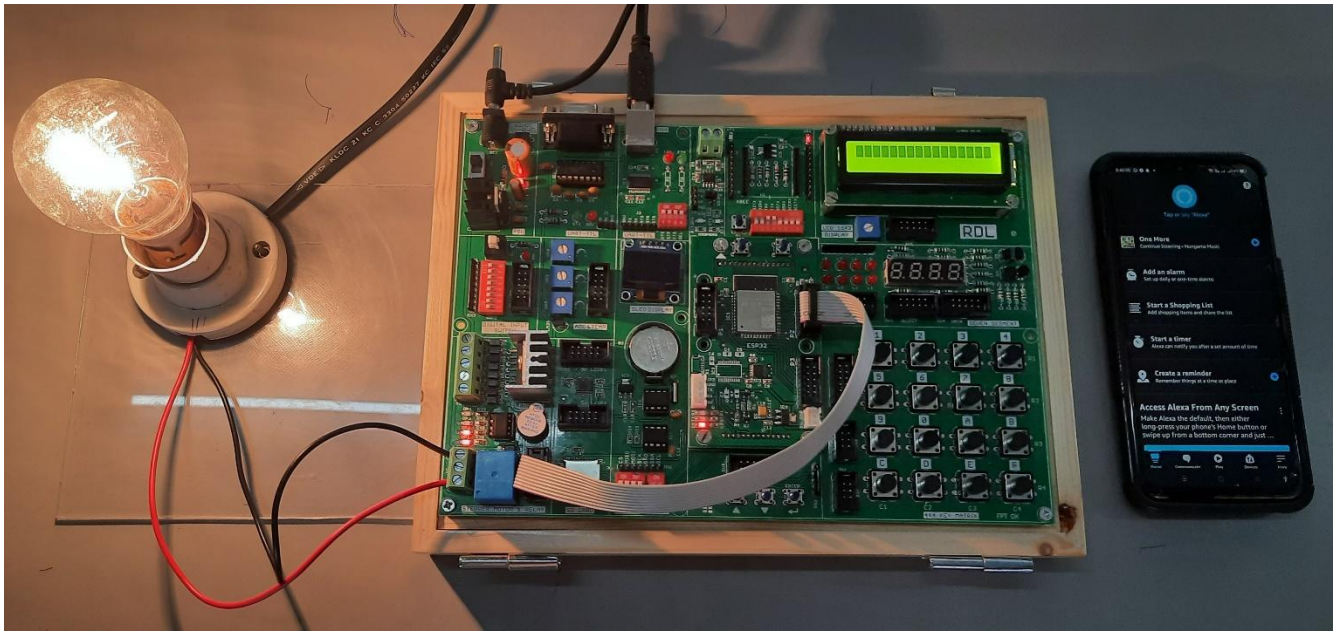
VOICE CONTROLLED APPLIANCES USING ALEXA AND ESP32 CONTROLLER



DOCUMENT VERSION: 1.0

- Switch off the supply voltage of this product as well as of attached devices before connecting or disconnecting them.
- Always use insulated tools while working.
- Do not touch any components of the board in open hand during power ON

Table of Contents

Contents

1. INTRODUCTION 3
2. LEARNING OBJECTIVE 3
3. REQUIRED COMPONENTS 4
4. REQUIRED CLOUD SERVICES 5
5. BLOCK DIAGRAM 5
6. PROCEDURE 6
6.1 : Setting up Adafruit IO: 6
6.2 : Setting up IFTTT: 12
6.3 : Setting up ESP32 Development Kit: 20
6.4 : Code: 20
6.5 : Setting up Amazon Alexa: 22
7. RESULT 23

1. INTRODUCTION

How to interface Amazon Alexa to the ESP32 Development Board to control Bulb. By using this power can be controlled (ON / OFF).

For more information about ESP32 Development Kit is available in the link below.

https://www.researchdesignlab.com/projects/ESP32%20User%20Manual-V1.0.pdf

2. LEARNING OBJECTIVE

- How to interface MQTT protocol and how to use it.
- Creating Applets, Trigger and connecting to Amazon Alexa Account in IFTTT.
- How to use Amazon Alexa in Appliances.
- Alexa recognizes the speech and sends command to the Appliances .

3. REQUIRED COMPONENTSs

ESP32 Development Board	
FRC Cable	
USB Cable	
Bulb	
Mobile Phone	

4. REQUIRED CLOUD SERVICES

Adafruit IO :

To create manual setup to turn on and off the **Bulb. And it receives the data from** IFTTT **and sends the data to** ESP32 Development Kit**.** It is advantage is that it is an **Open Source**.

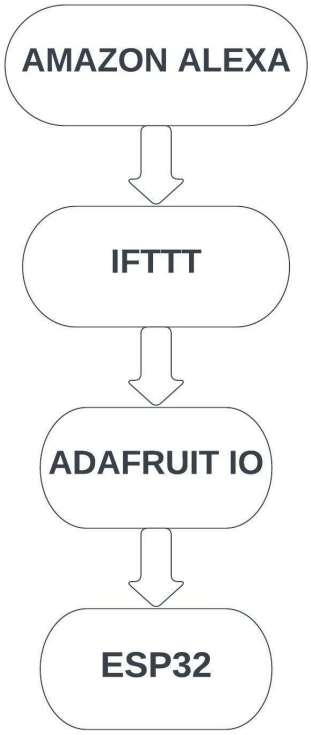
IFTTT **:**

To create the phrases and Feed data to Amazon Alexa. (**Note**: Only **5 Applets** are free to use. If you need more **Applets** you need to **Pay**)

Amazon Alexa **:**

To give voice commands to the **ESP32 Kit** to turn ON/OFF the **Bulb.** It is an **Open Source.**

5. BLOCK DIAGRAM

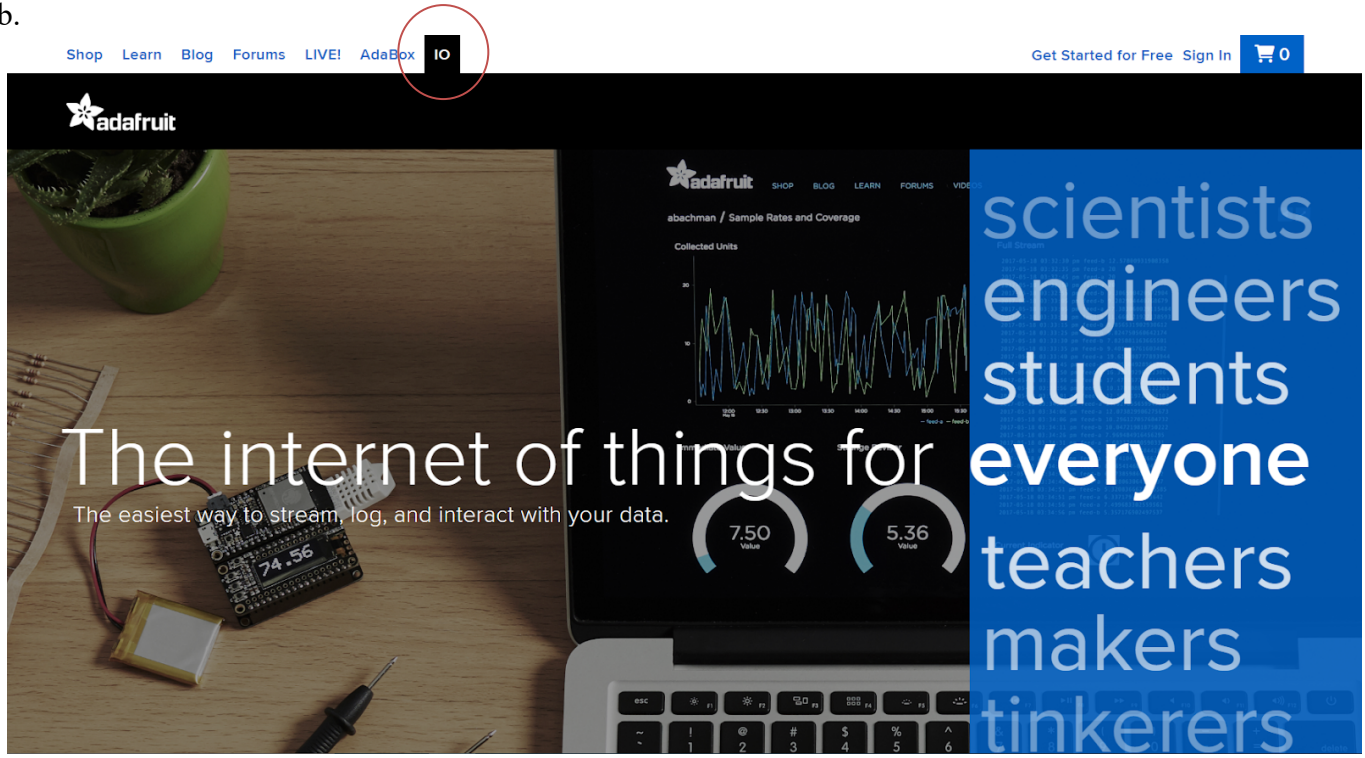


Working Block Diagram

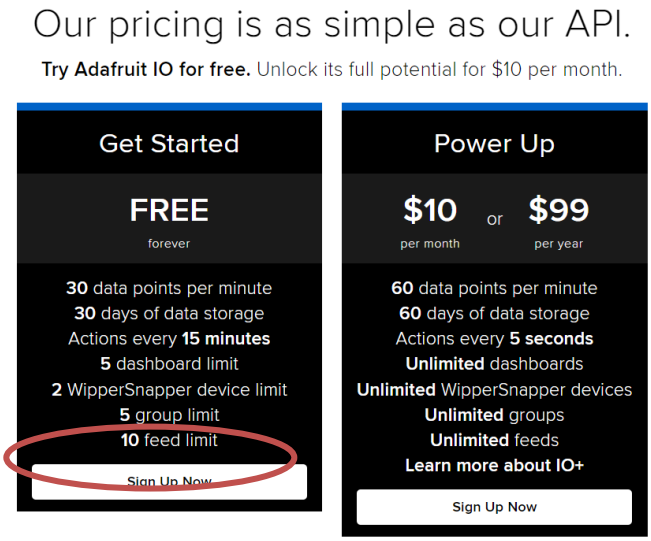
6. PROCEDURE

6.1: Setting up Adafruit IO:

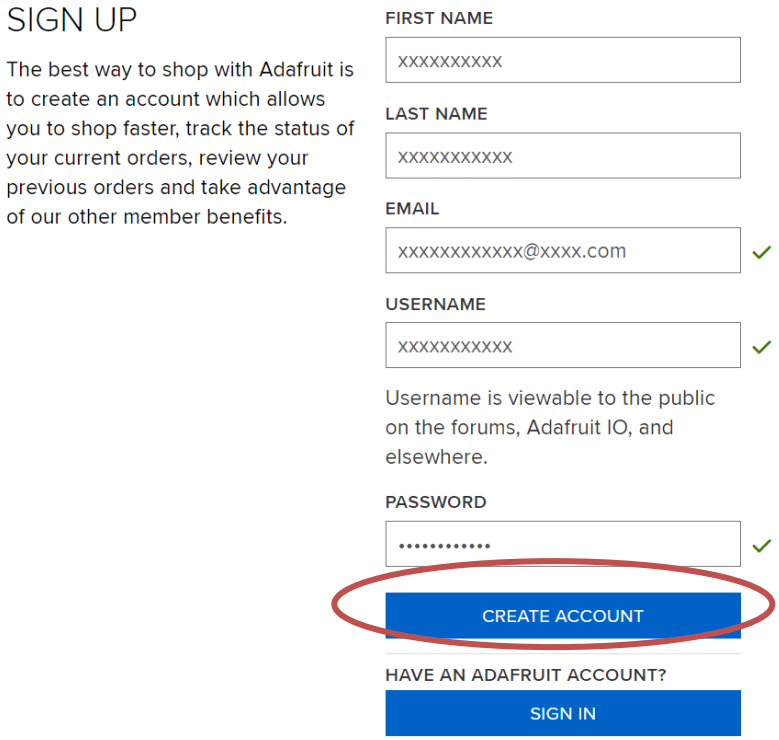
Open **Google Chrome** and Paste the URL (Welcome to Adafruit IO) given. Then click on **IO** Tab.



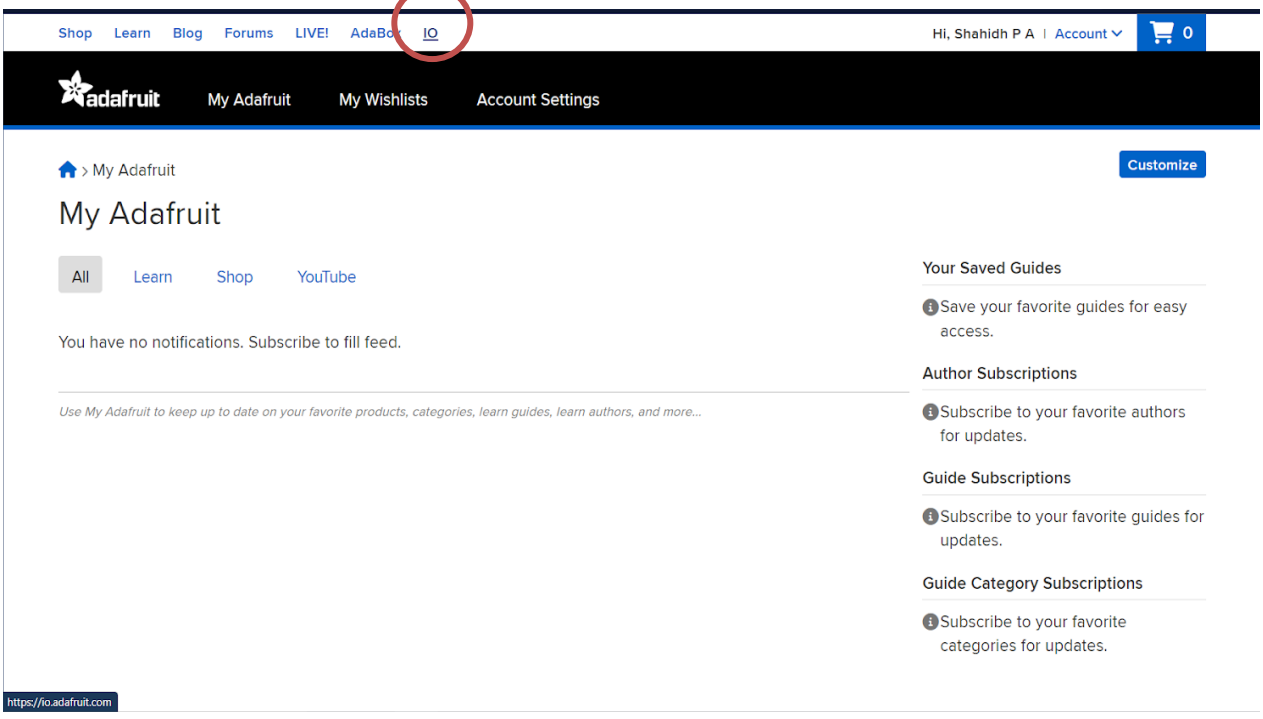
Scroll Down to Get Started For Free Account.



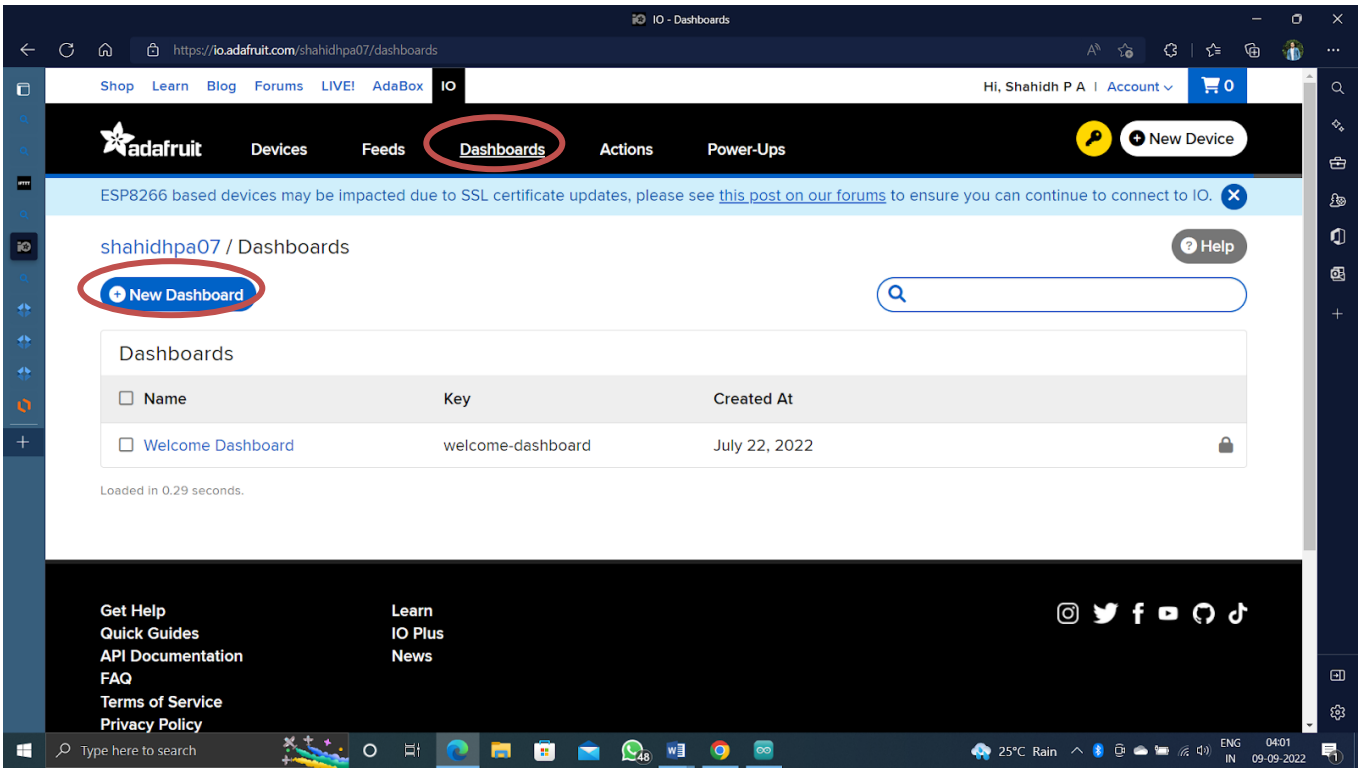
Sign Up for creating a **New Account** .



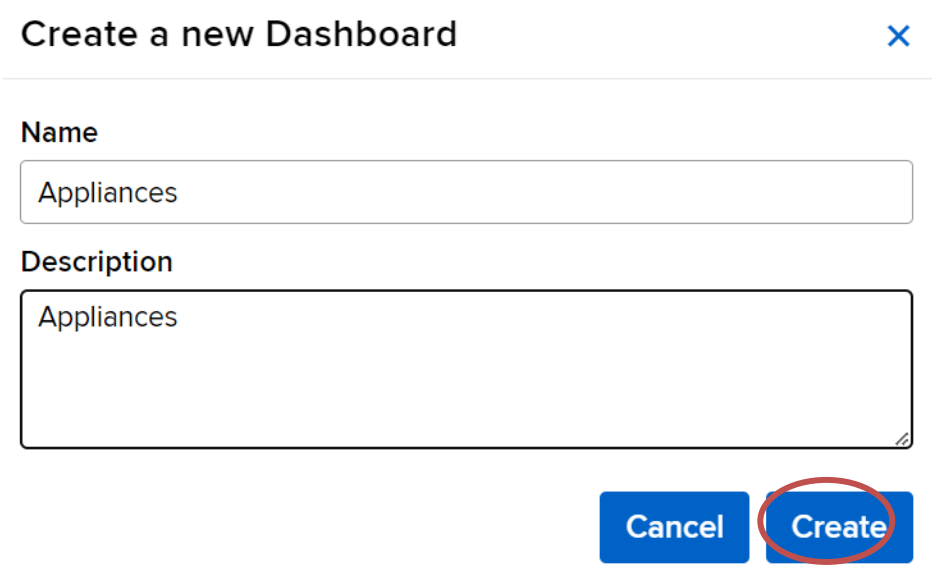
After signing in this page will be displayed and go to **IO** Tab.



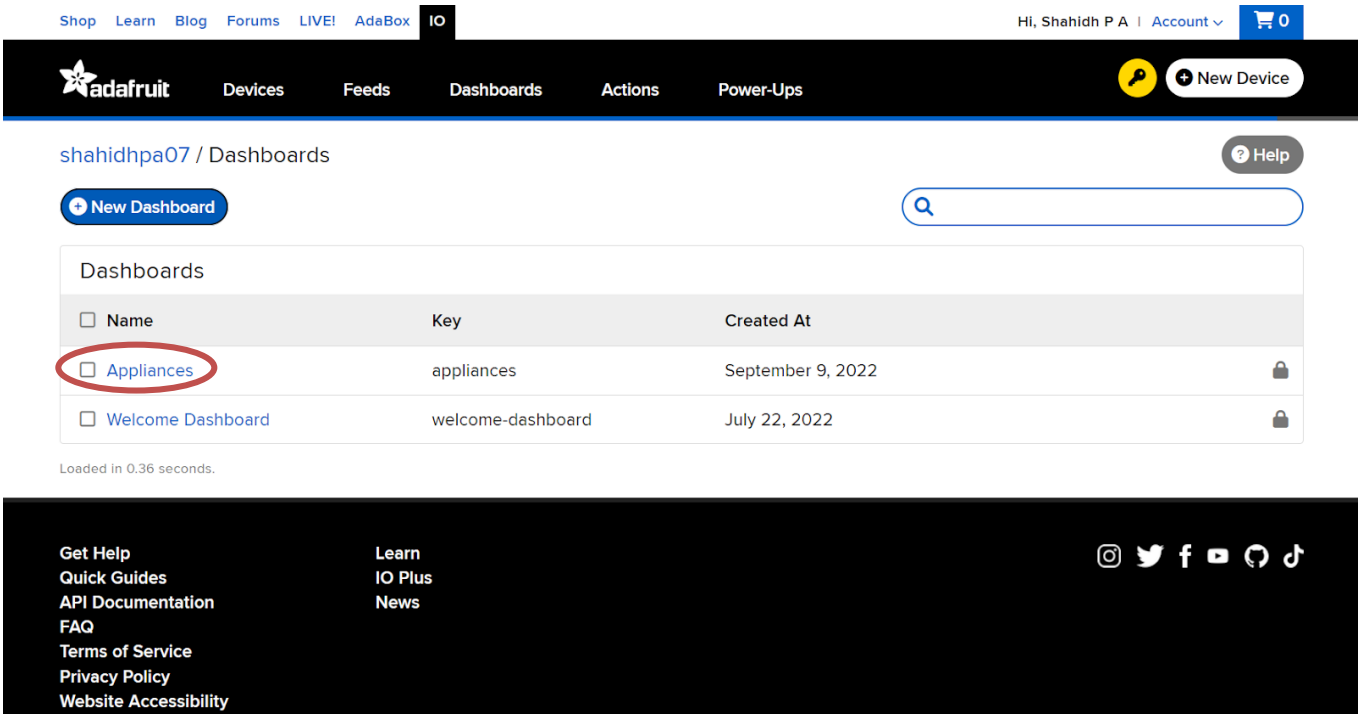
Next click on **Dashboards** and click on **New Dashboards.**



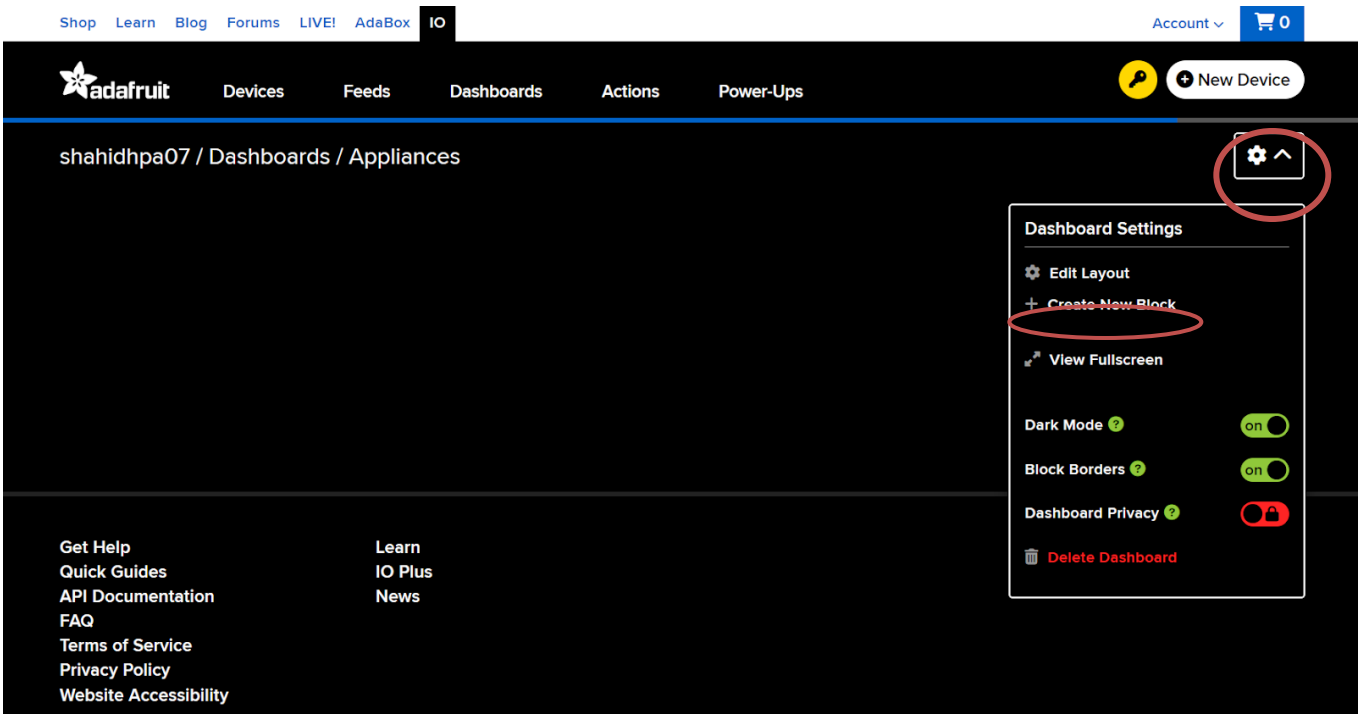
Then Name and Describe the **Dashboard** and Create it.



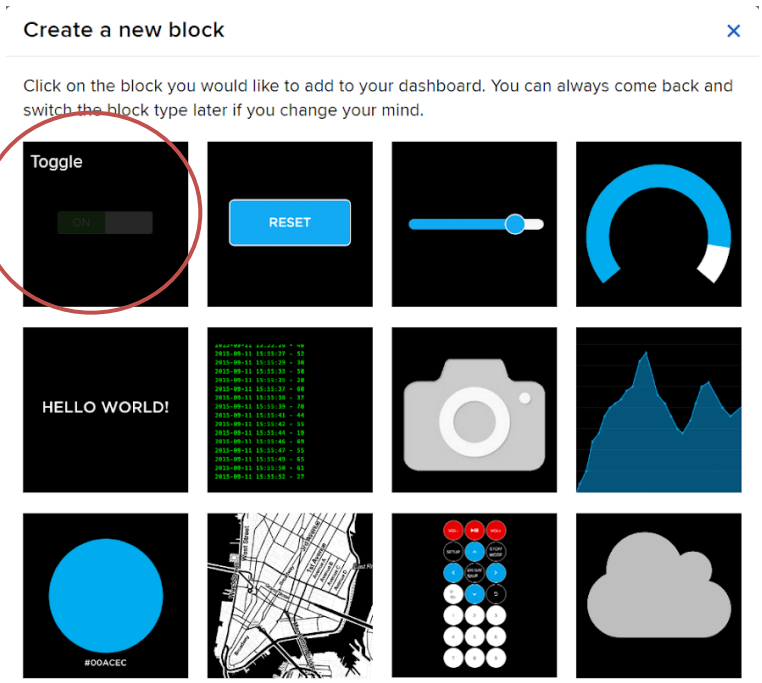
Now there is a **Dashboard** named as **Appliances**. Then click on the **Appliances Dashboard**.



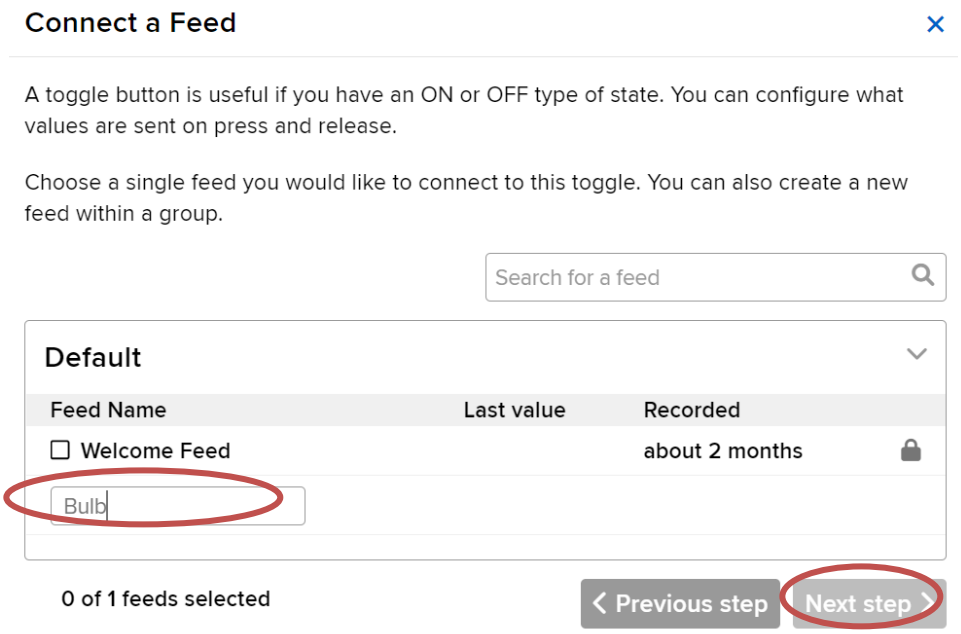
Then go to the **Settings** button and create a **Create New Block.**



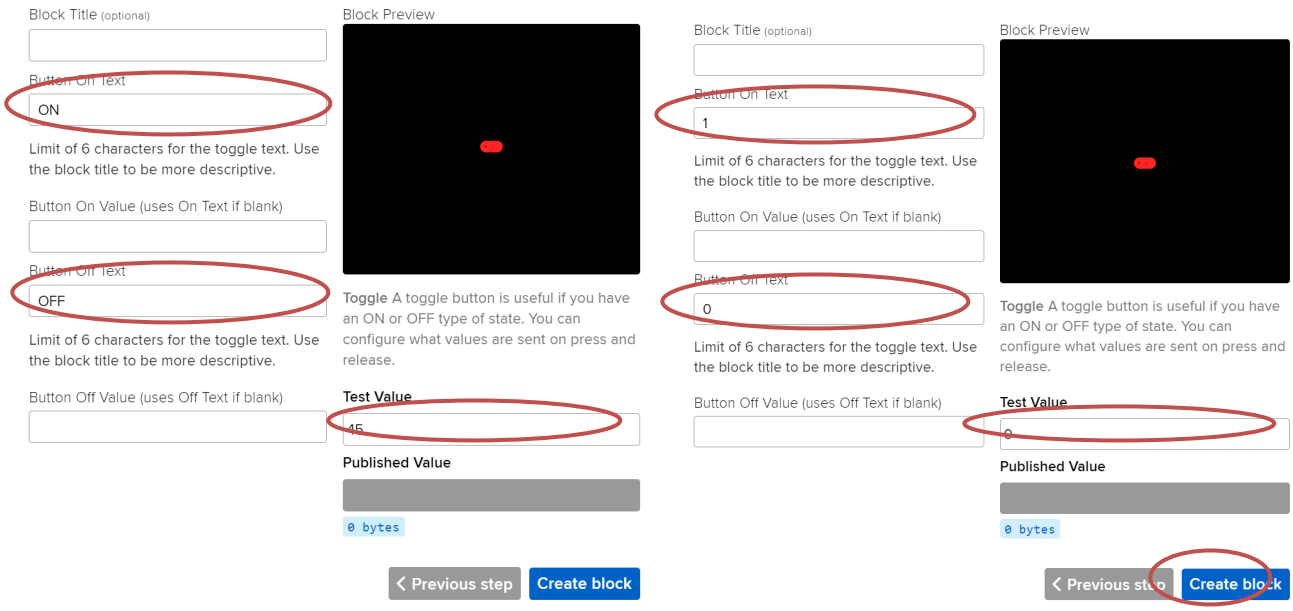
Select any **Block** (For example **Toggle**)



Next **Enter New Feed Name**. For Example **Bulb.** Then click on **Enter.** Then click on the Feed named **Bulb** and click on the **Next Step** as shown in the image below.



Change the **Button On Text** to **1** and **Button of Text** to **0.** Change the **Test Value** as **0.** Then Create Block.

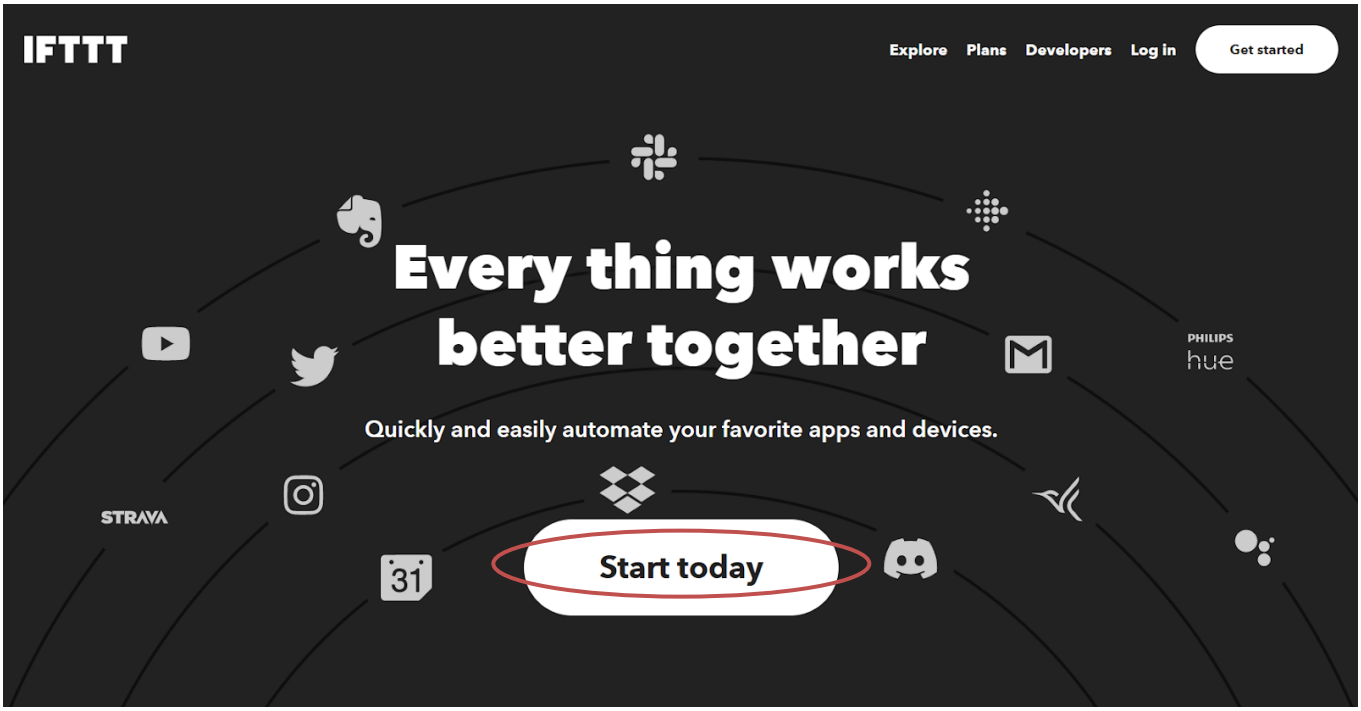


Similarly **Create New Block** for another **Toggle** if you need.

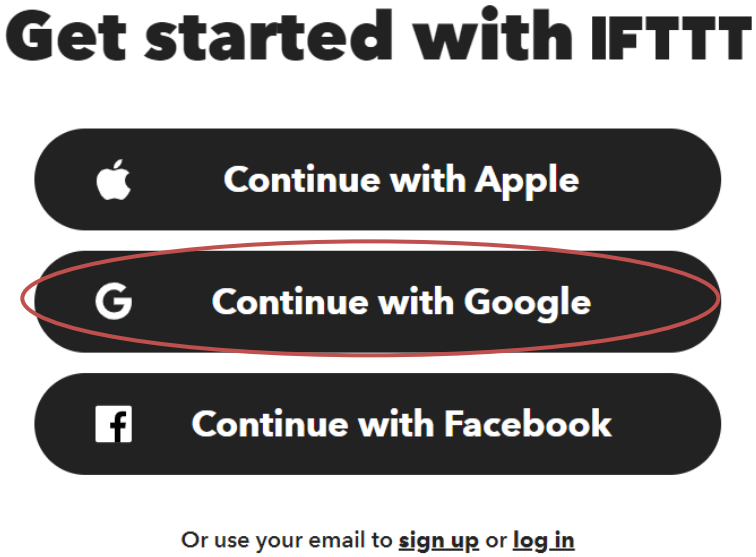
The procedures of **Adafruit IO** is completed.

6.2: Setting up IFTTT:

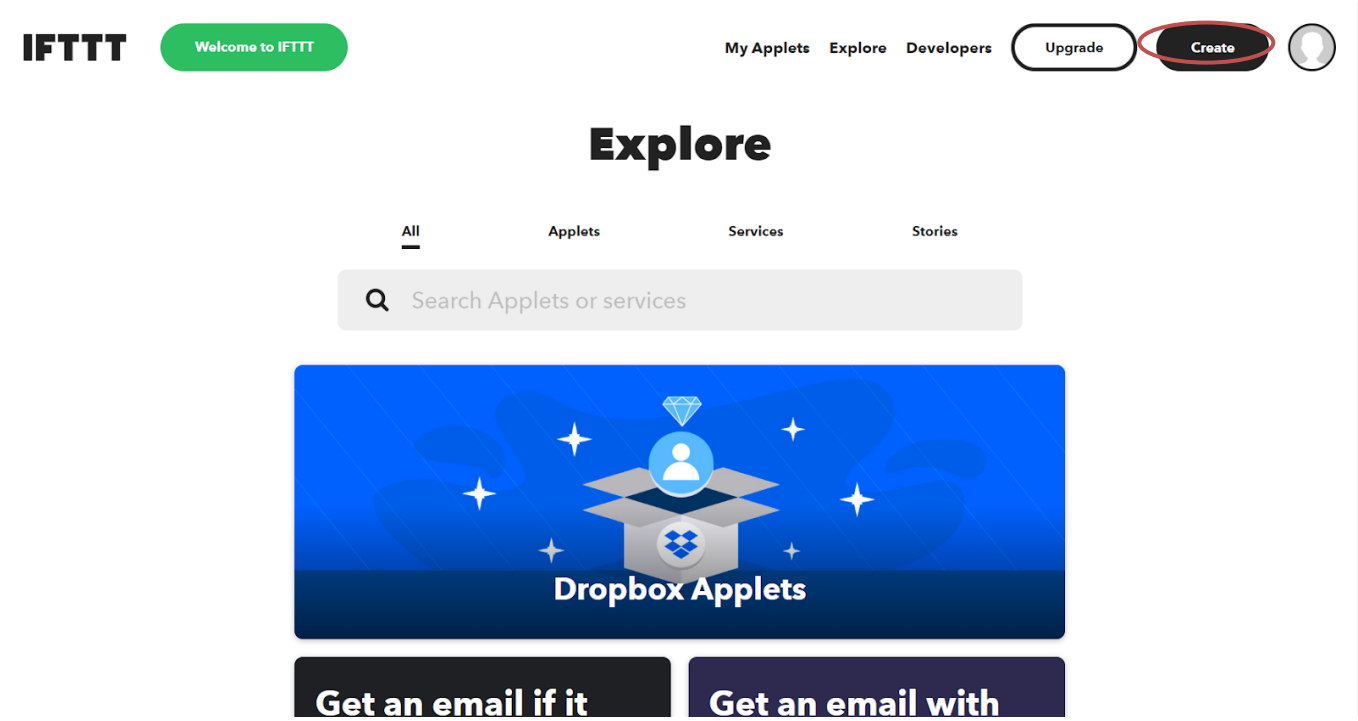
Open the Google Chrome and paste the URL for IFTTT (https://ifttt.com/). Click on **Start today.**



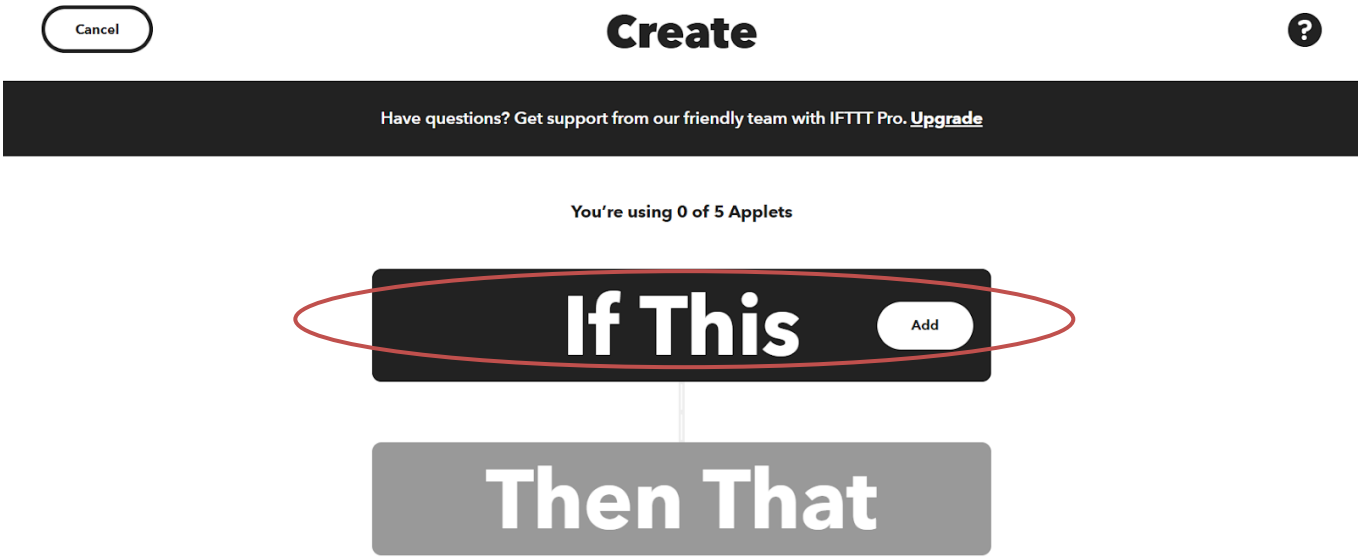
Then **Continue with Google** to create an account. Then sign in with credentials (Create Amazon Account if needed and sign in).



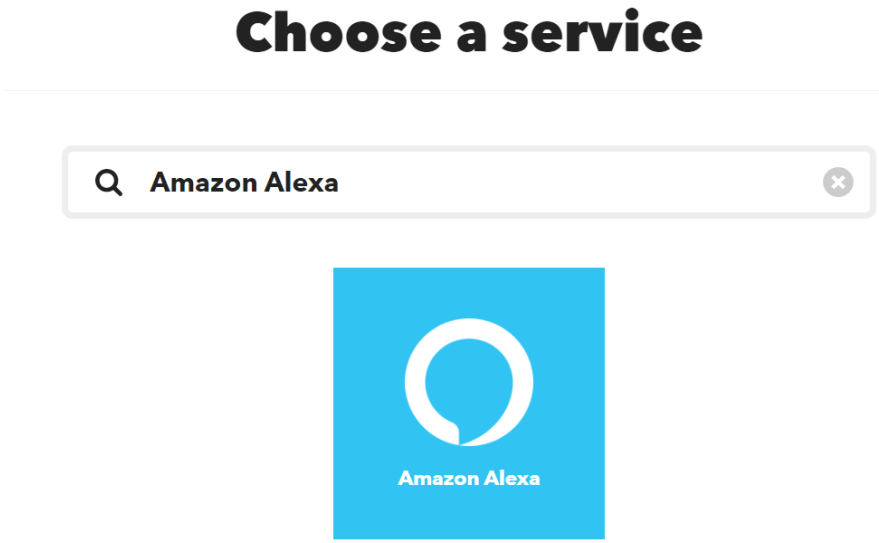
Then the Home Page will open and then Click on **Create.**



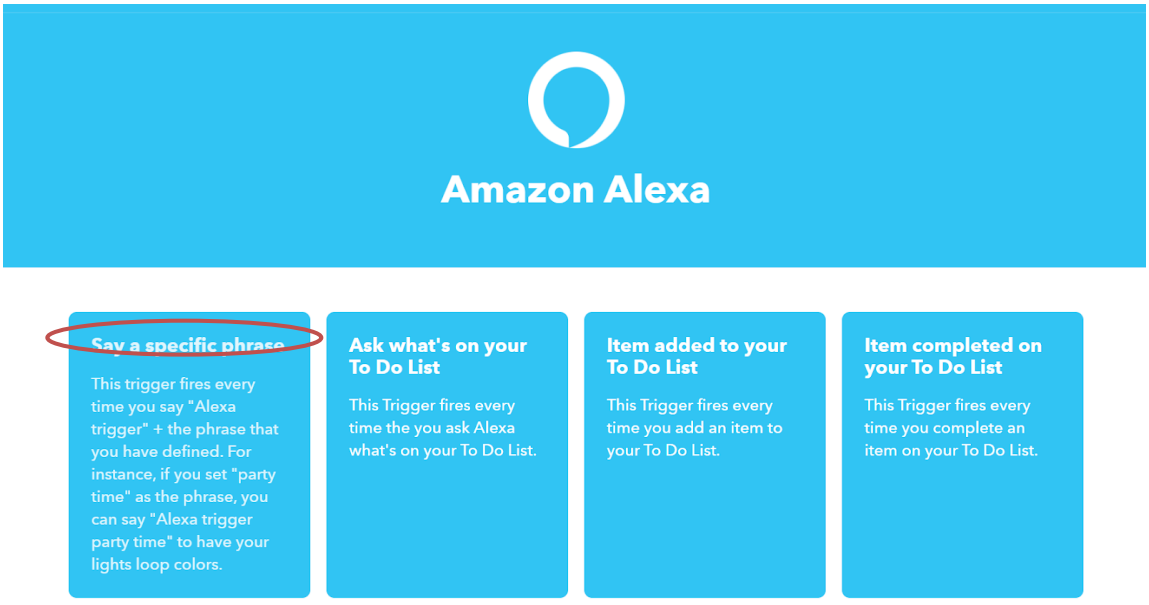
Click on **Add**.



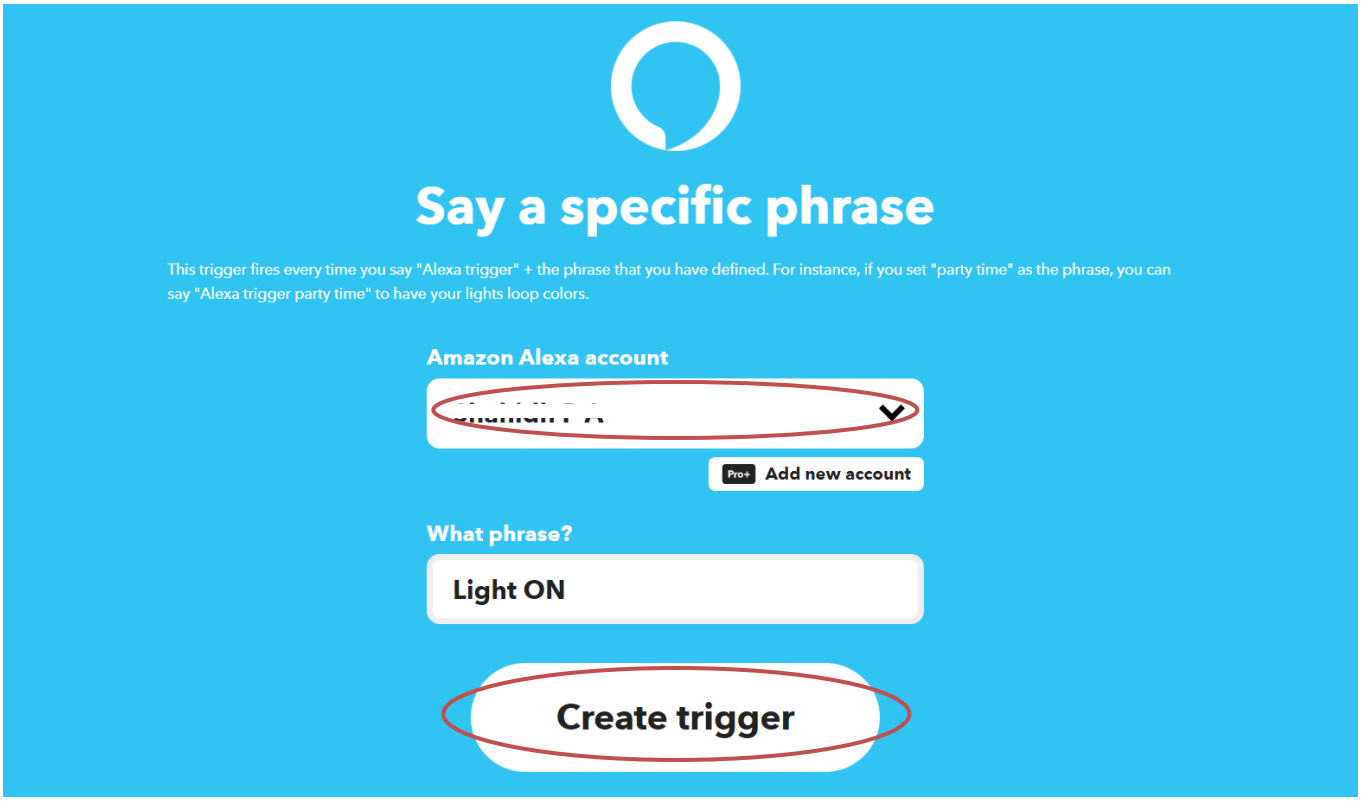
Then choose a service as **Amazon Alexa.**



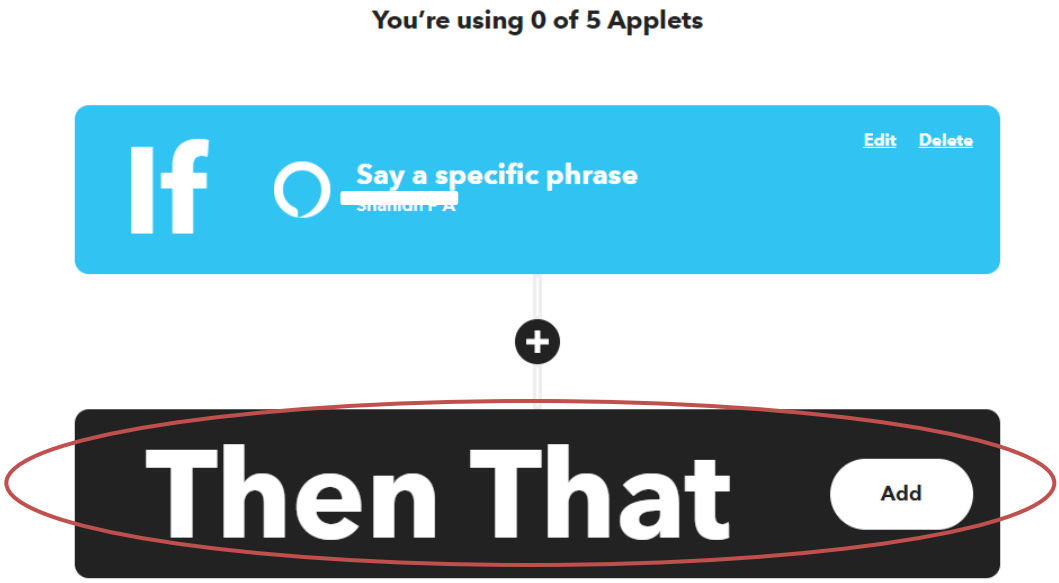
Choose a Trigger "**Say a Specific Phrase**"



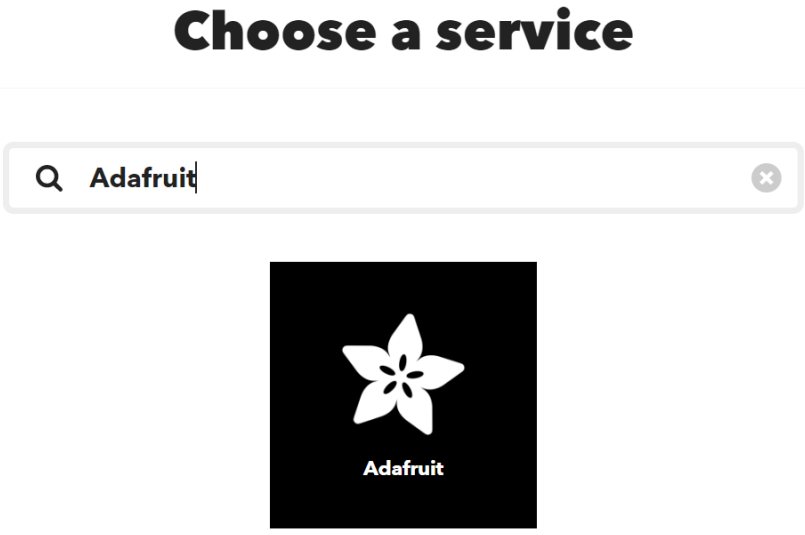
Select the **Amazon Alexa Account** and name a phrase for the trigger and **Create Trigger**.



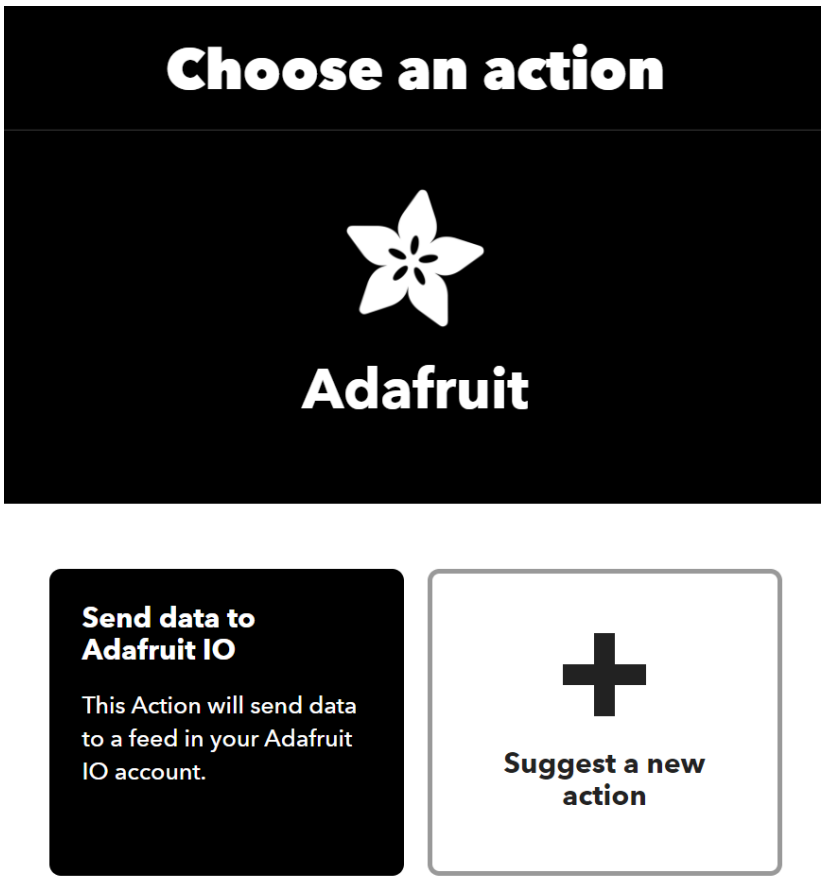
Click on **Add**



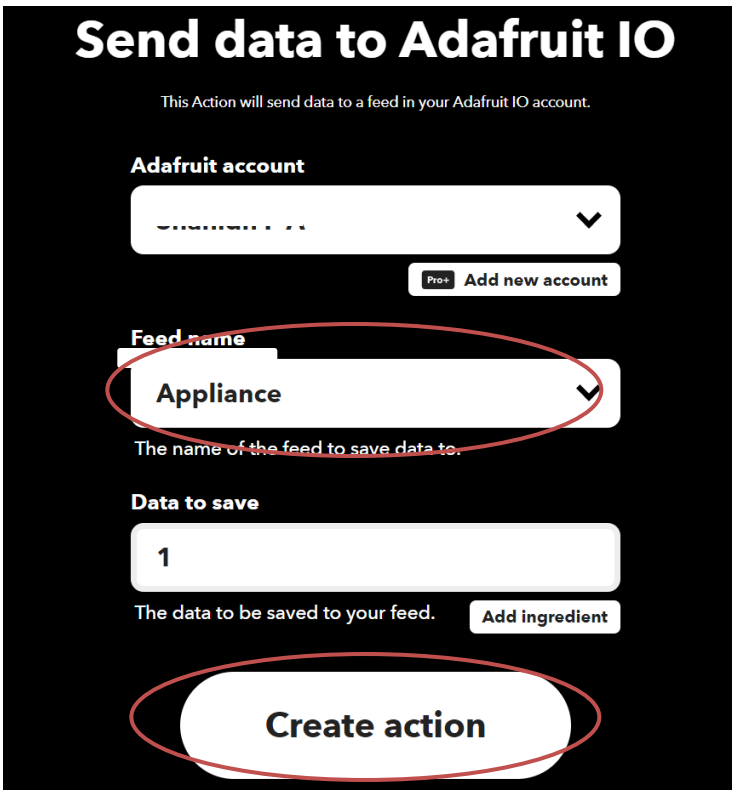
Choose a service as **Adafruit.**



Select "**Send data to Adafruit IO**"

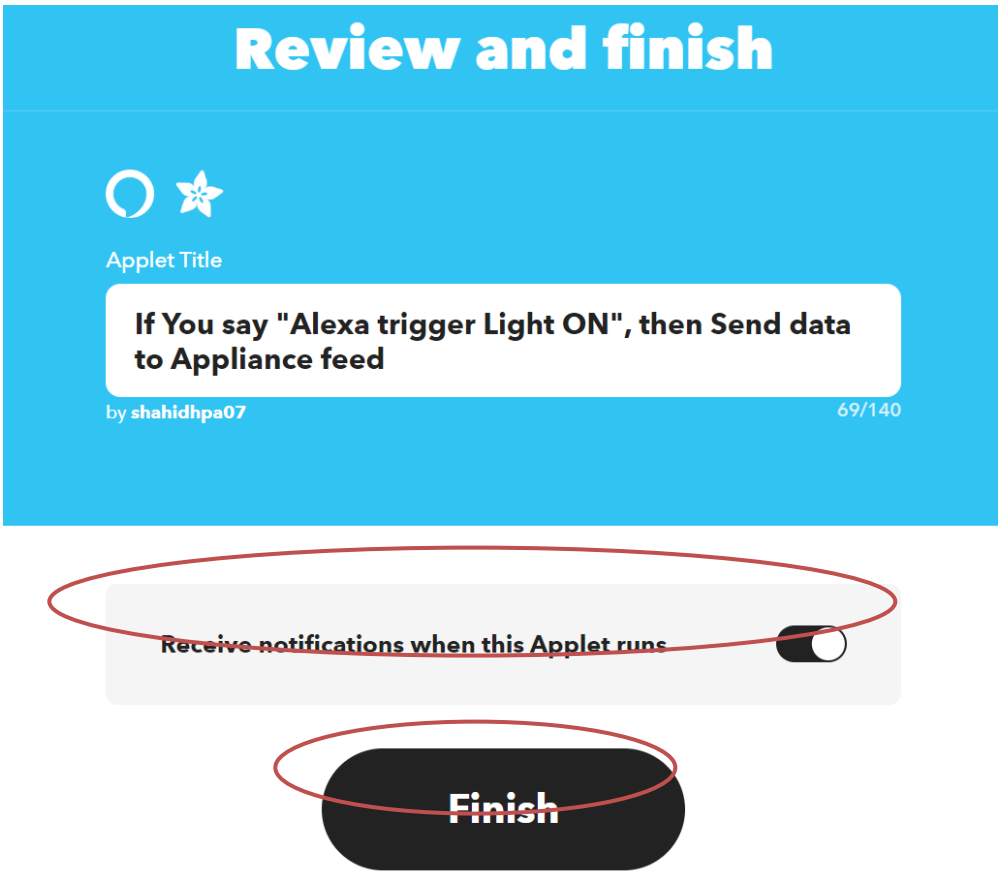


Select the **Adafruit Account**, then Select the **Feed Name** and **Data to save as 1.**

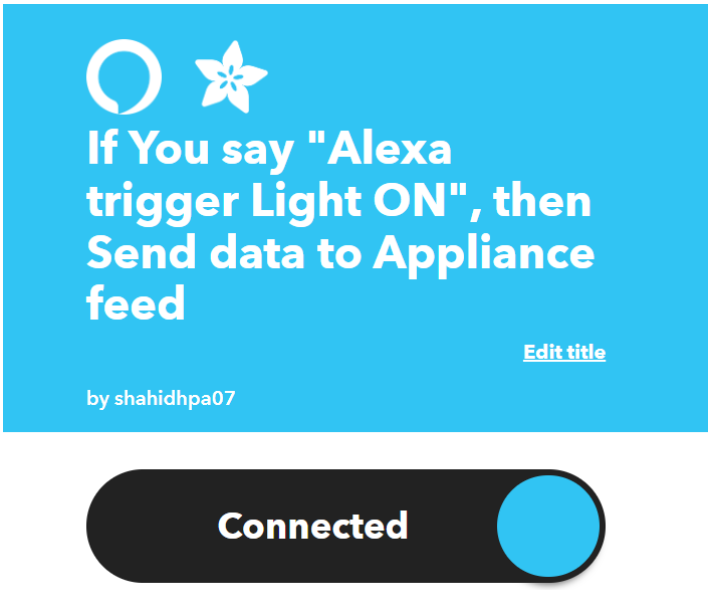


Then Click **Continue** button.

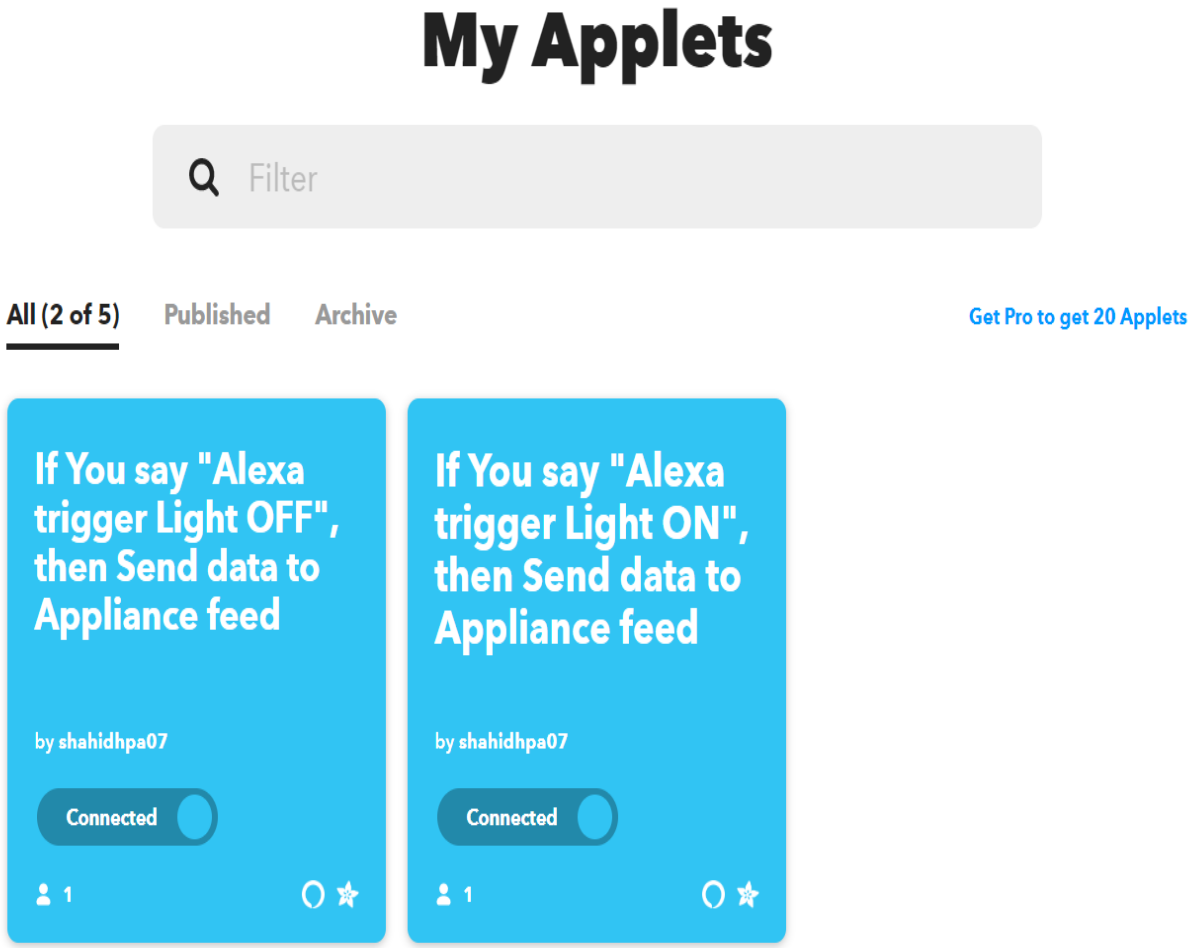
Turn on **Receive notification when this Applets runs.** Then click on **Finish** button.



Then Light ON will be connected.

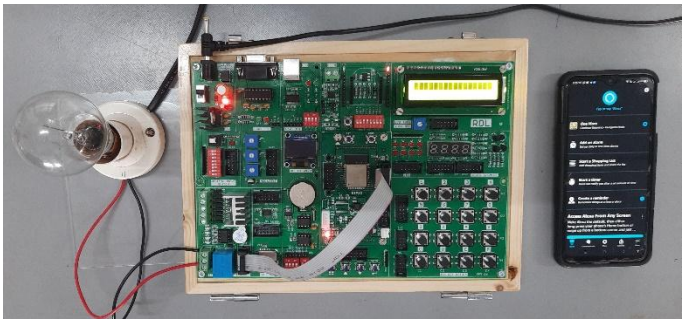


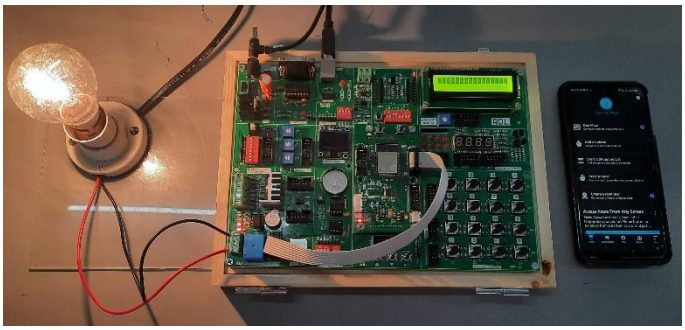
Similarly create another trigger for **Light OFF** (Put **0** instead of **1** in **Data to save).** And see the results in **My Applets.**



The procedure for IFTTT is completed. And then take the ESP32 Development board and connect Light to that board.

6.3: Setting up ESP32 Development Kit:





1. Connect Bulb for the Setup as shown above.
2. Connect the USB cable to the board.
3. Open Arduino IDE. Select DOIT ESP32 DEVKIT V1 in boards and select COM port.
4. Now write the program, verify and Upload it.
5. Now you can see the Blub blink on and off the ESP32 development board.

6.4: CODE

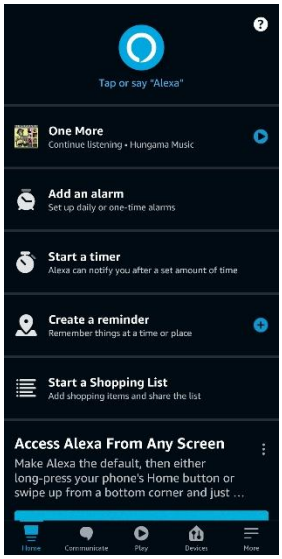
```
#include <WiFi.h>
#include <Adafruit_MQTT.h>
#include <Adafruit_MQTT_Client.h>

#define light           5


#define WLAN_SSID       "xxxxxxxxxx"             // Your SSID
#define WLAN_PASS       "xxxxxxxxxx"        // Your password

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

#define AIO_SERVER      "io.adafruit.com" //Adafruit Server
#define AIO_SERVERPORT  1883
#define AIO_USERNAME    "xxxxxxxxxx"            // Username
```

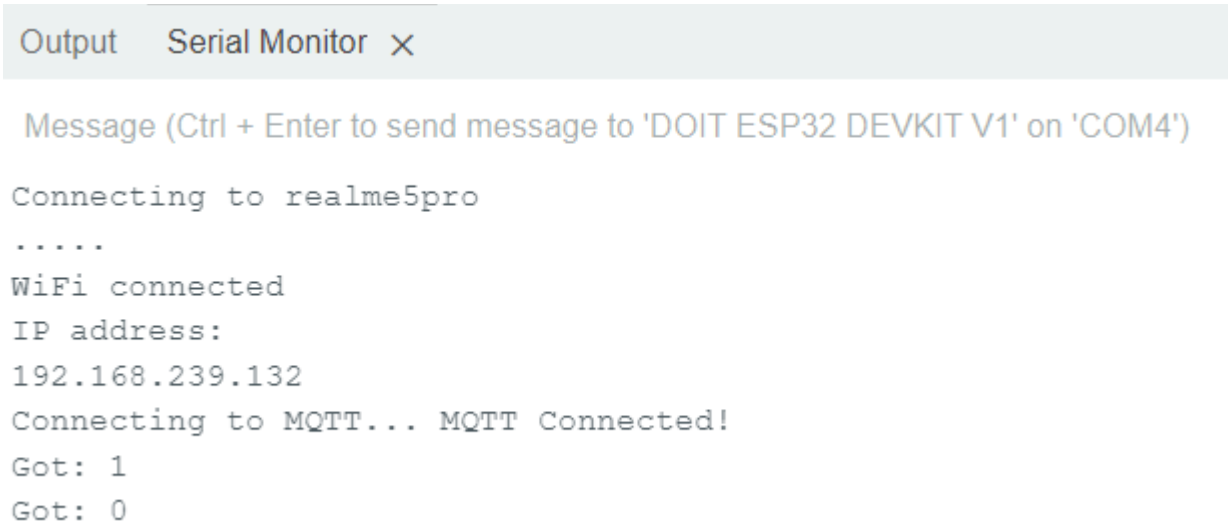
```
#define AIO_KEY         "aio_xxxxxxxxxxxxxxxxxxxxxxxxxxxx"   // Auth Key
/*AIO_USERNAME AND AIO_KEY will be getting from Adafruit IO by clicking on Key
Symbol.*/
//WIFI CLIENT
WiFiClient client;

Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME,
AIO_KEY);

Adafruit_MQTT_Subscribe Light1 = Adafruit_MQTT_Subscribe(&mqtt,
AIO_USERNAME"/feeds/Appliance"); // Feeds name should be same everywhere


void MQTT_connect();

void setup() {
  Serial.begin(115200);

  pinMode(light, 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(&Light1);
  mqtt.subscribe(&Light2);

}

void loop() {

  MQTT_connect();


  Adafruit_MQTT_Subscribe *subscription;
  while ((subscription = mqtt.readSubscription(20000))) {
    if (subscription == &Light1) {
      Serial.print(F("Got: "));
      Serial.println((char *)Light1.lastread);
      int Light1_State = atoi((char *)Light1.lastread);
      digitalWrite(light, Light1_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!");

}
```

6.5: Setting up Amazon Alexa:

Install Amazon Alexa Application on your mobile phone using the same credentials you are given previous. Tap or say “Alexa” for example “Alexa, Trigger Light ON”. One of the Bulb will turn on and repeat the same for the other Bulb as well.



7. RESULT

Click on Serial Monitor **on** Arduino IDE **and set the** Baud Rate **to** 115200.