

LED Controlling over IOT

In our project we are controlling the LED using IOT. For that NodeMCU esp8266 open-source controller board is used , which has an inbuilt wifi module using this we are able to send and receive the command from anywhere.

To control the LED we need a webpage (which is nothing but a dashboard to send the command to the controller via the internet), for that we are using the Blynk 2.0 it is a free platform to create the device dashboard.

Components Required

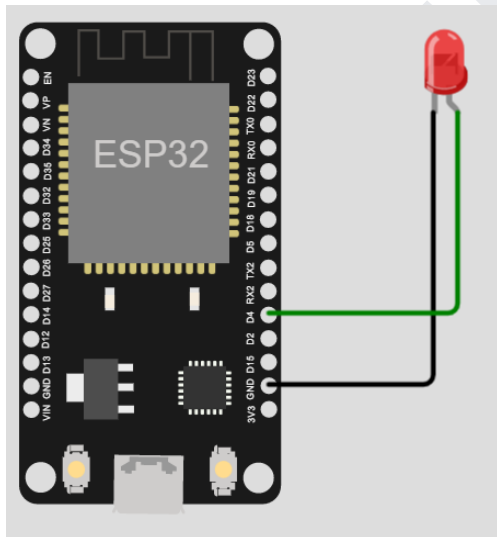
Hardware

- NodeMCU ESP8266
- LED
- Jumper wires

Software

- Arduino IDE
- Blynk 2.0 webpage

Circuit diagram



- Connecting LED positive to the D4 (pin 2).
- Connecting LED negative to the GND (ground).

Source Code :-

/* Remainder - Code explanation is also provided for proper understanding at the end of the document . */

```
#define BLYNK_TEMPLATE_ID "Enter your device ID"
#define BLYNK_DEVICE_NAME "Enter your device NAME"
#define BLYNK_AUTH_TOKEN "Enter your device TOKEN"
```

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
```

```
char auth[ ] = BLYNK_AUTH_TOKEN;
char ssid[ ] = "      "; // Enter your Wifi Username
char pass[ ] = "      "; // Enter your Wifi password
```

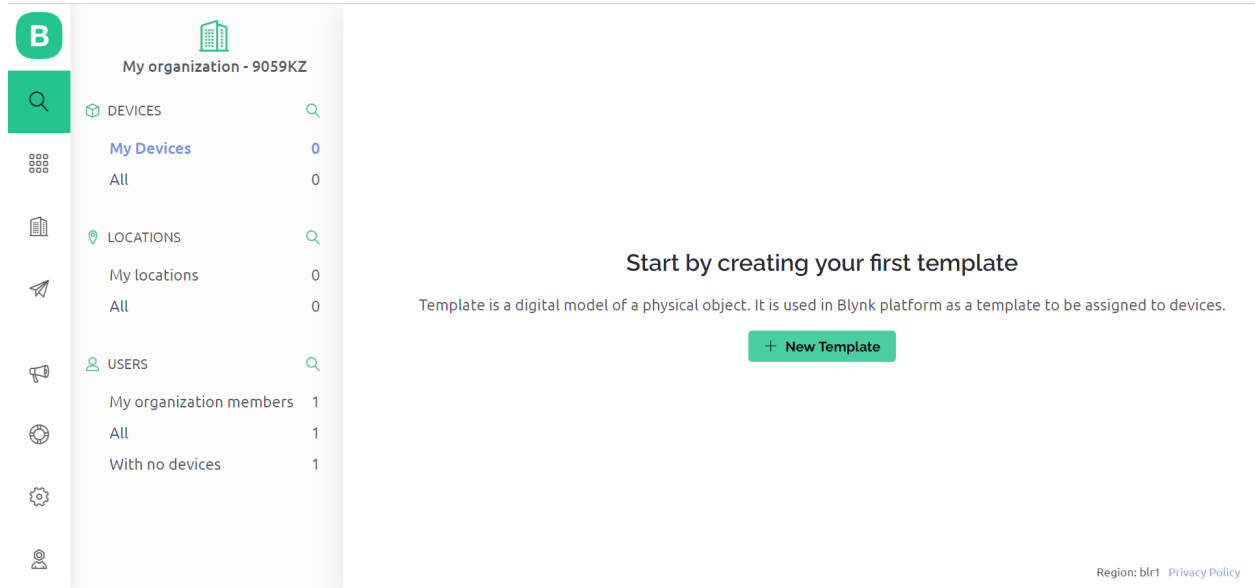
```
int ledpin = 2;
void setup( )
{
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
  pinMode(ledpin,OUTPUT);
}
```

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

Create a dashboard on Blynk 2.0

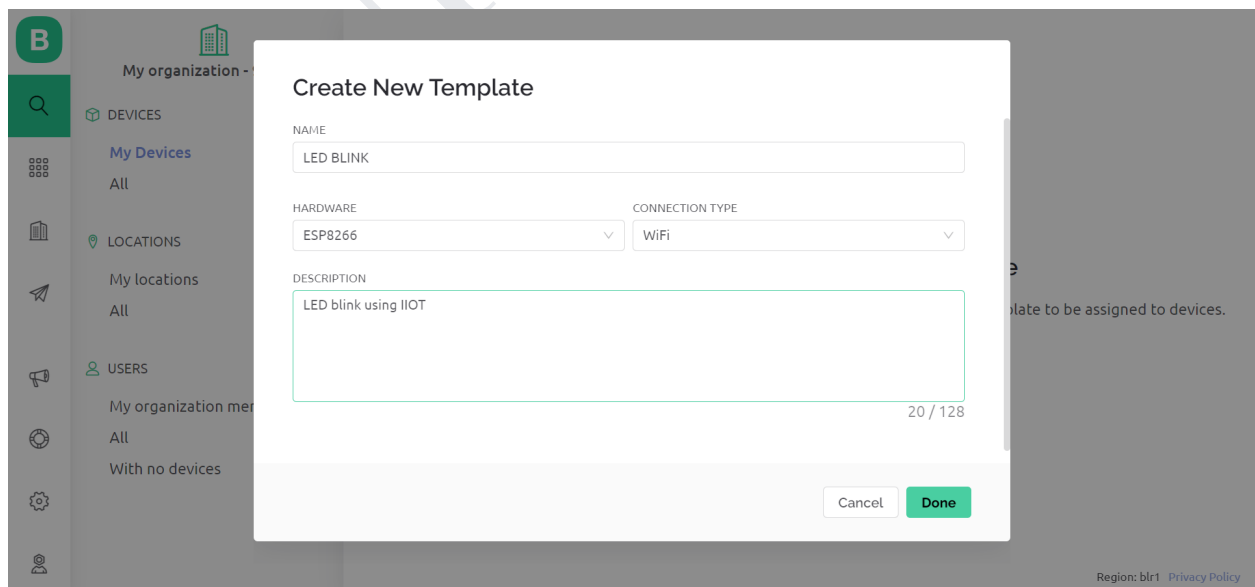
STEP 1:

Login with the Blynk 2.0 and start it by clicking **NEW TEMPLATE**.



STEP 2:

Create a New Template by filling the name, hardware, connection type and description then click the **DONE** button.



STEP 3:

Now your template has been modified as shown below.

B

LED BLINK

...

Cancel

Save

Info

Metadata

Datastreams

Events

Automations

Web Dashboard

Mobile Dashboard

TEMPLATE NAME

LED BLINK

HARDWARE

ESP8266

CONNECTION TYPE

WiFi

DESCRIPTION

LED blink using IOT

20 / 128

TEMPLATE ID

TMPLUEGPMd85

MANUFACTURER

My organization 9059KZ

OFFLINE IGNORE PERIOD

00 hrs 00 mins 00 secs

20 / 128

TEMPLATE IMAGE (OPTIONAL)

Add image

Upload from computer or drag-n-drop
.png or .jpg, minimum width 500px

FIRMWARE CONFIGURATION

```
#define BLYNK_TEMPLATE_ID "TMPLUEGPMd85"
#define BLYNK_TEMPLATE_NAME "LED BLINK"
```

Template ID and Device Name should be included at the top of
your main firmware

Region: blr1 [Privacy Policy](#)

STEP 4:

Then go to Datastreams and click **NEW DATASTREAM**.

B

LED BLINK

...

Cancel

Save

Info

Metadata

Datastreams

Events

Automations

Web Dashboard

Mobile Dashboard

Datastreams

Datastreams is a way to structure data that regularly flows in and out from device. Use it for sensor data, any telemetry, or actuators.

+ New Datastream

Region: blr1 [Privacy Policy](#)

STEP 5:

In the new datastream goto **Digital Datastream** , in that fill the name and enter the pin as 2, then pin mode as Output.

The screenshot shows the 'LED BLINK' interface with the 'Datastreams' tab selected. A modal titled 'Digital Datastream' is open, allowing configuration of a new digital datastream. The modal contains the following fields:

- NAME:** A text input field containing 'LED'.
- ALIAS:** A text input field containing 'LED'.
- PIN:** A dropdown menu with '2' selected.
- PIN MODE:** A dropdown menu with 'Output' selected.
- ADVANCED SETTINGS:** A link to expand additional configuration options.

At the bottom of the modal are 'Cancel' and 'Save' buttons. The background interface shows a sidebar with various icons and a top navigation bar with tabs for Info, Metadata, Datastreams, Events, Automations, Web Dashboard, and Mobile Dashboard.

STEP 6:

After saving that a template will be generated as shown below.

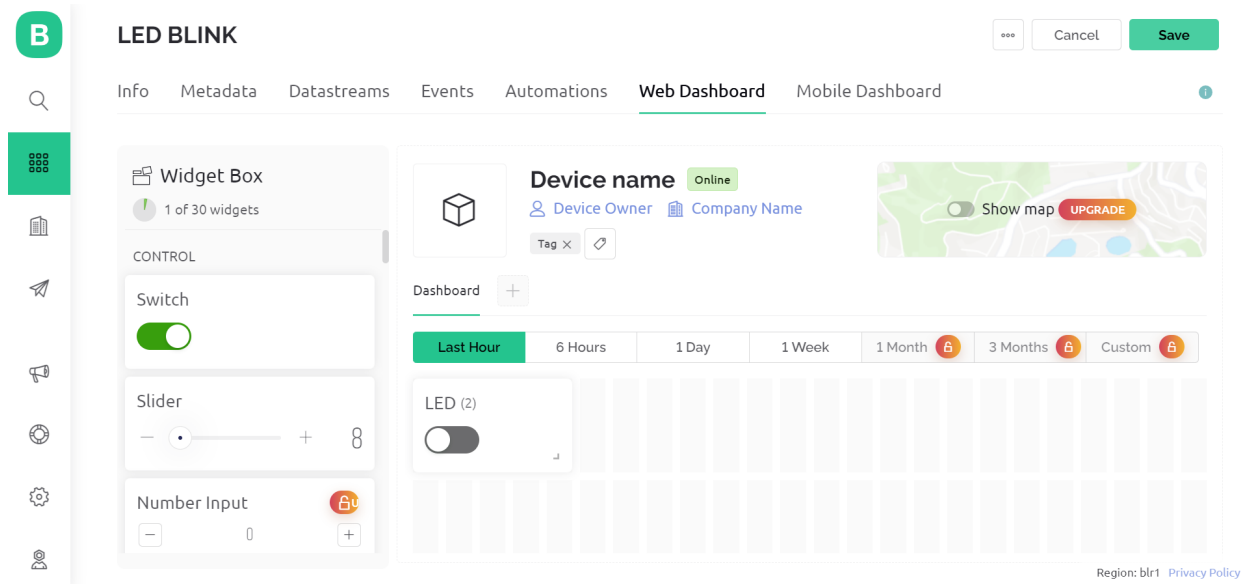
The screenshot shows the 'LED BLINK' interface with the 'Datastreams' tab selected. A table displays the generated datastream template. The table has the following columns: Id, Name, Alias, Color, Pin, Data Type, Units, Is Ra, and Actions.

Id	Name	Alias	Color	Pin	Data Type	Units	Is Ra	Actions
1	LED	LED		2	Integer	false		

The interface also includes a sidebar with icons, a top navigation bar with tabs for Info, Metadata, Datastreams, Events, Automations, Web Dashboard, and Mobile Dashboard, and a 'New Datastream' button.

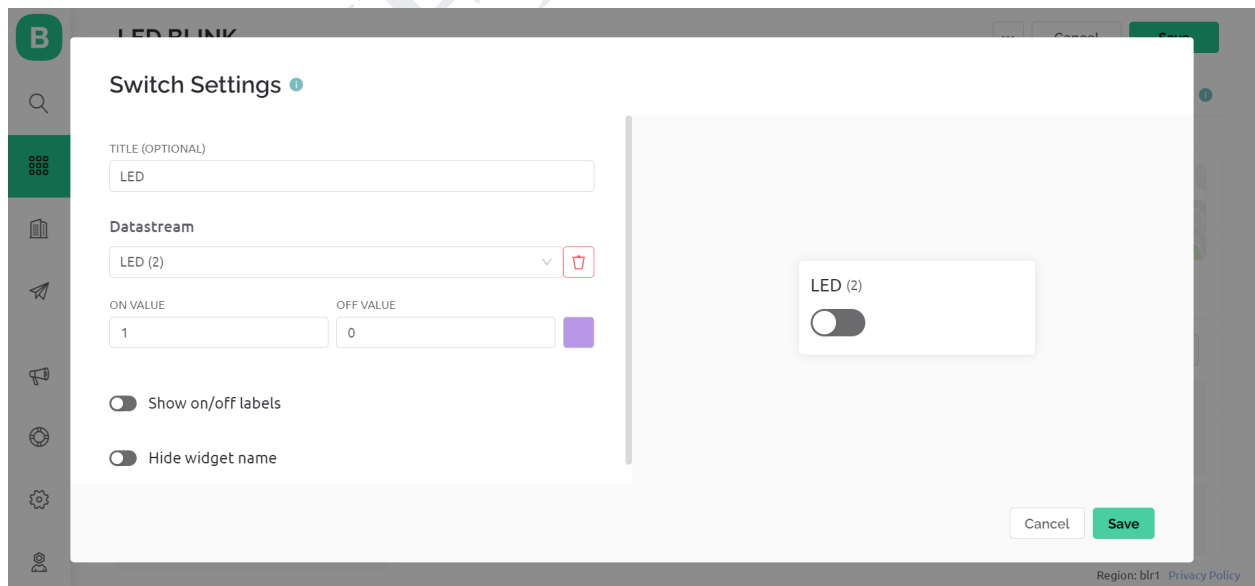
STEP 7:

Goto web dashboard take the switch and place it on the dashboard and then double click the switch which is on the dashboard in that go to settings.



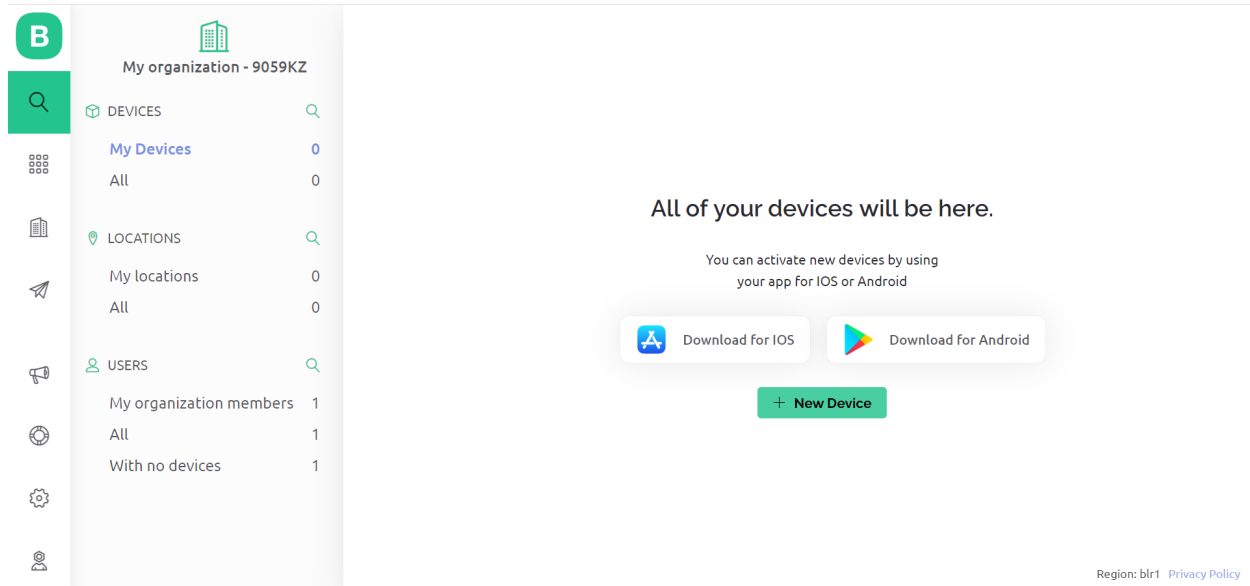
STEP 8:

You will get the switch setting panel enter the title, datastream and on value as 1 and save it, Note(after saving the switch setting save the entire template by clicking the save button on the right top corner).



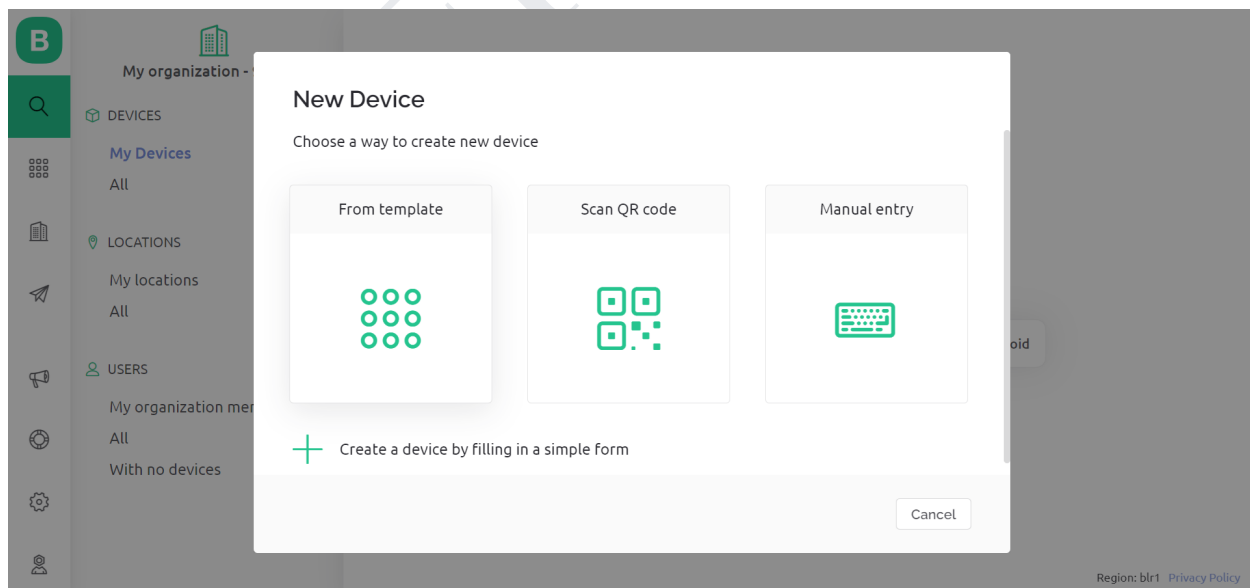
STEP 9:

By clicking the search symbol on your left you are able to move on to device pages in that click new device .



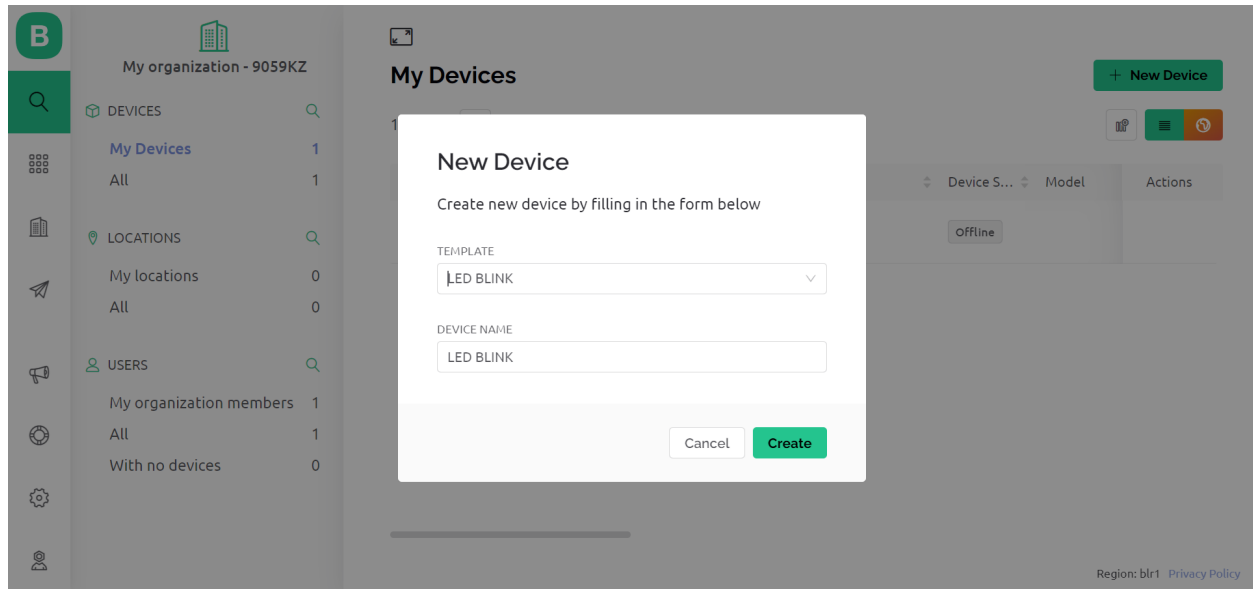
STEP 10:

In the new device click the **From template**.



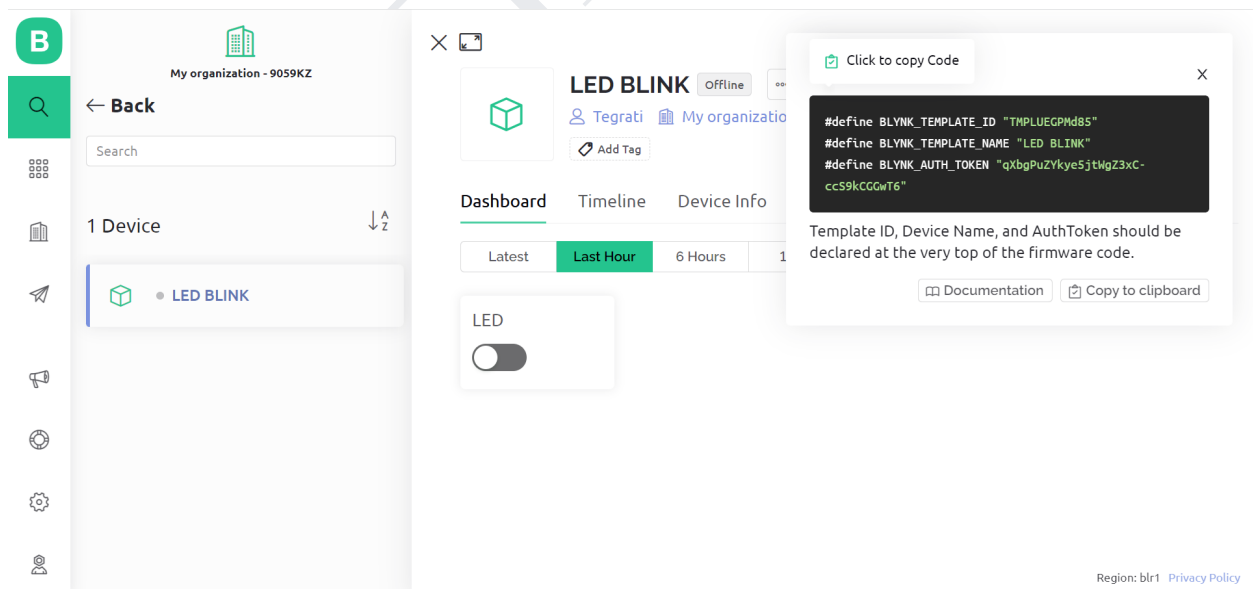
STEP 11:

Enter the template and device name then click create.



STEP 12:

Finally you have created the device on the server to make the communication between the esp8266 and server ,copy the template id, name and auth token by clicking the **Click to copy code** and paste it on the source code.



Code Explanation

This is how the code flow works.

First we need to define the device ID, device NAME, device TOKEN,
All these three entries must be defined correctly that have been given in the dashboard which we have created.

```
#define BLYNK_TEMPLATE_ID "Enter your device ID"  
#define BLYNK_DEVICE_NAME "Enter your device NAME"  
#define BLYNK_AUTH_TOKEN "Enter your device TOKEN"
```

Now define a BLYNK_PRINT Serial which will gives the option to control the LED using the Serial monitor,

And include the ESP8266WiFi.h to enable the wifi module in the ESP8266, Then include the BlynkSimpleEsp8266.h that library will make the communication to the Blynk 2.0.

```
#define BLYNK_PRINT Serial  
#include <ESP8266WiFi.h>  
#include <BlynkSimpleEsp8266.h>
```

Declaring the Token, ssid (username of your wifi network that will be connected to the esp8266), password (Enter the password of that username).

```
char auth[ ] = BLYNK_AUTH_TOKEN;  
char ssid[ ] = " "; // Enter your Wifi Username  
char pass[ ] = " "; // Enter your Wifi password
```

Declare the LED PIN as 2 (in ESP8266 it is a D4).

```
int ledpin = 2;
```

In this void function we need to set up the Serial baud rate to start the serial monitor, Blynk begin (to give the token,ssid,password), pinMode to set the pin as OUTPUT.

```
void setup( )  
{  
  Serial.begin(115200);  
  Blynk.begin(auth, ssid, pass);  
  pinMode(ledpin,OUTPUT);
```

```
}
```

This is a loop function that helps to repeat the process so inside the loop Blynk.run() has been called to run continuously.

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

CONFIDENTIAL