Pabna University of Science and Technology

DEPARTMENT OF ELECTRONIC AND TELECOMMUNICATION ENGINEERING

A Project on Voice Control Home Automation

COURSE TITLE- PROJECT

COURSE CODE- ETE 3000

REPORT ON FINAL PROJECT

SUBMITTED BY	SUPERVISED BY
SAJEEB CHANDRA DAS	Liton Chandra Paul
Roll: 150542	Assistant Professor
3rd Year 1st Semester	Department of Electronic and Telecommunication Engineering, PUST
Session 2014-15	

Date: - -

Pabna University of Science and Technology

DEPARTMENT OF ELECTRONIC AND TELECOMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the project entitled

“Voice Control Home Automation”

SUBMITTEDBY

SAJEEB CHANDRA DAS

Roll: 150542

The total project work carried out by him under my supervision and guidance.

(Supervisor)

Liton Chandra Paul

Assistant Professor

Department of Electronic and Telecommunication Engineering, PUST

CONTENT

ACKNOWLEDGEMENT

At first, I would like to thanks and give deep respect to my honorable supervisorLiton Chandra Paul, Assistant Professor, Department of Electronic and Telecommunication Engineering, PUST for giving me the opportunity to work to under his supervision, continuous guidance as well as constructive suggestions, advice and support me throughout the entire course of the work.

ABSTRACT

This paper presents a proposal for home automation using voice via Google Assistant. Home automation has been evolving drastically. We saw many home automation technologies introduced over these years from Zigbee automation to Amazon Echo, Google Home and Home from Apple. It has become a craze these days. Google Home price is around 12,000 TK(BDT) with an additional cost of the devices to be connected to, the total cost of the system reaches over 20,000 TK (BDT). Apple Home Kit too is pretty more expensive, over 8,500 TK(BDT) more than the Google Home just for a basic setup. Philips Hue, a smart light which is controlled by the Google Assistant, Amazon Echo and Siri, voice assistant by Apple is priced around 12,000 TK(BDT). So, overall we can see here that to make our home smart we need to invest quite a lot, let’s say some 20,000 TK (BDT) for a basic setup. What if we can automate our house within (cost of the Smartphone is not included as it is assumed to be owned by every individual these days) 1,500 TK (BDT) and can control up to 4 appliances using Google Assistant? Well, this paper describes the implementation of such a system. The system is implemented using ordinary household appliances Natural language voice commands are given to the Google Assistant and with the help of IFTTT (If This Then That) application and the Adafruit IO virtual switchboard application where the commands are decoded and then sent to the microcontroller, the microcontroller in turn controls the relays connected to it as required, turning the device connected to the respective relay On or OFF as per the users request to the Google Assistant. The microcontroller used is NodeMCU (ESP8266) and the communication between the microcontroller and the application is established via Wi-Fi (Internet).

Key Words: Home Automation, NodeMCU (ESP8266), IFTTT (If This Than That) Application, Adafruit IO, Internet of Things (IoT), Google Assistant, Voice Control, Smartphone.

INTRODUCTION

Home, it is the place where one fancies or desires to be after a long tiring day. People come home exhausted after a long hard working day. Some are way too tired that they find it hard to move once they land on their couch, sofa or bed. So any small device/technology that would help them switch theirs lights on or off, or play their favorite music etc. on a go with their voice with the aid of their smart phones would make their home more comfortable.

Moreover, it would be better if everything such as warming bath water and adjusting the room temperature were already done before they reach their home just by giving a voice command. So, when people would arrive home, they would find the room temperature, the bath water adjusted to their suitable preferences, and they could relax right away and feel cozier and rather, feel more homely.

Human assistants like housekeepers were a way for millionaires to keep up their homes in the past.However, not everyone is wealthy enough to be able to afford a human assistant, or some smart home kit. Hence, the need for finding an inexpensive and smart assistant for normal families keeps growing.

This paper proposes such inexpensive system. It uses the Google Assistant, the IFTTT application, the Adafruit IO application and the NodeMCU(ESP8266) microcontroller as the major components along with a relay board comprising of 4 relays along with LM 7805 IC. Natural language voice is used to give commands to the Google Assistant. All of the components are connected over the internet using Wi-Fi which puts this system under the IoT.

OBJECTIVES

- To control home appliances by voice commands with Google Assistant.
- To study how it works.

APPARATUS

1. ESP8266 NodeMCU.
2. 4 channel 5V Relay Module.
3. Breadboard.
4. Jumper Wires.
5. 9V Battery.
6. LM 7805 IC.
7. Electrical Switchboard.
8. Electric Wires.
9. Light(bulb).
10. Plugs.
11. Sockets.
12. AC Supply.

CIRCUIT DIAGRAM

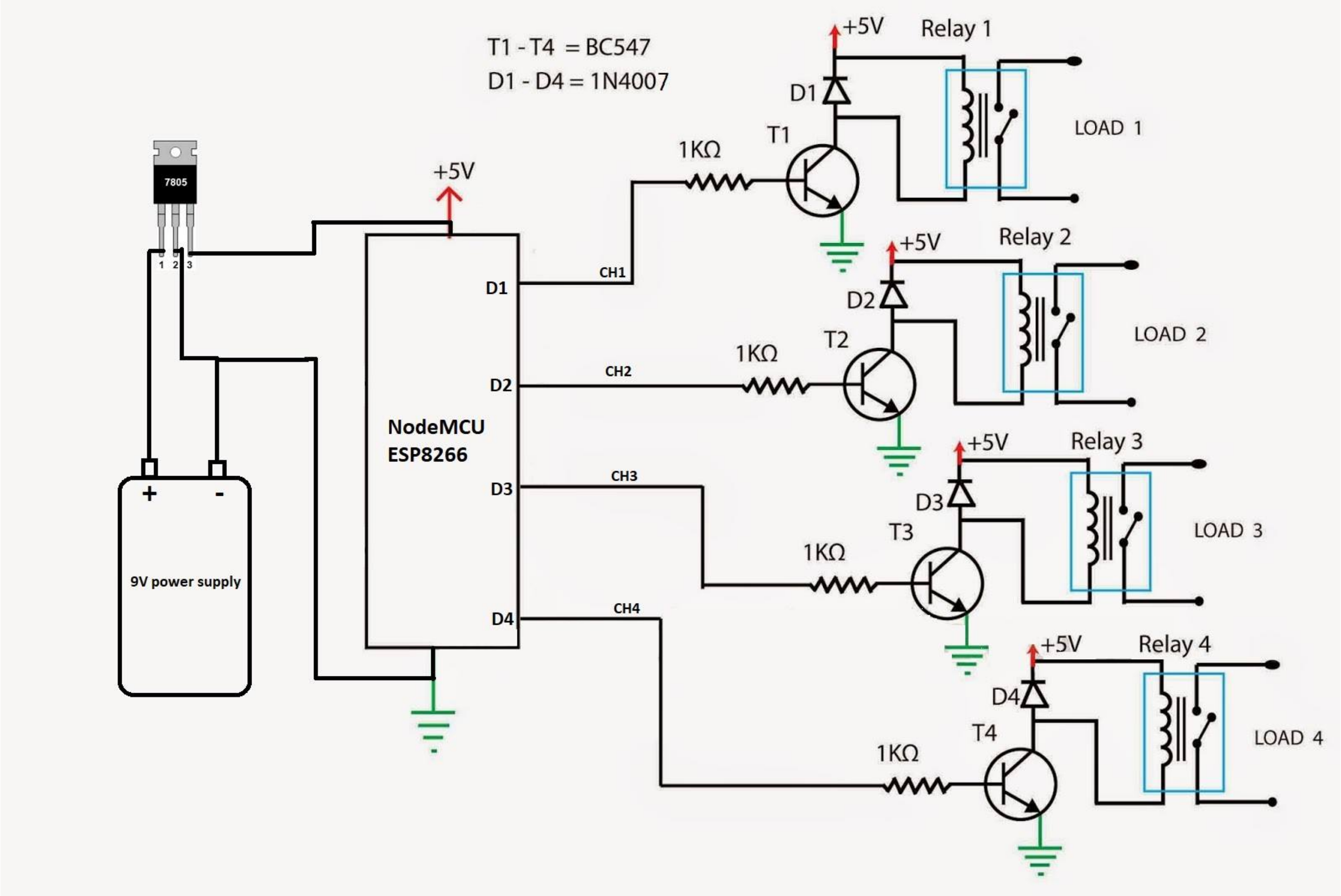


Figure: Circuit diagram of Voice Control Home Automation.

The Description of the Major Part of Voice Control Home Automation System Design

Internet of Things: The Internet of things (IoT) is the network of physical devices, vehicles, home appliances and other items embedded with electronics, software, sensors, actuators, and connectivity which enable these objects to connect and exchange data. Each thing is uniquely identifiable through its embedded computing system but is able to inter-operate within the existing Internet infrastructure.

The system design is broken down into two main categories,

i. The hardware- It has the capability to connect to the router. It would also be able to turn on/off specified devices, such as lights and fans etc. It is called the ‘Control Unit’. And,

ii. The Software- Adafruit IO, IFTTT and the Google Assistant constitute the software of the design and these applications would be integrated in the Android device.

The Control Unit comprises of the microcontroller- NodeMCU, LM7805 IC and the 4 Channel Relay board. Adafruit IO via IFTTT communicates with the microcontroller and sends the desired signal via the internet. Figure 1 below shows the basic system design architecture.

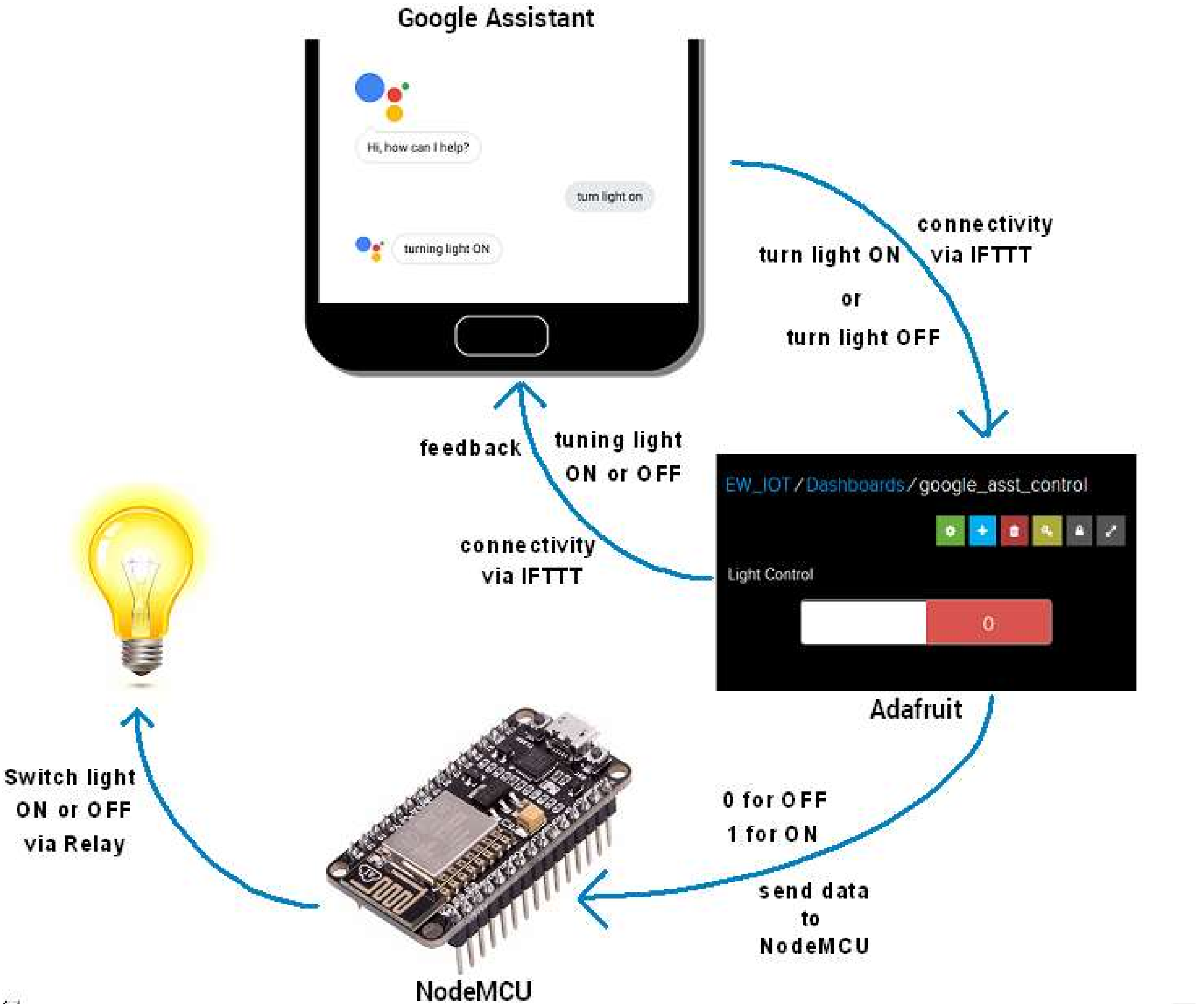


Fig -1: Basic System Architecture

The hardware also called the Control Unit comprises of the NodeMCU microcontroller and the Realy board. NodeMCU's digital output pins are connected to the Relay pins of the Relay board. Finally, each Relay is connected to an appliance.

- NodeMCU (ESP8266) :

The NodeMCU (Node Microcontroller Unit) is an open source software and hardware development environment that is built around a very inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266 is designed and manufactured by Express, contains all crucial elements of the modern computer: CPU, RAM, networking (wi-fi), and even a modern operating system and SDK. When purchased at bulk, the ESP8266 chip costs only 500TK (BDT) a piece. That makes it an excellent choice for this system design.

The NodeMCU aims to simplify ESP8266 development. It has two key components.

i. An open source ESP8266 firmware that is built on top of the chip manufacturer's proprietary SDK. The firmware provides a simple programming environment based on eLua (embedded Lua), which is a very simple and fast scripting language with an established developer community. For new comers, the Lua scripting language is easy to learn. And to add on NodeMCU can be programmed with the Android IDE too.

ii. A development kit board that incorporates the ESP8266 chip on a standard circuit board. The board has a built-in USB port that is already wired up with the chip, a hardware reset button, Wi-Fi antenna, LED lights, and standard-sized GPIO (General Purpose Input Output) pins that can plug into a bread board. Figure 2 below shows the NodeMCU development board.

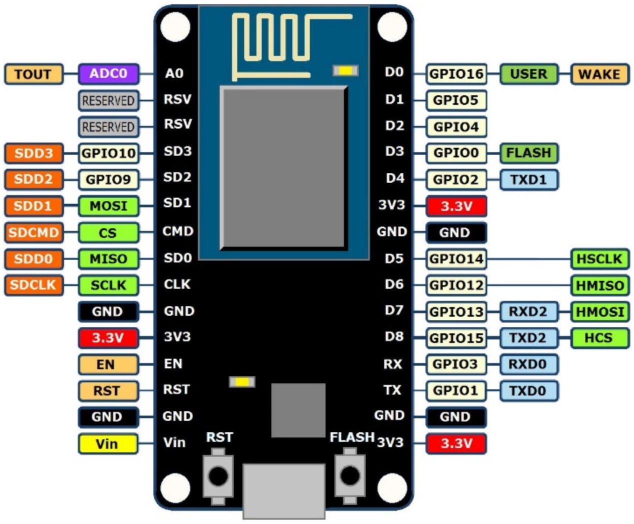


Fig -2: NodeMCU (ESP8266) Development Board.

- RELAY BOARD:

A relay is an electromagnetic switch. It is activated when a small current of some microampere is applied to it.Normally a relay is used in a circuit as a type of switch, an automatic switch. There are different types of relays and they operate at different voltages. When a circuit is built the voltage that will trigger it has to be considered. In this system the relay circuit is used to turn the appliances ON/OFF. The high/low signal is supplied from the NodeMCU microcontroller. When a low voltage is given to the relay of an appliance it is turned off and when a high voltage is given it is turned on. The relay circuit to drive four appliances in the Home automation system is shown below in figure 3. The number of appliances can be modified according to the user’s requirements.

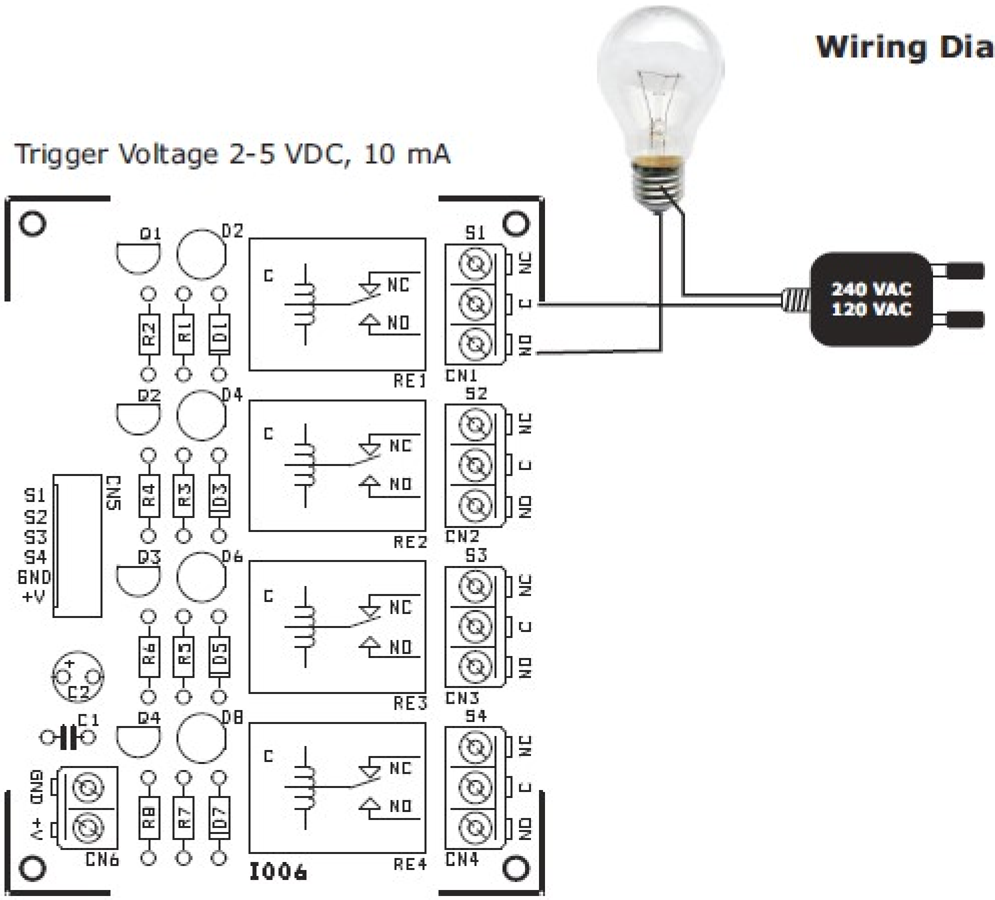


Fig-3: Relay module.

- IC 7805:

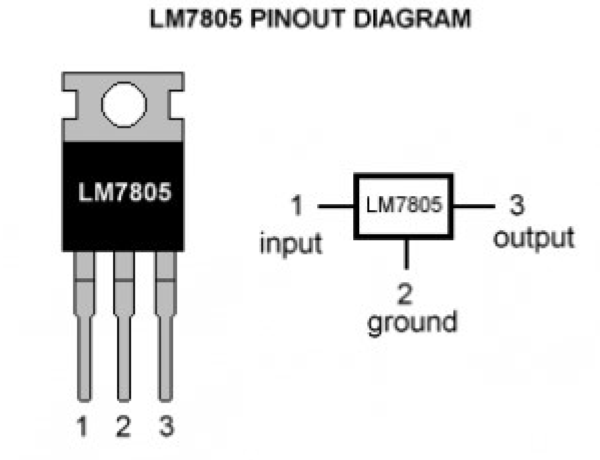
Voltage sources in a circuit may have fluctuations resulting in not providing fixed voltage outputs. A voltage regulator IC maintains the output voltage at a constant value. 7805 IC, a member of 78xx series of fixed linear voltage regulators used to maintain such fluctuations, is a popular voltage regulator integrated circuit (IC). The xx in 78xx indicates the output voltage it provides. 7805 IC provides +5 volts regulated power supply with provisions to add a heat sink.

7805 IC Rating:

Input voltage range 7V- 35V

Current rating Ic = 1A

Output voltage range V_{Max}=5.2V, V_{Min}=4.8V

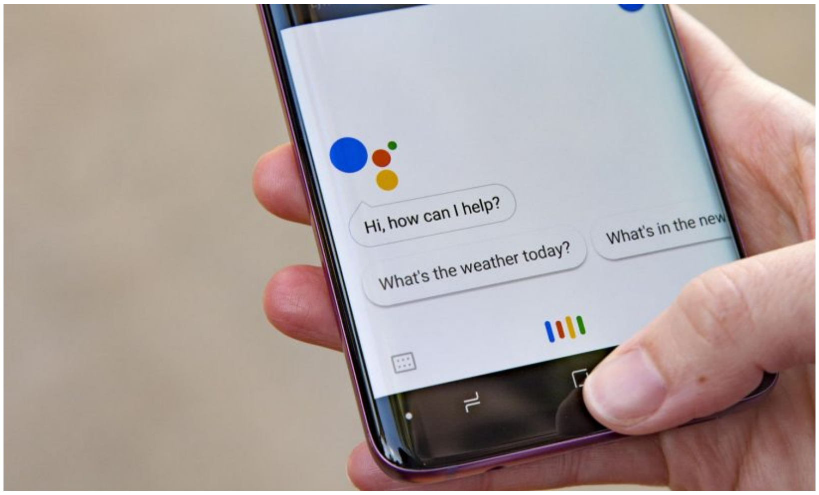


The software of the system proposed consists of mainly Google Assistant, Adafruit IO and the IFTTT application.

- Google Assistant:

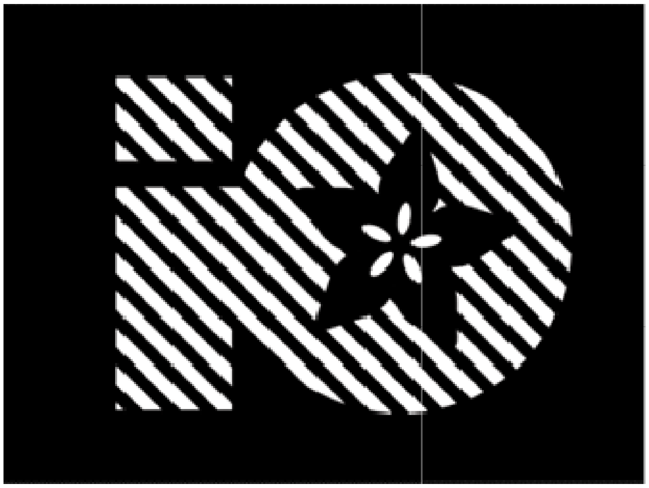
Google Assistant is an artificial intelligence-powered virtual assistant developed by Google that is primarily available on mobile and smart home devices. Unlike the company's previous virtual assistant, Google Now, Google Assistant can engage in two-way conversations. Assistant initially debuted in May 2016 as part of Google's messaging app Allo, and its voice-activated speaker Google Home. After a period of exclusivity on the Pixel and Pixel XL smartphones, it began to be deployed on other Android devices in February 2017, including third-party smartphones and Android Wear (now Wear OS). The Assistant has been, and is being, further extended to support a large variety of devices, including cars and third party smart home appliances. The functionality of the Assistant can also be enhanced by third-party developers. In 2017, Google Assistant was installed on more than 400 million devices.

Users primarily interact with Google Assistant through natural voice, though keyboard input is also supported.The Assistant is able to search the Internet, schedule events and alarms, adjust hardware settings on the user's device, and show information from the user's Google account. Google Assistant is able to identify objects and gather visual information through the device's camera, and support purchasing products and sending money, as well as identifying songs.



- Adafruit IO:

Adafruit IO is an IOT platform built around the Message Queue Telemetry Transport (MQTT) Protocol. MQTT is a lightweight protocol that allows multiple devices to connect to a shared server, called the MQTT Broker, and subscribe or write to user defined topics. When a device is subscribed to a topic, the broker will send it a notification whenever that topic changes. MQTT is best suited for applications with low data rates, strict power constraints, or slow Internet connections. In addition to providing the MQTT Broker service, Adafruit IO also allows you to set up dashboards that let you directly manipulate or view the current value of each topic. Since it can be accessed from a web browser, it makes it the ideal hub for monitoring and controlling all of your various IOT projects.After creating your Adafruit IO account, you should be taken to the homescreen. Select "Feeds" from the left-hand menu. Click the Actions drop-down menu, and create a new feed. I called mine "on-off". Next, go to Dashboards in the left-hand menu. Click the Actions drop-down menu, and create a new dashboard. I called mine "LightSwitch". Open the new dashboard, and you should be taken to a mostly blank page. Pressing the blue + button will let you add new UI components to the dashboard. For now, all we'll need is a toggle button, which should the first option. When prompted to choose a feed, select the one you just made, and keep the defaults for the rest of the settings.

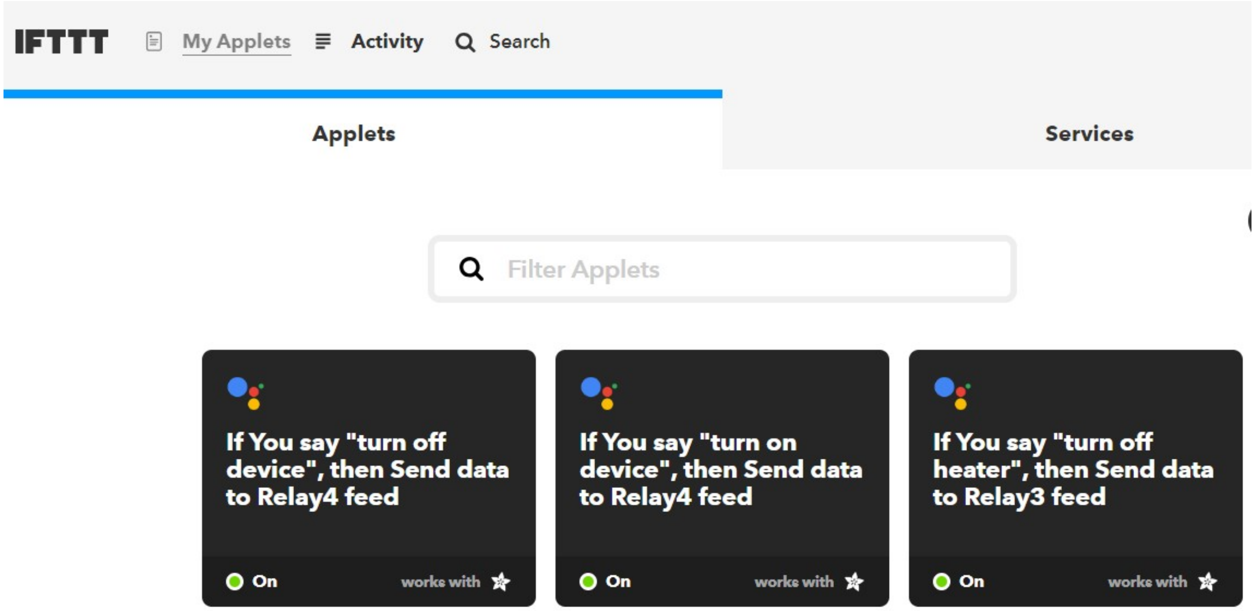


- IFTTT:

IFTTT is an initialism for "If This Then That" is a free web-based service to create chains of simple conditional statements, called applets.An applet is triggered by changes that occur within other web services such as Gmail, Facebook, Telegram, Instagram, and Pinterest.For example, an applet may send an e-mail message if the user tweets using a hashtag, or copy a photo on Facebook to a user's archive if someone tags a user in a photo.

IFTTT is both a website and a mobile app that launched in 2010 and has the slogan "Put the Internet to work for you". The idea is that you use IFTTT to automate everything from your favorite apps and websites to app-enabled accessories and smart devices. 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. Here, IFTTT application is used to bridge the gap between the Google Assistant commands and Adafruit IO.

IFTTT users created about 20 million recipes each day.



If You say "turn off device", then Send data to Relay4 feed

If You say "turn on device", then Send data to Relay4 feed

If You say "turn off heater", then Send data to Relay3 feed

CODE

```
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
#define Relay1          D1
#define Relay2          D2
#define Relay3          D3
#define Relay4          D4
#define WLAN_SSID       "PUST-Students"      // Your SSID
#define WLAN_PASS       "PUST#2008"       // Your password
/************ Adafruit.io Setup ***********/
#define AIO_SERVER      "io.adafruit.com"
#define AIO_SERVERPORT  1883                  // use 8883 for SSL
#define AIO_USERNAME    "sajeebchandra"            // Replace it with your username
#define AIO_KEY         "0be482a2c98749cca404335cad09de90"   // Replace with your Project Auth Key
/************ Global State (you don't need to change this!) ************/
// Create an ESP8266 WiFiClient class to connect to the MQTT server.
WiFiClient client;
// or... use WiFiFlientSecure for SSL
```

```
//WiFiClientSecure client;
// Setup the MQTT client class by passing in the WiFi client and MQTT server and login
details.
Adafruit_MQTT_Clientmqtt(&client, AIO_SERVER, AIO_SERVERPORT,
AIO_USERNAME, AIO_KEY);
/********************** Feeds **********************/
// Setup a feed called 'onoff' for subscribing to changes.
Adafruit_MQTT_Subscribe Light1 = Adafruit_MQTT_Subscribe(&mqtt,
AIO_USERNAME"/feeds/Relay1"); // FeedName
Adafruit_MQTT_Subscribe Light2 = Adafruit_MQTT_Subscribe(&mqtt,
AIO_USERNAME "/feeds/Relay2");
Adafruit_MQTT_Subscribe Light3 = Adafruit_MQTT_Subscribe(&mqtt,
AIO_USERNAME "/feeds/Relay3");
Adafruit_MQTT_Subscribe Light4 = Adafruit_MQTT_Subscribe(&mqtt,
AIO_USERNAME "/feeds/Relay4");
voidMQTT_connect();
void setup() {
Serial.begin(115200);
pinMode(Relay1, OUTPUT);
pinMode(Relay2, OUTPUT);
pinMode(Relay3, OUTPUT);
pinMode(Relay4, 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());
  // Setup MQTT subscription for onoff feed.
```

```
mqtt.subscribe(&Light1);
mqtt.subscribe(&Light3);
mqtt.subscribe(&Light2);
mqtt.subscribe(&Light4);
}
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(Relay1, Light1_State);}
if (subscription == &Light2) {
Serial.print(F("Got: "));
Serial.println((char *)Light2.lastread);
int Light2_State = atoi((char *)Light2.lastread);
digitalWrite(Relay2, Light2_State);}
if (subscription == &Light3) {
Serial.print(F("Got: "));
Serial.println((char *)Light3.lastread);
int Light3_State = atoi((char *)Light3.lastread);
digitalWrite(Relay3, Light3_State);
   }
if (subscription == &Light4) {
Serial.print(F("Got: "));
Serial.println((char *)Light4.lastread);
int Light4_State = atoi((char *)Light4.lastread);
digitalWrite(Relay4, Light4_State);
     }
}
}
voidMQTT_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!");
  }
```

WORKING PRINCIPLE

Relay Connections:

Connect **NodeMCU** Ground (**GND**) pin to **-ve** pin of **Relay**.

Connect **NodeMCU** Supply **(5V**) pin to **+ve** pin of **Relay**.

Connect **NodeMCU** Digital pin (**D1**) to **Input** pin of **Relay Channel 1**.

Connect **NodeMCU** Digital pin (**D2**) to **Input** pin of **Relay Channel 2**

Connect **NodeMCU** Digital pin (**D3**) to **Input** pin of **Relay Channel 3**

Connect **NodeMCU** Digital pin (**D4**) to **Input** pin of **Relay Channel 4**

COM - Common connection--> it is the center terminal, It is hot as power to the load is connected at this terminal.

NO Normally open ---> It acts like a switch, since it is open - there will be no contact between COM and NO, When we trigger the relay module, it connects to COM by the electromagnet inside the relay and supply to the load is provided, which powers up the light. Thus the circuit is closed until we trigger the state to low in relay.

NC Normally closed---->It is always in contact with COM, even when relay is not powered. When we trigger the relay it opens the circuit, so the connection is lost. It behaves just opposite to NO.

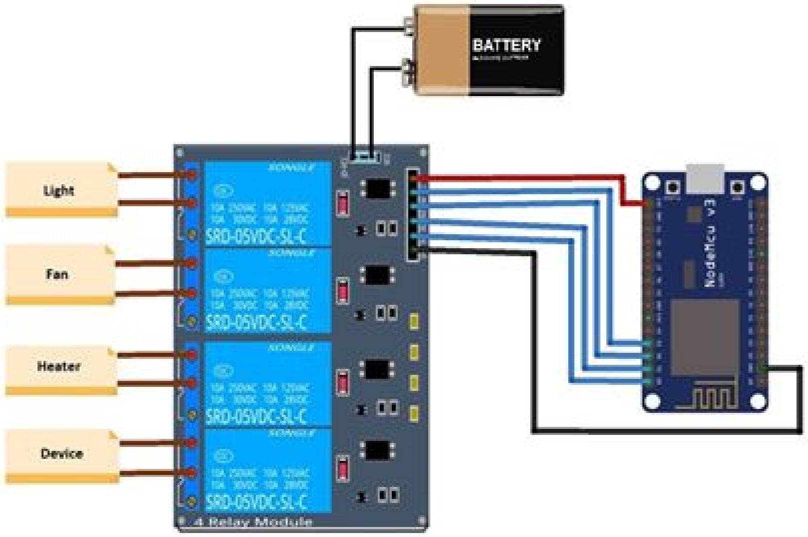
I'm using NO connection, but here in this type of relay "HIGH" state in code turns off the relay (opens the circuit). "LOW" state in code turns on the relay.

We are using normally open connection in relay. So that we can trigger on and off the light.

Hot line from supply is connected to COM

Supply line to the Ac light is connected to NO

Gnd or - or other terminal in light is connected directly.



Accounts

1. I go to platform.ifttt.com and sign up with the same Gmail account I use for my Google Assistant.
2. Next, I sign up with Adafruit IO at io.adafruit.com, again using the same Gmail ID.

NOTE: It is very important that both IFTTT, Adafruit.IO and Google Assistant accounts use the same Gmail ID.

Setting up Adafruit IO to talk to NodeMCU Board

Adafruit IO is a free MQTT server where I can publish or subscribe to a topic. If we want to learn more about MQTT, visit this blog: IOT Setup Using ESP8266 and HiveMQ Public MQTT

I signed in to Adafruit IO with the account I just created. Next, select Feeds from the left side menu. I clicked Actions drop down and create a new feed with the name "**onoff**".

I go to Dashboards from the left side menu. I created a dashboard with the name "**Lights**". Next, Click on Lights (The dashboard I created just now) and click on the + button (new block) on the right.

I choose toggle and select the feed I created ("onoff"). Now, named the block and without changing anything else, clicked create block.

Created another feed called "LightsStatus" and a text block in the dashboard Lights for NodeMCU to send status.Go to the previous page (Dashboards) and click View AIO key from the left side. It will shows my secret AIO key and username. I Note down my AIO key and username – this will be used in the program that needs to be uploaded to NodeMCU.

Adafruit IO Test

To test the connection to Adafruit IO, upload the code to NodeMCU and open the serial monitor. Wait till it connects to myWiFi and the mqtt server. When it's done connecting, go to Lights Dashboard in Adafruit IO webpage and try toggling the switch.

If the LED turns on and off when I toggle the switch, we are all set for the next step.

IFTTT Setup

I signed in to my new IFTTT account.

I have visited to https://ifttt.com/discover and click my applets from the top. Now, clicked New Applet and search for Google Assistant in the search box and I saw "**if this then that**" written in a big font. I noticed that the word **this** is written in blue color and it has a + sign on the left. I clicked on it.

When the page loaded, I searched Google Assistant in the search box and selected Google Assistant when it appears. I clicked on "Say a simple phrase". It has an option to place a maximum of three phrases to send a command. At least one is mandatory. The other two can be variations of the same phrase.

We can choose any phrase like "***turn on the light***" or "***switch on the light***". This phrase will be the one I will be saying to Google Assistant on my phone to turn the LED on. Now I filled the fourth box (Google Assistant's response). This will be something that I like Google Assistant to send in as a response to my "turn on light" command – E.g. "Ok. I'll do that".

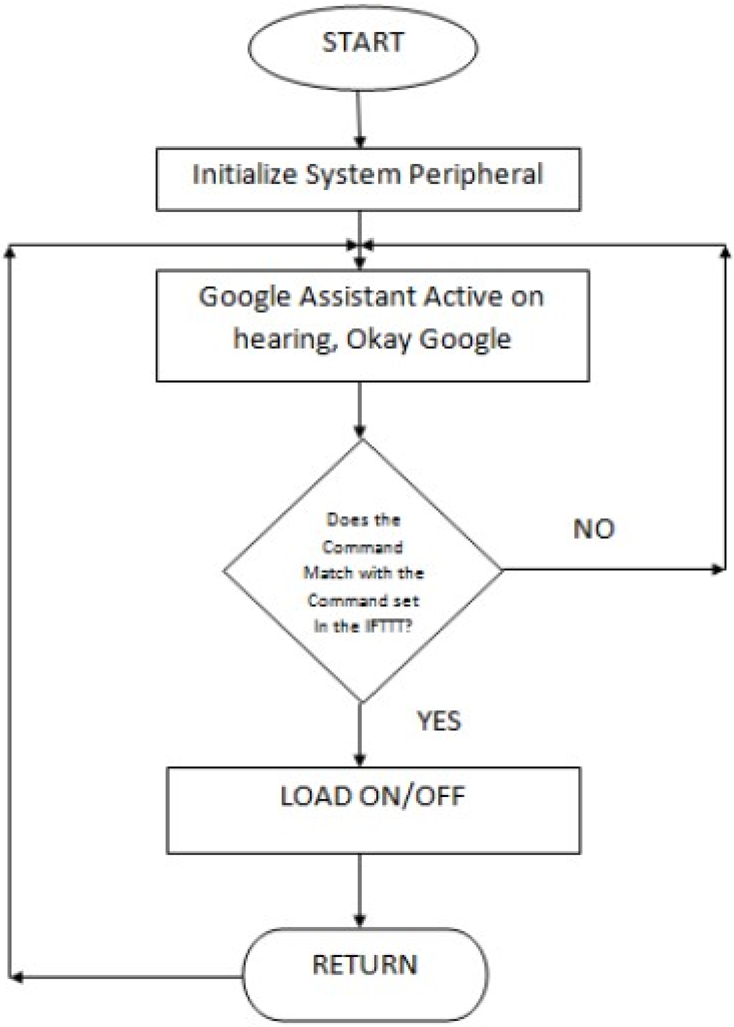
Next, I clicked create trigger. We will see the big **if this then that** again. This time, the +sign will be with the word **that**. Click **that** and we will see a search page again. Search for ***Adafruit*** and select it. I clicked Send data to Adafruit IO. Select my feed (in this case: **on-off**) and set the ***data to save*** to **ON** (remember we created the toggle switch block in Adafruit.IO with Button on Text as **ON**). Click create action. Now click FINISH.

Repeat these steps to create another trigger with action to turn the LED off. This time, set the phrase for the Google assistant to something like "***turn the light off***" and we must set the Adafruit IO section's data to save to OFF.That's it! My Google Assistant can now listen tomy command and turn LED on or off.

Finish

StartedMyNodemcu and wait for it to connect to my Wi-Fi router. Open Google Assistant on my phone and say ***turn on the light*** and we should see the LED turning on!

FLOWCHART


START
Initialize System Peripheral
Google Assistant Active on hearing, Okay Google
Does the Command Match with the Command set In the IFTTT?
NO
YES
LOAD ON/OFF
RETURN

RESULT

The result was positive and the system responded well. The diagram below shows the complete prototype implementation of the proposed system.

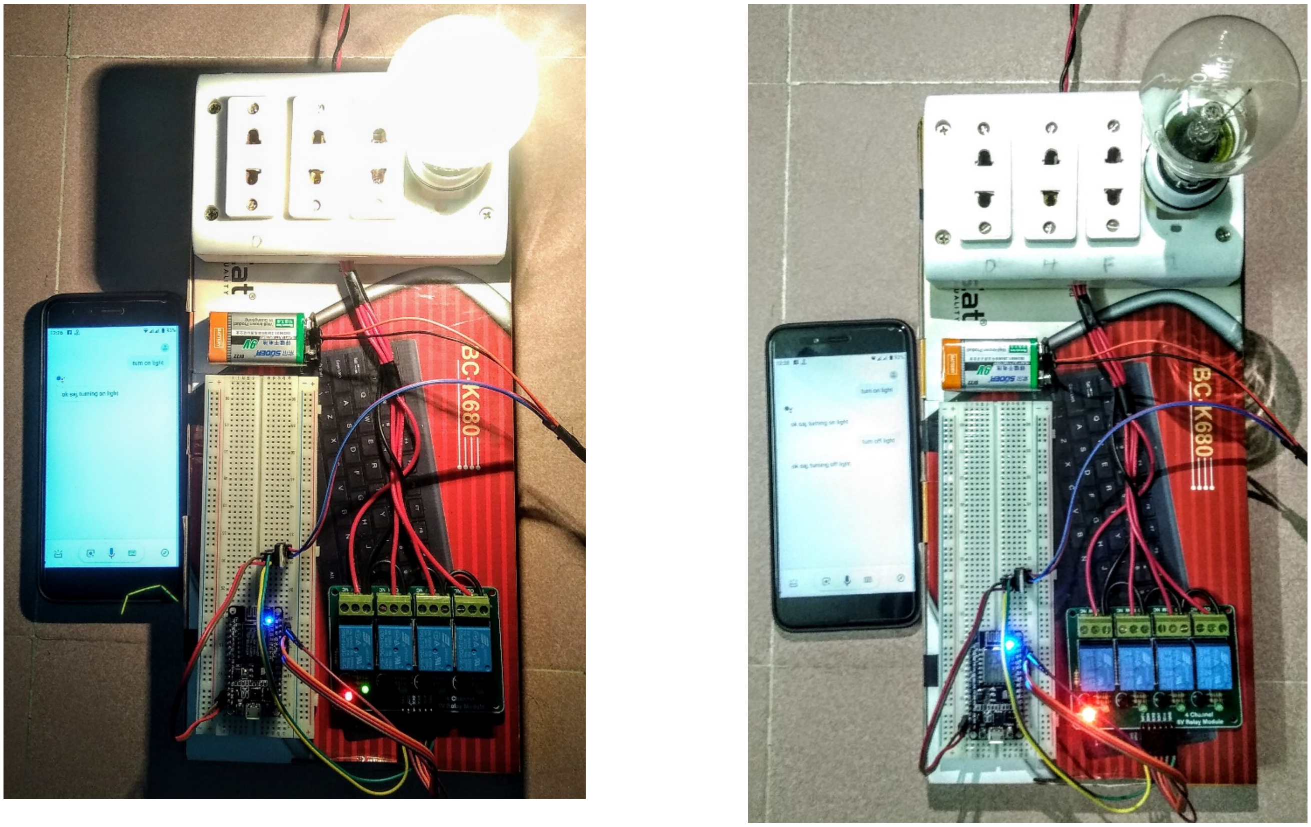


Fig -7: Light Turned ON and then OFF.

NOTE: 5V/1A Output from a 9V Battery through IC7805 were used to power the NodeMCU and the Relay Board.

APPLICATIONS

We can use Voice Control Home Automation to control huge number of home appliances. Such as,

- TV, Home Theater, Speaker.
- Refrigerator, Washer, Dryer, Geyser water heater.
- Light, Fan, Vacuum Cleaner, Air conditioner, Thermostat.
- Electric Door Lock, CC camera etc.

LIMITATIONS

In this system internet connection most important part to communicate between Google assistant and NodeMCU. So internet connection should be proper for speedy and continues services. The power supply for both NodeMCU and Relay module should be same 5V source otherwise the system will not work properly for different power rating. Also we should give voice command in noise free space for proper experience.

CONCLUSION AND FUTURE WORK

The aim of this paper was to propose a cost effective voice controlled (Google Assistant) home automation controlling general appliances found in one's home. The approach discussed in the paper was successful as GACHA's (Google Assistant Controlled Home Automation) design was successfully implemented. This system is highly reliable and efficient for the aged people and differently abled person on a wheel chair who cannot reach the switch for the switching ON/OFF the device and are dependent on others.

The future scope for GACHA can be huge. There are many factors to improve on to make GACHA more powerful, intelligent, scalable, and to become better overall for home automation. For example, controlling the speed of the fan, more number of devices can be integrated, like a coffee machine, air conditioner etc. To make the system respond more faster high speed internet can be used. Well, no system is ever perfect. It always has a scope for improvement. One just needs to put on a thinking cap and try and make the system more better.

REFERENCES

[1] IFTTT: https://ifttt.com/discover

[2] Adafruit: https://learn.adafruit.com/category/adafruit-io
https://io.adafruit.com/sajeebchandra/dashboards

[3] NodeMCU: https://nodemcu.readthedocs.io/en/master
https://iotbytes.wordpress.com/nodemcupinout/

[4] Google Assistant: https://assistant.google.com/intl/en_in/

[5]IoT:https://internetofthingsagenda.techtarget.com/definition/IoT-device

[6] Wikipedia: https://en.wikipedia.org/wiki/Home_automation