A Report on

“Home Automation Using Google-Assistant”

A technical project work submitted in partial fulfilment of requirement for the award of the degree of

Bachelor of Technology

In

Computer Science and Engineering

By

TEJASWINI G J	20181CSE0746
TANGIRALA RAMA NARAYANA REDDY	20181CSE0743
V MEGHANA	20181CSE0760
TATIKONDA SANDEEP	20181CSE0748
UPPATALA BHAVYA KRISHNA	20181CSE0759

Under the guidance of

Manasa C M

Assistant Professor,Dept. of CSE

Table of Content:

Aim: To build Home Automation for controlling the home appliances using Google-Assistant .

Components:

Hardware Used

- Node MCU – 32-bit ESP8266 development board with Wi-Fi Soc.
- Relay module 2 channel
- Bulb
- Jumper wires
- Electric wires
- Micro USB cable
- Power plug

To build a home automation application, I used 3 different platforms

- Google Assistant
- Adafruit
- IFTTT

Abstract:

- “Home automation” refers to the automatic and electronic control of household features, activities, and appliances. The utilities and features of our home can be easily controlled via Internet.
- By using IoT, we are successful in controlling the appliances by using Node Microcontroller. We can also use other boards like raspberry pi, beagle bone etc.
- Adafruit account which is a cloud based free IoT web server used to create virtual switches, is linking to IFTTT website abbreviated as “If This Than That” which is used to create if else conditional statements. The voice commands for Google assistant have been added through IFTTT website.
- The commands given through the Google assistant are decoded and then sent to the microcontroller, the microcontroller in turn control the relays connected to it. The device connected to the respective relay can be turned On or OFF as per the users request to the Google Assistant.

Introduction:

Google assistant is AI (Artificial Intelligence) based voice command service. Using voice, we can interact with the google assistant and it can search on the internet, schedule events, set alarms, control appliances, etc.

This service is available on smartphones and Google Home devices.

We can control smart home devices including lights, switches, fans, and thermostats using our Google Assistant.

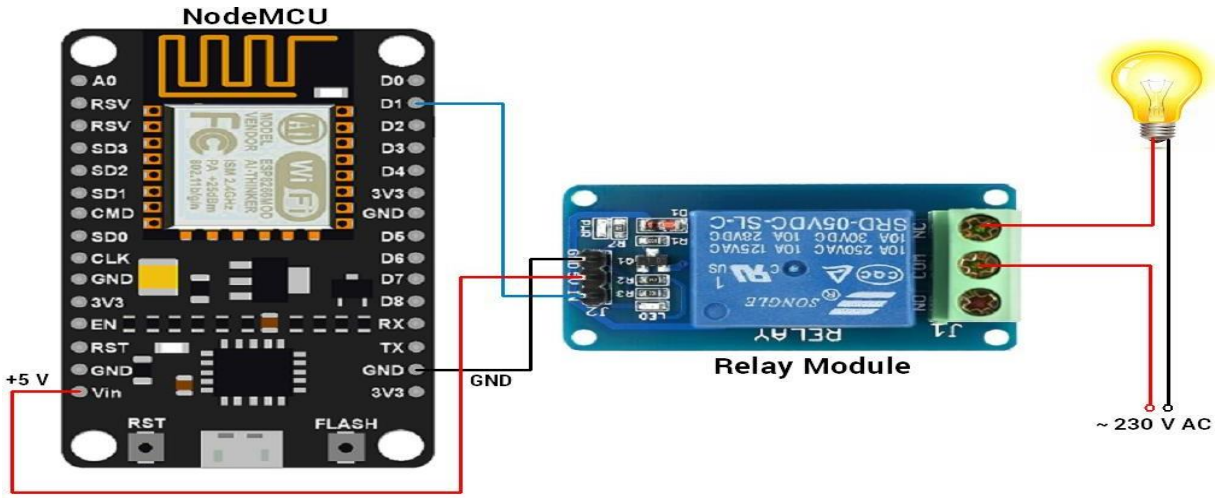
We will build an application that can control home appliances. Here, we will control a bulb using Google Assistant service.

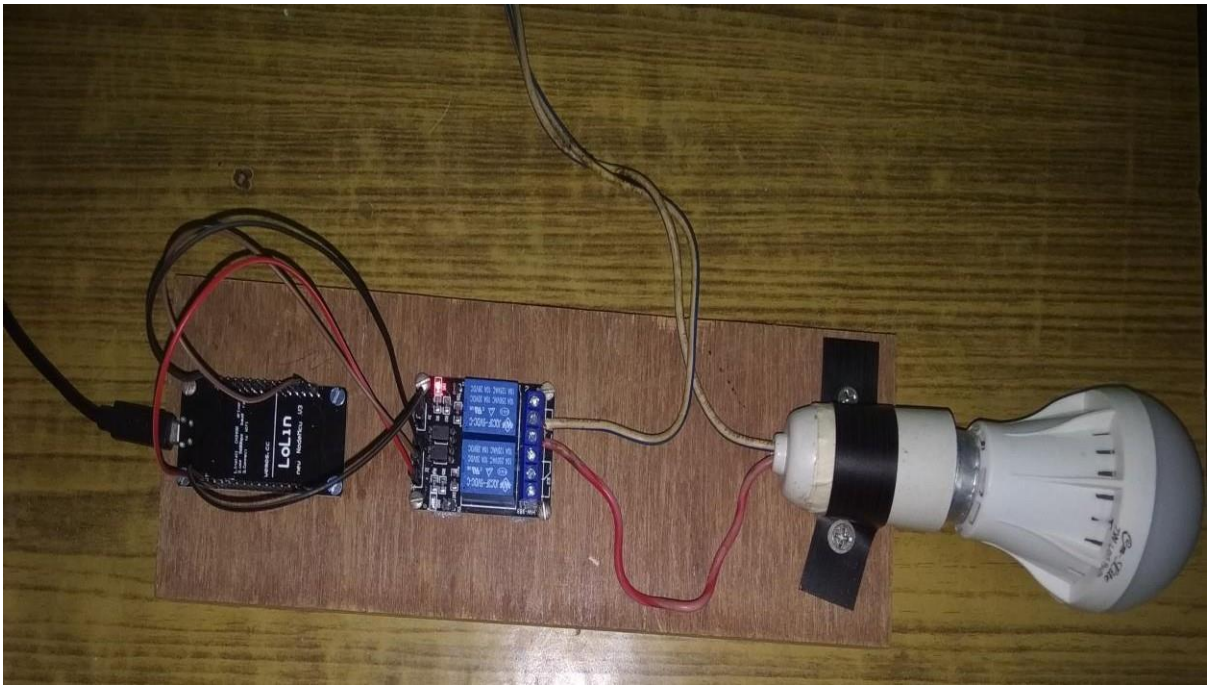
This application includes Google assistant along with Adafruit server and IFTTT service.

Architecture:

- In Google assistant-controlled home automation, first the user should have an Android smartphone with Google assistant installed in it.
- When the user gives commands to the Google assistant, the commands will be checked with the commands in the IFTTT website which are already set.
- Then the next step is setting up the virtual switches in Adafruit website. If the commands given by the user matches with the commands in the IFTTT website, then depending on that commands, the virtual switches in Adafruit will be turned ON or OFF.
- This will be sensed by the Node microcontroller and it will turn ON or OFF the relay depending on the commands.
- All this will be done over the Internet. In this, the relay will act as a switch and the Home appliances connected to the relay will be turned on or off. The number of Home appliances connected depends upon the number of relays.

Components Connection:





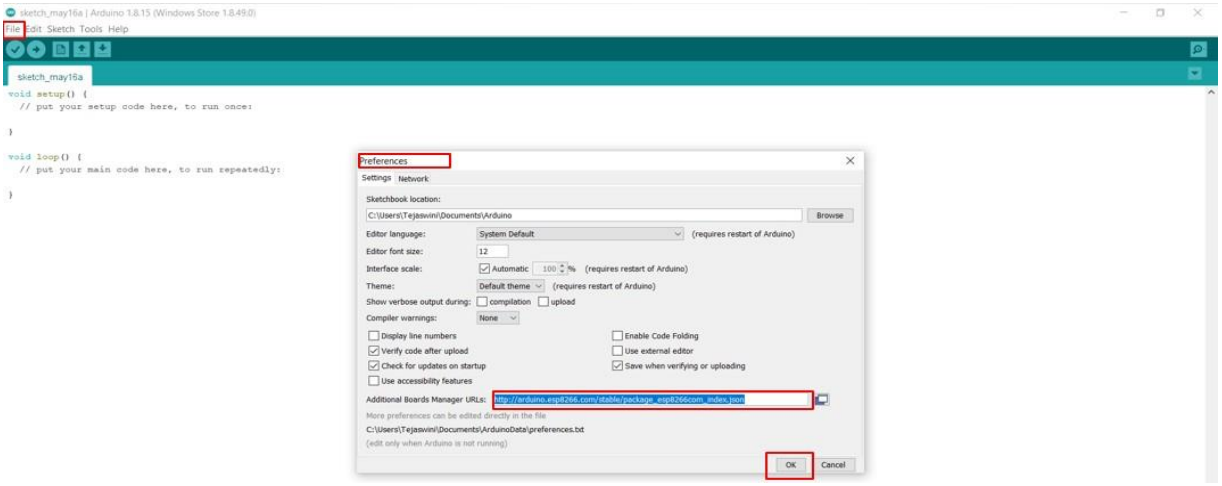
From Node MCU	To Relay 1
D1	IN1
G	GND
VIN	VCC

Connect power plug and electric wire then connect to relay module

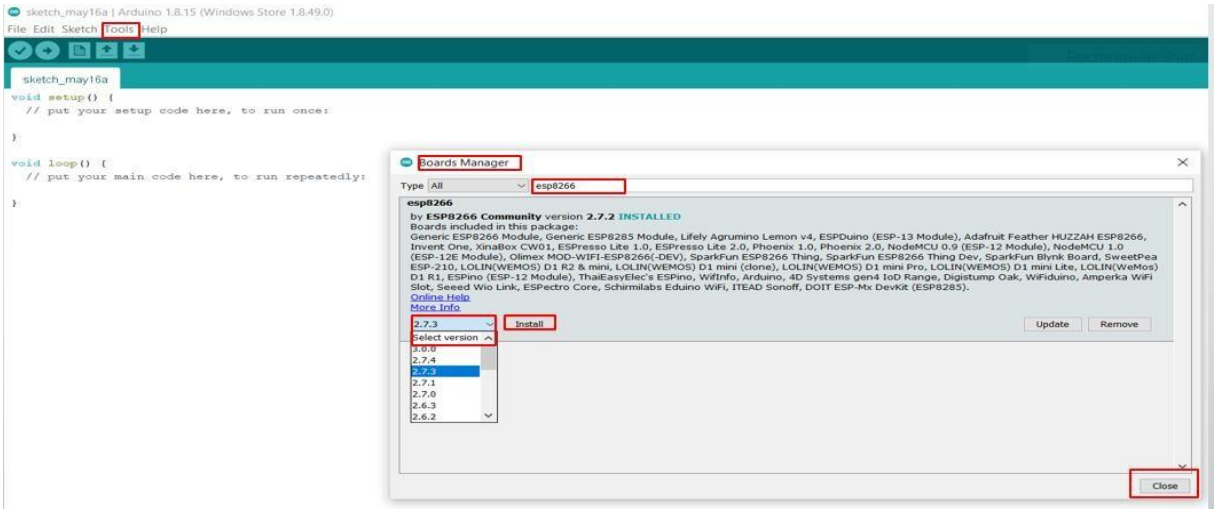
Connect Node MCU through Laptop Via USB cable

Software Setting up:

Download Arduino ide from the Microsoft store or from google
File--→Preference--→Paste the link--→click OK
Link: http://arduino.esp8266.com/stable/package_esp8266com_index.json

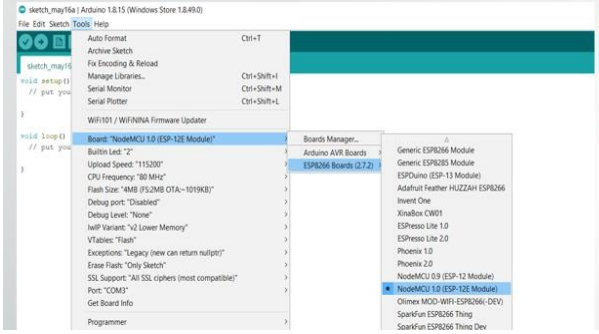


Tools→Board→Board Manager→Search for esp8266→Select Version then Install→Close

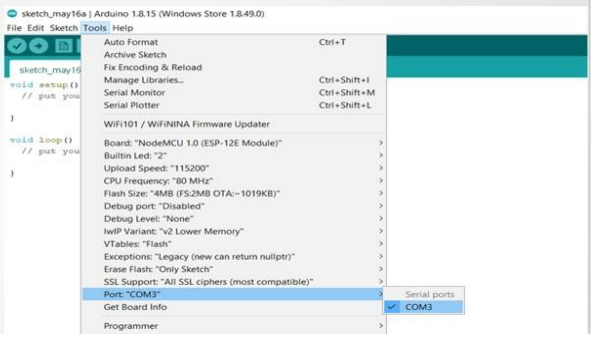


Set up these Information Required for our Project

Tool--→Board--→ESP8266(2.7.2)--→Node MCU 1.0(ESP-12E Module)

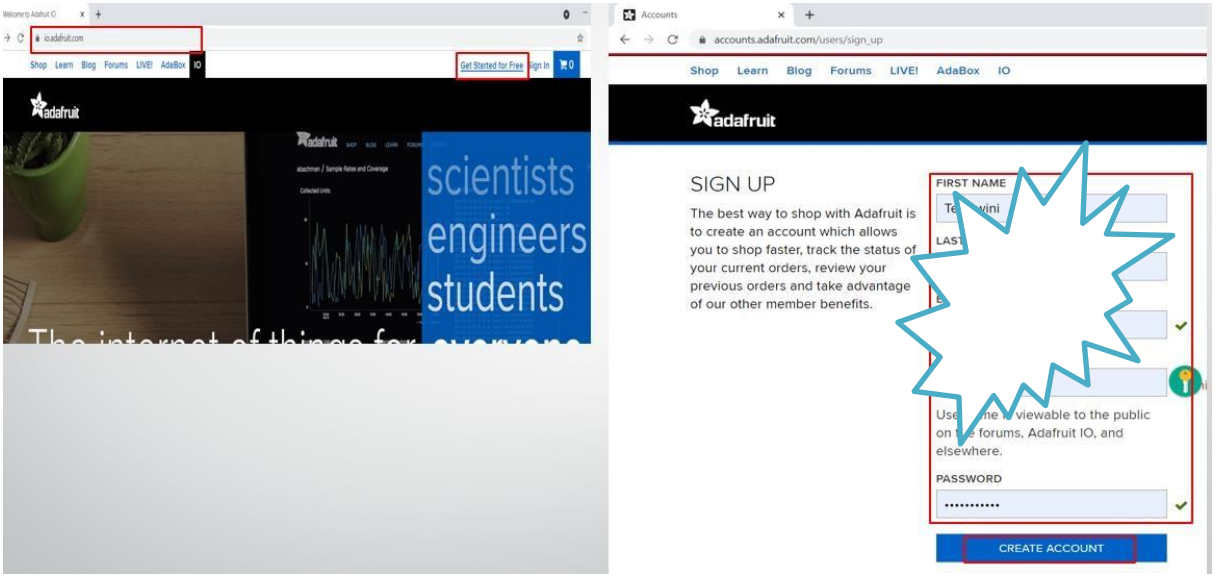


Tools--→Port--→COM3

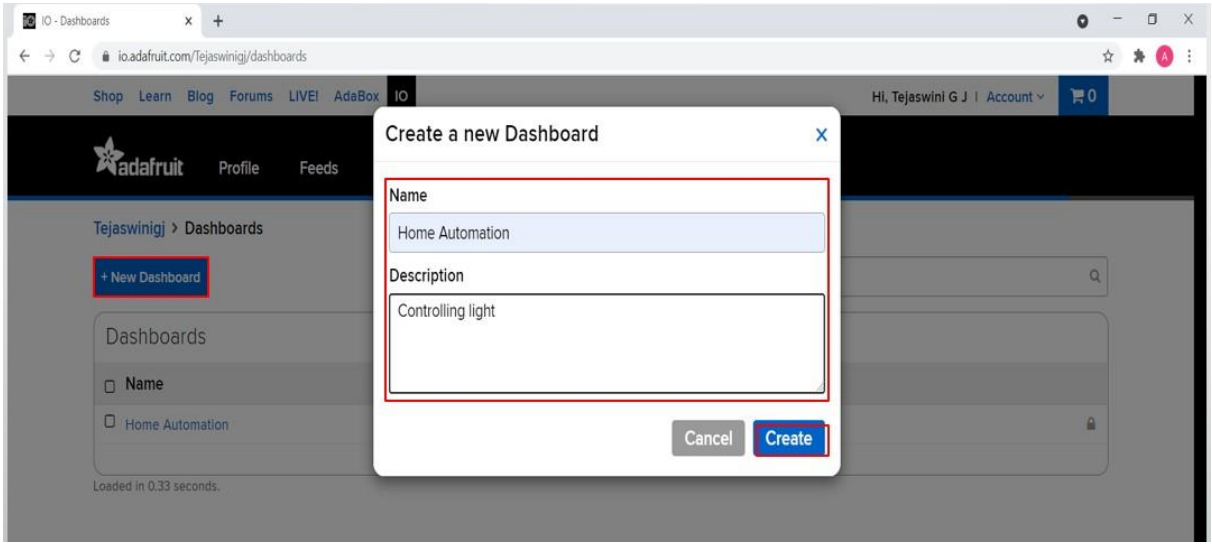


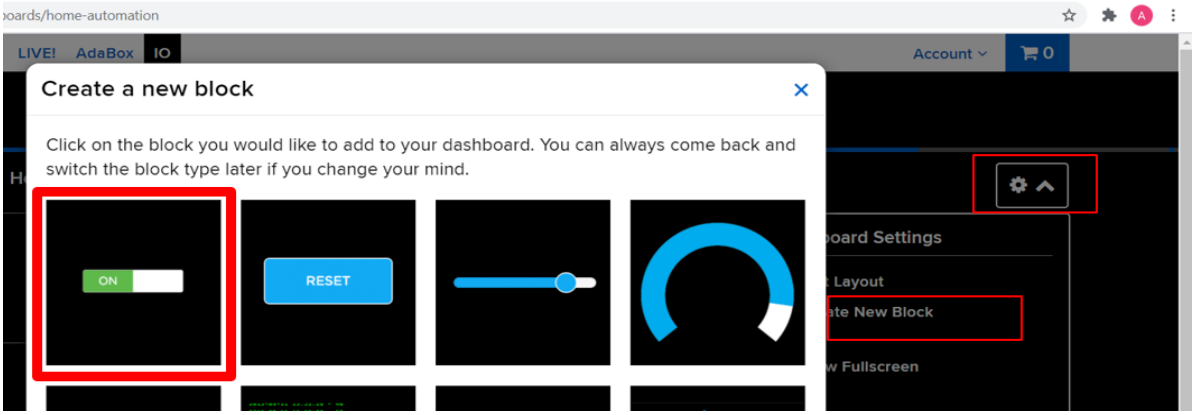
Accessing Adafruit Account:

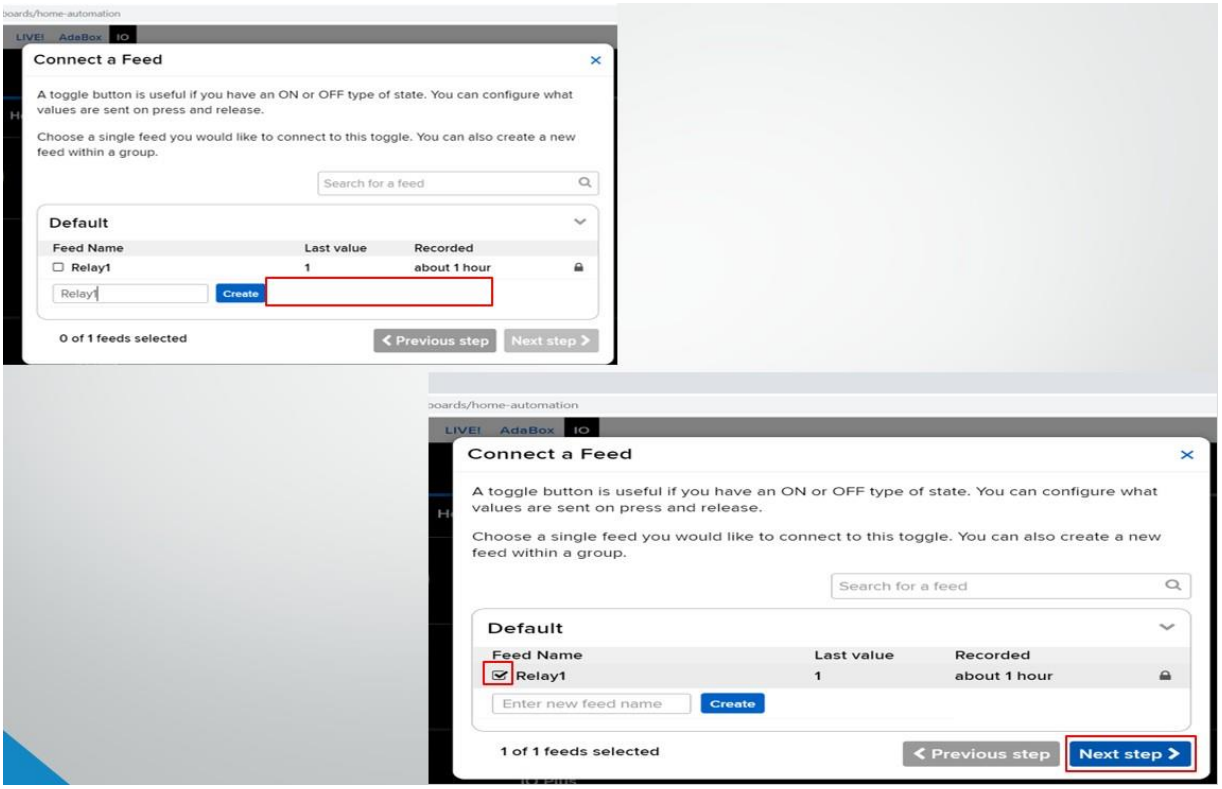
Web Browser--→Adafruit.io

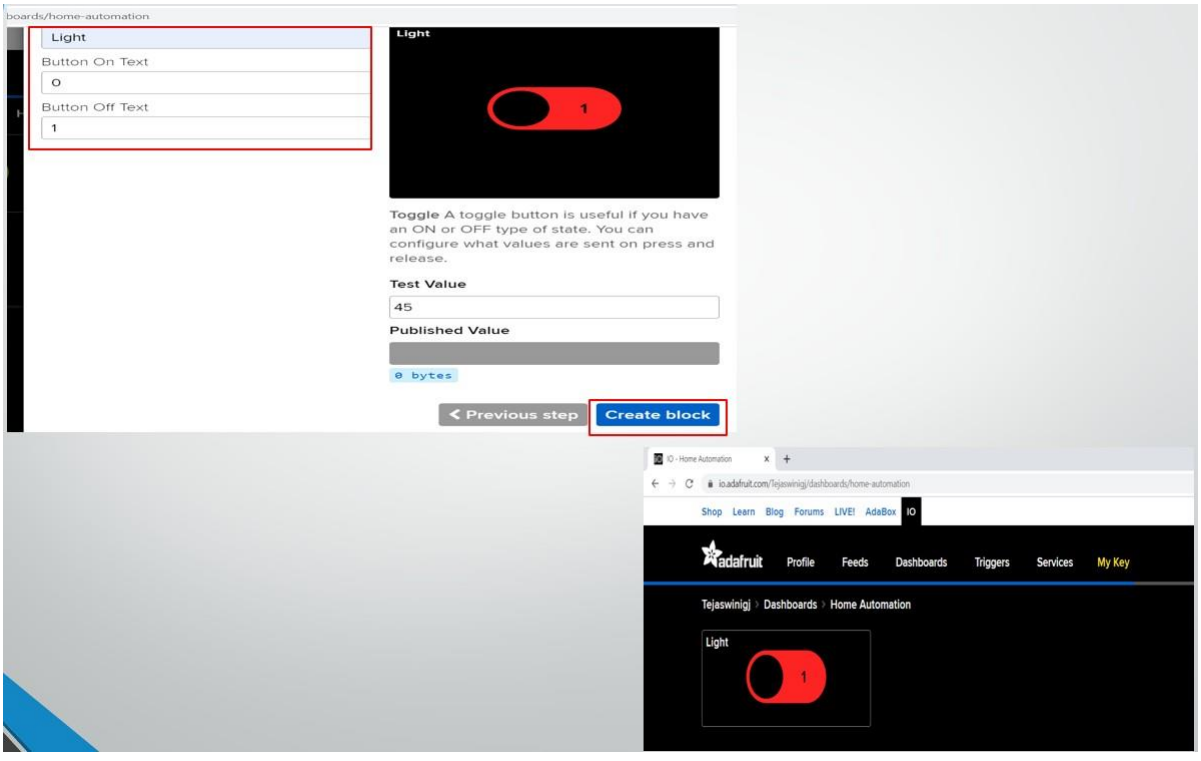


After creating adafruit account close the tab and open again. Then create New Dashboard



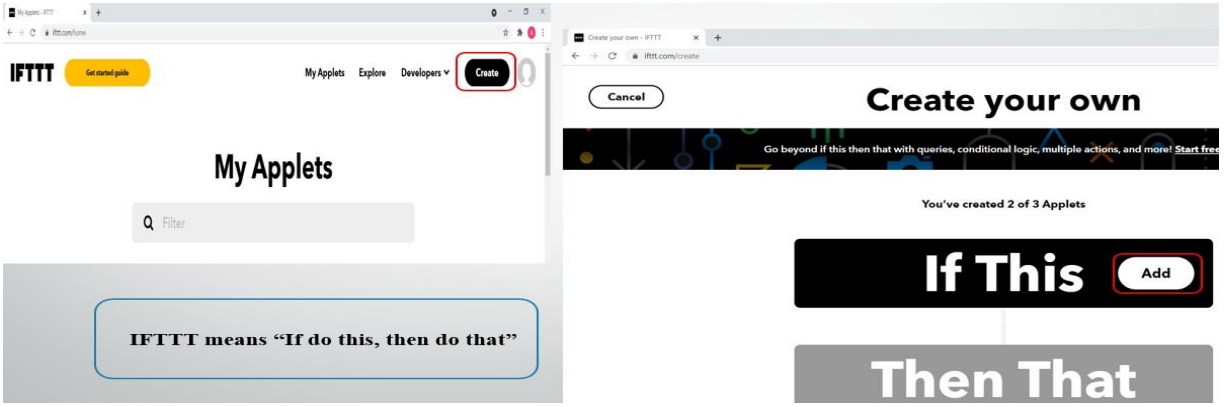


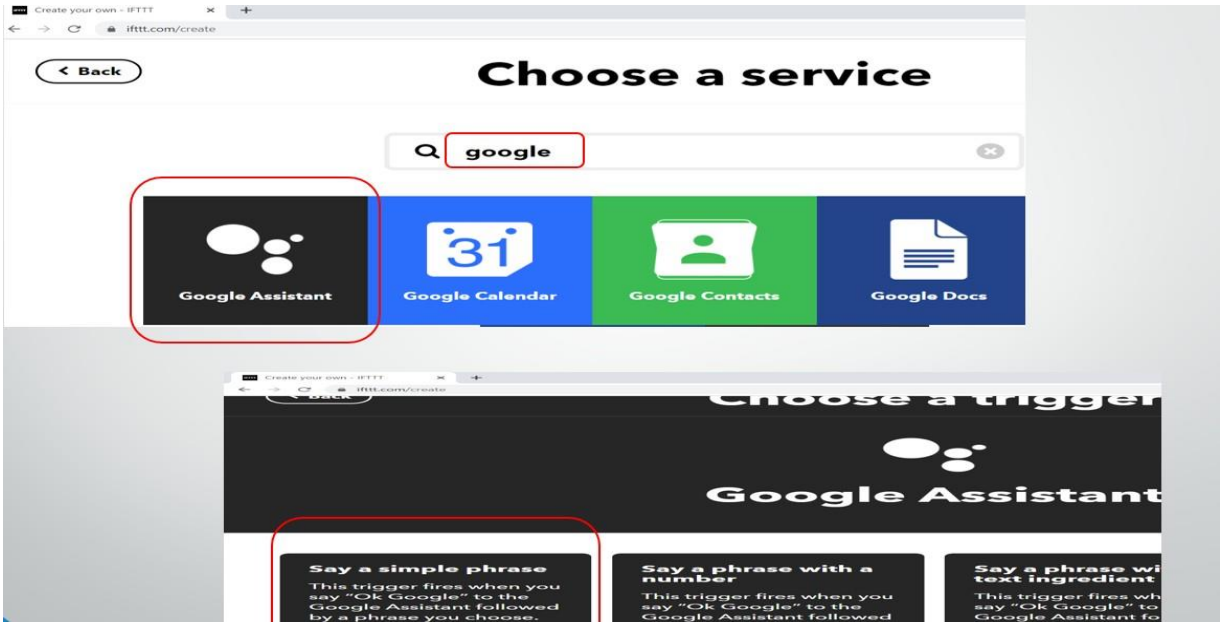

Connect a Feed
A toggle button is useful if you have an ON or OFF type of state. You can configure what values are sent on press and release.
Choose a single feed you would like to connect to this toggle. You can also create a new feed within a group.
Search for a feed
Default
Feed Name
Last value
Recorded
Relay1
1
about 1 hour
Relay1
Create
0 of 1 feeds selected
Previous step
Next step
Connect a Feed
A toggle button is useful if you have an ON or OFF type of state. You can configure what values are sent on press and release.
Choose a single feed you would like to connect to this toggle. You can also create a new feed within a group.
Search for a feed
Default
Feed Name
Last value
Recorded
Relay1
1
about 1 hour
Enter new feed name
Create
1 of 1 feeds selected
Previous step
Next step

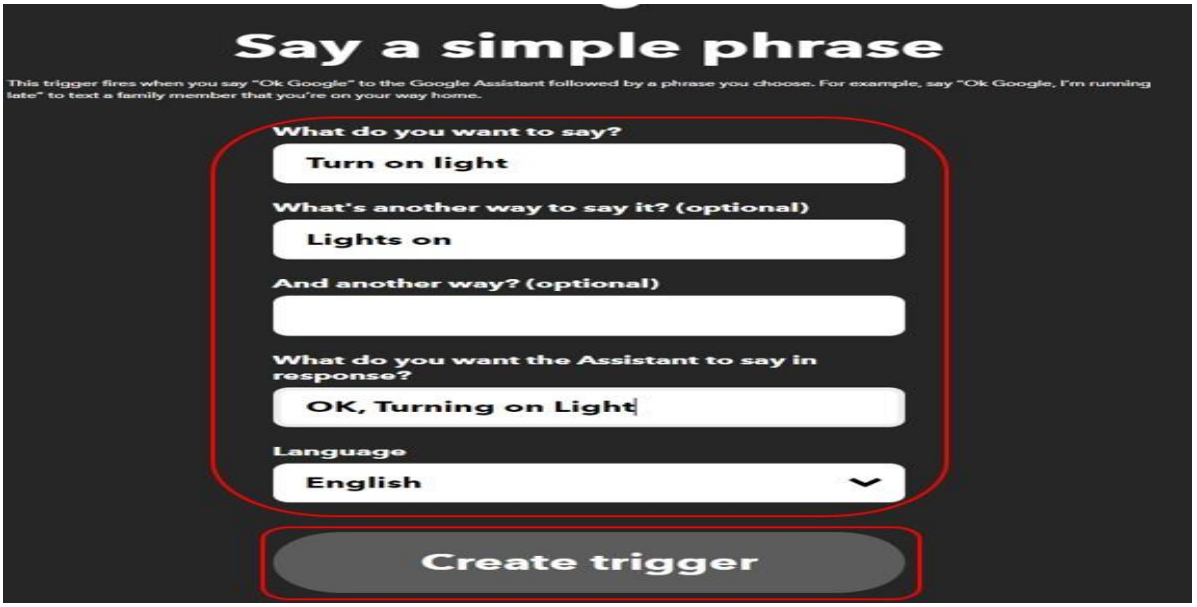

Light
Button On Text
0
Button Off Text
1
Light
1
Toggle A toggle button is useful if you have an ON or OFF type of state. You can configure what values are sent on press and release.
Test Value
45
Published Value
0 bytes
Previous step
Create block
Shop
Learn
Blog
Forums
LIVE!
AdaBox
IO
adafruit
Profile
Feeds
Dashboards
Triggers
Services
My Key
Tejaswinigj
Dashboards
Home Automation
Light
1

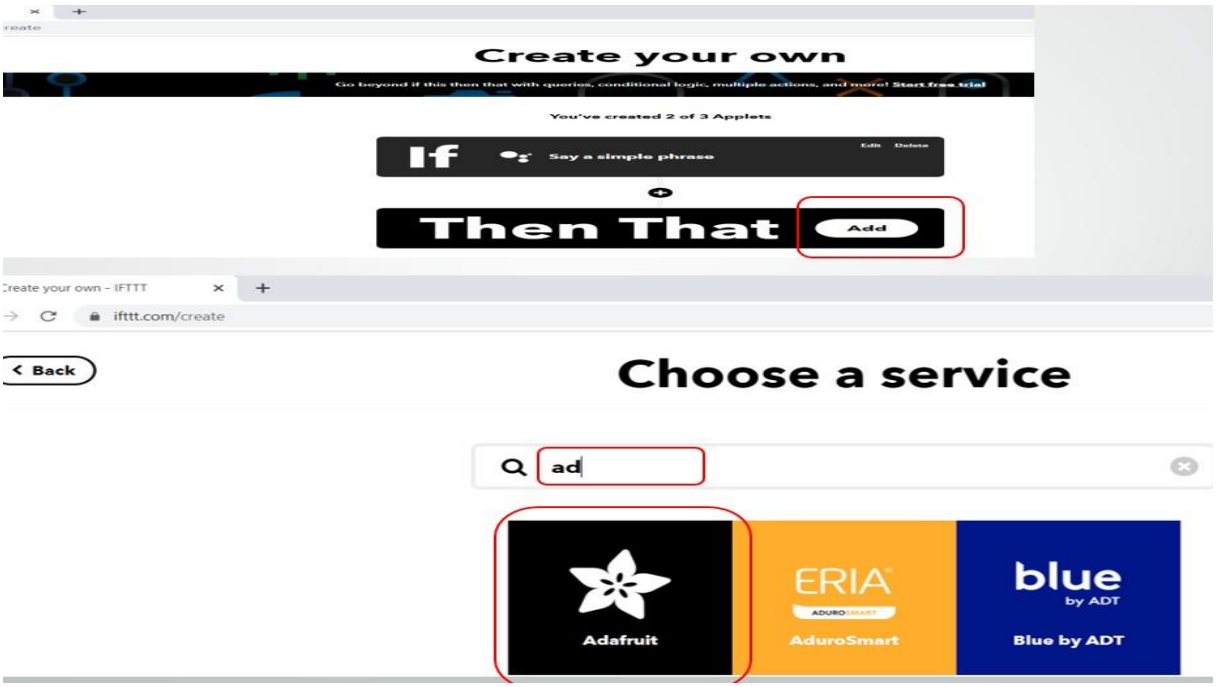
Configure IFTT:

Web Browser--→Create your IFTTT Account--→

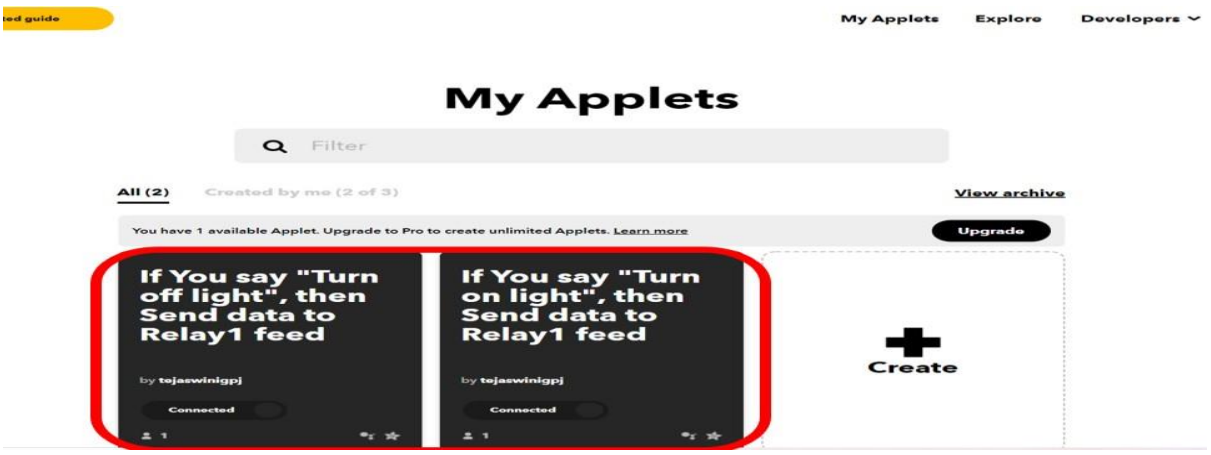







Create your own
Go beyond if this then that with queries, conditional logic, multiple actions, and more! Start free trial
You've created 2 of 3 Applets
If
Say a simple phrase
Edit Delete
Then That
Add
Create your own - IFTTT
ifttt.com/create
Back
Choose a service
ad
Adafruit
ERIA
AduroSmart
blue
by ADT
Blue by ADT


Complete action fields
Send data to Adafruit IO
This Action will send data to a feed in your Adafruit IO account.
Feed name
Relay1
The name of the feed to save data to.
Data to save
0
The data to be saved to your feed.
Add ingredient
Create action


My Applets
Explore
Developers
My Applets
Filter
All (2)
Created by me (2 of 3)
View archive
You have 1 available Applet. Upgrade to Pro to create unlimited Applets. Learn more
Upgrade
If You say "Turn off light", then Send data to Relay1 feed
by tejaswinigpj
Connected
If You say "Turn on light", then Send data to Relay1 feed
by tejaswinigpj
Connected
Create

Source Code:

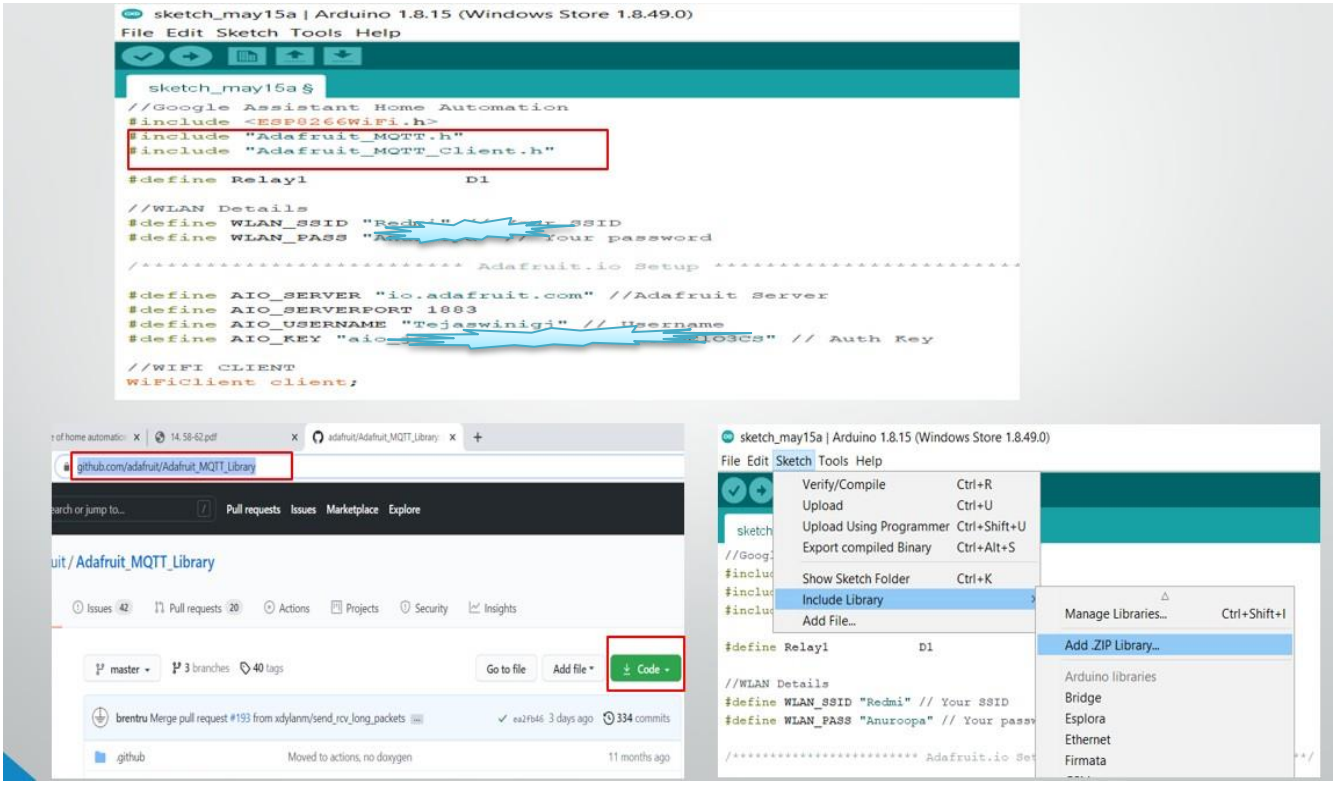
```
//Google Assistant Home Automation

#include <ESP8266WiFi.h>

#include "Adafruit_MQTT.h"

#include "Adafruit_MQTT_Client.h"

#define Relay1            D1



//WLAN Details

#define WLAN_SSID “-----" // Your SSID #define

WLAN_PASS "-------" // Your password



//Adafruit.io Setup

#define AIO_SERVER "io.adafruit.com" //Adafruit Server

#define AIO_SERVERPORT 1883

#define AIO_USERNAME "---------" //Adafruit Username

#define AIO_KEY "----------" // Auth Key

//WIFI CLIENT
WiFiClient client;
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);
Adafruit_MQTT_Subscribe Light = Adafruit_MQTT_Subscribe(&mqtt,
AIO_USERNAME"/feeds/Relay1"); // Feeds name should be same everywhere
void MQTT_connect();
```

```
void setup()
{
        Serial.begin(115200);
        pinMode(Relay1, OUTPUT);



        // Connect to WiFi access point.
        Serial.println(); Serial.println();
        Serial.print("Connecting to ");
        Serial.println(WLAN_SSID);

        WiFi.begin(WLAN_SSID, WLAN_PASS);
        while (WiFi.status() != WL_CONNECTED)
        {
                delay(500);
                        Serial.print(".");

         }

        Serial.println();
        Serial.println("WiFi connected");
        Serial.println("IP address: ");
        Serial.println(WiFi.localIP());

        mqtt.subscribe(&Light);

}

void loop()
{

         MQTT_connect();
        Adafruit_MQTT_Subscribe *subscription;

         while ((subscription = mqtt.readSubscription(20000)))

        {

           if (subscription == &Light)
```

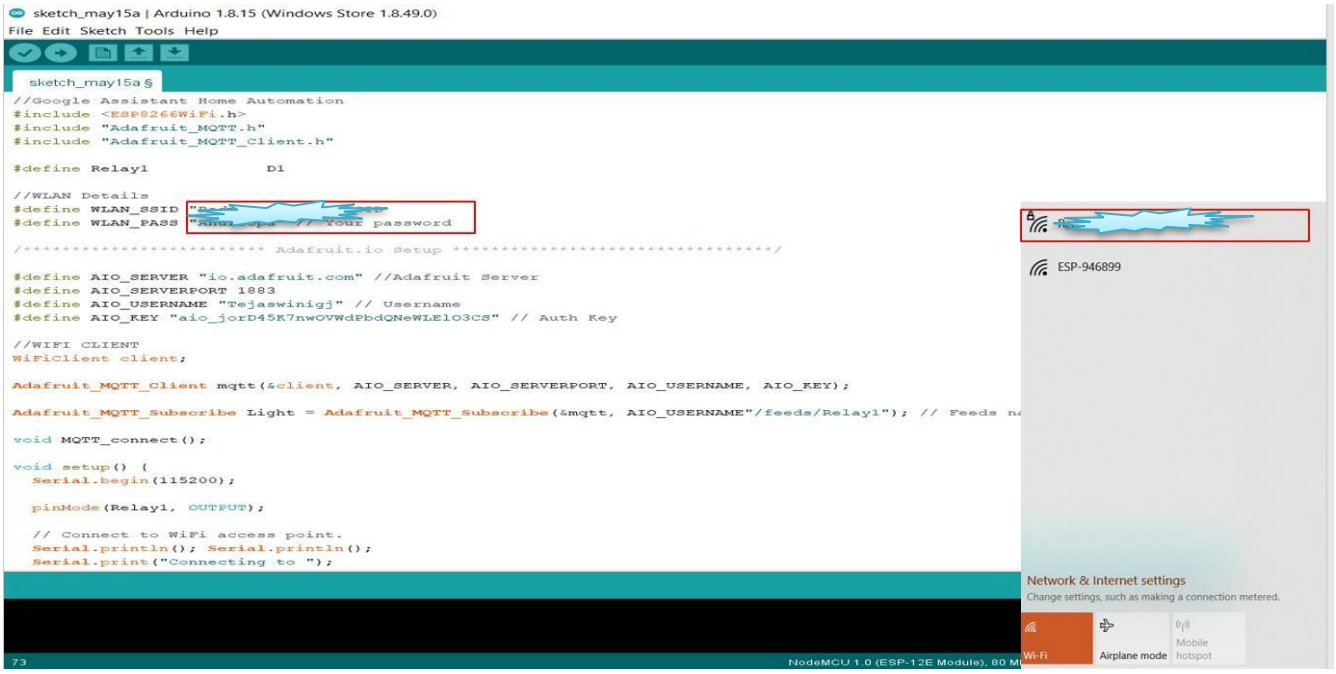
```
        {

                Serial.print(F("Got: "));
                Serial.println((char *)Light.lastread);
                int Light_State = atoi((char *)Light.lastread);
                digitalWrite(Relay1, Light_State);

         }

        }
}

void MQTT_connect()
{

        int8_t ret;
        if (mqtt.connected())
        {
        return;
        }

        Serial.print("Connecting to MQTT... ");
        uint8_t retries = 3;
        while ((ret = mqtt.connect()) != 0)
        {

                Serial.println(mqtt.connectErrorString(ret));
                        Serial.println("Retrying MQTT connection in 5 seconds...");
                mqtt.disconnect();
                        delay(5000);
                        retries--;
                        if (retries == 0)
                        {
                        while (1);
                        }
                }

Serial.println("MQTT Connected!");
}
```

Code Explaination:

Here We used the Adafruit MQTT library for receiving data from the Adafruit server.
To install this library, select option Sketch -> Include Library -> Add .ZIP Library
In that, search for Adafruit MQTT you have just installed and select open.



Modify your wife connection



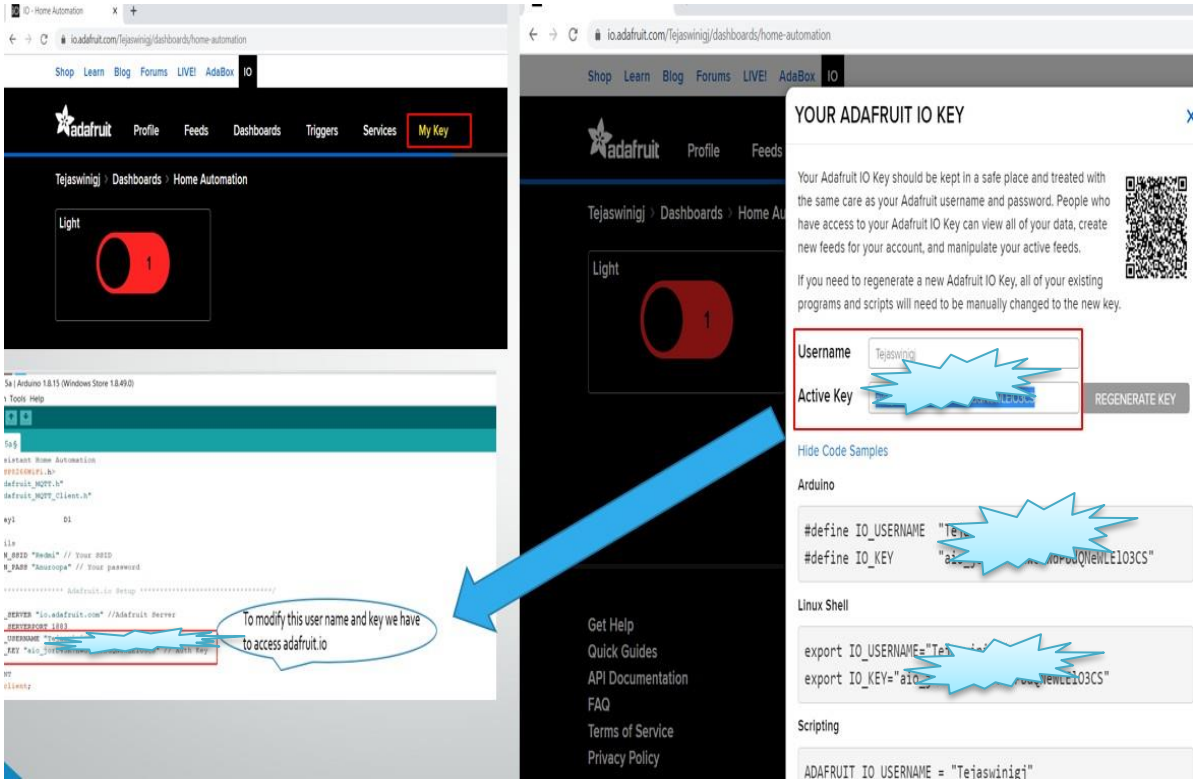
Modify your adafruit account information

sketch_may15a | Arduino 1.8.15 (Windows Store 1.8.49.0)

File Edit Sketch Tools Help

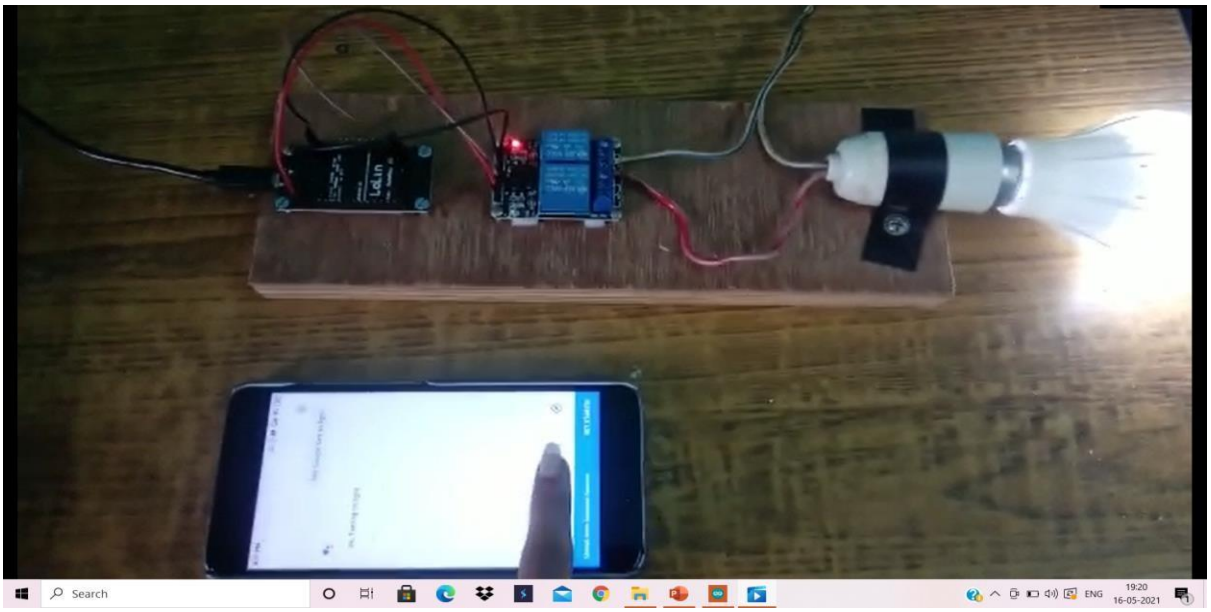
sketch_may15a §

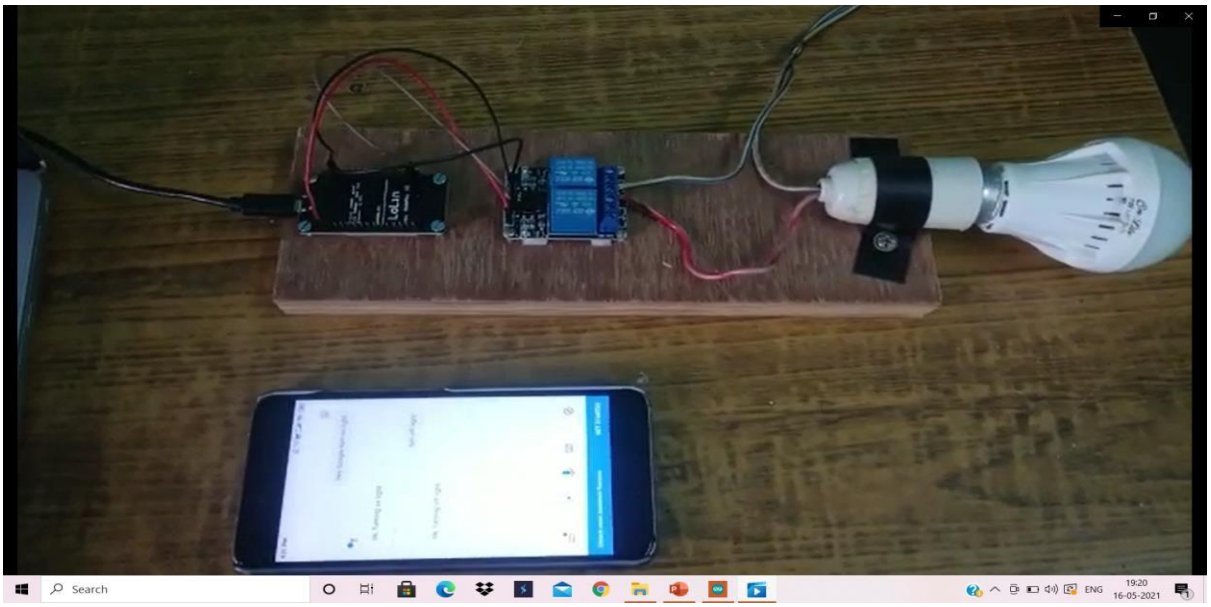
```
//Google Assistant Home Automation
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"

#define Relay1              D1

//WLAN Details
#define WLAN_SSID "Redmi" // Your SSID
#define WLAN_PASS "Anuroopa" // Your password

/************************* Adafruit.io Setup *********************

#define AIO_SERVER "io.adafruit.com" //Adafruit Server
#define AIO_SERVERPORT 1883
#define AIO_USERNAME "Tejaswinigj"
#define AIO_KEY "aio_JoRD[redacted]WLE1O3CS" // Auth Key

//WIFI CLIENT
WiFiClient client;

Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AI
```



Model Screenshot and Output:

Upload the code to the Node MCU then give the command to the google assistant.
The commands given through the Google assistant are decoded and then sent to the Node MCU, the Node MCU in turn control the relays connected to it. The device connected to the respective relay can be turned On or OFF as per the users request to the Google Assistant





Demo Link: https://drive.google.com/file/d/1NyzP3gPXE6bP-OJqFFrGebLBB_JV3g4X/view?usp=share_link

Contribution of Group Members:

Everyone has participated in both Review-1 and Review-2 of the project.

- ✓ **Tejaswini G J(20181CSE0746):** worked on preparing physical model such as editing the code part,gathering the required home autimation libraries and building the circuit and required software for the project. Prepared ppt for review-1. Preparation of report.

- ✓ **TANGIRALA RAMA NARAYANA REDDY(20181CSE0743):** Helped in giving ideas for preparing physical model, prepared ppt for review-1.Contributed for arranging the components for the project.

- ✓ **V Meghana(20181CSE0760):** Helped in preparing final report. Always gathered the team to work together.Contributed for arranging the components for the project.

- ✓ **UPPATALA BHAVYA KRISHNA(20181CSE0759):**Extracted information for preparing ppt for review-1. Contributed for arranging the components for the project.

- ✓ **Tatikonda Sandeep(20181CSE0748):** Contributed his work for collecting the information and source/related vedios to build physical model. Contributed for arranging the components for the project.