

**Home Automation Using IoT**

# Introduction

# 1.1 Background

Internet of Things is a concept where each device is assign to an IP address and through that IP address anyone makes that device identifiable on internet. The Internet is an evolving entity. It started as the Internet of Computers. Research studies have forecast an explosive growth in the number of things or devices that will be connected to the Internet. The resulting network is called the Internet of Things (IoT). IoT is having the potential to change the lifestyle of peoples. In day today’s life, people prefer more of automatic systems rather than any manual systems. The major elements of IoT based home automation system are Raspberry pi and the Relay along with their driving circuitry. Home automation can be defined as a mechanism removing as much human interaction as technically possible and desirable in various domestic processes and replacing them with programmed electronic systems. Ultimately it is a system that aims to heighten quality of life with the automation of household appliances that may be controlled over the Internet or Telephone. The overall design of Home Automation System implements low cost wireless communication between a Raspberry Pi module and an android cell phone.Raspberry pi is having different communication media like Ethernet port, HDMI port, USB port, Display Serial Interface, Camera Serial Interface, Bluetooth, Bluetooth low energy. It allows to control number of home appliances simultaneously. This project is intended to construct an automation system which can control your electrical appliances with voice commands using Google application programming interfaces (API). A home automation system that uses any mobile device to control the home appliances. This home automation is based on Internet of Things (IoT) & Raspberry pi which is credit card size computer. Raspberry pi supports large number of peripherals. The IoT based Home Automation will enable the user to use a Home Automation System based on Internet of Things (IoT). The user commands over the internet will be obtained by the Wi-Fi modems. The system status is displayed through the LCD display, along with the system data. This is a typical IoT based Home Automation system, for controlling all your home appliances.

# 1.2 Motivation

**You can be better about how much water you use.** In some locations, the cost of water can fluctuate during on- and off-peak times. Adding sprinkler control to your home automation system may qualify you for some rebates while allowing you to control, through a smartphone app, how long the water runs, potentially cutting your water bill significantly. A system like Rachio will even shut down the sprinklers automatically when rain is in the forecast. These days a lot of natural resources which we depend on our daily basis of life like water, electricity and many more but we have to save water from getting wasted.

**You'll use your appliances at just the right time**. Some home automation systems can measure the energy usage of individual appliances, allowing you to figure out whether an older unit is draining too much energy. (Most of these units are extremely complex to set up and use right now, so easy DIY options are still a few years off, unless you want to try an option like the Aeon Labs Z-Wave Smart Energy Switch.) Or you may be able to connect your system to the power company, allowing the home automation system to determine when power rates are at their lowest and then run the appliances automatically at the desirable off-peak times.

**You'll be alerted to maintenance issues.** When your home has smart appliances, they can alert you to potential problems they're having, such as a filter that needs changing. Perhaps your freezer is not reaching the proper temperature. It can alert you to this problem through the home automation system. It's nice to have a warning about a potential impending failure of your freezer, versus having no warning and coming home to a puddle on the floor and a unit filled with thawed, ruined food.

**You can control security lights.** If you have a series of security lights at your home, you can turn

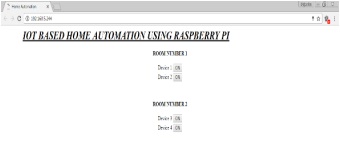
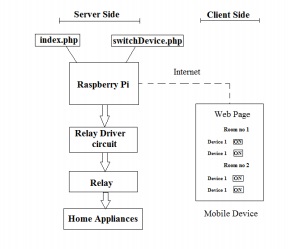
them on and off using a smartphone app or a motion detection system. Or you could have the motion detectors alert you through the home automation system any time they detect movement on your property. And the detectors could automatically turn on any security cameras you have as soon as they activate the lights.

**You can fool potential burglars.** Going on vacation? Rather than plugging your lamps into traditional timers, causing the same pattern of lights to activate every night, you can use your home automation system to turn different lights inside the home on and off — not just lamps — and use varying patterns, just as if someone were home and turning lights on and off as she moves through the house. You can turn a TV and radio on and off through the home automation system too.

**You can activate your security system.** If you pay for a home security system, or if you have your own DIY solution, you can tie these devices into your home automation system, receiving alerts if anything goes wrong while you're away from home. Or you can make changes to the security system's settings from anywhere through a smartphone app, giving you an option for battling potential burglars.

# Literature survey

# 2.1 Home Automation using Raspberry Pi controlled via an Android Application- *Phanm et al., IJISSET 2017*



*Figure 1* *Block diagram Figure 2 control panel*

This paper provides a simple introduction to the IoT, its application and potential benefits to the society. IoT has received much attention from scientists, industry and government all over the world for its potential in changing modern day living. IoT is envisioned as billions of sensors connected to the internet through wireless and other communication technologies. The sensors would generate large amount of data which needs to be analyzed, interpreted and utilized. [1] Home Automation System uses the technology of Internet of Things for monitoring and controlling of the electrical and

electronic appliances at home from any remote location by simply using a Smartphone. Implementation of a low cost, flexible home automation system is presented.

# 2.2 Home Automation using Raspberry Pi controlled via an Android Application – *Asmm et al. INPRESSCO (2017)*

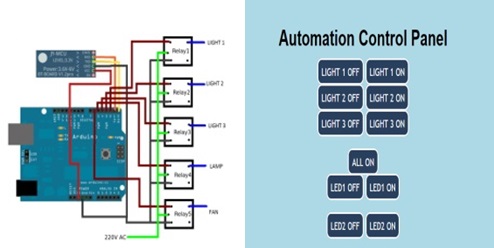


*Figure 3* *the control of appliances using mobile application.*

In order to address the mentioned issues of functionality and flexibility, we designed and implemented standalone, novel, flexible and low-cost home monitoring and controlling system using Web services as an interoperable application layer. The system consists of a micro Web - server based on Arduino Ethernet and hardware interface modules. The architecture presented in this work can be customized in several ways in order to accommodate different application scenarios with minimum design and recoding i.e. each time a new device is added to the micro Web-server, a new thread dedicated to the device can be added to the web server. [2] Hence, the aim

of the proposed work is not to incorporate expensive components such as high-end personal computers. This system allows authorized home owners to control and monitor connected devices at home. The proposed system permits the owner or the controller, all the rights to take decisions and to regulate the home appliances with the help of an android application. It provides effortless as well as varied ways to control the devices in the house, thus making one’s life comfortable and at the same time remotely accessible through portable devices like android phones. In the future extensions to the project, the smart Doorbell can be made by implementing voice and video calls with the person standing right outside the door and the owner remotely. Thereby increasing the safety quotient of the system.

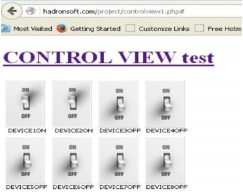
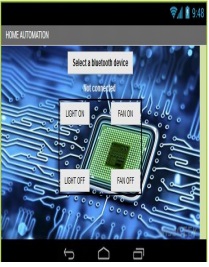
# 2.3 Home Automation Using aurdino and IoT- *MVIT et al. IRJET (2017)*



*Figure 4* *the connection of the system and control panel of the system to ON/OFF electrical appliances.*

The system consists of a micro Web - server based on Arduino Ethernet and hardware interface modules. The architecture presented in this work can be customized in several ways in order to accommodate different application scenarios with minimum design and recoding i.e. each time a new device is added to the micro Web-server, a new thread dedicated to the device can be added to the web server. Hence, the aim of the proposed work is not to incorporate expensive components such as high end personal computers. This system allows authorized home owners to control and monitor connected devices at home. The web server provides a graphical user interface (GUI) for accessing and controlling the devices. Once the connections have been made, the code written in the Arduino IDE must be uploaded to the device. The IP address of the device is to be entered in a web browser in order to open the front end web page. The web page has several buttons which may be used to control the given devices. [3] The home automation using Internet of Things has been experimentally proven to work by connecting simple appliances to it. These appliances were successfully controlled remotely through the internet. The designed system instigates a process according to the user’s requirements, for example switching on a fan when it gets hot. Sensors can be implemented to store data which can later be used to analyze the system at hand.

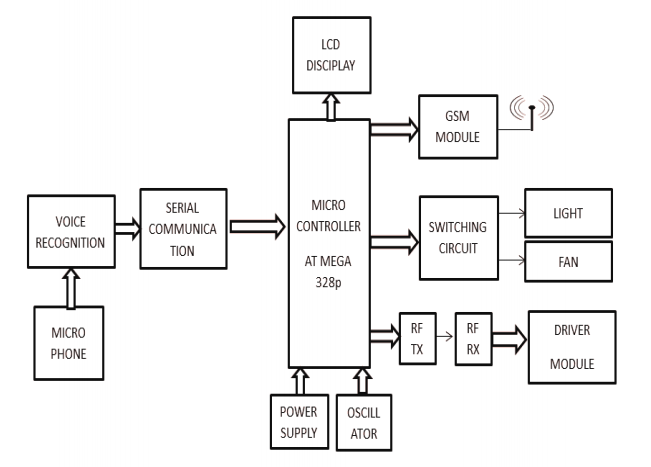
# 2.4 Smart home automation based on IOT and android technology- *ISSN et al. IJESC (2017)*



*Figure 5* *the control of appliances using mobile application and internet browser.*

The fundamental downside of Bluetooth based home mechanization can be defeat utilizing IOT innovation. In this segment, Ethernet module is utilized for interfacing Arduino board from any piece of the world. Arduino is associated with IOT and Port number can be utilized to find remote gadget associated with the Internet in a brilliant home condition. Android versatile application can be utilized to control electrical apparatuses from a remote area. Ethernet shield is put quite recently above Arduino board which is associated with RJ-45 for Internet availability [4].In this design, Arduino board is arranged as a server. At whatever point client enters IP address and Port number, demand will be sent to Arduino board, which thusly serves a HTML web page which is put away in Arduino's miniaturized scale SD card. Driven relating electrical machine can be exchanged ON/OFF utilizing program appeared.

# 2.5 Voice recognition based home automation system for paralyzed people - *NCEE-2017*



*Figure 6* *Block diagram of given system*

The voice recognition based home automation system for paralyzed people uses the speech recognition module to recognize the voice commands. The RF trans-receiver is used to send these commands to the controller to control the various electrical devices and robotic wheel chair. This proposed a home automation system uses the voice recognition module v3 for the voice recognition function, a microcontroller and relay module are used for the controlling functions like switching lights on and off etc. Here RF trancereciever is used which eliminates or reduces the need of additional wiring required for the signal transmission. We can also control the home appliances by two methods by voice commands or by using android mobile as a remote controller. The voice recognition is done by the android application and thus given to the controller to control the devices. When the key on remote controller is pressed the controller sends the commands via Bluetooth module to the receiver.

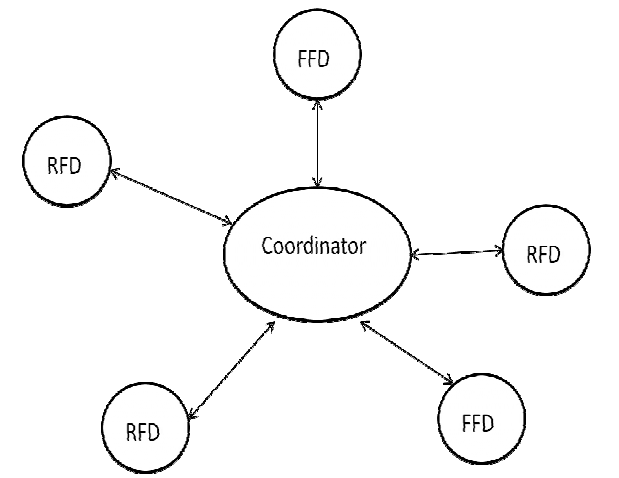


Figure 7 Star Network configuration

The ZigBee protocol uses the IEEE 802.15.4 specification as its Medium Access Layer (MAC) and Physical Layer (PHY). The IEEE 802.15.4 defines three frequency bands of operations: 2.4GHz, 915MHz and 868MHz. Each frequency band offers a fixed number of channels. For example, the 2.4 GHz frequency band offers 16 channels (channels 11-26), 915 MHz offers10 channels (channels 1- 10) and 868 MHz offers 1 channel (channel 0). The MRF24J40 is an IEEE 802.15.4 Standard compliant 2.4 GHz RF transceiver. It integrates the PHY and MAC functionality in a single chip solution. Here we have used star topology as shown in figure 1. The ZigBee protocol coordinator is a special variant of a Full Function Device configuration (FFD) that implements a larger set of ZigBee protocol services. An end device may be an FFD, like a Router, or a Reduced Function Device (RFD). RFD is the smallest and simplest ZigBee protocol node.

# 2.6 Home automation control system using SMS – *IRJET 2017*

The main concept behind this project is receiving the sent SMS and further processing it as required to perform several operations. The nature of the SMS sent decides the type of operation. The project is based on a very simple principle. First, the SMS sent is stored and collected from the receiving mobile station and then the required control signal is generated and sent to the intermediate hardware that we have designed according to the command received from the sent message. A microcontroller based using SMS system has been proposed for this project. GSM (Global System for Mobile Communications): It is a cellular communication standard. SMS (Short Message Service): it is a service available on digital mobile phones that permits the sending of short text messages (text messaging service). Short Message Service (SMS) is a text messaging service important component of any phone, web, or mobile communication systems. By using standardized communication protocols it allows the exchange of short text messages between mobile phone devices. SMS used for text messaging is the most widely used data application SMS is used on modern handsets was originated from radio telegraphy in radio memo-pagers by using standardized phone protocols and later it defined as the part of the Global System for Mobile Communications (GSM) as a mean of sending messages of up to 160 characters, to and from GSM mobile handsets. Since then, support for this service has been expanded to include other mobile technologies such as ANSI CDMA networks and Digital AMPS, with satellite and landline networks. Most SMS messages are mobile-to-mobile text messages though the standard supports of mobile network

# 2.7 Home automation using ZigBee – *IJETTCS 2017*

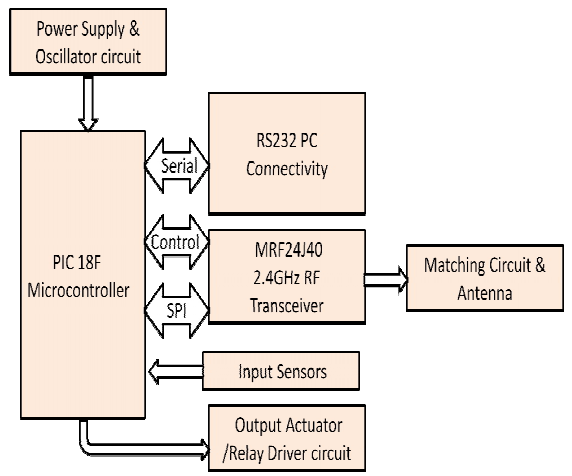


Figure 8 Block Diagram of ZigBee

The Home Automation system divides basically in three layers. The top-most layer is the input/sensor layer, which can be from sensors, limit switches, or any other kind of input. The middle layer is control unit. This layer reads the inputs and performs actions depending on the values of the inputs and the control program specified by the user. Finally, the control unit outputs commands to the actuator layer. The actuator layer is responsible for forming and sending commands to the real-world systems such as lights, fans, Air conditioner, security systems, and music systems etc. There are still some challenges of ZigBee based system like resources constraint- sensors (node) used in system, limited range, interference with other wireless system, technological limitation depend on IEEE 802.15.4 standard. This ZigBee based system is used for remote controlling and monitoring of various home loads/appliances. The ultimate objective is efficient power utilization through real time power level indicator with the help of a PC-based GUI application.

# 2.8 IOT Based Secured Smart Home Automation Using Raspberry Pi – *IJEEE 2017*

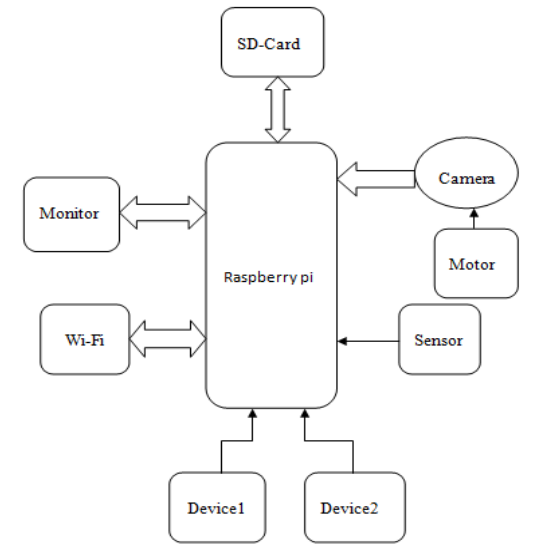
 

Figure 9 System Model

A custom-made Raspberry Pi will be fitted at each power points or switch boards. It will act as the control for all electrical appliances (lighting, fans, air conditioners etc.). There will be no work for the user regarding his/her appliance. One has to initialize the required settings at the time of setting up of the system. After that the system will be individual and self-sustained. The custom Raspberry Pi will have relays fixed on, which will control all lighting and fans or any other electrical appliances. This board will have a Wireless connection that connects to an Internet hub. This Internet hub will be connected to the internet via LAN or Wi-Fi (Depends upon the choice of the user). As mentioned earlier the internet acts as a master since the entire control process is taken care of by an online server-side program (ASP or PHP modules). The user just has to login in to the specified webpage during the time of initialization and in case there is a need to change the automation settings. The web-page will be coded in such a way that it provides complete control to the user over the automation process such as timing and conditions for the automation process.

# 2.9 Home Automation with Indoor Location Using Bluetooth Low Energy and Arduino-ZigBee Nodes – *IRJET 2017*

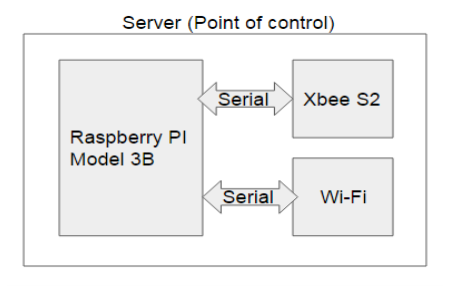
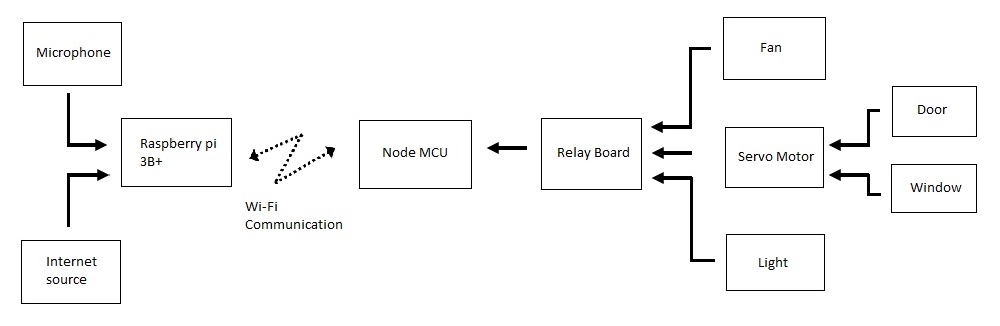


Figure 10 Block diagram

The smartphone/web client uses JSON objects and REST API implemented in the server to communicate directionally with the server, the server abstracts the ZigBee interface layer, uses a simple datagram to deliver information to the respective ZigBee device after which the paired Arduino will do the respective action. Use of REST API allows any device to communicate with server thus making it flexible. The implementation of the system was a success and is robust enough to be used on a 24x7 basis without failure with various normal domestic loads, the BLE based indoor location system is a game changer for low cost home automation systems as it allows for a new level of personalization without having to use expensive power consuming RFID based systems, the entire system has been designed with immense flexibility in mind and is suitable for medium industry level automation as well with users having Bluetooth tags and BLE receivers in each room allowing good ERP and employee tracking.

# Methodology

# 3.1 Block diagram of the proposed system



*Figure 11* *Block Diagram of the system*

# 3.2 Specifications of the components:

1. Raspberry pi 3B+:

Which contains following specification:

* **SOC**: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC
* **CPU**: 1.4GHz 64-bit quad-core ARM Cortex-A53 CPU
* **RAM**: 1GB LPDDR2 SDRAM
* **WIFI**: Dual-band 802.11ac wireless LAN (2.4GHz and 5GHz ) and Bluetooth 4.2
* **Ethernet**: Gigabit Ethernet over USB 2.0 (max 300 Mbps). Power-over-Ethernet support (with separate PoE HAT). Improved PXE network and USB mass-storage booting.
* **Thermal management**: Yes
* **Video**: Yes – Video Core IV 3D. Full-size HDMI
* **Audio**: Yes
* **USB** 2.0: 4 ports
* **GPIO**: 40-pin
* **Power**: 5V/2.5A DC power input
* **Operating system support**: Linux and Unix

**2.** Nodemcu: The Nodemcu (*N*ode *M*icro*c*ontroller *U*nit) is open-source software and hardware development environment built around an inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266, designed and manufactured by Espressif Systems, contains the crucial elements of a computer: CPU, RAM, networking (Wi-Fi), and even a modern operating system and SDK.

3. Relay Board:

Relay boards are computer boards with an array of relays and switches. They have input and output terminals and are designed to control the voltage supply. Relay boards provide independently programmable, real-time control for each of several onboard relay channels.

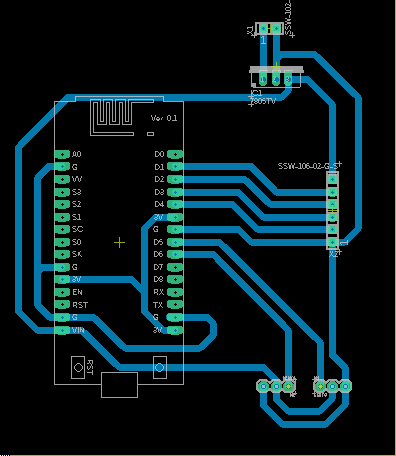
4. Water pump:

A pump is a device that moves fluids ([liquids](https://en.wikipedia.org/wiki/Liquid) or [gases](https://en.wikipedia.org/wiki/Gas)), or sometimes [slurries](https://en.wikipedia.org/wiki/Slurry), by mechanical action. Pumps can be classified into three major groups according to the method they use to move the fluid: *direct lift*, *displacement*, and *gravity* pumps.

5. Servo Motor 3.2k torque:

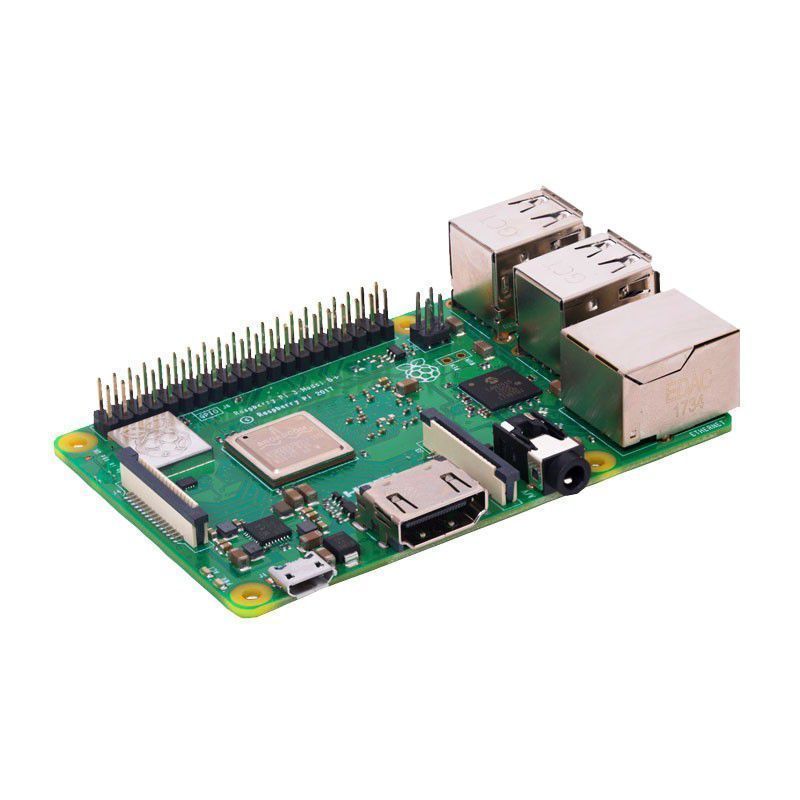
A servomotor is an electrical device which can push or rotate an object with great precision. If you want to rotate and object at some specific angles or distance, then you use servo motor. It is just made up of simple motor which run through servo mechanism.

# 3.3 PCB of Nodemcu unit also designed



*Figure 12* *pcb design (Eagle)*

# 3.4 Programmed Raspberry pi 3b+ to use it in headless mode for voice recognition using google API.



*Figure 13* *Raspberry 3B+*

# 3.5 Code used to program raspberry pi for setup

sudo apt-get purge python=2.7.0

sudo apt-get install python

sudo apt-get install git

git clone https://github.com/shivasiddharth/GassistPi

sudo apt-get update

cd /home/${USER}/

sudo chmod +x ./GassistPi/audio-drivers/USB-MIC-JACK/scripts/usb-mic-onboard-jack.sh

sudo ./GassistPi/audio-drivers/USB-MIC-JACK/scripts/usb-mic-onboard-jack.sh

speaker-test

cd /home/${USER}/

sudo chmod +x ./GassistPi/scripts/gassist-installer.sh

sudo ./GassistPi/scripts/gassist-installer.sh

# 3.6 Code used to get raspberry pi into headless mode

cd /home/${USER}

sudo chmod +x ./GassistPi/scripts/service-installer.sh  
sudo ./GassistPi/scripts/service-installer.sh  
sudo systemctl enable gassistpi.service  
sudo systemctl start gassistpi.service  
  
//--------------------For manual Start----------------------------  
  
sudo systemctl stop gassistpi.service  
/home/${USER}/env/bin/python -u /home/${USER}/GassistPi/src/main.py --device\_model\_id 'replace this with the model id'

# 3.7 Code for the nodemcu

#include <PubSubClient.h>

#include <ESP8266WiFi.h>

#include <ArduinoOTA.h>

#include <Servo.h>

void callback(char\* topic, byte\* payload, unsigned int length);

//EDIT THESE LINES TO MATCH YOUR SETUP

#define MQTT\_SERVER "192.168.43.104" //you MQTT IP Address

const char\* ssid = "cast";

const char\* password = "87654321";

//EJ: Data PIN Assignment on WEMOS D1 R2 https://www.wemos.cc/product/d1.html

// if you are using Arduino UNO, you will need to change the "D1 ~ D4" with the corresponding UNO DATA pin number

const int switchPin1 = D1;

const int switchPin2 = D2;

const int switchPin3 = D3;

const int switchPin4 = D4;

const int servoPin1 = D5;

const int servoPin2 = D6;

Servo myservo1;

Servo myservo2;

//EJ: These are the MQTT Topic that will be used to manage the state of Relays 1 ~ 4

//EJ: Refer to my YAML component entry

//EJ: feel free to replicate the line if you have more relay switch to control, but dont forget to increment the number suffix so as increase switch logics in loop()

char const\* switchTopic1 = "house/switch1/control";

char const\* switchTopic2 = "house/switch2/control";

char const\* switchTopic3 = "house/switch3/control";

char const\* switchTopic4 = "house/switch4/control";

char const\* servoTopic1 = "house/servo1/control";

char const\* servoTopic2 = "house/servo2/control";

WiFiClient wifiClient;

PubSubClient client(MQTT\_SERVER, 1883, callback, wifiClient);

void setup() {

//initialize the servo at servoPin and set to 180

myservo1.attach(servoPin1);

myservo1.write(180);

myservo2.attach(servoPin2);

myservo2.write(180);

//initialize the switch as an output and set to LOW (off)

pinMode(switchPin1, OUTPUT); // Relay Switch 1

digitalWrite(switchPin1, LOW);

pinMode(switchPin2, OUTPUT); // Relay Switch 2

digitalWrite(switchPin2, LOW);

pinMode(switchPin3, OUTPUT); // Relay Switch 3

digitalWrite(switchPin3, LOW);

pinMode(switchPin4, OUTPUT); // Relay Switch 4

digitalWrite(switchPin4, LOW);

ArduinoOTA.setHostname("NodeMcu\_Test"); // A name given to your ESP8266 module when discovering it as a port in ARDUINO IDE

ArduinoOTA.begin(); // OTA initialization

//start the serial line for debugging

Serial.begin(115200);

delay(100);

//start wifi subsystem

WiFi.begin(ssid, password);

//attempt to connect to the WIFI network and then connect to the MQTT server

reconnect();

//wait a bit before starting the main loop

delay(2000);

}

void loop(){

//reconnect if connection is lost

if (!client.connected() && WiFi.status() == 3) {reconnect();}

//maintain MQTT connection

client.loop();

//MUST delay to allow ESP8266 WIFI functions to run

delay(10);

ArduinoOTA.handle();

}

void callback(char\* topic, byte\* payload, unsigned int length) {

//convert topic to string to make it easier to work with

String topicStr = topic;

//EJ: Note: the "topic" value gets overwritten everytime it receives confirmation (callback) message from MQTT

//Print out some debugging info

Serial.println("Callback update.");

Serial.print("Topic: ");

Serial.println(topicStr);

if (topicStr == "house/switch1/control")

{

//turn the switch on if the payload is '1' and publish to the MQTT server a confirmation message

if(payload[0] == '1'){

digitalWrite(switchPin1, HIGH);

client.publish("house/switch1/status", "ON");

}

//turn the switch off if the payload is '0' and publish to the MQTT server a confirmation message

else if (payload[0] == '0'){

digitalWrite(switchPin1, LOW);

client.publish("house/switch1/status", "OFF");

}

}

// EJ: copy and paste this whole else-if block, should you need to control more switches

else if (topicStr == "house/switch2/control")

{

//turn the switch on if the payload is '1' and publish to the MQTT server a confirmation message

if(payload[0] == '1'){

digitalWrite(switchPin2, HIGH);

client.publish("house/switch2/status", "ON");

}

//turn the switch off if the payload is '0' and publish to the MQTT server a confirmation message

else if (payload[0] == '0'){

digitalWrite(switchPin2, LOW);

client.publish("house/switch2/status", "OFF");

}

}

else if (topicStr == "house/switch3/control")

{

//turn the switch on if the payload is '1' and publish to the MQTT server a confirmation message

if(payload[0] == '1'){

digitalWrite(switchPin3, HIGH);

client.publish("house/switch3/status", "ON");

}

//turn the switch off if the payload is '0' and publish to the MQTT server a confirmation message

else if (payload[0] == '0'){

digitalWrite(switchPin3, LOW);

client.publish("house/switch3/status", "OFF");

}

}

else if (topicStr == "house/switch4/control")

{

//turn the switch on if the payload is '1' and publish to the MQTT server a confirmation message

if(payload[0] == '1'){

digitalWrite(switchPin4, HIGH);

client.publish("house/switch4/status", "ON");

}

//turn the switch off if the payload is '0' and publish to the MQTT server a confirmation message

else if (payload[0] == '0'){

digitalWrite(switchPin4, LOW);

client.publish("house/switch4/status", "OFF");

}

}

if (topicStr == "house/servo1/control")

{

char \*lock = (char \*) payload; // convert mqtt byte array into char array

lock[length] ='\0';

int i = atoi(lock);

String paylo = String(i);

char attrib[5];

paylo.toCharArray(attrib,5);

myservo1.write(i);// tell servo to go to position in variable '(int)payload[i]'

client.publish("house/servo1/status", attrib );

}

if (topicStr == "house/servo2/control")

{

char \*lock = (char \*) payload; // convert mqtt byte array into char array

lock[length] ='\0';

int i = atoi(lock);

String paylo = String(i);

char attrib[5];

paylo.toCharArray(attrib,5);

myservo2.write(i);// tell servo to go to position in variable '(int)payload[i]'

client.publish("house/servo2/status", attrib );

}

}

void reconnect() {

//attempt to connect to the wifi if connection is lost

if(WiFi.status() != WL\_CONNECTED){

//debug printing

Serial.print("Connecting to ");

Serial.println(ssid);

//loop while we wait for connection

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

//print out some more debug once connected

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

}

//make sure we are connected to WIFI before attemping to reconnect to MQTT

if(WiFi.status() == WL\_CONNECTED){

// Loop until we're reconnected to the MQTT server

while (!client.connected()) {

Serial.print("Attempting MQTT connection...");

// Generate client name based on MAC address and last 8 bits of microsecond counter

String clientName;

clientName += "esp8266-";

uint8\_t mac[6];

WiFi.macAddress(mac);

clientName += macToStr(mac);

//if connected, subscribe to the topic(s) we want to be notified about

//dedsec995: Delete "mqtt\_username", and "mqtt\_password" here if you are not using any

if (client.connect((char\*) clientName.c\_str())) { //EJ: Update accordingly with your MQTT account

Serial.print("\tMQTT Connected");

client.subscribe(switchTopic1);

client.subscribe(switchTopic2);

client.subscribe(switchTopic3);

client.subscribe(switchTopic4);

client.subscribe(servoTopic1);

client.subscribe(servoTopic1);

//EJ: Do not forget to replicate the above line if you will have more than the above number of relay switches

}

//otherwise print failed for debugging

else{Serial.println("\tFailed."); abort();}

}

}

}

//generate unique name from MAC addr

String macToStr(const uint8\_t\* mac){

String result;

for (int i = 0; i < 6; ++i) {

result += String(mac[i], 16);

if (i < 5){

result += ':';

}

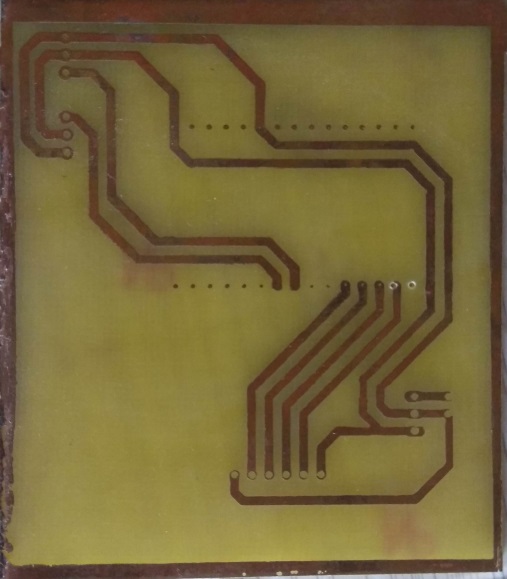
}

return result;

}

# Results and Discussion

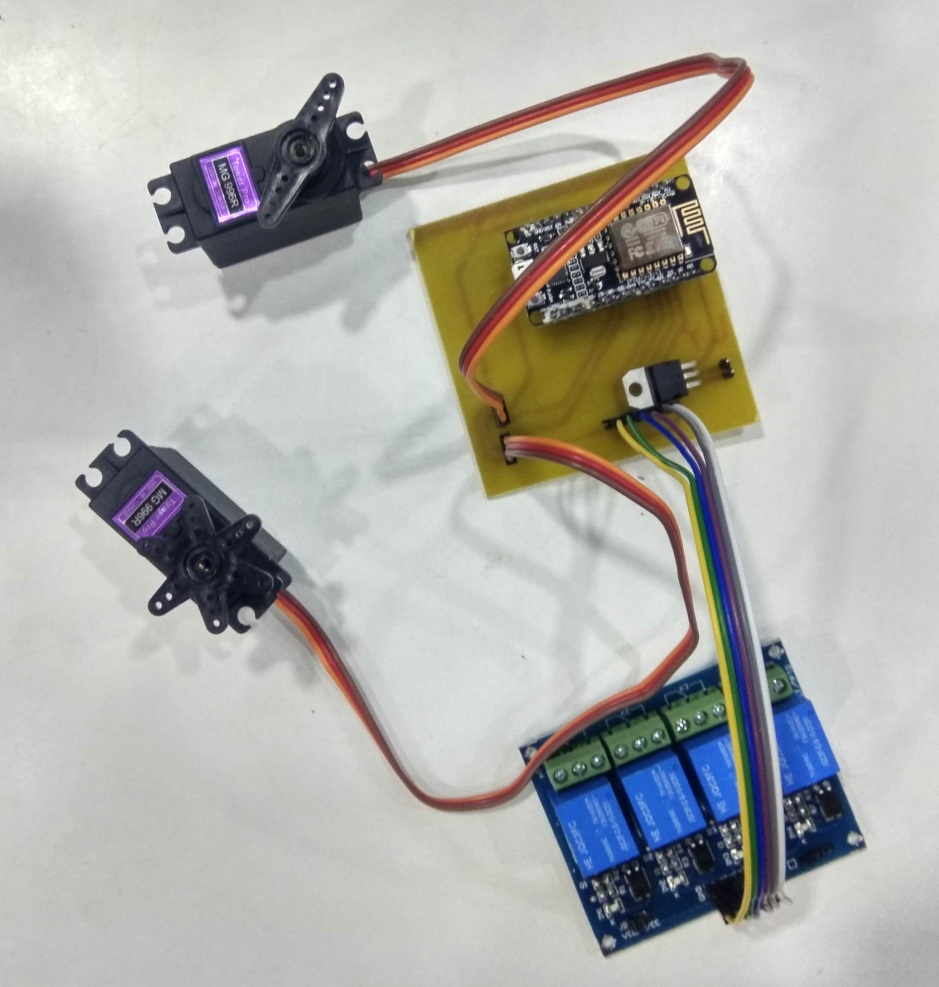
# 4.1 PCB designed and fabricated

****

*Figure 14* *pcb final*

This pcb was quite a challenge for us because the library of Node MCU was not found in the eagle (PCB design software).We made 3-4 times this unit’s pcb to get this result. So we made library of the nodemcu by ourself. It is not accurate as professionals but it can do the work. For designing this pcb we took a cooper clad and then firstly did the holes according to the requirement, Then we used nail paint as a resist to make tracks as you can see in figure. After that the etching is done to remove not required copper part.

# 4.2 After soldering components to pcb



*Figure 15* *pcb after soldering*

# 4.3 working and running of open hab instance

In this intermediate process we have now successfully linked the raspberry pi which is acting as the brain of the system and the nodemcu which is performing the function of automation and switching of the desired application using open hab. All the application such as light, fan and motor has been

connected via a four channel relay  to the nodeMcu and the raspberry Pi which forms a entire automated system.Now when a command such as turning on the fan is given to the raspberry Pi via voice command or the UI that has been created,the command is initialy processed in the raspberry Pi itself and after processing it the request is given from the local instance in the raspbian to the google servers for synthesizing the request and then after processing it on the servers it then forwards the request access to the openhab cloud server after identification of appropriate devices that has been created by the user. It then send this access to the local server which in turn commands the nodeMcu via The MQTT to perform the specified task. The entire process needs a fraction of milliseconds but the delay that we encounter is usually due to the latency in the network provided to the system. A strong reliable network can never create a latency or delay issue in the processing of requests.

# Conclusion

The home automation system using Internet of Things has been experimentally proven to work satisfactorily by integrating applications and appliances to the processor and creating a network of all the things which are connected. The applications were successfully controlled remotely through internet and voice commands. It was noted that as compared to conventional home appliances and its functioning and switching functionality, a simple home automation system using internet of things and sensors can provide a very high value of efficiency and functionality along with great amount of energy saving and conservative use as and when needed. This system is dynamic so that as per user needs multiple number of sensors and detectors can be used and integrated with the brain of the system that is the processor to perform endless number of applications and make user’s experience at home smart, comfortable as well as secure and efficient in every aspect possible. Hence providing an overall highly organized and systematic approach towards saving energy, improving functionality, providing security and enhancing user experience.