A NEW APPROACH TO THE DESIGN OF INTERNET OF THINGS SYSTEM (IOT), USING SMART DEVICE WITH TEMPERATURE SENSOR

BY

ETOR PROMISE

(B.SC Industrial Physics)

PSC1809211

SUPERVISED BY

DR. O. EBONWONIYI

FACULTY OF PHYSICAL SCIENCE

DEPARTMENT OF PHYSICS

UNIVERSITY OF BENIN

SEPTEMBER 2022

A NEW APPROACH TO THE DESIGN OF INTERNET OF THINGS SYSTEM (IOT), USING SMART DEVICE WITH TEMPERATURE SENSOR

BY

ETOR PROMISE

(B.SC Industrial Physics)

PSC1809211

SUPERVISED BY

DR. O. EBONWONIYI

A PROJECT SUBMITTED TO THE DEPARTMENT OF PHYSICS, FACULTY OF PHYSICAL SCIENCES, IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF DEGREE OF BACHELOR OF SCIENCE (B.SC) IN PHYSICS UNIVERSITY OF BENIN, BENIN CITY.

SEPTEMBER 2022

CERTIFICATION

This is to certify that this project work was carried out by ETOR PROMISE PSC1809211, student of the DEPARTMENT OF PHYSICS, UNIVERSITY OF BENIN, in partial fulfillment of the requirements for the award of the BACHELOR of INDUSTRIAL PHYSICS DEGREE in PHYSICS; under the supervision of ***DR. O, EBOMWONYI***

_____ _____

DR.O. EBOMWONYI

PROJECT SUPERVISOR DATE

_____ _____

PROF. O.D. OSAHON

HEAD OF DEPARTMENT DATE

_____ _____

EXTERNAL EXAMINER DATE

DEDICATION

This project is dedicated to my God Almighty, my parents for their contribution, my wonderful siblings and friends. I am very grateful and thank you for believing in me.

,

CERTIFICATION OF DISSERTATION OF PLAGIARISM

We the undersigned attest and declare that the dissertation of ETOR PROMISE titled "New approach to the design of internet of things system(IOT) using smart device with temperature sensor" has successfully passed the anti-plagiarism test and hence not violate any copyright@ regulations.

_____ _____

DR.O.EBOMWONYI

(Project Supervisor) **DATE**

_____ _____

PROF. O.D OSAHON

(Head of Department) **DATE**

ACKNOWLEDGEMENT

The level of achievement in this project work and my entire years in this great institution is as a result of hard work and the contribution of certain individuals who were concerned for my future.

Firstly I wish to express my profound gratitude to God almighty for his grace love and perfect heath that enabled me pass through this institution with great success.

I wish to express my gratitude to my project supervisor, DR.O. EBOMWONYI for his interest, support, consistent advice, academic assistance and in the amendment and correction he made on my write up with great supervision throughout this research.

Finally I wish to express my love and say a special thank you to my parents Mr. and Mrs. Etor for their great support morally, financially, spiritually and otherwise.

ABSTRACT

As technology improves, it has slowly and efficiently been embedded into every part of life. With the advent of communication protocols, organizations have turned to building gadgets that can interact with one other over communication protocols.

This has extended into the homes and how we can use these communication protocols to build a smart home where appliances and energy consumption can be controlled remotely.

This project focuses on the Smart Room Control System with Temperature Display using Arduino is a project that aims to create an automated and efficient solution for controlling various appliances like bulbs, fans, and an air conditioner (AC) within a room. The system will also provide real-time temperature monitoring for better comfort and energy management. The core of this project is an Arduino microcontroller that communicates with the appliances and temperature sensors to enable seamless automation and control.

TABLE OF CONTENTS

CHAPTER ONE

INTRODUCTION

1.1 PROJECT BACKGROUND

From the name Internet of things (IOT) consists devices with switches carrying information on the internet. Internet of things (IOT) shows the idea of remotely connecting and controlling real-world objects (things) through the internet (Asghar, 2015). The idea of IOT wasn't officially named until the late 1990s. From 1999 till now, IOT has evolved and produced profound results. The internet and computers we use this present days, can function on their own with or without operations. As opposed to how data was imputed for the internet which was susceptible to errors, computers can now gather such data for themselves. Entering the 2000s, Kevin Ashton said Radio Frequency Identification (RFID) was one of the key foundation for the Internet of Things. RFID (radio frequency identification) is a form of wireless communication to entails the use of electromagnetic or electrostatic coupling in the radio frequency portion of the electromagnetic diapason to uniquely identify an object, machine, or person (Sarah, 2021) Into the 21st century, The US Department of Defense, as well as Walmart, welcomed Ashton's model of tracking inventory using tagging, RFID. Fast forward to 2011 the Ring doorbell was developed by Jamie Siminoff because he wanted to see who was at his door while he was in the garage, working (Jamie, 2014). The opportunities and possibilities with IOT are seemingly limitless; small cities, companies, schools, and homes can now reduce losses by using IOT. For instance, a small city can use sensors and have systems work around them. This leads us into the industrial internet of things (IIOT). The industrial internet of things (IIOT) is an extension of the internet of things (IOT) in industrial sectors and its various applications. It focuses mainly on machine-to-machine

(M2M) communication, big data, and machine learning(AI). The IIOT helps industries and enterprises to have much better efficiency and reliability in their operations with devices.

In 2013 IOT had expanded to a higher scale. It is a system that entails numerous technologies. Almost anything you could think of have a relationship with IOT or built on IOT system. Smartphones majorly became a re-known part of IOT in 2015. Health industries also took advantage of the technology, devices such as smart watches and phones could now track the state of health of their patients, also accumulating data concerning heart rate, blood pressure, etc. (National Library of Medicine, 2018). With some health institutes bringing on board the idea of robots aiding difficult surgical operations. The robotic surgical system development includes a camera arm and mechanical arms with surgical instruments attached. A doctor, being the surgeon, controls the arms while seated at a computer console near the operating table. Seeming almost like a game, the console gives the surgeon a highly defined, magnified, 3D view of the surgical site. Like other surgical operations, the surgeon leads other team members who assist (The Mayo Clinic Staff, 2022). Fusing Artificial Intelligence (AI) with IOT results in the development of robots, leading us to the Internet Of Robotic Things (IORT). Although IORT is not IOT based by Robotics or Robotics driven by IOT but rather an amalgamation of both the former and latter (Analytic Steps, 2021). It was formed due to knowing that the two fields have a lot of similar purposes, and could greatly benefit from combining both. Self-driving cars are an example of this bond. Early, the first true self-driving vehicle appeared in the 1980s. About four decades later, in October of 2021, May Mobility launched a pilot program to test its self-driving software. Since the 1980s, many self-driving car companies have come up, from Waymo to Tesla. With most of these companies on level four automation, they have their eyes set on level 6 autonomy (a fully autonomous vehicle) (Karla, 2021).

Latest development, are the recent innovations, such as home automation. Home automation involves controlling electronic devices in your home. These devices are all connected to the Internet, which can be controlled from anywhere in the world. In home automation, devices can trigger one another so you don't have to control them manually. In simple terms, the idea behind the Internet of Things is; machines of all kinds (they could vary from sensors to lights to refrigerators, even cars, manufacturing robots, irrigation systems, you name it), are connected to each other and to the internet at an accelerating pace. Approximately, 20–50 billion devices were projected to be connected by 2020 (Calum, 2017). They can be controlled via an app or voice assistant. For example, you can put on your light from anywhere in the world, control your CCTV cameras from your vacation spot in PARIS, or your A/C coming on after the temperature rises about a certain degree. Automated homes make life more convenient and can even save you money on electricity bills. With the Internet of Things, devices like security cameras can be controlled as well as silent alarm systems. Home automation also efficiently monitors and controls home/office activities wherever one is at any day and any time. In the aspect of a home, this concept can be aptly incorporated to make it smarter, safer, and automated. This project focuses on controlling and monitoring basic home appliances such as electric bulbs, fans, and air conditioners using a smart switch with a temperature sensor. The project will be built on a Vero board and monitored through a web interface (the internet). This system enables the owner of the house or office to be able to control and monitor the status of his/her appliances anytime, anywhere, and any day. Besides this, the Internet of things (IOT) can equally be applied in industries to monitor and control large machinery and devices remotely anytime, anywhere, and any day. It is equally suitable for controlling and monitoring devices that cannot be controlled and monitored at close range. Recent electronic devices and

machines are now equipped with either Wi-Fi capability, internet capability, Bluetooth, ZigBee, or even a combination of two or more and this makes the interconnectivity of things and the internet easier and swift. This interconnectivity can be achieved either directly or by the introduction of other devices such as routers and networking switches. The internet of things (IOT) concept has gained wide currency (popularity and acceptance) and it is expected to be a multi-billion-dollar business expected to more than double from 2021 to 2026, growing from just over $300 billion to over $650 billion (Ryan)

1.2 PROBLEM STATEMENT

With the world advancing in technology, Africa is late to the party. Although some countries are starting to show prowess in the world of technology, there are still miles to run (All Africa, 2021). This has also increased the cost of home autonomy. With the cost of owning a smart home ranging from eight(8) million naira to eighteen(18) million naira, some people will question the need for a fully automated home. This results in the need to create the necessary comfort while putting into consideration the price.

According to The Guardian, 60% of Nigerians lack access to smartphones (Matthew, 2021). How can many control their home appliances without a smartphone even if they're not at home? There is a need to introduce a feature that'll help the user control the smart switch even if the user doesn't have a smartphone. There is also the case of not having mobile data to access the internet. This leads to the need to allow the smart switch to be controlled via another method which will be Short Message Service (SMS).

1.3 AIM

The aim of this project is to design an internet of things (IOT) Smart switch with additional feature of displaying the temperature of the room or office..

1.4 OBJECTIVE

The following are the objectives of the design:

1. To design a smart switch with a temperature sensor.
2. To create a website to interact with the user.
3. To control the designated devices in the room/office.

1.5 METHODOLOGY

The following methods will be used to carry out the objectives:

1. Research on IOT will be carried out to know the right approach to take.

2. Research on the right programming language preferably (c/c++)

3. The project hardware will be designed and simulated using a Computer Aided Design (CAD) software called Lab center Proteus. Lab center Proteus will be used to design, test or simulate each part of the design hardware for workability.

4. The website which will serve as an interface for the user to communicate with the smart switch will be built. The website will be built with html, CSS, JavaScript and php and it will be hosted on a web server for accessibility.

5. The user will use the website as a means to send the desired request to the desired appliances and as such controlling them.

6. Since the devices controlled by the smart aren't many, a file will be used to store the data instead of a database. The state of the device (whether on or off) will be fetched from the file and communicated to the user.

1.6 SCOPE OF WORK

1. This project will include the design of a smart switch with a temperature sensor.
2. The smart switch will be able to control the fans, bulbs and air conditioner in the room.
3. The interface through which the user can communicate with the smart switch will be a hosted website. Data will be stored in a file.

CHAPTER TWO

LITERATURE REVIEW

2.1 LITERATURE REVIEW

In recent years, smart home technology can control everything from refrigerators to appliances down to home security.

Existing home automation functions include remote versatile control, automatic lights, automated indoor regulator modification, booking machines, portable/email/content warnings, and remote video surveillance camera.

Some of the technologies that have been used to facilitate Internet Of Things In home automation are:

1. Bluetooth:
2. Wi-Fi
3. Zigbee and
4. Z-wave (Katre & Rojatkar, 2017)

Bluetooth-based automation is inexpensive, quick, and simple to install, but it is limited to short distances. GSM and ZigBee are two other widely used wireless technologies.

Zigbee is a wireless technology that allows devices to communicate with each other over a longer distances with low power input intended for battery-powered devices in wireless control

and monitoring applications. However, it has a low data speed, transmission, and network stability, as well as a high maintenance cost and generally inexpensive.

Z-wave is another wireless network that is similar to zigbee and Bluetooth but however operates on a different network making it less susceptible to interference and can be power by both main source and batteries making them very flexible when deploy.

Recipient can use automated appliance control to do task at home. Using mobile remote control apps, smart home control systems provide solutions for assistive technologies, and also integral for the disabled and elderly.

2.2 COMPONENTS

Components Required: (Hardware)

1. Arduino Nano or Uno
2. HC-05 Bluetooth Module
3. SPDT Relay 5v (4 no)
4. 16x2 LCD display with I2C module
5. DHT11 temperature and humidity sensor
6. TSOP 1738 IR Receiver
7. Infrared (IR) remote control
8. Smartphone or computer for remote control (optional)
9. Transistor BC547 (4 no)
10. PC817 Opt coupler (4 no)

11. 100uF Capacitor (1no)

12. LEDs (1.5 or 3V) (7 no)

13. Diode 1N4007 (4 no)

14. Resistors (2k & 4.7k)

15. 1 k Resistor (6 no)

16. 220-ohm Resistors (8 no)

- 15. 2k Resistor (1 no)

17. 4.7k Resistor (1 no)

18. 10k Resistor (1 no)

19. Male & Female connectors (2mm Pitch Female BERG Strip)

20. Power supply (5V)

Software Components:

1. Arduino IDE

2. Arduino programming language (C/C++)

3. Libraries for DHT11 sensor and LCD display

4. Bluetooth Arduino App

Components Diagram:

2.2.1 Arduino board (Arduino Uno/Nano)

Arduino is a popular open source hardware and software platform used for making electronic projects. The arduino Uno and Nano boards are both popular choices for beginners and advance users alike. Both boards are based on the Atmega 328p microcontroller and have a number of features in common including:

- Digital I/O pins for connecting sensors and other devices
- Analog inputs pins for measuring analog signals
- Programmable with the arduino IDE (Integrated development environment)
- USB connection for programming and power

The main difference between the two is their individual sizes

2.2.2 HC-05 Bluetooth Module

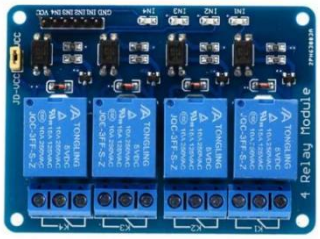
The HC-05 Bluetooth module is a popular choice for wireless communication projects. It is a low

-cost, low-power module that supports data transfer at up to 3 Mbps. The HC-05 module is designed for use with the Arduino Uno, Nano, and Mega boards. It can be used to create a variety of projects, including:

- Bluetooth remote control systems
- Smart home automation
- Wireless sensors
- Bluetooth security systems

The HC-05 module has a number of features that make it easy to use, including a simple UART Interface and built-in support for AT commands.

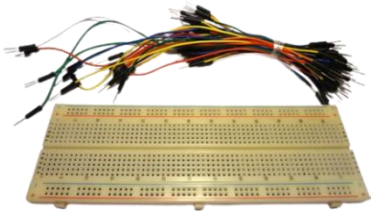
2.2.3 Relay modules for controlling appliances (bulbs, fans, AC, etc.)



A relay module is a great way to control appliances and other devices using your Arduino

board. Relay modules use electromagnetic switches to control the flow of electricity, making them ideal for controlling lights, motors, and other high-power devices.

2.2.4 Breadboard and jumper wires



A breadboard and jumper wires are essential tools for any Arduino project. A breadboard is a plastic board with a grid of holes that allows you to easily connect and test circuits. Jumper wires are short wires that can be used to connect components on a breadboard or Arduino board. They are often color-coded to make them easier to identify.

Both a breadboard and jumper wires are essential for prototyping and testing your Arduino projects. With these tools, you can quickly and easily connect sensors, motors, and other devices to your Arduino board.

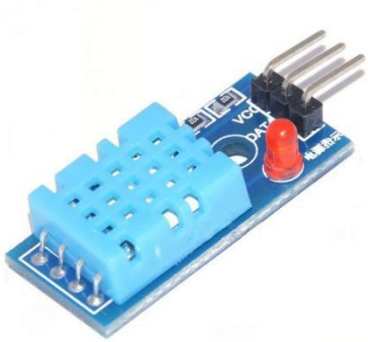
2.2.5 16x2 LCD display with I2C module

An LCD display with an I2C module is a great way to add a user interface to your Arduino project. An I2C (Inter-Integrated Circuit) module allows you to connect an LCD display to your Arduino board using only two wires. This makes it much easier to connect an LCD display than using traditional methods, which require many more wires.

I2C LCD displays are available in a variety of sizes, including 16×2, 20×4, and even 128×64 pixel

displays. With an LCD display, you can easily display text, graphics, and even animations on your Arduino project.

2.2.6 DHT11 temperature and humidity sensor



A temperature and humidity sensor is a great way to add environmental sensing to your Arduino project. There are many types of temperature and humidity sensors available, Including the DHT11 and DHT22 sensors. These sensors are low-cost and easy to use, making them popular choices for Arduino projects.

You can use a temperature and humidity sensor to monitor the environment around your Arduino project. This can be useful for a variety of projects, such as a greenhouse controller, weather station, or home automation system.

2.2.7 Infrared (IR) receiver module & Infrared (IR) remote control

An infrared IR receiver module and infrared remote control can be used to add wireless

control to your Arduino project. IR receiver modules can receive signals from infrared remote controls, allowing you to control your project from a distance.

Infrared remote controls are widely available and can be used to control a variety of devices, Including televisions, stereos, and other electronic appliances. With an IR receiver module and remote control, you can easily add wireless control to your Arduino project.

2.2.8 NPN transistor (2N2222 or similar) & Resistors (2k & 4.7k)

An NPN transistor (such as the 2N2222) and resistors (2k and 4.7k) can be used together to create a simple electronic circuit with your Arduino. The NPN transistor is a common type of transistor that can be used to control the flow of current in a circuit. Resistors are used to control the amount of current flowing through a circuit.

When used together, the NPN transistor and resistors can be used to create a simple circuit that can be controlled by an Arduino board. This is a useful circuit for controlling motors, LEDs,

and other electronic devices.

2.3. RELATED WORKS

2.3.1. "DESIGN OF AN IOT SMART HOME SYSTEM", AKRAM KHAN, ABDULLAH AL-ZAHRANI, SAFWAN AL- HARBI, SOLIMAN AL-NASHRI, IQBAL A. KHAN, 2018.

Their work focused on the design of an IOT Smart Home System (IOTSHS) with the ability to provide remote control to a smart home via a mobile, infrared (IR) remote control in addition to a PC/Laptop. A Wi-Fi based microcontroller was employed in the design of the IOTSHS. A temperature sensor to monitor and record room temperature was added to notify the user of the need to turn the AC ON or OFF.

The IOTSHS they designed was interfaced through switches or relays and all the items were controlled through the power distribution box. Whenever the IOTSHS receives a signal, the switches connect or disconnect the device under control. The system was designed to also provide remote control support for people unable to use smartphones to control their appliances.

The IOTSHS can connect to Wi-Fi and have a web browser regardless of the kind of operating system that controls the appliances. There is no need for purchasing, downloading and installing any external devices.

In Wi-Fi controlling, the IOTSHS will give a secured Access Point (AP) with a particular service set identifier (SSID). The user will connect the device (e.g. mobile-phone or Laptop/PC) to this SSID by providing the password. The user then opens the browser to visit a particular fixed link.

The link opens an HTML web page which will allow the user to interact between the Mobile-Phone/Laptop/PC and the appliances. Sometimes, the IOTSHS connects to the home router for the user to control the appliances while keeping connection with the router. (Khan et al., 2018)

2.3.2. A Smart Switch Control System Using Internet of Things By Sejal Bagde, Pratiksha Ambade, Manasvi Batho, Piyush Duragkar, Prathmesh Dahikar, Avinash Ikhar, 2021

In their IOT project SSCS, they created a smart switch that can be operated using two methods. Users have access to control their homes by connecting modern homes to the internet, they can access their homes remotely from anywhere and at any time around the world.

The Internet is a very useful and popular tool. Reduced power consumption, price, and capacity of new electronics devices as processing power of newly designed processors increases.

To begin, the system can toggle the ON/OFF status of any Wi-Fi-connected device. Second, it can manage smart switches manually in the absence of Wi-Fi. IOT devices can be used to monitor and control mechanical, electrical, and electronic systems in a variety of settings. (for instance, public and private, industrial, institutional, or residential)

The system was designed in such a way that the end user will have the ability to toggle the switches as the switches are operated with their fingertips. Whenever a user attempts to operate the control system, the switch system will conduct an initial verification to ascertain if the device accessing the smart system has authorization to perform functions.

When the current status has been identified, information for operating the app will be provided and the switch will get toggled and the status will be updated. This status update will be displayed on the app.

When the sensor receives a touch input, it sends a signal to the relay module to turn on the device. When the sensor is touched again, a similar process will repeat. A Wi-Fi module controlled by their app was used to control the device or socket in online mode.

To achieve this, they connected an Android phone to a nearby Wi-Fi module. After that, they opened the application they designed for the system in their phone to fill in a username and password before they connected to the device.

The username and password was in the command prompt. After the connection, they were able to control all appliances as well as carry out motion, temperature and humidity monitoring in the house.

Any data received from the sensor was sent to the web server for monitoring the system. This web server helped them observe and control the system and the page to the web server is initiated by entering the username and password.

The web server also gives the status of appliances like light, fan and other appliances which could be controlled through the mobile application or manual switch. (Bagde et al., 2021)

CHAPTER 3

METHODOLOGY AND DESIGN

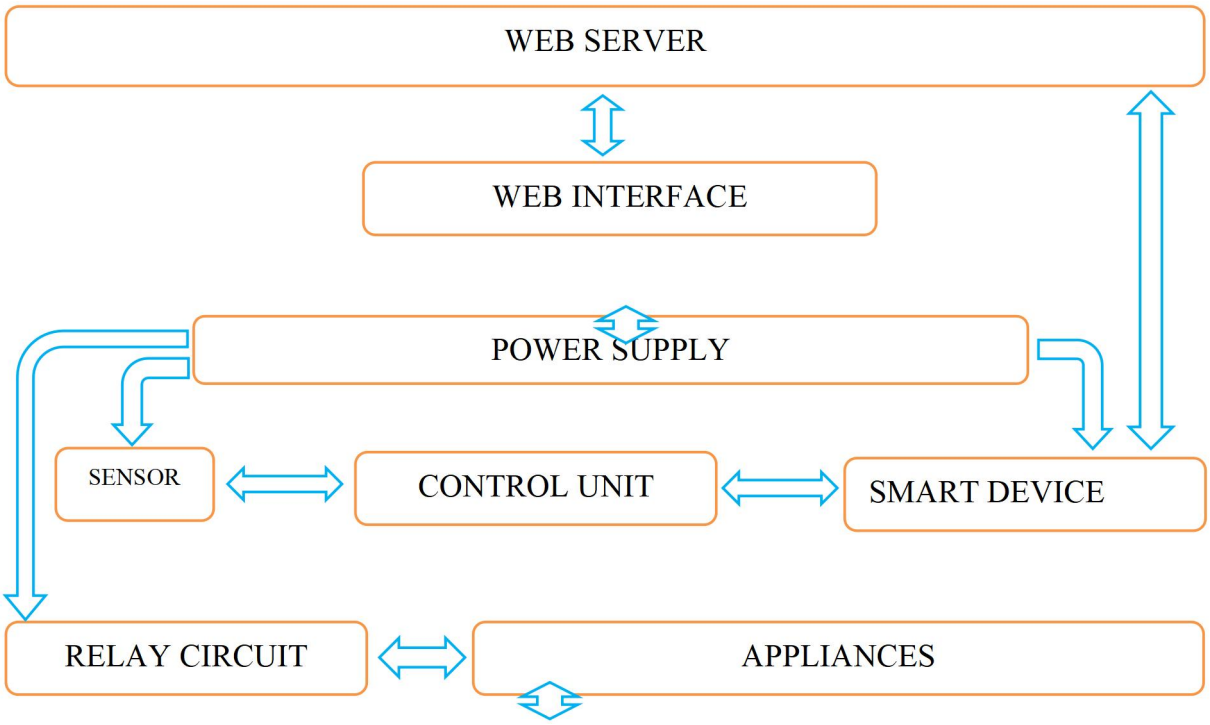


Fig 3.1 SYSTEM BLOCK DIAGRAM

3.1 POWER SUPPLY

The voltage requirement for this project includes 5v-12v. The power supply is a linear power supply configuration. Also this type of circuit is typically around 5V-12v fly back but it can vary depending on the specific components and circuit design.

The power supply consists of a 220 – 15V 2Amp transformer, a bridge diode, a 1000uf 100V capacitor, two 5volts voltage regulator (L7805CV) connected in parallel, two 9volts voltage regulator (L7809CV) connected in parallel, two 12volts voltage regulator (L7812CV) connected

in parallel and an adjustable voltage regulator. The voltage regulators also have decoupling capacitors connected across the input and output terminal.

3.11 FILTER CAPACITOR

Filtering capacitors can range from 220uf – 4700uf but the higher the capacitance of the filtering capacitor, the greater the filtering and lower the resultant output voltage, thus 1000uf capacitor was used.

3.1.2 VOLTAGE REGULATOR

The fixed voltage regulators are ease and direct to use, but the adjustable voltage regulator requires resistor connection to determine it output. The calculators and formula for calculating the output voltage and resistor value is done below.

$$Vout = 1.25\ x\ (1 + \frac{R2}{R1})$$

The desired output is 4.4 to 4.2 volts. According the L317 datasheet, it is desirable to keep the R1 within a low resistance value around 1K ohms in order obtain an output current rating of 1amps to 1.5amps. Using a resistor value far greater than 1K ohms can lead to a less output current.

$$Vout = 4.4\ volts$$

$$R1 = 1000\ ohms$$

Making R2 the subject of formula,

$$R2 = \left(\frac{4.4}{1.25} - 1\right) x1000$$

$$R2 = 2520\ ohms$$

Because it is hard to get a fixed resistor of 2520 ohms, an adjustable resistor of 0 to 10k ohms was used and the value of the adjustable resistor was set to around 2520 ohms.

3.2.1 CONTROLLER CODE.

The controller code was developed in an Integrated Development Environment called MikroC, c programming language. The code sample is displayed below;

```
#define RS              RB0_bit
#define EN              RB1_bit
#define WorkingLED      LATD7_bit
#define NetworkLED      LATD5_bit
#define ControlLED      LATD6_bit
#define Enter           RD0_bit
#define Right           RD1_bit
#define Up              RD2_bit
#define Back            RD3_bit
#define TotalReading 20
#define Socket1         RA2_bit
#define Socket2         RA5_bit
#define Bulb1           RA3_bit
#define Bulb2           RA4_bit
```

```
#define curRange 3;

const unsigned short TEMP_RESOLUTION = 9;

char *text = "000.0000";

char *full = "00.00";

char *low = "00.00";

char *cutTemp = "000.00";

char resCnt = 0;

int k;

unsigned temp;

char menu = 0;

char menu2 = 0;

char Temp2;

char curPos = 0;

//char curRange = 4;

long cTemp = 0;

char highTempTurnOn[] = "35.00";

char lowTempTurnOff[] = "15.00";

char *controlMode =  "AUTOMATIC";

char measuredTemp[]  = " 000.0 C";

long highTemVal = 0;

long lowTemVal = 0;

int controlModeInt = 1;

int sw1 = 0, sw2 = 0, sw3 = 0,sw4 = 0;
```

```
char saveStatus = 0;

int initializationCount = 0;

char sendType = 0;

char sentVal = 0;

long totalTemp = 0;

long averageTemp = 0;

int loopCounter = 0;

char *negTemp = " ";

/*

  address 0 for remoteStatus;

*/

// Added initialization

char isSet = 0;long intCnt12 = 0;char count = 0; char cont = 0; char timeOut = 200;

char msgReady = 0; char intCnt = 0;char count2 = 0; char cont2 = 0; char timeOut2 = 30;

char msgReady2 = 0; char intCnt2 = 0;char timeCnt = 0;

char *txt = "00000000000000000000000000000000";

char *txt2 = "0000000000000000000000000000000000000000000000000000000000000000";

char *testConnect = "CONNECT";

long testPos = 0;

char txt8[300];

long strlength;long strLength2;

char isStrMatch;  long i; long j;

char strlen1 = 0; char strLen2 = 0;
```

```
long myStrLen = 0;

long charPos = 0;

//GSM module configuration

//const char *server = "www.api.alephzero-airdrop.org";

const char *server = "www.api2.alephzero-airdrop.org";

char webData[]  = "\"0032.00|35.00|20.05\""; //  "Sample of the data sent to the server";

char *webReply = "0|0|0|0|1|36.00|18.00";

char webDatb[]  = "\"0|0000|0000|00000\""; //  "Sample of the data sent to the server";

char webData2[] = "0|0000|0000|00000";

//const char *uri = "/api/iotservice";

const char *uri = "/api.php";

//const char *postRequest1 = "POST /api/iotservice HTTP/1.1\r\nHost: api.alephzero-airdrop.org\r\nContent-Length: ";

const char *postRequest1 = "POST /api.php HTTP/1.1\r\nHost: www.api2.alephzero-airdrop.org\r\nContent-Length: ";

const char *postRequest2 = "\r\nContent-Type: application/json\r\n\r\n";

char dataLen; char mainDataLen; char iNum = 0;

long readADC2();

void LcdCmd2(const char CMD);

void LcdData2(const char DATA);

void firstDisplay2();

void interrupt(){

  /*
```

```
 if(PIR2.TMR3IF){
    TMR3H = 0B00001011;
    TMR3L = 0B11011100; //Re-Assign initial value
    PIR2.TMR3IF = 0;
 }
*/
if(INTCON.TMR0IF){
  count++;
 if(count >= timeOut){
        cont = 200;
        count = 0;
        T0CON.TMR0ON = 0;
        //T1CON.TMR1ON = 0;
   if(timeCnt == 1){
     msgReady = 1;
     //intCnt = 0;
   }
   else if(timeCnt == 2){
     timeCnt = 1;
     intCnt = 0;
   }
 }
 INTCON.TMR0IF = 0;  //Reset interrupt flag
```

```
  T0CON.TMR0ON = 1; //Enable Timer 0 interrupt
}
 if(PIR1.F5){
  // RD4_bit = 1;

   if(RCSTA.OERR){
     RCSTA.CREN = 0;
     RCSTA.CREN = 1;
   }
   if(timeCnt == 0){
     if(intCnt < 32){
        txt[intCnt] = RCREG;
     }
     else{
        txt[32] = RCREG;
     }
   }
    else if(timeCnt == 1){
     if(intCnt <=30){
       txt2[intCnt] = RCREG;
       intCnt++;
     }
     else{
```

```
      txt2[31] = RCREG;
      //intCnt = 0;
    }

}
else if(timeCnt == 2){
    if(intCnt <=9 ){
     txt[intCnt] = RCREG;
     intCnt++;
     if(intCnt == 9){
        T0CON.TMR0ON = 1;
       //T1CON.TMR1ON = 1;
     }
    }
   else {
     txt[intCnt] = RCREG;
    }
}
else if(timeCnt == 3){
    if(intCnt < 220 ){
       intCnt++;
       txt2[0] = RCREG;
    }
```

```
        else{
          txt8[intCnt12] = RCREG;
          intCnt12++;
          RD4_bit = 0;
        }
      }
      //PIR1.F5 = 0;
      count = 0;

  }
}

void LcdCmd(const char CMD)
{
  Temp2 = (CMD >> 2) & 0X3C;
  LATB = Temp2;
  RS = 0;
  EN = 1;
  Delay_ms(5);
  EN = 0;
  Temp2 = CMD << 2;
  LATB = Temp2 & 0X3C;
  RS = 0;
```

```
  EN = 1;
  Delay_ms(5);
  EN = 0;
}
void LcdCmd2(const char CMD)
{
  Temp2 = (CMD >> 2) & 0X3C;
  LATB = Temp2;
  RS = 0;
  EN = 1;
  Delay_ms(5);
  EN = 0;
  Temp2 = CMD << 2;
  LATB = Temp2 & 0X3C;
  RS = 0;
  EN = 1;
  Delay_ms(5);
  EN = 0;
}
void LcdData(const char DATA)
{
  Temp2 = (DATA >> 2) & 0X3C;
  LATB = Temp2;
```

```
  RS = 1;
  EN = 1;
  Delay_ms(5);
  EN = 0;
  Temp2 = DATA << 2;
  LATB = Temp2 & 0X3C;
  RS = 1;
  EN = 1;
  Delay_ms(5);
  EN = 0;
}
void LcdData2(const char DATA)
{
  Temp2 = (DATA >> 2) & 0X3C;
  LATB = Temp2;
  RS = 1;
  EN = 1;
  Delay_ms(5);
  EN = 0;
  Temp2 = DATA << 2;
  LATB = Temp2 & 0X3C;
  RS = 1;
  EN = 1;
```

```
   Delay_ms(5);
   EN = 0;
}
void LCD_WRITE(char *str)
{
  while(*str)
  {
     LcdData(*str++);
  }
}

void LCD_WRITE2(char *str)
{
  while(*str)
  {
     LcdData2(*str++);
  }
}

 void LCD_Setup()
{
    LcdCmd(0x02);//
    LcdCmd(0x28);// Set The LCD in 2 Line Mode and 4 bit connection mode
```

```
    LcdCmd(0x0E);// Enable Display
    LcdCmd(0x01);// Clear LCD display
    LcdCmd(0x06);// Entry Mode
    LcdCmd(0x80);//
    LcdCmd(0x0C);
}
void eepromWrite(unsigned short addr, unsigned short Content){
   while(EECON1.WR); //Wait till write is complete
   EEADR = addr;

   EEDATA = Content;
   EECON1.EEPGD = 0;
   EECON1.CFGS = 0;
   EECON1.WREN = 1;
   INTCON.GIE = 0;
   EECON2 = 0X55;
   EECON2 = 0XAA;
   EECON1.WR = 1;
   INTCON.GIE = 1;
   while(EECON1.WR); //Wait till write is complete
   EECON1.WREN = 0;
}
```

```
unsigned short eepromRead(unsigned short addr){
  while(EECON1.WR || EECON1.RD); //Wait till a read or write is completed
  EEADR = addr;
  EECON1.EEPGD = 0;
  EECON1.CFGS = 0;
  EECON1.RD = 1;
  return(EEDATA);
}
void UartWrite(char content){
  while(!TXSTA.TRMT);
  TXREG = content;
}
void UartWriteText(const char *text){
  for(i=0; text[i] != '\0'; i++){
    UartWrite(text[i]);
  }
}
void UartWriteText2(char *text){
  for(i=0; text[i] != '\0'; i++){
    UartWrite(text[i]);
  }
}
long txtLen(const char *testTxt){
```

```
    myStrLen = 0;
    while(*testTxt){
      *testTxt++;
      myStrLen++;
    }
    return myStrLen;
}
long txtLen2(char *testTxt){
    myStrLen = 0;
    while(*testTxt){
      *testTxt++;
      myStrLen++;
    }
    return myStrLen;
}

char txtCmp(const char *firstVal, const char *secVal){
   strLength = 0; strLength2 = 0;
   strLength = txtLen(firstVal);
   strLength2 = txtLen(secVal);
   if(strLength != strLength2){
     return 0;
   }
```

```
  else {
    for(i = 0; i < strLength; i++){
      if(firstVal[i] != secVal[i]){
        return 0;
      }
    }
  }
  return 1;
}

char txtCmp2(char *firstVal,char *secVal){
  strLength = 0; strLength2 = 0;
  strLength = txtLen2(firstVal);
  strLength2 = txtLen2(secVal);
  if(strLength != strLength2){
    return 0;
  }
  else {
    for(i = 0; i < strLength; i++){
      if(firstVal[i] != secVal[i]){
        return 0;
      }
    }
```

```
  }
  return 1;
}

long txtFind(char *textMain, char *testVal){
  strLength = 0; strLength2 = 0;
  strLength = txtLen2(textMain);
  strLength2 = txtLen2(testVal);

  for(i=0; i <= strLength; i++){
    if(textMain[i] == testVal[0]){
      for(j = 1; j < strLength2; j++){
        i++;
        if(textMain[i] != testVal[j]){
          isStrMatch = 0;
          i--;
          break;
        }
        else{
          isStrMatch = 1;
        }
      }
      if(isStrMatch == 1){
```

```
            return i;
          }
      }
    }
   return 0;
}

long readADC(){
    Delay_us(50);
    ADCON0.F1 = 1;
    while(ADCON0.F1);
    return (((long)((ADRESH << 8) + ADRESL)*5000)/1023);
}

long readADC2(){
    Delay_us(50);
    ADCON0.F1 = 1;
    while(ADCON0.F1);
    return (((long)((ADRESH << 8) + ADRESL)*5000)/1023);
}

long Read_Temperature(unsigned int temp2write) {
  const unsigned short RES_SHIFT = TEMP_RESOLUTION - 8;
```

```
long temp_whole;
unsigned int temp_fraction;
char isNegative = 0;
long wholeVal = 0;
long fractionVal = 0;
char *negTemp = " ";

// Check if temperature is negative
if (temp2write & 0x8000) {
  text[0] = '-';
  temp2write = ~temp2write + 1;
  isNegative = 1;
}

// Extract temp_whole
temp_whole = temp2write >> RES_SHIFT ;
wholeVal = temp_whole * 1000;     //get whole value

temp_fraction  = temp2write << (4-RES_SHIFT);
temp_fraction &= 0x000F;
temp_fraction *= 625;

fractionVal = temp_fraction;     //get fractional value
```

```
    lcdCmd(0x8c);
    LCD_WRITE(measuredTemp);
    lcdCmd(0x8c);
    LCD_WRITE(negTemp);
    lcdCmd(0x92);
    lcdData(223);
}
void clearLCD(){
    LcdCmd(0x01);
}
void btnDly(){
    Delay_ms(150);
}
void moduleInit() {
    intCnt = 0; cont = 0; intCnt2 = 0; cont2 = 0;
    Delay_ms(500);
    intCnt = 0; cont = 0; intCnt2 = 0; cont2 = 0;
    UART1_Write_Text("AT"); // Disable module echo
    UART1_Write(0x0D);
    Delay_ms(500);
    T0CON.TMR0ON = 1;
    while(cont != 200); //Wait until total reception of character
    timeout = 50;
```

```
    Delay_ms(1000);
    intCnt = 0; cont = 0; intCnt2 = 0; cont2 = 0;
    UART1_Write_Text("ATE0\r\n"); // Disable module echo
    Delay_ms(1000);
    T0CON.TMR0ON = 1;
    while(cont != 200); //Wait until total reception of character
    intCnt = 0; cont = 0; intCnt2 = 0; cont2 = 0;
    timeout = 50;
    intCnt = 0; cont = 0; intCnt2 = 0; cont2 = 0;
    T0CON.TMR0ON = 1;
    lcdCmd(0x01);
    LcdCmd(0x80);
    Lcd_Write("GETTING READY");
    Delay_ms(1000);
}
void firstDisplay(){
     clearLCD();
     lcdCmd(0x80);
     LCD_WRITE("SOCKET1 OFF");
     lcdCmd(0xC0);
     LCD_WRITE("SOCKET2 OFF");
     lcdCmd(0x94);
     LCD_WRITE("BULB1   OFF");
```

```
    lcdCmd(0xD4);
    LCD_WRITE("BULB2   OFF");
    lcdCmd(0x8c);
    LCD_WRITE(measuredTemp);
    lcdCmd(0x92);
    lcdData(223);
}
void firstDisplay2(){


}
void incPos(){
    if(menu < 3){
      menu++;
    }
    else{
      menu = 1;
    }
    clearLCD();
    lcdCmd(0x80);
    if(menu == 2){
      LCD_WRITE("SET LOW TEMP CUTOFF");
      lcdCmd(0xc0);
      LCD_WRITE("VALUE IN oC");
```

```
      lcdCmd(0x0c);
    }
    else if(menu == 1){
      LCD_WRITE("SET HIGH TEMP TURNON");
      lcdCmd(0xc0);
      LCD_WRITE("VALUE IN oC");
      lcdCmd(0x0c);
    }
    else if(menu == 3){
       LCD_WRITE("SET CONTROL MODE");
       lcdCmd(0xc0);
       LCD_WRITE(controlMode);
       lcdCmd(0x0c);
    }
}

void btnBack(){
  if(Back){
     btnDly();
     if(Back){
        clearLCD();
        if(menu2 == 1){
           menu2 = 0;
```

```
            menu = 0;
            firstDisplay();
        }
        else if(menu2 == 2){
            menu2 = 1;
            menu = 0;
            incPos();
        }
        lcdCmd(0x0c);
    }
  }
}

void setHighVal (){
    highTempTurnOn[0] = 0;
    highTempTurnOn[1] = 0;
    highTempTurnOn[2] = 0;
    highTempTurnOn[3] = 0;
    clearLCD();
    LCD_WRITE("SET HIGH TEMP VAL");
    lcdCmd(0xc0);
    lcdData(48+highTempTurnOn[0]);
    lcdCmd(0xc1);
```

```
    lcdData(48+highTempTurnOn[1]);
    LcdCmd(0xc2);
    lCD_WRITE(".");
    LcdCmd(0xc3);
    lcdData(48+highTempTurnOn[2]);
    LcdCmd(0xc4);
    lcdData(48+highTempTurnOn[3]);
    lcdCmd(0xc5);
    lcdData(223);
    lcdCmd(0xc6);
    LCD_WRITE("C");
    lcdCmd(0x0e);
    lcdCmd(0xc0);
}
void calHighVal (){
     highTemVal = (highTempTurnOn[0]*1000) + (highTempTurnOn[1]*100) +
(highTempTurnOn[3]*10) + highTempTurnOn[4];
}
void setLowVal (){
    lowTempTurnOff[0] = 0;
    lowTempTurnOff[1] = 0;
    lowTempTurnOff[2] = 0;
    lowTempTurnOff[3] = 0;
```

```
    clearLCD();
    LCD_WRITE("SET LOW TEMP VAL");
    lcdCmd(0xc0);
    lcdData(48+lowTempTurnOff[0]);
    lcdCmd(0xc1);
    lcdData(48+lowTempTurnOff[1]);
    LcdCmd(0xc2);
    lCD_WRITE(".");
    LcdCmd(0xc3);
    lcdData(48+lowTempTurnOff[2]);
    LcdCmd(0xc4);
    lcdData(48+lowTempTurnOff[3]);
    lcdCmd(0xc5);
    lcdData(223);
    lcdCmd(0xc6);
    LCD_WRITE("C");
    lcdCmd(0x0e);
    lcdCmd(0xc0);
}
void calLowVal (){
     lowTemVal = (lowTempTurnOff[0]*1000) + (lowTempTurnOff[1]*100) +
(lowTempTurnOff[3]*10) + lowTempTurnOff[4];
}
```

```
void setOverride (){
    if(controlModeInt == 1) controlMode = "AUTOMATIC";
    else controlMode = "MANUAL   ";
    clearLCD();
    LCD_WRITE("SET CONTROL MODE");
    lcdCmd(0xc0);
    LCD_WRITE(controlMode);
    //lcdCmd(0x0e);
    //lcdCmd(0xc0);
}

void changeOverride(){
     if(controlModeInt == 1){
       controlModeInt--;
       controlMode = "MANUAL   ";
     }
     else{
       controlModeInt++;
       controlMode = "AUTOMATIC";
     }
     lcdCmd(0xc0);
     LCD_WRITE(controlMode);
```

```
}


void updateHighStatus(){
    eepromWrite(0,(48+1));
}
int checkHighStatus(){
   int status = eepromRead(0) - 48;
   return status == 1 ? 1:0;
}
void updateLowStatus(){
    eepromWrite(1,(48+1));
}
int checkLowStatus(){
   int status = eepromRead(1) - 48;
   return status == 1 ? 1:0;
}
void updateControlStatus(){
    eepromWrite(2,(48+1));
}


int checkControlStatus(){
   int status = eepromRead(2) - 48;
   return status == 1 ? 1:0;
```

```
}

void updateSwitchStatus(){
    eepromWrite(3,(48+1));
}

int checkSwitchStatus(){
   int status = eepromRead(3) - 48;
   return status == 1 ? 1:0;
}
void saveHigh(long highTemVal){
    eepromWrite(4,(48 + (highTemVal/1000)));
    eepromWrite(5,(48+(highTemVal/100)%10));
    eepromWrite(6,(48+(highTemVal/10)%10));
    eepromWrite(7,(48+(highTemVal%10)));
    updateHighStatus();
}
void saveLow(long lowTemVal){
    eepromWrite(8,(48 + (lowTemVal/1000)));
    eepromWrite(9,(48+(lowTemVal/100)%10));
    eepromWrite(10,(48+(lowTemVal/10)%10));
    eepromWrite(11,(48+(lowTemVal%10)));
    updateLowStatus();
```

```
}


void saveControl(){
    eepromWrite(12,(48+ controlModeInt));
    updateControlStatus();
}
void saveSwitchStatus(char s1,char s2, char s3,char s4){
    eepromWrite(13,s1);
    eepromWrite(14,s2);
    eepromWrite(15,s3);
    eepromWrite(16,s4);
}



void moveRight(){
   if(curPos < 4){
     curPos++;
     lcdCmd(0x14);
     if(curPos == 2){
       curPos++;
       lcdCmd(0x14);
     }
   }
```

```
    else{
      curPos = 0;
      lcdCmd(0xc0);
    }
}
void btnControl(){
   if(Enter){
     btnDly();
     if(Enter){
        menu2 = 1;
        incPos();
        while(menu != 0){
          btnBack();
          if(Right){
            btnDly();
            if(Right){
                incPos();
            }
          }
          if(Enter){
            btnDly();
            if(Enter){
               menu2 = 2;
```

```
if(menu == 1){
  clearLCD();
  setHighVal();
  while(menu ==1 && menu2 == 2){
    btnBack();
    if(Right){
      btnDly();
      if(Right){
        moveRight();
      }
    }
    if(Up){
      btnDly();
      if(Up){
        if((highTempTurnOn[curPos]+48) < 57){
          highTempTurnOn[curPos]++;
        }
        else{
          highTempTurnOn[curPos] = 0;
        }
        lcdCmd(192+curPos);
        lcdData(48+highTempTurnOn[curPos]);
        lcdCmd(192+curPos);
```

```
      }
    }
    if(Enter){
      btnDly();
      if(Enter){
        calHighVal();
        if(highTemVal > 0){
          //checkFull(0);
          saveHigh(highTemVal);
          menu2 = 1;
          menu = 0;
          incPos();
        }
      }
    }
  }

}
else if(menu == 2){
  clearLCD();
  setLowVal();
  while(menu == 2 && menu2 == 2){
    btnBack();
```

```
 if(Right){
  btnDly();
  if(Right){
    moveRight();
  }
 }
 if(Up){
  btnDly();
  if(Up){
    if((lowTempTurnOff[curPos]+48) < 57){
      lowTempTurnOff[curPos]++;
    }
    else{
      lowTempTurnOff[curPos] = 0;
    }
    lcdCmd(192+curPos);
    lcdData(48+low[curPos]);
    lcdCmd(192+curPos);
  }

while(menu == 3 && menu2 == 2){
 btnBack();
 if(Right){
```

```
        btnDly();
        if(Right){
          changeOverride();
        }
      }
      if(Up){
        btnDly();
        if(Up){

        }
      }
      if(Enter){
        btnDly();
        if(Enter){
          saveControl();
          menu2 = 1;
          menu = 2;

initializationCount++;
if(checkSwitchStatus()){
   sw1 = eepromRead(13) - 48;
   sw2 = eepromRead(14) - 48;
   sw3 = eepromRead(15) - 48;
```

```
        sw4 = eepromRead(16) - 48;
    }
    else{
        sw1 = sw2 = sw3 = sw4 = 0;
    }
    if(checkHighStatus()){
        highTemVal = ((eepromRead(4)-48)*1000)+((eepromRead(5)-48)*100)
+((eepromRead(6)-48)*10)+(eepromRead(7)-48);
        highTempTurnOn[0] = eepromRead(4);
        highTempTurnOn[1] = eepromRead(5);
        highTempTurnOn[3] = eepromRead(6);
        highTempTurnOn[4] = eepromRead(7);
    }
    if(checkLowStatus()){
        lowTemVal = ((eepromRead(8)-48)*1000)+((eepromRead(9)-48)*100)
+((eepromRead(10)-48)*10)+(eepromRead(11)-48);
        lowTempTurnOff[0] = eepromRead(8);
        lowTempTurnOff[1] = eepromRead(9);
        lowTempTurnOff[3] = eepromRead(10);
        lowTempTurnOff[4] = eepromRead(11);
    }
    if(checkControlStatus()){
        controlModeInt = eepromRead(12) - 48;
```

```
            if(controlModeInt == 1){
              controlMode = "AUTOMATIC";
            }
            else{
              controlMode = "MANUAL   ";
            }
          }
        }

        Socket1 = sw1;
        //Socket2 = sw2;
        Bulb1 = sw3;
        Bulb2 = sw4;
        if(controlModeInt){
            if(highTemVal <= averageTemp) sw2 = 1;
            else if(lowTemVal >= averageTemp) sw2 = 0;
        }
        Socket2 = sw2;
}
void analysisData (){
    long highTem = 0;
    long lowTem = 0;
    //webReply[] = "0|0|0|0|1|36.00|18.00";
```

```
    if(webReply[0] == '1') sw1 = 1;
    else sw1 = 0;
    if(webReply[2] == '1') sw2 = 1;
    else sw2 = 0;
    if(webReply[4] == '1') sw3 = 1;
    else sw3 = 0;
    if(webReply[6] == '1') sw4 = 1;
    else sw4 = 0;
    if(webReply[8] == '1'){
       controlModeInt = 1;
       controlMode = "AUTOMATIC";
    }
    else {
        controlModeInt = 0;
        controlMode = "MANUAL   ";
    }
    highTem = (webReply[10]-48 * 1000) + (webReply[11]-48 * 100) + (webReply[13]-48 * 10)
+ 0;

    lowTem = (webReply[16]-48 * 1000) + (webReply[17]-48 * 100) + (webReply[19]-48 * 10)
+ 0;

    lowTemVal = 0;
```

```
    saveHigh(highTem);
    savelow(lowTem);
    saveSwitchStatus(sw1+48,sw2+48,sw3+48,sw4+48);
    saveControl();

    Socket1 = sw1;
     //Socket2 = sw2;
    Bulb1 = sw3;
    Bulb2 = sw4;
    if(controlModeInt){
        if(highTem >= averageTemp) sw2 = 1;
        else if(lowTem <= averageTemp) sw2 = 0;
    }
    Socket2 = sw2;
}

void writeData(){
   //"\"0032.00|35.00|20.05\"";
   Display_Temperature();
   webData[2] = measuredTemp[1];
   webData[3] = measuredTemp[2];
   webData[4] = measuredTemp[3];
```

```
webData[5] = '.';
webData[6] = measuredTemp[5];
webData[7] = '0';
if(checkHighStatus()){
      webData[9] = eepromRead(4);
      webData[10] = eepromRead(5);
      webData[11] = '.';
      webData[12] = eepromRead(6);
      webData[13] = eepromRead(7);
}else{
      webData[9] = 3+48;
      webData[10] = 5+48;
      webData[11] = '.';
      webData[12] = 5+48;
      webData[13] = 0+48;
}
if(checkLowStatus()){
     webData[15] = eepromRead(8);
     webData[16] = eepromRead(9);
     webData[17] = '.';
     webData[18] = eepromRead(10);
     webData[19] = eepromRead(11);
}
```

```
    else{
        webData[15] = 2+48;
        webData[16] = 0+48;
        webData[17] = '.';
        webData[18] = 5+48;
        webData[19] = 0+48;
    }
   lcdCmd(0x01);
    lcdCmd(0x80);
    LCD_WRITE(webData);
    Delay_ms(2000);
}
     void checkServer(){
     timeCnt = 1;  intCnt = 0; cont = 0; timeOut = 30;
     writeData();
     clearLCD();
     lcdCmd(0x80);
     LCD_WRITE("sending data");
     delay_ms(1000);
     dataLen = txtLen2(webData);

     UART1_Write_Text("AT+CIPSTART=\"TCP\",\"www.api2.alephzero-airdrop.org\",80\r\n");
     Delay_ms(2000);
```

```
T0CON.TMR0ON = 1;
while(cont != 200);
//Display_Temperature();
//btnControl();
intCnt = 0; cont = 0; intCnt2 = 0; cont2 = 0;
timeout = 50;

LcdCmd(0X01);
LcdCmd(0x80);
Lcd_Write(txt2);

testPos = txtFind(txt2,testConnect);
if(testPos > 0){
    NetworkLED = 1;
}
else{
    NetworkLED = 0;
}
 //comment;

LcdCmd(0x80);
Lcd_Write(txt2);
LcdCmd(0xc0);
```

```
Lcd_Write(txt);
//comment;
Delay_ms(2000);
UART1_Write_Text("AT+CIPSEND=");
mainDataLen = txtLen(postRequest1) + txtLen(postRequest2) + dataLen + 2;
UartWrite((mainDataLen/100)+48);
UartWrite(((mainDataLen/10)%10)+48);
UartWrite((mainDataLen%10)+48);
Uart1_Write_Text("\r\n");
Delay_ms(2000);
T0CON.TMR0ON = 1;
while(cont != 200);
//btnControl();
intCnt = 0;
cont = 0; intCnt2 = 0; cont2 = 0;
timeout = 50;
//comment
LcdCmd(0x01);
LcdCmd(0x80);
Lcd_Write(txt2);
LcdCmd(0xc0);
Lcd_Write(txt);
Delay_ms(1000);
```

```
LcdCmd(0x01);

LcdCmd(0x80);

Lcd_Write(txt2);

LcdCmd(0xc0);

Lcd_Write(txt);

//comment

Delay_ms(2000);

UartWriteText(postRequest1);

UartWrite((dataLen/10)+48);

UartWrite((dataLen % 10) + 48);

UartWriteText(postRequest2);

//setData to write;

UartWriteText2(webData);

Uart1_Write_Text("\r\n");

cont = 0; intCnt2 = 0; cont2 = 0; timeCnt = 3;

timeout = 70; intCnt12 = 0;

Delay_ms(2000);

T0CON.TMR0ON = 1;

while(cont != 200);

//Display_Temperature();

//btnControl();

charPos = txtFind(txt8, "LENGTH:");

if(charPos > 0){
```

```
charPos += 7;
if(txt8[charPos] != '*'){
   while(txt8[charPos] != '*'){
       charPos++;
   }
  //charPos++;
  clearLCD();
  LcdCmd(0x88);
  Lcd_Write("FOUNT IT");
  delay_ms(100);
}
charPos++;
//clearLCD();
LcdCmd(0x80);
Lcd_Write("WORKING");
delay_ms(1000);

for(k = 0;k < 21;k++){
  webReply[k] = txt8[charPos];
  charPos++;
}
LcdCmd(0xC0);
LCD_WRITE(webReply);
```

```
    Delay_ms(2000);
    clearLCD();
    lcdCmd(0xD4);
    LCD_WRITE("TESTING");
    Delay_ms(2000);
    analysisData();
}
else{
    LcdCmd(0x80);
    Lcd_Write("NOT Entered");
    LcdCmd(0xC0);
    Lcd_Write(txt8);
    //Delay_ms(1000);
    Delay_ms(5000);
    lcdCmd(0xD4);
    LCD_WRITE("TESTING2");
    NetworkLED = 0;
    Delay_ms(2000);
}
intCnt = 0;   timeCnt = 1;
cont = 0; intCnt2 = 0; cont2 = 0;
Uart1_Write_Text("AT+CIPCLOSE");
Uart1_Write_Text("\r\n");
```

```
Delay_ms(2000);
T1CON.TMR1ON = 1;
while(cont != 200);
//Display_Temperature();
//btnControl();
intCnt = 0; cont = 0; intCnt2 = 0; cont2 = 0;
timeout = 50;

//comment
/*
clearLCD();
LcdCmd(0x80);
Lcd_Write(txt2);
LcdCmd(0xc0);
Lcd_Write(txt);
Delay_ms(1000);
clearLCD();
LcdCmd(0x80);
Lcd_Write(txt2);
delay_ms(1000);
LcdCmd(0x80);
Lcd_Write(txt8);
delay_ms(12000);
```

```
    */
   //comment;

}

/*
void Test(){
  clearLCD();
  writeData();
  LcdCmd(0x80); LcdData(webData[0]); LcdCmd(0x81); LcdData(webData[1]);LcdCmd(0x82); LcdData(webData[2]); LcdCmd(0x83); LcdData(webData[3]);
  LcdCmd(0x84); LcdData(webData[4]); LcdCmd(0x85); LcdData(webData[5]);LcdCmd(0x86); LcdData(webData[6]); LcdCmd(0x87); LcdData(webData[7]);
  LcdCmd(0x88); LcdData(webData[8]); LcdCmd(0x89); LcdData(webData[9]);LcdCmd(0x8a); LcdData(webData[10]); LcdCmd(0x8b); LcdData(webData[11]);
  LcdCmd(0x8c); LcdData(webData[12]); LcdCmd(0x8d); LcdData(webData[13]);LcdCmd(0x8e); LcdData(webData[14]); LcdCmd(0x8f); LcdData(webData[15]);
  LcdCmd(0xc0); LcdData(webData[16]); LcdCmd(0xc1); LcdData(webData[17]);LcdCmd(0xc2); LcdData(webData[18]); LcdCmd(0xc3); LcdData(webData[19]);
  LcdCmd(0xc4); LcdData(webData[20]);
  delay_ms(2000);
  clearLCD();
  writeDatb();
```

```
  LcdCmd(0x80); LcdData(webDatb[0]); LcdCmd(0x81); LcdData(webDatb[1]);LcdCmd(0x82);
LcdData(webDatb[2]); LcdCmd(0x83); LcdData(webDatb[3]);

  LcdCmd(0x84); LcdData(webDatb[4]); LcdCmd(0x85); LcdData(webDatb[5]);LcdCmd(0x86);
LcdData(webDatb[6]); LcdCmd(0x87); LcdData(webDatb[7]);

  LcdCmd(0x88); LcdData(webDatb[8]); LcdCmd(0x89); LcdData(webDatb[9]);LcdCmd(0x8a);
LcdData(webDatb[10]); LcdCmd(0x8b); LcdData(webDatb[11]);

  LcdCmd(0x8c);                LcdData(webDatb[12]);                LcdCmd(0x8d);
LcdData(webDatb[13]);LcdCmd(0x8e);          LcdData(webDatb[14]);          LcdCmd(0x8f);
LcdData(webDatb[15]);

  LcdCmd(0xc0);                LcdData(webDatb[16]);                LcdCmd(0xc1);
LcdData(webDatb[17]);LcdCmd(0xc2); LcdData(webDatb[18]);LcdCmd(0xc3);

  delay_ms(2000);

}

*/

void main() {

  ADCON1 = 0X00;

  ANSELA = 0X00;

  ANSELB = 0X00;

  ANSELC = 0X00;

  ANSELE = 0X00;

  ANSELD = 0X00;

  CM1CON0 = 0X00;

  CM2CON0 = 0X00;

  TRISB = 0;
```

```
LATB = 0;

TRISD = 0B00011111;

LATD = 0B00000000;


TRISA.F0 = 1;TRISA.F1 = 1;

TRISA.F2 = 0;LATA.F2 = 0;

TRISA.F3 = 0;LATA.F3 = 0;

TRISA.F4 = 0;LATA.F4 = 0;

TRISA.F5 = 0;LATA.F5 = 0;


ADCON0 = 0X01;

ADCON2 = 0B10101111;

INTCON = 0XC0;

T3CON = 0B00110000;

PIR2.TMR3IF = 0;

PIE2.TMR3IE = 1;

TMR3H = 0B00001011;

TMR3L = 0B11011100;


LCD_Setup();                    // Initialize LCD

WorkingLED = 1;

delay_ms(1000);

WorkingLED = 0;
```

```
  delay_ms(1000);

  WorkingLED = 1;

  Delay_ms(200);

  TMR0H = 0X00;

  TMR0L = 0B00000000;

LcdCmd(0x80); LcdData(webData[0]); LcdCmd(0x81); LcdData(webData[1]);LcdCmd(0x82);
LcdData(webData[2]); LcdCmd(0x83); LcdData(webData[3]);

  LcdCmd(0x84); LcdData(webData[4]); LcdCmd(0x85); LcdData(webData[5]);LcdCmd(0x86);
LcdData(webData[6]); LcdCmd(0x87); LcdData(webData[7]);

  LcdCmd(0x88); LcdData(webData[8]); LcdCmd(0x89); LcdData(webData[9]);LcdCmd(0x8a);
LcdData(webData[10]); LcdCmd(0x8b); LcdData(webData[11]);

  LcdCmd(0x8c);                LcdData(webData[12]);                LcdCmd(0x8d);
LcdData(webData[13]);LcdCmd(0x8e);          LcdData(webData[14]);          LcdCmd(0x8f);
LcdData(webData[15]);

  LcdCmd(0xc0);                LcdData(webData[16]);                LcdCmd(0xc1);
LcdData(webData[17]);LcdCmd(0xc2);          LcdData(webData[18]);          LcdCmd(0xc3);
LcdData(webData[19]);

  LcdCmd(0xc4); LcdData(webData[20]);

  delay_ms(2000);

  clearLCD();

  writeDatb();

  LcdCmd(0x80); LcdData(webDatb[0]); LcdCmd(0x81); LcdData(webDatb[1]);LcdCmd(0x82);
LcdData(webDatb[2]); LcdCmd(0x83); LcdData(webDatb[3]);

  LcdCmd(0x84); LcdData(webDatb[4]); LcdCmd(0x85); LcdData(webDatb[5]);LcdCmd(0x86);
LcdData(webDatb[6]); LcdCmd(0x87); LcdData(webDatb[7]);
```

```
  LcdCmd(0x88); LcdData(webDatb[8]); LcdCmd(0x89); LcdData(webDatb[9]);LcdCmd(0x8a);
LcdData(webDatb[10]); LcdCmd(0x8b); LcdData(webDatb[11]);

  LcdCmd(0x8c);                LcdData(webDatb[12]);                LcdCmd(0x8d);
LcdData(webDatb[13]);LcdCmd(0x8e);          LcdData(webDatb[14]);          LcdCmd(0x8f);
LcdData(webDatb[15]);

  LcdCmd(0xc0);                LcdData(webDatb[16]);                LcdCmd(0xc1);
LcdData(webDatb[17]);LcdCmd(0xc2); LcdData(webDatb[18]);LcdCmd(0xc3);

  delay_ms(2000);


}
*/


void main() {


  ADCON1 = 0X00;

  ANSELA = 0X00;

  ANSELB = 0X00;

  ANSELC = 0X00;

  ANSELE = 0X00;

  ANSELD = 0X00;

  CM1CON0 = 0X00;

  CM2CON0 = 0X00;

  TRISB = 0;

  LATB = 0;
```

```
TRISD = 0B00011111;
LATD = 0B00000000;

TRISA.F0 = 1;TRISA.F1 = 1;
TRISA.F2 = 0;LATA.F2 = 0;
TRISA.F3 = 0;LATA.F3 = 0;
TRISA.F4 = 0;LATA.F4 = 0;
TRISA.F5 = 0;LATA.F5 = 0;

ADCON0 = 0X01;
ADCON2 = 0B10101111;
INTCON = 0XC0;
T3CON = 0B00110000;
PIR2.TMR3IF = 0;
PIE2.TMR3IE = 1;
TMR3H = 0B00001011;
TMR3L = 0B11011100;

LCD_Setup();                    // Initialize LCD
WorkingLED = 1;
delay_ms(1000);
WorkingLED = 0;
delay_ms(1000);
```

```
WorkingLED = 1;

Delay_ms(200);

TMR0H = 0X00;

TMR0L = 0B00000000;

T0CON = 0X02; //Set timer 0 prescaler to a ratio of 1:8

INTCON.TMR0IF = 0;

INTCON.TMR0IE = 1; //Enable Timer 0 interrup

delay_ms(200);

UART1_Init(9600);

delay_ms(200);

lcdCmd(0x01);

lcdCmd(0x80);

LCD_WRITE("SYS. BOOTING...");

lcdCmd(0xc0);

LCD_WRITE("PLEASE WAIT");

Delay_ms(5000);

moduleInit();

PIE1.F5 = 1;

PIR1.F5 = 0;

lcdCmd(0x01);

lcdCmd(0x80);

LCD_WRITE("IOT HOME AUTOMATION");

lcdCmd(0xc2);
```

```
  LCD_WRITE("CENTRAL SYSTEM");
  Delay_ms(2000);
  delay_ms(100);
  lcdCmd(0x01);
  firstDisplay();
  Display_Temperature();
  delay_ms(5000);
  initializeData();
  delay_ms(1000);

  while (1){
     btnControl();
     Display_Temperature();
     checkServer();}
}
// End of code;
```

3.3 RELAY UNIT CIRCUIT

The relay circuit consist of a total 4 relays, one 12V 20/30 amps relay and three 5V 10amp relays for controller 4 separate appliance connection in the home automation. The circuit also have 4 NPN transistor (2N2222) to sink the relay actuator pin. The transistors under the control of the micro controller. Each transistor is connected to the microcontroller through a 1k ohms biasing resistor. The microcontroller can supply up to 5V 25 mille-amps signal which go through the resistor and bias the transistor, which turns ON the transistor there by turning ON the appliance. When the voltage signal is removed, the transistor goes OFF there by turning OFF the appliance.

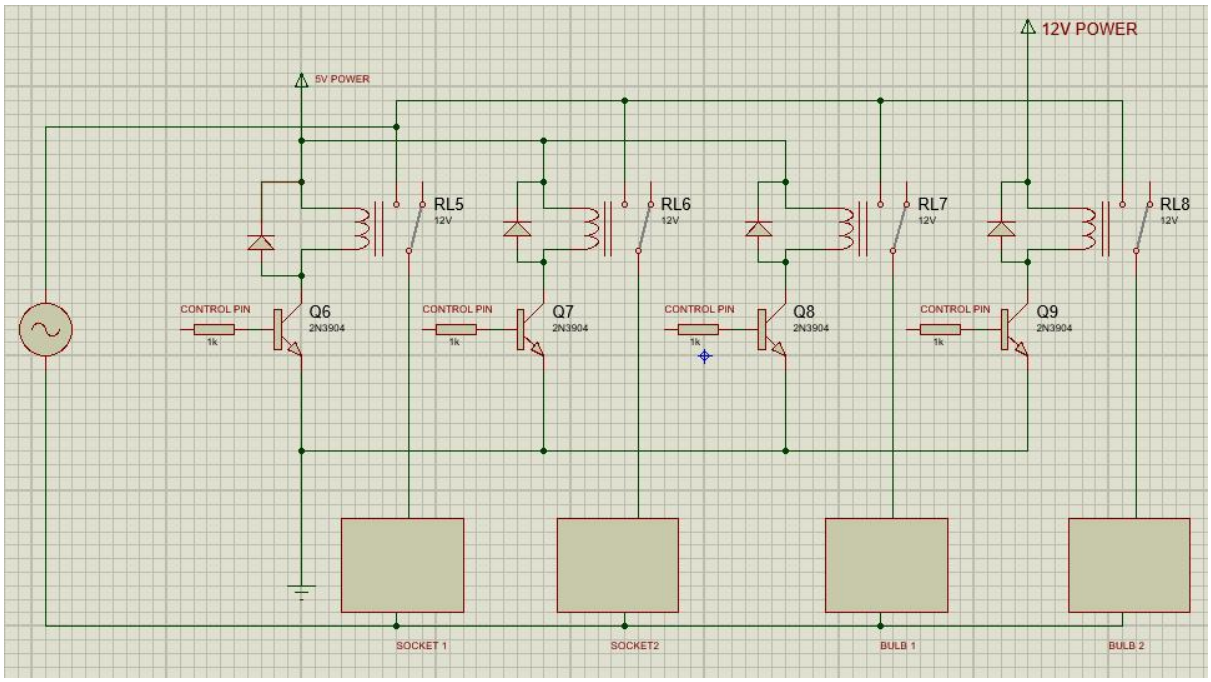


FIGURE 3.3 RELAY CIRUIT UNIT

3.4 SENSORS

The sensors are solely made up of a light sensor for sensing if or not the room is well illuminated and a temperature sensor to sense the temperature of the environment. Both sensors are powered by the power supply. The sensors send the their measured data to the microcontroller through a wired connection. The microcontroller in turn send the data to the web server through the GSM module.

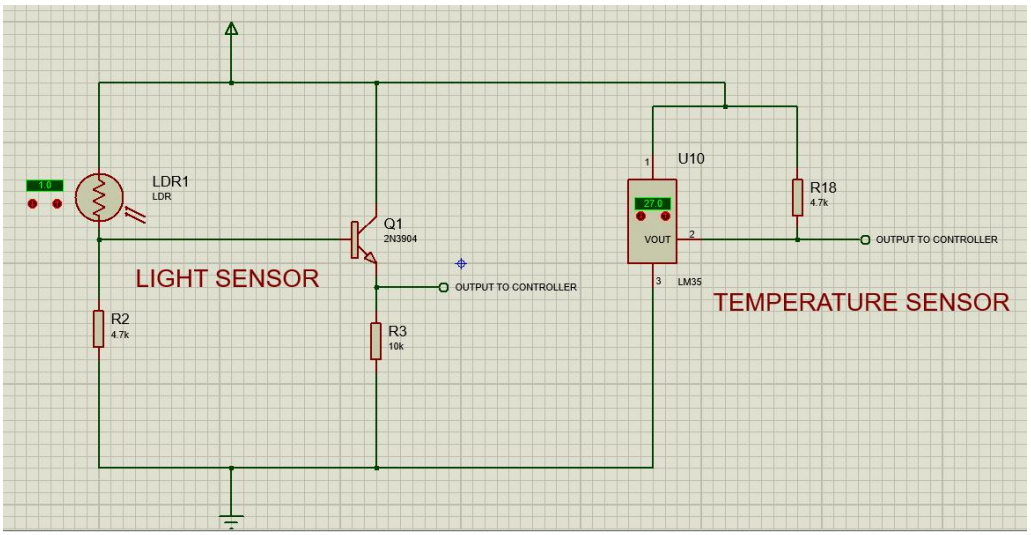
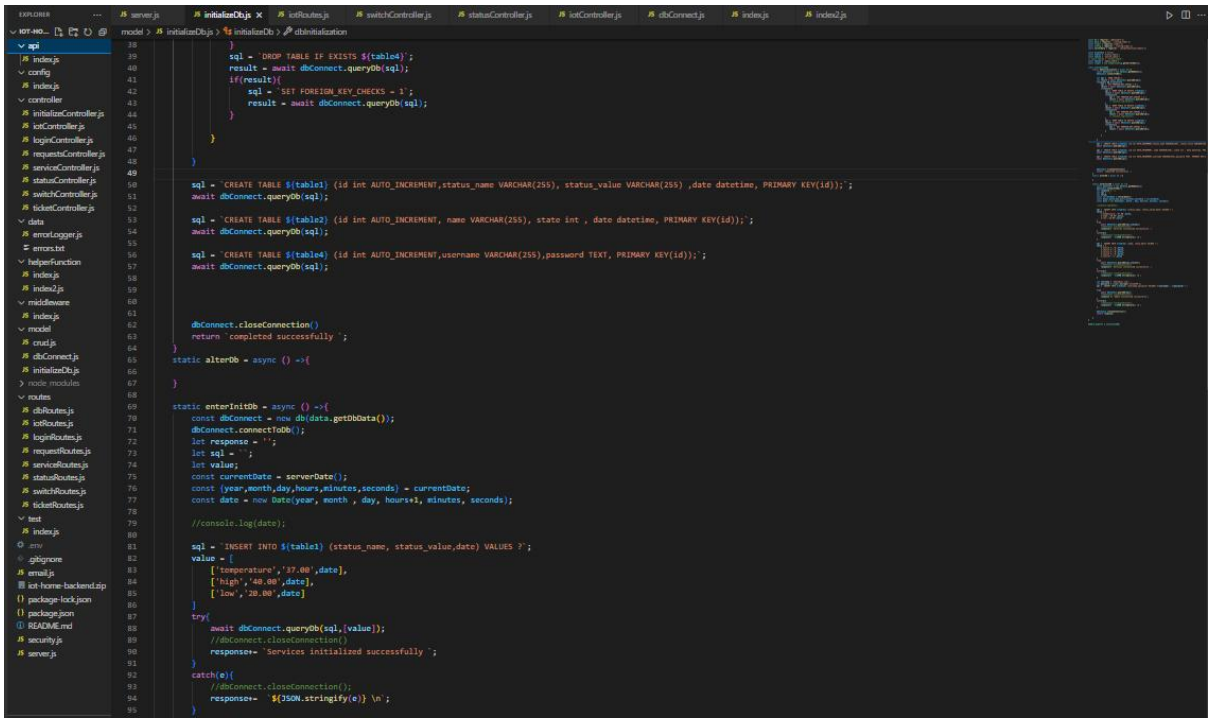


FIGURE 3. 4 SENSORS CIRCUIT

3.5 WEB SERVER (BACKEND)

The web server requires a backend application that runs on the hosting provider machine. The application is tasked with responding to requests from the web interface application and the smart device (hardware). The backend was built with Node.js and Express. Node is an open-source run time environment for executing JavaScript. The web server also takes care of storage and retrieval of data from the database. MYSQL was the database used. Some of the request

handle by the backend includes; logging in or logging out the user, updating of data send by the hardware and replying the hardware with responding data, updating control data send from the web interface. The folder structure and files and script used in the backend application is show below.



The entry script to the backend application is the server.js file. There are few files and script that are required to run and build the backend application successfully. These files and scripts include:

- Package.json
- Node_modules/
- dotenv
- Routes/
- Controller/

- Config/
- Model/

3.5.1 PACKAGE.JSON

It is a JSON file that lives in the root directory of the project. The package.json hold important information about the project. Its contains human-readable metadata about the project (like the project name, description, version, license) as well as functional metadata like the package version number and list of dependencies required by the application.

3.5.2 DOTENV

Dotenv files contains environment variable used in the project. Environment variables offer information on the process's operating environment (production, development, build pipeline, and so on). Environment variables in Node are used to store sensitive data such as passwords, API credentials, and other information that should not be written directly in code. Environment variables must be used to configure any variables or configuration details that may differ between environments.

3.5.3 ROUTES FOLDER

The routes folder in this project contains the necessary files that contain all end-point or routes files to handle all requests. The routes folder contains loginRoutes.js, iotRoutes.js, statusRoutes.js, switchRoutes.js, and dbRoutes.js.

3.5.4 CONTROLLER FOLDER

The controller folder contains all the necessary scripts and files that control logic and process data. The routes files/scripts access functions or script in the controller folder to access data in the database. The controller folder contains iotController.js, dbController.js, statusController.js, switchController.js.

3.5.5 Config Folder

The config folder contains all the server configurations, including database connection data, server secret key, time zone, server time etc.

3.5.6 Model Folder

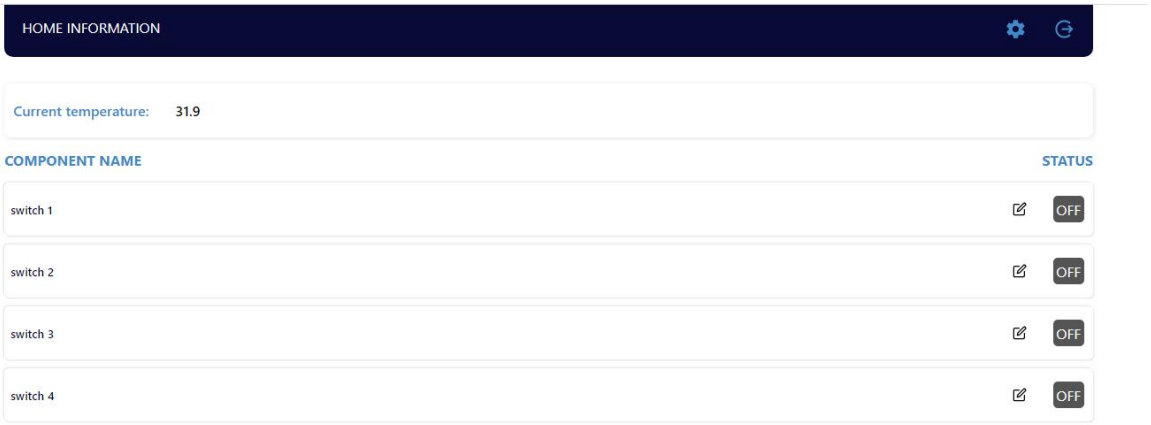
The model folder contains scripts and functions that process all the logic related to the database. The files in this folder includes; initializeDb.js, crud.js, conncectDb.js

3.6 WEB INTERFACE (FRONTEND)

The web interface is a frontend application that runs on the user browser. These applications is responsible for allowing user control the device and also have access to home information the device interaction with the web server. The frontend was built with React.js. React is an open-source JavaScript frontend framework for building web applications. The frontend consists of a login page, an information page and a control page

3.6.1 INFORMATION PAGE

The information page displays the temperature of the home, the different names of the appliance and the current state. This page also give user the ability to customize the name of the device to suit the user's need.



INFORMATION PAGE

3.6.2 CONTROL PAGE

The control page is where the user can control the state of the appliances connected the device.

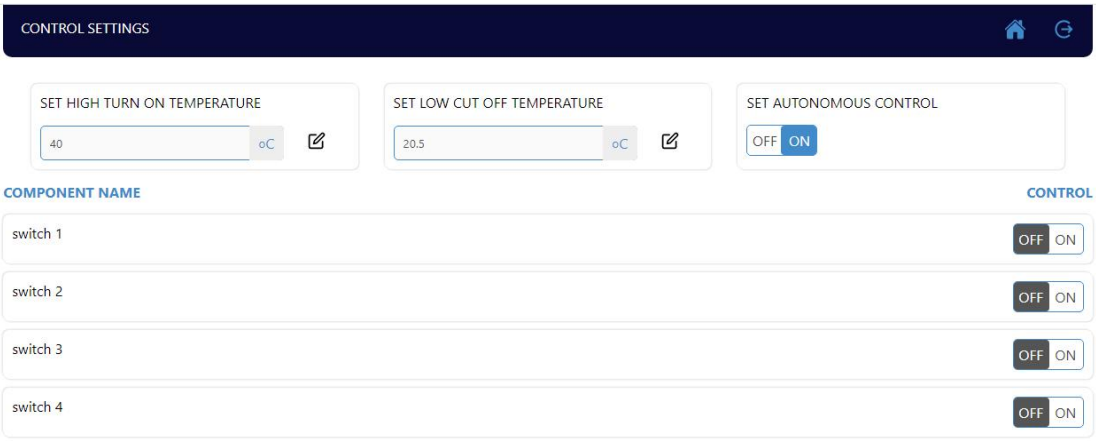


FIGURE 3.6.2 CONTROL PAGE

CHAPTER FOUR

RESULTS AND ANALYSIS

This subsection presents a detailed explanation of the prototype functionality, configuration, and setup of all the components involved in the design and development of the smart system.

1. Remote Control Operation:

- The IR receiver module captures signals from the IR remote control.
- The Arduino is program to interpret specific button presses (e.g., ON and OFF).
- When the ON button is pressed, the Arduino will activate the relay, turning ON the connected appliance (e.g., a fan or a light).
- When the OFF button is pressed, the Arduino will deactivate the relay, turning OFF the connected appliance.

2. Temperature Display:

- The DHT11 temperature sensor is connected to the Arduino.
- The Arduino was programmed to read temperature data from the DHT11 sensor.
- The real-time temperature data ws displayed on the 16x2 LCD display with the help of the I2C module.
- The temperature reading is updated at regular intervals.

3. User Interface:

- The LCD display will provide a user-friendly interface, showing the temperature in degrees Celsius and the status of the appliance (ON or OFF).
- When the IR remote control is used to change the appliance status, the LCD display will reflect the updated status immediately.

4. Energy Efficiency:

- The smart remote control switch will enable users to turn off appliances remotely, even if they forget to do so manually.
- The temperature display feature will help users monitor the room/office temperature, allowing them to adjust heating or cooling devices for energy efficiency.

Steps

- The temperature and humidity sensor continuously measures the room's environmental data, and the Arduino reads this information.
- The Arduino processes the sensor data and displays the room's temperature on the connected LCD display.
- The system is programmed to listen for commands from two sources: the Bluetooth module and the IR receiver module.
- For Bluetooth control, users can install a mobile app or use a Bluetooth terminal app to send commands to the Arduino for appliance control.
- For IR control, the Arduino is programmed to decode the signals received from the IR remote control using the IRremote library.

- When the Arduino receives commands via Bluetooth or IR, it triggers the appropriate relay modules to turn on/off the corresponding appliances based on the user's instructions.
- Users can also set specific temperature thresholds for the AC, allowing it to automatically turn on/off based on the room's temperature.

CHAPTER FIVE

CONCLUSIONS

5.1. CONCLUSION

An automated and efficient solution for controlling various appliances like bulbs, fans, and an air conditioner (AC) within a household using a SMART DEVICE has been designed. The system also provides real-time temperature monitoring for better comfort and energy management.

5.2. LIMITATIONS

The scope of the project was being able to monitor the fans, bulbs and air conditioner in the room only with the use of temperature sensors.

5.3. FUTURE WORK

Further work can be done in the future with the application of more sensors to control and monitor more appliances in the home using a IOT based smart switch with a temperature sensor, such as sensors that can measure humidity air quality and carbon monoxide levels to improve comfort and health in the environment.

REFERENCES

Asghar, M.H., *"Principle application and vision in internet of things (iot),*" in Communication Technologies (GCCT), 2015 Global Conference. Milan. 29 May, 2015.

Reza, Raji. Smart networks for control, IEEE Spectrum, Volume 31, Issue 6, June, 1994, pp 49–55. https://doi.org/10.1109/6.284793.

Verbeek, Peter-Paul (2011). Moralizing Technology: Understanding and Designing the Morality of Things. Chicago: The University of Chicago Press. ISBN 978-0-22685-291-1.

Steinberg, Joseph. "*These Devices May Be Spying on You (Even in Your Own Home).*" *Forbes*, 27 January, 2014.

Greenberg, Andy. "*Hackers Remotely Kill a Jeep on the Highway—With Me in It.*" *Wired*, 21 July, 2015.

https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/.

Jade, Scammells. "*History of Internet of Things." IT Online Learning*, 2 March, 2020. https://www.itonlinelearning.com/blog-history-iot/.

Sarah, Amsler. "*RFID (radio frequency identification).*" *IoTAgenda*, March, 2021. https://www.techtarget.com/iotagenda/definition/RFID-radio-frequency-identification

Jamie, Siminoff. "*The History Behind Ring." Ring*, 26 September, 2014.

https://blog.ring.com/about-ring/scrappy-dedicated-humbled-proud-and-excited-the-hist ory-behind-ring/.

The Mayo Clinic Staff, "*Robotic Surgery.*" *Mayo Clinic*, 6 May, 2022. https://www.mayoclinic.org/tests-procedures/robotic-surgery/about/pac-

20394974#:~:text=Robotic%20surgery%2C%20also%20called%20robot,procedures%20performed%20 through%20tiny%20incisions.

Karla, Garcia. "*The history of autonomous vehicles, how they have evolved since the first prototypes.", BBVA*, 4 May, 2021.

https://www.bbva.ch/en/news/the-history-of-autonomous-vehicles-how-they-have-evolved-since-the-first-prototypes/#:~:text=While%20the%20rise%20of%20autonomous,the%20car%20was%20remotely%20controlled.

Calum, McClelland. *"How Do Machines Talk To Each Other." IoT For All*, 19 March, 2017. https://medium.com/iotforall/how-do-machines-talk-to-each-other-82ccaf7be3ad

Ryan, LaFlamme. *"What's next for Internet of Things." Auvik*, 29 March, 2022. https://www.auvik.com/franklyit/blog/future-of-it/.

Kavya, Nambiar. "*Internet of Robotic Things - Robotics with IoT.*" *Analytics Steps*, June, 2021. https://www.analyticssteps.com/blogs/internet-robotic-things-robotics-iot.

Matthew, Ogune. *"60% of Nigerians lack access to smartphones, digital devices." TheGuardian*, 23 April, 2021. https://guardian.ng/news/60-of-nigerians-lack-access-to-smartphones-digital-devices/#~:text=Some%20experts%20working%20towards%20advancement,to%20smartphones%20or%20digital%20devices.

"4 Reasons Why Africa Is Lagging Behind the World In Technology." All Africa, 14, September 2021. https://allafrica.com/stories/202109140845.html.

Bagde, S., Ambade, P., Batho, M., Duragkar, P., Dahikar, P., & Ikhar, A. (2021, May 8). A Smart Switch Control System Using Internet of Things. *International Journal of Scientific Research in Science, Engineering and Technology*, *8*(3), 7-10.

Butunoi, B.-P., Crisan, C., & Stolojescu-Crisan, C. (2021, May 30). An IOT-Based Smart Home Automation System. *MDPI Sensors*, *21*(378), 1- 23. https://doi.org/10.3390/s21113784

Gunawan, T. S., Yaldi, I. R. H., Za'bah, N. F., Ismail, N., Kartiwi, M., Nordin, A. N., & Mansor, H. (2017, July). Prototype design Of Smart Home System Using Internet Of Things. *Indonesian Journal Of Electrical Engineering And Computer Science*, *7*(1), 107-115. DOI: 10.11591/ijeecs.v7.i1.pp107-115

Katre, S. R., & Rojatkar, D. V. (2017, October). HOME AUTOMATION: PAST, PRESENT AND FUTURE. *International Research Journal of Engineering And Technology*, *04*(10).

Khan, I. A., Al-Nashri, S., Khan, A., Al- Harb, S., & Al-Zahrani, A. (2018). Design of an IoT Smart Home System.

Ametherm Inc. (n.d.). *4 Most Common Types of Temperature Sensor*. www.ametherm.com. Retrieved 12 10, 2022, from https://www.ametherm.com/blog/thermistors/temperature-sensor-types

AspenCore, Inc. (n.d.). *Temperature Sensors*. Electronics-Tutorial. Retrieved 12 12, 2022, from https://www.electronics-tutorials.ws/io/io_3.html

Codrey. (2022, 11 17). *Microcontroller basics*. Codrey.com. Retrieved 12 12, 2022, from https://www.codrey.com/learn/microcontroller-basics/

Electrical Classroom. (n.d.). *Relay – Operation, types and applications*. Electrical Classroom. Retrieved 12 13, 2022, from https://www.electricalclassroom.com/relay-operation-type-application/

Electronics For You. (2020, 10 15). *What Is Light Dependent Resistor (LDR)?* www.electonics4u.com. Retrieved 12 11, 2022, from https://www.electronicsforu.com/technology-trends/learn-electronics/ldr-light-dependent-resistors-basics

Electronics Projects Focus. (n.d.). *What is a Flyback Converter : Design & Its Working*. El-Pro-Cus. Retrieved December 13, 2022, from https://www.elprocus.com/what-is-a-flyback-converter-design-its-working/

Jost, D. (2019, July 2). *What Is A Temperature Sensor*. www.fierceelectronics.com. Retrieved 12 13, 2022, from https://www.fierceelectronics.com/sensors/what-a-temperature-