



ISLAMIC UNIVERSITY OF TECHNOLOGY, IUT
ORGANISATION OF THE ISLAMIC COOPERATION, OIC
BACHELOR OF SCIENCE IN TECHNICAL EDUCATION
HOME AUTOMATION WITH GOOGLE ASSISTANT



PRESENTED BY:

OUN ALI	190033201
BILAL MOSLEH AZIZ	190033206
MUJAHID HAMEED	190033207

SUPERVISED BY:

Dr. MUHAMMAD REZAUL HOQUE KHAN
(ASSOCIATE PROFESSOR, EEE DEPT)

Department of Technical and Vocational Education (TVE)

MARCH – 2020-21

Certification:

This is to certify that this project report entitled "HOME AUTOMTION WITH GOOGLE ASSISTANT" is a true work of Oun Ali(190033201) , Bilal Mosleh Aziz (190033206) , Mujahid Hameed (190033207), who successfully carried out the work project under the supervision of Dr. Muhammad Rezaul Hoque Khan ,EEE Department, Islamic university of Technology (IUT).This project counts as our final year project which put an end to our one year program as B.Sc.TE (BACHELOR OF SCIENCE IN TECHNICAL EDUCATION) at the Islamic University of Technology (IUT) Dhaka, Bangladesh.

Author:


OUN ALI

190033201 

BILAL MOSLEH AZIZ

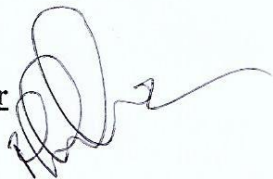
180033206 

MUJAHID HAMEED

190033207 

Date:

Supervisor



Name: Dr. Muhammad Rezaul Hoque Khan

Signature:

Date: 16/03/21

Head of department

Name: Prof. Dr. Md. Abu Raihan

Signature:

Date:

Department of Technical and Vocational Education (TVE)
Islamic University of Technology (IUT)

Abstract

Comfort is becoming a major priority in the 21st century. So, the revolutions of computing and smart environment came into existence. Some technologies like Ubiquitous/pervasive and ambient intelligence satisfy the maximum need of smart world but these technologies are not tightly coupled with the internet, so the people need another technology extension. Internet of Things (IoT) is an ideal buzzing technology to influence the internet and communication technologies. IoT allows people and things to be connected anytime, anyplace, with anything and anyone, by using ideally in any path/network and any service. Home automation system achieved great popularity in the last decades and it increases the quality of life. In this thesis, an overview of home automation systems is discussed. This thesis presents a design and prototype of the Home Automation system. The main part is NodeMCU, which has inbuilt Wi-Fi module, which will help in controlling devices over the Internet. It supports a wide range of home automation devices like power management systems. In addition to that, an app would be developed which will allow the user to control their devices using the Google Assistant.

Acknowledgements

We thank the most merciful Allah (SWT), by whose grace and blessing has spiritually help and give us strength to accomplish and complete this project. We would like to extent our sincerest Gratitude and thanks to our esteemed supervisor **Dr. Muhammad Rezaul Hoque Khan** Associate professor,

Department of Electrical and Electronics Engineering. He is not only helpful supervisor with deep vision but also most importantly a kind person. We sincerely think him for his guidance and encouragement. His trust and support inspired us in the most important moment of making right decisions and it was a great pleasure for us to work with him. We would like to thanks our beloved parents, who taught us the value of hard work by their own example. Lastly, we would also like to thanks all those who assisted us during this long journey of work.

Table of contents

Certification

Abstract

Acknowledgements

Table of contents

List of figures

Abbreviations

1. Chapter One: Introduction

1.1 Introduction

1.2 Problem Statement

1.3 Proposed Solutions

1.4 Aim and Objectives

1.5 Methodology

2. Chapter Two: Literature Review

2.1 Background

2.2 Review for foreign study

3. Chapter Three: Design and Implementation

3.1 Overview of the Smart Home System

3.2 System Description

3.3 Hardware Design

3.4 Software Design

3.5 Implementation

4. Chapter Four: Result and Discussion

4.1 Results

4.2 Discussion

5. Chapter Five: Conclusion and Future Work

5.1 Conclusion

5.2 Future Work

References

Appendix

ABBREVIATIONS

IoTs	Internet of Things
IFTTT	IF THIS THEN THAT
PIN	Personal Identification Number
IDE	Integrated Development Environment
Wi-Fi	Wireless Fidelity
GACHA	Google Assistant Controlled Home Automation
HTTP	Hyper Text Transfer Protocol
OS	operating system
WPANs	wireless personal area networks
PCB	Printed Circuit Board
USB	Universal Serial Bus
SMS	Short Message Service
TCP	Transmission Control Protocol
IP	Internet Protocol

List of Figures

Figure 1: IoT Objectives.....	09
Figure 2: ESP8266 WIFI Module.....	17
Figure 3: PIN Diagram Explanation of ESP8266.....	17
Figure 4: PIN Explanation of ESP8266	18
Figure 5: Relay Board.....	19
Figure 6: PIN Explanation of Relay Board.....	20
Figure 7: Arduino IDE.....	21
Figure 8: Functioning of Blynk App.....	22
Figure 9: IFTTT Application.....	24
Figure 10: Trigger creation in IFTTT.....	24

CHAPTER ONE

INTRODUCTION

1.1 Introduction

Every day modern people expect a new device and new technology to simplify their day to day life. The developers and researchers are always trying to find new things to provide comfort to the people. In the past 1990s, Internet usage increased rapidly in enterprise and consumer markets but was still limited in its use because of the low performance of the network interconnects. But in the 2000s Internet connectivity became a part for many applications to provide access to information. However, these devices require more human interaction and monitoring through apps and interface. The Internet of Things is a new era of intelligence computing and it's providing a privilege to communicate around the world. Figure 1 describes the coupling of C's and A's. That reveals people and things can be connected, ideally by using in Any path/network and any service.



Figure 1: IoT Objectives

1. Resource usage (water conservation and energy consumption)
2. Security and
3. Comfort.

This thesis introduces an IoT agent which is the brain of this architecture and it controls the devices through the internet. At a very short period of time, the IoT agent receives the user data in the form of commands, which will be sent to control the systems through Google commands. This command will be received by Google assistant and control the devices.

1.2 Problem Statement

When people left their homes, number of problems may exist, this include:

- Saving power from wasting electricity when forgetting electrical appliances open. Also saving effort and comfort for handicaps.
- Monitoring the appliances for protection from causing fires or Electric stuns.
- Monitoring the entries of the house to provide safeness from thieves or any interlopers.

The main problem is the lack of a user's ability to control home appliances more efficiently from anywhere.

1.3 Proposed Solution

The proposed solution is to provide remote control and monitoring system for home appliances and environment. The system uses NodeMcu with internet connectivity to communicate with Smartphone, the NodeMcu connected to household devices to perform control and monitor functions.

1.4 Aim and Objectives

Aim:

The aim of this project is to develop prototype for a Smart Home System that will enable homeowners to control and monitor electric devices such as lamps, fans, and other devices by using Blynk Application and Google Assistance.

Objectives:

- To design smart home based on Nodemcu that will run as central hub in home.
- To provide the communications mechanism between Smartphone and Nodemcu via internet.
- To develop software for Nodemcu, that will enable to control and monitor household devices from Smartphone application.

1.5 Methodology

The project describes the implementation of a system 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 Blynk application 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).

CHAPTER TWO
LITERATURE REVIEW

2.1 Background

When people think about home automation system, most of them get to a thought of living in a smart home: One remote controller for every household appliance, cooking the rice automatically, starting air conditioner automatically, heating water for bath automatically and shading the window automatically during the night. To some extent, home automation equals to the smart home. Then automatically these will bring out smart living condition and make our life more convenient and faster. Early home automation began with laborsaving machines. Self-contained electric or gas-powered home appliance became viable in the 1900s with the introduction of electric power distribution led to the introduction of washing machine (1904), water heater (1889), refrigerator, sewing machines, dishwashers and clothes dryers. As per our survey currently there exists system neither at cheaper rates nor easy to handle. Various systems are hard to install, difficult to use and maintain. Current systems are generally proprietary, closed and not very user friendly Based on Arduino or GSM or low-cost home security system and home automation system.

2.2 Review of Foreign Study

In their paper, Tan, Lee and Soh (2002) proposed the development of an Internet-based system to allow monitoring of important process variables from a distributed control system (DCS). This paper proposes hardware and software design considerations which enable the user to access the process variables on the DCS, remotely and effectively rent designations. Potamitis, Georgila, Fakotakis, and Kokkinoss, G. (2003) suggested the use of speech to interact

remotely with the home appliances to perform a particular action on behalf of the user. The approach is inclined for people with disability to perform real-life operations at home by directing appliances through speech. Voice separation strategy is selected to take appropriate decision by speech recognition. In the year 2006, S. M. AnamulHaque, S. M. Kamruzzaman and Md. Ashraful Islam proposed a system entitled “A System for Smart-Home Control of Appliances Based on Time and Speech Interaction” that controls the home appliances using the personal computer. This system is developed by using the Visual Basic 6.0 as programming language and Microsoft voice engine tools for speech recognition purpose. Appliances can be either controlled by timer or by voice command.

Vol-4 Issue-22018IJARIE-ISSN(O)-2395-43968241www.ijariie.com4011Jawarkar, Ahmed, Ladhake, and Thakare (2008) propose remote monitoring through mobile phone involving the use of spoken commands. The spoken commands are generated and sent in the form of text SMS to the control system and then the microcontroller on the basis of SMS takes a decision of a particular task. Prof. Era Johri Dept. Of Information and Technology K.J.Somaiya College of Engineering VIDYAVIHAR, MUMBAI in (2001)have successfully completed the project on “Remote Controlled Home Automation”.

CHAPTER THREE

DESIGN AND IMPLEMENTATION

3.1 Overview of the Smart House System

The project revolves around developing a smart home system, which allows homeowners to control household devices through a Smartphone application. Currently it is focused on being able to turn ON and turn OFF lights, fans, and other electronic home appliances.

This chapter describes the system and its components which might make into the final product and the technologies used for the software part of the project.

3.2 System Description

The structure of the system consists of application installed in Smartphone that connects to Blynk server through the internet to NodeMcu. The NodeMcu interacts with Blynk server to perform controlling and monitoring function by software program written in C/C++ programming language. Lights and household devices are connected to NodeMcu via relay (general purpose input output) pins. Figure 3.1 below describes the structure of the system. The designing of the system is divided into two main categories:

1. The Hardware - Hardware mainly deals with Control Unit which comprises of microcontroller NodeMCU, Relays, Driver IC (used for devices like motors).
2. The Software - The Arduino IDE Software, the Blynk app, the IFTTT app, and the Google Assistant constitute the software of the design and these applications would be integrated in the Android device. The Blynk app on an Android device communicates with the microcontroller and sends the desired signal via the internet.

3.3 Hardware Design

3.3.1 NodeMCU ESP8266 WIFI Module

The ESP8266 is a low cost Wi-Fi chip with full TCP/IP stack and microcontroller unit. This small module allows microcontroller to connect to a Wi-Fi network and make simple TCP/IP connection using Hayes style commands, The ESP8266 with 1MiB of built in flash, allowing for single chip device capable of connecting to Wi-Fi. Fig 2 shows the ESP8266 module structure.



Figure 2: ESP8266 WIFI MODULE

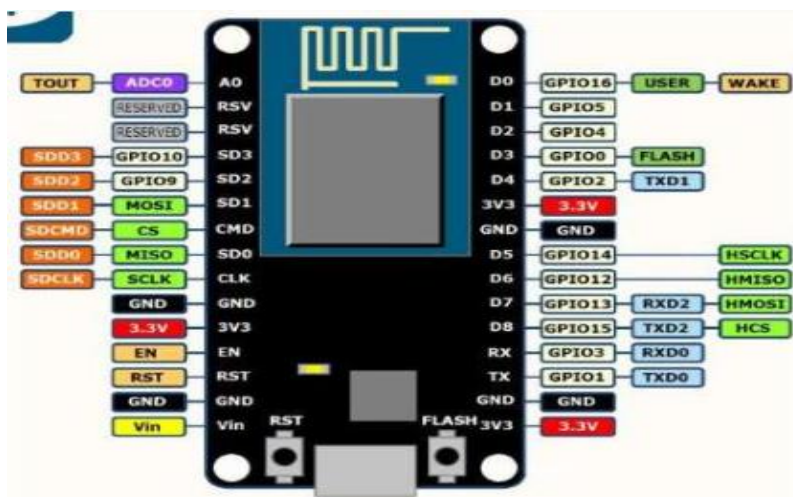


Figure 3: PIN Diagram explanation of ESP8266

The various Features of ESP8266 WIFI MODULE are as follows:

- Processor: L106 32-bit Reduced instruction set computer processor core based on Tensilica Xtensa Diamond Standard 106 micro running at 80 MHZ*.
- 64 KiB of instruction RAM, 96 KiB of data RAM
- External QSPI flash: 512 KiB to 4MiB
- WPA/WPA2 authentication, or open networks
- 16 General purpose input/output pins
- Serial Peripheral Interface Bus
- Inter-Integrated circuit
- Inter-IC sound interface with sharing pins with general purpose input output pins
- 10 bits Analog to digital converter

Name	Type	Function
VCC	Power	Power 3.0 – 3.6V
GND	Power	Ground
RESET	Input	External reset signal
ADC(TOUT) 0-1V	Input	ADC pin analog input
CH_PD Chip	Input	Chip Enable High: on. Works properly
GIPOO(FLASH) Output	Input/output	General purpose input If low while Takes chip into serial Programming mode
GP101(TX) Serial	Input/output	General purpose IO and RXd
GP103(RX)	Input/output	General purpose IO and RXd
GP104	Input/output	General purpose IO
GP105	Input/output	General purpose IO
GP1012	Input/output	General purpose IO

Figure 4: PIN Diagram explanation of ESP8266

3.3.2 Relay Board

A relay is an electrically operated switch. Many relays use an electromagnet to mechanically operate a switch, but other operating principles are also used, such as solid-state relay. Relays are used where it is necessary to control a circuit by a separate low power signal, or where several circuits must be controlled by one signal. Relays were used extensively in telephone exchange and early computers to perform logical operations.



Figure 5: Relay Board

A type of relay that can handle the high power required to directly control an electric motor or other loads is called a contactor. Solid state relays control power circuits with no moving parts, instead using a semiconductor device to perform switching. Relays calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from overload or faults; in modern electric power systems these functions are performed by digital instruments still called “protective relays”.

SR.	QTY.	REF.	DESCRIPTION
1	4	CN1, CN2, CN3, CN4	3 PIN PBT CONNECTOR
2	1	CN5	10 PIN BOX HEADER CONNECTOR
3	1	CN6	2 PIN PBT CONNECTOR
4	2	C1, C3	0.1 uF
5	1	C2	100 uF /25V
6	3	D1, D3, D5	RED LED
7	2	D2, D4	GREEN LED
8	4	RE1, RE2, RE3, RE4	12V GU RELAY
9	5	R1, R2, R3, R4, R5	1K
10	1	U1	ULN2003A
11	1	SOCKET	16 PIN DIP IC SOCKET

Figure 6: PIN Description of Relay Board

3.3.3 Channel Relay Board

Description

4-Channel Relay Board is a simple and convenient way to interface 4 relays for switching application in our project. The project has large Relay which can switch current up to 20Amps. Input voltage level support TTL as well as CMOS. Easy interface with Microcontrollers based projects and analog circuits.

Specifications

- Input supply 12 VDC @ 360 mA
- Output four SPDT Relay

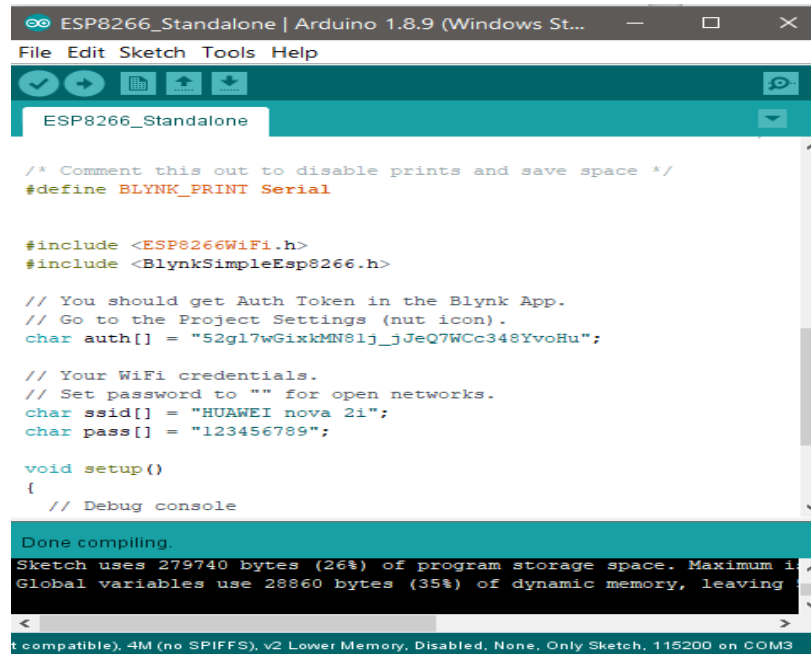
- Relay specification 20 A @ 230 VAC NC/30A NO
- Trigger level 2 ~ 5 VDC
- Box Header connector for connection of trigger signal
- LED on each channel indicates relay status
- Power-On LED indicator
- Screw terminal connector for easy relay output and power in connection
- Four mounting holes of 3.2 mm each
- PCB dimensions 65 mm x 116 mm

3.4 Software Design

The software part consists of mainly Arduino IDE software, Blynk Application, IFTTT Application.

3.4.1 Arduino IDE

Arduino IDE stands for Arduino Integrated Development Environment which is a cross-platform application for Windows, Linux. The programming language mainly used in this Arduino software is Java. Using special code structuring it also supports languages like C and C++. Figure 7 shows the picture of the Arduino IDE software.



```
ESP8266_Standalone | Arduino 1.8.9 (Windows St...
File Edit Sketch Tools Help

ESP8266_Standalone

/* Comment this out to disable prints and save space */
#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "52g17wGixkMN81j_jJeQ7WCc348YvoHu";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "HUAWEI nova 2i";
char pass[] = "123456789";

void setup()
{
  // Debug console
  Serial.begin(115200);

  Blynk.begin(auth, ssid, pass);
}

void loop()
{
  Blynk.run();
}

Done compiling.
Sketch uses 279740 bytes (26%) of program storage space. Maximum is 1024000 bytes.
Global variables use 28860 bytes (35%) of dynamic memory, leaving 51140 bytes free.
(ESP8266 compatible), 4M (no SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3
```

Figure 7: ARDUINO IDE

The Arduino IDE employs the program to convert the executable code into a text file in the hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

3.4.2 BLYNK APPLICATION

The Blynk application was mainly designed for the Internet of Things (IoT). It can control hardware Arduino, Raspberry Pi, NodeMCU and several other boards over the Internet using Android, IOS apps. It can show us the sensor data, store data, visualize it and do many other cool things. Figure 8 shows the functioning of Blynk App. There are three major components in the platform:

Blynk App - It allows us to create interfaces for our projects using various widgets that we can provide to it.

Blynk Server - It is responsible for all the communications between the smart phone and hardware kit. We can use the Blynk Cloud or run private Blynk server. Its open-source, could easily handle thousands of devices.

Blynk Libraries - It is mandatory for all the popular hardware platforms that enable communication with the server and process all the incoming and out coming commands.

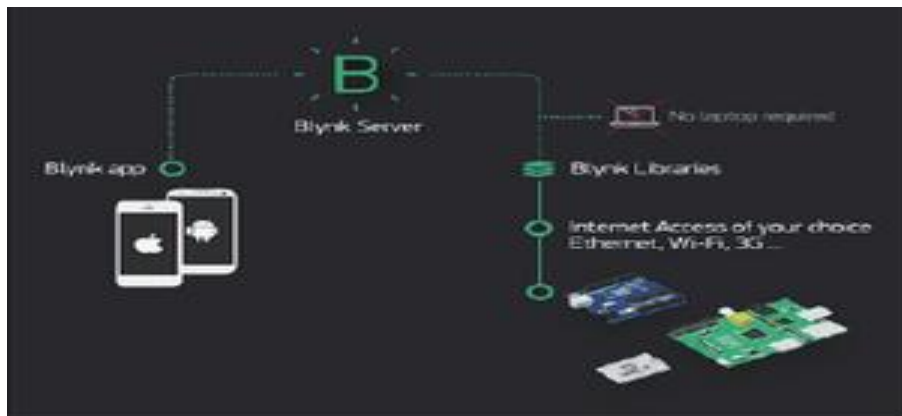


Figure 8: Functioning of Blynk App

Now imagine every time we press a Button in the Blynk app, the message travels to the Blynk Cloud, where it will find its way to our hardware. It works the same in the opposite direction and everything happens in a Blynk of an eye.

3.4.3 IFTTT Application

IFTTT stands for “IF THIS THEN THAT”. IFTTT is a website and a mobile app that was launched in 2010 and has the slogan. The idea of IFTTT is 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 automation involving those apps, devices, and services. Here, IFTTT application will bridge the gap between the Google Assistant commands and the

Blynk app. When we are setting up the IFTTT application first requires logging in after which we need to create an applet and then “This”, i.e. the trigger, we have to select is Google Assistant and then we will type in the commands to which the Google Assistant should respond and to this command it should control the appliance/relay associated with it. The response command from the Google Assistant can also be typed in as desired. After configuring the trigger, Google Assistant hears the command which we just configured. This will be decided by setting “That” of the app. We click “That” and then select webhooks and click connect. Webhooks service will allow us to send commands to the Blynk Server. Now, in the URL we type the IP address of the Blynk server followed by the Authentication token sent by the Blynk and then the pin number of the microcontroller to which the device to be controlled is connected.

The URL should be in the following format:
http://188.166.206.43/AuthToken/pin/CorrespondingDigitalPinNo Then in the method we select ‘PUT’ and the content type is ‘Application/JSON’ and in the body, we write [“0”] to turn ON and [“1”] to turn OFF. This creates the action for the trigger i.e. the Google Assistant command. Finally, the microcontroller is programmed with the actions it needs to do once it receives the signal from the Blynk application. Before that action, the Blynk and the microcontroller should communicate and the communication is done via the internet and since the microcontroller. Figure 9 shows the initial page setup of IFTTT application. Figure 10 shows the trigger creation.

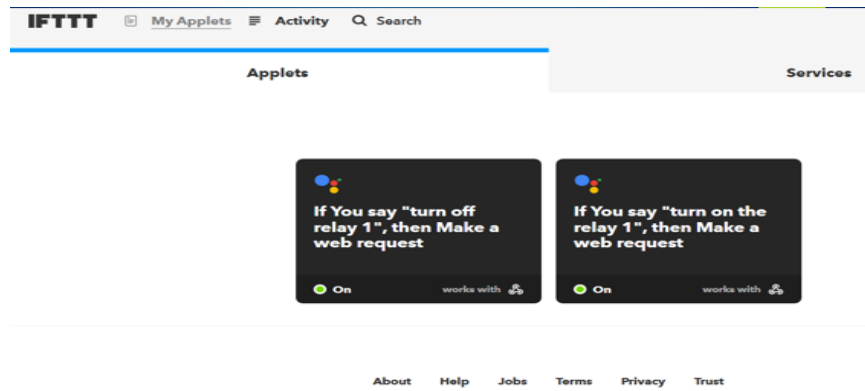


Figure 9: IFTTT Application

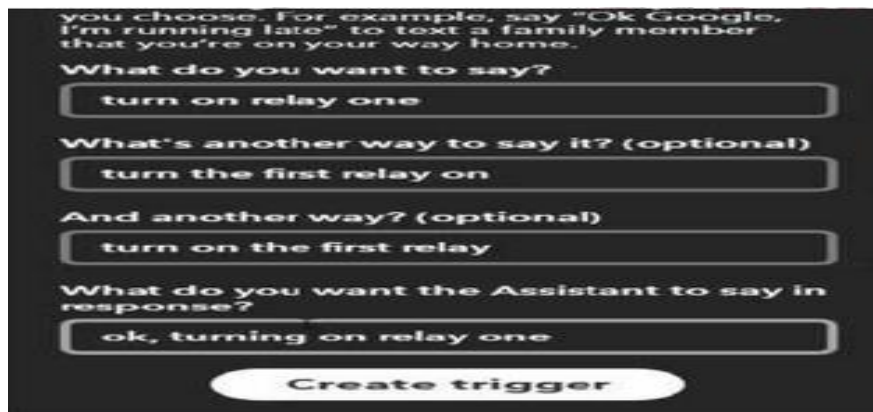


Figure 10: Trigger creation in IFTTT

3.5 Implementation

Step 1: Blynk

- Download the Blynk app from google play store
- Sign in using a valid E-mail address
- Create a new project
- Give a **project name** and choose **NodeMCU** from **choose device** dropdown list.
- Then choose **Wi-Fi** from **connection type** dropdown list and press Create.
- An **Auth Token** will be generated and send to the E-mail address which you signed in.

- Copy and save the **Auth Token** somewhere safe, we will use it in the next step **DO NOT** share it, anyone who is having it can control your device.
- Now click on the "+" symbol to add widgets, then add a **Button**. Now you can see a button will be added to the dashboard.
- Click on the **Button** widget from the dashboard to edit the button settings.
- Now from the settings choose the pin which you want to control, we are choosing digital pin D5
- Change the ON state to 0 and OFF state to 1, by default ON will be 1 ie HIGH and OFF will be 0 i.e. LOW. We are doing this because Our relay will turn ON on LOW and turn OFF on HIGH.
- Change the mode to switch instead of push.

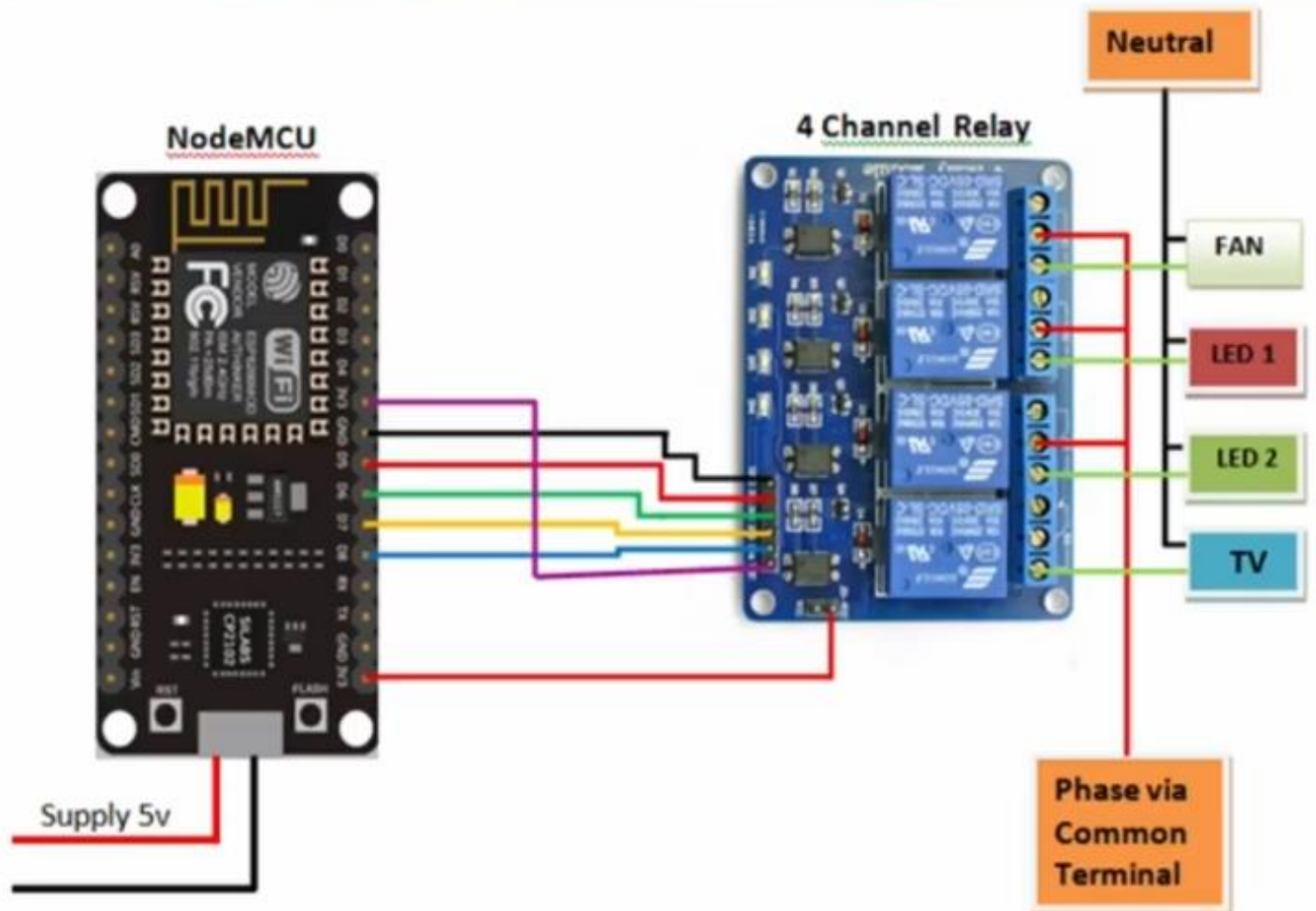
Step 2: NodeMCU

We are going to program NodeMCU with Arduino IDE. Open **node mcu blynk code** File in Arduino IDE.

- Then paste the **AuthToken** generated from Blynk app to the Blynk Arduino sketch.
- Type your Wi-Fi SSID and Password also to the Blynk Arduino sketch, where it is mentioned

Step 3: Circuit Diagram Connection

Circuit Diagram



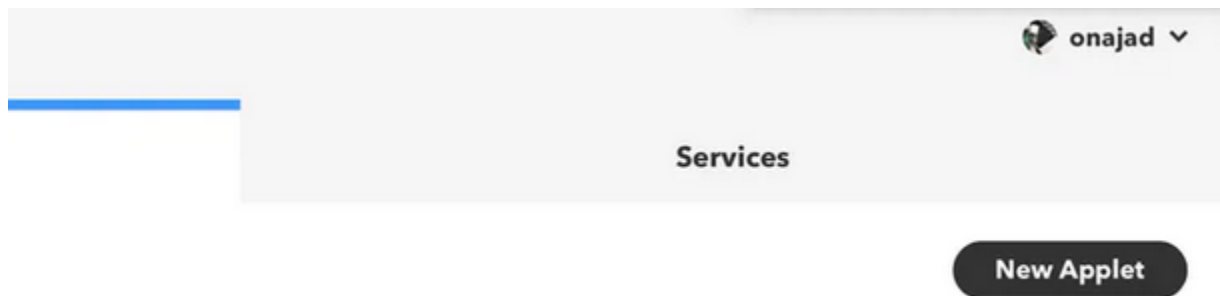
Step 4: IFTTT

Either you can use IFTTT website to IFTTT mobile app to create Applets. Here I'm using IFTTT website. Before start create IFTTT applet we need to create two HTTP request links, one for turning ON the relay and one for turning OFF the relay.

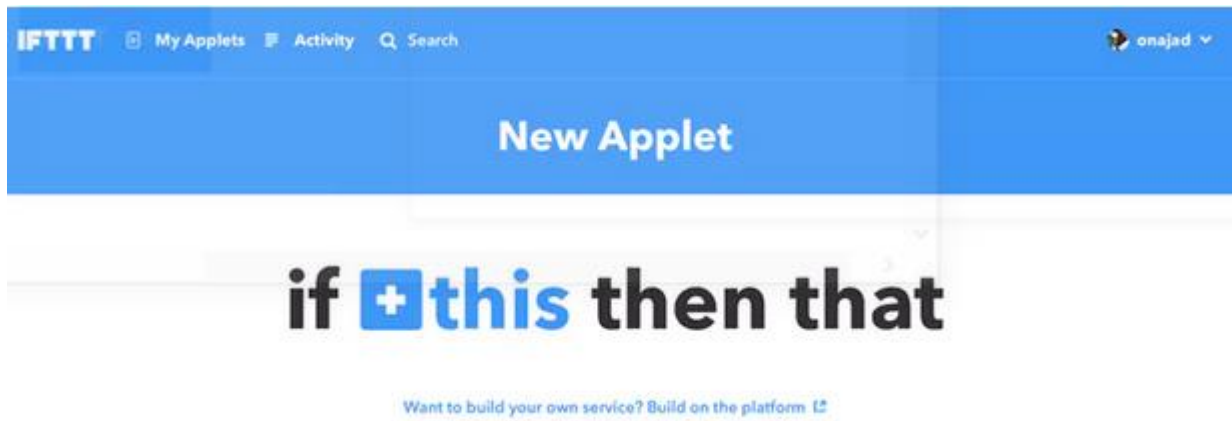
<http://188.166.206.43/blynkAuthToken/update/D14?value=0><http://188.166.206.43/blynkAuthToken/update/D14?value=1> Here,

- The 188.166.206.43 is the IP address of blynk server.
- You need to replace your blynk Auth Token, where it is mentioned in the request link.
- D14 is the pin which we have connected our relay, actually, we have connected relay to the D5 pin of NodeMCU but here in the HTTP request link we need to address the GPIO pin number, so the GPIO pin number for physical pin D5 is D14.
- When we call **value=0** it will make the pin D5 **LOW**, the relay is ON.
- When we call **value=1** it will make the pin D5 **HIGH**, the relay is OFF

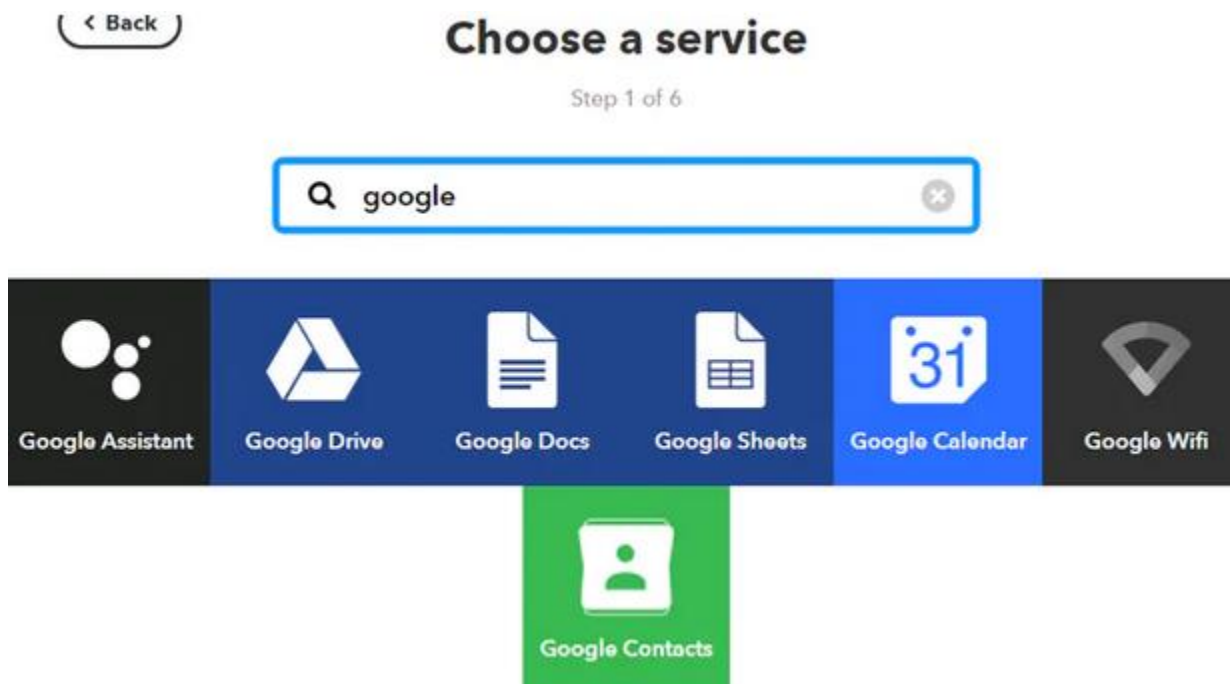
Creating Applet for Relay ON



- Go to www.ifttt.com then login using your account or signup if you don't have an account.
- Then under applets tab click on **new applets** button.



- Now, click on the **+this** button.
- Then choose a service page will open



- From "choose a service" page, search for Google Assistant and click on it.
- Then "choose trigger" page will open.

< Back



Choose trigger

Step 2 of 6

Say a simple phrase

This trigger fires when you say "Ok Google" to the Google Assistant followed by a phrase you choose. For example, say "Ok Google, I'm running late" to text a family member that you're on your way home.

Say a phrase with a number

This trigger fires when you say "Ok Google" to the Google Assistant followed by a phrase like "Set Nest thermostat to 68." **Use the # symbol to specify where you'll say the number ingredient

Say a phrase with a text ingredient

This trigger fires when you say "Ok Google" to the Google Assistant followed by a phrase like "Set Nest thermostat to 68." **Use the # symbol to specify where you'll say the number ingredient

Say a phrase with both a number and a text ingredient

- From "choose trigger" page click on Say Simple phrase.



Complete trigger fields

Step 2 of 6

Say a simple phrase

This trigger fires when you say "Ok Google" to the Google Assistant followed by a phrase you choose. For example, say "Ok Google, I'm running late" to text a family member that you're on your way home.

What do you want to say?

turn on light

What's another way to say it?
(optional)

And another way? (optional)

What do you want the Assistant to say in response?

Ok air

Language

English



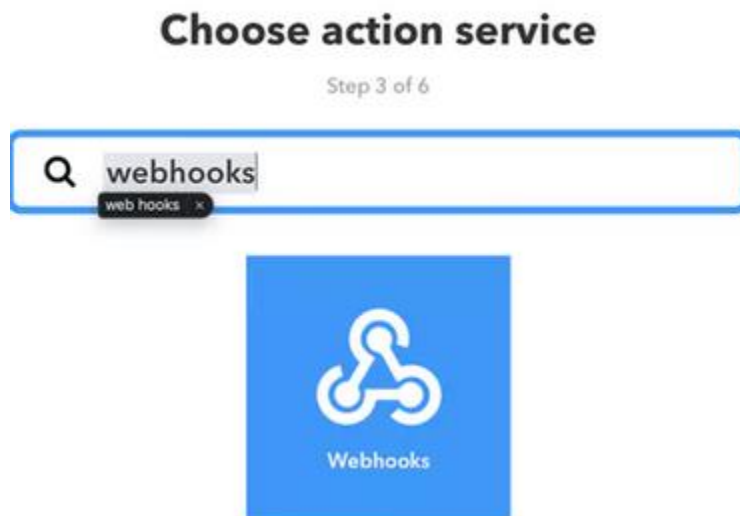
Create trigger

- **Now** Write a sentence you want to say to google assistant for light on
- and also add response sentence
- and click **create trigger** button.



if  then  that

- Now, choose **+that** button



- Then like before, search for **webhooks** and click on it.

< Back



Choose action

Step 4 of 6

Make a web request

This action will make a web request to a publicly accessible URL. NOTE: Requests may be rate limited.

- Now from "choose action" click on **Make a web request**.
- and fill the details as shown below

Make a web request

This action will make a web request to a publicly accessible URL. NOTE: Requests may be rate limited.

URL

Surround any text with "<<>>" to escape the content **Add ingredient**

Method

The method of the request e.g. GET, POST, DELETE

Content Type

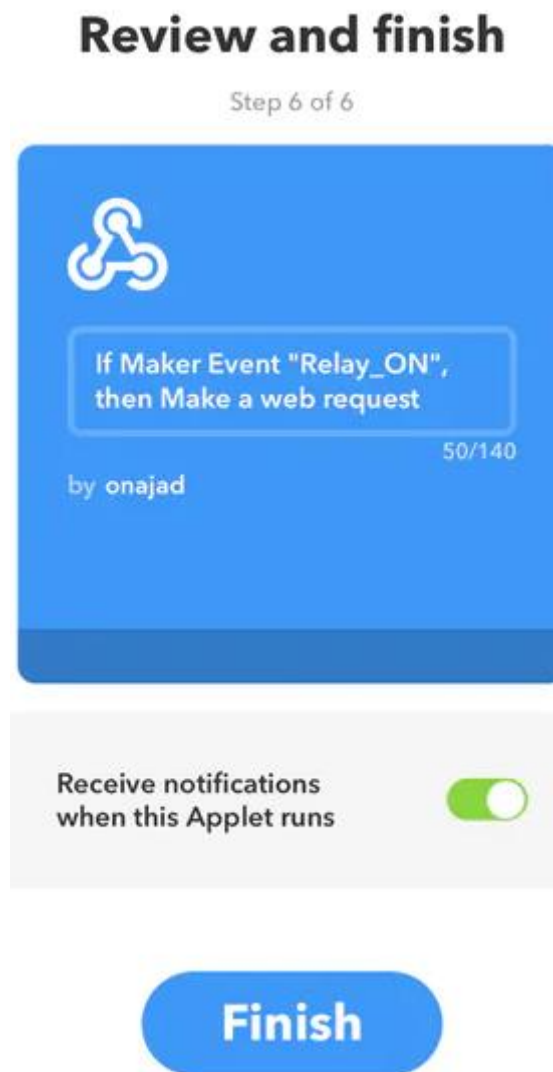
Optional

Body

Surround any text with "<<>>" to escape the content **Add ingredient**

Create action

- **URL:** In this field paste the HTTP request link which is created before, don't forget to edit the URL with your blynk Auth Token.
- **Method:** From the drop-down list choose **GET**.
- **Content/Type:** Choose text/plain from the dropdown list.
- **Body:** Leave it blank.
- Now, Click **Create action** Button

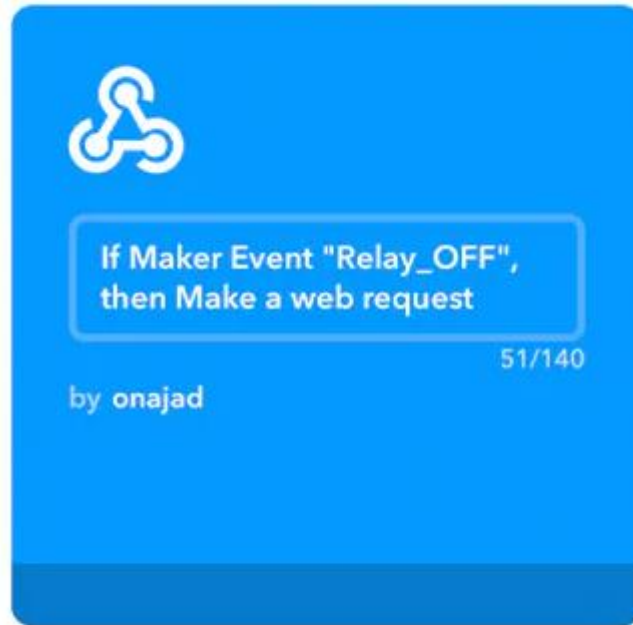


- Now we have successfully created an applet for turning the relay **ON**.
- like the same way, we need to create another applet for turning **OFF** the relay.

- Paste the HTTP request URL for turning **OFF** the relay, that is the **value=1**.
- and click on **create action** button

Review and finish

Step 6 of 6



Receive notifications
when this Applet runs



Now we have Successfully created Applets for both Relay ON and Relay OFF.

after that you login in google assistant using same Gmail account and simply say "Ok

google, Turn on light". NOW USING google assistant you can turn on/off your Home

Applications in this project we connect two lamps You can also connect More Devices based on

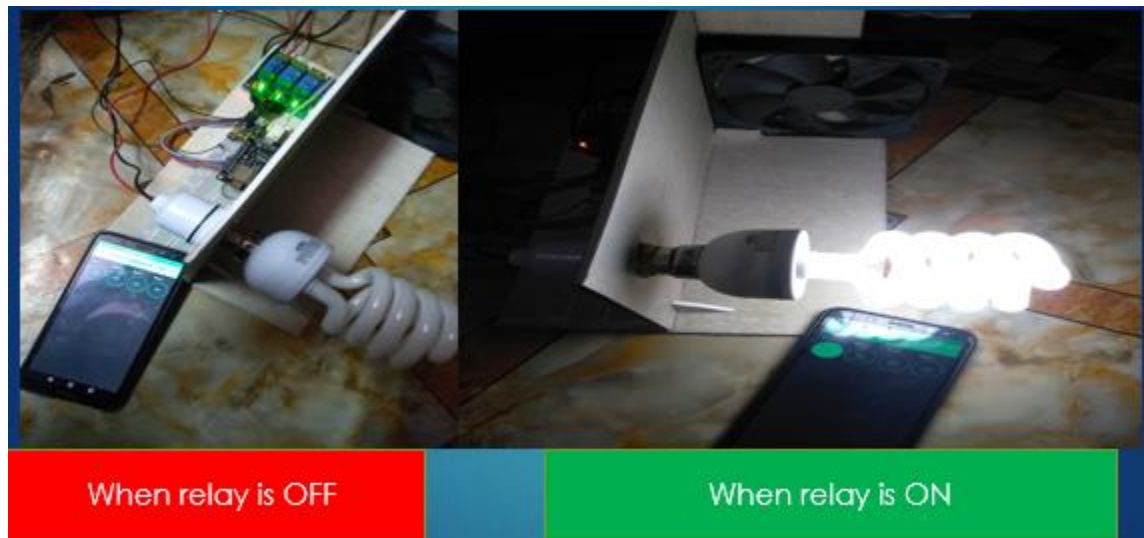
your requirement so you create more applets.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Results

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



Here once applet is created in IFTTT then the device to be turned on is given to that particular relay for which the applet is created. So, if we want to turn on the first device then it must be connected to relay 1. When we give command turn on the bulb then automatically the bulb will turn on.

4.2 Discussion

The Google Assistant is a software which allows its users to control all the apps in their device to be controlled directly through it. It allows the users to control and command most of the apps in their devices using voice commands. This provides more convenience to the people as they only have to command the google assistant thorough voice command. As for Google assistant, there was some problems with the system which is not affected by the signal of the voice to one of the appliances. The other appliances responded according to the voice received from the user as an input signal which lead to its response. Another issue faced was related to the interpretation of the voice received from the user (the pronunciation of words from users affect the way Google assistant will interpret.) Google assistant has the opportunity that we can used any language not only English but other languages too like Arabic, French, Spanish...instead of talking to the assistant, we can type what we want and get the same result. This is another way to act with Google assistant.

CHAPTER FIVE

CONCLUSION AND FUTURE WORK

5.1 Conclusion

The home automation using the Internet of Things has been proven to work by connecting simple appliances to it and the appliances were successfully controlled through internet. The designed system processes according to the requirement, for example switching on the light when we give the command. This will help the user to get an overview of various parameters in the home anytime anywhere. Low cost and flexible home Automation system using NodeMCU microcontroller is proposed and implemented. Overall, NodeMCU is easy to understand & its coding is easy. By implementing this type of system, we can ensure that the energy conservation can be done. By help of this system we can increase, the efficiency of the appliances. We can have the complete control over the home appliances from a long distance. This will increase the comfortability of human being and it will reduce the Human efforts.

5.2 Future Work

Using this home automation system as a reference, the system can be expanded to include various other options which could include home security feature like capturing the photo of a person moving around the house and storing it onto the cloud. Moreover, if we want to operate a particular device at some place then this will be helpful to do that. But this can make better by thinking out of the box and by innovating ideas.

There are other many factors to improve on to make GACHA (Google Assistant Controlled Home Automation) 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.

References

- [1]. IFTTT: <https://ifttt.com/discoverhttps://www.pocketlint.com/SmartHome/SmartHomenews/>
- [2]. Blynk: <https://www.blynk.cc/https://docs.blynk.cc/>
- [3]. NodeMCU:
<https://nodemcu.readthedocs.io/en/master> <https://iotbytes.wordpress.com/nodemcupinout/>
- [4]. Google Assistant: https://assistant.google.com/intl/en_in/
- [5]. <https://www.pocketlint.com/Apps/Appsnews/Googleappnews>
- [6]. (2002) The IEEE website. [Online]. Available: <http://www.ieee.org/>
- [7]. IoT: <https://internetofthingsagenda.techtarget.com/definition/IoT-device>.
- [8]. Singh, Sandeep Saurav, and Y. Bevish Jinila. "Sensor node failure detection using check point recovery algorithm." 2016 International Conference on Recent Trends in Information Technology (ICRTIT). IEEE, 2016.
- [9]. Seong Ro Lee and Rajeev Piyare "Smart Home-Control and Monitoring System Using Smart Phone" 1st International Conference on Convergence and its Application (ICCA), Volume: 24
- [10]. Devaraj, S. Allwin, R. Helen Vedanayagi Anita, and J. Jennifer Christa. "Comparative analysis of random based mobility models using TCP variant in MANETs."
- [11]. Kusuma S M, Assistant Professor, Department of telecommunication, MSRIT, Bangalore, India. "Home Automation Using Internet of Things" July 1999

Appendix

Code

node_mcu_blynk_code.ino

Arduino

```
#include <ESP8266WiFi.h>
```

```
#include <BlynkSimpleEsp8266.h>
```

```
// You should get Auth Token in the Blynk App.
```

```
// Go to the Project Settings (nut icon).
```

```
char auth[] = "YourAuthToken";
```

```
// Your WiFi credentials.
```

```
// Set password to "" for open networks.
```

```
char ssid[] = "YourNetworkName";
```

```
char pass[] = "YourPassword";
```

```
void setup()
```

```
{
```

```
  // Debug console
```

```
  Serial.begin(9600);
```

```
  Blynk.begin(auth, ssid, pass);
```

```
  // You can also specify server:
```

```
  //Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 80);
```

```
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);
```

```
}
```

```
void loop()
```

```
{
```

```
  Blynk.run();
```

```
  // You can inject your own code or combine it with other sketches.
```

```
  // Check other examples on how to communicate with Blynk. Remember
```

```
  // to avoid delay() function!
```

```
}
```