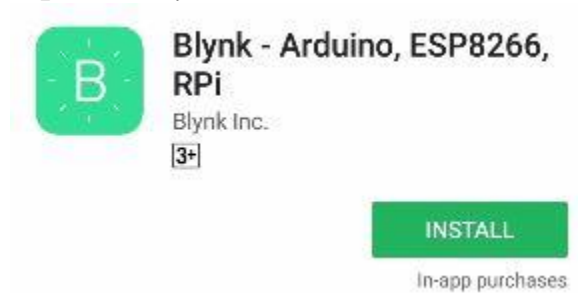**Things that you'll need:**

- NodeMCU
- Relay Board
- Jumper Cables
- Bread Board
- Smartphone.

Step 1: Downloading and Installing the Blynk App on the Smartphone.

1. Open Play store and download and install the Blynk App.

Blynk - Arduino, ESP8266, RPi
Blynk Inc.
[3+]
INSTALL
In-app purchases

2. Once the app is installed either login to it using Facebook or create a new account on Blynk and login using that. In this case I have logged in using Facebook.

3. After logging in, create a new project by clicking **'New Project'**.
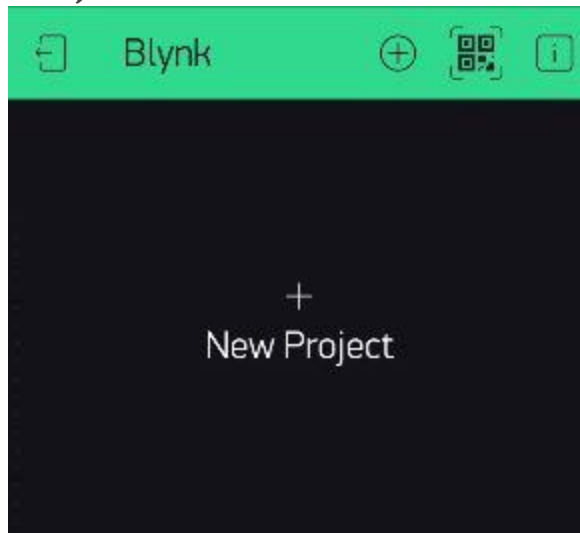
4. Give the project a name of your liking. Select the hardware device as NodeMCU and select the connection type as WIFI, and hit create.


5. At this point Blynk will send an *Auth token* to your email id. We will use this 'Auth token' later in the tutorial to link our app with the NodeMCU.

6. Now since I'm using a four-channel relay, I'll add 4 buttons on the blank project. To add a new button just click anywhere on the blank area and select button from the side menu that pops up. You can place the button anywhere on the screen.
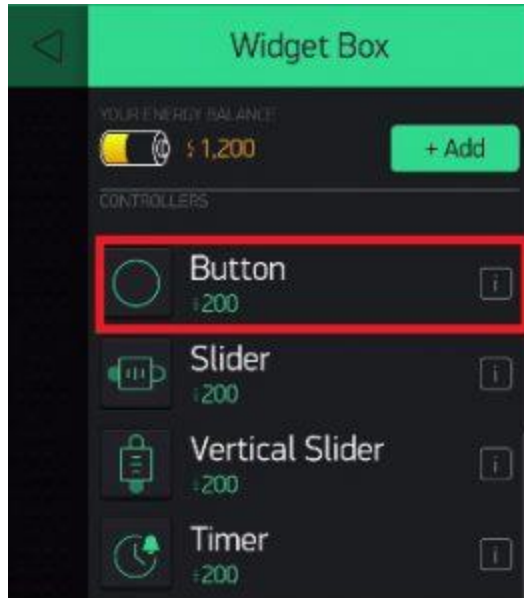
7. Click on the Button and give it a name. I'll name it *'Relay 1'* as I'll use it to control the first relay. Below the name, in

the textbox select the pin as digital pin D3.
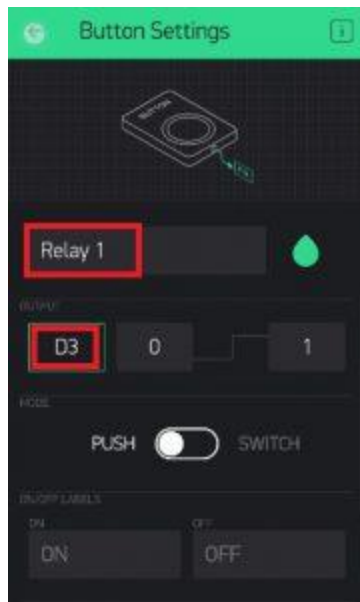
8. Repeat the process 3 more times and add three more buttons. Naming them *Relay 2*, *Relay 3* and *Relay 4* And choosing their Digital Pins as *D4*, *D5* and *D6* respectively.
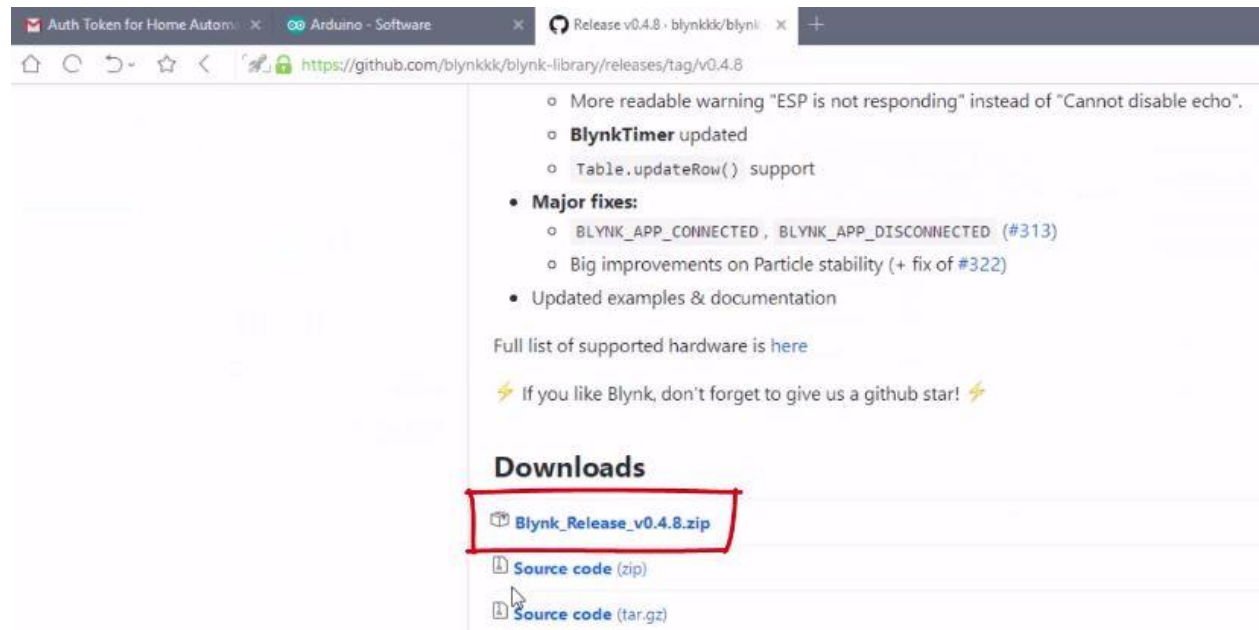

9. Now, the Blynk app is all set.

Step 2: Downloading Arduino IDE and Configuring Blynk Libraries.

1. Now let's install the Arduino IDE. Go to <u>this</u> link and download the IDE for your preferred operating system.
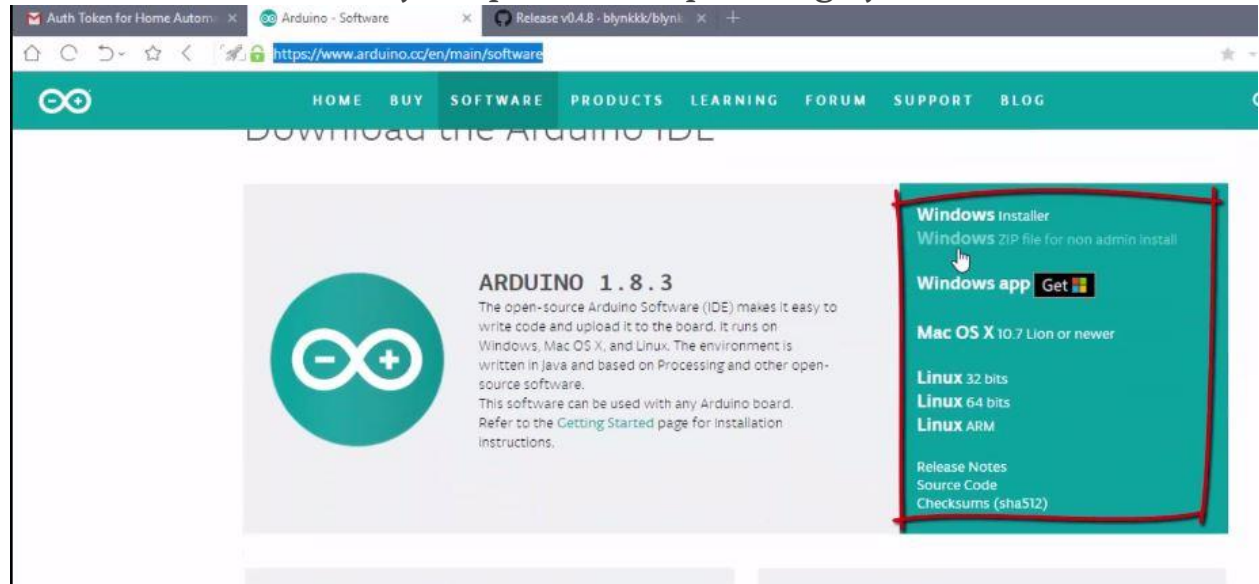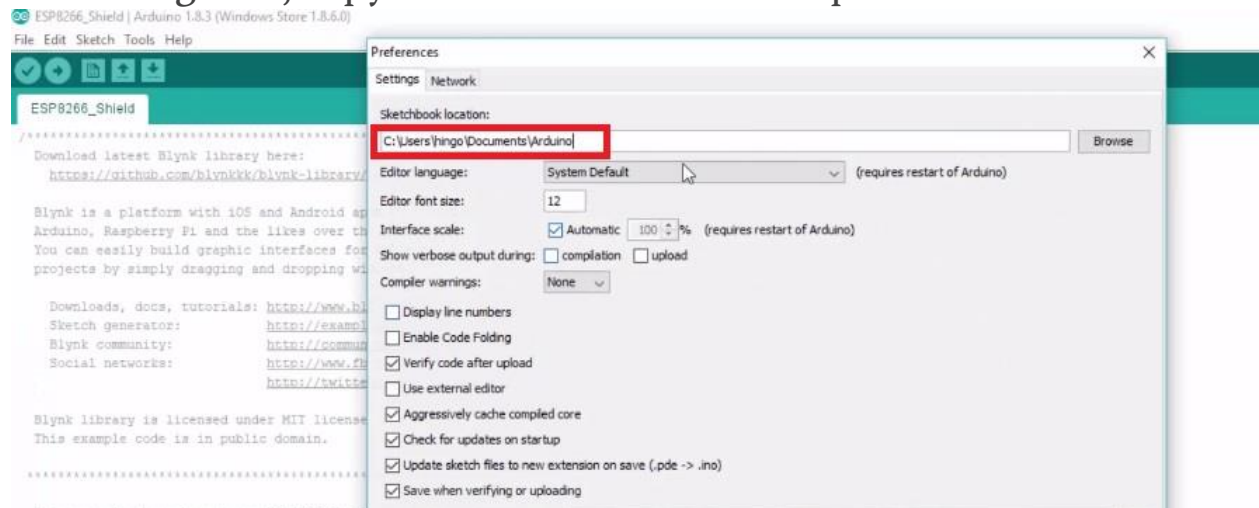

1. Download the latest Blynk libraries from this link. These libraries will help us connect the Blynk app with the

NodeMCU.

2. Extract the download zip file in a folder.

3. Open up *Arduino IDE,* go to: File > Preferences and under the *Settings* tab, copy the sketchbook location path.


4. Now open the file explorer and go to the copied path location. This is the path where all the Blynk libraries are installed. So, we'll have to copy all the newly downloaded Blynk libraries into this folder.

5. Copy the files/folders from the *Libraries* folder of the downloaded Blynk directory, and paste it to the *Libraries* folder of your Arduino IDE's directory (The path that we copied in step 5).

6. Similarly, copy the files/folder from the *Tools* folder of the downloaded Blynk directory, and paste it to the *Tools* folder of your Arduino IDE's directory.

Step 3: Uploading the code to NodeMCU.

1. Connect the NodeMCU to your PC using a USB cable.

2. Now, we'll set up the Arduino IDE by changing some settings. So, open up the Arduino IDE.

3. Go to *Tools > Port* and make sure an appropriate port is selected. In my case it's *COM 4.* This is the USB port in which
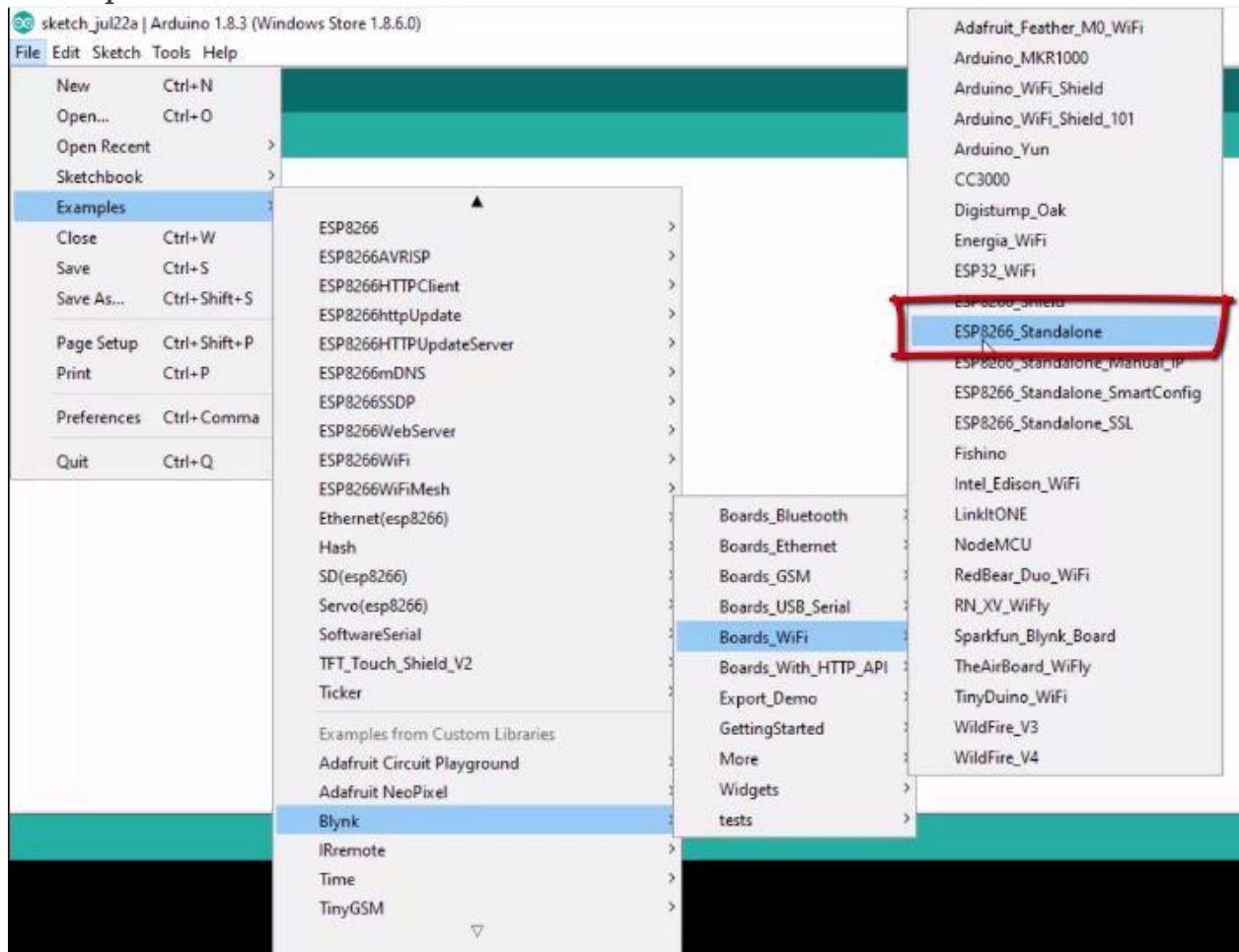
the NodeMCU is connected.
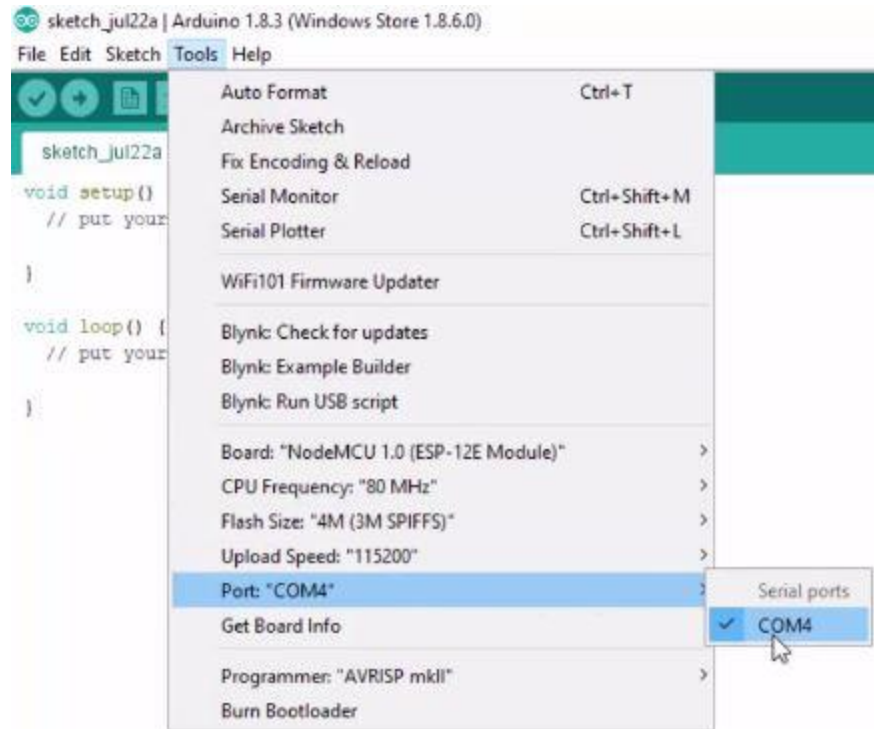
4. Now Go to *Tools > Board* and select '*NodeMCU 1.0 (ESP-12E Module)*' as the board. And that's all the settings we need to

change. So now let's begin writing some code.
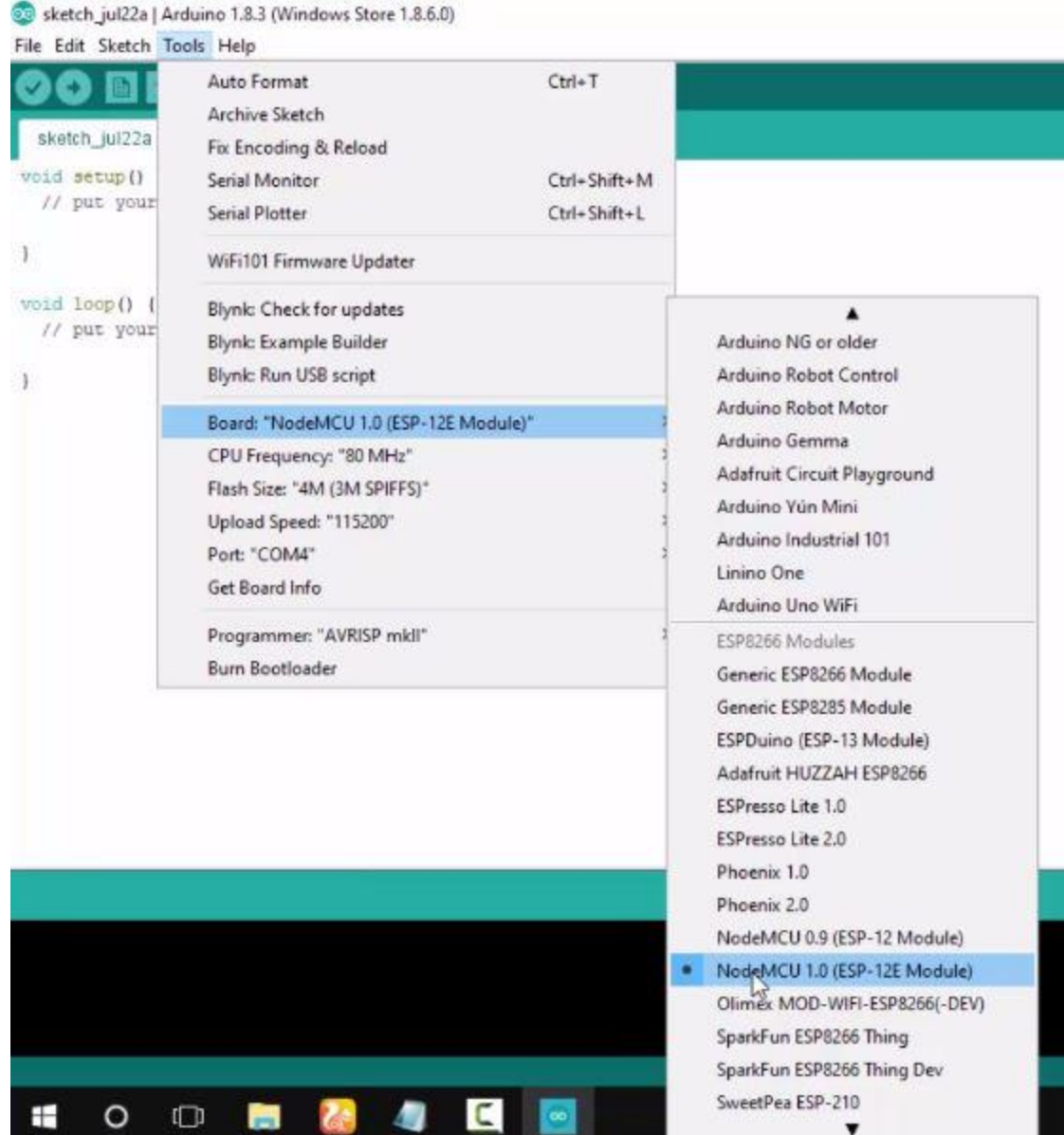
5. Go to *Files > Examples > Blynk > Boards_WIFI > ESP8266_Standalone.* A new file with some prewritten code

will open.

6. Now, in this file we only need to change 3 lines of code.

    1. Change the line where it says 'char auth[] =
       "YourAuthToken"' and replace the
       'YourAuthToken' part with your Blynk's auth token
       that you received in your email.Change the line
       where it says 'char ssid[] = "YourNetworkName"'
       and replace the 'YourNetworkName' part with the
       name of your WIFI network that you want your
       NodeMCU to connect to. In my case the name of my

WIFI network is '*The Network*'Change the line where it says 'char pass[] = "YourPassword"' and replace the 'YourPassword' part with the password of your WIFI network. In my case password of my WIFI network is '*The Network*' is 'abcd1234'.


7. That's really all the code that we need to write! We are now ready to upload this code to the NodeMCU. So directly hit upload button at the top (besides the button that has a checkmark), and wait for it to process.

8. The code will be uploaded to the NodeMCU and the next time you power it on, it will automatically connect to the specified WIFI network.

Step 4: Hardware Assembly.

1. We'll have to connect the NodeMCU with the Relay board, you can choose to do it with a bread board or without. But I prefer doing it using a Breadboard.
2. Connect the D3 pin of NodeMCU with Pin 1 of Relay. Similarly connect D4 pin of NodeMCU with Relay pin 2, D5 with Relay 3 and D6 with Relay 4.
3. Connect Ground Pin of Relay with Ground Pin of NodeMCU.

NodeMCU	Relay
D3 →	INI 1
D4 →	INI 2
D5 →	INI 3
D6 →	INI 4
GND →	GND
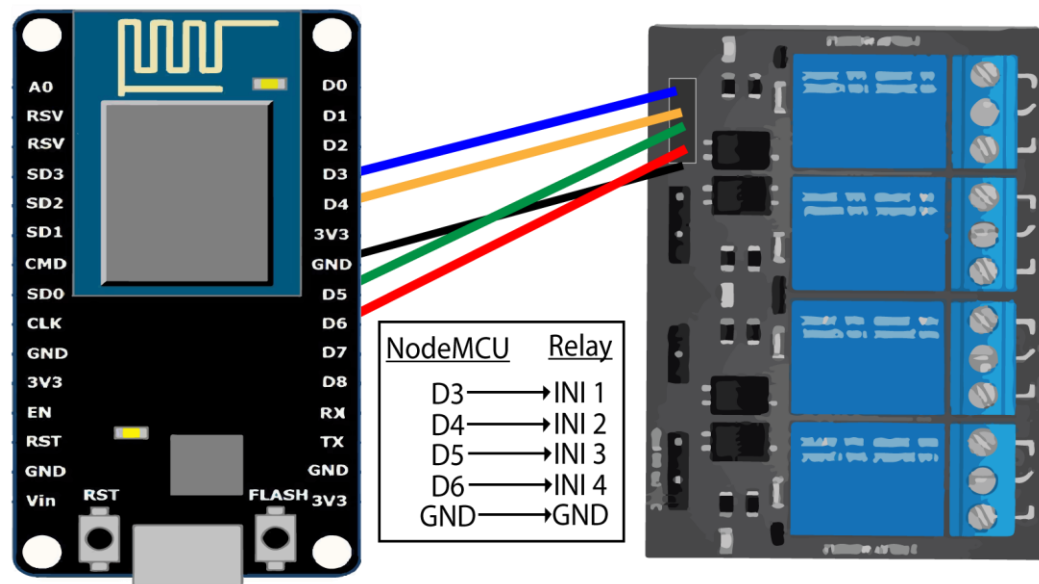
4. Now to power up the NodeMCU you can use a normal phone charger, just make sure its voltage is not too high. And to

power up the Relay board, you can use a battery or a separate breadboard power supplier.

5. As we are using a four-channel relay you can connect at most 4 electronic appliances to the Relay and control them over the internet.

6. Now if you want to connect your household appliances like Fan, Lights etc. which are connected to the main power of your house, I would recommend you take the help of a professional electrician and ask him/her to connect those appliances to the relay. Because working with the mains is no joke and if not done properly, can cause a serious damage.

At this point, we have a fully functional connection between the NodeMCU, Blynk app and our electrical appliances. So, you can directly run your Blynk project from your phone and turn the electrical appliances on or off using the buttons that we created in the app. And if you are satisfied with this and don't want to connect the NodeMCU with the Google Assistant and control the appliances using voice commands, then you don't have to read the remaining tutorial and you can stop right here. Otherwise let's move forward.

Step 5: Connecting Google Assistant (using IFTTT) to make the NodeMCU work with voice commands.

We cannot connect the Google Assistant to the NodeMCU directly, and that is the only reason we are using the Blynk app. Blynk app can

directly connect to the NodeMCU and send data to it. So, if we can send the voice commands interpreted by google assistant directly to the Blynk app, the Blynk app can then forward those commands to the NodeMCU. But the problem is Google Assistant cannot directly understand foreign commands like "turn on the fan" or "turn on relay one" etc. on its own. So, to solve this we use another intermediate app/website called 'IFTTT'.

Simply, to control our home appliances over the internet we are using NodeMCU and to connect NodeMCU with the home appliances we use a relay board. Now to send on or off signals to the NodeMCU we use our smartphone, and we do this using the Blynk app. But we want to send the on or off signals using voice commands. To do this we use google assistant in our smartphone and an app called IFTTT.

So, in the end what will happen is, when we say a voice command like "ok google turn on the light" to the Google Assistant, Google Assistant sends that this foreign command to IFTTT. IFTTT interprets this command and sends an On or Off signal to the Blynk app via the Blynk Server. Blynk will then send this signal to the NodeMCU and then to our electrical appliances.

1. Enough said, lets configure IFTTT. Go to IFTTT's website and sign up to it using your Google Account.
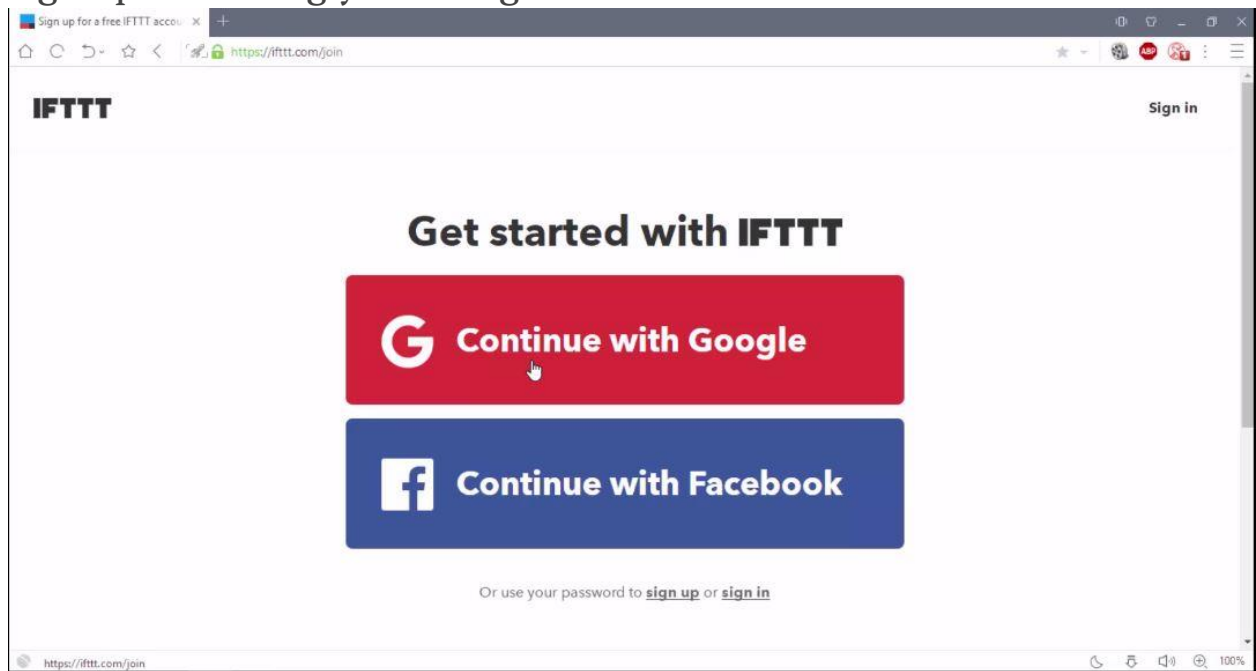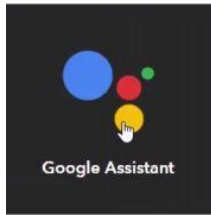

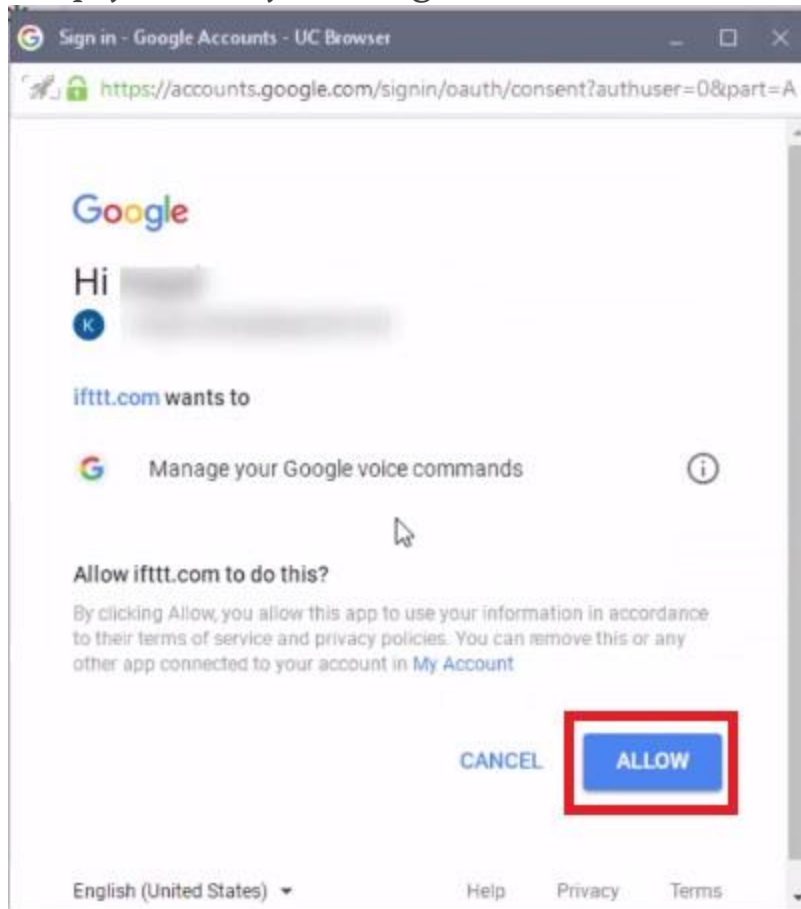2. After Signing in click on *My Applets* from the header and select *New Applet*.
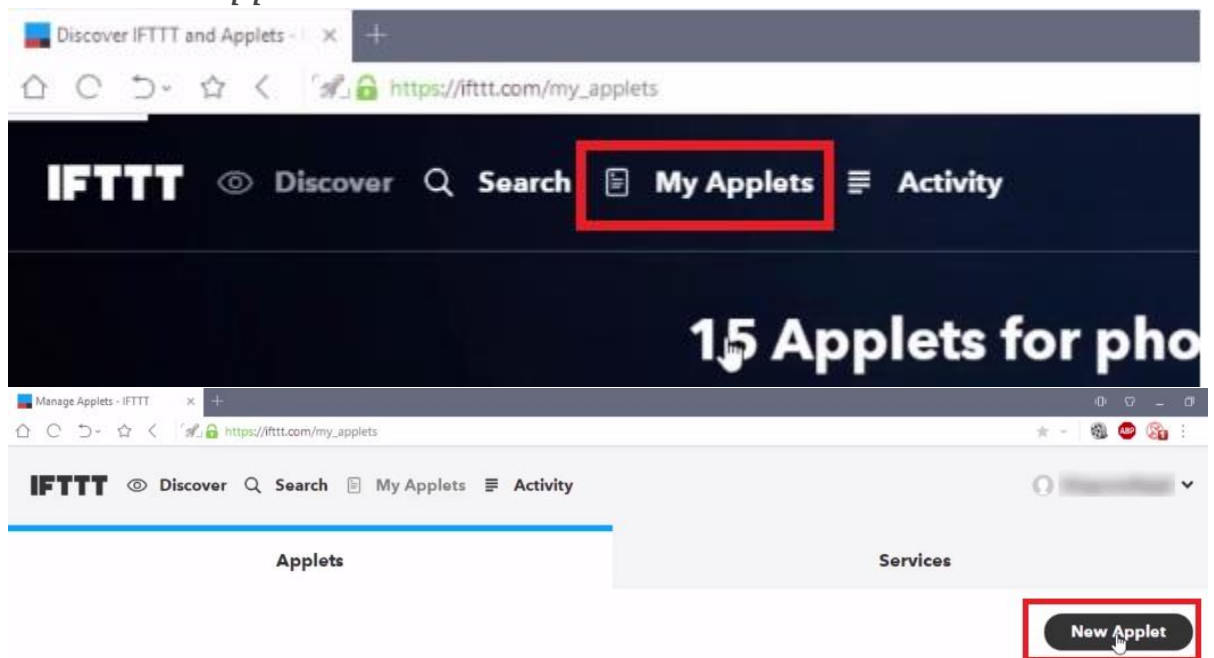
3. Click on '*this*'. **+this**

4. Search for *Google Assistant* and select it. And then Click on *Connect*.

Google Assistant

5. At this point IFTTT will ask you permission to use your google account to add voice commands to it. Which you simply allow by clicking on 'Allow'.

6. Select the card that says "Say a simple phrase".

7. Next, for the first textbox type the phrase that you want to say to Google Assistant. It can be anything such as "Turn on the T.V", "Turn on the fan" or anything you like.

8. For the next two text boxes, you write some other ways to say the first command. For example, if in the first textbox you wrote "Turn on the T.V", then in the second and third textboxes you can write something like "Turn the T.V On" or "Please Turn on the T.V" or "Turn the Idiot Box On".

9. In the fourth textbox type the reply that Google Assistant should respond with. For example, "Okay, Turning on the

T.V".

10. Finally, click on '*Create Trigger*'.

11. Now, click on that ➕**that** and type webhooks select it, and click connect. Webhooks will allow us to send **commands** to the Blynk Server.

12. Now, in the URL field type this URL:

http://188.166.206.43/ YourAuthTokenHere / update / DigitalPinToBeUpdateHere

This is the URL of Blynk Server of India, but it should work for other places as well. Replace the "YourAuthTokenHere" part with your Blynk Auth token that you received in the mail. And "DigitalPinToBeUpdateHere" part with the Digital pin of NodeMCU that is to be updated. So, as we assigned the Digital Pin D3 of NodeMCU to relay one we must write D3 in place of "DigitalPinToBeUpdateHere". But wait we cannot write D3 there, because when Blynk Server receives this command from IFTTT it assumes as if the command it received was to be sent to an 'Arduino Uno' board, but in our case, we are sending it to NodeMCU.

To solve this, we must type the Digital pin of Arduino which corresponds with the NodeMCU. You can find the mapping in the image below.
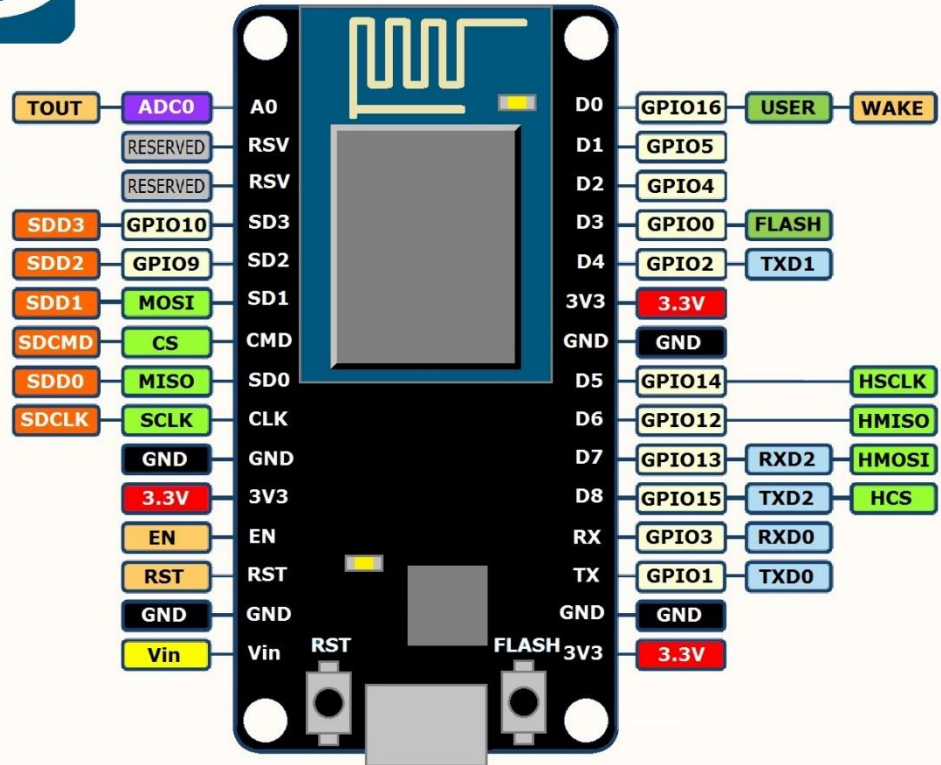
As you can see in the image, Digital Pin D3 of NodeMCU corresponds to Pin D0 of Arduino. So instead of *D3,* I'll write D0 as the pin. In the end the URL should look something like this:

http://188.166.206.43/ d5846384ef7140cfg54699bbc97f2ad8 / update / D0

13. Next, Select the 'Method' field as PUT

14. Select 'Content type' as *Application/JSON*.

15. For the 'Body' type this: ["0"]

Here '0' means to turn on,

so we are basically saying Blynk to turn on relay that is

connected to pin D3, which in our case is Relay one.

16. Now click on 'Create Action' and then Finish.

17. Similarly, we create another applet to turn off the relay. Repeat all the steps above from step 4 except the following changes:
    1. In step 8 and 9, instead of writing "Turn on the T.V", type "Turn off the T.V"
    2. In step 15, instead of ["0"], type ["1"].
18. So now we have successfully created two triggers to turn on and off one Relay. So Similarly, we create triggers for remaining 3 relays. Just change the phrase and Digital pin for each Relay. All the other steps will remain the same. So, in the end for 4 relays, we should have 8 triggers to turn each of them on or off.