

## Table of Contents

Table of Contents .....	2
1.0 Introduction .....	3
1.1 Internet of Things (IoT) .....	3
1.2 The birth of Internet of Things (IoT) .....	3
1.3 Project Documentation .....	3
1.3 Project Innovation and Motivation .....	4
1.4 Research and Problem Solving .....	4
2.0 Hardware and Software requirement .....	5
2.1 NodeMCU ESP-12E ESP8266 WiFi .....	5
2.2 5V 4 channel Relay Module for Arduino .....	6
2.3 Sover LED E14 Bulb 220V 3000K .....	7
2.4 Jumper wires and power cable wire .....	7
2.5 Breadboard .....	7
2.6 Adafruit IO .....	8
2.7 IFTTT Platform (If This Then That) .....	8
2.8 Google Assistant .....	8
2.9 MQTT IoT Messaging protocol .....	8
3.0 Project Design .....	9
3.1 Project flow diagram .....	9
3.2 Circuits Diagrams .....	10
3.3 External Interface Requirements .....	11
4.0 Prototyping and Implementation .....	12
4.1 Arduino IDE programing .....	12
4.2 Adafruit IO setup .....	13
4.3 IFTTT platform setup .....	15
5.0 Final Product .....	16
5.1 Connection of the product .....	16
5.2 Product presentation .....	18
5.3 Product cost estimation .....	18
5.4 Advantage and disadvantage discussion .....	19
Conclusion .....	19
Code of Node MCU .....	20

---

## 1.0 Introduction

---

### 1.1 Internet of Things (IoT)

In the current world with the new digitalization era, the internet has become the most powerful network that manages to connect the whole world in most ease and allow people to communicate. Along the way, most varied purposes have subsequently extended. With this revolutionary, more human activities move from manual to the digital age.

The change of internet evolution made more businesses adapt to new technology. The Internet of Things (IoT) becomes one of the systems of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

### 1.2 The birth of Internet of Things (IoT)



Kevin Ashton, inventor of the Internet of Things

<sup>1</sup>The term Internet of Things is 16 years old. But the actual idea of connected devices had been around longer, at least since the 70s. Back then, the idea was often called “embedded internet” or “pervasive computing”. But the actual term “Internet of Things” was coined by Kevin Ashton in 1999 during his work at Procter & Gamble. Ashton, who was working in supply chain optimization, wanted to attract senior management’s attention to a new exciting technology called RFID. Because the internet was the hottest new trend in 1999 and because it somehow made sense, he called his presentation “Internet of Things”. Even though Kevin grabbed the interest of some P&G executives, the term Internet of Things did not get widespread attention for the next 10 years.

### 1.3 Project Documentation

This project documentation specifies all the requirements for the Home Lights/Appliances automation using Arduino nodemcu via Google assistance. This documentation covers all that is related to the system functionality, constraints, performance, attribute and the system interface. The Home Lights/Appliances automation using Arduino nodemcu via Google assistance designed for ease human interaction to control the home light or home appliances using the digitization concept. This documentation focuses on end to end requirements till the final product.

---

<sup>1</sup> "Why it is called Internet of Things: Definition, history ...." 19 Dec. 2014, <https://iot-analytics.com/internet-of-things-definition/>. Accessed 2 Dec. 2020.

---

### 1.3 Project Innovation and Motivation

Internet of Things (IoT) technology has brought the revolution to each field of human's life by making everything smart and intelligent. Combination of Internet and google assistance services provides the daily activities of users more comfort and a trouble free system. The project scope to design a voice-controlled Home Lights/Appliances automation using Arduino via Google assistant using Wifi and IoT technology. which allows the users to control using a smartphone. Allows users to control basic home features automatically using mobile devices through voice over the internet.

The design also allows the person to control via google assistance anywhere in the world as long he/she connected the mobile phone with internet and Arduino nodemcu connected to home wifi. The development of Home Lights/Appliances automation using Arduino via Google assistance product not only enhancing normal daily actions but also:

- Cost effective - not necessary need to purchase Google Home-compatible smart bulb/fan/appliances
- Preventing theft - the person able to switch on/off the lights/home appliances in any part of the world. This can portrayer the house being occupied by the owners.
- Monitor the usage of the electricity data and pre schedule via google assistance home lights/appliances

### 1.4 Research and Problem Solving

Short research was carried out to understand how Home Lights/Appliances automation using Arduino via Google assistance will be very useful in a person's daily life.

Interviewed	Current situations and problems
<b>Parents / Adults</b>	Sometimes forget to switch off since it's manual to switch on/off lights.
	Require them to assist their child who unable to reach the switch
	Always need to remind the kids to handle the switch with a dry hand (not with wet hands). This is to prevent any unwanted incidents
	Slamming the switches may cause it to wear out
	Require relative or neighbour assistance to switch on/off light during long holidays. This is a precautionary action.

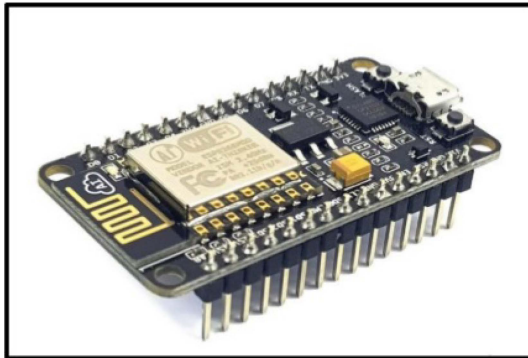
Home Lights/Appliances automation using Arduino via Google assistance product will be the right choice with a cost effective innovation to address all the current issues faced by the end users based on the research.

## 2.0 Hardware and Software requirement

---

The details of hardware components and software used the Project Home Lights/ Appliances automation using Arduino via Google assistance.

### 2.1 NodeMCU ESP-12E ESP8266 WiFi



<sup>2</sup>The ESP8266 NodeMCU V1.0 ESP-12E WiFi module is the latest version of this popular module and can be used as a WiFi enabled replacement for an Arduino in many applications.

#### *Key Features of ESP8266:*

- Microcontroller: ESP-8266 32-bit
- Clock Speed: 80 MHz
- USB Converter: CP2102
- USB Connector: Micro USB
- Operating Voltage: 3.3V
- Flash Memory: 4 MB
- Digital I/O: 11
- Analog Inputs: 1
- Communications: Serial, SPI, I2C and 1-Wire via software libraries
- WiFi: B

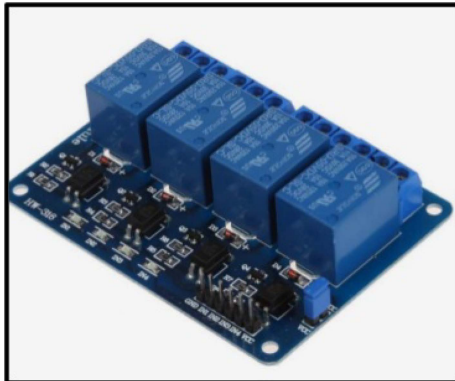
The ARDUINO integrated development environment (IDE) is a cross-platform application. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

---

<sup>2</sup> "ESP8266 NodeMCU V1.0 ESP-12E WiFi Module ...." <https://protosupplies.com/product/esp8266-nodemcu-v1-0-esp-12e-wifi-module/>. Accessed 2 Dec. 2020.

---

## 2.2 5V 4 channel Relay Module for Arduino



<sup>3</sup>The 4 Channel Relay Module is a convenient board which can be used to control high voltage, high current load such as motor, solenoid valves, lamps and AC load. It is designed to interface with microcontrollers such as Arduino, PIC and etc. The relay terminal (COM, NO and NC) is being brought out with a screw terminal. It also comes with a LED to indicate the status of relay.

### Specification:

- Digital output controllable
- Compatible with any 5V microcontroller such as Arduino.
- Rated through-current: 10A (NO) 5A (NC)
- Control signal: TTL level
- Max. switching voltage 250VAC/30VDC
- Max. switching current 10A
- Size: 76mm x 56mm x 17mm

A relay is a switch which is operated electrically. Most of the relays use an electromagnet to mechanically operate a switch, including other operating principles also, such as solid-state relays. Relays are used in applications where only one signal is necessary to control a circuit or several circuits must be controlled by one signal.

---

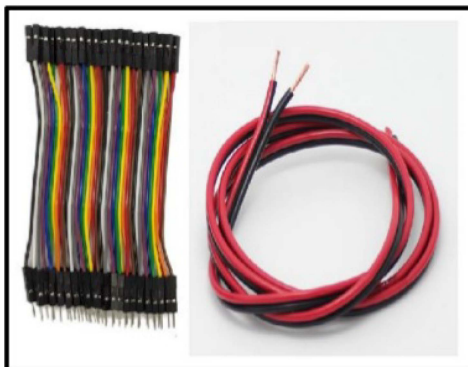
<sup>3</sup> "4 Channel 5V Relay Module - Mybotic." <https://www.mybotic.com.my/products/4-Channel-5V-Relay-Module/895>. Accessed 2 Dec. 2020.

### 2.3 Sover LED E14 Bulb 220V 3000K



<sup>4</sup>Voltage : AC/DC 220V  
Watts : 7W  
Size : 65mm  
Color : Warm White 3000K  
280 LUMEN  
Beam Angle : 360°  
Avg Life of 30,000 hours  
Environmental friendly because of high energy saving and long lasting

### 2.4 Jumper wires and power cable wire



A set of jumper wires male/female types to connect to the pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are generally used with prototyping tools and with breadboards in order to make prototyping easy to change a circuit as required.

Electrical power cable wire black red used to connect the bulb with the relay. We also require a 3 pin plug for the main socket

connection.

### 2.5 Breadboard



A breadboard is used to build and test circuits quickly before finalizing any circuit design. The breadboard has many holes into which circuit components like ICs and resistors can be inserted.

---

<sup>4</sup> "SOVER LED E14 BULB 220V 7W 3000K WARM WHITE ...." <https://www.lazada.com.my/products/sover-led-e14-bulb-220v-7w-3000k-warm-white-i1260880949.html>. Accessed 3 Dec. 2020.

---

## 2.6 Adafruit IO

This project used the Adafruit IO for an internet connection to control the relay. Adafruit IO is a cloud service that allows us to connect over the internet. It also stores the data for us to retrieve for monitoring. All these can be free without any charges by signing up an account with your Gmail id.

## 2.7 IFTTT Platform (If This Then That)

IFTTT or known as If This Then That is a web-based service that allows us to create or set up a list of conditional statements. This is a freeware that allows us to get different services from the channel. The combination of the services called applets. The Applets will be taking care of the daily workflow, integration to full automation. We can sign in to this platform using a Gmail account. Note both Adafruit IO and IFTTT should be signed in with the same Gmail account. Google Assistant is used for giving feed to IFTTT. IFTTT serves as a broker between them.

## 2.8 Google Assistant

<sup>5</sup> Google Assistant is an artificial intelligence powered virtual assistant developed by Google which is basically available on mobile and smart home devices. Google Assistant can engage in two-way conversations living behind the company's foregoing virtual assistant.

## 2.9 MQTT IoT Messaging protocol

MQTT: It's a standard for IoT Messaging protocol. A lightweight message queueing. Its publish/subscribe architecture is designed to be open and easy to implement, with up to thousands of remote clients capable of being supported by a single server.

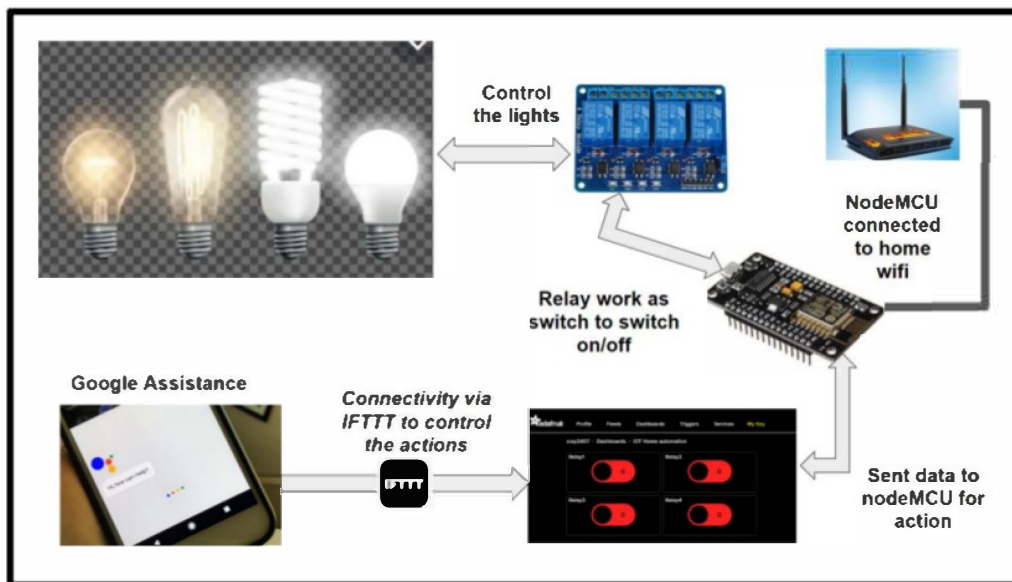
---

<sup>5</sup> "Google Assistant - Wikipedia." [https://en.wikipedia.org/wiki/Google\\_Assistant](https://en.wikipedia.org/wiki/Google_Assistant). Accessed 3 Dec. 2020.

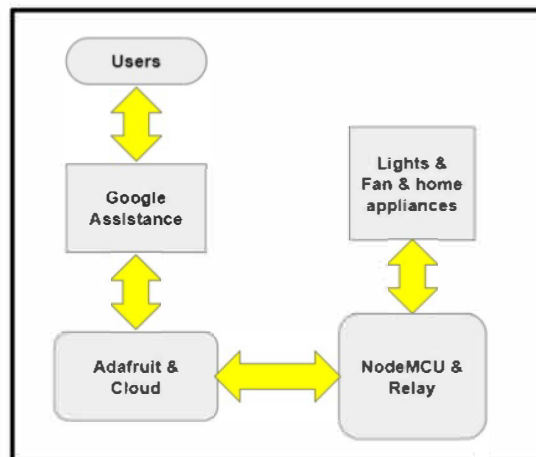
## 3.0 Project Design

### 3.1 Project flow diagram

Below shows the flow diagram of the project design showing the interaction of the components with the software for the Home lights/appliances automation using Arduino via Google assistant.



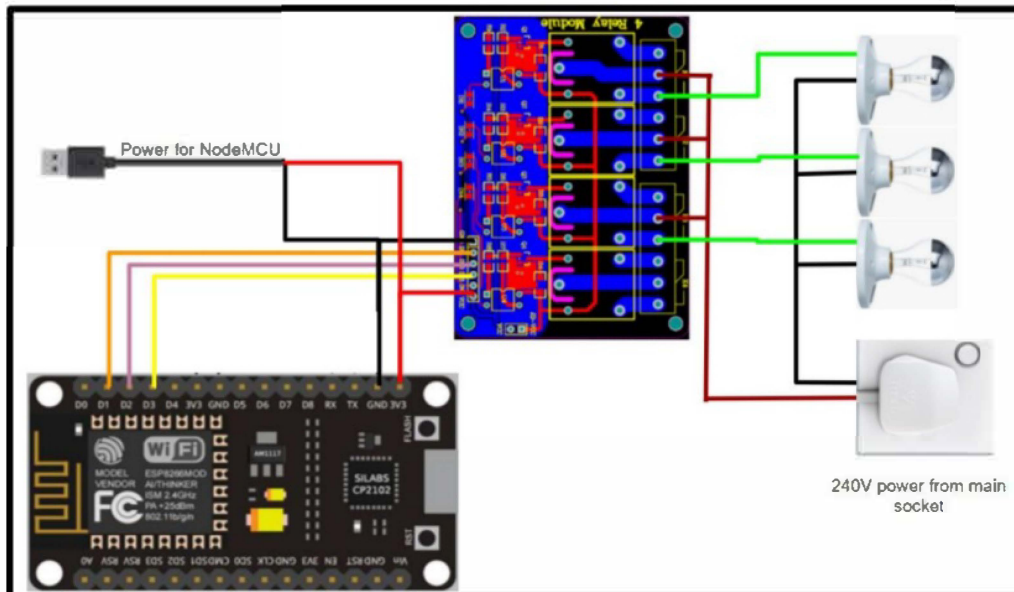
Below is the functional diagram





## 3.2 Circuits Diagrams

The circuit diagram shows the hardware connection for the Home lights/appliances automation using Arduino via Google assistant.

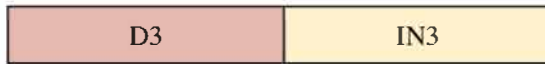


NodeMCU/ESP8266 3.3V pins of the onboard voltage regulator and can also supply 5V regulated to the VIN pin. The GND are the ground pins. At the same time, the connection from NodeMCU/ESP8266 to 5V 4 Channel relay. NodeMCU 3.3V pins will be connected to the VCC (Voltage Common Collector) pin of 5V 4 Channel relay. This supplies power according to relay voltage. NodeMCU GND pin will be connected to the 4 Relay board GND pin.

For this project, we will be using 3 GPIO pins whereby this can be assigned to functions. Each digital enabled GPIO can be configured to internal pull-up or pull-down, or set to high impedance. The GPIO pins will be connected with the IN pin terminal 1 of the relay.

Pin Connection table reference

NodeMCU/ESP8266	5V 4 Channel relay
3.3V pin	VCC pin
GND pin	GND pin
D1	IN1
D2	IN2



The lights / appliances will be connected to COM common pins and NO (Normally Open) pin. The normally open configuration works the other way around: the relay is always open, so the circuit is broken unless you send a signal from the Arduino to close the circuit.

### 3.3 External Interface Requirements

Internet connection is a required part of the project, NodeMCU/ESP8266 has built WIFI and can be programmed to be connected with the home wifi. The IFTTT will come as a broker between good assistance and IO adafruits.

---

## 4.0 Prototyping and Implementation

---

### 4.1 Arduino IDE programming

The Prototype will be designed as per project requirement as per circuit diagram in 3.2. Since the project connects to the main power supply with the output of the 240V, remember to place the prototype with an electricity insulator to prevent any accidents.

Once done with hardware setup, this when needed to work with programming in Arduino IDE. The ARDUINO integrated development environment (IDE) is a cross platform application (for Windows, macOS, Linux) that is written in the programming language Java. It's used to write and upload programming with compatible boards. The user-written code will be compiled and uploaded to the NodeMCU/ESP8266.

We need to add the following library in our Arduino IDE as part of the project.

- ❑ **ESP8266WiFi.h** - ESP8266 is all about Wi-Fi. for Wi-Fi networks to start sending and receiving data,
- ❑ **Adafruit\_MQTT.h & Adafruit\_MQTT\_Client.h** - MQTT is a lightweight publish-subscribe-based messaging protocol. This is based on APIs like http, allows remote location devices to connect, subscribe and publish. Works as a message broker between sender and receiver.

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

#define Relay1      D1
#define Relay2      D2
#define Relay3      D3

#define WIFI_ID      "XXXXXXXXXX-"
#define WIFI_PASSWRD "XXXXXXXXXXXX"

//Adafruit Server and the keys detail

#define AdaFIO_SERVER      "io.adafruit.com"
#define AdaFIO_SERVERPORT  1883
#define AdaFIO_USERNAME    "sray2407"
#define AdaFIO_KEY          "aio_BgxP57tptv7dVdK9Kz291SE2TTU9"

WiFiClient client;

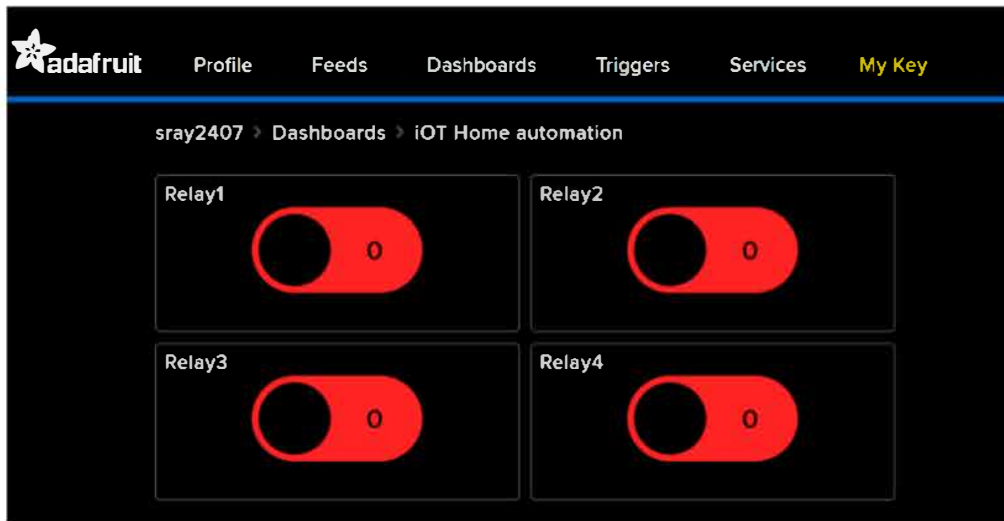
Adafruit_MQTT_Client mqtt(&client, AdaFIO_SERVER, AdaFIO_SERVERPORT, AdaFIO_USERNAME, AdaFIO_KEY);
```

We also need to define the home wifi ID and password for the NodeMCU/ESP8266 to be connected which allows the nodeMCU/ESP8266 to communicate with the adafruit IO. As part of the cording, it's required to set up the adafruit IO in the cording with IO server details, users name and the IO key.

---

## 4.2 Adafruit IO setup

Create an account on the Adafruit.IO (<https://io.adafruit.com/>) platform for a dashboard creation. We should be able to create the dashboard and setup the relay information in Adafruit.IO.



The Adafruit.IO also allow us to track the feed information using the dashboard which was created

+ Add Data

Download All Data

Filter

Prev

First

page 1 of 1

Next

Created at	Value	Location
2020/12/11 3:42:45AM	0	✖
2020/12/11 3:42:44AM	1	✖
2020/12/10 1:16:45AM	0	✖
2020/12/10 1:16:44AM	1	✖
2020/11/21 4:23:52PM	0	✖
2020/11/21 4:23:41PM	1	✖
2020/11/21 3:15:13PM	0	✖
2020/11/21 3:15:10PM	1	✖

We also require the Adafruit IO key details for the coding part in Arduino IDE where it allows for the board to be connected for the data feed. This when IFTTT will come as a broker or middleware platform. The values in the gauge will be updated automatically

## YOUR ADAFRUIT IO KEY



Your Adafruit IO Key should be kept in a safe place and treated with the same care as your Adafruit username and password. People who have access to your Adafruit IO Key can view all of your data, create new feeds for your account, and manipulate your active feeds.



If you need to regenerate a new Adafruit IO Key, all of your existing programs and scripts will need to be manually changed to the new key.

**Username**

**Active Key**

REGENERATE KEY

[Hide Code Samples](#)

Arduino

```
#define IO_USERNAME  "sray2407"  
#define IO_KEY       "aio_Bgxp57tpv7dVdK9KzZ91SEZTTU9"
```

Linux Shell

```
export IO_USERNAME="sray2407"  
export IO_KEY="aio_Bgxp57tpv7dVdK9KzZ91SEZTTU9"
```

Scripting

## 4.3 IFTTT platform setup

IFTTT stands for If This Then That. It is a free web-based service for creating chains of simple conditional statements, called applets. These applets are triggered by changes that occur within other web services. The web-based application, the service runs on IOS and Android. Create an account in IFTTT (<https://ifttt.com>) using the account used for IO.Adafruit.

Create the command that needs to be used in the applet setting and create the connection between IFTTT and data feed to Adafruit IO. We will be creating an applet for “on” and “off” functions for the same relay. Repeated the same action for all the relays.

Note: If you connect using more relays, you are required to create more applets with the same setup.

### Applet settings



**If You say "Turn on the hall light", then Send data to Relay 1 feed**

Edit title

by reanish

### ★ 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

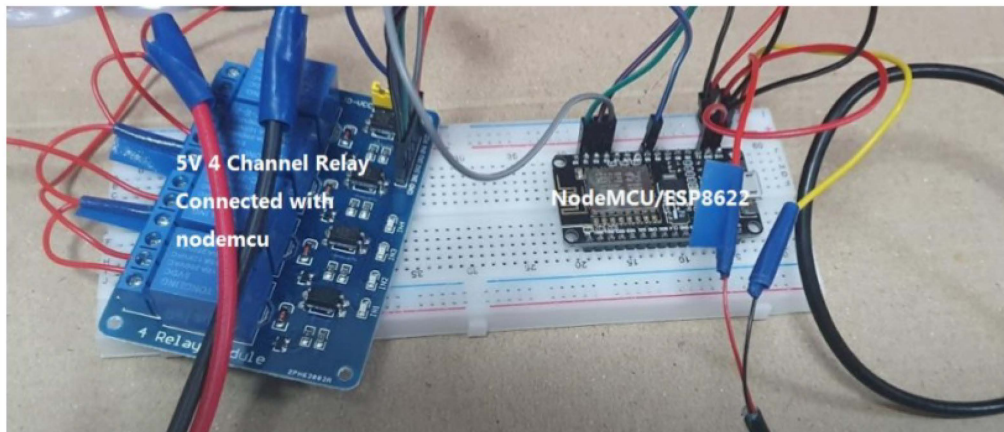
The data to be saved to your feed.

**Add ingredient**

## 5.0 Final Product

---

### 5.1 Connection of the product



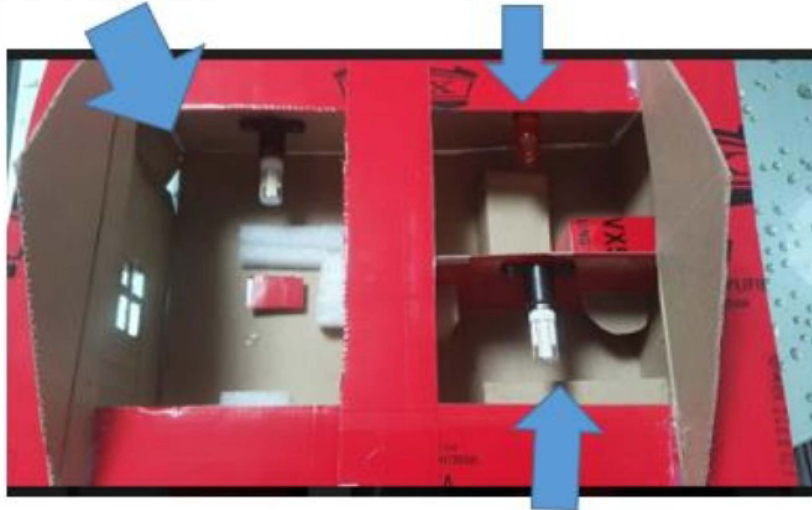
The final product will be connected as per the 3.2 Circuits Diagrams. The relay will be connected with the nodemcu/ESP8622.

To project the product, the Home Lights/Appliances automation using Arduino via Google assistant was designed using a home concept, whereby the lights were connected in a different part in the house model. The light 1 is placed in a hall whereby it is connected to a relay IN1, light 2 placed in a room connected to relay IN2 and relay 3 IN3 connected to kitchen light.

Please find the picture below of the home model

Light 1 for hall

Light 2 for room



Light 3 kitchen





## 5.2 Product presentation

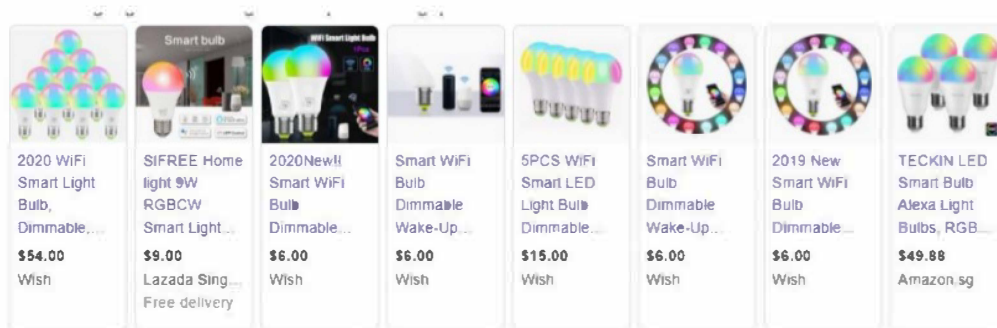
This is the presentation demo of the Home Lights/Appliances automation using Arduino via Google assistant. The lights will be voice controlled via google assistance. As per the introduction, we also can connect the relay IN to any of the home appliances using the same connection method.

Project presentation place in the google file

[https://drive.google.com/file/d/1TjH4qNVR2\\_ppwO1vHipZyA\\_PBay3zB4Y/view?usp=sharing](https://drive.google.com/file/d/1TjH4qNVR2_ppwO1vHipZyA_PBay3zB4Y/view?usp=sharing)

## 5.3 Product cost estimation

Here comes the product cost estimation, whereby it's shown this is a cost effective product that can be installed in any house. Below shows the price of google assistance bulbs in the Singapore market. It comes with specified looks, At least 14 bulbs required for a 3 room HDB flat in Singapore. Easily we need at least SGD100.00 to set up the house with google assistance lights.



However, using the Home Lights/Appliances automation using Arduino via Google assistant product and below are the cost estimations.

Hardware	SGD
NodeMCU/ESP8622	15
5v 4 channel relay	5
Male/female jumper wires	8
Red/Black cables - SGD 0.80 per meter	1.6
<b>Total cost</b>	<b>29.6</b>

Note: 5v 8 channel relay cost SGD8

There is no any cost of the lights because this product allows you to use your existing lights in the house. This clearly shows us Home Lights/Appliances automation using Arduino via Google assistant using nodeMCU/ESP8266 one of the cost effective

automation. This product also allows us to use the existing lights as per our own preference. The same ideas or innovation also can be setup for any power extension cord

## 5.4 Advantage and disadvantage discussion

The advantage of the product makes the human daily life better since the home light, fan controlled via voice. We simply sit and control everything using voice controlled. However, they are few disadvantage of the product such like: -

- The product only works with internet connection
- The layer electronic setup for any home wiring could lead to have more than one nodeMCU setup in the home
- Extra wall box setup might be required to place the nodeMCU and away from the kids.

## Conclusion

IOT is new area technology becoming human fingertips technology makes human life to be more into a new era. I have learned a lot of innovative ideas through this module. I also learn the basic understanding of electronics while learning the IOT module.

Using the innovation and ideas we can save more time and cost and also reduce home efforts. The data that's being collected can be studied, shows the pattern of human life which creates a new algorithm, tracking and etc. This module is really an exciting subject to learn.

---

## Code of Node MCU

---

```
//Home automation with Google assistance
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"

#define Relay1      D1
#define Relay2      D2
#define Relay3      D3

#define WIFI_ID      "----username---"
#define WIFI_PASSWRD  "---password---"

//Adafruit Server and the keys details

#define AdaFIO_SERVER    "io.adafruit.com"
#define AdaFIO_SERVERPORT 1883
#define AdaFIO_USERNAME  "sray2407"
#define AdaFIO_KEY       "aio_Bgxp57tpv7dVdK9KzZ91SEZTTU9"

WiFiClient client;

Adafruit_MQTT_Client mqtt(&client, AdaFIO_SERVER, AdaFIO_SERVERPORT,
AdaFIO_USERNAME, AdaFIO_KEY);

Adafruit_MQTT_Subscribe Light1 = Adafruit_MQTT_Subscribe(&mqtt,
AdaFIO_USERNAME"/feeds/Relay1");
Adafruit_MQTT_Subscribe Light2 = Adafruit_MQTT_Subscribe(&mqtt,
AdaFIO_USERNAME "/feeds/Relay2");
Adafruit_MQTT_Subscribe Light3 = Adafruit_MQTT_Subscribe(&mqtt,
AdaFIO_USERNAME "/feeds/Relay3");

void MQTT_connect();

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

  pinMode(Relay1, OUTPUT);
  pinMode(Relay2, OUTPUT);
  pinMode(Relay3, OUTPUT);
```

---

```
Serial.println(); Serial.println();
Serial.print("Connecting to home wifi ");
Serial.println(WIFI_ID);

WiFi.begin(WIFI_ID, WIFI_PASSWRD);
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println();

Serial.println("Home WiFi is connected");
Serial.println("Please find the IP address: ");
Serial.println(WiFi.localIP());

mqtt.subscribe(&Light1);
mqtt.subscribe(&Light3);
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(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);
  }
}

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!");
}
```

---