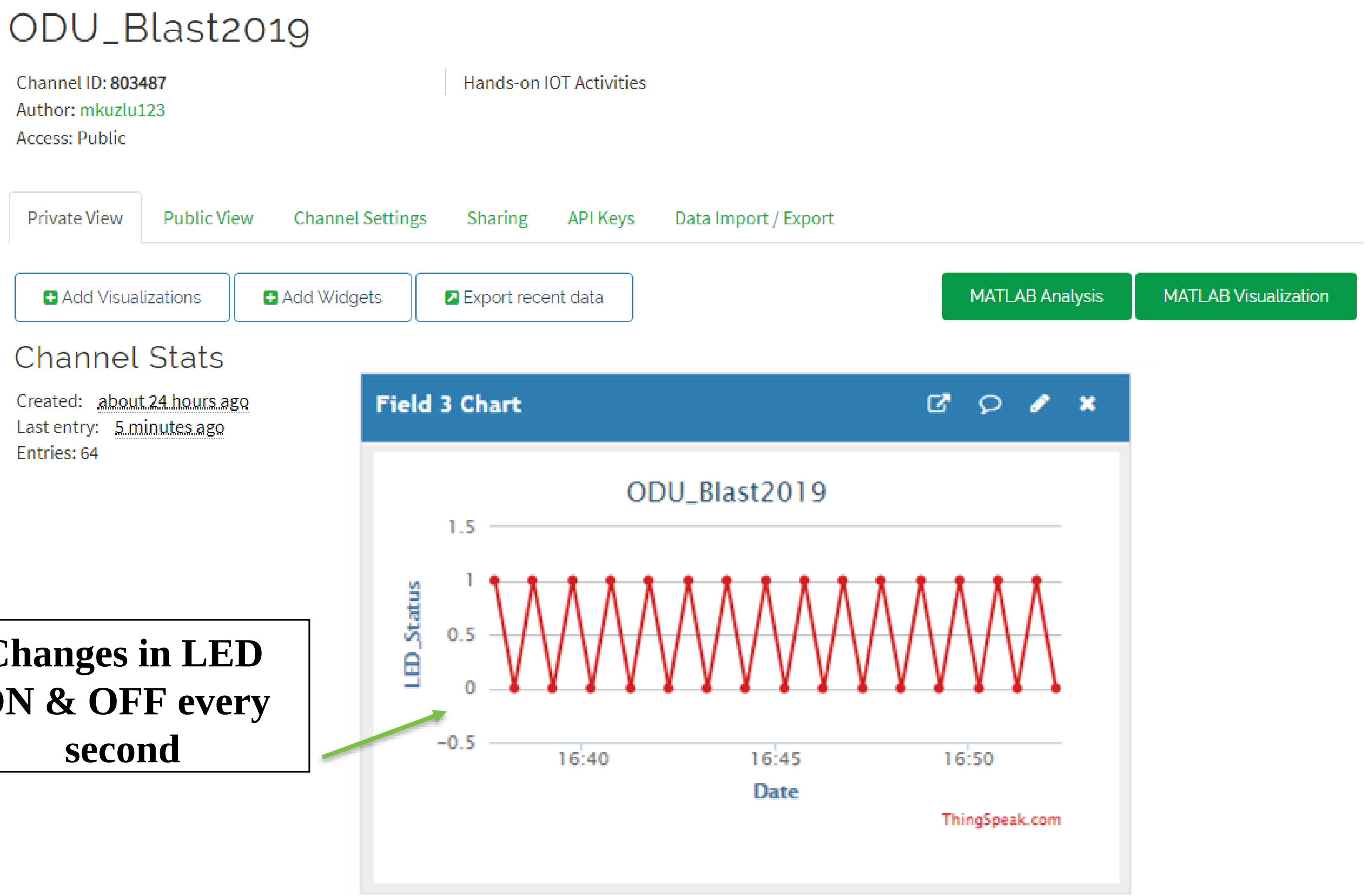
Digital output with Arduino and IoT - III

```
void loop() {
LED_Status = HIGH;
digitalWrite(LED, LED_Status);// turn the LED off.(Note that LOW is the voltage level
but actually
                    //the LED is on; this is because it is acive low on the ESP8266.
Serial.print("LED Status is :");
Serial.print(LED_Status);
ThingSpeak.writeField(myChannelNumber, 3, LED_Status, myWriteAPIKey);
delay(30000); // ThingSpeak will only accept updates every 15 seconds.

LED_Status = LOW;
digitalWrite(LED, LED_Status); // turn the LED on.
Serial.print("LED Status is :");
Serial.print(LED_Status);
ThingSpeak.writeField(myChannelNumber, 3, LED_Status, myWriteAPIKey);
delay(30000); // ThingSpeak will only accept updates every 15 seconds.
}
```

→ Upload

Digital output with Arduino and IoT -IV

- Check ThingSpeak



ODU_Blast2019

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

Field 3 Chart

Changes in LED ON & OFF every second

Hands-on Activity - II

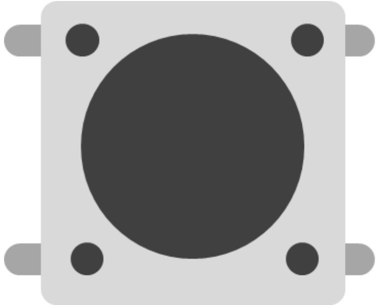
NodeMCU

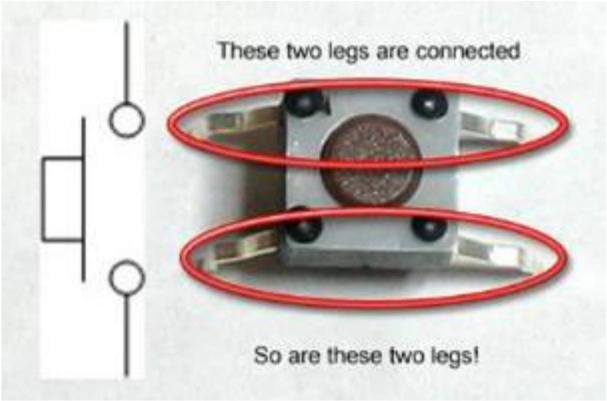
The switch

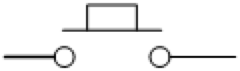
A switch is a simple, digital **sensor**

Switches come in different forms, but all of them in some way **open** or **close** a gap in a wire

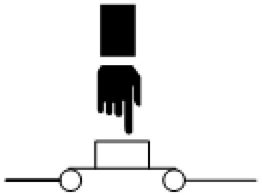
The **pushbutton** switch has four legs for easier mounting, but only two of them are needed







Normally Open
When not pressed



Closed
When pressed

Wiring a switch with Arduino

We will introduce how to blink the on-board LED and how to blink a external LED.

Hardware

- NodeMCU x1
- Push Button x1
- LED x1
- 10 K ohm Resistor x1
- 200 ohm Resistor x1
- Bread Board x1
- PC x1

Software

- Arduino IDE(version 1.6.4+)

Digital input with Arduino

Set up

Push Button connections :

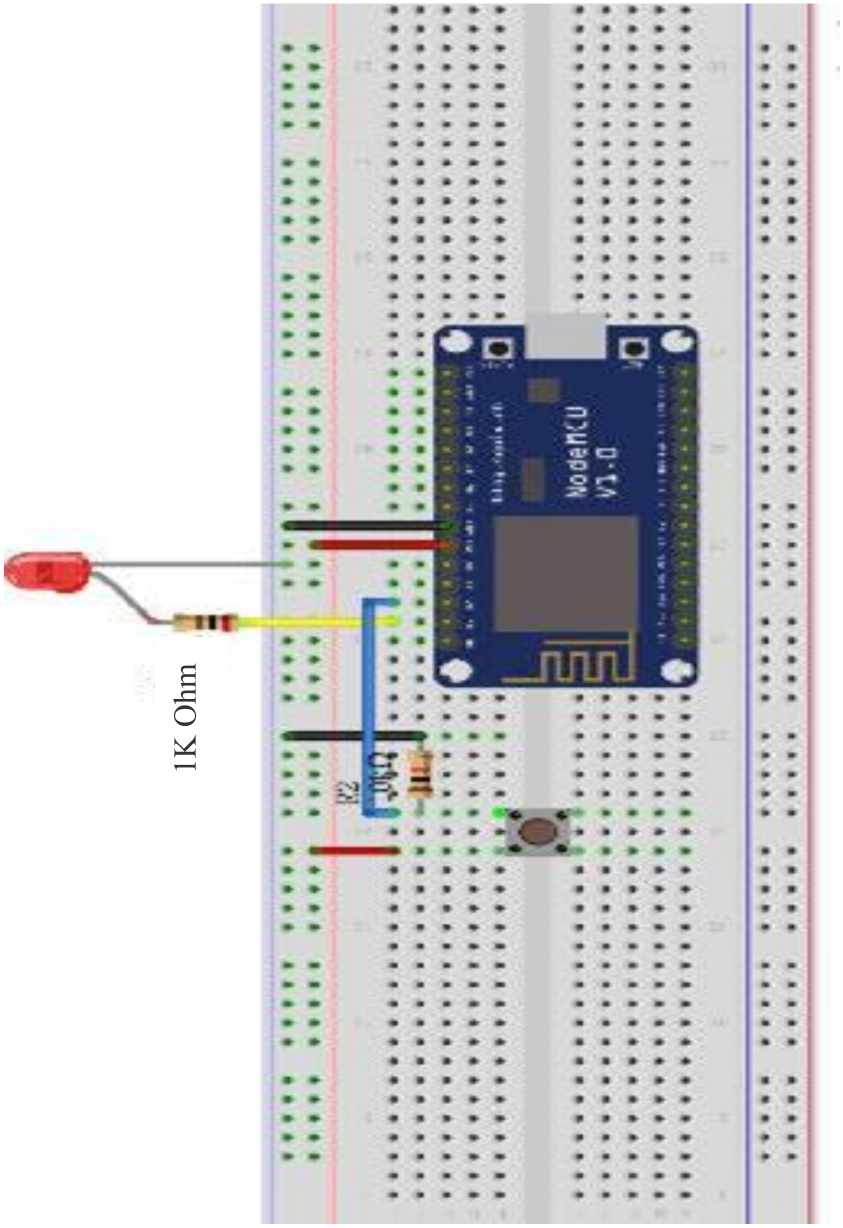
- ➔ The first pin goes from one leg of the pushbutton through a pull-up resistor(here 10K Ohms) to Ground (**GND**).

- ➔ The **second pin** goes from the corresponding leg of the pushbutton to the 3v supply pin.

- ➔ The **third pin** connects to a Digital I/O pin (here pin **D2**) which reads the button's state.

***LED connections**:*

- ➔ LED **Anode** is connected to Digital I/O pin (here pin **D1**) and Cathode to ground **(GND)** pin.



Digital input with Arduino

Code

- ➔ Copy the following code to the IDE

```
#define LED 5 // D1(gpio5)
#define BUTTON 4 //D2(gpio4)

int buttonState=0;
int switchState = 0; // actual read value from pin4
int oldSwitchState = 0; // last read value from pin4
int lightsOn = 0; // is the switch on = 1 or off = 0

void setup() {
pinMode(BUTTON, INPUT); // push button
pinMode(LED, OUTPUT); // anything you want to control using a switch e.g. a Led
}

void loop() {
switchState = digitalRead(BUTTON); // read the pushButton State
if (switchState != oldSwitchState) // catch change
{
oldSwitchState = switchState;
if (switchState == HIGH)
{
// toggle
lightsOn = !lightsOn;
}
}
if(lightsOn)
{
digitalWrite(LED, HIGH); // set the LED on
} else {
digitalWrite(LED, LOW); // set the LED off
}
}
```

➔ Upload

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 "ThingSpeak.h"
#include <ESP8266WiFi.h>
const char *ssid =  "AS2L-Room";     // replace with your wifi ssid and wpa2 key
const char *pass =  "as2l214c";
const char* server = "api.thingspeak.com";
const char * myWriteAPIKey = "3M0SBN71PI6UD1A4"; //  Enter your Write API key from
ThingSpeak

unsigned long myChannelNumber = 803487;

WiFiClient client;

#define LED 5 // D1(gpio5)
#define BUTTON 4 //D2(gpio4)

int buttonState=0;

int switchState = 0; // actual read value from pin4
int oldSwitchState = 0; // last read value from pin4
int lightsOn = 0; // is the switch on = 1 or off = 0
```

Digital input with Arduino and IoT - II

```
void setup()
{
      Serial.begin(115200);
      delay(10);

       pinMode(BUTTON, INPUT); // push button
       pinMode(LED, OUTPUT); // anything you want to control using a switch e.g. a Led

      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);
}
```

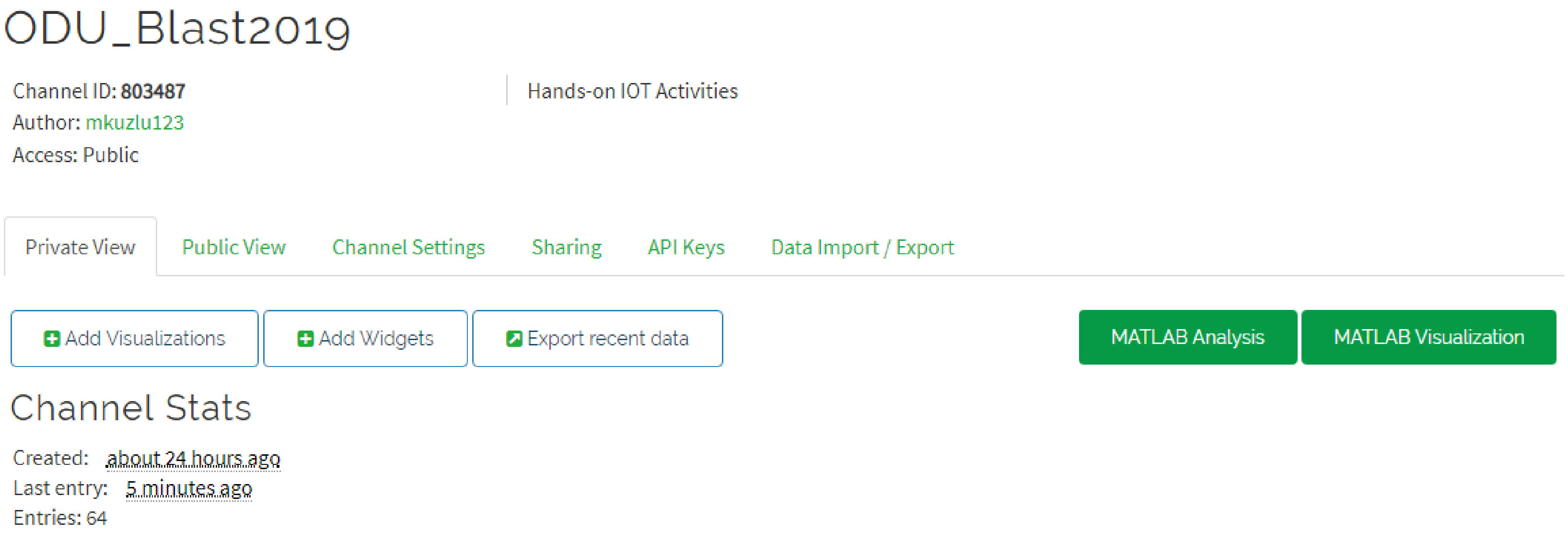
Digital input with Arduino and IoT - III

```
void loop() {
switchState = digitalRead(BUTTON); // read the pushButton State
if (switchState != oldSwitchState) // catch change
{
oldSwitchState = switchState;
if (switchState == HIGH)
{
// toggle
lightsOn = !lightsOn;
}
}
if(lightsOn)
{
digitalWrite(LED, HIGH); // set the LED on
buttonState = HIGH;
} else {
digitalWrite(LED, LOW); // set the LED off
 buttonState = LOW;
}
Serial.print("Button Status is :");
Serial.print(buttonState);
ThingSpeak.writeField(myChannelNumber, 4, buttonState,
myWriteAPIKey);
delay(1000);
}
```

➔ Upload

Digital output with Arduino and IoT -IV

- Check ThingSpeak



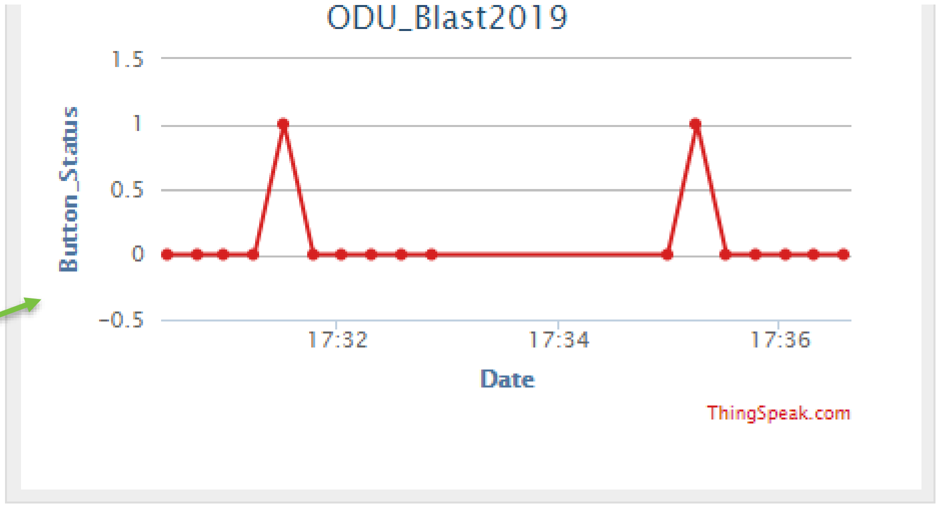
ODU_Blast2019

Channel ID: **803487**
Author: mkuzlu123
Access: Public

Hands-on IOT Activities

Channel Stats

Created: about 24 hours ago
Last entry: 5 minutes ago
Entries: 64



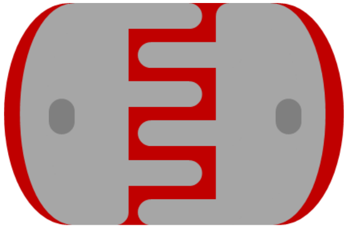
Changes in Button Status

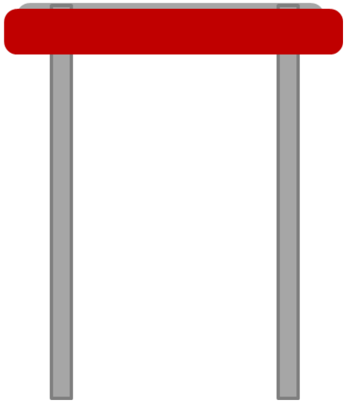
Hands-on Activity - III

NodeMCU

Photoresistor (LDR)

A photoresistor or **LDR** (**l**ight **d**ependent **r**esistor) is a resistor whose resistance depends on light intensity





An LDR can be used as a simple, **analog sensor**

The orientation of an LDR does not matter

Wiring a LED with Arduino

We will introduce how to blink the on-board LED and how to blink a external LED.

Hardware

- NodeMCU
- LDR / photoresistor
- 10k ohm resistor
- Breadboard
- Micro USB cable
- Connecting Wires

Software

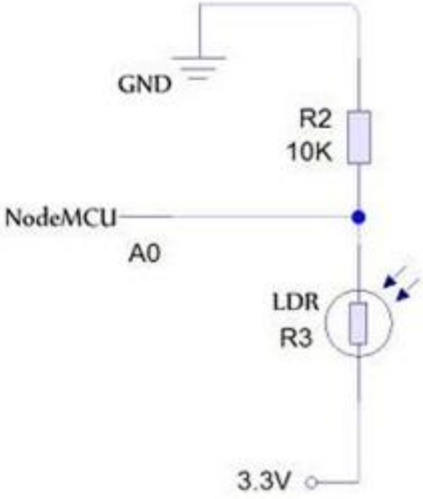
- Arduino IDE(version 1.6.4+)

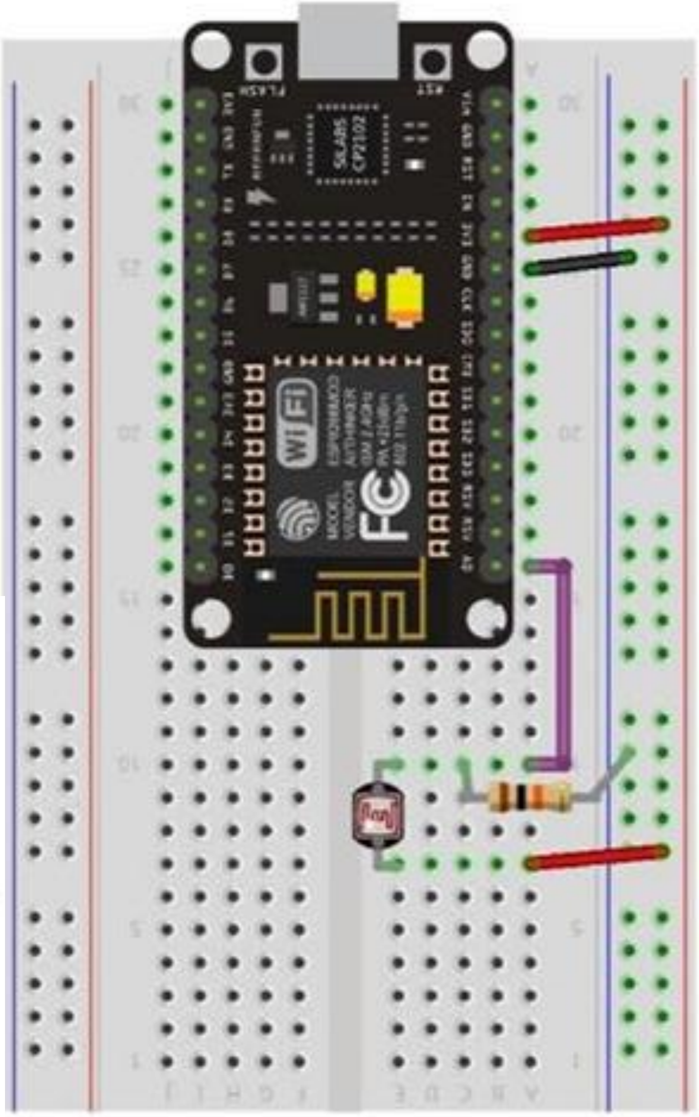
Wiring a LED with Arduino

Set up

The LDR output is actually analog in nature, so it gets connected to the A0 pin of the NodeMCU.

Note: this setup is a *voltage-divider*, *as* the total voltage is divided between LDR and resistor to keep 0V < **A0** < 2.5V





Analog input with Arduino

Code

- ➔ Copy the following code to the IDE

```
void setup()
{
      Serial.begin(115200);
      delay(10);
}

void loop() {

 int sensorValue = analogRead(A0);   // read the input on analog pin 0

 float voltage = sensorValue * (5.0 / 1023.0);   // Convert the analog
reading (which goes from 0 - 1023) to a voltage (0 - 5V)

 Serial.println(voltage);   // print out the value you read
}
```

➔ Upload

Analog input with Arduino and IoT - I

Now we are going to connect to IoT

- ➔ Copy the following code to a new IDE sketch

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

const char *ssid =  "AS2L-Room";     // replace with your wifi ssid and wpa2 key
const char *pass =  "as2l214c";
const char* server = "api.thingspeak.com";
const char * myWriteAPIKey = "3M0SBN71PI6UD1A4"; //  Enter your Write API key from ThingSpeak

unsigned long myChannelNumber = 803487;

WiFiClient client;
```

Analog input with Arduino and IoT - II

```
void setup()
{
    Serial.begin(115200);
    delay(10);

    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);
}
```

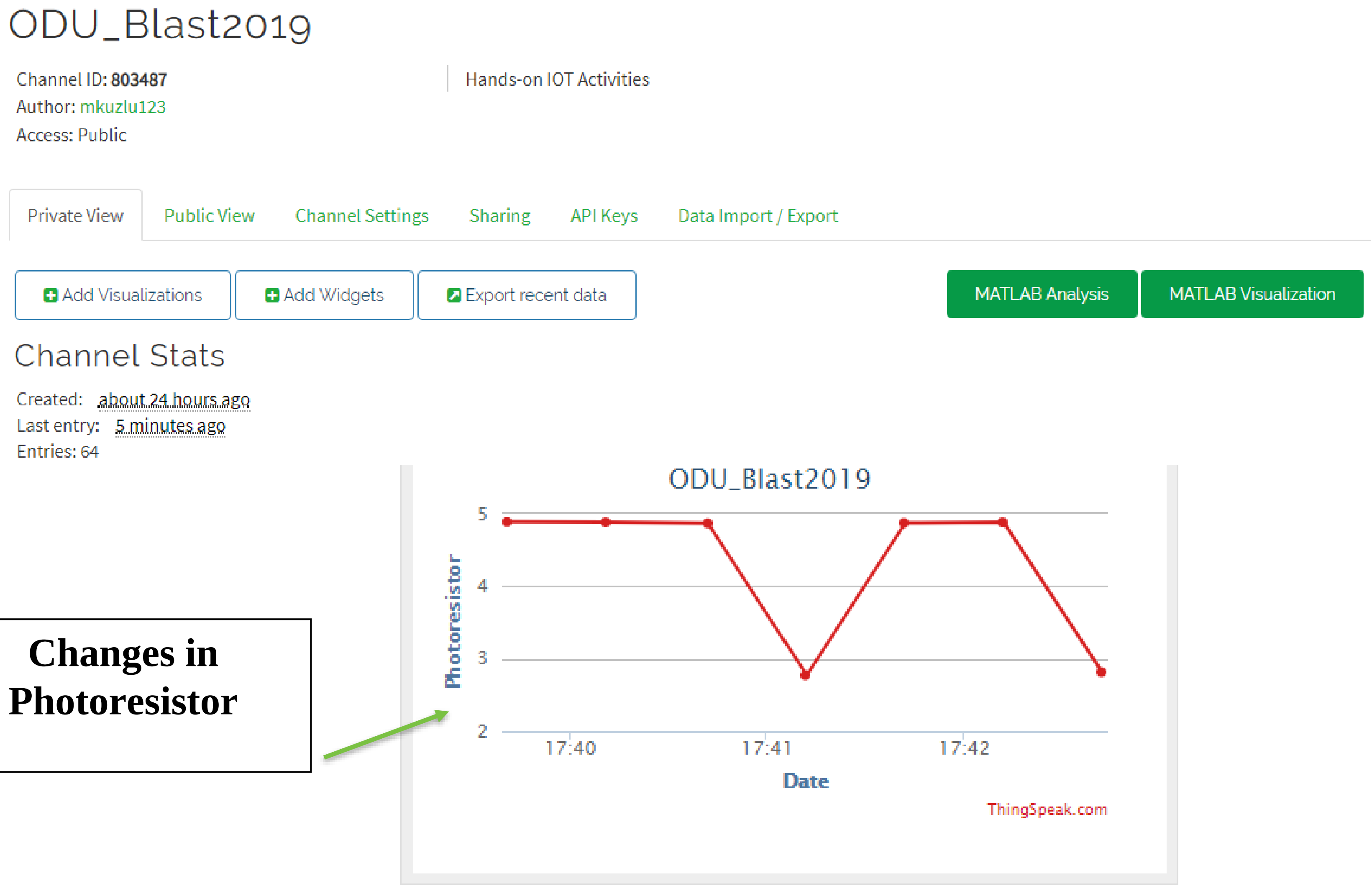
Analog input with Arduino and IoT - III

```
void loop() {

 int sensorValue = analogRead(A0);   // read the input on analog pin 0

 float voltage = sensorValue * (5.0 / 1023.0);   // Convert the analog reading (which goes from
0 - 1023) to a voltage (0 - 5V)

 Serial.println(voltage);   // print out the value you read
 Serial.print("Photoresistir value is :");
 Serial.print(voltage);
 ThingSpeak.writeField(myChannelNumber, 5, voltage, myWriteAPIKey);
 delay(30000); // ThingSpeak will only accept updates every 15 seconds.
}
```

→ Upload

Digital output with Arduino and IoT -IV

- → Check ThingSpeak



ODU_Blast2019

Channel ID: **803487**
Author: mkuzlu123
Access: Public

Hands-on IOT Activities

Channel Stats

Created: about 24 hours ago
Last entry: 5 minutes ago
Entries: 64

Changes in Photoresistor

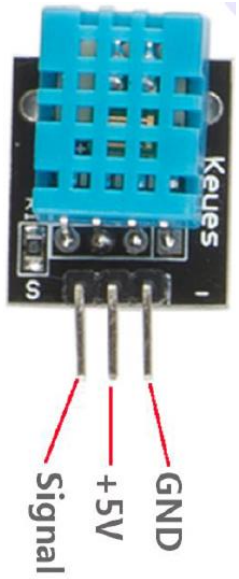
Hands-on Activity - IV

NodeMCU

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+)

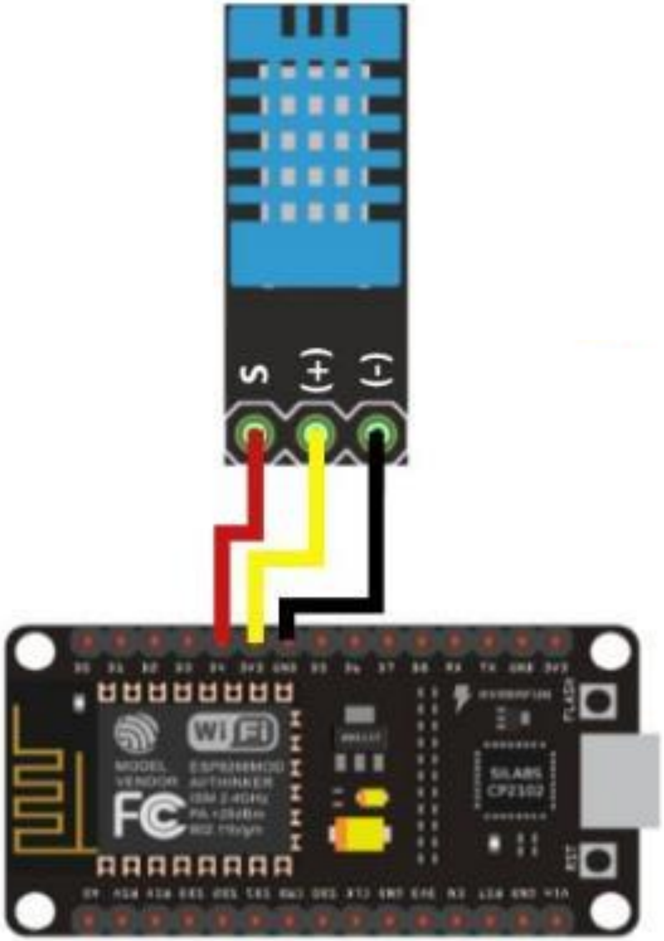
Digital input with Arduino

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 1** of the DHT11 goes into **+3.3v** of the NodeMCU.
- ➔ **Pin 2** of the DHT11 goes into Digital Pin **D4** of the NodeMCU.
- ➔ **Pin 3** of the DHT11 goes into Ground Pin (**GND**) of the NodeMCU.



Digital input with Arduino

Code

- ➔ Copy the following code to the IDE

```
#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
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 serialbegin number in your code is the same.

➔ Upload

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 =  "AS2L-Room";     // replace with your wifi ssid and wpa2 key
const char *pass =  "as2l214c";
const char* server = "api.thingspeak.com";
const char * myWriteAPIKey = "3M0SBN71PI6UD1A4";  //  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;
```

Digital input with Arduino and IoT - II

```
void setup()
{
    Serial.begin(115200);
    delay(10);
    //dht.begin();
    dht.setup(DHTPIN, DHTesp::DHT11); // data pin 2
    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);
}
```

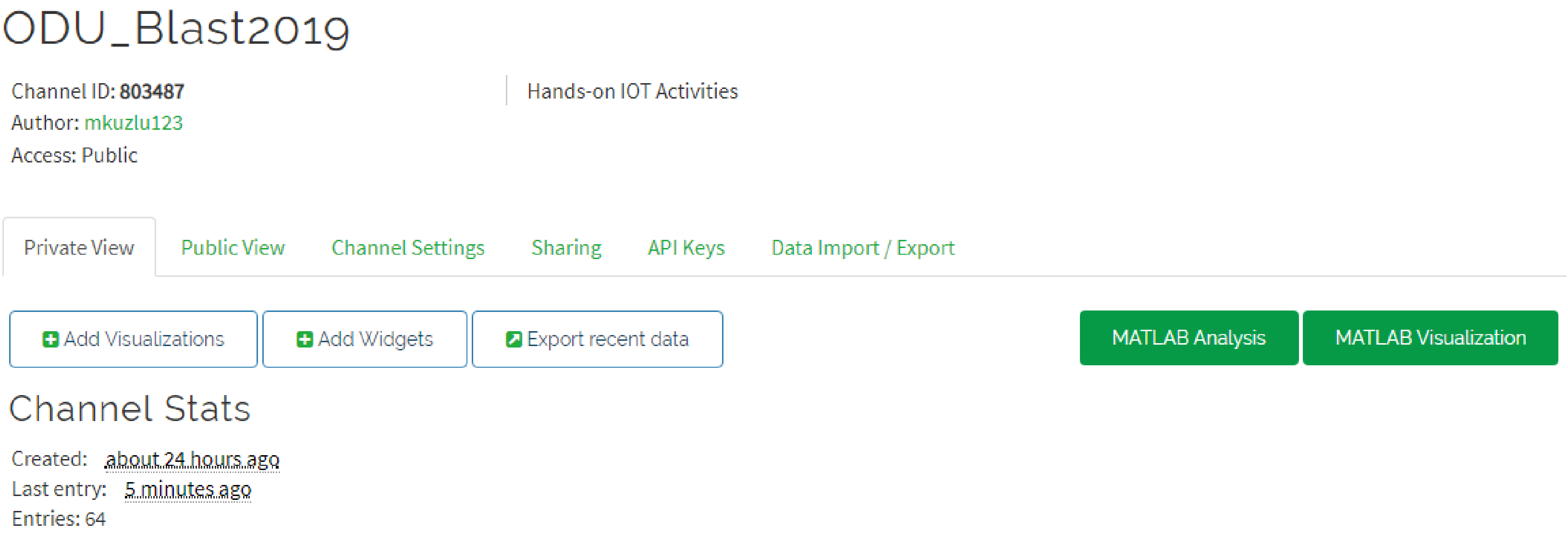
Digital input with Arduino and IoT - III

```
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;
 }
 delay(30000); // ThingSpeak will only accept updates every 15 seconds.
}
```

➔ Upload

Digital output with Arduino and IoT -IV

- Check ThingSpeak



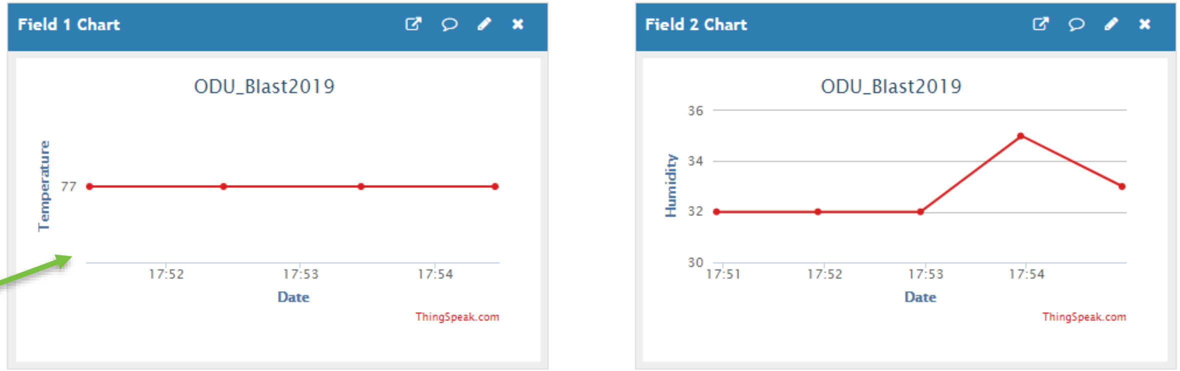
ODU_Blast2019

Channel ID: **803487**
Author: mkuzlu123
Access: Public

Hands-on IOT Activities

Channel Stats

Created: about 24 hours ago
Last entry: 5 minutes ago
Entries: 64



Changes in Temperature and Humidity

Hands-on Activity - V

NodeMCU

Interfacing a 4x4 Keypad with Arduino

We will introduce how to use a 4x4 matrix keypad with the NodeMCU ESP8266. We will then monitor the inputs of the keypad through IoT by using ThingSpeak.

Hardware

- NodeMCU x 1
- Breadboard x 1
- Micro USB cable x 1
- PC x 1
- Membrane Switch Module Keypad
- Male to Male Wires x 8

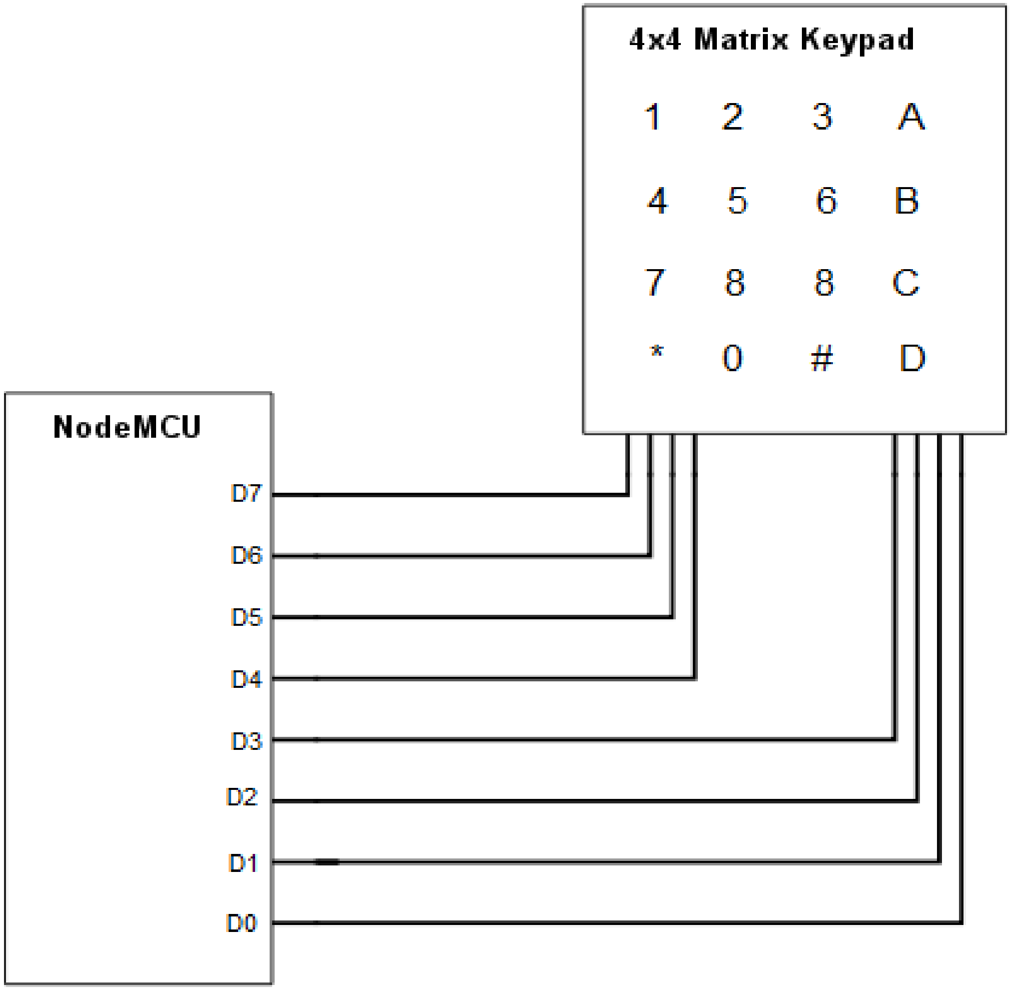
Software

- Arduino IDE(version 1.6.4+)

Wiring a 4x4 Keypad with Arduino

Set up

- With the screen of Membrane Switch Module facing you, start from left to right when wiring each of the eight male to male wires
- The first wire connects to pin (D7)
- The second wire connects to pin (D6)
- The third wire connects to pin (D5)
- The fourth wire connects to pin (D4)
- The fifth wire connects to pin (D3)
- The sixth wire connects to pin (D2)
- The seventh wire connects to pin (D1)
- The eighth wire connects to pin (D0)

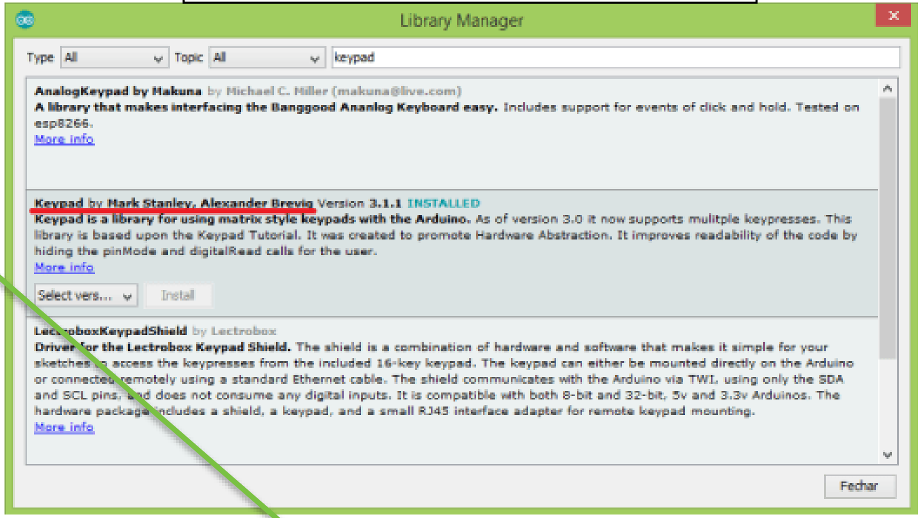


Digital output with Arduino - I

Code

```
#include <Keypad.h>

const byte n_rows = 4;
const byte n_cols = 4;

char keys[n_rows][n_cols] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};

byte colPins[n_rows] = {D3, D2, D1, D0};
byte rowPins[n_cols] = {D7, D6, D5, D4};
```

Add the keypad.h library



Library Manager

Type All Topic All keypad

AnalogKeypad by Makuna by Michael C. Miller (makuna@live.com)
A library that makes interfacing the Banggood Ananlog Keyboard easy. Includes support for events of click and hold. Tested on esp8266.
More info

Keypad by **Mark Stanley, Alexander Brevig** Version **3.1.1** INSTALLED
Keypad is a library for using matrix style keypads with the Arduino. As of version 3.0 it now supports mulitple keypresses. This library is based upon the Keypad Tutorial. It was created to promote Hardware Abstraction. It improves readability of the code by hiding the pinMode and digitalRead calls for the user.
More info

Select vers... Install

LectroboxKeypadShield by Lectrobox
Driver for the Lectrobox Keypad Shield. The shield is a combination of hardware and software that makes it simple for your sketches to access the keypresses from the included 16-key keypad. The keypad can either be mounted directly on the Arduino or connected remotely using a standard Ethernet cable. The shield communicates with the Arduino via TWI, using only the SDA and SCL pins, and does not consume any digital inputs. It is compatible with both 8-bit and 32-bit, 5v and 3.3v Arduinos. The hardware package includes a shield, a keypad, and a small RJ45 interface adapter for remote keypad mounting.
More info

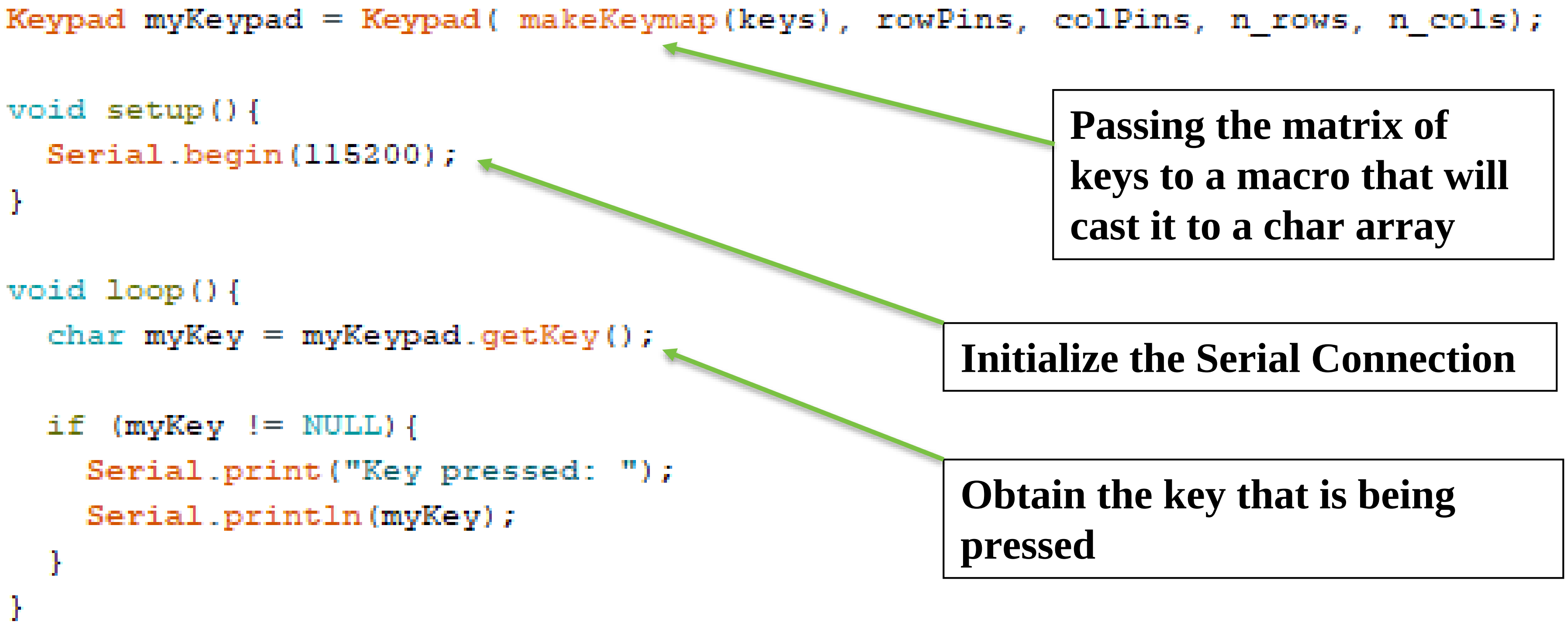
Fechar

Global Variables

Keypad Mappings

Array Declaration

Digital output with Arduino - II



```
Keypad myKeypad = Keypad( makeKeymap(keys), rowPins, colPins, n_rows, n_cols);

void setup(){
  Serial.begin(115200);
}

void loop(){
  char myKey = myKeypad.getKey();

  if (myKey != NULL){
    Serial.print("Key pressed: ");
    Serial.println(myKey);
  }
}
```

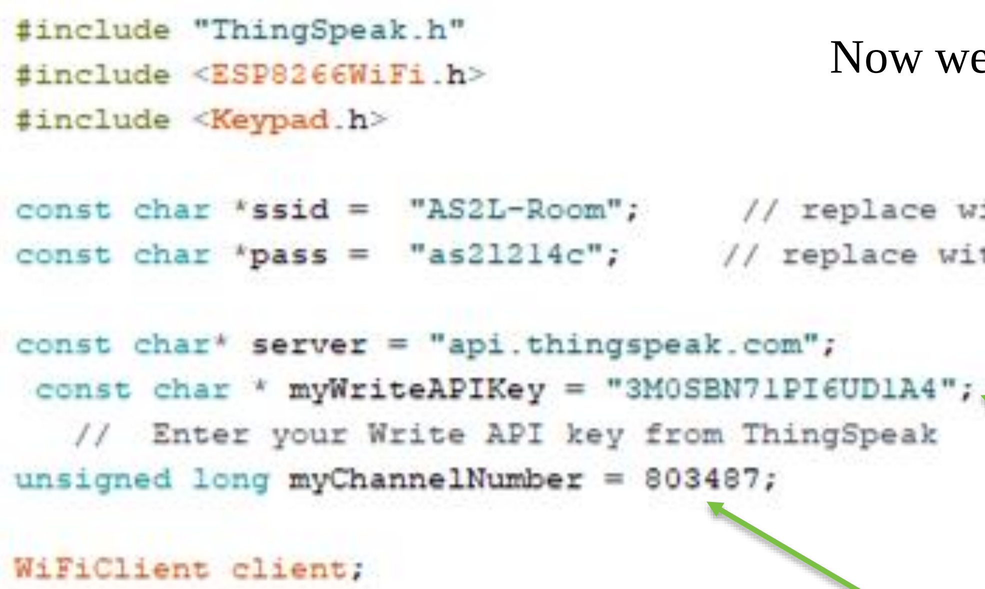
Passing the matrix of keys to a macro that will cast it to a char array

Initialize the Serial Connection

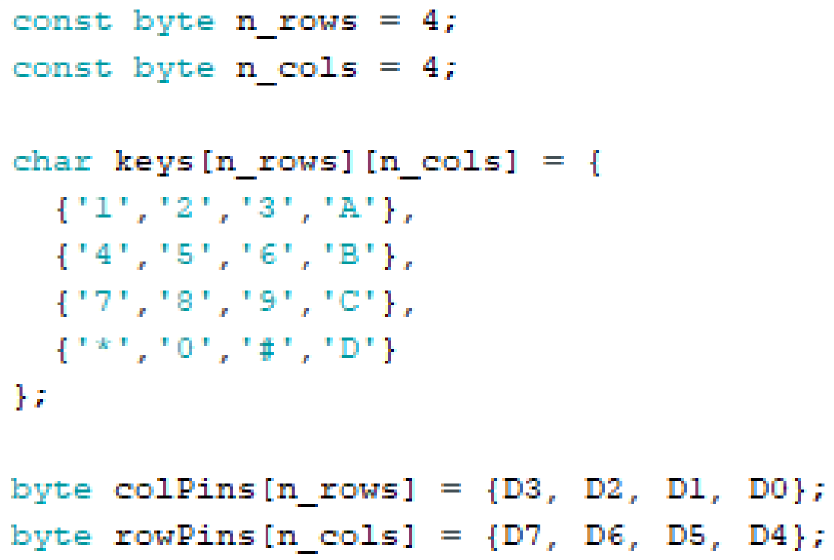
Obtain the key that is being pressed

Digital output with Arduino and IoT - I

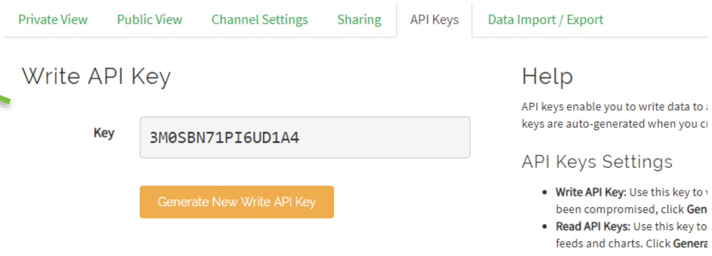
Now we are going to connect to IoT



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

const char *ssid =  "AS2L-Room";     // replace with your wifi ssid
const char *pass =  "as21214c";     // replace with your wifi password

const char* server = "api.thingspeak.com";
 const char * myWriteAPIKey = "3M0SBN71PI6UD1A4";
   //  Enter your Write API key from ThingSpeak
unsigned long myChannelNumber = 803487;

WiFiClient client;
```



```
const byte n_rows = 4;
const byte n_cols = 4;

char keys[n_rows][n_cols] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};

byte colPins[n_rows] = {D3, D2, D1, D0};
byte rowPins[n_cols] = {D7, D6, D5, D4};
```



Write API Key

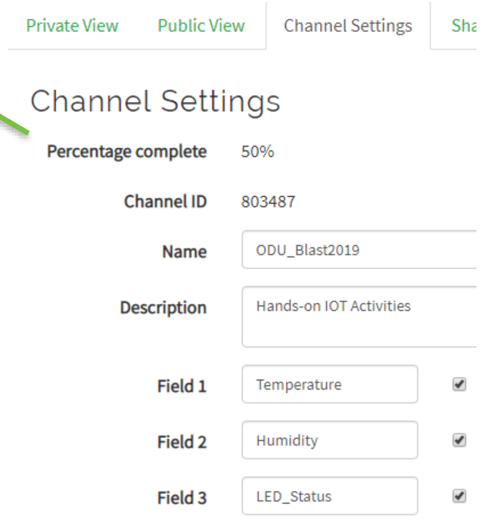
Key: 3M0SBN71PI6UD1A4

Help

API keys enable you to write data to
keys are auto-generated when you c

API Keys Settings

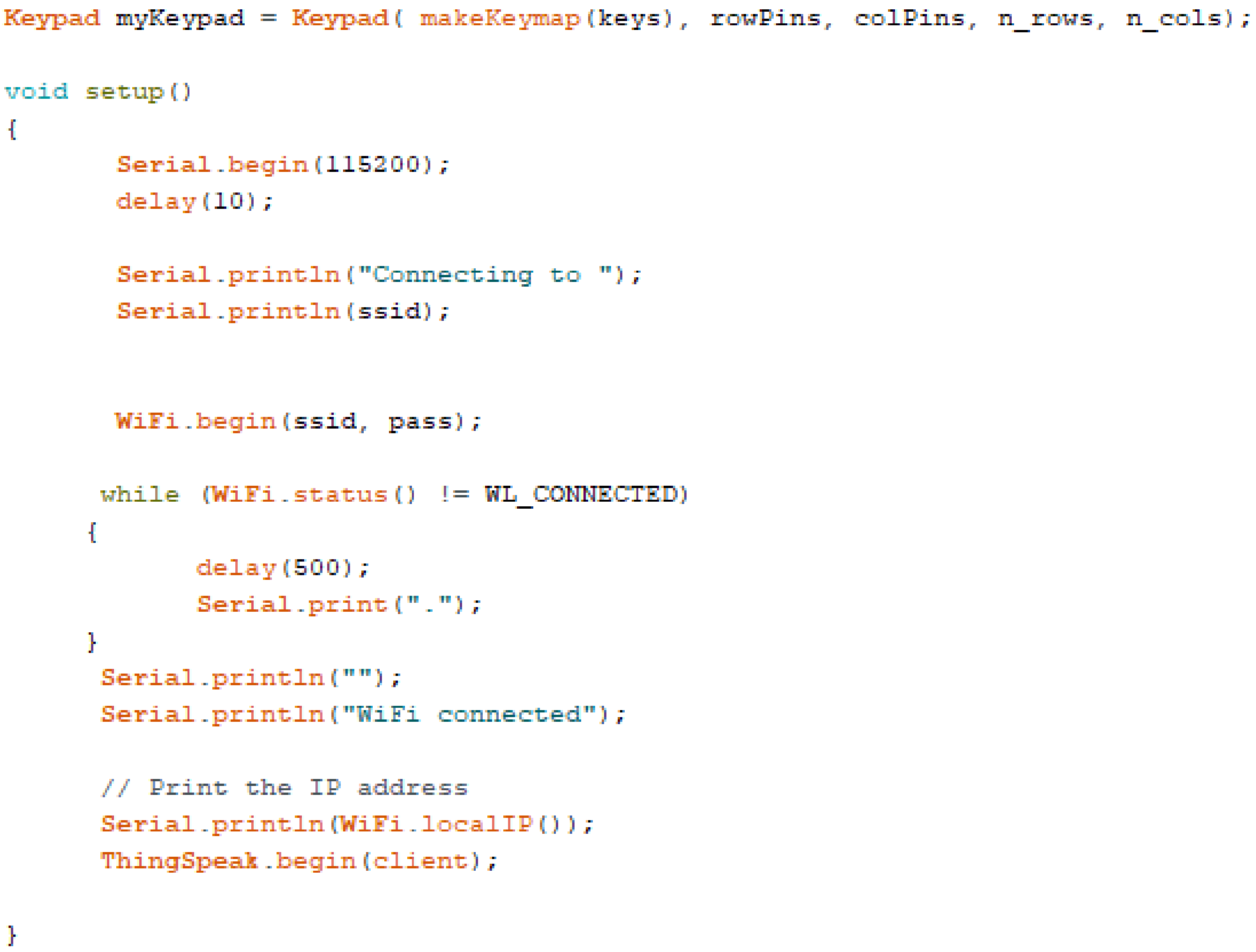
- **Write API Key:** Use this key to
been compromised, click Gen
- **Read API Keys:** Use this key to
feeds and charts. Click Genera



Channel Settings

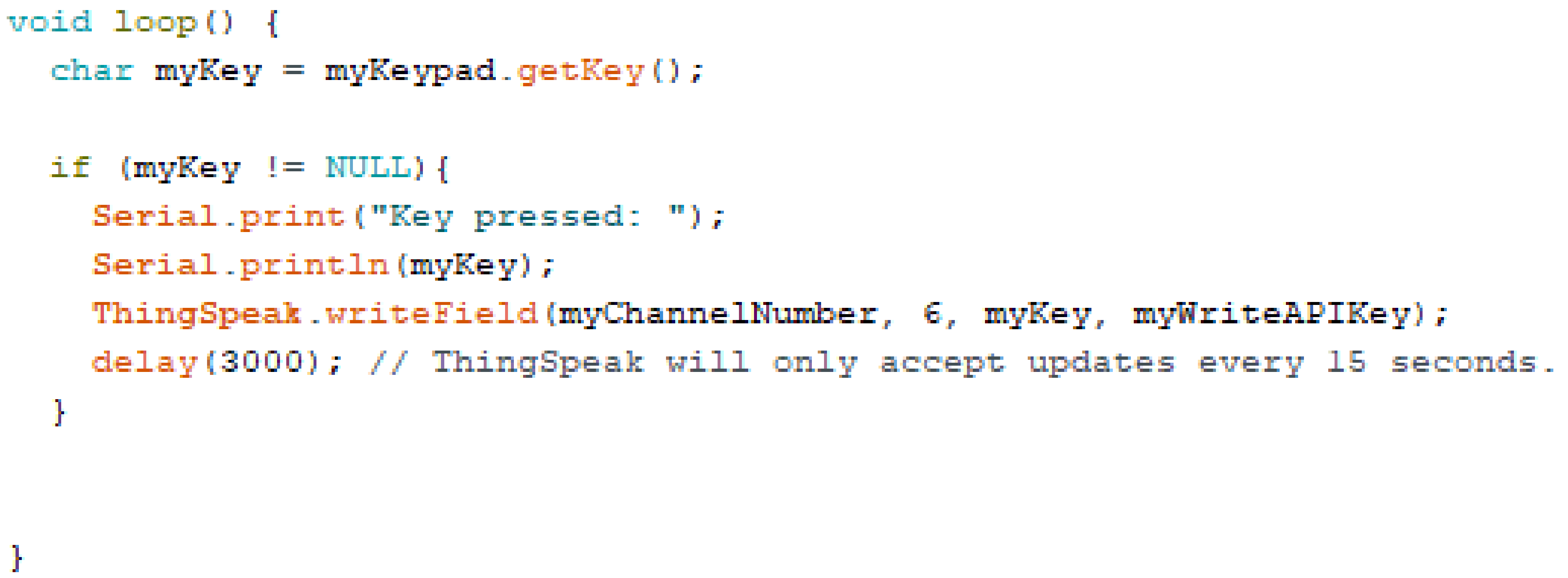
Percentage complete	50%
Channel ID	803487
Name	ODU_Blast2019
Description	Hands-on IOT Activities
Field 1	Temperature ☑
Field 2	Humidity ☑
Field 3	LED_Status ☑

Digital output with Arduino and IoT - II



```
Keypad myKeypad = Keypad( makeKeymap(keys), rowPins, colPins, n_rows, n_cols);

void setup()
{
        Serial.begin(115200);
        delay(10);

        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);

}
```

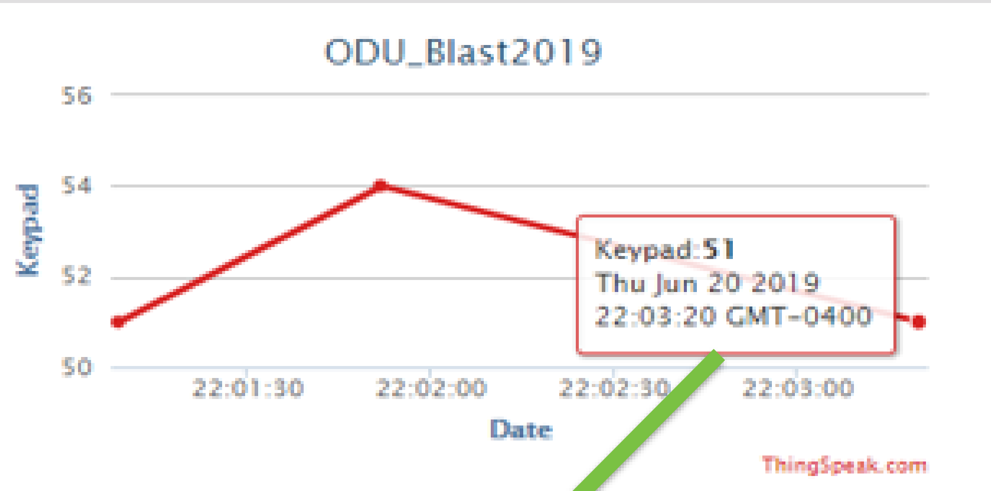
Digital output with Arduino and IoT - III

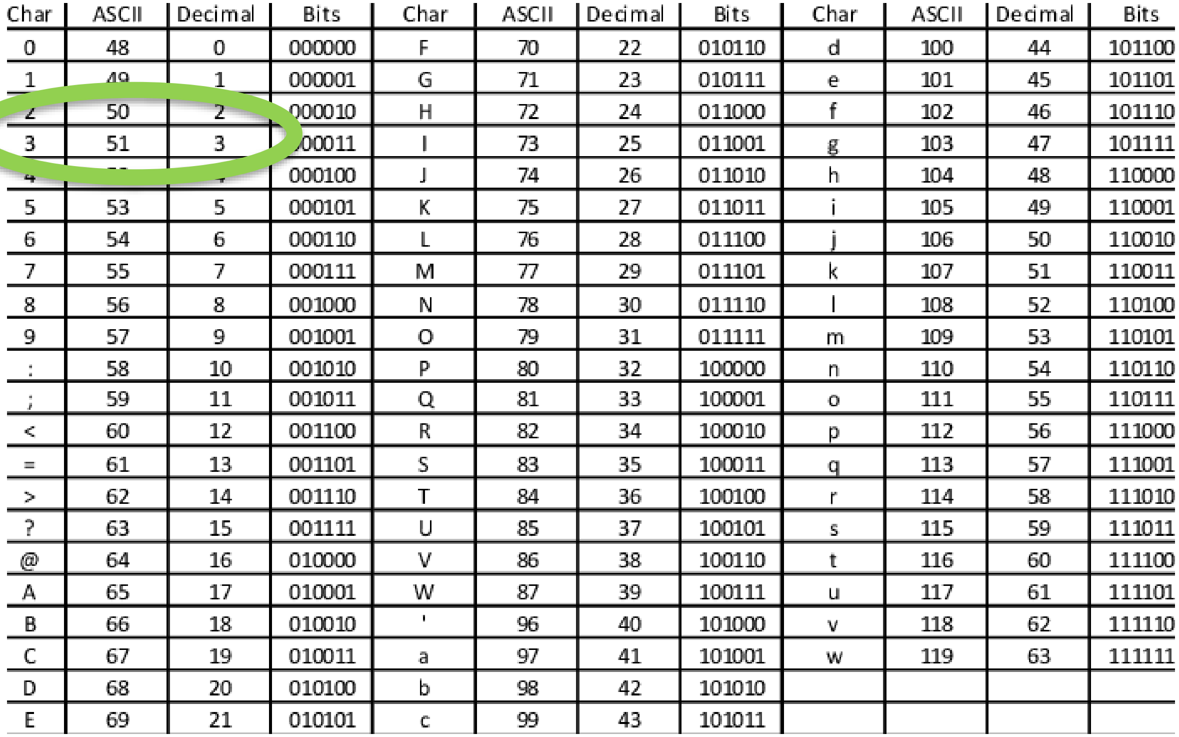


```
void loop() {
  char myKey = myKeypad.getKey();

  if (myKey != NULL){
    Serial.print("Key pressed: ");
    Serial.println(myKey);
    ThingSpeak.writeField(myChannelNumber, 6, myKey, myWriteAPIKey);
    delay(3000); // ThingSpeak will only accept updates every 15 seconds.
  }


}
```

→ Upload

Digital output with Arduino and IoT -IV





Char	ASCII	Decimal	Bits	Char	ASCII	Decimal	Bits	Char	ASCII	Decimal	Bits
0	48	0	000000	F	70	22	010110	d	100	44	101100
1	49	1	000001	G	71	23	010111	e	101	45	101101
2	50	2	000010	H	72	24	011000	f	102	46	101110
3	51	3	000011	I	73	25	011001	g	103	47	101111
4	[illegible]	[illegible]	000100	J	74	26	011010	h	104	48	110000
5	53	5	000101	K	75	27	011011	i	105	49	110001
6	54	6	000110	L	76	28	011100	j	106	50	110010
7	55	7	000111	M	77	29	011101	k	107	51	110011
8	56	8	001000	N	78	30	011110	l	108	52	110100
9	57	9	001001	O	79	31	011111	m	109	53	110101
:	58	10	001010	P	80	32	100000	n	110	54	110110
;	59	11	001011	Q	81	33	100001	o	111	55	110111
<	60	12	001100	R	82	34	100010	p	112	56	111000
=	61	13	001101	S	83	35	100011	q	113	57	111001
>	62	14	001110	T	84	36	100100	r	114	58	111010
?	63	15	001111	U	85	37	100101	s	115	59	111011
@	64	16	010000	V	86	38	100110	t	116	60	111100
A	65	17	010001	W	87	39	100111	u	117	61	111101
B	66	18	010010	'	96	40	101000	v	118	62	111110
C	67	19	010011	a	97	41	101001	w	119	63	111111
D	68	20	010100	b	98	42	101010				
E	69	21	010101	c	99	43	101011				

- **The number three was pressed on the keypad and the last entry recorded on ThingSpeak was 51. If you refer to the ASCII table shown to the right, the number 51 represents the decimal number 3.**

More

- Google & Youtube - search for projects, solutions to occurring problems and data sheets for components
- http://www.blynk.cc/ - Homepage of the Blynk software, getting started, community forums
- http://www.esp8266.com/ - Everything on ESP8266, wiki
- https://www.adafruit.com/ - Learning materials, guides, example projects, forums, store
- https://github.com/ - Largest code host, lots of projects and sample code
- http://allaboutee.com/ - good ESP8266 tutorials
- https://nurdspace.nl/ESP8266/ - ESP8266 info and basic list of AT commands

Dr. Murat Kuzlu

mkuzlu@odu.edu

Backup Slides

Hands-on Activity - I

MEGA2560

Wiring a RGB LED with Arduino

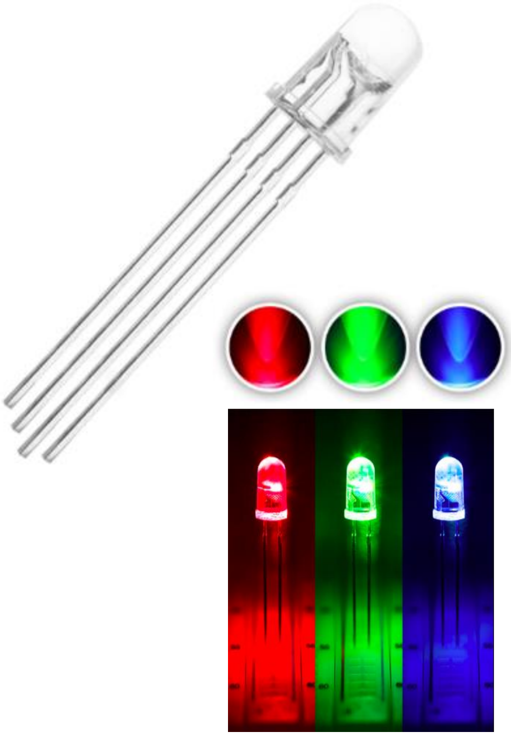
We will introduce how to use a RGB LED with the MEGA2560

Hardware

- (1) x Elegoo Mega 2560 R3
- (1) x 830 Tie PointsBreadboard
- (4) x M-M wires (Male to Male jumperwires)
- (1) x RGB LED
- (3) x 220 ohm resistors

Software

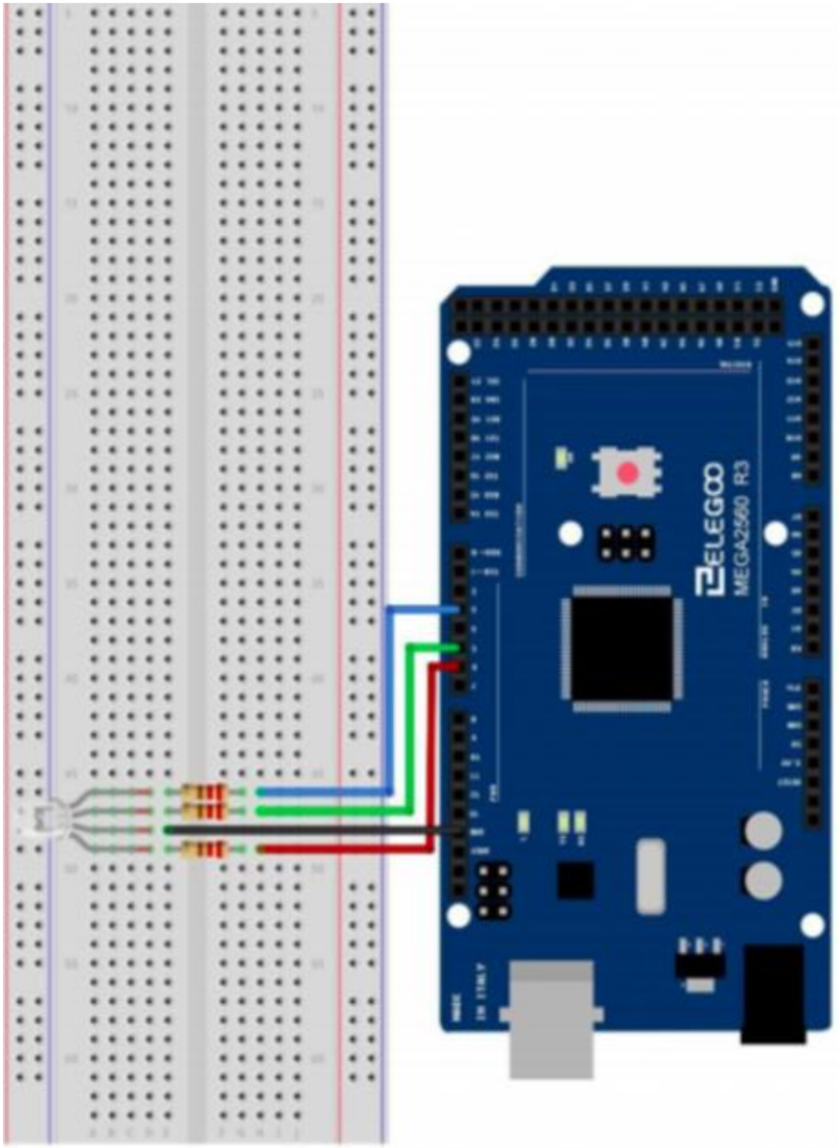
- Arduino IDE(version 1.6.4+)



Wiring a RGB LED with Arduino

Set up

- Connect the longest lead(common cathode) of the RGB to ground.
- Connect each of the other three leads to a 220Ω resistor
- These three positive leads of the LEDs (one red, one green and one blue) are connected to MEGA 2560 output pins using these resistors.



Digital output with Arduino

Code

→ Copy the following code to the IDE

```
// Define Pins
#define BLUE 3
#define GREEN 5
#define RED 6

void setup()
{
pinMode(RED, OUTPUT);
pinMode(GREEN, OUTPUT);
pinMode(BLUE, OUTPUT);
digitalWrite(RED, HIGH);
digitalWrite(GREEN, LOW);
digitalWrite(BLUE, LOW);
}

// define variables
int redValue;
int greenValue;
int blueValue;
```

Digital output with Arduino

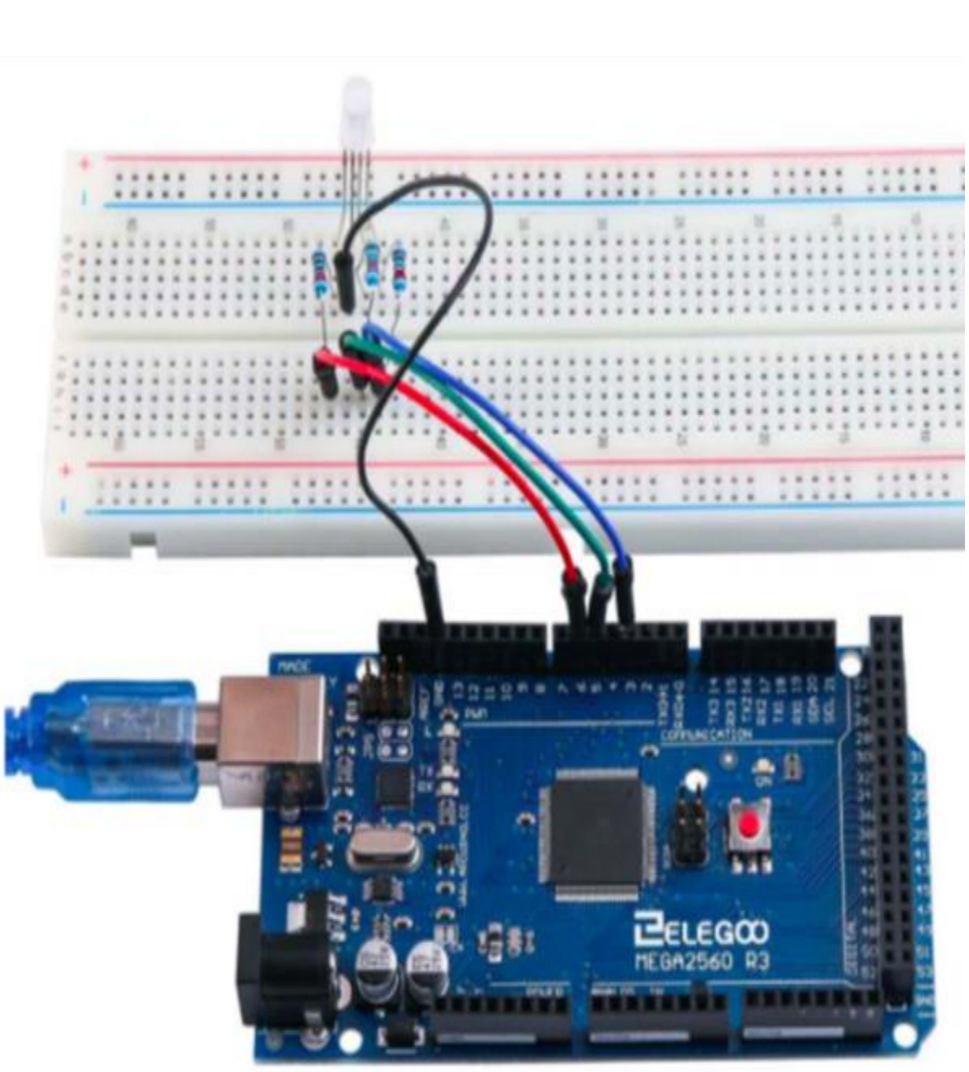
```
// main loop
void loop()
{
#define delayTime 20 // fading time between colors

redValue = 255; // choose a value between 1 and 255 to change
the color.
greenValue = 0;
blueValue = 0;

for(int i = 0; i < 255; i += 1) // fades out red bring green full when
i=255
{
redValue -= 1;
greenValue += 1;
// The following was reversed, counting in the wrong directions
// analogWrite(RED, 255 - redValue);
// analogWrite(GREEN, 255 - greenValue);
analogWrite(RED, redValue);
analogWrite(GREEN, greenValue);
delay(delayTime);
}
```

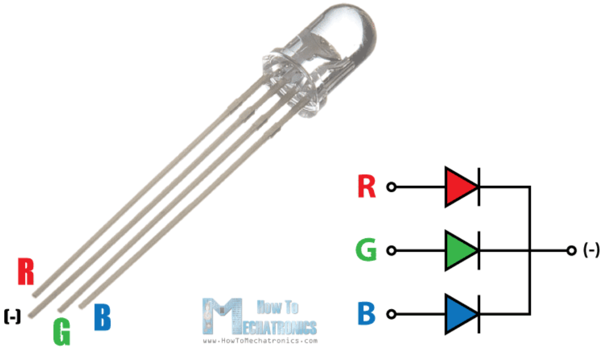
Digital output with Arduino

```
redValue = 0;
greenValue = 255;
blueValue = 0;

for(int i = 0; i < 255; i += 1) // fades out green bring blue full when i=255
{
greenValue -= 1;
blueValue += 1;
analogWrite(GREEN, greenValue);
analogWrite(BLUE, blueValue);
delay(delayTime);
}

redValue = 0;
greenValue = 0;
blueValue = 255;
}
for(int i = 0; i < 255; i += 1) // fades out blue bring red full when i=255
{

blueValue -= 1;
redValue += 1;

analogWrite(BLUE, blueValue);
analogWrite(RED, redValue);
delay(delayTime);
}
```

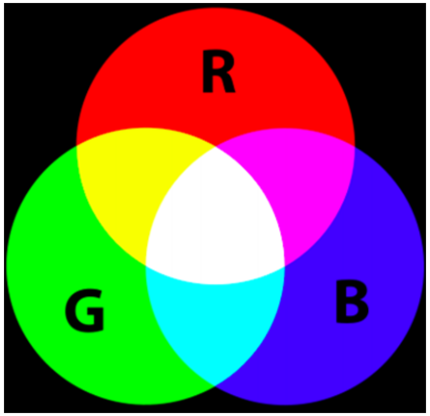
→ Upload

Digital output with Arduino





The RGB LED they are like three regular LEDs in one; Red, Green, Blue.



Combines these colors to produce the orders

Hands-on Activity - II

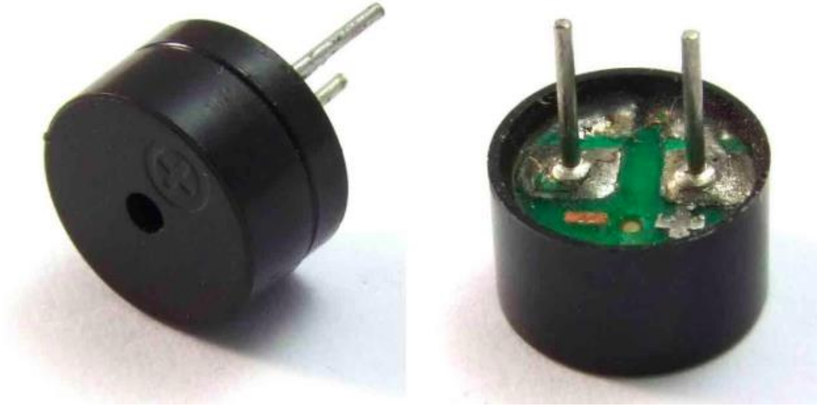
MEGA2560

Wiring a Buzzer with Arduino

We will introduce how to use a Passive Buzzer with the MEGA2560. You will generate eight different sounds, from Alto Do, Re, Mi, Fa, So, La, Si, to Treble Do.

Hardware

- (1) x Elegoo Mega 2560 R3
- (1) x Passive buzzer
- (2) x F-M wires (Female to Male DuPont wires)



Software

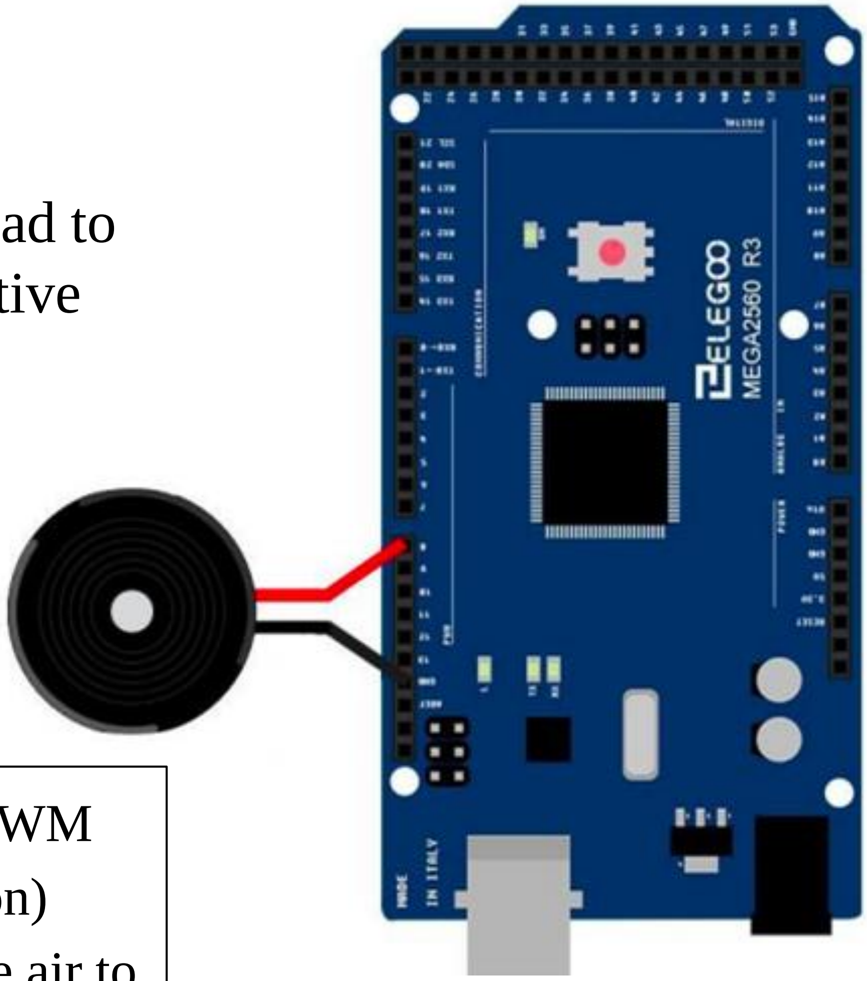
- Arduino IDE(version 1.6.4+)

Wiring a RGB LED with Arduino

Set up

- ➔ Connect the positive lead to the pin 8, and the negative lead to the GND.

That's it!



The passive buzzer uses PWM (Pulse-Width Modulation) generating audio to make the air to vibrate. It works through pulses

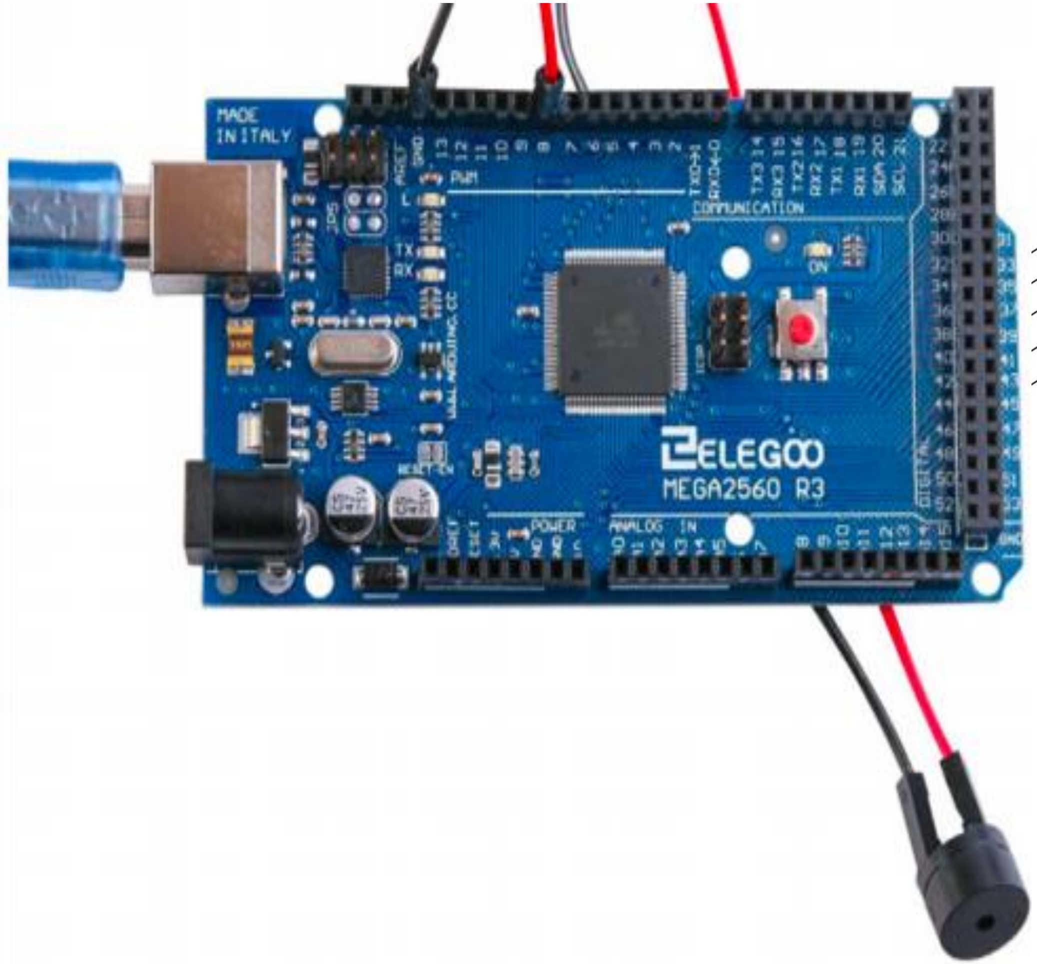
Digital output with Arduino

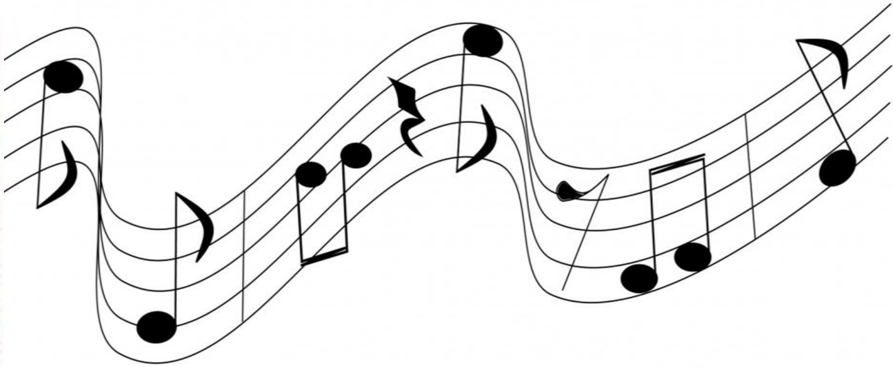
Code

➔ Copy the following code to the IDE

Include Library Pitches.h

```
#include "pitches.h"

// notes in the melody:
int melody[] = {
 NOTE_C5, NOTE_D5, NOTE_E5, NOTE_F5,
NOTE_G5, NOTE_A5, NOTE_B5, NOTE_C6};
int duration = 500;  // 500 miliseconds
void setup() {
}
void loop() {
 for (int thisNote = 0; thisNote < 8; thisNote++) {
   // pin8 output the voice, every scale is 0.5 sencond
   tone(8, melody[thisNote], duration);
   delay(500);
 }
 // restart after two seconds
 delay(2000);
}
```





Frequency of vibrations change sound/note

Example:

Alto Do (523Hz), Re (587Hz),

Mi (659Hz), Fa (698Hz)……