

ESP8266



Hardware Specifications

| Type | Value |
|----------------------|---|
| CPU | Tensilica L106 (32-bit @ 160 MHz) |
| Peripheral Interface | UART/SDIO/SPI/I2C/I2S/IR Remote Control GPIO/ADC/PWM |
| Operating Voltage | 2.5V ~ 3.6V |
| Operating Current | Average value: 80 mA |

Wi-Fi Protocols

- 802.11 b/g/n/e/i support
- Wi-Fi Direct (P2P) support
- WPA/WPA2 PSK, and WPS driver

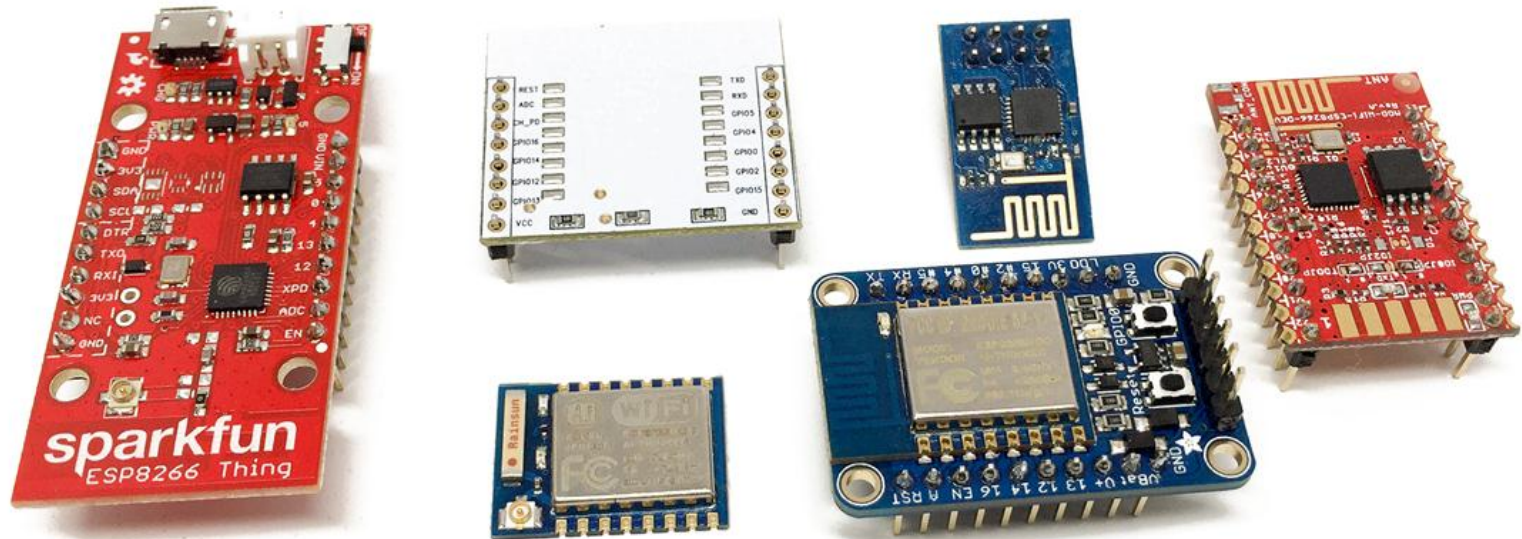
IEEE 802.11

IEEE 802.11 is a set of media access control (**MAC**) and physical layer (**PHY**) specifications for implementing wireless local area network (WLAN) computer communication in the 900 MHz and 2.4, 3.6, 5, and 60 GHz frequency bands.

WPA/WPA2

Wi-Fi Protected Access (WPA) and **Wi-Fi Protected Access II (WPA2)** are two security protocols and security certification programs developed by the Wi-Fi Alliance to secure wireless computer networks.

ESP8266 Flavors





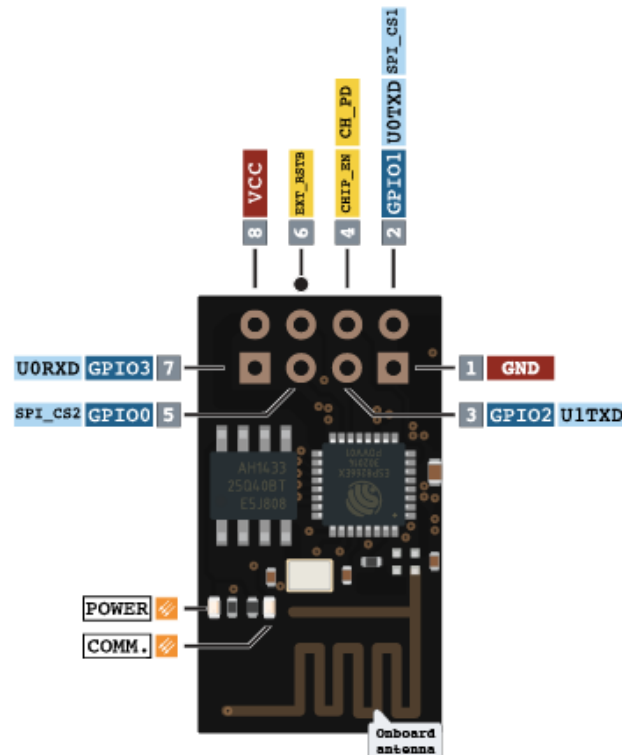
ESP-01

ESP-01

PINOUT

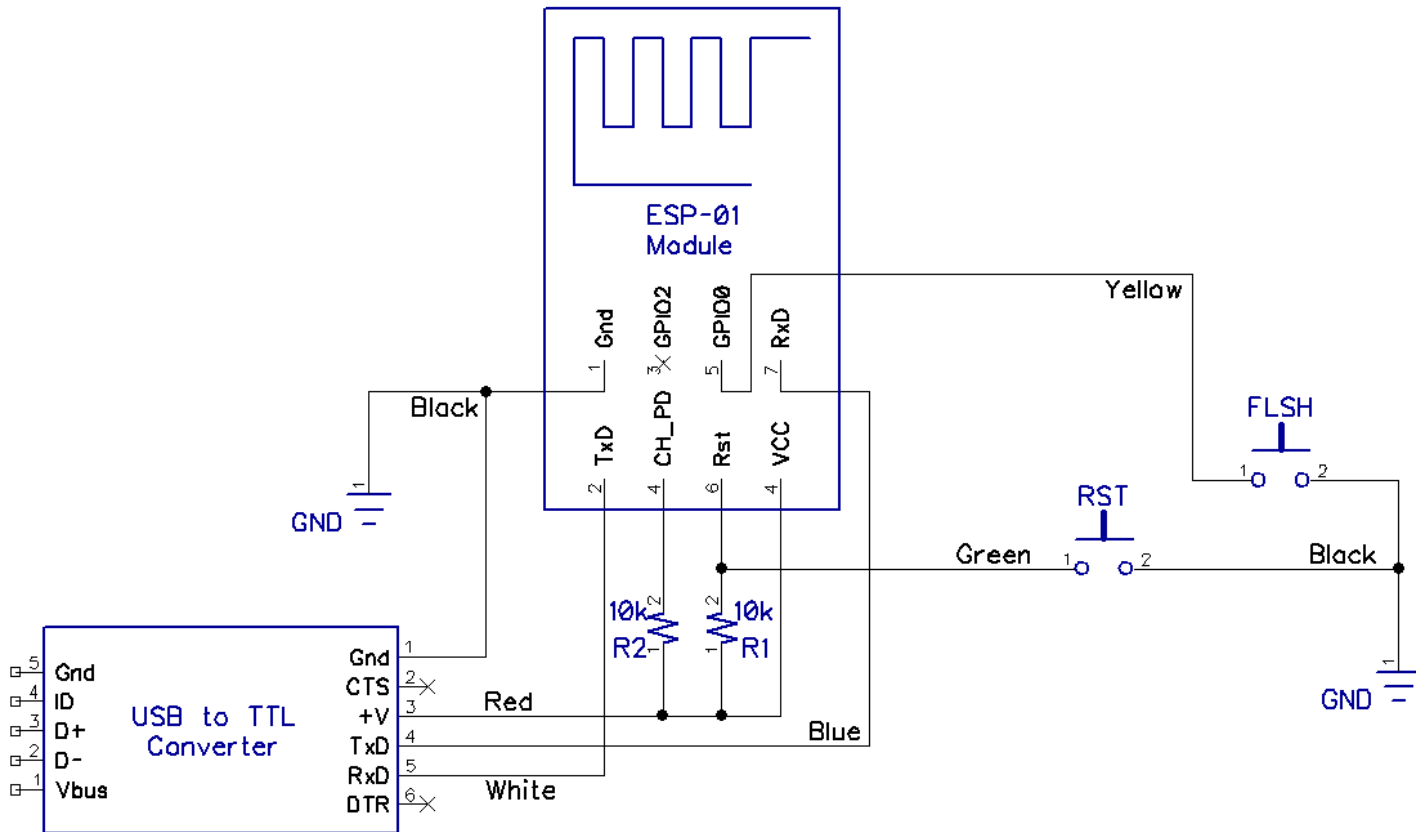
NOTES:

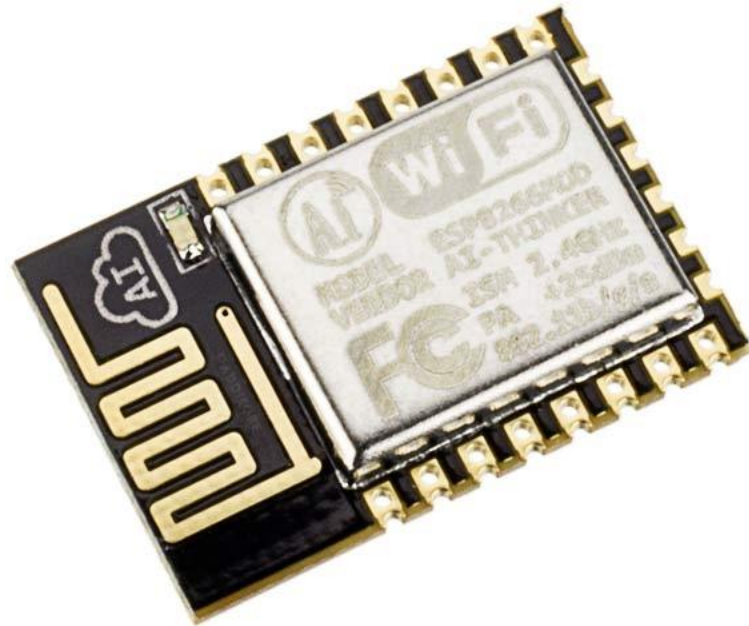
- ▲ Typ. pin current 6mA (Max. 12mA)
- ▲ For sleep mode, connect GPIO16 and EXT_RSTB. On wakeup, GPIO16 will output LOW for system reset.
- ▲ On boot/reset/wakeup, keep GPIO15 LOW and GPIO2 HIGH.



| | |
|---------|-----------------|
| POWER | SP. FUNCTION(S) |
| I/O | COMM. INTERFACE |
| ADC | PIN NUMBER |
| CONTROL | PWM |
| N/C | |

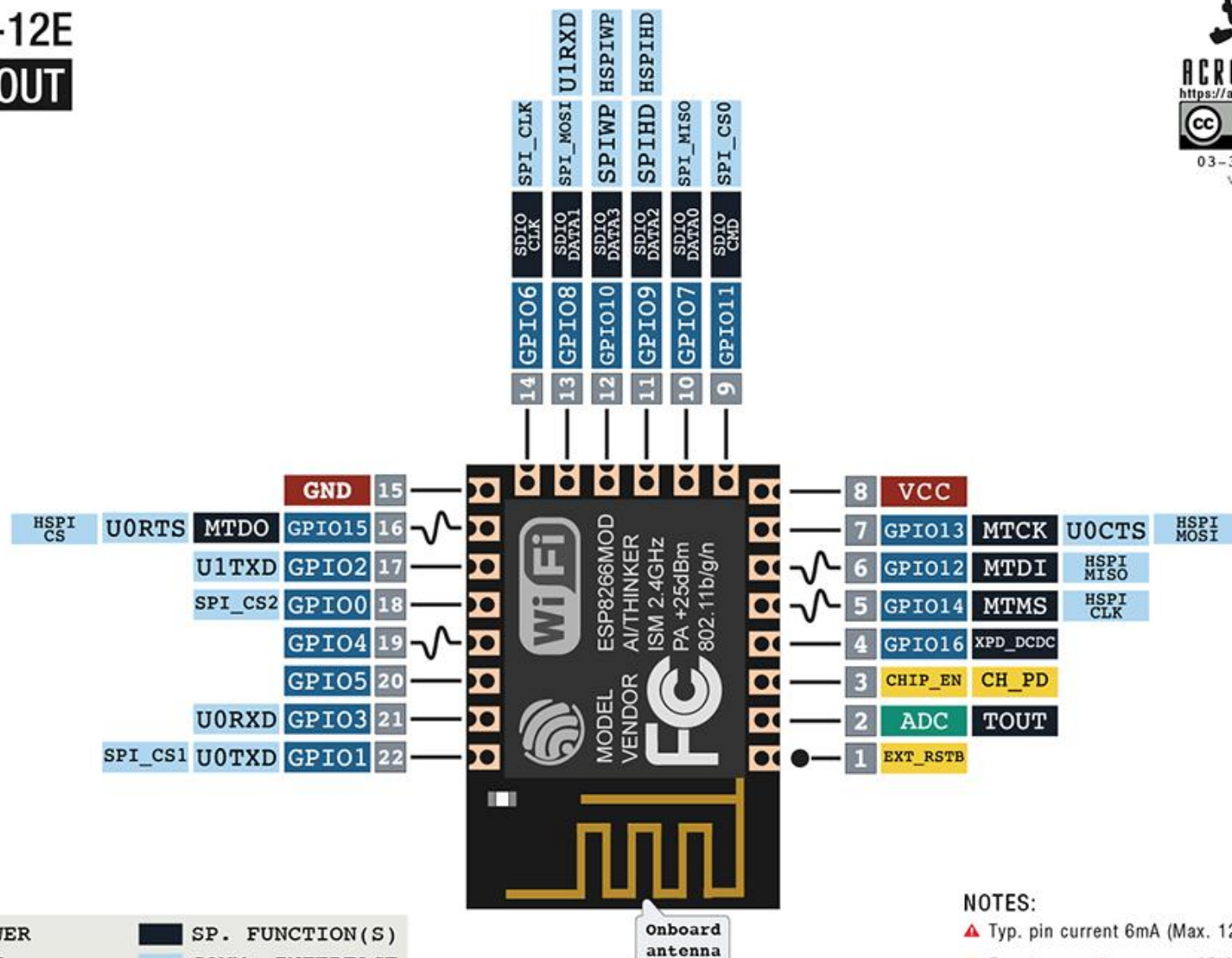
Programming ESP-01





ESP-12E

ESP-12E PINOUT

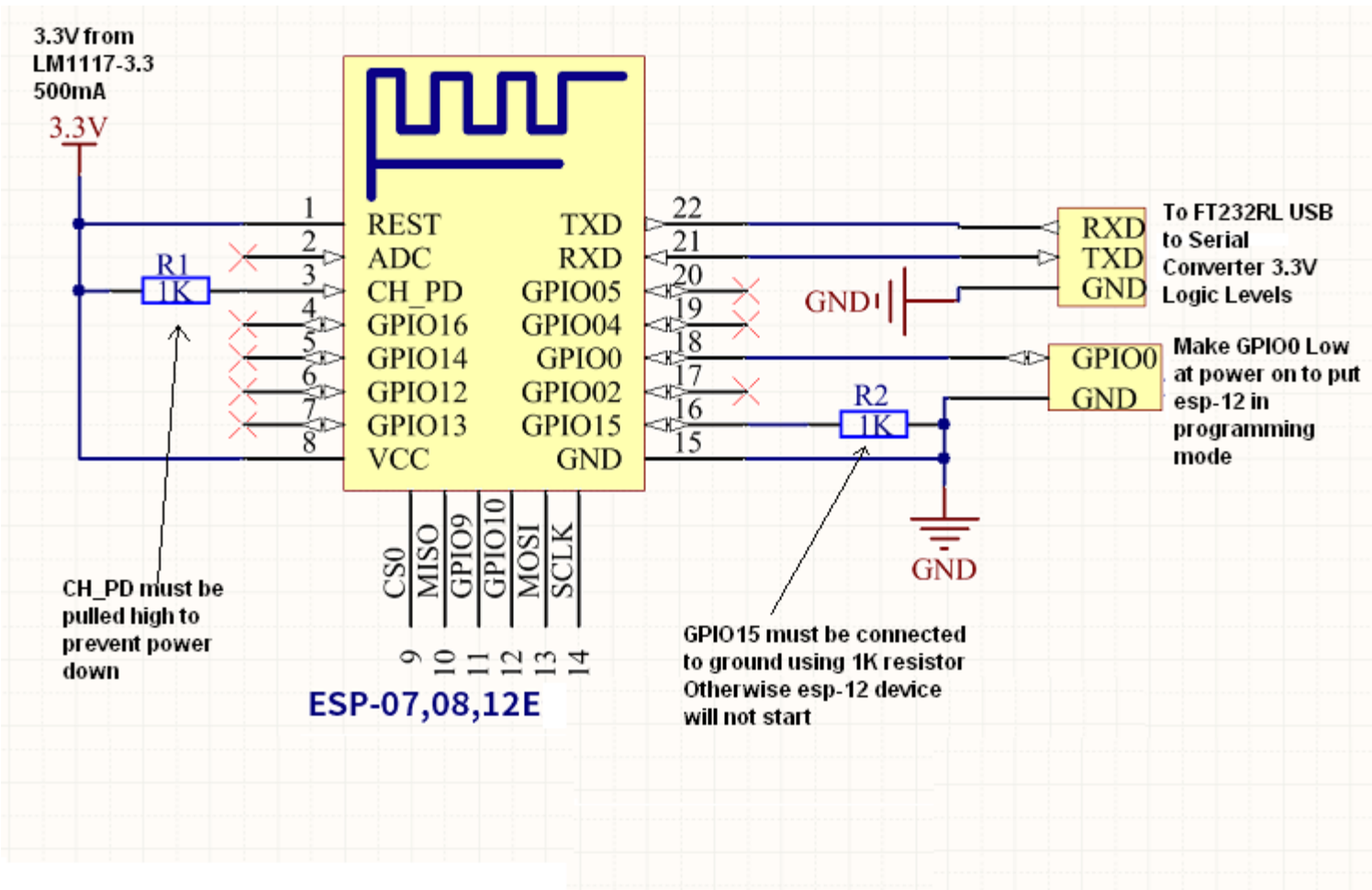


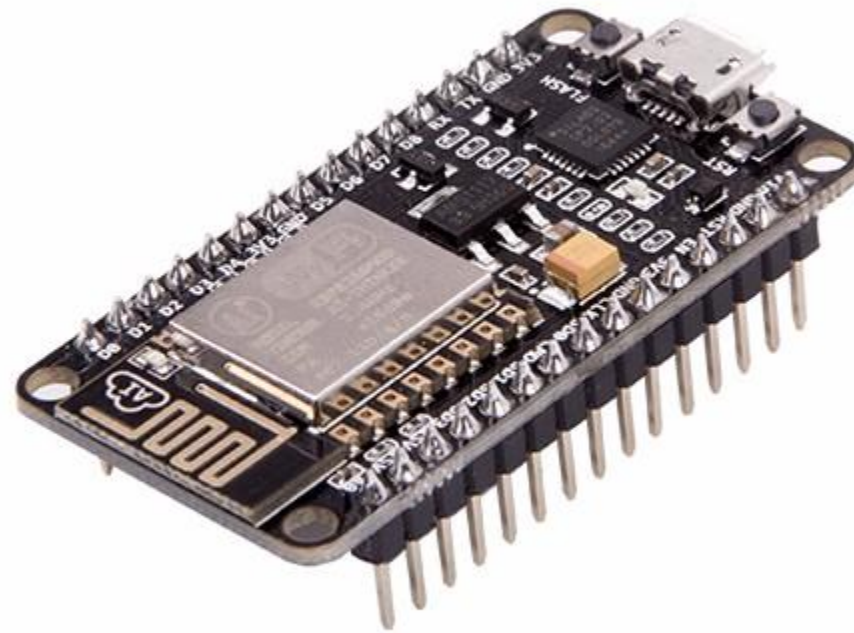
| | |
|--|---|
| POWER | SP. FUNCTION(S) |
| I/O | COMM. INTERFACE |
| ADC | PIN NUMBER |
| CONTROL | ~ PWM |
| N/C | |

NOTES:

- ▲ Typ. pin current 6mA (Max. 12mA)
- ▲ For sleep mode, connect GPIO16 and EXT_RSTB. On wakeup, GPIO16 will output LOW for system reset.
- ▲ On boot/reset/wakeup, keep GPIO15 LOW and GPIO2 HIGH.

Programming ESP-12E





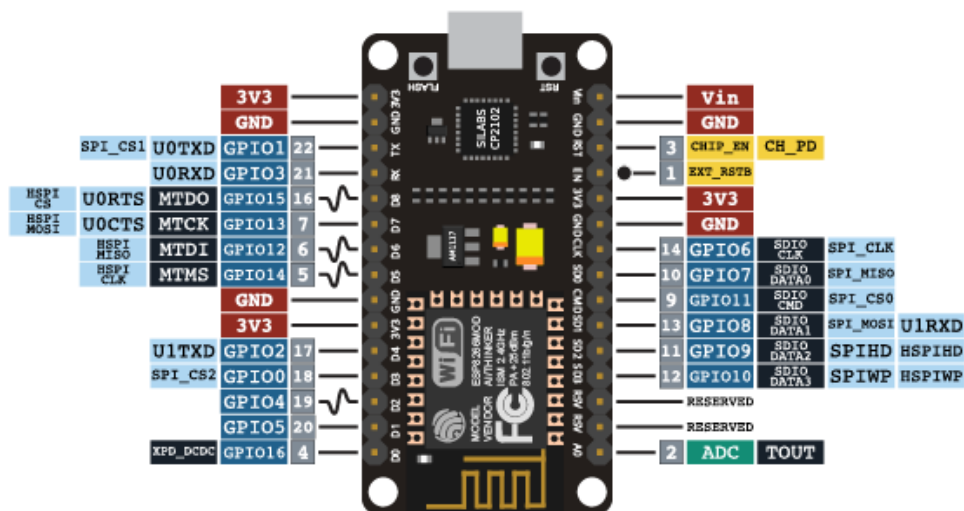
NodeMCU

ESP-12E DEVELOPMENT BOARD

PINOUT

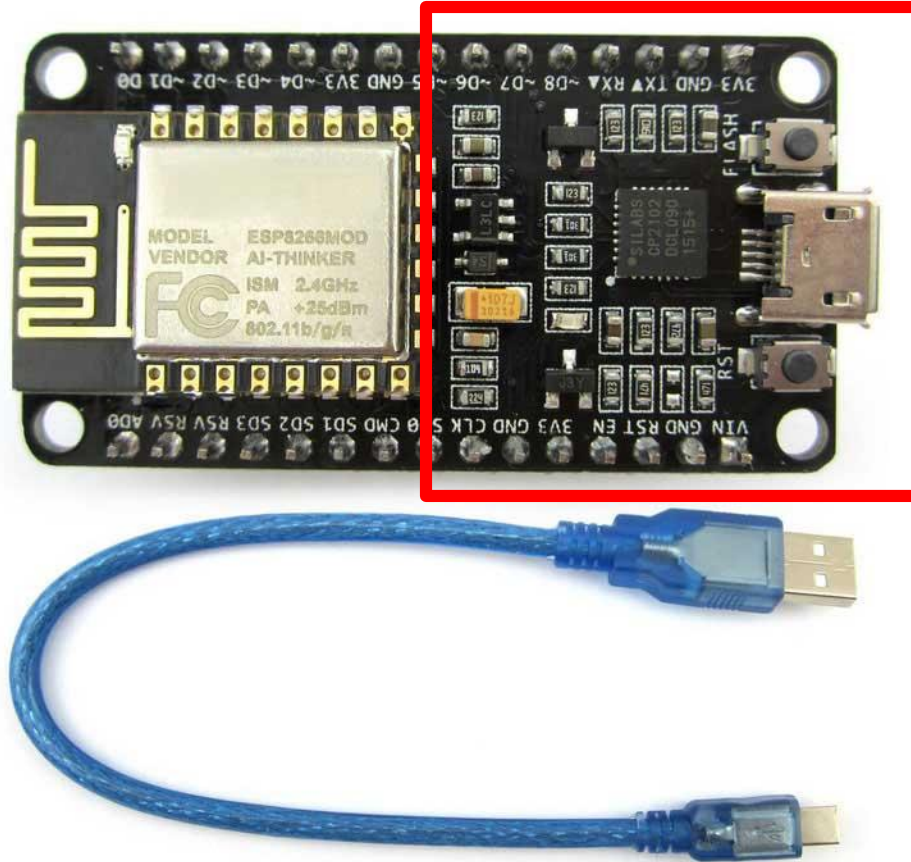
NOTES:

- ▲ Typ. pin current 6mA (Max. 12mA)
- ▲ For sleep mode, connect GPIO16 and EXT_RSTB. On wakeup, GPIO16 will output LOW for system reset.
- ▲ On boot/reset/wakeup, keep GPIO15 LOW and GPIO2 HIGH.



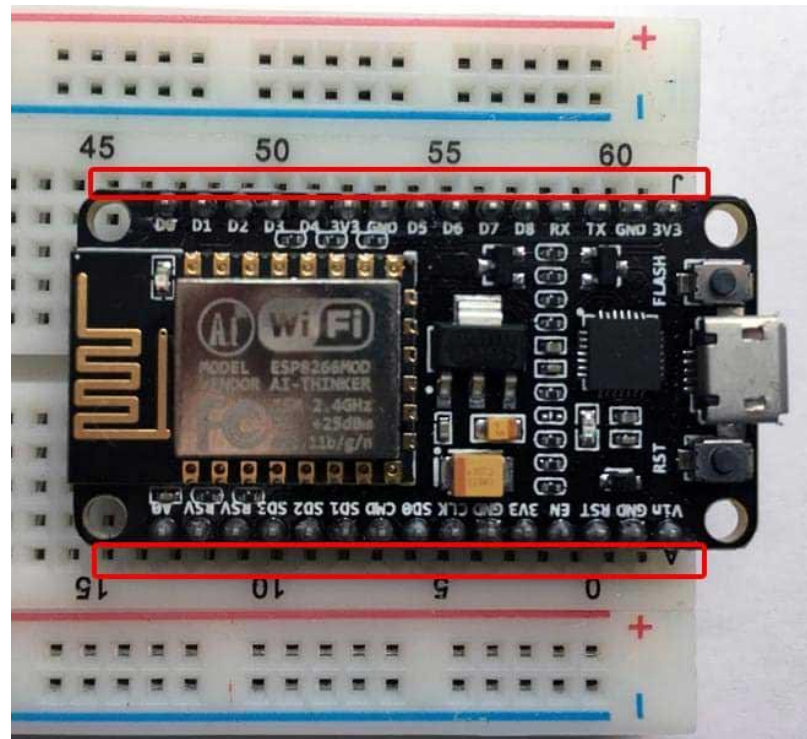
| | |
|---------|-----------------|
| POWER | SP. FUNCTION(S) |
| I/O | COMM. INTERFACE |
| ADC | PIN NUMBER |
| CONTROL | PWM |
| N/C | |

Programming NodeMCU



CP2102
USB-UART (VCP)

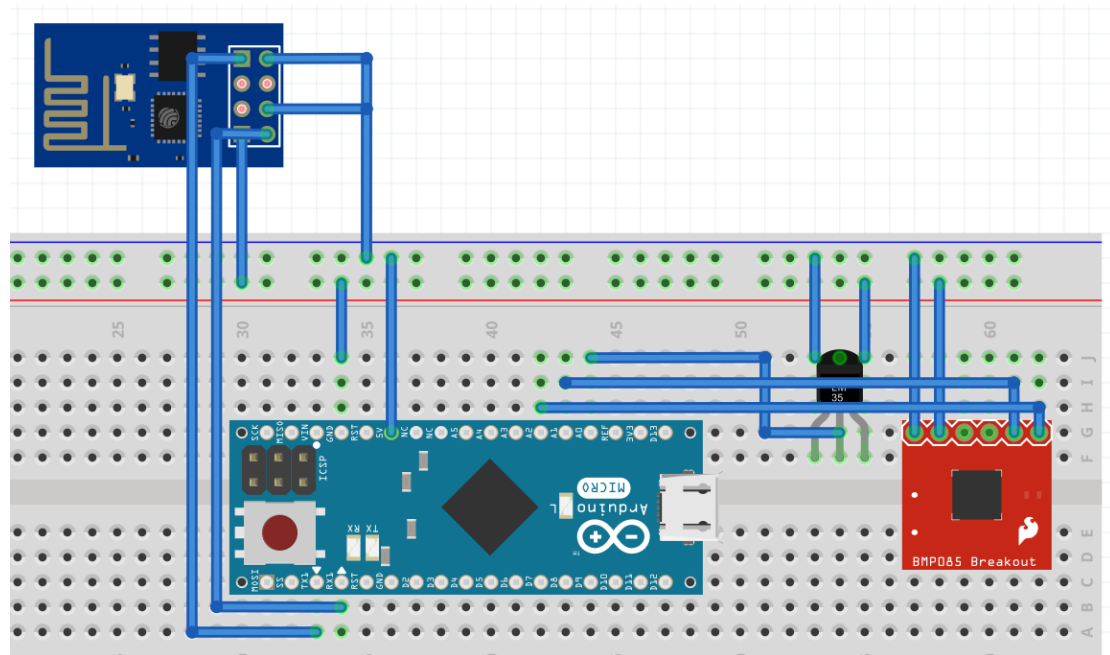
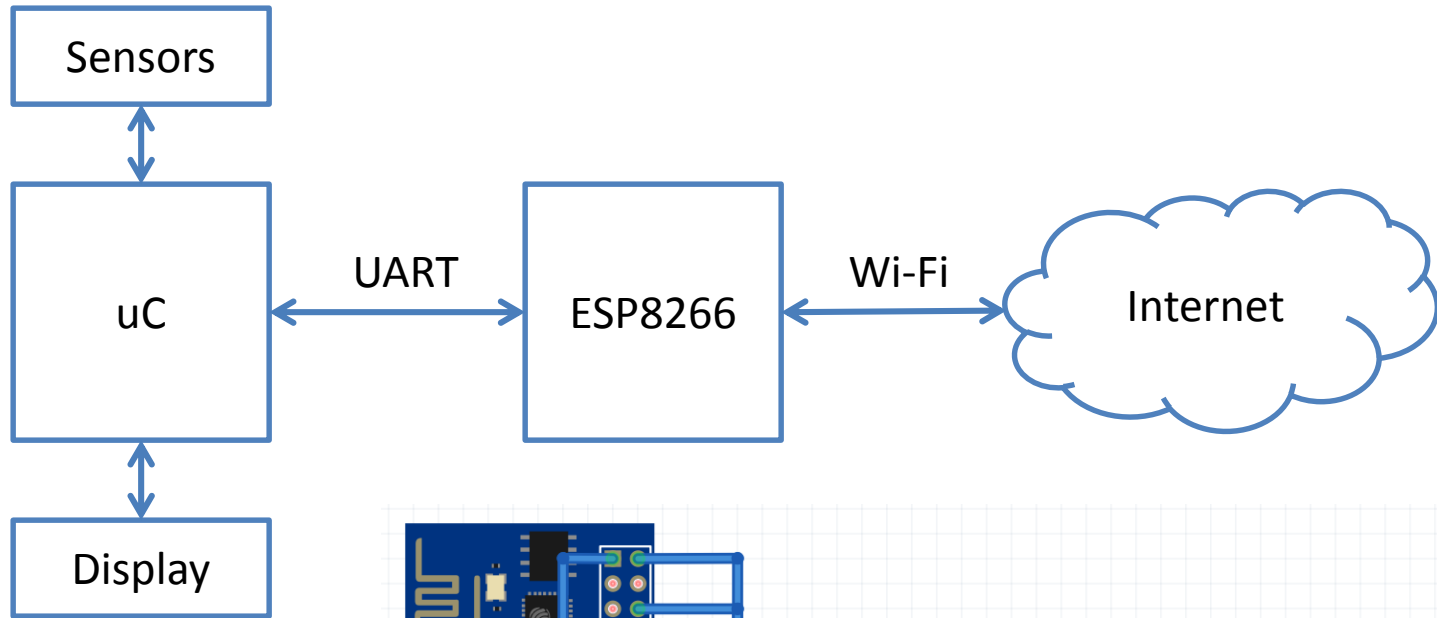
NodeMCU v2 is ProtoBoard friendly



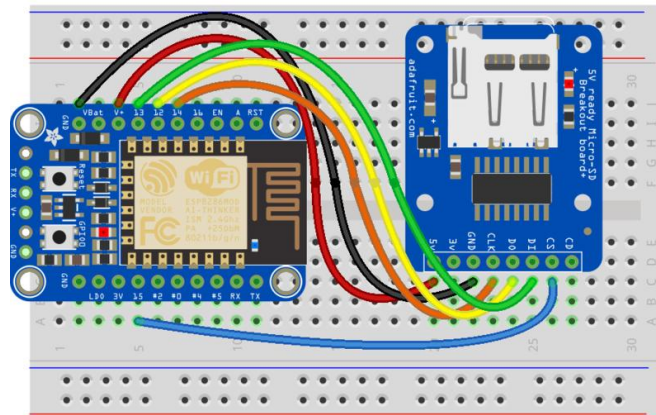
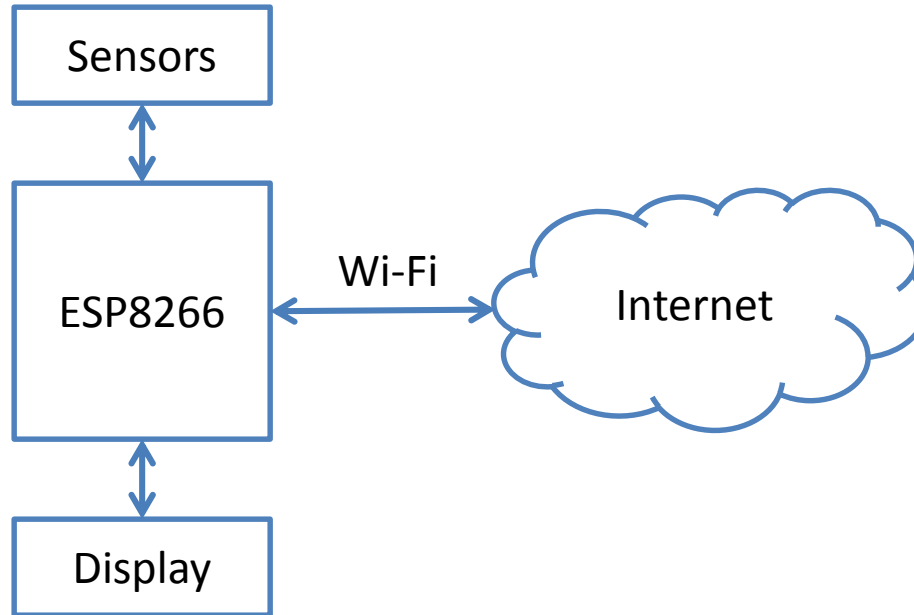
Firmware options:

- **Offical ESP8266 AT Firmware**
- Espressif FreeRTOS Firmware / SDK
- NodeMCU Lua Based Firmware
- Micro Python
- **Arduino core for ESP8266**
- Visuino
- ESP-Lisp
- Espruino
- Sming
- ESP8266 Basic
- ESPWebFramework

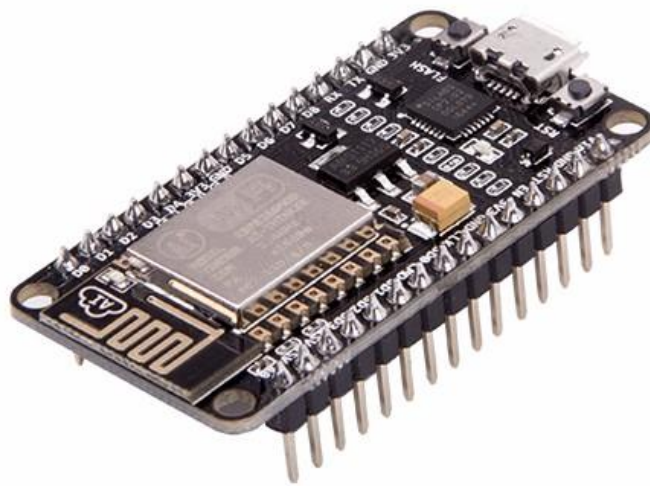
Slave Mode (AT Commands)



Standalone Mode



Hands on NodeMCU

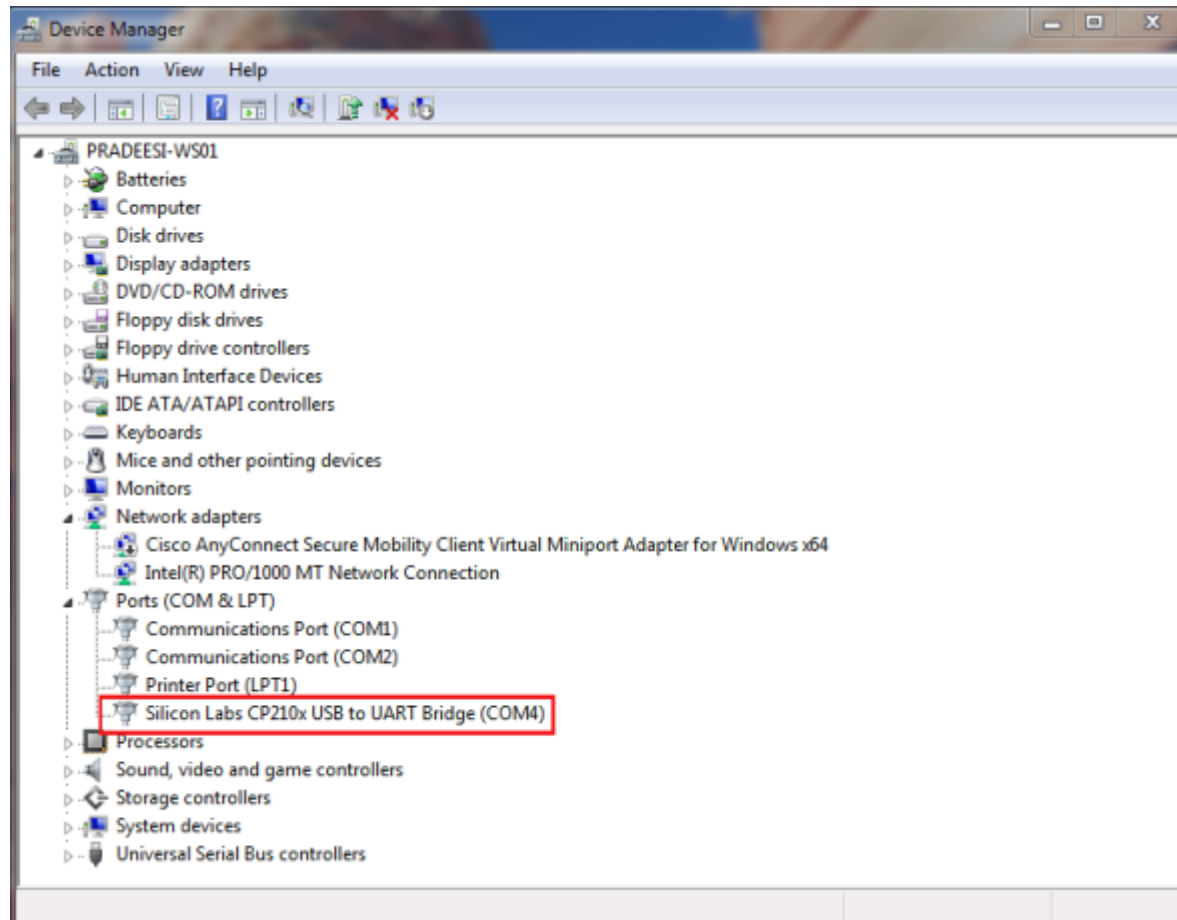


Prerequisites:

- NodeMCU Module
- USB to Micro USB cable
- Computer (Win OS) with Internet Connection
- Arduino IDE 1.6.4+

Configuration Steps:

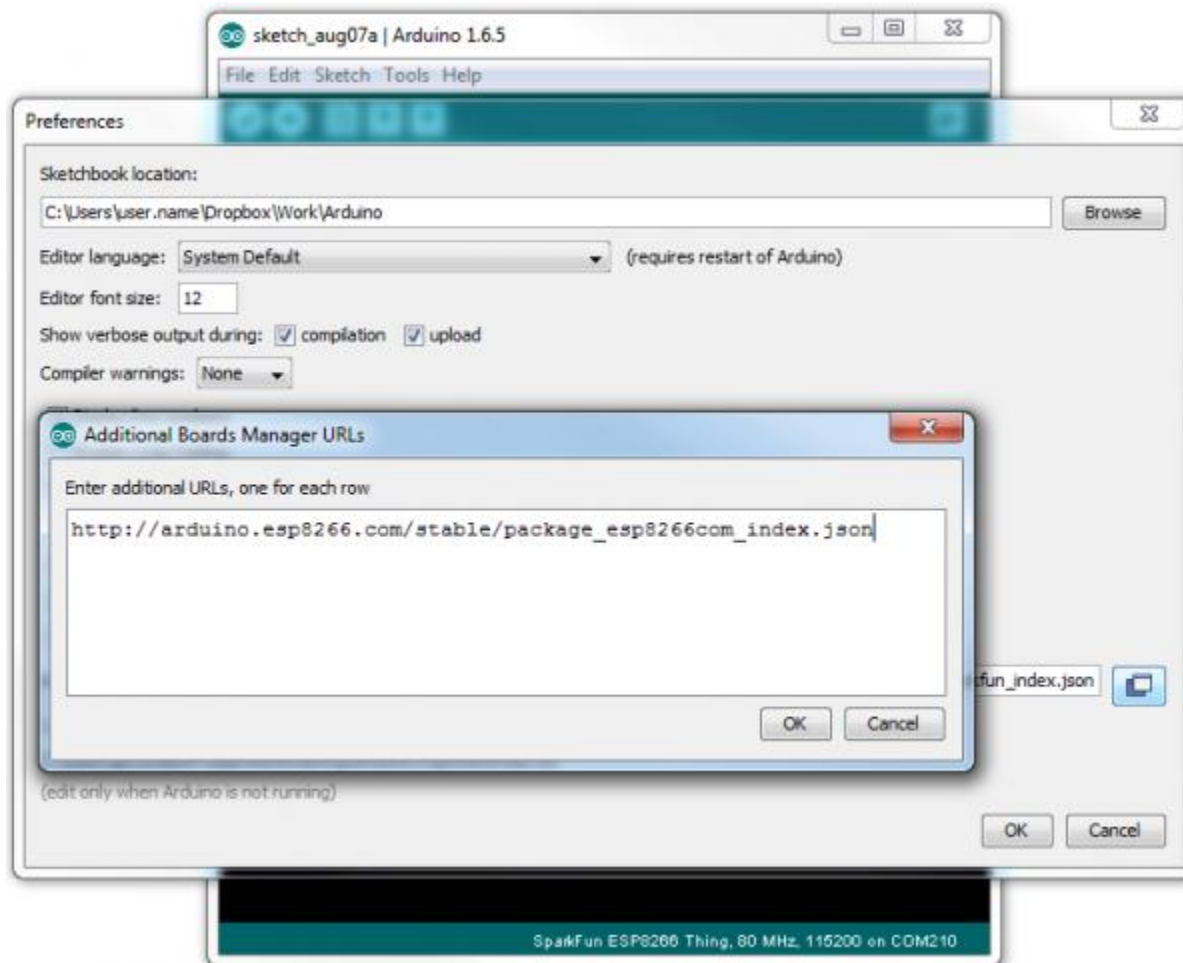
1. Download and Install “CP2102 USB to UART Bridge Controller” driver.



2. Install the ESP8266 Arduino Addon.

Open up Arduino, then go to the Preferences (**File> Preferences**). Then, towards the bottom of the window, copy this URL into the “**Additional Board Manager URLs**” text box:

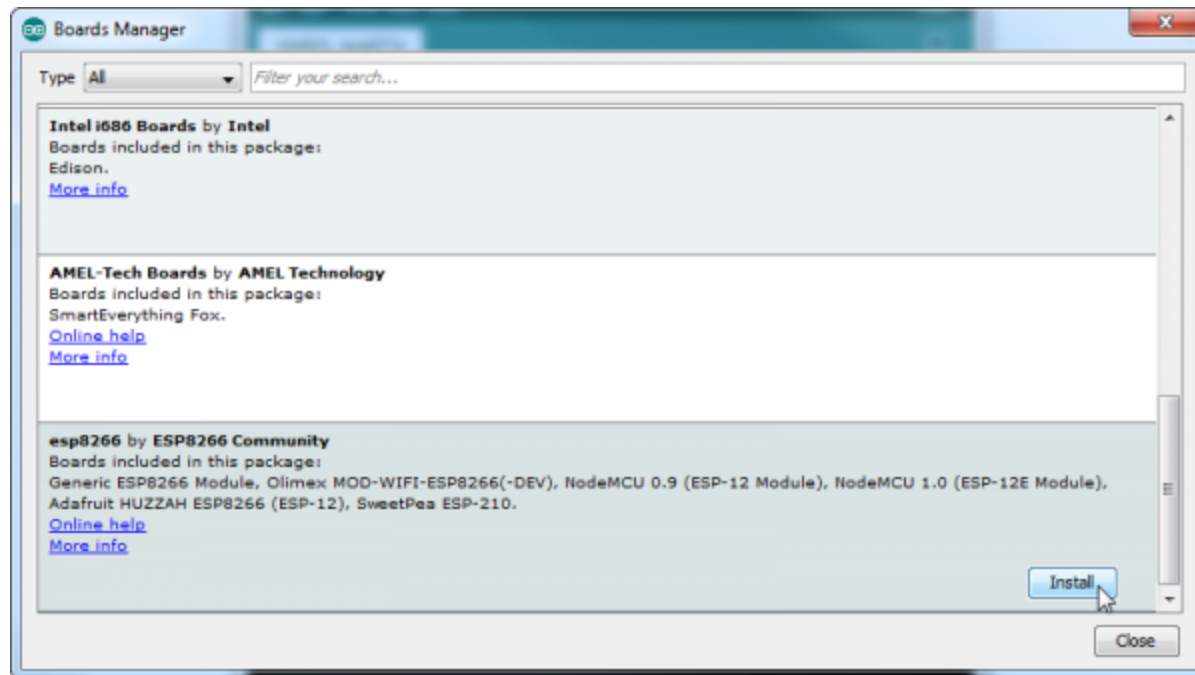
http://arduino.esp8266.com/stable/package_esp8266com_index.json



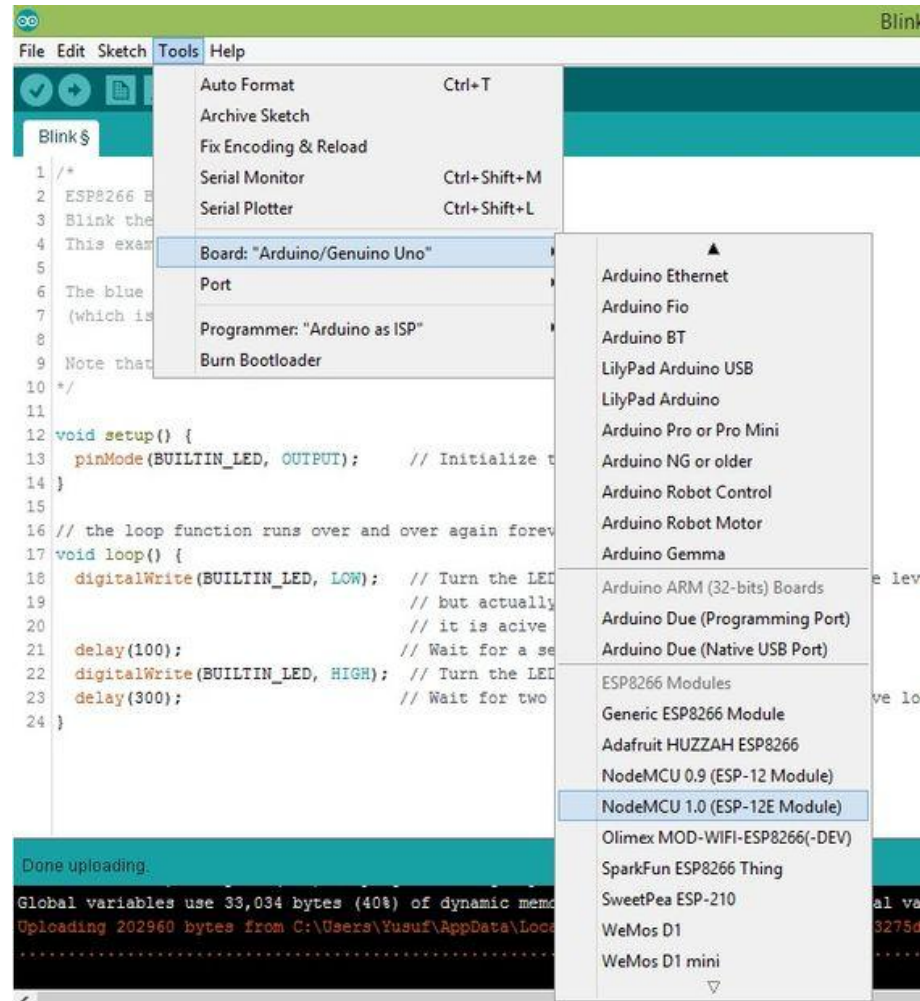
Then navigate to the Board Manager by going to:
 ”Tools > Boards > Boards Manager”.

There should be a couple new entries in addition to the standard Arduino boards. Look for **esp8266**.

Click on that entry, then select **Install**.

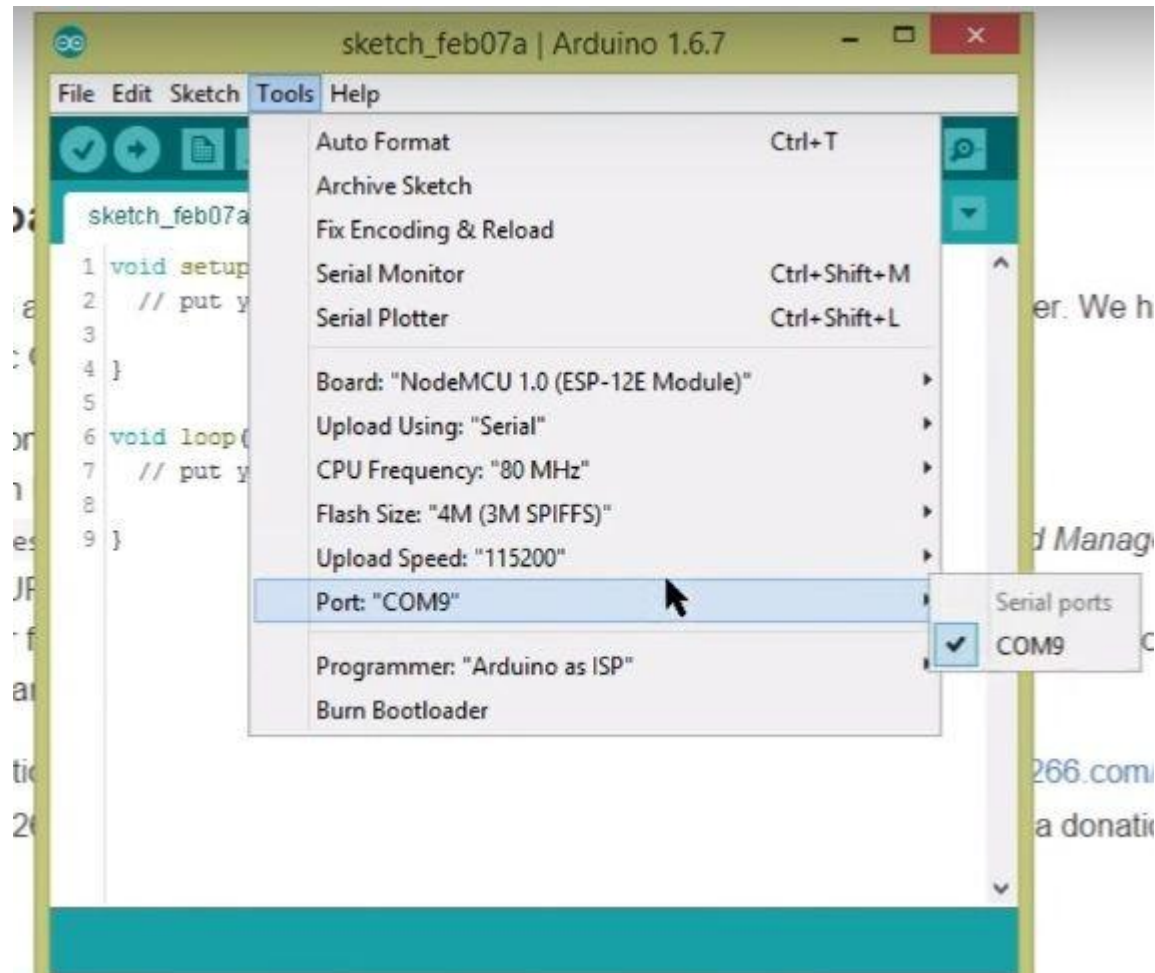


3. Select the NodeMCU board:
 "Tools > Boards > NodeMCU 1.0 (ESP-12E Module)".



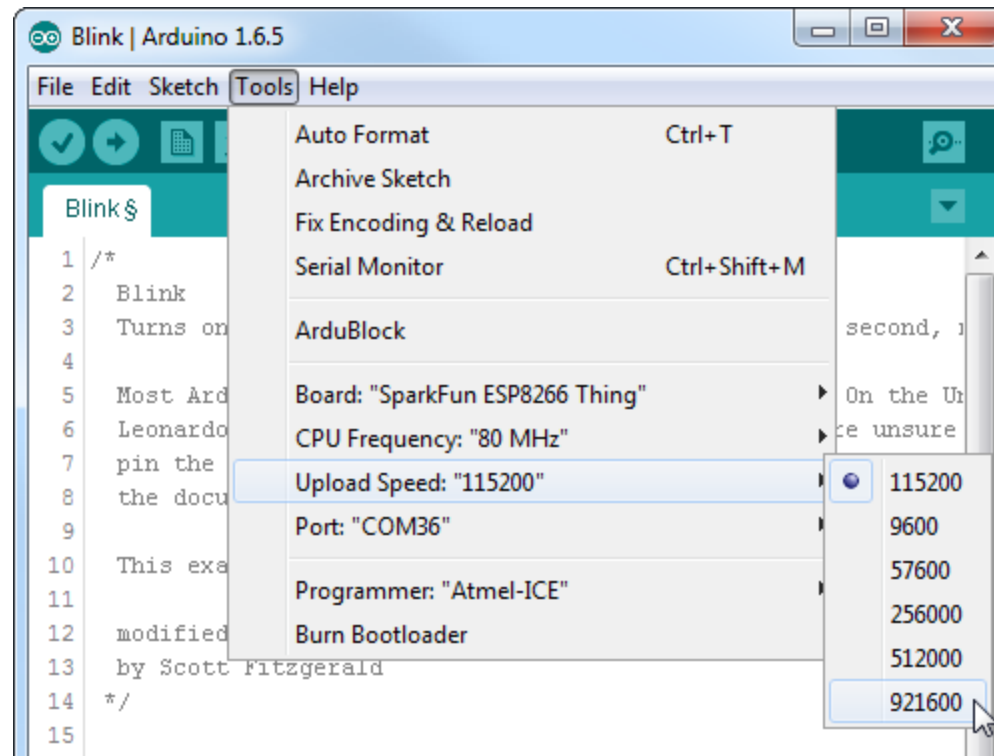
4. Select NodeMCU COM Port:

"Tools > Port > COM(n)".



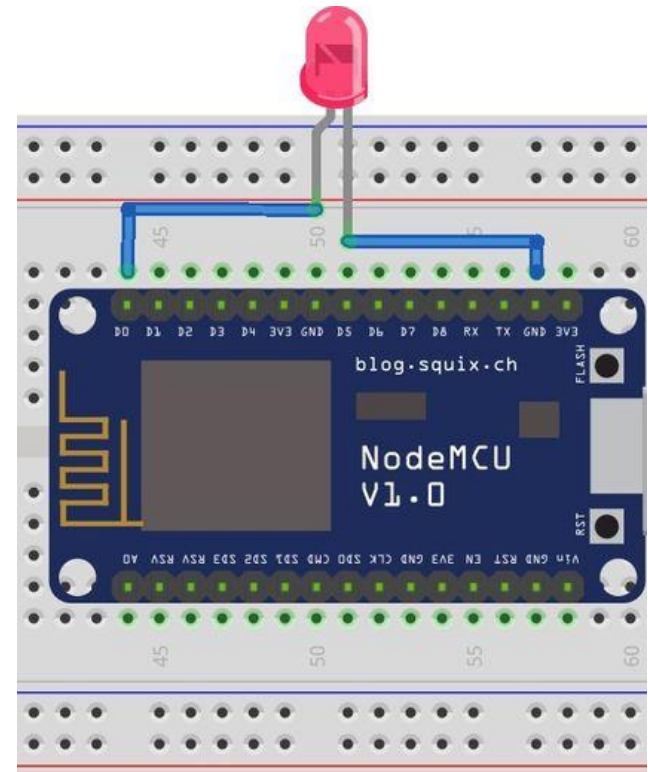
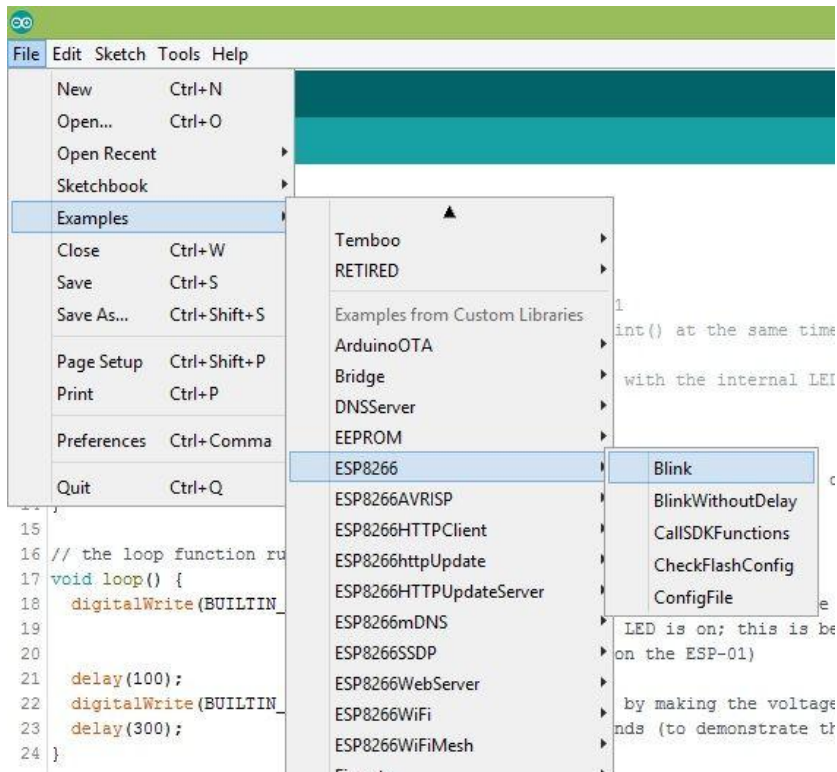
5. Select NodeMCU Upload Speed:

"Tools > Upload Speed > 921600".



5. Make it work:

”File > Examples > ESP8266 > Blink”.





ARE YOU

READY?

makeameme.org

Need help with your NodeMCU setup?

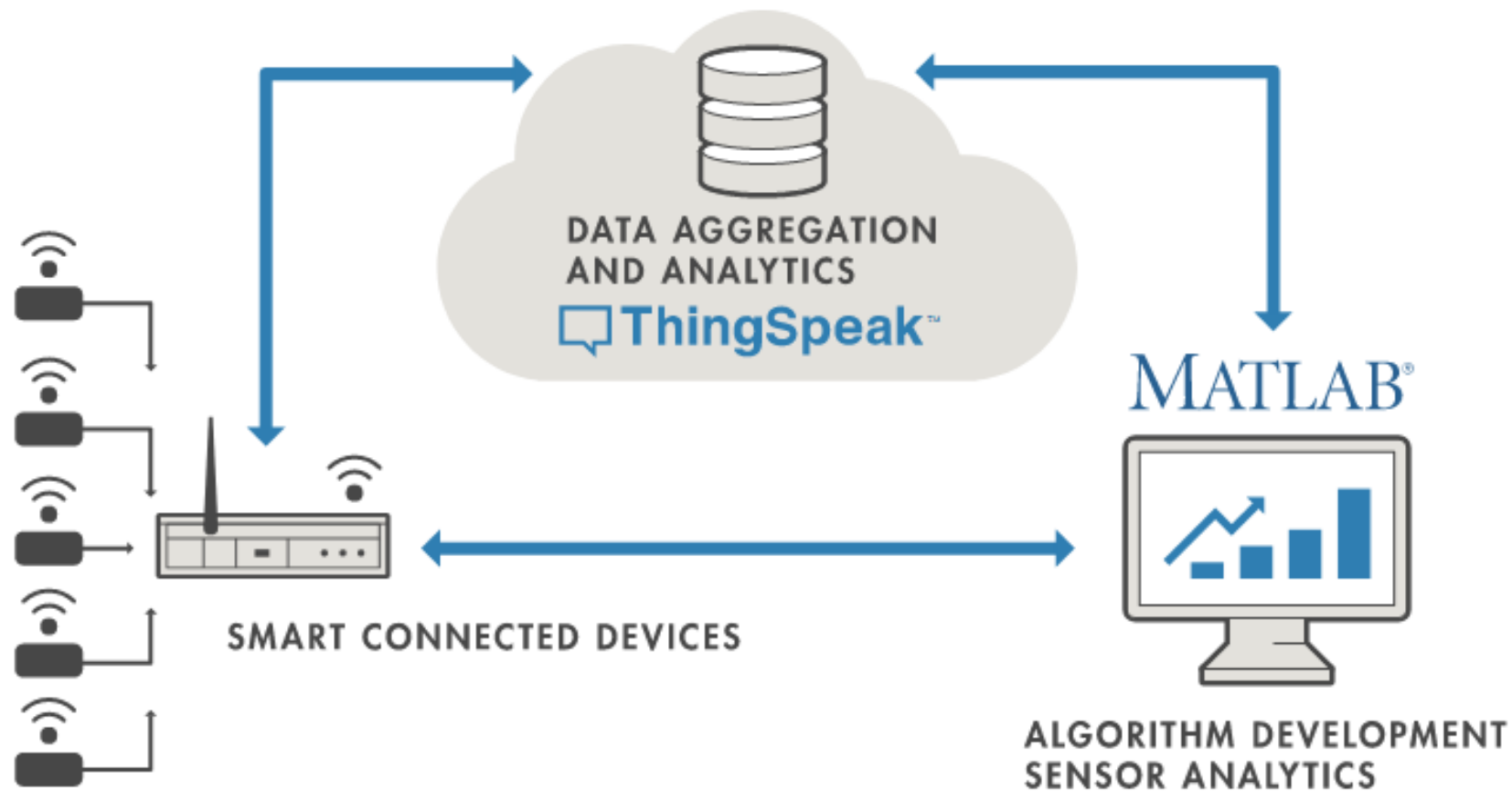


Still need help?



I can't take it anymore. I just want to die!

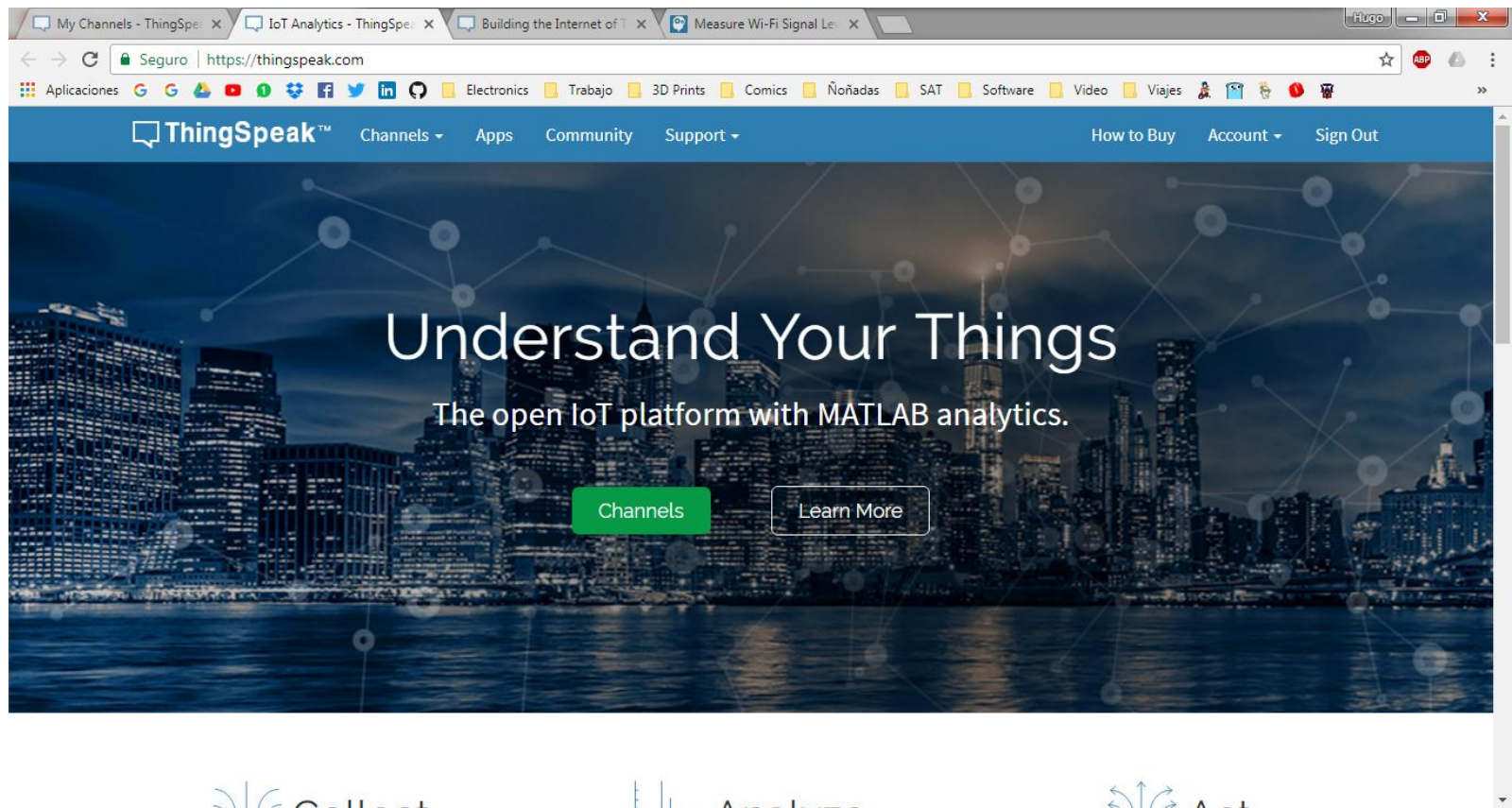
[adult swim]



Configuration Steps:

1. Sign up or register to Thingspeak:

https://thingspeak.com/users/sign_up



2. Create a new Channel:

<https://thingspeak.com/channels/new>

The screenshot shows the 'New Channel' page on the ThingSpeak website. The browser address bar shows the URL <https://thingspeak.com/channels/new>. The page has a blue header with the ThingSpeak logo and navigation links: Channels, Apps, Community, Support, How to Buy, Account, and Sign Out. The main content area is divided into two columns. The left column is titled 'New Channel' and contains a form with the following fields:

- Name:** A text input field.
- Description:** A larger text input field.
- Field 1:** A text input field containing 'Field Label 1' with a checked checkbox to its right.
- Field 2:** An empty text input field with an unchecked checkbox to its right.
- Field 3:** An empty text input field with an unchecked checkbox to its right.
- Field 4:** An empty text input field with an unchecked checkbox to its right.
- Field 5:** An empty text input field with an unchecked checkbox to its right.
- Field 6:** An empty text input field with an unchecked checkbox to its right.
- Field 7:** An empty text input field with an unchecked checkbox to its right.
- Field 8:** An empty text input field with an unchecked checkbox to its right.

The right column is titled 'Help' and contains a paragraph of text and a 'Channel Settings' section with a list of instructions:

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.
- **Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.

3. Get your channel API keys:

`https://thingspeak.com/channels/<channel_number>/api_keys`

The screenshot shows a web browser window with the URL `https://thingspeak.com/channels/327799/api_keys`. The page has a blue header with the Thingspeak logo and navigation links: Channels, Apps, Community, Support, How to Buy, Account, and Sign Out. Below the header, the author is listed as 'argandas' and the access is 'Private'. A tabbed interface shows 'API Keys' as the active tab, with other tabs for Private View, Public View, Channel Settings, Sharing, and Data Import / Export.

Write API Key

Key: `MYC91BJSZLFGGUWI`

[Generate New Write API Key](#)

Read API Keys

Key: `7QC1C0YLDT8NR5PO`

Note:

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

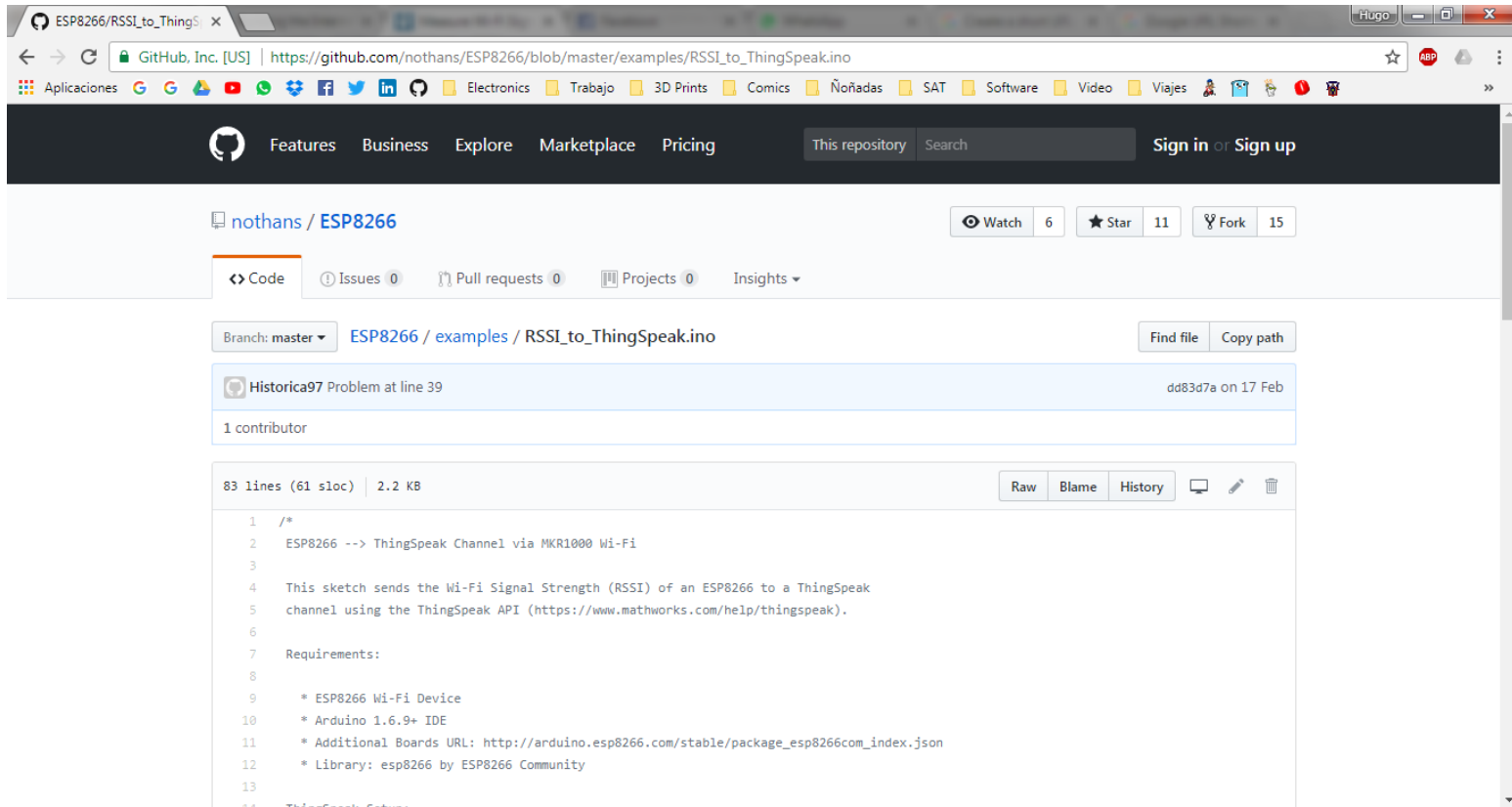
[Update a Channel Feed](#)

```
GET https://api.thingspeak.com/update?api_key=MYC91BJSZLFGGUWI&field=
```

4. Publish to your channel, download example code from:

<https://goo.gl/kkycnr>

(https://github.com/nothans/ESP8266/blob/master/examples/RSSI_to_ThingSpeak.ino)



The screenshot shows a web browser displaying the GitHub repository page for `nothans/ESP8266`. The browser's address bar shows the URL `https://github.com/nothans/ESP8266/blob/master/examples/RSSI_to_ThingSpeak.ino`. The repository page includes a navigation bar with links for Features, Business, Explore, Marketplace, and Pricing. Below the navigation bar, the repository name `nothans/ESP8266` is displayed, along with statistics: 6 Watchers, 11 Stars, and 15 Forks. The `Code` tab is selected, showing the file `RSSI_to_ThingSpeak.ino` from the `master` branch. A commit message `Historica97 Problem at line 39` is visible, dated `dd83d7a on 17 Feb`. The file content is displayed in a code editor, showing 83 lines of code (61 sloc) and a size of 2.2 KB. The code includes comments and requirements for the ESP8266 Wi-Fi device, Arduino IDE, and the ThingSpeak API.

```
1 /*
2  ESP8266 --> ThingSpeak Channel via MKR1000 Wi-Fi
3
4  This sketch sends the Wi-Fi Signal Strength (RSSI) of an ESP8266 to a ThingSpeak
5  channel using the ThingSpeak API (https://www.mathworks.com/help/thingspeak).
6
7  Requirements:
8
9  * ESP8266 Wi-Fi Device
10 * Arduino 1.6.9+ IDE
11 * Additional Boards URL: http://arduino.esp8266.com/stable/package_esp8266com_index.json
12 * Library: esp8266 by ESP8266 Community
13
14 ThingSpeak Setup...
```

5. Watch your data live!

https://thingspeak.com/channels /<channel_number>

NodeMCU Example

Channel ID: **210455**

Author: [iothans](#)

Access: Private

 esp8266

Private View

Public View

Channel Settings

API Keys

Data Import / Export

 Add Visualizations

 Data Export

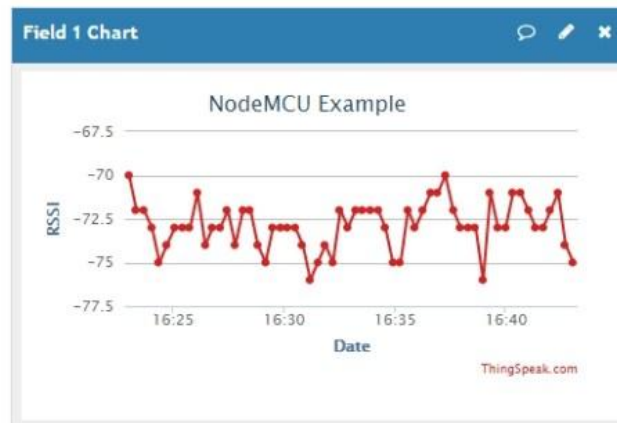
Channel Stats

Created: [27 days ago](#)

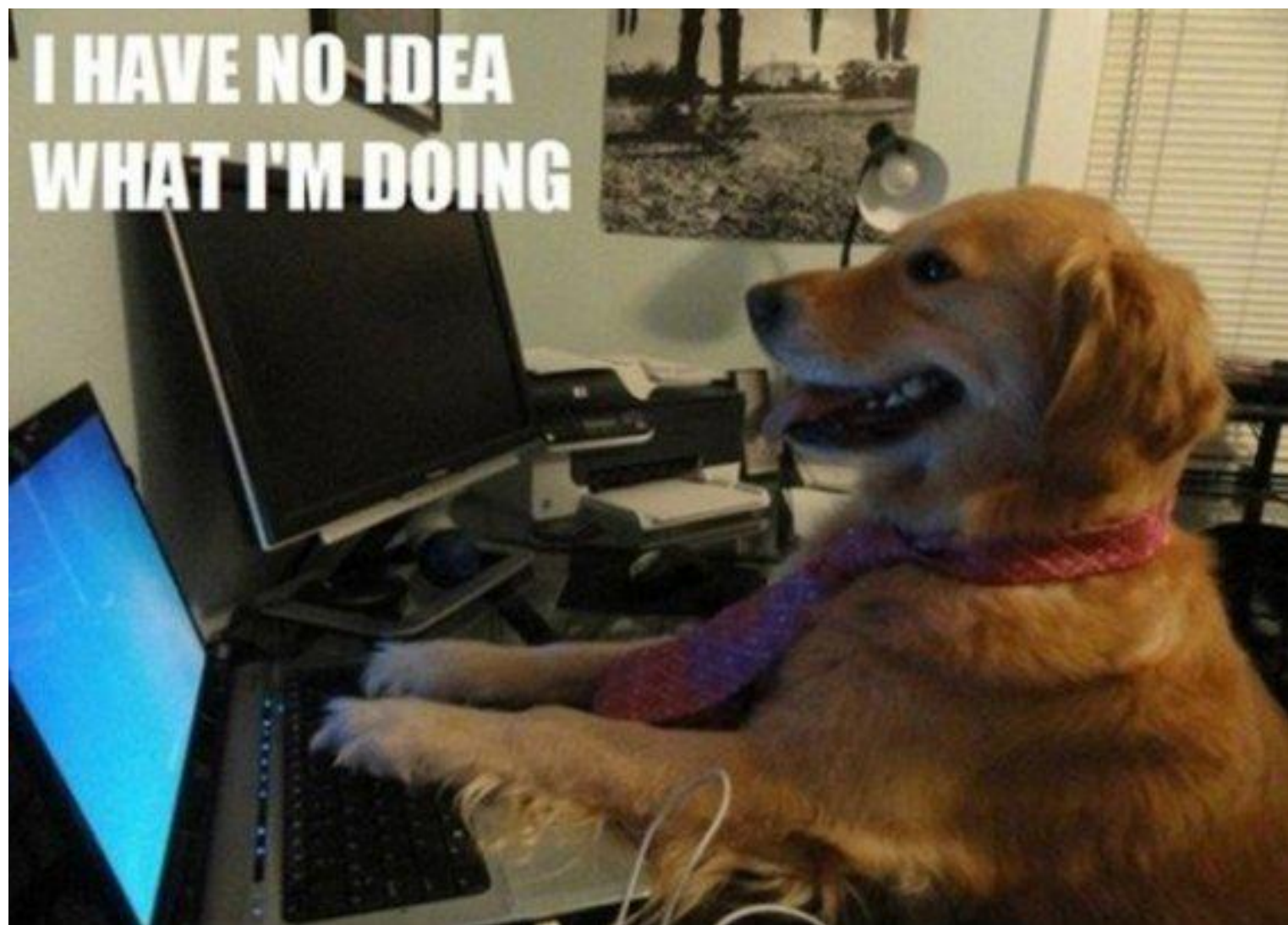
Updated: [3 minutes ago](#)

Last entry: [3 minutes ago](#)

Entries: 191



**I HAVE NO IDEA
WHAT I'M DOING**



Understand Thingspeak API

To update a channel feed, send an HTTP GET or POST request to:

`https://api.thingspeak.com/update`

Example:

POST /update HTTP/1.1

Host: api.thingspeak.com

Connection: close

X-THINGSPEAKAPIKEY: MYC91BJSZLFGGUWI

Content-Type: application/x-www-form-urlencoded

Content-Length: 22

field1=73&field2=100

HTTP

(Hypertext Transfer Protocol)

The Hypertext Transfer Protocol (HTTP) is an application protocol for distributed, collaborative, and hypermedia information systems.[1] HTTP is the foundation of data communication for the World Wide Web.

Hypertext is structured text that uses logical links (hyperlinks) between nodes containing text. HTTP is the protocol to exchange or transfer hypertext.

Define “protocol”



Define “protocol”

In information technology, a protocol is the special set of rules that end points in a telecommunication connection use when they communicate. Protocols specify interactions between the communicating entities.

HTTP Requests

- **GET**

The GET method requests a representation of the specified resource. Requests using GET should only retrieve data and should have no other effect

- **POST**

The POST method requests that the server accept the entity enclosed in the request as a new subordinate of the web resource identified by the URI. The data POSTed might be, for example, an annotation for existing resources; a message for a bulletin board, newsgroup, mailing list, or comment thread; a block of data that is the result of submitting a web form to a data-handling process; or an item to add to a database.

- **PUT**

The PUT method requests that the enclosed entity be stored under the supplied URI. If the URI refers to an already existing resource, it is modified; if the URI does not point to an existing resource, then the server can create the resource with that URI.

- **DELETE**

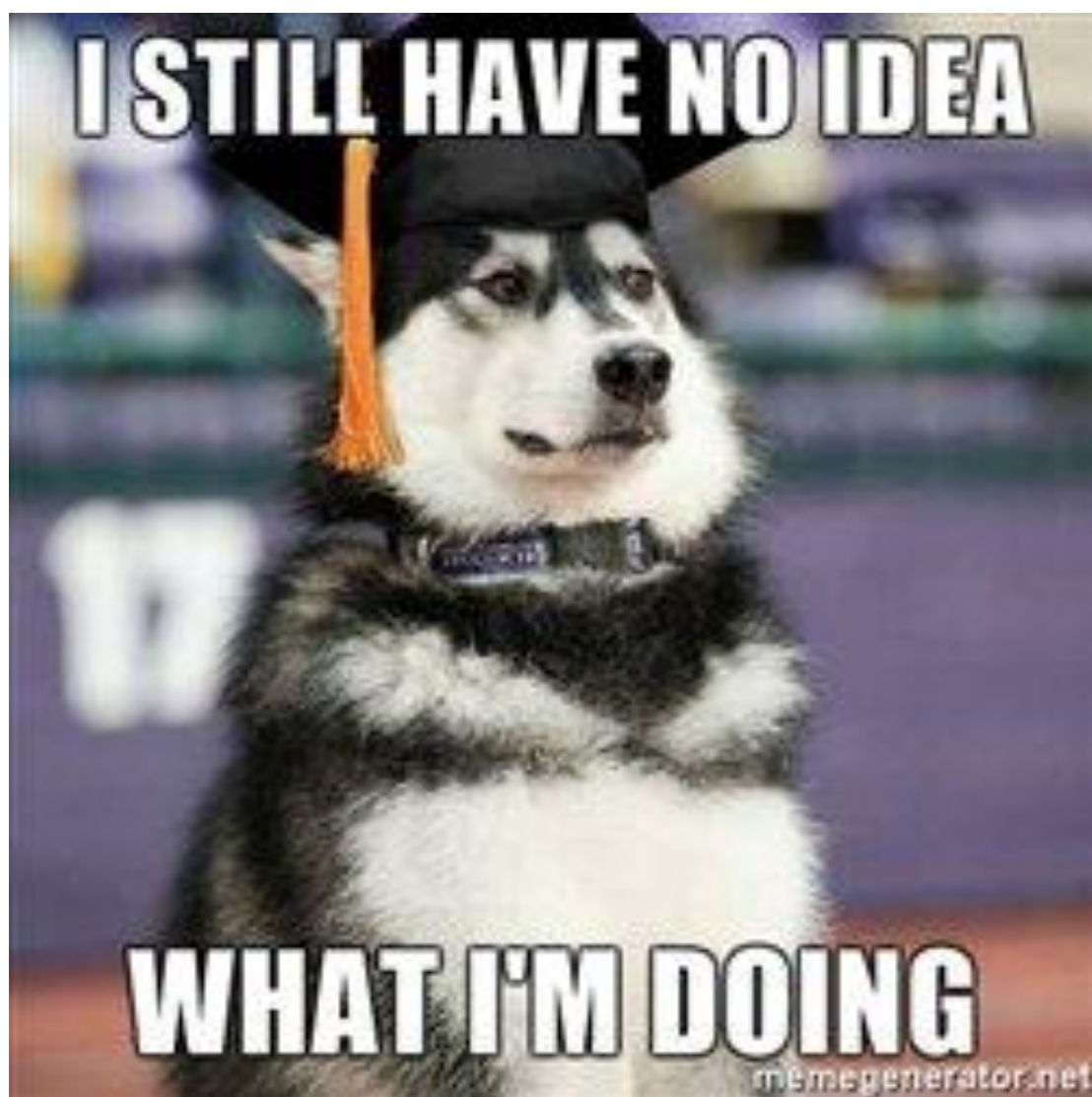
The DELETE method deletes the specified resource.

OSI Model

(Open Systems Interconnection model)

| OSI model | | |
|-----------|--------------------|--------------------------------------|
| Layer | Name | Example protocols |
| 7 | Application Layer | HTTP, FTP, DNS, SNMP, Telnet |
| 6 | Presentation Layer | SSL, TLS |
| 5 | Session Layer | NetBIOS, PPTP |
| 4 | Transport Layer | TCP, UDP |
| 3 | Network Layer | IP, ARP, ICMP, IPsec |
| 2 | Data Link Layer | PPP, ATM, Ethernet |
| 1 | Physical Layer | Ethernet, USB, Bluetooth, IEEE802.11 |

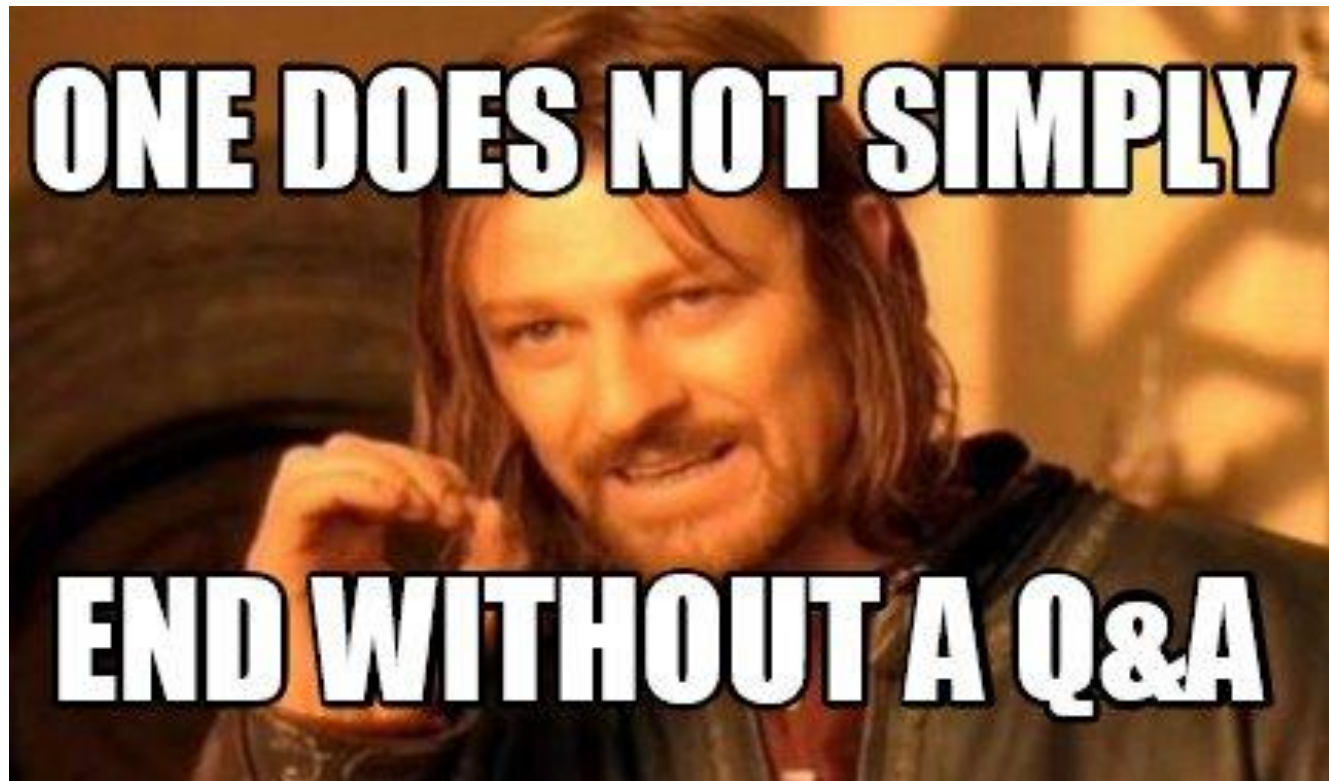
I STILL HAVE NO IDEA



WHAT I'M DOING

memegenerator.net

Q&A Time!



ARE YOU READY?

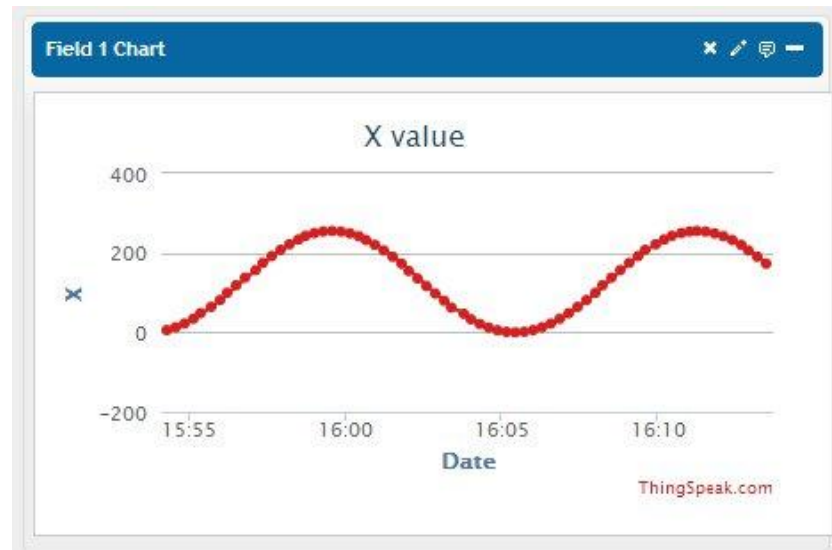
LET THE GAMES BEGIN

Challenge:

Draw a Sine wave on a Thingspeak chart, at least one complete cycle.
No rules on x-y axis range or sample timing, just draw the wave.

Duration: 30 minutes

The first one to show me the full wave and source code wins.



Prize:

Free admission on my next course, materials included & 1 beer...

Gracias por asistir!



Hugo Arganda
hugo.arganda@gmail.com
@argandas

*El 100% de las ganancias han sido donadas para apoyar a MakersGDL