IOT Appliance-- Controller

CONTENTS:

INTRODUCTION:

Congratulations for finding this document! You are going to gain a whole new insights from this document!

This document covers the development journey of speed regulator, which gives the user freedom to control the device from the BLYNK app on device, connected over a wifi connection. This speed controller circuit is developed as a part of Innovation lounge project, and comes under the **Smart** IOT converter.

Any load which is connected to this device, whether it is incandescent bulb, Fan, Dimmable LEDs can be controlled using this circuit. This is one of the reason its name.

This project comes under the domain of IOT- "Internet of Things" which is one of the emerging technology.

Document is divided into chapters, which cover the inspiration behind choosing this project, hardware information detailing about how it works, Software detailing about the BLYNK and Arduino Code. Finally all the references are mentioned from where resources and learning has been obtained.

To the conclusion of introduction, I hope This document pays an contribution in your understanding in field of IOT, Microcontrollers and electronic circuits.

INSPIRATION:

This project is done widely by many people around the world, Many companies and tech giants are on it innovating.

So why do this project? The main reason is to learn about home automation system using microcontrollers like ESP266,ESP32.

How to control our home using just a device. Our aim is to convert any device such as LED, fan to process of charging our devices into a smart system. By doing this project we have taken a step further towards this aim. We are a group consisting of hardware and software enthusiasts. This project truly is blend of the same from building a website, coding the microcontrollers, Interfacing the cloud platform to microcontrollers, designing the PCBs , actualising the PCBs, Working with electrical connections. This project has been a fruitful learning experience for all.

Hope all the of the readers, geeks, enthusiasts find this as helpful as its makers.

WORKING:

This chapter gives details of how the circuit works.

The circuit is controlled by a ESP8266(Node MCU) board. This Node MCU can easily be connected to the device, enabling the appliance control through the user's device. The platform on which controls are present is BLYNK. Operating it is easy via mobile app and web dashboard. In this chapter we are going to take a look at each part involved in some detail.

Parts include:

1) Components : Components details .
2) Electronic circuit : About connections, circuit diagram, working principle of circuit.
3) Arduino code : Arduino code with working explained.
4) PCB : About PCB designing process and PCB printing process.
5) BLYNK platform: Details about BLYNK platform, how to use it for projects.

COMPONETS USED:

N.o	List	Quantity

1	ESP8266 board	1
2	LM1117 3.3v regulator	1
3	0.1 uf ceramic capacitor	2
4	10uf electrolytic capacitor	1
5	220uf electrolytic capacitor	1
6	0.1uf polyster capacitor	1
7	0.1 uf metalised box capacitor	1
8	47Kohm 2W resistor	2
9	56 ohm 2W resistor	2
10	10K ohm resistor	1
11	1K ohm resistor	1
12	100ohm resitor	3
13	Screw Terminal	2
14	MOC3021 opto-Triac	1
15	PC817 opto coupler	1
16	Hi-Link 230V AC to 5V DC converter	1
17	Female header	1
18	Copper clad	1

19	Relay module	1
20	Connecting wires	3

ELECTRONICS CIRCUIT:

The electronic circuit of the dimmer circuit, basically controls the AC wave at output, which varies the power supplied to the appliance. The AC wave is varied according to the parameter set by user on the app/website.

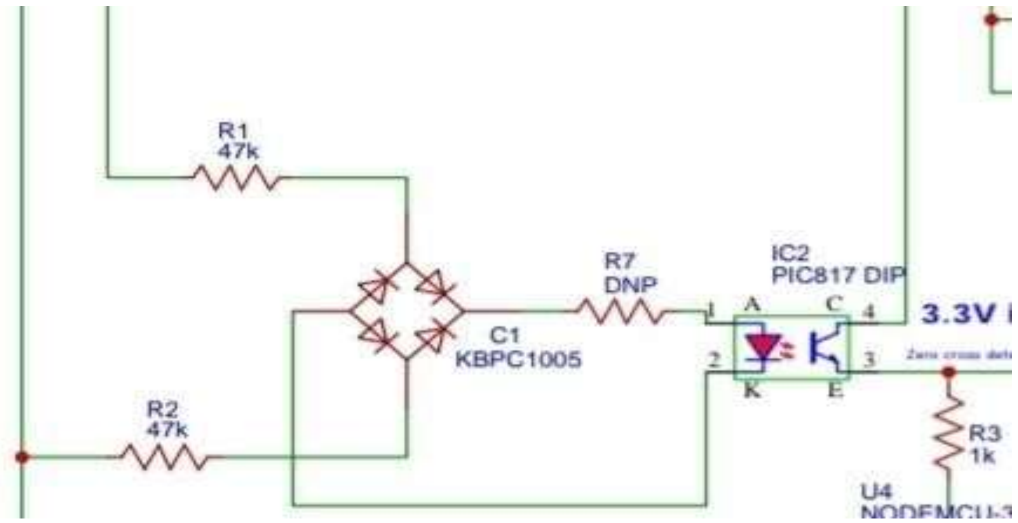
The circuit can be divided into three main parts :

1) Zero-Cross detect circuit
2) Power supply circuit
3) AC wave controller circuit

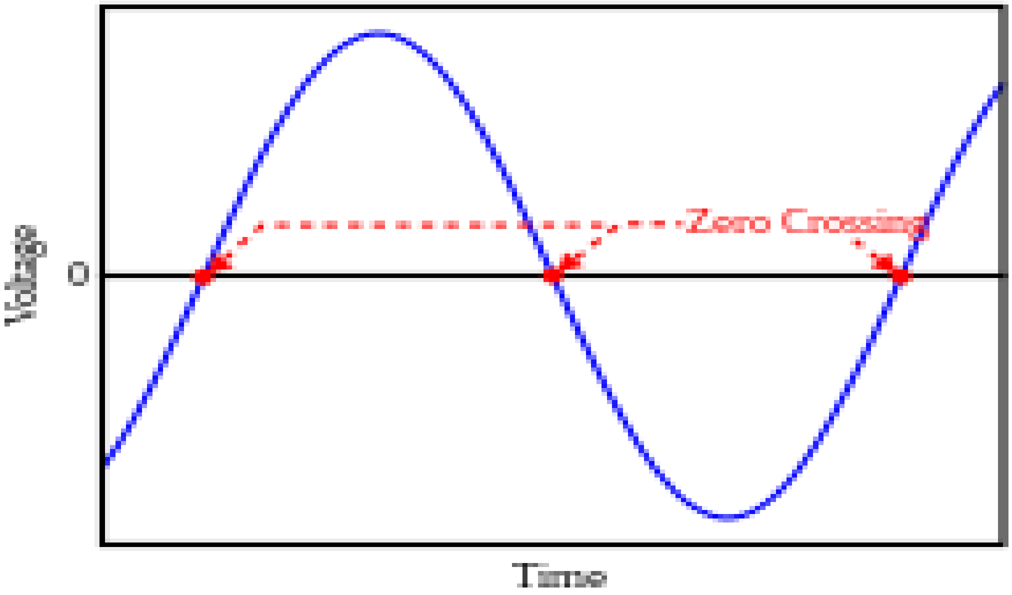
Zero-Cross detect:

This circuit comprises of a diode-bridge rectifier, 47K resistors,100ohm and 1k ohm resistors, PC817 opto-coupler.

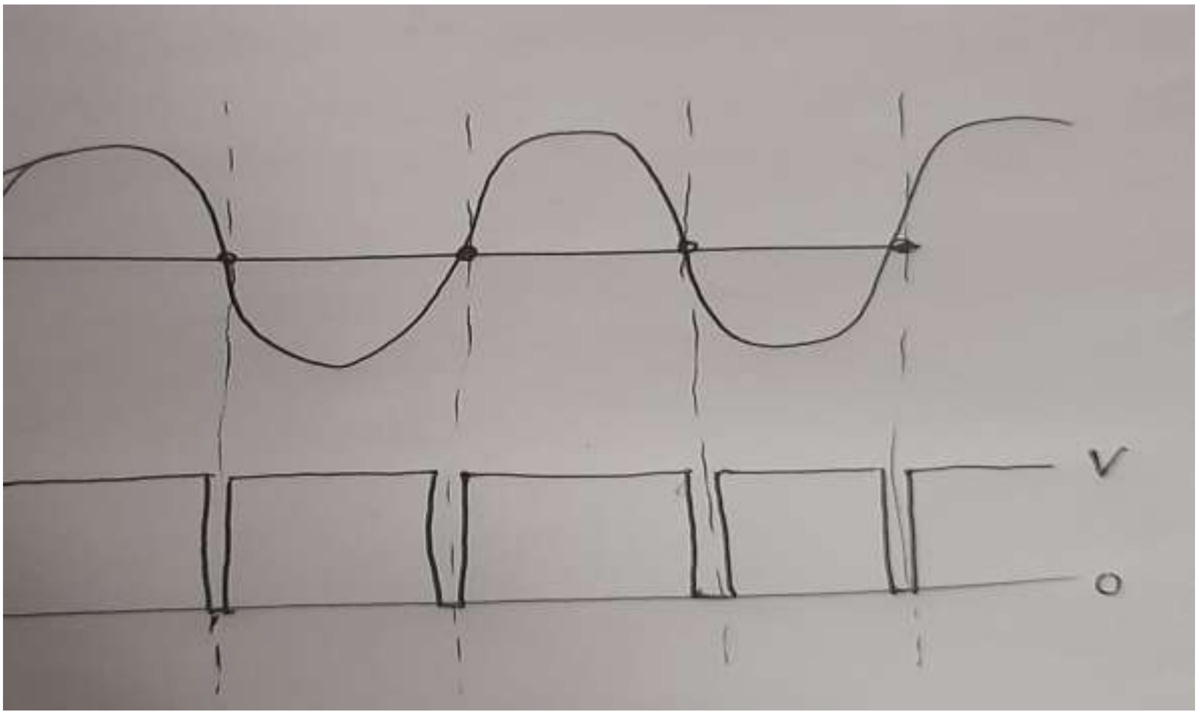
The connections are done as follows:



The pin 1,2 of opto-coupler is connected to output of diode bridge rectifier through a 100ohm resistor. Whereas the pin 4 is connected to 3.3v. at pin 3 of opto-coupler, there is a **pulldown** resistor to the zero-cross detect Output.



When the AC wave crosses the 0V mark, there is a 0V pulse generated. Represented below:



So, as the zero-crossing is detected 100times a second,this zero pulse is created 100times a second, This is the signal given to the ESP8266 pin number D2 which is IO4.

WHY 0V PULSE?

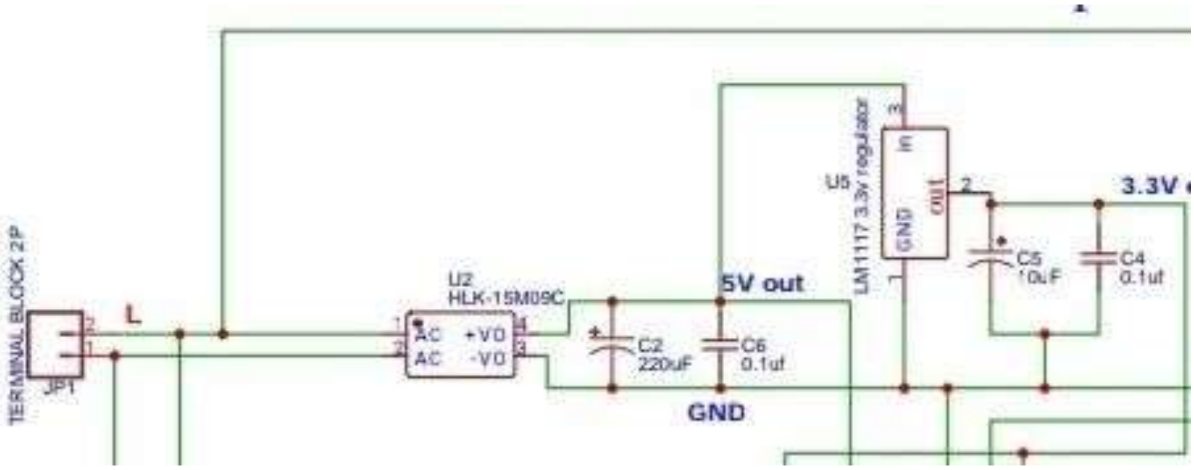
➔ The collector of transistor inside opto-coupler is connected to 3.3v supply. The emitter pin is taken as output to ESP8266, simultaneously connected to ground through 1k ohm resistor.

So, when the voltage of AC pulse is greater than 0V mark the transistor inside the opto-Coupler is ON, therefore giving HIGH signal to ESP8266. When the voltage starts decreasing and approaches 0V, The transistor turns off,giving the signal LOW .

Power Supply :

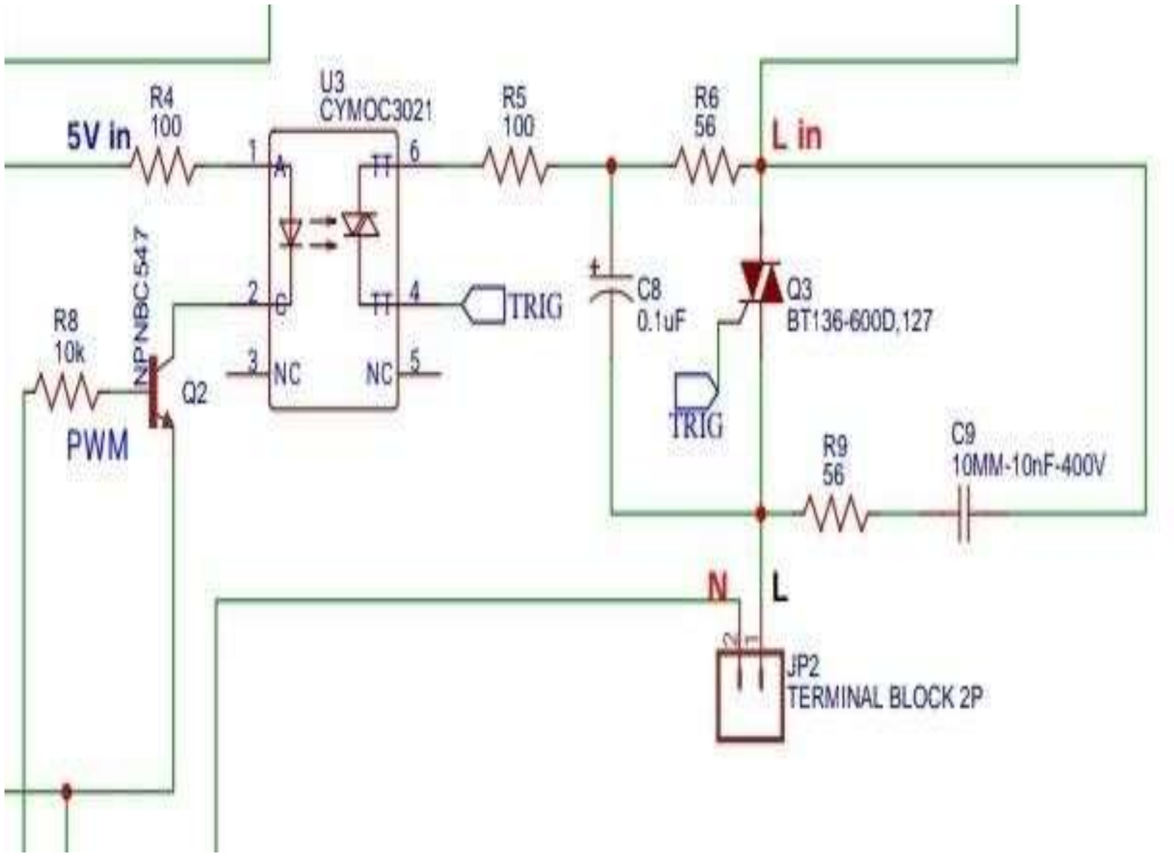
The circuit comprises of a Hi-Link module(230VAC5VDC), 220uf 0.1uf,10uf,0.1uf,Lm1117 3.3v regulator.

The Hi-link module gives 5VDC at output and 230 at INPUT. The capacitors 0.1uf,10uf work as a filter to give constant 5V DC supply. The 3.3V regulator Lm1117 further gives 3.3v regulated DC to the ESP8266.



AC wave controller:

This Circuit is very crucial part of device, This circuit consists of 100ohm and two 56ohm resistors, two 0.1uf capacitors , BT136 Triac, MOC3021 opto-Triac,BC547 npn transistor.



The Triac pin 1 is connected to live wire of AC supply,

Triac pin 2 is output, which is given to screw terminal.

Triac pin 3 is gate pin and connected to MOC3021 optotriac pin 4. The Triac BT136 can handle max current of 4A through it. Thus allowing 920W(max)[P=230V*4A] appliance to be controlled,through the circuit. However it is not advisable to connect such a high load, as it may heat up the Triac and circuit. The resistor and capacitor in parallel with triac is the snubber circuit, which enhances switching operations and protects the circuit from voltage spikes.

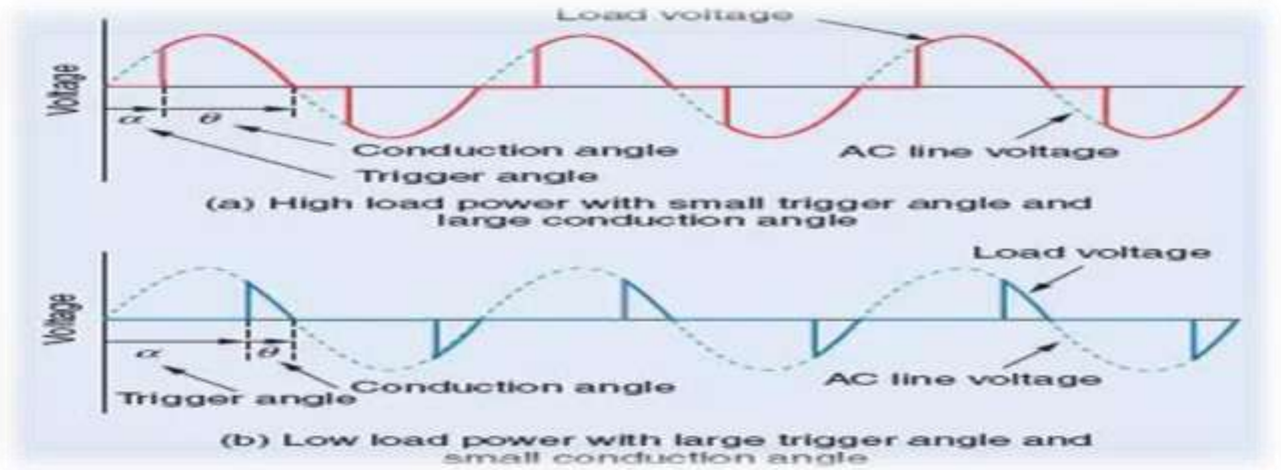
The Triac is switched on and off, from the Trigger pin of MOC3021. When Triac is switched on it allows AC current to passed, the AC current passage is allowed till the Triac is on. Otherwise no AC current is allowed to pass.

This principle is used for Manipulating the AC wave,

Which is the **phase angle control**.

Phase Angle Control:

This is the method in which a AC wave at output, is controlled. In this method only a portion of AC wave is allowed to pass , resulting to the subsequent reduction in voltage at output terminal and finally power.



(a) High load power with small trigger angle and large conduction angle

(b) Low load power with large trigger angle and small conduction angle

The trigger angle and conduction angle depends upon the position of slide icon position as set by user on the app.

Triac is controlled by the the opto-triac, which is controlled by the ESP8266. The ESP8266 is connected to the BLYNK app/website through a wifi connection.

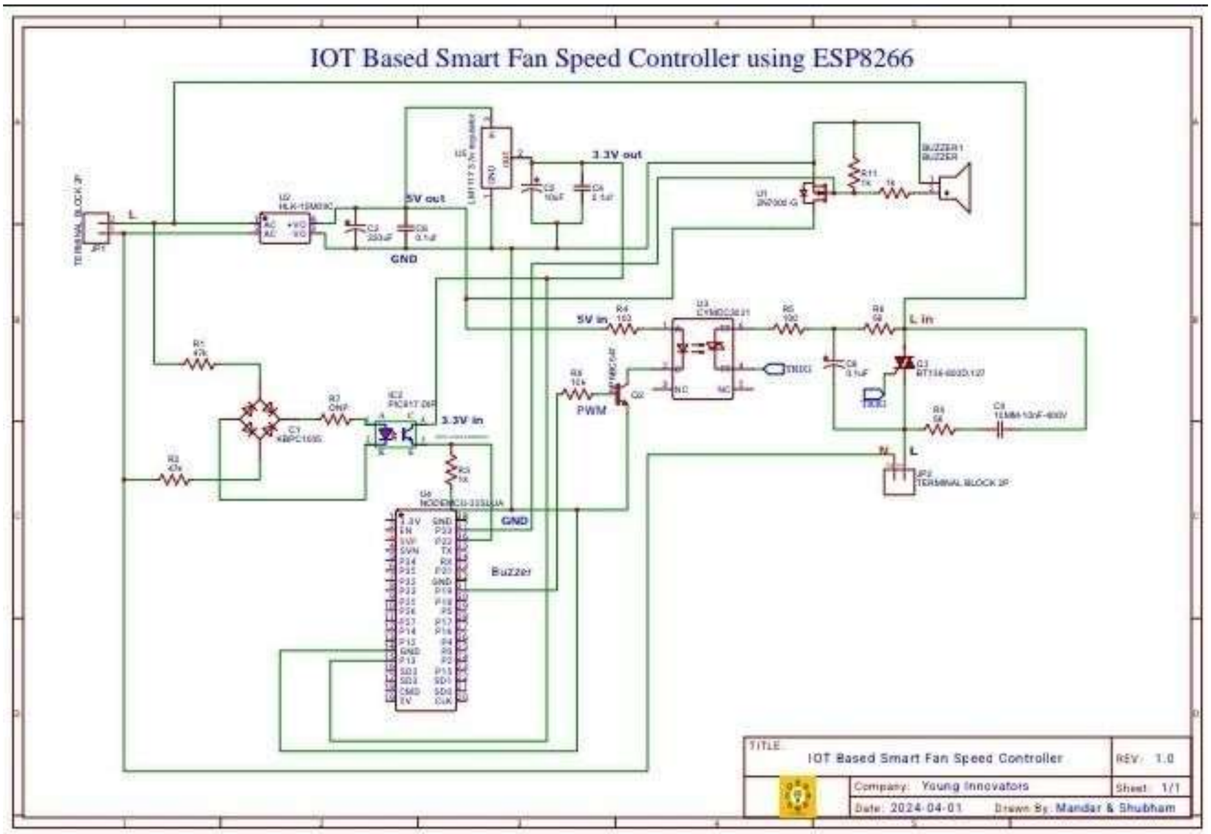
Thus, the user is able to control the AC output from his/her device remotely, while remaining connected to wifi network.

PCB

PCB which stands for printed circuit board, is used to hold all the components and their connections firmly and concisely on a piece of fibre.

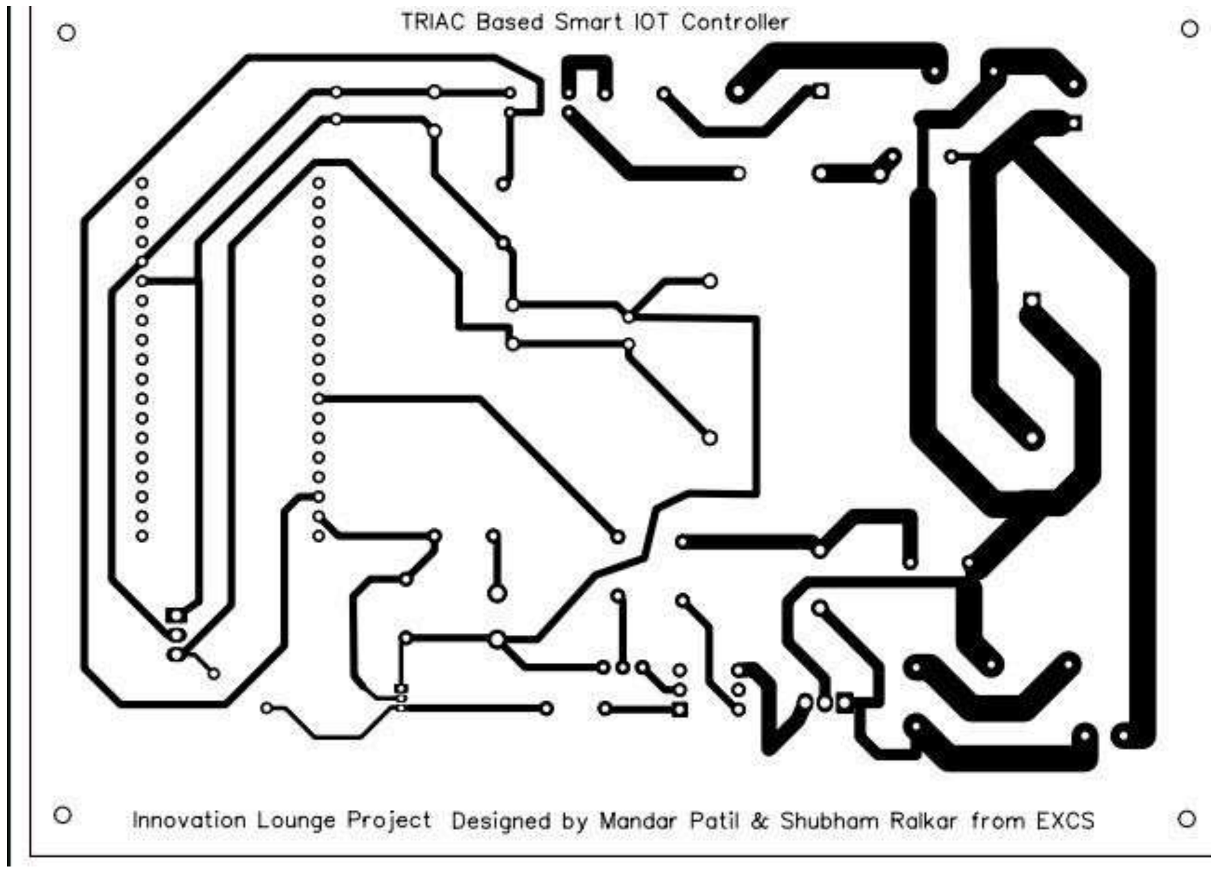
PCB connection layout for this project was designed on EasyEDA software:

Here is the full circuit diagram:



This schematic diagram has an additional circuit of buzzer, which is used as a indicator, whenever the ESP8266 recieves a notification from the user.

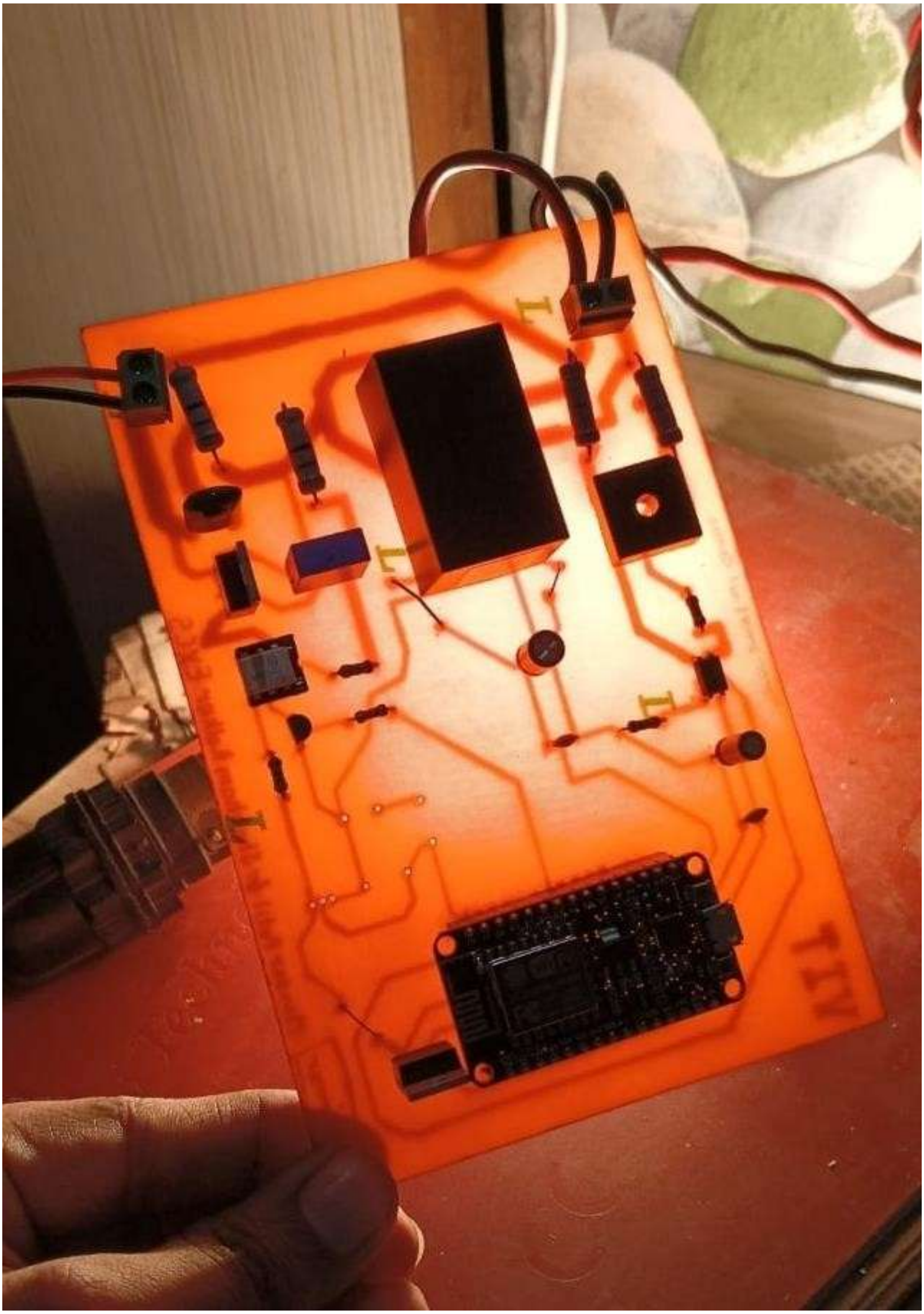
Here is the PCB layout:


TRIAC Based Smart IOT Controller
Innovation Lounge Project Designed by Mandar Patil & Shubham Ralkar from EXCS

The Journey to making a PCB is really challenging.

One has to constantly tryout new formations and keep in mind that connections don’t cross-over. The above layout was developed on EasyEDA software, it is a free and open software.

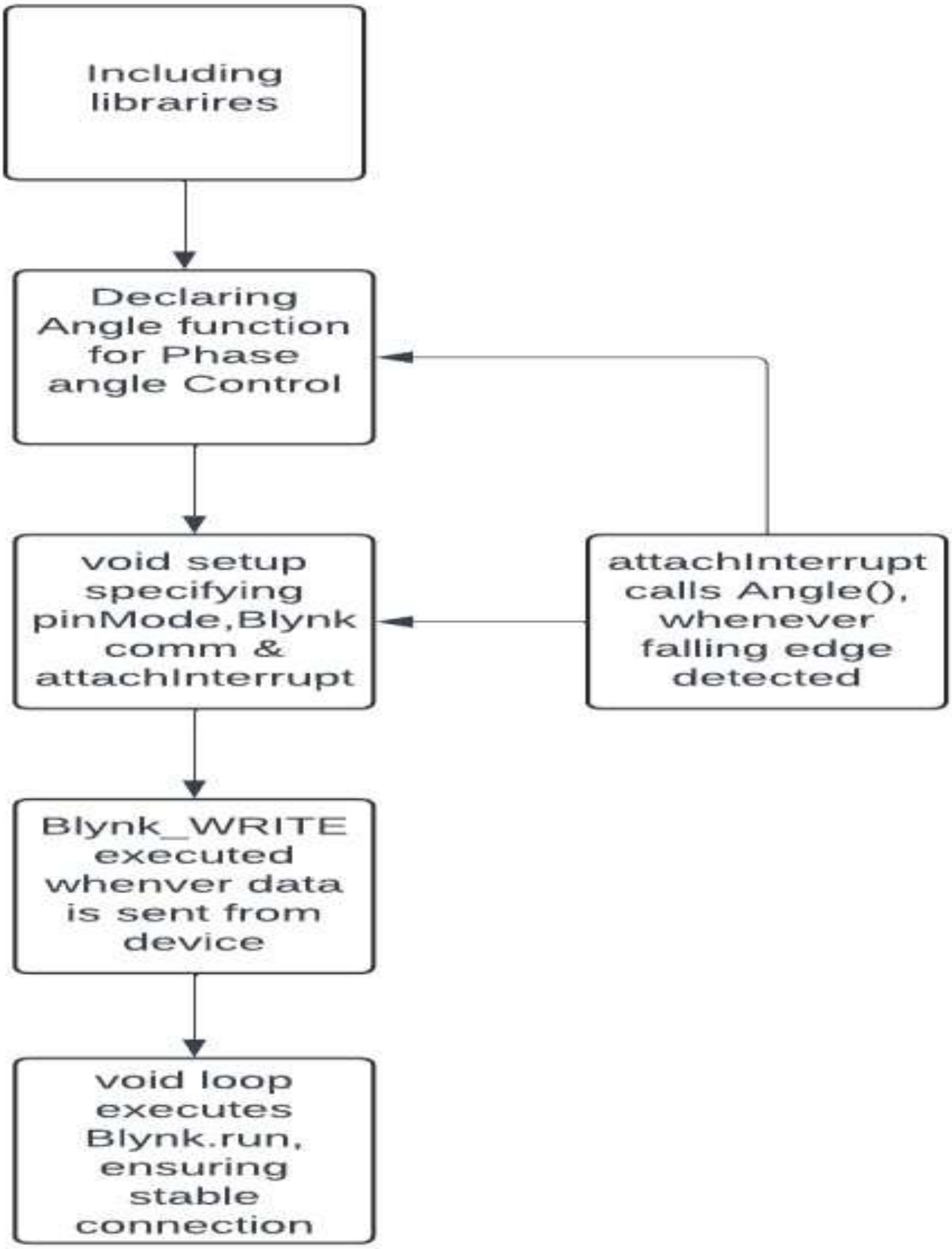
Further, this Layout was laser-printed on a sheet of glossy paper. This print was then ironed on the copper clad sheet for 3-4 minutes. At last the laser print was imprinted on the copper clad board. It was then etched with ferrous chloride solution. Holes were drilled, components soldered. Yeilding us a finished, working PCB for the circuit.

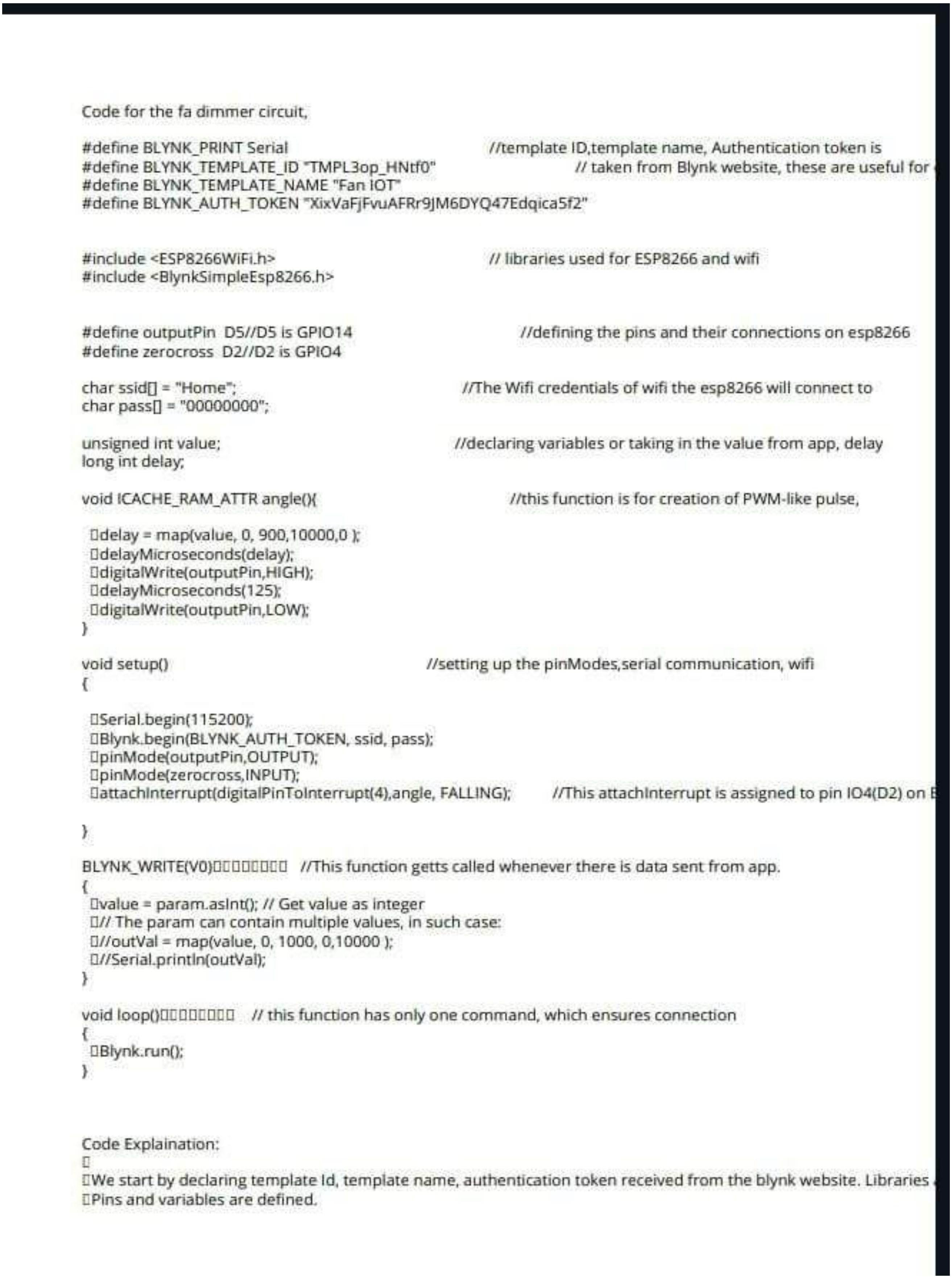


For more details about the PCB-making process checkout the PDF-2 at last of this document.

CODE

The code which is running on ESP8266,is compiled and uploaded from Arduino IDE. The flow of the code is as follows:


Including librarires
Declaring Angle function for Phase angle Control
void setup specifying pinMode,Blynk comm & attachInterrupt
attachInterrupt calls Angle(), whenever falling edge detected
Blynk_WRITE executed whenver data is sent from device
void loop executes Blynk.run, ensuring stable connection



Code for the fa dimmer circuit,

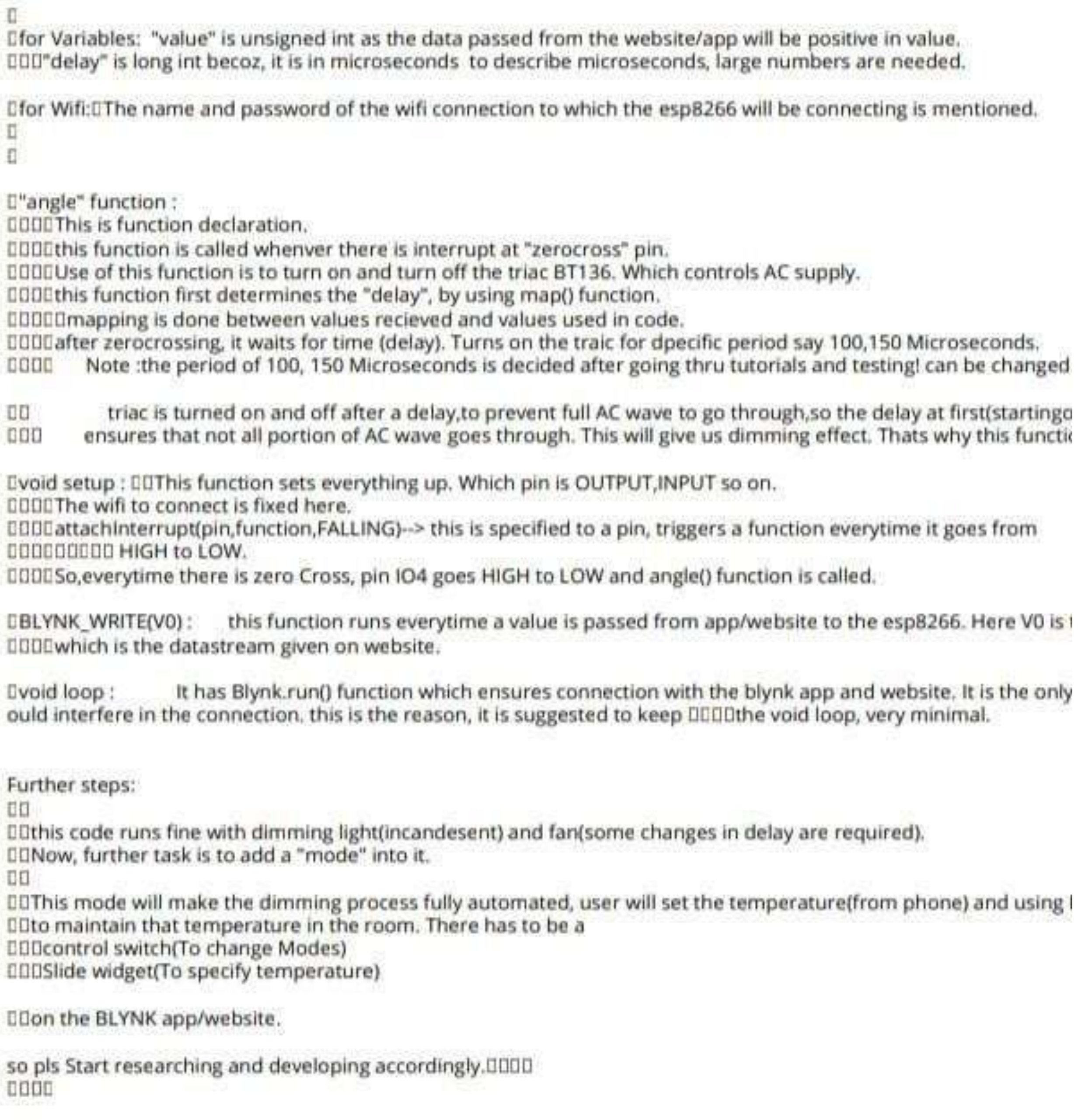
```
#define BLYNK_PRINT Serial                                    //template ID,template name, Authentication token is
#define BLYNK_TEMPLATE_ID "TMPL3op_HNtf0"                              // taken from Blynk website, these are useful for
#define BLYNK_TEMPLATE_NAME "Fan IOT"
#define BLYNK_AUTH_TOKEN "XixVaFjFvuAFRr9JM6DYQ47Edqica5f2"


#include <ESP8266WiFi.h>                                      // libraries used for ESP8266 and wifi
#include <BlynkSimpleEsp8266.h>


#define outputPin  D5//D5 is GPIO14                              //defining the pins and their connections on esp8266
#define zerocross  D2//D2 is GPIO4

char ssid[] = "Home";                                        //The Wifi credentials of wifi the esp8266 will connect to
char pass[] = "00000000";

unsigned int value;                                          //declaring variables or taking in the value from app, delay
long int delay;

void ICACHE_RAM_ATTR angle(){                                    //this function is for creation of PWM-like pulse,

 □delay = map(value, 0, 900,10000,0 );
 □delayMicroseconds(delay);
 □digitalWrite(outputPin,HIGH);
 □delayMicroseconds(125);
 □digitalWrite(outputPin,LOW);
}

void setup()                                             //setting up the pinModes,serial communication, wifi
{

 □Serial.begin(115200);
 □Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
 □pinMode(outputPin,OUTPUT);
 □pinMode(zerocross,INPUT);
 □attachInterrupt(digitalPinToInterrupt(4),angle, FALLING);       //This attachInterrupt is assigned to pin IO4(D2) on E

}

BLYNK_WRITE(V0)□□□□□□□□   //This function getts called whenever there is data sent from app.
{
 □value = param.asInt(); // Get value as integer
 □// The param can contain multiple values, in such case:
 □//outVal = map(value, 0, 1000, 0,10000 );
 □//Serial.println(outVal);
}

void loop()□□□□□□□□   // this function has only one command, which ensures connection
{
 □Blynk.run();
}
```

Code Explaination:

□

□We start by declaring template Id, template name, authentication token received from the blynk website. Libraries

□Pins and variables are defined.



□
□for Variables: "value" is unsigned int as the data passed from the website/app will be positive in value.
□□□"delay" is long int becoz, it is in microseconds to describe microseconds, large numbers are needed.

□for Wifi:□The name and password of the wifi connection to which the esp8266 will be connecting is mentioned.
□
□

□"angle" function :
□□□□This is function declaration.
□□□□this function is called whenver there is interrupt at "zerocross" pin.
□□□□Use of this function is to turn on and turn off the triac BT136. Which controls AC supply.
□□□□this function first determines the "delay", by using map() function.
□□□□□mapping is done between values recieved and values used in code.
□□□□after zerocrossing, it waits for time (delay). Turns on the traic for dpecific period say 100,150 Microseconds.
□□□□   Note :the period of 100, 150 Microseconds is decided after going thru tutorials and testing! can be changed

□□   triac is turned on and off after a delay,to prevent full AC wave to go through,so the delay at first(startingo
□□□   ensures that not all portion of AC wave goes through. This will give us dimming effect. Thats why this functi

□void setup : □□This function sets everything up. Which pin is OUTPUT,INPUT so on.
□□□□The wifi to connect is fixed here.
□□□□attachInterrupt(pin,function,FALLING)--> this is specified to a pin, triggers a function everytime it goes from
□□□□□□□□□ HIGH to LOW.
□□□□So,everytime there is zero Cross, pin IO4 goes HIGH to LOW and angle() function is called.

□BLYNK_WRITE(V0) :   this function runs everytime a value is passed from app/website to the esp8266. Here V0 is
□□□□which is the datastream given on website.

□void loop :   It has Blynk.run() function which ensures connection with the blynk app and website. It is the only
ould interfere in the connection. this is the reason, it is suggested to keep □□□□the void loop, very minimal.

Further steps:
□□
□□this code runs fine with dimming light(incandesent) and fan(some changes in delay are required).
□□Now, further task is to add a "mode" into it.
□□
□□This mode will make the dimming process fully automated, user will set the temperature(from phone) and using
□□to maintain that temperature in the room. There has to be a
□□□control switch(To change Modes)
□□□Slide widget(To specify temperature)

□□on the BLYNK app/website.

so pls Start researching and developing accordingly.□□□□
□□□□

The pdf of the above explaination is given at last.

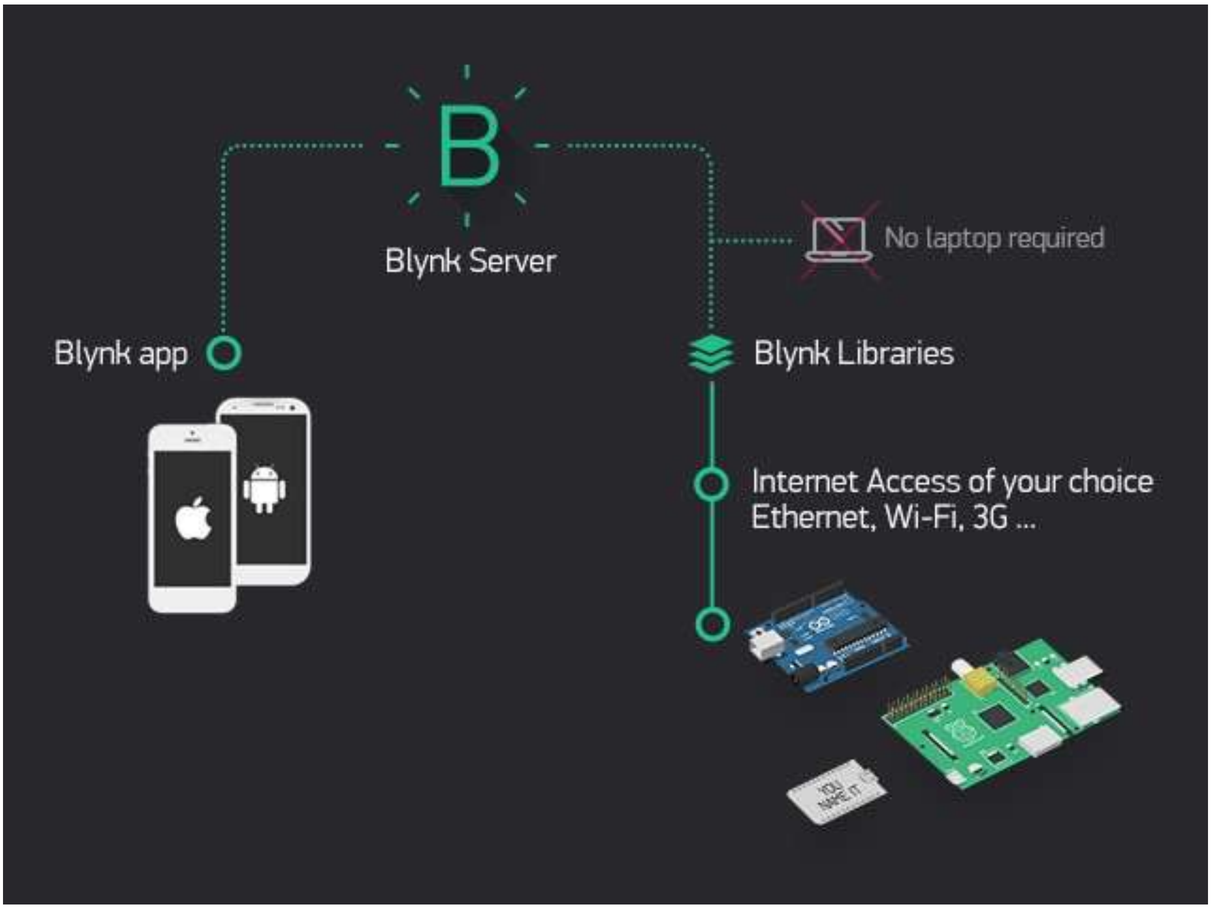
BLYNK:

Blynk platform is an open platform for making IOT based projects and products. To include BLYNK connection, there is a library named ESP8266Wifi.h which is to be included before.

Blynk library is available for download on their website:

https://docs.blynk.io/en/blynk-library-firmwareapi/installation/install-blynk-library-in-arduino-ide

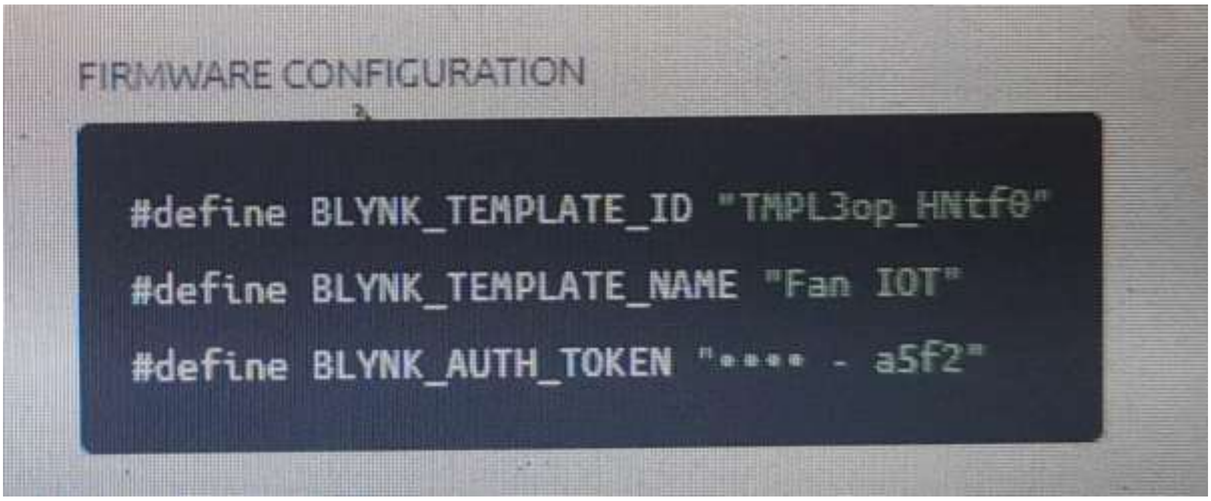
Here is the Blynk architecture:



Here is how I integrated BLYNK :

1) Create a new template on blynk.cloud 2) Create a new device for template.

3) Get the specifications and add it to code



FIRMWARE CONFIGURATION

```
#define BLYNK_TEMPLATE_ID "TMPL3op_HNtf0"
#define BLYNK_TEMPLATE_NAME "Fan IOT"
#define BLYNK_AUTH_TOKEN "•••• - a5f2"
```

Note: The Device(Phone/desktop) and microcontroller(ESP8266) should be connected to a common wifi network!

To know More about Blynk, kindly visit their website documentation section.

Link: https://docs.blynk.io/en

References:

As they say, we can achieve great feats only by standing on shoulders of Giants!! We as first year students too have experienced this. This device not only can be used for Fans but , Dimmable LEDs, Hairdryer and any appliance, using current less than 4A. Keeping the details aside, here are our Giants:

1)Mr Sourav Gupta:

https://circuitdigest.com/microcontrollerprojects/iot-based-smart-fan-control-usingesp8266-and-blynk

2) NodeMCU tutorial series by BINARYUPDATES:

https://youtube.com/playlist?list=PL4pptAPY2kIckUNYqpLpKE5qQISi1KvfS&si=8RwLQj6RYHCVwHrc

3)Phase Angle Control explained and shown:

https://youtu.be/g-2R0dNz6ik?si=N2g7M02vhGKcXJsV

Blynk Documentation and the open-source community have certainly helped in clearing our doubts and misconceptions.

Contributions:

This current project is controlling appliances using a smartphone/laptop, But further aim to add sensors and make the appliances more smarter and efficient. Here is the code PDF of the project:

https://drive.google.com/drive/folders/1709WRveohp1bBE-_E-p_j7fgjYFIXKln?usp=sharing

We would love to see any further changes and suggestions in improving the project. So please write to us for any suggestions or queries:

Contact:
shubham.ralkar@vit.edu.in
mandar.patil24@vit.edu.in aniket.saw@vit.edu.in