IoT Based Home Automation Control and Monitoring System using ESP32 and STM32

This report submitted in partial fulfillment for the requirements of the course of

Project Design Based on Communication Systems,
Course No.- ETE-3200

by

Mohammad Shihab Uddin
Roll:1704023

to the

Department of Electronics & Telecommunication Engineering,
Rajshahi University of Engineering & Technology

Rajshahi-6204, Bangladesh.

ACKNOWLEDGEMENT

At first, my deepest gratitude to the almighty that I have done the project successfully. However, it would not be possible for me without valuable support and help of many individuals. I would like to express my gratitude to all of them.

First of them, I am grateful to my respectful supervisor Jannatul Robaiat Mou mam for her excellent guidance and supervision , and also for helping me providing the required information and knowledge about this project. I highly appreciate her encouragement that helped me to complete the project. I would like to express my special thanks to our honorable Dr. Md. Kamal Hossain sir (head of the department of ETE) .My appreciation also goes to my beloved institution, Rajshahi University of Engineering and Technology for providing us such an opportunity to prove ourselves.

ABSTRACT

Internet of Things is going be the beginning of a new era of technology. IoT enables companies to automate processes and reduce labor costs. It also cuts down on waste and improves service delivery, making it less expensive to manufacture and deliver goods, as well as offering transparency into customer transactions The global IoT in the manufacturing market is expected to reach USD 53.8 billion by 2025. The reason for this increase is the growing demand for industrial automation that drives the development of IoT solutions.

Now days, if we think about smart technology, smart home is one of them. And talking about smart home will take us into IoT. The IoT home automation is the ability to control domestic appliances by electronically controlled, internet-connected systems. It may include setting complex heating and lighting systems in advance and setting alarms and home security controls, all connected by a central hub and remote-controlled by a mobile app.

IoT connected devices and machines can improve how we work and live. Real-world Internet of Things examples range from a smart home that automatically adjusts heating and lighting to a smart factory that monitors industrial machines to look for problems, then automatically adjusts to avoid failures

Now , if we talk about embedded system, there is a good relationship between IoT and Embedded system. Mainly, a system with embedded system technology becomes smarter using IoT.Any IoT device will have an embedded system. It is just used for connecting and controlling smart devices (read as embedded systems coupled with network connectivity) through Internet protocols. In both cases cases system program is required according to requirement. An embedded system may contains embedded operating system (OS) as the system requires to do a single or multiple tasks.

INDEX

Chapter-1: Introduction--01
1.1: Introduction about the Project--01
1.2: Brief Description and Operation---------------------------------------01
1.3: Application--02

Chapter-2: Project Details--03
2.1: Project Element and Required Apparatus----------------------------03
2.1.1:Hardware ---03
2.1.2: Software---07
2.2: Working Flowchart --09
2.2.1: Controlling Part --09
2.2.2: Monitoring Part --09
2.2.3: Load-side---10
2.3: Operation of Circuit and Other Peripherals-------------------------11
2.3.1: ESP-32 Part---13
2.3.2: STM-32 Part--15
2.3.3: Relay Module and Loads Part-----------------------------------18
2.4: Printed Circuit Board--19
2.4.1: Main PCB--19
2.4.2: Details about PCB --20
2.4.3: PCB View Based on Different Category-----------------------21

Chapter-3: Conclusion--23
3.1: Conclusion of the Project---23
3.2: Future Updates: --23

Chapter-1

1.1: Introduction about Project:

As the name suggests, the project is about IoT based home automation control and monitoring system. This project will give us an idea about smart home technology. The project is mainly combination of Iot and Embedded system. IoT device embedded systems are a combination of hardware and firmware along with internet connectivity to perform specific tasks. These devices transfer real-time data over the internet for a broader use case like monitoring, tracking, analysing, or more. Both hardware and programming are required in both part.

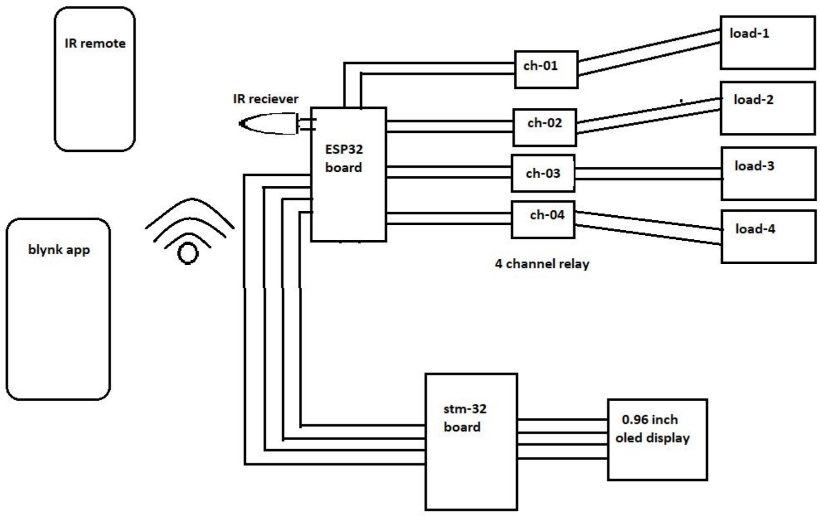

Figure 1.1: basic diagram for the project

One with STM32 part is mainly to monitor the system, which contains a display, where all the system functionality will be displayed. It is mainly connected with the main ESP32 board directly which performs the whole controlling part of the system.
The another part of the project is with ESP32 . This board mainly take command and give the input to the monitoring part so that all the real time responses can be displayed into the display

1.2 Brief Description and Operation:

As mentioned previously, this project can be divided into three main parts.
Those are:
Part-1: this portion of project is responsible for controlling all the loads of the project. The main part in here is the ESP32 board which has the built in wi-fi module. Also it performs the tasks according to given to it in program. Beside of controlling all the load , it also sends the signal to the monitoring part and then rest of the task is done by that monitoring part
Part-2: This part is used to monitor the whole system. A 0.96 inch OLED display is used to display all the status. Also in this part, an astable multivibrator is used to provide pulse into STM32 board. To drive the display,one STM32 board is used and the board contains all the program code for driving display.This program can be an

example of embedded OS which was developed based on some conditions, criteria and algorithm.

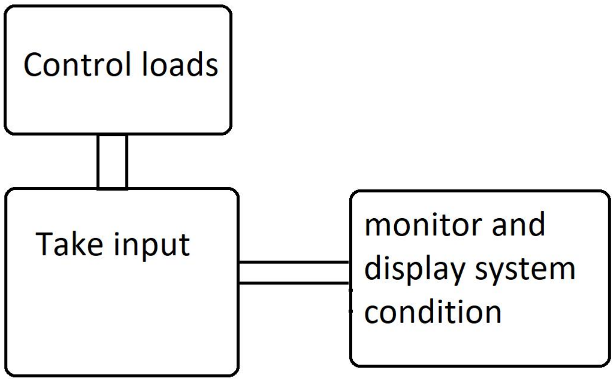

Figure 1.2: partition of the system in block diagram

Part-3: This is just an addition of the project, which contains a 4 channel 5 volt relay. As all the loads which are being controlled are electrical load, an isolation is required between main circuit part and higher voltage load

1.3: Application:

The application of IoT as well as embedded system is huge. The Fourth Industrial Revolution is the current and developing environment in which disruptive technologies and trends such as the Internet of Things (IoT).Embedded systems in IoT are used in home automation, health and wellness, and security, among others. Likewise, an embedded system in IIoT is used in remote sensing and control for water, gas, utility meters, shipping/transportation management, robotics integration, and so on. Consumer electronics include MP3 players, television sets, mobile phones, video game consoles, digital cameras, GPS receivers, and printers. Household appliances, such as microwave ovens, washing machines and dishwashers, include embedded systems to provide flexibility, efficiency and features. Also,IoT enables companies to automate processes and reduce labor costs. It also cuts down on waste and improves service delivery, making it less expensive to manufacture and deliver goods, as well as offering transparency into customer transactions.

Chapter-2: Project Details

This chapter contains all details about the project. To complete this project , several steps have been followed such as, creating schematic of the circuit, running trial in bread board, designing PCB , making the PCB, putting all component in the pcb , then running trial again, troubleshooting if required. All the steps and parts will be discussed here in details as much as possible.

2.1: Project Elements and Required Apparatus:

There are several numbers of software are required for designing and programming purpose. For programming, two Integrated Development Environment were used. Again, as the project is hardware based, significant number of hardware have been used here. All of them are given bellow:

2.1.1:Hardware :

- ESP32 development board
- STM32 development board
- ST-Link debugger
- SSD1306 OLED display
- 4 channel 5 volt relay
- IR remote with IR receiver
- 12-0-12 center tap transformer
- 12 volt DC bulb (4 pcs)

ESP32 development board: ESP32 is a low-cost System on Chip (SoC) Microcontroller from Espressif Systems, the developers of the famous ESP8266 SoC. It is a successor to ESP8266 SoC and comes in both single-core and dual-core variations of the Tensilica’s 32-bit Xtensa LX6 Microprocessor with integrated Wi-Fi and Bluetooth.

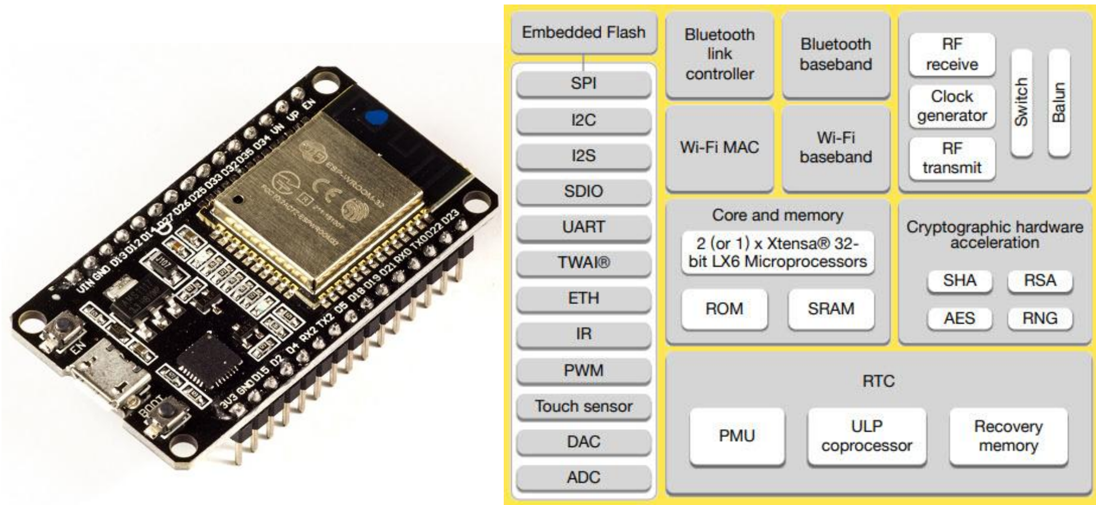

Figure 2.1.1: ESP-32 with it’s block diagram

Hardware specification for the board is given bellow:

- Single or Dual-Core 32-bit LX6 Microprocessor with clock frequency up to 240 MHz.
- 520 KB of SRAM, 448 KB of ROM and 16 KB of RTC SRAM.
- Supports 802.11 b/g/n Wi-Fi connectivity with speeds up to 150 Mbps.
- Support for both Classic Bluetooth v4.2 and BLE specifications.
- 34 Programmable GPIOs.
- Up to 18 channels of 12-bit SAR ADC and 2 channels of 8-bit DAC
- Serial Connectivity include 4 x SPI, 2 x I2C, 2 x I2S, 3 x UART.
- Ethernet MAC for physical LAN Communication (requires external PHY).

STM32 development board: The STM32 development board contains STM32F103C8T6 micro-controller which is a medium density performance line, ARM Cortex-M3 32bit microcontroller in 48 pin LQFP package. It incorporates high performance RISC core with 72MHz operating frequency, high speed embedded memories, extensive range of enhanced I/Os and peripherals connected to two APB buses. The STM32F103C8T6 features 12bit ADC, timers, PWM timer, standard and advanced communication interfaces. A comprehensive set of power saving mode allows the design of low power applications.

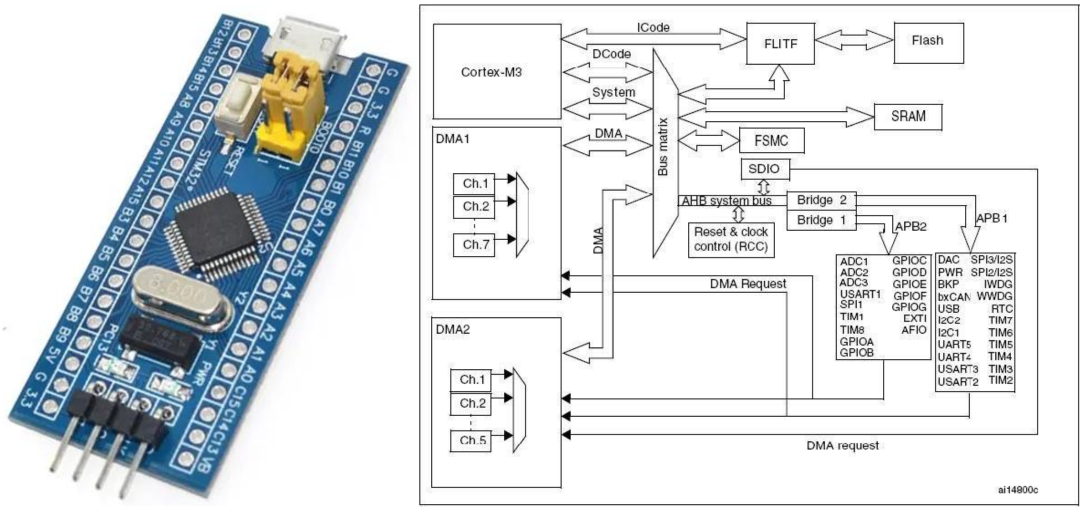

Figure 2.1.2: STM32 with it’s block diagram

It has following features:

- Operating voltage range from 2V to 3.6V
- 64Kbytes of flash memory
- 20Kbytes of SRAM
- CRC calculation unit, 96bit unique ID
- Two 12bit, 1µs A/D converter (up to 10 channels)

ST-Link debugger: Use this device to send program code to the 32-bit STM32 ARM Cortex or 8-bit STM8 microcontrollers. The header for this ST-Link v2 programmer contains connections for 5V, 3.3V, SWCLK, SWDIO, SWIM, Reset (RST/NRST) and GND. The connector on the opposite side of this device is a USB connector and is intended to be plugged into the computer.It is an in-circuit debugger and programmer for the STM8 and STM32 microcontroller families. ST Link V2 Mini Programmer is Compatible Programmer & Emulator / Debugging dongle. ST Link V2 Mini Programmer is little USB stick contains a microcontroller with the code for ST’s classic programmer/emulator for STM8 and STM32 bit processors.

Figure 2.1.3: ST-Link debugger and programmer

SSD1306 OLED display:SSD1306 is a single-chip CMOS OLED/PLED driver with controller for organic / polymer light emitting diode dot-matrix graphic display system. It consists of 128 segments and 64commons. This IC is designed for Common Cathode type OLED panel

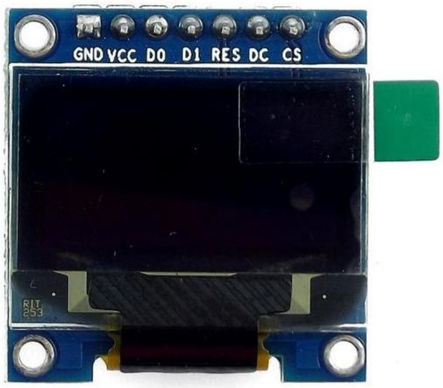

Figure 2.1.4: SSD1306 OLED display

4 channel 5 volt relay: The 4 Channel Relay Module is a convenient board which can be used to control high voltage, high current load such as motor, solenoid valves, lamps and AC load. It is designed to interface with microcontroller

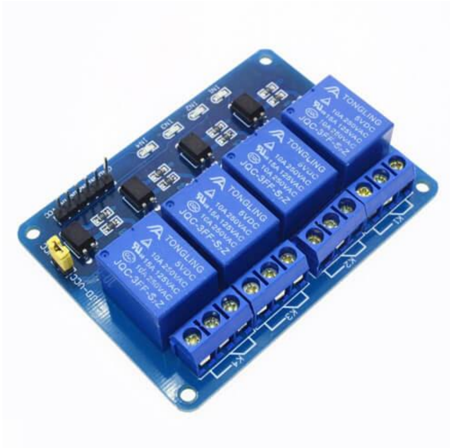

Figure 2.1.5: image of 4 channel 5 volt relay

IR remote with IR receiver: It is a complete Infrared remote control kit any microcontroller project.This kit comes with a remote commander, the IR Receiver, and additional LED for status control. This can be used by connecting the IR receiver to the microcontroller using IR receiver.

Figure 2.1.6: Image of IR remote with receiver

12-0-12 center tap transformer:A center-tap transformer is designed to provide two separate secondary voltages, VA and VB with a common connection. This type of transformer configuration produces a two-phase, 3-wire supply. The secondary voltages are the same and proportional to the supply voltage, VP, therefore power in each winding is the same.

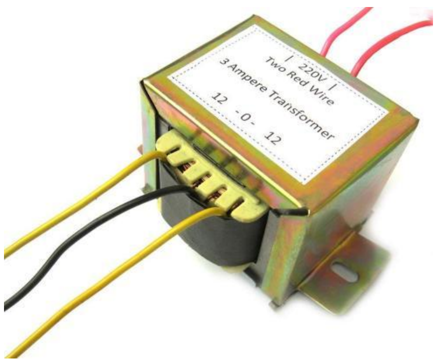

Figure 2.1.7 : center tap transformer

12 volt dc bulb: DC light bulbs are good for remote and self standing power supplies , such as DC batteries , cars , boats , yachts , solar off grid systems , trains , airplanes and more

2.1.2: Software:

- ❖ STM32CubeIDE
- ❖ BLYNK IoT platform
- ❖ Arduino IDE
- ❖ EasyEDA

STM32CubeIDE:STM32CubeIDE is an all-in-one multi-OS development tool, which is part of the STM32Cube software ecosystem. STM32CubeIde Board PhotoSTM32CubeIDE is an advanced C/C++ development platform with peripheral configuration, code generation, code compilation, and debug features for STM32 microcontrollers and microprocessors. It is based on the Eclipse®/CDT™ framework and GCC toolchain for the development, and GDB for the debugging. It allows the integration of the hundreds of existing plugins that complete the features of the Eclipse® IDE.

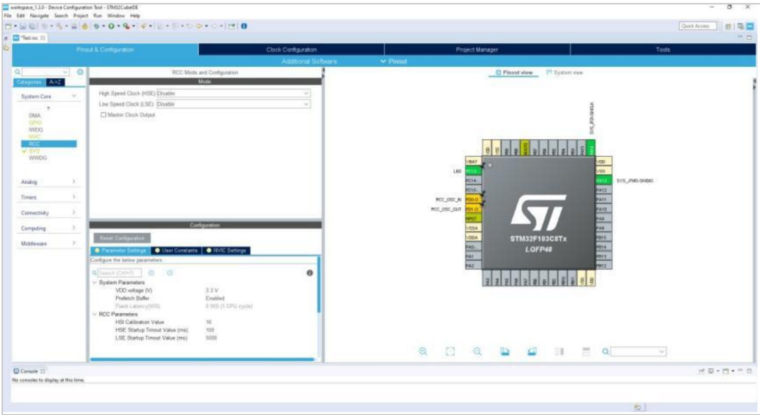

Figure 2.1.8:STM32CubeIDE interface

BLYNK IoT platform: Blynk is an IoT platform for iOS or Android smartphones that is used to control Arduino, Raspberry Pi and NodeMCU via the Internet. This application is used to create a graphical interface or human machine interface (HMI) by compiling and providing the appropriate address on the available widgets

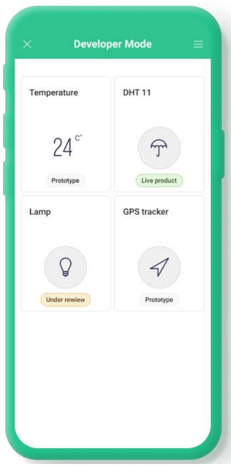

Figure 2.1.9: blynk mobile GUI

Arduino IDE: The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

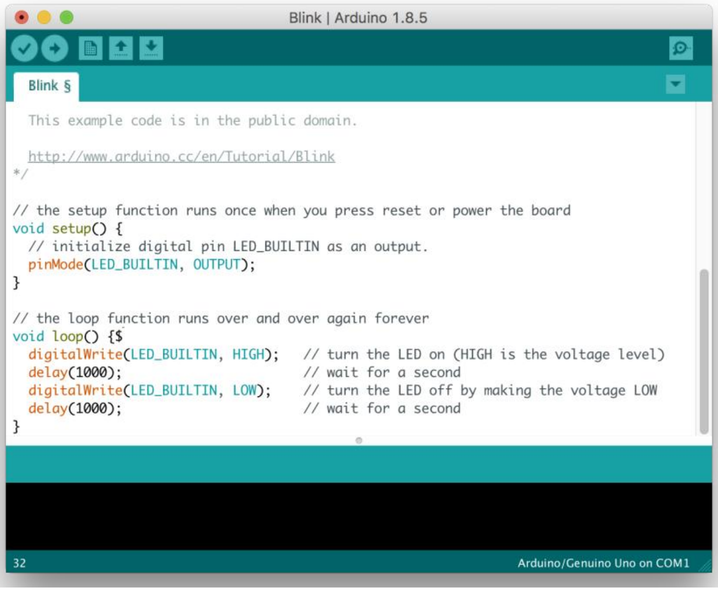

Figure 2.1.10: arduino IDE

EasyEDA: EasyEDA is an easier and powerful online PCB design tool that allows electronics engineers, educators, students, makers, and enthusiasts to design and share their projects. This is a design tool integrated LCSC components catalog and JLCPCB PCB service that helps users to save time to make their ideas into real products.It is a web-based EDA tool suite that enables hardware engineers to design, simulate, share - publicly and privately - and discuss schematics, simulations and printed circuit boards

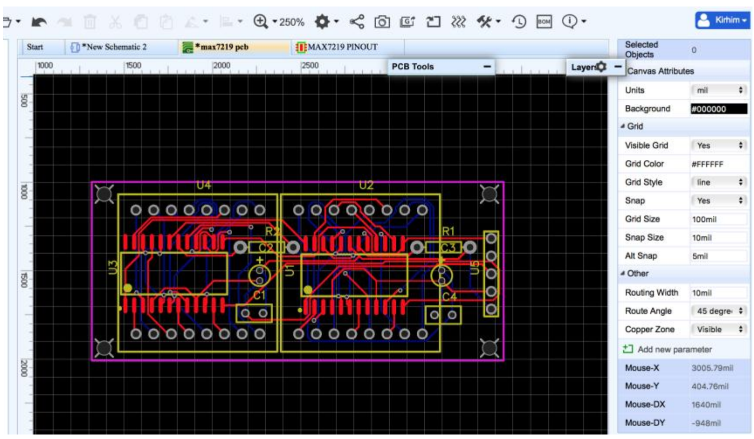

Figure 2.1.11: EasyEDA with PCB layout

2.2: Working Flowchart:

As the project has several number of functionality. It is designed based on some steps and algorithm. Also, the system can be represented by following flowchart.

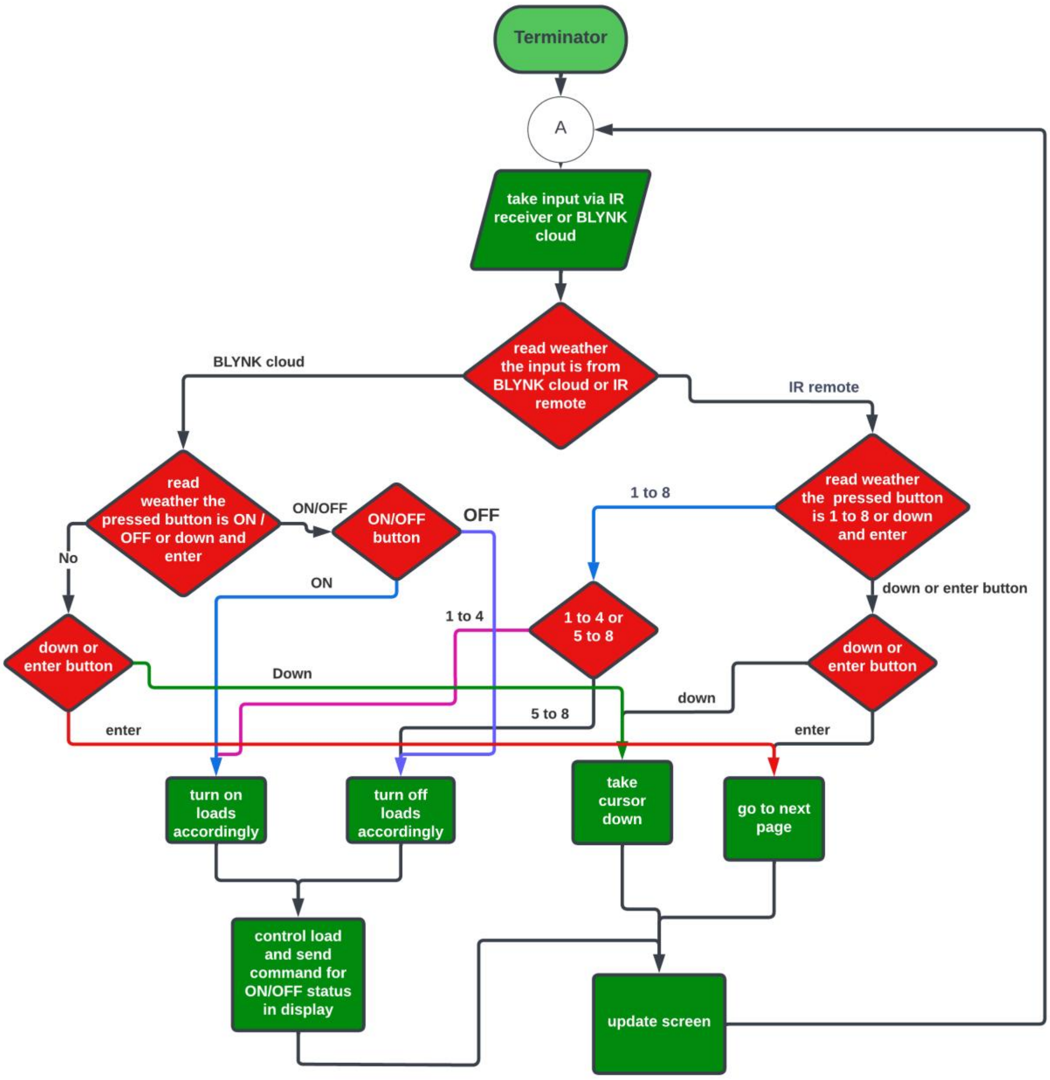

Figure 2.2.1: Flow-chart for the project

2.2.1: Controlling Part:

In the whole system , this part is responsible for the change of load state as well as monitoring display. The system just makes response according to this part. This can either turn ON the load as well as turn OFF the load. Also it can make change in the display screen. Now, the input in the controlling part can be given in two ways, one is via IR remote and another is via BLYNK IoT cloud or app. Both can do the same task. One is cloud based and another one does not require internet to work.

2.2.2: Monitoring Part:

This is the most significant part in this project. The 0.96 inch OLED display plays the rule to display all the real-time status of the whole system. Which is driven by the STM32 board. There are several numbers of interrupt buttons that work in this part.

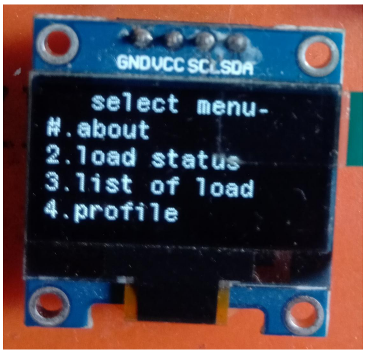

Figure 2.2.2: Main page of the display

Two of them are reserved for controlling display. And another one is to receive pulse from the astable multi-vibrator circuit. There are highest 3 pages in the display. Each of the menu can be selected from a page by taking cursor down.

2.2.3: Load-side:
This is an addition of the project. All the switches has been used are electronic switch and they have lower current and voltage rating .That's why a relay module has been used to turn on and off the loads. Those relay module can be controlled by 5 volt and can operate the load of up-to 220 volt. The relay module has been used in the project was 4 channel 5 volt relay module . The switch of the relay module can be turned on or off based on incoming output from the controlling part or more specifically, the ESP32 board.

2.3: Operation of Circuit and Other Peripherals:

With a closer view in the hardware section of this project, it can be explained in details by dividing it into some smaller parts. Each of the part works differently than other. There is no similarity of coding or hardware peripheral one with another. The circuit diagram for the project is given bellow. All the parts and peripherals are connected to it as given bellow.

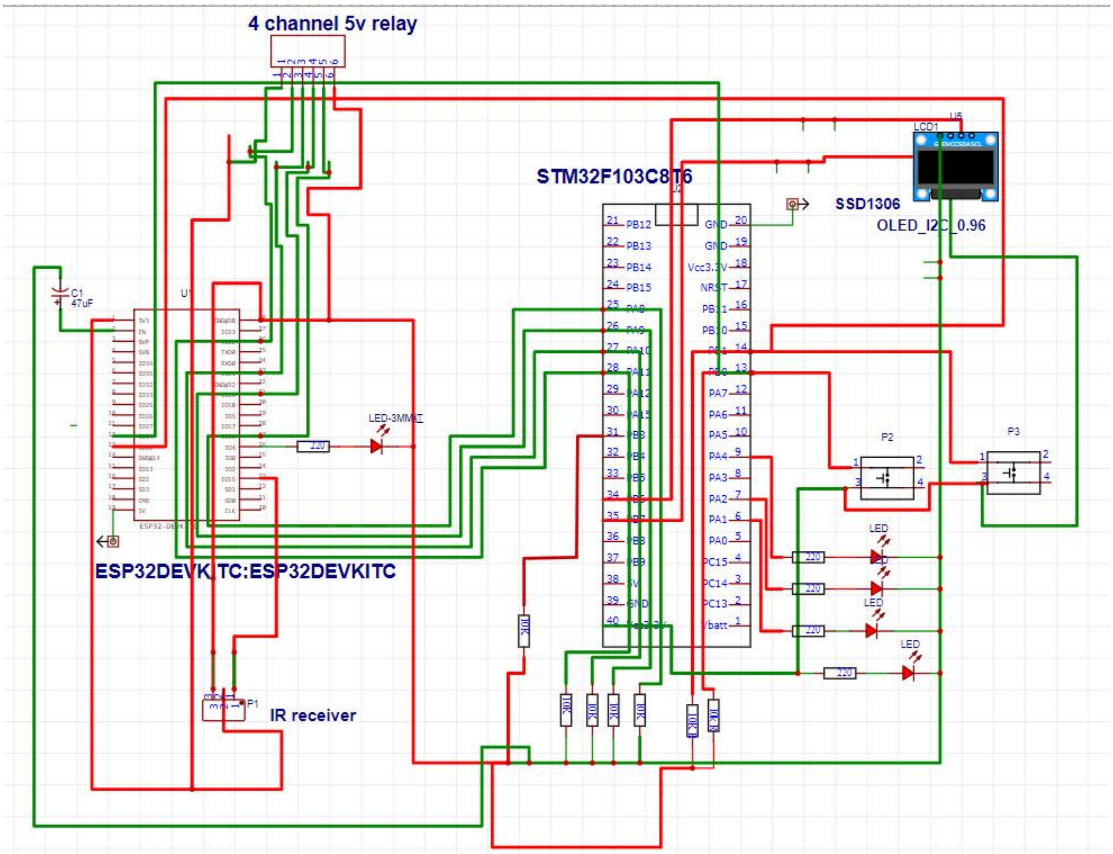

Figure2.3.1: Circuit schematic of the project

After schematic, the the PCB design and soldering all the components into it are required. The picture of whole project is as given bellow. 4 DC LED light bulbs were used as load which was connected with the relay module directly. Also the whole system is powered by a power source which is a 5 volt 10000 mAh DC power bank. Direct power supply is required for both ESP32 and STM32 and the power bank is the power supply source for both. Following picture is real working view of the project.

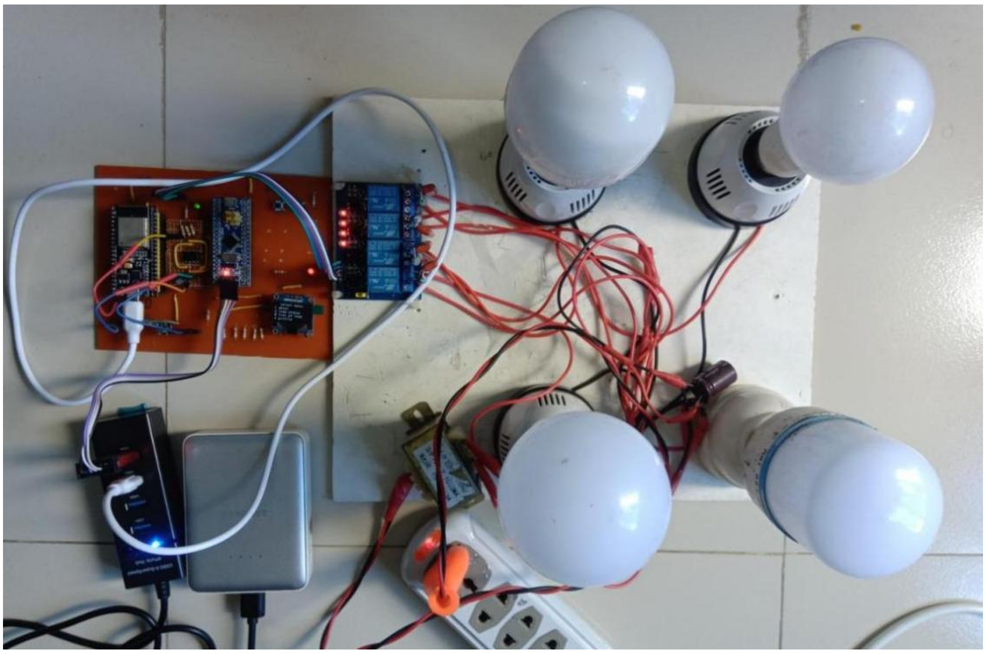

Figure 2.3.2: Complete working project

Here , in the PCB board , the left one is the ESP32 board , the right one is STM32 board and the middle one is the addition with the STM32 board which provides the pulse to count the time.All the parts will be discussed clearly as much as possible.

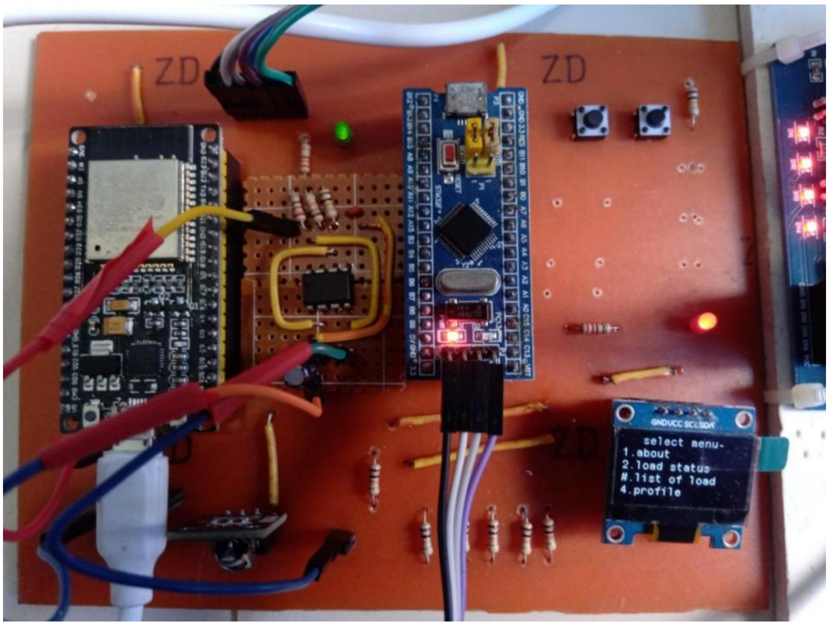

Figure 2.3.3: Main PCB board

However , as mentioned, before diving into project explanation, it can be divided as following:

- ❖ ESP32 part
- ❖ STM32 part
- ❖ Relay module and loads part

2.3.1: ESP32 Part:

The whole system responses based on input taken from this part more specifically, from the ESP32 board. Although , there are two methods of providing input to the ESP32 board, both of them do the same task and they do it in quite similar way. One is with IR remote which does not require any internet connection, another one is controlling over wi-fi.

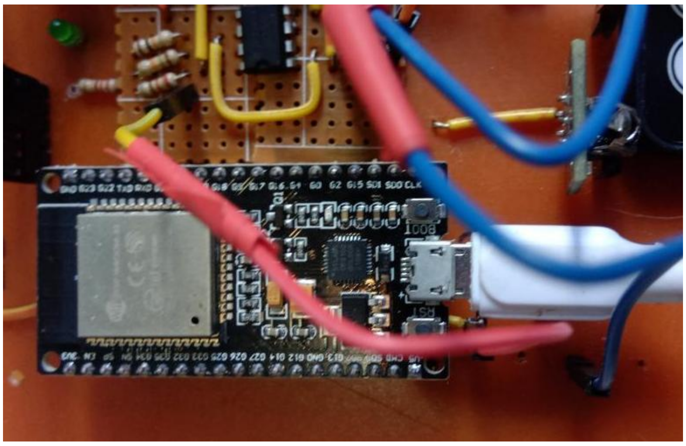

Figure 2.3.4: ESP32 and its part

In order to control the loads, first the board requires the connection of the wifi. The ESP32 board has been connected with a router at the time of setting up.

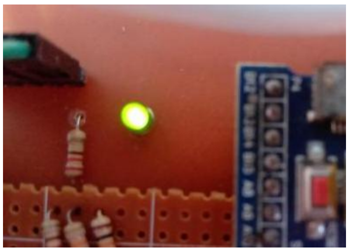

Figure 2.3.5:wi-fi LED indicator

When the board is connected to external wi-fi, the LED starts dimming, and before connecting to any network, the LED blinks continuously.
Also the IR remotes works properly after connecting the ESP32 board with wi-fi router.
The ESP32 board can be used using both web dashboard and mobile app. Both can be customized as required.

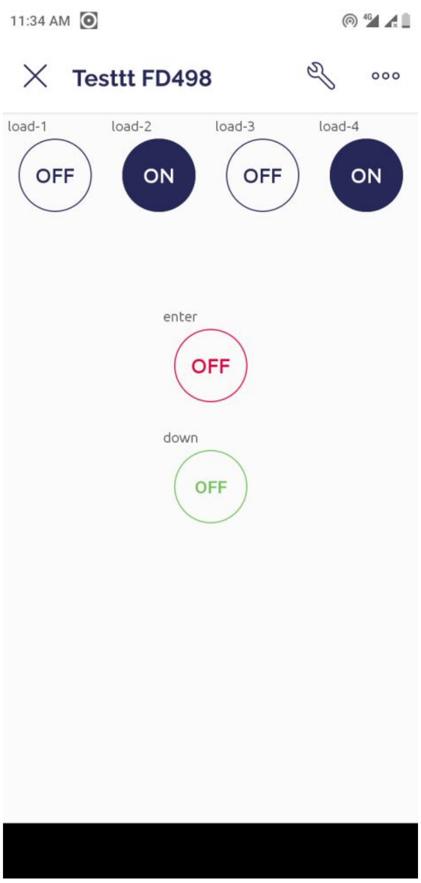

Figure 2.3.6: BLYNK app interface for this project

Another method is to use through IR remote. This remote sends some unique infrared signals every time the button is pressed. Based on that receiving signal, the board again sends signal to the relay and STM32 part.

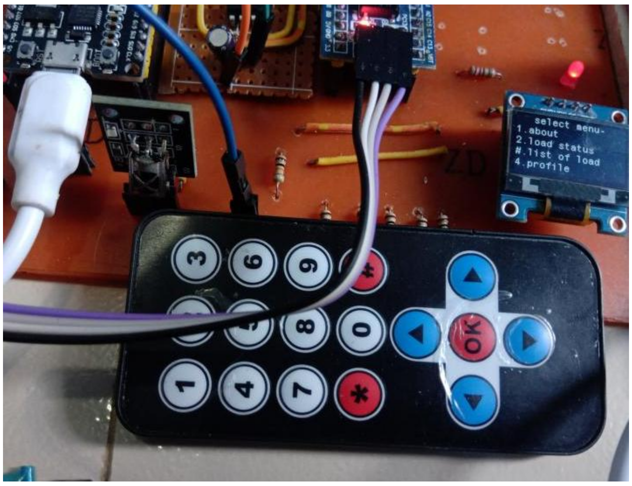

Figure 2.3.7: IR remote and receiver used in this project

The source code for programming ESP32 for this project can be found at-
https://github.com/mohammadshihab/ESP32-and-IR-remote

2.3.2: STM32 Part:

This part is responsible for receiving signal from ESP32 board and perform task by displaying according to loaded task. Here , there is one STM32 board one SSD1306 mono-crome OLED display and a astable multivibrator which will generate square wave pulse of 1Hz . This pulse mainly counts the on-time by taking pulse of 1Hz. The SSD1306 was connected with STM32 by I2C serial communication.

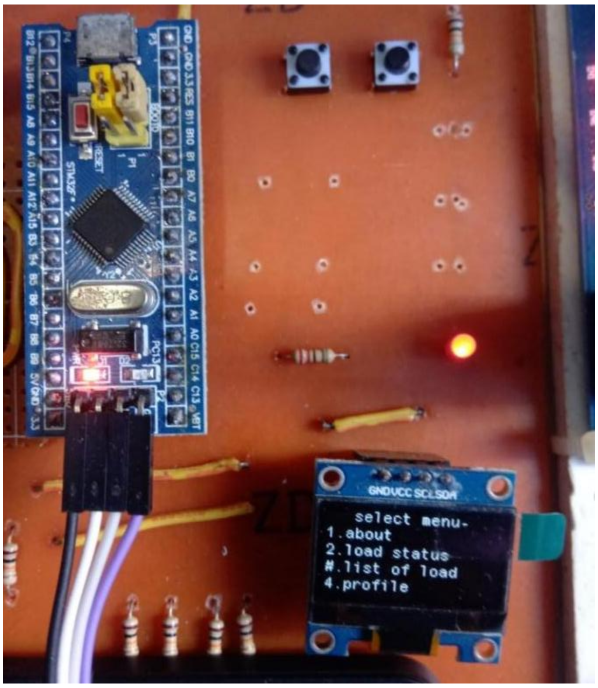

Figure 2.3.8:STM32 with OLED display

The asteable multivibrator used in this project is for providing pulse into the STM32 board . The picture of this circuit is given bellow.

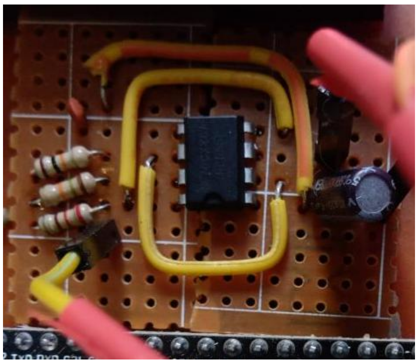

Figure 2.3.9: asteable multivibrator

Basically, the STM32 controls the display based on the output from the ESP32 board. There are two types of change in the display , one is because of ON/OFF of the loads, and the other one is the enter and cursor button. The picture given bellow is the homepage of the display. All the menu can be selected by pressing down button via IR remote or BLYNK over wi-fi

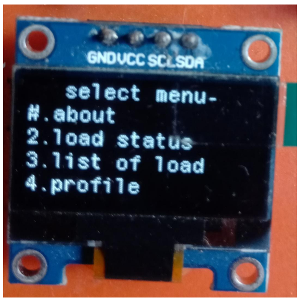

Figure 2.3.10: Main page of the display

Here, the first menu shows the basic about the project , which are some raw text. The 2nd and 3rd menu in the display mainly show the real time status of the system.

load status: This is the sub-menu page corresponding to the 2nd menu from the home page. As given to the picture. It is showing the list of the loads primarily.

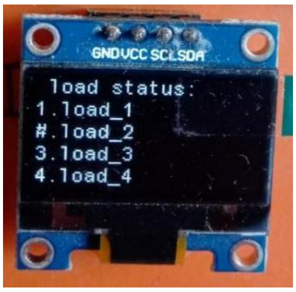

Figure 2.3.11: Load status page

Now , from the list , any load can be selected by using the cursor downing button. It will take to a specific load . By pressing enter in a selected load from the list, it will take to another sub page under the specific load from list of load. Now this page will show two parameters. One is ON or OFF parameter and the other one is the on-time status. If the load is ON , the display will show it as ON and the on-time will be shown beside of on-time. At the bottom part, the system on-time will be displayed.

Figure 2.3.12: Showing 2nd load ON with it's ontime as well as system ontime

Now , if the load is off, the on-time will be shown as 0 , but the system on-time will be counted as usual.

Figure 2.3.13: Showing 2nd load OFF

List of load: This is another sub-menu under 3rd menu of main page. After selecting this menu, it shows real-time ON/OFF status of the load. It shows that ON or OFF just beside of that load. Following picture shows load4 , load2 ON, others are OFF

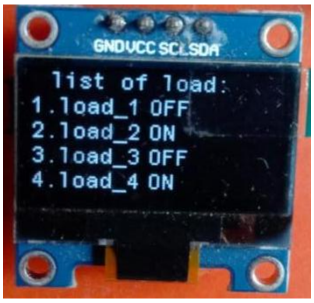

Figure 2.3.14: Showing 2nd and 4th load as ON

Following picture shows load1 and load3 as ON and others are OFF.

Figure 2.3.15: Showing 1st and 3rd load ON

The source code for programming STM32 for this project can be found at-
https://github.com/mohammadshihab/stm32-and-oled-display

2.3.3: Relay Module and Loads Part:

There are 4 channels in the relay, which can control up-to 4 loads at the same time. If any channel gets 5 volt supply , it will switches ON for that load, otherwise it turns OFF the load.

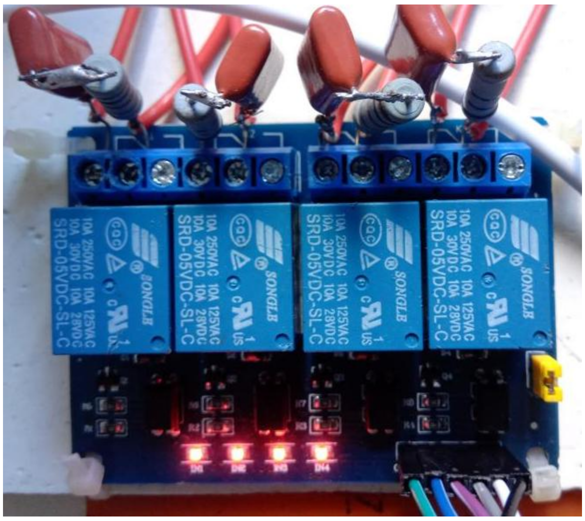

Figure 2.3.16: 4 channel relay with connected loads

4 snubber circuit were required in order to avoid EMI . All of those were connected in parallel with the electrical switches.

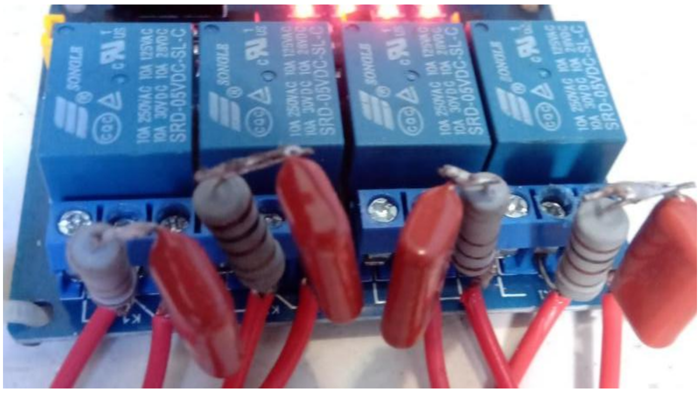

Figure 2.3.17: Snubber circuit

2.4: Printed Circuit Board:

Printed circuit board is one of the most important element in the project. To design the PCB, the EasyEDA , an online PCB design software has been used. To reduce fabrication complexity, the PCB was designed as single layer PCB. As there is a large number of connection, it was not possible to connect all the route directly, because they overlap with each other . That's why they were connected in a alternative way, which is known as jumper. The jumper connects the route using an external wired connection. Almost 7 or 8 jumpers were used in the PCB.

2.4.1: Main PCB

The main PCB contains all the routes only. As it mentioned earlier, the PCB contains single layer only . That single layer was the bottom layer only. All the components were at the top layer.

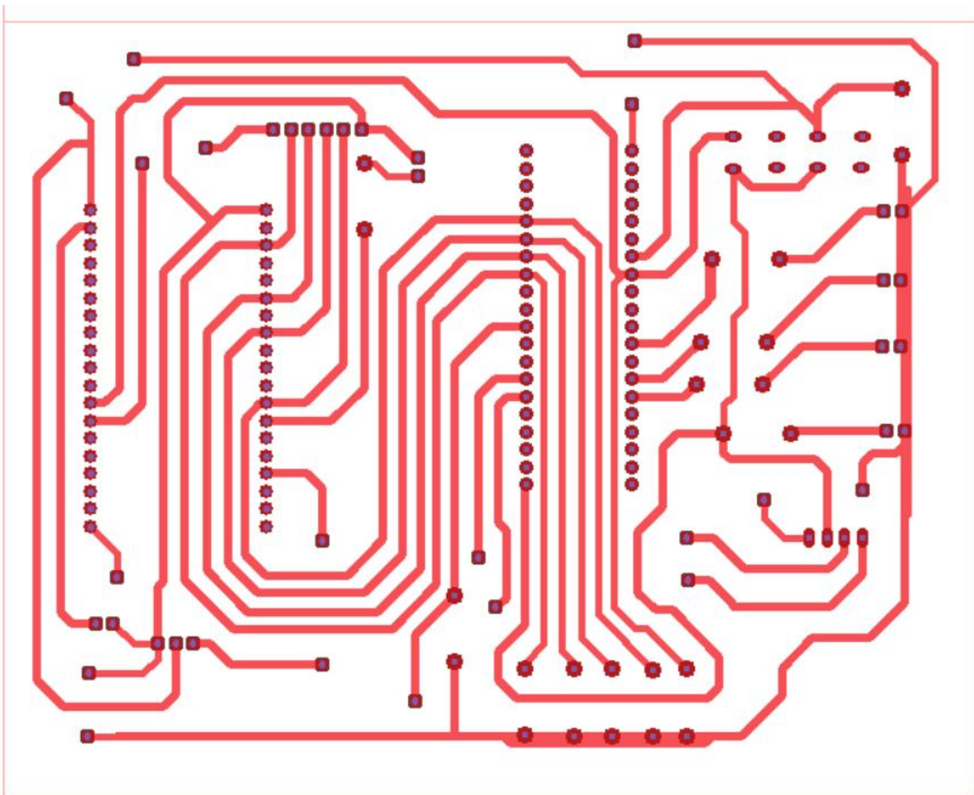

Figure 2.4.1: main PCB

After fabrication, all the routes were at the bottom side of the PCB board and all the components were at the top side. The 3D view of the PCB is given bellow:

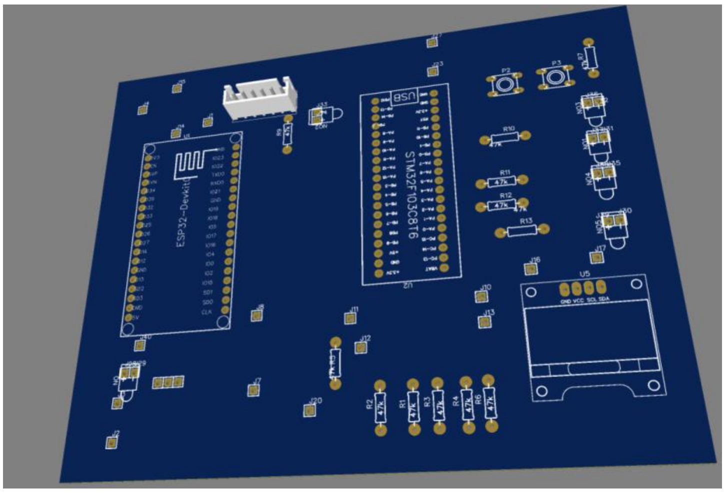

Figure2.4.2: 3D view of the main board

As mentioned earlier, an additional circuit with main PCB was in this circuit, this is the astable multi-vibrator , which was used to provide 1 Hz pulse to the STM32 board. The PCB of that circuit is given bellow:

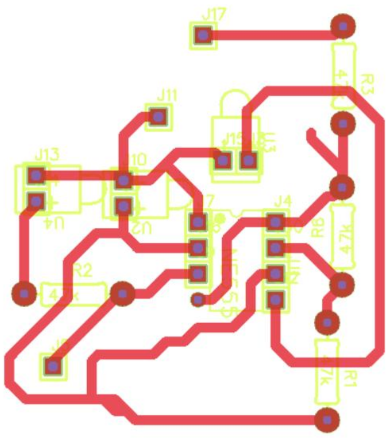

Figure2.4.3: PCB for aestable multi-vibrator

2.4.2: Details about PCB:

Using EasyEDA, it was possible to customize the design as long as required. The track width was taken as 1.1mm for most of the cases, somewhere it was taken as 1mm due to lake of space and even after this, it works properly.

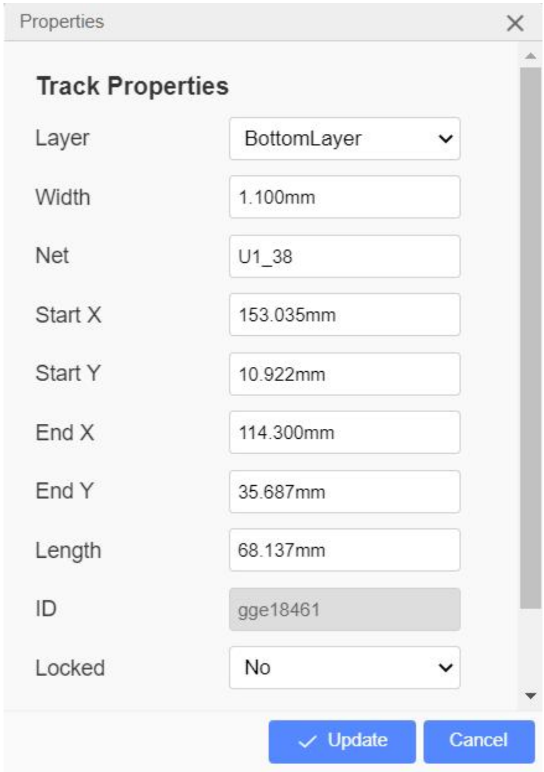

Figure 2.4.4: Track properties

2.4.3: PCB View Based on Different Category:

There are several types of layer and screen while designing a PCB such as top layer, bottom layer, drill, top silk screen , bottom silk screen etc. All the files with individual layer can be found on the gerber file after the PCB design is finished. Several view of the PCB is given bellow:

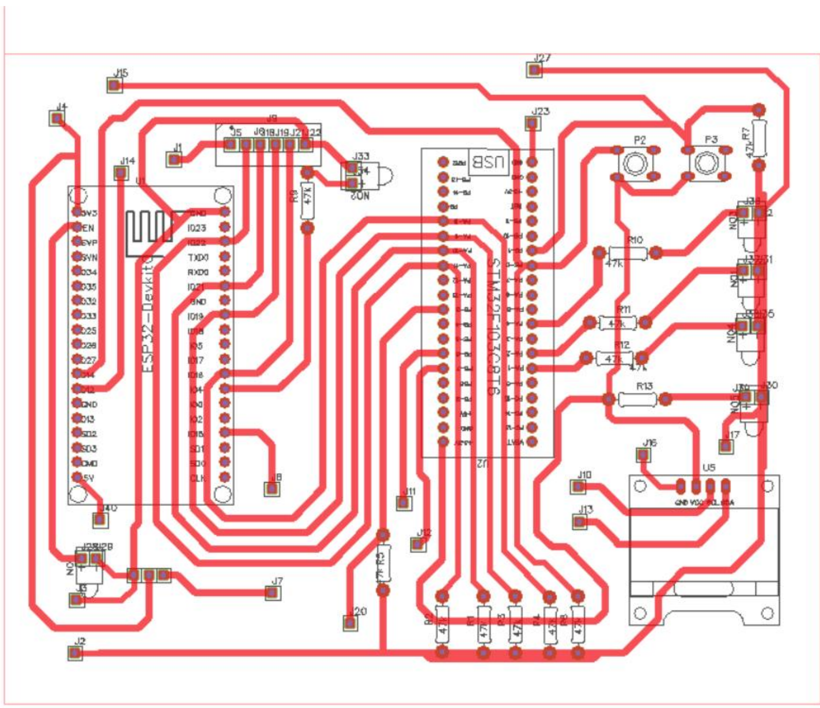

Figure 2.4.5: PCB with silk screen

Following figure provides the drilling position as well as silk screen,

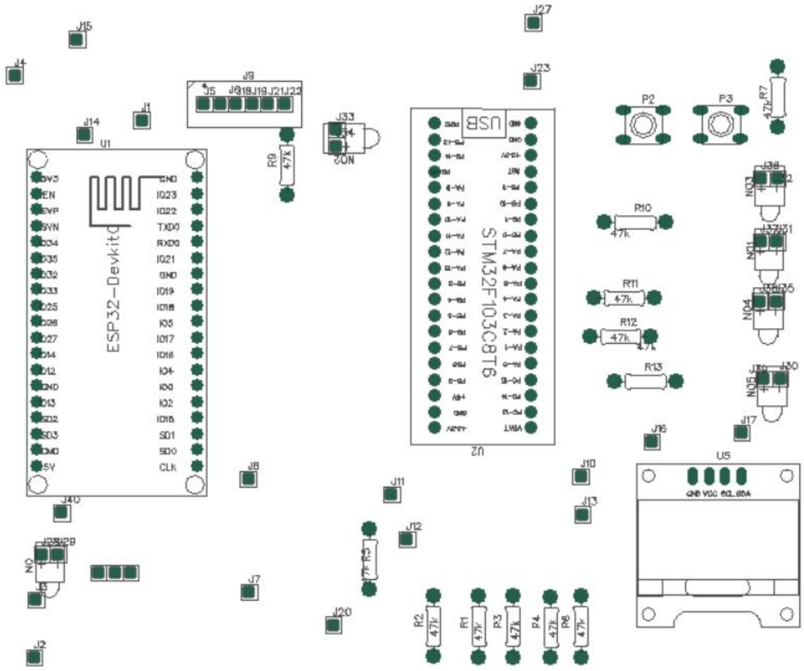

Figure 2.4.6 : PCB with only silk-screen and drill

Chapter-3: Conclusion

3.1: Conclusion of the Project

All the details about the project was mentioned in the report in details as much as possible. To explain the project more easily, it was divided into several parts. Details about all the components used in this project were explained in brief. There were several number of challenges while completing any part of the project.
To design the displaying part with multiple pages of the display was quite complicated, also selecting any menu from any page . To accomplish this, the interrupt routine for 2 pins were used. Also the incoming pulse from the multi-vibrator was causing interrupt continuously. The setup for I2C communication with OLED display was required to drive the display.
To control all the loads as well as the whole system , the ESP32 board was responsible for this. It was programmed based on the BLYNK IoT platform ,which is not that stable. Connecting the ESP32 board with a new router or network was bit struggling . Sometimes, designing the whole web dashboard was required for this .
Again , while controlling loads using 4 channel relay module, the display was blinking unexpectedly. It was happened due to inductive characteristics of the connected load. The loads were connected through wire and they show the inductive property which caused error in display. To avoid this problem , four snubber circuits were used and it prevented the unexpected EMI in the system.

3.2: Future Updates:

As the range of using of IoT and embedded system is increasing day by day. There is a lots of scope of applications that are being invented day by day. That's why there are several number of updates can be done in the project. For example , the monitoring part of the system was bounded in OLED display. This can be transformed in mobile app and web dashboard too. Where there might be a system to monitor the real time status of the whole system over wi-fi. The PCB board of the project quite large, several numbers of customization are possible in the PCB part. Also there were a several number of jumper connections in the project, which can be minimized. At the coding section of STM32, the size of the code for the board is quite large. Here, the code for the STM32 using CubeIDE can be optimized.