ABSTRACT

Automation is the main application of each and every technology now a day. Cloud automation helps to reduce the manual efforts associated with provisioning and managing computing power. Similarly, IoT automation helps to reduce manual work of actuating appliances or sensing by making these devices internet-enabled.

Home automation refers to the ability to automate manual tasks that we perform at home. It can be realized in many different ways. The home automation system discussed in this paper makes use of IOT and cloud automation both.

By applying home automation to our homes we are turning them to smart homes. These homes use a centralized control system to control the whole house. Things like controlling appliances, security alarm systems, fire alarm systems, temperature control etc. can be achieved from a single tap on your smartphone. Above all it’s a great help for elderly or physically handicapped people at home.

CHAPTER 1

INTRODUCTION

We live in an exciting time where more and more everyday things are becoming smart. Appliances have sensors and can communicate to other things and can provide control to more things.

The Internet of Things, IoT, is in a huge way and people are rapidly inventing new gadgets that enhances lives. The price of microcontrollers with the ability to talk over a network keeps dropping and developers can now tinker and build things inexpensively.

Homes of the 21st century will become more and more self-controlled and automated due to the comfort it provides, especially when employed in a private home. A home automation system is means that allow users to control electric appliances of varying kind.

Many existing, well-established home automation systems are based on wired communication. This does not pose a problem until the system is planned well in advance and installed during the physical construction of the building. But for already existing buildings the implementation cost goes very high.

In contrast, Wireless systems can be of great help for automation systems. With the advancement of wireless technologies such as Wi-Fi, cloud networks in the recent past, wireless systems are used every day and everywhere.

With advancement of Automation technology, life is getting simpler and easier in all aspects. In today's world Automatic systems are being preferred over manual system. With the rapid increase in the number of users of internet over the past decade has made Internet a part and parcel of life, and IoT is the latest and emerging internet technology.

Internet of things is a growing network of everyday object-from industrial machine to consumer goods that can share information and complete tasks while everybody is busy with other activities. Wireless Home Automation system(WHAS) using IoT is a system that uses computers or mobile devices to control basic home functions and features automatically through internet from anywhere around the world.

An automated home is sometimes called a smart home. It is meant to save the electric power and human energy. The home automation system differs from other system by allowing the user to operate the system from anywhere around the world through internet connection.

Home automation is building mechanization for a home, called a smart home. It includes the control and computerization of lighting, warming, (for example, brilliant indoor regulators), ventilation, aerating and cooling (HVAC), and security, and also home machines, for example, washer/dryers, stoves or fridges/coolers. Wi-Fi is regularly utilized for remote checking and control. Home devices, when remotely observed and controlled through the Internet, are an essential constituent of the Internet of Things.

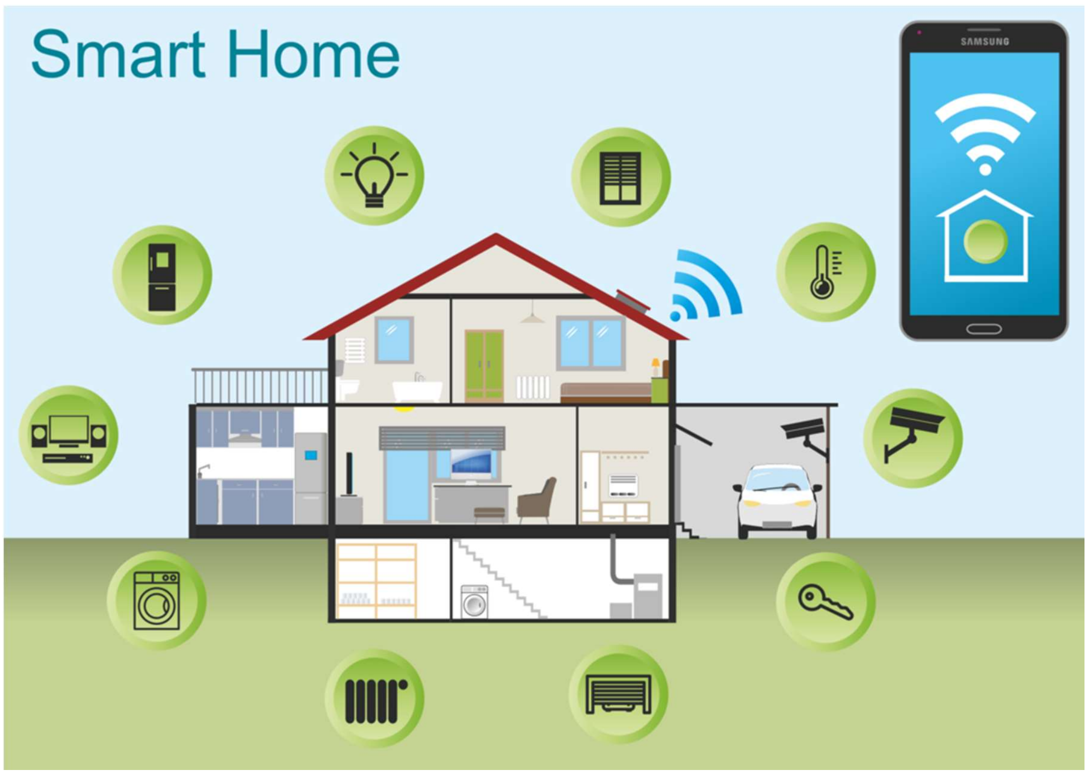


Fig 1.1 Smart Home

1.1 Objective

The objective of this project is to implement a low cost, reliable and scalable home automation system that can be used to remotely control different aspects of the house. The different aspects include controlling appliances, intruder alarm system, fire alarm system, temperature control, garage shutter control etc.

To realize these objectives, we aim to design this system using Arduino boards, Wi-Fi modules and a cloud storage.

To interact with this system remotely we provide android app for smartphone users and a website for other platform users.

CHAPTER 2

DESCRIPTION OF PROJECT

2.1 PROPOSED SYSTEM

We propose a basic home automation system with these following features:

1. Automated Lights

One can easily control the lights of the room with the help of the mobile app connected to his home Wi-Fi. These lights also have the capability of automatically turn on or off depending upon whether a person is inside or outside the room.

2. Automated Garage Door

Turns the garage door open or close with the help of a single click on the mobile or web app.

3. Temperature and Humidity Display

A LED display to display important information as well as temperature and humidity so that person is informed every time.

4. Intrusion Detection

Once a person is sleeping or not at home, the system has a capability to detect any intrusion in that person's home and notify that with an alarm.

5. Fire Alarm

If the house ever catches fire, the system will notify the user and set an alarm on, which can help notify others and those that can help us.

2.2 MODEL DIAGRAM

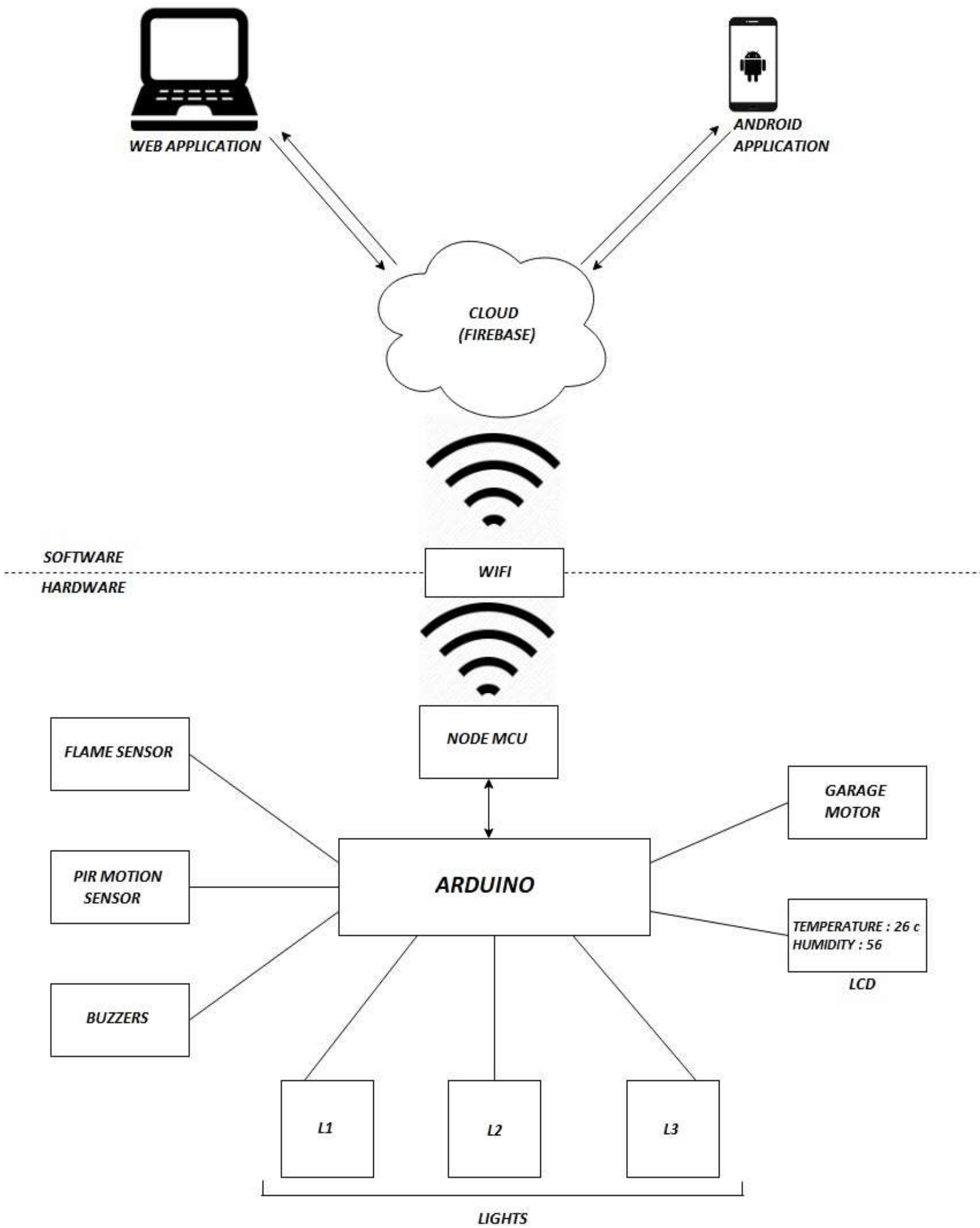


Fig:2.1 Model Diagram

2.3 WORKING

The model works in a master slave operation in which NodeMCU is a master and Arduino Uno controller is a slave. The NodeMCU requests the Arduino for all the sensors data in order to update them in cloud. Then the NodeMCU connects to the cloud and update the data in the cloud's database. When a user presses a button in the app to do anything, the value is changed inside the cloud. And these changed values are reflected back to the NodeMCU, which is constantly listening to the cloud. Then the NodeMCU uses the changed values to order the Arduino to do the required change.

For Example, when we push a led button off in the app the value of led stored inside the cloud is changed to 0 i.e. OFF. So NodeMCU notices the change because it is constantly listening to the cloud for any change. When NodeMCU notices that led variable is 0 then it orders the slave Arduino to turn off the light corresponding to led variable. This whole change is reflected in terms of a state which means the Arduino is constantly monitoring a state of the current house which contain all the information about all the sensors and light etc. The NodeMCU is monitoring the state of the cloud and the database. So any change in either of the state will always be represented.

The medium used to represent these states is JSON. It has the capability to monitor each state effectively. The whole information processing part is done on the NodeMCU that is why it is a master in this context. The job of the Uno controller is to only transfer the current state and update the state after getting order from the master i.e. NodeMCU.

CHAPTER 3

REQUIREMENT ANALYSIS

In this project there are two type of requirements:

1. Hardware Requirements
2. Software Requirements

3.1 HARDWARE REQUIREMENTS

The first and the foremost requirement is that of an infrastructure or a home for performing automation. A small model house is used as the infrastructure to perform automation upon. Multiple electronic components are used for building electronic circuits

Table 1: Specifications of electronic components used in to design the proposed system.

Components	**Specifications**
1. Arduino Uno	22 pins, operating voltage 6-20 v
2. NodeMCU	Esp8266 CPU
3. LED’S	5V
4. LCD	16 X 2 Display
5. PIR Sensor	Voltage: DC 3-5 v, Range: 2-30 cm
6. Step up Motor	24BYJ48 Stepper Motor
7. Infrared Flame Detection Sensor	3 Pins ; 5 v
8. Buzzer	5 v
9. Jump Wire	M-M, M-, F-F
10. Breadboard	Standard

3.2 SOFTWARE REQUIREMENTS

In order to design this system there are certain software requirements for this project also and these are:

- **Requirement for Arduino Programming**

It includes the Arduino programming language and the Arduino IDE.

- **Requirement for Website Development**

It includes the use of Nodejs, Express Framework, HTML, CSS and JavaScript.

The text editor used for website development was Sublime Text Editor.

GitHub is used as a version control system for the website.

- **Requirement for Android Application Development**

It includes the use of java and the official Android Studio IDE by google for the successful development of the application.

- **Requirement for Cloud**

We used Firebase by google for all the cloud based operation. The database used was the Firebase Real-time Database.

These Requirements will be discussed in detail in the Description of Software Chapter.

CHAPTER 4

DESCRIPTION OF HARDWARE USED

4.1 HARDWARE COMPONENTS

4.1.1 Arduino Uno

Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again. “Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

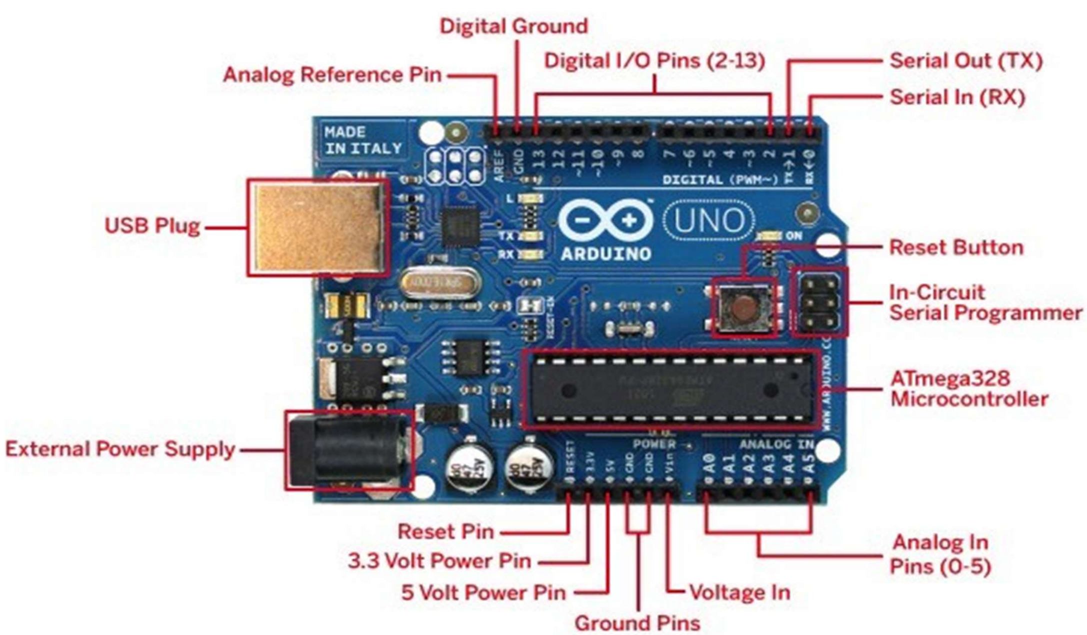


Fig: 4.1 Arduino Uno Board

Technical Specifications

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13

Fig: 4.2 Technical Specs

Programming

The Arduino Uno can be programmed with the (Arduino Software (IDE)). Select "Arduino/Genuino Uno from the Tools > Board menu (according to the microcontroller on your board). The ATmega328 on the Arduino Uno comes pre-programmed with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

4.1.2 NodeMCU

NodeMCU is an open source Lua based firmware for the ESP8266 Wi-Fi SOC from Espressif and uses an on-module flash-based SPIFFS file system. NodeMCU is implemented in C and is layered on the Espressif NON-OS SDK.

The firmware was initially developed as is a companion project to the popular ESP8266-based NodeMCU development modules, but the project is now community-supported, and the firmware can now be run on any ESP module.

It has following features:

- Built-in Micro-USB, with flash and reset switches, easy to program
- Full I/O port and Wireless 802.11 supported, direct download no need to reset
- Arduino compatible, works great with the latest Arduino IDE/Mongoose IoT/Micropython

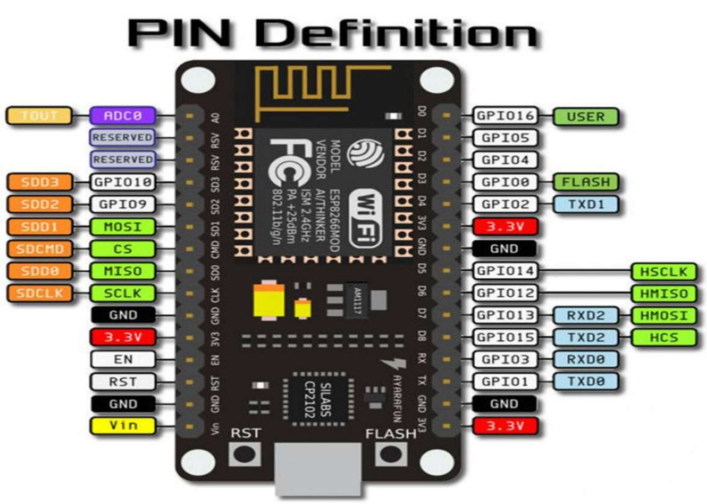


Fig 4.3 : NodeMCU

4.1.3 LED’s

A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons. This effect is called electroluminescence. The colour of the light (corresponding to the energy of the photons) is determined by the energy required for electrons to cross the band gap of the semiconductor. White light is obtained by using multiple semiconductors or a layer of light-emitting phosphor on the semiconductor device.

Appearing as practical electronic components in 1962, the earliest LEDs emitted low-intensity infrared light. Infrared LEDs are used in remote-control circuits, such as those used with a wide variety of consumer electronics. The first visible-light LEDs were of low intensity and limited to red. Modern LEDs are available across the visible, ultraviolet, and infrared wavelengths, with high light output.

Early LEDs were often used as indicator lamps, replacing small incandescent bulbs, and in seven-segment displays. Recent developments have produced white-light LEDs suitable for room lighting. LEDs have led to new displays and sensors, while their high switching rates are useful in advanced communications technology.

LEDs have many advantages over incandescent light sources, including lower energy consumption, longer lifetime, improved physical robustness, smaller size, and faster switching. Light-emitting diodes are used in applications as diverse as aviation lighting, automotive headlamps, advertising, general lighting, traffic signals, camera flashes, lighted wallpaper and medical devices.

Unlike a laser, the colour of light emitted from an LED is neither coherent nor monochromatic, but the spectrum is narrow with respect to human vision, and functionally monochromatic.

Fig 4.4 : LED

4.1.4 LCD Module

We come across LCD displays everywhere around us. Computers, calculators, television sets, mobile phones, digital watches use some kind of display to display the time. An LCD is an electronic display module which uses liquid crystal to produce a visible image. The 16×2 LCD display is a very basic module commonly used in DIYs and circuits. The 16×2 translates 0 a display 16 characters per line in 2 such lines. In this LCD each character is displayed in a 5×7 pixel

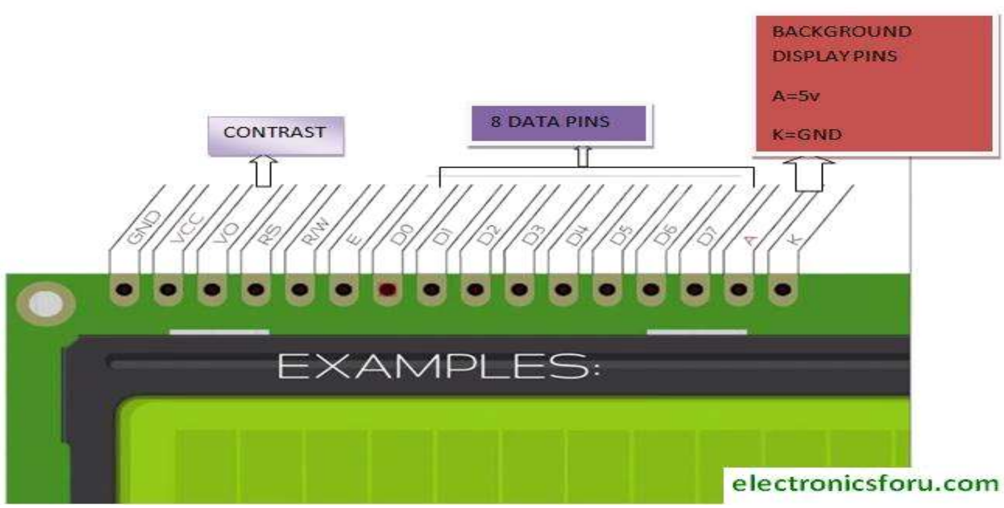


Fig 4.5 : LCD Module

4.1.5 PIR Sensor

A **Passive Infrared Sensor** (PIR sensor) is an electronic sensor that measures infrared (IR) light radiating from objects in its field of view. They are most often used in PIR-based motion detectors. PIR sensors are commonly used in security alarms and automatic lighting applications. PIR sensors detect general movement, but do not give information on who or what moved. For that purpose, an active IR sensor is required.

PIR sensors are commonly called simply "PIR", or sometimes "PID", for "passive infrared detector". The term passive refers to the fact that PIR devices do not radiate energy for detection purposes. They work entirely by detecting infrared radiation (radiant heat) emitted by or reflected from objects.

How PIRs Work

PIR sensors are more complicated than many of the other sensors (like photocells, FSRs and tilt switches) because there are multiple variables that affect the sensors input and output.

The PIR sensor itself has two slots in it, each slot is made of a special material that is sensitive to IR. The lens used here is not really doing much and so we see that the two slots can 'see' out past some distance (basically the sensitivity of the sensor). When the sensor is idle, both slots detect the same amount of IR, the ambient amount radiated from the room or walls or outdoors. When a warm body like a human or animal passes by, it first intercepts one half of the PIR sensor, which causes a positive differential change between the two halves. When the warm body leaves the sensing area, the reverse happens, whereby the sensor generates a negative differential change. These change pulses are what is detected.

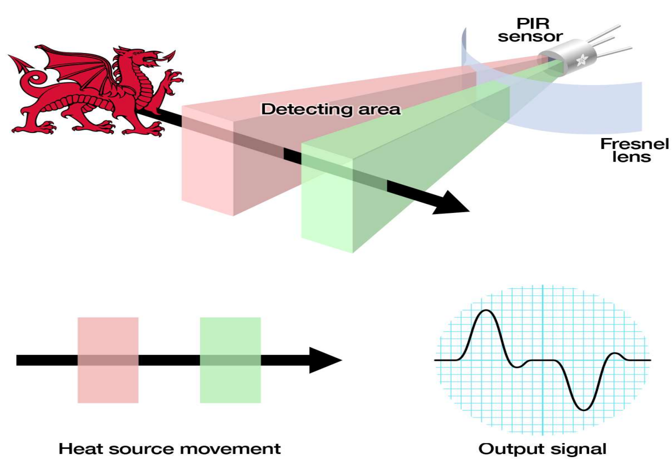


Fig 4.6 : PIR Sensor

4.1.6 IR Sensor

The IR sensor itself is housed in a hermetically sealed metal can to improve noise/temperature/humidity immunity. There is a window made of IR-transmissive material (typically coated silicon since that is very easy to come by) that protects the sensing element. Behind the window are the two balanced sensors.

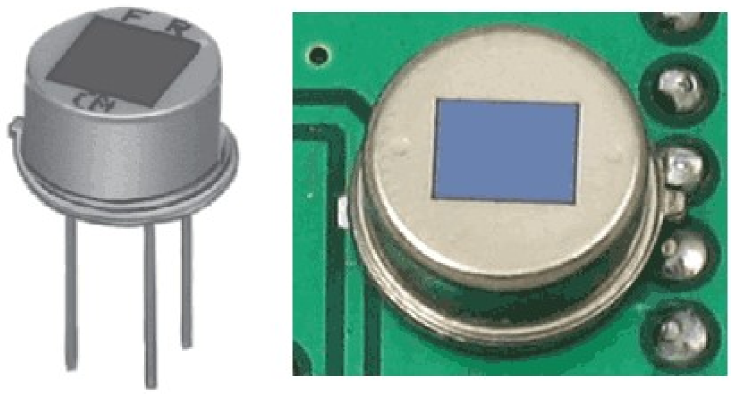


Fig 4.7: IR Sensor

4.1.7Stepper Motor

A Stepper Motor or a step motor is a brushless, synchronous motor which divides a full rotation into a number of steps. Unlike a brushless DC motor which rotates continuously when a fixed DC voltage is applied to it, a step motor rotates in discrete step angles. The Stepper Motors therefore are manufactured with steps per revolution of 12, 24, 72, 144, 180, and 200, resulting in stepping angles of 30, 15, 5,2.5, 2, and 1.8 degrees per step. The stepper motor can be controlled with or without feedback.

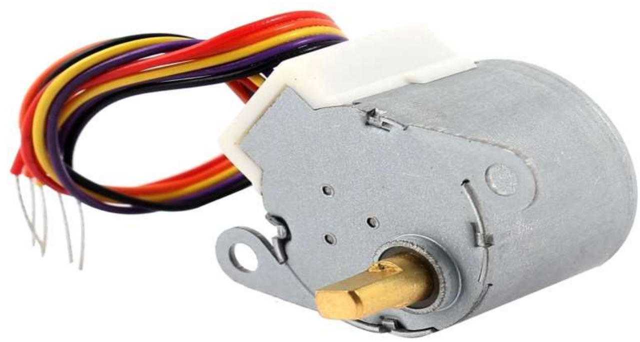


Fig: 4.8 Stepper Motor

4.1.8 Infrared Flame Detection Sensor

A flame detector is a sensor designed to detect and respond to the presence of a flame or fire, allowing flame detection. Responses to a detected flame depend on the installation, but can include sounding an alarm, deactivating a fuel line (such as a propane or a natural gas line), and activating a fire suppression system. When used in applications such as industrial furnaces, their role is to provide confirmation that the furnace is working properly; in these cases, they take no direct action beyond notifying the operator or control system. A flame detector can often respond faster and more accurately than a smoke or heat detector due to the mechanisms it uses to detect the flame.

Fig 4.9: Infrared Flame Detection Sensor

4.1.9 Buzzer

A buzzer or beeper is an audio signalling device, which may be mechanical, electromechanical, or piezoelectric (piezo for short). Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.

Fig 4.10: Buzzer

4.1.10 Jump Wires

A jump wire (also known as jumper wire, or jumper) 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.

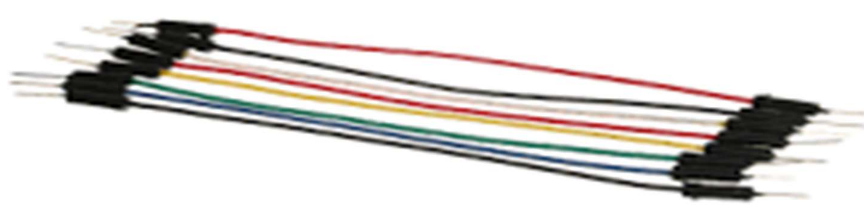


Fig 4.11: Jump Wires

4.1.11 Bread Board

A breadboard is a construction base for prototyping of electronics. Originally the word referred to a literal bread board, a polished piece of wood used for slicing bread.

In the 1970s the solderless breadboard (a.k.a. plugboard, a terminal array board) became available and nowadays the term "breadboard" is commonly used to refer to these.

Because the solderless breadboard does not require soldering, it is reusable. This makes it easy to use for creating temporary prototypes and experimenting with circuit design. For this reason, solderless breadboards are also popular with students and in technological education.

Older breadboard types did not have this property. A stripboard (Veroboard) and similar prototyping printed circuit boards, which are used to build semi-permanent soldered prototypes or one-offs, cannot easily be reused. A variety of electronic systems may be prototyped by using breadboards, from small analog and digital circuits to complete central processing units (CPUs).

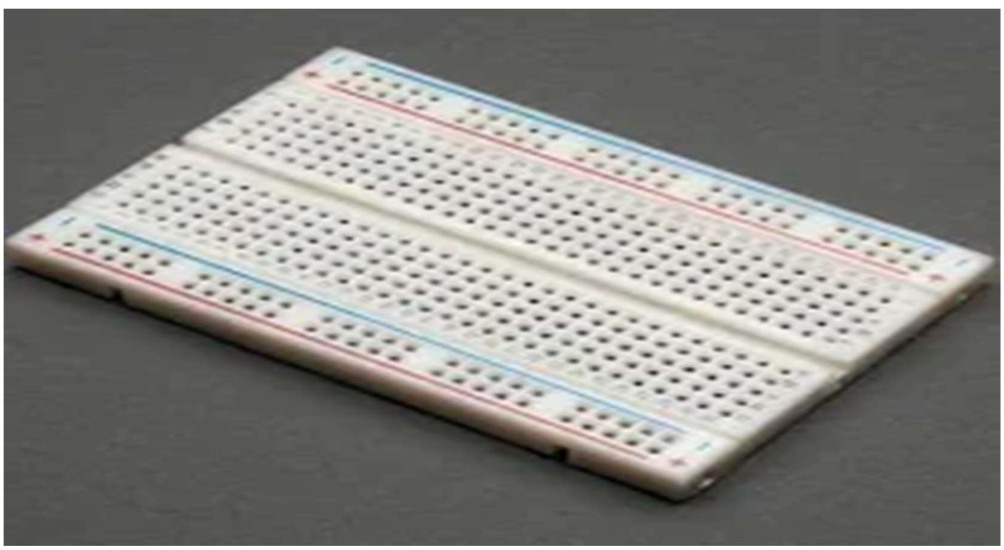


Fig 4.12: Bread Board

CHAPTER 5

DESCRIPTION OF SOFTWARE USED

5.1 PROGRAMMING IN ARDUINO

For programming in Arduino we use the Arduino IDE.

The **Arduino Integrated Development Environment (IDE)** is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

The source code for the IDE is released under the GNU General Public License, version 2.The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

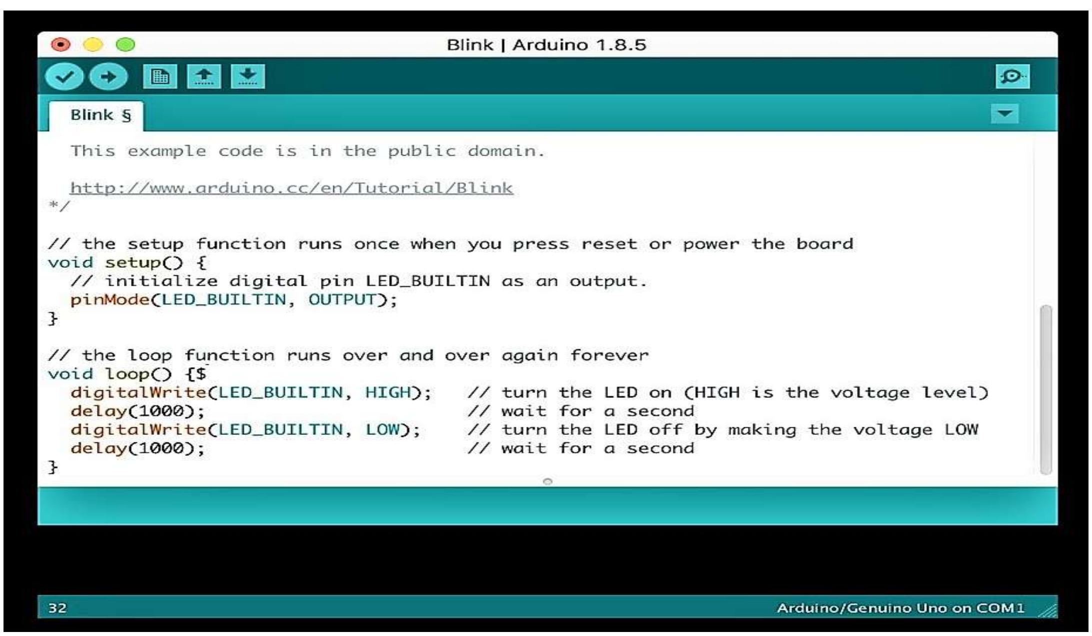


Fig: 5.1 Arduino IDE Screenshot Example

5.2 PROGRAMMING FOR WEBSITE

➢ **Nodejs**

Node.js is an open-source, cross-platform Java Script run-time environment that executes JavaScript code outside of a browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser.

Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web application development around a single programming language, rather than different languages for server- and client-side scripts.

➢ **ExpressJs**

Express.js, or simply Express, is a web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js.

➢ **HTML, CSS AND JavaScript**

HTML, CSS, and JavaScript are three district coding languages that together are used to build Website and Web Applications.

HTML

Hyper-Text Markup Language is used to put the structure of a website together. (Like a skeleton of a body)

CSS

Cascading Style Sheets acts like makeup for the HTML. CSS improves the colours and layout of a website structure built with HTML.

JavaScript

JavaScript is a full-on programming language that adds interactivity and functionality to a website.

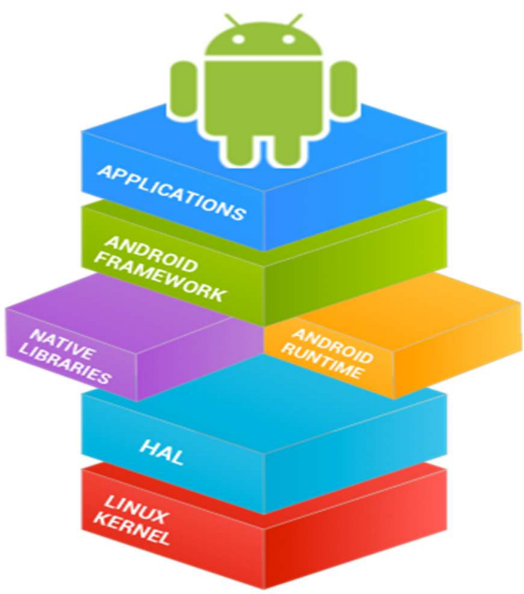
- **Sublime Text**

Sublime Text is a proprietarycross-platformsource code editor with a Pythonapplication programming interface (API). It natively supports many programming languages and markup languages, and functions can be added by users with plugins, typically community-built and maintained under free-software licenses.

5.3 PROGRAMMING IN ANDROID

- **Android Programming**

Android software development is the process by which new applications are created for devices running the Android operating system. Google states that "Android apps can be written using Kotlin, Java, and C++ languages" using the Android software development kit (SDK), while using other languages is also possible. All non-JVM languages, such as Go, JavaScript, C, C++ or assembly, need the help of JVM language code, that may be supplied by tools, likely with restricted API support. Some languages/programming tools allow cross-platform app support, i.e. for both Android and iOS. Third party tools, development environments and language support have also continued to evolve and expand since the initial SDK was released in 2008.



- **Android Studio**

Android Studio is the official integrated development environment (IDE) for Google's Androidoperating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

5.4 FIREBASE

Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014. As of October 2018, the Firebase platform has 18 products, which are used by 1.5 million apps.

Firebase Real-time Database

Firebase provides a real-time database and backend as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firebase's cloud.

The company provides client libraries that enable integration with Android, iOS, JavaScript, Java, Objective-C, Swift and Node.js applications. The database is also accessible through a REST API and bindings for several JavaScript frameworks such as AngularJS, React, Ember.js and Backbone.js.

The REST API uses the Server-Sent Events protocol, which is an API for creating HTTP connections for receiving push notifications from a server. Developers using the real-time database can secure their data by using the company's server-side-enforced security rules.

CHAPTER 6

DESIGN AND IMPLEMENTATION

The project is designed in number of steps and these steps are:

Step 1: Designing the House

First we design the model of the house, because in order to implement this project we need a house to work upon. We used a three dimensional sectioned model in order to replicate a house. For that we used mount sheet and some recycled material to be mounted on top of a hard cardboard. The cut outs in the sheet is used to represent various rooms in a house the detailed diagram of the top view of house is given below.

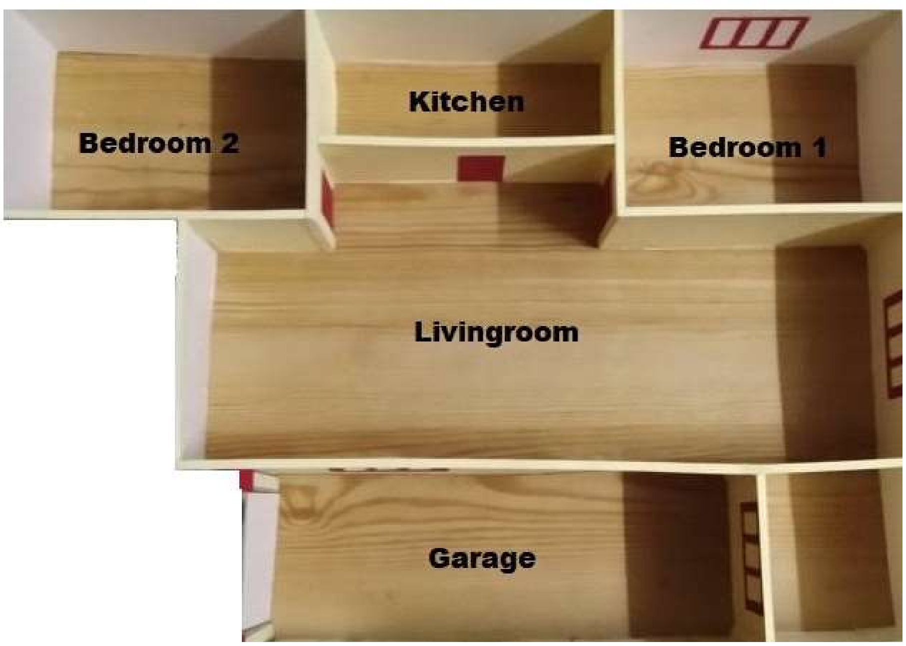


Fig 6.1 : Model of The House (Top View)

Step 2: Designing of Circuit Diagram

The step two is design of the circuit of what we need to implement, a circuit diagram can help easily model the connection so that they can be implemented very easily. For modelling of the circuit we need a software and for that we use a software called Fritzing.

Fritzing is an open-source initiative to develop amateur or hobby CAD software for the design of electronics hardware, to support designers and artists ready to move from experimenting with a prototype to building a more permanent circuit. It was developed at the University of Applied Sciences Potsdam.

CIRCUIT DIAGRAM

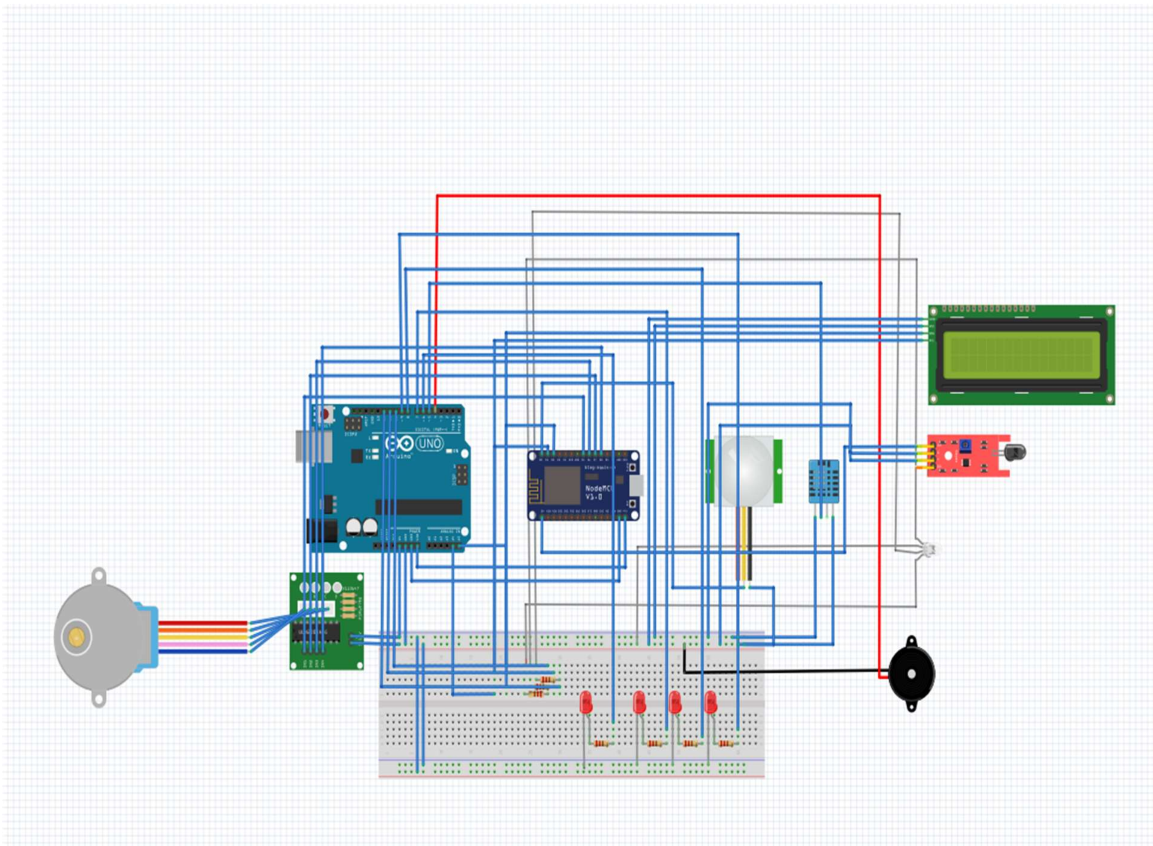


Fig 6.2: Circuit Diagram

Explanation

In order to explain the diagram, we will break down it in parts. First the NodeMCU is connected to the Arduino Uno controller in the master slave mode. In order to achieve this mode, the D1 pin of the NodeMCU is connected to the A4 pin of the Uno and the D2 pin of the NodeMCU is connected to A5 pin of the Arduino. Arduino and NodeMCU are connected using a common ground configuration and Vin of both Arduino and NodeMCU is connected together.

Then for the LCD display we use the i2c interface. I2C (Inter-Integrated Circuit) is serial bus interface connection protocol. It is also called as TWI (two wire interface) since it uses only two wires for communication. Those two wires are SDA (serial data) and SCL (serial clock). So for that SDA pin of LCD display is connected to SDA of NodeMCU that is pin D1 and SCL to D2 pin. Then 5v power and GND is connected to LCD using common power and GND on the breadboard.

The ULN2003 stepper motor driver PCB provides a direct drive interface between your microcontroller and stepper motor. The PCB provides 4 inputs for connection to your microcontroller, power supply connection for the stepper motor voltage, and ON/OFF jumper, a direct connect stepper motor header and 4 LEDs to indicate stepping state. These 4 PINS are connected to the NodeMCU at pin d5, d6, d7, d8, and the power is given from the NodeMCU itself.

Then 4 LEDs representing different rooms are connected with pin 6,7,8,9 to Arduino and common power is supplied. In order to prevent high voltage across LED's a 220 Ohm resistor is introduced between anode and the power supply.

After that for PIR sensor the Signal key is connected to D0 of the NodeMCU, and common power is supplied from the breadboard. Then for Fire Sensor module the A0 signal pin is connected to the A0 of the NodeMCU.

Then for the DHT module the signal (DATA) pin is connected to the pin 5 of the Arduino. For the RGB led a resistor is introduced for each R, G, B pin and these pin are connected to pin 10, 11, 12 of the Arduino board and GND is connected to GND on the breadboard. For buzzer pin 4 of Arduino is connected to the voltage pin of the buzzer and GND pin is connected to the breadboard.

Step 3: Testing of Individual Components

The step 3 is the most crucial step which involves testing of individual components and code them respectively.

Step 4: Coding and Integration

This step is concerned with realization of the project integrating each and every component and fully code the whole hardware i.e. the coding part.

The code for the both NodeMCU and the Arduino Uno is given in appendix

Step 5: Designing of website

This step requires to make a website for remotely controlling the hardware by manipulating data on the cloud. Technology stack used for the website consists of Nodejs for backend and HTML, CSS, JavaScript for the frontend.

Development environment consists of IDEs like Sublime and rest of the backend and testing is done using command line.

The code needs to be pushed on a git repository to keep track of its development.

Firebase real-time cloud database is integrated in to the website by using pre-built JavaScript functions and code snippets in the website.

Firebase hosting service is also used to host the website.

The directory structure and important code of the website is shown below:

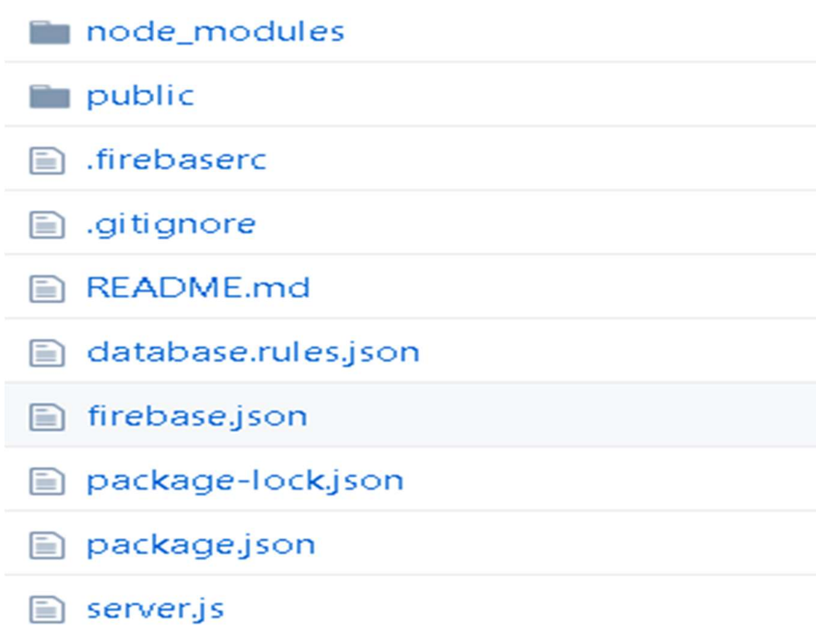


Fig 6.3 : Directory Structure

The node modules folder contains required packages for developing backend, these packages can be invoked by using require() function. A particular package contains a package.json file which confirms it is registered by the npm (node package manager).

As we can see our project folder also contains a package.json and a package-lock.json file. This is because our nodejs project is itself treated as a package by the npm. database.rules.json,

.firebaserc and firebase.json contain required metadata for using firebase cloud. README.md and .gitignore are some initially added files by github.

The server.js file calls require (‘express’) function to use express server from the node_modules directory as the default server. It contains that part of the backend which redirects the client request to the required pages of the website. As all the frontend is inside public folder thus it redirects a request for ‘domainname/public’ to the public folder.

Also it specifies the port number from which our web application will be served.

```
const express = require('express')
const server =  express()

server.get('/', function(req, res, next){
        res.send('Home Page')
})

server.use('/public', express.static(__dirname + '/public/'))



server.listen(2121);
```

The public folder mostly contains frontend and firebase code. Its directory structure is:

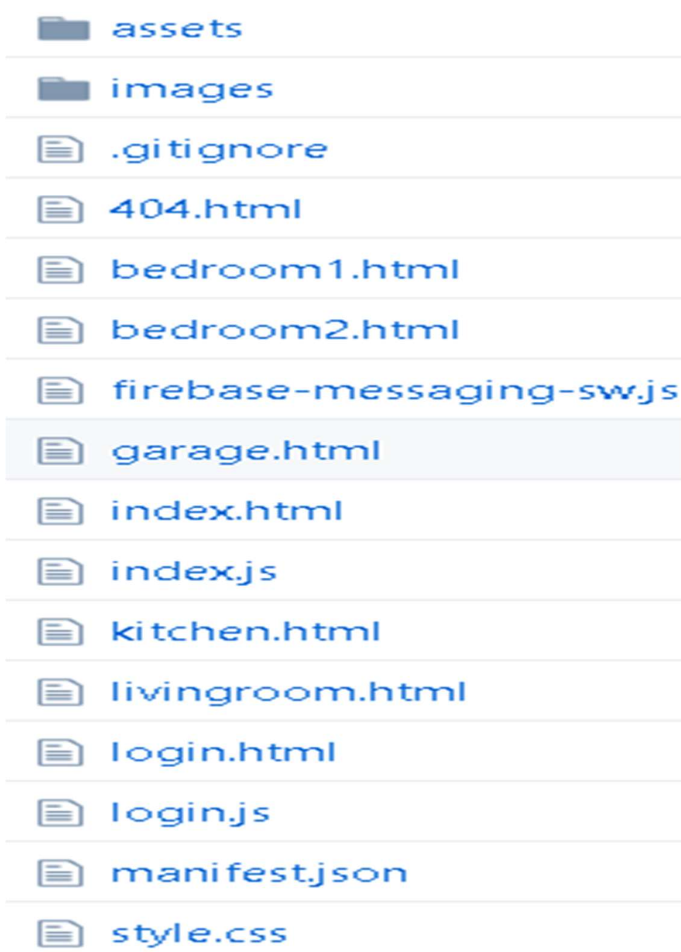


Fig 6.4 : Public Folder

The assets folder contains all the custom and required CSS as well as Javascript files. It also contains logos and fonts used in the website. The images folder contains images and graphics used in the website. All other html pages are made to represent the view of respective pages.

For example : Kitchen, living room, garage etc. html files represent respective rooms of the house. Login.html represents the login page for a user.

404.html shows up for the client requests which contain page names that are not present.

Index.html is the Home page or the dashboard of the website.

Javascript files like index.js and login.js are used to implement particular functions that get invoked when some action is performed on their respective html pages.

For example, here is the function inside login.js for logging in:

```
(function(){
// Initialize the FirebaseUI Widget using Firebase.
var ui = new firebaseui.auth.AuthUI(firebase.auth());
var uiConfig = {
  callbacks: {
    signInSuccessWithAuthResult: function(authResult, redirectUrl) {
      // User successfully signed in.
      // Return type determines whether we continue the redirect automatically
      // or whether we leave that to developer to handle.
      return true;
    },
    uiShown: function() {
      // The widget is rendered.
      // Hide the loader.
      document.getElementById('loader').style.display = 'none';
    }
  },
  // Will use popup for IDP Providers sign-in flow instead of the default, redirect.
  signInFlow: 'popup',
  signInSuccessUrl: 'index.html',
  signInOptions: [
    // Leave the lines as is for the providers you want to offer your users.
    //firebase.auth.GoogleAuthProvider.PROVIDER_ID,
    //firebase.auth.FacebookAuthProvider.PROVIDER_ID,
    //firebase.auth.TwitterAuthProvider.PROVIDER_ID,
    //firebase.auth.GithubAuthProvider.PROVIDER_ID,
    firebase.auth.EmailAuthProvider.PROVIDER_ID,
    //firebase.auth.PhoneAuthProvider.PROVIDER_ID
  ],
  // Terms of service url.
  tosUrl: 'index.html',
  // Privacy policy url.
  //privacyPolicyUrl: '<your-privacy-policy-url>'
};

ui.start('#firebaseui-auth-container', uiConfig);

})()
```

JavaScript is not only included in separate files. We have also use JavaScript code embedded inside html files for the related database manipulations on the cloud.

For example, the following code in livingroom.html represents a button and an RGB colour selector.

```
    <div class="buttonbox">
    <h1 class="Bulb"></h1>

    <button type="button" class="buttonone btn btn-primary btn-lg">Bulb</button>
    </div>


<div class="buttonbox">
<div class="rgbbox">


</div>


</div>


<div class="rgbselector">


<div class="redbox"></div>
<div class="greenbox"></div>
<div class="bluebox"></div>
<div class="raspberrybox"></div>
<div class="cyanbox"></div>
<div class="magentabox"></div>
<div class="yellowbox"></div>
<div class="whitebox"></div>
```

On a mobile view the above code looks as:

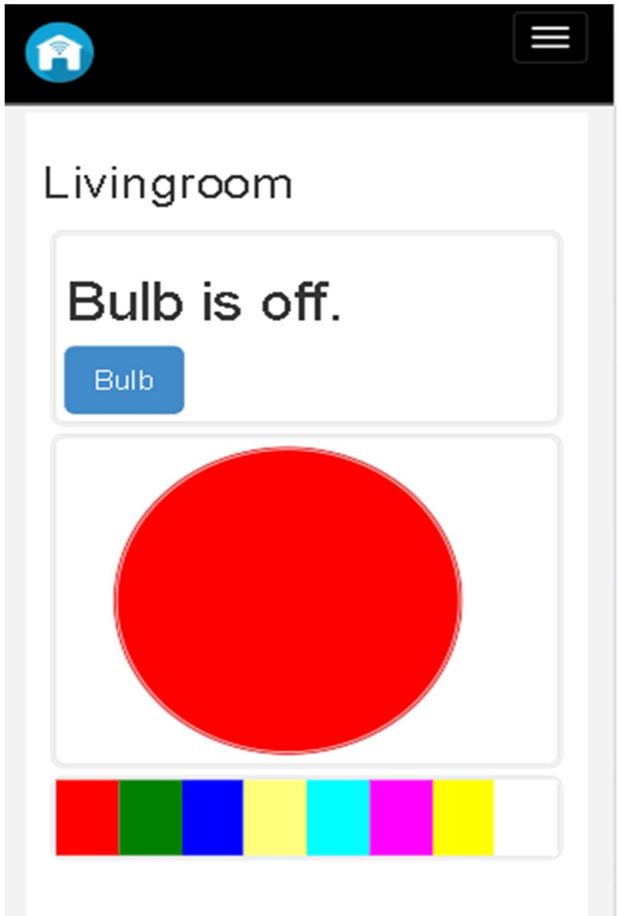


Fig 6.5 : Mobile view of Website

Now in the same livingroom.html file JavaScript script is embedded for the interacting with the button and RGB selector. Also JavaScript code for firebase is added to manipulate and track the present state of data in the cloud.

The code for this component is given in the appendix

The rest of the firebase and JavaScript code contains similar functions and logic for manipulating the DOM and the data in the cloud.

Step 6: Designing of Android App

In this step we designed an Android application for this project.

For this first we design the UI for the app using xml. We designed a login page, Main Menu page, and a room page for the application.

These layouts are given below:

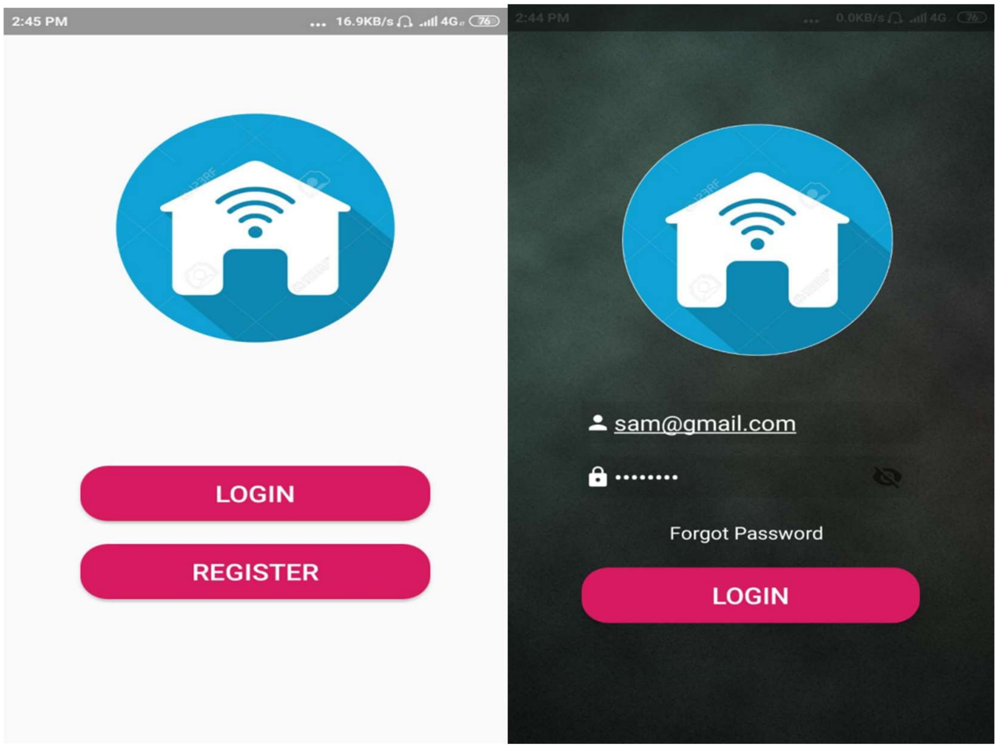


Fig 6.6: Start Page and Login Page for the Application

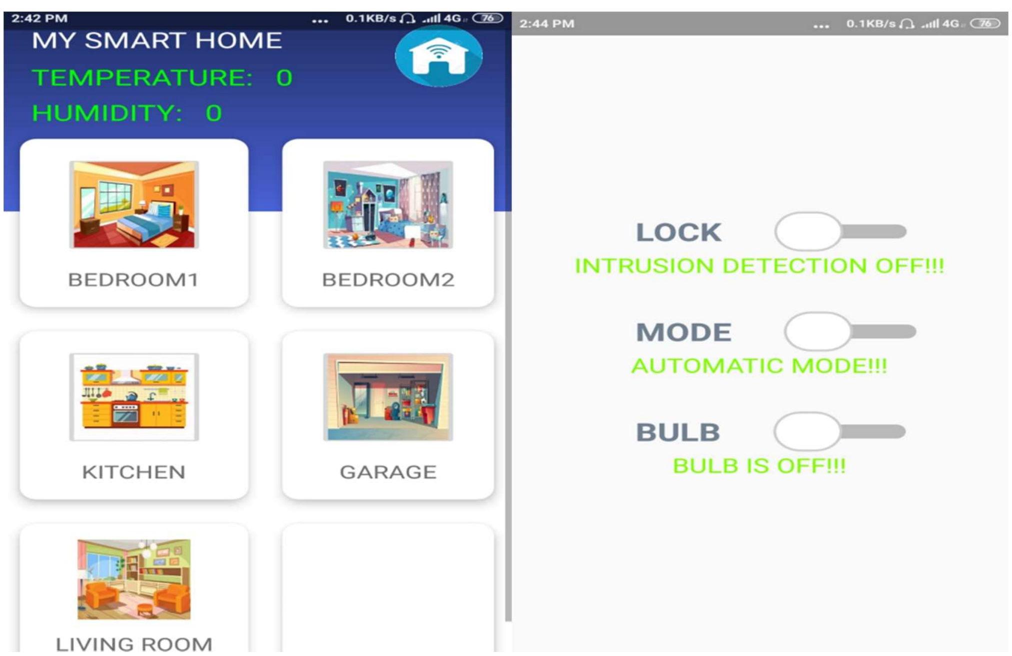


Fig 6.7 : Main Menu and Room Page of the App

The functionality of the app is similar to that of the website and it works in a similar way. The main reason of creating this app to provide users with a comfort of an app on a smart phone as Android covers most of the modern devices so it can run on any of them.

CHAPTER 7

ADVANTAGES AND DISADVANTAGES

Due to the increasing popularity of smart homes, many companies have been able to advance the technology necessary to bring affordably priced smart home products to the market.

People are investing in smart home products every day. Whether it's an automatic coffee maker, a thermostat, or an entire home security system, many families are making their homes a little bit smarter, and a lot more convenient.

Unfortunately, as with most things, a smart home has its advantages and disadvantages.

7.1 ADVANTAGES

1. Adding Convenience to Your Daily Life

When you convert your home into a smart home, you'll have all of your products programmed to your specific needs. Additionally, being able to control your home, no matter where you are, can be extremely beneficial.

2. Save Money on Utilities

Utilities can amount to several hundred dollars per month. Home automation can turn off lights or lower the thermostat automatically when you aren't using them and easily lower your utility bills by 10% to 25%.

3. Increased Home Safety

Many accidents happen in the home because of poor lighting. Home automation can automatically turn lights on in closets, stairways, and other dark places when you enter and decrease the chance of accidentally tripping or running into things.

4. Home Security

Although home security is a priority for everyone, high installation cost or monthly monitoring charges make security systems cost prohibitive for many homeowners. Home automation provides an inexpensive solution to home security.

2. Good for the Environment

In a time when we are all becoming more environmentally aware, home automation provides a good solution to help preserve our natural resources. Home automation products can reduce power consumption and automatically turn off lights and appliances when they aren't in use.

3. Peace of Mind

Never again worry about your home while you're away. Using cameras and an internet connection, you can check on the status of your home or kids from anywhere in the world using a PC or web-enabled phone.

4. Learning Experience for Children

Technology is here to stay and the more your children learn about upcoming technology the better prepared they are for the future. Turn your home into a classroom, as your home automation projects become a learning experience for your children.

7.2 DISADVANTAGES

1. Cost

Most families are able to purchase smart home products, but that doesn't mean it won't leave a dent in your wallet. You can purchase the products one at a time and it won't seem like too much, $50 here, $300 there, but by the time you have the smart home system you want, you will likely have spent a larger sum than you would have if you had purchased non-smart products.

2. Complex Technology

Automating everything in life may sound extremely appealing, but sometimes a good old-fashioned flip of the switch is a lot easier than reaching for your smart phone to turn lights on and off. Before you decide which system is right for you, think about how far you really want to take home automation in your household.

3. Installation

Depending on the complexity of the system, installing a home automation device can be a significant burden on the homeowner. It can either cost you money if you hire an outside contractor or cost you time if you venture to do it yourself.

4. Dependency on Internet

The basic requirement for the smart home system is the internet. Without a good and strong internet connection, you will not be able to take control of this. If there is no internet connection for some reason, there is no other way through which you can access and control your system.

5. Dependency on Professionals

In case there is a problem with the smart home system, you cannot simply call a handyman or someone similar to repair or manage the bug. You will have to depend on the professionals. Only the company professionals can help you to handle the problems. A professional would be able to take care of the disadvantages of home appliances.

6. Slight Learning Curve

I know I stated in the advantages that most smart home systems are actually very easy to use, but at the same time there is still somewhat of a learning curve for most people. For anyone already immersed in technology, converting your smart home will be a breeze, but for anyone not so tech savvy, it may make for a lot of time spent reading manuals.

7. Reliability

A smart home will be extremely reliant on your internet connection. If your connection drops, you’ll be left with a lot of smart products that won’t work. Additionally, wireless signals can possibly be interrupted by other electronics in your home and cause some of your smart products to function slowly or not at all.

CHAPTER 8

APPLICATIONS

1. Home Automation for The Elderly and Disabled

The form of home automation focuses on making it possible for older adults and people with disabilities to remain at home, safe and comfortable. Home automation is becoming a viable option for older adults and people with disabilities who would prefer to stay in the comfort of their homes rather than move to a healthcare facility. This field uses much of the same technology and equipment as home automation for security, entertainment, and energy conservation but tailors it towards older adults and people with disabilities.

There are two basic forms of home automation systems for the elderly: embedded health systems and private health networks. Embedded health systems integrate sensors and microprocessors in appliances, furniture, and clothing which collect data that is analysed and can be used to diagnose diseases and recognize risk patterns. Private health networks implement wireless technology to connect portable devices and store data in a household health database. Due to the need for more healthcare options for the aging population "there is a significant interest from industry and policy makers in developing these technologies".

2. Lighting Control

Smart lighting allows you to control wall switches, blinds, and lamps, but how intuitive is a lighting control system? It turns out, quite; its capabilities are extensive. You're able to schedule the times lights should turn on and off, decide which specific rooms should be illuminated at certain times, select the level of light which should be emitted, and choose how particular lights react through motion sensitivity, as seen with Belkin's WeMo Switch + Motion, which is both affordable and easy to use with its plug-and-play simplicity.

3. HVAC (Heating, Ventilation and Air Conditioning) Regulation

As fuel costs rise and the availability and sustainability of our resources becomes a greater concern, heating/cooling our homes efficiently is less a budgetary bonus and more of a necessity. Over the past year, smart thermostats and automated home heating systems have become more readily available and easily incorporate into any home. Heating and cooling our homes consumes an average of 50% of energy costs yearly, making daily HVAC regulation progressively rewarding. Maintaining a substantial lead among the nearly non-existent competition, the Nest Learning Thermostat, learns your heating and cooling preferences over time, eliminating the need for programming and is accessible from your smartphone app. With automated HVAC you are able to reduce the heat when a room is unoccupied, and increase or decrease it at specific times based on your schedule and occupancy.

4. Lawn Irrigation Systems

A lush and healthy lawn is a source of pride for most homeowners, but the weather doesn't always cooperate and provide the adequate elements for a flourishing landscape. For decades we've relied on sprinkler systems to keep our yards at peak presentation, but at what cost? The average American home spends approximately 30% of their daily water usage on lawn and garden maintenance. Nearly half of that amount is wasted due to inefficiency. If you apply that statistic to the national average, up to 4.5 billion gallons of water is wasted per day through ineffective watering methods. If we reflect upon the monetary impact of this, it results in Americans spending over a thousand dollars a year in water, with a portion of that being waste. The global effects are even greater when you consider the growing concern over climate change and the dramatic decrease in agricultural natural resources. However, sprinkler control systems, like Skydrop, are providing water regulation through real-time communication with local weather data. If a rainstorm develops and deposits two inches of rainwater on your lawn, the automated sprinkler detects the saturation and disables its scheduled watering. Conversely, the system will be alerted to dry conditions and supply the necessary amount of nourishment, without over-watering.

5. Smart Appliances

Smart refrigerators, such as LG's Smart ThinQ, allow you to scan grocery store receipts and keep an inventory of your items, and alerts you if an item is about to expire. More impressively, it suggests recipes based on your refrigerator's contents and lets you know when you need to replace items.

Smart ovens synch with your smartphone and automatically preheat to the correct temperature based on a recipe selected from your database. While these appliance options seem a bit superficial and convenience based, there is a conservation factor as well. By automating your kitchen appliance and making them accessible from your smart device, you're able to sever the electricity supplied to unused appliances and reduce your energy consumption and costs. Considering the number of appliances the average household owns; this could save a substantial amount of money over time.

6. Home Robots and Security

A household security system integrated with a home automation system can provide additional services such as remote surveillance of security cameras over the Internet, or access control and central locking of all perimeter doors and windows.

Instead of installing a surveillance camera you can always use the webcam for keeping track on your children as it can successfully perform the activity of taking note of all activities. Also it is cheap and very minute and the activities can be checked with the help of the internet

While efficiency and conservation are certainly IoT benefits, its potential to have improved control over home security is a primary focus. Smart locks, like Kwikset's Kevo, a Bluetooth enabled electronic deadbolt, and various connected home security systems, such as iSmart Alarm, offer a variety of features including door and window sensors, motion detectors, video cameras and recording mechanisms. All of which are connected to a mobile device and accessible via the cloud, thus enabling you to access real-time information on the security status of your home. Naturally, there is a great deal of scrutiny regarding the level of trust in controlling your home's security system via a mobile device, but it begs earnest exploration when weighing the potential benefits and peace of mind it provides homeowners.

7. Occupancy-Aware Control System

It is possible to sense the occupancy of the home using smart metersand environmental sensors like CO_2 sensors, which can be integrated into the building automation system to trigger automatic responses for energy efficiency and building comfort applications.

The quality of air indoors can also be monitored using various sensors which can be integrated in the home automation system. For example, Air Quality Egg is used by people at home to monitor the air quality and pollution level in the city and create a pollution map.

8. Indoor Positioning System

An indoor positioning system (IPS) is a system used to locate objects or people inside a building using lights, radio waves, magnetic fields, acoustic signals, or other sensory information.[1] There are several commercial systems on the market, but there is no standard for an IPS system.

Indoor positioning systems use different technologies, including distance measurement to nearby anchor nodes (nodes with known fixed positions, e.g. WiFi / LiFiaccess points or Bluetoothbeacons), magnetic positioning, dead reckoning. They either actively locate mobile devices and tags or provide ambient location or environmental context for devices to get sensed. Detecting the device's orientation (often referred to as the compass direction in order to disambiguate it from smartphone vertical orientation) can be achieved either by detecting landmarks inside images taken in real time, or by using trilateration with beacons. There also exist technologies for detecting magnetometric information inside buildings or locations with steel structures or in iron ore mines.

CHAPTER 9

FUTURE DEVELOPMENT

Future scope for the home automation systems involves making homes even smarter. Homes can be interfaced with sensors including motion sensors, light sensors and temperature sensors and provide automated toggling of devices based on conditions. More energy can be conserved by ensuring occupation of the house before turning on devices and checking brightness and turning off lights if not necessary. The system can be integrated closely with home security solutions to allow greater control and safety for home owners.

The next step would be to extend this system to automate a large scale environment, such as offices and factories. Home Automation offers a global standard for interoperable products. Standardization enables smart homes that can control appliances, lighting, environment, energy management and security as well as the expandability to connect with other networks.

Future homes will be able to offer almost all required services, e.g., communication, medical, energy, utility, entertainment, and security. As we move into the next generation, more and more devices will begin to connect to one another. The dream is a future in which data is communicated between devices and humans without relying on manual input of individual bytes.

For example, a smart thermostat that is able to automatically gauge the temperature of a room and then adjust the central heating and cooling units as necessary or a washing machine that automatically detects its contents and programs itself to be finished washing at a specified time.

These are all goals that engineers are working toward and depend not only on advances in data-mining technologies but also in big data computing. Pert is the next generation home automation innovation, that lets you control, monitor and secure your home with your smartphone. The future healthcare service provider will consider the smart home an effective way of providing remote healthcare services, especially to the elderly and disabled who do not require intensive healthcare support. As technologies continue to advance, you can expect the house of tomorrow to be even more automated than that of today.

CONCLUSION

The project has proposed the idea of smart homes that can support a lot of home automation systems. A smart home contains a connection between wireless communication, sensors, monitoring and tracking. Smart homes are a huge system that includes multiple technologies and applications that can be used to provide security and control of the home easily.

This project discussed the designed modules like sensors' circuits, monitoring and tracking of the home through mobile application and web platform. In this project, an efficient approach for smart homes was proposed and implemented.

A series of experiments have been carried out on the proposed smart home. These experiments show how to detect the fire, water leaking, smoke. Also how to detect any intruder to the home, detect and control the weather of the any room and how to secure the home through an access code. In addition, this project illustrates the way to send notifications to the homeowner about the actions in the home. Also this project showed the idea of measuring the temperature in all rooms and detect any fire happens and to detect any motion in the home by using ultrasonic sensors.

Central control for the entire home has been designed using three microcontroller system designs. These designs were for access control to the home, temperature validation, and control board system to connect all the security and control circuits together.

This project shows a general overview of smart home project that are masterminded by their expected services. It also covers the survey of various technologies which emphatically support the home automation systems in reliable way. This project recognizes a few future bearings of smart home research. The future healthcare service provider will consider the smart home powerful method for giving remote social insurance administrations, particularly to the elderly and disabled people. In future home automation will smarter and it would be extended to the large-scale environment.

REFERENCES

[1] R. A. Ramlee, D. H. Z. Tang, and M. M. Ismail, "Smart home system for Disabled People via Wireless Bluetooth," in System Engineering and Technology (ICSET), 2012 International Conference on, 2012, pp. 1-4.

[2] M. Yan and H. Shi, "SMART LIVING USING BLUETOOTH-BASED ANDROID SMARTPHONE," International Journal of Wireless & Mobile Networks (IJWMN), vol. 5, pp. 65-72, 2013.

[3] Deepali Javale, Mohd. Mohsin, Shreerang Nandanwar, and M. Shingate, "Home Automation and Security System Using Android ADK," International Journal of Electronics Communication and Computer Technology, vol. 3, pp. 382-385, 2013.

[4] S. V. A. Syed Anwaarullah, "RTOS based Home Automation System using Android," International Journal of Advanced Trends in Computer Science and Engineering, vol. 2, pp. 480-484, 2013.

[5] A. ElShafee and K. A. Hamed, "Design and Implementation of a WiFi Based Home Automation System," World Academy of Science, Engineering and Technology, pp. 2177-2180, 2012.

[6] A. Z. Alkar and U. Buhur, "An Internet based wireless home automation system for multifunctional devices," Consumer Electronics, IEEE Transactions on, vol. 51, pp. 1169-1174, 2005.

[7] http://ijettjournal.org/Special%20issue/NCEITCS-2017/NCEITCS-141

[8] https://www.irjet.net/archives/V4/i10/IRJET-V4I1061

[9] https://google.com

[10] https://wikipedia.com

Appendix

NodeMCU Code

```
/*----------------------------------------------------------Inculsion of various libraries to be used in this project-------------------

#include <SD.h>
#include "FirebaseESP8266.h"
#include <SoftwareSerial.h>
#include <ArduinoJson.h>
#include <ESP8266WiFi.h>
#include <Wire.h>
#include <dht.h>
#include "Stepper.h"

#include  <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,16,2);
/*-------------------------------------------------------Definations-----------------------------------------------*/

#define user "Dexter" // HOME WIfi SSID
#define pass "12345678" // WIFI PASSWORD
#define FIREBASE_AUTH "zfEOrJkczD1U75NVogSsHiumtqPfom1yfhOZF6f1" // FIREBASE AUTHORIZATION KEY
#define FIREBASE_HOST "test-3f5ea.firebaseio.com" // FIREBASE HOSTNAME

/*-------------------------------------------------------Pins-----------------------------------------------------*/
//GIVING THE DEFINATIONS TO THE PIN WHICH MEANS WHICH PIN OF SENSORS AND DEVICES IS CONNECTED TO WHICH PIN OF NODEMCU



#define pir_r2pin D0 //PIR SENSOR OF ROOM 2 IS CONNECTED TO D0 PIN OF NODEMCU
#define systemled D4 //LED to show whether the system is on or off
#define fire_pin A0  //Fire sensor pin is connected to the Ao pin of Nodemcu


Stepper myStepper(32, D5, D7, D6, D8); // Stepper motor connected to D5,6,7,8 PIN OF NODEMCU

/*-------------------------------------------------------Global Variables------------------------------------------*/
//Global variables to be used to implement program logic

int garage_done = 0;
const int stepsPerRevolution = 90;
int state = LOW;                                          // by default, no motion detected
int val = 0;                                              // variable to store the pir_r2pin status (value)
long timersr1, timerer1, diff;
long timersr2,timerer2;// variables for timer delay to get nonblocking delay
int master1 = 0;

int fbtstart, fbtend;
int fire_logic;
int fire_buzzer;
int intrusion_buzzer;
int intr_logic;
int intrstart, intrend;

/*-----------------------------------------------------------Buffers-------------------------------------------------*/
//INCLUDES VARIOUS JSON BUFFERS TO HOLD THE VALUES OF CURRENT JSON STATE OF THE SYSTEM

DynamicJsonBuffer jsonBuffer;
DynamicJsonBuffer jsonBuffer2;
DynamicJsonBuffer jsonBuffer3;
DynamicJsonBuffer jsonBuffer4;

FirebaseData firebaseData; //Buffer to hold the data from the firebase


/*-------------------------------------------------------Function Declaration------------------------------------*/
void sendToArduino(String test);                          // Function to take a String and send it to Arduino via I2C Interface
```

```
/*--------------------------------------------------------Main Setup ------------------------------------------*/
void setup() {

  Serial.begin(9600); // Serial interface to begin at 9600 baud rate
  Wire.begin(D1, D2); // I2C interface at pin D1 and D2

  myStepper.setSpeed(600); // Setting the motor speed to 600 steps

  pinMode(pir_r2pin, INPUT); // setting the modes of the various pins to input and output
  pinMode(systemled, OUTPUT);
  pinMode(fire_pin, INPUT);

  lcd.init();                      // initialize the lcd
  lcd.backlight();
  //pinMode(pir_r1pin, INPUT);

  digitalWrite(systemled, HIGH);

  Serial.println("Ready");
  delay(60000); // A delay in order to system to boot up effectively so that all sensors are heated effective for effective working


  WiFi.begin(user, pass); //Starting of wireless connection
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.println("connected");
  Serial.println(WiFi.localIP());

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);


  //Initializing of various non blocking timers because the delay library function cause system to block
  timersr2 = millis();
  timersr1 = millis();
  fbtstart = millis();
  intrstart = millis();

  fire_logic = 1;
  intr_logic = 1;

  digitalWrite(systemled, LOW);
}

/*------------------------------------------------------------Loop------------------------------------------------------*/

void loop() {

// Various local variables to be used in the code
  int val = 0;
  String json;
  String json_data;
  String json_data2;
  char arr[1000];
  int led_r2;
  int pir_r2;
  int pir_r2status;
  int led_r2cntrl;
  int led_r1;
  int pir_r1;
  int pir_r1status;
  int led_r1cntrl;
  bool successful;
  int led_r2mode;
  int led_r1mode;
  int humidity;
  int temperature;
```

```
int fan;
int fan_tempset;
int fan_temp;
int fan_cntrl;
int fire;
int fire_alarm;
int intrusion_set;
int intrusion_alarm;
int garage;
int led_g;
int led_k;
int led_h;
int r,g,b;


/*-------------------------------------------------------------------------Getting Json Data----------------------------------------------*/

successful = Firebase.getJSON(firebaseData, "/");
//  if  (!successful) {
//    return;
//  }
json = firebaseData.jsonData();
json.toCharArray(arr, 1000);
Serial.println(arr);
JsonObject& root = jsonBuffer.parseObject(arr);
if (!root.success()) {
  Serial.println("parseObject() failed");
  return;
}
/*---------------------------------------------------------------------------Getting arduino Data------------------------------------------------*/
char mcudata[32];
int mcudataindex =0;
  Wire.requestFrom(8, 32);
 while(Wire.available()){
  mcudata[mcudataindex] = Wire.read();

   mcudataindex++;
   }


int mcuval[2];
String datamcu = String(mcudata);

strtodata(mcuval, datamcu);
Serial.println("MY MCU");
Serial.println(mcuval[1]);
Serial.println(mcuval[0]);

/*------------------------------------------------------------------------------Led and PIR functionality --------------------
  intrusion_set = root["intrusion_set"];

  led_r2cntrl = root["led_r2cntrl"];
  led_r2mode = root["led_r2mode"];
  pir_r2 = digitalRead(pir_r2pin);
  Serial.println("room 2 pir");
  Serial.println(pir_r2);
  pir_r2status = pirCheck(pir_r2, &timerer2, &timersr2);
  Serial.println("room 2 pir status");
  Serial.println(pir_r2status);

  led_r1cntrl = root["led_r1cntrl"];
  led_r1mode = root["led_r1mode"];
 // pir_r1 = digitalRead(pir_r1pin);
  Serial.println("room 1 pir");
  Serial.println(pir_r1);
 // pir_r1status = pirCheck(pir_r1, &timerer1, &timersr1);

  intrusioncheck(&intrusion_set, &intrusion_alarm, &intrstart, &intrend, &intrusion_buzzer, &intr_logic, pir_r2status);
```

```
ledCheck(&led_r2, led_r2mode, pir_r2status, led_r2cntrl);
ledCheck(&led_r1, led_r1mode, pir_r1status, led_r1cntrl);

led_g = root["led_g"];
led_k = root["led_k"];


/*--------------------------------------------------------------------------------------fan-----------------
temperature = mcuval[0];
humidity = mcuval[1];
fan_tempset = root["fan_tempset"];
fan = root["fan"];
fan_cntrl = root["fan_cntrl"];
fan_temp = root["fan_temp"];

if (fan_tempset == 1) {
  if (temperature > fan_temp) {
    fan = 1;
  }
  else {
    if (fan_cntrl == 1) {
      fan = 1;
    }
    else {
      fan = 0;
    }
  }
}
else {
  if (fan_cntrl == 1) {
    fan = 1;
  }
  else {
    fan = 0;

   }
 }


 /*---------------------------------------------------Fire Alarm-----------------------------------
 fire = analogRead(fire_pin);
 firecheck(&fire_buzzer, &fbtstart, &fbtend, &fire, &fire_alarm, &fire_logic);

/*----------------------------------------------------------Garage--------------------------------
garage = root["garage"];
 if(garage == 1&& garage_done == 0)
  {
   myStepper.step(-800);

   garage_done =1;
   delay(500);
   }
  if(garage ==0 && garage_done == 1){
    myStepper.step(800);
    garage_done = 0;
    delay(500);
    }
    /*----------------------------------------------------------Control hall rgb led --------------

    led_h = root["led_h"];
    if(led_h == 0){
      r=0;
      g=0;
      b=0;

      }else{
        r= root["r"];
        g = root["g"];
        b = root["b"];
        }
```

```
/*--------------------------------------------------Uploading Data------------------------------------------

JsonObject& object = jsonBuffer2.createObject();// data to update
object["pir_r2"] = pir_r2status;
object["led_r2"] = led_r2;
//object["pir_r1"] = pir_r1status;
object["led_r1"] = led_r1;
object["fan"] = fan;
object["fire_alarm"] = fire_alarm;
object["intrusion_alarm"] = intrusion_alarm;
object["temp"] = mcuval[0];
object["humidity"] = mcuval[1];


object.printTo(json_data2);
Firebase.updateNode(firebaseData, "/", json_data2);




/*--------------------------------------------------------Sending data to arduino----------------------------

JsonObject& arduinoData = jsonBuffer3.createObject();
arduinoData["r2"] = led_r2;
arduinoData["r1"] = led_r1;
arduinoData["f"] = fan;
arduinoData["fire"] = fire_buzzer;
arduinoData["intr"] = intrusion_buzzer;
arduinoData["led_g"] = led_g;
arduinoData["led_k"] = led_k;
arduinoData["r"] = r;
arduinoData["g"] = g;
arduinoData["b"] = b;
//  arduinoData["garage"] = garage;


 arduinoData.printTo(json_data);
 String bytes = String(json_data.length());

 sendToArduino(bytes);
 sendToArduino(json_data);

 lcd.setCursor(1,0);
 lcd.print("Temprature: ");
 lcd.print(temperature);
 lcd.setCursor(1,1);
 lcd.print("Humidity: ");
 lcd.print(humidity);


 jsonBuffer.clear();
 jsonBuffer2.clear();
 jsonBuffer3.clear();



}


/*----------------------------------------------------------------------Function to send data-----------------
void sendToArduino(String test) {
  char ch[1000];
  int i;
  for (i = 0; i < test.length(); i += 31) {
    test.substring(i).toCharArray(ch, 32);

    Wire.beginTransmission(8);
    Wire.write(ch);

    Wire.endTransmission();
    Serial.print(ch);
    Serial.println();
  }
}
```

Arduino Code

```
#include <Wire.h>

#include <SoftwareSerial.h>
#include <ArduinoJson.h>
#include <dht.h>

StaticJsonBuffer<120> jsonBuffer;

#define led_r2pin 7
#define fire_buzzer_pin 4
#define fan_pin 3
#define intrusion_buzzer_pin 2
#define dht_pin 5
#define led_r1pin 6
#define led_kpin 9
#define led_gpin 8
#define red 10
#define blue 11
#define green 12

dht DHT;

char mcudata[20];
char mculen[3];
int done = 0;
int no=0;
char data[100];
int index=0;
int bytes = 0;
char bytesbuff[4];
int buffindex = 0;
bool datamark=true;
int garage;
```

```
void setup()
{

 Wire.begin(8);
 Wire.onReceive(receiveEvent);
 Wire.onRequest(requestEvent);


 pinMode(led_r2pin,OUTPUT);
 pinMode(fire_buzzer_pin, OUTPUT);
 pinMode(fan_pin, OUTPUT);
 pinMode(intrusion_buzzer_pin, OUTPUT);
 pinMode(dht_pin, INPUT);
 pinMode(led_r1pin, OUTPUT);
 pinMode(led_gpin, OUTPUT);
 pinMode(led_kpin, OUTPUT);
   pinMode(red, OUTPUT);
  pinMode(green, OUTPUT);
  pinMode(blue, OUTPUT);

 digitalWrite(intrusion_buzzer_pin, LOW);
 digitalWrite(fire_buzzer_pin, LOW);
 Serial.begin(9600);
 int val = DHT.read11(dht_pin);
 String temp = String(DHT.temperature)+';';

 String humidity = String(DHT.humidity)+';';

 String final  = temp+humidity;
 String len = String(final.length());
 Serial.println(final);
 len.toCharArray(mculen,3);
 Serial.println(mculen);
 final.toCharArray(mcudata,20);


 delay(5000);
 }

 void loop(){

 }

 void receiveEvent(int howMany) {

//Serial.println(howMany);
 if(howMany<4&&datamark){

  while(Wire.available()){
    char c = Wire.read();

    bytesbuff[buffindex] = c;
    buffindex++;
    }
    buffindex=0;
    Serial.println(bytesbuff);
    datamark = false;
    bytes = String(bytesbuff).toInt();
   Serial.println("bytesbuff");
    Serial.print(bytes);
    Serial.println();

  }
 else{
 while (Wire.available()) {
    char c = Wire.read();
     data[index]=c;
     index++;
     /* print the character */
  }
```

```
if(index>=bytes){
  jsonBuffer.clear();

  index=0;
  bytes = 0;
  datamark = true;
  no=0;
  Serial.println(data);
  JsonObject& root = jsonBuffer.parseObject(data);

  if(!root.success()){
    //Serial.println("wrong");

    return;
    }
  else{

  int led_r1 = root["r1"];
  int fan = root["f"];
  int fire_buzzer = root["fire"];
  int intrusion_buzzer = root["intr"];
  int led_g = root["led_g"];
  int led_k = root["led_k"];
  int led_r2 = root["r2"];
  int r = root["r"];
  int g = root["g"];
  int b = root["b"];


  analogWrite(red, r);
analogWrite(green, g);
analogWrite(blue, b);


  digitalWrite(led_r2pin,led_r2);
```

```
    digitalWrite(fire_buzzer_pin, fire_buzzer);
    digitalWrite(intrusion_buzzer_pin, intrusion_buzzer);
    digitalWrite(led_r1pin, led_r1);
    digitalWrite(led_gpin, led_g);
    digitalWrite(led_kpin, led_k);

    int val = DHT.read11(dht_pin);
    Serial.println("gsdfd");
    Serial.println(DHT.temperature);
  String temp = String(DHT.temperature)+';';

 String humidity = String(DHT.humidity)+';';
 String final  = temp+humidity;
 String len = String(final.length());
 Serial.println(len);
 len.toCharArray(mculen,3);
 Serial.println(mculen);
 final.toCharArray(mcudata,20);
    }
    }
    else{
      }
      }
 }
void requestEvent(){
    Wire.write(mcudata);
 }
```

RGB Component of Website

```
$(".redbox").click(function(){
  var rRef = firebase.database().ref().child("r");
  var gRef = firebase.database().ref().child("g");
  var bRef = firebase.database().ref().child("b");
  rRef.set(255);
  gRef.set(0);
  bRef.set(0);
  $(".rgbbox").css( "background-color", "red" );

    });

$(".greenbox").click(function(){
  var rRef = firebase.database().ref().child("r");
  var gRef = firebase.database().ref().child("g");
  var bRef = firebase.database().ref().child("b");
  rRef.set(0);
  gRef.set(255);
  bRef.set(0);
  $(".rgbbox").css( "background-color", "green" );

    });

$(".bluebox").click(function(){
  var rRef = firebase.database().ref().child("r");
  var gRef = firebase.database().ref().child("g");
  var bRef = firebase.database().ref().child("b");
  rRef.set(0);
  gRef.set(0);
  bRef.set(255);
  $(".rgbbox").css( "background-color", "blue" );

    });
```

```
 $(".raspberrybox").click(function(){
var rRef = firebase.database().ref().child("r");
var gRef = firebase.database().ref().child("g");
var bRef = firebase.database().ref().child("b");
rRef.set(255);
gRef.set(255);
bRef.set(125);
$(".rgbbox").css( "background-color", "#ffff7d" );

  });

  $(".cyanbox").click(function(){
var rRef = firebase.database().ref().child("r");
var gRef = firebase.database().ref().child("g");
var bRef = firebase.database().ref().child("b");
rRef.set(0);
gRef.set(255);
bRef.set(255);
$(".rgbbox").css( "background-color", "cyan" );

  });

   $(".magentabox").click(function(){
var rRef = firebase.database().ref().child("r");
var gRef = firebase.database().ref().child("g");
var bRef = firebase.database().ref().child("b");
rRef.set(255);
gRef.set(0);
bRef.set(255);
$(".rgbbox").css( "background-color", "magenta" );

  });
```

```
    $(".yellowbox").click(function(){
  var rRef = firebase.database().ref().child("r");
  var gRef = firebase.database().ref().child("g");
  var bRef = firebase.database().ref().child("b");
  rRef.set(255);
  gRef.set(255);
  bRef.set(0);
  $(".rgbbox").css( "background-color", "yellow" );

   });

     $(".whitebox").click(function(){
  var rRef = firebase.database().ref().child("r");
  var gRef = firebase.database().ref().child("g");
  var bRef = firebase.database().ref().child("b");
  rRef.set(255);
  gRef.set(255);
  bRef.set(255);
  $(".rgbbox").css( "background-color", "white" );

   });


var bulb;

database.ref().on("value", function(snap){
  bulb = snap.val().led_h;
  if(bulb == 1){
    $(".Bulb").text("Bulb is on.");
  } else {
    $(".Bulb").text("Bulb is off.");
  }
});
```

```
    $(".buttonone").click(function(){
      var led_hRef = firebase.database().ref().child("led_h");
      if(bulb == 1){
        led_hRef.set(0);
        bulb = 0;
      }else {
        led_hRef.set(1);
        bulb = 1;
      }
    });



  });


</script>
```