MOBILE APPLICATION BASED HOME AUTOMATION SYSTEM USING IoT

A MINI PROJECT 2 REPORT

Submitted by

GOVINDHRAJULU M - E19EER013
KABILAN V- E19EER017
KULOTHUGAN S - E19EER024
MANOJKUMAR S - E19EER027

In partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRICAL AND ELECTRONICS ENGINEERING

E.G.S. PILLAY ENGINEERING COLLEGE (AUTONOMOUS)

NAGAPATTINAM-611002

MAY 2022

BONAFIDE CERTIFICATE

Certified that this project report titled, **"MOBILE APPLICATION BASED HOME AUTOMATION SYSTEM USING IoT"** is the bonafide work of **GOVINDARAJULU M** (Reg. No. E19EER013), **KABILAN V (**Reg. No. E19EER017), **KULOTHUGAN S** (Reg. No. E19EER024) **and MANOJKUMAR S** (Reg. No. E19EER027) who carried out the project work under my supervision.

SIGNATURE

Dr. V.MOHAN M.E., Ph.D.,

HEAD OF THE DEPARTMENT

Professor

Dept. of Electrical and Electronics Engineering

E.G.S. Pillay Engineering College

Nagapattinam-611 002

SIGNATURE

Mr.S.SIVAMANI M.E.,

SUPERVISOR

Assistant Professor

Dept. of Electrical and Electronics Engineering

E.G.S. Pillay Engineering College

Nagapattinam-611 002

ACKNOWLEDGMENT

It is our indeed responsibility to express our heartfelt gratitude to our founder Chairman Late **Chevalier Dr.G.S.PILLAY**. We would like to be obedient for our chairman **Mrs.S.JOTHIMANI AMMAL G.S.PILLAY** a goodhearted, vital of our institution. We would like to thank our honourable Advisor **Chevalier Dr. S.PARAMESVARAN, M.Com., FCCA (London),** Respectable Secretary **Sri S.SENTHIL KUMAR** and our loveable joint Secretary **Sri S.SANKAR GANESH** for having a great dream to serve for the nation by educating the rural community with the help of their institution. We also express our sincere thanks to our Principal **Dr. S.RAMABALAN M.E., Ph.D.,** for providing us with the necessary facilities to carry out this project work.

We are very much honoured and would like to profound our deep grateful to our Head of the dept. **Dr. V.MOHAN, M.E., Ph.D.,** for his expert, sincere and valuable guidance to us. We all are must to express our sincere thanks to our project coordinator **Mr. P.J SURESH BABU M.E.,** for his full support and motivation given to us.

We would like to express our sincere thanks to our project guide **Mr. S.SIVAMANI, M.E.,** for his wide knowledge, serious research attitude and enthusiasm in work deeply impressed and made us to complete this project to the at most satisfaction of our superiors in a stipulated period.

We express our hearty render and thanks to all the staff members and technical assistants of the Dept. of Electrical and Electronics Engineering for their timely assistance. Our humble gratitude and thanks to our parents and friends who have supported and helped us to complete the project work

ABSTRACT

Everyday Home automation is gaining popularity day by day, because of large advantages. One can achieve home automation by simply connecting home appliance electrical devices to the internet or cloud storage. the reason for this surge demand of network enabled home automation is reaching the zenith in recent days for its simplicity and comparable affordability. Platforms based on cloud computing help to connect to the things surroundings everyone so that one can find it easy to access anything and everything at any time and place in a user friendly manner using custom defined portals. Hence, cloud act as a front end to access IOT. Here we are assuming a system which can control devices through wireless based network or cloud based approach. In project we use IOT based home automation system which goal is to develop a home automation system that gives the user complete control over all remotely controllable aspects of his or her home. The automation system will have ability to be controlled from a central host PC, the internet, and also remotely accessed via a packet PC with a windows mobile based application.

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In this IoT project, I have explained how to make Smart Home with Android Mobile App . With this NodeMCU ESP8266 project, you can control 2 home appliances with Android Mobile App. You can also control the relays from Android Mobile App from anywhere in the world. You can control the relay module from the manual switches if there is no internet available. And you don't need any Google Nest or Amazon Echo Dot devices for this voice control home automation project.With this home automation project, you can control & monitor the real-time feedback of the relays in Android Mobile App from anywhere in the world. If the WiFi is available, the NodeMCU will automatically connect with the Wi-Fi.For this project, I have used all the FREE tools. So if you follow all steps, you can easily make this Smart Home System with Android Mobile App Using MIT App Inventor.

FIG:1.1 OPERATION SYSTEM

Nodemcu Esp8266

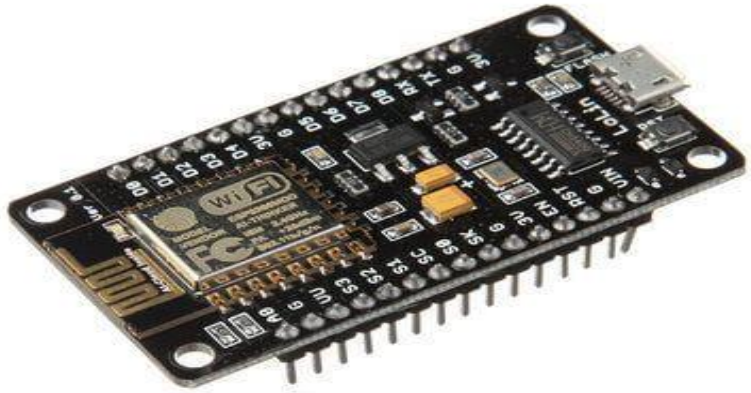

FIG: 1.2 NODEMCU ESP8266

NodeMCU is a low-cost open source IoT platform. It initially included firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which was based on the ESP-12 module. Later, support for the ESP32 32-bit MCU was added. NodeMCU is an open source firmware for which open source prototyping board designs are available. The name "NodeMCU" combines "node" and "MCU" (micro-controller unit). The term "NodeMCU" strictly speaking refers to the firmware rather than the associated development kits Both the firmware and prototyping board designs are open source. The firmware uses the Lua scripting language. The firmware is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson and SPIFFS. Due to resource constraints, users need to select the modules relevant for their project and build

a firmware tailored to their needs. Support for the 32-bit ESP32 has also been implemented. The prototyping hardware typically used is a circuit board functioning as a dual in-line package (DIP) which integrates a USB controller with a smaller surface-mounted board containing the MCU and antenna. The choice of the DIP format allows for easy prototyping on breadboards. The design was initially based on the ESP-12 module of the ESP8266, which is a Wi-Fi SoC integrated with a Tensilica Xtensa LX106 core, widely used in IoT applications .

Nodemcu Pins

NodeMCU provides access to the GPIO (General Purpose Input/Output) and a pin mapping table is part of the API documentation.

I/O index	ESP8266 pin
0 [*]	GPIO16
1	GPIO5
2	GPIO4
3	GPIO0
4	GPIO2
5	GPIO14
6	GPIO12
7	GPIO13
8	GPIO15
9	GPIO3
10	GPIO1
11	GPIO9
12	GPIO10

Channel 5V Relay Module

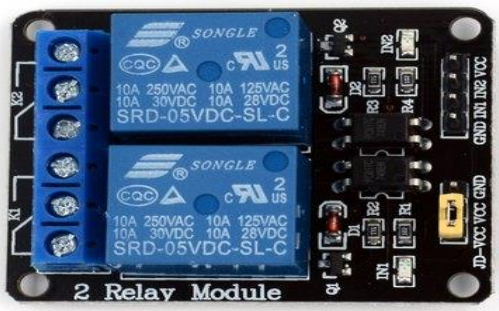

FIG: 1.3-CHANNEL 5V RELAY MODULE

2-Channel 5V Relay Module is a relay interface board, it can be controlled directly by a wide range of microcontrollers such as Arduino, AVR, PIC, ARM and so on. It uses a low level triggered control signal (3.3-5VDC) to control the relay. Triggering the relay operates the normally open or normally closed contacts. It is frequently used in an automatic control circuit. To put it simply, it is an automatic switch to control a high-current circuit with a low-current signal.5V relay signal input voltage range, 0-5V. VCC power to the system. JD-VCC relay in the power supply. JD-VCC and VCC can be a shorted.

The features of 2-Channel Relay module:

- Good for safe control of higher amperage circuits. In power systems, the lower current can control the higher one.

- 2-channel high voltage system output, meeting the needs of dual channel control.
- Brand new and high quality.
- Standard interface that can be controlled directly by microcontroller (Arduino , 8051, AVR, PIC, DSP, ARM)]
- Wide range of controllable voltages.
- Being able to control high load current, which can reach 250V, 10A or 125V, 15A
- With a normally-open (NO) contact and a normally-closed (NC) contact.
- Around the board with 4 mounting holes, easy installation and fixing
- It has a common end, a beginning, a closed-end

Specification of 2-Channel Relay module:

- Relay Module; Model : JQC-3FF-S-Z, 2 Channel
- Voltage to operate: 5V D
- Color : Blue Relays on a black PCB
- Load : 10A, AC 250V/ 15A, 125V

Jumper Wire

FIG:1 4 JUMPER WIRE

A jumper wire (also known as jumper, jumper wire, jumper cable, DuPont wire or cable) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

Power Adapter

FIG: 1.5 POWER ADAPTER

Power Adapter is a source of input supply to the project. It is an essential thing in any project. If we provide you all your project requirement how can miss the power adapter. Here in this category, we stock power Adapter/ DC power supply with differential power output ratings ranging from 5V to 12V dc @ 1A to 5A. You can also find the replacement cable for your charger or Adapter. This adapter is compatible to handle up to 3A current so applications like toy cars, CCTV Cameras, Routers, Modems, Cordless Phones, Set-Top Boxes, Wireless Devices, and POS Machines are compatible with this adapter.

CHAPTER 2

HOME AUTOMATION SYSTEM

2.1 Home Automation Block Diagram

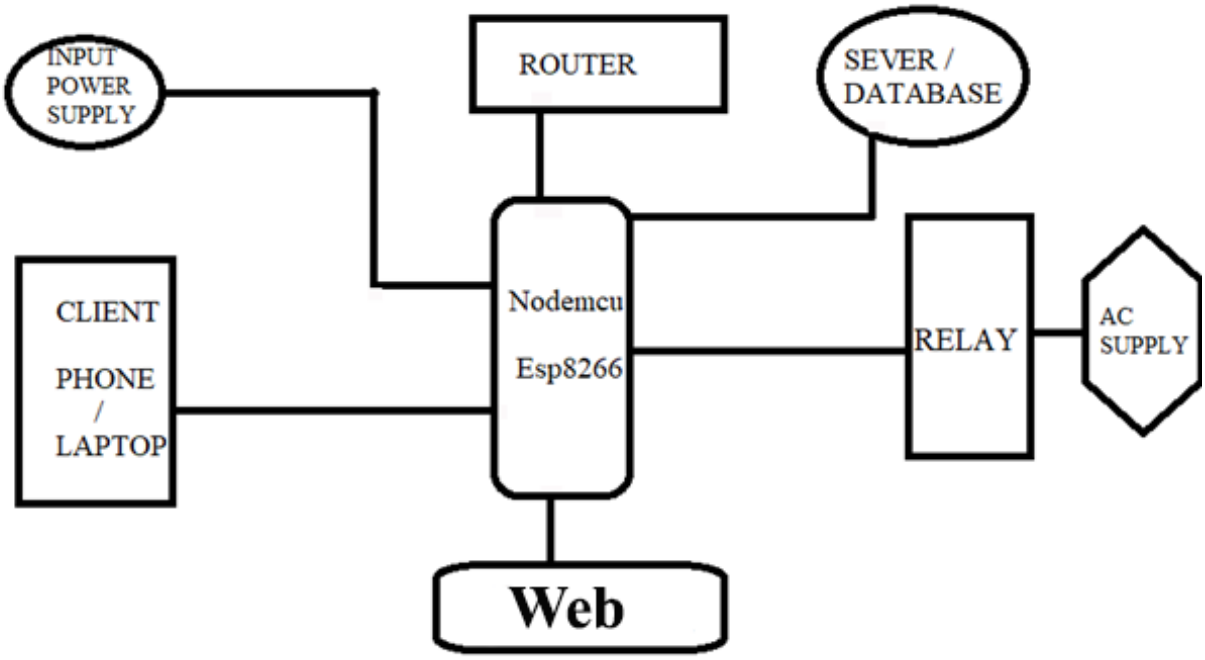

FIG: 2.1 BLOCK DIGRAM

Input Power Supply 5v for Nodemcu and relay – Connected To Router – Router to Connect Sever and Database – Nodemcu Communicate with Sinric pro , Google Home and Amazon Alexa – We use to work Sinric pro , Google Home and Amazon Alexa using Client Phone / laptop using Recommended There Own Website Or App to Communicate Nodemcu – Nodemcu to Operate the Relay to On / Off Light by Client Comands – And Also Operate using Manual switch Relay to On / Off Light by Client Comands.

2.2 Circuit Diagram Home Automation

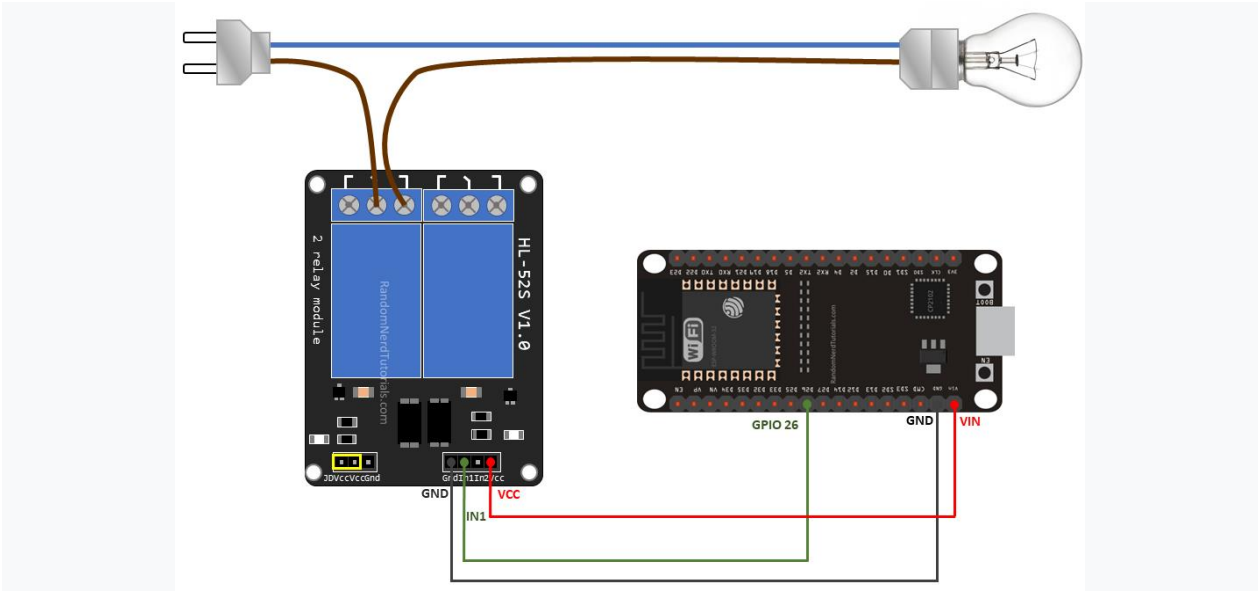

FIG: 2.2 CIRCUIT DIAGRAM

Circuit Connections

- The circuit is very simple, I have used D1,D2 to control the 4-channel relay module.And the IN1,IN2 connected with manual switches to control the relay module manually.
- I have used the INPUT_PULLUP function in Arduino IDE instead of using the pull-up resistors with each switch.
- As per the source code, when the control pins of the relay module receive the LOW signal the respective relay will turn on and the relay will turn off for the HIGH signal in the control pin.
- The Boot will fail if SD3 and D3 are grounded during the Boot process. So manual switch-S1 and switch-S2 must be OFF during NodeMCU Boot.
- Now, if you want to use pushbuttons then just connect the pushbuttons across the GPIO pins and GND pin instead of switches.

CHAPTER 3

PROGRAM FOR DEVELOPER BOARD

In the Your Project, I have explained all the steps to program the NodeMCU using Arduino IDE.

1. Update the **Preferences** –> Aditional boards Manager URLs: https://dl.espressif.com/dl/package_esp32_index.json, http://arduino.esp8266.com/stable/package_esp8266com_index.json
2. Then install the **ESP8266** board from the Board manager or **Click Here** to download the **ESP8266** board.
3. Download the required libraries from the following links:

- **WebSockets** by Markus Sattler (minimum Version 2.3.5)
- ArduinoJson by Benoit Blanchon (minimum Version 6.12.0)

CODE FOR HOME AUTOMATION

```
#include <WiFi.h>

#include <Wire.h>

#include<DHT.h>

#define DHTPIN 14 //dht pin defined

#define DHTTYPE DHT11 //dht type defined
WiFiClient client;
DHT dht(DHTPIN,DHTTYPE);

#define ssid      "Wifi"

#define password  "`12367dh"
```

```
WiFiServer server(80);

IPAddress local_IP(192,168,1,112);
IPAddress gateway(192,168,1,1);
IPAddress subnet(255, 255, 0, 0);



void setup()
{
Serial.begin(115200);
dht.begin();
WiFi.begin(ssid,pass);
Serial.println("Connecting");
while(WiFi.status()!=WL_CONNECTED)
{
Serial.print(".");
delay(100);
}
Serial.println("Connected");
}
    Serial.begin(115200);
    pinMode(14, OUTPUT);      // set the LED pin mode

    delay(10);

    // We start by connecting to a WiFi network
    WiFi.config(local_IP,gateway,subnet);
    Serial.println();
```

```
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);

    WiFi.begin( ssid,password);
//     WiFi.config(local_IP,gateway);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.println("");
    Serial.println("WiFi connected.");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());

    server.begin();

}

int value = 0;

void loop(){
// put your main code here, to run repeatedly:
float temperature = dht.readTemperature();
float humidity = dht.readHumidity();
Serial.println("Temperature:" + String(temperature) + " " +
```

```
"Humidity:" + String(humidity));
client.stop();
if(client.connect(server,80))
 WiFiClient client = server.available();   // listen for incoming
clients


  if (client) {                             // if you get a client,
    Serial.println("New Client.");           // print a message out
the serial port
    String currentLine = "";                // make a String to hold
incoming data from the client
    while (client.connected()) {            // loop while the client's
connected
      if (client.available()) {             // if there's bytes to
read from the client,
        char c = client.read();             // read a byte, then
        Serial.write(c);                    // print it out the serial
monitor
        if (c == '\n') {                    // if the byte is a
newline character


          // if the current line is blank, you got two newline
characters in a row.
          // that's the end of the client HTTP request, so send a
response:
          if (currentLine.length() == 0) {
            // HTTP headers always start with a response code (e.g.
HTTP/1.1 200 OK)
            // and a content-type so the client knows what's coming,
then a blank line:
            client.println("HTTP/1.1 200 OK");
```

```
            client.println("Content-type:text/html");

            client.println();

            // the content of the HTTP response follows the header:

            client.print("Click <a href=\"/H\">here</a> to turn the
LED on pin 14 on.<br>");

            client.print("Click <a href=\"/L\">here</a> to turn the
LED on pin 14 off.<br>");


            // The HTTP response ends with another blank line:

            client.println();

            // break out of the while loop:

            break;

          } else {    // if you got a newline, then clear currentLine:

            currentLine = "";

          }

        } else if (c != '\r') {  // if you got anything else but a
carriage return character,

          currentLine += c;      // add it to the end of the
currentLine

        }


        // Check to see if the client request was "GET /H" or "GET
/L":

        if (currentLine.endsWith("GET /H")) {

          digitalWrite(14, HIGH);               // GET /H turns the
LED on

        }

        if (currentLine.endsWith("GET /L")) {
```

```
          digitalWrite(14, LOW);                // GET /L turns the
LED off}
    }
    // close the connection:
    client.stop();
    Serial.println("Client Disconnected.");
  }
}
```

CHAPTER 4

MIT APP INVENTOR

FIG: 4.1 MIT APP INVENTOR LOGO

MIT App Inventor is a web application integrated development environment originally provided by Google, and now maintained by the Massachusetts Institute of Technology (MIT). It allows newcomers to computer programming to create application software(apps) for two operating systems (OS): Android, and iOS, which, as of 8 July 2019, is in final beta testing. It is free and open-source software released under dual licensing: a Creative Commons Attribution ShareAlike 3.0 Unported license, and an Apache License 2.0 for the source code.

It uses a graphical user interface (GUI) very similar to the programming languages Scratch (programming language) and the StarLogo, which allows users to drag and drop visual objects to create an application that can run on Android devices, while a App-Inventor Companion (The program that allows the app to run and debug on) that works on iOS running devices are still under development. In creating App Inventor, Google drew upon significant prior research in educational computing, and work done within Google on online development environments.

MIT APP INVENTOR DASHBOARD

The Designer lets you create the app's interface

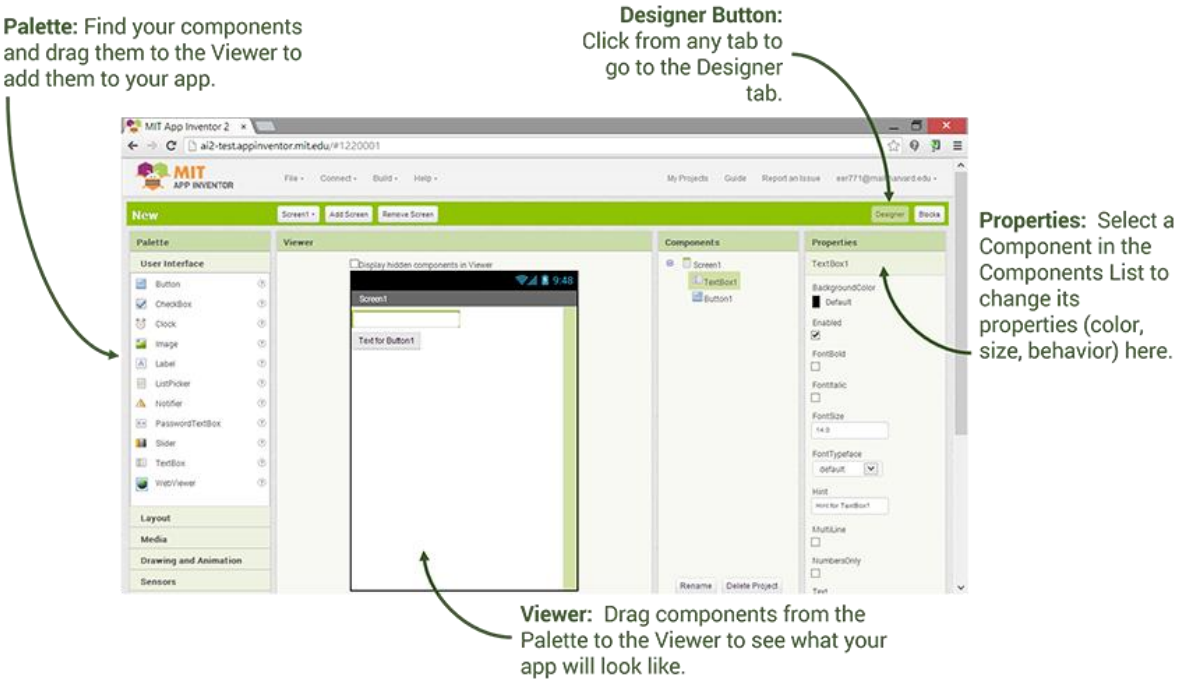

FIG: 4.2 DESIGNER INTERFACE

Blocks Editor lets you program the app's behavior by putting blocks together

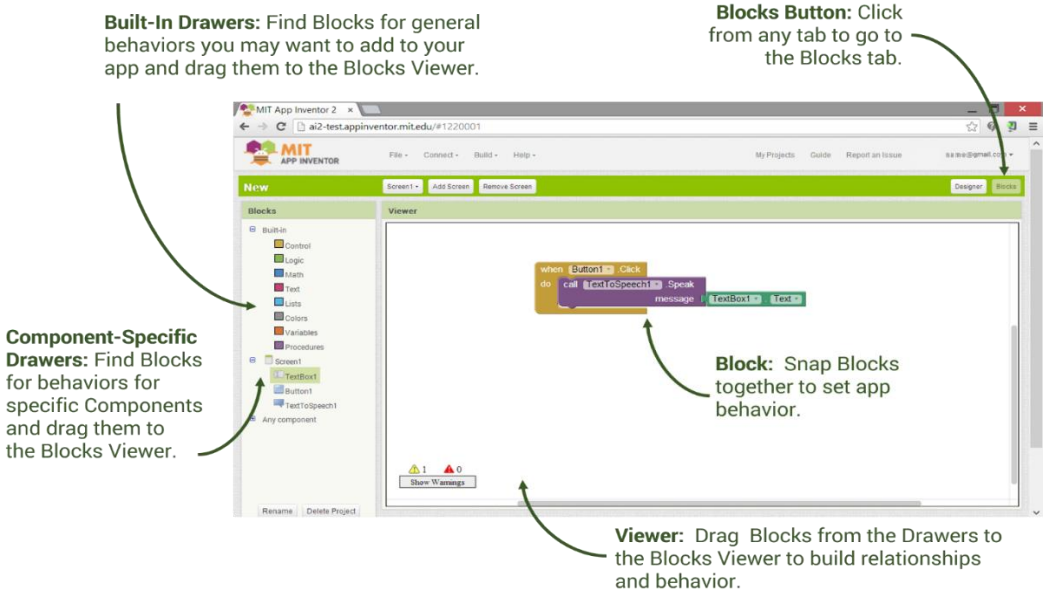

FIG: 4.3 BLOCKS INTERFACE

YOUR OWN CREATED ESPEC HOME AUTOMATION SYSTEM ANDROID MOBILE APP

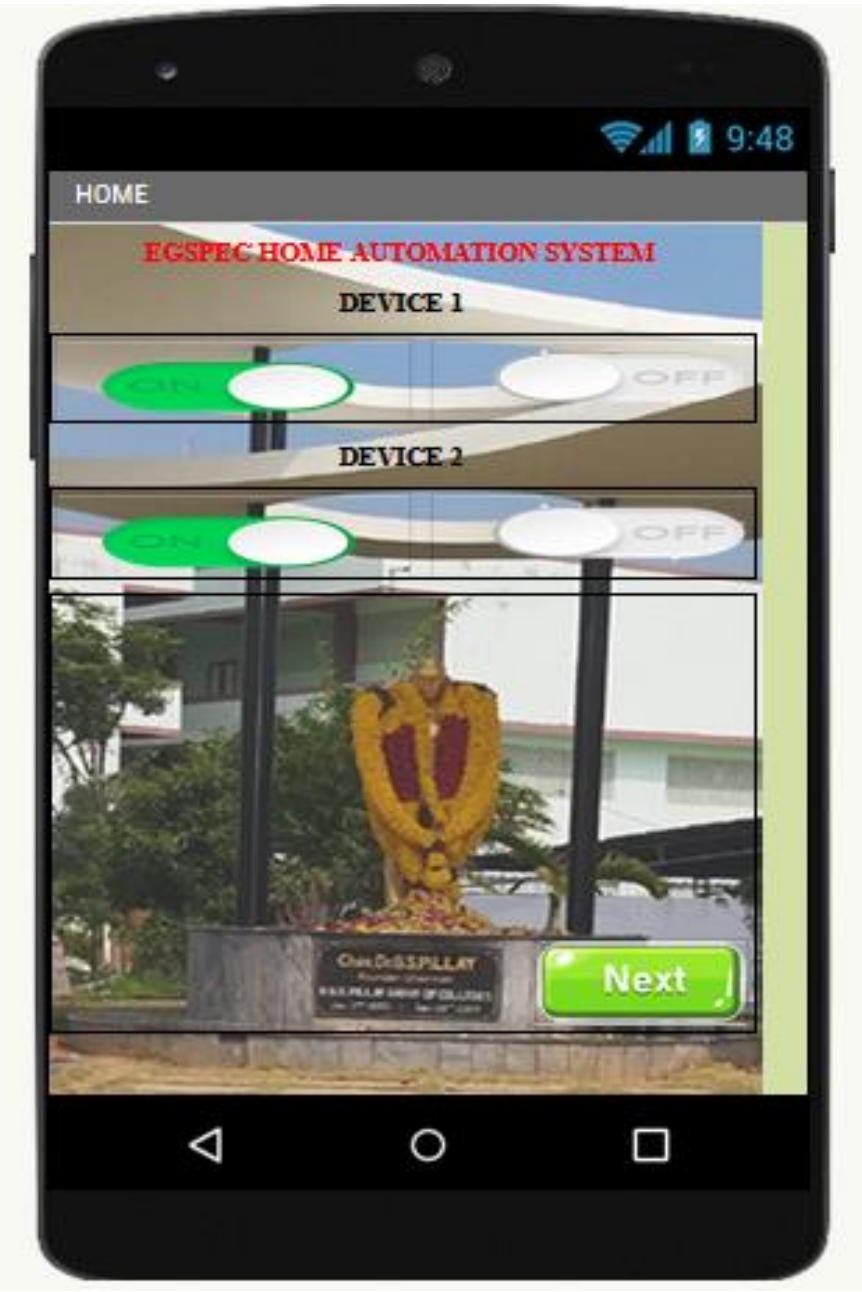

FIG: 4.4 APP INTERFACE

Advantages of Home Automation system

- **Energy-saving**

Home automation manages control elements that contribute to saving water, electricity, and gas. That is, we can program all the devices to turn on or off at the necessary time. Home automation control of lighting and air conditioning controls the management of 70% of energy consumption. Most air conditioners suffer from the same issues causing them to use up additional units. The automation can keep that in check.

- **Security**

Another of its important advantages is being able to detect fires, intruders, gas leaks or a water leak. You can see everything that happens from anywhere through cameras and simulate presence by turning lights on and off remotely.

- **Communication**

It is essential nowadays to establish correct communication between people and housing. New technologies and the Internet are a natural part of home automation and become intuitive and practical tools. Even the recognition of voice or body movements can become a channel of communication with our home. With all these elements, the house can interact with people through the home automation elements of the installation, text messages, emails, and voice calls.

- **Comfort**

The tasks to be carried out in our homes are much easier, and you can do many actions comfortably from a screen.

- **Wellness**

Through home automation, we can automatically close the blinds, detecting the amount of sunlight that enters the rooms or the wind that causes it; control the degree of light in the different rooms, and be able to direct the different environments of the home.

- **Telecare**

The system consists of a set of sensors that monitors the user's life habits, such as the time spent in bed, bath, taking medications. The parameters obtained by these sensors configure a profile that is stored on a central server supervised by healthcare professionals 24 hours a day.

Disadvantages of Home Automation system

- **Initial cost**

The price of the home automation installation is still very high. The initial investment that must be made is very important since the entire home must be wired.

- **Maintenance**

In the event of some type of breakdown, its repair can be complex and expensive. In addition to this, it is possible that an important part of the system will be blocked and more functions will be canceled. Therefore, the cost of any type of breakdown can be very high.

- **Data transmission speed**

Depending on the number of systems that are connected, when transferring a large amount of data, the network can become congested and decrease the transmission speed, causing the functions to slow down.

- **Ring connection**

When the information is connected in the form of a ring, there may be some delay that will also depend on the number of points that are connected to the network, which gives little reliability to the system.

Applications of Home Automation system

Rebuilding consumer expectations, home automation has been projected to target wide array applications for the new digital consumer. Some of the areas where consumers can expect to see home automation led IoT-enabled connectivity are:

- Lighting control
- HVAC
- Lawn/Gardening management
- Smart Home Appliances
- Improved Home safety and security
- Home air quality and water quality monitoring
- Natural Language-based voice assistants
- Better Infotainment delivery
- AI-driven digital experiences
- Smart Switches
- Smart Locks
- Smart Energy Meters

The list is still not exhaustive and will evolve over the time to accommodate new IoT use cases.Now that you are familiar with home automation applications, let’s have a detailed look at what components are involved in building a typical home automation prototype.

CHAPTER 5

Conclusion

A IoT system integrates electrical devices in a house with each other. The techniques which are going to use in home automation include those in building automation as well as the control of domestic activities, such as TV, fan, electric tubes, refrigerator etc. After studying and understanding literature survey and other existing works, we proposed a technique that will gives us better understanding of the Environmental conditions in home. We also provide notification to the user about any error occurs in the devices. In this paper we are planning to eliminate most of the human interaction by providing intelligent system. Development of such system by using Internet of Things technology. By using these system we can actually manage to make low cost, flexible smart homes to adjust its environmental conditions and resolve its errors with energy saving.

CHAPTER 6

References

- Smart Energy Efficient Home Automation System using IOT", by Satyendra K. Vishwakarma, Prashant Upadhyaya, Babita Kumari, Arun Kumar Mishra.

- "IOT Based Smart Security and Home Automation", by Shardha Somani, Parikshit Solunke, Shaunak Oke, Parth Medhi, Prof. P. P. Laturkar.

- A Low Cost Home Automation System Using Wi-Fi based Wireless Sensor Network Incorporating internet of Things", by Vikram.N, Harish.K.S, Nihaal.M.S, Raksha Umesh, Shetty Aashik Ashok Kumar; in 2017 IEEE 7th International Advance Computing Conference.

- Rajeev Piyare "Internet of Things: Ubiquitous Home Control and Monitoring System using Android based Smart Phone" International Journal of Internet of Things 2013 .

- About Access point for web: https://randomnerdtutorials.com/esp8266-nodemcu-access-point-ap-web-server/