TABLE OF CONTENTS

1. OBJECTIVE OF THE PROJECT

We propose an automated home automation that works on speech processing. System eases the home automation task by listening to user's speech and switching appliances as per user spoken commands. Here we use a mic to record user's speech and transfer these commands to the Raspberry Pi through our circuitry. The Pi processor now processes user's speech to extract keywords related to load switching.

2. COMPONENTS REQUIRED

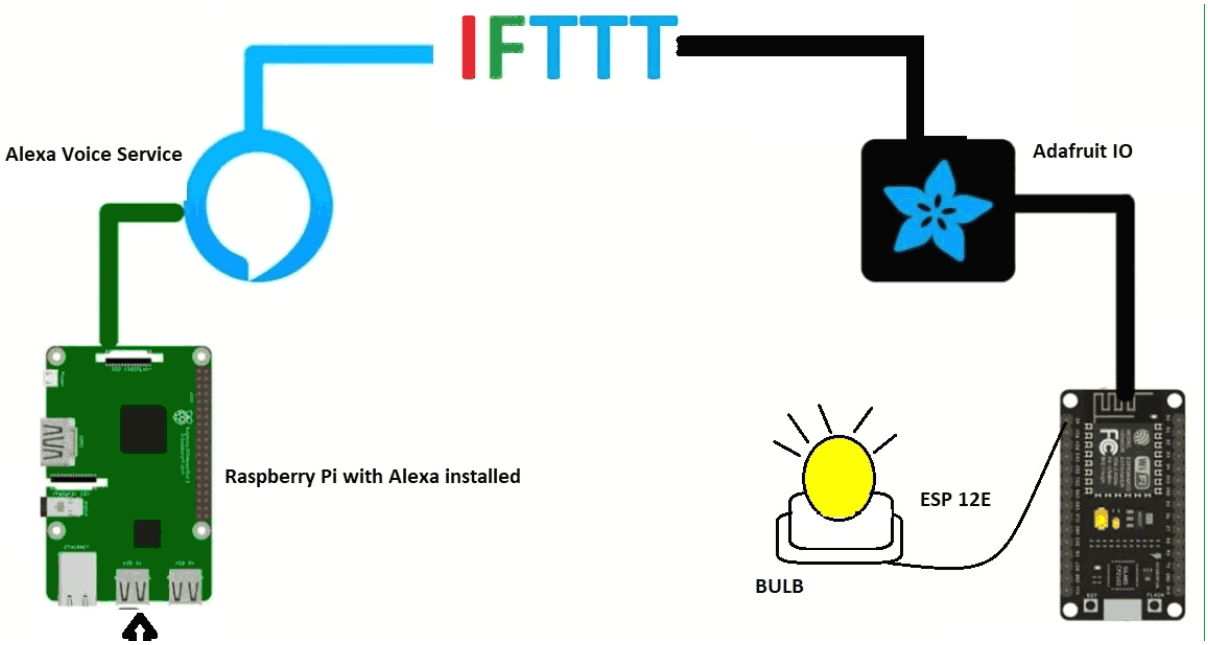
Raspberry Pi 3	LCD Display	Crystal Oscillator
Resistors	Capacitors	Transistors
Cables &	Connectors	Diodes
PCB	LEDs	Transformer/Adapter
Push Button	Push Button	External Speaker with 3.5 mm AUX
Cable	USB 2.0	Microphone
Relay Module		

Software Requirements include:

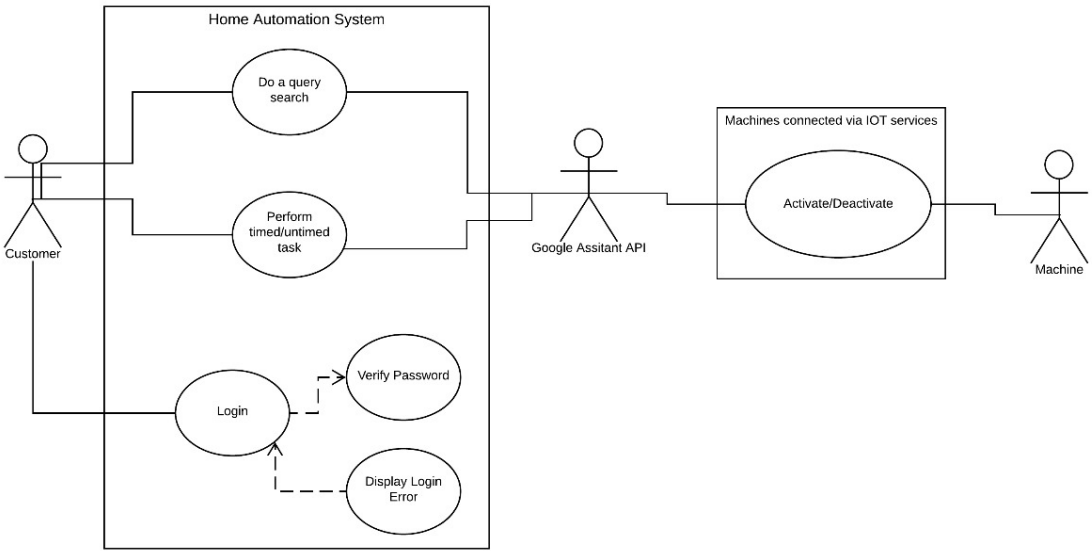
- **Linux/Windows OS**
- **Registered account with Alexa Voice Services**
- **Registered account with PubNub**
- **Registered account with IFTTT**

3. DIAGRAMS

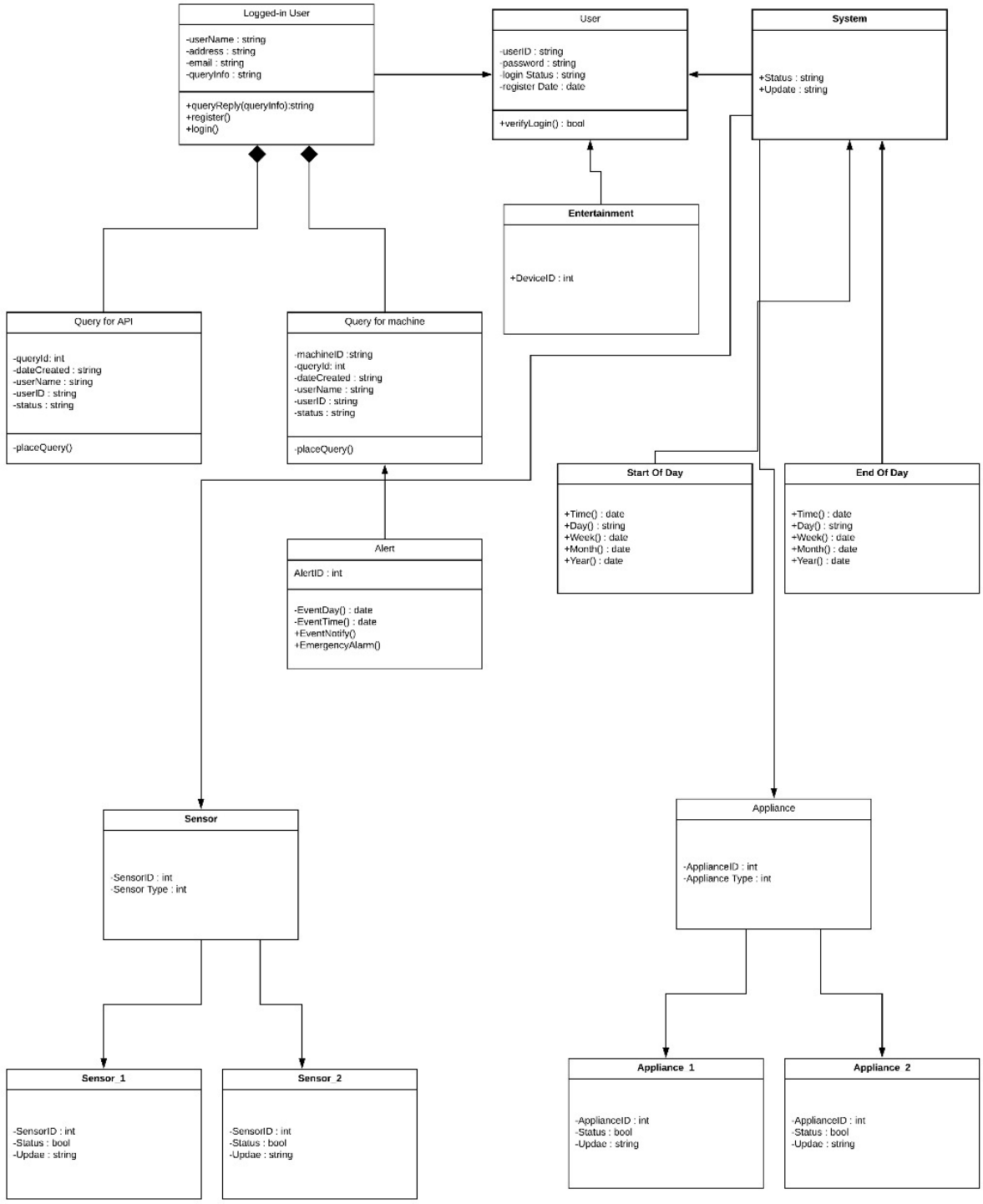
Flow Diagram:



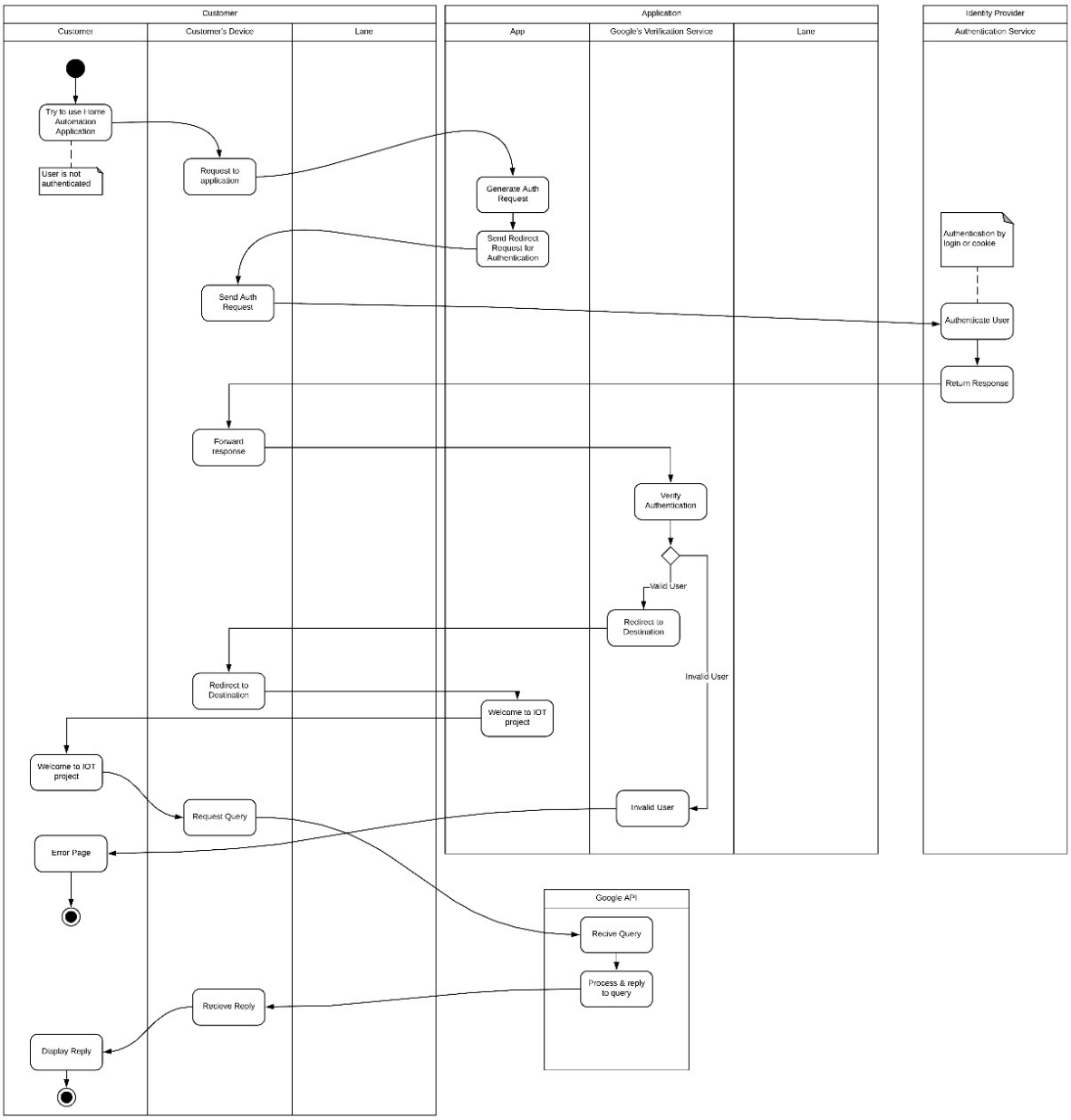
Use Case Diagram:



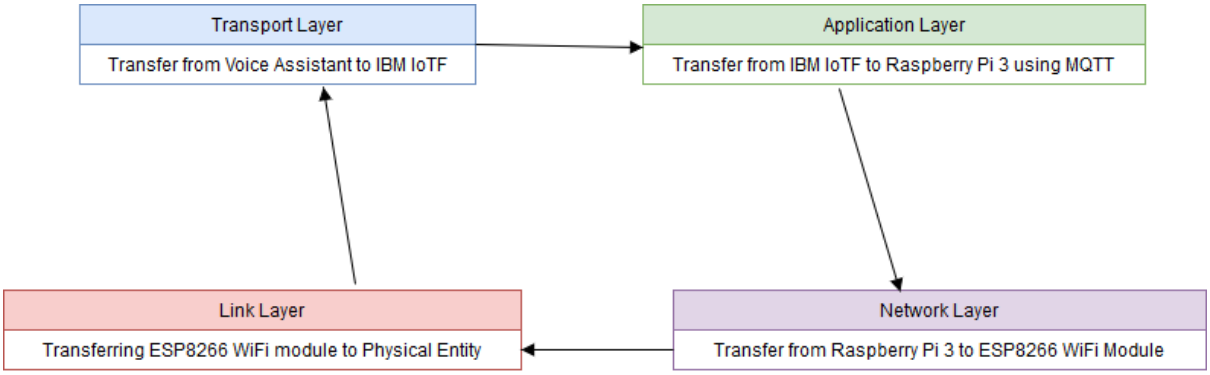
Class Diagram:



Activity Diagram:



Communication Model:



4. PROJECT DESCRIPTION

In our project, the process includes several steps.

- First, input is given to Raspberry Pi through the USB Mic.
- Now, this recording is sent to Alexa voice services and after voice recognition, AVS sent the data to IFTTT and it triggers the condition in IFTTT.
- According to the recipe, IFTTT will send the command to Adafruit IO which is the MQTT broker to perform an action.
- Then ESP12e will receive the data from Adafruit IO via MQTT protocol and LED will turn ON/OFF according to the command.

Overview of hardware components and service platforms

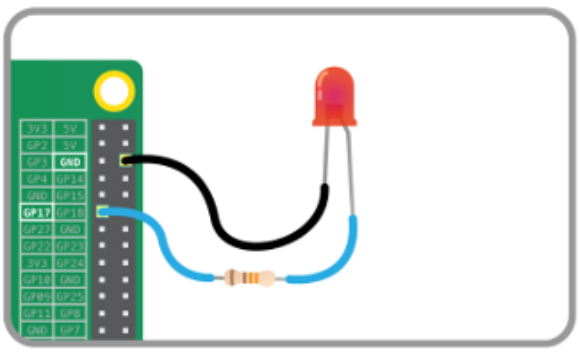
Raspberry Pi

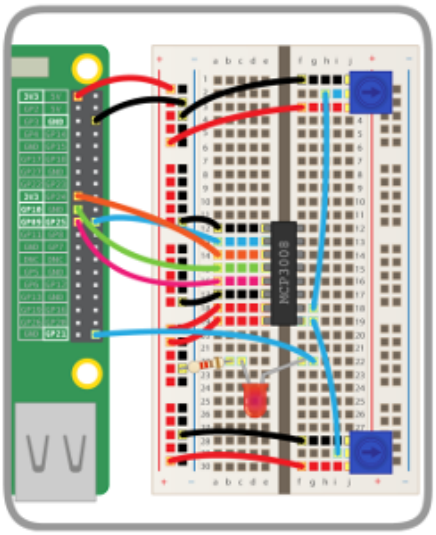
Raspberry Pi is the name of a series of single-board computers made by the Raspberry Pi Foundation, a UK charity that aims to educate people in computing and create easier access to computing education.

The Raspberry Pi launched in 2012, and there have been several iterations and variations released since then. The original Pi had a single-core 700MHz CPU and just 256MB RAM, and the latest model has a quad-core 1.4GHz CPU with 1GB RAM. The main price point for Raspberry Pi has always been $35 and all models have been $35 or less, including the Pi Zero, which costs just $5.

All over the world, people use Raspberry Pis to learn programming skills, build hardware projects, do home automation, and even use them in industrial applications.

The Raspberry Pi is a very cheap computer that runs Linux, but it also provides a set of GPIO (general purpose input/output) pins that allow you to control electronic components for physical computing and explore the Internet of Things (IoT).





NodeMCU ESP8266

NodeMCU is an open-source firmware and development kit that helps you to prototype or build IoT product. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266.

IFTTT Platform

FTTT derives its name from the programming conditional statement “if this, then that.” What the company provides is a software platform that connects apps, devices and services from different developers in order to trigger one or more automations involving those apps, devices and services.

- **How it works**

The automations are accomplished via applets — which are sort of like macros that connect multiple apps to run automated tasks. You can turn on or off an applet using IFTTT’s website or mobile apps (and/or the mobile apps’ IFTTT widgets). You can also create your own applets or make variations of existing ones via IFTTT’s user-friendly, straightforward interface.

- **IFTTT and Alexa**

An increasingly popular way to use IFTTT is in conjunction with Amazon’s Alexa voice assistant. Much of these applets center around internet-of-things use cases such as controlling smart home devices with voice commands directed at Echo and Echo Dot speakers. This could entail telling Alexa to make a cup of coffee with WeMo’s connected coffee maker or changing the color of Hue smart lights each time Alexa plays a new song.

Amazon is keen to push its A.I. assistant into corporate environs, and there are uses emerging for IFTTT and Alexa in the workplace context. For instance, when you ask Alexa to add a to-do item, this can automatically be added to a workspace within Asana’s project management app. The same can be done with Evernote, Google Docs spreadsheets and more. You can also sync your to-do list with Google Calendar.

It is also simple to connect IFTTT with Google’s Assistant, which powers its Google Home speakers.

Adafruit.IO

Adafruit.io is a *cloud service* - that just means we run it for you and you don't have to manage it. You 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!

IO includes client libraries that wrap their REST and MQTT APIs. IO is built on Ruby on Rails, and Node.js.

- **Triggers**

  Use triggers in Adafruit IO to control and react to your data. Configure triggers to email you when your system goes offline, react to a temperature sensor getting too hot, and publish a message to a new feed.

- **Integration with IFTTT and Zapier**

  Want to make your project react to an email, display trending tweets, or turn on the front lights when your pizza is on the way? We baked in integrations with IFTTT and Zapier to connect your project's sensors to hundreds of web services.

- **MQTT API Documentation**

  MQTT, or message queue telemetry transport, is a protocol for device communication that Adafruit IO supports. Using a MQTT library or client you can publish and subscribe to a feed to send and receive feed data.

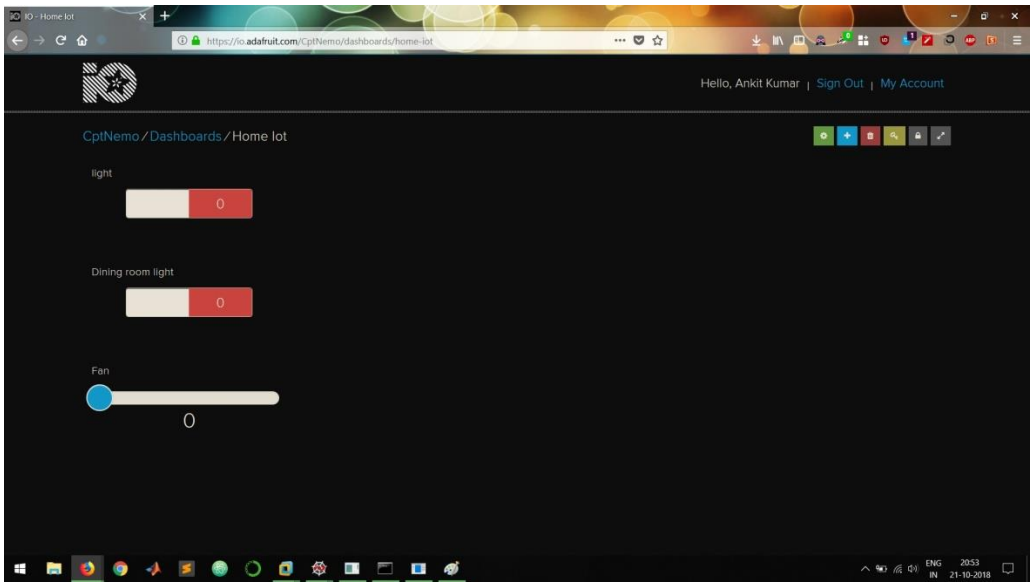
  To use the MQTT API that Adafruit IO exposes you'll need a MQTT client library. For Python, Node.js, and Arduino you can use Adafruit's IO client libraries as they include support for MQTT (see the client libraries section). For other languages or platforms look for a MQTT library that ideally supports the MQTT 3.1.1 protocol.

Setting Up the Project.

Setting Up an Adafruit account for Communication

First, we will make a feed in Adafruit IO. Feed stores the data sent by IFTTT. To make feed follow these steps:

- Log in to **Adafruit IO** with your credentials or Sign up if you don't have an account.
- Click on My account -> Dashboard
- Give name and description to your feed and click on Create.
- Click on **Key** button and note down the AIO Keys, we will use this key in our code.
- Click on '+' button to create a new block and click on **Toggle button**.
- Now, Enter Name of Feed and click on Create. Then Select the feed and click on Next step.
- In block settings, Write '1' in Button ON text field and '0' in Button OFF Text field.
- Your Feed is successfully created.

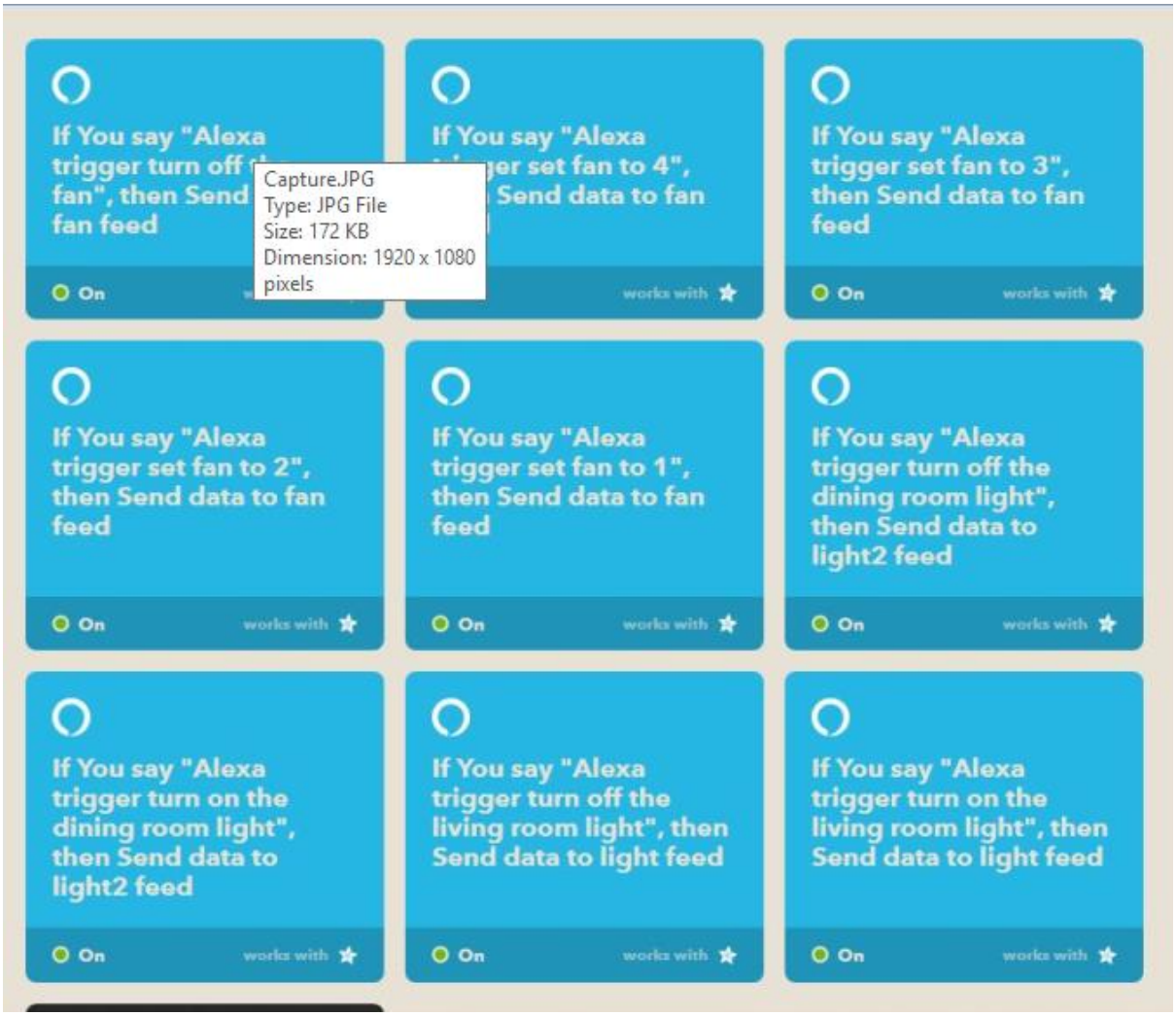


Linking Alexa to Adafruit IO using IFTTT

Follow these steps to make an Applet/Recipe in IFTTT:

- Login to IFTTT with your credentials or Sign Up if you don't have an account on it.
- On My Applets, Click on **New Applet.**
- Click on +**this**
- Search Amazon Alexa and click on it, sign in with your amazon developer account details.
- Choose the trigger, **Say a specific phrase.**
- Provide ***"turn on the light"*** as the phrase, click on **Create Trigger**.
- Click on **+that**
- Search *Adafruit* and click on it.
- Login to Adafruit Account using your credentials. Click on ***Send data to Adafruit***. Select feed name that you just created in Adafruit IO. Now, give '1' in data to save, this implies LED will be ON. Click on ***Create Action***.
- Follow the same steps to *make applets to turn the LED OFF*. Just put '0' in the Data to save field. All steps remain the same.

Trigger phrases added in our project are:



5. PROJECT CODE

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

#define WLAN_SSID       "cptNemo"
#define WLAN_PASS       "Valarr@Anm%"
#define AIO_SERVER      "io.adafruit.com"
#define AIO_SERVERPORT  1883
#define AIO_USERNAME    "CptNemo"
#define AIO_KEY         "a377d9597b8442199c8b8130f773d89d"

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/light");
Adafruit_MQTT_Subscribe light2 = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME
"/feeds/light2");
Adafruit_MQTT_Subscribe fan   = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/fan");

void MQTT_connect();

void setup() {
  Serial.begin(115200);
  delay(10);
  pinMode(D0, OUTPUT);
  pinMode(D1, OUTPUT);
  pinMode(D2, OUTPUT);

  Serial.println(F("Adafruit MQTT demo"));

  // 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());

  // Setup MQTT subscription for light and fan feed.
  mqtt.subscribe(&light);
  mqtt.subscribe(&light2);
  mqtt.subscribe(&fan);
}
```

```
uint32_t x=0;

void loop() {
  MQTT_connect();

  Adafruit_MQTT_Subscribe *subscription;
  while ((subscription = mqtt.readSubscription(5000))) {
    if (subscription == &light) {
      Serial.print(F("Got_light: "));
      Serial.println((char *)light.lastread);
      uint16_t num = atoi((char *)light.lastread);
      digitalWrite(D0,num);
    }

       if (subscription == &fan){
      uint16_t num1 = atoi((char *)fan.lastread);
      analogWrite(D1, num1);
    }
        if (subscription == &light2) {
      Serial.print(F("Got_light: "));
      Serial.println((char *)light2.lastread);
      uint16_t num2 = atoi((char *)light2.lastread);
      digitalWrite(D2,num2);
    }

  }
}
void MQTT_connect() {
  int8_t ret;

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

  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) {
         // basically die and wait for WDT to reset me
         while (1);
       }
  }
  Serial.println("MQTT Connected!");
}
```

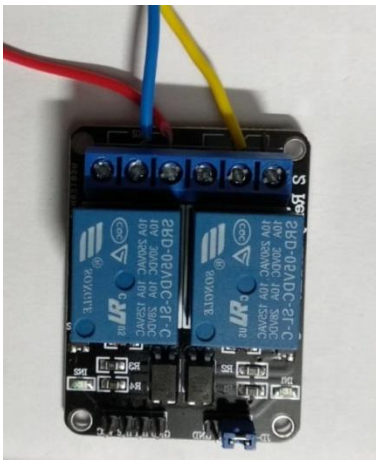
6. PROJECT OUTPUT

Project Components:

1. Raspberry Pi 3

2. WiFi Module

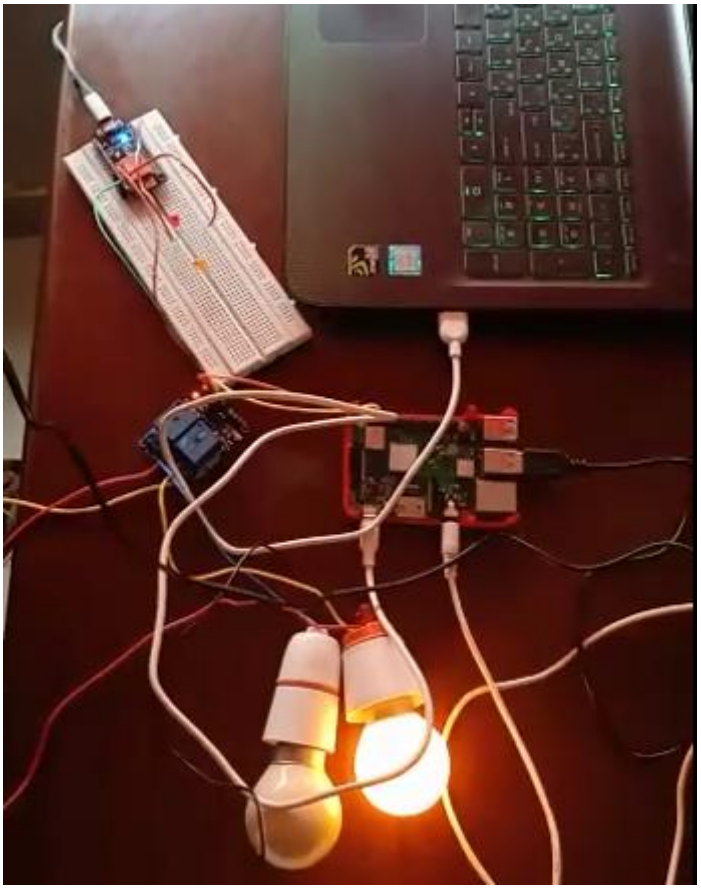
3. Relay Module



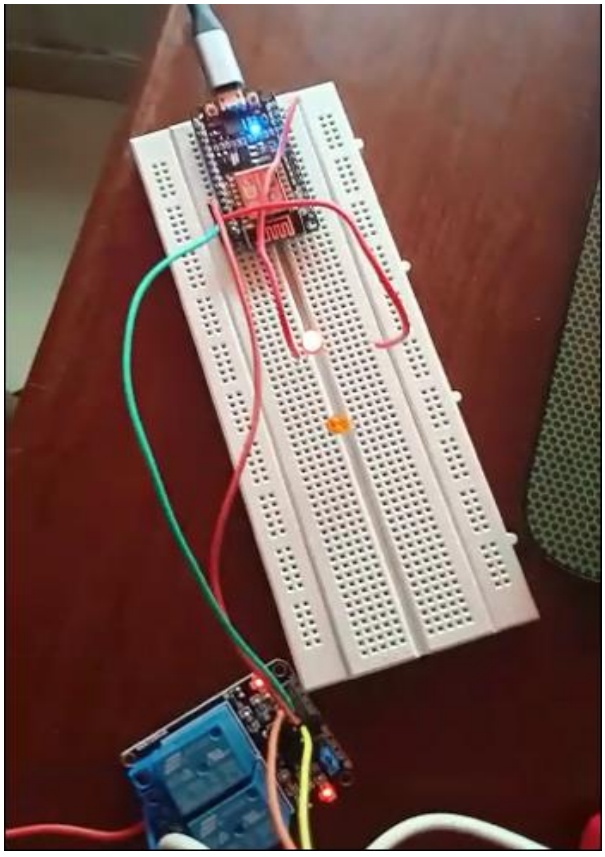
4. **USB Microphone**

5. **AC Bulb**

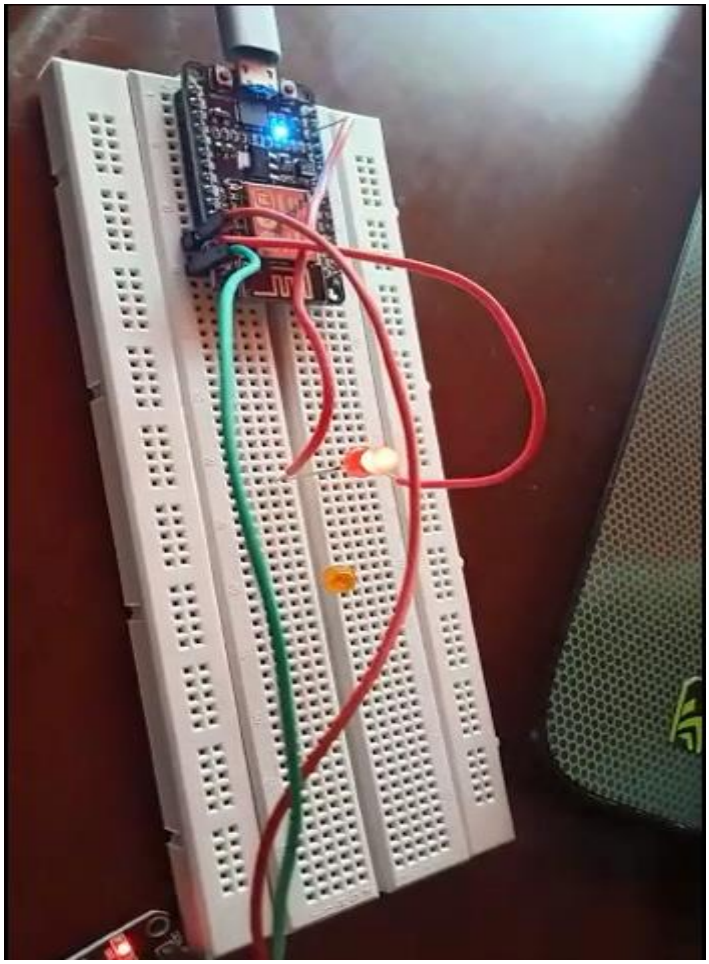
Turning on the light:



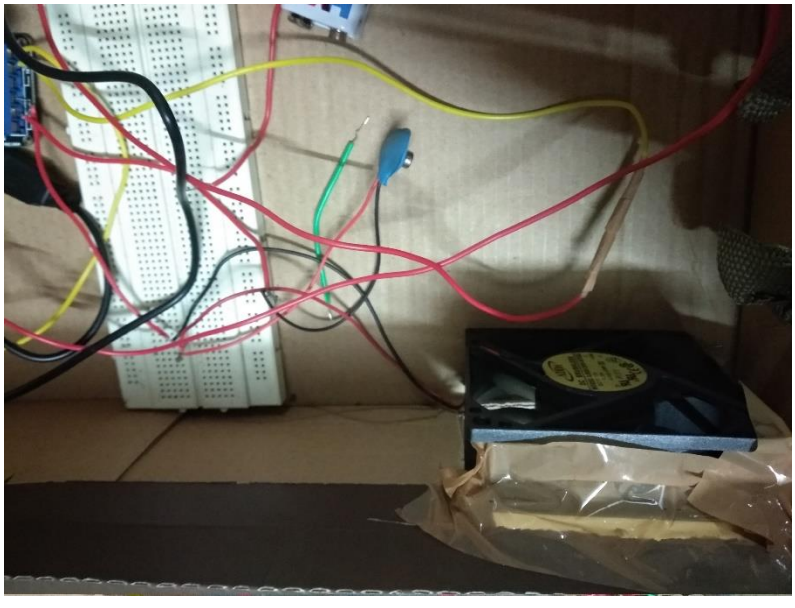
Switching ON Fan (here, simulated as bulb):



Decreasing Fan Speed (here, simulated as bulb's brightness):



Controlling Fan(DC)



7. INFERENCE

After connecting ESP-12E with the laptop and uploading below code with edited credentials in the code.

We Connect an LED or relay to pin D0. Alexa service is running on RPi.

To give any command, it is needed to wake up Alexa service by calling “Alexa” each time we wanted to send a command.

Then we will hear a beep sound. Then, say “*Alexa Trigger Turn on the Light.*”

We can see the light turns ON within a moment. And then if we say “*Alexa Trigger Turn off the Light*”, the light should turn OFF.

Therefore, the project was successfully implemented using Alexa Voice Assistant on home Automation System.

8. CONCEPTS LEARNED

Following concepts, we learned in our project:

- Setting up the Raspberry Pi with SSH and VNC
- Set up Amazon Developer Account and Configuring Alexa Voice Services on Raspberry Pi
- Setting up IFTTT for Amazon Alexa Service.
- Working of MQTT protocol.
- Setting up MQTT broker.
- NodeMcu working and application

9. REFRENCES

- https://circuitdigest.com/microcontroller-projects/iot-based-alexa-voice-controlled-led-using-raspberry-pi-and-esp12
- https://opensourceforu.com/2016/11/mqtt-get-started-iot-protocols/
- https://www.raspberrypi.org/magpi/ssh-remote-control-raspberry-pi/
- https://developer.amazon.com/docs/alexa-voice-service/set-up-raspberry-pi.html
- https://learn.adafruit.com/adafruit-io/getting-started
- https://www.electronicwings.com/nodemcu/introduction-to-nodemcu
- https://help.ifttt.com/hc/en-us