

SMART HOME MONITORING SYSTEM AND PREDICTION OF POWER CONSUMPTION

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology degree in
Information Technology

By

SOUNDHARYA S (38120081)

SANGEETHA K K (38120073)



**DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

March – 2022

DECLARATION

I **Soundharya S (38120081)** and **Sangeetha K K(38120073)** hereby declare that the Project

Report entitled "**Smart Home Monitoring System and prediction of Power Consumption**" done by

me under the guidance of **Dr. Mary Gladence (Internal)** and is submitted in partial

fulfillment of the requirements for the award of Bachelor of Technology degree in Information

Technology.

DATE: 15/04/2022

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC I 12B Status by UGC I
Approved by AICTE

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

BONA FIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Soundharya S (38120081)** and **Sangeetha (38120073)** who carried out the project entitled "**Smart Home Monitoring System and prediction of Power Consumption**" under my supervision from December 2021 to May 2022.

Internal Guide

Dr. MARY GLADENCE M.E., PhD.,

Dr. R. Subhashini, M.E., Ph.D.,

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

I am pleased to acknowledge our sincere thanks to Board of management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey our thanks to **Dr. T. SASIKALA, M.E., Ph.D., Dean, School of Computing** and **Dr.R.SUBHASHINI, M.E., Ph.D.,Head of the Department, Department of Information Technology** for providing us the necessary support and details at the right time during the progressive reviews.

I would like to express our sincere and deep sense of gratitude to our Project **Dr. R. Subhashini, M.E., Ph.D.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express our thanks to all Teaching and Non-teaching staff members of the Department of **INFORMATION TECHNOLOGY** who were helpful in many ways for the completion of the project.

ABSTRACT

Automation is a popular topic in the twenty-first century, and it has become an indispensable part of everyday life. Human labor, time, work, and some human errors are all eliminated under automation. And also many natural resources are on the verge of Renewable energy sources that have not yet achieved the point of total self-sufficiency are now in danger. In this situation, energy efficiency and appropriate power sharing are essential. In this article, we employed the smart meter concept to continuously monitor energy usage and warn the consumer when it surpasses a specified threshold. A customized Mobile Application and hardware module are created to measure the energy used by each item at home, estimate the monthly electricity bill, and anticipate power usage for future dates. In this method, the user could now create a preferred bill, and when the electricity use reaches the specified threshold, it will be notified via application. This system furthermore employs a technology to establish a connected device's tracking rate, making it an energy-efficient tool in and of itself. A current sensor, Voltage sensor and a Micro-controller were used to operate the hardware components. The software application was created to keep track of electricity costs. The system uses current data to forecast meter readings for the coming months and predicts the electricity usage based on their power consumption.

TABLE OF CONTENTS

Chapter No	Page No
1	Introduction.....1
2	Literature Survey.....2
3	Aim and Objective.....3
4	Algorithms Used.....7
4.1	Proposed System.....7
4.1.1	Device Control.....7
4.1.2	Dataset Collection.....8
4.1.3	Preprocessing and Prediction.....9
4.2	Smart Home Monitoring.....10
4.3	System Architecture.....11
4.4	Hardware Components used.....12
4.4.1	Power Supply.....12
4.4.2	Arduino UNO.....12
4.4.3	NodeMCU.....15
4.4.4	Voltage Sensor.....18
4.4.5	Current Sensor.....20
4.4.6	Relay Switch.....21
4.5	Software Components used.....22
4.5.1	Arduino IDE.....22
4.5.2	Android Studio.....22
4.5.3	Python 3.7.....24
4.5.4	Anaconda Software.....25

4.5.5 Firebase.....	25
4.6 Algorithm Used.....	26
4.6.1 Random Forest Algorithm.....	26
 5Results and Discussion	
5.1 Android Application.....	27
5.2 Firebase Cloud Database.....	28
5.3 Calculated Output Values.....	29
 6 Summary and Conclusion.....	
References.....	31
 Source Code.....	 35

ABBREVIATIONS

IOT	Internet Of Things
NodeMCU	Node Micro Controller Unit
ML	Machine Learning
AC	Alternating Current
DC	Direct Current
Wi-Fi	Wireless Fidelity
IDE	Integrated Development Environment

LIST OF FIGURES

Fig.No	Title	Page No
4.1	Description of Device Control.....	7
4.2	Illustration of Dataset Collection.....	8
4.3	Representation of preprocessing and Prediction.....	9
4.4	System Architecture Diagram.....	11
4.5	Arduino UNO.....	12
4.6	Arduino Pin Diagram.....	13
4.7	NodeMCU.....	15
4.8	NodeMCU Pin Diagram.....	16
4.9	Voltage Sensor.....	18
4.10	Voltage Sensor Pin Diagram.....	19
4.11	Current Sensor.....	20
4.12	Relay Switch.....	21
4.13	Relay Switch Pin Diagram.....	22
4.14	Firebase	26
5.1	Android Application Controls.....	28
5.2	Firebase Cloud.....	29
5.3	Screenshot of Output Values.....	30
5.4	Embedded Circuit.....	30

LIST OF TABLES

Table No	Title	Page No
4.1	Arduino Specifications.....	14
4.2	NodeMCU Specifications.....	17
4.3	Voltage Sensors Specifications.....	19

CHAPTER 1

INTRODUCTION

Smart home automation helps us in leading a comfortable life and quality of lifestyle is gradually uplifted. The ever increasing demand for power has driven many natural resources on the verge of depletion. In such a case, optimal power sharing and efficient energy consumption is needed. It typically connects the home appliances to the Controller. It has a high potential of sharing data between members or trusted individuals for personal security and could lead to energy saving measures with a positive environmental impact in the future.

With the help of IoT, the smart home concept adds comfort and ease to our lives. A network of devices connected to the Internet via various communication protocols such as Wi-Fi, Bluetooth, ZigBee, and others is used to achieve home automation. Electronic interfaces, such as Alexa or Google Assistant, or applications, can be used to control the devices remotely. Many of these IoT devices have sensors that monitor motion, temperature, and other variables to provide the user relevant information for the device's surroundings.

The traditional meter that we use in our homes to monitor energy consumption is an offline equipment which must be manually monitored. However, Smart Energy Meters are now available on the market, with readings that can be viewed from anywhere via the internet. Not only can we monitor energy consumption, but we can also monitor a number of parameters including voltage, current, power factor, frequency, and soon using IoT on a portable device. The goal of the project is to use an android application to control and monitor home appliances, as well as to use machine learning technique to anticipate the power usage of each device.

CHAPTER 2

LITERATURE SURVEY

In today's world, several technologies such as IoT and cloud are being used to design home automation systems. There are numerous research works [1]-[19] already proposed and currently available.

Hadoop is used to correctly obtain meter data as well as to calculate energy use per day, weekly, and monthly [1]. Furthermore, Hadoop operates on commodity hardware, energy distribution businesses can save expenses. In a survey paper [2] examines smart energy meters and its applications in depth, focusing on key aspects of the measurement process, economic factors, and the technology is used to suit those desires. Smart meters have been installed in a number of countries around the world since the early 2000s. The problems and opportunities that have developed as a result of the introduction of AI and the expanding use of cloud settings are also discussed in this study.

Lalitha Sankar et al., [3] discussed the User privacy in smart meter measures which is a prevalent issue in the smart grid, and it has been linked to a variety of technologies such as storage or power usage forecast without evaluating the loss of utility that has resulted from any such method to date. A variety of previously proposed smart meter privacy techniques are incorporated into this strategy. Tushar Chaurasia et al., in their work [4] created an Upgraded Smart Home Automation System. They are concerned with ensuring the secrecy, originality, and integrity of the information sensed and transferred by smart home objects. This study proposes an IoT-based smart home automation solution that is both secure and efficient.

A cloud-based, integrated smart home automation system that could be controlled remotely [5]. Smart houses powered by IoT and cloud computing provide real-time power consumption analysis. This paper describes the design of a minimal-cost smart switch that combines a Smart Home System and a cloud infrastructure. This research also covers the issues that have arisen, as well as a possible remedy.

In IoT applications and implementation, artificial intelligence is becoming more prominent. The implementation of AI in the Internet of Things is explained in A.A. Osuwa's works at el., [6]. The goal of AI research was to collect more useful data from IoT devices. The goal of this research is to demonstrate how artificial intelligence systems are utilized in the Internet of Things.

In addition, Artificial Intelligence technologies such as queue logics and neural networks are investigated in relation to the IoT network. The self-optimizing and server-based network are critical components [7] gave a presentation on a LoRa-based AI-powered remote smart-home monitoring system. LoRa is a WAN (LPWAN) system that is essential for IoT and M2M applications. The purpose of the paper is to offer an AI-based long-range (LoRa) system. A data flow mechanism for cloud and IoT Server is also included in this article.

IoT-enabled wireless smart homes and home automation were the subject of an evaluation and performance study. This research article [8], [21] describes a strategy for using Smart Home to control automation. The use of IoT to remotely monitor and control domestic devices is referred to as home automation. They use mobile phones or laptops to manage and run basic household appliances using a specially customized website and internet connection in this technology. Anuja Shinde et al., [9] designed a automation system by implementing android, IR, bluetooth and GSM. This paper describes a home automation system that uses an android app, an IR remote, Bluetooth, and GSM to control AC appliances, making it easier to use than the standard switch. The goal of this

paper is to teach people about these technologies and make the system as simple as possible to understand. This research resulted in the development of a system that allows users to operate and automated devices using mobile applications from remote [22].

The survey paper [10] discussed briefly about an energy management and analysis system that attempts to improve solar energy supply and grid supply. Essential component of a smart grid is reducing energy costs by intelligently switching between the two sources of electricity. Switching between power supplies is achieved to get power from the most cost-effective source while also balancing the energy supply at any given time. The system examines the model's energy consumption routine and guarantees that energy is delivered efficiently.

The use of Smart Home in conjunction with Grid Connection is outlined in the paper [11] to provide, more cost-effective, and loss-free electricity, as well as to aid in energy conservation through waste minimization. IOT is also used to feed real-time data to the Smart Grid, which helps with energy conservation when demand rises or falls. IoT has established links between users and suppliers in this proposed technique.

The Microcontroller-based Smart Power Meter using GSM [12] offered a method for connecting an LDR sensor to a flashing Light Emitting Diode (LED) and transferring data to the microcontroller over the GSM shield. Real-Time-Clock is a timer that also works as an interrupt. The system contains the ability to send an SMS to the user for updates on energy use as well as final bill creation, and the ability to re-configure loads via SMS. The system is expected to be less expensive than currently available smart meters with similar features.

An IoT-based Energy Utilization Tracking and Warning System [13] is also suggested in the earlier works. This article describes a domestic energy usage surveillance system with notifications on IoT platform Cloud Services. This paper describes an IOT architecture for tracking power consumption statistics at residence and sending the information to Cloud.

This system employs an ACS712 sensor module connected to an Intel Edison microprocessor to assess power use in the home. Cloud performs activity processing upon that collected electricity usage data and sends an email and telegram warning and suggestion to the homeowner, informing them that their electrical power use has surpassed their power consumption limit, thereby minimizing overuse of electricity [23].

Eleni Sardianou [14] presented a work on Predicting energy conservation patterns. This research provides an empirical model to analyze the important variables of home energy conservation behavior in Greece, using cross-section data. One of the most critical needs for attaining sustainable development and maintaining consumer quality of life is an energy policy framework [24]. The findings suggest that socioeconomic factors such as consumer income and family size can help explain differences in energy-saving priorities.

CHAPTER 3

Aim and Scope of the present investigation

Aim:

To build a Smart Home Monitoring System and Prediction of Power Consumption

Scope:

To enhance the standard of living, the appliances need to be wholly automated without any user intervention in any form whatsoever. This enables the end user hassle-free interaction with the appliances as the appliances learn and react as per the user's requirements without him physically pressing a button. The ever increasing demand for power has driven many natural resources on the verge of depletion. In such a case, optimal power sharing and efficient energy consumption is needed. The scope of the project is to build a smart home automation system to control home appliances, monitor the power consumed by the appliances and predict the power consumption using machine learning algorithm.

CHAPTER 4

4.1 PROPOSED SYSTEM

Proposed system can control the home appliances from the user-end and it will monitor the power consumption of each home appliance connected in the circuit. Using python, the daily power consumption of the electrical devices are taken and uploaded to cloud. The data will be then fed to Random Forest, a well-known machine learning technique that belongs to the Supervised Learning category, to predict the power usage of each electrical appliance in future days.

4.1.1 DEVICE CONTROL

Here an Android application will be developed to control home appliances through the internet. Using Arduino microcontroller, different devices will be controlled with the help of a relay module. Current and voltage sensors will be built into the devices. During ON/OFF process current and voltage values update to the cloud database as mentioned in Fig 4.1.

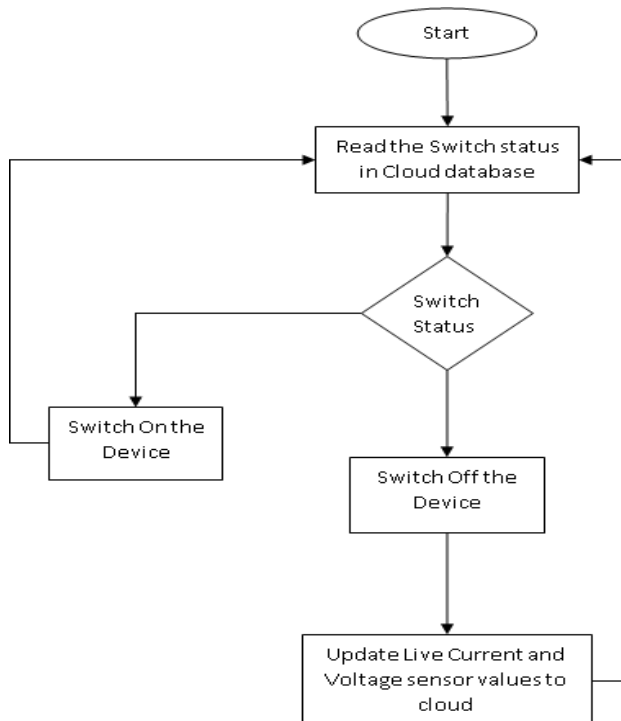


Fig 4.1 Description of Device Control

4.1.2 DATASET COLLECTION

A server side python application will be developed to collect the dataset from the user side. During the ON/OFF process this python app collects the data of current, voltage and total time of power consumption as shown in Fig 4.2. These data will be saved in CSV format in day wise order.

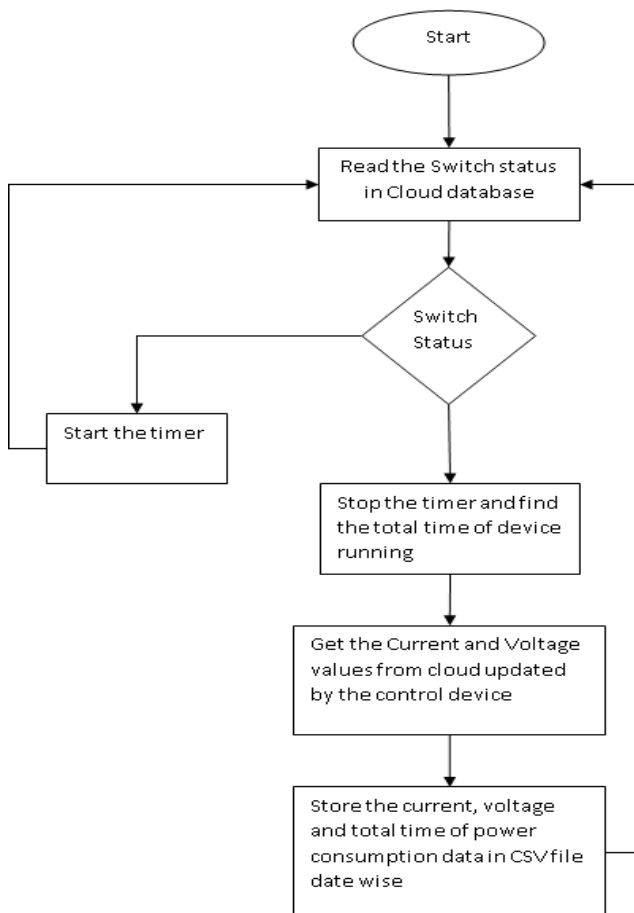


Fig 4.2 Illustration of Dataset Collection

4.1.3 PREPROCESSING AND PREDICTION

The entire CSV file will be collected and power consumption data will be arranged in kWh format. These preprocessed data are fed into the Random Forest algorithm which is depicted in Fig 4.3 and predict the power consumption of individual devices for next 30 days which will save as CSV format. Further these results will be uploaded to a cloud database to view from an android application.

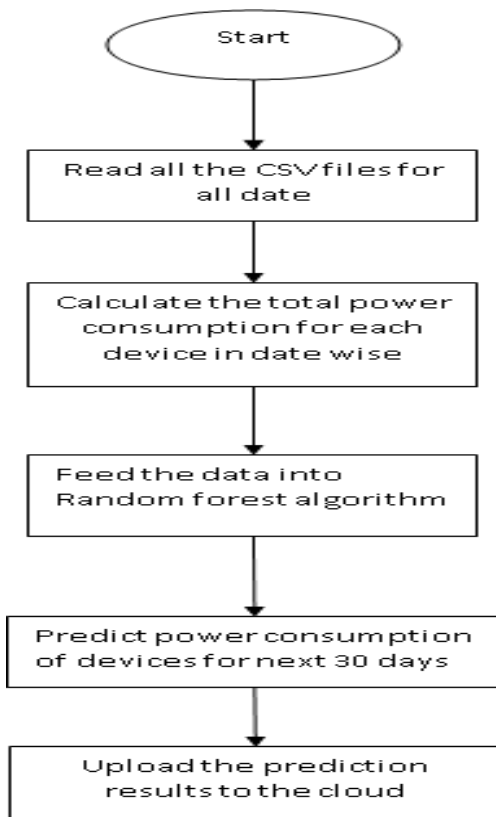


Fig 4.3 Representation of preprocessing and prediction

4.2 SMART HOME MONITORING SYSTEM

This project aims to create a prototype for power usage monitoring in smart homes in terms of putting smart household automation concepts into practice. The overall process is explained in the architecture diagram shown in Fig 4. We include a voltage sensor and current sensor for each socket in this device, allowing us to track the power use over time.

A customized Android mobile application is designed to track electricity consumption and control home appliances from anywhere via the internet built in Fig 5.

The NodeMCU is used as the master microcontroller to monitor and control the power consumption of devices. This controller can connect directly to a network and a cloud database, as well as receive control signals from an Android mobile device.

The power consumption data will be kept in the cloud database on a daily basis. We can anticipate future

power using machine learning algorithms. Predicting power consumption assists consumers in calculating electricity costs, individual block usage, and so on. The overall process of the smart home monitoring system is depicted in the architecture diagram shown in Fig 4.4.

4.3 SYSTEM ARCHITECTURE

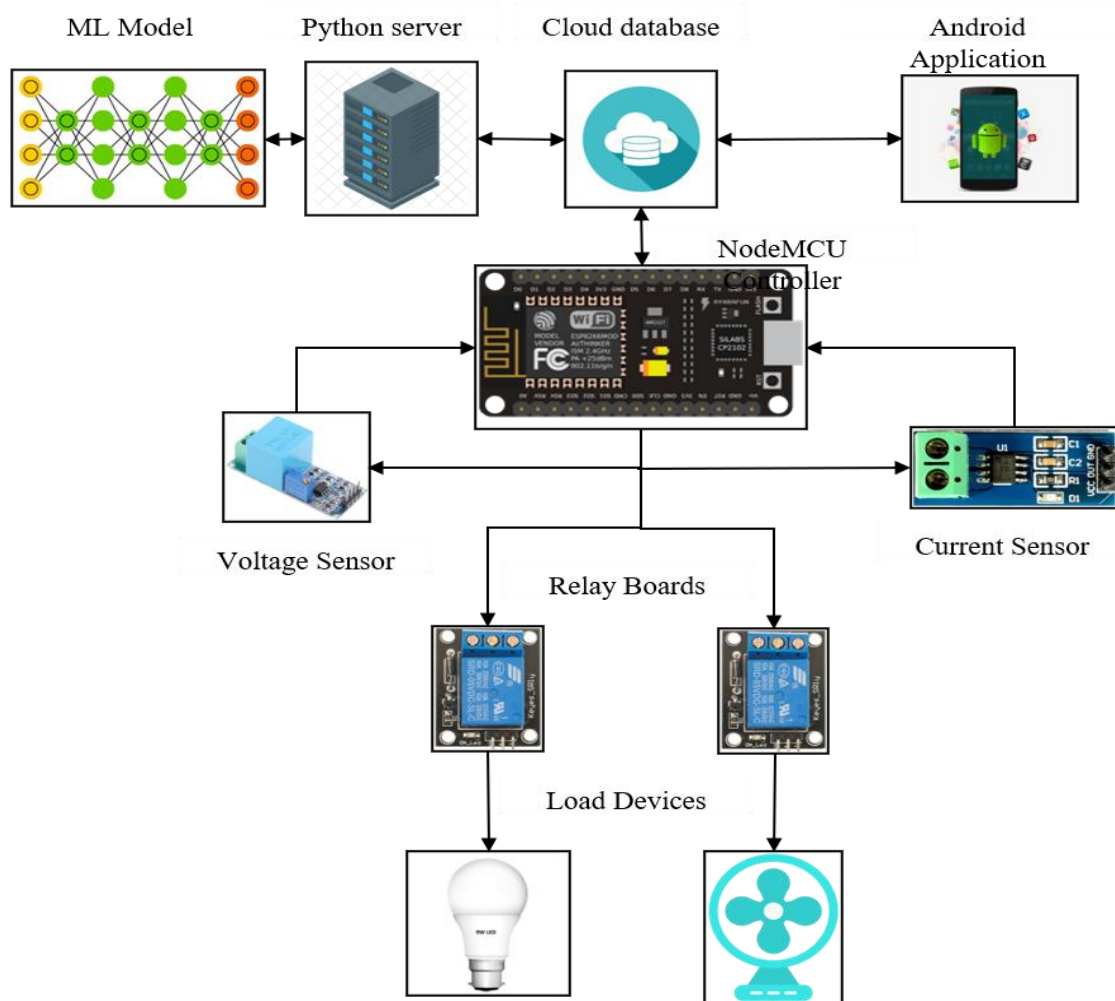


Fig 4.4 System Architecture Diagram

4.4 HARDWARE COMPONENTS USED

4.4.1 POWER SUPPLY

Alternating current is required for power line transmission and for high-power devices like appliances and lighting. AC's qualities make it perfect for long-distance transmission and supplying huge volumes of electricity for uncontrolled needs like generating heat and light. For lower-power appliances and gadgets, direct current electricity must be carefully handled. Because a typical home is powered by AC, it must be converted to DC for a variety of applications.

4.4.2 ARDUINO UNO

The Arduino Uno is a type of Arduino board shown in Fig 4.5 uses an ATmega328p microprocessor but also is available as an open-source project. Arduino boards take input signals like light from a sensor, a touch on a button etc - and convert them to outputs - such as turning on an LED, triggering a motor, or posting content online. Arduino has a number of compelling features, including a wide user base, a free and extensive library of code, and comparatively small components.



Fig 4.5 Arduino UNO

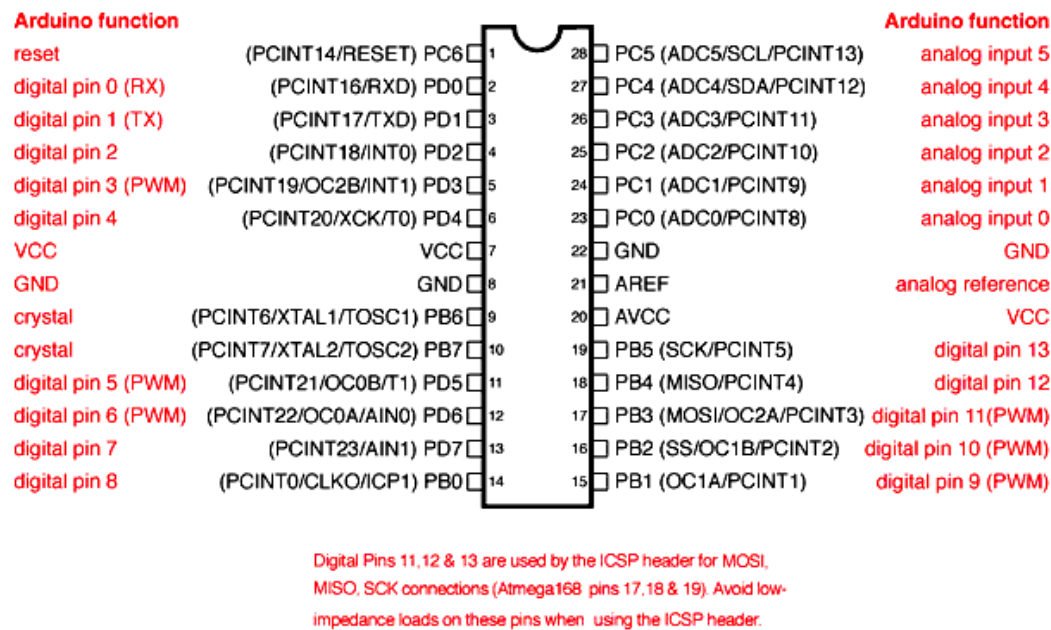


Fig 4.6 Arduino Pin diagram

Specifications:

- The ATmega2560 is a Microcontroller
- The operating voltage of this microcontroller is 5volts
- The recommended Input Voltage will range from 7volts to 12volts
- The input voltage will range from 6volts to 20volts
- The digital input/output pins are 54 where 15 of these pins will supply PWM o/p.
- Analog Input Pins are 16
- DC Current for each input/output pin is 40 mA
- DC Current used for 3.3V Pin is 50 mA
- Flash Memory like 256 KB where 8 KB of flash memory is used with the help of boot loader
- The static random access memory (SRAM) is 8 KB
- The electrically erasable programmable read-only memory (EEPROM) is 4 KB
- The clock (CLK) speed is 16 MHz
- The USB host chip used in this is MAX3421E

Table 4.1 Arduino Specifications

Microcontroller	ATmega1280
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	128 KB of which 4 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Benefits:

- Low cost
- Consistent board format
- 10x faster processing
- Added connectivity

Applications:

- Low cost PC/tablet/laptop
- IoT applications
- Media centre
- Robotics
- Industrial/Home automation
- Server/cloud server
- Print server
- Security monitoring

4.4.3 NODEMCU

The NodeMCU (shown in Fig 4.7) is widely used to connect microcontrollers and other integrated circuits. It is a minimal cost chip that is shown in Fig 6 that acts as a master microcontroller using TCP or IP protocol and . Furthermore, by providing some of the most important microcontroller functions such as GPIO, PWM, ADC, and so on, it may be capable of meeting many of the project's requirements on its own.



Fig 4.7 Node MCU

NodeMCU is an open-source firmware and development kit that helps you to prototype or build IoT product. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module.

The ESP8266 WiFi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor.

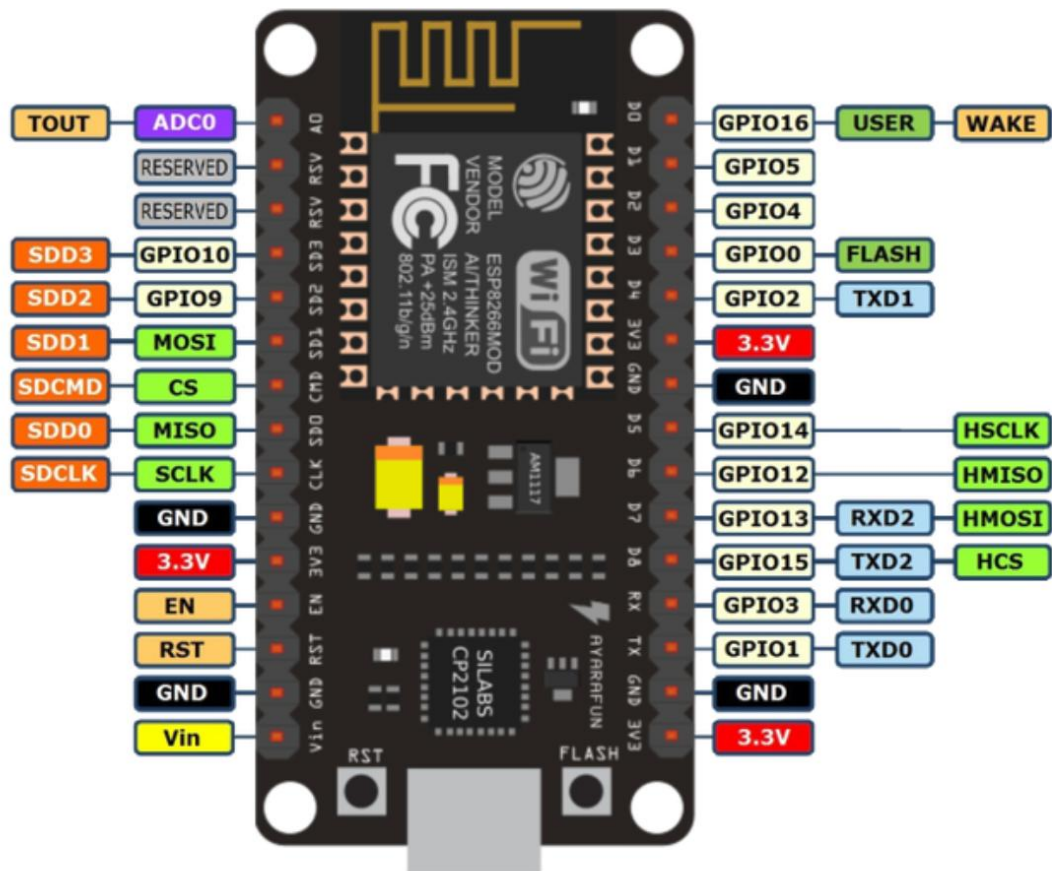


Fig 4.8 NodeMCU Pin Diagram

Table 4.2 NodeMCU Specifications

Label	GPIO	Input	Output	Notes
D0	GPIO16	no interrupt	no PWM or I2C support	HIGH at boot used to wake up from deep sleep
D1	GPIO5	OK	OK	often used as SCL (I2C)
D2	GPIO4	OK	OK	often used as SDA (I2C)
D3	GPIO0	pulled up	OK	connected to FLASH button, boot fails if pulled LOW

D4	GPIO2	pulled up	OK	HIGH at boot connected to on-board LED, boot fails if pulled LOW
D5	GPIO14	OK	OK	SPI (SCLK)
D6	GPIO12	OK	OK	SPI (MISO)
D7	GPIO13	OK	OK	SPI (MOSI)
D8	GPIO15	pulled to GND	OK	SPI (CS) Boot fails if pulled HIGH
RX	GPIO3	OK	RX pin	HIGH at boot
TX	GPIO1	TX pin	OK	HIGH at boot debug output at boot, boot fails if pulled LOW
A0	ADC0	Analog Input	X	

4.4.4 VOLTAGE SENSOR

A voltage sensor calculates and measures the amount of voltage present in an element. In Fig 4.9 , we have used the ZMPT101B voltage transformer to measure the voltage in the circuit. It can calculate the AC voltage or DC voltage .The voltage is the sensor's input, while analogue voltage signal, current signal, switches or audible signal are the sensor's output



Fig 4.9 Voltage Sensor

ZMPT101B Voltage Module Pinout

ZMPT101B module has 4 pins:

- **VCC:** Module power supply: 5 V
- **GND:** Ground
- **OUT:** Module output which is analog.

POWER ■
GND ■
OUTPUT ■

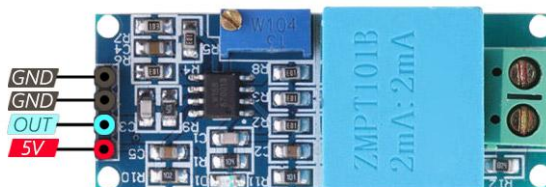


Fig 4.10 Voltage Sensor Pin Diagram

Table 4.3 Voltage Sensors Specifications

Specifications	Descriptions
Rated input current	2mA
Rated output current	2mA
Linear range	0~1000V 0~10mA
Isolation withstand voltage	4000V
Turns Ratio	1000:1000
Measurement accuracy Class	0.2
Linearity	0.1%
Rated Burdon	$\leq 200\Omega$
Operating Frequency	50-50Hz
DC coil Resistance	110 @ 20 deg Cel

4.4.5 CURRENT SENSOR

Current sensors, also known as current transformers or CTs, are devices (shown in Fig 4.11) that use the magnetic field to detect and provide a proportional output to measure the current flowing through a wire. The generated signal will then be used to represent the detected current in an ammeter, or it can be stored in a data gathering system for further analysis or used for control.

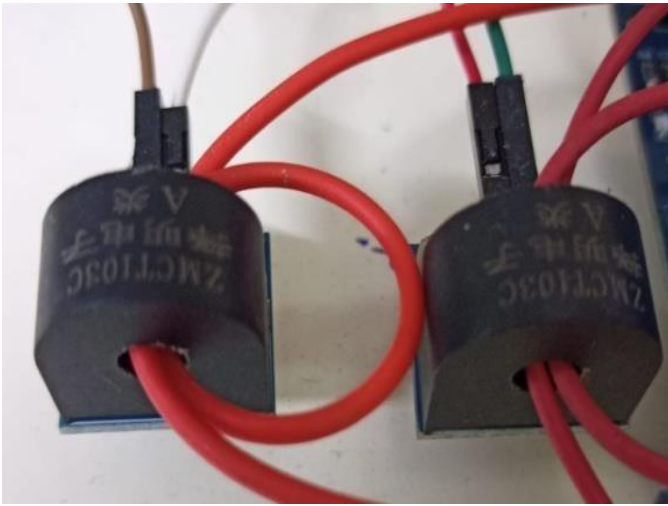


Fig 4.11 Current Sensor

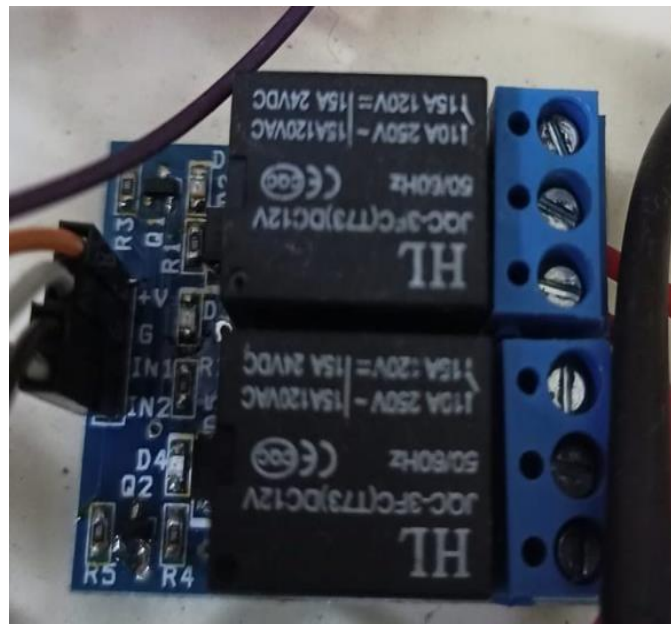
The sensed current and the output signal can be:

- Alternating current input,
 - Analog output, which duplicates the wave shape of the sensed current.
 - Bipolar output, which duplicates the wave shape of the sensed current.
 - Unipolar output, which is proportional to the average or RMS value of the sensed current.
- Direct current input,
 - unipolar, with a unipolar output, which duplicates the wave shape of the sensed current
 - digital output, which switches when the sensed current exceeds a certain threshold

4.4.6 RELAY SWITCH

Relays are electromechanical or electronic switches that open and close connections. Relays open and close contacts in another circuit to control the electrical circuit which is shown in Fig 4.12. They are commonly used in switchboards, industries, and smart buildings to manage power and switch lower current levels in a control system.

Fig 4.12 Relay Switch



A **relay switch** can be divided into two parts: input and output. The input section has a coil which generates magnetic field when a small voltage from an electronic circuit is applied to it. This voltage is called the operating voltage. Commonly used relays are available in different configuration of operating voltages like 6V, 9V, 12V, 24V etc. The output section consists of contactors which connect or disconnect mechanically. In a basic relay there are three contactors: normally open (NO), normally closed (NC) and common (COM). At no input state, the COM is connected to NC. When the operating voltage is applied the relay coil gets energized and the COM changes contact to NO. Different relay configurations are available like SPST, SPDT, [DPDT](#) etc, which have different number of changeover contacts. By using proper

combination of contactors, the electrical circuit can be switched on and off. Get inner details about structure of a [relay switch](#).

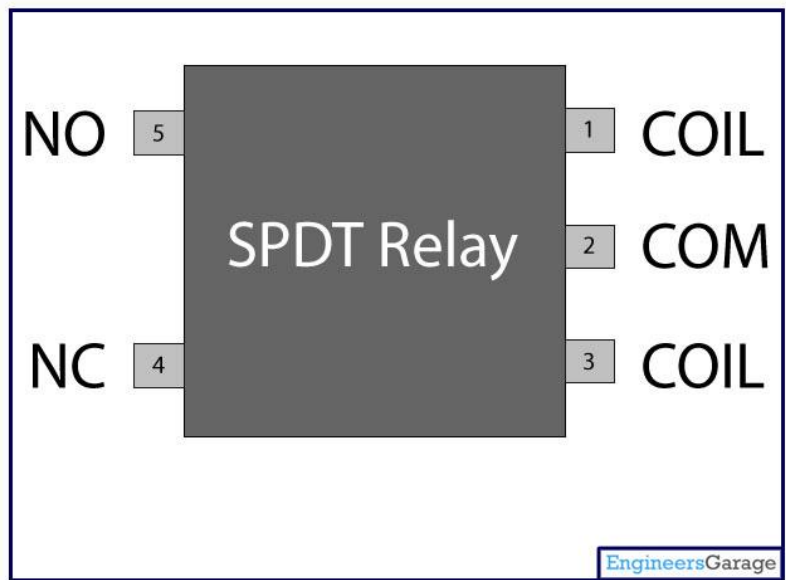


Fig 4.13 Relay Switch Pin Diagram

4.5 SOFTWARE COMPONENTS USED

4.5.1 ARDUINO IDE

Arduino IDE is an open and free platform with simple software. The coding part for the embedded circuit is written in the Arduino IDE. Writing and uploading the code to the board is very simple with this Arduino IDE. It supports cross platform usage. The IDE environment is implemented and developed in Java.

Arduino is based on easy-to-use hardware and software. [Arduino boards](#) are able to

read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

4.5.2 ANDROID STUDIO

Android Studio is the official integrated development environment (IDE) for Android application development. It is based on the IntelliJ IDEA, a Java integrated development environment for software, and incorporates its code editing and developer tools.

To support application development within the Android operating system, Android Studio uses a Gradle-based build system, emulator, code templates, and Github integration. Every project in Android Studio has one or more modalities with source code and resource files. These modalities include Android app modules, Library modules, and Google App Engine modules.

Features of Android Studio:

- It has a flexible Gradle-based build system.
- It has a fast and feature-rich emulator for app testing.
- Android Studio has a consolidated environment where we can develop for all Android devices.

- Apply changes to the resource code of our running app without restarting the app.

4.5.3 Python 3.7

Python is a high-level, general-purpose and a very popular programming language. Python programming language (latest Python 3) is being used in web development, Machine Learning applications, along with all cutting edge technology in Software Industry.

Python makes it applicable in almost every domain of software development. Python makes its presence in every emerging field. It is the fastest-growing programming language and can develop any application.

Features of Python:

- Easy to Learn and Use
- Expressive Language
- Interpreted Language
- Cross-platform Language
- Free and Open Source
- Object-Oriented Language
- Extensible
- Large Standard Library
- Embeddable

4.5.4 ANACONDA SOFTWARE

Anaconda is the Open Source data science distribution for the Python

and R programming languages scientific computing, such as data science, machine learning applications, predictive analysis, large-scale data processing, etc.

This platform is developed and maintained by Anaconda Inc. that was founded by Travis Oliphant and Peter Wang in the Year 2012. It can also be called Anaconda Distribution or Anaconda Individual Edition because it is an Anaconda Inc. product.

Benefits of using Anaconda Software:

- It is open-source, i.e., available for use free.
- It has more than 1500 R/ Python packages.
- It simplifies the package management and deployment.
- It is an environment where we can easily manage deployment for any project.

4.5.5 FIREBASE:

Firebase is a mobile and web app development platform that provides developers with a plethora of tools and services to help them develop high-quality apps, grow their user base, and earn more profit.



Fig 4.14 Firebase

Realtime Database

The Firebase Realtime Database is a cloud-hosted NoSQL database that lets you store and sync between your users in realtime.

The Realtime Database is really just one big JSON object that the developers can manage in realtime.

4.6 ALGORITHM USED

4.6.1 RANDOM FOREST ALGORITHM

A random forest is a supervised machine learning technique developed using decision tree algorithms. This method is used in a wide range of industries, like banking and e-commerce, to forecast behavior and outcomes. The Random Forest Algorithm provides more accurate and reliable predictions. Random forest has the benefit of being able to address classification and regression difficulties, which are common in machine learning systems.

Advantages of Random Forest Algorithm:

- Random Forest is capable of performing both Classification and Regression tasks.
- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the overfitting issue.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 ANDROID APPLICATION

To automate and control the household appliances, a customized android application shown in Fig 5.1, is created using java. The user will be able to keep track of how much energy each device consumes with this software. With the help of an Android application, the end-user can be able to access and view the status of an electronic appliance from anywhere via the Internet. The user also can predict the power consumption usage for next 30 days by clicking the prediction button Fig. 5.1

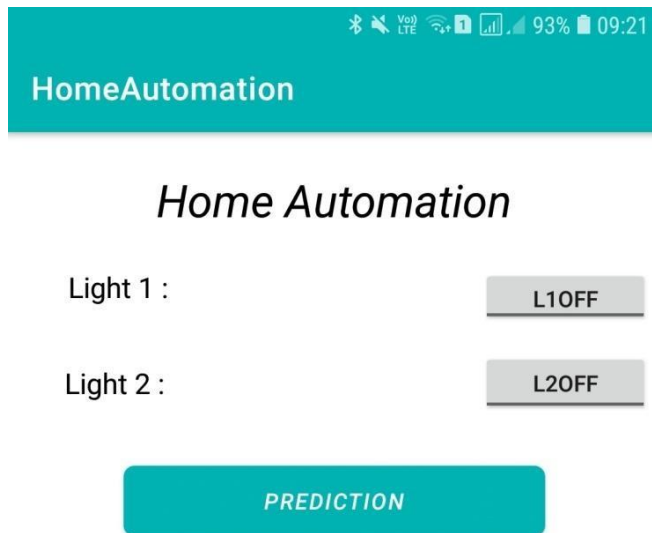


Fig 5.1 Android Application Controls

5.2 FIREBASE CLOUD DATABASE

Firebase is a document-model-based, cloud-hosted NoSQL database. It can be horizontally scaled while storing and synchronizing data for users in real time. This is extremely effective for programmes that are utilized on various devices, such as smartphone apps. Here we have created a real time database named Home Automation with two fields Light1 and Light2 for each device which is shown in Fig 5.2. Firebase is an excellent way of keeping sensory data acquired at the device level organized, and it works well with the Android APIs, which is supported by Android Things.

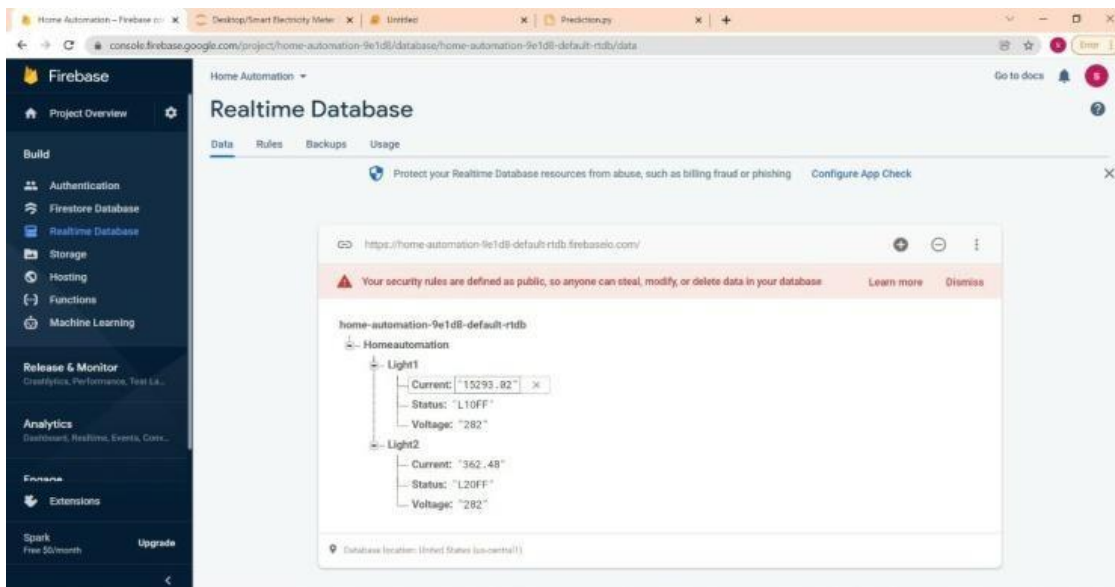


Fig 5.2firebase cloud

5.3 CALCULATED OUTPUT VALUES

The Electrical devices are turned ON/OFF through the customized Android Application.

During each ON/OFF process the current and voltage values will be generated with the help of the current sensor and voltage sensor respectively. The generated values are collected in the form of CSV file as shown in the fig.5.3 when the python server starts executing.

	A	B	C	D	E	F	G	H	I	J
1	Time	L1Current	L2Current	L1Voltage	L2Voltage	L1Status	L2Status	L1hrs	L2hrs	
2	51:19.8	0	0	295	0	L1OFF	0	0.006543	0	
3	51:59.8	1432.64	0	280	0	L1OFF	0	0.006657	0	
4	52:40.0	1415.38	0	280	0	L1OFF	0	0.008959	0	
5	53:28.6	1449.9	0	292	0	L1OFF	0	0.006638	0	
6										

Fig 5.3 Screenshot of Output Values

5.4 FINAL OUTPUT OF EMBEDDED CIRCUIT

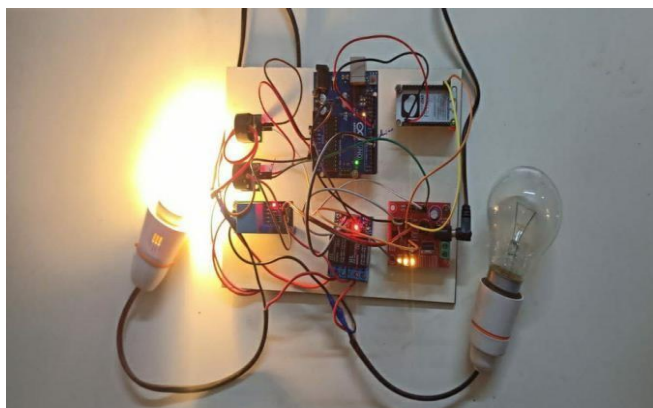


Fig 5.4 Embedded Circuit

CHAPTER 6

SUMMARY AND CONCLUSION

The project proposes the concept of smart homes, which can accommodate a wider range of home automation devices. Wireless connectivity, sensors, monitoring, and tracking are all part of a smart home. Smart homes are a large system that comprises a variety of technologies and apps that can be utilized to provide home protection and control.

This type of automation system is also beneficial to the elderly, as it facilitates their work and ensures that they live in a safe, secure, and comfortable environment.

REFERENCES

1. Bhardwaj, Ashutosh and Dr. William Jeet Singh. "Smart Meter Data Analytics using Hadoop." (2017) in 5th International Conference on Emerging Trends in Engineering Technology, Science and Management.; ISBN: 978-93-86171-60-3.

2. D. Alahakoon and X. Yu, "Smart Electricity

- MeterDataIntelligenceforFutureEnergySystems:ASurvey,"inIEEETransactionso
nIndustrialInformatics, vol. 12, no. 1, pp. 425-436, Feb.
2016,doi:10.1109/TII.2015.2414355.
3. L. Sankar, S. R. Rajagopalan, S. Mohajer and H. V.Poor,"SmartMeterPrivacy:ATheoreticalFramework," in IEEE Transactions on Smart Grid,vol.4,no.2,pp.837-846,June2013,doi:10.1109/TSG.2012.2211046.
 4. T. Chaurasia and P. K. Jain, "Enhanced Smart HomeAutomation System based on Internet of Things," 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2019, pp. 709-713, doi: 10.1109/I- SMAC47947.2019.9032685.
 5. A.Hanumanthaiah, D. Arjun,M. L. Liya, C. ArunandA.Gopinath,"IntegratedCloudbasedSmartHome with Automation and Remote controllability,"2019InternationalConferenceonCommunicationand Electronics Systems (ICCES), 2019, pp. 1908-1912,doi:10.1109/ICCES45898.2019.9002245.
 6. A.Osuwa,E.B.EkhoragbonandL.T.Fat,"Applicationof artificial intelligence inInternet ofThings,"20179thInternationalConferenceonComputationalIntelligenceandCo mmunicationNetworks(CICN),2017,pp.169-173,doi:10.1109/CICN.2017.8319379.
 7. M.Shahjalal,M.K.Hasan,M.M.Islam,M.M.Alam, M. F. Ahmed and Y. M. Jang,

- "An Overview of AI-Enabled Remote Smart-Home Monitoring System Using LoRa," 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC), 2020, pp. 510-513, doi:10.1109/ICAIIIC48513.2020.9065199.
8. K. Agarwal, A. Agarwal and G. Misra, "Review and Performance Analysis on Wireless Smart Home and Home Automation using IoT," 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2019, pp. 629-633, doi: 10.1109/I-SMAC47947.2019.9032629.
 9. A. Shinde, S. Kanade, N. Jugale, A. Gurav, R. A. Vatti and M. M. Patwardhan, "Smart Home automation system using IR, bluetooth, GSM and android," 2017 Fourth International Conference on Image Information Processing (ICIIP), 2017, pp. 1-6, doi:10.1109/ICIIP.2017.8313770.
 10. S. Oswal, V. Modani, S. Gundawar and V. Pawar, "Energy Management And Analysis For Smart Homes Using IoT," 2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA), 2019, pp. 1-7, doi:10.1109/ICCUBEA47591.2019.9128496.
 11. M. Kumar, A. F. Minai, A. A. Khan and S. Kumar, "IoT based Energy Management System for Smart Grid," 2020 International Conference on Advances in Computing, Communication & Materials (ICACCM), 2020, pp. 121-125, doi:10.1109/ICACCM50413.2020.9213061.
 12. H. K. Patel, T. Mody and A. Goyal, "Arduino Based Smart Energy Meter using GSM," 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), 2019, pp. 1-6, doi:10.1109/IoT-SIU.2019.8777490
 13. Z. H. Che Soh, I. H. Hamzah, S. A. Che Abdullah, M. A. Shafie, S. N. Sulaiman

- and K. Daud, "Energy Consumption Monitoring and Alert System via IoT," 2019 7th International Conference on Future Internet of Things and Cloud (FiCloud), 2019, pp. 265-269, doi: 10.1109/FiCloud.2019.00044.
14. E.Sardianou,"HouseholdEnergyConservationPatterns:EvidenceFromGreece",J .Chem.Inf.Model,vol.53,pp.1689-1699, 2013.
 15. Mussab Alaa, A.A. Zaidan, B.B. Zaidan, MohammedTalalandM.L.M.Kiah,"AreviewofsmarthomeapplicationsbasedonInternetofThings",J.Netw. Comput. Appl, vol. 97, no. C, pp. 48-65, November2017.
 16. A.Frank,R.AsuncionandM.Frank,"SmartOptimizationofEnergyConsumptionusingIoT,"20192ndIEEEMiddleEastandNorthAfricaCOMMunications Conference (MENACOMM), 2019,pp. 1-6,doi: 10.1109/MENACOMM46666.2019.8988549
 17. P. Bandarra, M.T. Valdez and A. Pereira, "Solutionsformonitoringandanalyzingforenergyconsumption Energy management systems", 51st International Universities Power Engineering Conference (UPEC), 2016
 18. S. Thakare, A. Shriyan, V. Thale, P. Yasarp and K.Unni, "Implementation of an energy monitoring andcontrol device based on IoT," 2016 IEEE Annual IndiaConference(INDICON),2016,pp.1-6,doi:10.1109/INDICON.2016.7839066.
 19. DepartmentofEEE,SirMVIT,Bangalore,IndiaMuhammadAsadullahandRazaAhshan,"Anoverview of home automation systems", In 2016 2ndInternationalConferenceonRoboticsand ArtificialIntelligence(ICRAI), pp. 27-31, 2016.

20. S.Ivanović, S.Milivojša, T.Erić and M.Vidaković, "Collection and Analysis of System Usage Data in Smart Home Automation Systems," 2017 IEEE 7th International Conference on Consumer Electronics - Berlin (ICCE-Berlin), 2017, pp. 65-66, doi:10.1109/ICCE-Berlin.2017.8210592.
21. Gladence, L. Mary, V. Maria Anu, R. Rathna, and E.Brumancia. "Recommender system for home automation using IoT and artificial intelligence." *Journal of Ambient Intelligence and Humanized Computing* (2020): 1-9.
22. Gladence, L.M., Anu, V.M., Revathy, S. et al. Security management in smart home environment. *Soft Comput* (2021). <https://doi.org/10.1007/s00500-021-06054-z>
23. Sungheetha, Akey, and Rajesh Sharma. "Real Time Monitoring and Fire Detection using Internet of Things and Cloud based Drones." *Journal of Soft Computing Paradigm (JSCP)* 2, no. 03 (2020): 168-174.
24. Agarwal, Arun, Kabita Agarwal, and Gourav Misra. "Is internet becoming a Major Contributor for Global warming-The Online Carbon Footprint." *Journal of Information Technology and Digital Worlds* 2, no .4 (2020): 217-22

SOURCE CODE

Home_Node

```
#define WIFI_SSID "iotkitt"
```

```

#define WIFI_PASSWORD "123456789"
#include "FirebaseESP8266.h"
#include <ESP8266WiFi.h>
#include <SoftwareSerial.h>
SoftwareSerial mySerial(5, 4); //(RX, TX)
#define FIREBASE_HOST "https://home-automation-9e1d8-default-
rtdb.firebaseio.com/"
#define FIREBASE_AUTH "smbserjnlCssGbZ4kN3yTKGdRMv8wlnIOEfTVyMA"
FirebaseData firebaseData;

String S, S1, S4, S5, S6, S7, S8;
char *S2;
char *S3[5];
String Light1S, Light1n, Light2S, Light2n;
int i=0;
int j=0;
//int lightpin = D3;
//const char* host = "10.0.0.101";
void setup()
{
    Light1n="L1OFF";
    Light2n="L2OFF";
    Serial.begin(9600);
    //pinMode(lightpin, OUTPUT);
    mySerial.begin(9600);
    WiFi.mode(WIFI_STA);
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
    while (WiFi.status() != WL_CONNECTED)
    {
        Serial.print(".");
        delay(300);
    }
}

```

```

}
Serial.println();
Serial.print("Connected with IP: ");
Serial.println(WiFi.localIP());
Serial.println();
Firebase.begin(FIREBASE_HOST,FIREBASE_AUTH);
Firebase.reconnectWiFi(true);
{
  Serial.println("reconnected");
}

}

void loop() {
  Firebase.getString(firebaseData,"/Homeautomation/Light1/Status");
  Light1S = (firebaseData.stringData());// put your main code here, to run repeatedly:
  if(Light1S!=Light1n){
    Serial.println(Light1S);
    Serial.println(Light1n);
    Serial.println("");
    mySerial.print(Light1S);
    Light1n=Light1S;
    getValue();

  }

  Firebase.getString(firebaseData,"/Homeautomation/Light2/Status");
  Light2S = (firebaseData.stringData());// put your main code here, to run repeatedly:
  if(Light2S!=Light2n){
    mySerial.print(Light2S);
    Light2n=Light2S;
    getValue();
  }
}

```

```
}  
}
```

```
void getValue(){  
  if(mySerial.available())  
  {  
    //WiFiClient client;  
    S=mySerial.readString();  
    int e=S.indexOf("@");  
    int e1=S.indexOf("#");  
    int e2=S.indexOf("$");  
    int e3=S.indexOf("%");  
    //int e4=S.indexOf("^");  
    //int e5=S.indexOf("&");  
    S4=S.substring(e+1,e1);  
    S5=S.substring(e1+1,e2);  
    S6=S.substring(e2+1,e3);  
    //S7=S.substring(e3+1,e4);  
    //S8=S.substring(e4+1,e5);  
    Serial.println(S4);  
    Serial.println(S5);  
    Serial.println(S6);  
    Firebase.setString(firebaseData,"/Homeautomation/Light1/Current",S4);  
    Firebase.setString(firebaseData,"/Homeautomation/Light2/Current",S5);  
    Firebase.setString(firebaseData,"/Homeautomation/Light1/Voltage",S6);  
    Firebase.setString(firebaseData,"/Homeautomation/Light2/Voltage",S6);  
  
  }  
}
```

Home_UNO

```
#include <SoftwareSerial.h>
SoftwareSerial swSerial(10, 11); // RX, TX

//#define esp8266_reset_pin 5
#define Current_sensor1 A0
#define Current_sensor2 A1
#define Voltage_sensor1 A2
#define Relay1 7
#define Relay2 6
float Device1_current, Device2_current, Device1_voltage;
float nVPP1; // Voltage measured across resistor
float nVPP2;
float nCurrThruResistorPP; // Peak Current Measured Through Resistor
float nCurrThruResistorRMS; // RMS current through Resistor
float nCurrentThruWire1; // Actual RMS current in Wire
float nCurrentThruWire2;

double sensorValue1 = 0;
int val[100];
int max_v = 0;
double VmaxD = 0;
double VeffD = 0;
double Veff = 0;

void setup() {
  Serial.begin(9600);
```

```

swSerial.begin(9600);
pinMode(Current_sensor1,INPUT);
pinMode(Current_sensor2,INPUT);
pinMode(Voltage_sensor1,INPUT);
pinMode(Relay1,OUTPUT);
pinMode(Relay2,OUTPUT);
digitalWrite(Relay1,LOW);
digitalWrite(Relay2,LOW);
}

void loop() {
  if(swSerial.available()>0){
    String Val=swSerial.readString();
    Serial.println(Val);
    if(Val=="L1ON"){
      digitalWrite(Relay1,HIGH);
    }
    if(Val=="L1OFF"){
      digitalWrite(Relay1,LOW);
    }
    if(Val=="L2ON"){
      digitalWrite(Relay2,HIGH);
    }
    if(Val=="L2OFF"){
      digitalWrite(Relay2,LOW);
    }
  }

  nVPP1 = getVPP1();
  nCurrThruResistorPP = (nVPP1/200.0) * 1000.0;
  nCurrThruResistorRMS = nCurrThruResistorPP * 0.707;

```

```

nCurrentThruWire1 = nCurrThruResistorRMS * 1000;
Serial.print("Current Through Wire1 : ");
Serial.println(nCurrentThruWire1,3);
nVPP2 = getVPP2();
nCurrThruResistorPP = (nVPP2/200.0) * 1000.0;
nCurrThruResistorRMS = nCurrThruResistorPP * 0.707;
nCurrentThruWire2 = nCurrThruResistorRMS * 1000;
Serial.print("Current Through Wire2 : ");
Serial.println(nCurrentThruWire2,3);


int v = voltageM();
Serial.print("Voltage: ");
Serial.println(v);


//Device1_current=analogRead(Current_sensor1);
//Device2_current=analogRead(Current_sensor2);
/*Device1_voltage=analogRead(Voltage_sensor1);
Serial.print("Current1:");
Serial.println(Device1_current);
Serial.print("Current2:");
Serial.println(Device2_current);
Serial.print("voltage:");
Serial.println(Device1_voltage);
Serial.println("");*/


swSerial.println("@"+String(nCurrentThruWire1)+"#"+String(nCurrentThruWire2)+"$"+
String(v)+"%");
delay(1000);
}

```

```

float getVPP1()
{
    float result;
    int readValue;          //value read from the sensor
    int maxValue = 0;        // store max value here
    uint32_t start_time = millis();
    while((millis()-start_time) < 1000) //sample for 1 Sec
    {
        readValue = analogRead(Current_sensor1);
        // see if you have a new maxValue
        if (readValue > maxValue)
        { /*record the maximum sensor value*/
            maxValue = readValue;
        }
    }
    // Convert the digital data to a voltage
    result = (maxValue * 5.0)/1024.0;
    return result;
}

float getVPP2()
{
    float result;
    int readValue;          //value read from the sensor
    int maxValue = 0;        // store max value here
    uint32_t start_time = millis();
    while((millis()-start_time) < 1000) //sample for 1 Sec
    {
        readValue = analogRead(Current_sensor2);
        // see if you have a new maxValue

```

```

    if (readValue > maxValue)
    { /*record the maximum sensor value*/
        maxValue = readValue;
    }
}
// Convert the digital data to a voltage
result = (maxValue * 5.0)/1024.0;
return result;
}

```

```

float voltageM(){

for ( int i = 0; i < 100; i++ ) {
    sensorValue1 = analogRead(A2);
    //Serial.println(sensorValue1);
    if (analogRead(A2) > 511) {
        val[i] = sensorValue1;
    }
    else {
        val[i] = 0;
    }
    delay(1);
}

```

```

max_v = 0;

```

```

for ( int i = 0; i < 100; i++ )
{
    if ( val[i] > max_v )
    {

```

```

        max_v = val[i];
    }
    val[i] = 0;
}
if (max_v != 0) {

    VmaxD = max_v;
    VeffD = VmaxD / sqrt(2);
    Veff = (((VeffD - 420.76) / -90.24) * -210.2) + 210.2;
}
else {
    Veff = 0;
}
//Serial.print("Voltage: ");
//Serial.println(Veff);
VmaxD = 0;
return Veff;
    delay(100);
}

```

Firebasepost.py

```

from firebase import firebase
import os
import time
import datetime
L1Statusn="L1OFF"

```

```

L2Statusn="L2OFF"
timeout = time.time() + 60*5
a=str(datetime. datetime.now())
spltt=a.split(' ')
date=spltt[0]
time1=spltt[1]
s1=date.split('-')
cday=int(s1[2])
cmonth=int(s1[1])
cyear=int(s1[0])

l1ontime=0
l1offtime=0

firebase = firebase.FirebaseApplication('https://home-automation-9e1d8-default-
rtadb.firebaseio.com/')
#file = open("dataset.csv", "a")
#if os.stat("dataset.csv").st_size == 0:
file = open(date+"data_log.csv", "a")
if os.stat(date+"data_log.csv").st_size == 0:

file.write("Time,L1Current,L2Current,L1Voltage,L2Voltage,L1Status,L2Status,L1hrs,L
2hrs\n")

while True:

    L1Status= firebase.get('/Homeautomation/Light1/Status',None)
    L2Status= firebase.get('/Homeautomation/Light2/Status',None)
    print(L1Status)
    print(L2Status)

```

```

if(L1Status != L1Statusn):
    if(L1Status=="L1ON"):
        l1ontime=time.time()
        L1Current= firebase.get('/Homeautomation/Light1/Current',None)
        L1Voltage= firebase.get('/Homeautomation/Light1/Voltage',None)
    if(L1Status=="L1OFF"):
        l1offtime=time.time()
        l1time= l1offtime - l1ontime
        l1time=l1time/3600

file.write(str(now)+","+str(L1Current)+","+str(0)+","+str(L1Voltage)+","+str(0)+","+str(L
1Status)+","+str(0)+","+str(l1time)+","+str(0)+"\n")
    print(l1time)
    L1Statusn=L1Status
    now =datetime. datetime.now()
    time.sleep(1);

L2Current= firebase.get('/Homeautomation/Light2/Current',None)

L2Voltage= firebase.get('/Homeautomation/Light2/Voltage',None)

file.flush()
if(L2Status != L2Statusn):
    L2Statusn=L2Status
    now =datetime. datetime.now()
    time.sleep(1);
    L1Current= firebase.get('/Homeautomation/Light1/Current',None)
    L2Current= firebase.get('/Homeautomation/Light2/Current',None)
    L1Voltage= firebase.get('/Homeautomation/Light1/Voltage',None)
    L2Voltage= firebase.get('/Homeautomation/Light2/Voltage',None)
    #file.write(str(now)+","+str(i)+","+str(-i)+","+str(i-10)+","+str(i+5)+","+str(i*i)+"\n")

```

```
file.write(str(now)+","+str(L1Current)+","+str(L2Current)+","+str(L1Voltage)+","+str(L2
Voltage)+","+str(L1Status)+","+str(L2Status)+"\n")
    file.flush()
```

Prediction.py (google collab)

```
#!/usr/bin/env python
```

```
# coding: utf-8
```

```
# In[151]:
```

```
import pandas as pd
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
Energy1 = pd.read_excel(r'C:\Users\Swathee\Desktop\Smart Electricity Meter
Monitoring and Prediction using iSocket\Computer\Prediction\Building energy
consumption racord.xlsx')
```

```
Energy1
```

```
# In[152]:
```

```
Energy = Energy1.set_index('Time')
```

```
Energy
```

```
Energy.describe()
```

```
# In[153]:
```

```
plt.boxplot(Energy['building 41']) # Energy.loc[:, 'building 12'] or Energy.iloc[:,0]
plt.show()
import seaborn as sns
sns.boxplot(Energy['building 41'])
```

```
# In[154]:
```

```
Path= r'C:\Users\Swathee\Desktop\Smart Electricity Meter Monitoring and Prediction
using iSocket\Computer\Prediction\WeatherData.xlsx'
knmi= pd.read_excel(Path)
knmi
```

```
# In[155]:
```

```
knmi = knmi.set_index("Time")
knmi
```

```
# In[156]:
```

```
df = pd.concat([knmi, Energy], axis=1) #axis =1 for considering the columns
```

```
df
```

```
# In[157]:
```

```
df.isna().sum()
```

```
# In[158]:
```

```
import missingno as msno  
msno.bar(df)
```

```
# In[159]:
```

```
plt.figure(figsize = (16,6))  
sns.heatmap(df.corr(), annot = True, linewidths=1, fmt=".2g", cmap= 'coolwarm')  
plt.xticks(rotation='horizontal')
```

```
# In[160]:
```

```
df_sum_weekly = df['building 41'].resample('W').mean()  
df_feature1= df["Temp"].resample("W").mean()  
df_feature2 = df["U"].resample("W").mean()
```

```
# In[161]:
```

```
fig,ax = plt.subplots(figsize=(24,8))
ax.plot(df_sum_weekly.index, df_sum_weekly, color="red",marker="o")
ax.set_ylabel("KWh")
ax.set_xlabel('Date')
ax2 = ax.twinx()
ax3 = ax.twinx()
ax2.plot(df_sum_weekly.index, df_feature1, color="blue", marker="o")
ax2.set_ylabel("'°C")
ax3.plot(df_sum_weekly.index, df_feature2, color="green", marker="o")
ax3.set_ylabel("grams/cubic meter")
ax3.spines["right"].set_position(("axes", 1.05))
fig.legend(["Resampled Weekly Energy Consumption","Resampled Weekly
Temperature","Resampled Weekly Relative Humidity"], loc='upper right')
fig.show()
```

```
# In[162]:
```

```
from scipy.stats import spearmanr
```

```
#filter columns from dataframe
energy = np.array(df["building 41"])
hour = np.array(df["HH"])
month= np.array(df["month"])
```

```
# calculate spearman's correlation
```

```
corr1, _ = spearmanr(energy, hour)
corr2, _ = spearmanr(energy, month)
print('Spearman correlation between Energy and hour feature: %.3f' % corr1)
print('Spearman correlation between Energy and month feature: %.3f' % corr2)
```

```
# In[163]:
```

```
knmi_updated= knmi.loc[:, ~knmi.columns.isin(["TD","U","DR","FX"])] # ~ sign drops
the columns we select
knmi_updated
```

```
# In[164]:
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(knmi_updated, Energy, test_size =
0.2, random_state = 0)
y_train
```

```
# In[165]:
```

```
y_train = y_train.values.ravel() #ravel is a numpy function to change a 2-dimensional
array or a multi-dimensional array into a continuous flattened array.
y_train
```

```
# In[166]:
```

```
y_test = y_test.values.ravel()
y_test
```

```
# In[167]:
```

from sklearn.preprocessing import StandardScaler #standardizes the data to a range in which the mean is equal to 0 and the standard deviation is 1. It assumes the data is normally distributed.

from sklearn.preprocessing import MinMaxScaler #normalizes the data and brings the values between 0 (lowest value) and 1 (highest value)

from sklearn.preprocessing import RobustScaler #standardizes the data. But is more robust to outliers because it only scales the data according to the Interquartile Range (IQR) between the 1st and 3rd quartile.

```
#Generate the scaler
```

```
sc1= StandardScaler()
```

```
sc2= MinMaxScaler()
```

```
sc3= RobustScaler()
```

```
X1 = sc1.fit_transform(knmi_updated)
```

```
X2 = sc2.fit_transform(knmi_updated)
```

```
X3 = sc3.fit_transform(knmi_updated)
```

```
sns.distplot(X1,color="red",label="StdScaler")
```

```
sns.distplot(X2,color="blue",label="MinMax")
```

```
sns.distplot(X3,color="black",label="RobustScaler")
```

```
plt.legend()
```

In[168]:

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score #(coefficient of determination) regression score
function.
from sklearn.metrics import mean_squared_error #The MSE indicates the average
distance of the best fit regression line to the observed values.
RFReg = RandomForestRegressor(max_depth=10, random_state=0)
X_train2, X_test2, y_train2, y_test2 = train_test_split(X1, Energy, test_size=0.2,
random_state=0, shuffle= "False")
y_train2 = y_train2.values.ravel()
y_test2 = y_test2.values.ravel()
RFReg.fit(X_train2, y_train2)
Predicted_Train2= RFReg.predict(X_train2)
print(r2_score(y_train2, Predicted_Train2))
print(mean_squared_error(y_train2, Predicted_Train2))
```

In[169]:

```
#importing the ensemble module for the random forest regressor from sklearn library
from sklearn.ensemble import RandomForestRegressor

# Creating an instance of the random forest regressor
RFReg = RandomForestRegressor(max_depth=10, random_state=0)
X4 = sc1.fit_transform(knmi.loc[:, ~knmi.columns.isin(['U'])])
# fitting the regression model to the training data
X_train2, X_test2, y_train2, y_test2 = train_test_split(X4, Energy, test_size=0.2,
```

```

random_state=0, shuffle= "False")
y_train2 = y_train2.values.ravel()
y_test2 = y_test2.values.ravel()
RFReg.fit(X_train2, y_train2)

#Predicting on the training data
Predicted_Train2= RFReg.predict(X_train2)

#Caculating R2 score and Root mean square error
print(r2_score(y_train2, Predicted_Train2))
print(mean_squared_error(y_train2, Predicted_Train2))

```

In[170]:

```

plt.plot(y_train2, color="b",label= 'Original Data')
plt.plot(Predicted_Train2, color ="red", label="RFReg predicted data")
plt.xlabel('Time (Hours)')
plt.ylabel('kWh')
plt.legend(loc='best')
plt.show()

```

In[171]:

```

Predicted_Test2 = RFReg.predict(X_test2)

#Caculating R2 score and Root mean square error
print(r2_score(y_test2,Predicted_Test2))

```

```
print(mean_squared_error(y_test2,Predicted_Test2))
```

```
# In[172]:
```

```
plt.plot(y_test2, color='blue', label="Actual")
plt.plot(Predicted_Test2, color='Red', label="Predicted")
plt.xlabel('Time (Hours)')
plt.ylabel('kWh')
plt.legend(loc='best')
```

```
# In[173]:
```

```
from sklearn.model_selection import GridSearchCV
```

```
# In[174]:
```

```
check_parameters = {'max_depth':[8,9,11,12]}
```

```
gridsearchcv = GridSearchCV(RFReg, check_parameters, n_jobs=-1, cv=3)
gridsearchcv.fit(X_train, y_train)
```

```
print('Best parameters found:\n', gridsearchcv.best_params_)
```

```
# In[175]:
```

```
#importing the ensemble module for the random forest regressor from sklearn library
from sklearn.ensemble import RandomForestRegressor
```

```
# Creating an instance of the random forest regressor
RFReg = RandomForestRegressor(max_depth=12, random_state=0)
```

```
# fitting the regression model to the training data
X_train2, X_test2, y_train2, y_test2 = train_test_split(X4, Energy, test_size=0.2,
random_state=0, shuffle= "False")
y_train2 = y_train2.values.ravel()
y_test2 = y_test2.values.ravel()
RFReg.fit(X_train2, y_train2)
```

```
#Predicting on the training data
Predicted_Train2= RFReg.predict(X_train2)
```

```
#Calculating R2 score and Root mean square error
print(r2_score(y_train2, Predicted_Train2))
print(mean_squared_error(y_train2, Predicted_Train2))
```

```
# In[176]:
```

```
Predicted_Test2 = RFReg.predict(X_test2)
```

```
#Calculating R2 score and Root mean square error
print(r2_score(y_test2,Predicted_Test2))
print(mean_squared_error(y_test2,Predicted_Test2))
```

```
# In[177]:
```

```
check_parameters = {'max_depth':[15,20,30]}
```

```
gridsearchcv = GridSearchCV(RFReg, check_parameters, n_jobs=-1, cv=10)
```

```
gridsearchcv.fit(X_train, y_train)
```

```
print('Best parameters found:\n', gridsearchcv.best_params_)
```

```
# In[178]:
```

```
from sklearn.ensemble import RandomForestRegressor
```

```
# Creating an instance of the random forest regressor
```

```
RFReg = RandomForestRegressor(max_depth=30, random_state=0)
```

```
# fitting the regression model to the training data
```

```
X_train2, X_test2, y_train2, y_test2 = train_test_split(X4, Energy, test_size=0.2,
```

```
random_state=0, shuffle= "False")
```

```
y_train2 = y_train2.values.ravel()
```

```
y_test2 = y_test2.values.ravel()
```

```
RFReg.fit(X_train2, y_train2)
```

```
#Predicting on the training data
```

```
Predicted_Train2= RFReg.predict(X_train2)
```

```
#Calculating R2 score and Root mean square error
print(r2_score(y_train2, Predicted_Train2))
print(mean_squared_error(y_train2, Predicted_Train2))
```

```
# In[179]:
```

```
Predicted_Test2 = RFReg.predict(X_test2)
```

```
#Calculating R2 score and Root mean square error
print(r2_score(y_test2,Predicted_Test2))
print(mean_squared_error(y_test2,Predicted_Test2))
#Predicted_Test2=pd.DataFrame(Predicted_Test2, columns=['kWh'])
#Predicted_Test2
```

```
# In[180]:
```

```
# Import the weather cost file
path= "/home/gts/Documents/PYTHON/samir1206-main/Weather_Cost.xlsx"
weather_cost = pd.read_excel(path)
weather_cost
```

```
# In[181]:
```

```
weather_cost = weather_cost.set_index('Time')
weather_cost.isna().sum()
```

```
weather_cost_updated= weather_cost.loc[:, ~weather_cost.columns.isin(['U'])]
```

```
# In[182]:
```

```
X5 = sc1.transform(weather_cost_updated)
```

```
# In[183]:
```

```
predicted = RFReg.predict(X5)
```

```
predicted.shape
```

```
# In[184]:
```

```
#Converting the predicted array into a dataframe so it is easier when plotting to show  
the time index
```

```
predicted1= pd.DataFrame(predicted, columns=['kWh'])#.to_csv('predictions.csv')
```

```
predicted1
```

```
# In[185]:
```

```
#Import the index from the weather cost file
```

```
predicted1['Time']= weather_cost.index
```

```
predicted1
```

```
#predicted= pd.DataFrame(predicted).to_csv('predictions.csv')  
#predicted= pd.DataFrame(predicted, columns=['Time']).to_csv('predictions.csv')
```

```
# In[186]:
```

```
header=["kWh","Time"]  
predicted1.to_csv('output.csv', columns = header,index=False)
```