A

Project Report

on

Simulation of Home Automation Appliances using Android App & Google Assistant

By

Abhay Pratap Srivastava (1784110002) (TL)

Arun Kumar (1784110017)

Bhanu Pratap Yadav (1784110023)

Rahul Gupta (1784110037)

Under the Supervision of

Dr. Anurag Sewak

(Assistant Professor, CSED)

Submitted to the department of Computer Science and Engineering
For the partial fulfillment of the requirements
for award of Bachelor of Technology

in

Computer Science and Engineering

Rajkiya Engineering College Sonbhadra
Dr. A.P.J. Abdul Kalam Technical University, Lucknow,
Uttar Pradesh, India
July, 2020-21

Certificate

This is to certify that the Project report entitled **"SIMULATION OF HOME AUTOMATION APPLIANCES USING ANDROID APP & GOOGLE ASSISTANT"** is a record of the work done by the following students:

Student Name	**Roll No.**
Abhay Pratap Srivastava	1784110002
Arun Kumar	1784110017
Bhanu Pratap Yadav	1784110023
Rahul Gupta	1784110037

This work is done under our supervision and guidance during the academic year of **2020-21**. This report is submitted to the **Rajkiya Engineering College, Sonbhadra** for partial fulfillment for the degree of **B.Tech. (Computer Science and Engineering) of Dr. A.P.J. Abdul Kalam Technical University, Lucknow, Uttar Pradesh, India.**

We wish him all the best for all the endeavors

Signature Guide:
Dr. Anurag Sewak
(Assistant Professor, CSED)

ACKNOWLEDGEMENT

We would like to place on record our deep sense of gratitude to **Dr. Anurag Sewak, Assistant Professor, Department of Computer Science and Engineering, Rajkiya Engineering College, Sonbhadra** for his generous guidance, help and useful suggestions.

We express our sincere gratitude to **Prof. Amod Tiwari, HOD CSE, Rajkiya Engineering College, Sonbhadra** for his stimulating guidance, continuous encouragement and supervision throughout the course of present work.

Date: Student Name:

Abhay Pratap Srivastava
Arun Kumar
Bhanu Pratap Yadav
Rahul Gupta

ABSTRACT

Home Automation is need of hour for reducing energy wastage, to manage cost and moreover for the convenience. Even People want home automation for comfort and peace of mind.

This project is based on the construction of a soft model simulating a home appliances automation which can be controlled also by an android mobile application and Google Assistant remotely. To achieve this objective i.e., simulation, we have used a software- Proteus 8 Professional and platforms like Adafruit, Blynk and IFTTT. For the regulation and control it has a lamp, some LED, and fan.

The core is an Arduino UNO R3 board, Node MCU 8266 (For Google Assistant) that allows the application operation and receives, from a Blynk Android mobile application and Google Assistant, operating modes commands. For the data transmission from the mobile to the board, is used communication via Internet.

This project is especially for handicapped people in house and they are not able to move very frequently for controlling appliances in house, so using home appliance automation system these people can easily control all the appliances. For them (handicapped people) it is essential to develop such home appliance automation system which required less and easy user interaction.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

The main points of this project are simulation, automation and control, electronic boards and mobile applications and particularly home automation, Arduino boards and Android applications.

This is the result of the connection of two modern technologies, including hardware and software. Despite the actuality of these components, they all appeared long ago and have been evolving throughout history.

1.1 Motivation & Origin

History of automation

This project is based in automation technology and more specifically in home automation systems.

Automation is the transfer of tasks normally performed by humans to a set of technological elements.

An automated system consists of two parts:

Operation: Part formed by elements that act directly on the machine and make it perform desired operations. These elements are called actuators and some examples are engines, cylinders or photodiodes.

Control: Brain of system, normally constituted by a programmable automaton, able to communicate with all constituents of the operation part.

The inclusion of control in the automation system, allows to decide on the development of a process, manipulating certain variables to get these or other variables to act in the desired way.

Although it seems a recent technology and currently is in full development, automation dates back to ancient times.

Nowadays nobody thinks of divine forces seeing an automatic door, it is something of the day to day that we have very internalized. It was not the same with people of

Alexandria when Herón, the great genius of mechanical engineering of classical Greece, invented an automatic door opening system for a temple in Alexandria.

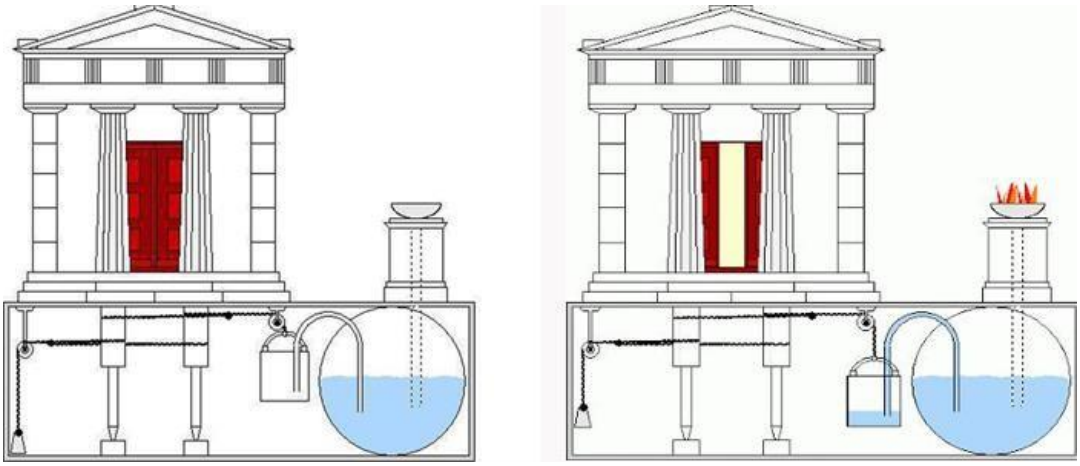

Figure 1.1. Automatic door designed by Herón of Alexandria.

In spite of being antecedents of automation much earlier, like Osiris statue of ancient Egypt, crankshaft and mechanic watches of the engineer Al-Jazari (1260) or automats designed by Leonardo Da Vinci (1452-1519), the industrial revolution is considered as the greatest technological, socioeconomic and cultural change in history.

It is not completely clear what year the industrial revolution began, but it is speculated that it was in 1775, when James Watt invented the steam engine. Even so, around 50's, semiconductors began to be used, the fact that turned out to be a breakthrough because before this, only mechanical and electromagnetic elements were used.

In 1968 Ford and General Motors raised the specifications that had to have a programmable electronic controller to be useful in the industry, the fact that arises from the need of programming to optimize processes. In addition, the company Bedford Associates developed a prototype of industrial controller obtaining what is considered as the first PLC (Programmable Logic Controller) in history, which was named Modular Digital Controller or MODICON, used by a car manufacturer in the United States.

Figure 1.2. MODICON. First PLC of the history.

Later, in the early 70's, the Intel company developed the first microprocessor, a 4-bit CPU named i4004, for the Japanese company Busicom to incorporate it into calculators and by the middle of that decade it was already included into automatons. It contributed to give more flexibility because of the ease of programming, as the wired memories disappeared.

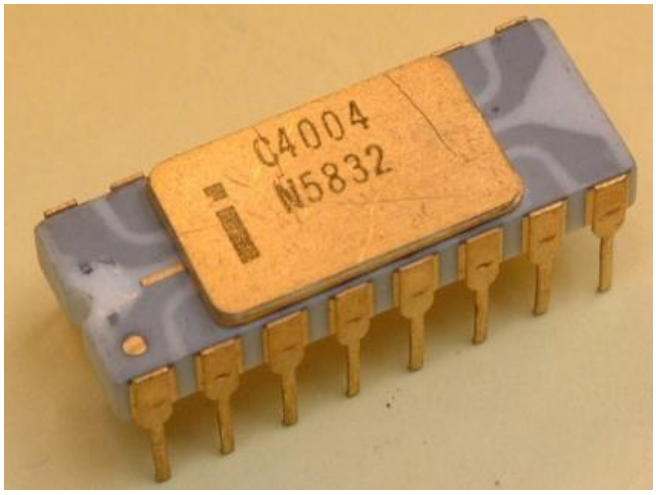

Figure 1.3. i4004. First microprocessor of the history.

Over years the improvements were continuous, the automaton was gaining memory, ability to govern control loops, communications and programming languages became more powerful, obtaining faster processing speed, more complex control techniques… Until nowadays, having the great number of existing automatons, increasingly

powerful and useful in different fields, even disengaging from the industry to open new roads, such as home automation applications.

1.2 Existing Approaches

Currently thanks to the permanent technological evolution the offer is better and have grater quality, its use is now more intuitive and perfectly manageable by any user.
Today's home automation contributes to increase the quality of life, makes the distribution of the house more versatile, changes environmental conditions by creating different predefined scenes and makes the home more functional by allowing the development of domestic, professional and leisure under one single roof.

1.3 Objectives

- The main objective of this project is to design and simulate a prototype of a home automation devices controllable from an Android mobile application and Google Assistant.
- Application must be able to perceive and act and to have controlling operation in order to obtain the purpose for which this technology was invented: maximizing user's comfort offering an easy way to personalize home.

The steps that should be taken to achieve the expected result are the following:

1. Determine the scope of the application and delimit the points that each mode of operation must deal with.
2. Select the components and software.
3. Electronic design.
4. Program the board.
5. Integration of mobile application.
6. Build the applet and Integration of Google Assistant.
7. Test and debug the application.

1.4 System Overview

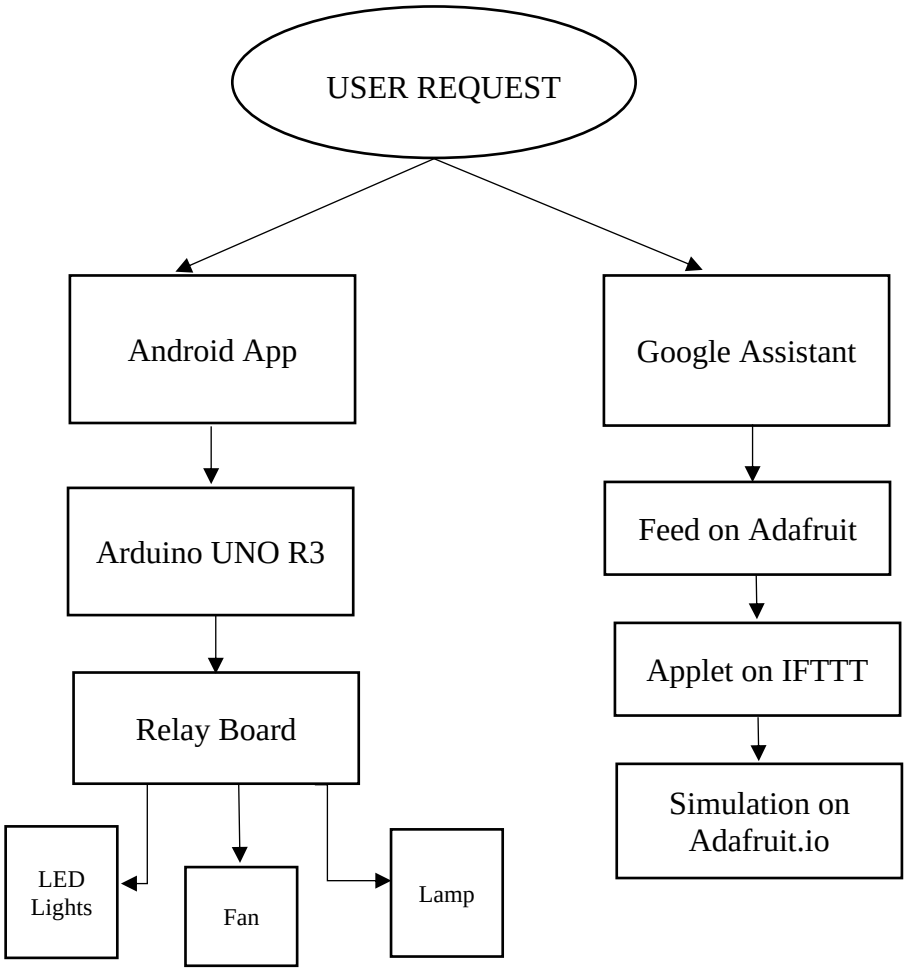

Fig 1.4 System Overview

The system is divided into two modes as the request from user through two communication channel-

- Through Android Mobile App
- Through Google Assistant

First through Android App where User sends request to ON/OFF the electric appliances through Mobile App. This request first received by the Arduino UNO R3 which instruct relay module to pass the current or stop the flow of current in the appliance/device.

On the other communication channel, User request Google assistant through phone and give instruction to either ON/OFF appliance. As soon as request arrives to IFTTT it instructs the Adafruit.io and requested action performed.

CHAPTER 2

LITERATURE REVIEW/ SURVEY

In Current Scenario, India's most of the homes are far away from being called as Smart Home. This is supposed that in India smart home market will expand by 2022 and we have start seeing the expansion as soon as Make in India IoT things started to come in our daily life.

In Existing system, there is normal electric appliances working on physical interaction that leads to lot and lot of wastage of energy in daily life, not only the energy but due to this wastage and high consumption, cost factor comes into the play which is impacting the lives of Indian People.

Also, in current scenario where handicapped who are at home and do not move very frequently. They sometimes feel ashamed of themselves as they are not able to the basic tasks their own.

2.1 Scope Of the Project

The purpose of this project is to develop and simulate using software and online platform, working of home automation appliances such as LED Bulbs, Fan, and Lamps which can be controlled through Google Assistant and Android Mobile App remotely from anywhere in the world using internet connectivity in user's mobile phone. Even check the status of the appliance.

2.2 Proposed System

In proposed system the user can easily turn the switch off or on the different home appliances either staying in home without physically navigating to the switch or outside the home. They have the connectivity to the board using internet. User can even check the status of the appliance which is either running or inactive state.

Having Home automation specifically Control of appliances through Smart Phone or even Google Assistant, it'll make task very easy. It'll help in saving energy, cost factor will be in budget, provide convenience and for handicapped or differently abled person

they’ll find ease in controlling these appliances. They do not need to move having smartphone.

2.3 Problem Definition

- To develop, design and simulate a system which can control different home appliances using Android Mobile App or Google Assistant over Internet from anywhere in the world.

CHAPTER 3

METHODOLOGY

3.1 Proteus 8 Professional Simulator

The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. The software is used to create schematics and electronic prints for manufacturing printed circuit boards. Proteus 8 is one of the tools needed for circuit design and simulation. This software is developed by Labcenter Electronics.

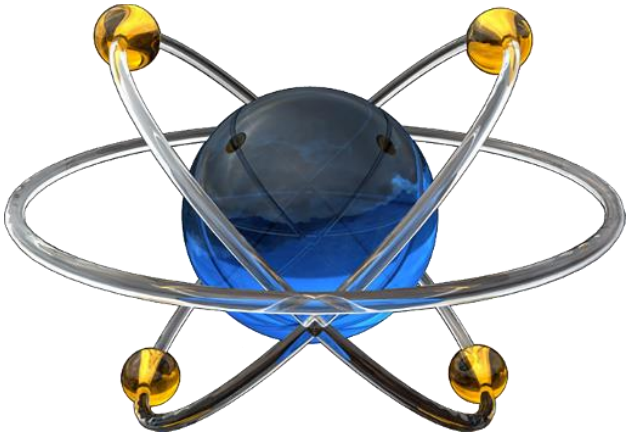

Fig. 3.1 Proteus 8 Profession Software Icon

Proteus virtual system modelling (VSM)

The most exciting and important feature of Proteus VSM is its ability to simulate the interaction between software running on a microcontroller and any analogue or digital electronics connected to it. The micro-controller model sits on the schematic along with the other elements of your product design. It simulates the execution of your object code (machine code), just like a real chip. If the program code writes to a port, the logic levels in circuit change accordingly, and if the circuit changes the state of the processor's pins, this will be seen by your program code, just as in real life.

The VSM CPU models fully simulate I/O ports, interrupts, timers, USARTs and all other peripherals present on each supported processor. It is anything but a simple software simulator since the interaction of all these peripherals with the external circuit

is fully modelled down to waveform level and the entire system is therefore simulated. With over 750 supported micro-processor variants, many thousands of embedded SPICE models and one the world's largest libraries of embedded simulation peripherals, Proteus VSM remains the first choice for embedded simulation.

Proteus VSM combines circuit simulation, animated components and microprocessor models to facilitate co-simulation of complete microcontroller-based designs. Proteus VSM makes it possible to develop and test designs before a physical prototype is constructed. The functions and advantages of Proteus VSM are introduced.

3.2 Arduino IDE

The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of third-party cores, other vendor development boards.

Fig. 3.2 Arduino IDE Icon

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program avrdude *(AVRDUDE - AVR Downloader Uploader - is a program for downloading and uploading the on-chip memories of*

Atmel's AVR microcontrollers.) to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware. By default, avrdude is used as the uploading tool to flash the user code onto official Arduino boards.

3.3 Blynk Android App

Blynk is a Platform which provides a platform for building mobile applications that can connect electronic devices to the Internet and remotely monitor and control these devices.

Blynk was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data, vizualize it and do many other cool things.

There are three major components in the platform:

- **Blynk App -** allows to you create amazing interfaces for your projects using various widgets we provide.
- **Blynk Server -** responsible for all the communications between the smartphone and hardware. You can use our Blynk Cloud or run your private Blynk server locally. It's open-source, could easily handle thousands of devices and can even be launched on a Raspberry Pi.
- **Blynk Libraries -** for all the popular hardware platforms - enable communication with the server and process all the incoming and outcoming commands.

Fig. 3.3 Blynk App for IoT Icon

3.4 IFTTT (If This Then That)

IFTTT (If This Then That) is a service that allows a user to program a response to events in the world.

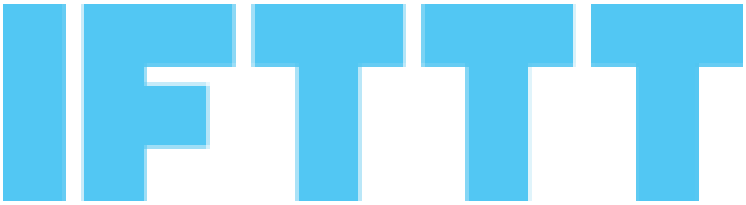

Fig. 3.4 IFTTT Icon

IFTTT has partnerships with different service providers that supply event notifications to IFTTT and execute commands that implement the responses. Some event and command interfaces are simply public APIs.

The programs, called applets, are simple and created graphically. An automation platform that communicates between different apps, web services, and devices to trigger user-specific actions through Applets.

User can create programs and otherwise control IFTTT with a web interface or iOS or Android application.

3.5 Adafruit.io

Adafruit.io is a cloud service - that just means we run it for user and user don't have to manage it. User can connect to it over the Internet. It's meant primarily for storing and then retrieving data but it can do a lot more than just that!

Fig. 3.5 Adafruit.io

Dashboards

Adafruit.io can handle and visualize multiple feeds of data Dashboards are a feature integrated into Adafruit IO which allow you to chart, graph, gauge, log, and display data. This data can view your dashboards from anywhere in the world.

Triggers

Triggers in Adafruit IO to control and react to data.

3.6 VSPE

The Free Virtual Serial Port Emulator or VSPE is a program that allows IT engineers and developers to create, test, and debug applications that use the parallel port.

It allows the creation of virtual devices to send and receive data. Unlike parallel ports, virtual devices can be used multiple times by multiple applications. VPSE allows you to share physical data from multiple applications, expose ports to local network, create virtual parallel ports, etc.

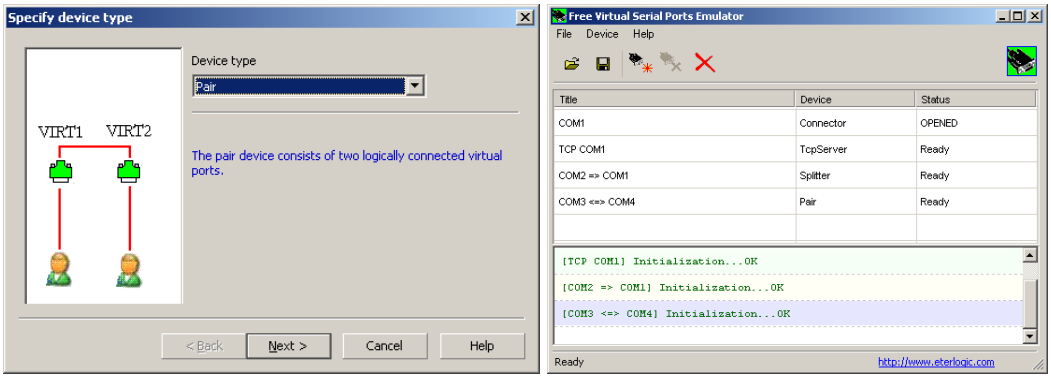

Fig. 3.6 VSPE Software Screenshots

CHAPTER 4

SYSTEM ANALYSIS AND DESIGN

4.1 Arduino UNO in Proteus

Proteus 8 professional doesn't include Arduino UNO R3 in their preloaded library so we need to download this library from other source and then move this into the destination folder of Proteus 8 Profession so that we can include it in the project.

These are steps followed to download and import in the Proteus Library-

Step 1- Downloaded the Compressed type of Arduino UNO R3 Library for Proteus.

Step 2- Extract files and copy two files namely-

ArduinoUNO2TEP.IDX, and *ArduinoUNO2TEP.LIB.*

Step 3- Navigate to the Drive C Program Files(x86) and *Labcenter Electronics* Folder where Proteus 8 Professional is already installed. In our case path of folder is –

C:\Program Files (x86)\Labcenter Electronics\Proteus 8 Professional\DATA\LIBRARY

Step 4- And then paste copied two files. Now we are set to use Arduino Uno R3 in Proteus 8 Professional.

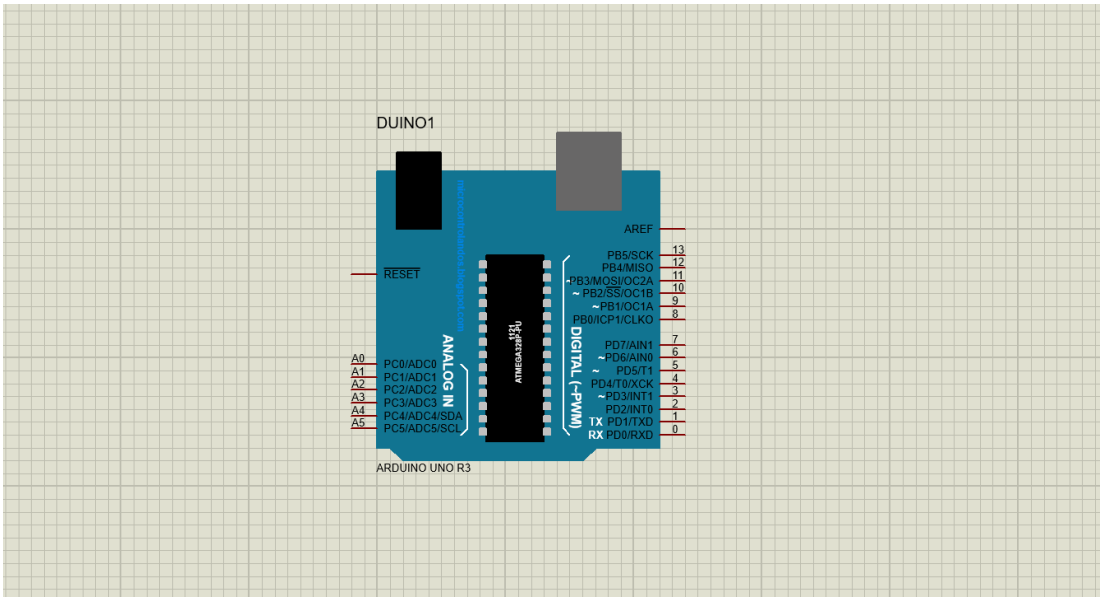

Fig. 4.1 Arduino UNO R3 in Proteus

4.2 Arduino IDE- Installation, Board, Port and Library Configuration

4.2.1 Arduino IDE Installation

To Simulate any IoT project we need to code the microcontrollers. Either we upload code to physical microcontrollers or we simulate the microcontroller using code written in Arduino IDE.

To simulate or run the IoT project we need to follow these steps to download and install Arduino IDE in native computer-

Step 1: First and foremost, we need to visit the Arduino official website- https://www.arduino.cc/en/software

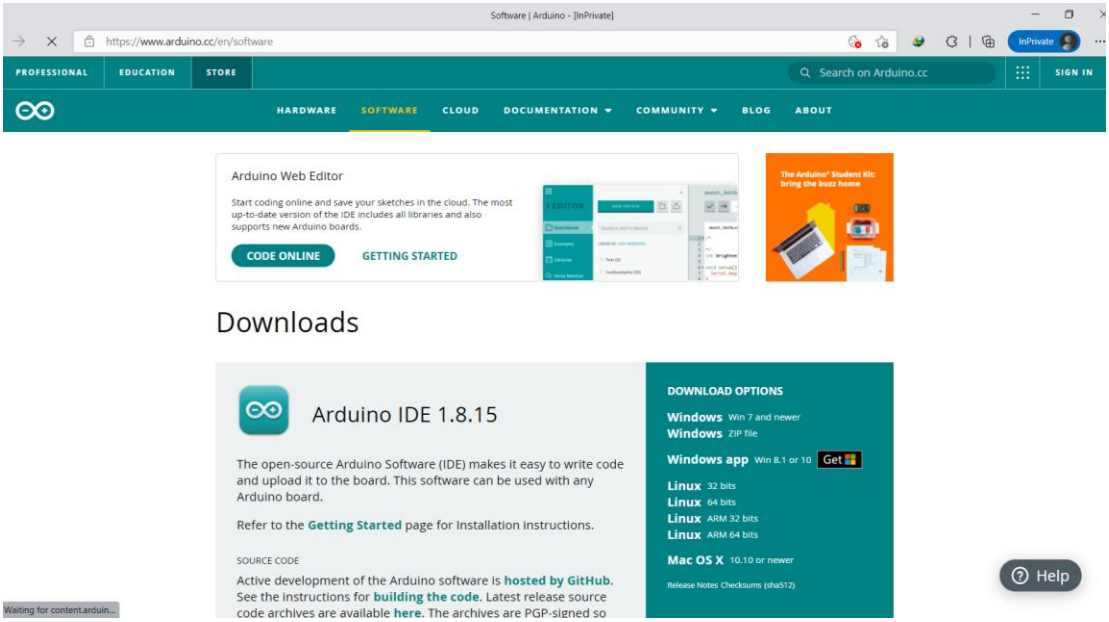

Fig. 4.2 Arduino IDE Official Website Download Page

Step 2: Following download Link for Windows 10 or higher, download Arduino IDE in computer and install in the system.

Step 3: After Successful Installation, Open Arduino IDE

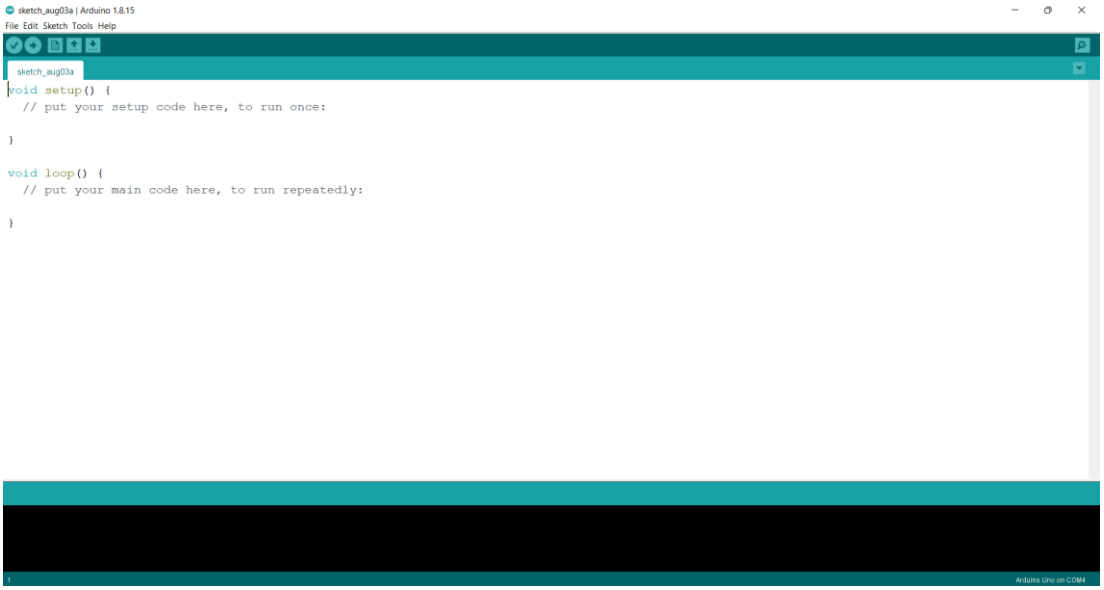

Fig. 4.3 Arduino IDE after successful Installation

4.2.2 Arduino IDE Board

In Arduino we select Board before coding, every code cannot run on every board (Board- Electronic Board Like- Arduino UNO, Node MCU) so we select one board on which we're going to write program to run/simulate our project.

In our project we are making home automation of appliances using Arduino UNO R3 so we'll select by going to top of section in Arduino IDE.

Tools >Board >Arduino AVR Boards > Arduino UNO

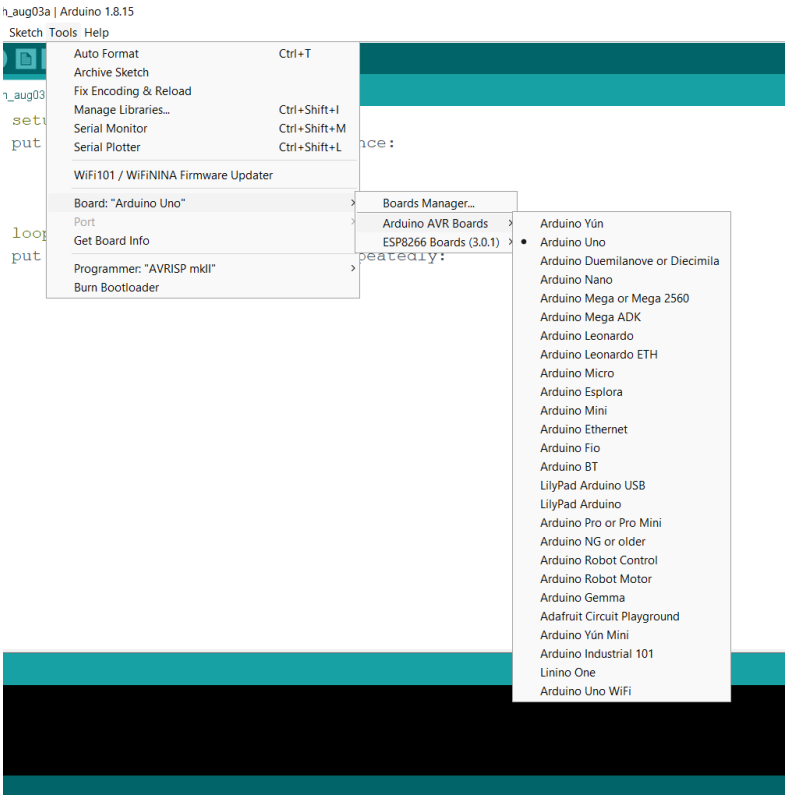

Fig. 4.4 Selecting Arduino UNO Board in Arduino IDE

4.2.3 Arduino IDE Port

This menu contains all the serial devices (real or virtual) on our machine. It automatically refreshes every time you open the top-level tools menu. This port basically comes into the play when we've plugged multiple hardware in our computer so by selecting this, we select on which port or hardware we have to upload the program.

*It is located in **Tools Section of Main Menu > Port***

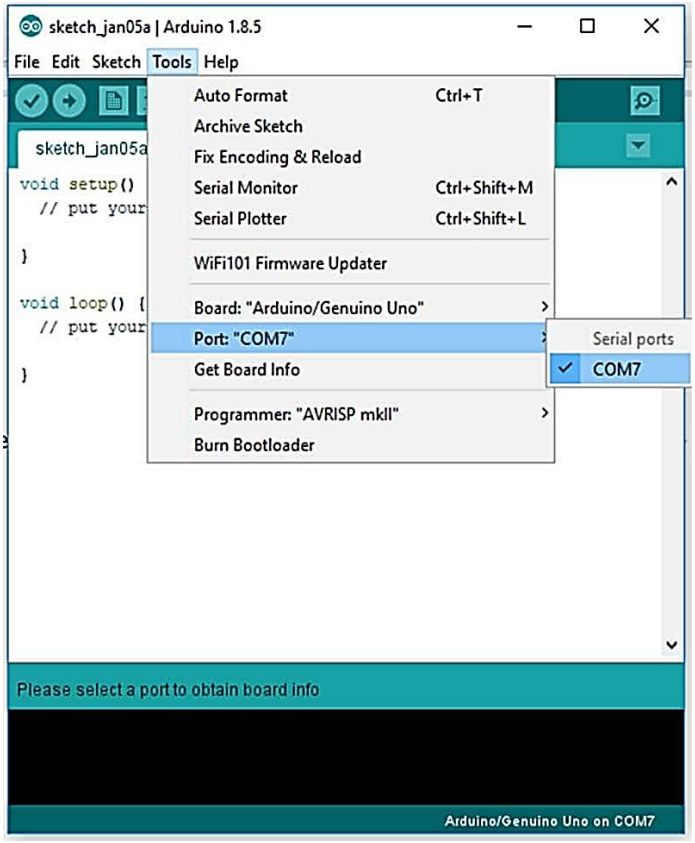

Fig. 4.5 Selecting Port in Arduino IDE

4.2.4 Arduino IDE Library Configuration

The Arduino environment can be extended through the use of libraries, just like most programming platforms. Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data.

To use a library in a sketch, select it from ***Sketch > Import Library.*** A number of libraries come installed with the IDE, but we can also download or create our own.

In our project we are managing with Blynk App for IoT so we need to Download the library in the IDE before writing #include.

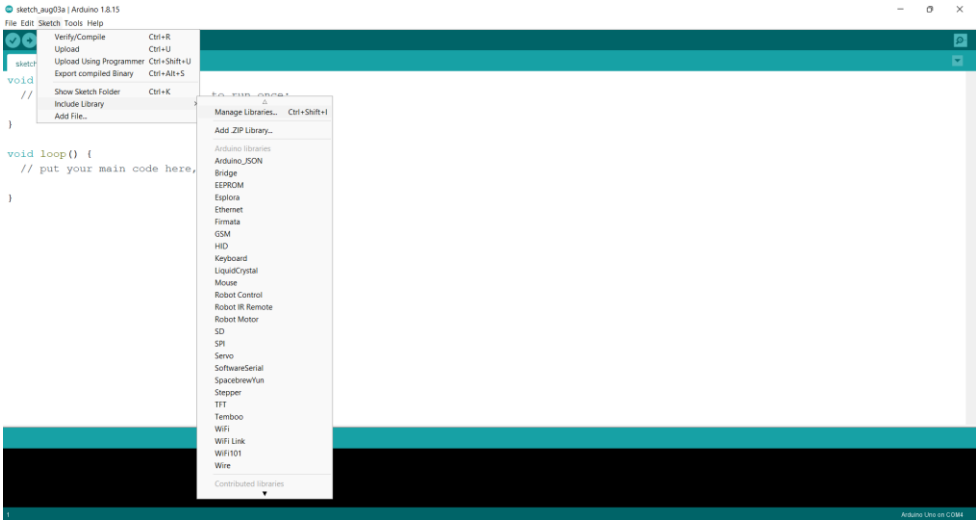

Fig. 4.6 Library Include in Arduino IDE for Program (Sketch > Include Library> Manage Libraries)

Fig. 4.7 Manage Libraries Window (Libraries can be installed or uninstalled)

4.2.5 Arduino IDE- Preference Setup

4.2.5.1 Preference- Selecting Compilation

By Default, Arduino IDE is setup with only upload preference. On verifying it will generate only program can be uploaded in hardware or not. But when we do simulation, we need to have compilation code (hex files) in Proteus. This can be only generated when we select compilation as well.

For that we have to follow these steps:

Step 1 – Go to Main Menu and *File > Preferences*

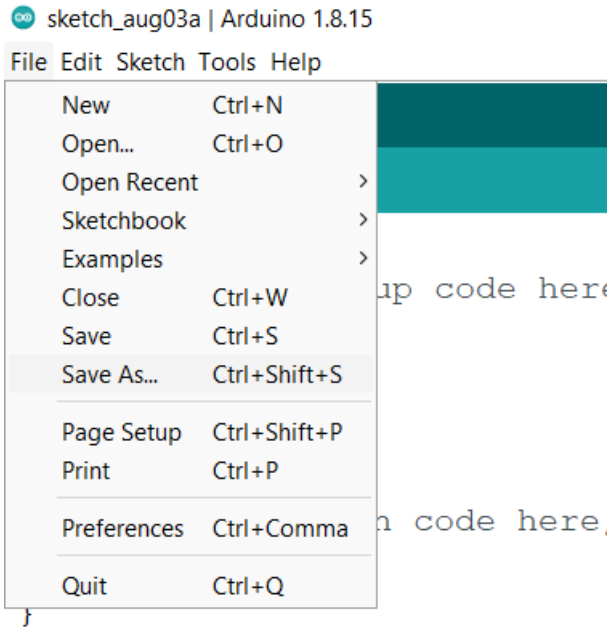

Fig. 4.8 Selecting Preferences in File Menu

Step 2 - Tick on Compilation and Upload Both or If only doing simulation we can go with compilation only as this will generate hex code.

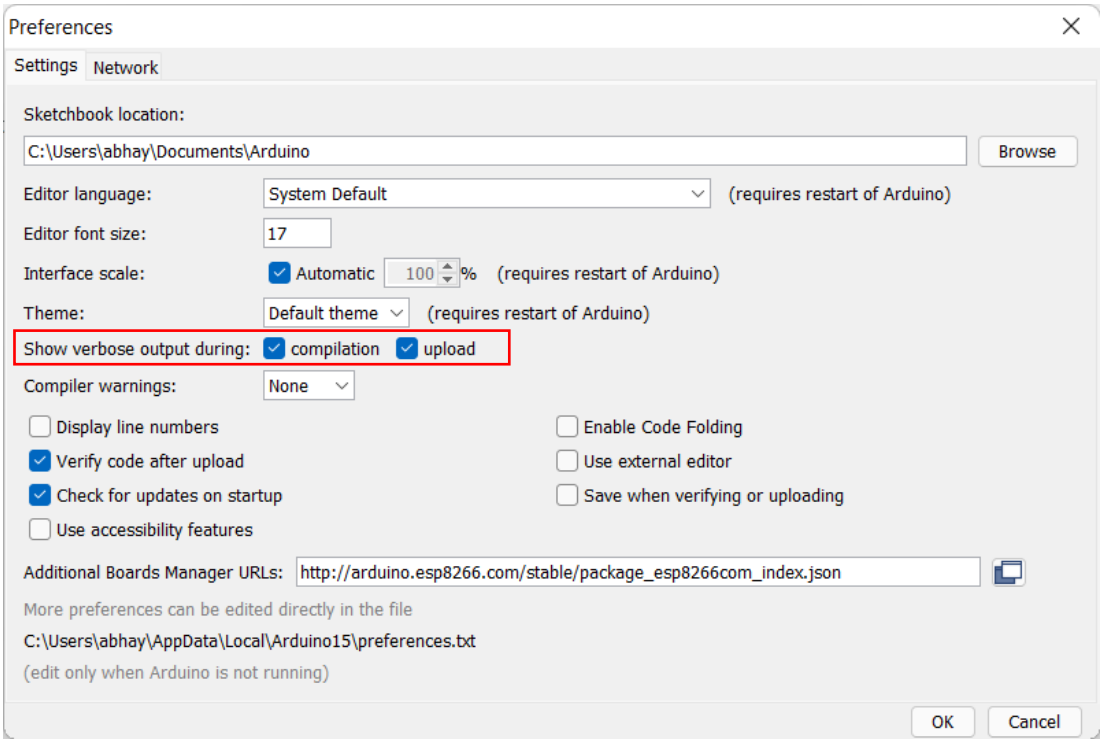

Fig. 4.9 Selecting Compilation in Preferences

4.2.5.2 Preference- Additional Board Manager URLs –

For 3rd party hardware packages, it is first necessary to add the URL of their Boards Manager JSON file in: *File > Preferences > Additional Boards Manager URLs.* The URLs point to JSON index files that Arduino IDE uses to build the list of available installed boards.

*URL-**http://arduino.esp8266.com/stable/package_esp8266com_index.json***

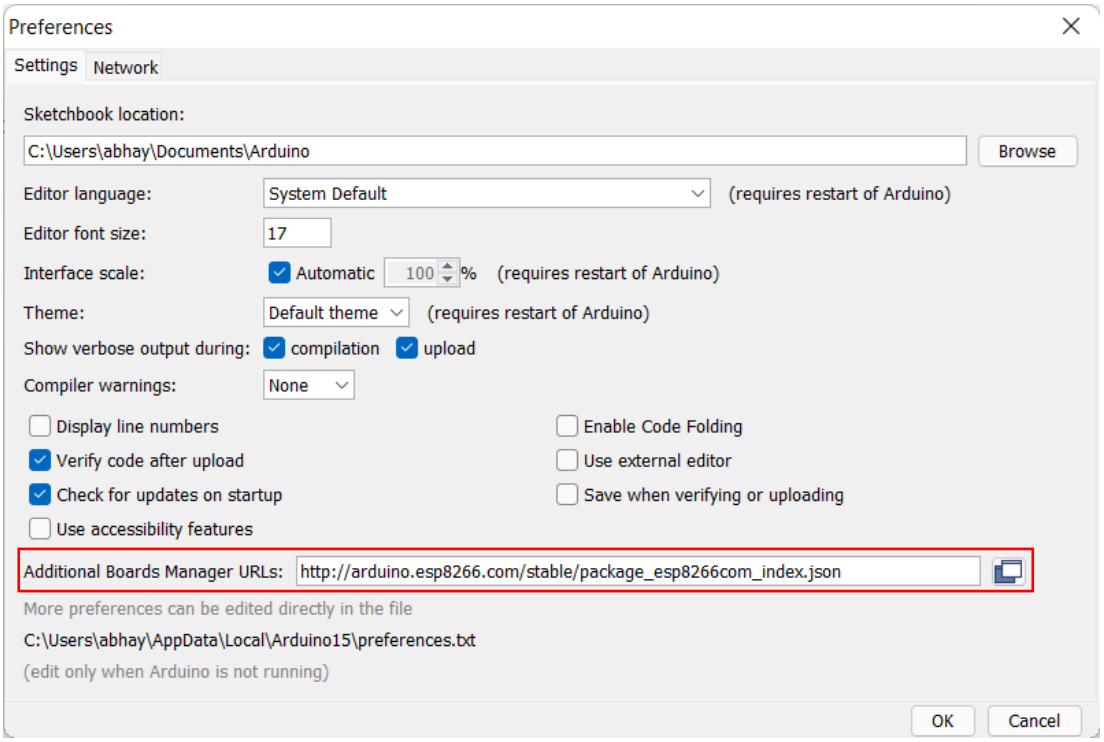

Fig. 4.10 Additional Board Manager URLs

4.3 Circuit Design in Proteus

4.3.1 Creating Project in Proteus 8

To create project in proteus, we have to open Proteus 8 profession and Navigate to New Project from option available on front or start page.

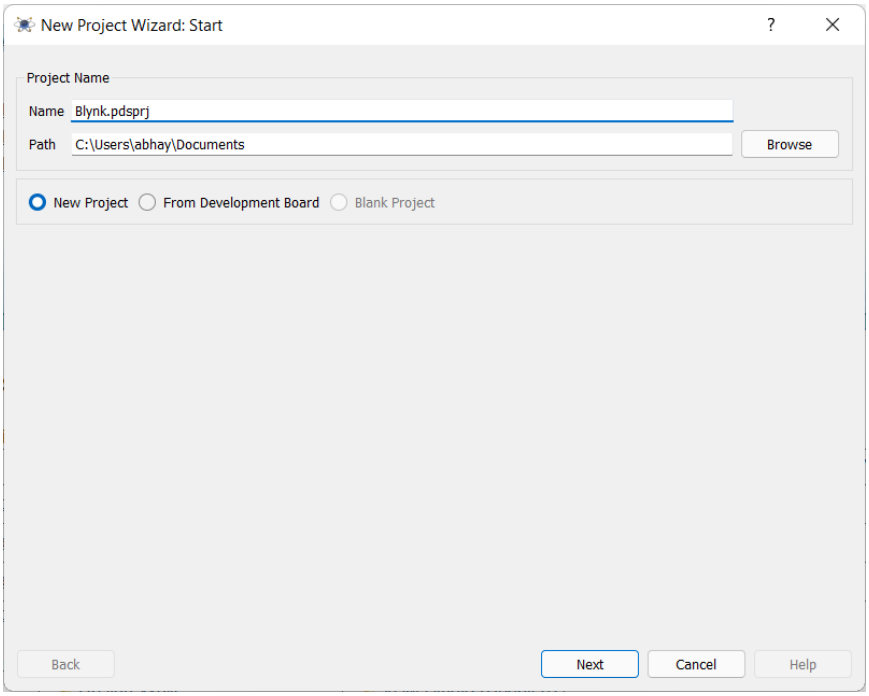

Fig 4.11 Creating New Project in Proteus

After creating project we'll on project main page where we'll drag and drop elements, devices and create circuits.

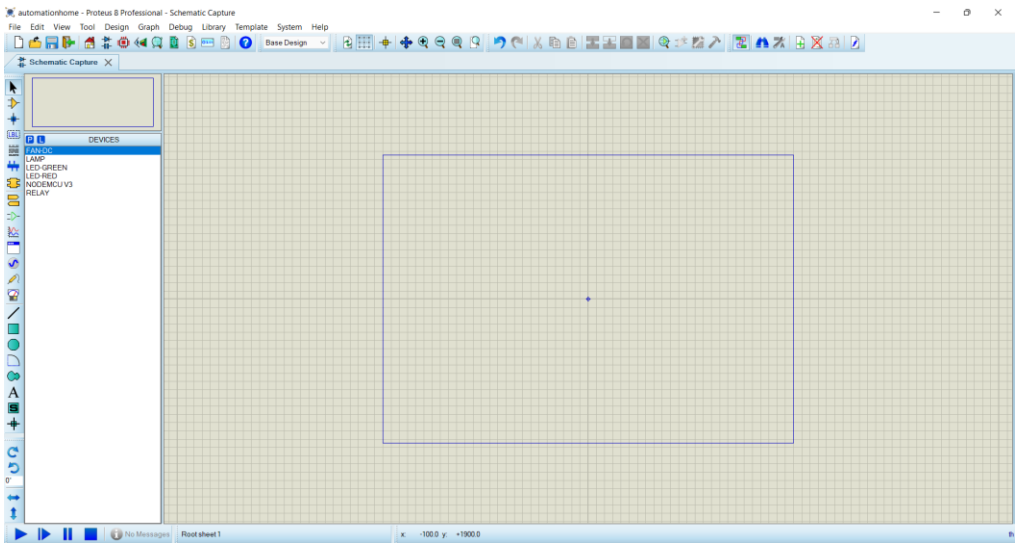

Fig 4.12 GUI Proteus 8

4.3.2 Design of Circuit using Components

To implement simulation of electric devices on Proteus using Android App we have first drawn a circuit diagram which we've to implement in proteus.

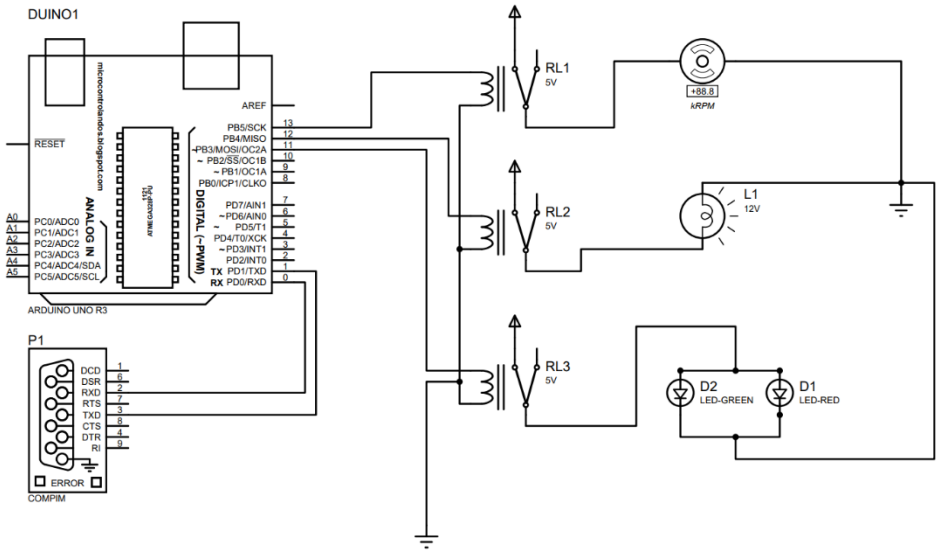

Fig. 4.13 Circuit Diagram

For Adding the devices and Terminals in the working area we search device in Pick Device Section and for terminals we pick from terminal section.

4.3.2.1 Device

There are multiple device devices used in the circuit –

i. Arduino UNO R3
ii. COMPIM
iii. FAN DC
iv. LAMP
v. LED (Green & Red)
vi. Relay

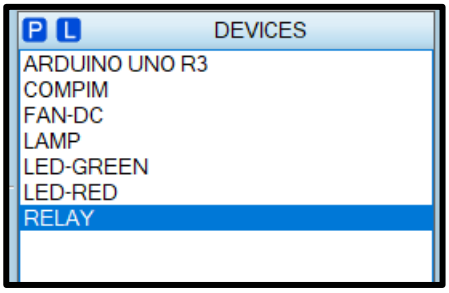

Fig. 4.14 Devices used in Proteus

i. Arduino UNO R3:

The Arduino Uno is a microcontroller board based on the ATmega328. It has 20 digital input/output pins (of which 6 can be used as PWM outputs and 6 can be used as analog inputs), a 16 MHz resonator, a USB connection, a power jack, an in-circuit system

programming (ICSP) header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features an ATmega16U2 programmed as a USB-to-serial converter. This auxiliary microcontroller has its own USB bootloader, which allows advanced users to reprogram it.

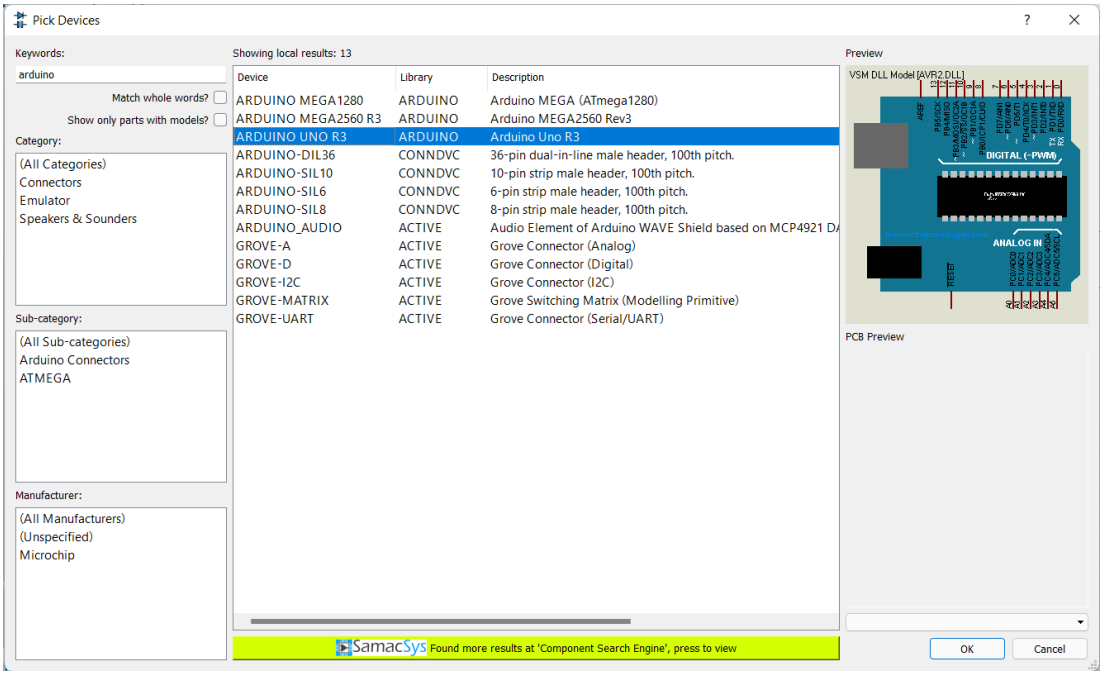

Fig. 4.15 Selecting Arduino UNO R3 from Pick Devices in Proteus

In circuit, we have connected:

Table 1.1 Arduino PINs and Device

Arduino PIN 13	Relay 1 – Fan
Arduino PIN 12	Relay 2 - Lamp
Arduino PIN 11	Relay 3 – LED
RXD PIN 0	COMPIM RXD Pin 2
TXD PIN 1	COMPIM TXD Pin 3

ii. COMPIM

COMPIM is used to model physical COM interfaces in Proteus. It works by capturing and buffering serial signals which it then presents to the electrical circuit. The computer's serial ports will be used to conduct all serial data originating from the CPU or the UART model.

Virtual serial ports can also be created using USB or Bluetooth connectivity by using several technical workarounds. Baud rate conversion is possible when using the COMPIM model. Verification of the virtual and physical characteristics of the device can be implemented through the addition of optional hardware to software.

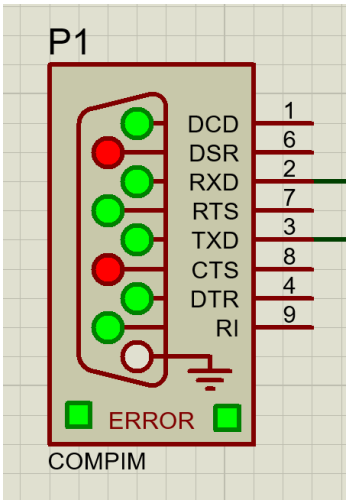

Fig. 4.16 COMPIM in Proteus

iii. FAN DC

DC Fan in Proteus, we've used it as part of circuit.

iv. LAMP-

Lamp is used as another electric device to show simulation of the circuit

v. LED (Green & Red)-

Two LED connected in Parallel mode to show simulation in Proteus via Android App.

vi. Relay

A power relay module is an electrical switch that is operated by an electromagnet. The electromagnet is activated by a separate low-power signal from a micro controller. When activated, the electromagnet pulls to either open or close an electrical circuit.

A simple relay consists of wire coil wrapped around a soft iron core, or solenoid, an iron yoke that delivers a low reluctance path for magnetic flux, a

movable iron armature and one or more sets of contacts. The movable armature is hinged to the yoke and linked to one or more set of the moving contacts. Held in place by a spring, the armature leaves a gap in the magnetic circuit when the relay is de-energized. While in this position, one of the two sets of contacts is closed while the other set remains open.

When electrical current is passed through a coil, it generates a magnetic field that in turn activates the armature. This movement of the movable contacts makes or breaks a connection with the fixed contact. When the relay is de-energized, the sets of contacts that were closed, open and breaks the connection and vice versa if the contacts were open. When switching off the current to the coil, the armature is returned, by force, to its relaxed position. This force is usually provided by a spring, but gravity can also be used in certain applications. Most power relays are manufactured to operate in a quick manner.

4.3.2.2 Terminals used in circuit-

i. **Power**

ii. **Ground**

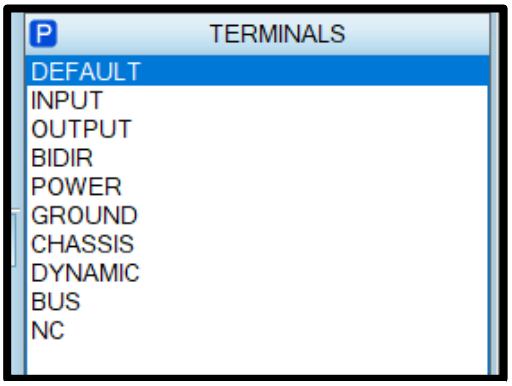

Fig. 4.17 Terminal in Proteus

i. **Power-**

In the circuit design, we have used 5V Power Supply in Proteus power in the circuit.

ii. **Ground**

Another terminal interface Ground used at different places of circuit to provide balance to the circuit.

After adding terminal and device components and following circuit diagram, we've finally designed a full circuit in proteus which we use for simulation using Android App.

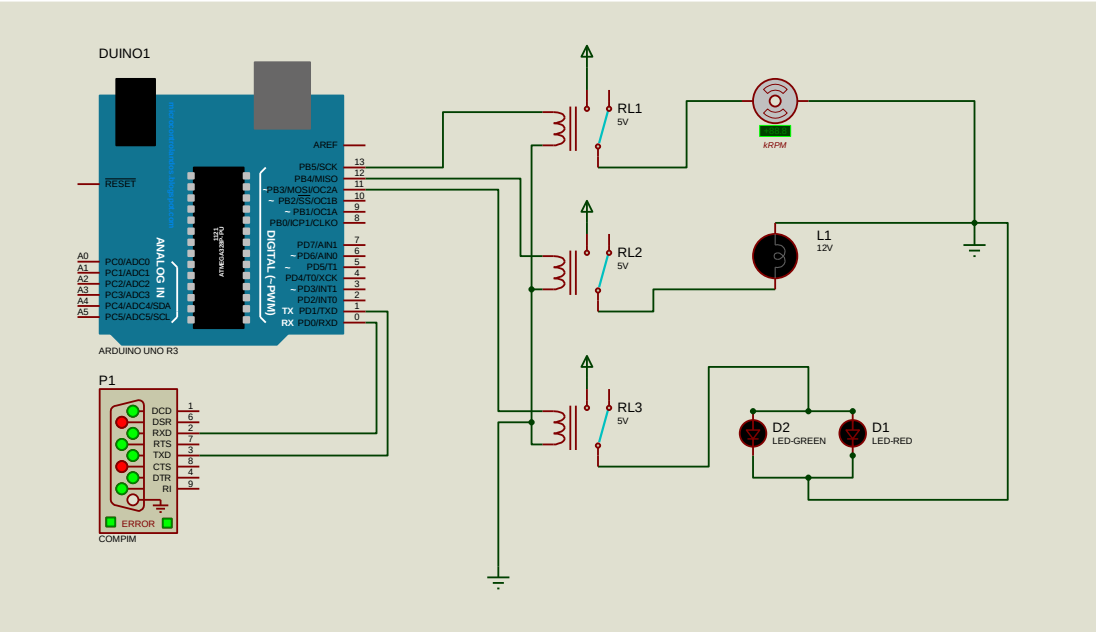

Fig. 4.18 Circuit in Proteus 8 Professional

4.4 Accounts on Adafruit & IFTTT

4.4.1 Adafruit.io

Adafruit IO is a platform designed to display, respond, and interact with project's data. It also keeps our data private (data feeds are private by default) and secure. It's the internet of things - for everyone!

4.4.1.1 Creating Account on Adafruit IO

It is very simple to create account on Adafuit.io. We just have to visit sign up of Adafruit (https://accounts.adafruit.io/users/sign_up) and filling necessary details such as First Name, Last Name then we have to select a username and then password. That's all we're ready to enter in Adafruit.IO platform.

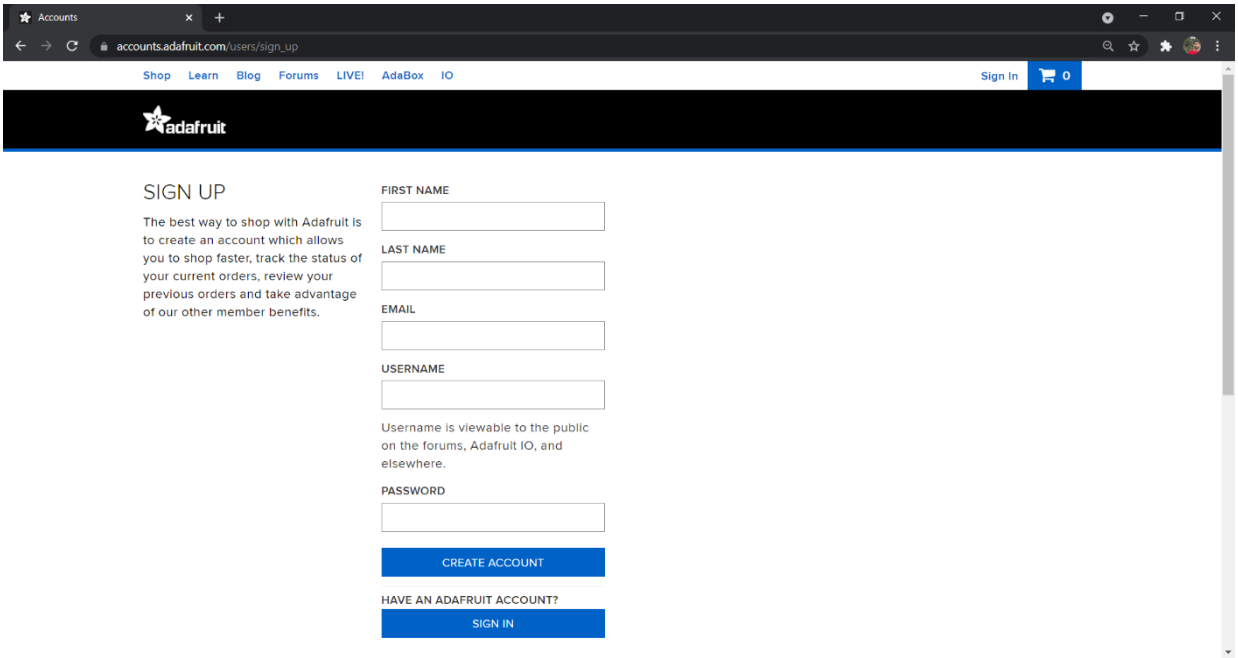

Fig. 4.19 Adafruit Sign Up Page

4.4.1.2 Creating Dashboard on Adafruit.io

Dashboards are a feature integrated into Adafruit IO which allow us to chart, graph, gauge, log, and display our data. We can view our dashboards from anywhere in the world.

Creating dashboard is first and foremost thing on this platform. To create the dashboard, we have to follow following steps: -

Step 1: First, we have to navigate to *Dashboards*

Step 2: Then we click on *Create New Dashboard* Button

Step 3: Then Enter *Name* and *Description* of Dashboard then click on *Create.*

Step 4: Now our dashboard is ready for use.

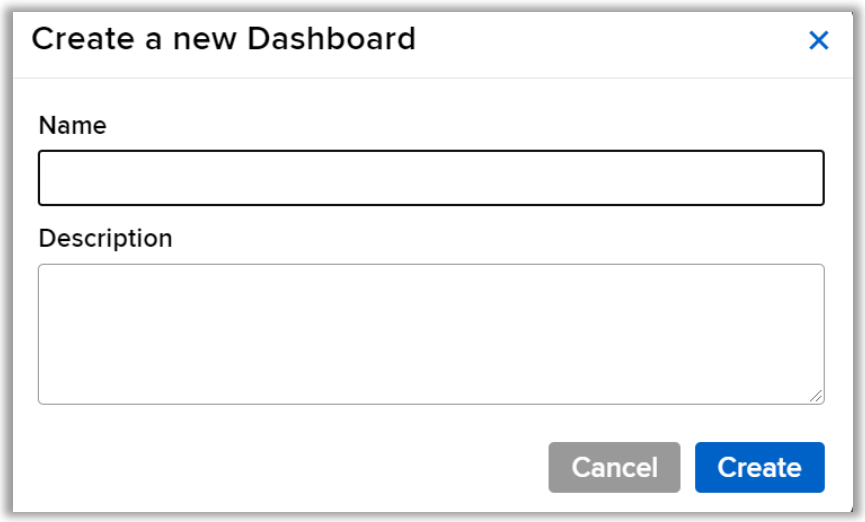

Fig 4.20 Create a new Dashboard snippet

We have created our own dashboard with name ***"fyphomeauto"***

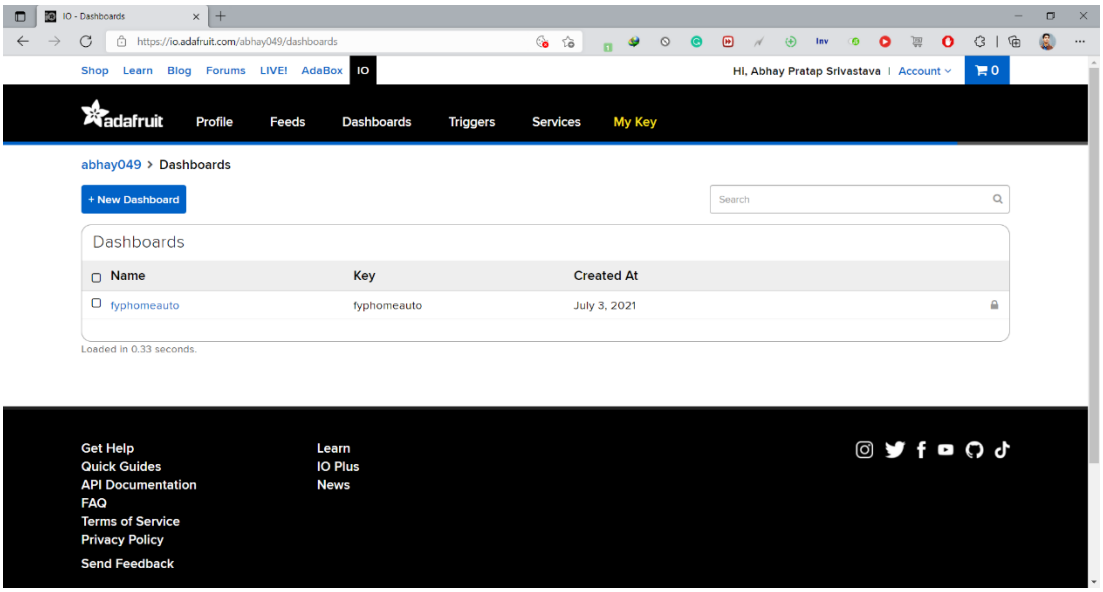

Fig 4.21 Snippet: New Dashboard with name "fyphomeauto"

4.4.1.3 Adafruit IO Key:

On creating account on Adafruit.io, it provides us key that we use in Arduino IDE or in scripting.

Adafruit Key of our account.

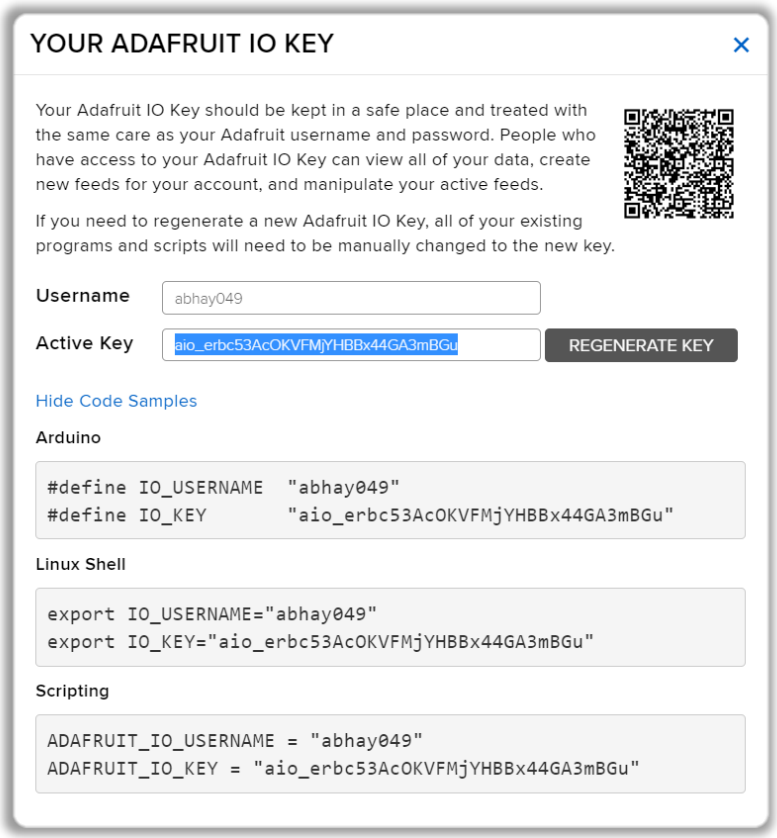

Fig 4.22 Adafruit IO Key

4.4.2 IFTTT

If This Then That is a service that allows a user to program a response to events in the world. IFTTT has partnerships with different service providers that supply event notifications to IFTTT and execute commands that implement the responses. Some event and command interfaces are simply public APIs.

4.4.2.1 Create Account on IFTTT

To create account on IFTTT, we have to visit https://ifttt.com and then we have to navigate to Login.

There is option to sign up or just login through Google, Facebook or Apple.

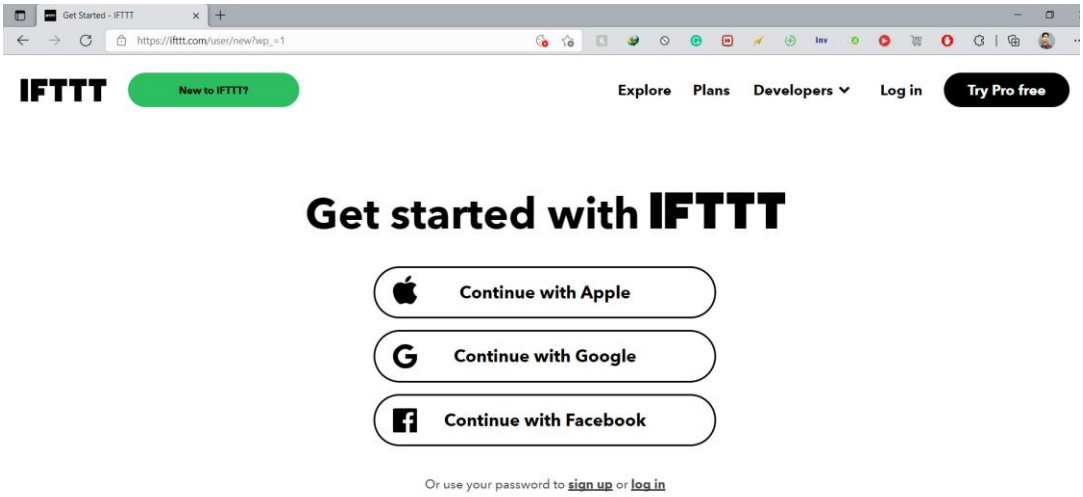

Fig 4.23 Sign Up & Login using Google, Apple or Facebook

We've to sign up on both the platform with same Google Account. Also make sure we select only that google account which is login in android phone having Google Assistant.

4.5 Creating Project in Blynk Android App

To control home appliance in Proteus we have used Blynk IoT App. Blynk is an IoT platform for iOS or Android smartphones that is used to control Arduino, Raspberry Pi and NodeMCU via the Internet.

Steps followed to setup account on Blynk:

Step 1: First we have to create account using email and then Blynk will send a Token Key that we'll use in our program/code for Arduino UNO R3.

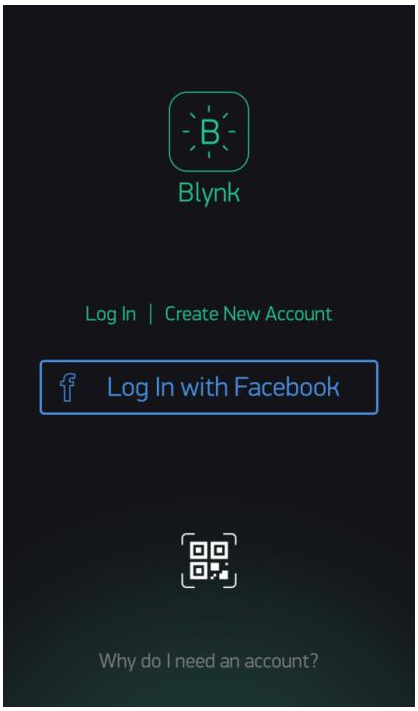

Fig 4.24 Sign up Blynk

Step 2: After login into the account, we have to create news project by tapping on ***+New Project.*** In the very next step, we have to enter Project Name ***"homeuno"*** Select Device as ***Arduino UNO*** from drop down list and then we have to decide connection type as ***"Wi-Fi".*** Then selecting the theme ***Dark/Light*** we create our project.

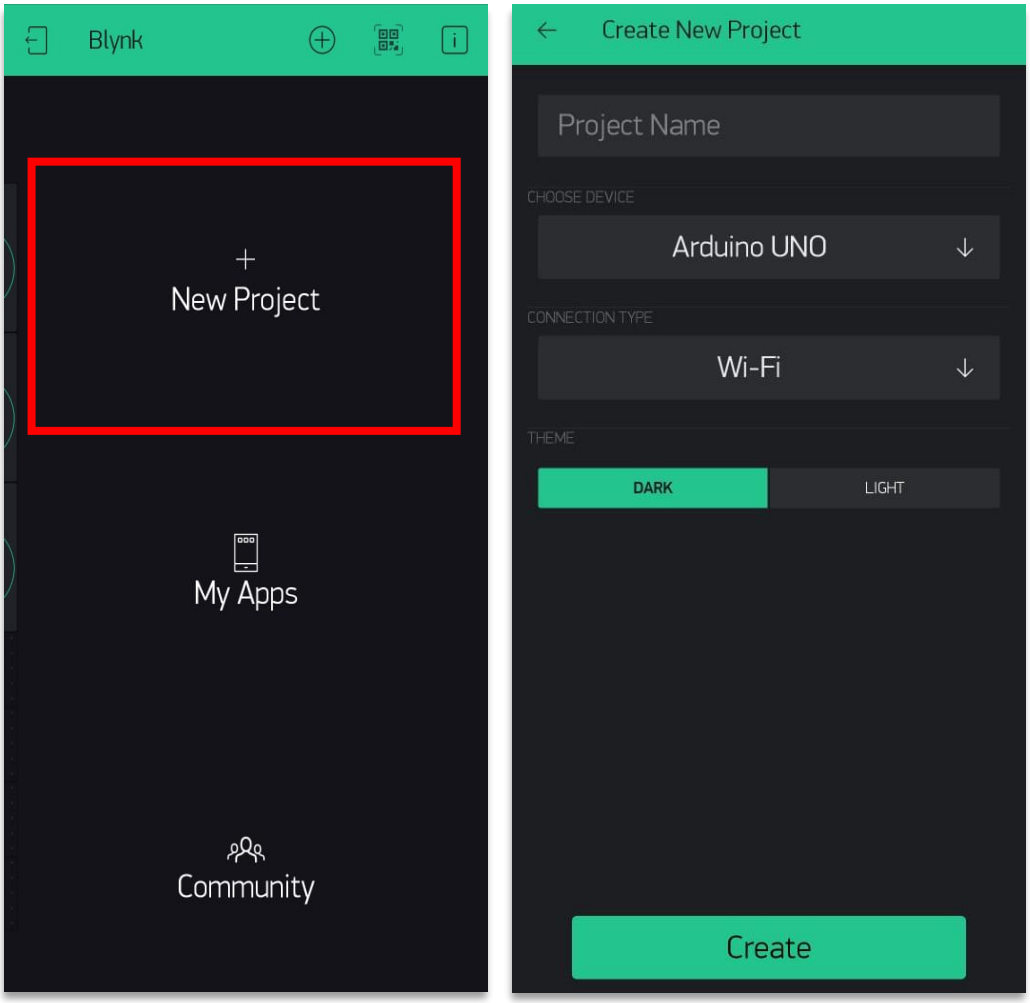

Fig 4.25 Project Configuration in Blynk

Step 3: Now we have successfully created our project on Blynk App. To add widget like Toggle Button. We top on + icon top of screen in Blynk and add the BUTTON.

Step 4: First we select the ***Button,*** then we need to configure button for specific pin. As in proteus we've connected LED at D11 (pin 11), In the same way, we add D11 for LED in Blynk. Also, we will opt for SWITCH instead of push button. And 0 for Off and 1 for ON.

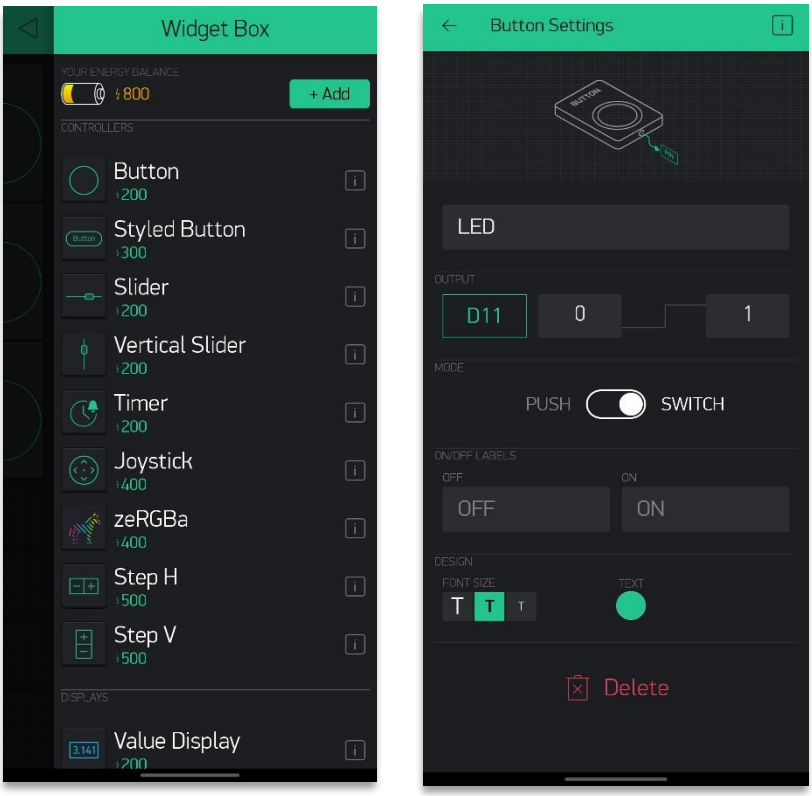

Fig 4.26 Widget and Button Setting in Blynk

Step 5: In the same way. we have created two other switches, for Lamp and Fan at output pin on D12 and D13 respectively.

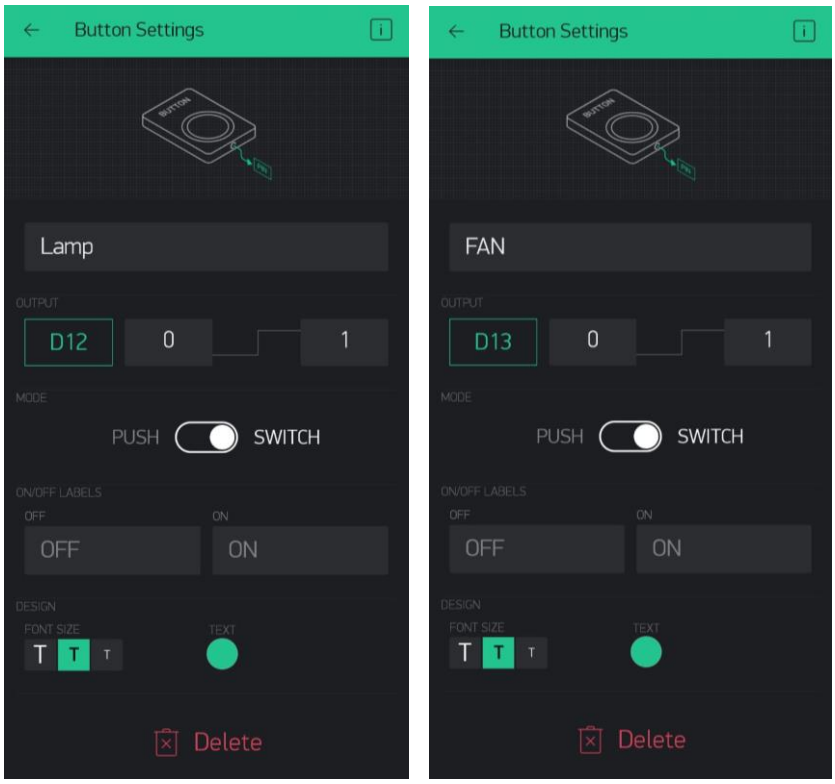

Fig 4.27 Button Setting in Blynk

Step 6: After successfully adding all our switches, our project in Blynk Android App. It looks like this figure.

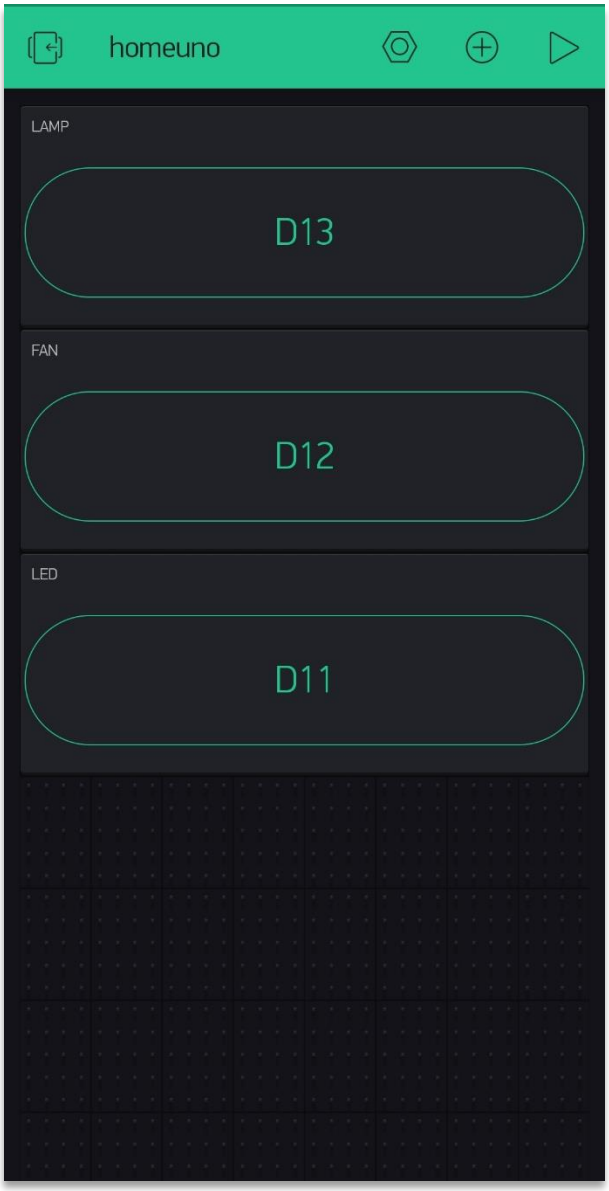

Fig 4.28 After successfully configuring switches for device in Android App

CHAPTER 5
IMPLEMENTATION

5.1 Arduino IDE code to Arduino UNO

We follow these steps to move our code to Arduino UNO in Proteus: -

Step 1: First we write code for our program in Arduino IDE and by including Blynk Library. Also, we'll include Token sent by Blynk while creating the project.

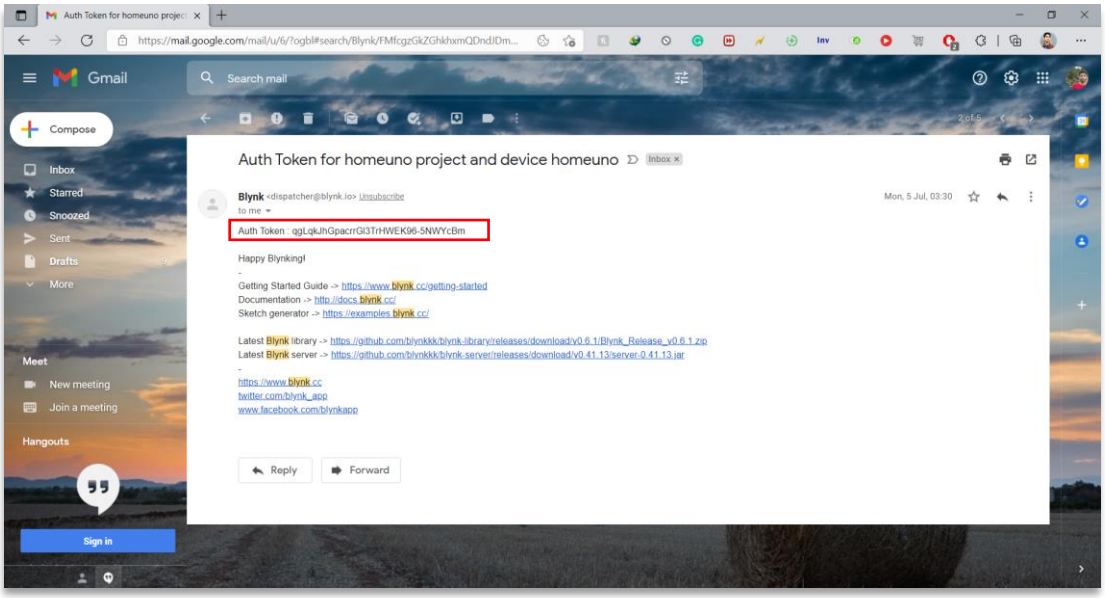

Fig 5.1 Auth Token on Email received from Blynk

Arduino IDE Code for home appliances automation using Android App.

```
#include <BlynkSimpleStream.h>

char auth[] = "qgLqkJhGpacrrGl3TrHWEK96-5NWYcBm";

 void setup()
 {
   Serial.begin(9600);
   Blynk.begin(auth, Serial);
```

```
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
 }

 BLYNK_WRITE(V1) //Button Widget is writing to pin V1
{
  int pinData = param.asInt();
  if(pinData==1){
    digitalWrite(11, HIGH);
  }else{
    digitalWrite(11, LOW);
  }
}

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

Fig 5.2 Snipped of Code from Arduino IDE

Step 2: After writing code we'll compile this code and generate a hex file. When Arduino compiles your sketches, it produces an output file called Hex file.

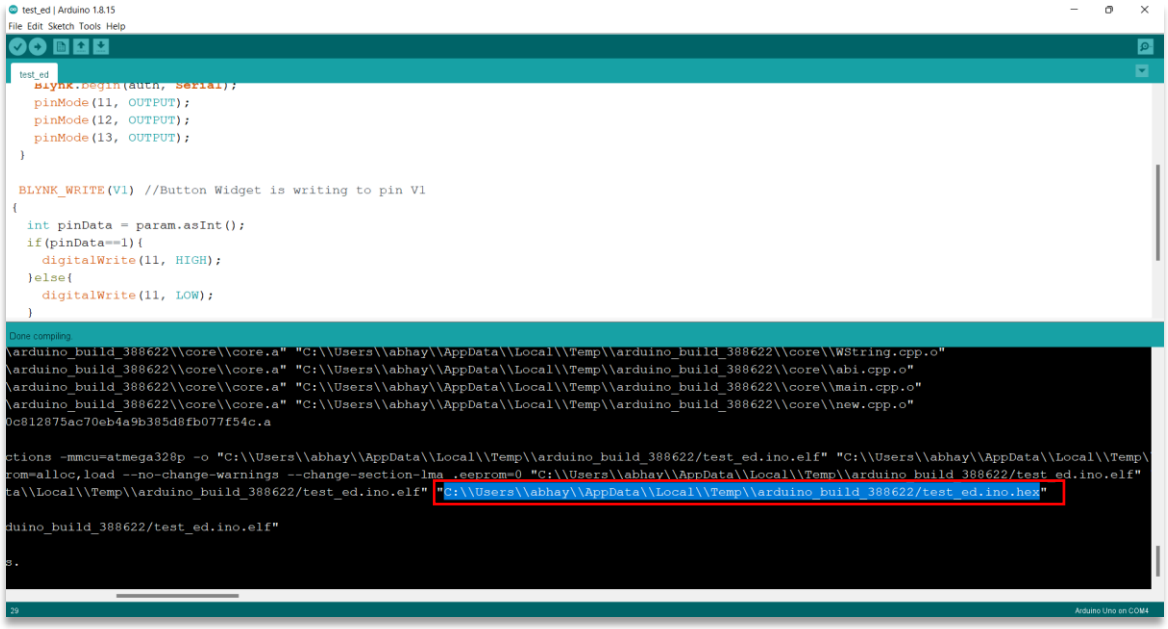

Fig 5.3 Hex Code path generated in Arduino IDE

Our hex file path is:

"c:\\users\\abhay\\appdata\\local\\temp\\arduino_build_388622/test_ed.ino.hex"

Step 3: After copying the hex file path. Now we'll open Proteus 8 Professional and To paste this path we'll double click on Arduino UNO R3. It opens a new window called "Edit Component".

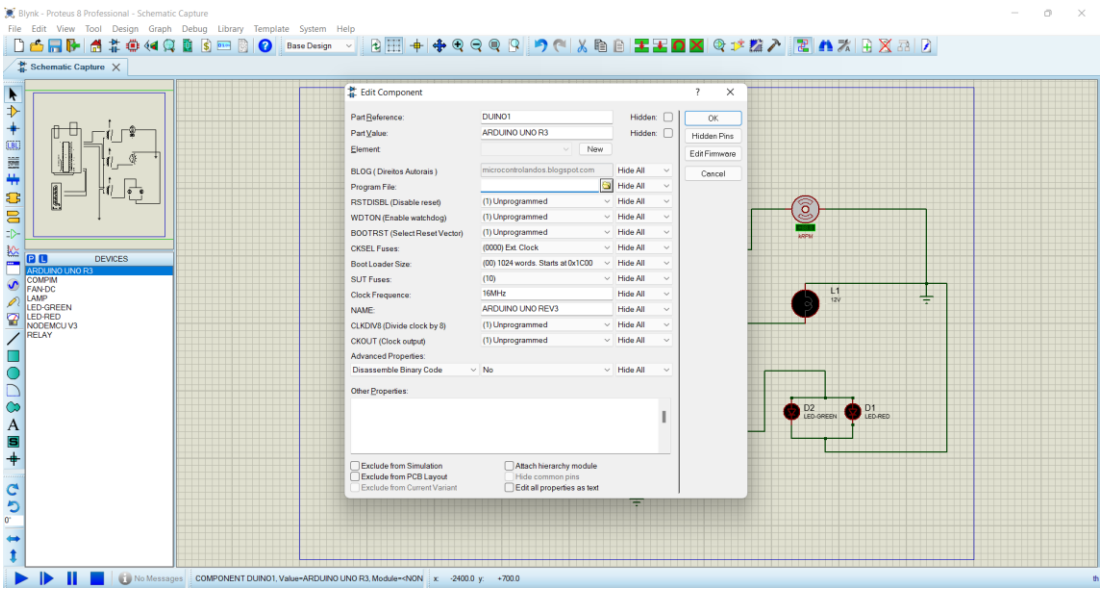

Fig 5.4 After double clicking on Arduino UNO, we see a new window "Edit Component"

Step 4: In *Edit Component* Window, there is option to add path file for the program i.e., ***Program File***. This is where we have to paste copied hex file path code.

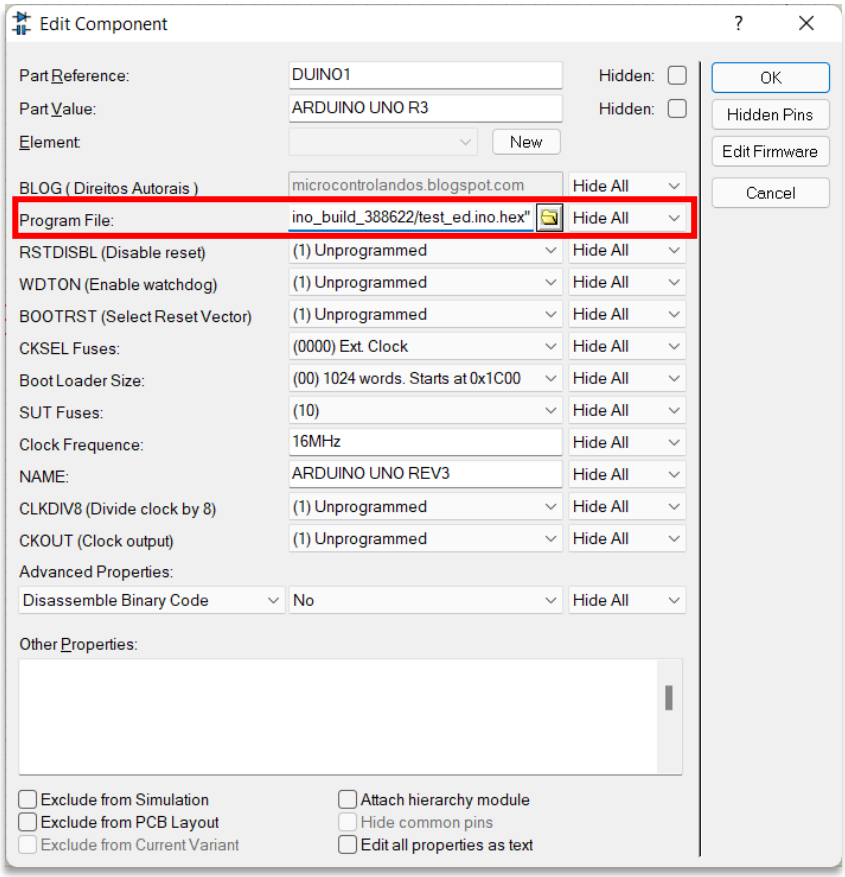

Fig 5.5 Program File in Edit Component Window

5.2 Proteus Connection to Android App Blynk

To connect Proteus 8 to Blynk Android App. we'll use COMPIM which will provide virtual serial port.

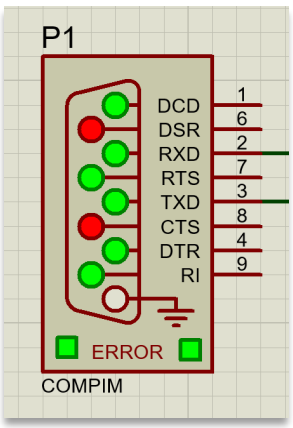

Fig 5.6 COMPIM in Proteus

First, we choose port for COMPIM device in Proteus and then connect this virtual port with Blynk default port number COM1 using VSPE software. To do so we follow these steps: -

Step 1: Double click on COMPIM in Proteus to get to Edit Component window. And then from there we'll change *Physical Port from Default to COM 2*. Also, *Physical Baud Rate to 9600* and *Virtual Baud Rate to 9600.*

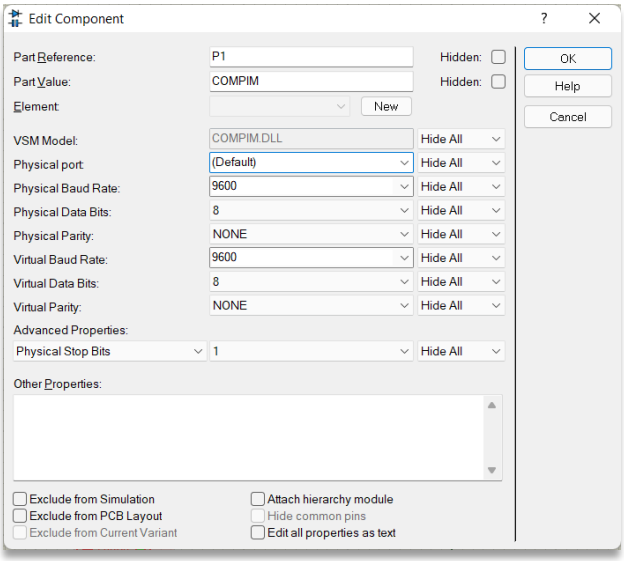

Fig 5.7 ***Before*** *changing COMPIM settings.*

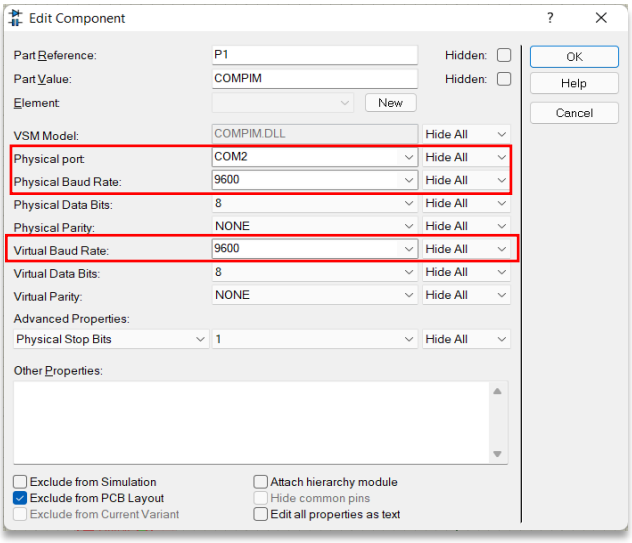

Fig 5.8 ***After*** *Changing Settings of COMPIM.*

Step 2: After successfully changing these settings, we save our project.

Step 3: By default, Blynk Android App COMP port is COM1. We can check by navigating to –

"C:\Users\abhay\Documents\Arduino\libraries\Blynk\scripts\blynk-ser.bat"

When we edit or open this file ***"blynk-ser.bat"*** with Notepad, we can see the default values of Blynk Android App settings.

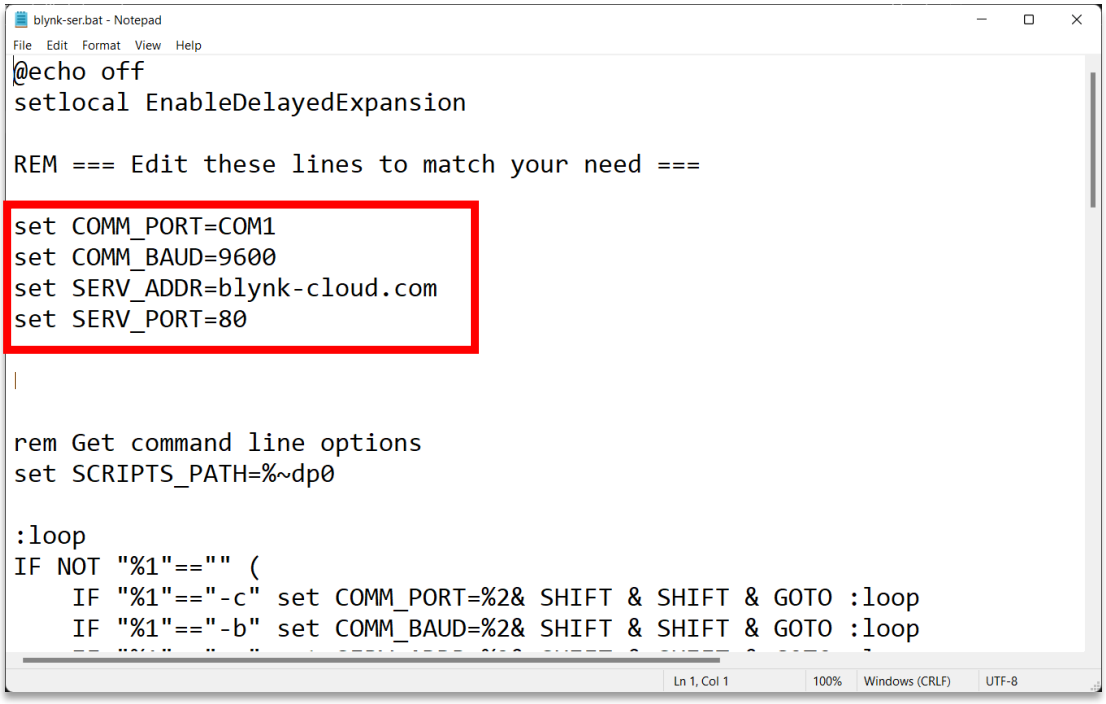

Fig 5.9 blynk-ser.bat file in Notepad.

Line by Line command of "blynk-ser.bat" **file -**

```
@echo off
setlocal EnableDelayedExpansion

REM === Edit these lines to match your need ===

set COMM_PORT=COM1
set COMM_BAUD=9600
set SERV_ADDR=blynk-cloud.com
set SERV_PORT=80


rem Get command line options
set SCRIPTS_PATH=%~dp0

:loop
IF NOT "%1"=="" (
    IF "%1"=="-c" set COMM_PORT=%2& SHIFT & SHIFT & GOTO :loop
    IF "%1"=="-b" set COMM_BAUD=%2& SHIFT & SHIFT & GOTO :loop
    IF "%1"=="-s" set SERV_ADDR=%2& SHIFT & SHIFT & GOTO :loop
    IF "%1"=="-p" set SERV_PORT=%2& SHIFT & SHIFT & GOTO :loop

    CALL :usage
    GOTO :eof
)

rem Find ports
set PORTS=
for /f "tokens=4 delims=: " %%A in ('mode^|findstr "COM[0-9]*:"') do IF not
[%%A] == [] set PORTS=!PORTS! %%A
```

```
set PORTS=!PORTS:~1!

rem Check port
rem Skip check if no ports at all - Windows bug?
if not "x%PORTS%"=="x~1" (
    if "x!PORTS:%COMM_PORT%=!"=="x%PORTS%" (
        echo %COMM_PORT% not found, or may be busy.
        set /p COMM_PORT="Select serial port [ %PORTS% ]: "
    )
)

rem Create exe
if not exist "%SCRIPTS_PATH%\com2tcp.exe" (
    copy "%SCRIPTS_PATH%\com2tcp.bin" "%SCRIPTS_PATH%\com2tcp.exe" > NUL
)

rem Do the job
echo Connecting device at %COMM_PORT% to %SERV_ADDR%:%SERV_PORT%...

rem Try resetting board
rem mode %COMM_PORT%:%COMM_BAUD%,N,8,1 >nul

:restart
  "%SCRIPTS_PATH%\com2tcp.exe"     --baud     %COMM_BAUD%     --ignore-dsr
\\.\%COMM_PORT% %SERV_ADDR% %SERV_PORT%
  echo Reconnecting in 3s...
  timeout /T 3
goto restart


goto:eof

:usage
    echo.
    echo.   This script redirects serial communication to the server.
    echo.
    echo.   You can specify port, baud rate, and server endpoint like this:
    echo.     blynk-ser.bat -c ^<serial port^> -b ^<baud^> -s ^<server^> -p
^<port^>
    echo.
    echo.   The defaults are:
    echo.      -c    /dev/ttyUSB0       (on Linux)
    echo.            COM1               (on Windows)
    echo.            /dev/tty.usbserial (on OSX)
    echo.      -b    9600
    echo.      -s    blynk-cloud.com
    echo.      -p    80
    echo.
    echo.   If the specified serial port is not found, it will ask to enter
another one.
    echo.   The script also tries to reestablish connection if it was lost.
goto:eof
```

We need to run this file while running our project, this is the program which will connect Proteus to Blynk.

5.3 Virtual Serial Port Implementation

Virtual Serial Ports Emulator (VSPE) is one of the key software or programs that is used to connect port Proteus 8 with Blynk. We follow these steps to connect both of these software for simulation: -

Step 1: Download VSPE from http://www.eterlogic.com/Downloads.html website.

Step 2: Install the software.

Step 3: *Open VSPE* >

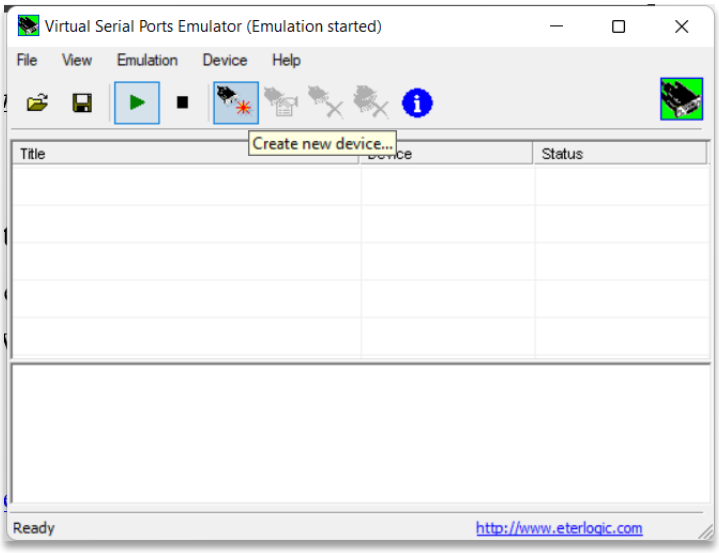

Fig 5.10 VSPE

Step 4 : Create New Device > *Device Type: Pair*

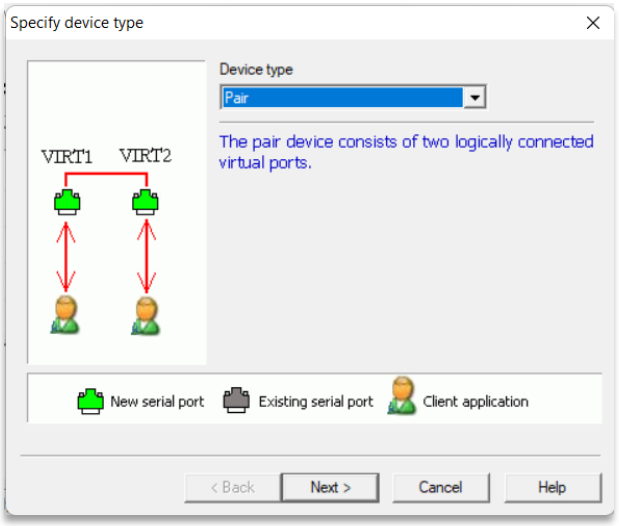

Fig 5.10 Pairing in VSPE

***Step 5**:* Serial Port 1: COM 2 to Serial Port 2: COM 1 > Finish

***Step 6**:* Now COM 2 ⇔ COM 1 simulation working.

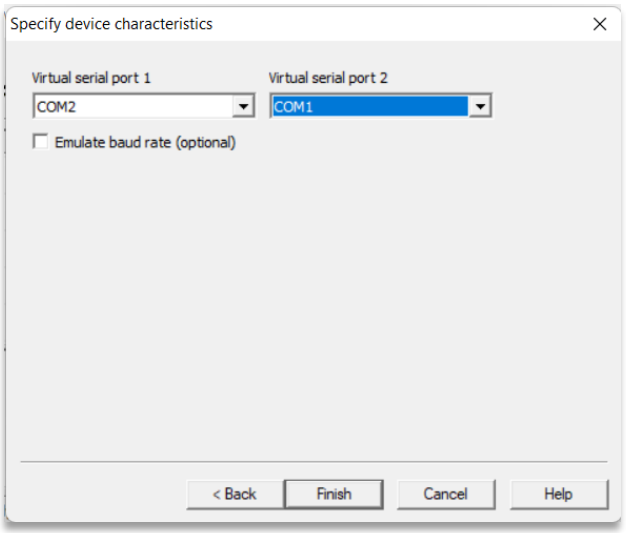

Fig 5.11 Specifying Devices Characteristics

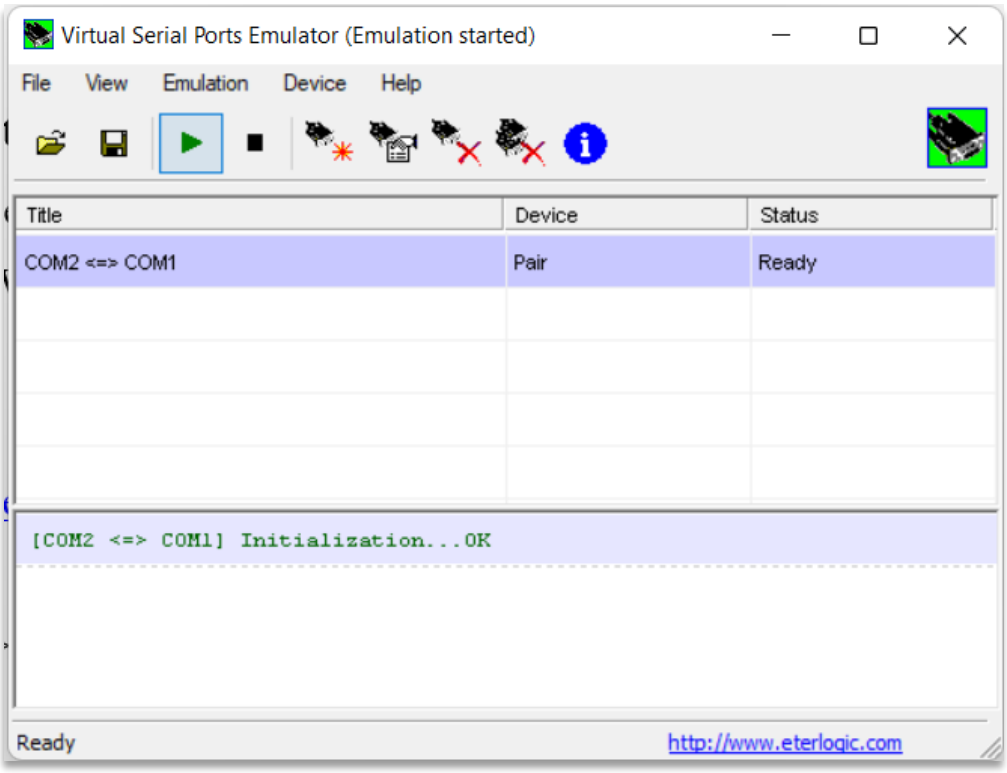

Fig 5.12 Emulation Started

5.4 Working with Proteus Simulator

Now we have setup everything, it's time to simulate the appliances using Android App. We just have to press Play button to simulate.

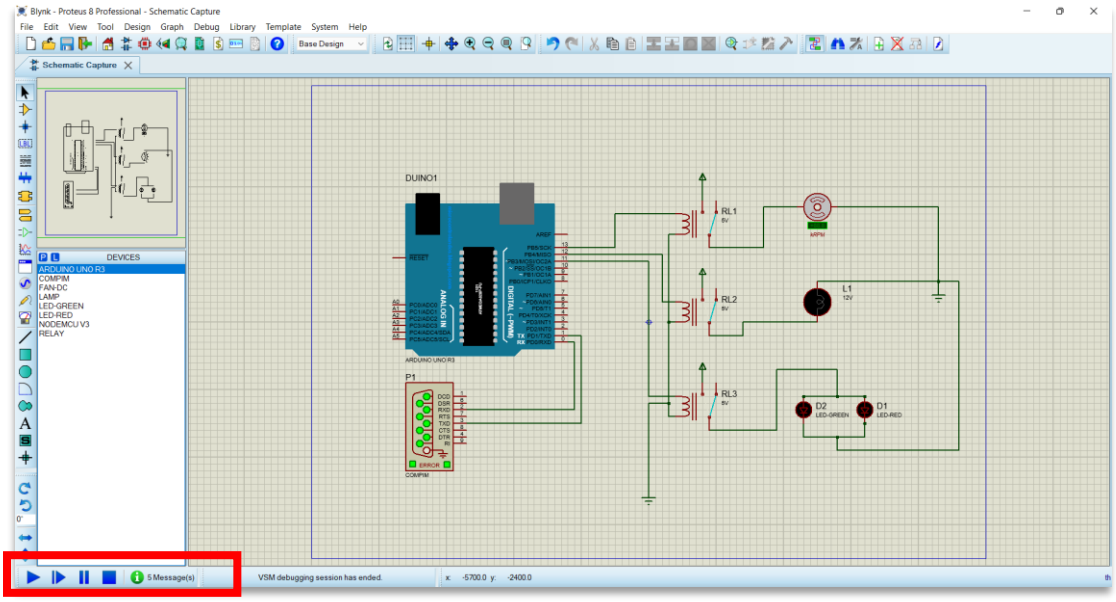

Fig 5.13 Click on play button to start simulation

Now we've to navigate to this path to run .bat file in order to connect proteus with Blynk App –

C:\Users\abhay\Documents\Arduino\libraries\Blynk\scripts and run blynk-ser.bat file.

Fig 5.14 Running blynk-ser.bat file.

When all these programs run simultaneous, we'll get notification on Mobile App that we're now connected and from there we can control the home appliances.

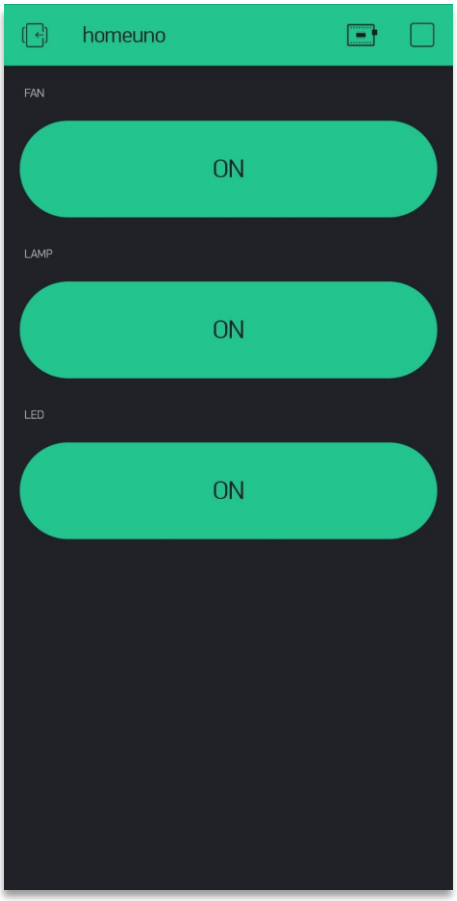

Fig 5.15 On Giving Instruction to on all switches.

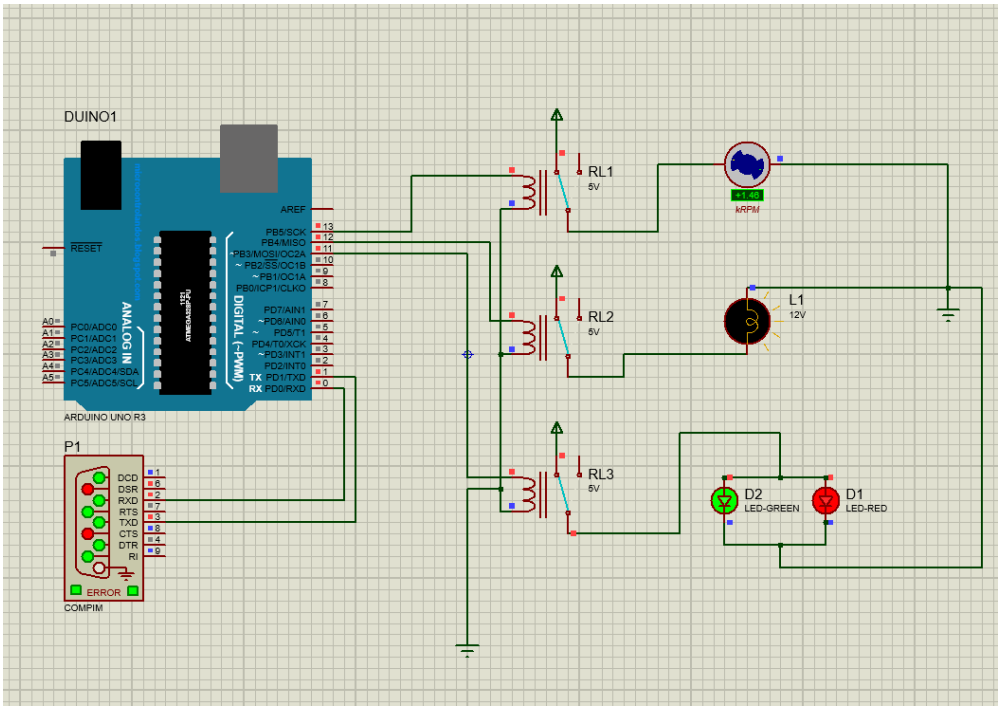

Fig 5.16 All appliances are in working mode.

5.5 Adafruit.io- Dashboard & Feed

In 4.4.1.2 we have created Dashboard on Adafruit

We have created our own dashboard with name ***"fyphomeauto"***

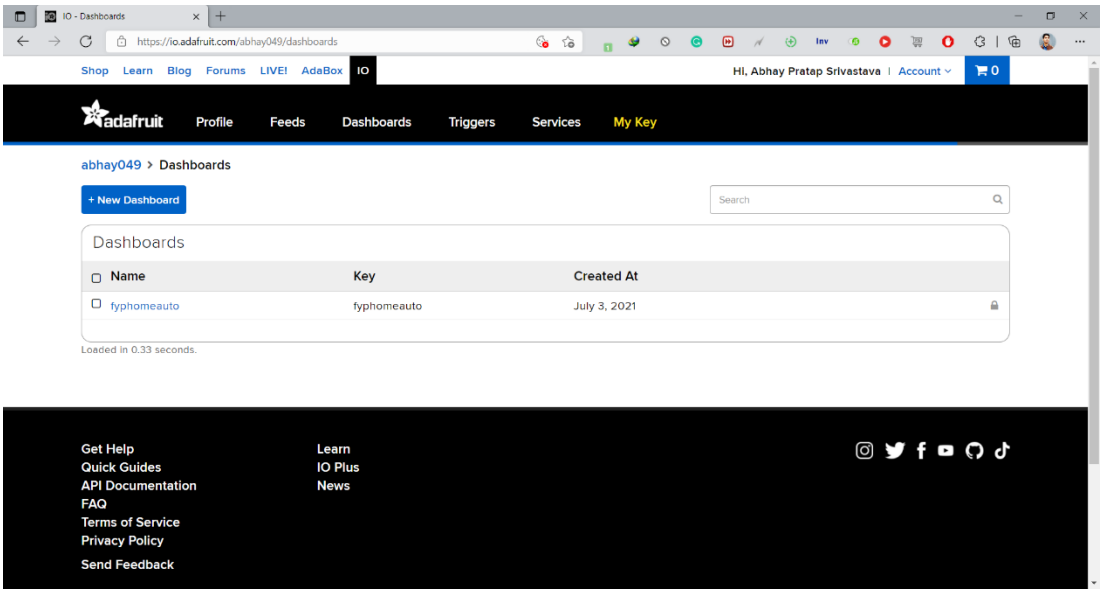

Fig 5.17 Snippet: New Dashboard with name "fyphomeauto"

5.5.1 Creating Feeds on Adafruit.io

Now, Feeds are the core of the Adafruit IO system. The feed holds metadata about the data we push to Adafruit IO. This includes settings for whether the data is public or private, what license the stored sensor data falls under, and a general description of the data. The feed also contains the sensor data values that get pushed to Adafruit IO from device.

We will need to create one feed for each unique source of data we send to the system.

Creating a Feed on Adafruit IO is a very simple process. When we login to our Adafruit account, we will see Feeds top in Menu, when we click on Feeds we'll directed to a new page of Feeds as shown in Figure.

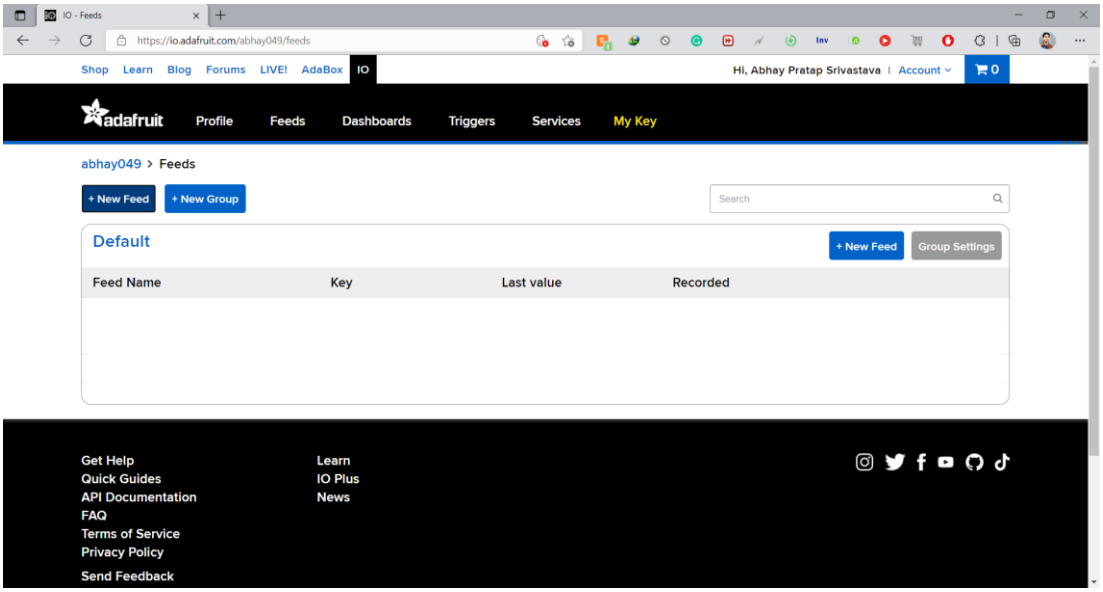

Fig 5.18 Feeds on Adafruit.io

Steps to create a New Feed: -

Step 1: Click on *+New Feed* Left of screen to create a feed.

Step 2: Enter Name and Description of Feed as shown in figure. We have entered feed name as relay 1.

Step 3: Similarly. we'll create three feeds namely- relay 1, relay 2 and relay 3. Each for our device- Fan, Lamp and LED.

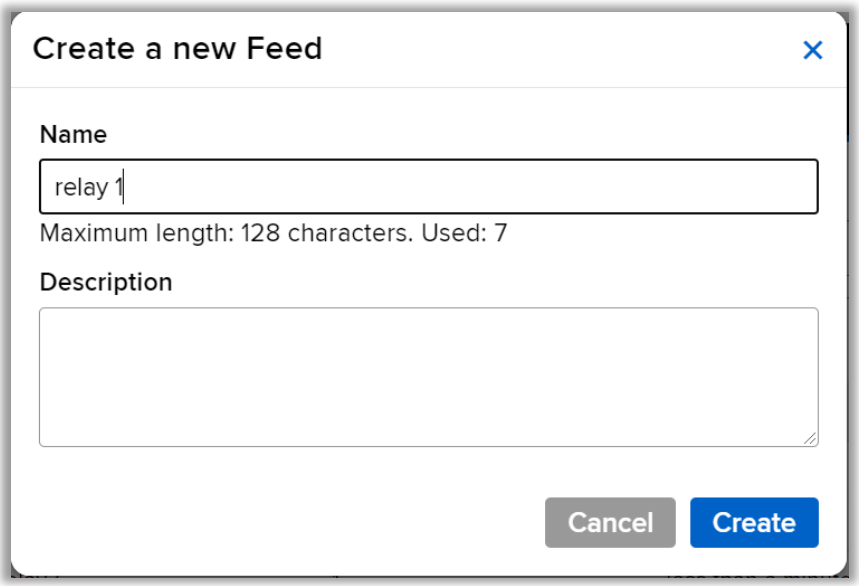

Fig 5.19 Creating New Feed in Adafruit.io

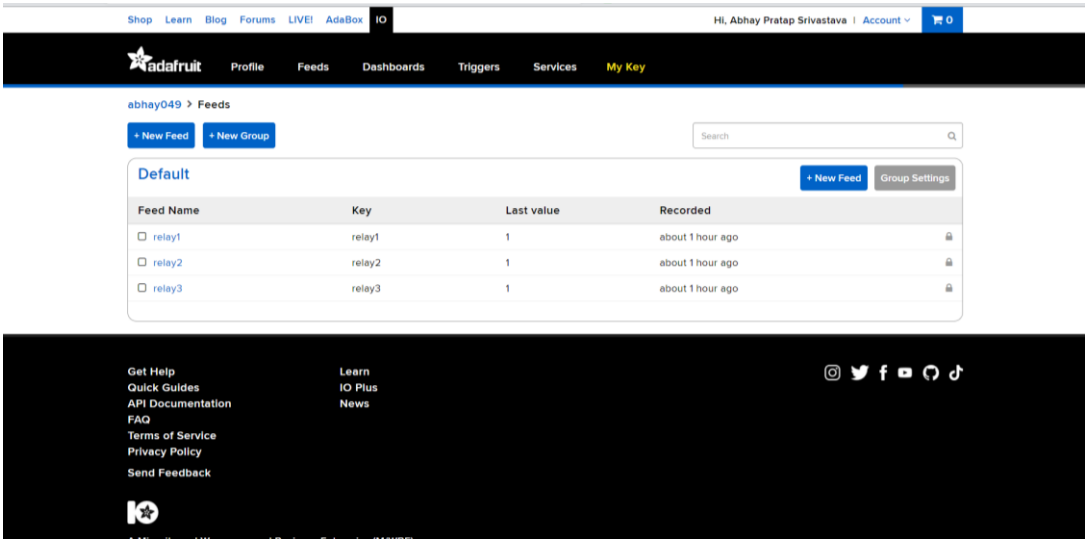

Fig 5.20 After creating all three feeds

5.5.2 Creating Toggle Block and Connecting with Feeds

After creating feeds, it is now turn to connect these feeds with Toggle Block so that they can work and give us simulation/visual results of project.

For that we'll follow these steps in Adafruit.io: -

Step 1: Go to *Dashboard* and select your *Dashboard*. In our case we select our dashboard *"fyphomeauto"*

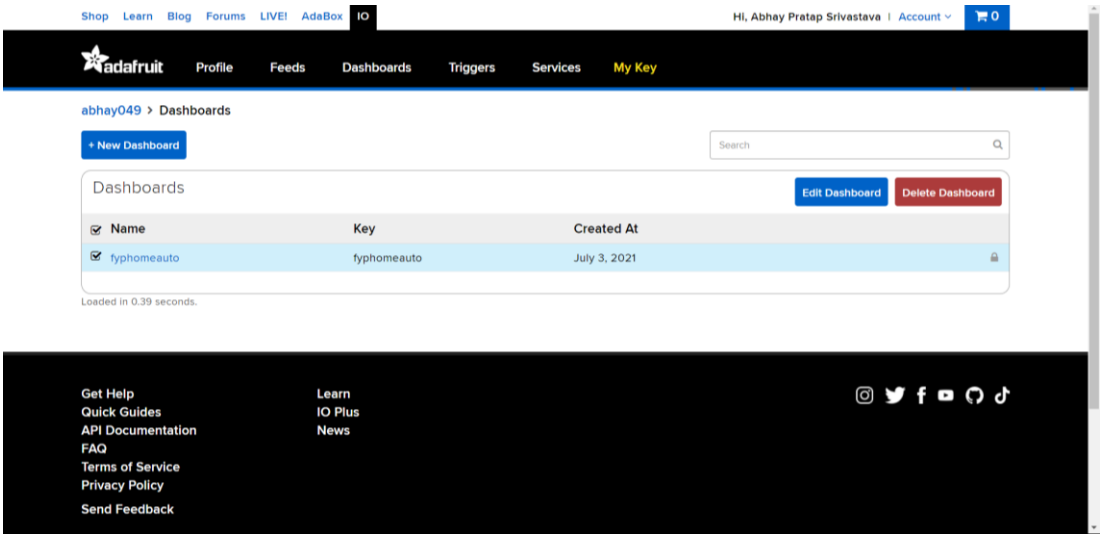

Fig 5.21 Selecting Dashboard in Adafruit

Step 2: Now Click on *Setting* Icon located on left side of screen. And then from dropdown menu select *Create New Block.*

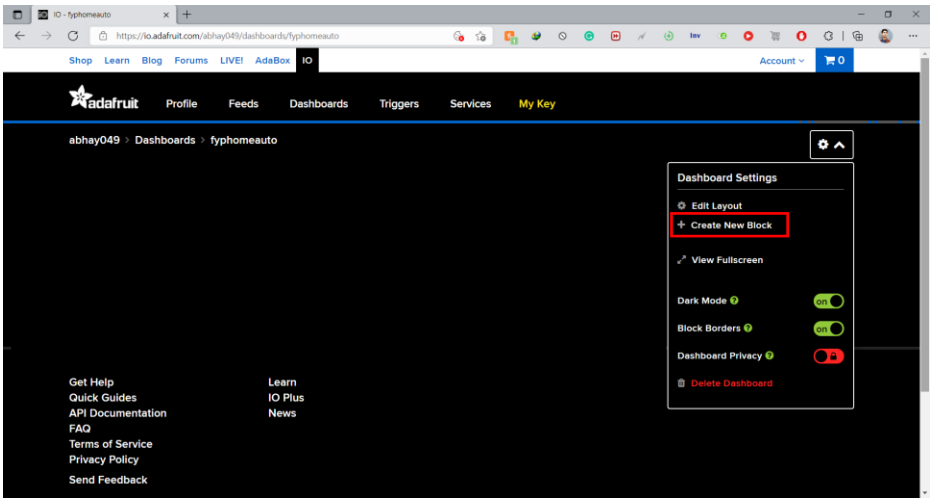

Fig 5.22 Creating New Block on Adafruit

Step 3: After clicking on *Create New Block*, it will give option to select one block from list of blocks. In our case we'll sect *Toggle (ON/OFF) block.*

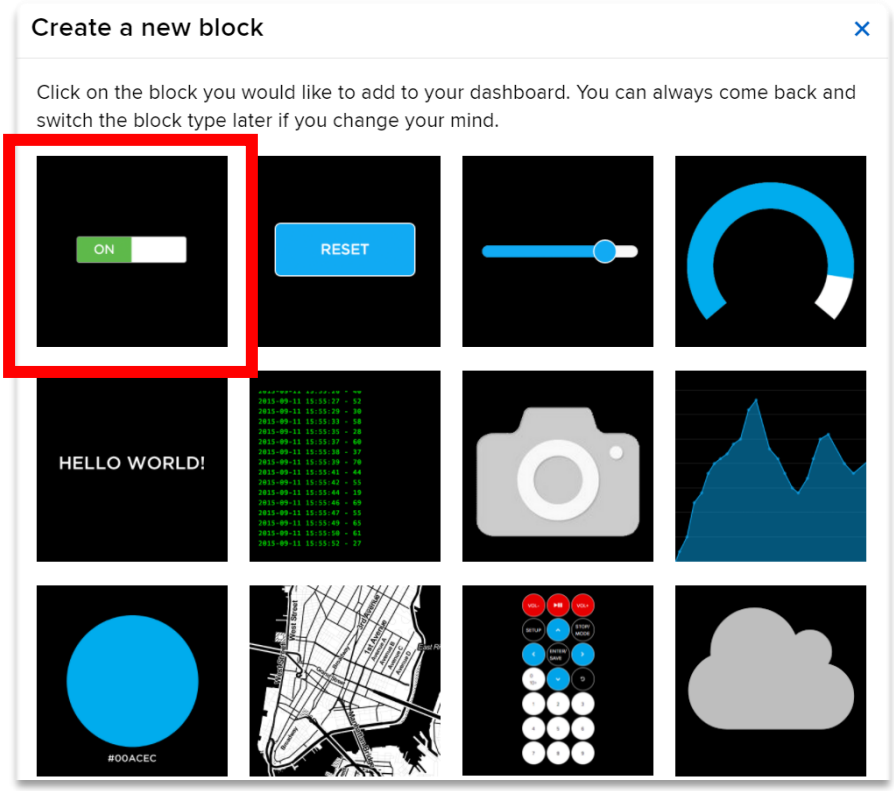

Fig 5.23 Selecting Toggle from Blocks List

Step 4: As soon as we select Toggle as block, in the very next step it asks to connect the Feed (that we've created earlier). One at a time for each block we'll select one relay/feed as shown in the figure.

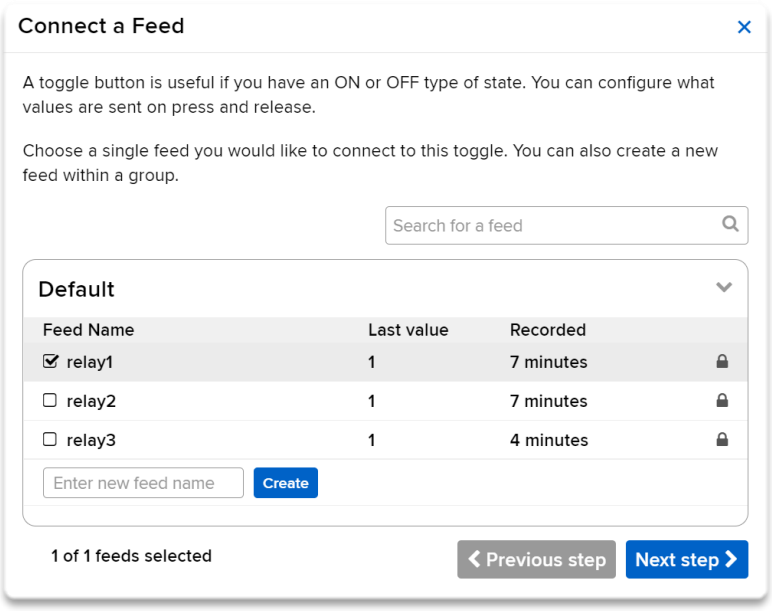

Fig 5.24 Connecting Feed to Block

Step 5: Clicking on Next Step we need to configure *Block Title, Button on Text, Button on Value, Button Off Text and Button Off Value.*

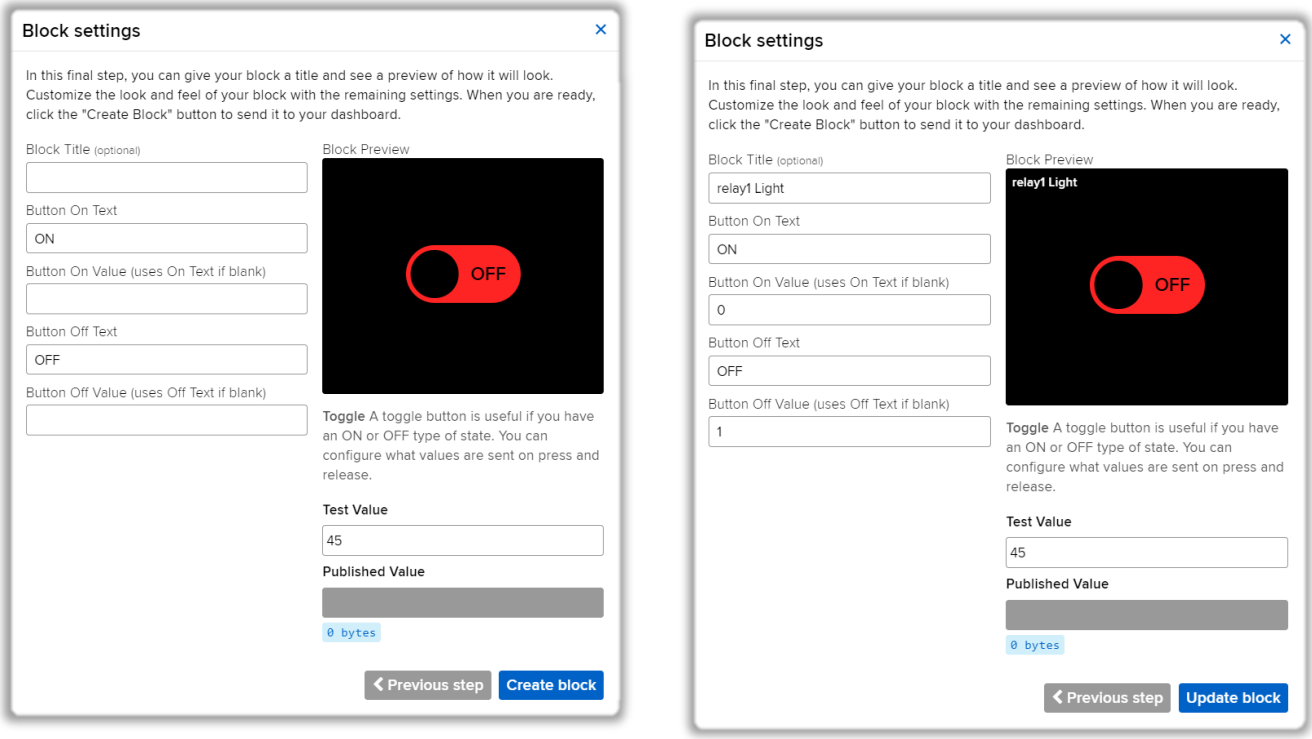

Fig 5.25 Block Setting

We'll configure it for all three relay as:

Table 1.2 Relay1 Light Settings

Block Title:	relay1 Light
Button On Text:	ON
Button On Value:	0
Button Off Text:	OFF
Button Off Value:	1

Table 1.4 Relay3 Lamp Settings

Block Title:	relay3 Lamp
Button On Text:	ON
Button On Value:	0
Button Off Text:	OFF
Button Off Value:	1

Table 1.3 Relay2 Fan Settings

Block Title:	relay2 Fan
Button On Text:	ON
Button On Value:	0
Button Off Text:	OFF
Button Off Value:	1

After creating and configuring we can see our feeds connected to Toggle block and now we can view it on Dashboard as shown in figure.

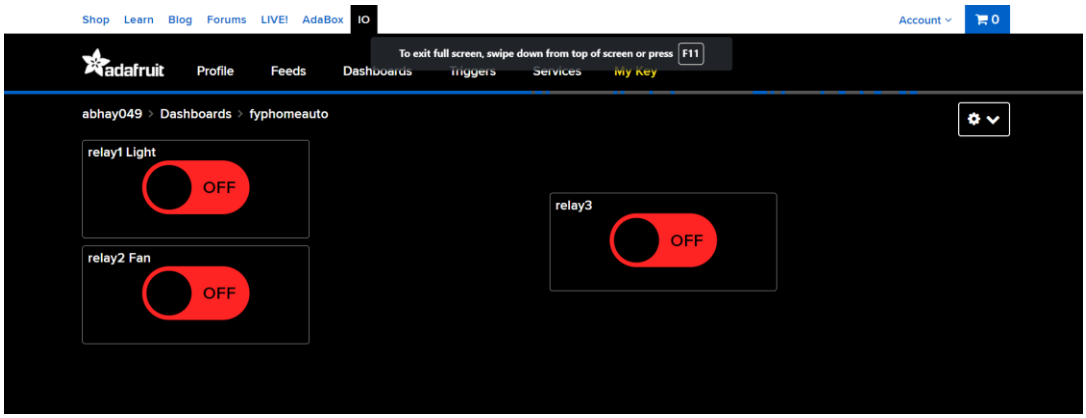

Fig 5.26 Dashboard Adafruit after configuration

5.6 IFTTT Applet & Integration with Adafruit

Following *4.4.2 IFTTT*, after we successfully Login into IFTTT, we have to create our own Applet to connect Google Assistant and Adafruit together.

Applet-

An Applet connects two or more apps or devices together. It enables you to do something that those apps or devices couldn't do on their own.

In Free account of IFTTT, any user can create a maximum of **three applet.**

We're going to create applet for three actions-

-To turn on Light

-To turn off Light

-To turn on Fan

In order to create applet on IFTTT we have to go by these steps-

Step 1: Go to top corner > ***Create*** Button

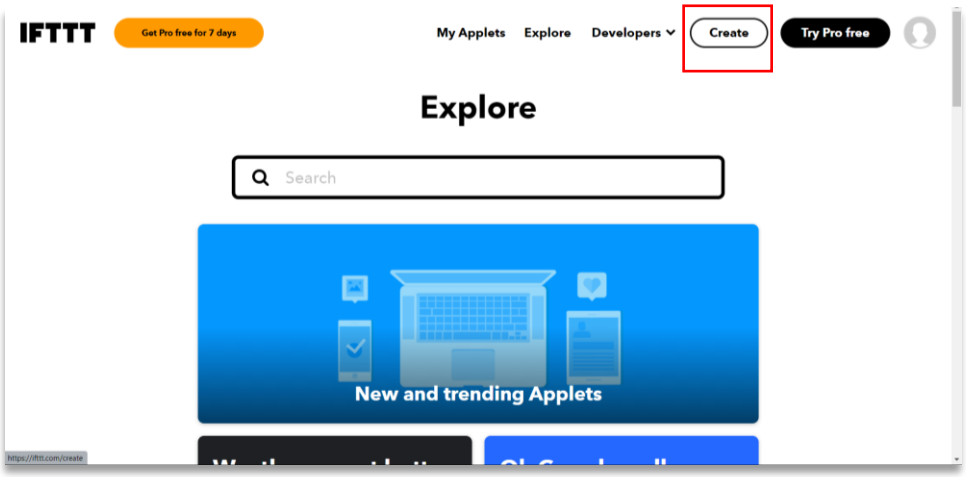

Fig 5.27 IFTTT Explore Page

Step 2: Go to Create or directly via link https://ifttt.com/create . Now on create page we have to connect Google Assistant with Adafruit.io

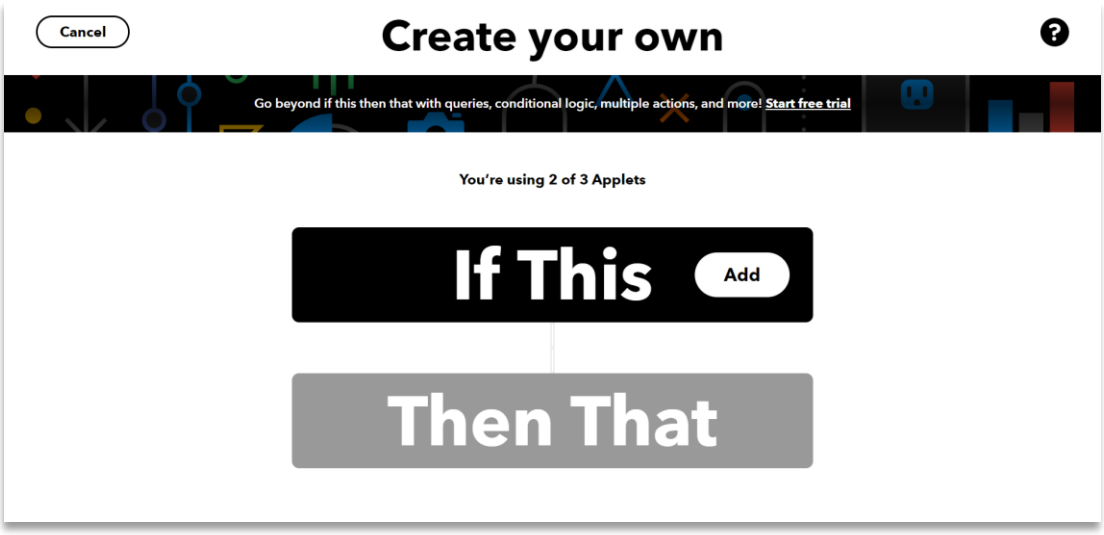

Fig 5.28 Creating Applet on IFTTT

Step 3: Now we have to add *Google Assistant* at *If This* and *Adafruit* at *Then That*. As soon as we tap to Add Google Assistant on If This case it will ask to connect service Google Assistant.

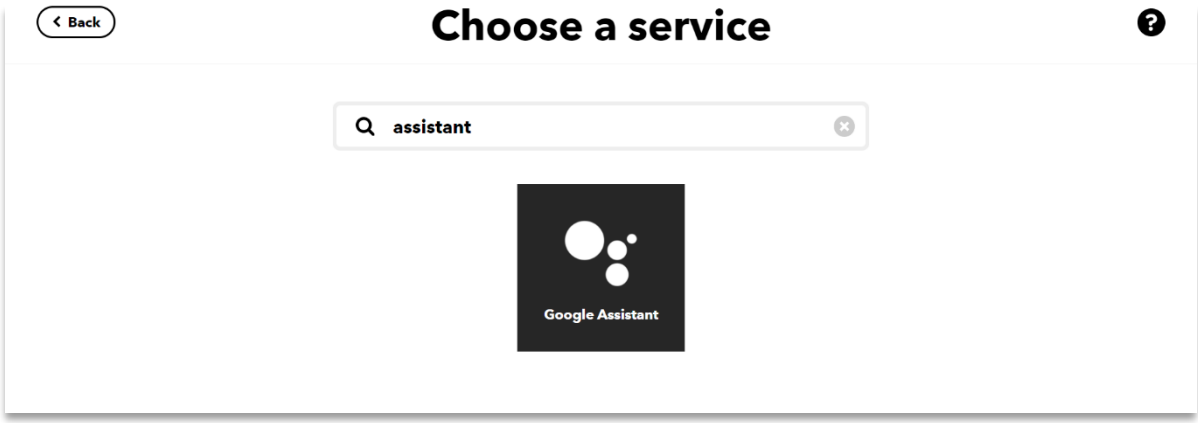

Fig 5.29 IFTTT Choosing First Service- Google Assistant

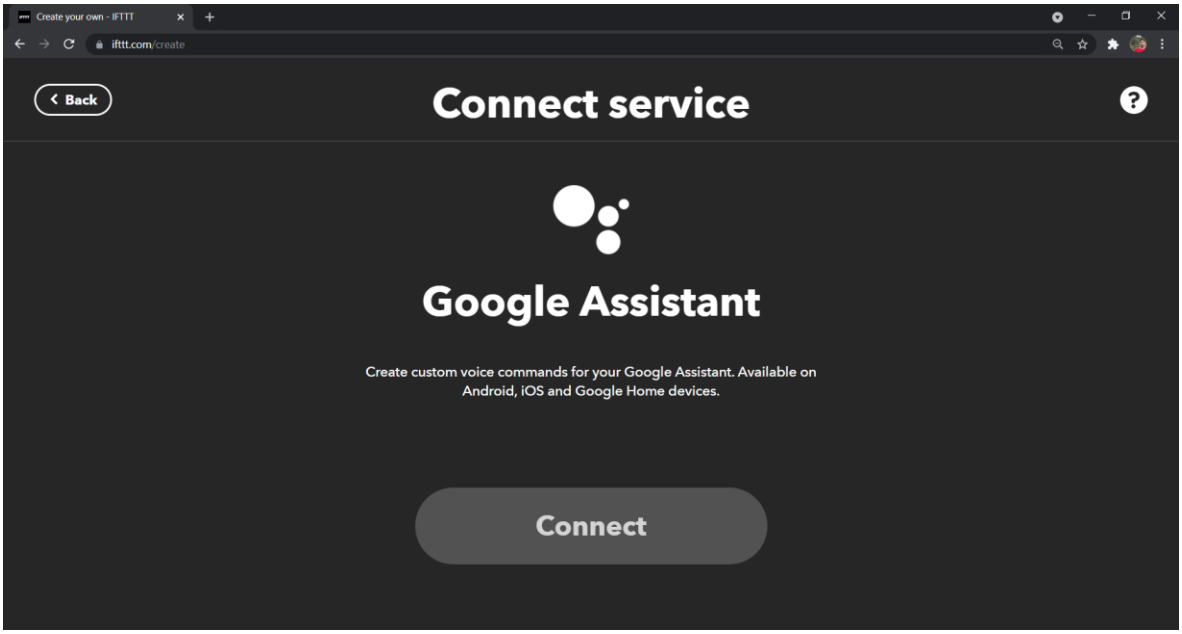

Fig 5.30 Connecting Google Assistant

Step 4: Connecting Google assistant will lead to a pop-up window to verify email again. We've to verify the email.

Step 5: Now we have successfully connected our Google Assistant to IFTTT platform.

Step 6: When we added Google Assistant it asks to select a phrase that we'll say to give command to Google Assistant. We have to select first trigger "Say a Simple Phrase".

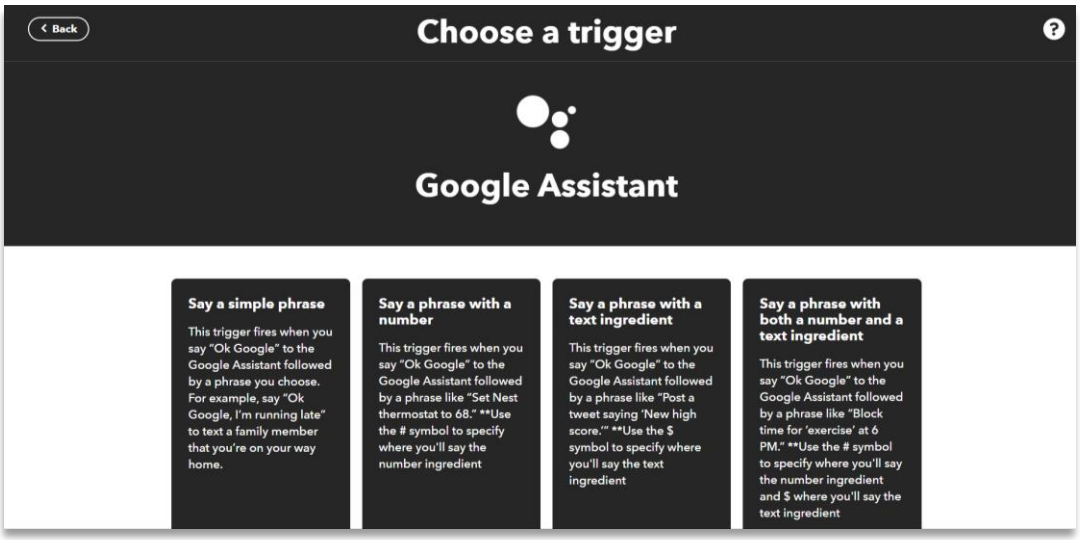

Fig 5.31 Choosing a Trigger

Step 7: Now we have to complete the Trigger field with What we want to say, what's other way of saying and what Google Assistant will say in reply. We've to write all in the trigger field.

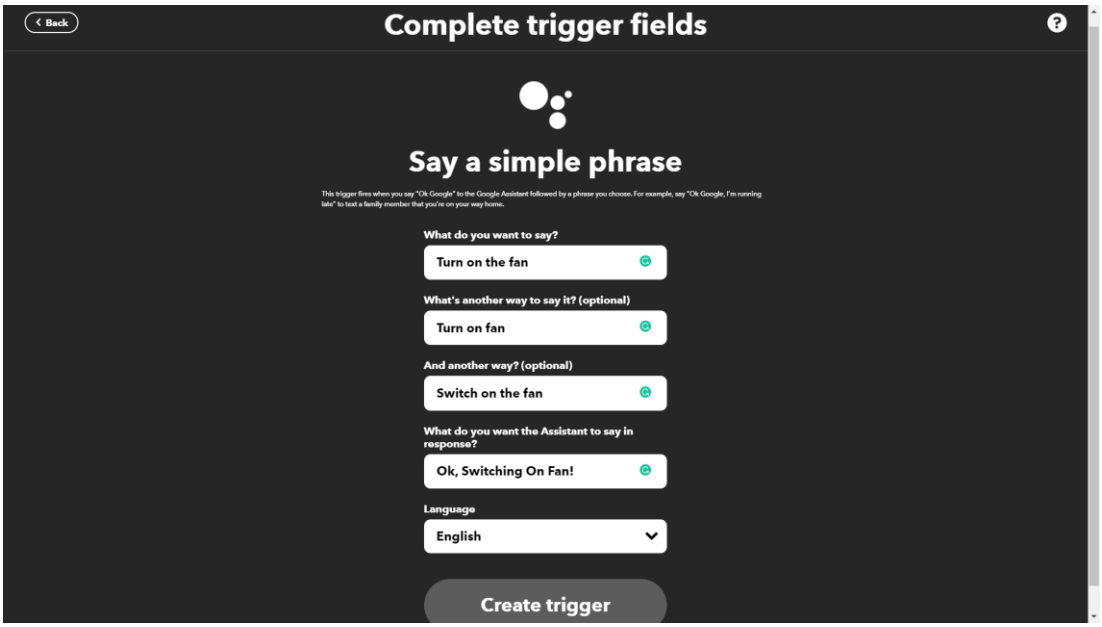

Fig 5.32 Completing Trigger Field

Step 8: Now we've successfully added Google Assistant Trigger to the IFTTT account. Now next step is to add Adafruit in similar way to IFTTT in Then That option.

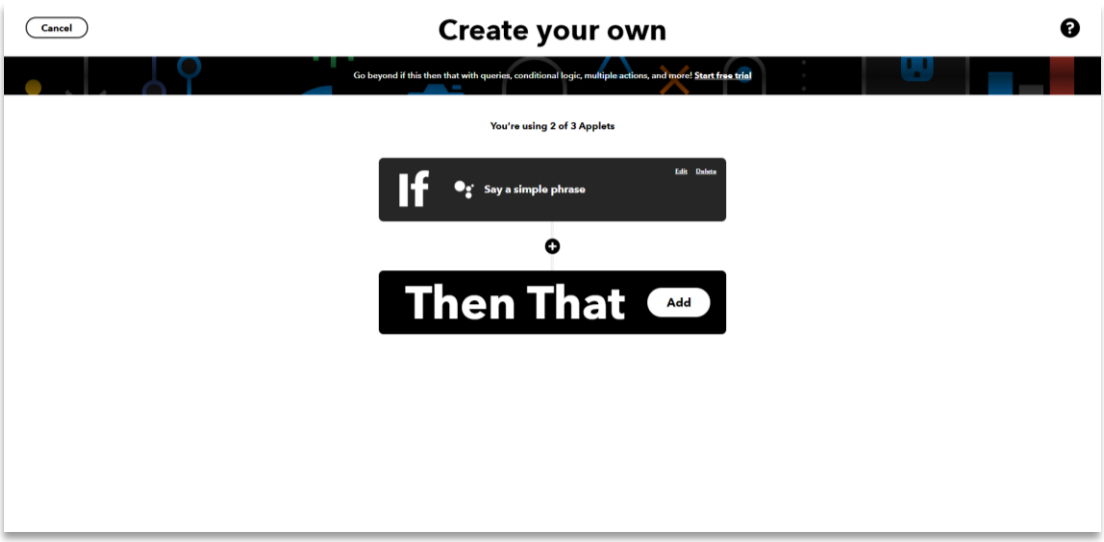

Fig 5.33 Choosing second Trigger

Step 9: By clicking on Add we get an option to choose a service. Service we select in Then that case will be Adafruit.

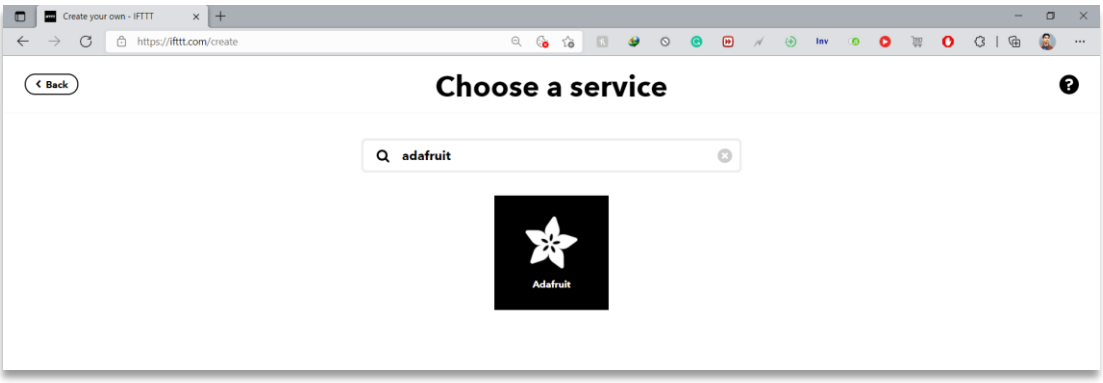

Fig 5.34 Choosing second Trigger as Adafruit.io

Step 10: As soon as we select service it will ask to connect the service (Adafruit) by clicking on connect Button.

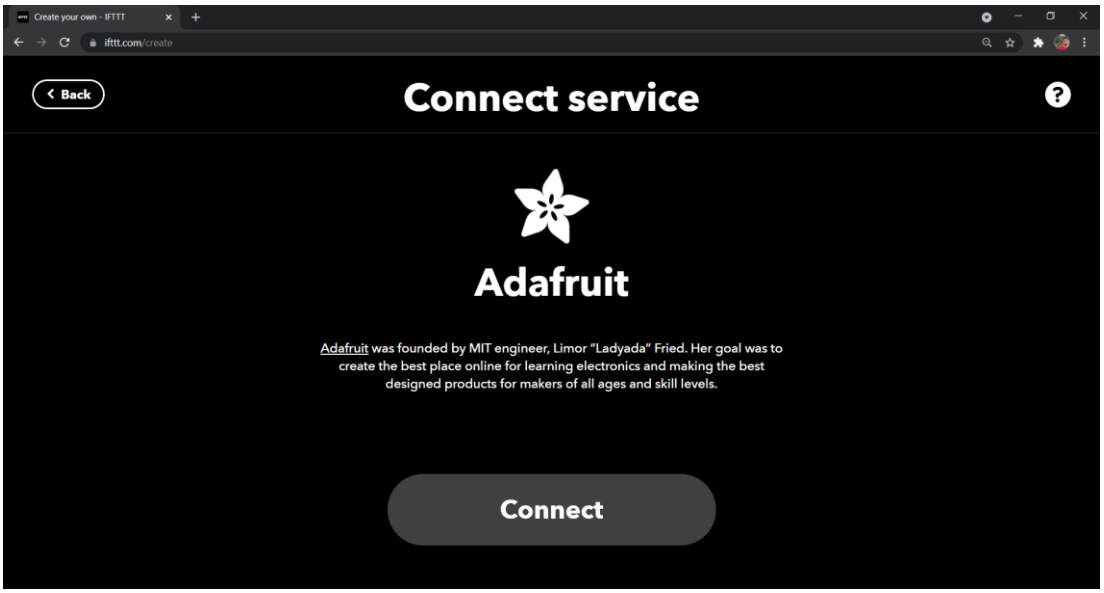

Fig 5.35 Connecting second service Adafruit

Step 11: Clicking on connect button it will pop up another window that'll give option to login at Adafruit.io account.

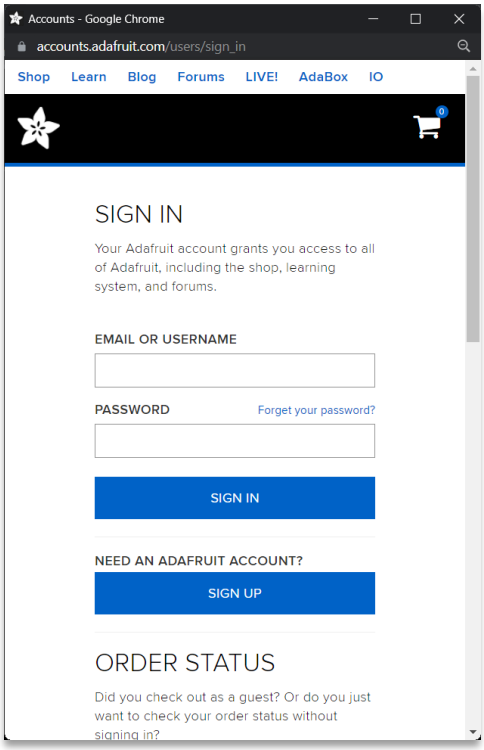

Fig 5.36 Authentication of Adafruit in IFTTT

Step 12: Then we choose an action i.e., send data to Adafruit IO.

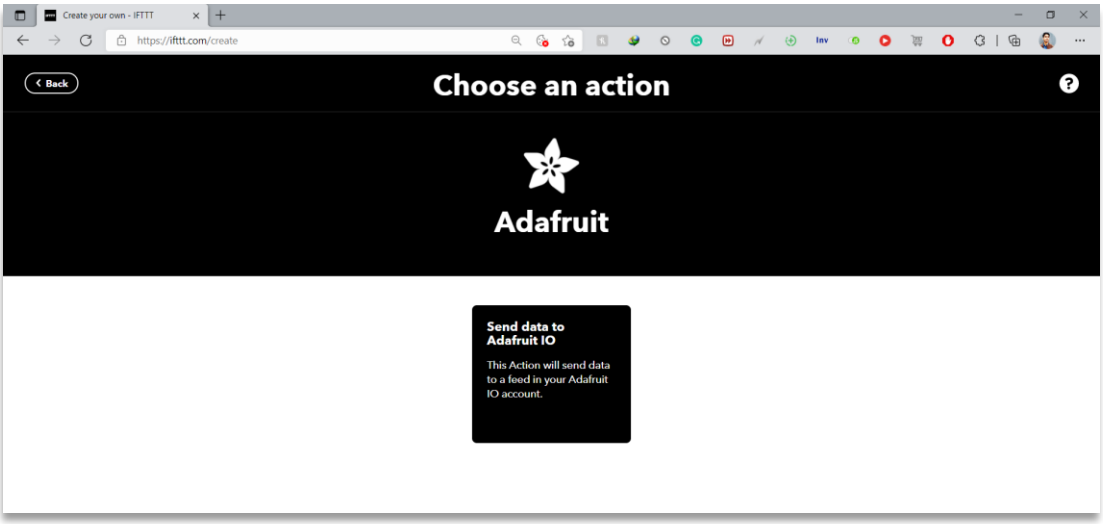

Fig 5.37 Choosing Action of Adafruit on IFTTT

Step 13: Now we have to complete action fields. We have to select Feed Name and Data to save on Feed. (0- ON, 1- OFF)

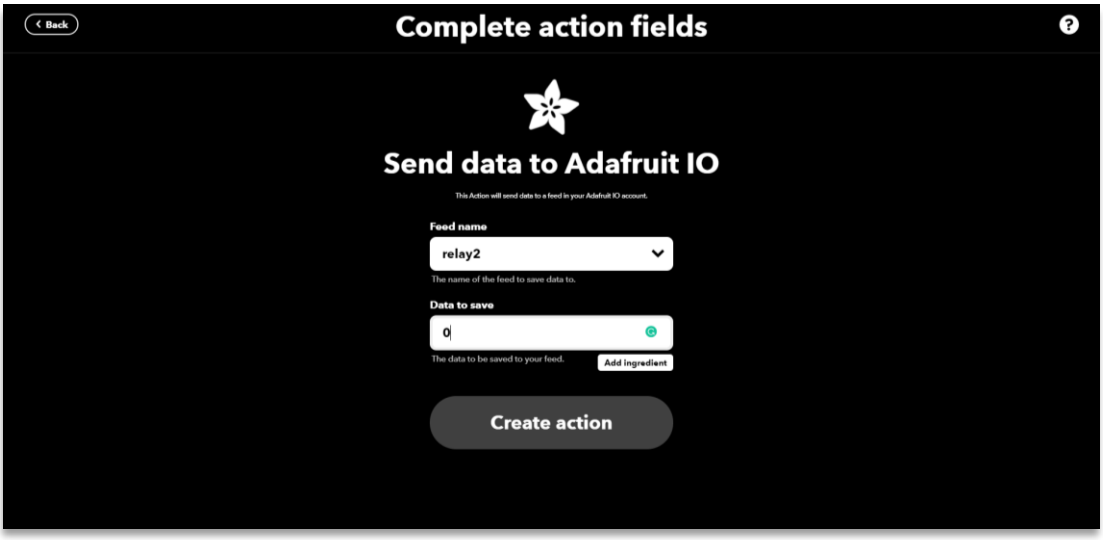

Fig 5.38 Completing Adafruit action

Step 14: When both services added successfully, we have to click continue button to move forward.

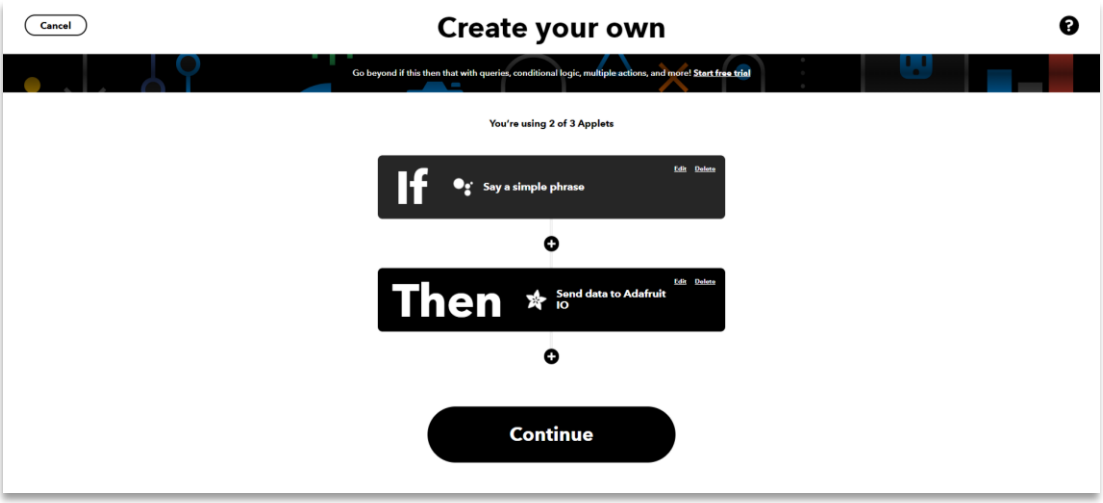

Fig 5.39 Both services added in Applet

Step 15: We have to Review all the applet settings and then click on Finish Button.

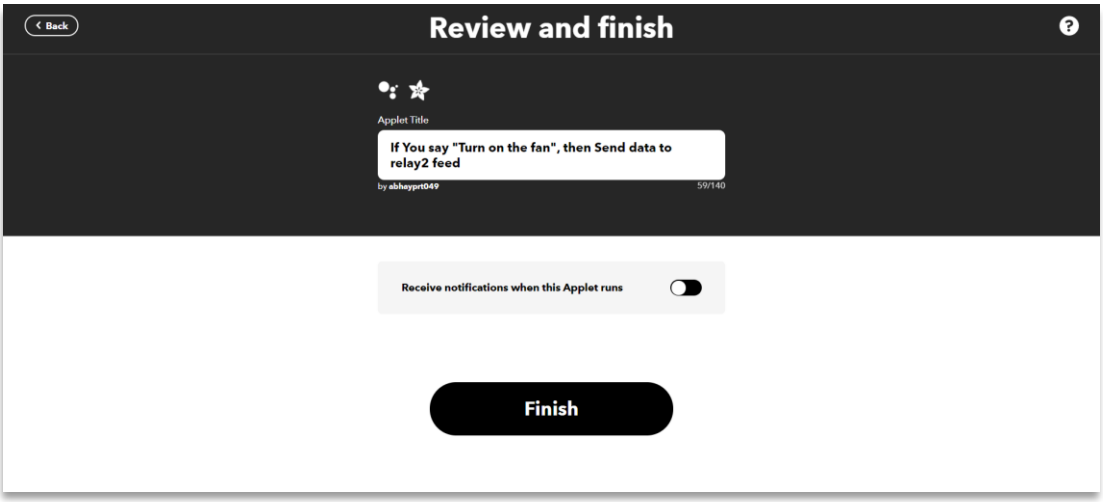

Fig 5.40 Review and Finish applet

Step 16: On finishing, we have successfully crated and connected two services (Google Assistant and Adafruit.io on IFTTT)

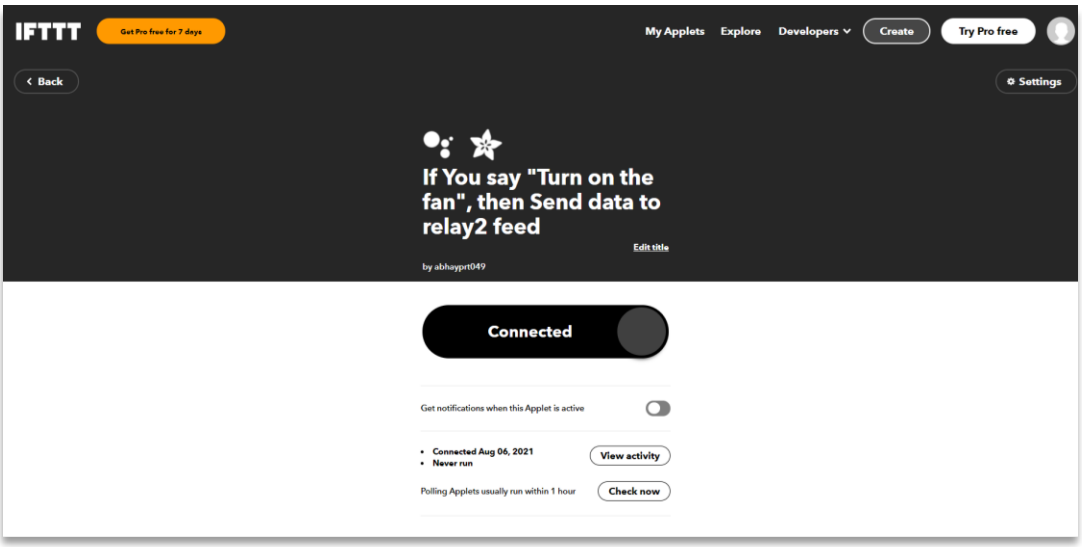

Fig 5.41 Both services connected on IFTTT

Step 17: Similarly, we'll create Toggle (ON/OFF) applet for each relay. For 3 appliances there will be a total 6 applets. But in free version of IFTTT only we can create 03 applets.

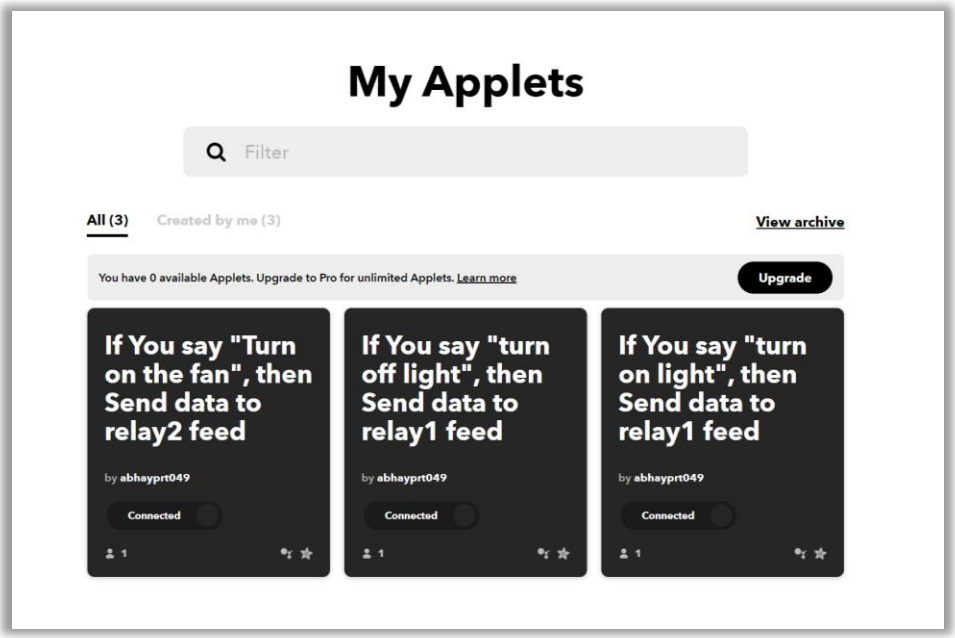

Fig 5.42 Multiple Applets for Simulation

CHAPTER 6

LIBRARIES, ONLINE PLATFORM & SOFTWARE

Libraries used:

1. Arduino UNO R3 Library
2. Blynk Library
3. ESP8266 Wi-Fi Library
4. Adafruit Library

Software, App & Online Platforms Used:

Software:

1. Arduino IDE

2. Proteus 8 Professional

3.VSPE (Virtual Serial Port Emulator)

Online Platform-

1. Adafruit.io

2. IFTTT

Android App-

1. Blynk

Chapter 7

CONCLUSION & FUTURE SCOPE

Conclusion

In this report, we have successfully done simulation of home automation system using Arduino UNO R3 and Internet of Things technology. It has been experimentally proven to work satisfied by connecting sample appliances to it and the application were successfully controlled from a wireless android mobile device using app. Also, with help of Google Assistant it can be controlled and its successful simulation is shown on Adafruit. The automated mode makes life easier for users by complete automation of necessary appliances without any human effort.

Future Scope

Future scope for the home automation systems involves making homes even smarter. Homes can be interfaced with sensors including motion sensors, light sensors and temperature sensors and provide automated toggling of devices based on conditions. More energy can be conserved by ensuring occupation of the house before turning on devices and checking brightness and turning off lights if not necessary. The system can be integrated closely with home security solutions to allow greater control and safety for home owners. The next step would be to extend this system to automate a large-scale environment, such as offices and factories. Home Automation offers a global standard for interoperable products. Standardization enables smart homes that can control appliances, lighting, environment, energy management and security as well as the expandability to connect with other networks.

Reference

[1] Home Automation Application Based On Arduino Controllable From Mobile by Anna Merino Herreros

[2] Adafruit Documentation - *https://io.adafruit.com/api/docs/ - adafruit-io-http-api*

[3] IFTTT Documentation- *https://platform.ifttt.com/docs*

[4] Home automation – Wikipedia - *https://en.wikipedia.org/wiki/Home_automation*

[5] Blynk Documentation- *https://docs.blynk.cc*

[6] Pololu - Arduino Uno R3 - *https://www.pololu.com/product/2191*

[7] VSPE- *http://www.eterlogic.com/Products.VSPE.html*