1. Required Components

1.1 Hardware:

1) ESP8266: The ESP8266 is a System on a Chip (SoC), manufactured by the Chinese company Espressif. It consists of a Tensilica L106 32-bit micro controller unit (MCU) and a Wi-Fi transceiver. It has 11 GPIO pins* (General Purpose Input/Output pins), and an analog input as well.

2) Relay: An electrical switch that is operated by an electromagnet, such as a solenoid. When a small current passes through the electromagnet's coiled wire, it produces a magnetic field that attracts a movable iron bar, causing it to pivot and open or close the switch.

3) USB Type A 2.0 To Micro US 3.0 Cable

4) 5V Charger Circuit

5) LED Light

6) Wires

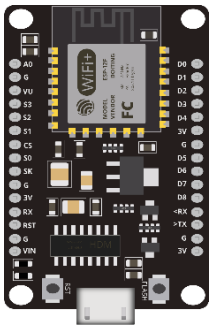
1.2 Software:

1) Adafruit IO: Adafruit IO is a cloud service available on the network and with which it is possible to connect our devices, such as an Arduino board. Its main function is to store the data acquired by one or more boards connected to sensors, to show them both in real time and subsequently, but it can also perform other interesting functions.

2) IFTTT: If This Then That (commonly known as IFTTT) 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.

3) Google Assistance: Google Assistant is an artificial intelligence–powered virtual assistant developed by Google that is primarily available on mobile and smart home devices.

Note: Use The same Gmail account



ESP8266

Relay

LED

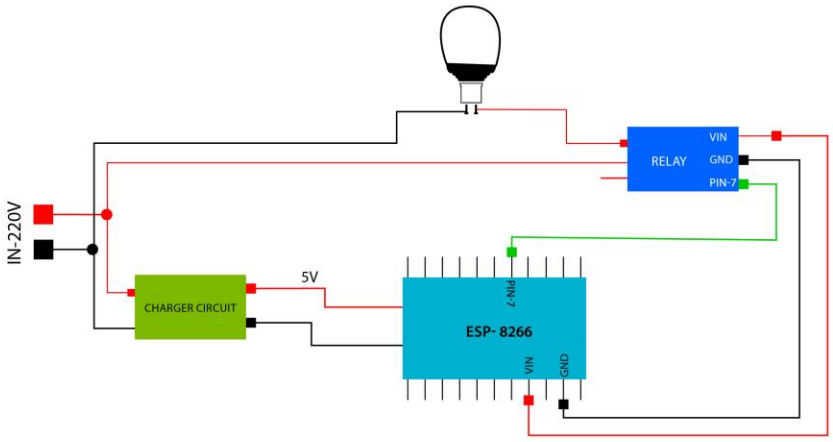
USB cable

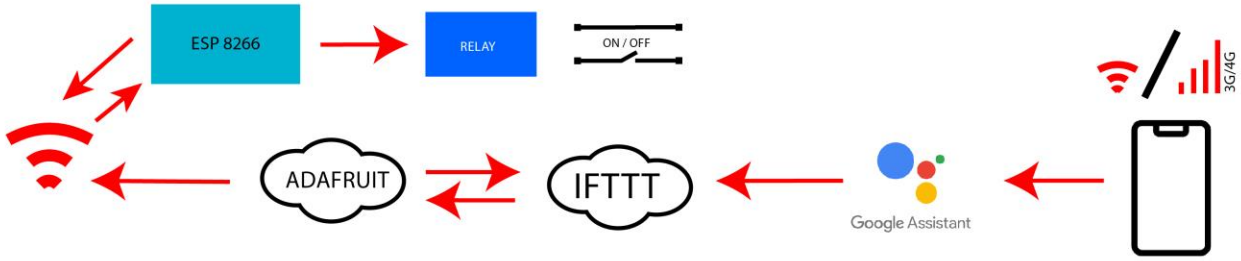
Charger Circuit

Jumper Cable

Figure 1.1: Hardware

2. Circuit Diagram Project work Flow



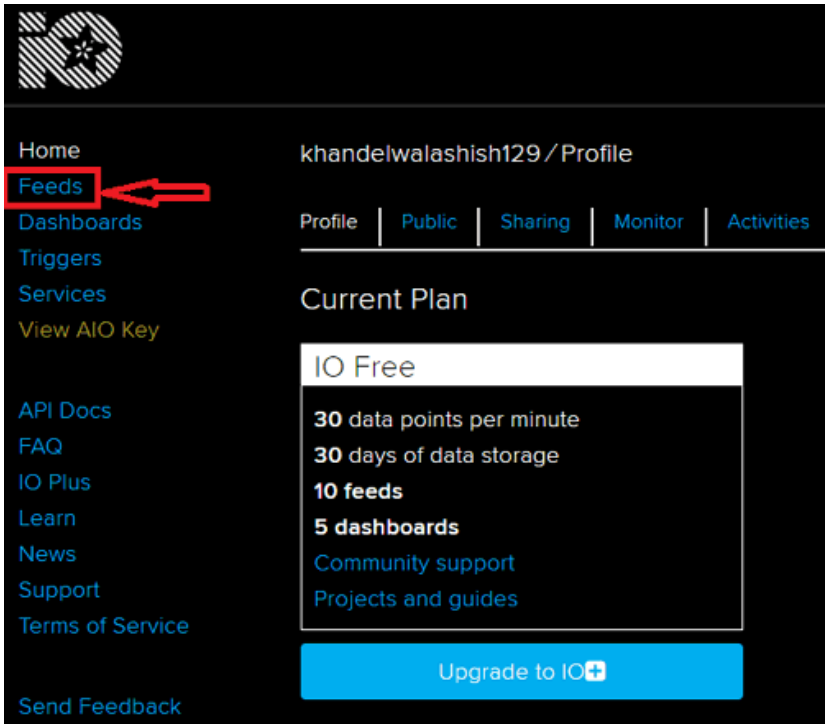


3. Methodology

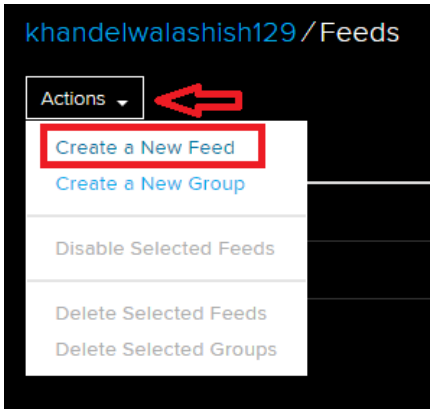
For this project you have to follow following steps to getting started with Adafruit IO:

3.1: Setting Up Adafruit IO

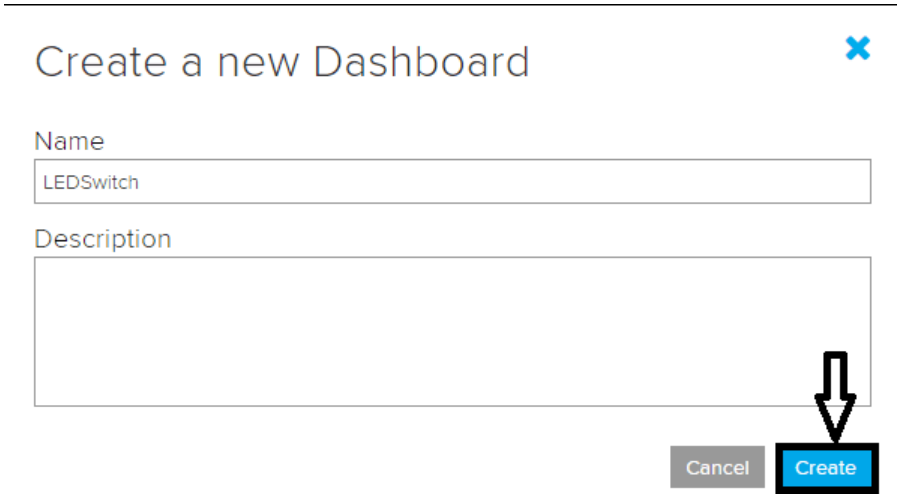
- Visit **https://io.adafruit.com** and create an account.
- After creating your Account, you will be taken to your home screen. Click on **"Feeds"** from the left hand menu.



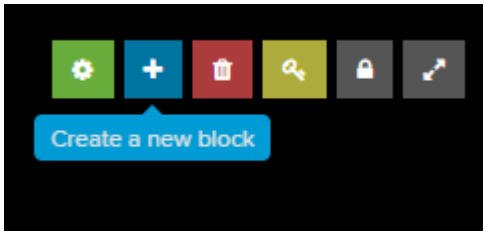
- Now click on Actions and then create a New feed. Then it will ask you to name your feed I am giving it **LightsStatus**, you can give according to you and then create and your feed is created. Create another feed name **ONOFF**



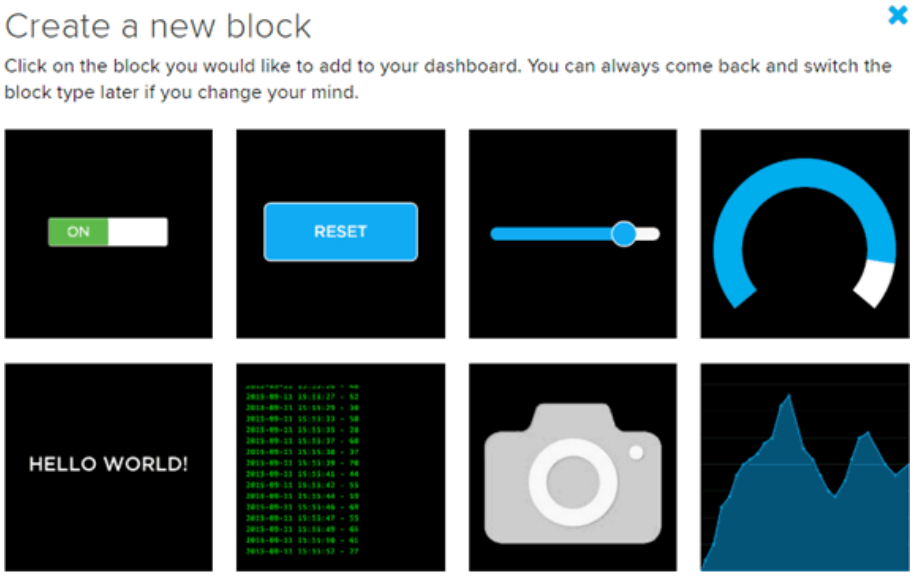
- Now go to **"Dashboards"** from the left hand menu. Click on Actions and then click on create a new dashboard, give it name as you want; I am giving **"LEDSwitch"** and then click on **Create**.



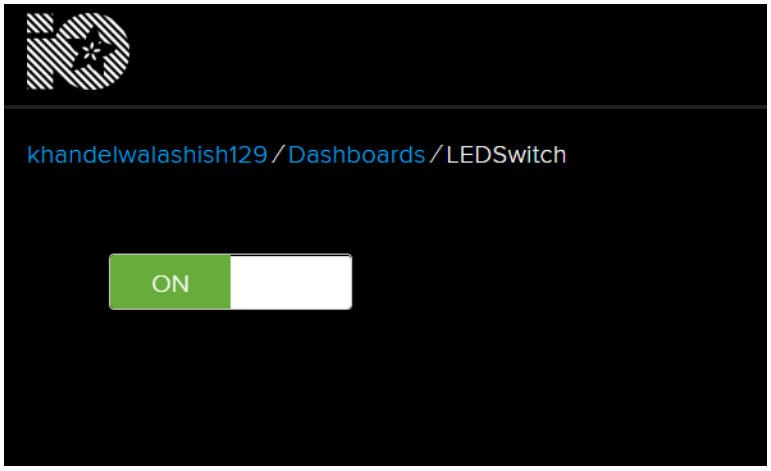
- Now open your new dashboard by simply clicking on it and you should be taken to a mostly blank page. Clicking on **blue** + button will let you add new UI components to the dashboard.



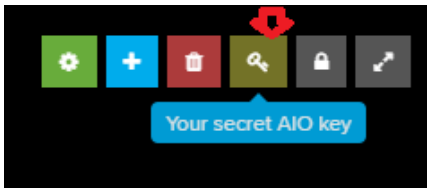
- For this project I just need a button, so select first option, it will ask you to select the feed, so select the one you just made and keep the defaults for the rest of the settings.



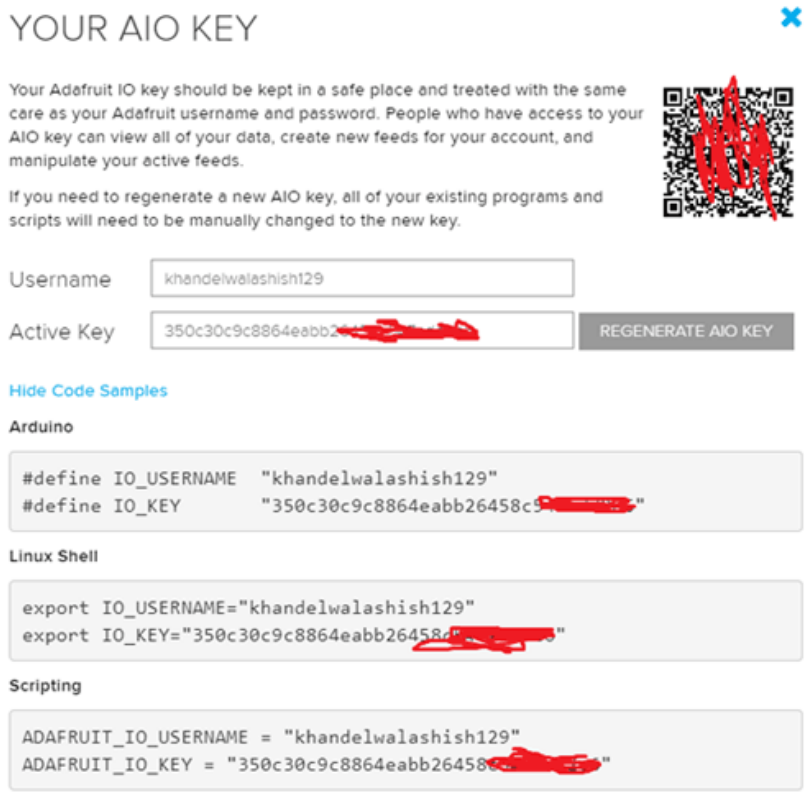
- After selecting your dashboard window will look like this:



- During programming you will required your unique AIO key so for this click on key button at right hand corner of your window.



- After clicking on key button your Active key for this project is generated, don't share this key with anyone this must be confidential.



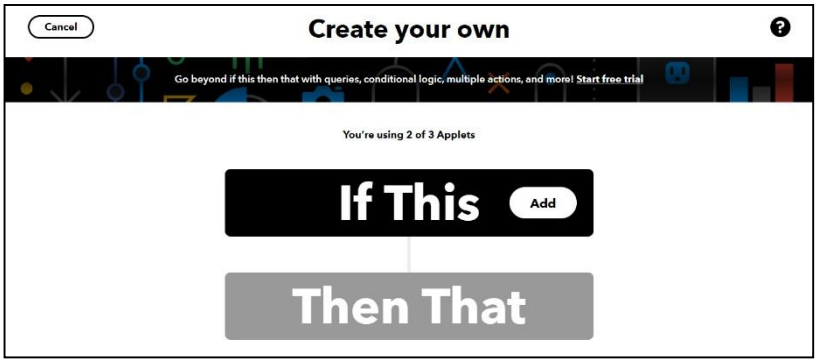
3.2 Connecting to Google Assistant through IFTTT

In this step, we will connect our Google Assistant to the Adafruit IO MQTT Broker to allow us to control the lights with voice commands. To do this I am using IFTTT (If This Then That) platform.

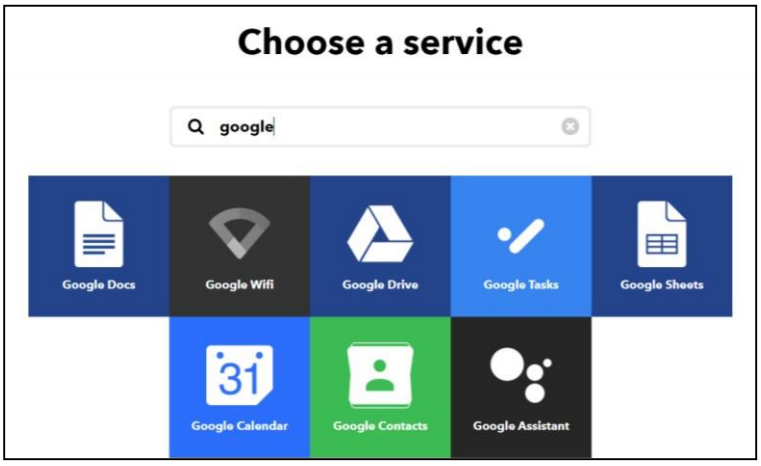
3.2.1 Go to www.IFTTT.com website and create a new account if you are not having already. You can sign up using your Google account also.

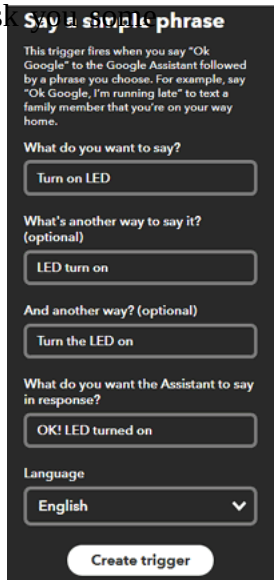
3.2.2 After creating an account click on “Create” button.

3.2.3 After clicking on New applet you will find a window which ask you ‘If this then that’. The term IF THIS THEN THAT means if something happens on the “This” then we have do something on “that”.



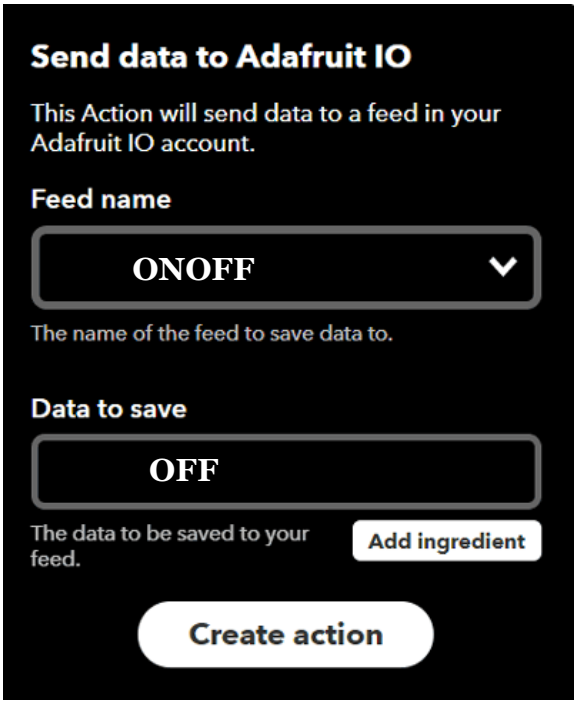
3.2.4 Click on + blue button and search for “Google Assistant”, and then select “Say a simple phrase” from the menu of specific triggers. This will ask you to complete details, fill according to you and create trigger.





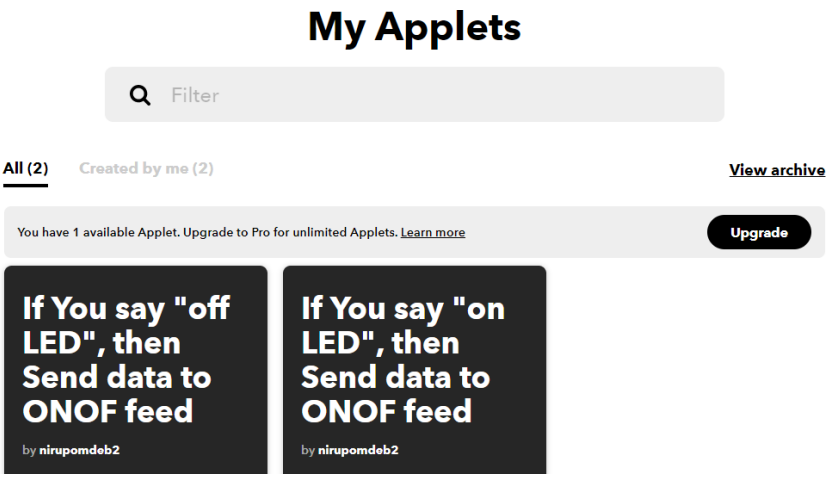
3.2.5 Now you have to give Action so click on + button of “That”, and search for Adafruit and click on “Send data to Adafruit IO”

3.2.6 Now it will ask you to select the Feed name so select the feed that you made earlier for this project and in Data to save we will send ON for this applet and click on Create action.



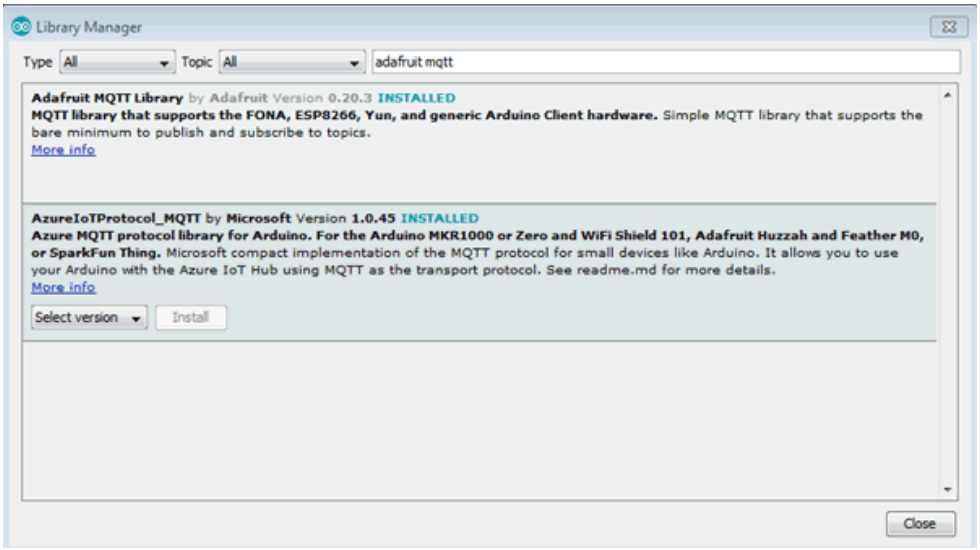
3.2.7 Once you have created this applet, you have to create another applet for turning the LED “OFF”. You have to follow the same steps to create another applet.

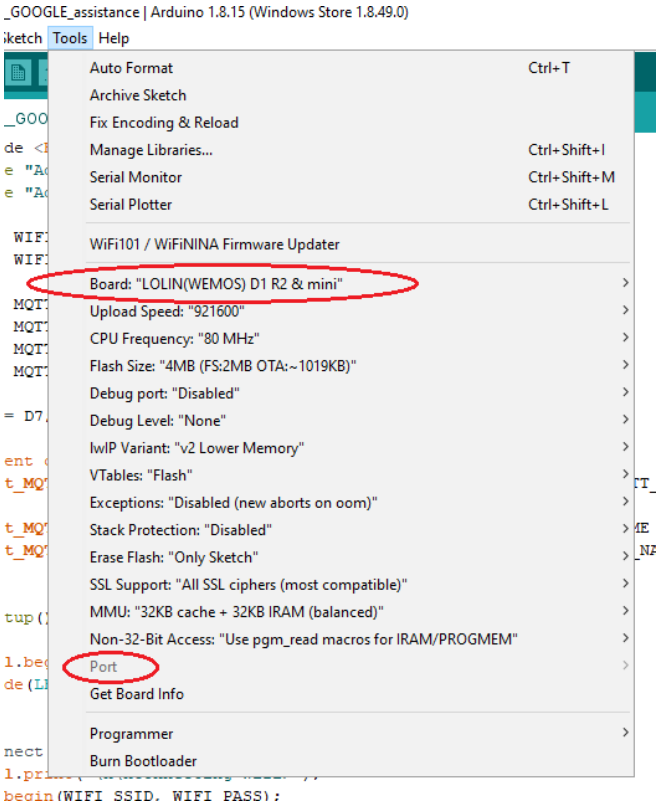
3.2.8 After creating both the applets go to “My Applets” and you can see both the applets here.



3.3 Programming ESP8266 for IoT Based LED Light

3.3.1 Use **Arduino IDE** for programming
3.3.2 Setting up IDE for ESP8266
3.3.3 Connect ESP8266 with your Computer
3.3.4 Select **Board** and **Port**





3.4 Program:

```
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"

#define WIFI_SSID "YOUR_WIFI_NAME"
#define WIFI_PASS "YOUR_WIFI_PASSWORD"
#define MQTT_SERV "io.adafruit.com"
#define MQTT_PORT 1883
#define MQTT_NAME "YOUR_ADAFRUIT"
#define MQTT_PASS "YOUR_ADAFRUIT_KEY"

int led= D7;

WiFiClient client;
Adafruit_MQTT_Client mqtt(&client, MQTT_SERV, MQTT_PORT, MQTT_NAME, MQTT_PASS);

Adafruit_MQTT_Subscribe ONOFF = Adafruit_MQTT_Subscribe(&mqtt, MQTT_NAME
    "/f/ONOFF");
Adafruit_MQTT_Publish LightsStatus = Adafruit_MQTT_Publish(&mqtt, MQTT_NAME
    "/f/LightsStatus");



void setup()
{
 Serial.begin(9600);
 pinMode(LED_BUILTIN, OUTPUT);



 //Connect to WiFi
 Serial.print("\n\nConnecting Wifi>");
 WiFi.begin(WIFI_SSID, WIFI_PASS);
 digitalWrite(LED_BUILTIN, LOW);

 while (WiFi.status() != WL_CONNECTED)
 {
  Serial.print(">");
  delay(50);
```

```
  }

  Serial.println("OK!");

  //Subscribe to the onoff topic
  mqtt.subscribe(&ONOFF);

  pinMode(led, OUTPUT);
 digitalWrite(LED_BUILTIN, HIGH);

}

void loop()
{
  //Connect/Reconnect to MQTT
  MQTT_connect();

  //Read from our subscription queue until we run out, or
  //wait up to 5 seconds for subscription to update
  Adafruit_MQTT_Subscribe * subscription;
  while ((subscription = mqtt.readSubscription(5000)))
  {
    //If we're in here, a subscription updated...
    if (subscription == &ONOFF)
    {
      //Print the new value to the serial monitor
      Serial.print("ONOFF: ");
      Serial.println((char*) ONOFF.lastread);

      //If the new value is  "ON", turn the light on.
      //Otherwise, turn it off.
      if (!strcmp((char*) ONOFF.lastread, "ON"))
      {
        //active low logic
        digitalWrite(led, HIGH);
        LightsStatus.publish("ON");
      }
      else if (!strcmp((char*) ONOFF.lastread, "OFF"))
```

```
      {
        digitalWrite(led, LOW);
        LightsStatus.publish("OFF");

      }
      else
      {
        LightsStatus.publish("ERROR");
      }
    }
    else
    {
      //LightsStatus.publish("ERROR");
    }
  }
  //  if (!mqtt.ping())
  //  {
  //    mqtt.disconnect();
  //  }
}


void MQTT_connect()
{

  //  // Stop if already connected
  if (mqtt.connected() && mqtt.ping())
  {
    //    mqtt.disconnect();
    return;
  }

  int8_t ret;

  mqtt.disconnect();

  Serial.print("Connecting to MQTT... ");
  uint8_t retries = 3;
```

```
while ((ret = mqtt.connect()) != 0) // connect will return 0 for connected
{
  Serial.println(mqtt.connectErrorString(ret));
  Serial.println("Retrying MQTT connection in 5 seconds...");
  mqtt.disconnect();
  delay(5000);  // wait 5 seconds
  retries--;
  if (retries == 0)
  {
    ESP.reset();
  }
}
Serial.println("MQTT Connected!");
    }
```

4 Experimental Result

4.1 Testing

Now all that's left is to test the system. To do this, you'll need a device with the Google Assistant enabled. This is built into the latest versions of the Android operating system, as well as the Google Home series of devices. If you don't have either of these, the Google Allow messaging app, available for Android and iOS, also includes the Google Assistant. Start up the assistant, make sure it's logged into the proper account, and say the activation phrase you used in the previous step. After a 2-5 second delay, the light on your ESP8266 board should switch on or off. Try the other phrase, and make sure it works too.

4.2 Advantages of our project

4.2.1 We can control LED without physical switch.
4.2.2 We can control LED with Adafruit Toggle Switch.
4.2.3 We can control LED with Google Voice command.
4.2.4 We can control this Light from anywhere if it connected to the internet.
4.2.5 We Can use this technology many useful project.

We can control smart home devices including lights, switches, fans, and thermostats using our Go