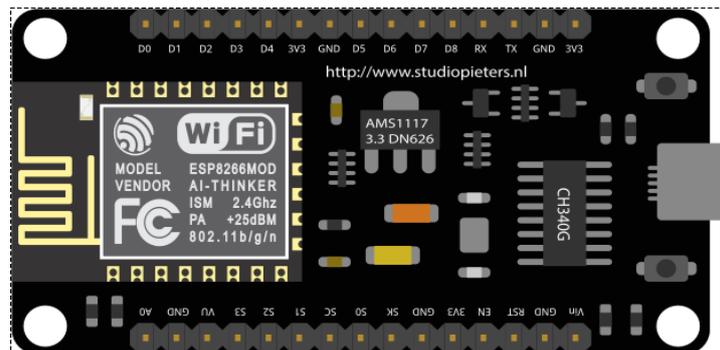


# NodeMCU V3 For Fast IoT Application Development

Authors: Ali Al Dahoud and Mohamed Fezari  
Faculty of IT, Al-Zaytoonah University Amman



fritzing

The best way to develop quickly an IoT application with less Integrated circuits to add is to choose this circuit “NodeMCU”. Today, we will give a detailed Introduction on NodeMCU V3. It is an open-source firmware and development kit that plays a vital role in designing a proper IoT product using a few script lines.

The module is mainly based on ESP8266 that is a low-cost Wi-Fi microchip incorporating both a full TCP/IP stack and microcontroller capability. It is introduced by manufacturer Espressif Systems. The ESP8266 NodeMcu is a complex device, which combines some features of the ordinary Arduino board with the possibility of connecting to the internet.

Arduino Modules and Microcontrollers have always been a great choice to incorporate automation into the relevant project. But these modules come with a little drawback as they don't feature a built-in WiFi capability, subsequently, we need to add external WiFi protocol into these devices to make them compatible with the internet channel.

This is the famous NodeMCU which is based on ESP8266 WiFi SoC. This is version 3 and it is based on ESP-12E (An ESP8266 based WiFi module). NodeMCU is also an open-source firmware and development kit that helps you to prototype your IOT product within a few LUA script lines, and of course you can always program it with Arduino IDE.

In this article, We will try present useful details related to this WiFi Development Kit, its main features, pinout and everything we need to know about this module and the application domain.

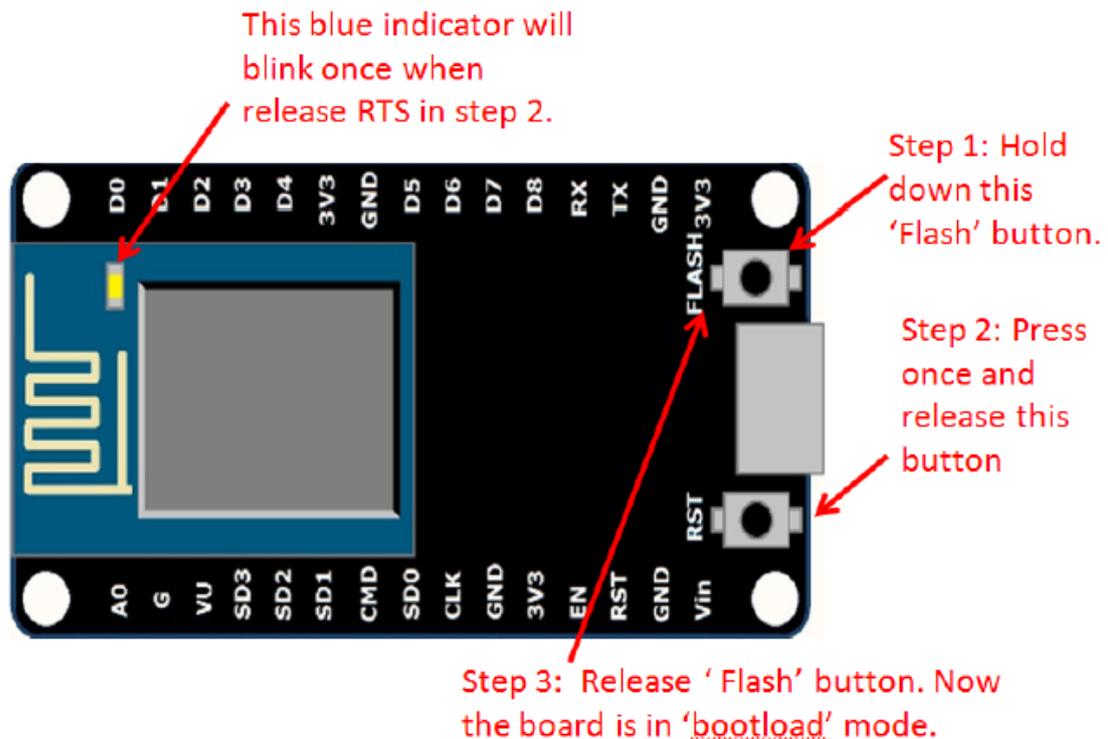
## Introduction NodeMCU V3

NodeMCU V3 is an open-source firmware and development kit that plays a vital role in designing an IoT product using a few script lines.

Multiple GPIO pins on the board allow us to connect the board with other peripherals and are capable of generating PWM, I2C, SPI, and UART serial communications.

- The interface of the module is mainly divided into two parts including both Firmware and Hardware where former runs on the ESP8266 Wi-Fi SoC and later is based on the ESP-12 module.

The firmware is based on Lua – A scripting language that is easy to learn, giving a simple programming environment layered with a fast scripting language that connects you with a well-known developer community.



And open source firmware gives you the flexibility to edit, modify and rebuild

the existing module and keep changing the entire interface until you succeed in optimizing the module as per your requirements.

- USB to UART converter is added on the module that helps in converting USB data to UART data which mainly understands the language of serial communication.

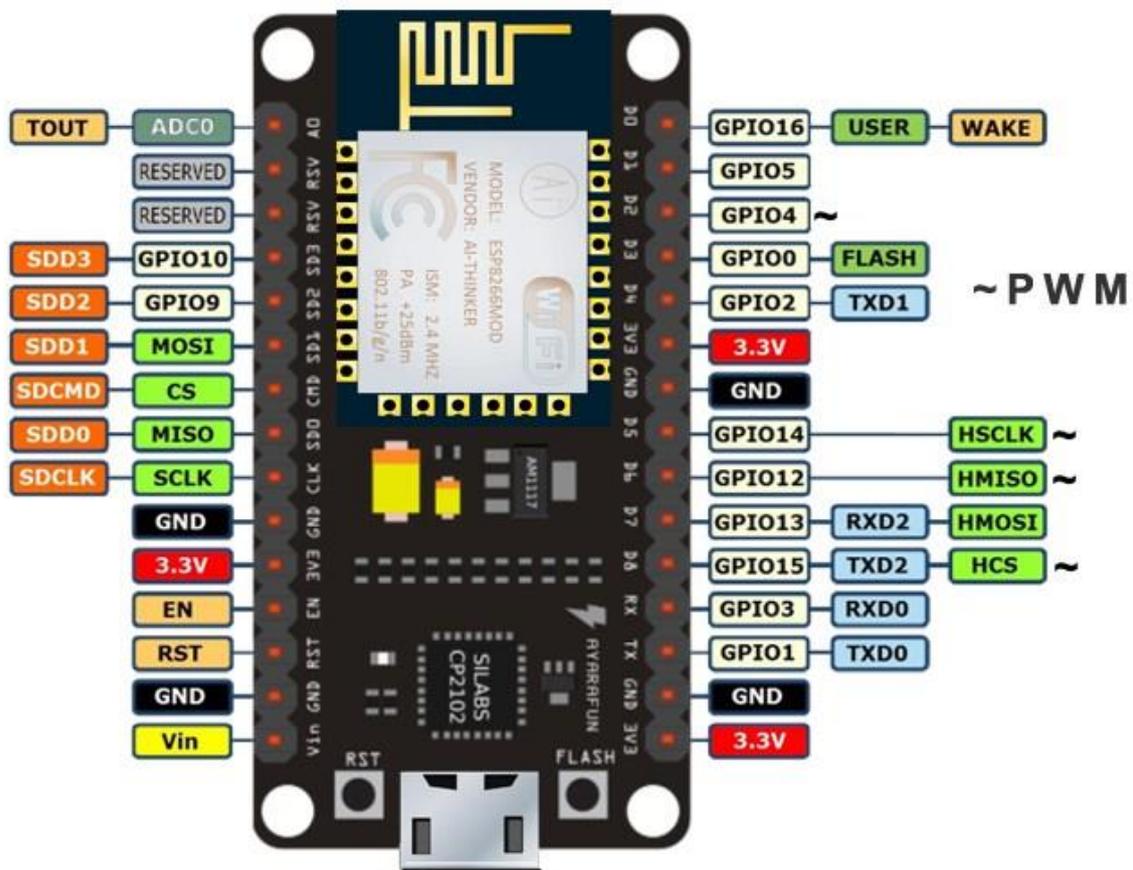
Instead of the regular USB port, MicroUSB port is included in the module that connects it with the computer for dual purposes: programming and powering up the board.

- The board incorporates status LED that blinks and turns off immediately, giving you the current status of the module if it is running properly when connected with the computer.

The ability of module to establish a flawless WiFi connection between two channels makes it an ideal choice for incorporating it with other embedded devices like Raspberry Pi.

## **NodeMCU V3 Pinout**

NodeMCU V3 comes with a number of GPIO Pins. Following figure shows the Pinout of the board.



- There is a candid difference between Vin and VU where former is the regulated voltage that may stand somewhere between 7 to 12 V while later is the power voltage for USB that must be kept around 5 V.

## Features

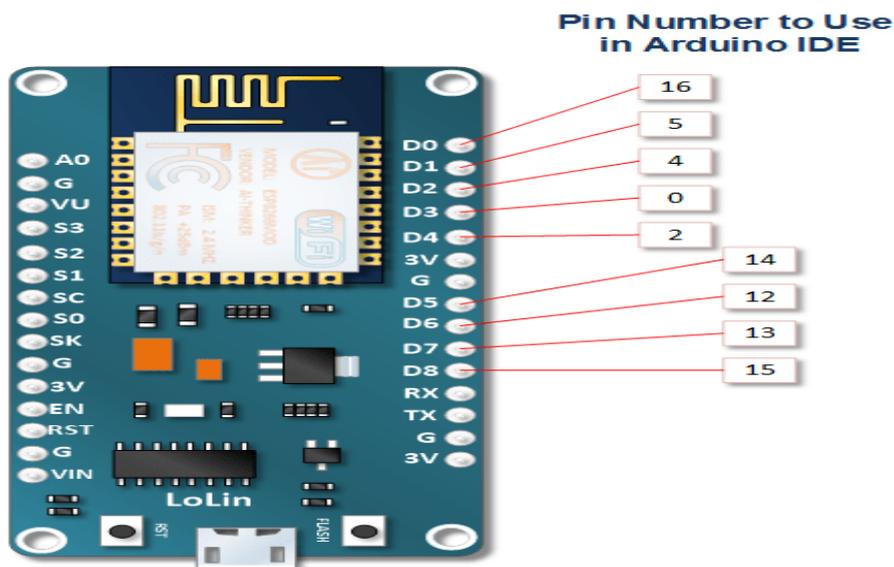
1. Open-source
2. Arduino-like hardware
3. Status LED
4. MicroUSB port
5. Reset/Flash buttons
6. Interactive and Programmable
7. Low cost
8. ESP8266 with inbuilt wifi
9. USB to UART converter
10. GPIO pins
11. Arduino-like hardware IO
12. Advanced API for hardware IO, which can dramatically reduce the redundant work for configuring and manipulating hardware.
13. Code like arduino, but interactively in Lua script.

14. Nodejs style network API
15. Event-driven API for network applicaitons, which faciliates developers writing code running on a 5mm\*5mm sized MCU in Nodejs style.
16. Greatly speed up your IOT application developing process.
17. Lowest cost WI-FI
18. Less than \$2 WI-FI MCU ESP8266 integrated and esay to prototyping development kit.
19. We provide the best platform for IOT application development at the lowest cost.

As mentioned above, a cable supporting micro USB port is used to connect the board. As you connect the board with a computer, LED will flash. You may need some drivers to be installed on your computer if it fails to detect the NodeMCU board. You can download the driver from [this](#) page.

**Note:** We use Arduino IDE software for programming this module. It is important to note that the pin configuration appearing on the board is different from the configuration we use to program the board on the software i.e. when we write code for targeting pin 16 on the Arduino IDE, it will actually help is laying out the communication with the D0 pin on the module.

Following figure the shows the pin configuration to use in Arduino IDE.



## How to Power NodeMCU V3

We can see from the pinout image above, there are five ground pins and three 3V3 pins on the board. The board can be powered up using the following three ways.

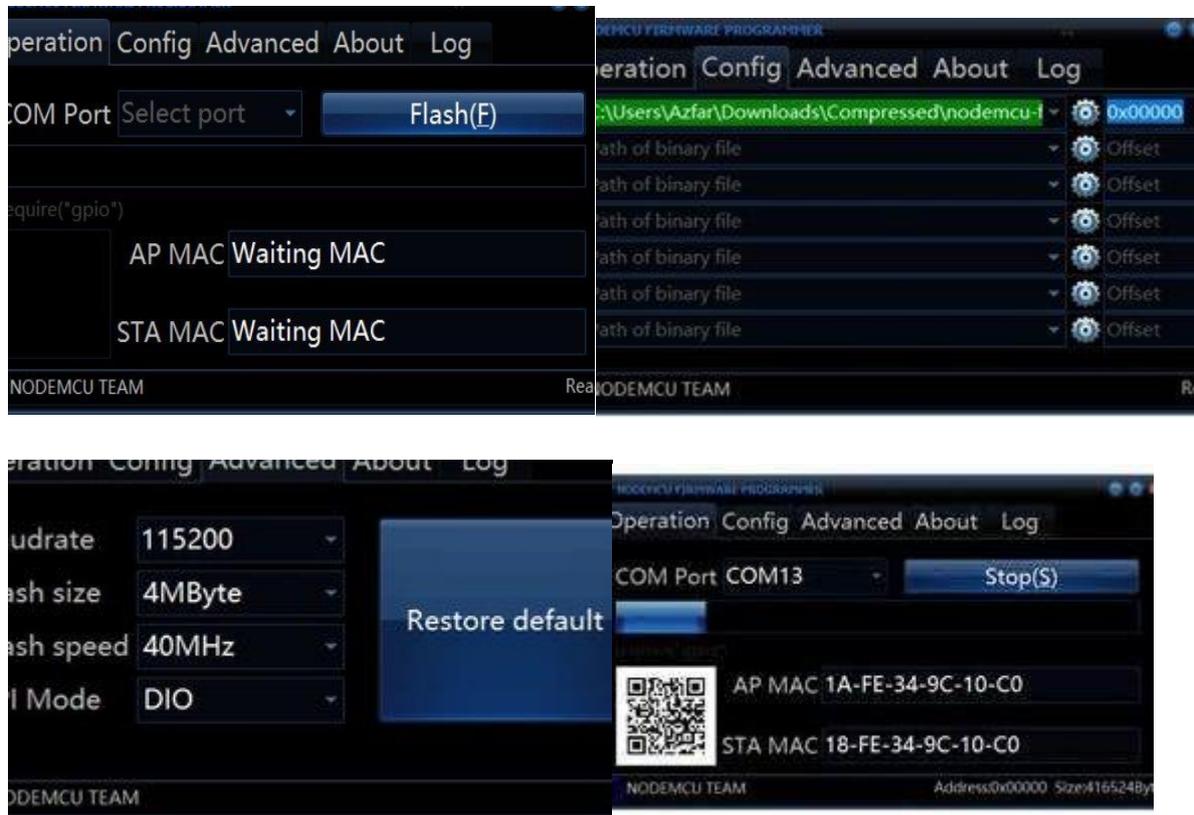
**USB Power.** It proves to an ideal choice for loading programs unless the project you aim to design requires separate interface i.e. disconnected from the computer.

**Provide 3.3V.** This is another great option to power up the module. If you have your own off-board regulator, you can generate an instant power source for your development kit.

**Power Vin.** This is a voltage regulator that comes with the ability to support up to 800 mA. It can handle somewhere between 7 to 12 V. You cannot power the devices operating at 3.3 V, as this regulator unable to generate as low as 3.3V.

## Programming

### Step 1: Installing the Firmware



In NodeMCU Boards the first thing you need is to install the Firmware to the board

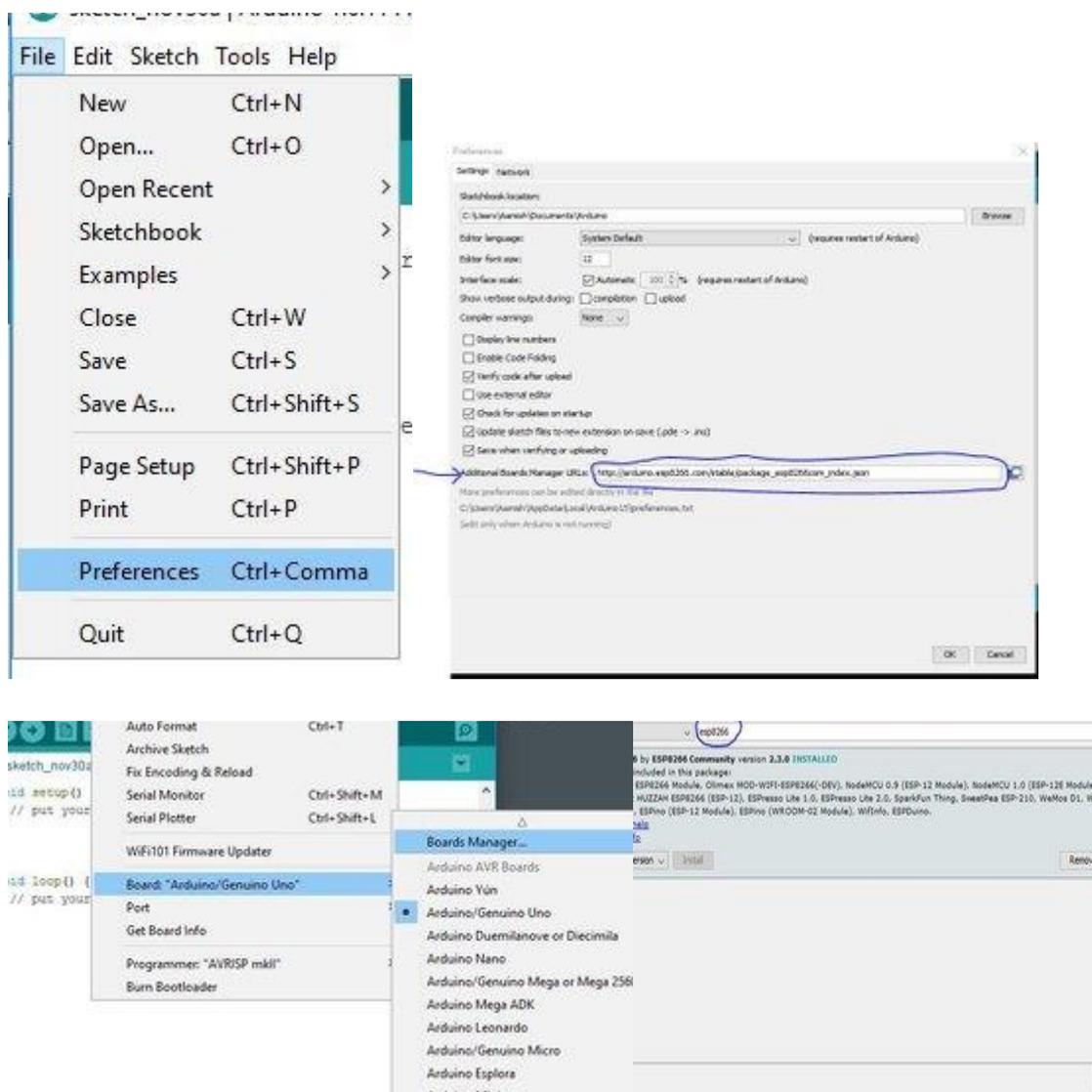
the following method works for all NodeMCU Boards

1. Open the NodeMCU flasher master folder than open the win32/win64 folder as your computer. now open the folder Release than double click ESP8266Flasher.

2. Select the COM Port.
3. Goto config tab
4. click on the small gear and open up the firmware which you have downloaded
5. go to the advanced tab and select the desired Baudrate
6. Goto the Operation tab and click on Flash Button.

Add TipAsk QuestionCommentDownload

## Step 2: Preparing the Arduino IDE



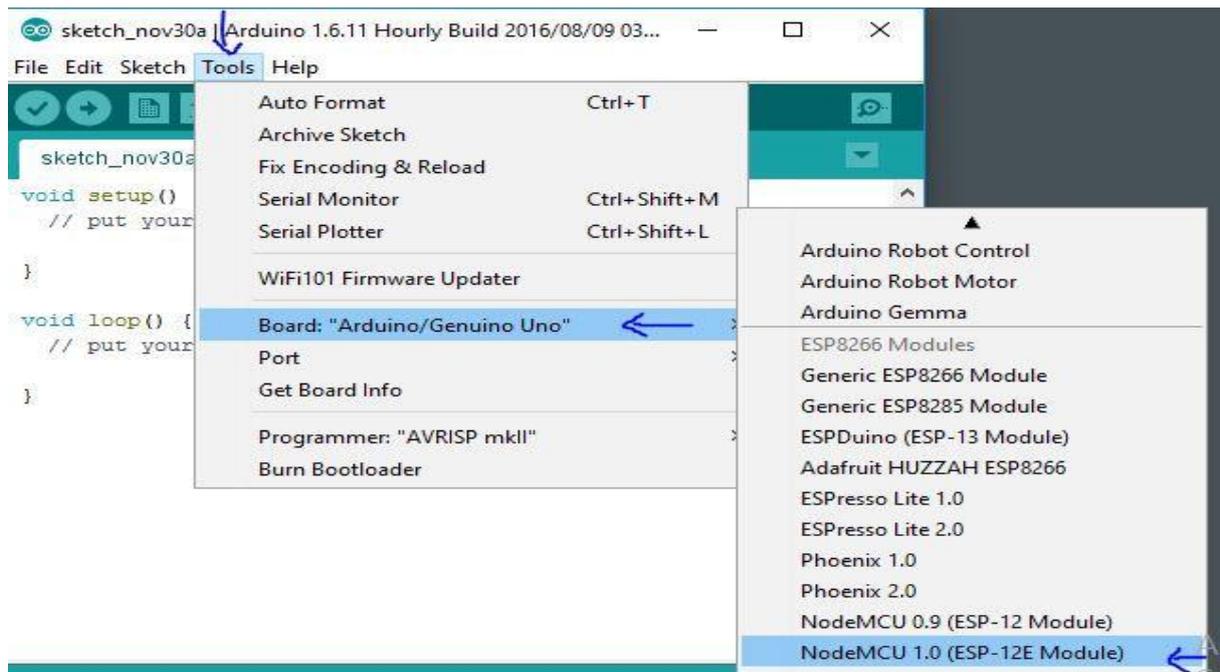
After Installing the firmware you are ready to do the programming with the ESP8266

1. Install the Arduino IDE
2. open the Arduino IDE from the desktop icon
3. Click on File tab and then open preferences
4. In the additional Boards Manager URLs add the following link ([http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)) and click OK
5. Goto Tools>Boards>Boards Manager
6. In the search field type esp8266 click the esp8266 by ESP8266 Community option and click Install

### Step 3: Code...

Now we can do whatever you want with your NodeMCU board  
Following is an example for led blinking with NodeMCU board via webserver

- In arduino IDE goto tools>Boards>select NODEMCU 1.0 (ESP - 12E Module)
- again goto tools and select port.
- Change the Wifi name and password from the following code.
- Now click on Upload button to upload the following code.
- Connect the led's positive leg on D9 pin of board and negative to the ground of the code.
- Power up the board and open the serial monitor from arduino IDE
- after connecting to the wifi it will show you the IP address.
- type that IP address on the web browser(Edge, Chrome, Firefox etc..)
- A webpage will open you can change the status of LED by turning it ON or OFF.



## Copy, Paste and Upload the Tutorial Sketch

The sketch is one that comes as an example from ESP8266.COM.

```
#include "ESP8266WiFi.h"

void setup() {
  Serial.begin(115200);
  // Set WiFi to station mode and disconnect from an AP if it was
  // previously connected
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(2000);
  Serial.println("Setup done");
}

void loop() {
  Serial.println("scan start");

  int n = WiFi.scanNetworks(); // WiFi.scanNetworks will return the
  // number of networks found
  Serial.println("scan done");
  if (n == 0)
    Serial.println("no networks found");
  else
  {
    Serial.print(n);
    Serial.println(" networks found");
    for (int i = 0; i < n; ++i)
    {
      // Print SSID and RSSI for each network found
    }
  }
}
```

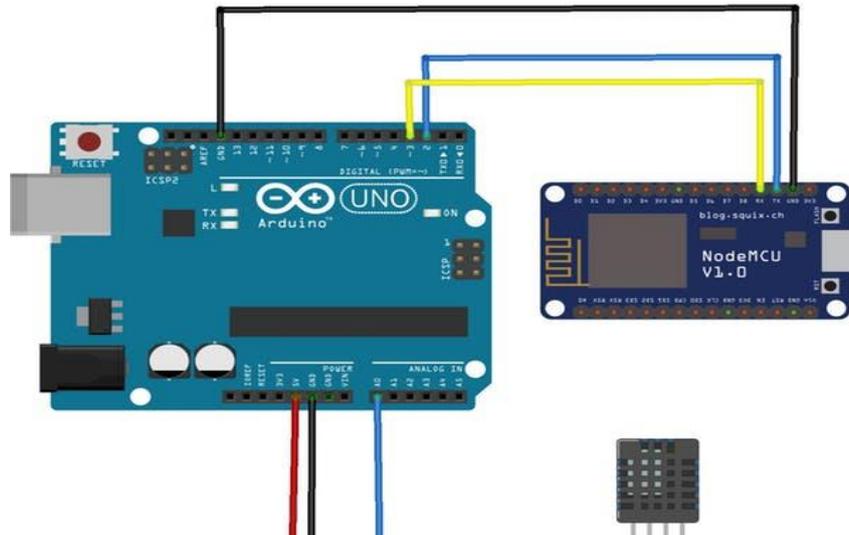
```

Serial.print(i + 1);
Serial.print(": ");
Serial.print(WiFi.SSID(i));
Serial.print(" ");
Serial.print(WiFi.RSSI(i));
Serial.print(" ");
Serial.println((WiFi.encryptionType(i) == ENC_TYPE_NONE)?" ":
"*");
    delay(10);
}
}
Serial.println("");

// Wait a bit before scanning again
delay(5000);
}

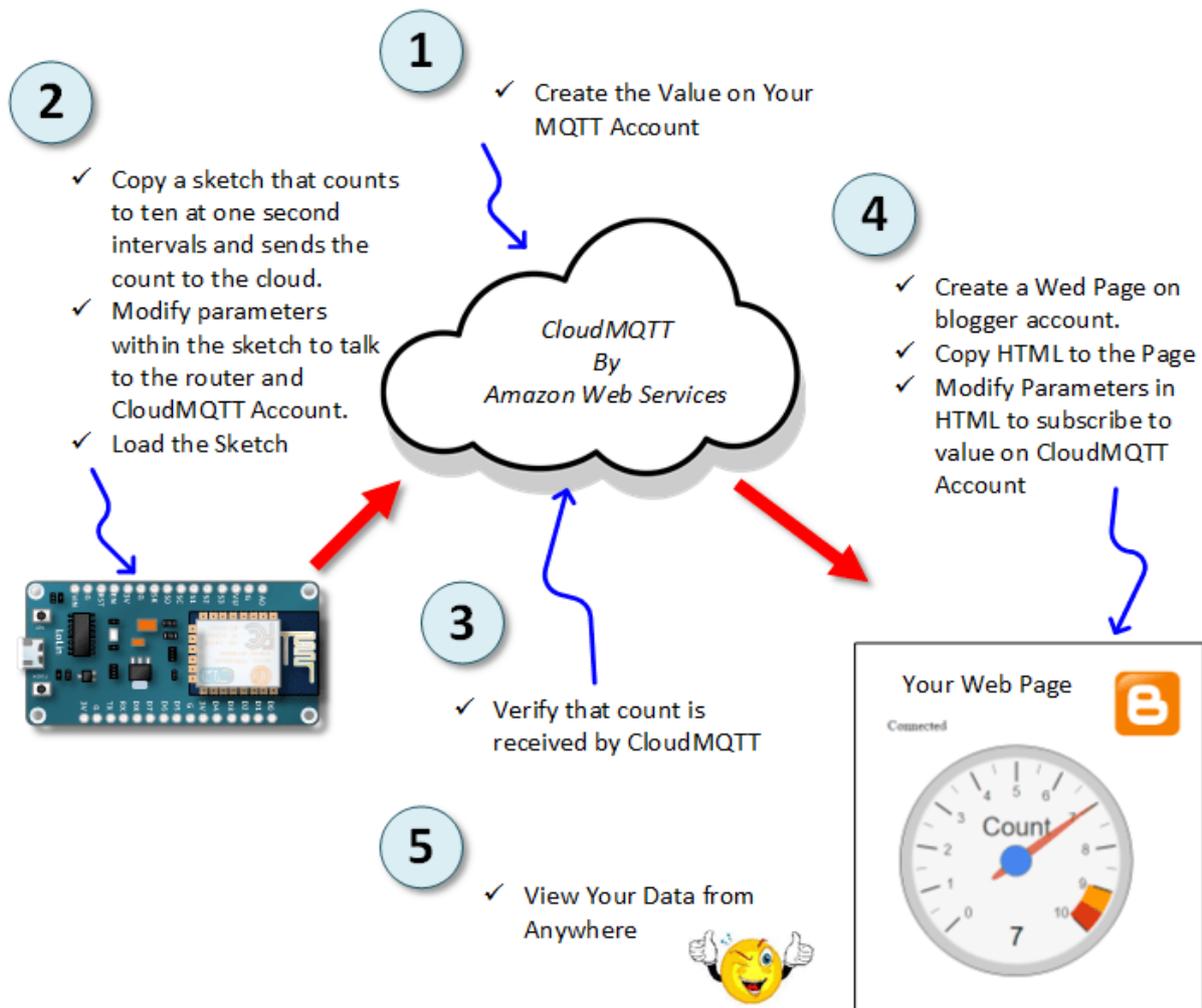
```

## Serial Communication Between NodeMCU and Arduino



### Projects and Applications

NodeMCU V3 is mainly used in the WiFi Applications which most of the other embedded modules fail to process unless incorporated with some external WiFi protocol. Following are some major applications used for NodeMCU V3.



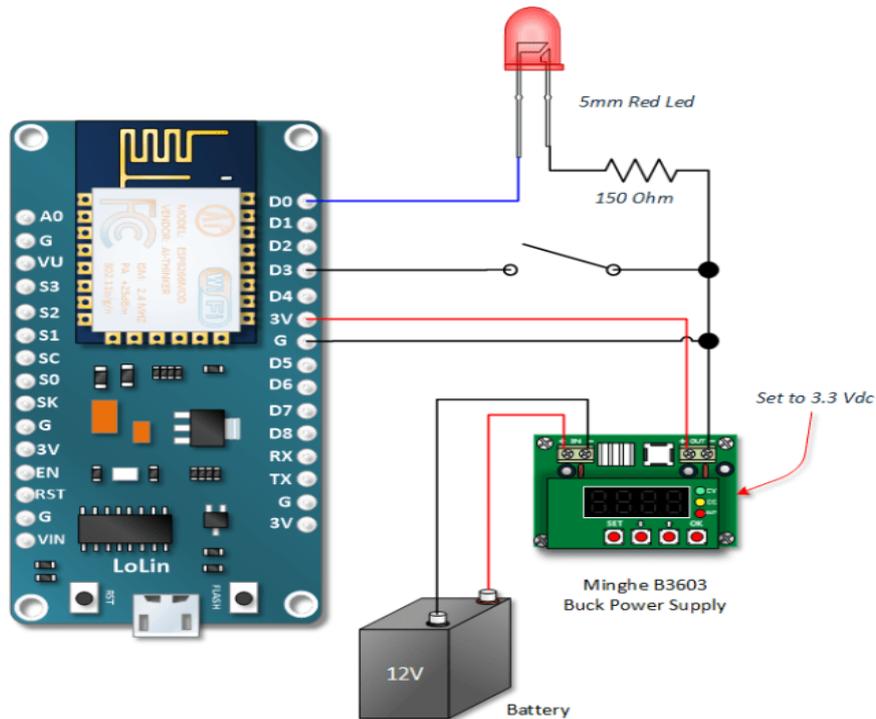
- Internet Smoked Alarm
- VR Tracker
- Octopod
- Serial Port Monitor
- ESP Lamp
- Incubator Controller
- IoT home automation
- Security Alarms

### Application Example:

#### NodeMCU ESP-12E Arduino IDE Digital Input Tutorial

#### Connect the Circuit

You may wish to power your NodeMCU another way. You can read about it [HERE](#).



## Copy, Paste and Upload the Code

The code is real simple works as follows:

- The input is read from switch pin If switch is closed, it will read a low.
- The LED output pin is set to the opposite of the switch pin. If the switch is closed, the output pin will be set to a high. A high will turn on the LED.

```
int switchPin = 0;
int LedPin = 16;
int switchValue;

void setup() {
  pinMode(LedPin, OUTPUT);
  pinMode(switchPin, INPUT_PULLUP);
}

void loop() {
  // Read the switch value
  switchValue = digitalRead(switchPin);
  /* Set the LED output pin the opposit of what is read on the sw
  itch
  * input pin
  */
  digitalWrite(LedPin, !switchValue);
}
```

## References

1. <https://www.instructables.com/id/Getting-Started-With-ESP8266LiLon-NodeMCU-V3Flashi/>
2. <https://www.c-sharpcorner.com/article/blinking-led-by-esp-12e-nodemcu-v3-module-using-arduinoide/>
3. <https://www.cytron.io/p-nodemcu-v3>
4. <http://henrysbench.capnfatz.com/henrys-bench/arduino-projects-tips-and-more/arduino-esp8266-lolin-nodemcu-getting-started/>
5. <http://henrysbench.capnfatz.com/henrys-bench/arduino-projects-tips-and-more/nodemcu-esp-12e-use-cloudmtt-to-display-on-a-gauge-on-blogger/>
6. <http://henrysbench.capnfatz.com/henrys-bench/arduino-projects-tips-and-more/arduino-ide-nodemcu-esp-12e-digital-input-basics/>
7. <http://henrysbench.capnfatz.com/henrys-bench/arduino-projects-tips-and-more/nodemcu-io-basics-pwm/>
8. [www.handsontec.com/pdf\\_learn/esp8266-V10.pdf](http://www.handsontec.com/pdf_learn/esp8266-V10.pdf)
9. <https://einstronic.com/wp-content/uploads/2017/06/NodeMCU-ESP8266-ESP-12E-Catalogue.pdf>
10. <https://airccj.org/CSCP/vol7/csit77405.pdf>
11. R.Aravindhan, M.Ramanathan, D.SanjaiKumar, R.Kishore, "HOME AUTOMATION USING Wi-Fi INTERCONNECTION", Department of Mechatronics engineering, Chennai Institute of Technology, Anna University Chennai, Tamil Nadu, India- 600069, (IRJET) e-ISSN: 2395 -0056 Volume: 04 Issue: 03 | Mar -2017 ,www.irjet.net.2542-2544pp.
12. [https://www.researchgate.net/publication/319253026\\_Counter\\_with\\_ESP8266\\_NodeMcu\\_and\\_Blynk](https://www.researchgate.net/publication/319253026_Counter_with_ESP8266_NodeMcu_and_Blynk)
13. <https://www.elecrow.com/download/ESP8266%20NodeMCU%20kit.pdf>