SYNACORP
Your Strategic Solutions Provider

SYNACORP TECHNOLOGIES SDN. BHD. (1310487-K)
No.25 Lorong 1/SS3, Bandar Tasek Mutiara,
14120 Simpang Ampat, Penang, Malaysia.
T: +604.586.0026 F: +604.586.0026
www.synacorp.my | Email: sales@synacorp.com.my

WEMOS D1 Mini Nodemcu Wireless Wifi IOT Arduino ESP8266 ESP-12F

Introduction :

WEMOS D1 mini NodeMcu wireless Wifi IoT module is a compatible breadboard and Arduino IDE that supported development board available in the market. It is one of the board from ESP8266 family which contains 11 digital ports and one analog pin for analog inputs. It has an on board CH340 USB-TTL convert reset button and offers 4MB flash memory, All of the Digital I/O support PWM and interrupts excepts D0. It can support much wide range of sensor by utilizing 3V/ 5V supply provided by onboard regulator. The functioning is similar to that of NodeMcu, except that the hardware is built resembling Arduino UNO. The D1 board can helped to eradicate some complex issues using wifi based chips and modules in IoT based projects.

Objectives :

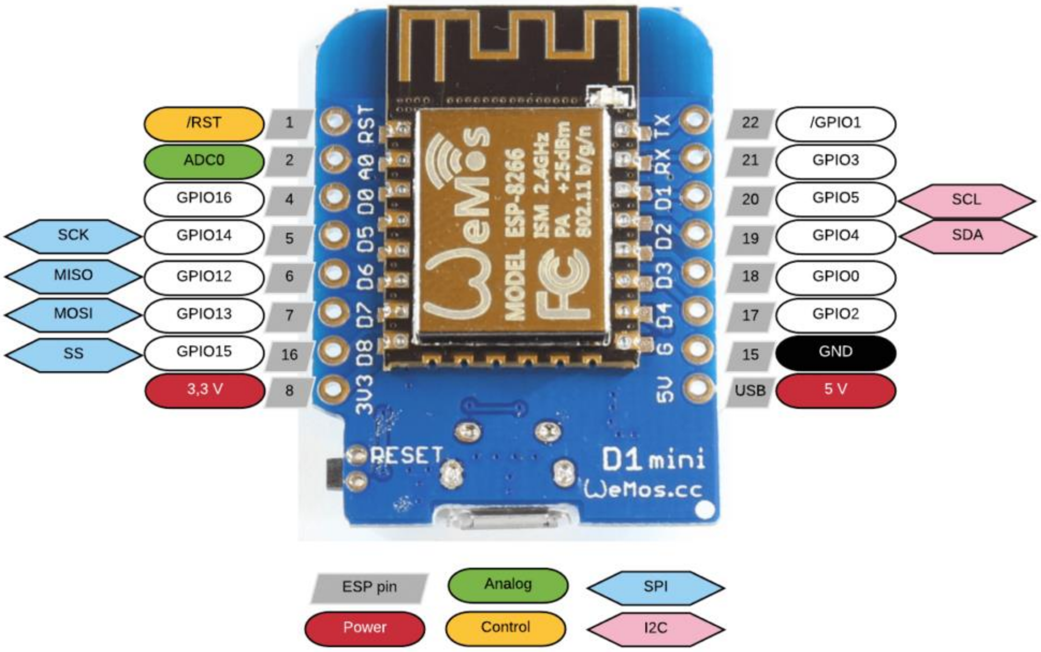
This is a tutorial for Internet of Things (IoT) to teaches how to control stuff over wifi with the Wemos mini board and blynk app. It also covers how to connect and code the Wemos D1 mini board and how to get started with Blynk app.

Specifications :

- Microcontroller: ESP 8266 32-bit
- Serial to USB Converter: CH340
- USb Connector: Micro USB
- Operating voltage: 3.3 V
- Input Voltage: 3.3V / 5V
- DC Current per pin: 12mA(Max)
- Flash Memory: 4MB
- Clock Speed: 80MHz
- Digital I/O: 11
- Analogue I/O: 1 (10 bit ADC, Max Input 3.2V)
- Communications: Serial, SPI., I2C interrupt/PWM and 1
- Wire via software libraries
- WIFI: Built in 802.11 b/g/n
- Has built-in LED connected to pin 4
- Compatble with Arduino IDE and NodeMCU
- Dimension: 34 x 26mm

Features :

- Based on ESP-8266
- Can be programmed with Arduino IDE
- Powered through micro usb

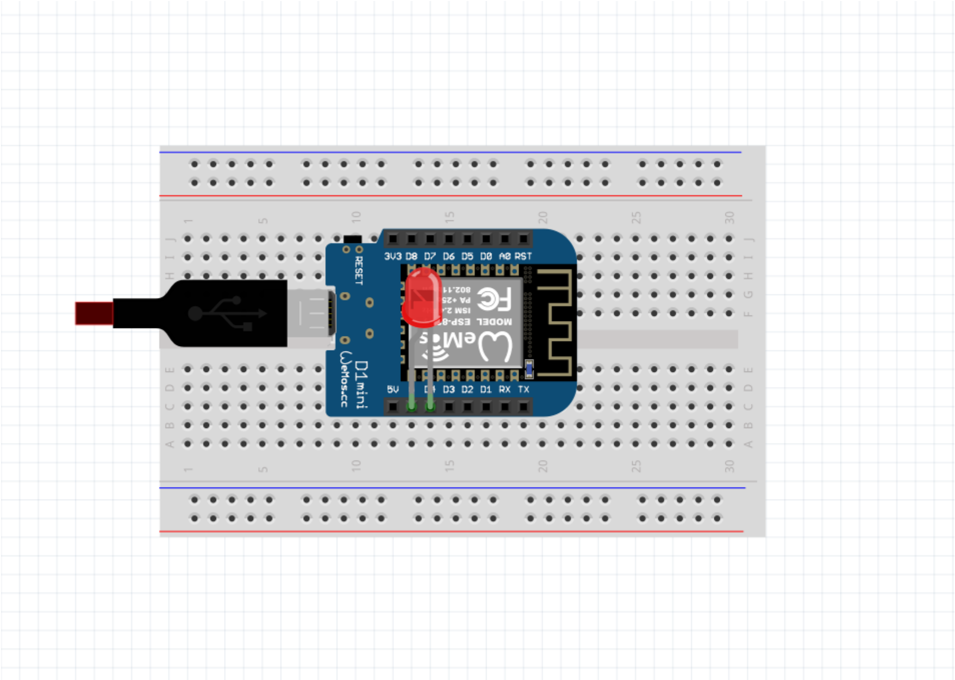


Components Needed :

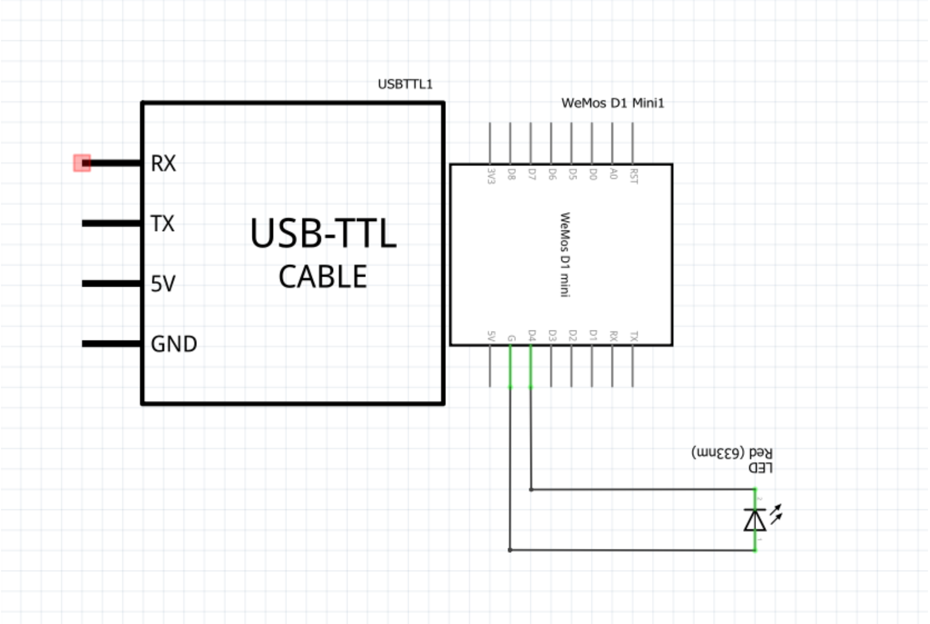
- LED
- Wemos D1 mini Nodemcu
- USB cable
- Blynk app

Procedures :

Step 1 : Connect the LED to Wemos D1 mini nodemcu based on diagram below.

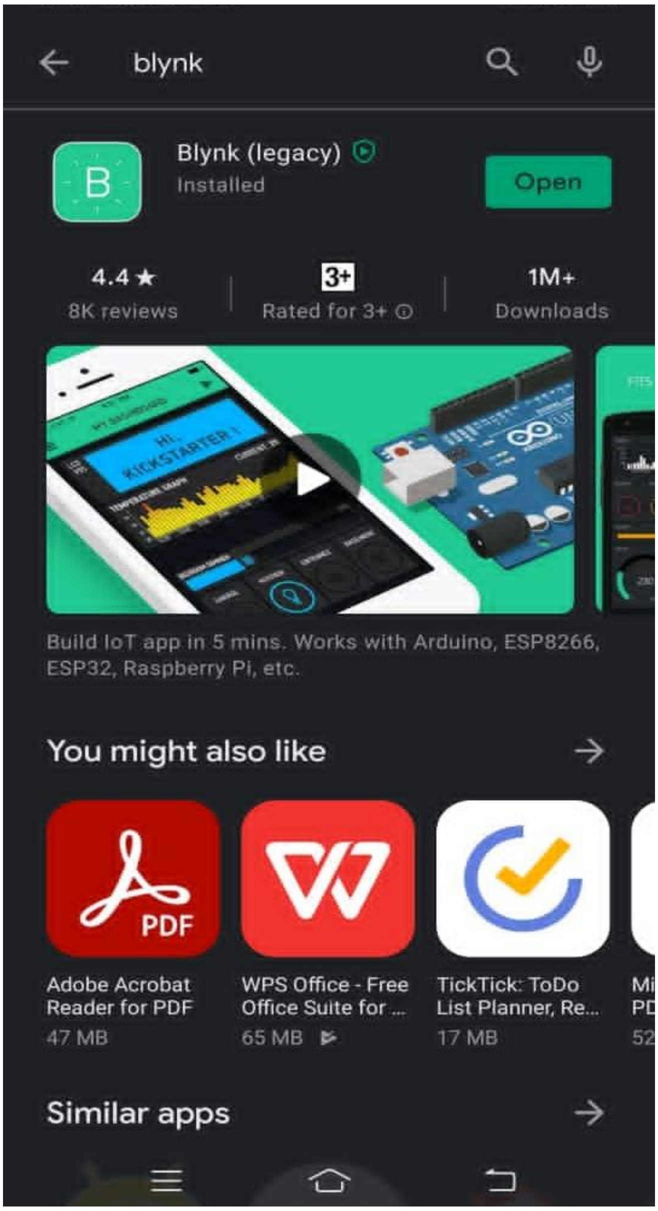


Breadboard layout

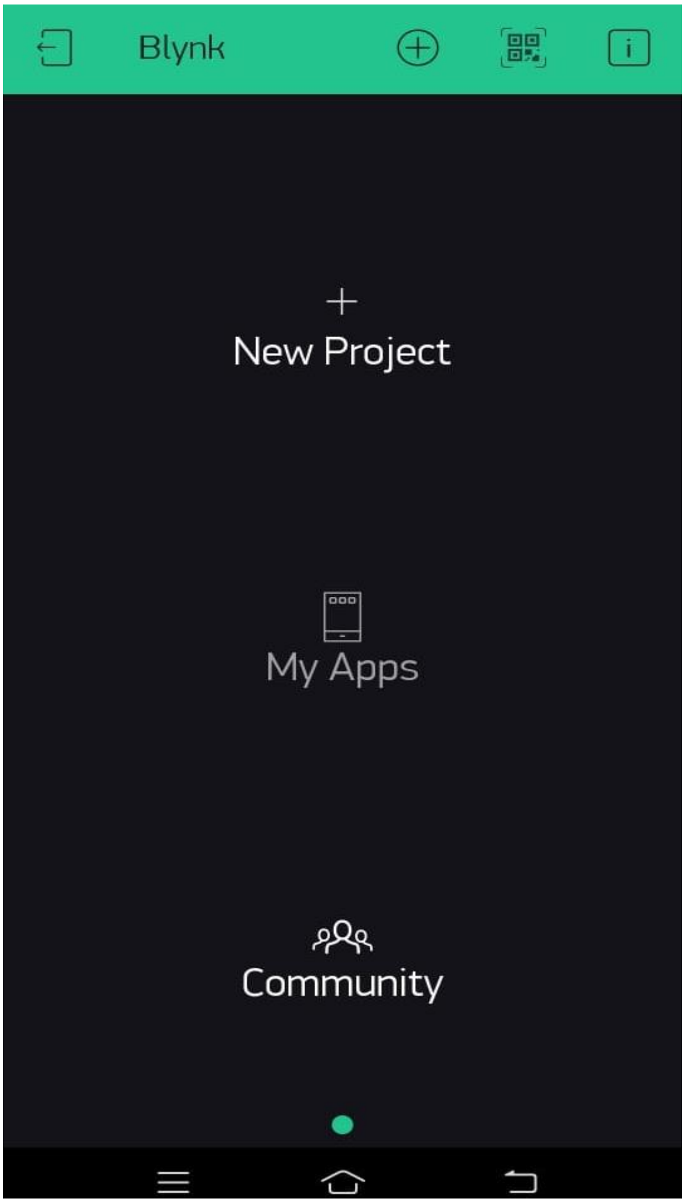


Schematic layout

Step 2 : Getting started with the ***Blynk*** **App.** Make sure you installed the application on App store or Google Play store.

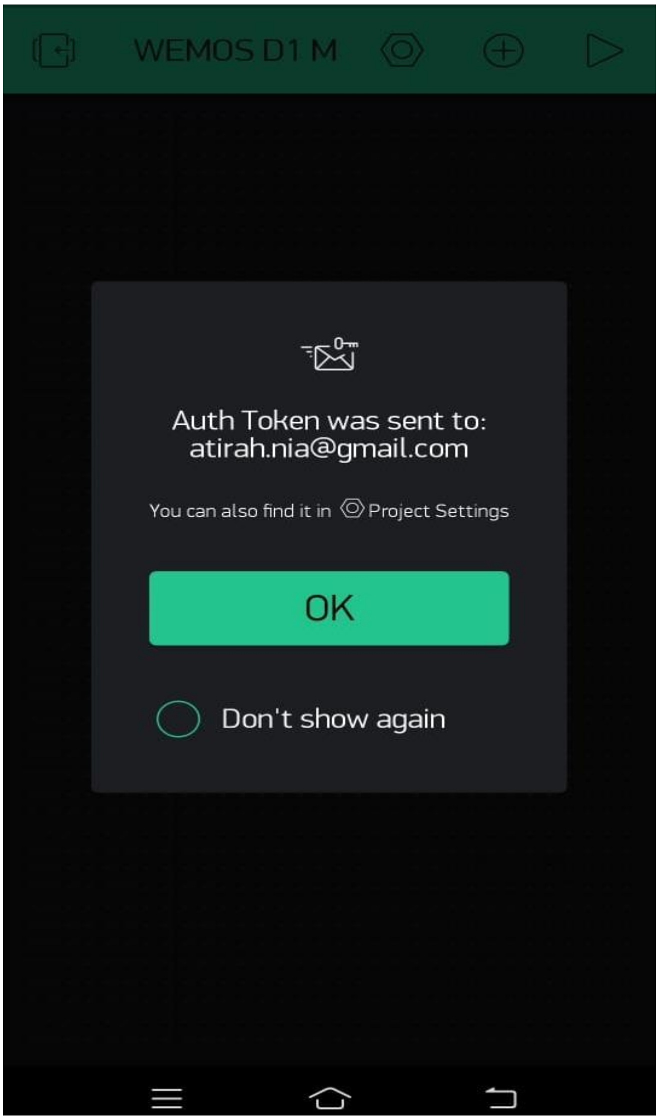

blynk
Blynk (legacy)
Installed
Open
4.4 ★
8K reviews
3+
Rated for 3+
1M+
Downloads
Build IoT app in 5 mins. Works with Arduino, ESP8266, ESP32, Raspberry Pi, etc.
You might also like
Adobe Acrobat Reader for PDF
47 MB
WPS Office - Free Office Suite for ...
65 MB
TickTick: ToDo List Planner, Re...
17 MB
Similar apps

Step 3 : Create a new poject to select the hardware model you use.



Step 4 : Authentication Token.

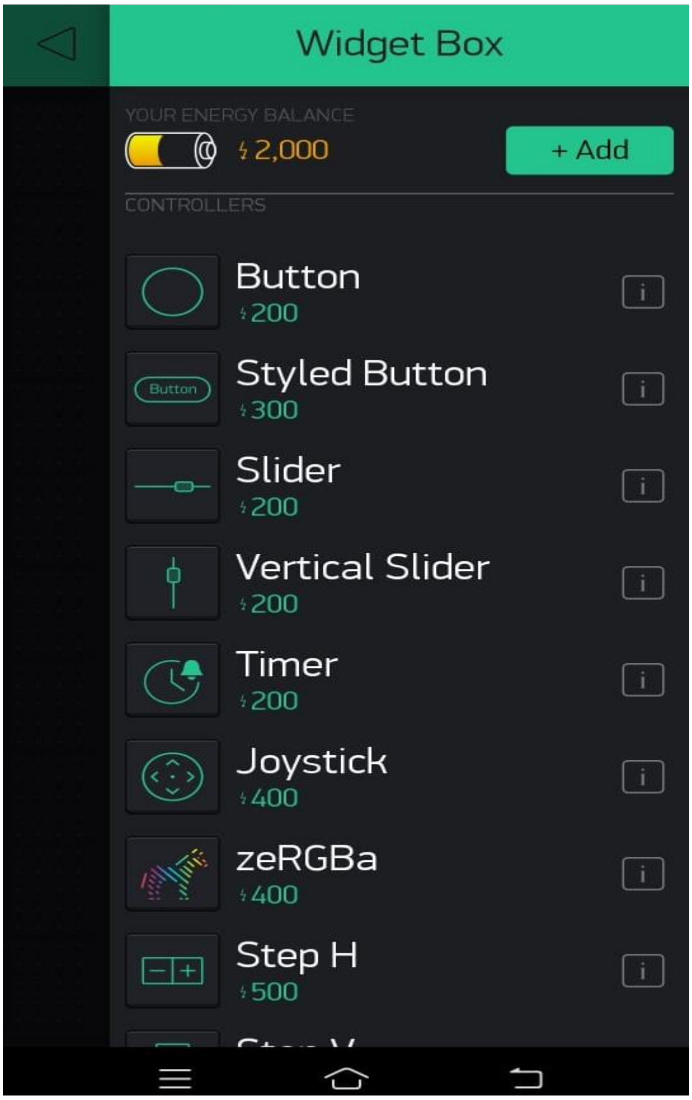
You will get Auth Token automatically on your email after project creation. You can also copy it manually. Click on devices section and selected required device.



Auth Token was sent to:
atirah.nia@gmail.com

You can also find it in Project Settings

Step 5 : Add a widget for adding a button to control the LED.



Widget Box

YOUR ENERGY BALANCE

ϟ2,000 + Add

CONTROLLERS

Button ϟ200

Styled Button ϟ300

Slider ϟ200

Vertical Slider ϟ200

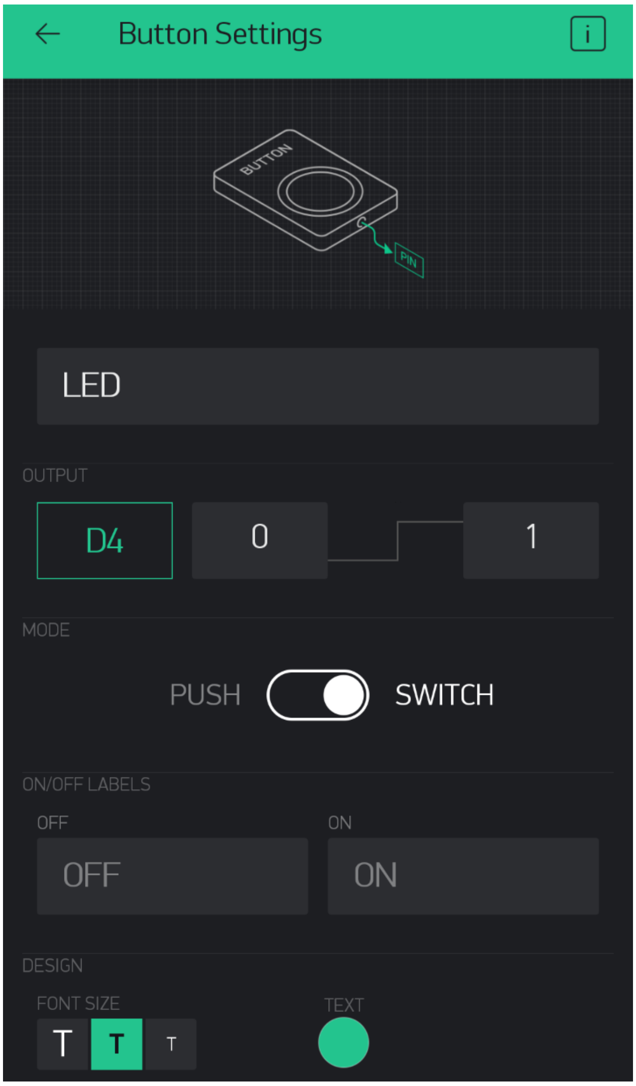
Timer ϟ200

Joystick ϟ400

zeRGBa ϟ400

Step H ϟ500

Step 6 : Set the output pin and mode at the Button Settings.



Button Settings

LED

OUTPUT

D4 0 1

MODE

PUSH SWITCH

ON/OFF LABELS

OFF: OFF

ON: ON

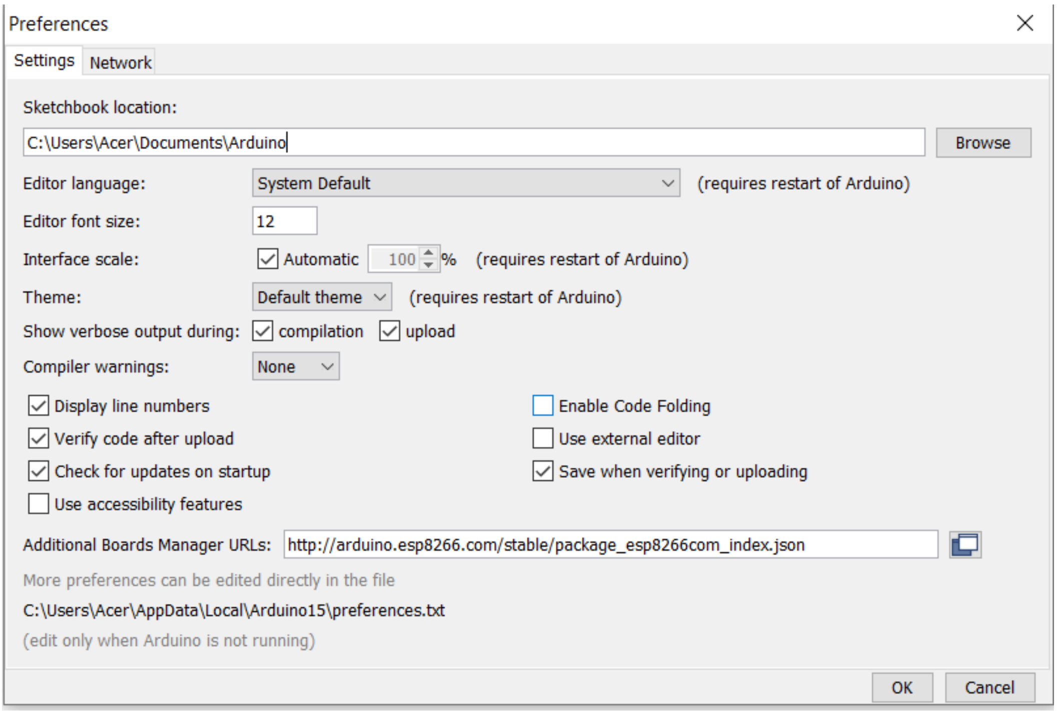
DESIGN

FONT SIZE

TEXT

Step 7: Now launch **Arduino IDE** and locate **Board Manager URLS** and copy paste the bellow url and click **OK**.

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



Preferences

Settings | Network

Sketchbook location:
C:\Users\Acer\Documents\Arduino [Browse]

Editor language: System Default (requires restart of Arduino)

Editor font size: 12

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

Theme: Default theme (requires restart of Arduino)

Show verbose output during: ☑ compilation ☑ upload

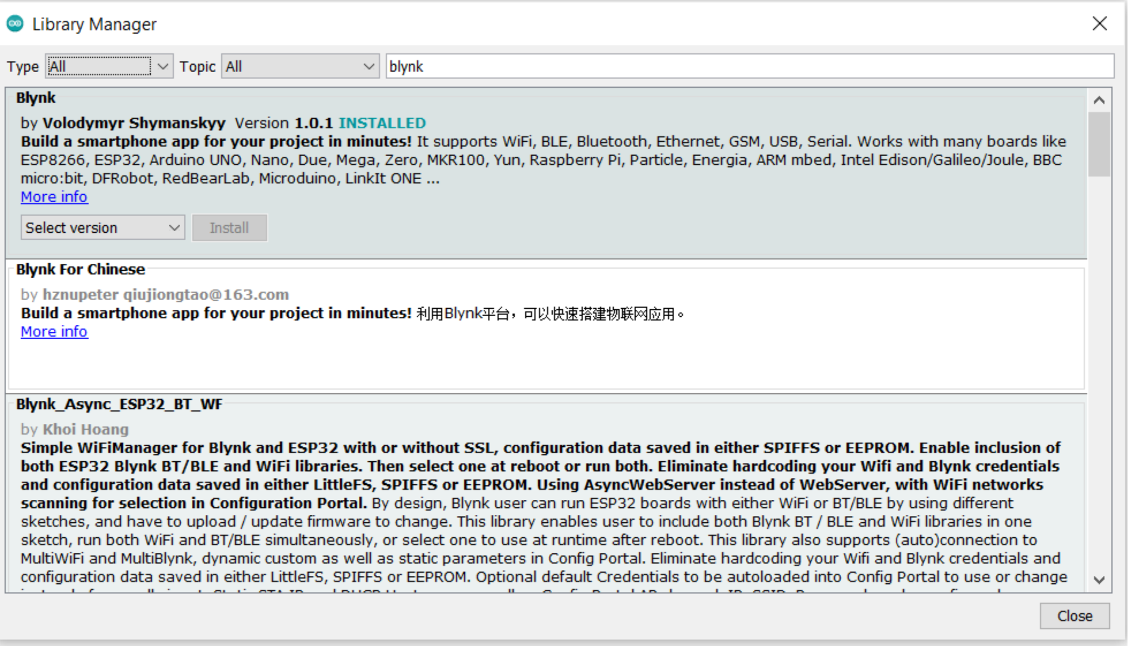
Compiler warnings: None

☑ Display line numbers
☑ Verify code after upload
☑ Check for updates on startup
□ Use accessibility features
□ Enable Code Folding
□ Use external editor
☑ 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
C:\Users\Acer\AppData\Local\Arduino15\preferences.txt
(edit only when Arduino is not running)

Step 8: Then, upload the Arduino code by installed **Blynk** and **ESP8266** in the **library manager**.



Library Manager

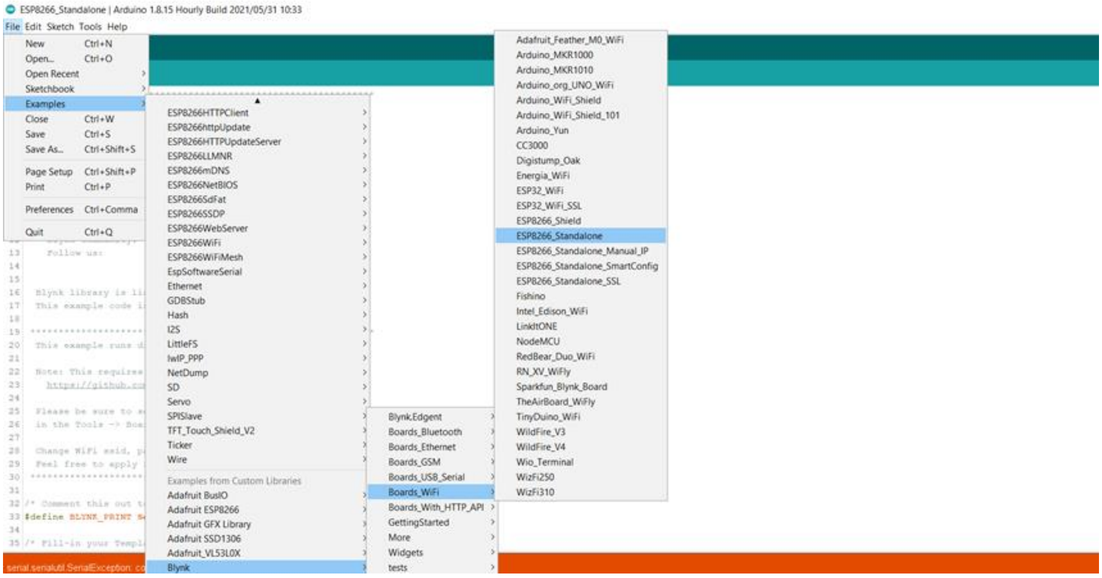
Type All Topic All blynk

Blynk
by **Volodymyr Shymanskyy** Version **1.0.1** INSTALLED
Build a smartphone app for your project in minutes! It supports WiFi, BLE, Bluetooth, Ethernet, GSM, USB, Serial. Works with many boards like ESP8266, ESP32, Arduino UNO, Nano, Due, Mega, Zero, MKR100, Yun, Raspberry Pi, Particle, Energia, ARM mbed, Intel Edison/Galileo/Joule, BBC micro:bit, DFRobot, RedBearLab, Microduino, LinkIt ONE ...
More info

Blynk For Chinese
by **hznupeter qiujiongtao@163.com**
Build a smartphone app for your project in minutes! 利用Blynk平台，可以快速搭建物联网应用。
More info

Blynk_Async_ESP32_BT_WF
by **Khoi Hoang**
Simple WiFiManager for Blynk and ESP32 with or without SSL, configuration data saved in either SPIFFS or EEPROM. Enable inclusion of both ESP32 Blynk BT/BLE and WiFi libraries. Then select one at reboot or run both. Eliminate hardcoding your Wifi and Blynk credentials and configuration data saved in either LittleFS, SPIFFS or EEPROM. Using AsyncWebServer instead of WebServer, with WiFi networks scanning for selection in Configuration Portal. By design, Blynk user can run ESP32 boards with either WiFi or BT/BLE by using different sketches, and have to upload / update firmware to change. This library enables user to include both Blynk BT / BLE and WiFi libraries in one sketch, run both WiFi and BT/BLE simultaneously, or select one to use at runtime after reboot. This library also supports (auto)connection to MultiWiFi and MultiBlynk, dynamic custom as well as static parameters in Config Portal. Eliminate hardcoding your Wifi and Blynk credentials and configuration data saved in either LittleFS, SPIFFS or EEPROM. Optional default Credentials to be autoloaded into Config Portal to use or change

Step 9: For uploading the code, go to **Files** > **Examples** > **Blynk** > **Board_Wif**i > **ESP8266_StandAlone**.

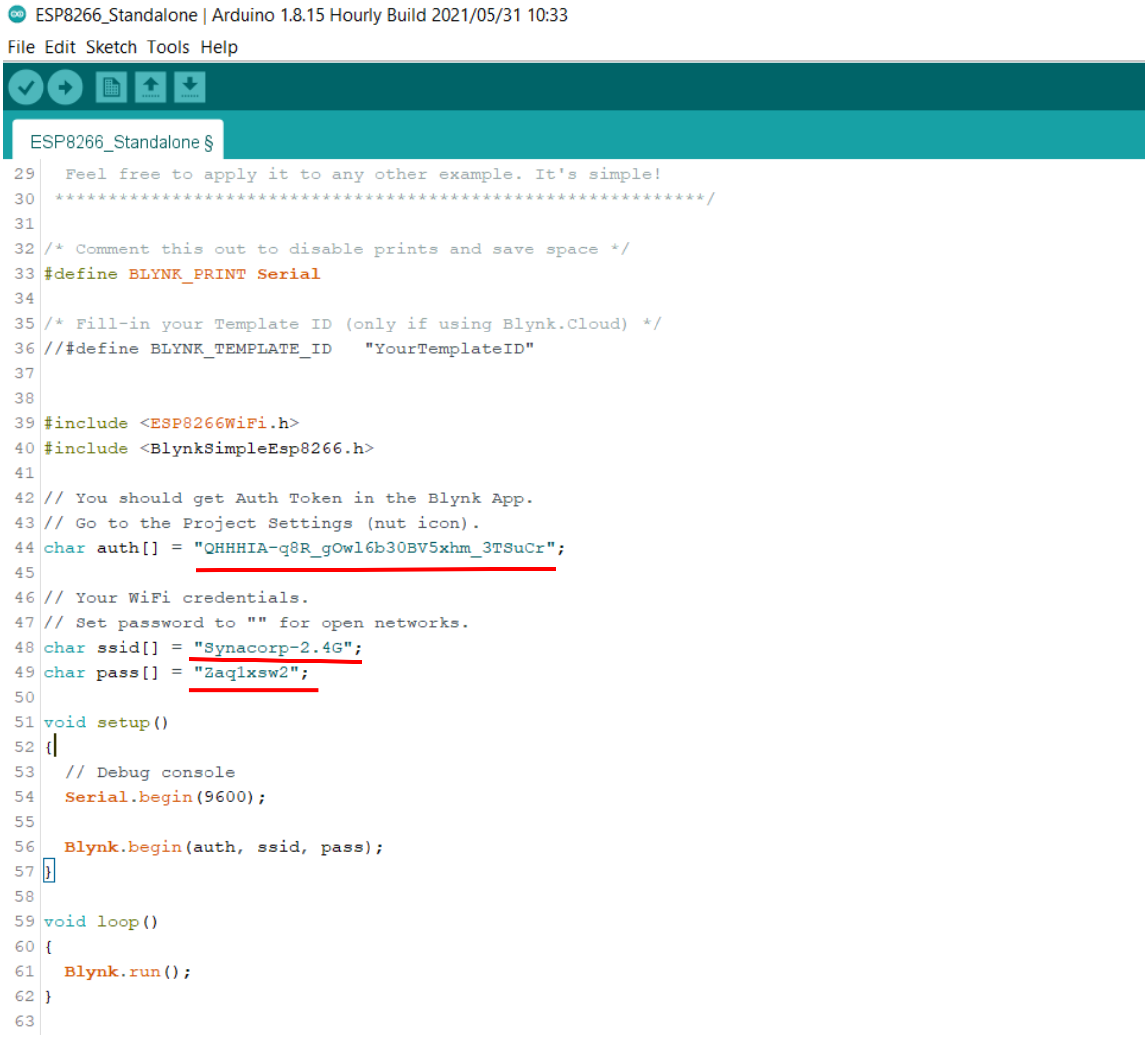


Step 10 : Edit the three key components that need to be specificied :

char auth[] = ""; Specific to your project (Blynk App).

char ssid[] = ""; Specific to the network that we are connecting to (network name).You can "hotspot" from your phone also.

char pass[] = ""; Specific to the network we are connecting to (password).

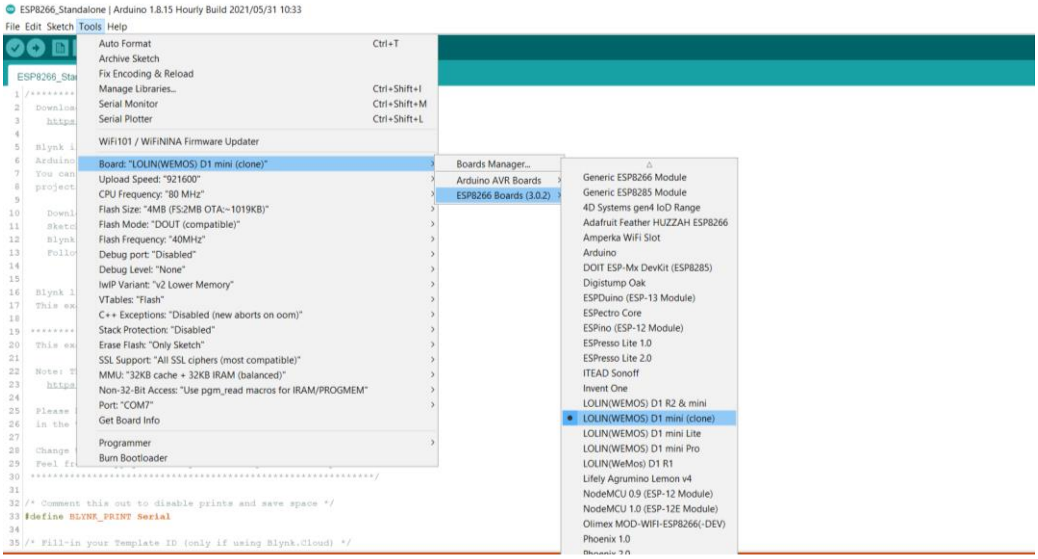


ESP8266_Standalone | Arduino 1.8.15 Hourly Build 2021/05/31 10:33

```
  Feel free to apply it to any other example. It's simple!
 *************************************************************/

/* Comment this out to disable prints and save space */
#define BLYNK_PRINT Serial

/* Fill-in your Template ID (only if using Blynk.Cloud) */
//#define BLYNK_TEMPLATE_ID   "YourTemplateID"


#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "QHHHIA-q8R_gOwl6b30BV5xhm_3TSuCr";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Synacorp-2.4G";
char pass[] = "Zaq1xsw2";

void setup()
{
  // Debug console
  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);
}

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

```

Step 11: Select the proper port number and locate the board.

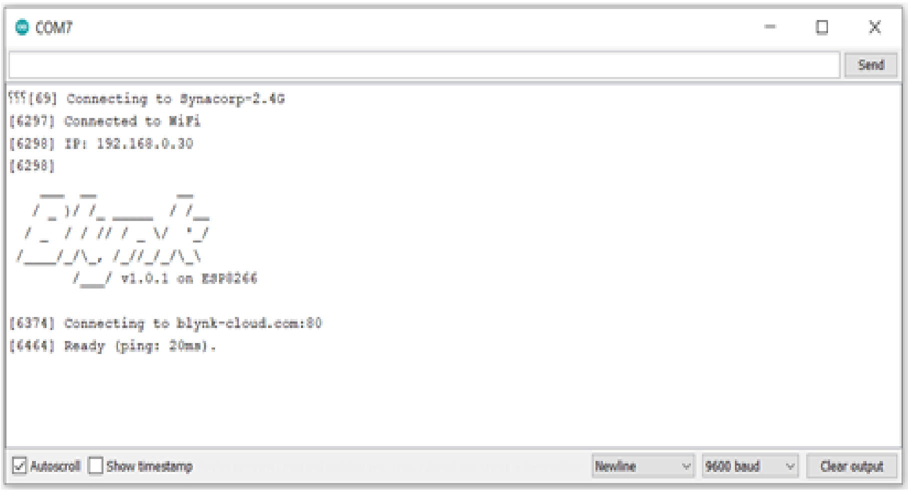
Select **Tools > Port .** Select **Tools > Board > Board Manager**.


ESP8266_Standalone | Arduino 1.8.15 Hourly Build 2021/05/31 10:33
File Edit Sketch Tools Help
ESP8266_Sta
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Get Board Info
Programmer
Burn Bootloader
Boards Manager...
Arduino AVR Boards
ESP8266 Boards (3.0.2)
Generic ESP8266 Module
Generic ESP8285 Module
4D Systems gen4 IoD Range
Adafruit Feather HUZZAH ESP8266
Amperka WiFi Slot
Arduino
DOIT ESP-Mx DevKit (ESP8285)
Digistump Oak
ESPDuino (ESP-13 Module)
ESPectro Core
ESPino (ESP-12 Module)
ESPresso Lite 1.0
ESPresso Lite 2.0
ITEAD Sonoff
Invent One
LOLIN(WEMOS) D1 R2 & mini
● LOLIN(WEMOS) D1 mini (clone)
LOLIN(WEMOS) D1 mini Lite
LOLIN(WEMOS) D1 mini Pro
LOLIN(WeMos) D1 R1
Lifely Agrumino Lemon v4
NodeMCU 0.9 (ESP-12 Module)
NodeMCU 1.0 (ESP-12E Module)
Olimex MOD-WIFI-ESP8266(-DEV)
Phoenix 1.0

Board: "LOLIN(WEMOS) D1 mini (clone)"
Upload Speed: "921600"
CPU Frequency: "80 MHz"
Flash Size: "4MB (FS:2MB OTA:~1019KB)"
Flash Mode: "DOUT (compatible)"
Flash Frequency: "40MHz"
Debug port: "Disabled"
Debug Level: "None"
IwIP Variant: "v2 Lower Memory"
VTables: "Flash"
C++ Exceptions: "Disabled (new aborts on oom)"
Stack Protection: "Disabled"
Erase Flash: "Only Sketch"
SSL Support: "All SSL ciphers (most compatible)"
MMU: "32KB cache + 32KB IRAM (balanced)"
Non-32-Bit Access: "Use pgm_read macros for IRAM/PROGMEM"
Port: "COM7"

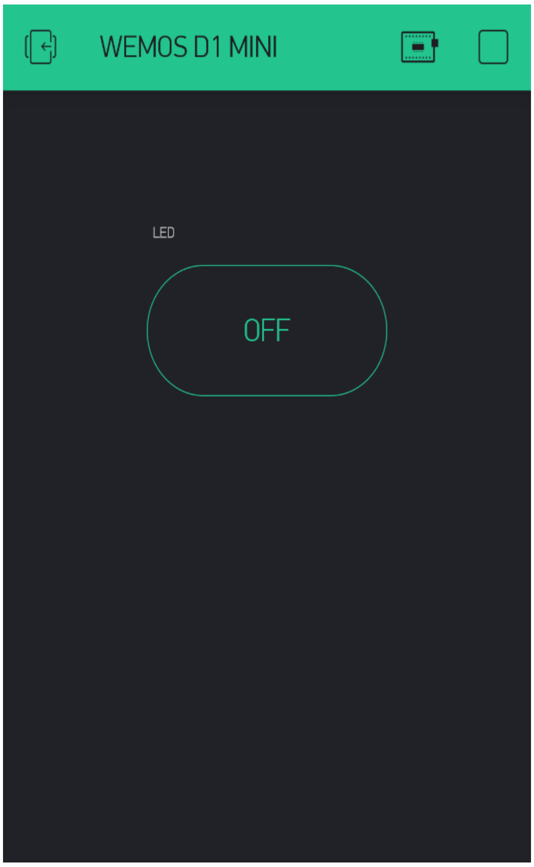
```
/* Comment this out to disable prints and save space */
#define BLYNK_PRINT Serial

/* Fill-in your Template ID (only if using Blynk.Cloud) */
```

Step 12: Run and upload the project.Then, open the **serial monitor** to see the receiving messages from *Blynk App*.

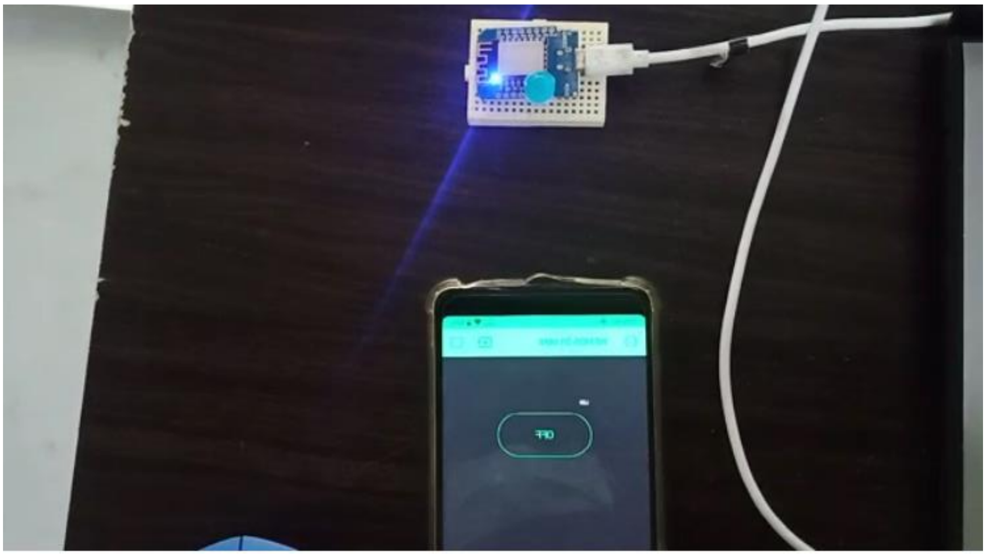


```
⸮⸮⸮[69] Connecting to Synacorp-2.4G
[6297] Connected to WiFi
[6298] IP: 192.168.0.30
[6298]
    ___  __          __
   / _ )/ /_ _____  / /__
  / _  / / // / _ \/  '_/
 /____/_/\_, /_//_/_/\_\
        /___/ v1.0.1 on ESP8266

[6374] Connecting to blynk-cloud.com:80
[6464] Ready (ping: 20ms).
```

Step 13 : Blynking.Press the play button and turn the LED on and off.



Conclusion :



Once you upload the code successfully, green LED will start blinking if you switch the button ON.This tutorial was easy ousing Wmos D1 mini NodeMcu programming with simple steps.