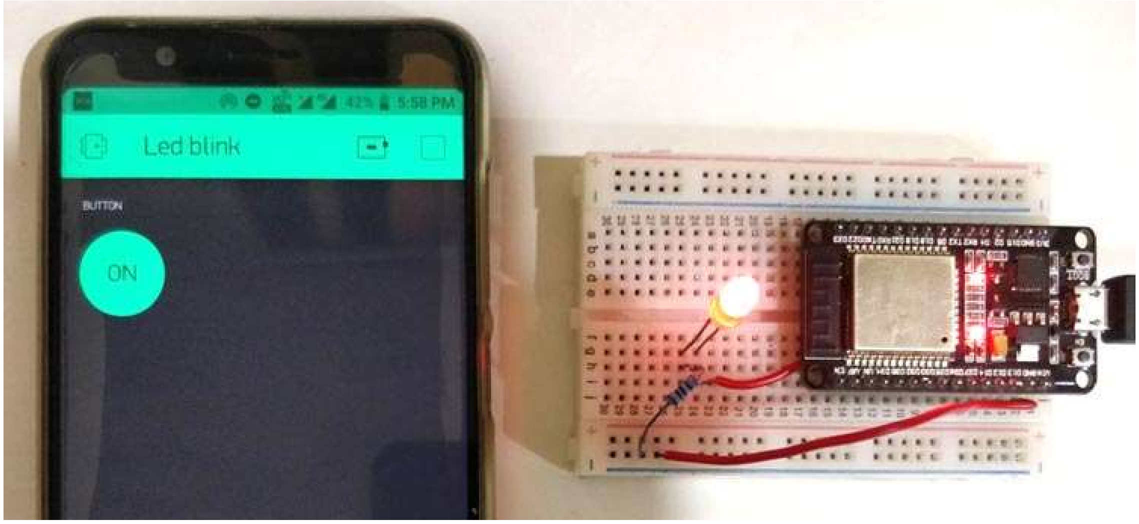
LAB -16

Controlling LED Using Node with Mobile Application



Blynk is an IoT Platform to control Arduino, Raspberry Pi, NodeMCU and other microcontrollers over the Internet. Blynk app can be downloaded from Google play store or Apple store.

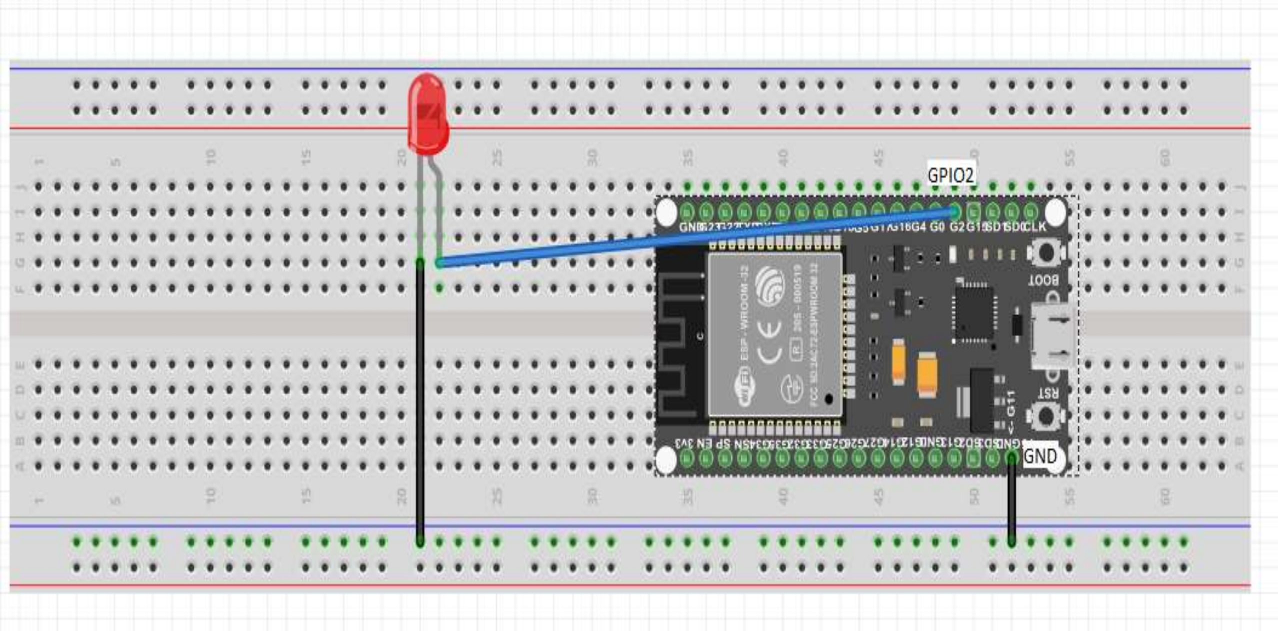
Blynk app provides a digital dashboard where you can build a graphic interface for any IoT based project by simply dragging and dropping widgets. It's a simple and easy to use IoT platform to build complex applications. Blynk is not bound to some specific board or platform but it can be used with any microcontroller, provided that the microcontroller is connected to the internet. Raspberry Pi has inbuilt Wi-Fi and other microcontrollers like Arduino can be connected to the internet using some Wi-Fi module like ESP8266 etc.

Component Required

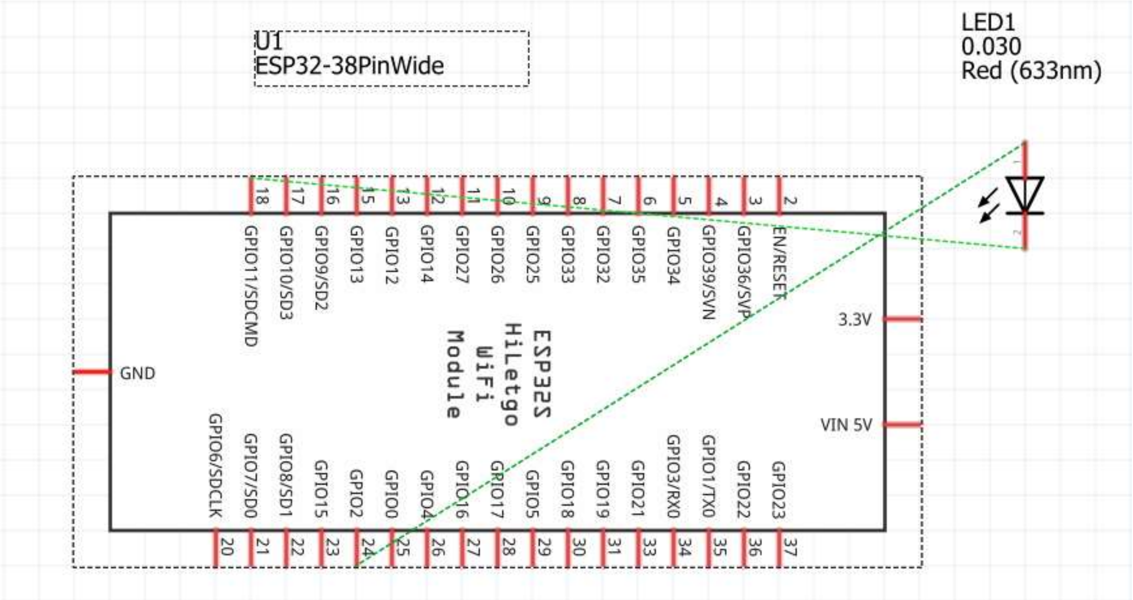
1. ESP32 module
2. USB Cable
3. Arduino IDE
4. Blynk App
5. LED
6. Breadboard

7. Jumper wires

Connection Diagram



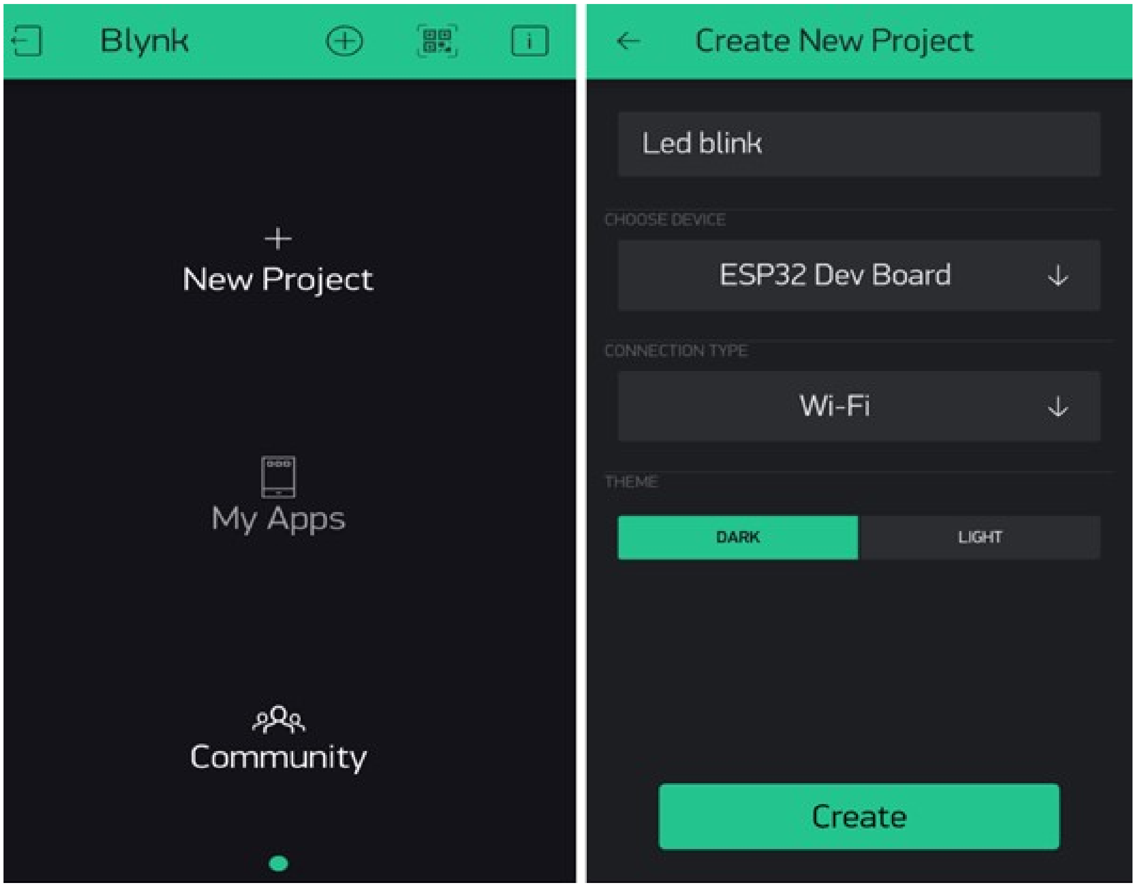
Circuit Diagram



Configuring Blynk App for ESP32

Following are the steps to configure Blynk app in your phone and use it for a project:

1. Firstly, download the Blynk app on your phone from Google play store and install it.
2. After installing, you need to create an account in this app; you may use your current Gmail account.
3. After creating an account a window will open, in this click on **New Project.**



Led blink

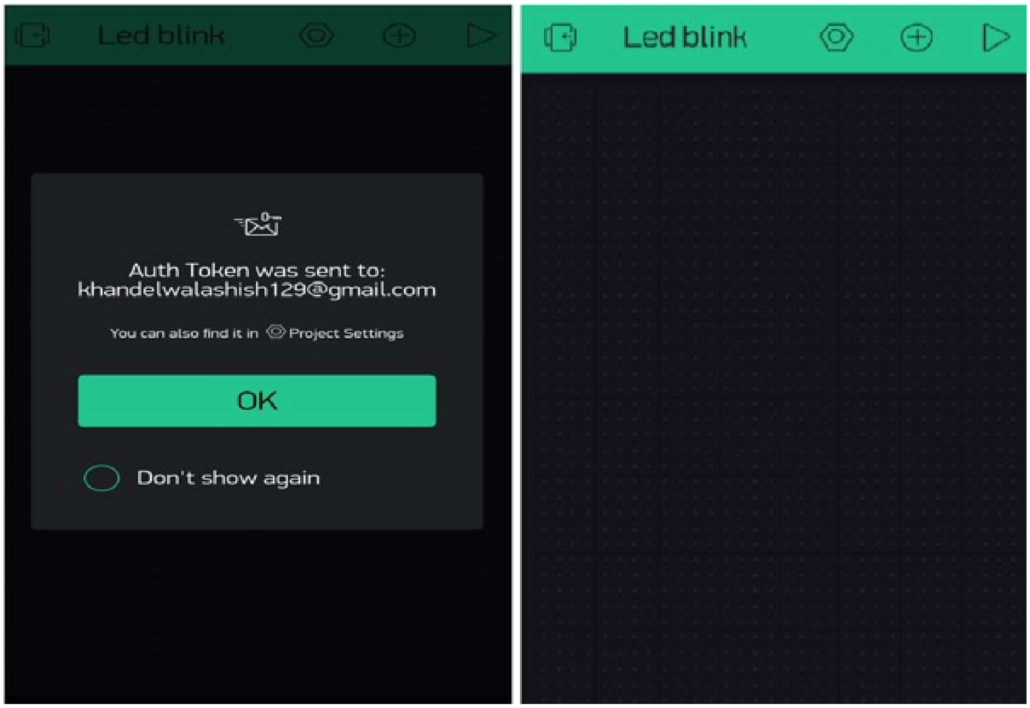
CHOOSE DEVICE
ESP32 Dev Board

CONNECTION TYPE
Wi-Fi

THEME
DARK

4. Now give the project a name according to your choice and in *device* choose **ESP32 Dev Board** and in *Connection type* choose **Wi-Fi** and then click on Create.

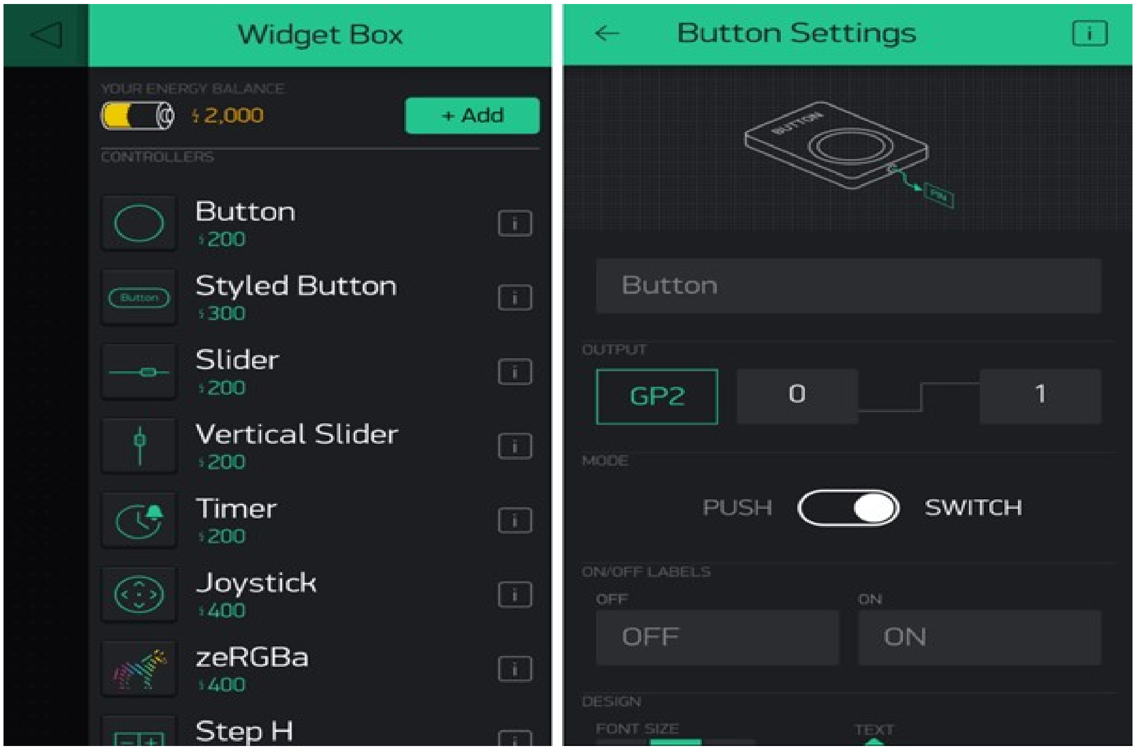
5. Now a window will come which shows your **authentication token** which you will need later sent to your concerned mail ID. You can open your email to check the authentication key.



Auth Token was sent to:
khandelwalashish129@gmail.com

You can also find it in Project Settings

6. After clicking on OK, you will find a canvas window.
7. Now, tap anywhere on the canvas to open the widget box. All the available widgets are located here. Now choose a button.



Button

OUTPUT
GP2 0 1

MODE
PUSH SWITCH

ON/OFF LABELS
OFF ON
OFF ON

8. Click on the Button widget to change the setting.
9. Set output pin to gp2 as I am taking output here from GPIO2 pin, you can change according to you. In *Mode* select to switch.
10. When you are done with this setting, you are ready to work with this app. On pressing the Play button it will switch you from EDIT mode to PLAY mode where you can interact with the hardware. In PLAY mode, you will not be able to drag or set new widgets, press STOP to get back to STOP mode.

Programming ESP32 for Blynk Application

First of all, you have to download the Blynk library from the given link.

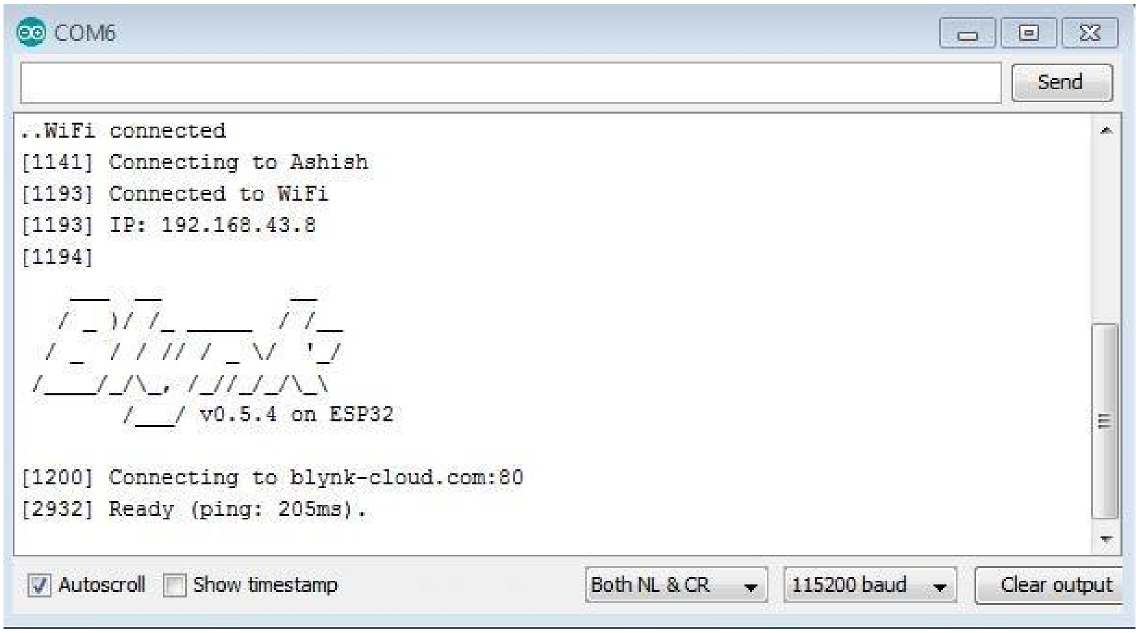
https://github.com/blynkkk/blynk-library

It is a zip file, download it and extract it and then copy this library file to Arduino Library files. You can find your Arduino library file in ***Documents--> Arduino--> libraries***. Copy this library file here.

Programming of ESP32 with Arduino IDE and Blynk app

Now you are in a stage to upload the code in your ESP32 and perform the project, for this following steps should be performed:

1. Connect ESP32 to your PC via USB cable and make a circuit as given above, here I am using GPIO2 you can use according to you.
2. Open your Arduino IDE and copy this code to your IDE window but make sure that you provide correct WiFi credentials of your network.
3. Now go to *Tools--> Board--> ESP32* Dev module.
4. Then *Tools-->Port* and select the port to which your ESP32 is connected.
5. Now click on upload to upload the code.
6. After complete uploading you will find a message like this in your output console.
7. Now open your serial monitor and press reset button of ESP32 and now ESP starts connecting to your network, once connected it will give you IP of your ESP and message like this:



```
..WiFi connected
[1141] Connecting to Ashish
[1193] Connected to WiFi
[1193] IP: 192.168.43.8
[1194]
    ___  __          __
   / _ )/ /_ _____  / /__
  / _  / / // / _ \/  '_/
 /____/_/\_, /_//_/_/\_\
        /___/ v0.5.4 on ESP32

[1200] Connecting to blynk-cloud.com:80
[2932] Ready (ping: 205ms).
```

☑ Autoscroll □ Show timestamp Both NL & CR 115200 baud

8. Now open the Blynk app and go to the project you created earlier and tap on PLAY.
9. Now you can see on clicking on the button your LED will change its state.

This is how you are successfully able to control LED using ESP32 with Blynk App.

Algorithm:

1. Define GPIO2 as a pin for LED.
2. Define authentication token, ssid and the password for communication.
3. In setup, Define pin as OUTPUT pin.
4. begin the serial communication with a certain baudrate.
5. Check if the wifi is connected or not.
6. Connect to the Blynk application using the authentication token generated.
7. In loop, start the blynk application using Blynk.run().

Code:

```
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

// Your Blynk Auth Token
char auth[] = "";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = “OppoA58";
char pass[] = "So@1234@890";

// Define the LED pin
const int ledPin = 2;

void setup()
{
  // Debug console
  Serial.begin(115200);

  // Set LED pin as output
  pinMode(ledPin, OUTPUT);
  // Start Blynk
  Blynk.begin(auth, ssid, pass);
}

// This function will be called every time the Blynk button state changes
BLYNK_WRITE(V1)
{
  int pinValue = param.asInt(); // Get value as integer
  digitalWrite(ledPin, pinValue); // Set LED accordingly
}

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

Output:

Thus when the switch is clicked, the LED turn on.

Else the LED is in OFF State.

Result:

Thus the Blynk IOT is connected with ESP 32 for turning