

**A Project report on**

**Home Automation using IOT**

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

**Bachelor of Technology**

**In**

**Computer Science & Engineering**

Submitted by

**(Goli Srinath)**

(14H51A0519)

**(G. Sindhuja Sharma)**

(14H51A0521)

**(Himanshu Joshi)**

(14H51A0523)

**(Nishanth Reddy)**

(14H51A0524)

Under the esteemed guidance of

(Mrs. P. Sruthi)

(Assistant Professor)



**Department of Computer Science & Engineering**

**CMR COLLEGE OF ENGINEERING AND TECHNOLOGY**

(An Autonomous Institution under UGC & JNTUH, Approved by AICTE, Permanently Affiliated to JNTUH, Accredited by NAAC with 'A' Grade.)

**2015- 2019**



# **CMR COLLEGE OF ENGINEERING & TECHNOLOGY**

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

## **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



### **CERTIFICATE**

This is to certify that the Project report entitled "**Digit Recognition using MLP Classifier**" being submitted by **Goli Srinath(14H51A0519)**, **G. Sindhuja Sharma (14H51A0521)**, **Himanshu Joshi(14H51A0523)**, **Nishanth(14H51A0524)**, in partial fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering** is a record of bonafide work carried out his/her under my guidance and supervision.

The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

**Mrs. P. Sruthi**  
Assistant Professor  
Dept. of CSE

**Dr. K. Vijaya Kumar**  
Professor and HOD  
Dept. of CSE

---

Submitted for viva voice Examination held on \_\_\_\_\_

**External Examiner**

## ACKNOWLEDGEMENT

With great pleasure I want to take this opportunity to express my heartfelt gratitude to all the people who helped in making this project work a grand success.

I am grateful to **Mrs. P. Sruthi**, Assistant Professor, Department of Computer Science and Engineering for his valuable suggestions and guidance during the execution of this project work.

I would like to thank **Dr. Vijaya Kumar Koppula**, Head of the Department of Computer Science and Engineering, for his moral support throughout the period of my study in CMRCET.

I am highly indebted to **Dr. V. A. Narayana**, Principal CMRCET for giving permission to carry out this project in a successful and fruitful way.

I would like to thank the Teaching & Non- teaching staff of Department of Computer Science and Engineering for their co-operation

Finally I express my sincere thanks to **Mr. Ch. Gopal Reddy**, Secretary, CMR Group of Institutions, for his continuous care. I sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project work.

**Goli Srinath**  
(14H51A0519)

**G. Sindhuja Sharma**  
(14H51A0521)

**Himanshu Joshi**  
(14H51A0523)

**Nishanth Reddy**  
(14H51A0524)

# TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	<b>LIST OF FIGURES</b>	i
	<b>ABSTRACT</b>	ii
<b>1</b>	<b>INTRODUCTION</b>	1
	1.1 Existing Problem	1
	1.2 Main Objective	1
	1.3 Scope	2
	1.4 Internet of Things	
<b>2</b>	<b>BACKGROUND WORK</b>	3
	2.1 Existing Systems	3
	2.2 Android	3
	2.2.1 About Native Code	4
	2.2.2 Creating an Android Project	5
	2.2.3 Creating AVD	6
	2.2.4 Modify Launch Configuration	7
<b>3</b>	<b>PROPOSED SYSTEM</b>	9
	3.1 Block Diagram	9
	3.1.1 IR Sensor	10
	3.2 Controller & Display Block	10
	3.2.1 Micro Controller	10
	3.2.2 LCD Display	11
	3.2.3 ESP-8266 Wi-Fi Module	12
	3.3 Appliances Block	13
	3.3.1 Buzzer	13
	3.3.2 Mobile Application	13
	3.3.3 Bread Board	14
	3.4 Android Coding	15
	3.5 Arduino Code for Sensors	20
<b>4</b>	<b>DESIGNING</b>	24
	4.1 Circuit Diagram	24
	4.2 UML Diagram	25
<b>5</b>	<b>RESULTS AND DISCUSSION</b>	26
<b>6</b>	<b>CONCLUSION AND FUTUREWORK</b>	27
	6.1 Future Scope	27
<b>7</b>	<b>REFERENCES</b>	28

## LIST OF FIGURES

<b>FIGURE NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
3.1	Block Diagram	10
3.2	IR Sensor	11
3.3	Arduino Uno	12
3.4	LCD	12
3.5	ESP-8266 Wi-Fi	13
3.6	Buzzer	14
3.7	Bread Board	15
3.4	LCD	12
3.5	ESP-8266 Wi-Fi	13
3.6	Buzzer	14
3.7	Bread Board	15
4.1	Circuit Diagram	25
4.2	UML Class Diagram	26
5.1	Android App Interface	27

## **ABSTRACT**

This project presents the overall design of Home Automation System (HAS) with low cost and wireless system. This system is designed to assist and provide support in order to fulfill the needs of elderly and disabled in home. Also, the smart home concept in the system improves the standard living at home. . The switch mode and voice mode are used to control the home appliances. The video feedback is received in the android application which streams the video of IP-Camera. The main control system implements wireless technology to provide remote access from smart phone. The design remains the existing electrical switches and provides more safety control on the switches with low voltage activating method. The switches status is synchronized in all the control system whereby every user interface indicates the real time existing switches status. The system intended to control electrical appliances and devices in house with relatively low cost design, user-friendly interface and ease of installation.

# CHAPTER 1

## INTRODUCTION



## **INTRODUCTION**

### **1.1 Existing Problem:**

While people are pursuing ever-growing high quality of their lives today. This leads to more and more facilities and home appliances poured into their buildings. How to control and manage these versatile facilities and appliances in a house. Usually conventional wall switches are located in different corners of a house and, thus necessitate the need of manual operations like pressing to turn the loads on or off. It becomes very difficult for elderly or physically handicapped people to operate them. How to help them?

As the world is trending into new technologies and implementations it is a necessary goal to trend up in home automation also. Most projects signify the use of wireless sensor network collect data from different sensors deployed at various nodes and send it through the wireless protocol. The collected data provide the information about the various environmental factors. Monitoring the environmental factors is not the complete solution to increase the effectiveness. Hence automation must be implemented to overcome these problems. So, in order to provide solution to all such problems, it is necessary to develop an integrated system which will take care of all the devices connected through your android device.

### **1.2 Main Objective:**

Android controlled Smart Home Automation should be able to control the home appliances wirelessly with effectively and efficiently. To develop an application that includes the features of switches mode application. Switch Mode can be used to control the switches of home appliances.

### **1.3 Scope:**

- The system can be used in home, small offices to the big malls
- For remote access of appliances in internet or intranet.
- For the development of technology friendly environment

#### **1.4 Internet of things:**

The **Internet of things (IOT)** is the network of physical devices, vehicles, home appliances and other items embedded with electronics, software, sensors, actuators, and connectivity which enable these objects to connect and exchange data. The term Internet of Things was first coined by Kevin Ashton in 1999 which refers to uniquely identifiable objects and their virtual representation in an Internet like structure. IOT is a network with an infrastructure able to link physical and virtual objects. The link would be simple network between two devices. Each thing is uniquely identifiable through its embedded computing system but is able to inter-operate within the existing Internet infrastructure.

The IOT allows objects to be sensed or controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic benefit in addition to reduced human intervention. When IOT is augmented with sensors and actuators, the technology becomes an instance of the more general class of cyber-physical systems, which also encompasses technologies such as smart grids, virtual power plants, smart homes, intelligent transportation and smart cities.

# CHAPTER 2

## BACKGROUND WORK

## BACKGROUND WORK

### 2.1. Existing Systems

In previous years of automation which was mainly through heavy duty relay controlling of it was a tedious task. Lot of maintenance was required due to lot of wiring. These cause space and made the system bulky. Relays only used as controlling system such as switching ON and OFF but this caused wear and tear of system. Further improvements in industry made modification of control panels which was used for multiple connection and these panel gave the status of working of machines. But these control panels were not mobile and stagnant at place. A person was need to appoint for its status maintenance.

### 2.2 Android

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language.

**Application framework** enabling reuse and replacement of components

**Dalvik virtual machine** optimized for mobile devices

**Integrated browser** based on the open source [WebKit](#) engine

**Optimized graphics** powered by a custom 2D graphics library; 3D graphics based on the OpenGL ES 1.0 specification (hardware acceleration optional)

**SQLite** for structured data storage

**Media support** for common audio, video, and still image formats (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)

**GSM Telephony** (hardware dependent)

**Bluetooth, EDGE, 3G, and WiFi** (hardware dependent)

**Camera, GPS, compass, and accelerometer** (hardware dependent)

**Rich development environment** including a device emulator, tools for debugging, memory and performance profiling, and a plugin for the Eclipse IDE

Developers have full access to the same framework APIs used by the core applications. The application architecture is designed to simplify the reuse of components; any application can publish its capabilities and any other application may then make use of those capabilities (subject to security constraints enforced by the framework). This same mechanism allows components to be replaced by the user.

Android includes a set of C/C++ libraries used by various components of the Android system. These capabilities are exposed to developers through the Android application framework. Some of the core libraries are listed below:

**System C library** - a BSD-derived implementation of the standard C system library (libc), tuned for embedded Linux-based devices

**Media Libraries** - based on PacketVideo's OpenCORE; the libraries support playback and recording of many popular audio and video formats, as well as static image files, including MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG

**Surface Manager** - manages access to the display subsystem and seamlessly composites 2D and 3D graphic layers from multiple applications

**LibWebCore** - a modern web browser engine which powers both the Android browser and an embeddable web view

**SGL** - the underlying 2D graphics engine

**3D libraries** - an implementation based on OpenGL ES 1.0 APIs; the libraries use either hardware 3D acceleration (where available) or the included, highly optimized 3D software rasterizer

**Free Type** - bitmap and vector font rendering

**SQLite** - a powerful and lightweight relational database engine available to all applications

Android includes a set of core libraries that provides most of the functionality available in the core libraries of the Java programming language.

Every Android application runs in its own process, with its own instance of the Dalvik virtual machine. Dalvik has been written so that a device can run multiple VMs efficiently. The Dalvik VM executes files in the Dalvik Executable format which is optimized for minimal memory footprint. The VM is register-based, and runs classes compiled by a Java language compiler that have been transformed into the .dex format by the included "dx" tool.

The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management.

An export your project into a signed APK, which can be distributed to users.

To begin developing Android applications in the Eclipse IDE with ADT, you first need to download the Eclipse IDE and then download and install the ADT plugin. To do so, follow the steps given in [Installing the ADT Plugin](#).

## **Developing in eclipse with ADT:**

The Android Development Tools (ADT) plugin for Eclipse adds powerful extensions to the Eclipse integrated development environment. It allows you to create and debug Android applications easier and faster. If you use Eclipse, the ADT plugin gives you an incredible boost in developing Android applications:

It gives you access to other Android development tools from inside the Eclipse IDE. For example, ADT lets you access the many capabilities of the DDMS tool: take screenshots, manage port-forwarding, set breakpoints, and view thread and process informationd directly from Eclipse.

It provides a New Project Wizard, which helps you quickly create and set up all of the basic files you'll need for a new Android application.

It automates and simplifies the process of building your Android application.

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language.

The Android SDK includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator (based on QEMU), documentation, sample code, and tutorials. Currently supported development platforms include x86-architecture computers running Linux (any modern desktop Linux distribution), Mac OS X 10.4.8 or later, Windows XP or Vista. The officially supported integrated development environment (IDE) is Eclipse (3.2 or later) using the Android Development Tools (ADT) Plug in, though developers may use any text editor to edit Java and XML files then use command line tools to create, build and debug Android application

### **2.2.1 About Native code:**

Libraries written in C and other languages can be compiled to ARM native code and installed, but the Native Development Kit is not yet officially supported by Google. Native classes can be called from Java code running under the Dalvik VM using the System. Load Library call, which is part of the standard Android Java classes.

### **2.2.2 Creating an android project**

*The ADT plug-in provides a New Project Wizard that you can use to quickly create a new Android project (or a project from existing code). To create a new project:*

- *Select File > New > Project.*
- *Select Android > Android Project, and click Next.*

- *Enter a Project Name. This will be the name of the folder where your project is created.*

*Under Contents, select Create new project in workspace. Select your project workspace location*

- *Under Target, select an Android target to be used as the project's Build Target. The Build Target specifies which Android platform you'd like your application built against.*
- *Unless you know that you'll be using new APIs introduced in the latest SDK, you should select a target with the lowest platform version possible, such as Android 1.1.*
- *Under Properties, fill in all necessary fields :*

*Enter an Application name. This is the human-readable title for your application — the name that will appear on the Android device.*

- Enter a Package name. This is the package namespace (following the same rules as for packages in the Java programming language) where all your source code will reside.*
- Select Create Activity (optional, of course, but common) and enter a name for your main Activity class.*
- Enter a Min SDK Version. This is an integer that indicates the minimum API Level required to properly run your application. Entering this here automatically sets the min Sdk Version attribute in the <uses-sdk> of your Android Manifest file. If you're unsure of the appropriate API Level to use, copy the API Level listed for the Build Target you selected in the Target tab.*
- Click Finish.*

Once you complete the New Project Wizard, ADT creates the following folders and files in your new project:

`src/`

Includes your stub Activity Java file. All other Java files for your application go here.



`<Android Version>/` (e.g., Android 1.1/)

Includes the `android.jar` file that your application will build against. This is determined by the build target that you have chosen in the *New Project Wizard*.

`gen/`

This contains the Java files generated by ADT, such as your `R.java` file and interfaces created from AIDL files.

`assets/`

This is empty. You can use it to store raw asset files. See [Resources and Assets](#).

`res/`

A folder for your application resources, such as drawable files, layout files, string values, etc. See [Resources and Assets](#).

`AndroidManifest.xml`

The Android Manifest for your project. See [The AndroidManifest.xml File](#).

`default.properties`

This file contains project settings, such as the build target. This file is integral to the project, as such, it should be maintained in a Source Revision Control system. It should never be edited manually — to edit project properties, right-click the project folder and select "Properties".

### **2.2.3 To create an AVD with the AVD manager:**

- *Select Window > Android SDK and AVD Manager, or click the Android SDK and AVD Manager icon (a black device) in the Eclipse toolbar.*
- *In the Virtual Devices panel, you'll see a list of existing AVDs. Click New to create a new AVD.*
- *Fill in the details for the AVD.*
- *Give it a name, a platform target, an SD card image (optional), and a skin (HVGA is default).*

*Click Create AVD.*

*When you first run a project as an Android Application, ADT will automatically create a run configuration. The default run configuration will launch the default project Activity and use automatic target mode for device selection (with no preferred AVD).*

## **2.2.4 To Create or Modify a Launch Configuration**

*Follow these steps as appropriate for your Eclipse version:*

- *Open the run configuration manager.*
- *In Eclipse 3.3 ,select **Run > Open Run Dialog** (or Open Debug Dialog)*
- *In Eclipse 3.4 (Ganymede), select **Run > Run Configurations** (or Debug Configurations)*
- *Expand the Android Application item and create a new configuration or open an existing one.*

# CHAPTER 3

## PROPOSED SYSTEM

## PROPOSED SYSTEM

### 3.1 Block Diagram

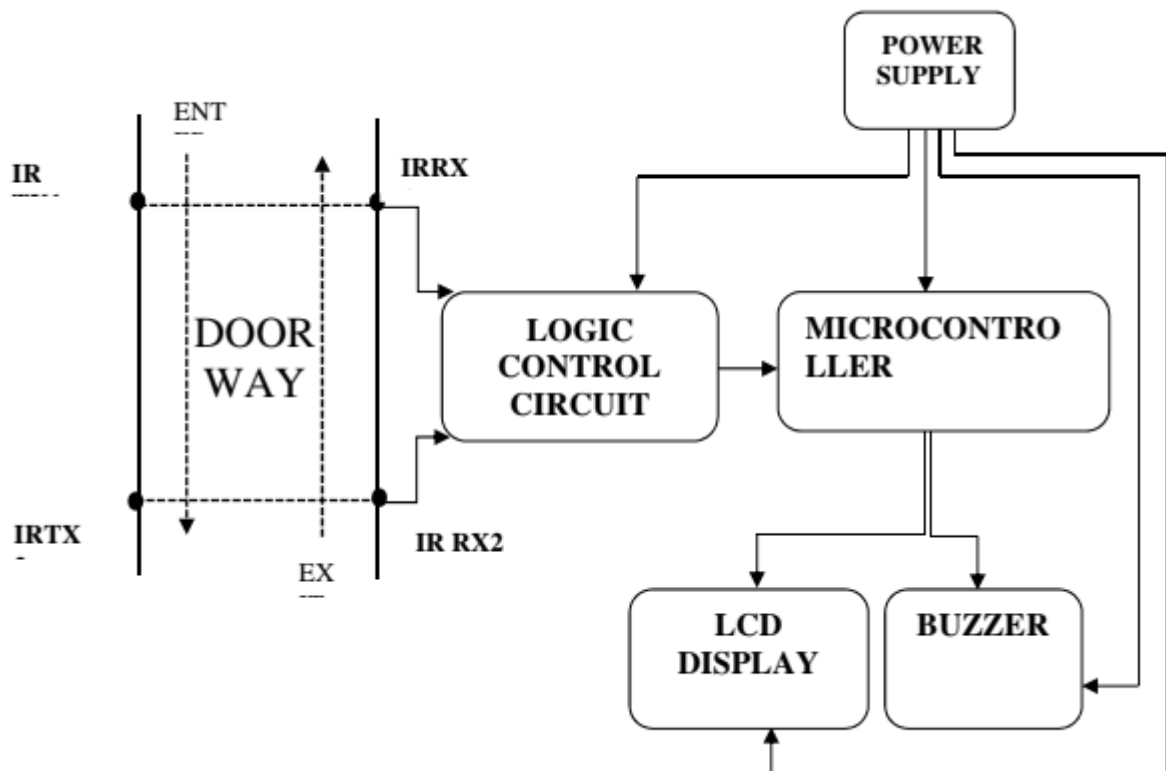


Figure 3.1: Block Diagram

This block diagram can be separated into three blocks,

- Sensor block
- Controller and Display block
- Appliances block

#### Sensor Block

The sensor block consists of IR sensor. IR sensor is used to count the number of persons in the room.

### 3.1.1 IR Sensor



Figure 3.2: Infrared Sensor

An infrared sensor is an electronic device that emits infrared radiation in order to sense some aspects of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. The radiations which are invisible to our eyes can also be detected by an infrared sensor. The emitter is simply an IR LED (Light Emitting Diode) as shown in the Fig:2 and the detector is simply an IR photodiode which is sensitive to IR light of the same wavelength as that emitted by the IR LED.

## 3.2 Controller and Display Block

The Arduino board receives the messages from sensor block and controls the appliances in the room. Arduino switches ON the appliances based on number of persons in the room. LCD display is used to display the visitors count.

### 3.2.1 Micro Controller

The microcontroller used for this project is Arduino uno.

#### Arduino Uno

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller operations. It is connected to a computer with a help of USB cable. The Uno varies from other boards because it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board. The Arduino does

not need a separate piece of hardware (called a programmer) in order to load new code onto the board – you can simply use a USB cable.

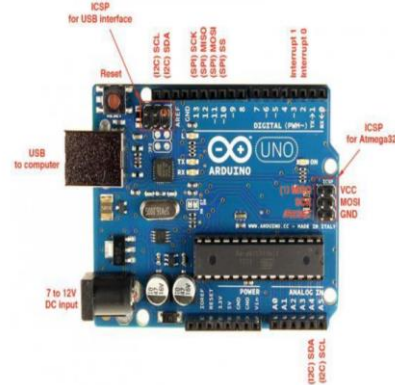


Figure 3.3: Arduino Uno

### 3.2.2 LCD Display

The LCD screen is an electronic display module with a wide range of applications. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data. The command register stores the command instructions given to the LCD.

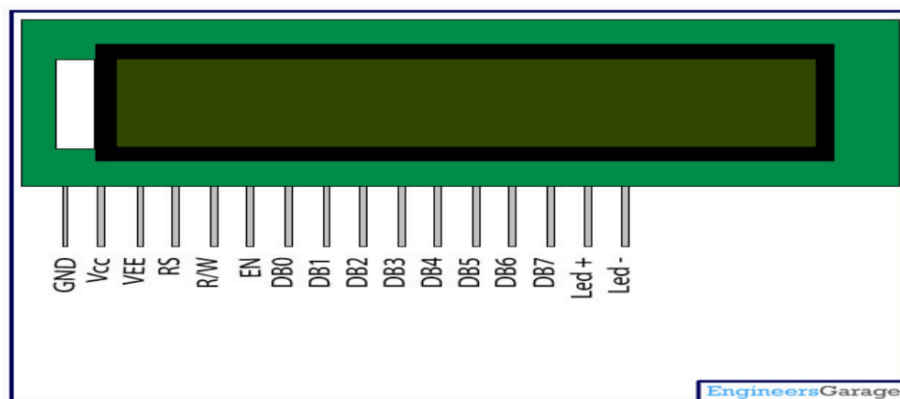


Figure 3.4: LCD

A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The LCD screen displays the present the exact number of visitors in a building, and operates in 4-bit 'nibble' mode to save I/O pins. For this project, the LCD is connected directly to Port B of the microcontroller. The LCD screen will display “Capacity Full” when the microcontroller detects that the required number of visitors

supposed to occupy a particular building is exceeded .

The connections which are done for LCD are given below:

- PIN1 or VSS to ground.
- PIN2 or VDD or VCC to +5v power.
- PIN3 or VEE to ground (gives maximum contrast best for a beginner).
- PIN4 or RS (Register Selection) to PIN12 of ARDUINO UNO.
- PIN5 or RW (Read/Write) to ground(puts LCD in read mode eases the communication for user).
- PIN6 or E (Enable) to PIN10 of ARDUINO UNO.
- PIN11 or D4 to PIN5 of ARDUINO UNO.
- PIN12 or D5 to PIN4 of ARDUINO UNO.
- PIN13 or D6 to PIN3 of ARDUINO UNO.
- PIN14 or D7 to PIN2 of ARDUINO UNO.
- PIN15 to +5V with 221 ohm resistor in series.
- And the last PIN16 to ground.

### **3.2.3 ESP 8266 Wi-Fi Module**

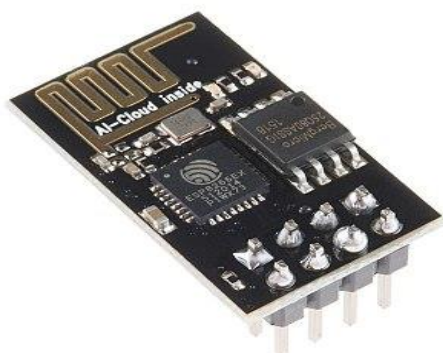


Figure 3.5: ESP-8266 Wi-Fi Module

The ESP8266 is a System on a Chip (SoC), manufactured by the Chinese company Espressif. It consists of a Tensilica L106 32-bit micro controller unit (MCU) and a Wi-Fi transceiver. It has 11 GPIO pins (General Purpose Input/Output pins), and an analog input as well. This means that you can program it like any normal Arduino or other microcontroller. And on top of that, you get Wi-Fi communication, so you can use it to connect to your Wi-Fi network, connect to the Internet, host a web server with real web pages, let your smartphone connect to it, etc., this chip has become the most popular IOT device available.

### 3.3 Appliances block

#### 3.3.1 Buzzer

The piezo buzzer is a device or a transducer which converts electrical energy to sound energy. When electrical energy is applied to it, it buzzes or sounds until the energy applied to it is stopped. The buzzer used in this project alert user of the event corresponding to capacity changes by producing a noisy sound irrespective of the voltage variation applied to it. The buzzer has two terminals (positive and ground).



Figure 3.5: Buzzer

#### 3.3.2 Mobile Application

The output of the reading are also displayed in application developed for mobile and PC which are displayed in the form of graph. This application gets the data through wifi-module transmitting the data through internet.

#### 3.3.3 Bread Board

A breadboard is a solder less device for temporary prototype with electronics and test circuit designs. Most electronic components in electronic circuits can be interconnected by inserting their leads or terminals into the holes and then making connections through wires where appropriate.

The breadboard has strips of metal underneath the board and connect the holes on the top of the board. The metal strips are laid out as shown below. Note that the top and bottom rows of holes are connected horizontally and split in the middle while the remaining holes are connected vertically.



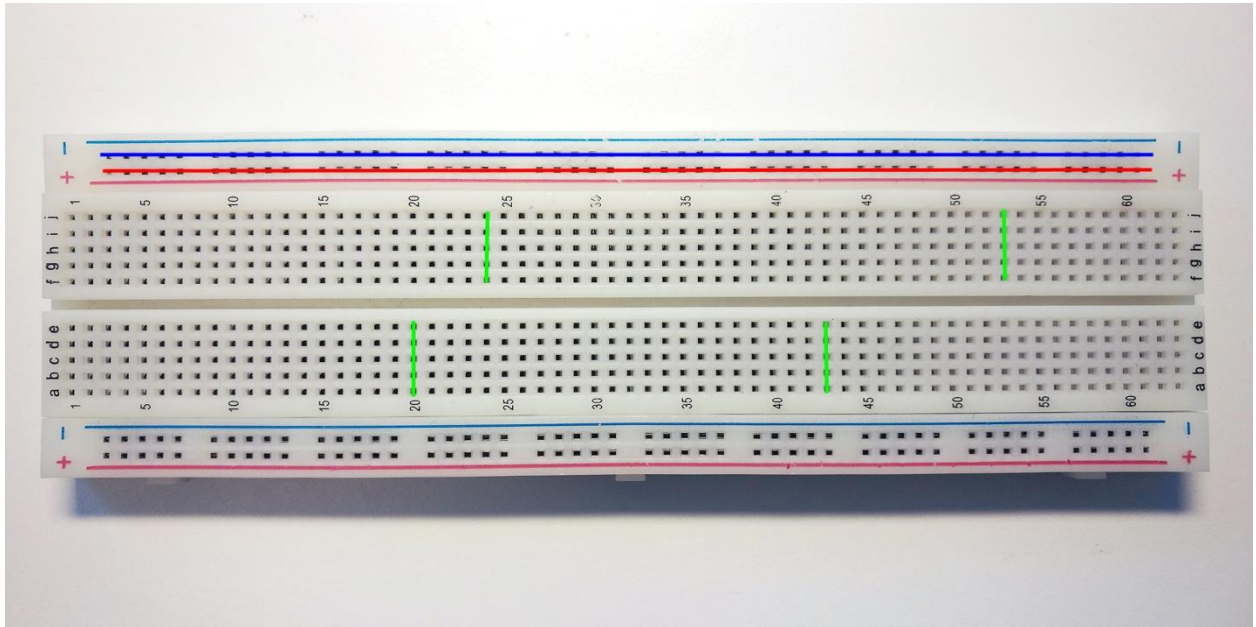


Figure 3.7: Bread Board

### 3.4 ANDROID CODING

#### *MAIN ACTIVITY.JAVA*

```
package com.example.iotvisitorcounters;

import android.os.Bundle;
import android.os.Handler;
import android.app.Activity;
import android.app.DownloadManager.Request;
import android.content.Intent;
import android.view.Menu;
import android.view.Window;
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_main);
    new Handler().postDelayed(new Runnable() {

        @Override
        public void run() {
            // TODO Auto-generated method stub
            Intent it =new Intent(MainActivity.this,HomeActivity.class);
            startActivity(it);
            finish();
        }
    }, 3000);
}

}
```

#### 3.3.4.2 HOMEACTIVITY.JAVA

```
package com.example.homeautomation;

import java.io.BufferedOutputStream;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.net.URLConnection;
import java.net.URL;

import android.annotation.SuppressWarnings;
import android.app.Activity;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.StrictMode;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Toast;
```

```
@SuppressWarnings("NewApi")
public class HomeActivity extends Activity {
    String aal;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (android.os.Build.VERSION.SDK_INT > 9) {

            StrictMode.ThreadPolicy policy = new
            StrictMode.ThreadPolicy.Builder().permitAll().build();
            StrictMode.setThreadPolicy(policy);
        }

        setContentView(R.layout.activity_home);
        ImageButton
        bt=(ImageButton) findViewById(R.id.imageButton1);
        ImageButton
        bt2=(ImageButton) findViewById(R.id.imageButton2);
        // Button bt3=(Button) findViewById(R.id.button3);

        bt.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View arg0) {
                // TODO Auto-generated method stub
                new Lichterkette2().execute();
                Toast.makeText(getApplicationContext(),
                "posted",100).show();
            }

        });

        bt2.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View arg0) {
                // TODO Auto-generated method stub
                new Lichterkette().execute();
                Toast.makeText(getApplicationContext(),
                "posted",100).show();
            }

        });
    }
}
```

```
class Lichterkette extends AsyncTask<String,Void,String>{
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
    }

    @Override
    protected String doInBackground(String... params) {
        StringBuilder sb=null;
        BufferedReader reader=null;

String serverResponse=null;
        try {

            URL url = new
URL("https://api.thingspeak.com/update?api_key=GRF6C4K6UQB9UNK
U&field1=1");
            HttpURLConnection connection = (HttpURLConnection)
url.openConnection();

            connection.setRequestProperty("Content-Type", "application/x-
www-form-urlencoded");
            connection.setConnectTimeout(5000);
            connection.setRequestMethod("POST");
            connection.connect();
            int statusCode = connection.getResponseCode();

//Log.e("statusCode", "" + statusCode);
            if (statusCode == 200) {
                sb = new StringBuilder();
                reader = new BufferedReader(new
InputStreamReader(connection.getInputStream()));
                String line;
                while ((line = reader.readLine()) != null) {
                    sb.append(line + "\n");
                }
            }

            connection.disconnect();
            if (sb!=null)
                serverResponse=sb.toString();
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            if (reader != null)

```

```
e.printStackTrace();
    }
}
    }
    return serverResponse;
}
```

```
@Override
    protected void onPostExecute(String s) {
```

```
    super.onPostExecute(s);
    //All your UI operation can be performed here
    System.out.println(s);
    }
}
```

```
class Lichterkette2 extends AsyncTask<String,Void,String>{
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
    }
}
```

```
@Override
    protected String doInBackground(String... params) {
        StringBuilder sb=null;
        BufferedReader reader=null;
        String serverResponse=null;
        try {

            URL url = new
            URL("https://api.thingspeak.com/update?api_key=GRF6C4K6UQB9UNK
            U&field1=0");
            URLConnection connection = (URLConnection)
            url.openConnection();
            connection.setRequestProperty("Content-Type",
            "application/x-www-form-urlencoded");
            connection.setConnectTimeout(5000)

        }
    }
}
```

```
        return serverResponse;
    }

    @Override
    protected void onPostExecute(String s) {
        super.onPostExecute(s);
        //All your UI operation can be performed here
        System.out.println(s);
    }
}
```

### **GRAPHACTIVITY.JAVA**

```
package com.example.iotvisitorcounters;
```

```
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.webkit.WebView;
```

```
public class GraphicActivity extends Activity {
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_graphic);
        String html = "<iframe width=\"450\" height=\"250\" style=\"border: 1px solid
#cccccc;\"
src=\"http://thingspeak.com/channels/503672/charts/1?api_key=REYNO4FKZL3TDJXK&d
ynamic=true\"></iframe>";
        WebView webview = (WebView) findViewById(R.id.webView1);
        webview.getSettings().setJavaScriptEnabled(true);
        webview.setInitialScale(210);

        return true;
    }
}
```

### 3.5 Arduino Code For Sensors, Relay And LED:

```
int relay = 12;
int led=11;
const char* ssid = "Cmrcet-IOT10";
const char* password = "12345678";

char res[130];

char check(char* ex, int timeout)
{
    int i = 0;
    int j = 0, k = 0;
    while (1)
    {
sl:
        if (Serial.available() > 0) //serial data present
        {
            res[i] = Serial.read();//reading into res[i]//0
            if (res[i] == 0x0a || res[i] == '>' || i == 100) //res[i]=enter//res[i]='>'//i=100
            {
                i++;
                res[i] = 0; break; //enters into loop when enter,> and i=100
            }
            i++;
        }

j++;
        if (j == 30000)
        {
            k++;
            //Serial.println("kk");
            j = 0;
        }
        if (k > timeout)

    {
        // Serial.println("timeout");
    }
}
```

```
//while 1
if (!strcmp(ex, res, strlen(ex)))

{
    //Serial.println("ok.");
    return 0;
}
else
{
    // Serial.print("Wrong ");
    // Serial.println(res);
    i = 0;
    goto sl;
}
}
void serialFlush() {
    while (Serial.available() > 0) {
        char t = Serial.read();
    }
}

char buff[200];
int  sen1, sen2;

void setup() {

char ret;

    Serial.begin(115200); //baudrate
    //Serial.begin(9600);
    pinMode(led,OUTPUT);
    pinMode(relay,OUTPUT);
    digitalWrite(relay,LOW);

    delay(3000);
    serialFlush();
```



```
st:
  Serial.println("ATE0"); // disable the echo of AT
  ret = check((char*)"OK", 50);
  Serial.println("AT");
  ret = check((char*)"OK", 50);

  if (ret != 0) {

delay(100);
  goto st;
}

  Serial.println("AT+CWMODE = 1"); // client and access point

ret = check((char*)"OK", 50);

connectagain:

  serialFlush();
  Serial.print("AT+CWJAP=\""); // connect to a particular access point specified by ssid and
  pswd
  Serial.print(ssid);
  Serial.print "\",\"");
  Serial.print(password);
  Serial.println("\"");

  if (check((char*)"OK", 300)) {
    goto connectagain;
  }

  Serial.println("AT+CIPMUX=1"); // 4 connections at a time to 4 diff channels; = 0 for single
  connection
}
int cl = 0;

void loop() {
  cl++;

  if (cl > 4)
  {
```

```
sen1 = analogRead(A0)/2;
//sen2 = analogRead(A1);
serialFlush();
Serial.println("AT+CIPSTART=4,\"TCP\", \"184.106.153.149\",80"); // connectoin id = 4;
type = TCP (wireless UDP); thinkspeak ip address; port no for tcp/ip
if (!check((char*)"4,CONNECT", 200))
{

    serialFlush();
    Serial.println("AT+CIPSEND=4,76"); // 76 is the length of data
    if (!check((char*)">", 50))
    {

        delay(1000);
        serialFlush();
        Serial.print("GET /update?api_key=GRF6C4K6UQB9UNKU&");
        sprintf(buff, "field1=%04u", sen1);
        Serial.print(buff);

        if(sen1>=47)
        {

            digitalWrite(led,HIGH);
            digitalWrite(relay,HIGH);
        }

    else
    {
        digitalWrite(led,LOW);
        digitalWrite(relay,LOW);
    }
    // sprintf(buff, "&field2=%04u", sen2);
    // Serial.print(buff);
    /*sprintf(buff, "&field3=%s", buf3);
    Serial.println(buff);*/
    Serial.println(""); //no choice
```

# CHAPTER 4

## DESIGNING

## DESIGNING

### 4.1 Circuit Diagram

The circuit works on the principle of IR sensing. Infrared or simply IR Sensors are devices that work with Infrared Light Source and a Photo Detector like a Photo Diode or a Photo Transistor that act as a Transmitter and Receiver respectively.

In this project, we have used an IR LED as the IR Transmitter and a Photo Diode as the IR Receiver. Two sets of IR sensors consisting of an IR LED and Photo Diode are placed at two ends of the entrance of a room.

Output from each sensor is fed to the microcontroller. In normal operation, IR light from the LED would not fall on the Photo Diode as it is a Reflective type IR Sensor. The output from the sensor would be a logic LOW signal in this case.

In case of any interruption (due to any person crossing the path), the Photo Diode would start receiving the IR Light and start conducting. As a result, the output from the sensor would be a logic HIGH signal. The transition from low to high, for each sensor pair is detected by the microcontroller and accordingly the count would be increased or decreased.

The total amount of transition between increase and decrease is noted by Arduino Controller and it sends data to Mobile application through ESP8266 Wi-Fi Module

the Mobile calculates and calibrates the total data of visitor counting of in and out of museum in a particular duration of time and displays it in the form of graph for better understanding.

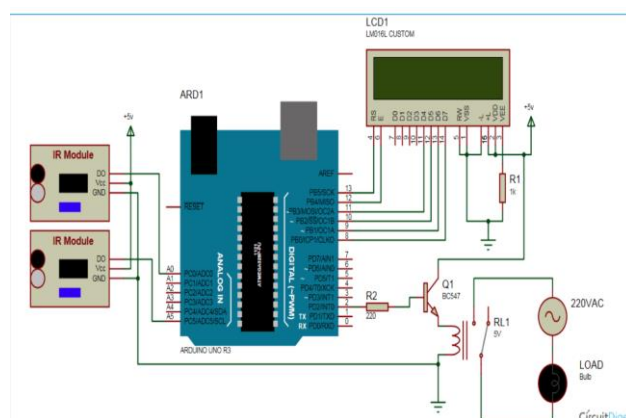


Figure 4.1: Circuit Diagram

## 4.2 UML Diagram:

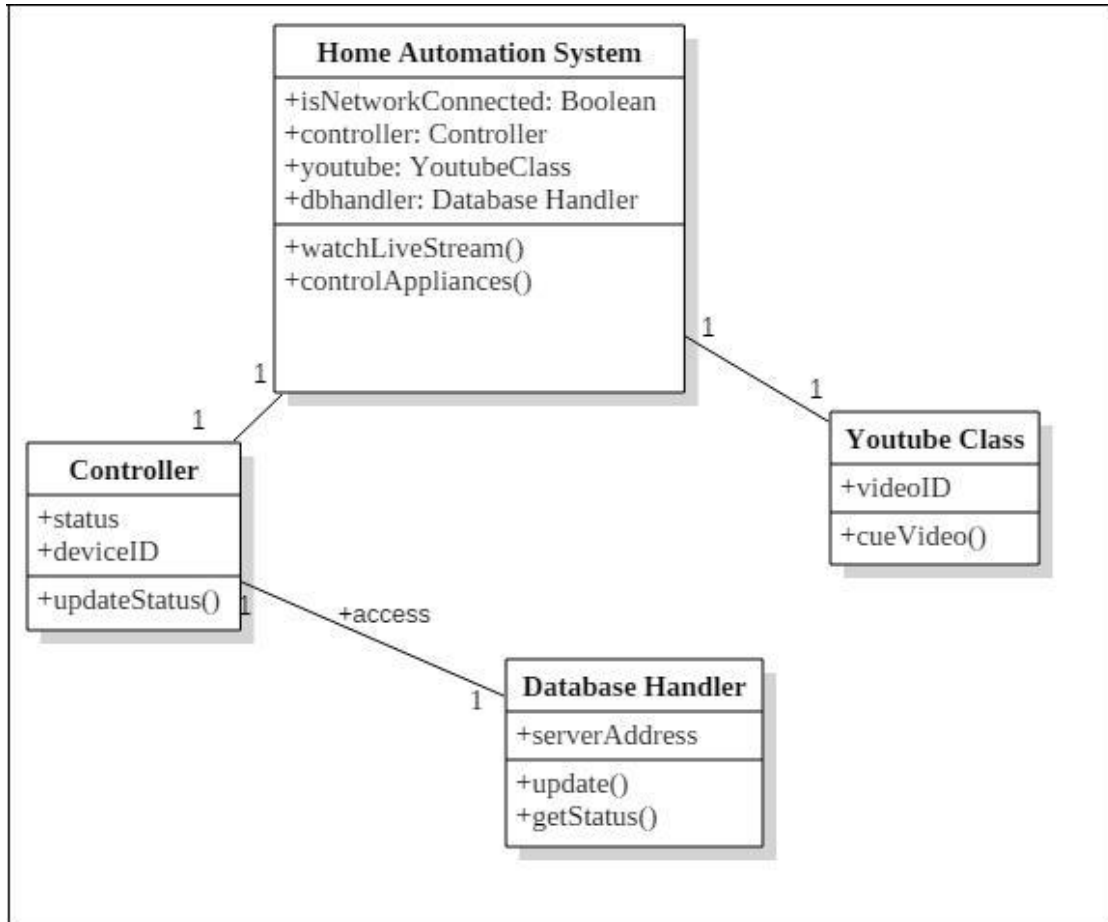


Figure 4.2: UML Class Diagram

# CHAPTER 5

## RESULTS AND DISCUSSION

## RESULTS AND DISCUSSION

The final outcome of this project is a prototype for a simple home automation system which can control home AC appliances with a touch of a finger and is equally capable of monitoring with live stream video feed of the site. On the front end, there is an android app to control the home appliances and PCB board to which a number of home appliances are connected. Then on the backend, there is raspberry which has been programmed to control the status of GPIO pins and thereby controlling the appliances, and also capture video using camera module and stream it live to YouTube server.

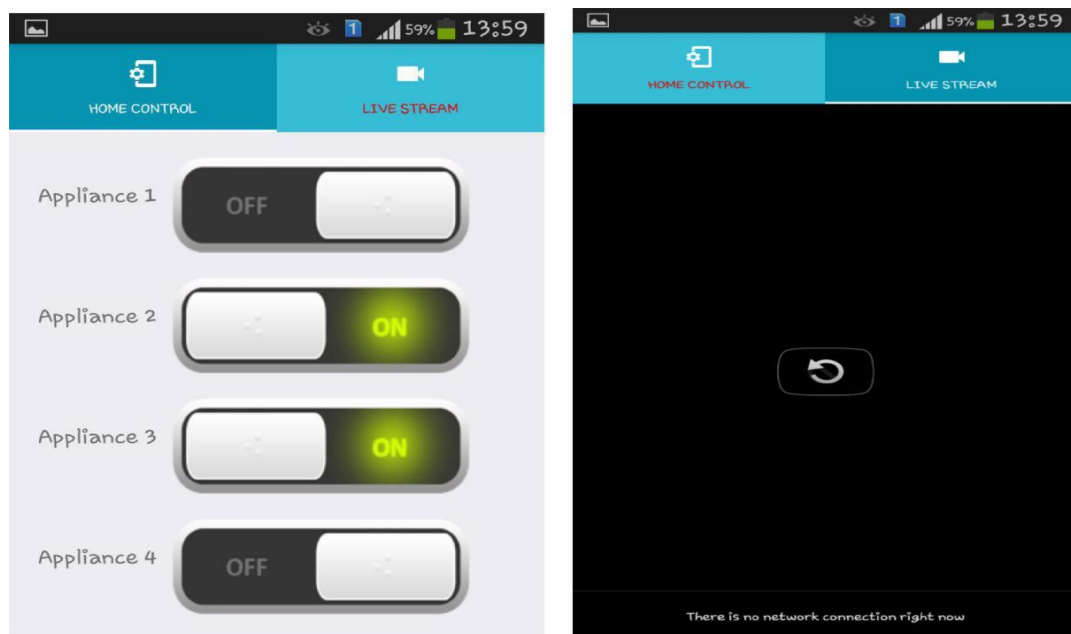


Figure 5.1: Android app interfaces for control (left) and live stream

The results of this project have been up to the mark as expected when the project began initially. The user can remotely control the appliances and also watch the live stream on the same mobile app. Hence, the target we set were successfully reached on time effectively.

# CHAPTER 6

## CONCLUSION AND FUTURE WORK



## **CONCLUSION AND FUTURE WORK**

The prime objective of our project is to use the Smartphone to control the home appliances effectively. The switch mode and voice mode are used to control the home appliances. The video feedback is received in the android app which streams the video of IP- Camera.

This project is based on the Arduino, Android platform Java. These platforms are Free Open Source Software. So the overall implementation cost is low and can be easily configured.

User can easily interact with the android phone/tablet. The user can send commands via the switch mode or speech mode. The data are being analyzed by the application and are sent over a network. The Raspberry pi acts as a server, analyses the data and activates the GPIO (General Purpose Input Output) Pins. The GPIO Pins are connected to the relays switch which activated the required home appliances.

In this way, automation process is carried out. This is a simple prototype. Using this as a reference further it can be expanded to many other programs.

### **6.1 Future Scope:**

Looking at the current situation we can build cross platform system that can be deployed on various platforms like iOS, Windows. Limitation to control only several devices can be removed by extending automation of all other home appliances. Network can be connected to internet and Security cameras can be controlled from other places, allowing the user to observe activity around a house or business. Security systems can include motion sensors that will detect any kind of unauthorized movement and notify the user. Scope of this project can be expanded to many areas by not restricting to only home.

# CHAPTER 6

## REFERENCES

## REFERENCES

- [1] Jain, Sarthak, Anant Vaibhav, and Lovely Goyal, "Raspberry Pi based interactive home automation system through E-mail." Optimization, Reliability, and Information Technology (ICROIT), 2014 International Conference on. IEEE, 2014.
- [2] YAN Wenbo, WANG Quanyu, "Smart Home Implementation Based on Internet and WiFi Technology" GAO Zhenwei School of Computer Science And Technology, Beijing Institute of Technology, Beijing 100081, P. R. China.
- [3] Nitin Sinha, Korrapati Eswari Pujitha, John Sahaya Rani Alex, "Xively Based Sensing and Monitoring System for IOT" 2015 International Conference on Computer Communication and Informatics (ICCCI -2015).
- [4] Assaf, Mansour H., et al. "Sensor based home automation and security system." Instrumentation and Measurement Technology Conference (I2MTC), 2012 IEEE International. IEEE, 2012.
- [5] Kelly, Sean Dieter Tebje, Nagender Kumar Suryadevara, and Subhas Chandra Mukhopadhyay. "Towards the implementation of IOT for environmental condition monitoring in homes." Sensors Journal, IEEE 13.10 (2013): 3846-3853.
- [6] Mr. Pranay P. Gaikwadli Mrs. Jyotsna P. Gabhane Mrs. Snehal S. Golait "A Survey based on Smart Homes System Using Internet-of-Things ", 2015 International Conference On Computation Of Power, Energy, Information And Communication .
- [7] J. JeyaPadmini, K.R.Kashwan, "Effective Power Utilization and Conservation in Smart Homes Using IOT", 2015 International Conference On Computation Of Power, Energy, Information And Communication.
- [8] Ming Wang, Guiqing Zhang, Chenghui Zhang, Jianbin Zhang, Chengdong Li "An IOT-based Appliance Control System for Smart Homes", 2013 Fourth International Conference on Intelligent Control and Information Processing (ICICIP),2013.
- [9] Dongyu Wang, Dixon Lo, Janak Bhimani and Kazunori Sugiura, "AnyControl - IOT based Home Appliances Monitoring and Controlling", 2015 IEEE 39th Annual International Computers, Software & Applications Conference.
- [10] Pavithra.D, Ranjith Balakrishnan, "IOT based Monitoring and Control System for Home Automation", Proceedings of 2015 Global Conference on Communication Technologies(GCCT 2015).