**Turn on/off LED with Blynk**

1. **Requirements**

* Use Wemos D1 wifi board
* Setup Blynk app on your smart phone
* Use Blynk to turn ON/OFF the LED

1. **Devices:**

|  |  |
| --- | --- |
| **Devices** | **Description** |
| WeMos D1 board |  |
| LED |  |
| Resistor |  |

1. **Blynk installation**

Download Blynk from Apple AppStore or Google Play Store

Graphical user interface, application

Description automatically generated

1. **Sign up for a Blynk account**

Graphical user interface, text, application, chat or text message

Description automatically generated

1. **Login and create the template**

A screenshot of a phone

Description automatically generated with low confidence

Click at the icon  to switch to Developer Mode

A screenshot of a phone

Description automatically generated with medium confidence

At this mode you can create a Template or modify the current template on this screen.

Tap the icon  to Add Template or tap on a template on this screen (if it exists) to modify.

But the best way to add a template and create some Virtual Pin on your application is using the Blynk Console website (<https://blynk.cloud/>). After login with your account, it looks like this:

Graphical user interface, application

Description automatically generated

Click New Template to add a template into your current account.

Graphical user interface, application, email

Description automatically generated

Save your template

Graphical user interface, application

Description automatically generated

Now, we need to add an IoT device to our template (LAB 7)

Click on the  icon to manage your devices.

Graphical user interface, text, application

Description automatically generated

Click the “+ New Device button”

Graphical user interface, application, website

Description automatically generated

Select the option “From template”

Graphical user interface, text, application

Description automatically generated

Click create button and you will be informed the 3 keys like this:

#define BLYNK\_TEMPLATE\_ID "TMPL\_MOwRKvB"

#define BLYNK\_DEVICE\_NAME "ESP8266"

#define BLYNK\_AUTH\_TOKEN "N2JYxK7R45TfG0tFjn7IZXtNdlTw1G3B"

Copy and paste them at the TOP of your sketch.

These keys will be used to authenticate your application.

Graphical user interface, application

Description automatically generated

You can collect all keys by select your device in above screen, like this:

Graphical user interface, text, application

Description automatically generated

Next step, we need to create our blynk application.

Remember that, you can create both web application and mobile application.

For the web app: go to Templates, select our template (LAB 7) then go to the Web Dashboard to edit the web app interface.

Timeline

Description automatically generated with low confidence

You can Edit the web dashboard as your wishes later.

But now, we need to create a button to control the LED on ESP8266 board.

Go to the Datastream and add one.

Graphical user interface, text, application

Description automatically generated

Click **+New Datastream** and select **Virtual Pin,** then fill information about this button

Graphical user interface, application, email

Description automatically generated

Note: this button has a virtual Pin is V0, we will use this pin in our sketch to read the controlling signal.

At now, you can go to Web Dashboard to create your app. It’s really easy and you can finish it at once.

But here, we will focus on Mobile Application

Click on a template LAB 7 that we have just created.

A screenshot of a phone

Description automatically generated with medium confidenceBackground pattern

Description automatically generated

Click at the  icon to modify your application items with the Widget Box

We will add a button to control the LED

Graphical user interface, application

Description automatically generatedChart

Description automatically generated with low confidence

Tap on the Button to edit the datastream and pin. Mode will be SWITCH, then tap Select Data Stream and choose the Switch Button that we have just created.

Graphical user interface, application, Teams

Description automatically generatedText

Description automatically generated with low confidenceGraphical user interface, application, Teams

Description automatically generatedA picture containing diagram

Description automatically generated

Click the  icon to exit developer mode, turn back to App mode.

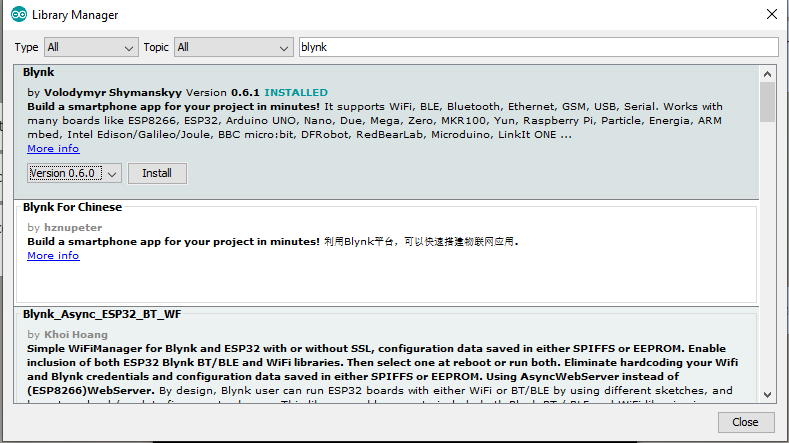
A picture containing chart

Description automatically generated

We have just finished create a simple app on blynk. Next step is code the sketch

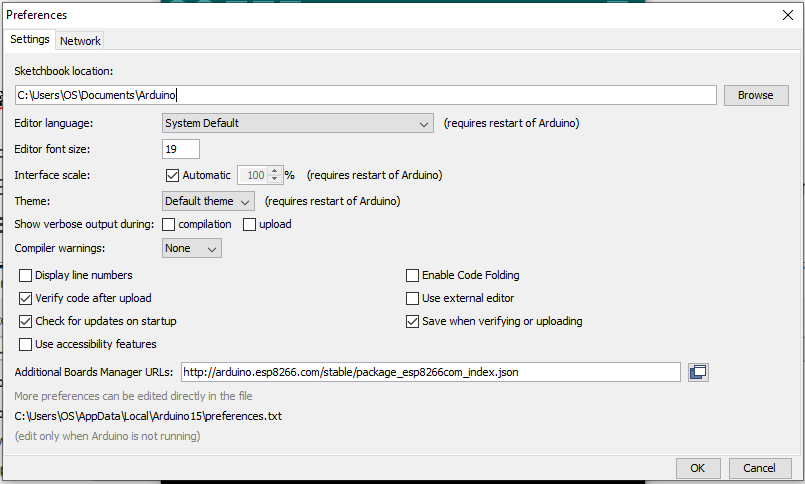
1. **Install Blynk library**

(How to install Blynk library: <http://help.blynk.cc/en/articles/512105-how-to-install-blynk-library-for-arduino-ide>)

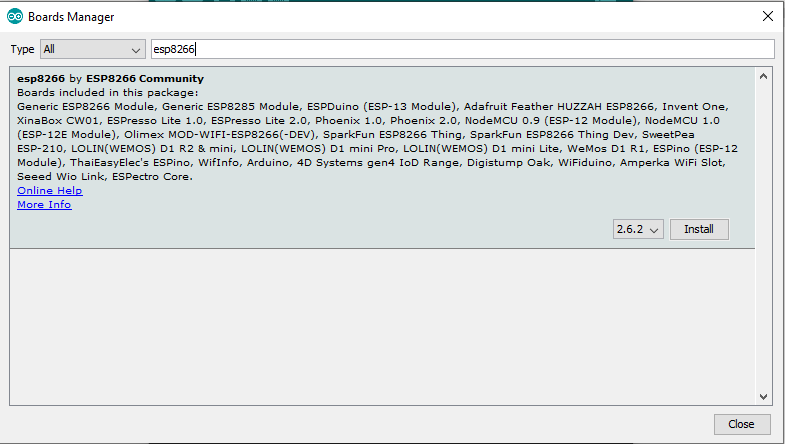


1. **Board Setup**

* From the Arduino IDE, select File>Preferences in menu bar
* Enter http://arduino.esp8266.com/stable/package\_esp8266com\_index.json in the **Additional Board Manager URLs** box.



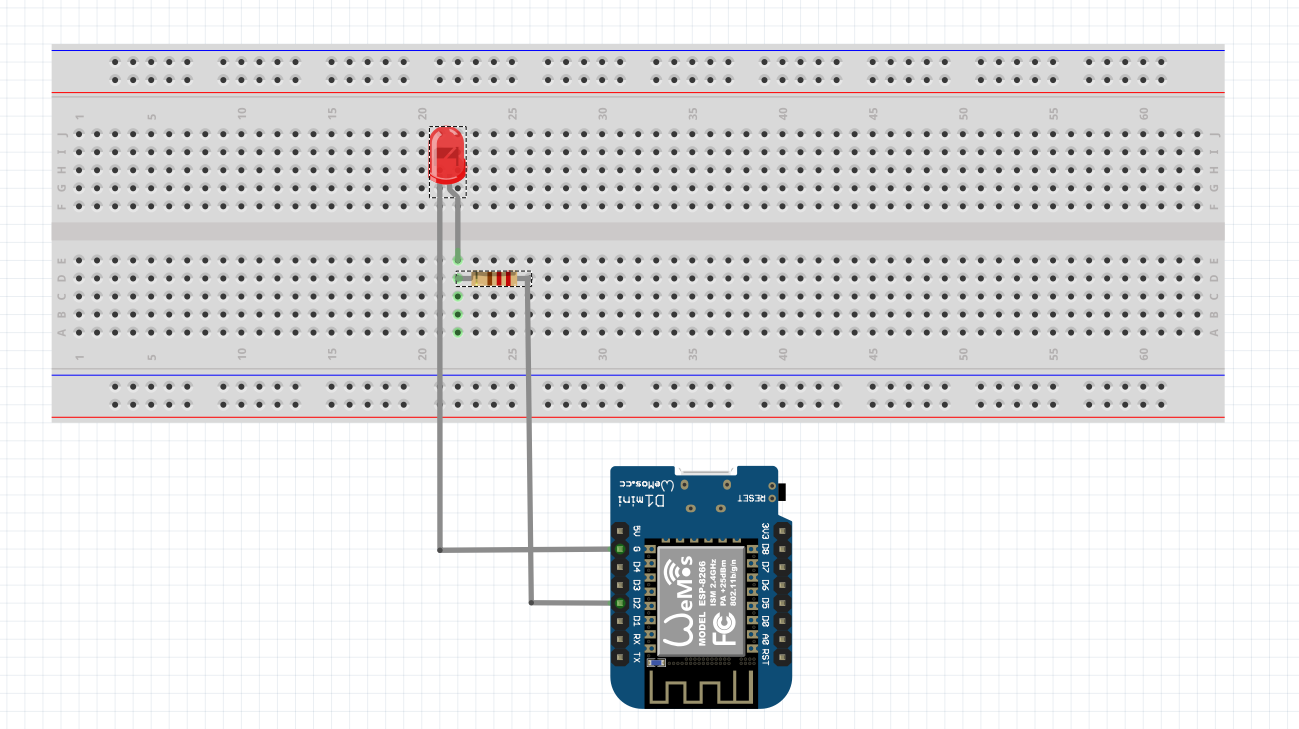
* SelectTools> Board menu > Boards Manager and search for **esp8266** platform.



* Select the version from Drop-down list.
* Click **install.**

1. **The Schematic**

* We use the D2 pin for LED controlling.



1. **Code the sketch**

// Template ID, Device Name and Auth Token are provided by the Blynk.Cloud

// See the Device Info tab, or Template settings

// You need to change 3 keys as your Device Info keys

#define BLYNK\_TEMPLATE\_ID "TMPL\_MOwRKvB"

#define BLYNK\_DEVICE\_NAME "ESP8266"

#define BLYNK\_AUTH\_TOKEN "N2JYxK7R45TfG0tFjn7IZXtNdlTw1G3B"

// Comment this out to disable prints and save space

#define BLYNK\_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

char auth[] = BLYNK\_AUTH\_TOKEN;

// Your WiFi credentials.

char ssid[] = "Your\_wifi\_name";

char pass[] = "Your\_wifi\_password";

void setup()

{

// Debug console

Serial.begin(115200);

Blynk.begin(auth, ssid, pass);

//Setup LED

pinMode(D5, OUTPUT);

}

//Setup for Blynk App to control Led

BLYNK\_WRITE(V0)

{

int pinValue = param.asInt();

if(pinValue == 1)

{

digitalWrite(D5, HIGH);

Serial.println("Button switched to 1");

}

else

{

digitalWrite(D5, LOW);

Serial.println("Button switched to 0");

}

}

void loop()

{

Blynk.run();

}