PROJECT REPORT

On

"SMART HOME"

Submitted by.

Name: SONALI, CHANDRAKANTH.

USN: 3gn17ec061, 3gn16ec017.

Dept.:ECE

Date: 25-10-2021

UNDERTAKING

I declare that the work presented in this project titled "SMART HOME", submitted to the All India council of robotics and Automation, for the award of the Internship in INTERNET OF THINGS, is my original work. I have not plagiarized or submitted the same work for the award of any other Internship. In case this undertaking is found incorrect, I accept that my Project may be unconditionally withdrawn.

CERTIFICATE

Certified that the work contained in the project titled "SMART HOME, by CHANDRAKANTH and SONALI has been carried out under my supervision and that this work has not been. submitted elsewhere for a Internship..

All India Council of Robotics and Automation

Name of the Internship

Delhi-110020

PREFACE

The fundamental purpose of monitoring electronics appliances in the modern world by using IOT (Internet of Things) is to make everything in the house automatically controlled using technology to control them and perform the job that we normally do manually. Nowadays, efficient control is more and moreneeded to optimize performance and saves unnecessary wastage of power. The basics home appliances are light, fan and refrigerators, by controlling them unnecessary wastageof power and resources by turning on lights during day time and high speed fan while no one is around can be avoided. A system has been developed to control household appliances anywhere at any time across the globe. Node MCU, an android operating system is used to achieve the aforementioned automation.

This project proposes an efficient implementation for IOT (Internet of Things) used for monitoring and controlling the home appliances via Internet. Home automation uses portable devices as user interface. The Node MCU server communicates with the corresponding relay hardware circuits that control the appliances running at home. With this we provide a comfortable and effective home automation system.

ACKNOWLEDGMENTS

I take upon this opportunity to acknowledge the many people whose prayers and support meant a lot to me I am deeply indebted to my mentor SUMIT CHATTERJEE RESEARCH ENGINEER, ALAAUDEEN KM who motivated me along the way I would like to thank all my teachers who help me in this project I further thank to my friends I owe my sincere gratitude towards the God My heartfelt thanks to parents who supported me a lot.

Finally, I would like to wind up by paying my heartfelt thanks to AICRA institute who provided me with this great opportunity

CONTENTS

CHAPTER 1:

INTRODUCTION

1.1 INTRODUCTION

The main objective of this project is to develop a home control system using Node MCU board with internet being remotely controlled by android or any operating system. As technology is advancing so houses are also getting smarter. Modern houses are gradually shifting from conventional switches to centralized control system, involving google assistant and sensors.Automation is integrating our surrounding, physical world, into computer-based system.This project aims at controlling home appliances via web browser using Wi-Fi as communication protocol and Node MCU as server system.

Presently, conventional wall switches located in different parts of the house makes it difficult for the user to go near them to operate, it is even more difficult for the handicapped people. Home automation with google assistant provides a most modern solution with smart phones.

1.2 HOME AUTOMATION

Home automation means automation of the home, housework or household activity. Home automation is also defined as the control of lighting, heating, ventilation, air conditioning, security locks of gates and doors and other system to provide improved convenience,effective and comfort including efficient use of power.

Home automation can provide a better life quality for the persons who might require a caregivers or institutional care.IOT allows objects to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into digital world. Home automation includes the control of domestic activities such as lighting control system, the use of domestic robots, changing the ambiance of the house according to circumstances, pet feeding etc. Through the integration of information technologies with the home environment, systems and appliances

can communicate in an integrated manner which results in convenience, energy efficiency, and safety benefits.Thepopularity of home automation has been increasing greatly in recent years due to much higher affordability and simplicity through smartphone connectivity.

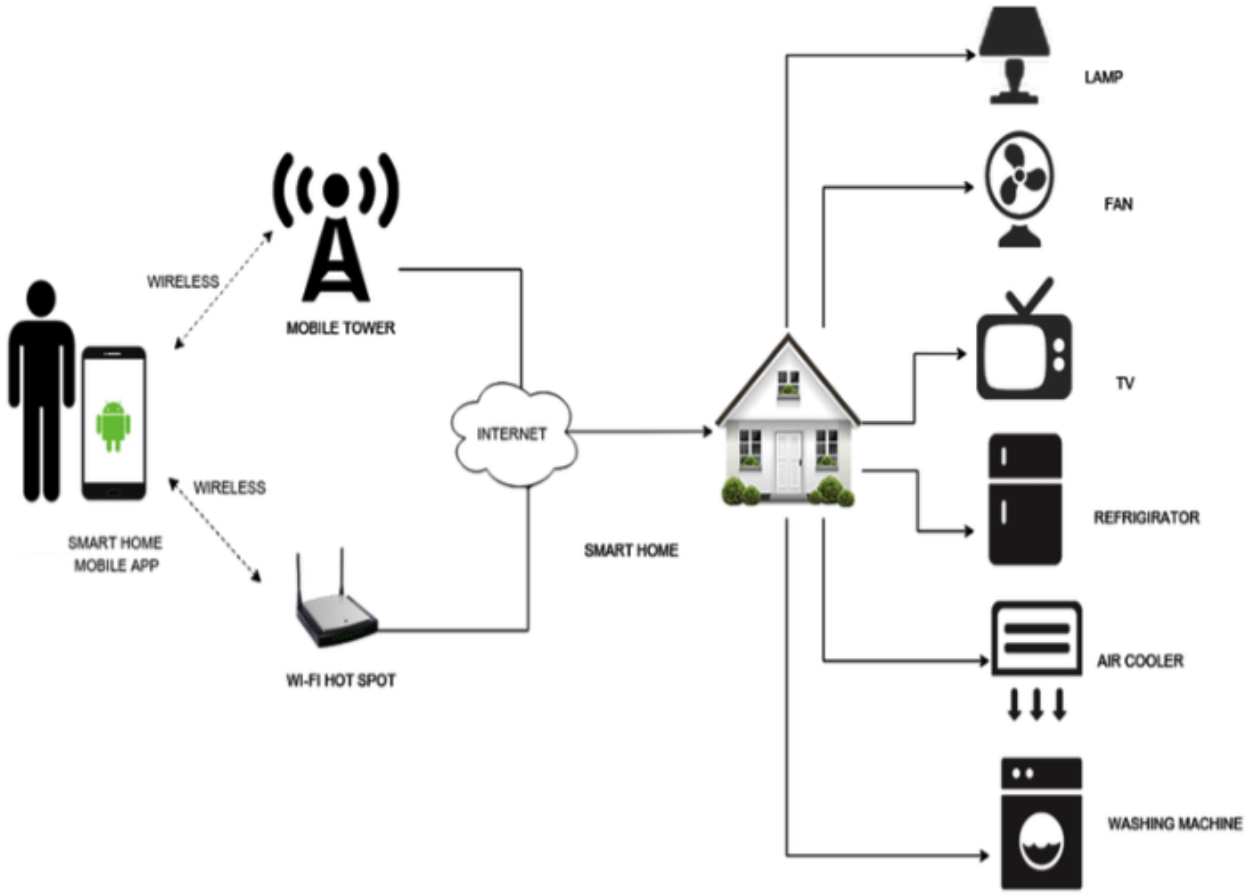


Figure 1: Concept of home automation

1.3 OVERVIEW

Home automation refers to the use of smartphone or computer and internet to control home appliances and features (such as lights, fan, doors, etc.). Systems is ranged from simple remote control of lighting to complex computer/micro-controller-based networks with varying degrees of intelligence

and automation. Home automation is adopted for reasons of simplicity, security and energy efficiency including digitalizing the world.

In industrialized world, most homes have beenwired for electrical power, telephone lines, TV outlets (cable and antenna). Many household tasks were automated by the development of specialized automated appliances. The use of gaseous or liquid fuels, and later the use of electricity enabled increased automation in heating, reducing the labor necessary to manually refuel heaters and stoves. Introduction of automatic washing machines was developed to reduce the manual labor of cleaning clothes, and water heaters reduced the labor necessary for bathing and other purposes.

Asthe number of controllable devices in the house rises interconnection and communication becomes a useful and desirable feature. For example, an LDR (Light Detector Sensor) can send an alert message when it gets dark and light is needed or a person when he needs service. If no one is supposed to be home and the alarm system is set, the home automation system could call the owner, or the neighbors, or an emergency number if an intruder is detected. Or if the owner is at his work place and realized that he left his fan and some lights on, he can turn them off from there without moving.

In project, our automation installation is asstraightforward as turning on the lights when a person says“turn ON light” or turning on the fan when someone says “turn ON fan” to his google voice from his mobile phone. In advanced automation, installations can sense not only the words of a person talking but know who that person is and perhaps set appropriate lighting, fan speed, temperaturetaking into account the day of the week, the time of day, and other factors.

CHAPTER 2:

WORKING OF THE PROJECT

In this chapter we have given a description of the whole project. We describe in detail the project and also, we provided a physical view, a block diagram and flowchart are also attached to describe an overview and a deep description respectively.

2.1 Block Diagram

Block diagram is a diagram of a system in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks. They are heavily used in engineering in hardwaredesign, electronic design, software design, and process flow diagrams.

Block diagrams are typically used for higher level, less detailed descriptions that are intended to clarify overall concepts without concern for the details of implementation. Contrast this with the schematic diagrams and layout diagrams used in electrical engineering, which show the implementation details of electrical components and physical construction. In our project the central device which is “NODE MCU” is connected to IFTTT server which interconnect Google Assistant to Adafruit.io platform. The MCU is also connected via GPIO (General Purpose Input Output) pins to 3 Channel Relays and to a Night Lamp circuit. The diagram is given below.

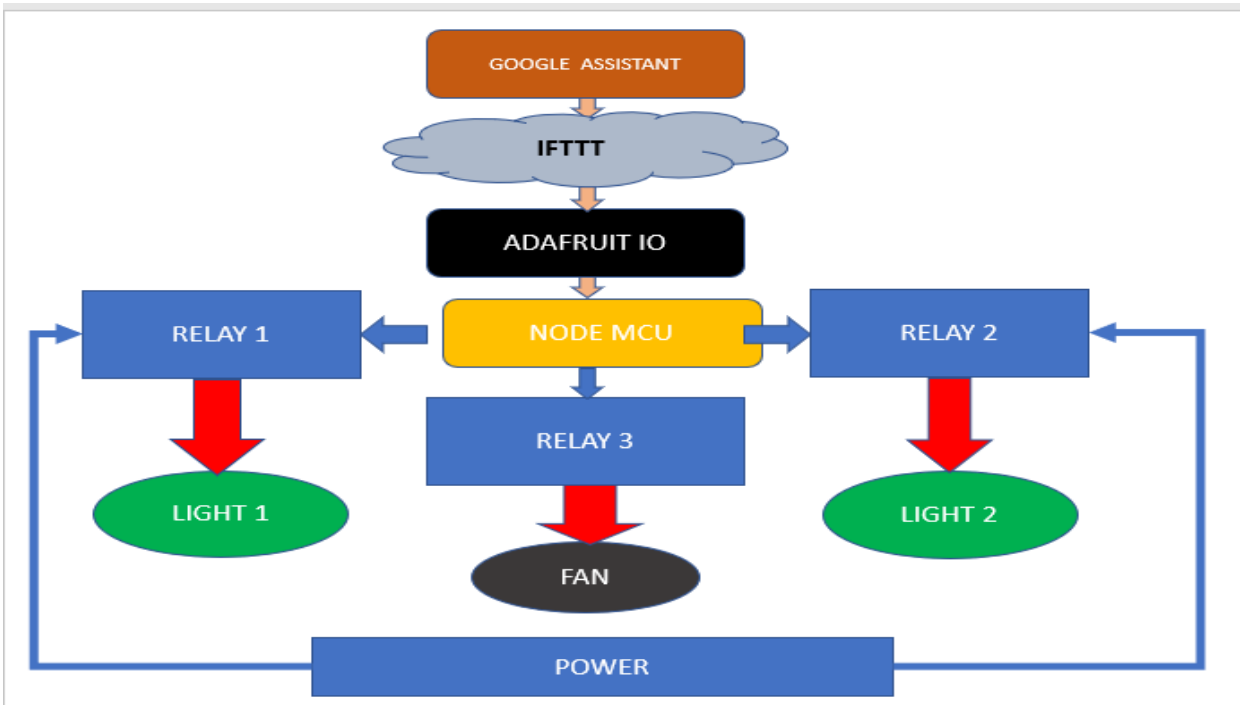


Fig 2.1 Block diagram of IOT based home automation system

2.2 Flowchart

A flowchart is a formalized realistic portrayal of a rationale succession, work or assembling process, association outline, or comparable formalized structure. The motivation behind a stream outline is to furnish individuals with a typical dialect or reference moment that managing a venture or process. Flowcharts utilize straightforward geometric images and bolts to characterize connections. In programming, for example, the start or end of a program is spoken to by an oval. A procedure is spoken to by a square shape, a choice is spoken to by a precious stone and an I/O process is spoken to by a parallelogram. The Internet is spoken to by a cloud. Below is the flowchart.

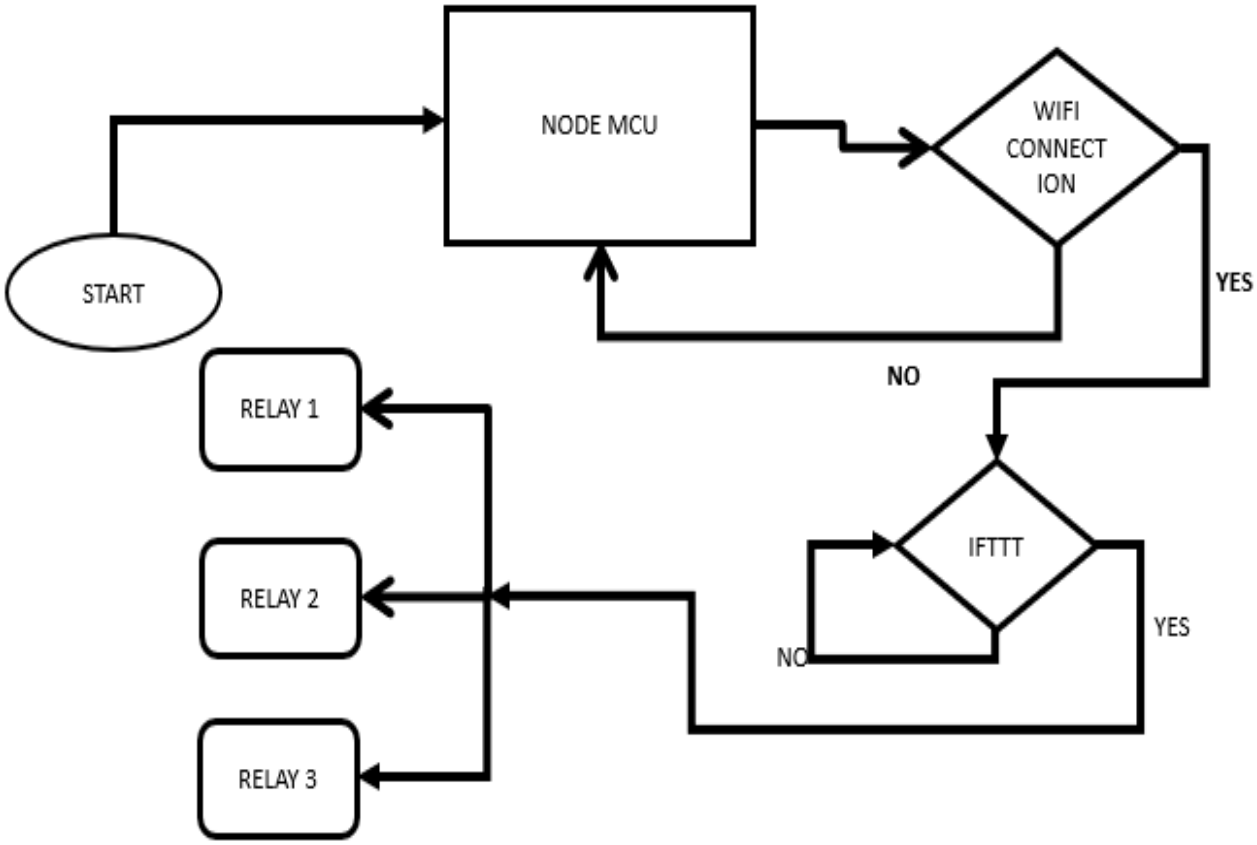


Fig 2.2 Flowchart of the system

2.3 project view

Here, we can see the project on Fig 2.3

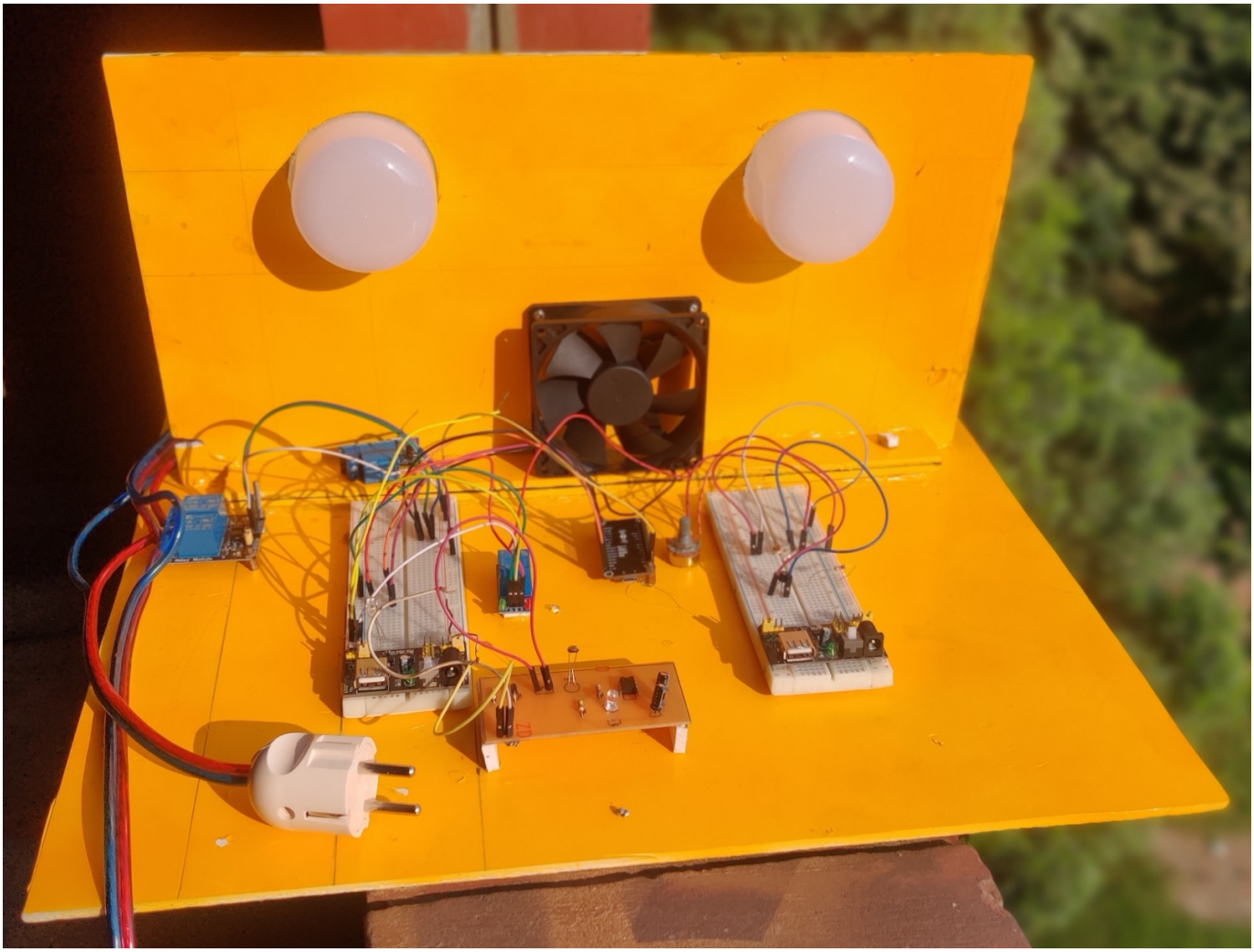


Fig 2.3 Hardware view

2.4 Code:

```
//Google Assistant Home Automation
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"

#define Relay1          D1
#define Relay2          D5
#define Relay3          D3
#define Relay4          D4

#define WLAN_SSID       "Insert WLAN_SSID"            // Your SSID
#define WLAN_PASS       "Insert WLAN_Pass"        // Your password

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

#define AIO_SERVER      "io.adafruit.com" //Adafruit Server
#define AIO_SERVERPORT  1883
#define AIO_USERNAME    "Insert AIO_ Username"           // Username
#define AIO_KEY         "Insert AIO_Key"   // Auth Key

//WIFI CLIENT
WiFiClient client;
Adafruit_MQTT_Clientmqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);
Adafruit_MQTT_Subscribe Light1 = Adafruit_MQTT_Subscribe(&mqtt,
AIO_USERNAME"/feeds/Relay1"); // Feeds name should be same everywhere
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());
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;
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!");
}
```

CHAPTER 3:
HARDWARE

It is the body of the model. We have used several components in order to complete the prototype model. We will describe every component not in deep but at least how it works.

3.1 Choosing the right component

Knowing the importance of components in a project we must wisely and carefully select those. In case we do not, we might encounter some careless errors. For instance, when we will mount the whole circuit, some of the components might not perform well, they can have certain voltage drop or some elements might not get proper power supply. In this case the whole system can fail while the whole circuit integration is correct. So, perfect choice of components is a big concern while making a prototype model.Meanwhile the cost of the model is also to be considered here as in the market, everyone generally prefers a good output at an affordable budget. We have selected different components to make the
Device physically feasible. We tried also to use cheap and effective components as well as availability of the components. We have tried to show almost all the characteristics and behavior of different component that are used as a hardware part in our model.

3.1.1 Node MCU

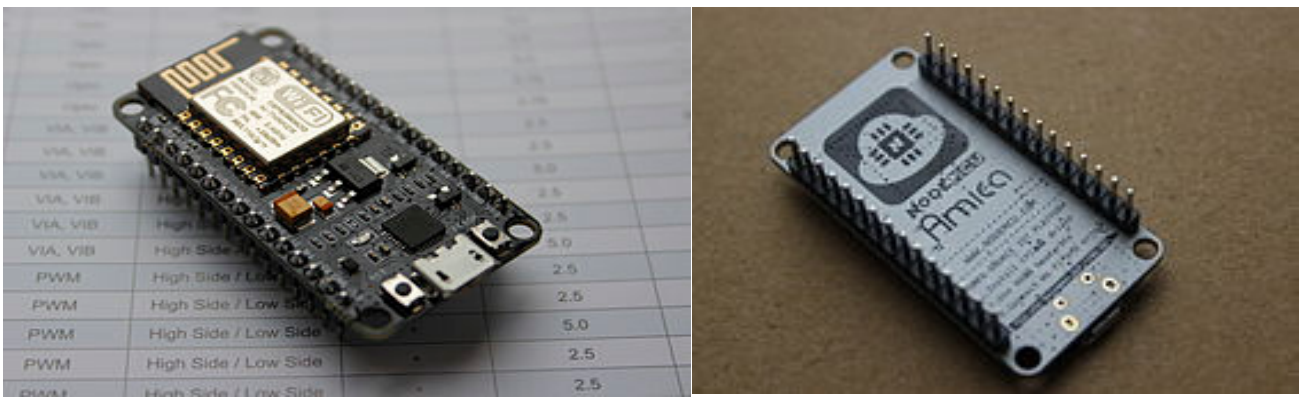


Fig 3.1 Node MCU

NodeMCU is an open source IoT stage. It incorporates firmware which keeps running on the ESP8266 Wi-Fi SoC from Expressive Systems, and equipment which depends on the ESP-12module. The term NodeMCU typically refers to the firmware, whereas the board is termed Devkit. It additionally contains a transformer, a USB interface. The expression "NodeMCU" of course alludes to the firmware as opposed to the improvement units.

Model specification:

Developer	ESP8266 Open source community
type	Single board microcontroller
Introductory price	$5
Operating system	XTOS
CPU	Esp8266(LX106)
Memory	128KBytes
storage	4MBytes
Power	USB

Table 3.1 Node MCU specification

Pins description

While the ESP8266 is often used as a 'dumb' Serial-to-WIFI bridge, it's a very powerful microcontroller on its own. In this chapter, we'll look at the non-Wi-Fi specific functions of the ESP8266.

Digital I/O

Just like a normal Arduino, the ESP8266 has digital input/output pins (I/O or GPIO, General Purpose Input/output pins). As the name implies, they can be used as digital inputs to read a digital voltage, or as digital outputs to output either 0V (sink current) or 3.3V (source current).

Voltage and current restrictions

The ESP8266 is a 3.3V microcontroller, so its I/O operates at 3.3V as well. The pins are **not 5V tolerant, applying more than 3.6V on any pin will kill the chip.**

The maximum current that can be drawn from a single GPIO pin is **12mA**.

Usable pins

The ESP8266 has 17 GPIO pins (0-16), however, you can only use 11 of them, because 6 pins (GPIO 6 - 11) are used to connect the flash memory chip. This is the small 8-legged chip right next to the ESP8266. If you try to use one of these pins, you might crash your program.

GPIO 1 and 3 are used as TX and RX of the hardware Serial port (UART), so in most cases, you can't use them as normal I/O while sending/receiving serial data.

Boot modes

As mentioned in the previous chapter, some I/O pins have a special function during boot: They select 1 of 3 boot modes:

GPIO15	GPIO0	GPIO2	Mode
0V	0V	3.3V	Uart Bootloader
0V	3.3V	3.3V	Boot sketch (SPI flash)
3.3V	x	x	SDIO mode (not used for Arduino)

Table 3.2 Boot mode setting

Note: you don't have to add an external pull-up resistor to GPIO2, the internal one is enabled at boot. We made sure that these conditions are met by adding external resistors in the previous chapter, or the board manufacturer of your board added them for you. This has some implications, however:

- GPIO15 is always pulled low, so you can't use the internal pull-up resistor. You have to keep this in mind when using GPIO15 as an input to read a switch or connect it to a device with an open-collector (or open-drain) output, like I²C.
- GPIO0 is pulled high during normal operation, so you can't use it as a Hi-Z input.
- GPIO2 can't be low at boot, so you can't connect a switch to it.

Internal pull-up/-down resistors

GPIO 0-15 all have a built-in pull-up resistor, just like in an Arduino. GPIO16 has a built-in pull-down resistor.

PWM

Unlike most Atmel chips (Arduino), the ESP8266 doesn't support hardware PWM, however, software PWM is supported on all digital pins. The default PWM range is 10-bits @ 1kHz, but this can be changed (up to >14-bit@1kHz)

Analog input

The ESP8266 has a single analog input, with an input range of 0 - 1.0V. If you supply 3.3V, for example, you will damage the chip. Some boards like the NodeMCU have an on-board resistive voltage divider, to get an easier 0 - 3.3V range. You could also just use a trim pot as a voltage divider. The ADC (analog to digital converter) has a resolution of 10 bits.

Communication

Serial

The ESP8266 has two hardware UARTS (Serial ports):
UART0 on pins 1 and 3 (TX0 and RX0 resp.), and UART1 on pins 2 and 8 (TX1 and RX1 resp.), however, GPIO8 is used to connect the flash chip. This means that UART1 can only transmit data.

UART0 also has hardware flow control on pins 15 and 13 (RTS0 and CTS0 resp.). These two pins can also be used as alternative TX0 and RX0 pins.

I²C

The ESP doesn't have a hardware TWI (Two Wire Interface), but it is implemented in software. This means that you can use pretty much any two digital pins. By default, the I²C library uses pin 4 as SDA and pin 5 as SCL. (The data sheet specifies GPIO2 as SDA and GPIO14 as SCL.) The maximum speed is approximately 450 kHz.

SPI

The ESP8266 has one SPI connection available to the user, referred to as HSPI. It uses GPIO14 as CLK, 12 as MISO, 13 as MOSI and 15 as Slave Select (SS). It can be used in both Slave and Master mode (in software).

GPIO overview

GPIO	Function	State	Restrictions
0	Boot mode select	3.3V	No Hi-Z
1	TX0	-	Not usable during Serial transmission
2	Boot mode select TX1	3.3V (boot only)	Don’t connect to ground at boot time Sends debug data at boot time
3	RX0	-	Not usable during Serial transmission
4	SDA (I²C)	-	-
5	SCL (I²C)	-	-
6 - 11	Flash connection	x	Not usable, and not broken out
12	MISO (SPI)	-	-
13	MOSI (SPI)	-	-
14	SCK (SPI)	-	-
15	SS (SPI)	0V	Pull-up resistor not usable
16	Wake up from sleep	-	No pull-up resistor, but pull-down instead Should be connected to RST to wake up

Table3.3 general GPIO description

3.1.2 Relay module

Fig 3.2 Relay module

This relay module allows to combine the processing power of Arduino to devices that use higher current and voltage. It does so by providing four relays that are rated for 7A at either 28VDC or 10A at 125VAC. Each relay has a Normally Open (NO) and a Normally Closed (NC) contact.

Specification:

• 4-Channel Relay interface board, and each one needs 15-20mA Driver Current
• Both controlled by 12V and 5V input Voltage
• Equipped with high-current relay, AC250V 10A; DC30V 10A
• Standard interface that can be controlled directly by microcontroller (Arduino, 8051, AVR, PIC, DSP, ARM, ARM, MSP430, TTL logic active low)
• Opto-isolated inputs
• Indication LED's for Relay output status.

3.1.3 Breadboard

Fig 3.3 Breadboard

A breadboard is a solder less device for temporary prototype with electronics and test circuit designs. Most electronic components in electronic circuits can be interconnected by insertingtheir leads or terminals into the holes and then making connections through wires where appropriate. The breadboard has strips of metal underneath the board and connect the holes on the top of the board. The metal strips are laid out as shown below. The top and bottom rows of holes are connected horizontally and split in the middle while the remaining holes are connected vertically.

3.1.4 Connecting Wires

Fig 3.4 Connecting wires

Connecting wires allows an electrical current to travel from one point on a circuit to another because electricity needs a medium through which it can move. Most of the connecting wires are made up of copper or aluminum. Copper is cheap and good conductivity. Instead of the copper, we can also use silver which has high **conductivity** but it is too costly to use.

3.1.5 Led light bulb

Fig 3.5 Led Light Bulb

An LED light bulb is a solid-state lighting (SSL) device that fits in standard screw-in connections but uses LEDs (light-emitting diodes) to produce light. LED light bulbs are a more environmentally-friendly alternative to incandescent bulbs. LED bulbs use a semiconductor device that emits visible light when an electric current passes through it.

3.1.6 Breadboard power supply

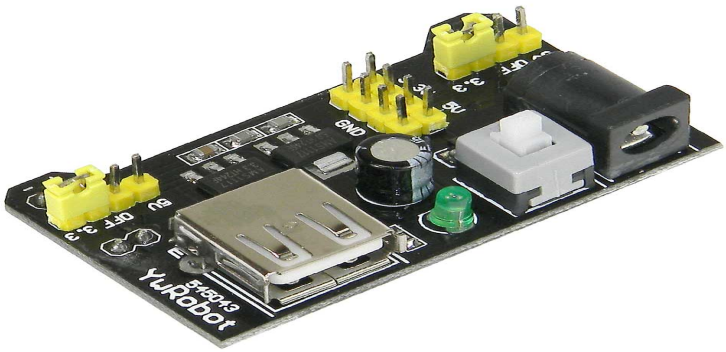


Fig 3.6 Breadboard power supply

A breadboard power supply is a companion module, which provides 5 V, 3 V, and 12 V rails on a solderless breadboard. It is an extremely useful facility, which provides regulated voltage rails for project circuits. The module design is usually in the form of a plug-in, which connects onto the breadboard, and the power to the module usually comes from an adapter or USB port of a PC. Some modules provide a +5 V regulated output through a female USB socket to power other USB devices, which can be extremely useful. Some modules also allow the board to receive power through the same socket.

3.1.7 Automatic Night Light

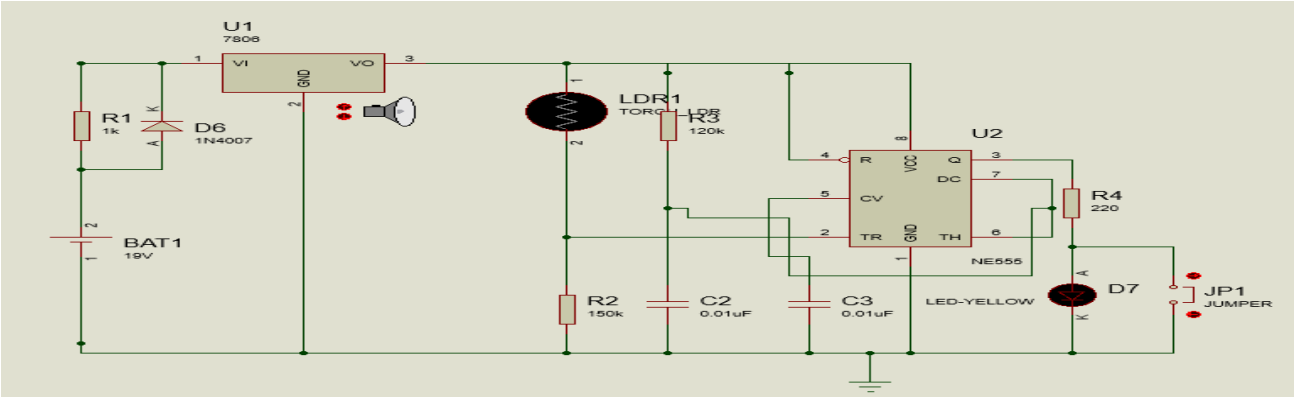


Fig 3.7 Automatic Night Light on proteus software

This circuit automatically turns on a night lamp when bedroom light is switched off. The lamp remains 'ON' until the light sensor senses daylight in the morning. A super-bright white LED is used as the night lamp. It gives bright and cool light in the room. The circuit is powered from a breadboard power supply. Regulator IC 7806 gives regulated 6V DC to the circuit. Diode D5 prevents the battery from discharging backwards following the mains failure and diode D6 provides current path from the battery the circuit utilizes light-dependent resistors (LDRs) for sensing darkness and light in the room. The resistance of LDR is very high in darkness, which reduces to minimum when LDR is fully illuminated. LDR detects darkness, while the circuit is designed around the popular timer IC NE555 (IC2), which is configured as a monostable.
IC2 is activated by a low pulse applied to its trigger pin 2. Once triggered, output pin 3 of IC2 goes high and remains in that position until IC2 is triggered again at its pin 2.

Chapter 4
Software and platforms used

4.1 Software

In this project we are using the Arduino IDE software to program the NodeMCU ESP8266 micro controllers and Adafruit IO to control the system.

4.1.1 Arduino IDE

Here, the open-source Arduino software (IDE) makes it easy to write code and upload it to the board. It runs on windows, mac os x, and Linux. Figure- shows the Arduino IDE software editing window. The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring.[4] The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution.

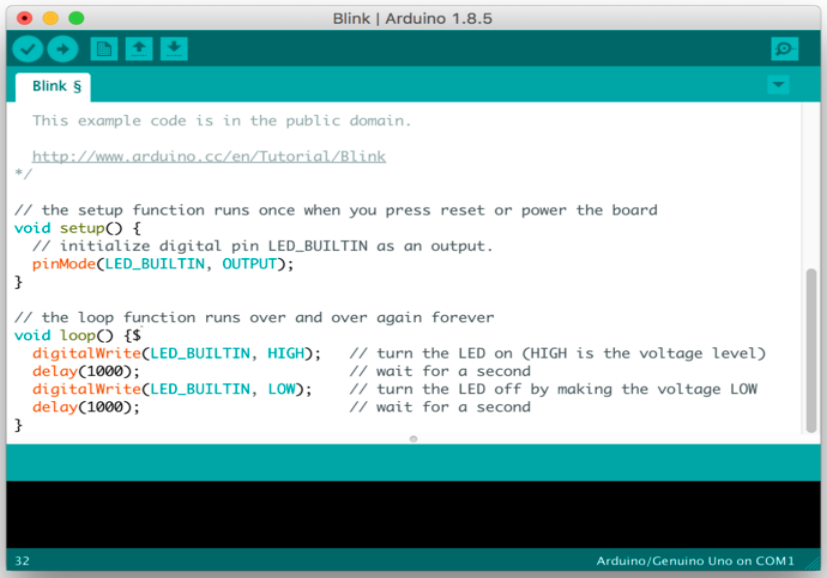


Fig 4.1 Arduino IDE

4.2 Platforms

To build home automation application, we used three different platforms

- Google Assistant
- Adafruit
- IFTTT

4.2.1 Google Assistant

Google assistant is AI (Artificial Intelligence) based voice command service. Using voice, we can interact with google assistant and it can search on internet, schedule events, set alarms, control appliances, etc. This service is available on smartphones and Google Home devices. We can control smart home devices including lights, switches, fans and thermostats using our Google Assistant. Here, we control two LED bulb a 5V DC fan and Automatic Night Lamp using Google Assistant service. This application includes Google assistant along with Adafruit server and IFTTT service.

Fig 4.2 Google Assistant

To use IFTTT and Adafruit IO we need to configure them.

4.2.2 Adafruit IO

First, created account at www.Adafruit.io

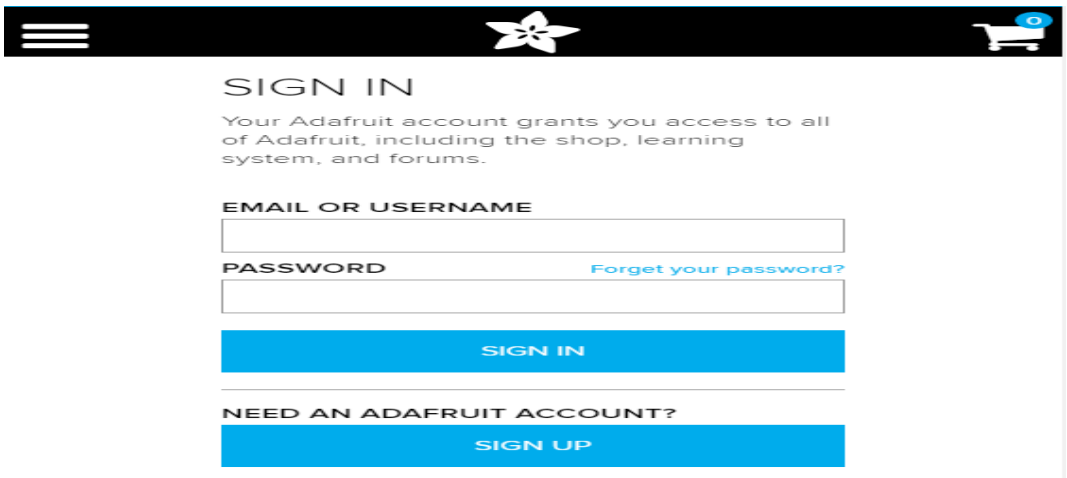


Fig 4.3 creating Adafruit account

Now, create dashboard at Adafruit. This dashboard is a user interface to control things remotely.

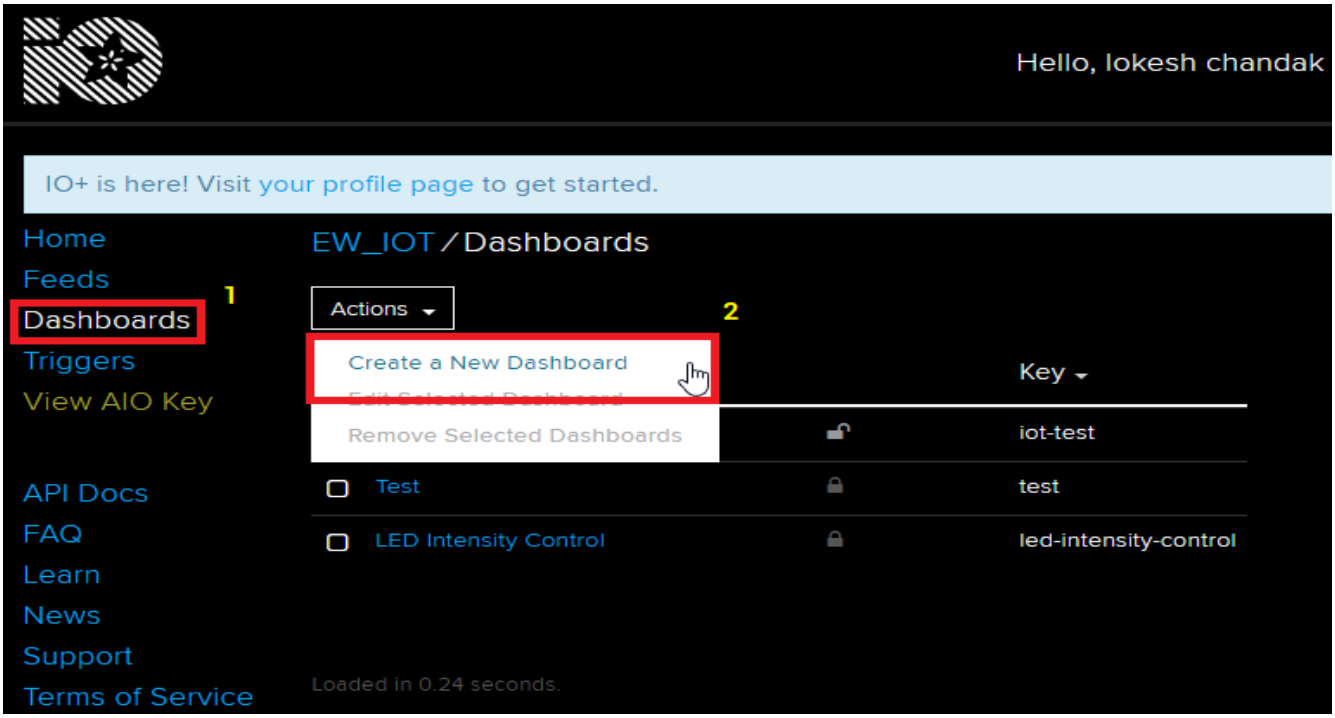


Fig 4.4 creating Dashboard

After following above steps, provide name to the dashboard and save it. We can see our dashboard as follows,

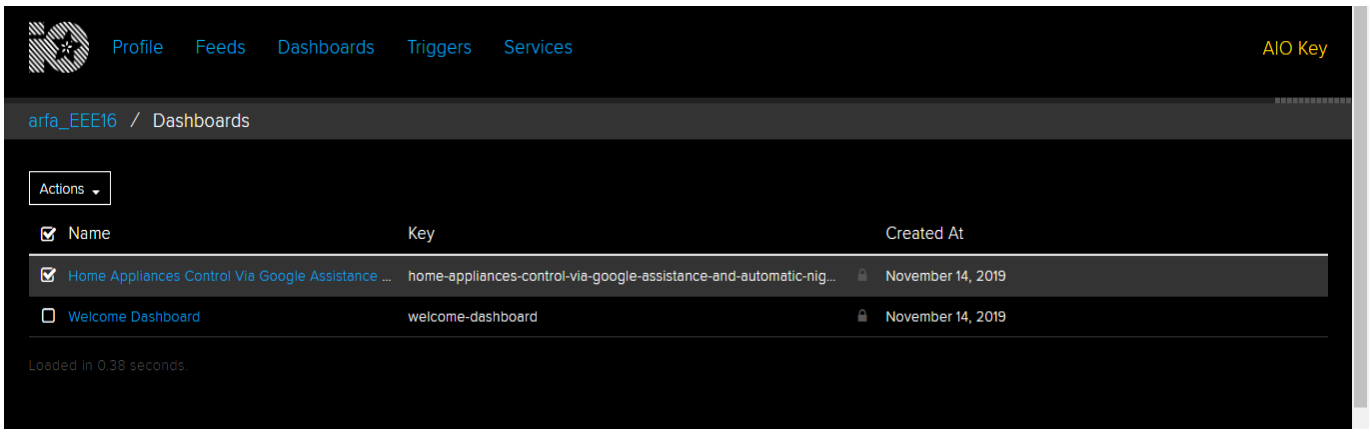


Fig 4.5 Dashboard display

Now, create feed (user interface) to control light On-Off. To create it, just click on **'+'** symbol and select toggle feed shown below,

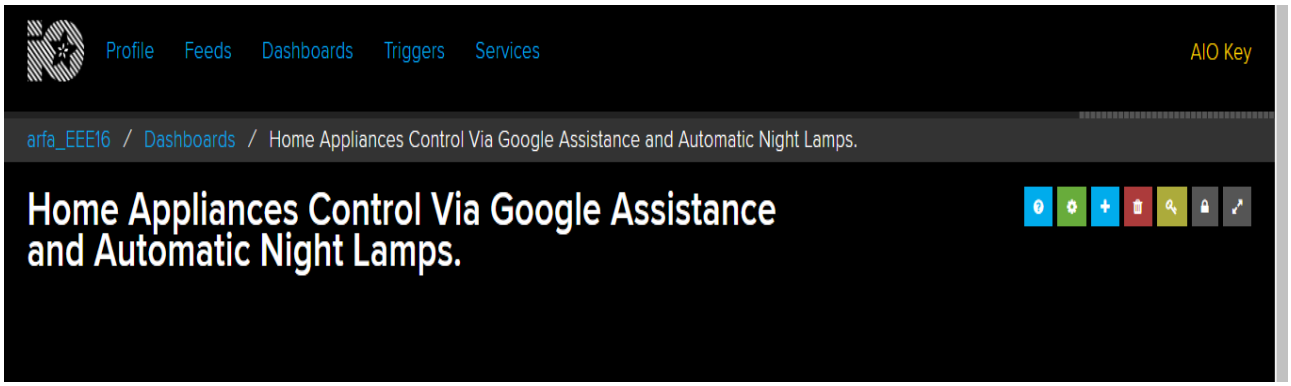


Fig4.6 creating feed

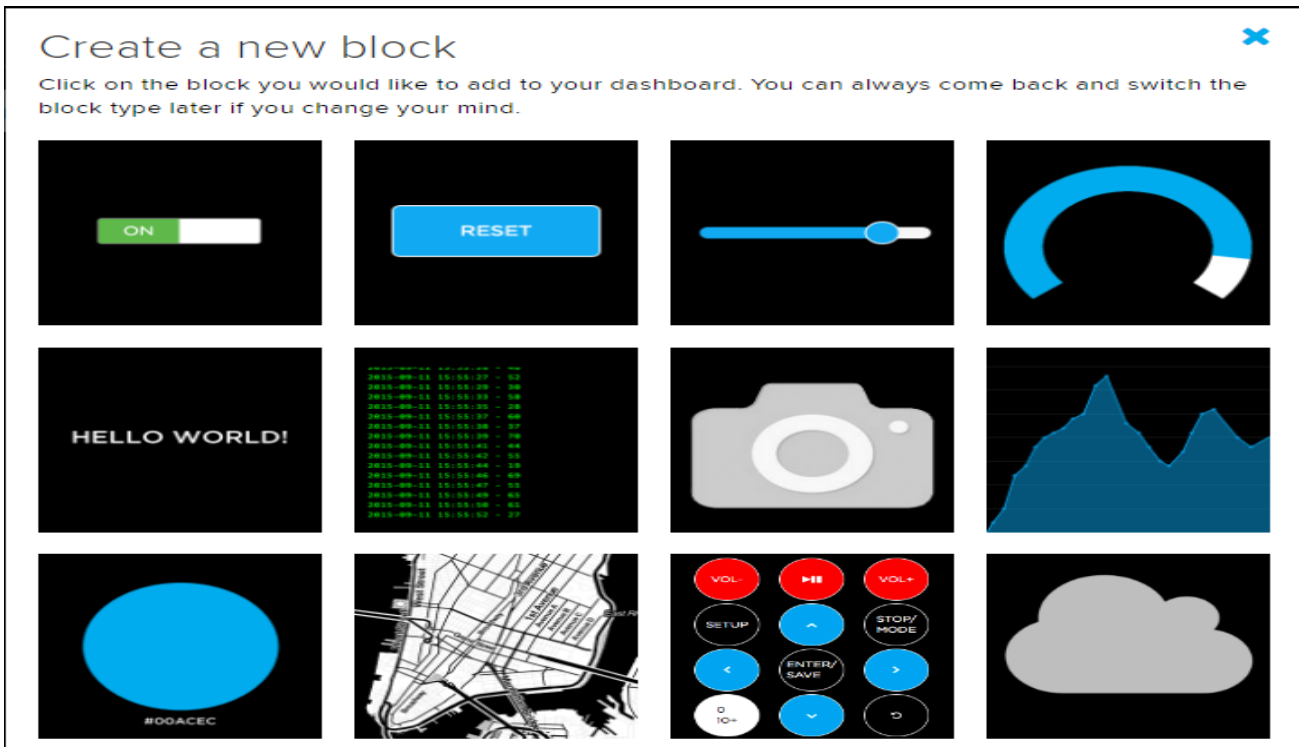


Fig 4.7 Toggle feed

After selecting toggle feed, pop-up window appears as shown below.

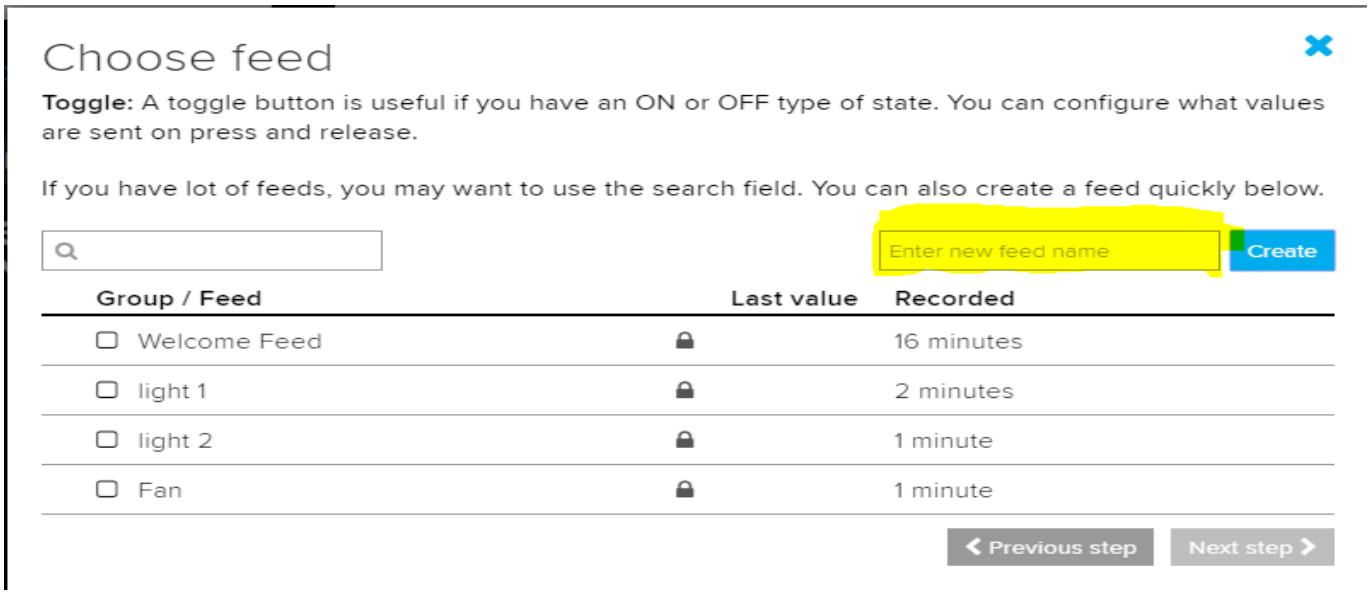


Fig 4.8 list of created feed

Enter name of our feed (shown by the highlighter) and create it. After creation, select the created feed (here mine is light**1**) and then click on **Next step.**

In the next step configure the feed which is shown below,

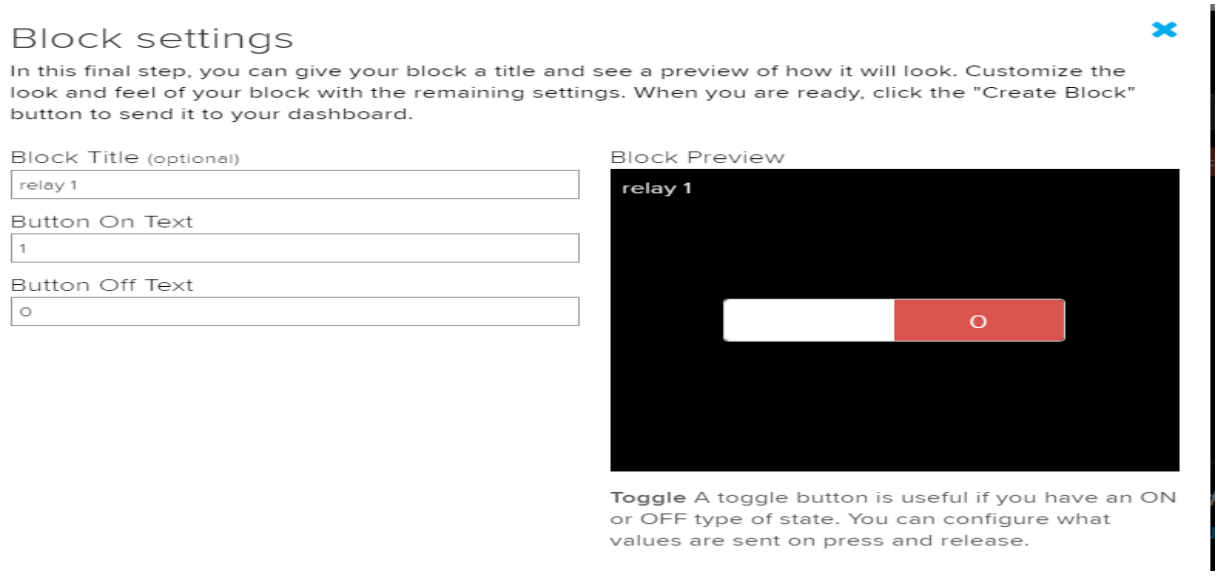


Fig 4.9 setting a feed

Here, we used **0**(OFF) and **1**(ON) text for button and then click on create. This will create toggle button on your dashboard which can be used to control things remotely. After following same steps, you can create and set more feeds. In our case we created more 3 which change our Dashboard as follow,

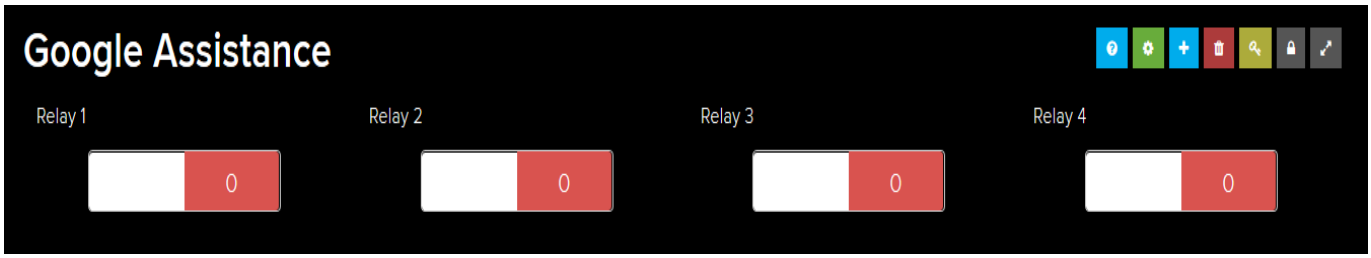


Fig 4.10 Dashboard with all control switches

Now, my dashboard is ready for IoT application like home automation.

4.2.3 IFTTT (If This Then That)

If This Then That, also known as IFTTT 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, or 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. Here, I used IFTTT to use google assistant service and Adafruit service in chain. So, when I use google assistant to control light of my home by saying Ok google, turn the light ON or OFF. Then IFTTT interpret the message and can send it to Adafruit's dashboard as an understandable command to the created feed.

4.2.3.1 Configure IFTTT

First step is creating account on IFTTT

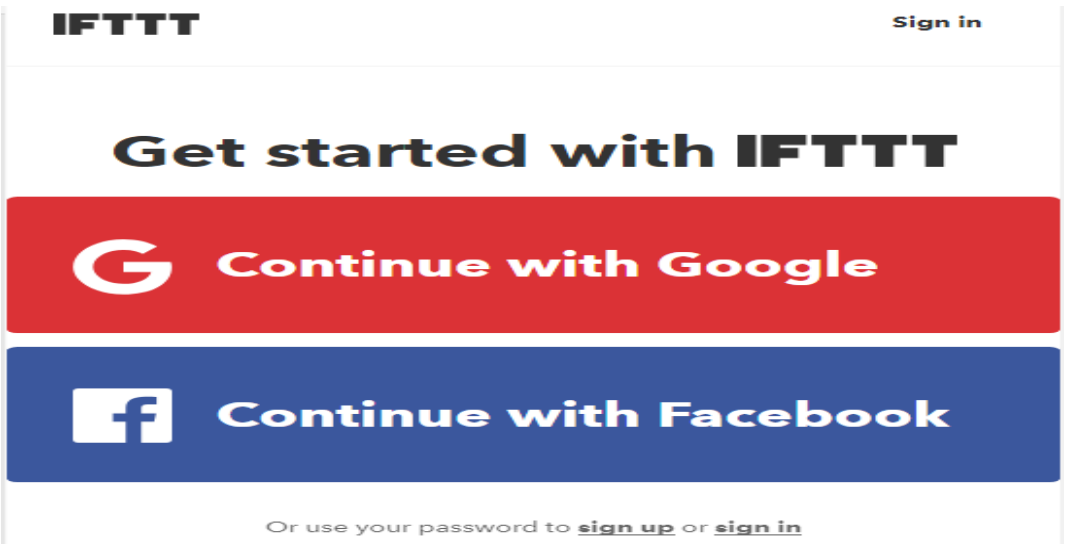


Fig 4.11 creating IFTTT account

Note: Create account on IFTTT by using same e-mail id which you have used for Adafruit.

After account creation, click on **My Applets** and then select **New Applet** shown below

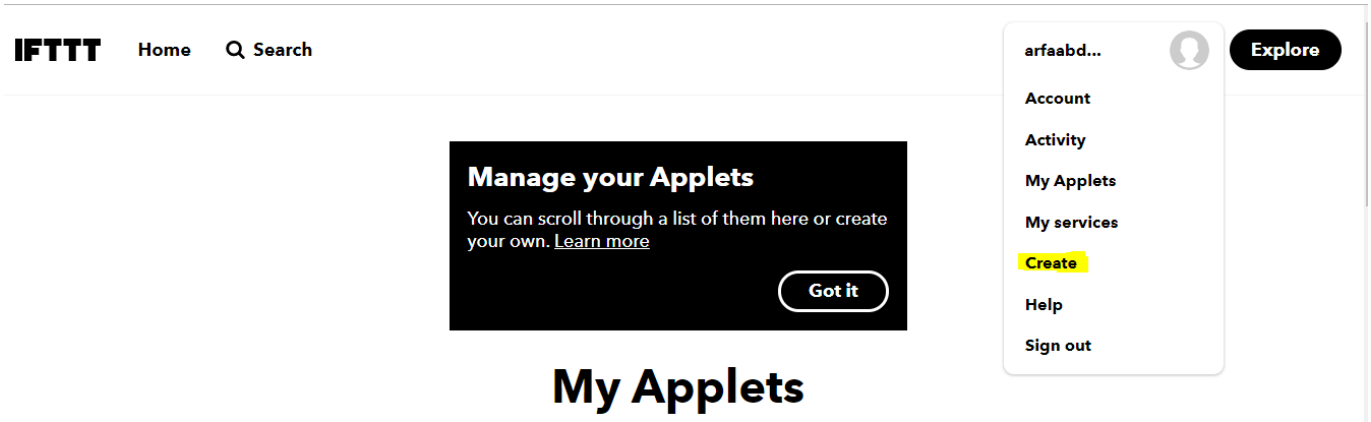


Fig 4.12 creating Applet

After clicking on create button we get a new page in which we should click on to **This** as shown in below image.

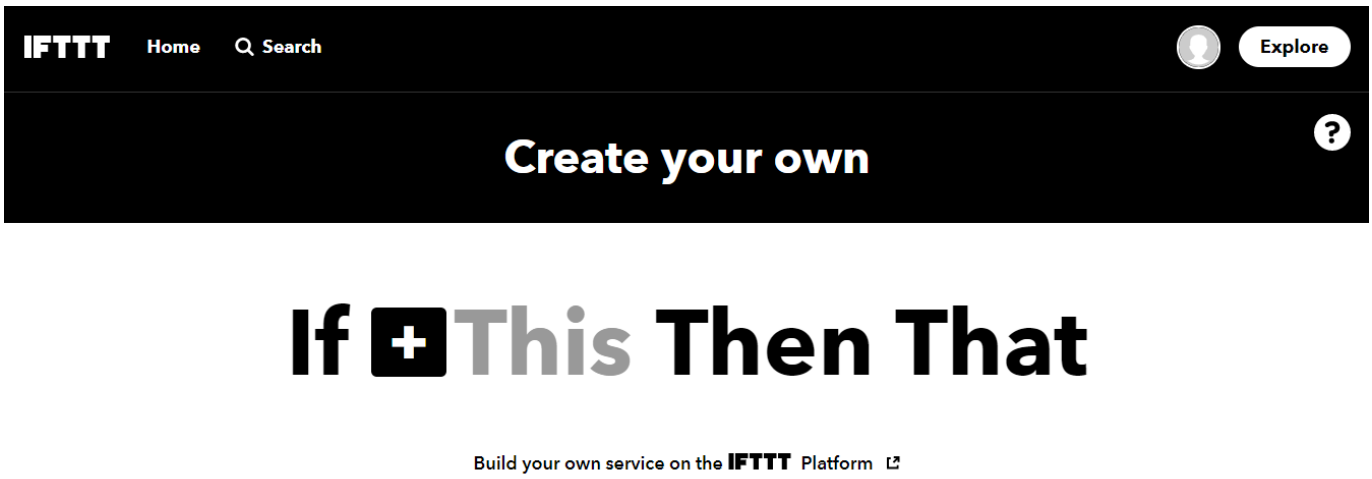


Fig 4.13 crating a trigger

Click on 'THIS' then search for **Google Assistant** and select it.

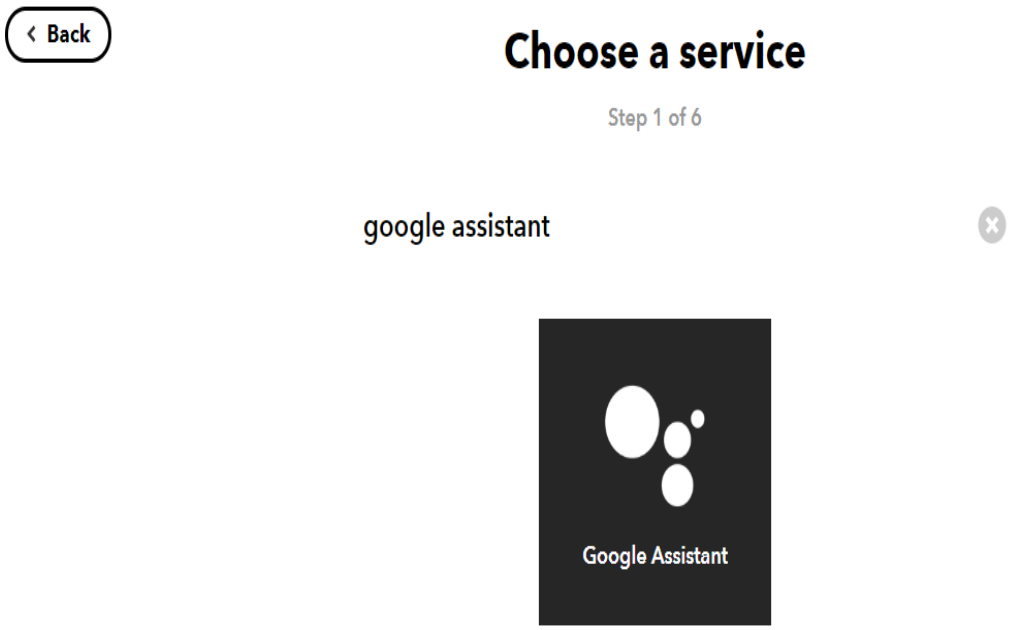


Fig 4.14 creating the trigger

Now, enter voice phrases which we will use as a command for google Assistant.

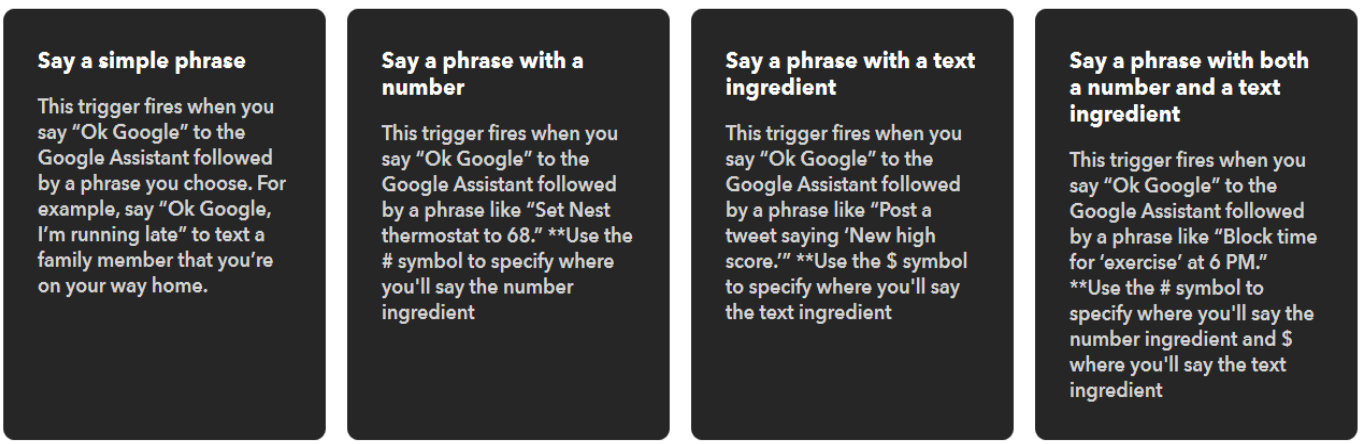


Fig 4.15 different trigger options

Select the first one and fill it according to your need. In our case we created and fill them according to light 1, light 2 and fan. Then after clicking you will receive the following window,

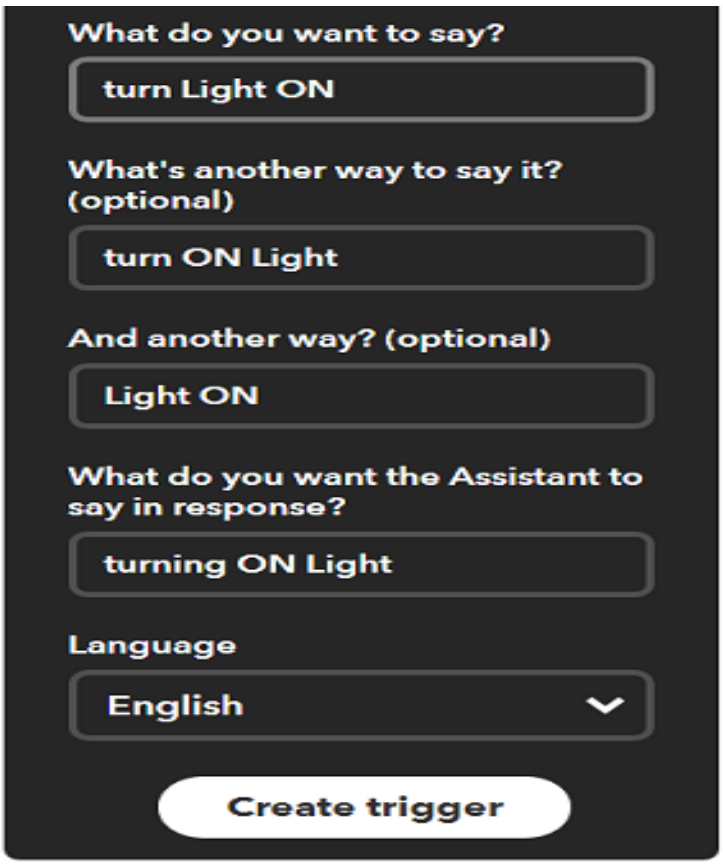


Fig 4.16 setting a trigger for on

We can enter any phrase as per our application. As you can see, the phrases entered in the above fields is for making **Light ON.** For making **Light OFF**, we have to create another applet with different phrases.

Now, we get another page on which we have to click on **that** option which is used to connect Google Assistant with Adafruit.

Fig 4.16 creating action

Click on “THAT” and search for Adafruit. After selecting Adafruit, choose action as shown below,

Fig 4.17 creating action

Now enter what data we need to send to which feed of Adafruit dashboard

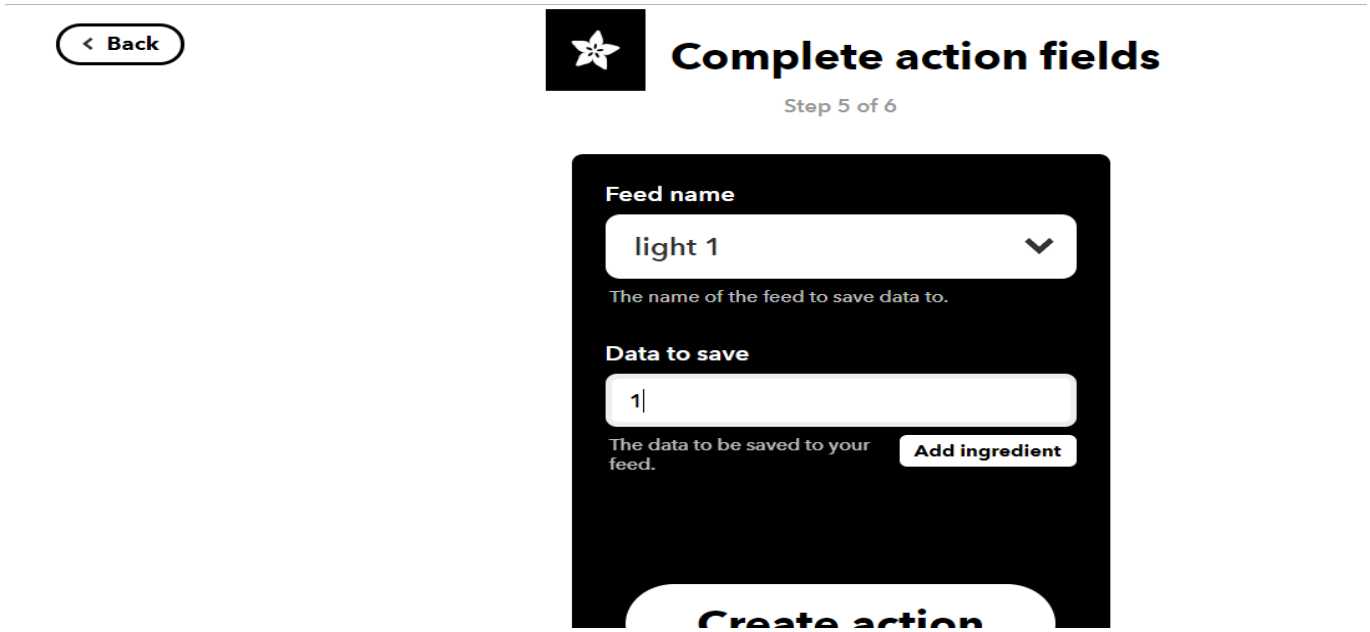


Fig 4.18 setting action

Similarly, we have created an Applet for turning off the light and others to act on respective devices for this project.

Click on **Create Action.**

So, when we use Google Assistant on my mobile and give voice command as “Ok Google, Turn Light 1 ON”, applet created in IFTTT receive this command and will send data ‘1’ to the Adafruit feed. This will trigger the event on Adafruit dashboard which is continuously monitored by the microcontroller (here NodeMCU). This microcontroller will take action as per the data change on the Adafruit dashboard.

4.2.4 Analysis of the physical system

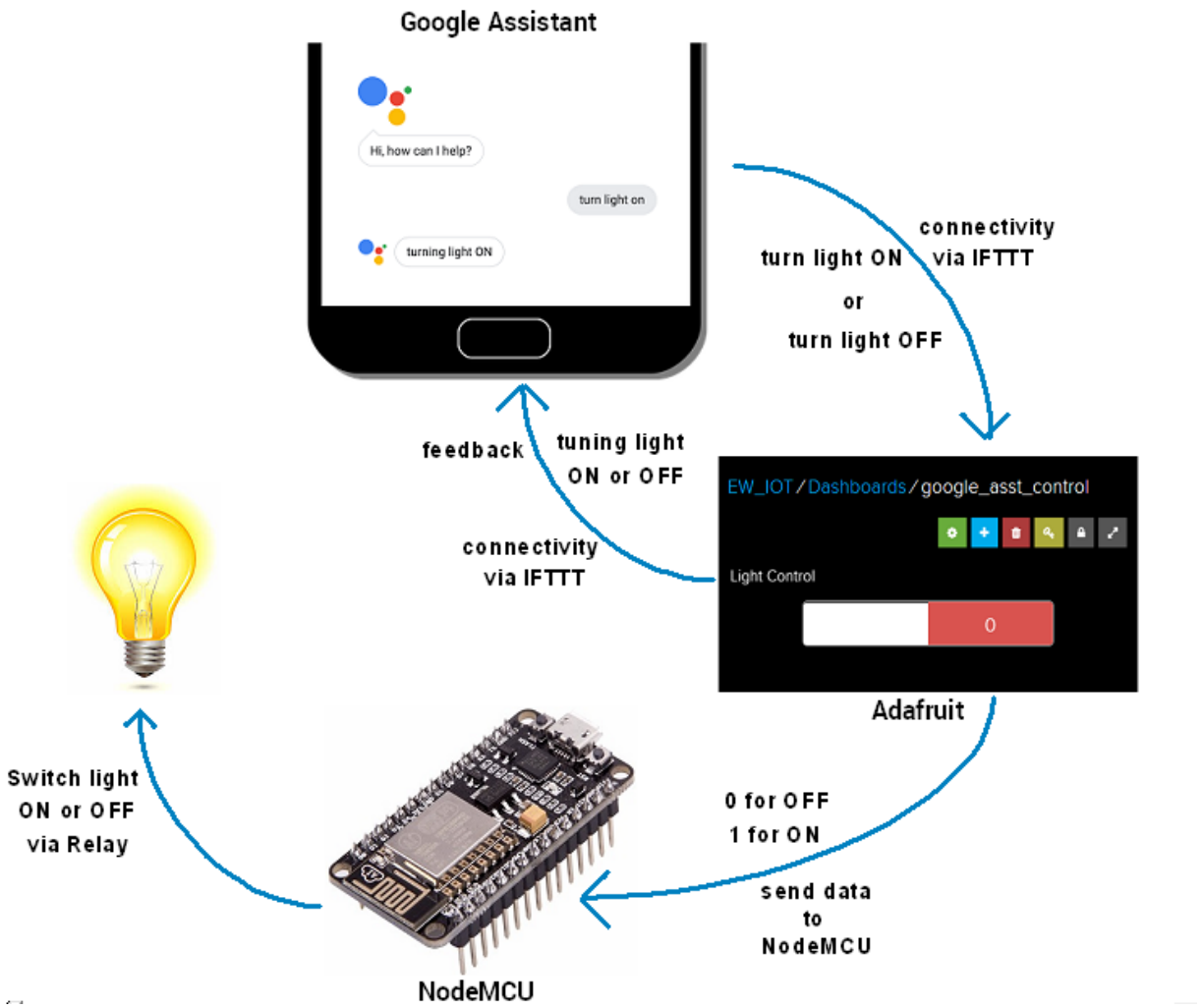


Fig 3.8 simplified system

Here we be describing how we control all devices which are connected to our system by voice through google assistant. Google assistant is a very powerful AI build by google. We have added this feature to our project so that our system will be a lot of machine-controlled and smart as well. We are using NodeMCU Wi-Fi module for this purpose where it is connected through home Wi-Fi network of user or any WIFI provider such as smart phones. In this case, user just need to say on his phone “Ok google” to turn google assistant on and then voice control feature of our model will be activated. When any instruction is given to google assistant by saying “Ok google” it sends request to IFTTT. IFTTT is a free web-based service to create chains of simple conditional statements, called applets. IFTTT is connected through our google account and when google assistant finds the match, it passes the instruction to NodeMCU. Relays are connected with the NodeMCU Wi-Fi module. After getting the instruction from
IFTTT, NodeMCU starts executing the instruction through the relay.

Applications:

IoT based home automation can revive the way people use technology. There is a considerable range of possibilities when we speak about applications of home automation.
We'll start with the basics. The most common applications *of* homeautomation are **lighting control, HVAC, outdoor lawn irrigation, kitchen appliances**, and **security systems.**

- ## Lighting Control: Leaving the Dark Ages and Stepping Into the Light

  Smart lighting allows you to control wall switches, blinds, and lamps, but how intuitive is a lighting control system? It turns out, quite; its capabilities are extensive. You're able to schedule the times lights should turn on and off, decide which specific rooms should be illuminated at certain times, select the level of light which should be emitted, and choose how particular lights react through motion sensitivity, as seen with Belkin's WeMo Switch + Motion, which is both affordable and easy to use with its plug-and-play simplicity.

- ## HVAC Regulation: No Longer Burned by Your Heating Bill

  As fuel costs rise and the availability and sustainability of our resources becomes a greater concern, heating/cooling our homes efficiently is less a budgetary bonus and more of a necessity. Over the past year, smart thermostats and automated home heating systems have become more readily available and easily incorporate into any home. Heating and cooling our homes consumes an average of 50% of energy costs yearly, making daily HVAC regulation progressively rewarding. Maintaining a substantial lead among the nearly non-existent competition, the Nest Learning Thermostat, learns your heating and cooling preferences over time, eliminating the need for programming and is accessible from your smartphone app. With automated HVAC you are able to reduce the heat when a room is unoccupied, and increase or decrease it at specific times based on your schedule and occupancy.

- ## Lawn Irrigation Systems: The Grass is Always Greener

  A lush and healthy lawn is a source of pride for most homeowners, but the weather doesn't always cooperate and provide the adequate elements for a flourishing landscape. For decades we've relied on sprinkler systems to keep our yards at peak presentation, but at what cost? The average American home spends approximately 30% of their daily water usage on lawn and garden maintenance. Nearly half of that amount is wasted due to inefficiency. If you apply that statistic to the national average, up to 4.5 billion gallons of water is wasted per day through ineffective watering methods. If we reflect upon the

monetary impact of this, it results in Americans spending over a thousand dollars a year in water, with a portion of that being waste. The global effects are even greater when you consider the growing concern over climate change and the dramatic decrease in agricultural natural resources. However, sprinkler control systems, like Skydrop, are providing water regulation through real-time communication with local weather data. If a rainstorm develops and deposits two inches of rainwater on your lawn, the automated sprinkler detects the saturation and disables its scheduled watering. Conversely, the system will be alerted to dry conditions and supply the necessary amount of nourishment, without over-watering.

- Smart Appliances

  Will smart kitchen appliances actually make you a better cook? Maybe. Smart refrigerators, such as LG's Smart ThinQ, allow you to scan grocery store receipts and keep an inventory of your items, and alerts you if an item is about to expire. More impressively, it suggests recipes based on your refrigerator's contents and lets you know when you need to replace items. Smart ovens synch with your smartphone and automatically preheat to the correct temperature based on a recipe selected from your database. While these appliance options seem a bit superficial and convenience based, there is a conservation factor as well. By automating your kitchen appliance and making them accessible from your smart device, you're able to sever the electricity supplied to unused appliances and reduce your energy consumption and costs. Considering the number of appliances the average household owns; this could save a substantial amount of money over time

- Security Systems

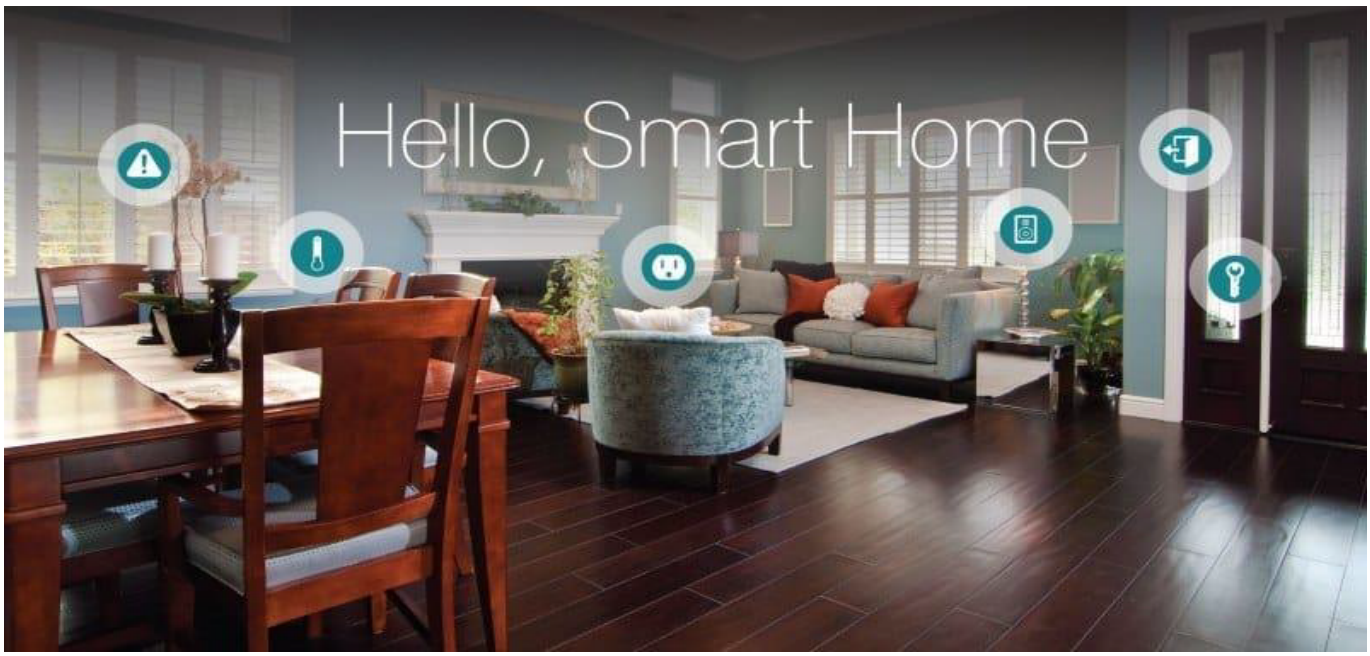
  Smart locks, like Kwikset's Kevo, a Bluetooth enabled electronic deadbolt, and various connected home security systems, such as iSmartAlarm, offer a variety of features including door and window sensors, motion detectors, video cameras and recording mechanisms. All of which are connected to a mobile device and accessible via the cloud, thus enabling you to access real-time information on the security status of your home. Naturally, there is a great deal of scrutiny regarding the level of trust in controlling your home's security system via a mobile device, but it begs earnest exploration when weighing the potential benefits and peace of mind it provides homeowners.

Conclusion:

The system as the name indicates, 'Home automation' makes the system more flexible and provides attractive user interface compared to other home automation systems. In this system we integrate mobile devices into home automation systems.

The system consists of mainly three components which are NodeMCU, Google Voice Assistance and relay circuits. Internet Connection Wi-Fi is used as the communication channel between android phone and the NodeMCU
Hence, the proposed system is better from the scalability and flexibility point of view than the commercially available home automation systems.
With that said, we can conclude that the required goals and objectives of home automation system have been achieved.

Future of the Home Automation



In the near future smart home system will become as common as the internet today, Otherwise it will be outdate just like homes nowadays without internet.

References

i. Wikipedia

ii. Kumar, Abhijeet, and Apoorva Sharma. "Internet of Life (IOL)." (2015). ISBN 978-93-5156-328-0*Brian Benchoff. "An SDK for the ESP8266 Wi-Fi chip". Hackaday. Retrieved 2 April 2015.*

iii. Vowstar. "NodeMCU Devkit". *Github*. NodeMCU Team. Retrieved 2 April 2015.

iv. "IFTTT Platforms" ULR:

v. https://www.mpja.com/33V-5V-Breadboard-Power-Supply/productinfo/30175%20PS/

vi. IFTTT: https://ifttt.com/discover
https://www.pocketlint.com/SmartHome/SmarHomenew

vii. Google Assistant: https://assistant.google.com/intl/en_in

viii. Arduino IDE: https://www.arduino.cc/en/Guide/Environment

ix. "Home Automation Using IoT" ULR: https://dzone.com/articles/home-automation-using-iot

x. "Google Assistant Platforms" ULR: https://en.wikipedia.org/wiki/Google_Assistant

xi.

Image References:

i. "Google Image" ULR:
https://www.google.com/search?q=IOT+Images&client=firefox-b-d&sxsrf=ACYBGNSnh891w4ckluSyy-l-9pZs6lucuw:1573527118186&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiQ1LGn1ePlAhUGU30KHfy3DEcQ_AUIESgB&biw=1366&bih=664