SEARCH ASSISTANT BASED HOME AUTOMATION [SABHA]

A PROJECT REPORT

Submitted by

MUKESH.G	**(412514106111)**
SAIDEEPAK.R	**(412514106158)**
VENKATASUBRAMANIAN.S	**(412514106211)**

In partial fulfillment for the award of the degree

Of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

SRI SAIRAM ENGINEERING COLLEGE

ANNA UNIVERSITY:: CHENNAI 600 025

APRIL 2018

ANNA UNIVERSITY::CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report titled **"SEARCH ASSISTANT BASED HOME AUTOMATION SYSTEM [SABHA]"** is the bonafide work of "**MUKESH.G (412514106111), SAIDEEPAK.R (412514106158), VENKATASUBRAMANIAN.S (412514106211)"** who carried out the project work under my supervision.

SIGNATURE

Ms. A.R. RAJINI

PROFESSOR

HEAD OF THE DEPARTMENT

Department of Electronics and Communication Engineering

Sri Sai Ram Engineering College

Chennai – 600 044.

SIGNATURE

Ms. S. USHA

ASSISTANT PROFESSOR

SUPERVISOR

Department of Electronics Communication Engineering

Sri Sai Ram Engineering College

Chennai – 600 044.

Submitted for VIVA-VOCE Examination held on …….

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We thank our Founder Chairman **Thiru. MJF. Ln. LEO MUTHU** for his great endeavors in establishing this institution and standing as a figure of guidance.

We also thank our C.E.O **Mr. J. SAI PRAKASH LEO MUTHU** and Principal **Dr. C.V. JAYAKUMAR** for their kind cooperation and inspiration.

We thank **Prof. A.R. RAJINI**, Head of the Department, Electronics and Communication Engineering for giving us freedom to carry out the project work in the chosen domain.

We thank our Project Coordinator **Ms.V.SASIKALA** and **Ms.M.SHABANA PARVEEN,** Assistant Professor, for thier constant support right from the commencement of the project work and also for providing us necessary details with regard to presentation and documentation.

We are ever grateful to our Project Guide **Ms.S.USHA,** Assistant Professor, who was a buttress to carry out our project and for his/her valuable suggestions at every stage of the project. We sincerely thank him/her for the support rendered from the day we commenced our work.

Our gratitude extends to the **staff of ECE Department** whose words of encouragement kept our spirits high throughout the course of the project.

We thank the entire people who contributed directly and indirectly for the completion of this project.

ABSTRACT

Home automation has been a booming industry providing different solutions for the consumers, but a low cost search assistant based home automation system is yet to be introduced. The main objective is to build a low cost search assistant based home automation system. The system mainly consists of two modules:- The Master Module (which is an raspberry pi based search assistant module) and Control Module (which is like a switch board cover used to control the appliances). The Master Module takes the speech signal as the input and plays a speech signal as an output thereby performing actions of a search assistant (like surfing the web, setting alarms, etc.,), it also gives an output to the control module to control home appliances (commands like: “switch on the light”). The Control Module is in form of a switch board cover with some projection accompanying electronic components and servo motors that are used to control the switching action of the switches. In addition to these there is an OLED Display for the consumer to know about the status of the switches and it also shows useful information like time, date, weather and even ring an alarm thereby without using any kind of PLC or any other high cost controllers to control the action of the switches.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ACRONYM	EXPANSION
oled	Organic Light Emitting Diode
Plc	Programmable Logical Controller
hvac	Heating, Ventilation and AirConditionng
wi fi	Wireless Fidelity
IOT	Internet Of Things
SMS	Short Message Service
NLP	Natural Language Processing
GSM	Global Systeam for Mobile
NFC	Near Field Communication
6LoWPAN	LowPower Wireless Personal Area Network
GPIO	General Purpose Input/Output
ARM	Advanced RISC Machine
LAN	Local Area Network
RAM	Random Access Memory
CPU	Central Processing Unit
SD	Secure Digital
PWM	Pulse Width Modulation
ADC	Analog to Digital Convertor
Hz	Hertz

CHAPTER 1

INTRODUCTION

Home automation has been a booming industry providing different solutions for the consumers, World's top Techgaints like Google, amazon and even Facebook are being heavily investing in Home automation, knowing that it is the next basic need of the future. They have introduced Home automation solutions like Alexa by Amazon, Google Home by Google Corp and Jarvis by Facebook. But, these are constrained only to few nations due to lack of technological developments in other nations and due to its high cost of implementation. There by a low cost search assistant based home automation system is yet to be introduced.

1.1 OBJECTIVE

The main objective, to build a low cost search assistant based home automation system. The system mainly consists of two modules: *The Master Module*_(which is an raspberry pi based search assistant module) and *Control Module* (which is like a switch board cover used to control the appliances). In addition to these there is an OLED Display for the consumer to know other useful information like time, date, weather and even ring an alarm. Thereby without using any kind of PLC or any other high cost controllers to control the action of the switches to provide a low cost search assistant based home automation solution for the consumer.

1.2 MOTIVATION

In order to improve the quality of life for common people, in these work we focused on developing new technologies to help these persons to access the home appliances from any outdoor environment. With an exponential growth of

relatively old concept of home automation, the installation of the system and managing it are still very complex to deal with. In this project we are creating a simple way to manage multiple end points by the average user using Voice recognition. The system intends to help them in providing the information using internet. This solution can be used in various sectors for making daily tasks comfortable and easy.

1.3 PROPOSED SYSTEM

The Master Module takes the speech signal as the input and gives a speech signal as an output thereby performing actions of a search assistant (like surfing the web, setting alarms, etc.,) , it also gives an output to the control module to control home appliances(commands like: ”switch on the light”). The Control Module is in form of a switch board cover with some projection accompanying some electronic components and servo motors that are used to control the switching action of the switches.

1.3.1 MERITS

- Efficiency
- Speed
- Availability
- Low power consumption
- Peace of mind
- Control
- Convenience
- Safety
- Savings
- Ease of installation

1.4 IMPORTANCE OF PROJECT

- To enhance instant control on gadgets
- Economic project can be affordable for common people
- Reduce the manpower
- Saving lot of time to operate home appliances

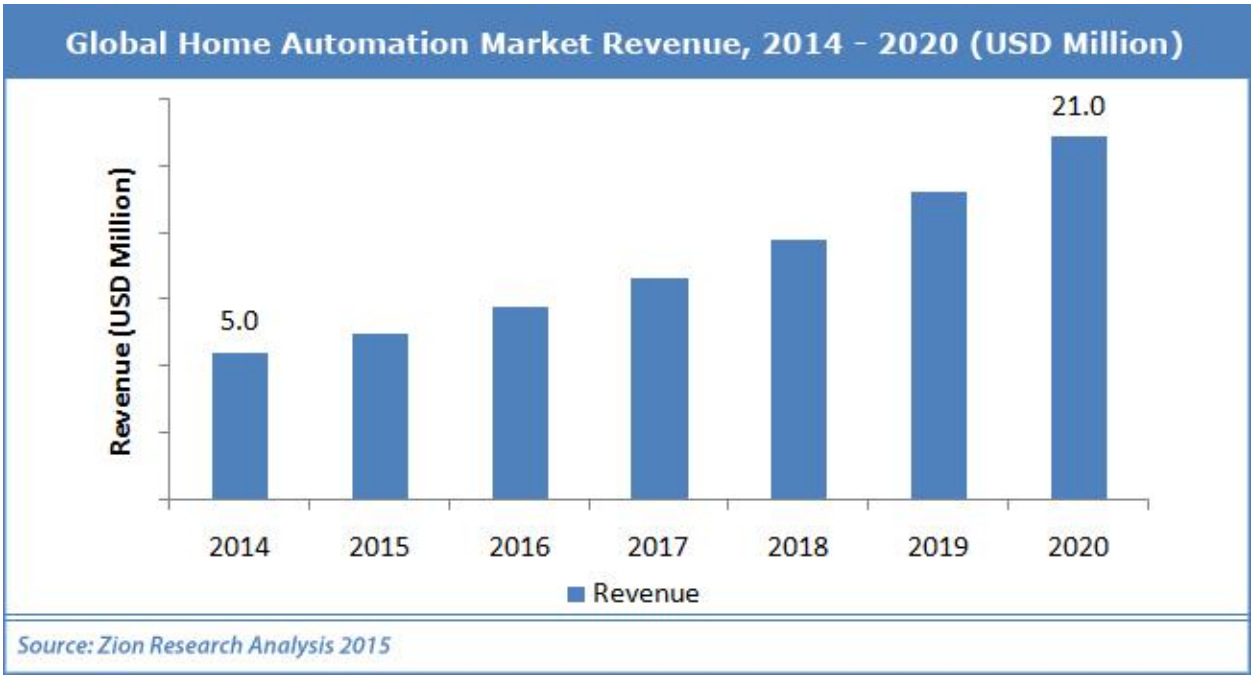

Fig: 1.1 increasing demand for home automation

1.5 APPLICATION

- Appliance control
- Automation for elderly and disabled
- Increased household security
- Occupancy aware control
- Leak, smoke and water detection

CHAPTER 2

LITERATURE REVIEW

2.1 1NTRODUCTION

Home automation or domotics is building automation for a home, called a smart home or smart house. It involves the control and automation of lighting, heating (such as smart thermostats), ventilation, air conditioning (HVAC), and security (such as smart locks), as well as home appliances such as washer/dryers, ovens or refrigerators/freezers. Wi-Fi is often used for remote monitoring and control. Home devices, when remotely monitored and controlled via the Internet, are an important constituent of the Internet of Things. Modern systems generally consist of switches and sensors connected to a central hub sometimes called a "gateway" from which the system is controlled with a user interface that is interacted either with a wall-mounted terminal, mobile phone software, tablet computer or a web interface, often but not always via Internet cloud services. While there are many competing vendors, there are very few worldwide accepted industry standards and the smart home space is heavily fragmented. Manufacturers often prevent independent implementations by withholding documentation and by litigation. The home automation market was worth US$5.77 billion in 2013, predicted to reach a market value of US$12.81 billion by the year 2020.

2.2 HISTORY

Early home automation began with labor-saving machines. Self-contained electric or gas powered home appliances became viable in the 1900s with the introduction of electric power distribution and led to the introduction of washing machines (1904), waterheaters (1889), refrigerators, sewing

machines, dishwashers, and clothes dryers. In 1975, the first general purpose home automation network technology, X10, was developed. It is a communication protocol for electronic devices. It primarily uses electric power transmission wiring for signalling and control, where the signals involve brief radio frequency bursts of digital data, and remains the most widely available. By 1978, X10 products included a 16 channel command console, a lamp module, and an appliance module. Soon after came the wall switch module and the first X10 timer. By 2012, in the United States, according to ABI Research, 1.5 million home automation systems were installed. According to Li et al. (2016) there are three generations of home automation:

1. First generation: wireless technology with proxy server, e.g. Zigbee automation;
2. Second generation: artificial intelligence controls electrical devices, e.g. Amazon Echo;
3. Third generation: robot buddy who interacts with humans, e.g. Robot Rovio, Roomba

2.3 LITERATURE SURVEY

Advanced IOT based combined remote health monitoring, home automation and alarm system - Jayeeta Saha - Arnab Kumar – 17600370 - IEEE

Abstract:

Nowadays, automation and Internet of Things are changing the world. The day-by-day development of the Internet of Things causes a revolution in modern technology, which makes our life easier and automated. Due to a busy schedule and irregular lifestyle, health hazard is not an age-dependent factor in the recent

era. Under these circumstances, Internet of Things has provided a much easier solution for remote real-time health monitoring of patients from the hospital as well as home. Sensors acquire the data of various parameters regarding patients' health, and the Internet of Things stores that data and displays through the website, which provides access for remote monitoring. Use of Sensor reduces the human error, and the size of the system reduces the occupied space of the room. The unique part of this proposed solution is the alarm generation to provide the prescribed medicine to the patient in time. The other beneficial area of the system is the scheme of sending the notification through email and SMS alert if any of the health parameters crosses the threshold value. Notification scheme will keep the respective authority conscious of the situation. Another significant area of the proposed solution is to create the optimum surroundings as per the requirement of patient's health condition. In this paper, we have discussed the monitoring of heart rate, blood pressure, respiration rate, body temperature, body movement and saline levels.

Voice controlled home automation system using Natural Language Processing (NLP) and Internet of Things (IoT) - Paul Jasmin Rani - Jason Bakthakumar - B. Praveen Kumaar – 17521288 - IEEE

Abstract:

The primary objective of our project is to construct a fully functional voice based Home automation system that uses Internet of Things, Artificial Intelligence and Natural Language Processing (NLP) to provide a cost-effective, efficient way to work together with home appliances. There are many smart home solutions in the market that aim to automate the basic operations of these home appliances using various technologies such as GSM (Global System for Mobile), NFC (Near-Field

Communication) etc. However, most of these systems focus on mimicking the basic operation of the electrical switch. Our project aims at providing a fully automated voice based solution that our users can rely on, to perform more than just switching on/off the appliances. The user sends a command through speech to the mobile device, which interprets the message and sends the appropriate command to the specific appliance. We plan on implementing four basic home appliances as a "Proof-of-Concept" for this project which includes Fan, Light, Coffee Machine and Door Alarms. The voice command given by the user is interpreted by the mobile device using Natural Language processing. The mobile device acts as a central console; it determines what operation must be completed by which appliance to fulfill the user's request. The central console might likewise be either a desktop application, web application or a smart phone application as nearly all of the data transferred can be processed by the cloud. However, for the convenience of the user and increased mobile capabilities we will be using a smart phone in this project. The appliances are associated with the mobile device through an Arduino Board that establishes the concept of Internet of Things. The Arduino Boards are interfaced with the appliances and programmed in a manner that they respond to mobile inputs.

Home automation using IoT and a chatbot using natural language processing – Cyril Joe Baby - Faizan Ayyub Khan - J. N. Swathi - 17471058 – IEEE

Abstract:

Home automation - controlling the fans, lights and other electrical appliances in a house using Internet of things is widely preferred in recent days. In this paper, we propose a web application using which the fans, lights and other electrical

appliances can be controlled over the Internet. The important features of the web application is that firstly, we have a chatbot algorithm such that the user can text information to control the functioning of the electrical appliances at home. The messages sent using the chatbot is processed using Natural Language processing techniques. Secondly, any device connected to the local area network of the house can control the devices and other appliances in the house. Thirdly, the web application used to enable home automation also has a security feature that only enables certain users to access the application. And finally, it also has a functionality of sending an email alert when intruder is detected using motion sensors.

On cohabitating networking technologies with common wireless access for home automation system purposes – Jordi Mongay Batalla - George Mastorakis - Constandinos X. Mavromoustakis - 16448415 - IEEE Communications Society

Abstract:

An increasing number of home automation systems using wireless devices compete for the radio access in the same space and time. Lately, a stressing trend consists of aggregating home automation systems to save energy consumption, while at the same time avoiding wireless interference. This article proposes virtualization, open software deployment, and separation of radio and higher layers as the response to the increasing expandability of home automation systems combined with the increasing number of technologies for connecting wireless devices. A system has been developed, containing three different technologies: ZigBee, Idsecom, and 6LoWPAN simultaneously working over a virtualization platform with access to a common antenna. The evaluation tests performed on the

system validate the solution and separately show the performance capacity of virtualization platform, software (ZigBee, Idsecom and 6LoWPAN) nodes, and 802.15.4 wireless antennas.

Eyrie smart home automation using Internet of Things – 17502684 - Ayush Panwar - Anandita Singh - Renu Kumawat – IEEE

Abstract:

This paper describes the use of various open source hardware such as Arduino, Raspberry Pi, etc. to build smart and secure homes. The hardware is open source and hence cost efficient. This home automation system allows the end user to monitor his home or office with a smartphone, tablet, or any computer. This paper also explains the use of the security system for fire hazards that may occur due to a gas leakage and can be detected using a smoke sensor. It uses a low power NRF24L trans-receiver at each node around the house to create a mesh network that connects to a Linux based central hub. Users can monitor the house from anywhere and get periodic alerts. In the proposed work, the house can also be controlled using voice commands such as Google Voice, Apple HomeKit and Alexa.

An Internet based wireless home automation system for multifunctional devices - A.Z. Alkar - U. Buhur – 8755275 - IEEE Consumer Electronics Society

Abstract:

The aim of home automation is to control home devices from a central control point. In this paper, we present the design and implementation of a low cost but yet

flexible and secure Internet based home automation system. The communication between the devices is wireless. The protocol between the units in the design is enhanced to be suitable for most of the appliances. The system is designed to be low cost and flexible with the increasing variety of devices to be controlled.

Design of a phoneme based voice controlled home automation system - G B Karan - Kiran Pai – 17616391 – IEEE

Abstract:

Speech recognition has found a variety of applications in consumer electronics, such as smart phones, laptops, speech to text converters, live subtitling on television, dictation tool and many more. In this paper, design of a Voice Controlled Home Automation System using phoneme based speech recognition is proposed. The proposed system uses Mel Frequency Cepstral Coefficients for feature extraction and a two stage Support Vector Machine for classification. In order to assess the performance of system designed, voice samples were recorded in the laboratory to train the system. 86.00-100.0% recognition accuracy is obtained on using the proposed system with a recognition time of 0.9063s. The proposed work can be easily extended for various other voice controlled pattern recognition applications.

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

It mainly consists of two modules one is the Master module another one is the control module. The master module is a static one which takes the speech signal as the input processes it identifies whether it is a triggering command (Commands to control the electronic equipment's) or search commands. Based on the commands the decision is taken by the master module to act as Search assistant or a controller. If the given in command is a search command than the master module performs as search assistant surfing the web for the required results and provides us speech signal as the output with the help of the speaker. If the input command is given is a trigger command than the control is shifted to the control module, were the appropriate switch is controlled. This is done with the help of updating the control values in the cloud database (Google FireBase). The control module is in the form of the switch cover which consists of gear setup fitted with the servo motors. Which are interconnected with the help of the Wifi module. If the user wants to manually control the switches than a manual interrupt option is provided in the control module itself by pressing the button on it. The status of the switches as well as useful information's such as date, time and weather are displayed in the OLED screen present in the control module.

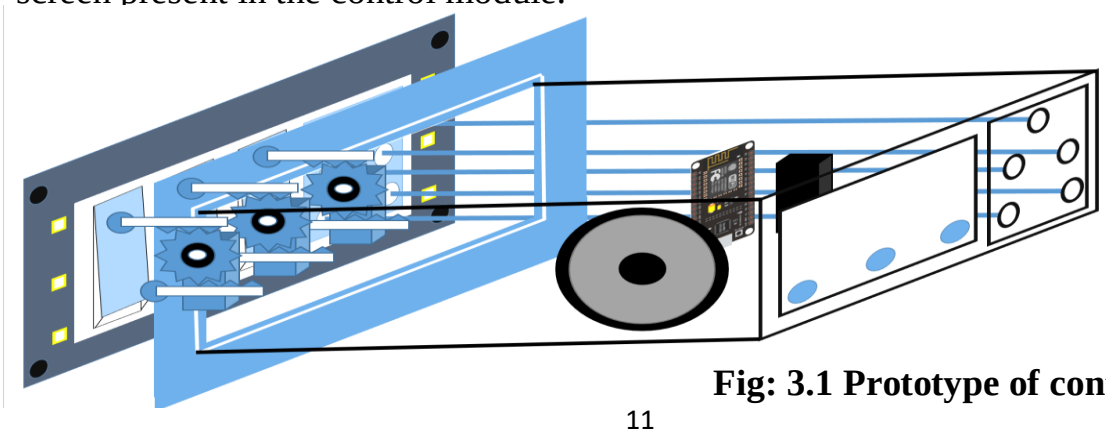

Fig: 3.1 Prototype of control module

. 3.2 BLOCK DIAGRAM

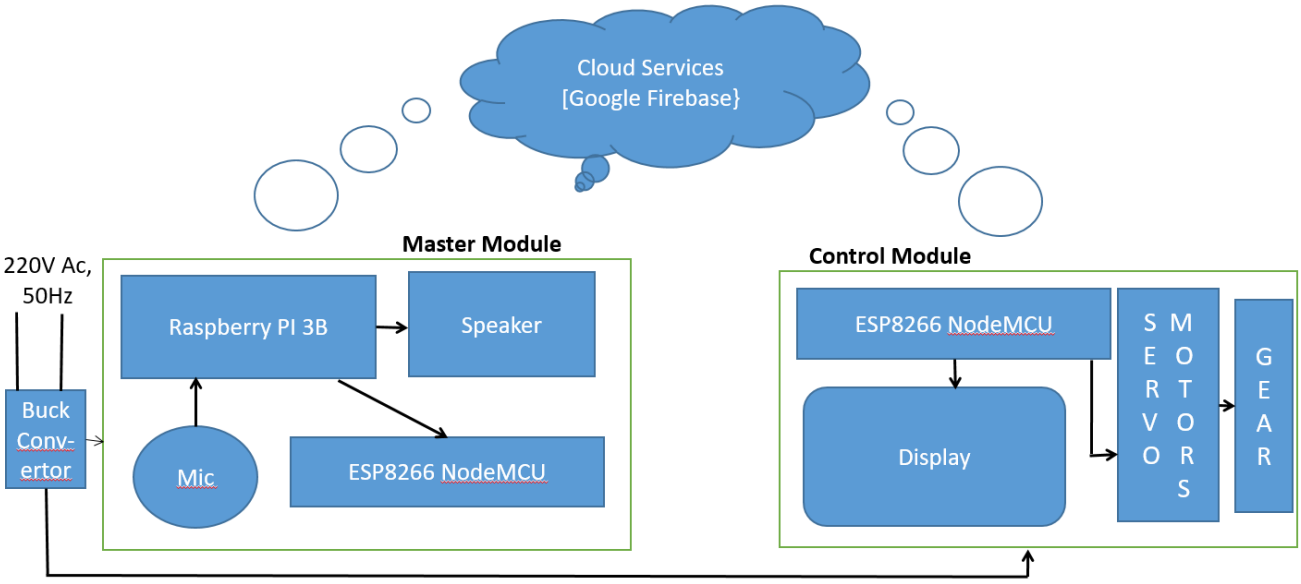

Fig: 3.2 Block Diagram of the proposed system

3.3 WORK FLOW

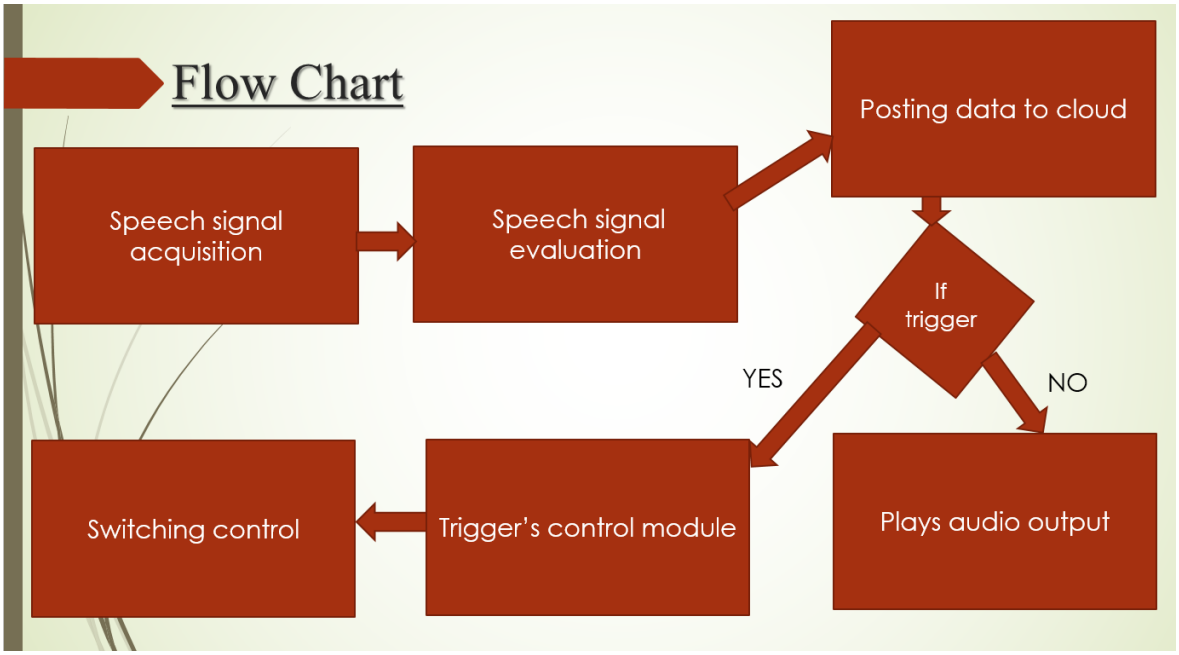

Fig: 3.3 Flow Chart of the proposed system

3.4 WORKING PRINCIPLE

The lack of technological development in the Electronic Equipment used day to day is the main problem in the above existing system. To overcome this we are going for a more versatile solution by controlling the switch itself to control the electronic equipments using a worm gear mechanism

3.5 DESCRIPTION OF BLOCK DIAGRAM

3.5.1 RASPBERRY PI 3B

3.5.1.1 PRODUCT DESCRIPTION

The Raspberry Pi 3 Model B is the third generation Raspberry Pi. This powerful credit-card sized single board computer can be used for many applications and supersedes the original Raspberry Pi Model B+ and Raspberry Pi 2 Model B. Whilst maintaining the popular board format the Raspberry Pi 3 Model B brings you a more powerful processer, 10x faster than the first generation Raspberry Pi. Additionally it adds wireless LAN & Bluetooth connectivity making it the ideal solution for powerful connected designs.

3.5.1.2 TECHNICAL SPECIFICATION

- Broadcom BCM2387 chipset
- 1.2GHz Quad-Core ARM Cortex-A53
- 802.11 bgn Wireless LAN and Bluetooth 4.1 (Bluetooth Classic and LE)
- 1GB RAM
- 64 Bit CPU

- 4 x USB ports
- 4 pole Stereo output and Composite video port
- Full size HDMI
- 10/100 BaseT Ethernet socketbr
- CSI camera port for connecting the Raspberry Pi camera
- DSI display port for connecting the Raspberry Pi touch screen display
- Micro SD port for loading your operating system and storing data
- Micro USB power source

3.5.1.3 FEATURES

- Now **10x Faster** - Broadcom BCM2387 ARM Cortex-A53 Quad Core Processor powered Single Board Computer running at 1.2GHz!
- 1GB RAM so you can now run bigger and more powerful applications
- Fully HAT compatible
- 40pin extended GPIO to enhance your “real world” projects.
- Connect a Raspberry Pi camera and touch screen display (each sold separately)
- Stream and watch Hi-definition video output at 1080
- Micro SD slot for storing information and loading your operating systems.
- 10/100 BaseT Ethernet socket to quickly connect the Raspberry Pi to the Internet

3.5.1.4 PIN CONFIGURATION

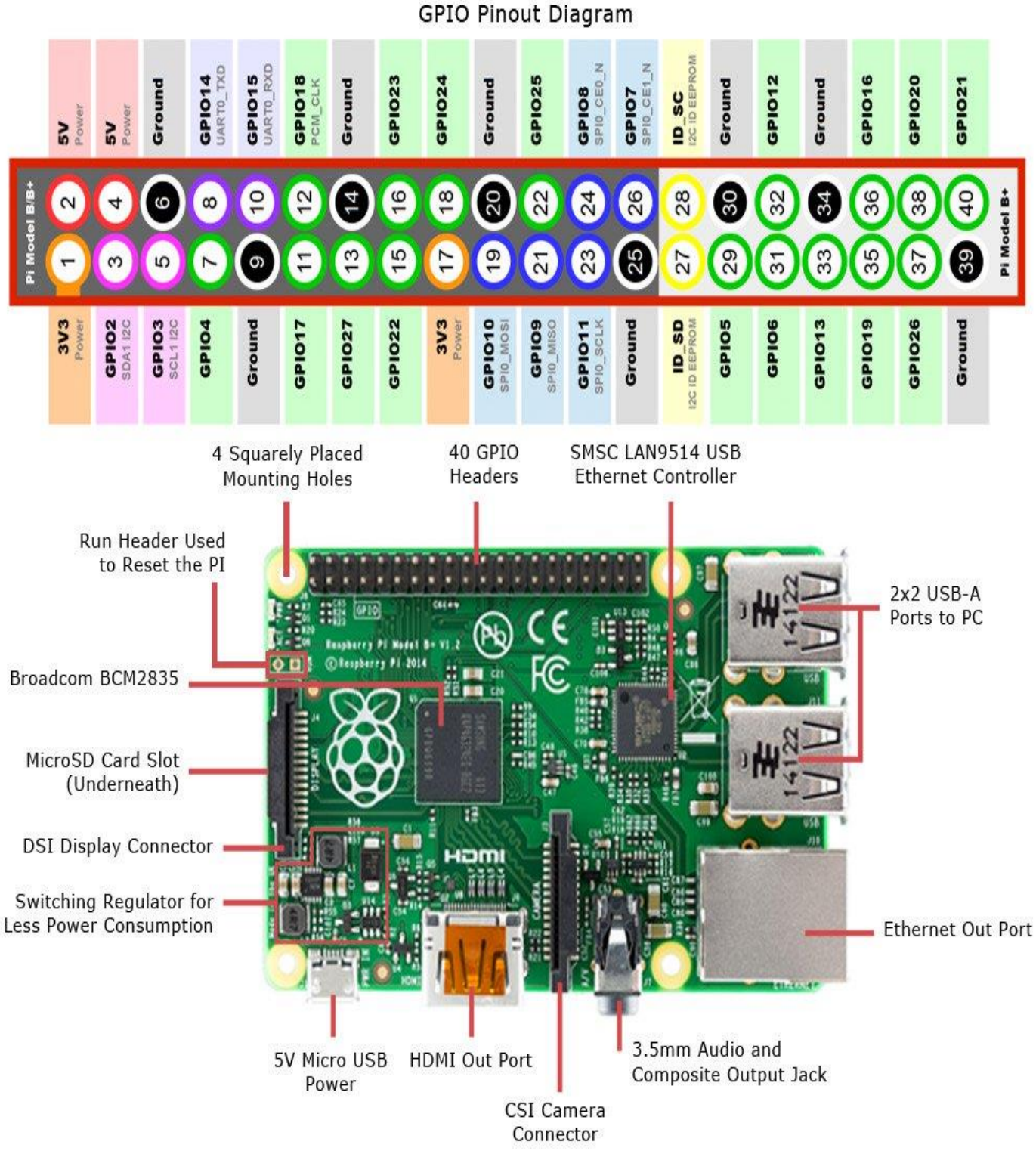

Fig: 3.4 pin diagram of Raspberry Pi Model 3B

3.5.2 NODEMCU

3.5.2.1 PRODUCT DESCRIPTION

The ESP8266 is the name of a micro controller designed by Espressif Systems. The ESP8266 itself is a self-contained WiFi networking solution offering as a bridge from existing micro controller to WiFi and is also capable of running self-contained applications. This module comes with a built in USB connector and a rich assortment of pin-outs. With a micro USB cable, you can connect NodeMCU devkit to your laptop and flash it without any trouble, just like Arduino. It is also immediately breadboard friendly.

3.5.2.2 SPECIFICATION

- Voltage:3.3V. • Wi-Fi Direct (P2P), soft-AP.
- Current consumption: 10uA~170mA.
- Flash memory attachable: 16MB max (512K normal).
- Integrated TCP/IP protocol stack.
- Processor: Tensilica L106 32-bit.
- Processor speed: 80~160MHz.
- RAM: 32K + 80K
- GPIOs: 17 (multiplexed with other functions).
- Analog to Digital: 1 input with 1024 step resolution.
- +19.5dBm output power in 802.11b mode
- 802.11 support: b/g/n.
- Maximum concurrent TCP connections: 5.

3.5.2.3 HARDWARE SPECIFICATION

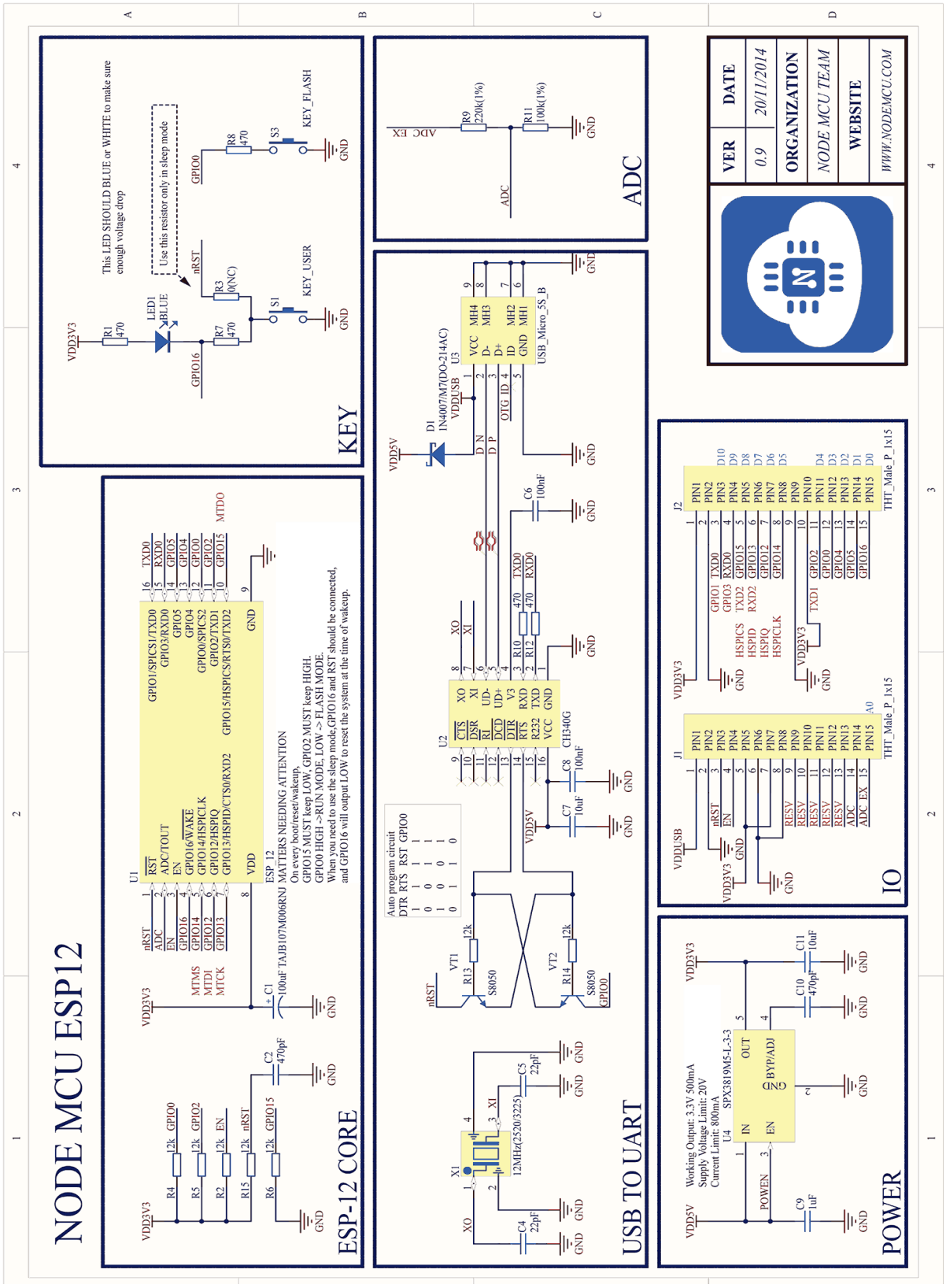

Fig: 3.5 Hardware Architecture of NodeMCU

3.5.2.4 PIN CONFIGURATION

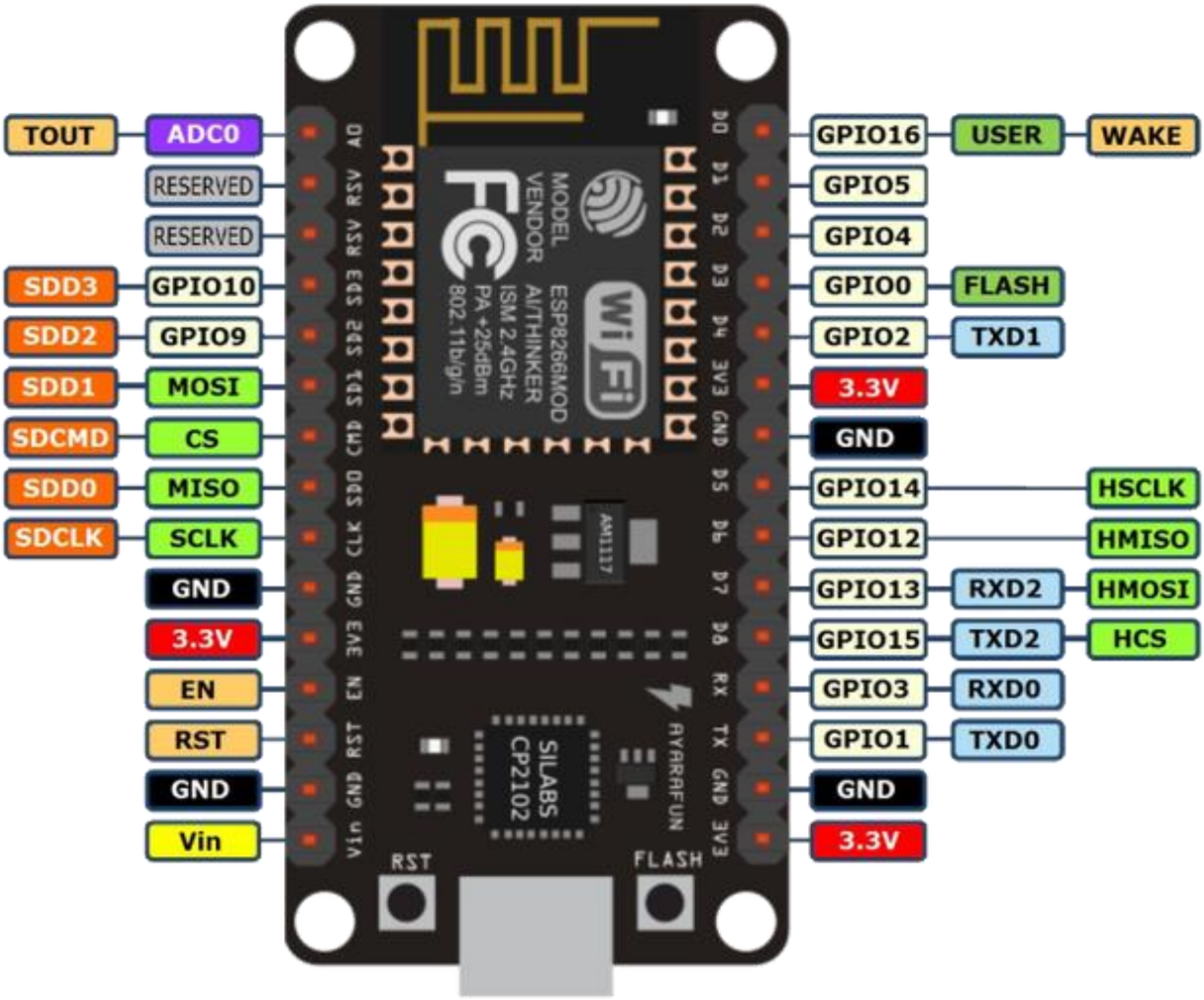

Fig: 3.6 pin diagram of NodeMCU

GPIO (General Purpose Input Output) Pins:

NodeMCU has general purpose input output pins on its board as shown in above pinout diagram. We can make it digital high/low and control things like LED or switch on it. Also, we can generate PWM signal on these GPIO pins.

ADC (Analog to Digital Converter) channel (A0):

NodeMCU has one ADC channel/pin on its board.

SPI (Serial Peripheral Interface) Pins:

NodeMCU based ESP8266 has Hardware SPI (HSPI) with four pins available for SPI communication. It also has SPI pins for Quad-SPI communication. With this SPI interface, we can connect any SPI enabled device with NodeMCU and make communication possible with it.

I2C (Inter-Integrated Circuit) Pins:

NodeMCU has I2C functionality support on ESP8266 GPIO pins. Due to internal functionality on ESP-12E we cannot use all its GPIOs for I2C functionality. So, do tests before using any GPIO for I2C applications

UART (Universal Asynchronous Receiver Transmitter) Pins:

NodeMCU based ESP8266 has two UART interfaces, UART0 and UART1. Since UART0 (RXD0 & TXD0) is used to upload firmware/codes to board, we can't use them in applications while uploading firmware/codes.

3.5.3 GOOGLE'S VOICE HAT AIY KIT

3.5.3.1 PRODUCT DESCRIPTION

The Voice HAT connects the Raspberry Pi to the Google Assistant and is part of Google's AIY Voice Kit. It was created with the Raspberry Pi 3 Model B in mind but may be used with any Raspberry Pi featuring a 40-way header, such as the Raspberry Pi Zero.

The HAT includes on-board hardware to facilitate audio capture and playback, connectors for the dual mic daughter board and speaker, GPIO breakouts to connect low-voltage components like micro-servos and sensors, and an optional barrel connector for dedicated power supply.

3.5.3.2 PIN CONFIGURATION

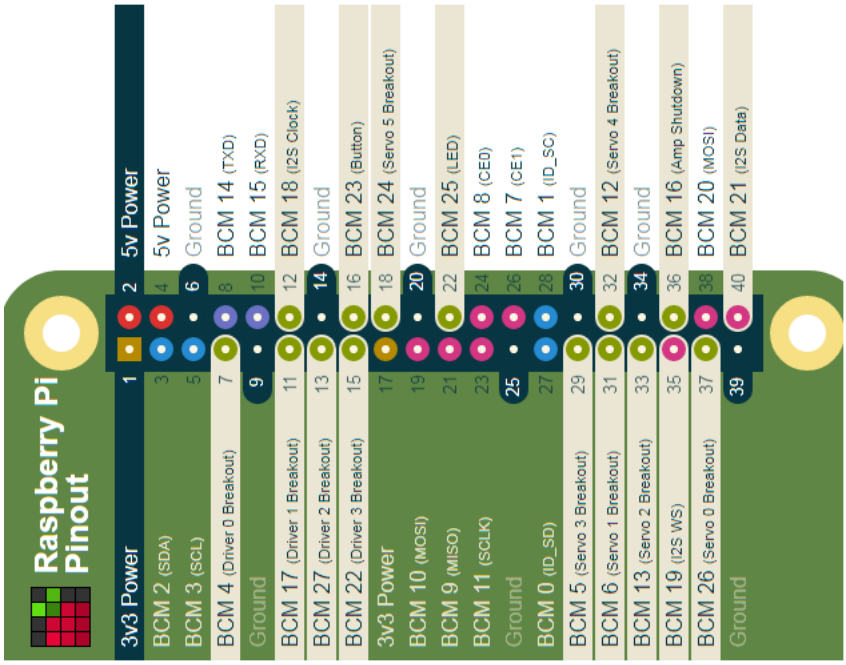

Fig: 3.7 pin connection of Voice HAT to Raspberry Pi

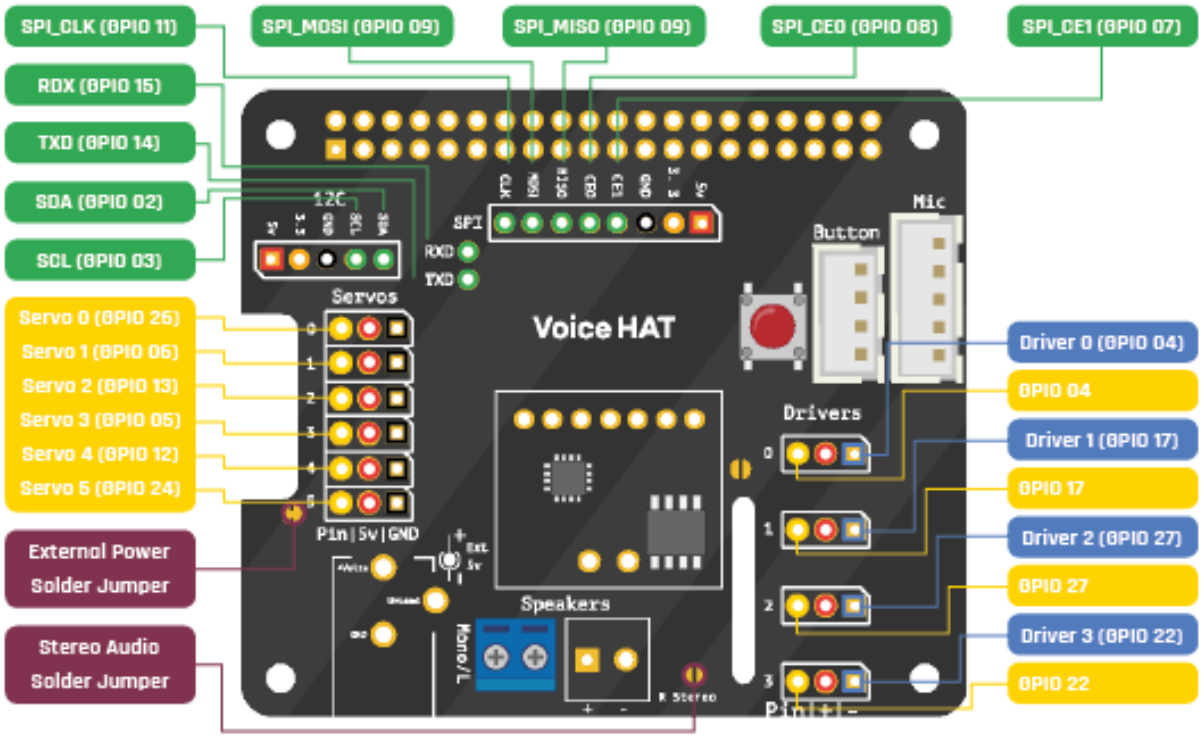

Fig: 3.8 Pin Description of Voice HAT

3.5.3.3 CONNECTIONS

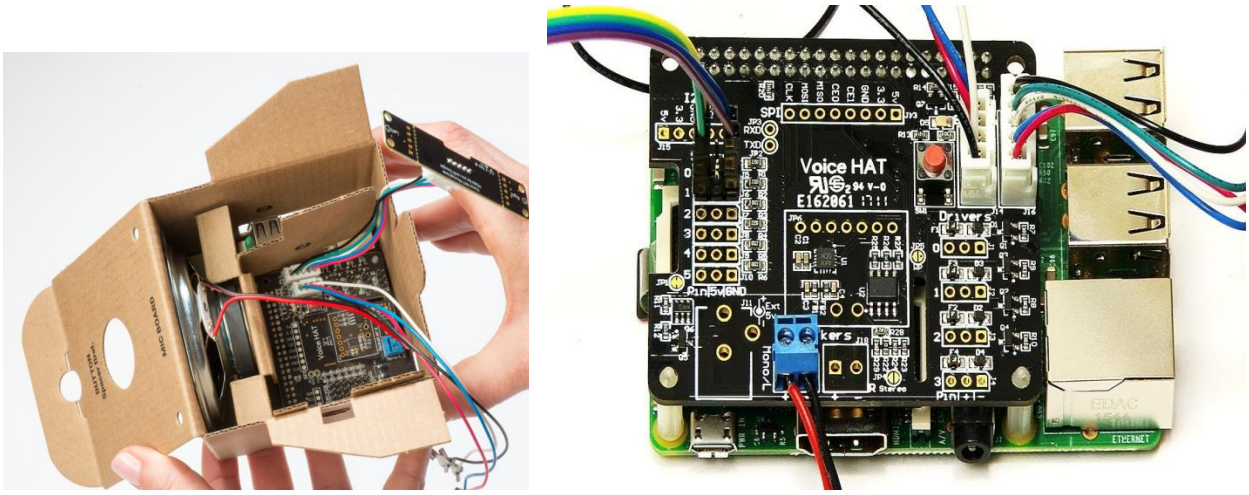

Fig: 3.9 Connections between Voice HAT and Raspberry Pi as well as other components inside the Kit

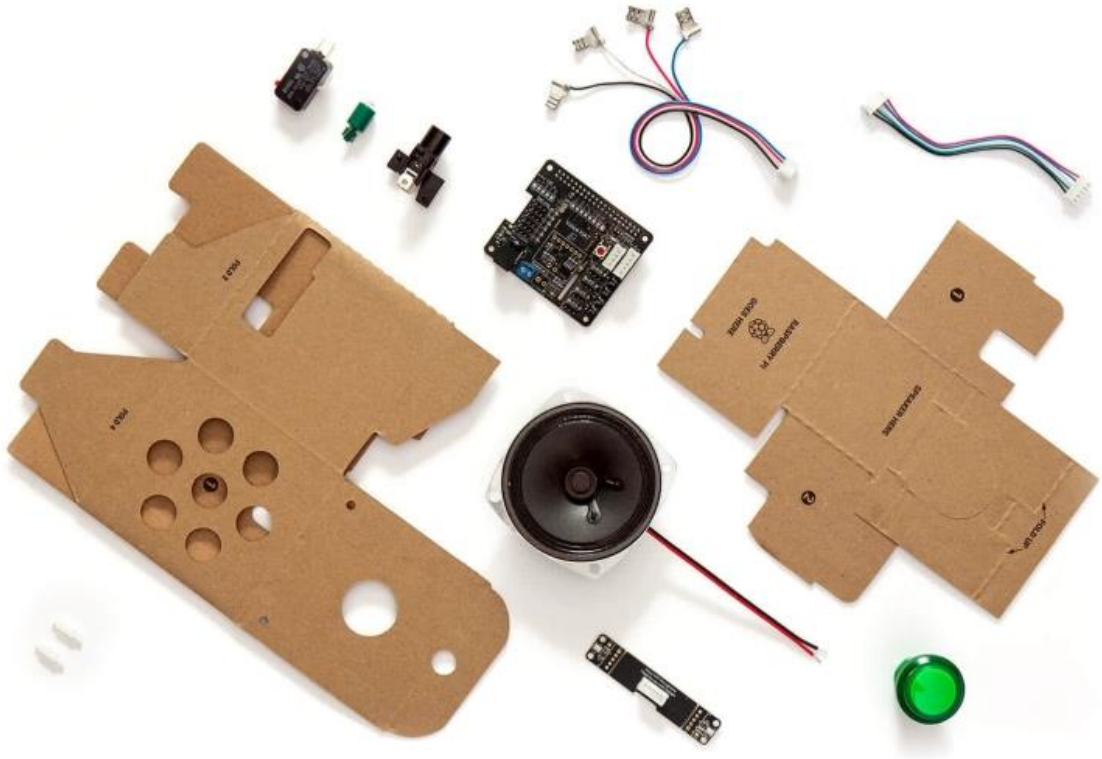

Fig: 3.10 Components inside the Kit

3.5.4 SERVO MOTOR

3.5.4.1 PRODUCT DESCRIPTION

VS-11 is analog servo with 19.0 kgf.cm torque. This servo motor is ideal for making your own hexapod, walking insects, heavy duty sensor,camera pod..They are very useful in robotics applications because of there small size and low cost. Servo motor has built in motor, gearbox, position feedback mechanism and motor controller. The servo motor can be controlled to move any position just by using simple pulse controlling. This motor has three wire interfaces for control and power supply.On servo motor black/brown wire is battery ground (negative), the red wire is battery power (positive) and the white/yellow/orange wire is receiver signal.

3.5.4.2 SPECIFICATION

Operating Voltage: **4.8-6.0V**
STD Direction: **Counter Clockwise / Pulse Travelling 800 to 2200µsec**
Stall Torque: **15kg/cm @ 4.8v, 19kg/cm @ 6v**
Operating Speed: **0.22sec/60deg @ 4.8v, 0.19sec/60 deg @ 6v**
Weight: **103g**
Size: **60 x 28 x 52 mm**
Lead Length: **250mm, JR style plug**
Spline Count: **25**

3.5.4.3 FEATURES

- 2pc Ball raced
- Alloy heatsink [middle case]

3.5.4.4 HARDWARE CONNECTION

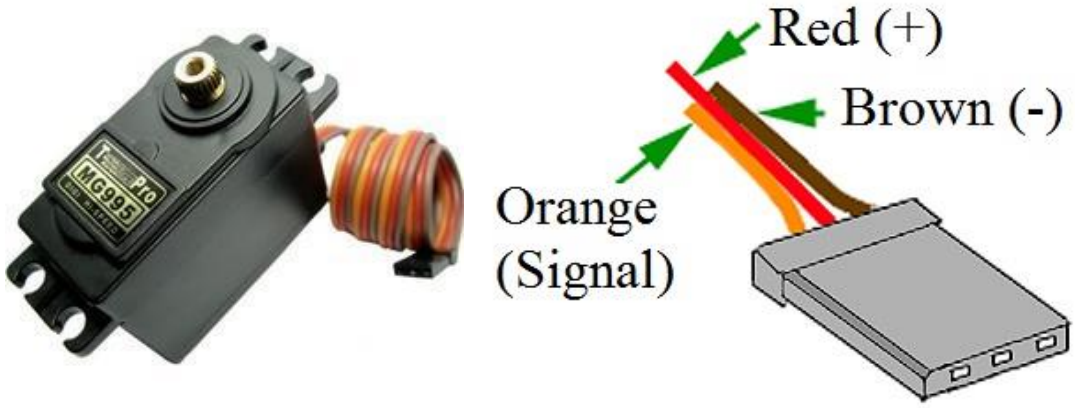

Fig: 3.11 Description of a Servo Motor

3.5.5 BUCK CONVERTOR

All electronic circuits works only in low DC voltage, so we need a power supply unit to provide the appropriate voltage supply for their proper functioning. This unit consists of transformer, rectifier, filter & regulator. AC voltage of typically 230 volts rms is connected to a transformer voltage down to the level to the desired ac voltage. A diode rectifier that provides the full wave rectified voltage that is initially filtered by a simple capacitor filter to produce a dc voltage. This resulting dc voltage usually has some ripple or ac voltage variation. A regulator circuit can use this dc input to provide dc voltage that not only has much less ripple voltage but also remains the same dc value even the dc voltage varies somewhat, or the load connected to the output dc voltages changes.

General Block of Power Supply Unit

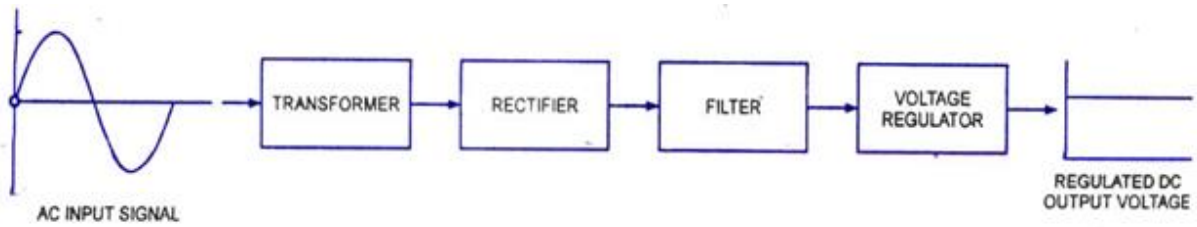

Fig: 3.12 Functional Block Diagram of Buck Convertor

3.5.5.1 TRANSFORMER

A transformer is a static piece of which electric power in one circuit is Transformed into electric power of same frequency in another circuit. It can raise or lower the voltage in the circuit, but with a corresponding decrease or increase in current. It works with the principle of mutual induction. In our project we are using a step down transformer to providing a necessary supply for the electronic circuits. Here we step down a 230volts ac into 12volts ac.

3.5.5.2 RECTIFIER

A dc level obtained from a sinusoidal input can be improved 100% using a process called full wave rectification. Here in our project for full wave rectification we use bridge rectifier. From the basic bridge configuration we see that two diodes (sayD2 & D3) are conducting while the other two diodes (D1 & D4) are in off state during the period t = 0 to T/2.Accordingly for the negative cycle of the input the conducting diodes are D1 & D4 .Thus the polarity across the load is the same. In the bridge rectifier the diodes may be of variable types like 1N4001, 1N4003, 1N4004, 1N4005, IN4007 etc… can be used. But here we use 1N4007, because it can withstand up to 1000v.

3.5.5.3 FILTERS

In order to obtain a dc voltage of 0 Hz, we have to use a low pass filter. So that a capacitive filter circuit is used where a capacitor is connected at the rectifier output & a dc is obtained across it. The filtered waveform is essentially a dc voltage with negligible ripples & it is ultimately fed to the load.

3.5.5.4 REGULATORS

The output voltage from the capacitor is more filtered & finally regulated. The voltage regulator is a device, which maintains the output voltage constant irrespective of the change in supply variations, load variations & temperature changes. Here we use fixed voltage regulator namely LM7805.The IC LM7805 is a +5v regulator which is used for microcontroller.

3.5.5.5 FEATURES AND DESCRIPTION

- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V
- Thermal Overload Protection
- Short Circuit Protection

3.5.6 DISPLAY

3.5.6.1 SPECIFICATION

Table: 3.1 Specification of OLED Display

VSS = 0 V, Ta = 25 °C

Item	Symbol	Min	Max	Unit	Note
Supply Voltage for Operation	V_{DD}	-0.3	4	V	Note 1, 2
Supply Voltage for Display	Vcc	0	15	V	
Operating Temperature	Top	-30	70	°C	
Storage Temperature	Tstg	-40	80	°C	
Static Electricity	Be sure that you are grounded when handling displays.				

Note 1: All the above voltages are on the basis of "VSS=0V".

3.5.6.2 PIN CONFIGURATION

Table: 3.2 Pin Configuration of OLED Display

Characteristics	Symbol	Conditions	Min	Typ	Max	Unit
Supply Voltage for Logic	V_{DD}		1.65	2.8	3.3	V
Supply Voltage for Display	V_{CC}	Note 3	7.0	7.25	7.5	V
Supply Voltage for DC/DC	V_{BAT}	Internal DC/DC	3.3	-	4.2	V
High Level Input	V_{IH}	-	$0.8xV_{DD}$	-	V_{DD}	V
Low Level Input	V_{IL}	-	0	-	$0.2xV_{DD}$	V
High Level Output	V_{OH}	Iout=100μA,3.3 MHz	$0.9xV_{DD}$	-	V_{DD}	V
Low Level Output	V_{OL}	Iout=100μA,3.3 MHz	0	-	$0.1xV_{DD}$	V
Operating Current for VDD	I_{DD}	Note 4 Note 5	-	180	300	μA
Operating Current for VBAT(VCC Generated by Internal DC/DC)	IBAT	Note 4	-	7.5	9.4	mA
		Note 5	-	12.6	15.8	mA
Sleep Mode Current for VCI	$I_{DD, SLEEP}$	-	-	1	5	μA
Sleep Mode Current for VCC	$I_{CC, SLEEP}$	-	-	1	5	μA

Note 3 Brightness (L_{br}) and Supply Voltage for Display (V_{CC}) is subject to the change of the panel characteristics and the customer's request.
Note 4 V_{DD}= 2.8V, V_{CC} = 7.25V, 50% Display area turned on.
Note 5 V_{DD} = 2.8V, V_{CC} = 7.25V, 100% Display area turned on

Fig: 3.13 OLED Display Module

3.5.6.3 MECHANICAL CHARACTERISTICS

Table: 3.3 Mechanical Characteristics of OLED Display

ITEM	CHARACTERISTIC	UNIT
Display Format	96 x 16	Dots
Overall Dimensions	29.10×9.20× 1.60	mm
Viewing Area	23.104 x 5.504	mm
Active Area	21.104 × 3.504	mm
Dot Size	0.204 × 0.204	mm
Dot Pitch	0.22 × 0.22	mm
Weight	0.89	g
IC Controller/Driver	SSD1306	

3.5.7 GOOGLE FIREBASE

Firebase is a platform that makes developing android apps easier. it is owned by Google and is easy to Integrate to your project It combines Analytics, Database, Authentication, Storage, Hosting, Crash Reports, AdMob, etc.. Google is trying to Integrate all basic services needed for an android app through Firebase Analytics- Provides Insights about your app like number of users, what are they doing with your app (you can track users based on buttons they click ,activities through which they navigate etc..). It also has other features such as:- Real time database, Push notification, Firebase Analytics, Firebase Authentication, Firebase Cloud Messaging, Firebase Storage, Firebase Hosting, Firebase Test Lab For Android, Firebase Crash reporting, Firebase Notification, Firebase App Indexing, Firebase Dynamic link, Firebase Invites and Firebase Adwords, etc.,

3.6 HARDWARE CONNECTION

3.6.1 MASTER MODULE

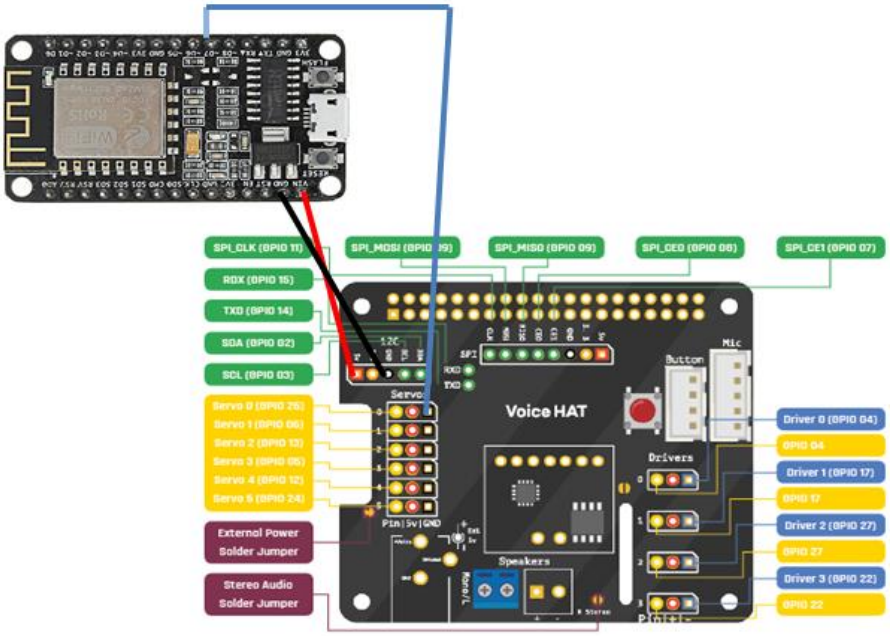

Fig: 3.14 Master Module Connection

3.6.2 CONTROL MODULE

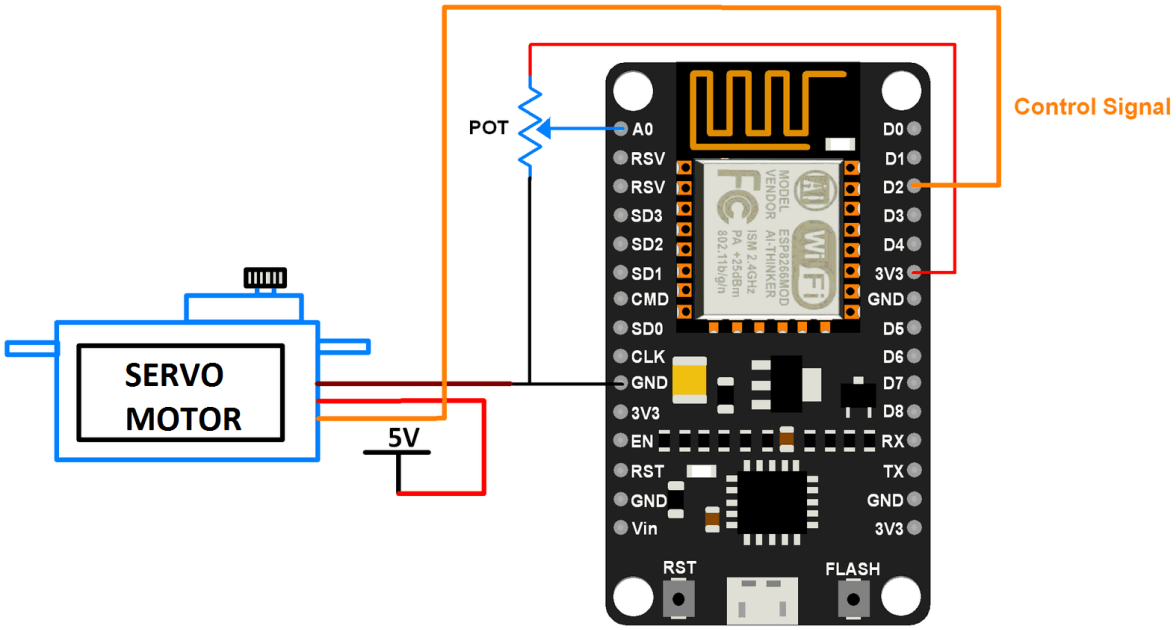

Fig: 3.15 Control Module Connection

3.7 SOFTWARE CODING

3.7.1 NODEMCU AT MASTER MODULE

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

// Set these to run example.
#define FIREBASE_HOST "nodemcu-17c74.firebaseio.com"
#define FIREBASE_AUTH "tt1J5M522Ww8UJpCGRCIqoxue1zTVb9Il1HzRAEp"
#define WIFI_SSID "sairam"
#define WIFI_PASSWORD "mukesh9739"

void setup() {

  Serial.begin(9600);
  // connect to wifi.
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);

  }

  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());
  pinMode(D0,INPUT);
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);

}

void loop() {

  // set value
  int a=0;
  a=digitalRead(D0);
  if(a==1)
     Firebase.setInt("number", 1);
  if(a==0)
     Firebase.setInt("number", 0);

}
```

3.7.2 RASPBERRY PI AT MASTER MODULE

```
import aiy.audio
import aiy.cloudspeech
import aiy.voicehat

import logging
import platform
import sys

import aiy.assistant.auth_helpers
from aiy.assistant.library import Assistant
from google.assistant.library.event import EventType

import RPI.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(26,GPIO.OUT)

logging.basicConfig(
    level=logging.INFO,
    format="[%(asctime)s] %(levelname)s:%(name)s:%(message)s"
)

def process_event(event):
    status_ui = aiy.voicehat.get_status_ui()
    if event.type == EventType.ON_START_FINISHED:
        status_ui.status('ready')
        if sys.stdout.isatty():
            print('Say "OK, Google" then speak, or press Ctrl+C to quit...')

    elif event.type == EventType.ON_CONVERSATION_TURN_STARTED:
        status_ui.status('listening')
        recognizer = aiy.cloudspeech.get_recognizer()
        recognizer.expect_phrase('turn on the light')
        recognizer.expect_phrase('turn off the light')
        text = recognizer.recognize()
        if 'turn on the light' in text:
            print (“hi”)
            GPIO.output(26,GPIO.HIGH)
        if 'turn off the light' in text:
            print(“hello”)
            GPIO.output(26,GPIO.LOW)

    elif event.type == EventType.ON_END_OF_UTTERANCE:
        status_ui.status('thinking')
```

```
    elif (event.type == EventType.ON_CONVERSATION_TURN_FINISHED
          or event.type == EventType.ON_CONVERSATION_TURN_TIMEOUT
          or event.type == EventType.ON_NO_RESPONSE):
        status_ui.status('ready')

    elif event.type == EventType.ON_ASSISTANT_ERROR and event.args and
event.args['is_fatal']:
        sys.exit(1)

def main():
    recognizer = aiy.cloudspeech.get_recognizer()
    recognizer.expect_phrase('turn off the light')
    recognizer.expect_phrase('turn on the light')
    recognizer.expect_phrase('blink')

    button = aiy.voicehat.get_button()
    led = aiy.voicehat.get_led()
    aiy.audio.get_recorder().start()

    while True:
        print('Press the button and speak')
        button.wait_for_press()
        print('Listening...')
        text = recognizer.recognize()
        if not text:
            print('Sorry, I did not hear you.')
        else:
            print('You said "', text, '"')
            if 'turn on the light' in text:
                led.set_state(aiy.voicehat.LED.ON)
            elif 'turn off the light' in text:
                led.set_state(aiy.voicehat.LED.OFF)
            elif 'blink' in text:
                led.set_state(aiy.voicehat.LED.BLINK)
            elif 'goodbye' in text:
                break
            else
                credentials = aiy.assistant.auth_helpers.get_assistant_credentials()
with Assistant(credentials) as assistant:
                for event in assistant.start():
                        process_event(event)

if __name__ == '__main__':
    main()
```

3.7.2 NODEMCU AT CONTROL MODULE

```
#include <ESPWiFi.h>
#include <ESPHTTPClient.h>
#if defined(ESP8266)
#include <Ticker.h>
#endif
#include <JsonListener.h>
#include "SSD1306Wire.h"
#include "OLEDDisplayUi.h"
#include "Wire.h"
#include "WundergroundClient.h"
#include "WeatherStationFonts.h"
#include "WeatherStationImages.h"
#include "TimeClient.h"
#include "ThingspeakClient.h"
#include <SPI.h>
#include "SSD1306Spi.h"
#include <Servo.h>
Servo myservo;

// WIFI
const char* WIFI_SSID = "sairam";
const char* WIFI_PWD = "mukesh9739";

// Setup
const int UPDATE_INTERVAL_SECS = 10 * 60; // Update every 10 minutes

// Display Settings
const int I2C_DISPLAY_ADDRESS = 0x3c;
#if defined(ESP8266)
const int SDA_PIN = D3;
const int SDC_PIN = D4;
#else
const int SDA_PIN = 5; //D3;
const int SDC_PIN = 4; //D4;
#endif

// TimeClient settings
const float UTC_OFFSET = 2;

// Wunderground Settings
const boolean IS_METRIC = true;
const String WUNDERGRROUND_API_KEY = "4cb629a83bff0210";
const String WUNDERGRROUND_LANGUAGE = "EN";
const String WUNDERGROUND_COUNTRY = "INDIA";
```

```
const String WUNDERGROUND_CITY = "CHENNAI";

//Thingspeak Settings
const String THINGSPEAK_CHANNEL_ID = "67284";
const String THINGSPEAK_API_READ_KEY = "L2VIW20QVNZJBLAK";
SSD1306Spi        display(D0, D2, D8);
OLEDDisplayUi   ui( &display );
WundergroundClient wunderground(IS_METRIC);
ThingspeakClient thingspeak;
bool readyForWeatherUpdate = false;
String lastUpdate = "--";

#if defined(ESP8266)
Ticker ticker;
#else
long timeSinceLastWUpdate = 0;
#endif

//declaring prototypes
void drawProgress(OLEDDisplay *display, int percentage, String label);
void updateData(OLEDDisplay *display);
void drawDateTime(OLEDDisplay *display, OLEDDisplayUiState* state, int16_t x, int16_t y);
void drawCurrentWeather(OLEDDisplay *display, OLEDDisplayUiState* state, int16_t x, int16_t y);
void drawForecast(OLEDDisplay *display, OLEDDisplayUiState* state, int16_t x, int16_t y);
void drawThingspeak(OLEDDisplay *display, OLEDDisplayUiState* state, int16_t x, int16_t y);
void drawForecastDetails(OLEDDisplay *display, int x, int y, int dayIndex);
void drawHeaderOverlay(OLEDDisplay *display, OLEDDisplayUiState* state);
void setReadyForWeatherUpdate();

FrameCallback frames[] = { drawDateTime, drawCurrentWeather, drawForecast, drawThingspeak };
int numberOfFrames = 4;

OverlayCallback overlays[] = { drawHeaderOverlay };
int numberOfOverlays = 1;
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>

// Set these to run example.
#define FIREBASE_HOST "nodemcu-17c74.firebaseio.com"
#define FIREBASE_AUTH "tt1J5M522Ww8UJpCGRCIqoxue1zTVb9Il1HzRAEp"
#define WIFI_SSID "sairam"
#define WIFI_PASSWORD "mukesh9739"
```

```
void setup() {
  Serial.begin(115200);
  Serial.println();
  Serial.println();
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  // initialize dispaly
  display.init();
  myservo.attach(D4);
  display.clear();
  display.display();
  //display.flipScreenVertically();
  display.setFont(ArialMT_Plain_10);
  display.setTextAlignment(TEXT_ALIGN_CENTER);
  display.setContrast(255);
  WiFi.begin(WIFI_SSID, WIFI_PWD);
  int counter = 0;
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
    display.clear();
    display.drawString(64, 10, "Connecting to WiFi");
    display.drawXbm(46, 30, 8, 8, counter % 3 == 0 ? activeSymbole : inactiveSymbole);
    display.drawXbm(60, 30, 8, 8, counter % 3 == 1 ? activeSymbole : inactiveSymbole);
    display.drawXbm(74, 30, 8, 8, counter % 3 == 2 ? activeSymbole : inactiveSymbole);
    display.display();
    counter++;
  }
  ui.setTargetFPS(30);
  ui.setActiveSymbol(activeSymbole);
  ui.setInactiveSymbol(inactiveSymbole);
  ui.setIndicatorPosition(BOTTOM);
  ui.setIndicatorDirection(LEFT_RIGHT);
  ui.setFrameAnimation(SLIDE_LEFT);
  ui.setFrames(frames, numberOfFrames);
  ui.setOverlays(overlays, numberOfOverlays);
  ui.init();
  Serial.println("");
```

```
  updateData(&display);
#if defined(ESP8266)
  ticker.attach(UPDATE_INTERVAL_SECS, setReadyForWeatherUpdate);
#endif
}

void loop() {
 int a = Firebase.getInt("number");
 Serial.println(a);
 if(a==0)
  { myservo.write(180);
 Serial.println("180");
 delay(2000);}
 if(a==1){
myservo.write(90);
Serial.println("90");
delay(2000);
 }
#if !defined(ESP8266)
  if (millis() - timeSinceLastWUpdate > (1000L*UPDATE_INTERVAL_SECS)) {
    setReadyForWeatherUpdate();
    timeSinceLastWUpdate = millis();
  }
#endif
  if (readyForWeatherUpdate && ui.getUiState()->frameState == FIXED) {
    updateData(&display);
  }
 int remainingTimeBudget = ui.update();
  if (remainingTimeBudget > 0) {
   delay(remainingTimeBudget);
  }
}
void drawProgress(OLEDDisplay *display, int percentage, String label) {
  display->clear();
  display->setTextAlignment(TEXT_ALIGN_CENTER);
  display->setFont(ArialMT_Plain_10);
  display->drawString(64, 10, label);
  display->drawProgressBar(2, 28, 124, 10, percentage);
  display->display();
}
void updateData(OLEDDisplay *display) {
  drawProgress(display, 10, "Updating time...");
  timeClient.updateTime();
  drawProgress(display, 30, "Updating conditions...");
```

```
  wunderground.updateConditions(WUNDERGRROUND_API_KEY,
WUNDERGRROUND_LANGUAGE, WUNDERGROUND_COUNTRY,
WUNDERGROUND_CITY);
  drawProgress(display, 50, "Updating forecasts...");
  wunderground.updateForecast(WUNDERGRROUND_API_KEY,
WUNDERGRROUND_LANGUAGE,WUNDERGROUND_COUNTRY,
WUNDERGROUND_CITY);
  drawProgress(display, 80, "Updating thingspeak...");
  thingspeak.getLastChannelItem(THINGSPEAK_CHANNEL_ID,
THINGSPEAK_API_READ_KEY);
  lastUpdate = timeClient.getFormattedTime();
  readyForWeatherUpdate = false;
  drawProgress(display, 100, "Done...");
  delay(1000);
}
void drawDateTime(OLEDDisplay *display, OLEDDisplayUiState* state, int16_t x, int16_t y) {
  display->setTextAlignment(TEXT_ALIGN_CENTER);
  display->setFont(ArialMT_Plain_10);
  String date = wunderground.getDate();
  int textWidth = display->getStringWidth(date);
  display->drawString(64 + x, 5 + y, date);
  display->setFont(ArialMT_Plain_24);
  String time = timeClient.getFormattedTime();
  textWidth = display->getStringWidth(time);
  display->drawString(64 + x, 15 + y, time);
  display->setTextAlignment(TEXT_ALIGN_LEFT);
}
void drawCurrentWeather(OLEDDisplay *display, OLEDDisplayUiState* state, int16_t x,
int16_t y) {
  display->setFont(ArialMT_Plain_10);
  display->setTextAlignment(TEXT_ALIGN_LEFT);
  display->drawString(60 + x, 5 + y, wunderground.getWeatherText());
  display->setFont(ArialMT_Plain_24);
  String temp = wunderground.getCurrentTemp() + "°C";
  display->drawString(60 + x, 15 + y, temp);
  int tempWidth = display->getStringWidth(temp);
  display->setFont(Meteocons_Plain_42);
  String weatherIcon = wunderground.getTodayIcon();
  int weatherIconWidth = display->getStringWidth(weatherIcon);
  display->drawString(32 + x - weatherIconWidth / 2, 05 + y, weatherIcon);
}
void drawForecast(OLEDDisplay *display, OLEDDisplayUiState* state, int16_t x, int16_t y) {
  drawForecastDetails(display, x, y, 0);
  drawForecastDetails(display, x + 44, y, 2);
  drawForecastDetails(display, x + 88, y, 4);
}
```

```
void drawThingspeak(OLEDDisplay *display, OLEDDisplayUiState* state, int16_t x, int16_t y)
{
  display->setTextAlignment(TEXT_ALIGN_CENTER);
  display->setFont(ArialMT_Plain_10);
  display->drawString(64 + x, 0 + y, "Outdoor");
  display->setFont(ArialMT_Plain_16);
  display->drawString(64 + x, 10 + y, thingspeak.getFieldValue(0) + "°C");
  display->drawString(64 + x, 30 + y, thingspeak.getFieldValue(1) + "%");
}
void drawForecastDetails(OLEDDisplay *display, int x, int y, int dayIndex) {
  display->setTextAlignment(TEXT_ALIGN_CENTER);
  display->setFont(ArialMT_Plain_10);
  String day = wunderground.getForecastTitle(dayIndex).substring(0, 3);
  day.toUpperCase();
  display->drawString(x + 20, y, day);
  display->setFont(Meteocons_Plain_21);
  display->drawString(x + 20, y + 12, wunderground.getForecastIcon(dayIndex));
  display->setFont(ArialMT_Plain_10);
  display->drawString(x + 20, y + 34, wunderground.getForecastLowTemp(dayIndex) + "|" +
wunderground.getForecastHighTemp(dayIndex));
  display->setTextAlignment(TEXT_ALIGN_LEFT);
}
void drawHeaderOverlay(OLEDDisplay *display, OLEDDisplayUiState* state) {
  display->setColor(WHITE);
  display->setFont(ArialMT_Plain_10);
  String time = timeClient.getFormattedTime().substring(0, 5);
  display->setTextAlignment(TEXT_ALIGN_LEFT);
  display->drawString(0, 54, time);
  display->setTextAlignment(TEXT_ALIGN_RIGHT);
  String temp = wunderground.getCurrentTemp() + "°C";
  display->drawString(128, 54, temp);
  display->drawHorizontalLine(0, 52, 128);
}
void setReadyForWeatherUpdate() {
  Serial.println("Setting readyForUpdate to true");
  readyForWeatherUpdate = true;
}
```

CHAPTER 4

RESULTS

4.1 MASTER MODULE

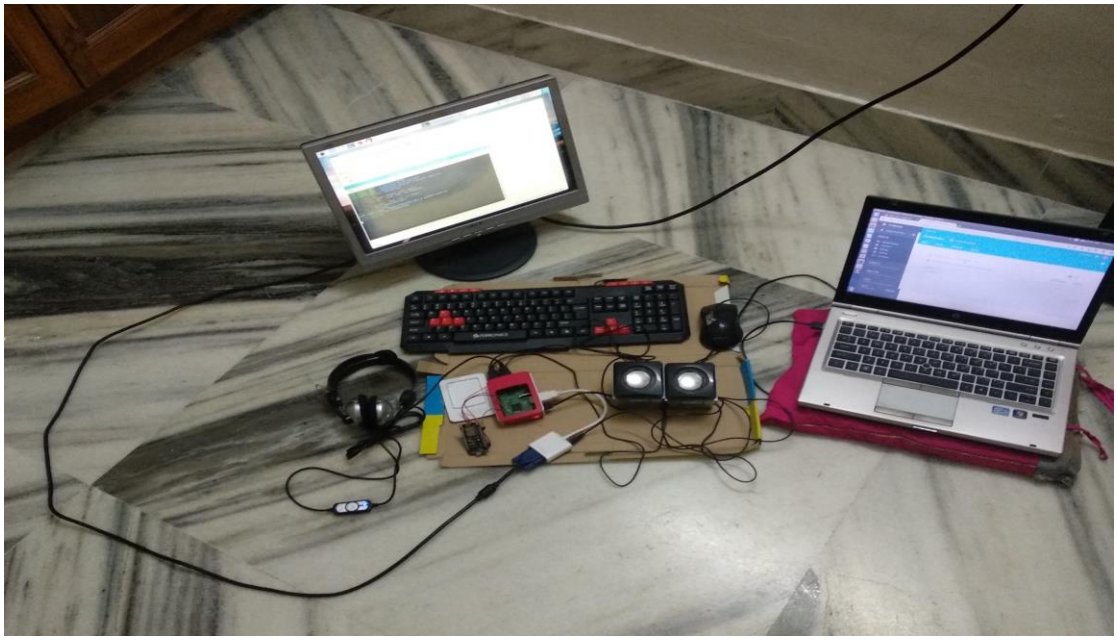

Fig: 4.1 Master Module Configuration

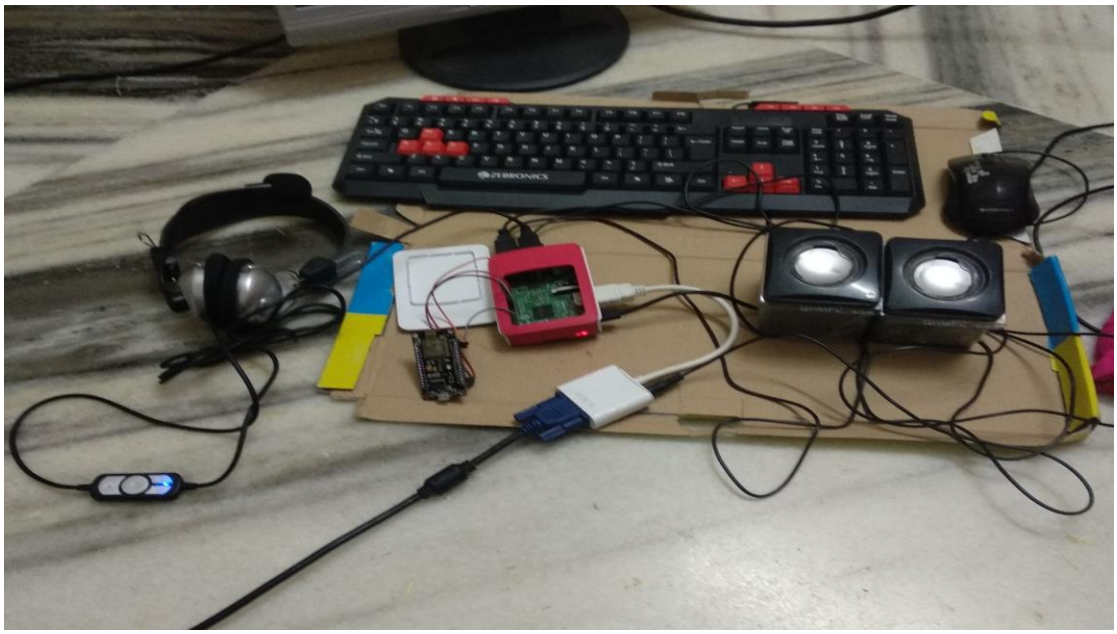

Fig: 4.2 closer look into Master Module Configuration

4.2 FIREBASE CONSOLE

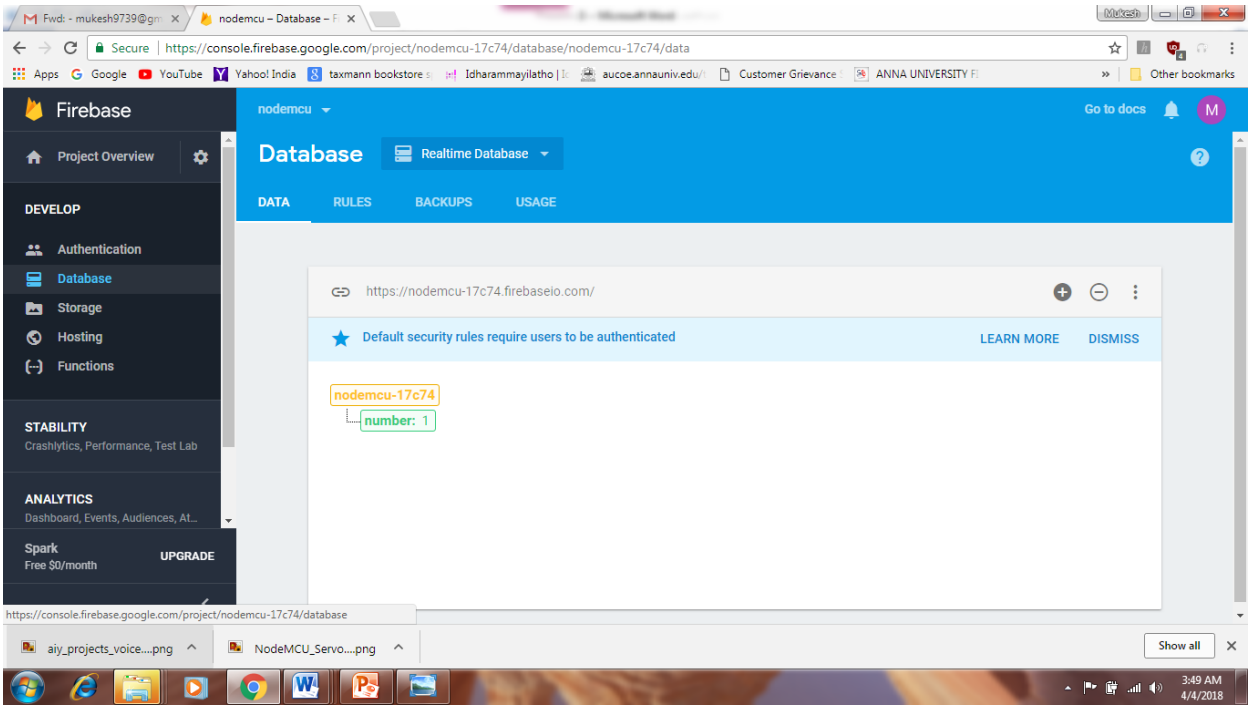

Fig: 4.3 Firebase console – Real time DataBase

4.3 CONTROL MODULE

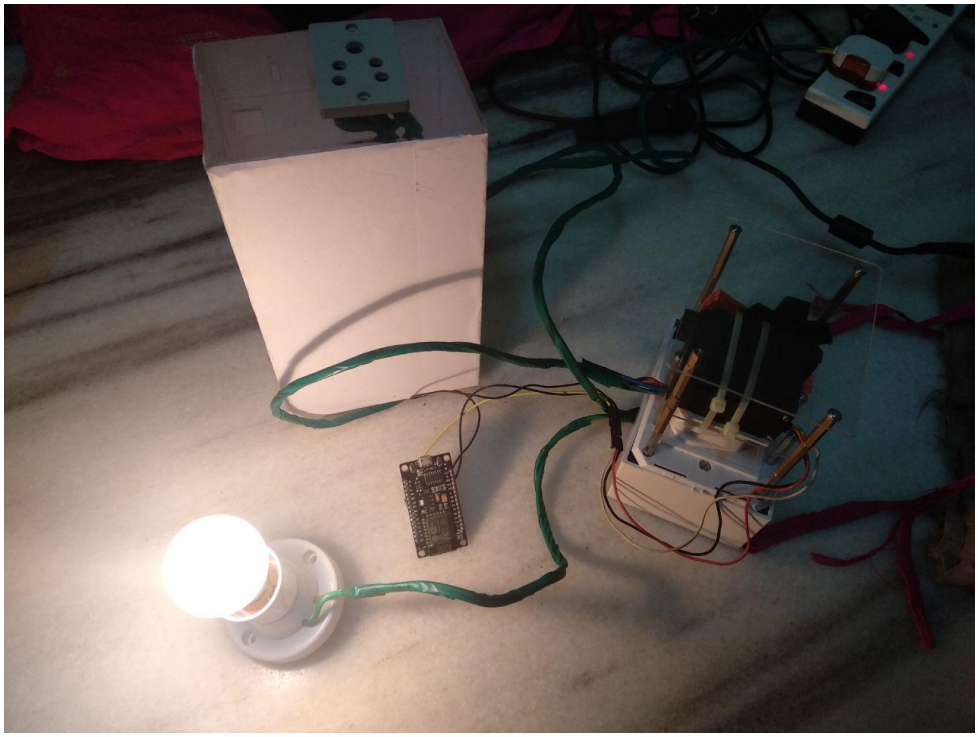

Fig: 4.4 Control Module Setup

CHAPTER 5

CONCLUSION

5.1 FUTURE SCOPE

- Automatic lighting control based on moods
- HVAC regulation
- Lawn irrigation systems
- Highly secure systems
- Connecting with Robotics and healthcare systems
- The control module can be made for all standard sizes of switch covers and for all types of switches.
- Further add-ons can be a Wi-Fi interface and Application to the display present in the control module.
- The size of the master module can be further reduced.

5.2 CONCLUSION

The possibilities are numerous and can be explored further. A smart home contains a connection between wireless communication, sensors, monitoring and tracking. Smart homes are a huge system that includes multiple technologies and applications that can be used to provide security and control of the home easily. In this project, an efficient approach for smart homes was proposed and implemented. In future, this project have high potential for marketing. This project offers best in class service and more affordable for common people.