



Introduction - Blynk - Firebase

# NodeMCU

Eueung Mulyana

<https://eueung.github.io/012017/nodemcu>

CodeLabs | Attribution-ShareAlike CC BY-SA

---

NodeMCU DEVKIT V1.0

Firebase Version: 3.6.4 | Blynk Library v0.4.4

Arduino IDE 1.8.1

## Outline

Introduction

Using Arduino IDE

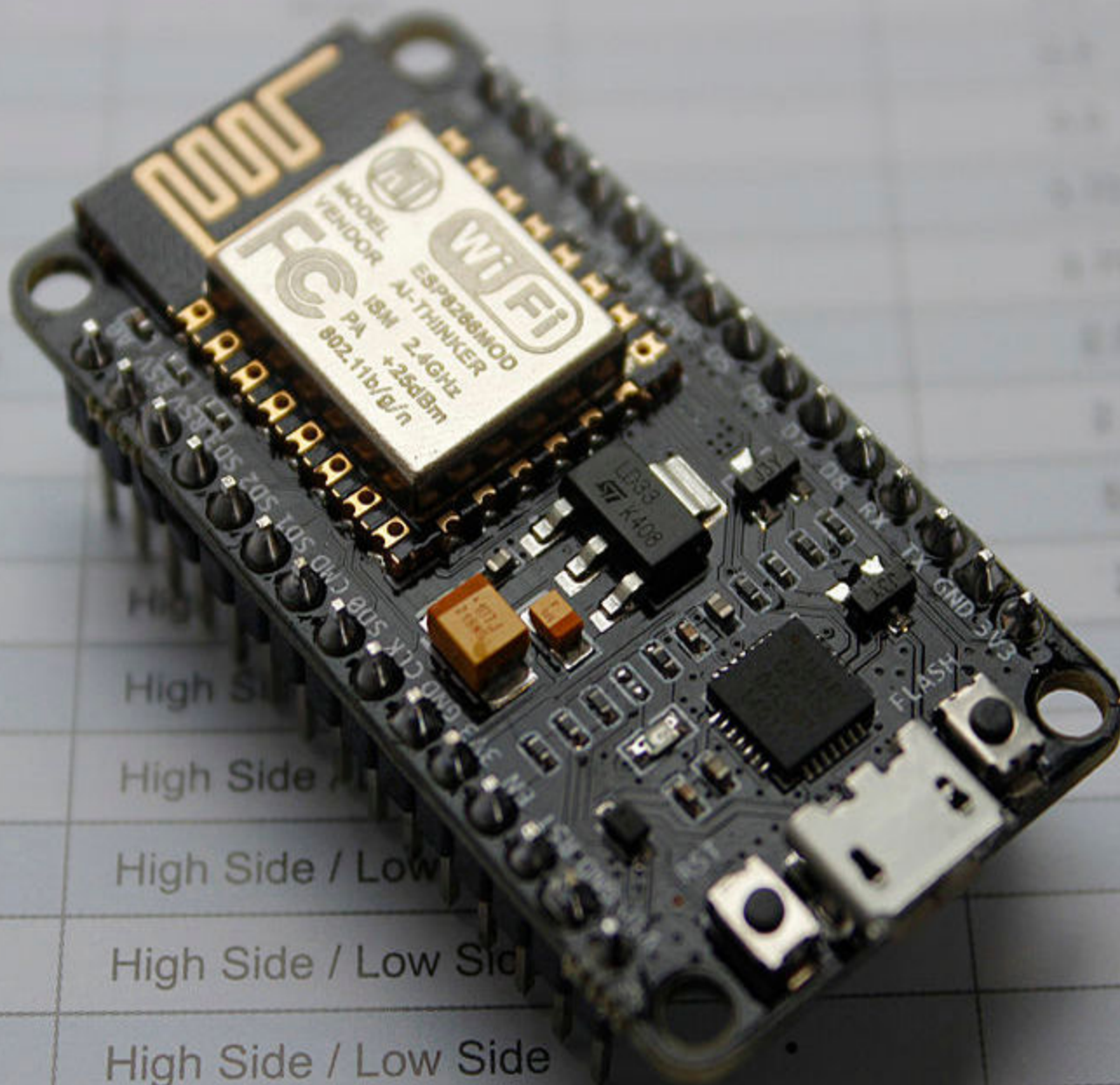
NodeMCU + Blynk

NodeMCU + Firebase



# Introduction





# NodeMCU

NodeMCU is a **LUA** based interactive **firmware** for Expressif **ESP8622** Wi-Fi SoC, as well as an open source hardware **board** that includes a (CP2102/CH340/..) TTL to USB chip for programming and debugging, is breadboard-friendly, and can simply be powered via its micro USB port.

Ref: [cnx-software](#)

# NodeMCU

## Firmware + DevKits

NodeMCU is an eLua based **firmware** for the ESP8266 WiFi SOC from Espressif. The firmware is based on the Espressif **NON-OS SDK** and uses a file system based on spiffs.

The NodeMCU firmware is a companion project to the popular NodeMCU **DevKits**, ready-made open-source **development boards** with ESP8266-12E chips.

The NodeMCU programming model is similar to that of Node.js, only in **Lua**. It is asynchronous and event-driven. Many functions, therefore, have parameters for callback functions.

Ref: [NodeMCU Documentation](#)

# NodeMCU

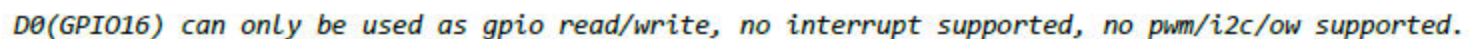
DevKit v1.0

1. **ESP-12E** WiFi Module (**16** GPIOs)
2. **4 MB** (32 Mb) Flash Storage
3. 128 KB Memory
4. **Micro USB** (**5V** Power, Program, Debug)
5. 2x15-Pin Headers (GPIOs, SPI, UART, ADC, Power)

Notes: Pins still at **3.3V**!



# V1.0 Pin Map





# Arduino IDE

# Using Arduino IDE

## Develop Sketches and Flash NodeMCU with Arduino IDE (C-Style)!

Starting with 1.6.4, Arduino allows installation of third-party platform packages using Boards Manager.

1. Install Arduino IDE
2. Start Arduino and open [Preferences](#) window
3. Enter board URL into [Additional BM URLs](#)
4. Open [Boards Manager](#) and install **esp8266**
5. Select your NodeMCU board

[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

- New Ctrl+N
- Open... Ctrl+O
- Open Recent ▸
- Sketchbook ▸
- Examples ▸
- Close Ctrl+W
- Save Ctrl+S
- Save As... Ctrl+Shift+S
- Page Setup Ctrl+Shift+P
- Print Ctrl+P
- Preferences Ctrl+Comma**
- Quit Ctrl+Q

to run once:

to run repeatedly:

## #2 | File | Preferences

**Preferences**

Settings Network

Sketchbook location:

Editor language:  (requires restart of Arduino)

Editor font size:

Interface scale: ☒ Automatic  % (requires restart of Arduino)

Show verbose output during: ☐ compilation ☐ upload

Compiler warnings:

☐ 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:

More preferences can be edited directly in the file  
/home/em/.arduino15/preferences.txt  
(edit only when Arduino is not running)

OK Cancel

# #3 | File | Preferences | BM URLs



sketch\_mar04a

```
void setup() {
  // put your set
```

```
void loop() {
  // put your mai
```

- Auto Format Ctrl+T
- Archive Sketch
- Fix Encoding & Reload
- Serial Monitor Ctrl+Shift+M
- Serial Plotter Ctrl+Shift+L
- WiFi101 Firmware Updater
- Board: "NodeMCU 1.0 (ESP-12E Module)"**
- CPU Frequency: "80 MHz"
- Flash Size: "4M (3M SPIFFS)"
- Upload Speed: "115200"
- Port
- Get Board Info
- Programmer: "AVRISP mkII"
- Burn Bootloader

**Boards Manager...**

- Arduino AVR Boards
- Arduino Yún
- Arduino/Genuino Uno
- Arduino Duemilanove or Diecimila
- Arduino Nano
- Arduino/Genuino Mega or Mega 2560
- Arduino Mega ADK
- Arduino Leonardo
- Arduino Leonardo ETH
- Arduino/Genuino Micro
- Arduino Esplora
- Arduino Mini
- Arduino Ethernet
- Arduino Fio
- Arduino BT
- LilyPad Arduino USB
- LilyPad Arduino
- Arduino Pro or Pro Mini
- Arduino NG or older
- Arduino Robot Control
- Arduino Gemma

**#4 | Tools | Board | Boards Manager**





sketch\_mar04a

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

void loop() {
  // put your
```

Type

All

esp

**Arduino AVR Boards** Built-In by **Arduino** version **1.6.17** **INSTALLED**

Boards included in this package:

Arduino Yún, Arduino/Genuino Uno, Arduino Uno WiFi, Arduino Diecimila, Arduino Nano, Arduino/Genuino Mega, Arduino MegaADK, Arduino Leonardo, Arduino Leonardo Ethernet, Arduino/Genuino Micro, Arduino Esplora, Arduino Mini, Arduino Ethernet, Arduino Fio, Arduino BT, Arduino LilyPadUSB, Arduino Lilypad, Arduino Pro, Arduino ATmegaNG, Arduino Robot Control, Arduino Robot Motor, Arduino Gemma, Adafruit Circuit Playground, Arduino Yún Mini, Arduino Industrial 101, Linino One.

[Online help](#)

[More info](#)

**esp8266** by **ESP8266 Community**

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](#)

2.3.0

▼

Install

Close

#4 | esp8266



sketch\_mar04a s

```
void setup() {
  // put your set
```

```
void loop() {
  // put your mai
```

- Auto Format Ctrl+T
- Archive Sketch
- Fix Encoding & Reload
- Serial Monitor Ctrl+Shift+M
- Serial Plotter Ctrl+Shift+L
- WiFi101 Firmware Updater
- Board: "NodeMCU 1.0 (ESP-12E Module)"**
- CPU Frequency: "80 MHz"
- Flash Size: "4M (3M SPIFFS)"
- Upload Speed: "115200"
- Port
- Get Board Info
- Programmer: "AVRISP mkII"
- Burn Bootloader

- Linino One
- Arduino Uno WiFi
- ESP8266 Modules
- Generic ESP8266 Module
- Generic ESP8285 Module
- ESPduino (ESP-13 Module)
- Adafruit HUZZAH ESP8266
- ESPRESSO Lite 1.0
- ESPRESSO Lite 2.0
- Phoenix 1.0
- Phoenix 2.0
- NodeMCU 0.9 (ESP-12 Module)
- NodeMCU 1.0 (ESP-12E Module)**
- Olimex MOD-WIFI-ESP8266(-DEV)
- SparkFun ESP8266 Thing
- SparkFun ESP8266 Thing Dev
- SweetPea ESP-210
- WeMos D1 R2 & mini
- WeMos D1(Retired)
- ESPino (ESP-12 Module)
- ThaiEasyElec's ESPino
- Core Development Module

#5 | Tools | Board | NodeMCU 1.0



# NodeMCU & Blynk

Using Arduino IDE

1. Configure Arduino IDE for NodeMCU (as before)
2. Install **Blynk Library**
3. Prepare/Configure **Blynk App**
4. Prepare the Board, Create the **Sketch** & Flash
5. Play the Project (Blynk App)!

## #2

# Install Blynk Library

---

```
~/Arduino$ tree
```

```
.
|-- blank
|   |-- blank.ino
|-- libraries
|   |-- readme.txt
```

```
2 directories, 2 files
```

```
#-----
```

```
~/Arduino$ tree -L 2
```

```
.
|-- blank
|   |-- blank.ino
|-- libraries
|   |-- readme.txt
|   |-- Adafruit_NeoPixel
|   |-- Blynk
|   |-- BlynkESP8266_Lib
|   |-- SimpleTimer
|   |-- Time
|   |-- TinyGSM
|-- tools
|   |-- BlynkUpdater
|   |-- BlynkUsbScript
```

```
11 directories, 2 files
```

---

URL: [blynkcc/blynk-library](https://blynk.cc/blynk-library)

- Verify/Compile Ctrl+R
- Upload Ctrl+U
- Upload Using Programmer Ctrl+Shift+U
- Export compiled Binary Ctrl+Alt+S
- Show Sketch Folder Ctrl+K
- Include Library**
- Add File...

- Recommended libraries
- Adafruit Circuit Playground
- Adafruit NeoPixel
- Contributed libraries
- ArduinoOTA
- Blynk
- BlynkESP8266\_Lib**
- DNSServer
- EEPROM
- ESP8266
- ESP8266AVRISP
- ESP8266HTTPClient
- ESP8266HTTPUpdateServer
- ESP8266SSDP
- ESP8266WebServer
- ESP8266WiFi
- ESP8266WiFiMesh
- ESP8266httpUpdate
- ESP8266mDNS
- Ethernet(esp8266)
- GDBStub
- GSM
- Hash

**#2 - Check**



## Create New Project

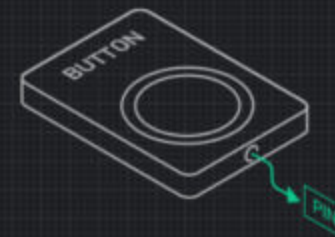
ltk

CHOOSE DEVICE

ESP8266

Create

## Button Settings



led



OUTPUT

GP12

0

1

MODE

PUSH



SWITCH

ON/OFF LABELS

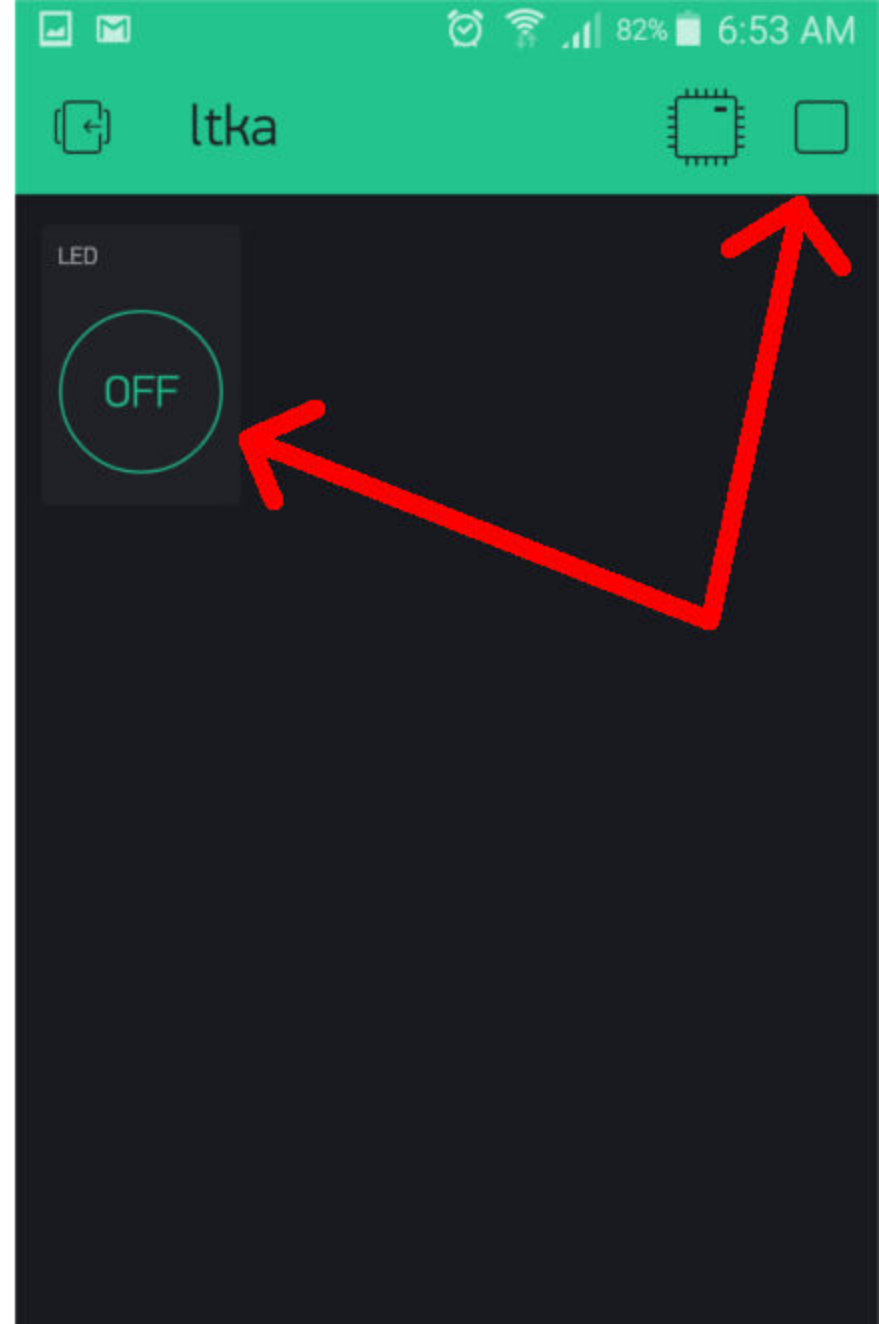
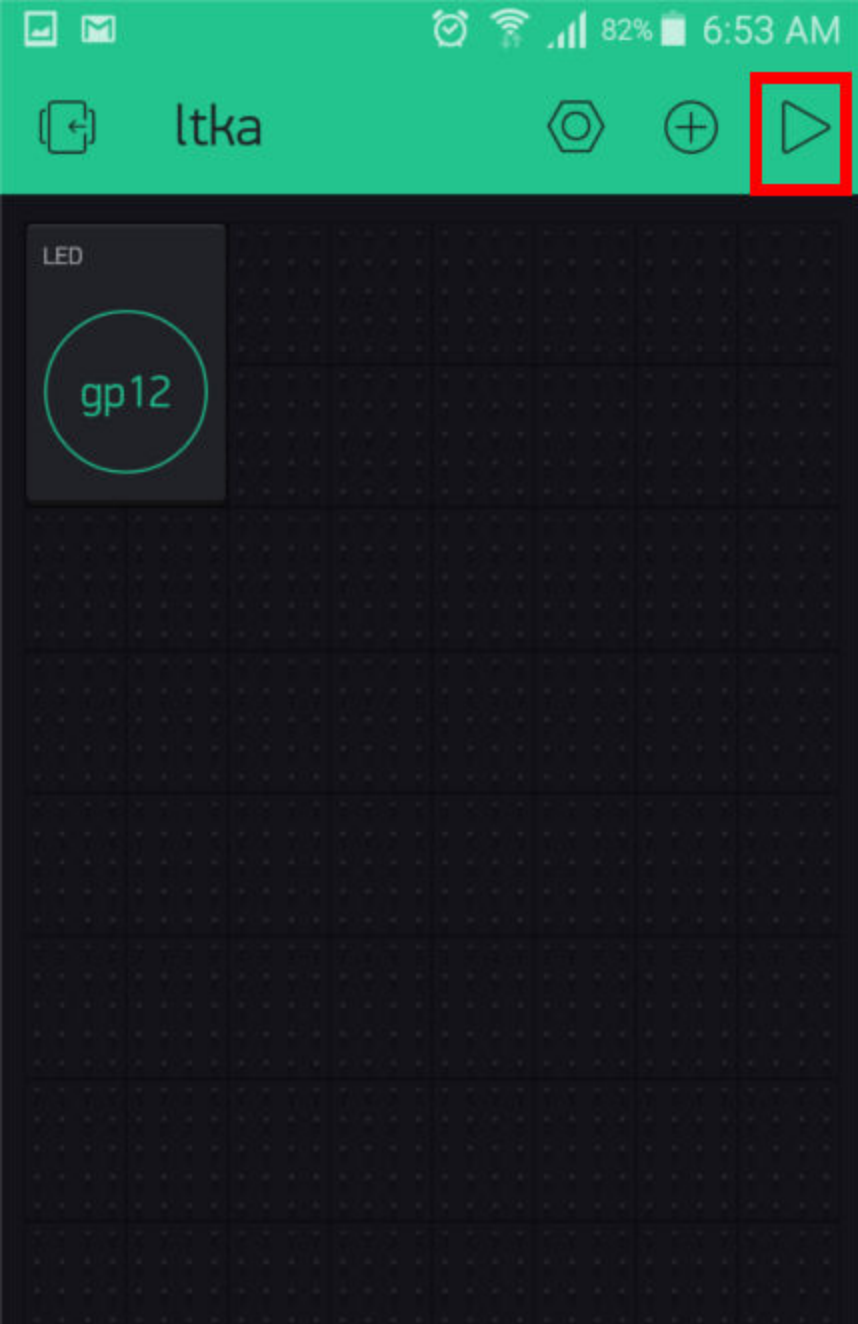
ON

OFF

ON

OFF

#3 - Configure Blynk App



**#3** - Configure Blynk App

- New Ctrl+N
- Open... Ctrl+O
- Open Recent
- Sketchbook
- Examples
- Close Ctrl+W
- Save Ctrl+S
- Save As... Ctrl+Shift+S
- Page Setup Ctrl+Shift+P
- Print Ctrl+P
- Preferences Ctrl+Comma
- Quit Ctrl+Q

to run once:

- DNSServer
- EEPROM
- ESP8266
- ESP8266AVRISP
- ESP8266HTTPClient
- ESP8266httpUpdate
- ESP8266HTTPUpdateServer
- ESP8266mDNS
- ESP8266SSDP
- ESP8266WebServer
- ESP8266WiFi
- ESP8266WiFiMesh
- Ethernet(esp8266)
- Hash
- SD(esp8266)
- Servo(esp8266)
- SoftwareSerial
- TFT\_Touch\_Shield\_V2
- Ticker
- Examples from Custom Libraries
- Adafruit NeoPixel
- Blynk
- Time

- Boards\_Bluetooth
- Boards\_Ethernet
- Boards\_GSM
- Boards\_USB\_Serial
- Boards\_WiFi
- Boards\_With\_HTTP\_API
- Export\_Demo
- GettingStarted
- More
- Widgets
- tests

- Arduino\_MKR1000
- Arduino\_WiFi\_Shield
- Arduino\_WiFi\_Shield\_101
- Arduino\_Yun
- CC3000
- Digistump\_Oak
- Energia\_WiFi
- ESP32\_WiFi
- ESP8266\_Shield
- ESP8266\_Standalone
- ESP8266\_Standalone\_Manual\_IP
- ESP8266\_Standalone\_SmartConfig
- ESP8266\_Standalone\_SSL
- Fishino
- Intel\_Edison\_WiFi
- LinkItONE
- RedBear\_Duo\_WiFi
- RN\_XV\_WiFly
- Sparkfun\_Blynk\_Board
- TheAirBoard\_WiFly
- TinyDuino\_WiFi
- WildFire\_V3
- WildFire\_V4

# #4 - Sketch



ESP8266\_Standalone

```
* https://github.com/esp8266/Arduino
*
* Please be sure to select the right ESP8266 module
* in the Tools -> Board menu!
*
* Change WiFi ssid, pass, and Blynk auth token to run :)
*
*****/

#define BLYNK_PRINT Serial // Comment this out to disable prints and save space
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "YourAuthToken";

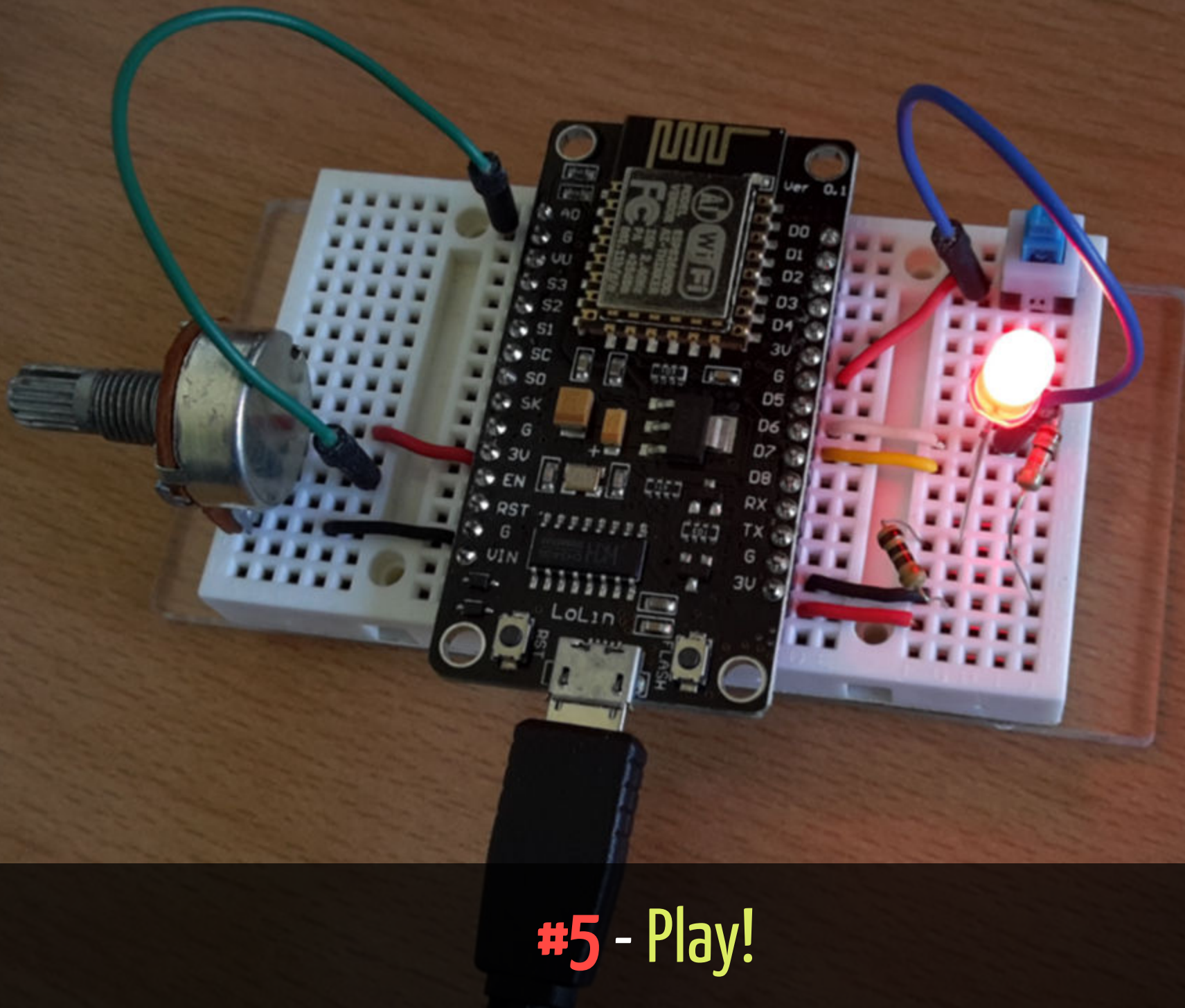
// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";

void setup()
{
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
}

void loop()
{
  Blynk.run();
}
```

## #4 - Sketch - cf. Sketch Builder





#5 - Play!

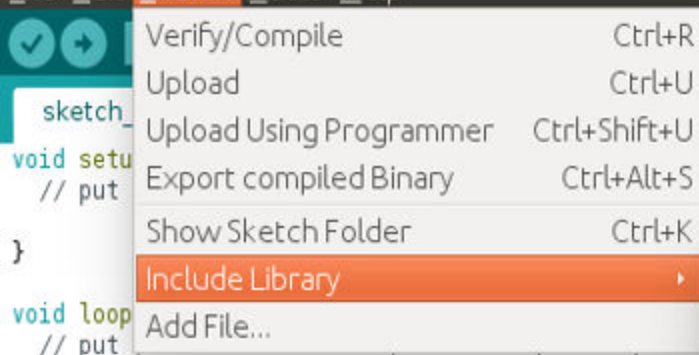




# NodeMCU & Firebase

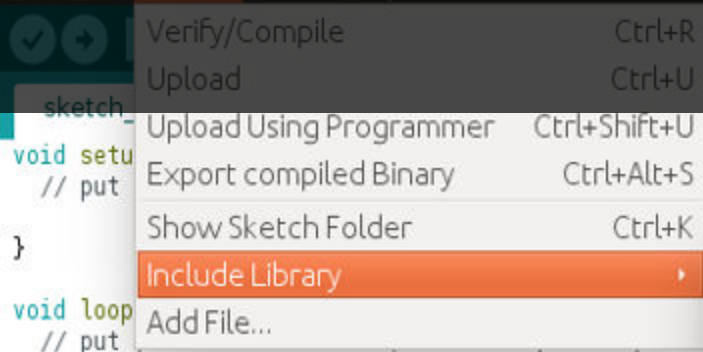
Using Arduino IDE

1. Configure Arduino IDE for NodeMCU (as before)
2. Install **FirebaseArduino Library**
3. Create a Firebase Project
4. Use Example Sketch **FirebaseRoom\_ESP8266**
5. Prepare the Circuit & Flash
6. Play!



## #2 - Install FirebaseArduino

## #2 - Check



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



## Welcome back to Firebase

Continue building your apps with Firebase. Here are some of the resources below.

[Documentation](#) <> [Sample code](#)

Your projects using Firebase

**fbase-test**

fbase-test-54a62

fbase-vue

### Create a project



Project name

Country/region ?

United States ▼

By default, your Firebase Analytics data will enhance other Firebase features and Google products. You can control how your Firebase Analytics data is shared in your settings at anytime. [Learn more](#)

CANCEL

CREATE PROJECT

IMPORT GOOGLE PROJECT

## #3 - Create Firebase Project



Overview



Analytics

DEVELOP



Authentication



Database



Storage



Hosting



Test Lab



Crash Reporting

GROW



Notifications



Remote Config



Dynamic Links

Spark

Free \$0/month

UPGRADE

## Realtime Database



DATA

RULES

USAGE

BACKUPS

<https://fbase-nodemcu.firebaseio.com/>

Default security rules require users to be authenticated

[LEARN MORE](#)[DISMISS](#)

fbase-nodemcu: null



Store and sync data in realtime across all connected clients

[Learn more](#)

# #3 - Create Firebase Project : Host



Overview



Analytics

DEVELOP



Authentication



Database



Storage



Hosting



Test Lab



Crash Reporting

GROW



Notifications



Remote Config



Dynamic Links

Spark

Free \$0/month

UPGRADE

## Settings

GENERAL

CLOUD MESSAGING

ANALYTICS

ACCOUNT LINKING

SERVICE ACCOUNTS

[Manage all service accounts](#)

Firebase Admin SDK



Crash Reporting

Legacy credentials



Database Secrets

## Database Secrets



Database secrets are currently deprecated and use a legacy Firebase token generator. Update your source code with the Firebase Admin SDK.

[LEARN MORE](#)

Create custom database authentication tokens using a legacy Firebase token generator. At least one secret must exist at all times. [Learn more](#)

fbase-nodemcu

ADD SECRET

Secrets

#3

- Create Firebase Project : DB Secret



- New Ctrl+N
- Open... Ctrl+O
- Open Recent
- Sketchbook
- Examples**
- Close Ctrl+W
- Save Ctrl+S
- Save As... Ctrl+Shift+S
- Page Setup Ctrl+Shift+P
- Print Ctrl+P
- Preferences Ctrl+Comma
- Quit Ctrl+Q

to run once:

- ESP8266
- ESP8266AVRISP
- ESP8266HTTPClient
- ESP8266httpUpdate
- ESP8266HTTPUpdateServer
- ESP8266mDNS
- ESP8266SSDP
- ESP8266WebServer
- ESP8266WiFi
- ESP8266WiFiMesh
- Ethernet(esp8266)
- Hash
- SD(esp8266)
- Servo(esp8266)
- SoftwareSerial
- TFT\_Touch\_Shield\_V2
- Ticker
- Examples from Custom Libraries
- Adafruit NeoPixel
- Blynk
- FirestoreArduino**
- Time
- TinyGPS

- contrib
- FirebaseCloudMessaging\_Send\_ESP8266
- FirebaseDemo\_ESP8266
- FirebaseNeoPixel\_ESP8266
- FirebaseRoom\_ESP8266**
- FirebaseStream\_ESP8266
- src

Library added to your libraries. C

## #4 - Example Sketch FirebaseRoom



FirebaseRoom\_ESP8266

```
// FirebaseRoom_ESP8266 is a sample that demo using multiple sensors
// and actuator with the FirebaseArduino library.
```

```
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
```

```
// Set these to run example.
#define FIREBASE_HOST "example.firebaseio.com"
#define FIREBASE_AUTH "token_or_secret"
#define WIFI_SSID "SSID"
#define WIFI_PASSWORD "PASSWORD"
```

```
const int grovePowerPin = 15;
const int vibratorPin = 5;
const int lightSensorPin = A0;
const int ledPin = 12;
const int buttonPin = 14;
const int fanPin = 13;
```

```
void setup() {
  Serial.begin(9600);

  pinMode(grovePowerPin, OUTPUT);
  digitalWrite(grovePowerPin, HIGH);
```

```
  pinMode(vibratorPin, OUTPUT);
  pinMode(lightSensorPin, INPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
  pinMode(fanPin, OUTPUT);
```

```
  // connect to wifi.
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
```

## #4 - Example Sketch FirebaseRoom



# #5

## Prepare the Circuit

- Vibration Motor (**brrr**): pin 5 (D1) - Blue LED
- Light Sensor (**sunlight**): pin A0 - Potentio
- Red LED (**redlight**): pin 12 (D6)
- Button (**pushbutton**): pin 14 (D5)
- Mini Fan (**cooldown**): pin 13 (D7) - Orange LED

```

#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>

// Set these to run example.
#define FIREBASE_HOST "fbase-nodemcu.firebaseio.com"
#define FIREBASE_AUTH "g3mvizMBfpXpoSa446BXxPiiBduXFRhZUhgzMjDL"
#define WIFI_SSID "lk8fm"
#define WIFI_PASSWORD "1234567890"

const int grovePowerPin = 15;
const int vibratorPin = 5;
const int lightSensorPin = A0;
const int ledPin = 12;
const int buttonPin = 14;
const int fanPin = 13;

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

  pinMode(grovePowerPin, OUTPUT);
  digitalWrite(grovePowerPin, HIGH);

  pinMode(vibratorPin, OUTPUT);
  pinMode(lightSensorPin, INPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
  pinMode(fanPin, OUTPUT);

  // connect to wifi.
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Firebase.set("pushbutton", 0);
  Firebase.set("sunlight", 0);
  Firebase.set("redlight", 0);

```

FirebaseRoom\_ESP8266





Overview



Analytics

DEVELOP



Authentication



Database



Storage



Hosting



Test Lab



Crash Reporting

GROW



Notifications

## Realtime Database



DATA

RULES

USAGE

BACKUPS

<https://fbase-nodemcu.firebaseio.com/>

Default security rules require users to be authenticated

[LEARN MORE](#)[DISMISS](#)

fbase-nodemcu

brrr: 0

cooldown: 0

pushbutton: 0

redlight: 1

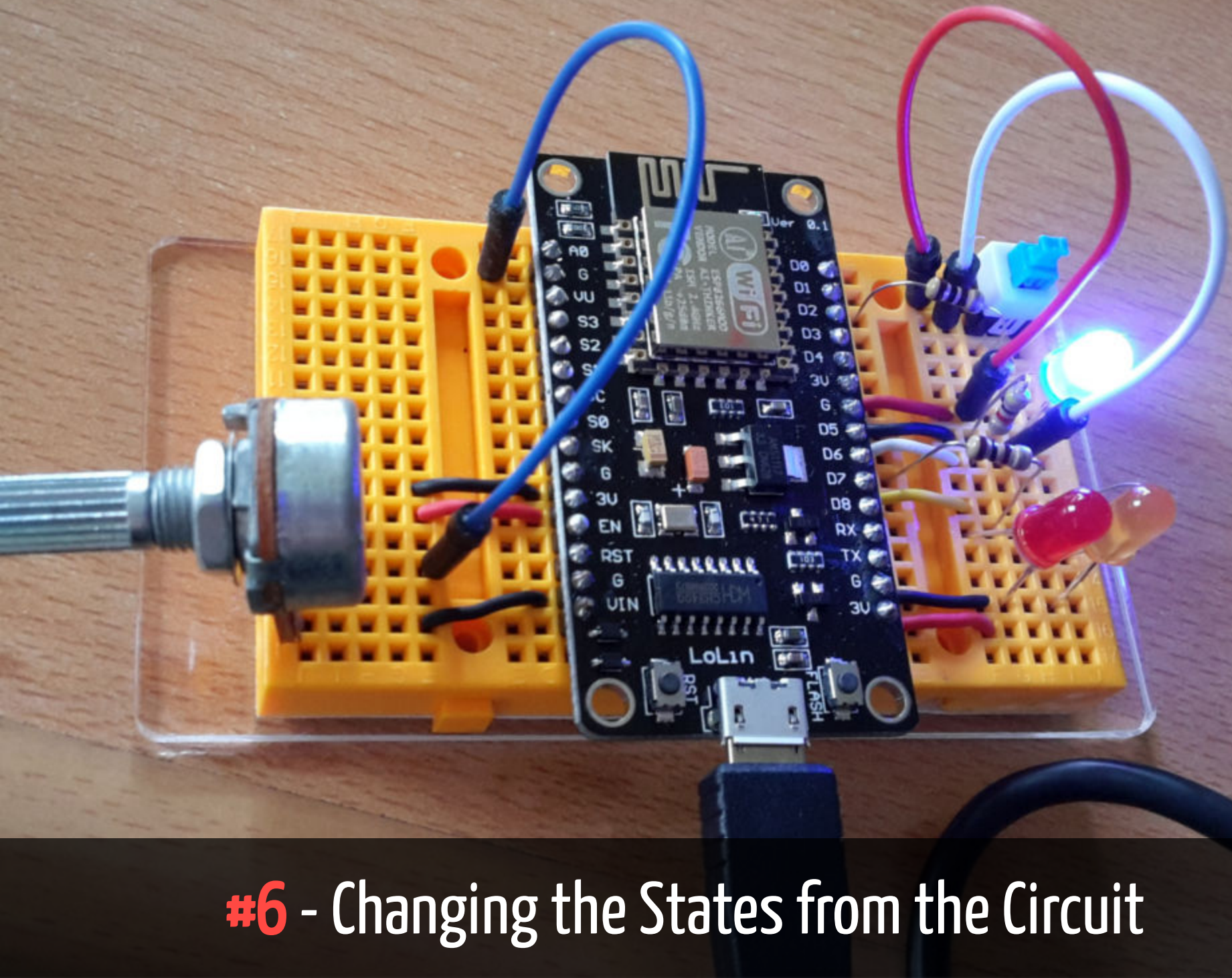
sunlight: 906

Spark

Free \$0/month

UPGRADE

## #6 - Changing the Values from Firebase



## #6 - Changing the States from the Circuit





# Refs/Resources

# Refs/Resources

1. [NodeMcu](#) - An open-source firmware based on ESP8266 wifi-soc
2. [nodemcu/nodemcu-devkit-v1.0](#)
3. [NodeMCU Documentation](#)
4. [esp8266/Arduino](#): ESP8266 core for Arduino
5. [Getting Started - Blynk](#)
6. [Simple Led Control With Blynk and NodeMCU Esp8266 12E](#)
7. [FirebaseDemo\\_ESP8266](#) - firebase/firebase-arduino
8. [NodeMCU](#) is both a Breadboard-Friendly ESP8266 Wi-Fi Board and a LUA based Firmware
9. [NodeMCU](#) - Wikipedia

# More Refs/Resources

- [nodemcu/nodemcu-firmware](#): lua based interactive firmware for mcu like esp8266
- [NodeMCU custom builds](#)
- [marcelstoer/nodemcu-build](#) - Docker Hub
- [Comparison of ESP8266 NodeMCU development boards](#)
- [Comparison of tools and IDEs for NodeMCU](#)



# END

Eueung Mulyana

<https://eueung.github.io/012017/nodemcu>

CodeLabs | Attribution-ShareAlike CC BY-SA

---