A

REPORT

ON

INTELLIGENT IOT FOR HOME AUTOMATION

BY

ABODE DANIEL OHIZIMEDE

EEE/12/9493

SUBMITTED TO

THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

SCHOOL OF ENGINEERING AND ENGINEERING TECHNOLOGY

FEDERAL UNIVERSITY OF TECHNOLOGY, AKURE

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF BACHELOR OF ENGINEERING IN ELECTRICAL AND ELECTRONICS ENGINEERING.

27/10/2017

Certification

This is to certify that this project was carried out by ABODE DANIEL OHIZIMEDE with matriculation number EEE/12/9493, in the department of Electrical and Electronics Engineering. Federal University of Technology, Akure, Ondo State, in partial fulfillment of the award of Bachelors of Engineering (B. Eng.) in Electrical and Electronics Engineering.

……………………………..   ………………………………

ABODE D.O.   DR S.A. Oyetunji

EEE/12/9493   Supervisor

………………………………

DR A.A. PONNLE

H.O.D

Dedication

This Undergraduate Final Year Project is dedicated to God and to my Parents for their enormous support throughout the period of my undergraduate studies.

Acknowledgement

This work would not have materialized without the fatherly guidance of my supervisor DR S.A. Oyetunji who provided all the assistance I needed even with his very busy schedule and commitments. My sincere acknowledgement also goes to Engr Sola of the PCB design LAB for making the LAB available for me when I needed it, and I also thank Miss Ibukun for her guidance. This acknowledgement will not be complete if I do not appreciate my parents Mr. and Mrs. Abode and my siblings for all their support morally and financially during the progress of this work. I really acknowledge their support.

Finally, all praises and adorations belong to God who had spared my life till this time and for his provision for me and for my family.

Table of Contents

LIST OF FIGURES

Abstract

Generally, Home automation is as diverse as the various uses of the home, it involves the control and automation of lightning, heating equipment, ventilation, air conditioning, security, as well as home appliances such as heaters, coffee maker, refrigerators and other home devices. This project inquire into how cloud computing, internet of things, and machine learning can be used to enhance home automation, in order to improve control over the various switches and sockets that control the loads in the house and also to make these devices smarter by making them artificially intelligent, so that they can learn from the way the user interact with them, and use this experience to serve the user better. This project will develop a board prototype of home electrical installation, which will be used for testing and exhibition purposes. To improve the idea of home automation, this project proposes a platform to manage the home automated devices as services accessed via the cloud using google firebase cloud service. This report shows the design of IOT modules that connect the home automated devices to the cloud, it uses the cheap ESP8266 node MCU interfaced with various sensors and relays acting as a gateway between the home and the cloud. It also shows in detail, information about other aspect of the project such as the method used to train a machine learning algorithm that allows the prediction of user interaction with the switches and sockets, prediction of the energy consumption statistics of the user, and also the ambient condition of the home, all these were learnt from sensors data collected during the course of the project. In addition, it shows the development of a mobile application to be used by user to interact with these home electrical installation over the cloud. The result shows that home automation can be achieved at low cost and that IOT and machine learning can help improve the quality of home automation.

CHAPTER ONE

1. Introduction

Imagine how helpful it will be to be able to turn on your air condition few minutes before getting home (Olafusi, 2009), to have the control of all the appliances in your home with you everywhere you go. To be able to control and monitor energy consumption in your home. To know if some lights that should be off are on, even while you are miles away from home. More than this, to have your home devices learn from the way you interact with them in order to serve you better.

Home automation has generally been around for some time in history, since man always wants more convenience. Having being adopted and implemented in different ways, home automation still refers to the act of making the home or better still the home devices smart and easier to control. Home automation ranges from automating lightning, heating, ventilation, air conditioning, refrigerators, sockets and even total control of the electrical installation in the home.

Over the years, different methods have been used in automating the home. Home automation had evolved from the era of wired communication as used in X10, past short distance wireless technology using ZigBee, Bluetooth, Infrared, to enjoying the widespread area of development, Internet of Things (IOT).

With the advent of cloud computing, internet of things, machine learning, and the future of having everything connected to the internet, home automation is taking a new turn. These current trends in technology have enjoyed rapid applications in our daily live and home automation is just one of it.

According to Li et al. (2016) there are three generations of home automation. The first generation uses wireless technology e.g. ZigBee automation. The second generation involve the use of artificial intelligence control of electrical devices e.g. Amazon Echo. Third generation of home automation is in form of a robot buddy who interacts with humans, e.g. Robot Rovio, Roomba.

For this thesis, internet of things and machine learning will be applied to the automation of lighting switches and sockets via a cloud service platform. This thesis also involves the design of a mobile application for monitoring and controlling the home devices over the cloud. In addition, machine learning algorithm will be developed to allow the home devices to learn from experience. A demo of this algorithm will be developed using MATLAB and its performance will be analyze using the same tool. The cloud service will act as a data storage medium, hence it will be the bridge between the user and the home devices.

1.1 Aim

The aim of the project is to design an artificial intelligent home automation system that allow ubiquitous user control and monitoring and have machine learning capabilities using the following technologies; IOT, cloud computing, mobile app development and machine learning.

1.2 Objectives

The objectives of the project include;

1. To design a low cost cloud connected home automation system using google firebase cloud and ESP8266 NodeMCU IOT module, an android mobile application for controlling and monitoring the system and a machine learning algorithm that allows a home automation system to learn from user usage.

2. To implement the embedded system on a printed circuit board and to develop a board prototype that allows for testing and exhibition.

3. To evaluate and test the system by controlling and monitoring it with the mobile app, and to visualize a graphical result of the machine learning process.

1.3 Motivation and Justification

Hitherto, Africa has always been the last place where the latest technologies are used, after years of serving other part of the world. Home automation is already trending in most part of the world but still less used in Africa because of cost of implementation and maintenance. There is of course the need for smart homes in Nigeria due to its enormous advantages. The successful design of this project will mean that advance home automation system can now be easily implemented in Nigeria at the cheapest possible price.

1.4 Problem Statement

There is always a need to be able to control and monitor the various switches and sockets in the house. Although, the use of switches mounted on walls of the home have always been the traditional means of doing this, it is quite limited as users now wish ubiquitous control and monitoring, with the advent of artificial intelligence this capability can even be extended.

1.5 Project Scope and Limitation

The project is complete on its own in remote and ubiquitous control and monitoring of various switches and sockets in the home. In addition, the project provides a mobile application that enhances this control and provide users with visualization of the present states of the switches and sockets. Furthermore, the project extends the traditional means of home automation to include IOT, Cloud computing and Machine learning.

Home automation is generally wide, this project did not use cameras and surveillance, or voice control. Other forms of home automation which include design of smart home appliances, security system, voice recognition, face recognition are not included in this project, the project is also not solving the security challenges of internet of things.

1.6 Expected Contribution to Knowledge

The project will contribute to the knowledge of the design of smart homes and low cost implementation of home automation in Nigeria, making use of the trending technologies, including the use of smart phones which have gained rampart usage by both young and old in the country.

It will contribute to the existing application of IOT, mobile application, cloud computing and machine learning.

1.7 Report Layout

The project report is composed of Six chapters, each covering different sections of the work as summarized below:

- Chapter One gives an introduction to home automation, the project's aim, objectives, motivation, problem statement, scope and expected contribution to knowledge.
- Chapter two covers an extensive literature review on the previous works on home automation systems, the different established standards and protocols, and the different technologies that has been applied over the years.
- Chapter three highlights the project methodology, giving reasons for choice of methods, specific platforms, components and software developed. It also shed light on several aspects of the project design
- Chapter four covers on results and discussion of the results.
- Chapter five includes conclusion and recommendations based on the project work.

CHAPTER TWO

LITERATURE REVIEW

2 Introduction

This literature review will shed light on the different areas of research and past papers related to this project. It will give detail explanation on home automation, how it has evolved and the different technologies used over the years to enhance it. Explanation and description will also be given about the concepts of Internet of Things, Cloud computing, machine learning and how they have been applied to home automation since their inception.

2.1 Home Automation

Home automation may designate an emerging practice of automating household appliances and features in residential dwellings, particularly through electronics means. Home automation system is getting popular and widely used in lots of houses worldwide. It has tons of advantages to users, even more to the handicapped and/or elderly users to whom it will make easier to control their home appliances (Alhaj, 2015).

Home automation can also include the use of various sensors that studies the environment and causes home devices to act accordingly, e.g. A security light turning on automatically in the night, turning off in the day, or the air conditioner turning on or adjusting itself base on the temperature in the room. In addition to inclusion of sensors, home automation can also include some form of convenient control that allow the user holding just a remote control to control virtually anything he points to. Talking about controlling, different technologies has been used to achieve this over the years, ranging from wired to wireless technology (infrared, Bluetooth, Wi-Fi). Now, with the use of the internet and GSM technology, it is possible to control these home devices from virtually anywhere you might be around the world, adding some sort of elegance to home automation.

Another form of home automation, uses some form of scheduling, allowing the user to create schedules for the home devices and leave them to work according to such schedules, example of this is timing a smart socket or timing your electric cooker.

A more elegant form of home automation allows a complete form of interaction between the user and the home, the user can talk to the home and schedule operations using voice control. The home on recognizing this control acts accordingly. For example, imagine a situation where a mother wants to know which room the toddler is, she can just ask the home and the home will reply.

Home automation can improve home experience in different ways;

1. Ease of control over home devices,

2. Ease of monitoring of home devices,

3. A means of power management,

4. Improve Security.

2.1.1 History of Home Automation

Home automation has been around since the world war 1, in fact, the television remote (a simple home automation system) was patented in 1893 (Wikipedia, 2017). Since then different home automation systems have evolved with a sharp rise after the second World War. Its growth has been through various informal research and designs by technology enthusiasts who want a better way of getting things done at home without much effort on their part. The systems evolved from one that can automatically do routine chores like switch on and off security lights, to more sophisticated ones that can adjust lighting, put the television channel to favorite station and control doors. Now in the 21st century, due to the unending growth in technology, home automation has taken a new turn. With the arrival of IoT, machine learning, cloud computing, and their application in voice recognition, face recognition, robotics etc. home automation has enjoyed improvement, as there has been production of devices that self-automate the home, allow the user to communicate with the house through voice control which has now developed into voice discussion instead of just single word command, example of such devices include the Amazon Echo and wink hub 2.

Face recognizing cameras using cloud feature such as google machine learning, allows the home to differentiate its owner from strangers, example of such devices is Nest Cam outdoor. Recently, in 2017, a company started investing in what it calls the robot home, that is fully automated while it is being built.

2.2 Internet of Things (IOT)

Almost everybody knows what the internet is, but the phrase 'Internet of things' IOT for short has been somehow confusing. It can be clarified this way; The internet is the connection of computers together all over the world. The internet of things is a network of objects, sensors and everyday items not normally considered computers, allowing these devices to generate, exchange and consume data with minimal human intervention· The Internet of Things is an emerging topic of technical, social, and economic significance. Projections for the impact of IOT on the Internet and economy are impressive, with some anticipating as many as 100 billion connected IOT devices and a global economic impact of more than $11 trillion by 2025. (Marsan, 2015). Challenges of IOT include; security, privacy, interoperability standard, Legal regulatory, rights and adoption by developing economy.

2.2.1 History of Internet of Things

As of 2016, the vision of the Internet of things has evolved due to a convergence of multiple technologies, including ubiquitous wireless communication, real-time analytics, machine learning, commodity sensors, and embedded systems.

The concept of a network of smart devices was discussed as early as 1982, with a modified Coke machine at Carnegie Mellon University becoming the first Internet-connected appliance, able to report its inventory and whether newly loaded drinks were cold. Mark Weiser's seminal 1991 paper on ubiquitous computing, "The Computer of the 21st Century", as well as academic venues such as UbiComp and PerCom produced the contemporary vision of IOT. However, only in 1999 did the field start gathering momentum. Bill Joy envisioned Device to Device communication as part of his "Six Webs" framework, presented at the World Economic Forum at Davos in 1999.

2.2.2 Internet of Things Communication Models

In March 2015, the Internet Architecture Board (IAB) released a guiding architectural document for networking of smart objects (RFC 7452), which outlines a framework of four common communication models used by IOT devices. Three of this framework are shown in figure 2.1, 2.2 and 2.3.

Device-to-Device communication

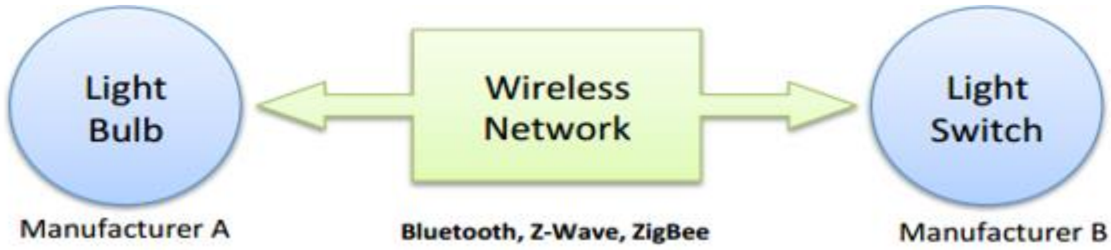

Figure 2.1 Example of device to device communication (Source: Tschofenig, 2015)

Device-to-Cloud Communication

Figure 2.2 Example of device to cloud model (Source: Tschofenig, 2015)

Device-to-Gateway-Model

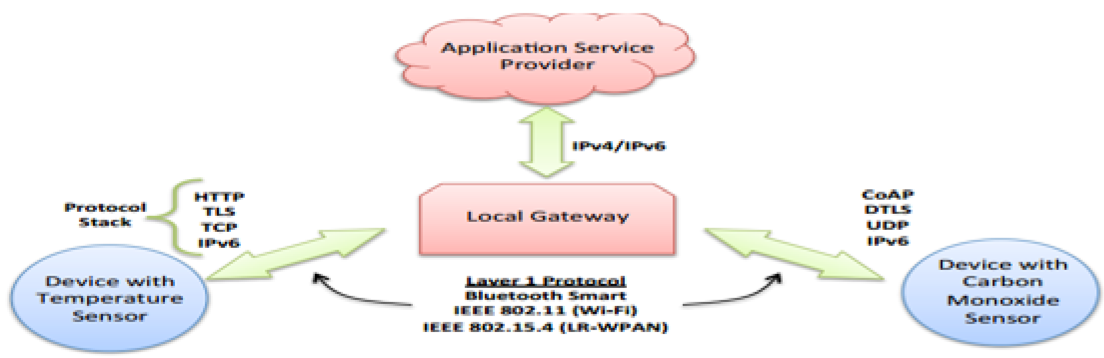

Figure 2.3: Example of device to gateway model (Source: Tschofenig, 2015)

2.2.3 Problem Facing Internet of Things

Security: While security considerations are not new in the context of information technology, the attributes of many IOT implementations present new and unique security challenges.

Privacy: The full potential of the Internet of Things depends on strategies that respect individual privacy choices across a broad spectrum of expectations.

Interoperability / Standards: A fragmented environment of proprietary IOT technical implementations will inhibit value for users and industry.

Legal, Regulatory and Rights: The use of IOT devices raises many new regulatory and legal questions as well as amplifies existing legal issues around the Internet.

2.2.4 Application of IOT to Home Automation

Since the inception of IOT, it has enjoyed application in different fields including home automation. With the application of IOT to home automation, ubiquitous control and monitoring has been possible, devices in the home are beginning to communicate in order to optimize energy consumption. The user can now own the power to control modern home appliances on their smart phones. For example, the SmartThings hub is a stand-alone gateway device that connects to the SmartThings cloud service, allowing the user to gain access to the devices using a smartphone app and an Internet connection.

2.3 Cloud Computing

Cloud computing is the practice of using a network of remote servers hosted on the Internet to store, manage, and process data, rather than a local server or a personal computer. It allows the hosting of complex algorithm, and large data on the internet, thereby reducing task on local computers and IOT devices.

Cloud computing offer several services, notable among them are infrastructure as a service, platform as a service and software as a service. Cloud computing has been extremely useful in enhancing internet of things applications.

2.3.1 History of cloud computing

Cloud computing is a type of Internet-based computing that provides shared computer processing resources and data to computers and other devices on demand. It is a model for enabling ubiquitous, on-demand access to a shared pool of configurable computing resources (e.g., computer networks, servers, storage, applications and services) (Hassan and Qusay, 2011). The origin of the term cloud computing is unclear (Wikipedia, 2017). The word "cloud" is commonly used in science to describe a large agglomeration of objects that visually appear from a distance as a cloud and describes any set of things whose details are not further inspected in a given context (Hassan and Qusay, 2012).

References to "cloud computing" in its modern sense appeared as early as 1996, with the earliest known mention in a Compaq internal document (Regalado 2011). The popularization of the term can be traced to 2006 when Amazon.com introduced its Elastic Compute Cloud.

2.3.2 Cloud Services

A Cloud is essentially a class of systems that deliver IT resources to remote users as a service. The resources encompass hardware, programming environments and applications. The services provided through cloud systems can be classified as follows;

i. **Infrastructure as a Service**

The IaaS is categorized into:

1) Computation as a Service (CaaS), in which virtual machine based servers are rented and charged per hour based on the virtual machine capacity – mainly CPU and RAM size, features of the virtual machine, OS and deployed software.

2) Data as a Service (DaaS), in which unlimited storage space is used to store the user's data regardless of its type, charged per GB for data size and data transfer.

ii. **Platform as a Service**

Platform as a Service (PaaS) cloud systems provide an execution environment that application services can run on. The environment is not just a pre-installed operating system but is also integrated with a programming-language-level platform, which users can be used to develop and build applications for the platform (Furth, 2011).

iii. **Software as a Service**

Software-as-a-Service (SaaS) is based on licensing software use on demand, which is already installed and running on a cloud platform. These on-demand applications may have been developed and deployed on the PaaS or IaaS layer of a cloud platform. SaaS replaces traditional software usage with a Subscribe/Rent model, reducing the user's physical equipment deployment and management costs.

2.3.3 Cloud Providers

Due to the promising future of cloud computing, several companies have started investing in high capacity computing infrastructure in order to provide cloud services. Examples are;

i. Microsoft Azure
ii. Amazon Web Services
iii. Google Cloud Engine
iv. Google Firebase
v. Century Link Cloud
vi. Rackspace Cloud
vii. OpenStack Cloud Services

and the likes.

2.3.4 Problem Facing Cloud Computing

i. performance: Cloud may lack performance in some intensive transaction oriented and other intensive applications. High latency delays may be observed by the users who are at far long distance from cloud.

ii. Security and Privacy

Security is still a major criterion when coming to cloud computing, customers are worried about the attacks which are vulnerable, when information and other important resources are kept outside the firewall. Standard security practices should be done to overcome this problem.

iii. Control

Some IT departments are concerned because cloud computing providers have a full control of the platforms. Cloud computing providers typically do not design platforms for specific companies and their business practices.

iv. Bandwidth costs

With cloud computing, companies can save money on hardware and software; however they could incur higher network bandwidth charges. Bandwidth cost may be low for smaller Internet-based applications, which are not data intensive, but could significantly grow for data-intensive applications.

v. Reliability

Cloud computing still does not always offer round the clock reliability. There were cases where cloud computing services suffered few-hours outages. In the future, we can expect more cloud computing providers, richer services, established standards, and best practices.

2.4 Machine Learning

Arthur Samuel describe machine learning as the field of study that gives computer the ability to learn without being explicitly programmed. Over the past two decades Machine Learning has become one of the mainstays of information technology and with that, a rather central, although usually hidden, part of our life. With the ever increasing amounts of data becoming available there is good reason to believe that smart data analysis will become even more pervasive as a necessary ingredient for technological progress (Smola and Vishwanathan,2008).

2.4.1 History of Machine Learning

As a scientific endeavor, machine learning grew out of the quest for artificial intelligence. Already in the early days of AI as an academic discipline, some researchers were interested in having machines learn from data (Wikipedia, 2017).

Machine learning, reorganized as a separate field, started to flourish in the 1990s. The field changed its goal from achieving artificial intelligence to tackling solvable problems of a practical nature. It shifted focus away from the symbolic approaches it had inherited from AI, and toward methods and models borrowed from statistics and probability theory (Langley and Pat, 2011). It also benefited from the increasing availability of digitized information, and the possibility to distribute that via the Internet

2.4.2 Types of Machine Learning Problem

As Stated by Russell (2003), Machine learning tasks are typically classified into three broad categories, depending on the nature of the learning "signal" or "feedback" available to a learning system. These are;

Supervised learning: The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs.

Unsupervised learning: No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

Reinforcement learning: A computer program interacts with a dynamic environment in which it must perform a certain goal (such as driving a vehicle or playing a game against an opponent). The program is provided feedback in terms of rewards and punishments as it navigates its problem space.

2.4.3 Algorithm use in solving Machine Learning Problem

As the field of machine learning started enjoying several applications, different algorithms has been developed to generate the best hypothesis for wide range of learning problems, example of such algorithms are Linear Regression, Logistic Regression, Neural Network, Support Vector machines, Clustering and the likes. This algorithm fits an application depending on whether it is a classification or regression problem. Other factors to be considered in choosing a learning algorithm are; Number of input features, number of input units, efficiency of the algorithm and the likes.

2.4.4 Application of Machine Learning

Machine learning as found application in several fields, examples are; Adaptive websites, Affective computing, Bioinformatics, Brain-machine interfaces, Computer vision, including object recognition, Game playing, Natural language processing, Natural language understanding, Online advertising, Speech and handwriting recognition, Financial market analysis, User behavior analytics.

2.4.5 Challenges Facing Machine Learning

Some of the key challenges of applying machine learning to a problem are;

i. problem of choosing enough input features,
ii. problem of sourcing enough input units,
iii. Problem of selecting the best algorithm,
iv. Problem of data storage etc.

CHAPTER 3

METHODOLOGY

3. Introduction

There are several methods of designing a home automation system and also there are several platforms that can be use as shown in the previous chapter. This chapter contains the methods and choices that were made on the type of platform, hardware component, mobile app development platform, cloud service provider and the machine learning algorithm use to realize this project. Before the actual design of the project work, specific deliberate choices were made on the type of platform, hardware component, mobile app development platform, cloud service provider and the machine learning algorithm to be applied. These choices were made bearing in mind their feasibility, low cost, availability, reliability, flexibility, simplicity and efficiency.

3.1 Selection of Implementation Platform

In the last chapter, seven different platforms over which home automation system can be implemented were discussed. Most of the platforms are outdated and are no longer in use. Due to low cost, simplicity, reliability and widespread usage, Wi-Fi has been selected as the platform for the home automation system in this project. Wi-Fi network is available both in the developed and developing nations, and has gained popularity due to high speed of data transfer. Wi-Fi enables local area network, wide area network and internet.

3.1.1 Selection of Hardware Components

Each platform has a set of hardware components over which it can be implemented, there are several Wi-Fi modules that can allow the IOT project to access Wi-Fi network, examples are NL6621-Y1 2.4G Wi-Fi transceiver module, ESP-01 Wi-Fi module, ESP8266 Wi-Fi Module. ESP8266 was chosen due to low cost, availability, portability, low power consumption and more importantly it is embedded with Tensilica L106 32-bit micro controller (MCU), so that in addition to being a Wi-Fi module it is also a microcontroller that can be programmed. In addition, a 10A, 240V ac relay was chosen as the actuator, the system also includes sensor network that includes, PIR motion sensor, DHT11 temperature and humidity sensor and also the MAX44009 light sensor.

3.1.2 Selection of Mobile App Development Platform

Selection of mobile app development platform depends on the mobile phone operating system, and ease of use of the platform, example of mobile phone operating system include Microsoft, Android, iOS. Due to wide spread usage, the Android platform was chosen. There are several android app development environments such as, android studio, python IDE, MIT app inventor etc. MIT app inventor was chosen because of its ease of use in building an android app and it also contains all the features we need to build a successful android mobile application for the home automation system.

3.1.3 Selection of Cloud Service Provider

As stated in the previous chapter, there are several cloud service providers, such as Microsoft Azure, Amazon Web Services, Google Cloud Engine, Google Firebase Cloud, Century Link Cloud, Rackspace Cloud, OpenStack Cloud Services and the likes. The google firebase cloud was chosen because of its supports for IOT devices, it is easy to use with the ESP8266 module, and it is supported by MIT app inventor, so that a developer can create app that access the cloud services.

3.1.5 Selection of Machine Learning Algorithm

There are several machine learning algorithms which include support vector machine, logistic regression, naïve bayes, kernel approximation, neural network etc. We chose the neural network algorithm because it can solve both regression and classification problem which the project entails and also it is available as a toolbox in MATLAB.

3.2 System Design

The system designed for the home automation project consist of complex integration of both hardware and software somewhat call embedded system and an android mobile phone application that allows a user to exert both control and monitoring over the embedded system. The system also includes the use of google firebase cloud which serve as a database, and also as a sort of bridge between the mobile application and the embedded system installed in the home. The interaction between the various parts of the system is shown figure 3.1.

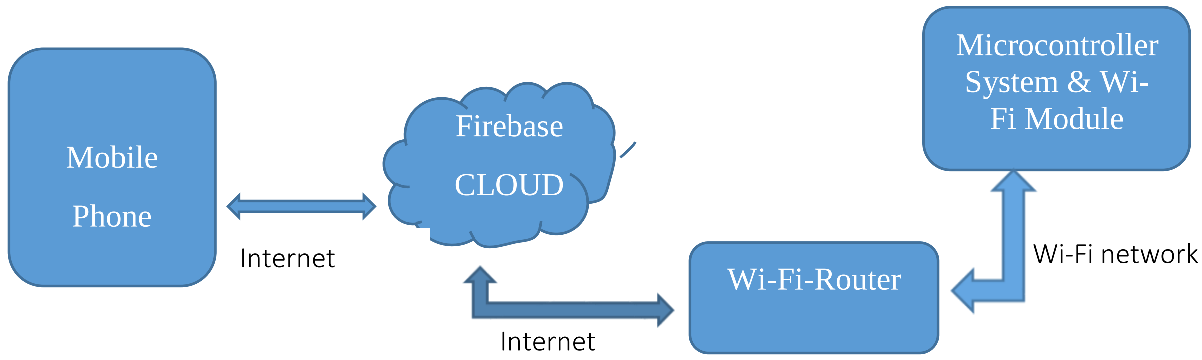

Figure 3.1 Overview block diagram of the

3.2.1 System Block Diagram

A more explicit diagram of the system is shown below;

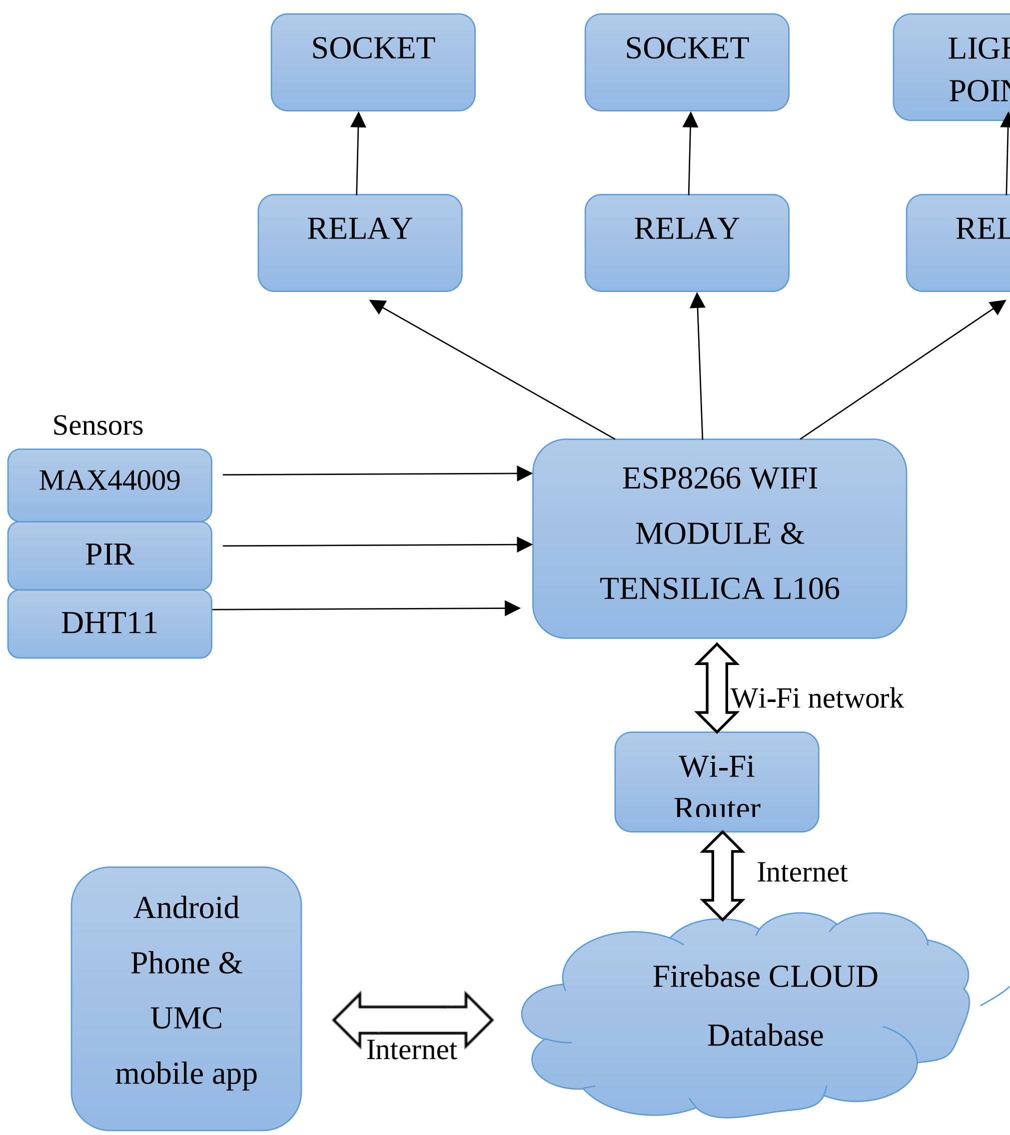

Figure 3.2 Detailed block diagram of the system

3.2.1 ESP8266 NodeMCU

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC (System On Chip) from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the dev kits. The firmware uses the Lua scripting language. ESP8266EX also integrates an enhanced version of Tensilica’s L106 Diamond series 32-bit processor, with on-chip SRAM, besides the Wi-Fi functionalities.

A typical ESP8266 NodeMCU module has 30 pins, it has 17 GPIO pins which can be used for digital, PWM (pulse width modulation), one-wire and I2C (Inter Integrated Circuit) communication. It has a 3.3V pins, pins for ground connection, 3 set of serial communication pins, an analog input pin, and the MOSI, SCLK, MISO pins for SPI communication. A picture of the nodeMCU and its pin mapped out is shown in figure 3.3.

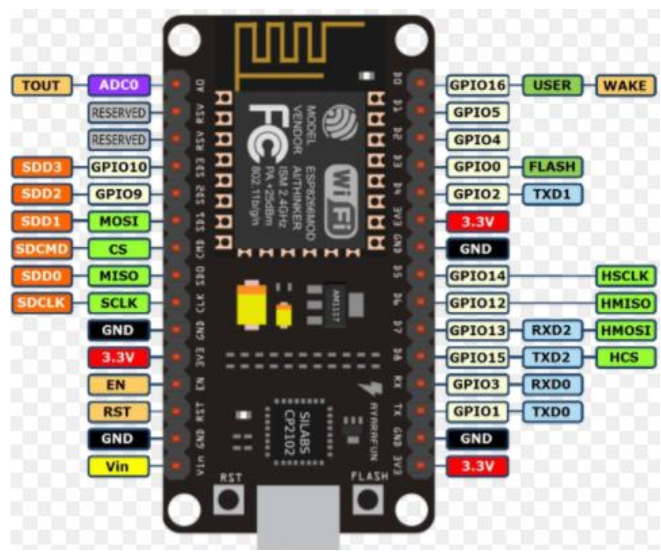

Figure 3.3 NodeMCU and its pin map-out diagram

3.3.1 ESP8266 Architecture

The ESP8266 is an SoC 32 pins QFN package with a typical working voltage of 3.3V, it exhibits 3 sleep modes (Modem sleep, light sleep, deep sleep) in order to enhance good functionalities while saving power. The ESP8366 integrated circuit pin label is shown in figure 3.4.

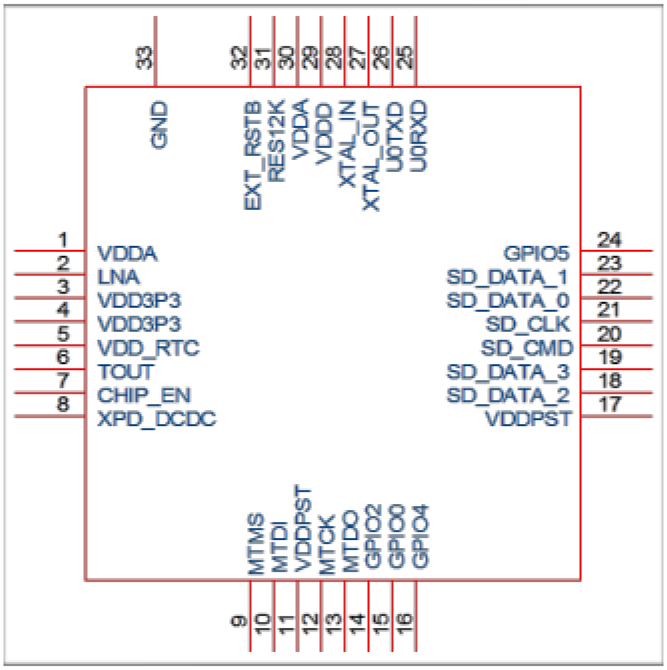

Figure 3.4 ESP8266 and its pin map-out diagram

Features:

- 802.11 b/g/n
- Integrated low power 32-bit MCU
- Integrated 10-bit ADC
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLL, regulators, and power management units
- Supports antenna diversity
- WiFi 2.4 GHz, support WPA/WPA2
- Support STA/AP/STA+AP operation modes
- Support Smart Link Function for both Android and iOS devices

• SDIO 2.0, (H) SPI, UART, I2C, I2S, IR Remote Control, PWM, GPIO
• STBC, 1x1 MIMO, 2x1 MIMO
• A-MPDU & A-MSDU aggregation & 0.4s guard interval
• Deep sleep power <10uA, Power down leakage current < 5uA
• Wake up and transmit packets in < 2ms
• Standby power consumption of < 1.0mW (DTIM3)
• +20 dBm output power in 802.11b mode
• Operating temperature range -40C ~ 125C
• FCC, CE, TELEC, WiFi Alliance, and SRRC certified

ESP8266EX is among the most integrated Wi-Fi chip in the industry; it integrates the antenna switches, RF balun, power amplifier, low noise receive amplifier, filters, power management modules as shown in figure 3.5, it requires minimal external circuitry, and the entire solution, including front-end module, is designed to occupy minimal PCB area.

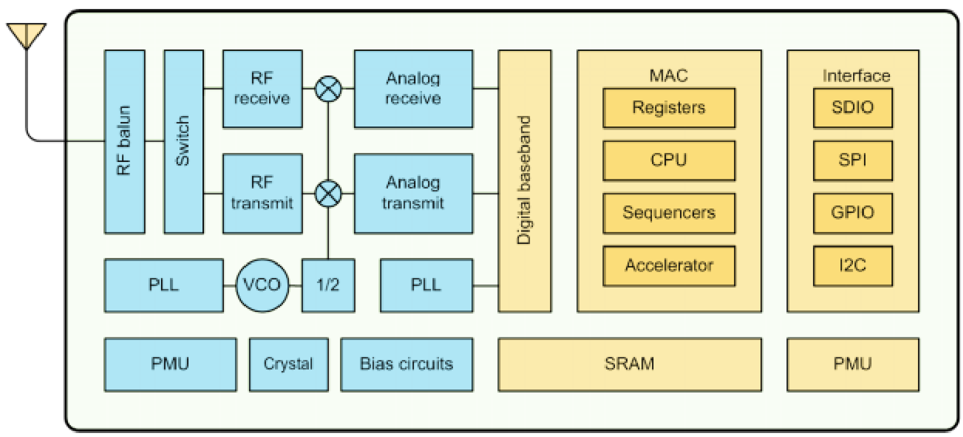

Figure 3.5 ESP8266 internal block diagram

3.3.2 ESP8266 Wi-Fi Characteristics

The ESP8266 radio contains a 2.4GHz receiver, 2.4GHz transmitter, High speed clock generators and crystal oscillator, Real time clock, Bias, regulators and Power management. The transceiver supports 14 channels between 2412MHz and 2484MHz frequency range. The 2.4GHz receiver down-converts the RF signal to quadrature baseband signals and converts them to the digital domain with 2 high resolution high speed ADCs. The 2.4GHz transmitter up-converts the quadrature baseband signals to 2.4GHz, and drives the antenna with a high powered CMOS power amplifier.

The Wi-Fi characteristics of the ESP8266 is shown in table 3.1;

Table 3.1 Wi-Fi characteristics of ESP8266

Categories	Items	Values
WiFi Parameters	Certificates	FCC/CE/TELEC/SRRC
	WiFi Protocols	802.11 b/g/n
	Frequency Range	2.4GHz – 2.5GHz
	Tx Power	802.11 b: +20dBm
		802.11 g: +17dBm
		802.11 n: +14dBm
	Rx Sensitivity	802.11 b: -91dBm (11 Mbps)
		802.11 g: -75dBm (54Mbps)
		802.11 n: -72dBm (MCS7)
	Types of Antenna	PCB Trace, External, IPEX Connector, Ceramic Chip

3.3.3 Tensilica L106 MCU Architecture

The Tensilica L106 MCU is a low power microcontroller, consumes about 75microW/MHz for a 7 stage pipeline. It is special microcontroller that carries a user configurable and extensible processor core. It uses the xtern instruction set architecture which is a new post-RISC ISA. Its architecture is shown in figure 3.6 and it generally includes;

Processor Configuration

- Power usage: 76microW/MHz, 47microW/MHz (5 and 7 stage pipeline)
- Clock Speed: 359MHz, 400MHz (5 and 7 stage pipeline)

- Cache: up to 32KB
- 64 general purpose registers
- Extensible via use of TIE and FLIX instructions
- 32-bit ALU

Registers

- 32-bit general purpose register file
- 32-bit program counter
- 16 optional 1-bit Boolean registers
- 16 optional 32-bit floating point registers
- 4 optional 32-bit MAC16 data registers
- Optional Vectra LX DSP registers

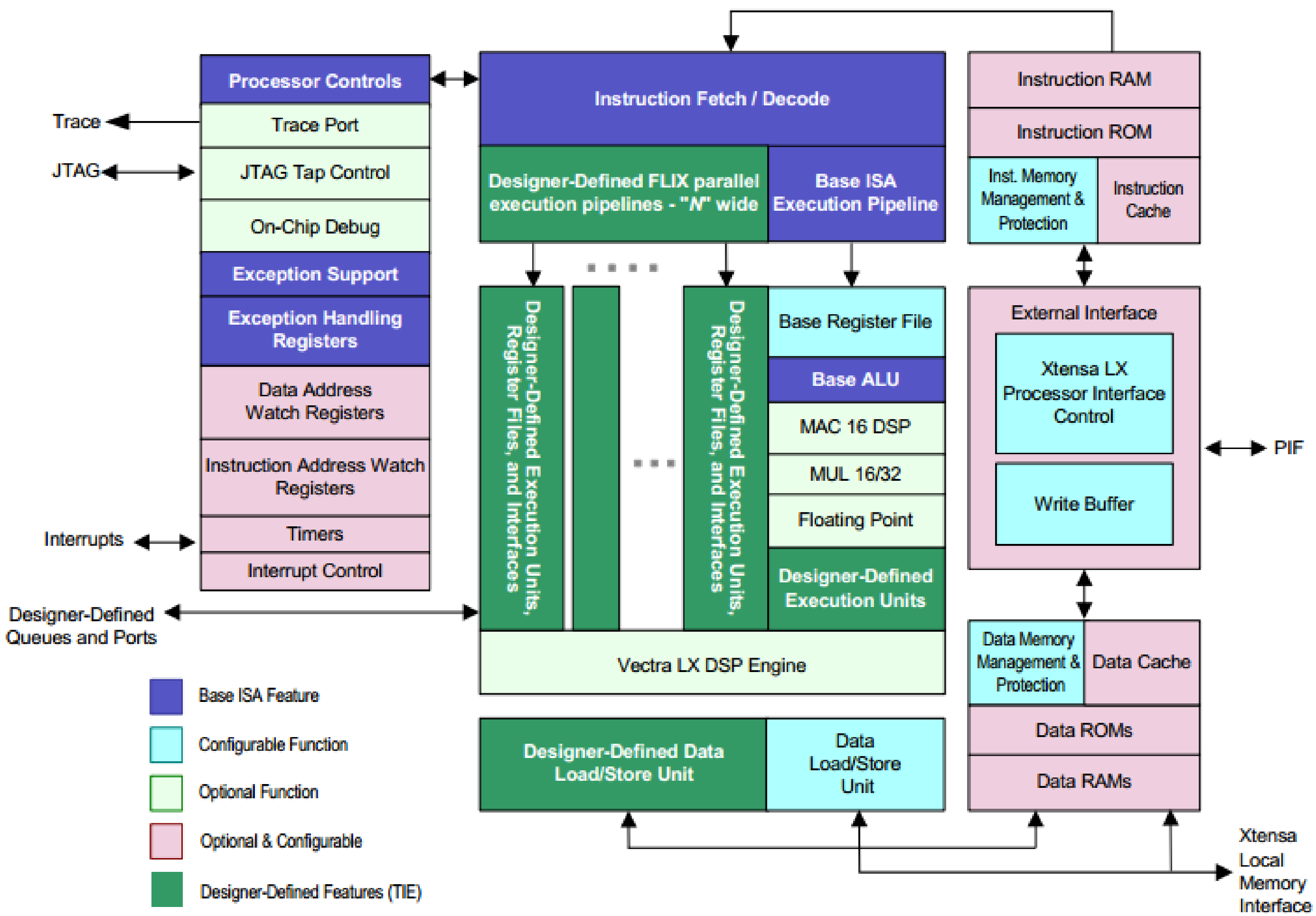

Figure 3.6 Tensilica L106 MCU architecture

3.3.4 ESP8266 Programming

The official firmware for ESP8266 NodeMCU was developed with LUA script, but some creative mind developed the Arduino core for ESP8266 enabling the ESP8266 to be programmed with the Arduino IDE using C/C++. The Arduino IDE is a free integrated development environment which has gained popularly usage.

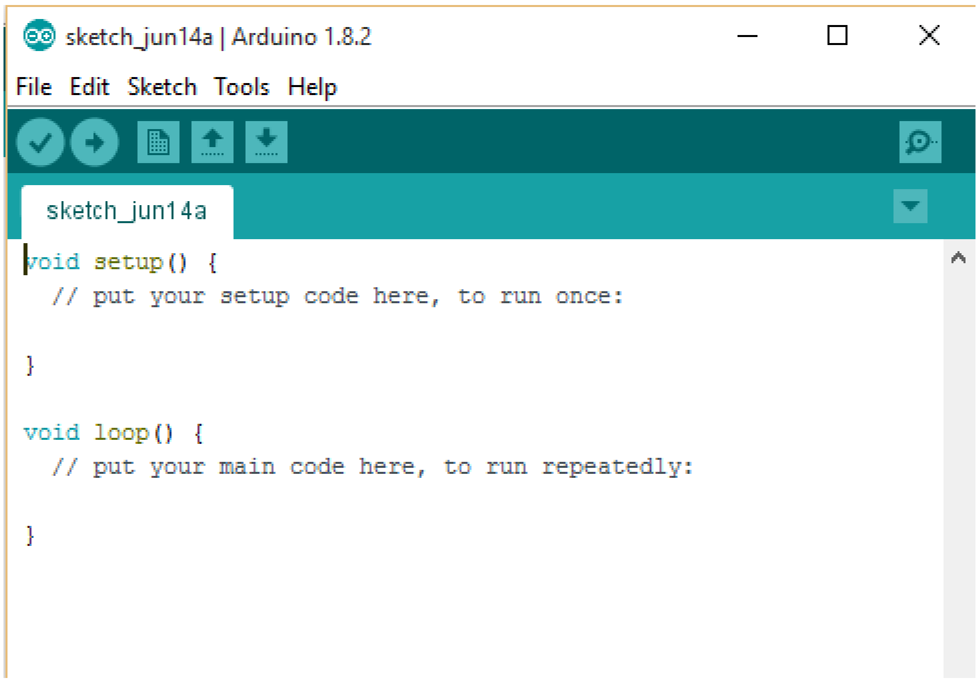

Figure 3.7 The Arduino IDE

3.3.5 Sensor Network

The home automation system also includes some sensors that senses the ambient condition of the home in order to provide a form of ambience intelligence, the data from these sensors are read and stored in the cloud by the ESP8266 NodeMCU. These data are made available to the user on the mobile app. In addition, these data will be fed to a machine learning algorithm hosted in the cloud to serve an artificial intelligence purpose. The ambient condition to sensed include, light intensity, humidity, temperature, human presence, etc. The sensors that was used includes the DHT11 temperature and humidity sensor, MAX44009 light sensor, and the PIR motion sensor.

3.4 Hardware Design

The embedded system design is a prototype of the part of the project that will be installed in the home, this prototype contains the hardware components of the project. A board casing was made that housed both the prototype, 2 sockets and a lighting point for testing and exhibition purpose.

3.4.1 Components and Tools Used

Components

ESP8266 NodeMCU, MAX44009 Light Sensor, PIR motion sensor, DHT11 humidity and temperature sensor, 3 x 5V relay module, 2 x LED, 2 x 1k resistors, 2.5V power supply module, PCB board, 2 x Sockets, Lamp Holder, plug, 60W bulb, Wires, Wooden board.

Tools

Soldering Iron, Breadboard, Screw Drivers, Wood Saw, Cutter, PCB drill, Laptop Computer

Software Tools

Arduino IDE, Proteus

3.4.2 Circuit Design of the Embedded System

The circuit was designed using proteus and it is shown in figure 3.8. There was no simulation model for most of the components, hence simulation and testing of the project was done on a breadboard before it was transferred to a PCB board.

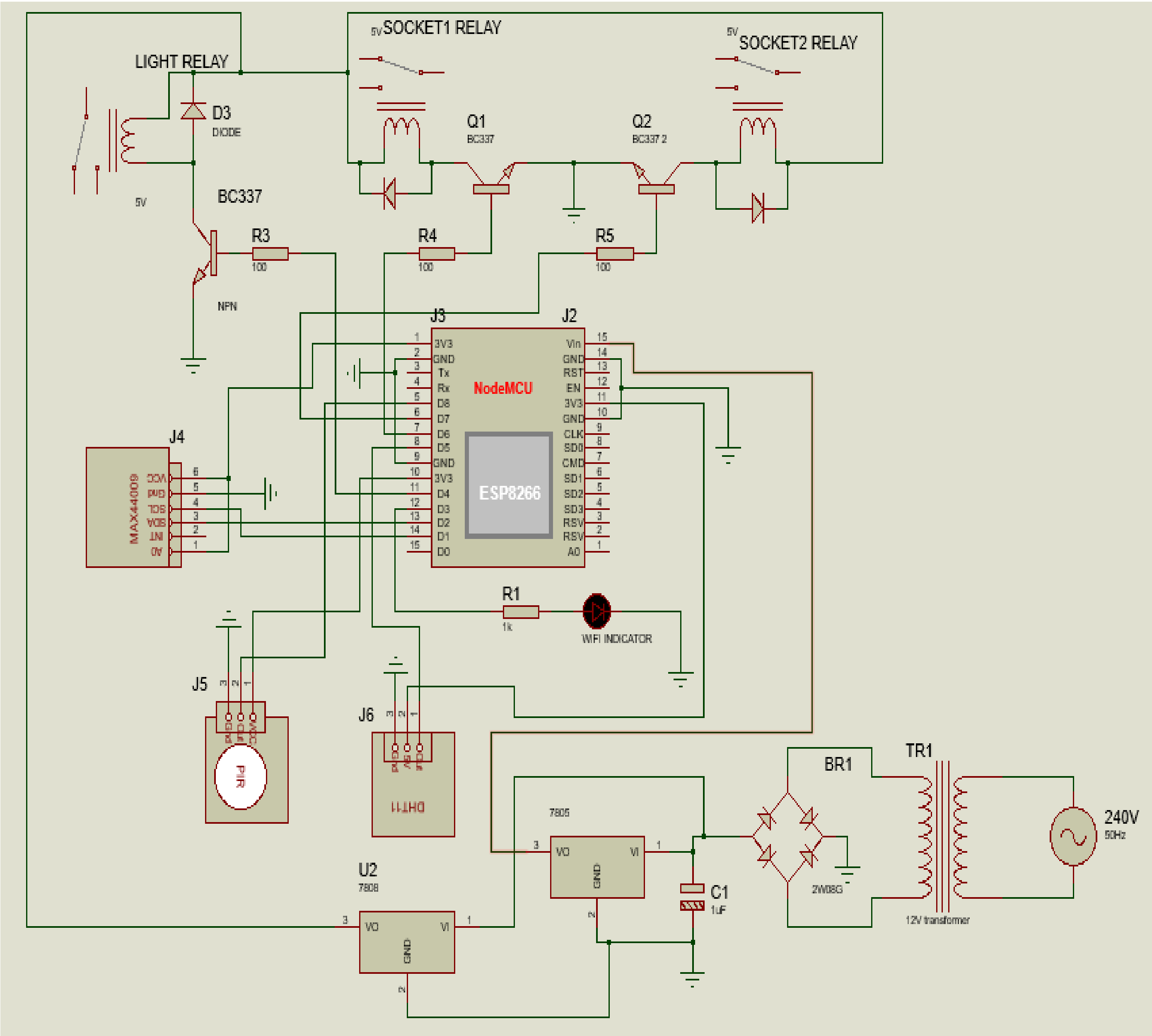

Figure 3.8 Hardware system circuit diagram

3.4.3 Current requirement and power supply design of the project

The total current requirement of the project is the sum of power needed to power all the components involved. The ESP8266 nodeMCU consume a maximum current of 170mA, MAX44009 light sensor consumes 0.65uA operating current, PIR motion sensor draws a maximum current of 100uA in full operation, DHT11 draws 0.1mA, this indicate that the sensors are low current components. This leaves us with the relays, their current consumption is shown in calculation below.

$$I = \frac{V}{R} \qquad (3.1)$$

The relays used are 6V relays, since 6V regulators are not currently available, we decide to power the relay with 8V. The resistance of the relay was measured with a multimeter gave us 110ohms. hence;

$$I = \frac{8}{110}$$

$$I = 0.073A$$

Each Relay will consume 73mA of current maximum, 3 relays will consume 3*73 = 218mA

Table 3.2: Major components and current requirement

Component	Current Requirement (mA)
ESP8266 NodeMCU	170
MAX44009	0.00065
PIR motion sensor	0.1
Relays x3	218
Total	436.1

Hence, we used a 600mA 12V step down transformer. There will be additional loss in power due to the efficiency of the full wave bridge rectifier that will be used. The efficiency of a full wave rectifier is derived below;

$$instantaneous\ current\ i = \frac{v}{rf+Rl} = \frac{VmSin\theta}{rf+Rl} \qquad (3.2)$$

Where rf is the diode resistance and Rl is load resistance.

$$I_{dc} = \frac{2I_m}{\pi} \qquad (3.3)$$

$$dc\ power\ output\ P_{dc} = I_{dc}^2 * \ R_l = \left(\frac{2I_m}{\pi}\right)^2 * R_l \qquad (3.4)$$

Ac input power is given by;

$$P_{ac} = \ I_{rms}^2\left(r_f + \ R_l\right) \qquad (3.5)$$

For a full wave rectification wave, we have;

$$I_{rms} = \frac{I_m}{\sqrt{2}} \qquad (3.7)$$

$$P_{ac} = \left(\frac{I_m}{\sqrt{2}}\right)^2 \left(r_f + \ R_l\right) \qquad (3.8)$$

Full wave rectification efficiency is;

$$\in = \frac{P_{dc}}{P_{ac}} = \frac{\left(\left(\frac{2I_m}{\pi}\right)^2 * R_l\right)}{\left(\frac{I_m}{\sqrt{2}}\right)^2 \left(r_f + R_l\right)} \qquad (3.9)$$

$$\in = 0.812 \frac{R_l}{r_f + R_l} = \frac{0.812}{1+\frac{r_f}{R_f}} \qquad (3.10)$$

If rf is negligible compared to Rl then we have maximum efficiency to be

Maximum efficiency = 81.2%

The means that only 81.2% of the power supplied by the transformer will be available at the output of the rectifier.

The ESP8266 nodeMCU requires 5V input for operation and the relays require 6V for operation, because of this we made use of two voltage regulator, a 5V voltage regulator 7805 and 8V voltage regulator 7808.

The complete circuit of power supply is shown in fig 3.9.

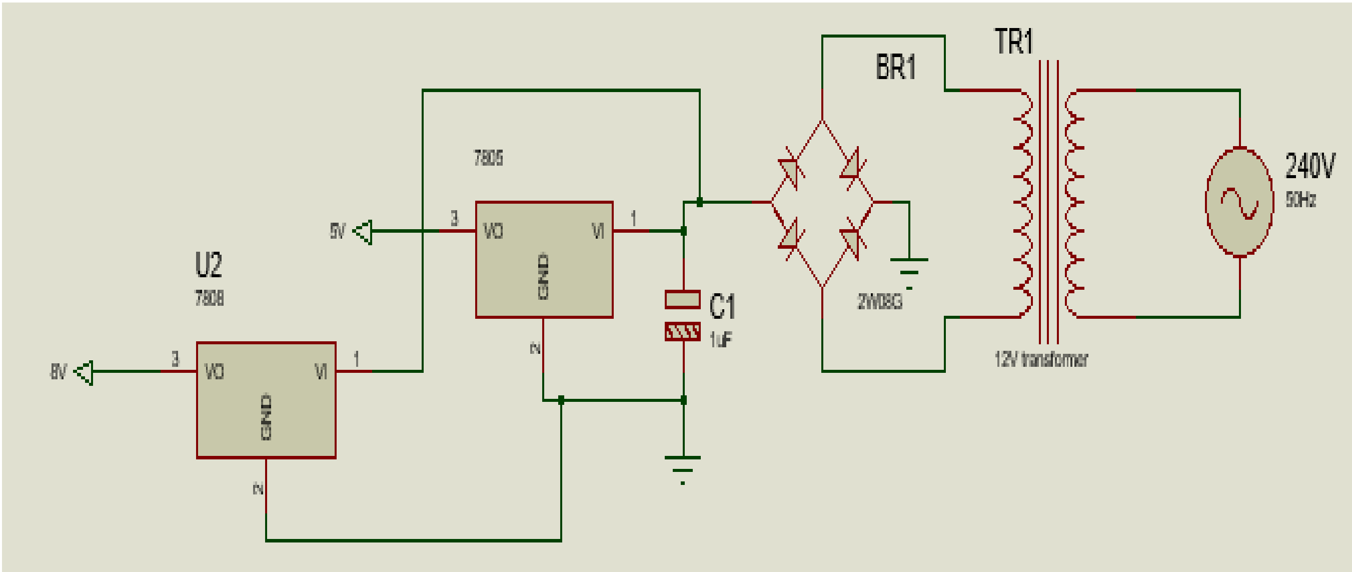

Figure 3.9 Power supply circuit diagram

3.4.4 Programming the ESP8266

The ESP8266 was programmed with the Arduino IDE, the Arduino IDE was configured as shown in session 3.3.4. The program code implemented was developed stepwise as shown below; the complete code is available on appendix 1.

- Include the required header files, which are Wire.h, MAX44009.h, dht11.h, ESP8266WiFi.h, FirebaseArduino.h
- Declare objects of the header files
- Define the firebase credentials which are FIREBASE_HOST and FIREBASE_AUTH
- Define the WIFI login details
- Declare the pins and other variables
- Setup the pins functions, Wi-Fi and Firebase
- Create a function that get the state of the switches from firebase and returns acknowledgements
- Create a function that read the light intensity
- Create a function that reads human presence
- In the void loop, call the required functions, read temperature and humidity from the DHT11, update the value of the ambient data on firebase.

3.4.5 Printed Circuit Design (PCB) of the project.

The printed circuit board design of the project was done following the steps below;

A new circuit was designed on proteus with connectors and terminals replacing the components of the project without a PCB simulation model. Then the PCB design was done with ARES. The design is shown in figure 3.10.

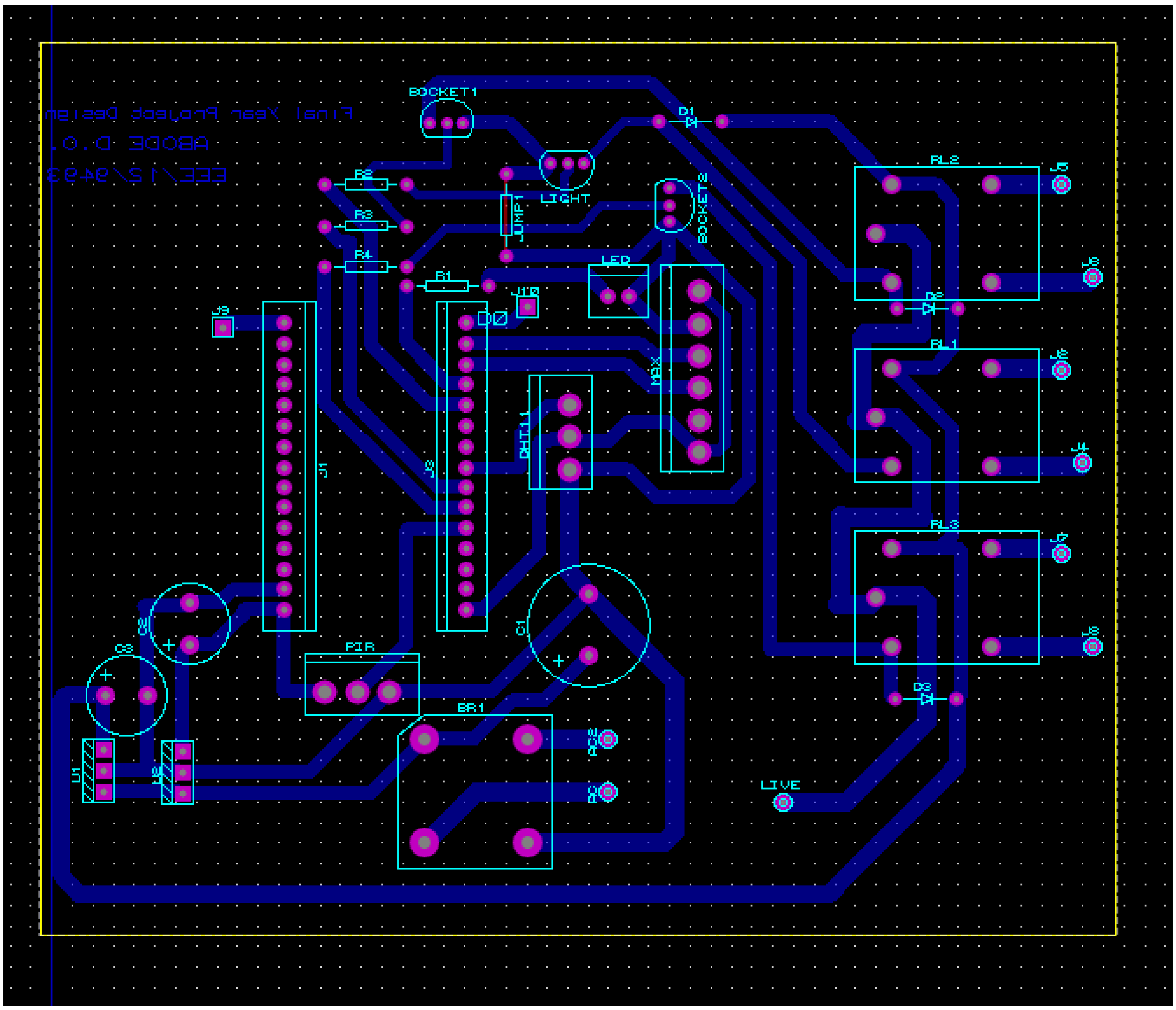

Figure 3.10 Printed circuit board design

The 3D view of the design was also developed using the 3D view on proteus as shown in figure 3.11.

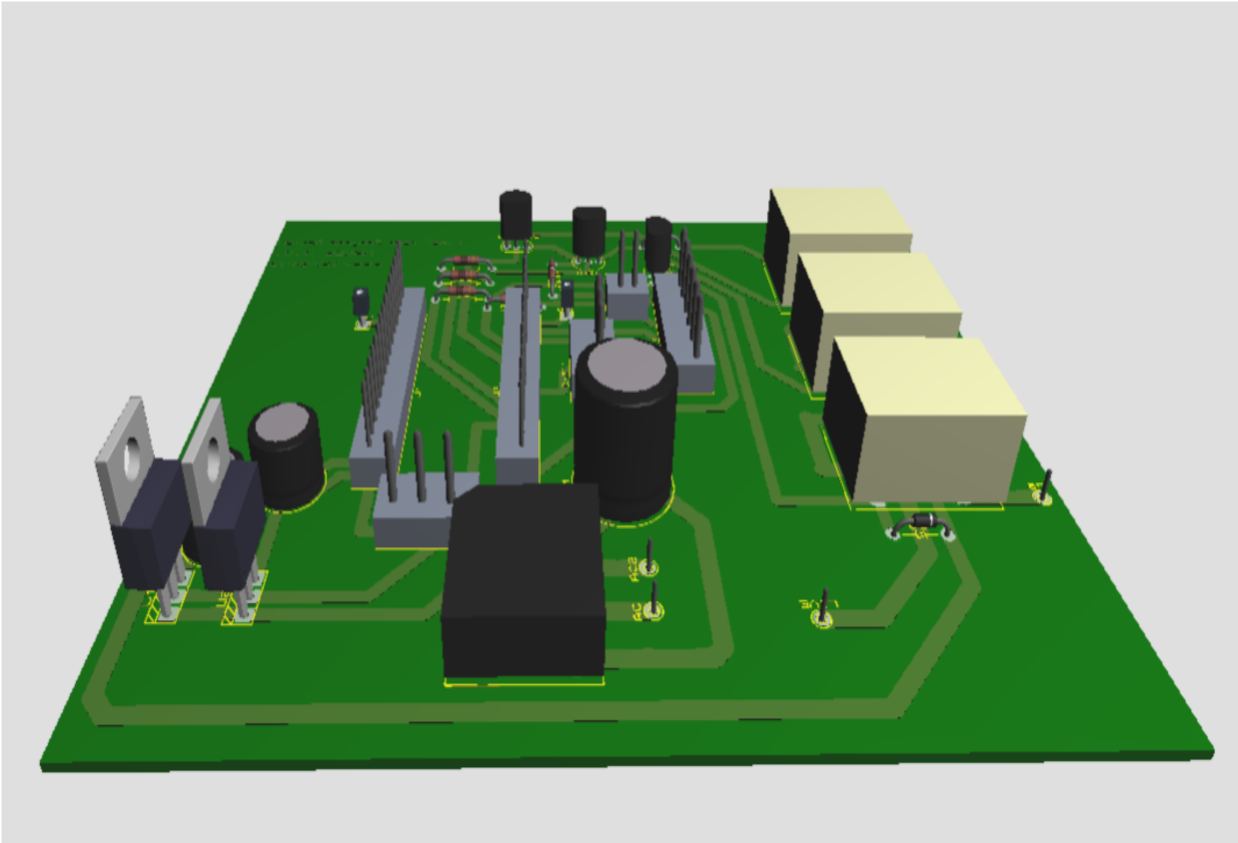

Figure 3.11 3D diagram of Printed circuit board

Next, the design was printed on a glossy paper and with a hot pressing iron, it was transferred to a copper board. The copper board was etched with a solution of iron chloride. After the etching, through holes were drilled where appropriate and the components were transferred to the board and soldered.

3.4 Android Mobile Application Design

Android application can be developed with different integrated development environment, such as B4U IDE, android studio, visual studio, MIT app inventor platform, etc. The MIT app inventor provide contains everything needed to build the mobile application for the project. The project android app was called Ubiquitous Monitor and Controller (UMC).

3.4.1 MIT App Inventor Platform

MIT app inventor is an online platform for developing mobile application, it allows android developer to develop android apps and also to build it into a .apk extension. It provides both the GUI design environment and also the coding environment. It also provides an emulator that allows the user to view the app in real time as it is been developed.

3.4.3 UMC Android App Design

MIT app inventor platform was use in creating the UMC app. In creating the app, a new project was chosen and named Home automation app.

3.4.4 Designing the Graphical User Interface

The UMC app uses a single page interface, the graphical user interface makes use of some visible components such as horizontal bars, vertical bars, buttons, labels, list pickers and also some invisible components such as firebaseDB, tinyDB, Clock. The interface is shown figure 3.12.

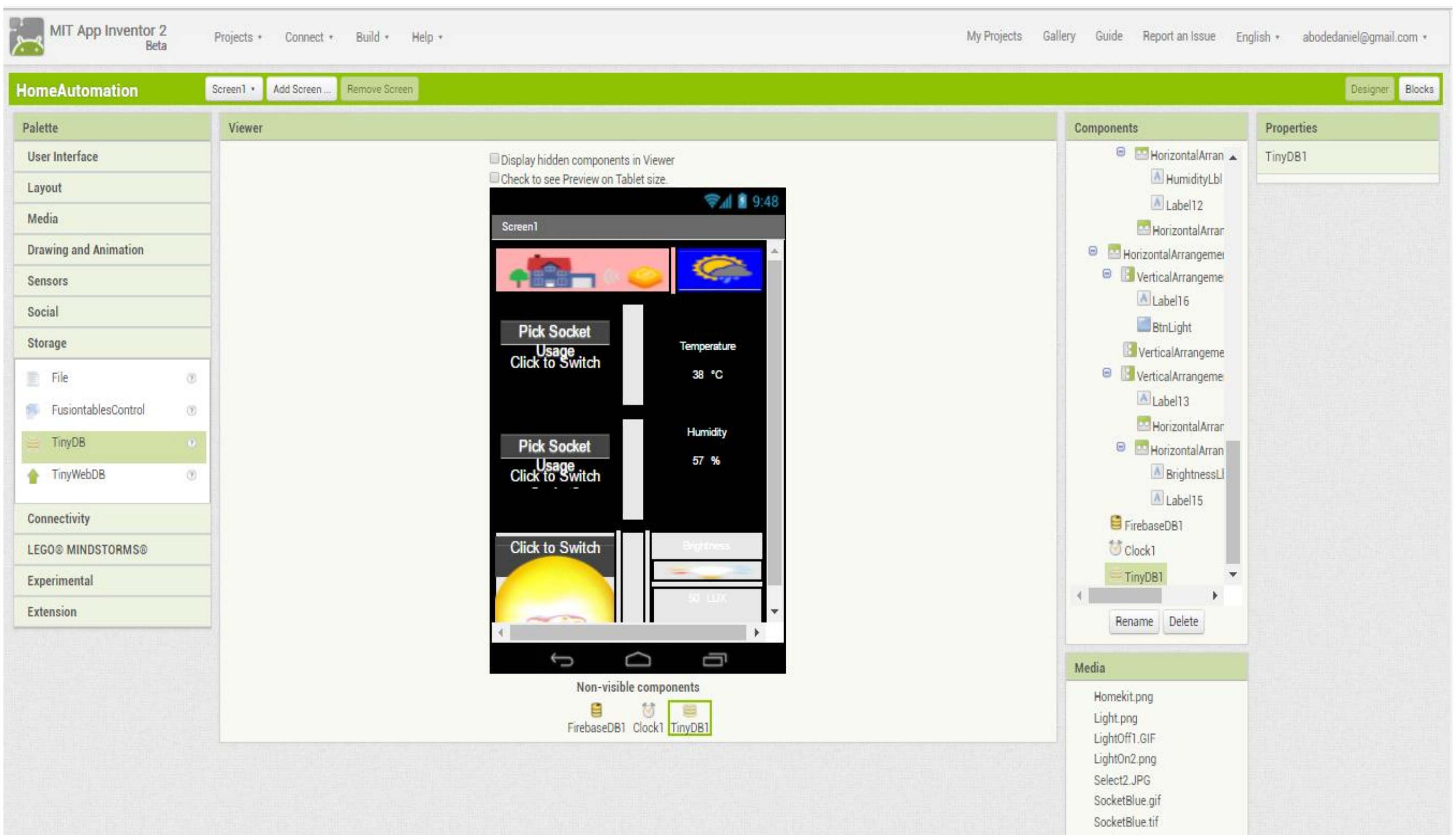

Figure 3.12 App development GUI design

3.4.5 Programming the Android App

The steps followed to program the app is shown below

- When screen initializes get the value stored in the tinyDB and use it to sets the pictures on the buttons
- When button socket1 is clicked, change the state of socket1 in firebase database, wait for acknowledgement to change the picture on the button.
- When button socket2 is clicked, change the state of socket2 in firebase database, wait for the acknowledgement to change the picture on the button.
- When button Light is clicked, change the state of Light in firebase database, wait for acknowledgment to change the picture on the button.
- When clock ticks, check for the ambient condition values on firebase database and updates the text on the corresponding label.

3.5 Google Firebase Cloud Service

3.5.1 Google Firebase Cloud Service

Firebase is a real time database in the cloud that provides an API to store and sync data in real time. It offers a several cloud services such as, cloud storage, cloud functions, database, hosting, authentication etc. These project will be using two of firebase services (cloud functions and database). The database will be used to store the sensor data from the home and also the state of the switch.

3.5.2 Creating the Cloud Application

For the Home automation project, the google firebase was used both as a database and also as a platform for hosting the machine learning functions. These process was accomplished with the following steps.

Step 1: A firebase project was created and named “Home automation project”

Step 2: On clicking the tab “Home automation project”, we were able to explore the cloud services provided by firebase.

Step 3: From the database tab, we click on Rules and change the security rules so that the embedded system and the mobile app will be able to read and write to the database.

Step 4: From the database tab, we added the fields – HomeData> States. We created the tag that would hold the value for the different data, including the switch states, the sensor data and acknowledge value from the ESP8266 when controlled.

With steps above, we created the firebase database needed for the cloud functionalities of the home automation project, the function of the database is to store data in the tags created and make it available for the user, the ambient data are updated every five seconds while the data that stores the switch states changes with the user interaction with the UMC android app.

3.6 Machine Learning Algorithm Development

The machine learning algorithm will be developed in MATLAB to show the possibility of a home installation learning from the user and to provide visualization tools that can help us to visualize the performance of the algorithm. To generate data for this algorithm we design two modules that is attached to two sockets to log the data offline on an SD_CARD. This data was logged every 2 minute for 30 days on 2 sockets being used in an office and a light source.

3.6.1 Description of data collected

Day type: day type distinguish weekend from work day, 1 was chosen for weekend and 2 for week day and it was label DT.

Day of the week: day of the week helps to label each day in the week, Monday was counted as 1 up till Sunday which was labelled 7. The label for this data is DOW.

Hour of the day: Hour of the day was labelled 0-24 for the 24-hour clock.

Minute of the hour: Minute of the way was logged using number 0-60 to cover the 60 min in an hour range. DS1302 RTC module was used to perform the real time clock function.

Human presence: Human presence was detected with the use of a PIR motion sensor, it outputs 1 when it detects a human movement and a zero otherwise, the result of the output was averaged over 2 minute before it is logged, its value ranges from 0.00 – 1.00.

Light Intensity: The light intensity of the room was detected using MAX44009 light sensor, it measures the light intensity in lux.

Temperature: The temperature of the room was also logged, the temperature was sensed using DHT11 temperature and humidity sensor.

Humidity: The humidity of the room was sensed using DHT11.

Light State: By using the MAX44009 the sensor was position to be shielded from ambient light, when the light is switched on there is a sudden increase in the light intensity, this behavior was used to detect when the light is on or off.

Socket1 state: The state of the first socket was detected using an LDR, placed inside the socket and focusing on its indicator, when the socket is off the reading of the LDR is low and when on the reading is high, this behavior was used to determine when the socket is on (1) and off (0).

Socket2 State: The state of the second socket was also detected as that of the formal and was also logged.

From the data collected we were able to learn and predict:

1. The state of the sockets and light at any time to some degree of accuracy.
2. The total number of load points the user uses any time as a form of energy consumption statistics.
3. The temperature and humidity of the room at any time.

3.6.2 Design of modules used to collect data for demo ML Algorithm.

Two modules were designed each attach to a socket, the client module and the server was designed using esp8266 nodeMCU module as the microcontroller board. The client module is the only one that carries the SD_CARD module, hence a Wi-Fi communication was set up between the two modules using the Wi-Fi capability of the esp8266, so that the server modules can send its own data to the client module to log on the SD_CARD.

Design of the Server Module: The components used for the design is shown on the circuit diagram below, the module carries the light intensity sensor (MAX44009), DHT11 temperature and humidity sensor and the LDR for sensing the socket state. The circuit diagram for the server module is shown in figure 3.13.

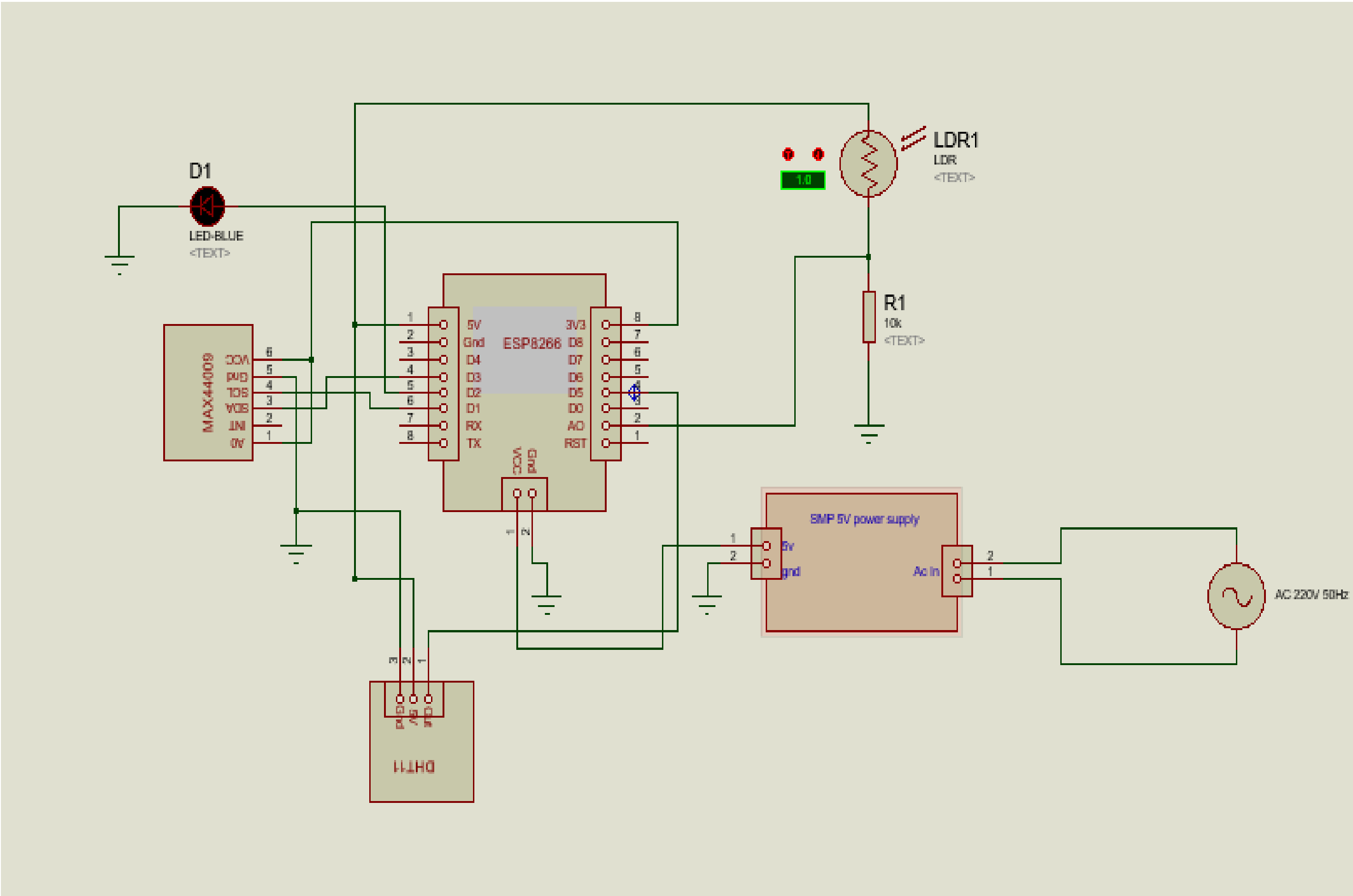

Figure 3.13 Circuit diagram of server module

Design of client module

The client module carried an LDR for sensing the socket state, a PIR motion sensor, an RTC module and a SD_CARD module. The circuit used is shown in figure 3.14.

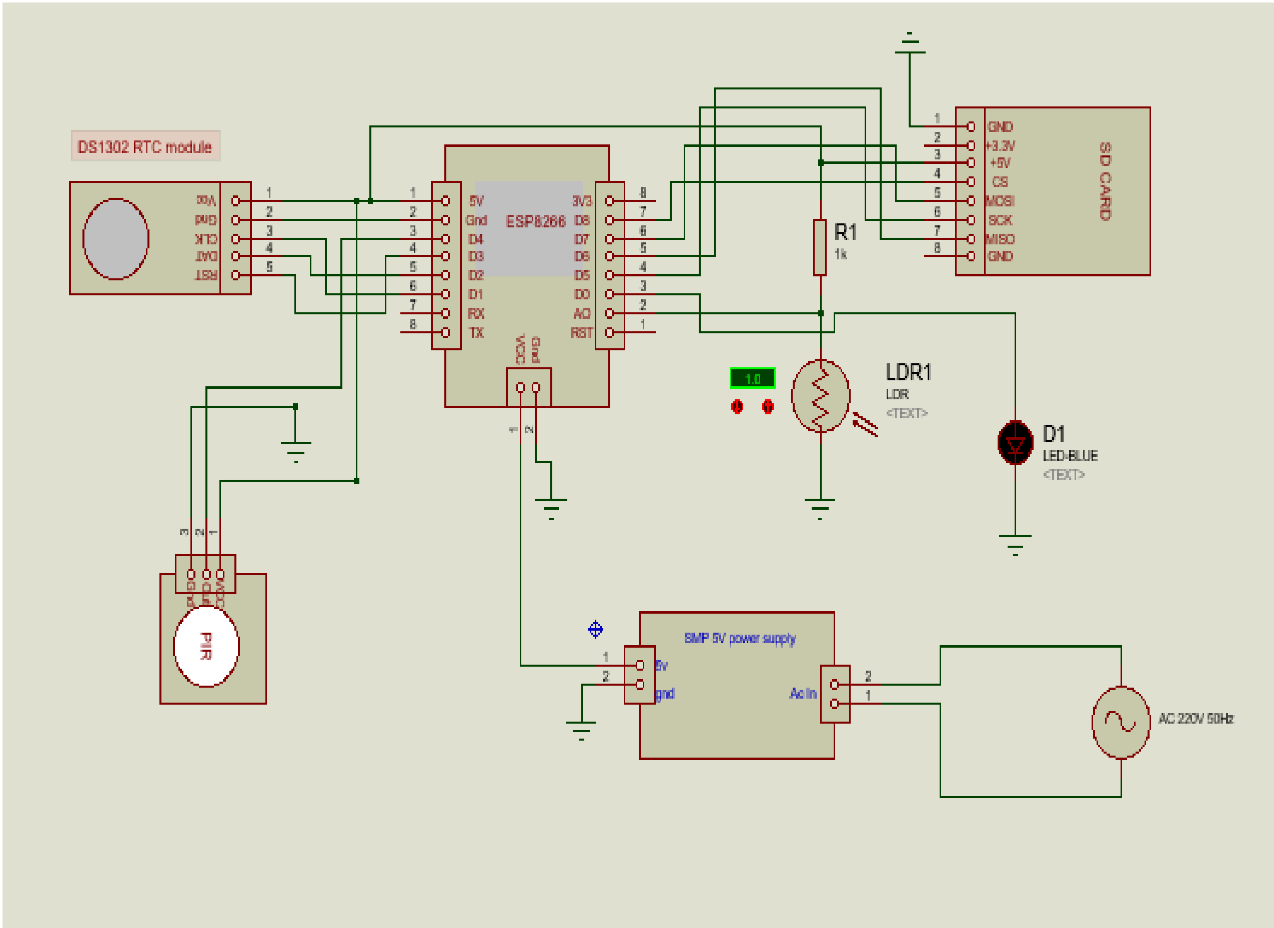

Figure 3.14 Circuit diagram of client module

Wi-Fi Communication between Server and Client module

The client module carries the only available SD_CARD, hence there is need for communication between the server module and the client module, the communication protocol used was Wi-Fi as esp8266 carries a Wi-Fi modem on board. The communication was made possible using the <ESP8266WiFiMesh> library. The client module sends hello request and the server module replies it with the required data.

Client request

sprintf(request, "Hello world request #%d from Mesh_Node%d.", request_i++, ESP.getChipId());

Client response

sprintf(response, "%d %d %d %d %d",LightReading(),tempValue,humValue,LightState,S1S);

3.6.3 The data logged

The data logged include the date and the time, the day type which is 2 for weekday and 1 for weekend, the day of the week which is 1-7 for Monday – Sunday, the hour of the day, the minute of the hour, the human presence rating, light intensity, temperature, humidity, light state, socket1 state and socket2 state.

An extract of the data is shown in figure 3.15.

		DT	DOW	HD	MH	HP	LI	TP	HM	LS	S1S	S2S
'04.08.2017'	'12:46:01'	2	5	12	46	0.33	27	26	69	1	1	1
'04.08.2017'	'12:48:01'	2	5	12	48	0.42	27	26	69	1	1	1
'04.08.2017'	'12:50:00'	2	5	12	50	0.48	28	25	69	1	1	1
'04.08.2017'	'12:52:00'	2	5	12	52	0.40	27	26	69	1	1	1
'04.08.2017'	'12:54:00'	2	5	12	54	0.37	26	25	68	1	1	1
'04.08.2017'	'12:56:01'	2	5	12	56	0.75	28	25	68	1	1	1
'04.08.2017'	'12:58:00'	2	5	12	58	0.58	23	25	68	1	1	1
'04.08.2017'	'13:00:01'	2	5	13	00	0.53	22	25	68	1	1	1
'04.08.2017'	'13:02:00'	2	5	13	02	0.68	23	26	67	1	1	1
'04.08.2017'	'13:06:01'	2	5	13	06	0.40	22	26	66	1	1	1
'04.08.2017'	'13:10:00'	2	5	13	10	0.54	25	27	64	1	1	1
'04.08.2017'	'13:12:01'	2	5	13	12	0.20	23	27	63	1	1	1
'04.08.2017'	'13:14:01'	2	5	13	14	0.62	24	27	63	1	1	1

Figure 3.15 extract of data logged

4.5.4 Learning from the data using Neural network in MATLAB

The prediction problem to be solved involve both classification and regression. The problem of predicting the state of the switch is a classification problem, this can be expanded to predicting the energy consumption statistics or pattern of the user. The problem of predicting the ambient condition of the room is a regression problem. The neural network toolbox has the ability to solve both regression and classification problem.

The step by step process in developing the learning algorithm of the learning process is shown in figure 3.16.

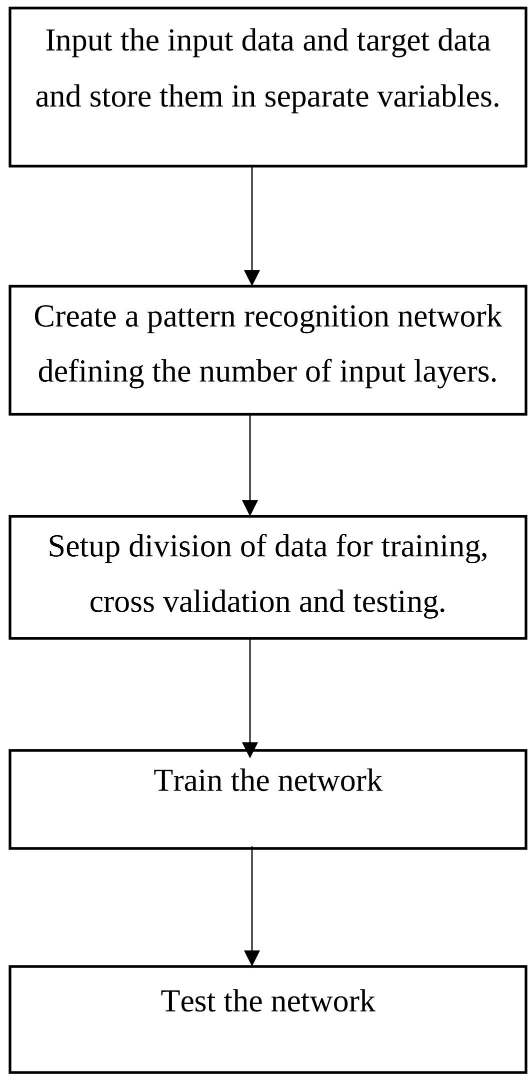

Figure 3.16 Program flow for Neural network algorithm

The training function generated from this process individually for socket1state, socket2state, and lightstate was called in another function that combines their result from a single input query and also to predict user energy usage statistics.

The classification problems follow the same process but the regression problem have some little changes, instead of creating a pattern recognition network, a training function and a fitting network was created.

CHAPTER 4

RESULT AND DISCUSSION

The design of the project involves coupling several hardware components together, developing a mobile application, developing a firebase cloud database and finally training a machine learning algorithm on MATLAB to predict user's attitude towards using the electrical installation.

4.1 The mobile app interface

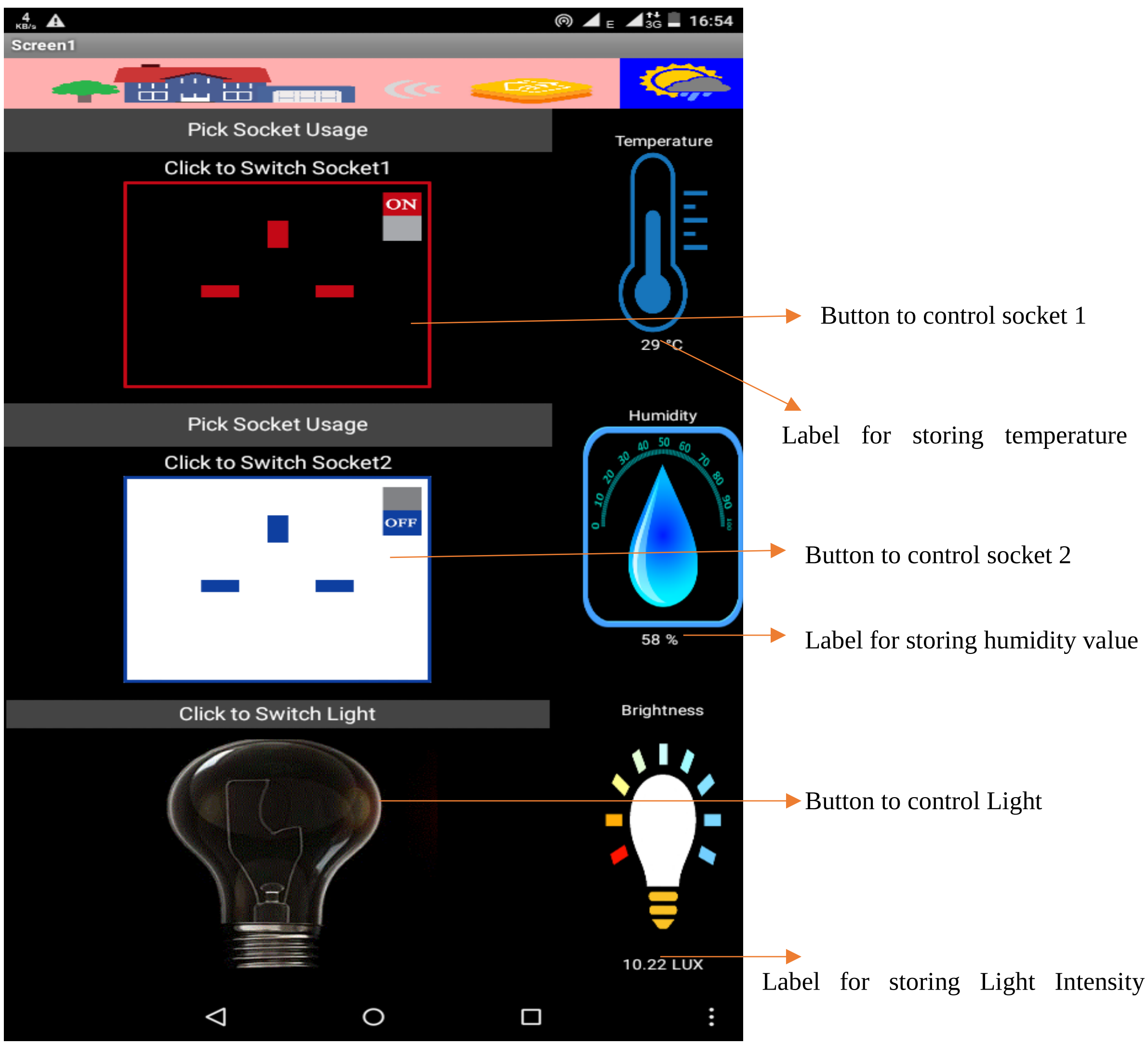

Figure 4.1 Graphical user interface of the mobile app

The mobile app interface as shown in fig 4.1 includes buttons to control the light source and 2 sockets, it also gives a visual feedback of the state of the load points in form of pictures, the red socket means an ON state, the blue socket means an OFF state, the bulb is either lit or dark indicating it's on and off state respectively. In addition, it includes label for showing the value of the humidity, temperature and light intensity of the room which constitute the ambient condition of the room.

4.2 The Google Firebase Real-time Database

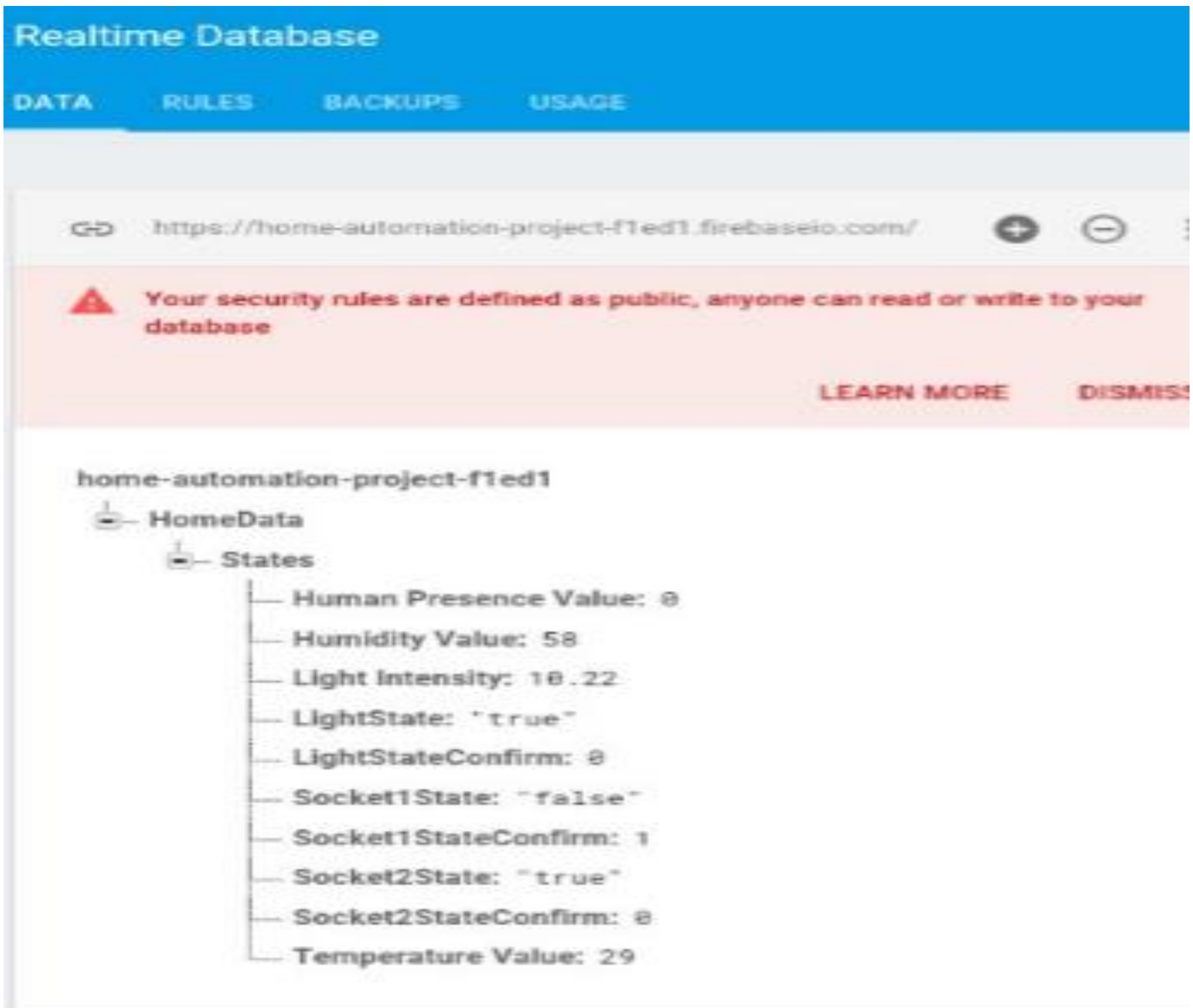

Figure 4.2 The Firebase real-time database

Some of the data stored are the human presence value, the humidity value, the temperature value, the light intensity value, the states of the switches and confirmation of the states in the real time database is as shown in figure 4.2.

4.3 The hardware

The hardware includes a board carrying 2 sockets, one lamp holder, the sensors, and the embedded system PCB board design that carries the microcontroller board and the relays as shown in figure 4.3.

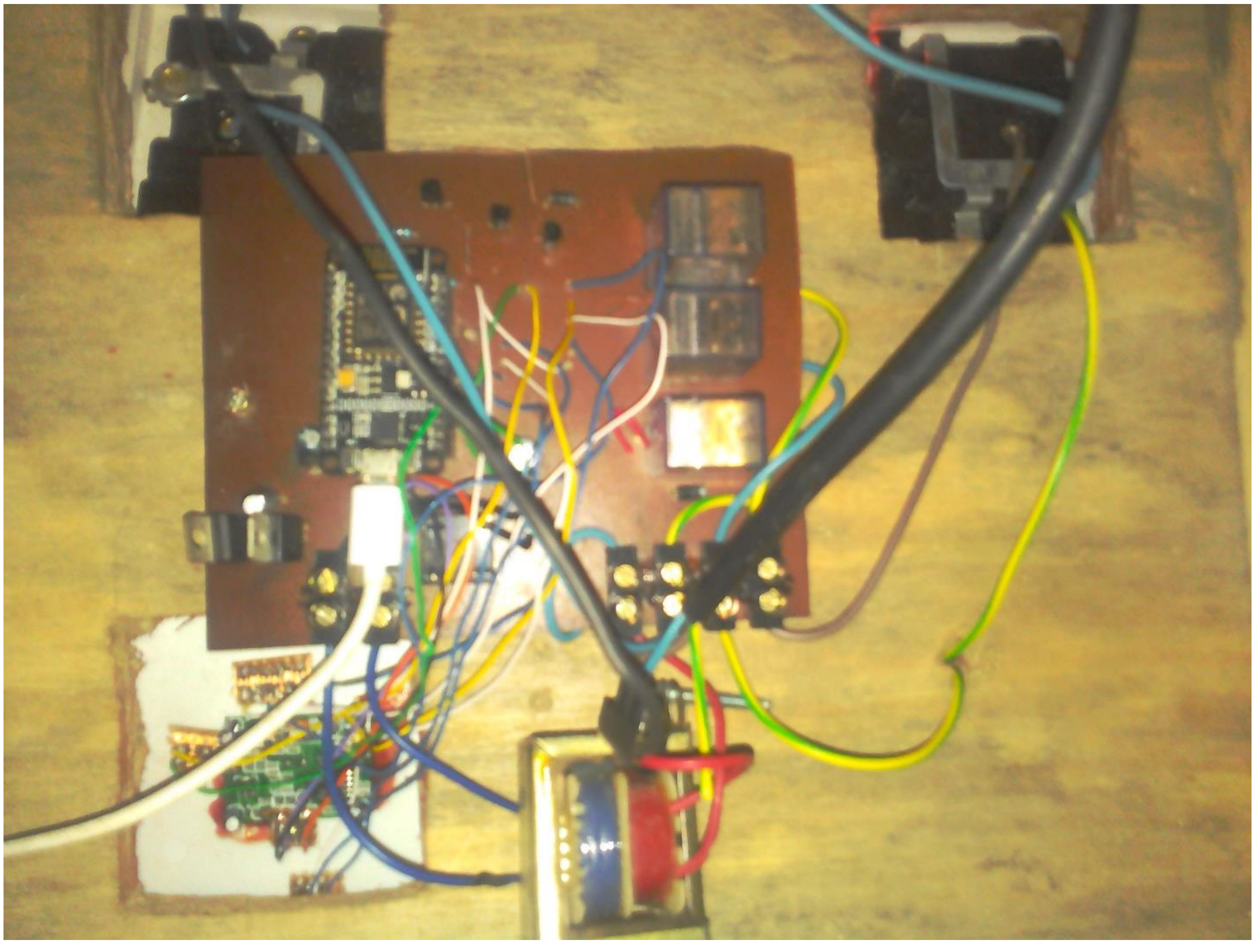

Figure 4.3 The Hardware design

4.4 The Machine Learning Algorithm Result and Discussion

4.4.1 Neural network classification confusion matrix

The confusion matrix shows the error and the success of the training in form of percentage. Each for the training, validation and testing, and it also shows the error and success for the total training under "All Confusion Matrix". Figure 4.4 shows the confusion matrix for socket1state training.

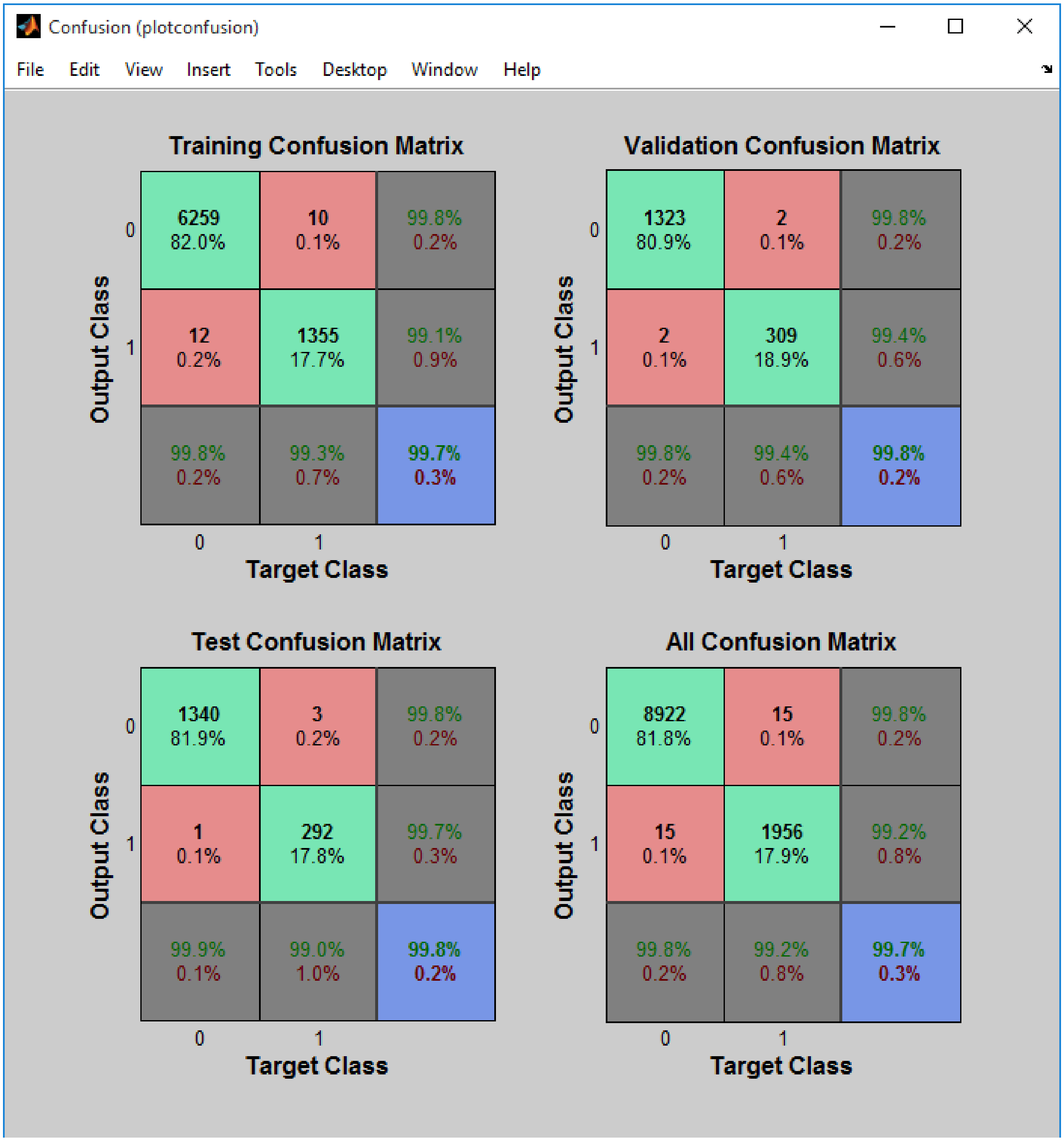

Figure 4.4 Confusion matrix of socket1 state training

As shown in figure 4.4 The network outputs are very accurate, as you can see by the high numbers of correct responses in the green squares and the low numbers of incorrect responses in the red squares. The lower right blue squares illustrate the overall accuracies.

4.4.2 Neural network regression for humidity and temperature prediction

For a regression problem such as the prediction of the humidity and temperature value, a regression plot is more suitable for viewing the training. As shown in figure 4.5, the line of best fit generated by the algorithm is not well fitted because of the complexity of the data.

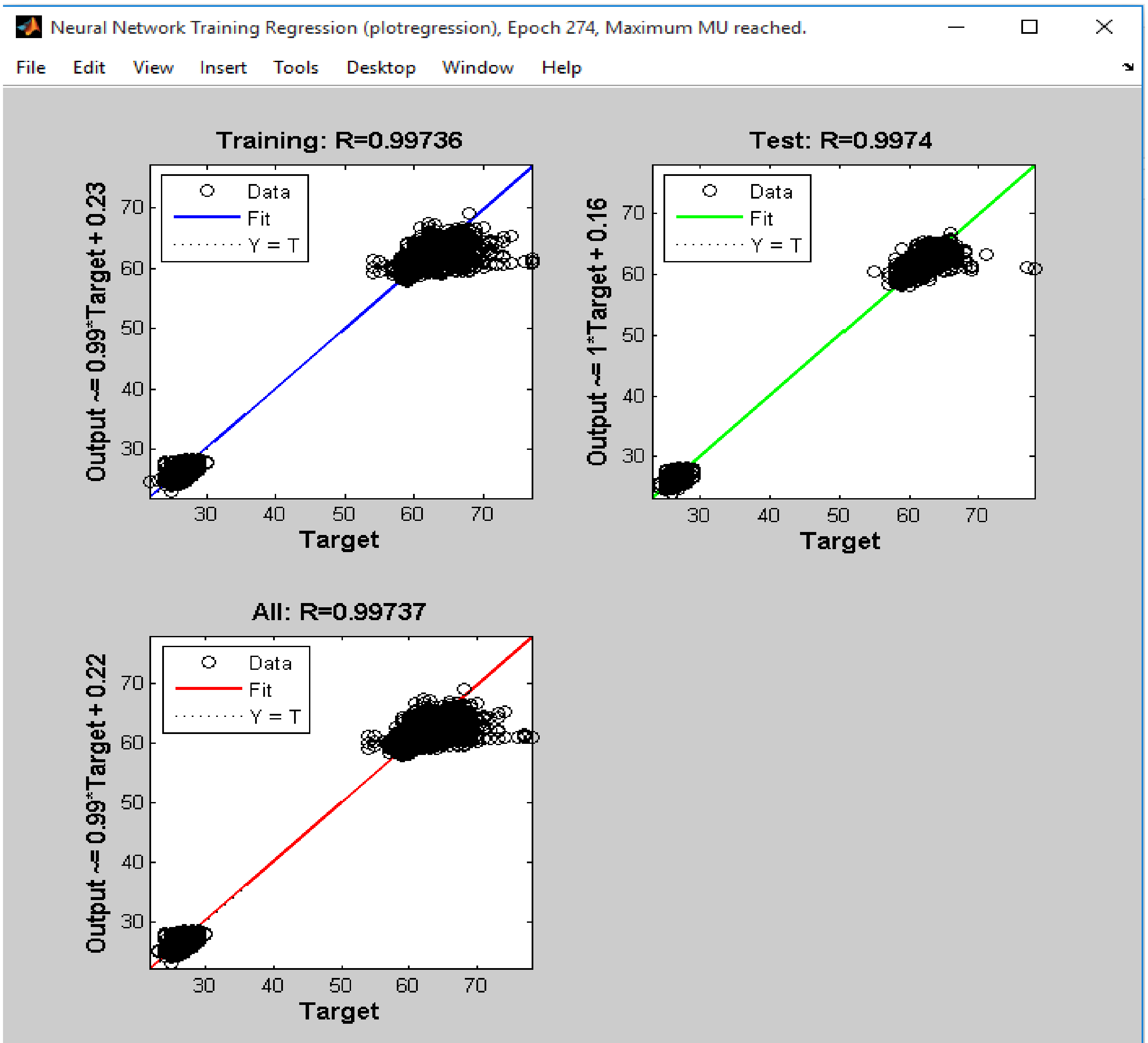

Figure 4.5 Regression plot of humidity and temperature state

4.4.3 Result of a prediction example

When the day type, day of the week, hour of the day, minute of the hour, human presence rating, humidity, temperature and light intensity were input into the prediction function generated by the algorithm, the results is shown in figure 4.6, figure 4.7 and figure 4.8. For figure 4.6 the input is 10:36 am, on Tuesday with a human presence rating of 0.69, light intensity of 53Lux, temperature of 25 deg C, and relative humidity of 61%. The predicted output is in form of a probability, the light is 0.8596 probable to be ON, socket1 is 1.0 probable to be ON, and socket2 is 0.9486 probable to be ON. A function takes the probability as an input and if it is greater than 0.7 it predicts it as an ON state. Hence as shown, Light should be on, socket1 should be ON and socket2 should be ON. And the energy consumption statistics shows that there is maximum load utilization at that particular time.

```
>> PredictSwitchState(2,2,10,36,0.690000000000000,53,25,61)
   0.8596

   1.0000

   0.9489


ans =

Light should be: On
Socket1 should be: On
Socket2 should be: On

Maximum Load Utilization
```

Figure 4.6 Predicting for 10:36 am on Tuesday

```
>> PredictSwitchState(2,4,1,42,0,0,26,61)
   7.0416e-05


   1.2350e-04


    0.0649



ans =


Light should be: Off
Socket1 should be: Off
Socket2 should be: Off


No load In Use
```

Figure 4.7 Predicting for 01:42 am on Thursday

For figure 4.7, the input is 01:42 am in the midnight, on Thursday with a human presence rating of 0, light intensity of 0 lux, temperature of 26 deg C, and relative humidity of 61%. The predicted output is in form of a probability, the light is 7e-05 probable to be ON, socket1 is 1.23e-04 probable to be ON, and socket2 is 0.0649 probable to be ON. A function takes the probability as an input and if it is less than 0.3 it predicts it as an OFF state. Hence as shown, Light should be OFF, socket1 should be OFF and socket2 should be OFF. And the energy consumption statistics shows that there is no load in use at that particular time.

```
>> PredictSwitchState(1,7,21,10,0,0,28,62)
  3.9611e-10

  4.2185e-06

  1.4624e-09


ans =

Light should be: Off
Socket1 should be: Off
Socket2 should be: Off

No load In Use
```

Figure 4.8 Predicting for 21:10 pm on Sunday

For fig 4.8, the input is 21:10 pm, on sunday with a human presence rating of 0, light intensity of 0 lux, temperature of 28 deg C, and relative humidity of 62%. The predicted output is in form of a probability, the light is 3.9e-10 probable to be ON, socket1 is 4.22e-06 probable to be ON, and socket2 is 1.46e-09 probable to be ON. A function takes the probability as an input and if it is less than 0.3 it predicts it as an OFF state. Hence as shown, Light should be OFF, socket1 should be OFF and socket2 should be OFF. And the energy consumption statistics shows that there is no load in use at that particular time.

As the data was taken from an office space in the lab, the result of the prediction is reasonable.

4.4.5 Comparing actual state to prediction result for socket1

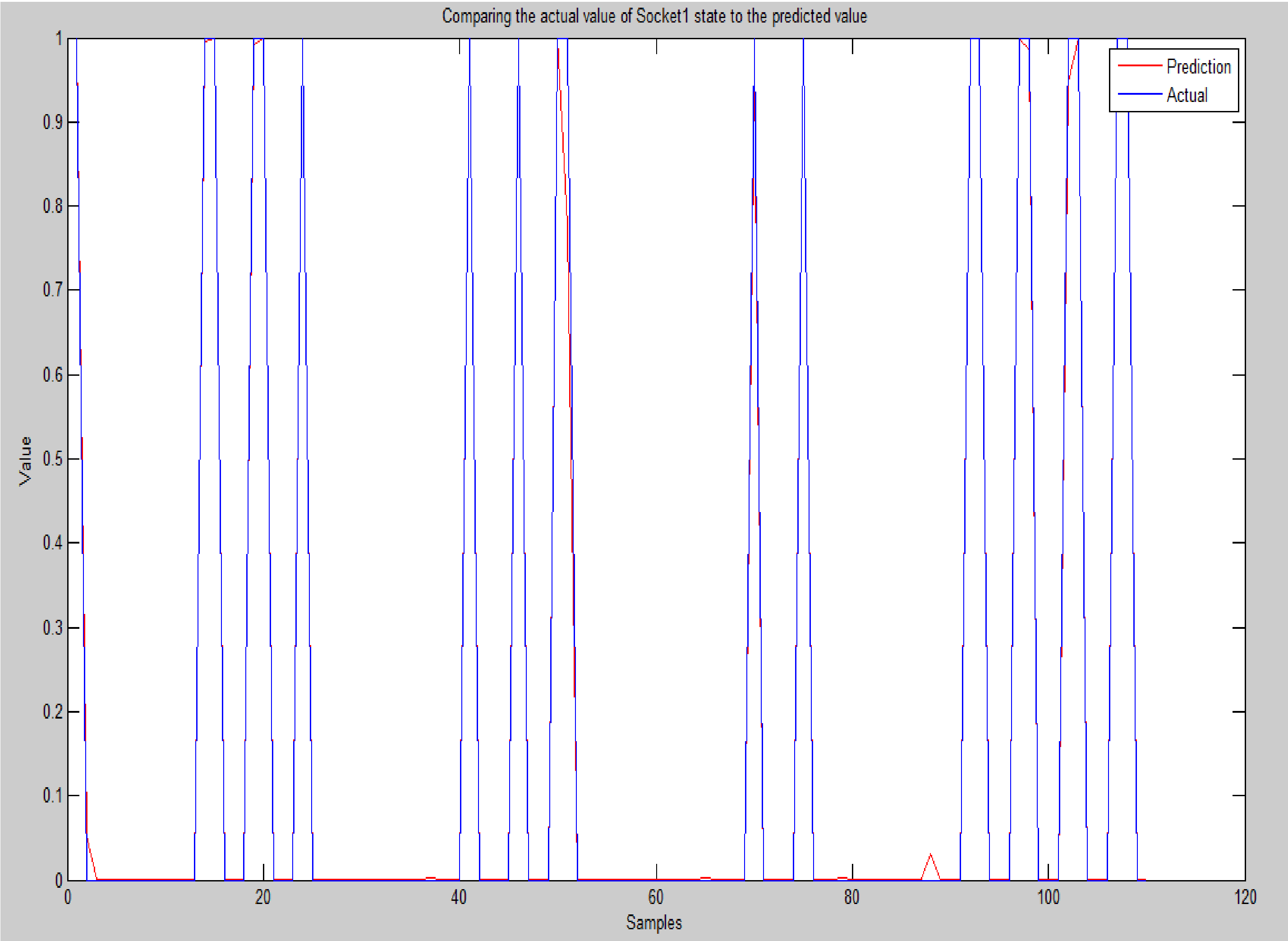

Figure 4.9 Plot comparing actual value to predicted value for socket1

Figure 4.9 shows a plot of predicted value to actual value for socket1, for some randomly chosen weights, the red line shows the predicted value, and the blue line show the actual value. It can be seen that the prediction is not completely perfect at some weights and seems to be perfect at some other weight.

4.4.6 Comparing actual state to prediction result for socket2

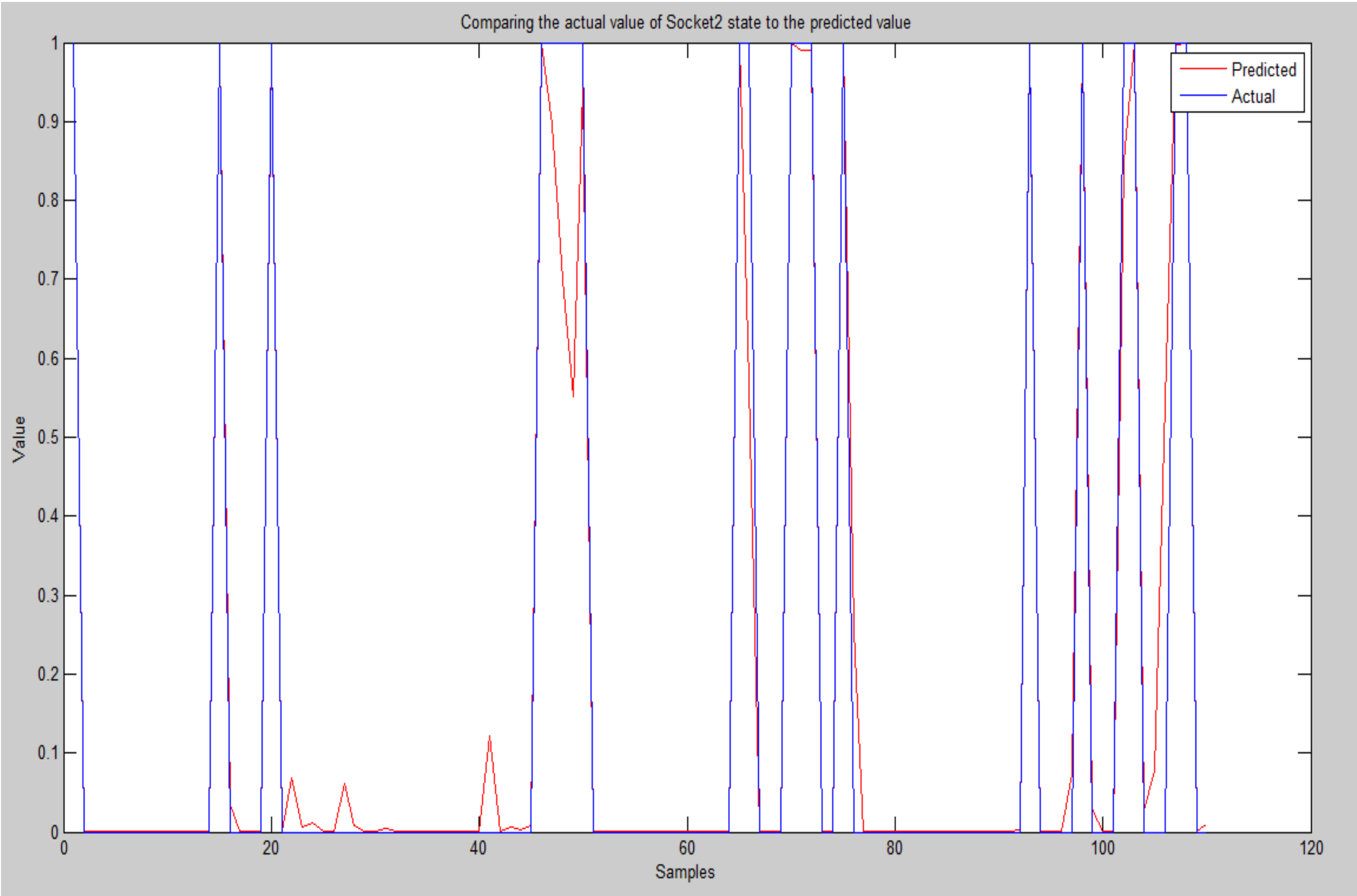

Figure 4.10 Plot comparing actual value to predicted value for socket2

Figure 4.10 shows a plot of predicted value to actual value for socket2, for some randomly chosen weights, the red line shows the predicted value, and the blue line show the actual value. It can be seen that the prediction is not completely perfect at some weights and seems to be perfect at some other weights. There is more deviation in the result of the predicted value for socket2 because some of the data used to train its algorithm are redundant.

4.4.7 Comparing actual state to prediction result for light

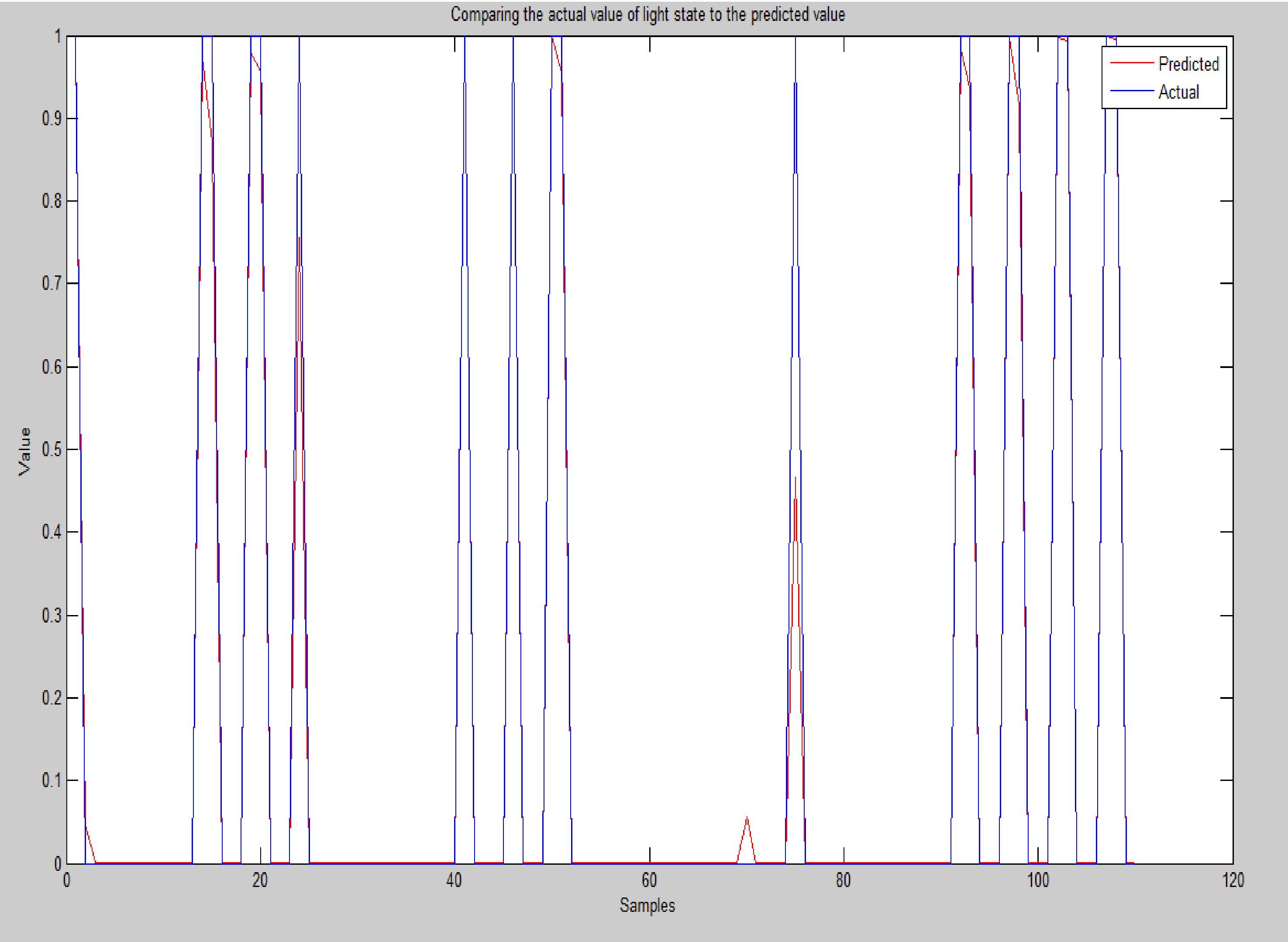

Figure 4.11 Plot comparing actual value to predicted value for light

Figure 4.11 shows a plot of predicted value to actual value for light, for some randomly chosen weights, the red line shows the predicted value, and the blue line show the actual value. It can be seen that the prediction is not completely perfect at some weights and seems to be perfect at some other weights.

4.5 Testing and Evaluation

Figure 4.12, 4.13 and 4.14 show the project in use with the lamp and one socket ON and the other socket OFF, the data on the firebase database and also the feedback on the mobile app. When a user clicks the switch button on the mobile app to control the switch, the state of the switch will change on the firebase database, when the embedded system detects that change, it will act on the relay, and give a feedback as switch state confirmation to the database. The mobile app on seeing the confirmation gives a visual feedback to the user by changing the picture on the button pressed.

Figure 4.12 Testing the hardware

As shown in figure 4.12, one of the socket is ON as indicated by its indicator, the lamp holder is also ON and the second socket is OFF.

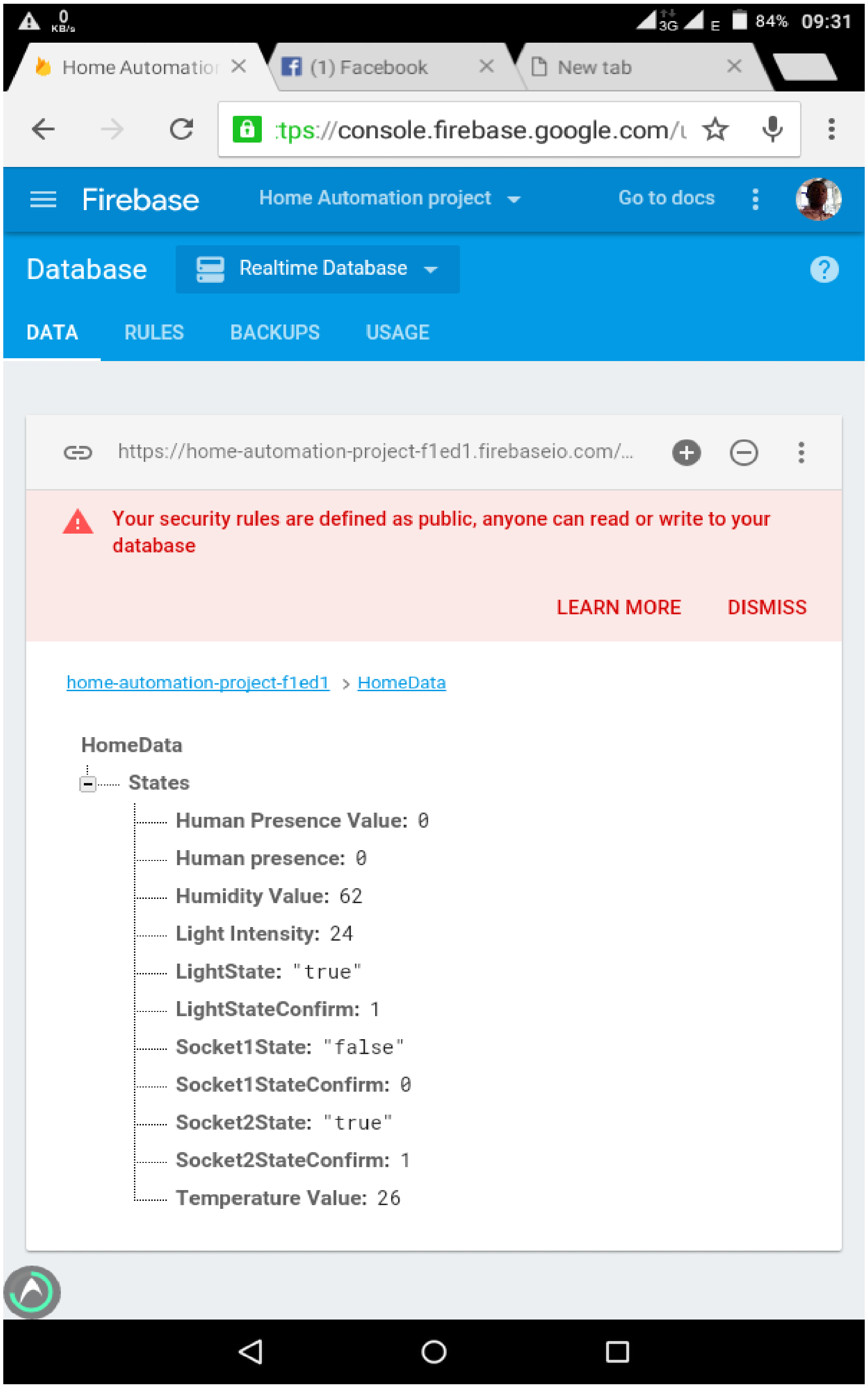

Fig 4.13 The database

Fig 4.13 shows the data in the firebase database.

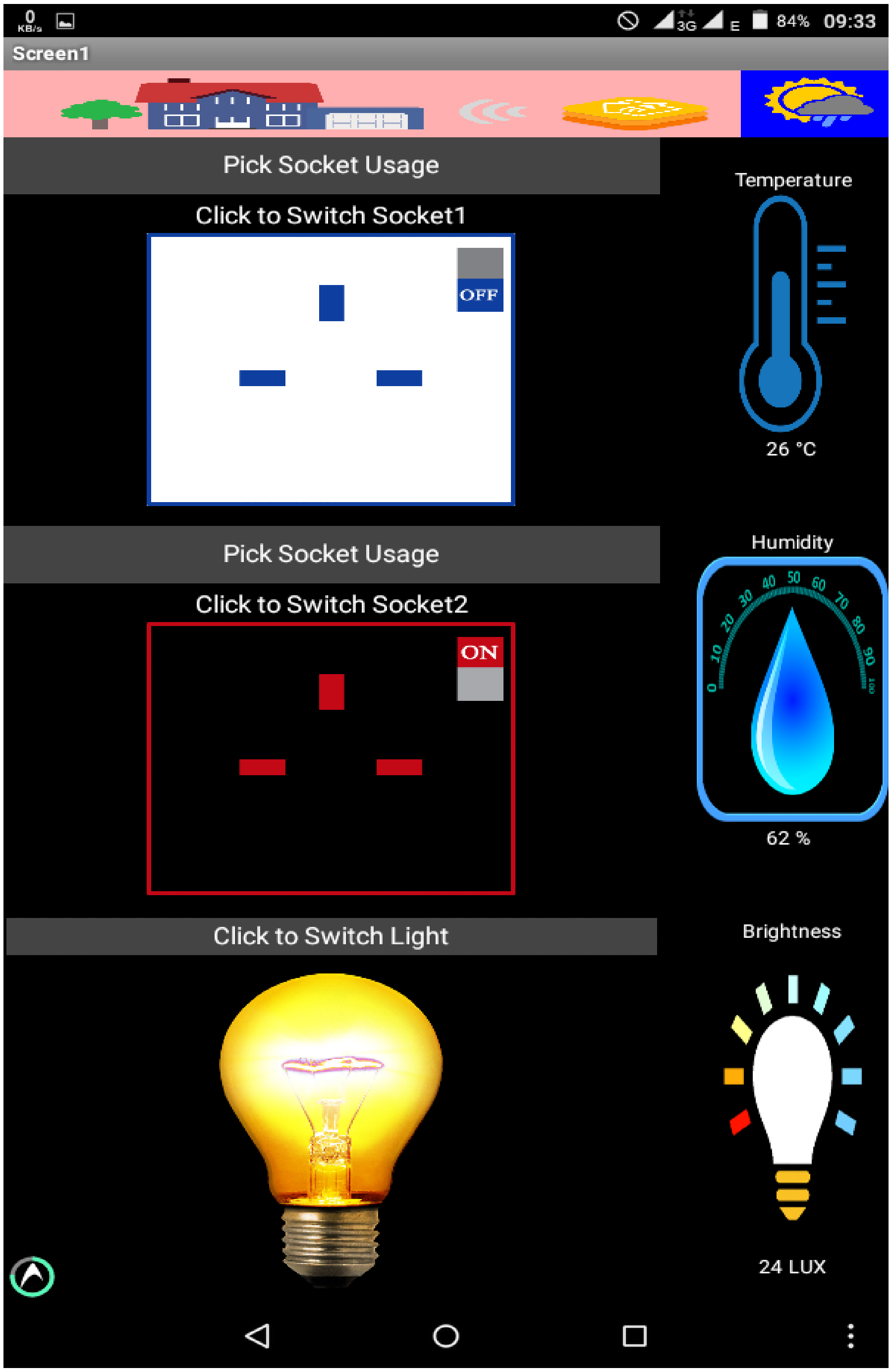

Figure 4.14 The mobile app in use

The blue button on socket1 shows that the socket is ON, the red button on socket2 shows that the socket is ON, and the bulb picture being lit shows that the bulb is ON, this is in agreement with Figure 4.12 and Figure 4.13. The temperature, humidity and light intensity are also shown in their respective labels.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

2.5 Conclusion

It is evident from this project work that an intelligent IoT home automation system can be cheaply made from low-cost locally available components and can be used to control multifarious home appliances ranging from the security lamps, the television to the air conditioning system and even the entire house lighting system from anywhere the user might be, as far as the user is connected to the internet. And better still, the components required are so small and few that they can be packaged into a small inconspicuous container. Using a mobile app with easy to understand graphical user interface will give user a wonderful user experience. The machine learning results show that it is possible to learn from the way a user a user uses her home automation and use this data to automate a process that predicts what the states of the system should be at any time, the energy consumption statistics of the user and the ambient condition of the room.

The designed home automation system was tested a number of times and certified to control different home appliances, including lighting system, air conditioning system, heating system, home entertainment system and many more (this is as long as the maximum power and current rating of the appliance does not exceed that of the used relay).

2.6 Recommendations

From the experience gained and challenges faced during the course of this project work, the following recommendation is made;

- Most cloud system with a complete cloud infrastructure don’t have support for cheap IOT modules, I recommend that they update their system so that it can be used for cheap IOT applications.
- I recommend that future studies, use online learning mode of machine learning as human behavior towards usage of home appliances varies with time.
- Due to the difficulty of getting some of the component for this project, I recommend that the department should acquire some of this IOT modules and embedded system board in the lab.

- The result of this project shows what can be indigenously done in Nigeria, therefore I recommend that the government should support and create opportunities for some of this project work to get to the local industry.

REFERENCES

Russell S., Stuart K.D. (2003). Artificial Intelligence, Prentice Hall, Upper Saddle River, USA.

Michell T.(1997). Machine Learning, McGraw Hill, Louisiana, USA.

Alex Smola, Vishwanathan S.V.N. (2008). Introduction to Machine Learning, Cambridge University press, Edinburgh Building, Cambridge, UK.

Langley H, Pat L.S. (2011). The changing science of machine learning, Machine learning, https//doi.org?10.1007%2Fs10994-011-5242-y.

Rita Yi Man; Li J., Herru Ching Yu; Mak, Cho Kei; Tang, Tony Beigi. (2015), "Sustainable Smart Home and Home Automation: Big Data Analysis Approach", International Journal of Smart Home.

Olafusi M.O. (2009). Individual Control Home Automation, Federal University of Technology Akure.

Alhaj A. (2015). Home Automation System Using Arduino, Universiti Teknologi Malaysia, Malaysia.

Javier C., Adrián N. and Ernesto P., (2007). A Cloud-Based Internet of Things for Assisted Living, Universidad de Málaga, Departamento de Lenguajes y Ciencias de la Computación, Campus de Teatinos, 29071 Málaga, Spain.

Mehta V.K., Mehta R. (2014). Principle of Electronics, S Chand publishers, Ram Nager, New Delhi.

Dr. Ovidiu V., Dr Peter F. (2014). Internet of Things- From Research and Innovation to Market Deployment, River Publishers, Aalborg.

Charalampos D., LLias M. (2011). Managing Wearable Sensor Data through Cloud Computing, IEEE computer society, 2011 Third IEEE international conference on cloud computing technology and science.

Moraru A., Marko P., Maria P., Carolina F., Dunja M., (2010). Using Machine Learning on Sensor Data, Journal of Computing and Information Technology- CIT 18,2010, 2 , 342-347.

APPENDIX 1

1. Code for the home automation system

```
#include <Wire.h>
#include <MAX44009.h>
#include <dht11.h>
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
MAX44009 light;
#define FIREBASE_HOST "home-automation-project-f1ed1.firebaseio.com"
#define FIREBASE_AUTH "sF4fmvhzuSem8q75XYnjzbmi7W6pktTTGPTNfUAb"
#define WIFI_SSID "dan"
#define WIFI_PASSWORD "Danick007@ROSS"

 dht11 DHT11;
const int Light1 = D4;
const int Socket1 = D6;
const int Socket2 = D7;
const int PIRpin = D8;
const int WifiIndicator = D3;
String HomePath = "HomeData";
String path = HomePath + "/States";

  void setupPins(){
    pinMode(Light1, OUTPUT);
    pinMode(Socket1, OUTPUT);
    pinMode(Socket2, OUTPUT);
    pinMode(PIRpin,INPUT);
    pinMode(WifiIndicator,OUTPUT);
    }

    void setupWifi(){
      WiFi.begin(WIFI_SSID,WIFI_PASSWORD);
```

```
    Serial.println("Hey I 'm connecting...");
    while(WiFi.status()!= WL_CONNECTED ){
      Serial.println(".");
      digitalWrite(WifiIndicator,HIGH);
      delay(500);
      }
      Serial.println();
      Serial.println("I am connected and my IP address is ");
      Serial.println(WiFi.localIP());
      digitalWrite(WifiIndicator,LOW);
  }

    void setupFirebase()
    {
      Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
      if(Firebase.failed()){
        Serial.print("setting/number failed:");
        Serial.println(Firebase.error());
        return;
        }
    }


void setup() {
  // put your setup code here, to run once:
light.begin();
Serial.begin(9600);
digitalWrite(WifiIndicator,LOW);
setupWifi();
setupFirebase();
DHT11.attach(D5);
Serial.println("DHT11 TEST PROGRAM ");
Serial.print("LIBRARY VERSION: ");
```

```
Serial.println(DHT11LIB_VERSION);
}

 void Switching()
 {
  FirebaseObject object = Firebase.get(path);
 if(Firebase.failed()){
   Serial.print("Failed to access path");
   digitalWrite(WifiIndicator,HIGH);
   delay(100);
   digitalWrite(WifiIndicator,LOW);
   delay(100);
   return;
   }

  String LightState = Firebase.getString("HomeData/States/LightState");
  String Socket1State = Firebase.getString("HomeData/States/Socket1State");
  String Socket2State = Firebase.getString("HomeData/States/Socket2State");
 Serial.println(LightState);
 Serial.println(Socket1State);
 Serial.println(Socket2State);
 digitalWrite(WifiIndicator,LOW);
 if(LightState == "true"){
 digitalWrite(Light1,HIGH);
 Firebase.setInt("HomeData/States/LightStateConfirm",1);
 }
 else{
 digitalWrite(Light1,LOW);
 Firebase.setInt("HomeData/States/LightStateConfirm",0);
 }
  if(Socket1State == "true"){
 digitalWrite(Socket1,HIGH);
 Firebase.setInt("HomeData/States/Socket1StateConfirm",1);
```

```
   }
  else{
  digitalWrite(Socket1,LOW);
  Firebase.setInt("HomeData/States/Socket1StateConfirm",0);
  }
   if(Socket2State == "true"){
  digitalWrite(Socket2,HIGH);
Firebase.setInt("HomeData/States/Socket2StateConfirm",1);
   }
  else{
  digitalWrite(Socket2,LOW);
  Firebase.setInt("HomeData/States/Socket2StateConfirm",0);
  }
  }

    float LightReading()
    {
      float x = light.get_lux();
      return x;
    }

void loop() {
  // put your main code here, to run repeatedly:
  setupPins();
  int x = digitalRead(PIRpin);
  Serial.println(x);
Switching();
float lightLevel = LightReading();
int tempValue = 0;
int humValue = 0;
Serial.println("\n");
int chk = DHT11.read();
  switch (chk)
```

```
  {
   case 0:
   Serial.print("Humidity (%): ");
   humValue = (DHT11.humidity);
   tempValue = (DHT11.temperature);
   break;
   case -1: Serial.println("Checksum error"); break;
   case -2: Serial.println("Time out error"); break;
   default: Serial.println("Unknown error"); break;
  }
Switching();
Firebase.setFloat("HomeData/States/Light Intensity", lightLevel);
Switching();
Firebase.setInt("HomeData/States/Temperature Value",tempValue);
Switching();
Firebase.setInt("HomeData/States/Humidity Value", humValue);
Firebase.setInt("HomeData/States/Human presence",x);
Serial.print("Light Level : ");
Serial.println(lightLevel);
Serial.print("Temperature value: ");
Serial.println(tempValue);
Serial.print("Humidity value: ");
Serial.println(humValue);
Switching();
delay(100);
}
```

APPENDIX 2

Machine learning prediction functions code.

```
function [ x1,x2,x3] = PredictSwitchState(DT,DOW,HD,MH,HP,LI,TP,HM)


switch nargin
    case 8
        p = [DT;DOW;HD;MH;HP;LI;TP;HM];
        x1 = LightStatePredictionFunc(p);
        x2 = Socket1StatePredictionFunc(p);
        x3 = Socket2StatePredictionFunc(p);
         test(x1,x2,x3)
    case 7
        p = [DT;DOW;HD;MH;HP;LI;TP;60];
         x1 = LightStatePredictionFunc(p);
        x2 = Socket1StatePredictionFunc(p);
        x3 = Socket2StatePredictionFunc(p);
        test(x1,x2,x3)
    case 6
        p = [DT;DOW;HD;MH;HP;LI;25;60];
        x1 = LightStatePredictionFunc(p);
        x2 = Socket1StatePredictionFunc(p);
        x3 = Socket2StatePredictionFunc(p);
        test(x1,x2,x3)
    case 5
        if (HD > 20 & HD < 6)
         p = [DT;DOW;HD;MH;HP;0;25;60];
         x1 = LightStatePredictionFunc(p);
        x2 = Socket1StatePredictionFunc(p);
        x3 = Socket2StatePredictionFunc(p);
        test(x1,x2,x3)
        else
            p = [DT;DOW;HD;MH;HP;20;25;60];
            x1 = LightStatePredictionFunc(p);
          x2 = Socket1StatePredictionFunc(p);
           x3 = Socket2StatePredictionFunc(p);
           test(x1,x2,x3)
        end
```

```
    case 4
        if(HD > 17 & HD < 7)
            p = [DT;DOW;HD;MH;0;0.25;60];
             x1 = LightStatePredictionFunc(p);
            x2 = Socket1StatePredictionFunc(p);
            x3 = Socket2StatePredictionFunc(p);
            test(x1,x2,x3)
        else
            p = [DT;DOW;HD;MH;0.1;30;25;60];
            x1 = LightStatePredictionFunc(p);
            x2 = Socket1StatePredictionFunc(p);
            x3 = Socket2StatePredictionFunc(p);
            test(x1,x2,x3)
        end
    case 3
           if(HD > 17 & HD < 7)
             p = [DT;DOW;HD;0;0;0;25;60];
             x1 = LightStatePredictionFunc(p);
            x2 = Socket1StatePredictionFunc(p);
            x3 = Socket2StatePredictionFunc(p);
            test(x1,x2,x3)
           else
            p = [DT;DOW;HD;0;0.1;30;25;60];
            x1 = LightStatePredictionFunc(p);
            x2 = Socket1StatePredictionFunc(p);
            x3 = Socket2StatePredictionFunc(p);
            test(x1,x2,x3)
           end
    otherwise
        error('not enough input argument, minimum of 3 input argument (Day type, Day of the Week, Hour of
the day), try help PredictSwitchState for more information')
end
end
function [] = test(a,b,c)
    if (a >= 0 & a <= 0.3 )
        d = 'Off';
        x = 0;
```

```
    elseif (a <= 1 & a >= 0.7)
         d = 'On';
         x = 1;
    else
        d = 'Algorithm is undecisive';
        x = 2;
    end

    if (b >= 0 & b <= 0.3 )
        e = 'Off';
        y = 0;
    elseif (b <= 1 & b >= 0.7)
         e = 'On';
         y = 1;
    else
        e = 'Algorithm is undecisive';
        z = 2;
    end

     if (c >= 0 & c <= 0.3 )
        f = 'Off';
        z = 0;
    elseif (c <= 1 & c >= 0.7)
         f = 'On';
         z = 1;
    else
        f = 'Algorithm is undecisive';
        z = 2;
     end
disp(a);
disp(b);
disp(c);
sprintf('Light should be: %s ¥nSocket1 should be: %s ¥nSocket2 should be: %s ',d,e,f)
if (x == 1 & y == 1 & z == 1)
    disp('Maximum Load Utilization');
elseif (x == 0 & y == 1 & z == 1)
    disp('Two-Third of Load in Use');
  elseif (x == 1 & y == 0 & z == 1)
```

```
    disp('Two-Third of Load in Use');
   elseif (x == 1 & y == 1 & z == 0)
    disp('Two-Third of Load in Use');
     elseif (x == 0 & y == 0 & z == 1)
    disp('One-Third of Load in Use');
     elseif (x == 0 & y == 1 & z == 0)
    disp('One-Third of Load in Use');
     elseif (x == 1 & y == 0 & z == 0)
    disp('One-Third of Load in Use');
 else
    disp('No load In Use')
 end

end
```