DESIGN OF AN IOT-BASED LOADS PRIORITY AND PREDICTIVE MAINTENANCE FOR INDUSTRIAL TRANSFORMER SYSTEMS

By

Eric TUYISHIMIRE

Reference number: 221030901

A dissertation submitted to the African Centre of Excellence in Energy for Sustainable Development (ACE-ESD)

College of Science and Technology, University of Rwanda

In Partial Fulfilment of the requirement for the Degree of Master of Science in Electrical Power Systems

Supervisor: Dr Innocent NKURIKIYIMFURA

November, 2023
KIGALI-RWANDA

Declaration

I, the undersigned, declare that this Project proposal is my original work, and has not been presented for a degree in University of Rwanda or any other universities. All sources of materials that will be used for the thesis work will have been fully acknowledged in the correct academic format.

Names: Eric TUYISHIMIRE

Signature:

Approval

This thesis proposal has been submitted for examination with my approval as a university advisor.

Supervised: Dr. Innocent NKURIKIYIMFURA

Signature:

Date of Submission: ……., 2023

Acknowledgements

I am appreciative of the Almighty God's mercies in my life and his guidance in getting me this far. I give him thanks for my excellent health and the resources that allowed me to carry out this research.

I want to express my gratitude for the outstanding assistance provided by my supervisor, Dr Innocent NKURIKIYIMFURA. I appreciate all of your committed help and advice during the procedure. I appreciate your support and openness to sharing your insights. You have given me the self-assurance and competence I need to carry out this research and others in the future. I will always be appreciative of all that you taught me when you were in charge.

I also value the moral support I receive from my friends, colleagues, and the ACE-ESD team. We appreciate your support, suggestions, and presence. To all of my professors who helped me finish the course. I appreciate it.

Abstract

This research project's goal is to address the issues with electricity supply that customers encountered in developing industries, where most of the time the power outage is the big issue to deal with. The industrial transformer is the key component of the industry for power transition to satisfy electrical requirements of loads. In this context, the lack of Load priority and Predictive Maintenance of Industrial Transformers causes the problems to for ensuring power stability for the machines. Establishing a reliable and affordable electricity supply is crucial for power utilities to maintain a positive image and business orientation. modern technologies, particularly the Internet of Things (IoT), can be a good solution for Load priority and Predictive Maintenance. Many Industrial Transformers do not yet have a remote mechanism in place to quickly Load priority and Predictive Maintenance of Industrial Transformers. Due absence of this good technology, it is very difficult to deal and react on the abnormality of power supply systems, and can result to damage to of the connected loads and transformers. The IoT system and machine learning composed of microcontroller has been introduced to deal with these issues. The main parameters that are automatically detected are: voltage, current and temperature. When the transformer is overloaded, the system switching relay to disconnect the same machines to ensure the safety of the transformer. Remote control has also the ability to activate the buzzer when there is any fault. The integration of an IoT devices will support in Loads priority and Predictive Maintenance.

Keywords: IoT, industrial transformers, predictive maintenance, load prioritization, remote monitoring, microcontroller, current sensor, voltage sensor, machine learning, power distribution, operational efficiency.

Table of Contents

List of figures

List of tables

Abbreviations

AC: alternating current

AT: Attention

DC: direct current

GPRS: General Packet Radio Service

GSM: Global System for Mobile Communications

IoT: Internet of Things

SMS: Short Message Service

TDMA: Time Division Multiple Access

Chapter 1 Introduction

1.1 Background

The industrial sector has a long-standing history of striving for improved efficiency and cost reduction. Over the years, various technologies have been employed to enhance operational processes. Before the emergence and integration of the Internet of Things (IoT), traditional approaches were employed to operate transformers in power distribution systems. However, these methods could not monitor transformers in real time and optimize their performance, leading to challenges in ensuring reliability and minimizing downtime. Before IoT technology, operators heavily relied on periodic manual inspections to assess the condition of transformers. These inspections involved visual assessments, temperature measurements, and basic diagnostic tests [1]. Maintenance activities were primarily reactive, with operators responding to issues or failures as they occurred. This reactive approach resulted in unplanned downtime and potential capacity accordingly. However, this approach did not account for dynamic variations in load patterns, potentially resulting in overloading or underutilization of transformer capacity [2]. Limited Performance Insights: Without continuous monitoring capabilities, operators had limited visibility into the real-time performance of transformers. This limited insight made it challenging to identify potential issues, optimize load distribution, and make informed decisions about maintenance and replacement strategies [3]. Load prioritization and management relied on general guidelines and predefined load-shedding schemes. There was limited flexibility to dynamically adjust load distribution based on criticality or real-time demand fluctuations. This approach often resulted in suboptimal performance and increased the risk of transformer failures during peak demand periods [4]. However, with the introduction of IoT technology and its integration into transformer systems, a transformative shift occurred. This proposal introduces an IoT-based system that leverages a network of sensors, communication devices, and cloud-based analytics tools. This integration enables real-time monitoring of transformer performance and load characteristics. The proposed system prioritizes loads based on their criticality, allowing the most important loads to receive priority during periods of high demand. This approach ensures that transformers are not overloaded, reducing the risk of failure and extending their operational lifespan. Furthermore, the system employs predictive maintenance algorithms that utilize machine learning techniques. These algorithms analyze historical data to identify patterns and anomalies that may indicate impending failure. Such proactive maintenance reduces downtime and maintenance costs while improving overall system reliability. To evaluate the effectiveness of the proposed system, a series of simulations have been conducted. The results demonstrate its capability to reduce transformer failures and improve system performance. Real-time monitoring of transformer performance and load characteristics empowers operators to make informed decisions about load prioritization and maintenance scheduling.

1.2 Problem statement

Transformers are critical components in power distribution systems and their failure can result in significant downtime and maintenance costs, as well as safety hazards. Therefore, it is crucial to ensure their reliability and minimize downtime. Traditionally, transformers are monitored through periodic inspections and maintenance, which can be time-consuming and costly. In addition, the loads on the transformer are not prioritized, which can result in overloading during periods of high demand, leading to potential failures.

1.3 Objectives

1.3.1 Main objective

The main objective of this research is to design and implement a prototype of an industrial transformer with remote control and monitoring system for predictive maintenance and load prioritization using IoT-based technologies.

1.3.2 Specific Objective

- To design t a prototype of IoT-based system used in loads prioritation and predictive maintenance for industrial transformers system.
- To design a Remote monitoring and control of the designed prototype.
- To implement the designed prototype.
- To test and characterize the designed prototype

1.4 Hypotheses

The hypotheses for this developed IoT system in addressing electricity supply challenges in developing industries are threefold. First, the implementation hypothesis posits that deploying the system significantly enhances electricity supply reliability by swiftly detecting and isolating abnormalities in distribution transformers. Second, the cost-effectiveness hypothesis suggests that the affordable IoT solution is economically viable compared to traditional monitoring methods, making it conducive to widespread adoption in resource-constrained settings. Lastly, the intervention speed hypothesis proposes that the integration of WiFi technology and real-time notifications expedites interventions, thereby reducing power outage durations and mitigating potential damage to industrial transformers and connected industrial machines.

1.5 Study Scope

This project of IoT-Loads priority and Predictive Maintenance for Industrial Transformers system, collect and generates essential technical parameters (current, voltage, temperature) on the secondary side of Industrial transformers. It performs analytics to have a real time decision-making of what to do in terms of control, Loads priority, Predictive Maintenance, quick interventions and avoiding the hazard incidents related to vandalism, especially in the industrial sector for transformer.

Chapter 2 Literature review

2.1. Introduction to transformer

2.1.1 Construction of transformer

In general, the transformer is the major component in the power transmission, distribution and consumption systems. It is composed of primary and secondary windings. It can be step up or down transformer type. The step up transformer is used to level up the voltage for transmission purpose while the step down one is used of power distribution at load side. Figure 1 describes the typical transformer providing the final level in the power grid, stepping down the voltage to match with the nominal voltages of loads at consumption side.

In general, transformer has four parts: The Input Connection, the Output Connection, the Windings or Coils, and the Core.

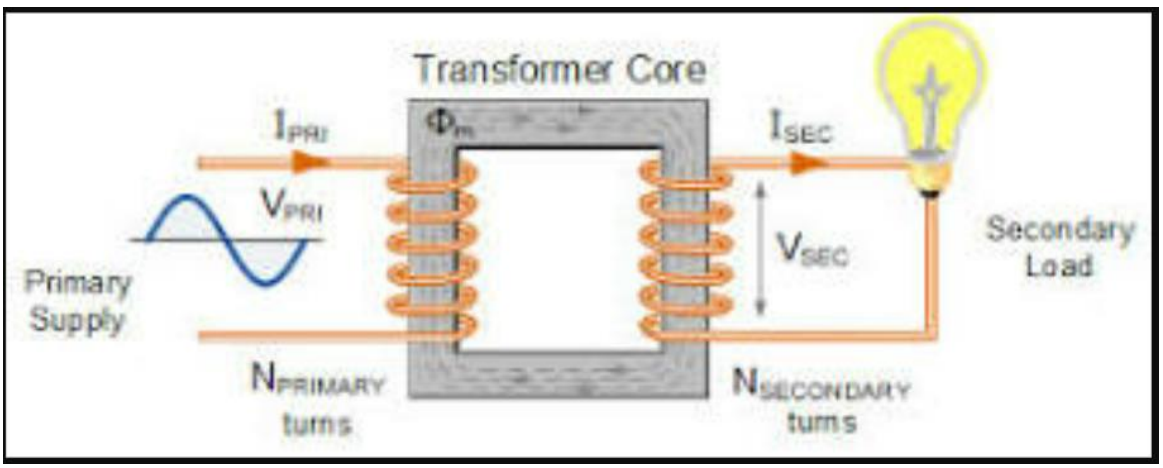

Figure 1: Schematic of a Transformer

Mainly, below are major parts of the transformer:

- **Input terminal:** also called the primary connection of the transformer
- **Output terminal of a Transformer**: also called the secondary side of the transformer, Usually, the voltage in the transformer's output side (or secondary side) is lower than in the primary side.

- **Winding:** All types of transformers have two windings. It is divided into the primary and the secondary winding. The primary one has the function of drawing power from the source. The secondary winding transfers electric power to electrical equipment.
- **Core:** The Core of the transformer core provides a path that controls the magnetic flux created in the transformer. Typically, the core is not a solid bar of steel. It includes many laminated steel sheets, or layers folded neatly. This design is to eliminate or decrease heating.

  If the output winding voltage is higher than the input voltage, then the secondary winding has more wire turns than the primary one. So, the output voltage increased higher, also called "a step-up transformer." Whereas, if the output voltage is less than the input voltage, it's called "a step-down transformer. Figure 2 shows the typical electrical power system [5]

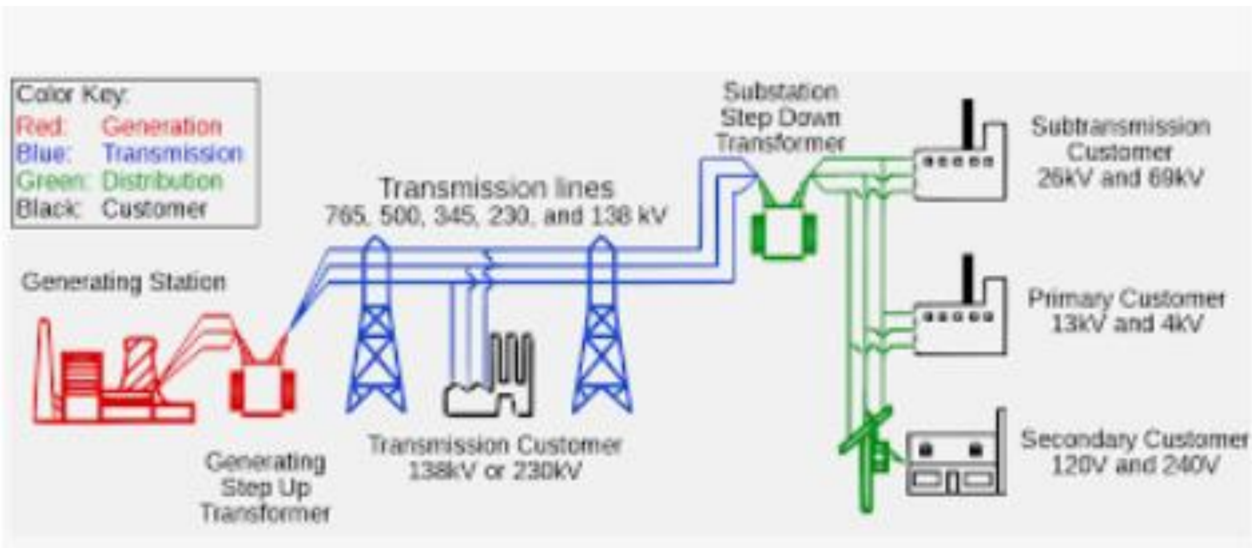

Figure 2: Electric power system

In fact, in an electric power system; power is generated by power generation stations, transmitted by transmission network systems and distributed to the load systems.

2.1.2. Transformers Faults

- **Internal faults of transformer**

Earth fault

Earth fault winding will result in currents that depend on the source, neutral grounding impedance, leakage of reactance of the transformer, and the position of the fault in the windings. The winding connections also influence the magnitude of the fault current

Core Faults

Core faults are caused by insulation breakdown and permitting sufficient eddy current to flow to cause overheating and damage to windings

Inter-turn faults

A short circuit of a few turns of the winding will give rise to high currents in the short-circuited loops

Phase to Phase faults: Phase-to-phase faults will result in the flowing of high current

Tank Faults: These faults cause the loss of oil and reduce winding insulation and abnormal rises in temperature.

- **E*xternal factor of transformer faults.***

Apart from internal faults that may occur in the transformer, there are also some faults caused by external abnormal conditions that cause stresses to the transformer resulting in over or under voltage, over current and overheating of the transformer. In this project, we shall develop a remote control system that will prevent or minimize the above faults.

2.1.3 Types of transformer maintenance

- **Unscheduled Maintenance:**

This type of maintenance is based on a reactionary mode of operation. That is to say, maintain the equipment when it breaks down, otherwise leave it alone.

Ordinary Maintenance:

This type of maintenance takes irregular visual inspection and making repairs, adjustments, and replacements as necessary

Routine Basis (Preventative Maintenance):

This maintenance consists of performing preventive maintenance, predictive maintenance, and corrective maintenance. Preventive maintenance involves scheduled maintenance and testing regularly. Predictive maintenance involves additional monitoring and testing, whereas corrective maintenance involves repairing and restoring transformer integrity to its original condition when exacerbated conditions are discovered.

This maintenance can be summarized by the following:

- Maintain transformer protective coating
- Test and maintain transformer insulation systems
- Inspect and maintain transformer auxiliary devices
- Control transformer heat
- Maintain transformer bushing insulation

Testing and Checking

- Always maintain to keep oil filled up to the desired level in Magnetic Oil Gauge (MOG)
- Change the silica gel if its color changes to pink.
- Seal any leakage detected.

Monthly Interval Transformer Maintenance

- The oil level in the oil cap must be checked a monthly interval.
- Breathing holes in the silica gel breather should also be checked and properly cleaned for proper breathing action.
- If your electrical transformer has oil-filling bushing, make sure that the oil is filled up to the correct level.

Maintenance of Transformer on Half Yearly Interval

The transformer oil must be checked on a half-yearly interval, for dielectric strength, water content, acidity, sludge content, flash point, and resistivity for transformer oil.

Yearly Transformer Maintenance

- Oil pumps, air fans, among other items that are used to cool down a transformer and control circuit must be inspected annually.
- Ensure to clean all the bushings of your electrical transformer with only soft cotton.
- Oil conditions should be carefully examined yearly.
- Ensure to clean out the inside of all of the marshalling boxes yearly.
- Check the proper functioning of the space and illumination heaters.

- Terminal connections of control and relay wiring should be tightened at least once a year.
- The Control Panel has to be cleaned with a proper cleaning agent.
- Ensure to measure the resistive value of the earth connection
- Mechanical inspection of Buchholz relays should be carried out yearly. [7]

2.2 Related work

IoT-based load priority and predictive maintenance for industrial transformers is a relatively new field of research, and there are only a few studies that have been published in this area. Here are some of the related works and their references:

In the dynamic field of IoT-based load prioritization and predictive maintenance for industrial transformers,Kumar . El-Sayed contribute a predictive maintenance system for electrical transformers. Their research integrates IoT technologies like sensors and cloud computing, aiming to enhance the efficiency of maintenance practices and streamline operations, yet potential gaps in data security, real-time analysis, scalability, and algorithmic refinement warrant attention for enhanced efficiency and seamless integration within existing infrastructure [8]

Hancke et al. present a specialized IoT-based predictive maintenance system for power transformers. By incorporating machine learning algorithms, their study predicts the remaining useful life of transformers, optimizing maintenance strategies and significantly contributing to overall system reliability and longevity yet potential gaps in scalability, real-time analysis optimization, and integration with current infrastructure present opportunities for refining maintenance strategies and fortifying overall system dependability [9]

Jena and Panda advance the discourse with an IoT-based transformer predictive maintenance system. This research focuses on predicting the likelihood of transformer failure and prioritizing maintenance activities through machine learning algorithms, promoting proactive maintenance practices and system longevity, application may involve addressing data security concerns in IoT integration, refining predictive accuracy in diverse environmental conditions, and optimizing cost-effective implementation for widespread industry adoption, thus ensuring robustness and practicality in real-world applications [10]

Joshi and Bhadane introduce a smart transformer monitoring system utilizing IoT. This comprehensive approach monitors transformer health and promptly alerts maintenance personnel to faults, emphasizing real-time response and proactive safeguarding of transformers, thereby ensuring the reliability of power distribution systems, gaps include the system's scalability to accommodate diverse transformer types, addressing interoperability challenges with existing infrastructure, and optimizing energy-efficient protocols for prolonged IoT device operation, aiming to ensure broader applicability, seamless integration, and sustainable long-term performance in varied operational environments [11]

Jadhav et al. presents an IoT-based transformer monitoring and control system. Integrating sensors and wireless communication, their study emphasizes remote monitoring and control of transformer operations, highlighting the importance of real-time oversight for optimal system performance and the efficient management of industrial transformers, gaps include refining the system's data processing capabilities for handling large-scale sensor data, ensuring robust cybersecurity measures to safeguard remote access, and addressing compatibility challenges with existing infrastructure for seamless integration and industry-wide adoption, aiming to enhance scalability, security, and interoperability while optimizing overall system efficiency and reliability [12]

IOT-based health monitoring of electrical equipment could assist in replacing the equipment before it fails and ensure that power is uninterrupted. The proposed health monitoring system uses a temperature sensor, potential transformer, and current transformer to monitor the distribution transformer's temperature, voltage, and current in real-time. This data is sent to a remote server so that it can be tracked and appropriate action can be taken to prevent an electricity supply outage. [13]

The transformer needs to be inspected frequently to avoid expensive repairs being needed to maintain a trouble-free power supply. These are expensive components of the distribution system. We frequently have severe power outages as a result. To ensure the security and dependability of a transformer's power function, the research focuses on live monitoring and fault detection techniques for intelligent power transformers. The primary goal is to meet the deadlines for quick measurements and inspections of parameters such as transformer voltage and current, oil temperature, and oil level. We also regularly check for sparks, flames, or smoke, and we send out alerts as needed. [14]

Chapter 3 Research methodology

3.1 Methodology

In this project, data were collected on IoT-Loads priority and Predictive Maintenance for Industrial Transformers systems using data collection methods such as Interviews, Observation and the internet.

An interview is a research method used to determine how effective this study solution is in Industrial. The data were collected on IoT-Loads priority and Predictive Maintenance for the Industrial Transformers system. The collected data gave information on various issues Industrial sector and accessible techniques to solve those challenges; as a result, we discovered gaps between those strategies and sought solutions to close those gaps.

3.2 Observation

It is a method of gathering data through watching behavior and describing the seen objects. It helped in comprehending the system's parameters and operating principles.

3.3 Internet

This approach was used to gather secondary data from reports, files, and online literature discussing the relevance of this project, how current technology is being investigated, and how existing systems are operating. It will assist me in identifying prior research gaps and determining strategies to close those gaps.

3.4 Technology Used

To enhance the Industrial Transformers for the industrial sector in this thesis, a Loads priority and Predictive Maintenance system of a smart Industrial Transformers system prototype was implemented, for that the following technologies were used:

3.5 Data acquisition technology

Data acquisition technology was used to collect signal data from physical devices, such as microcontrollers, temperature, current, and voltage sensors, and then convert the data to a digital signal that the microcontroller can understand.

For communication, the prototype system employs a WiFi internet gateway, which allows for speedier connection and data transmission.

3.6 Requirements

As shown below, this IoT-Loads priority and Predictive Maintenance for Industrial Transformers system prototype is made up of two parts: hardware and software:

Hardware Requirements composed of temperature sensor, current sensor, voltage sensor, NodeMcu, PCB, Jumper wires, and power supply.

After Developing, testing and verification of the implemented system, the findings are shown on a web base application called transformer control on the following link transformer.vrt.rw. Where the results are stored, analyzed, visualized and processed. This Application helps to send Text Messages to the pertinent parties when the stated threshold value reaches the greatest level and triggers an SMS depending on the occurrence. This web based application was developed using HTML5, CSS, PHP and JS technologies.

3.6.1 Functionality

The system is composed of three main parts namely the sensing part, data storage &analysis and data visualization part.

The acquired data from the sensors will be continually sent to the microcontroller for processing, and all recorded and processed data will be sent to the cloud for storage and analysis through WiFi internet gateway. These data will be evaluated and a report generated using a dashboard platform before a decision is made. In the event of an aberration relating to the threshold value specified, the system will be able to notify.

3.7 System structure

It shows all diagrams used to describe the system’s structure, codes and testing activities.

System structure is composed of planning, analyzing, design, building of prototype, implementation and testing

The models specify the stages and sequences of processes to be performed (Figure 3).

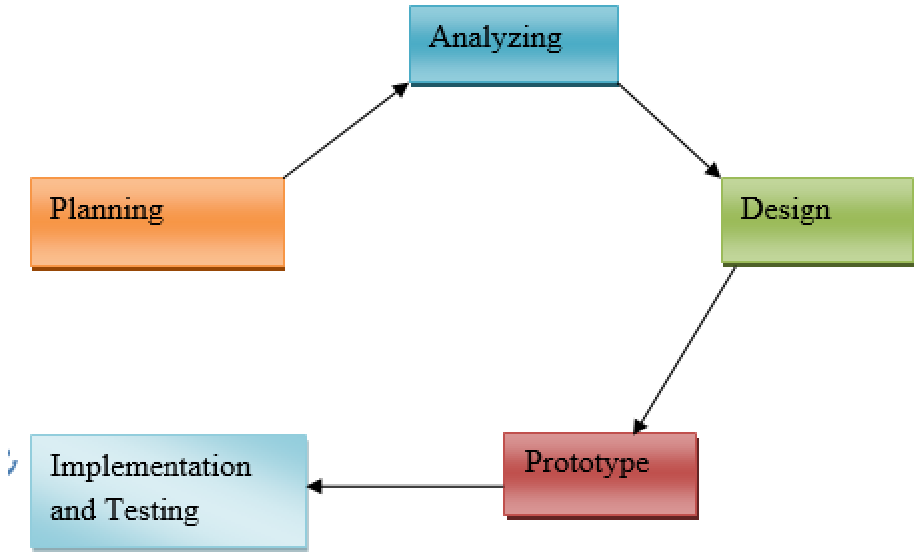

Figure 3: System model development process

3.8 Planning stage

Following the discovery of a topic, planning is the process of considering the activities necessary to reach a desired outcome. It also deals with budget, project timeline, cost projection, and risk, all of which must be updated on a regular basis.

3.8.1 Analysis and Defining Requirements

This phase entails examining the project's performance at various phases of development as well as examining all system needs. Data will be collected using various ways throughout this phase, and the resulting information will be evaluated to aid in the system's design and implementation.

3.9 Design

The design phase comes before the coding phase. The application's details are now set and described. The team will work on the application architecture and programming during this stage, which will include establishing the application's programming language/industry standards and methodologies for solving issues and completing certain activities. The team starts working on the user interface and deciding which platforms the program will operate on. Last but not least, there's security. What safeguards will be in place for the software? How will user data and passwords be protected? This is the time to address these concerns.

System Prototype: It is the other stage to make a working product model, demonstrating its purpose.

System testing: The system will be tested for checking if the software and hardware component are working properly. After developing the proposed system, I will test the following materials to see if they function as expected:

- Temperature, current sensor and voltage sensor should measure temperature, current and voltage on transformer
- NodeMcu should receive the sensed data, and process it.
- WiFi module should receive the processed data and send it to the cloud and also it should send the SMS to industrial manager.
- The dashboard platform should provide a graph of temperature, current and voltage changes, as well as the data's corrected history

Chapter 4 System analysis and design

4.1 Conceptual system architecture

The below Figure 4, shows the conceptual system architecture through a block diagram, which is made up of three primary components: data processing, data storage, and data analysis. Sensors collect data from the physical world, the microcontroller processes the stimuli, and then via WiFi Module, the data is sent to the cloud platform for processing, analysis, and storage, where choices are then made.

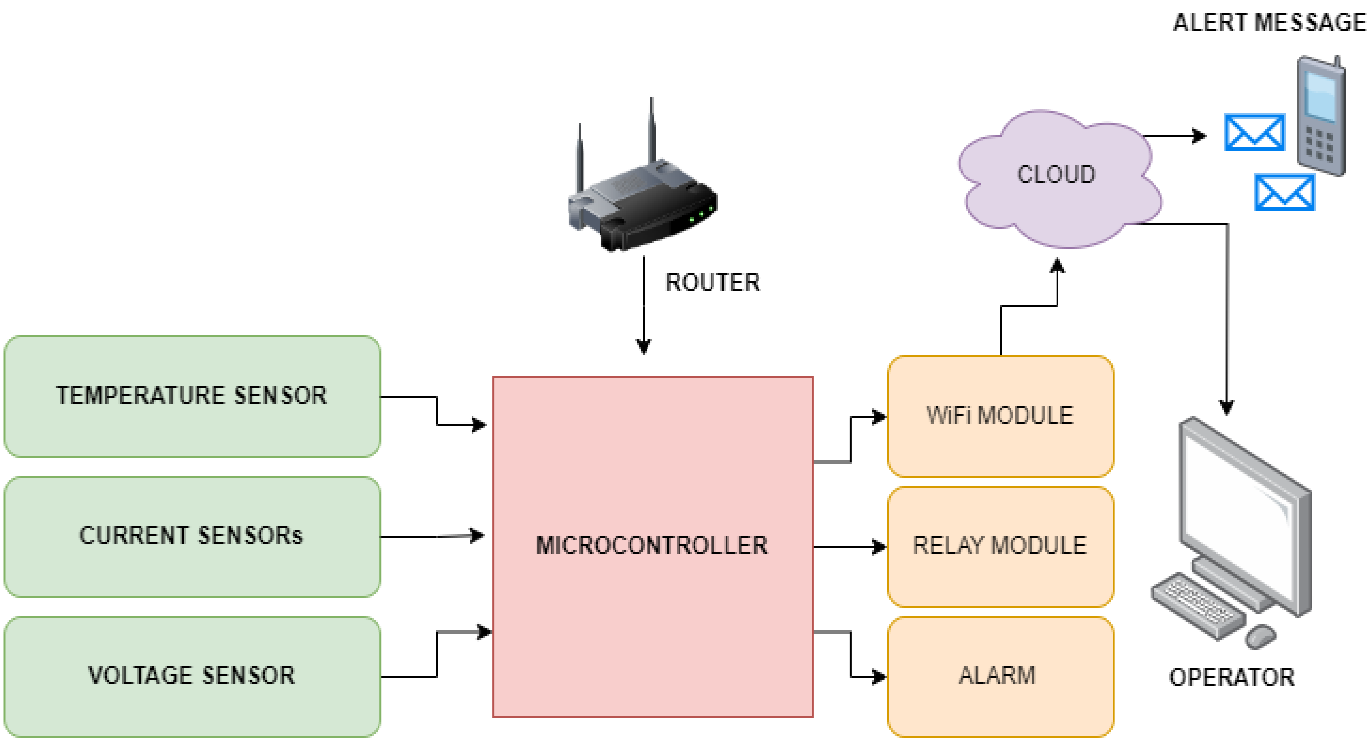

Figure 4: The block diagram

4.2 Circuit diagram

Figure 5, shows the circuit diagram of the prototype system design for single-phase and three phase transformers. It shows how the prototype performs its tasks, its main parts from the power source, the sensing data, data processing, communication channel, server and data visualization on the IOT Platform. Temperature sensor, current sensor and voltage sensor are used to capture data from the physical environment, Microcontroller processes the information before sending it immediately to the server application over the internet for further processing and analysis.

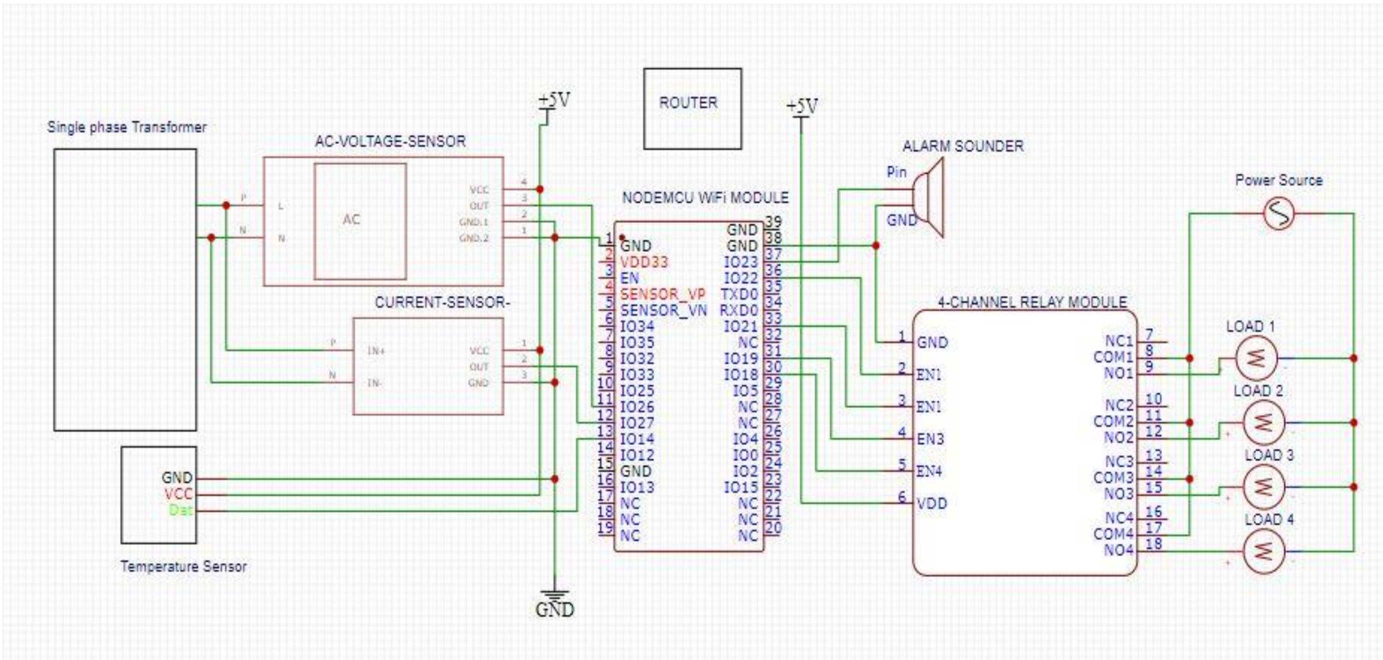

Figure 5: Circuit diagram of the single-phase system

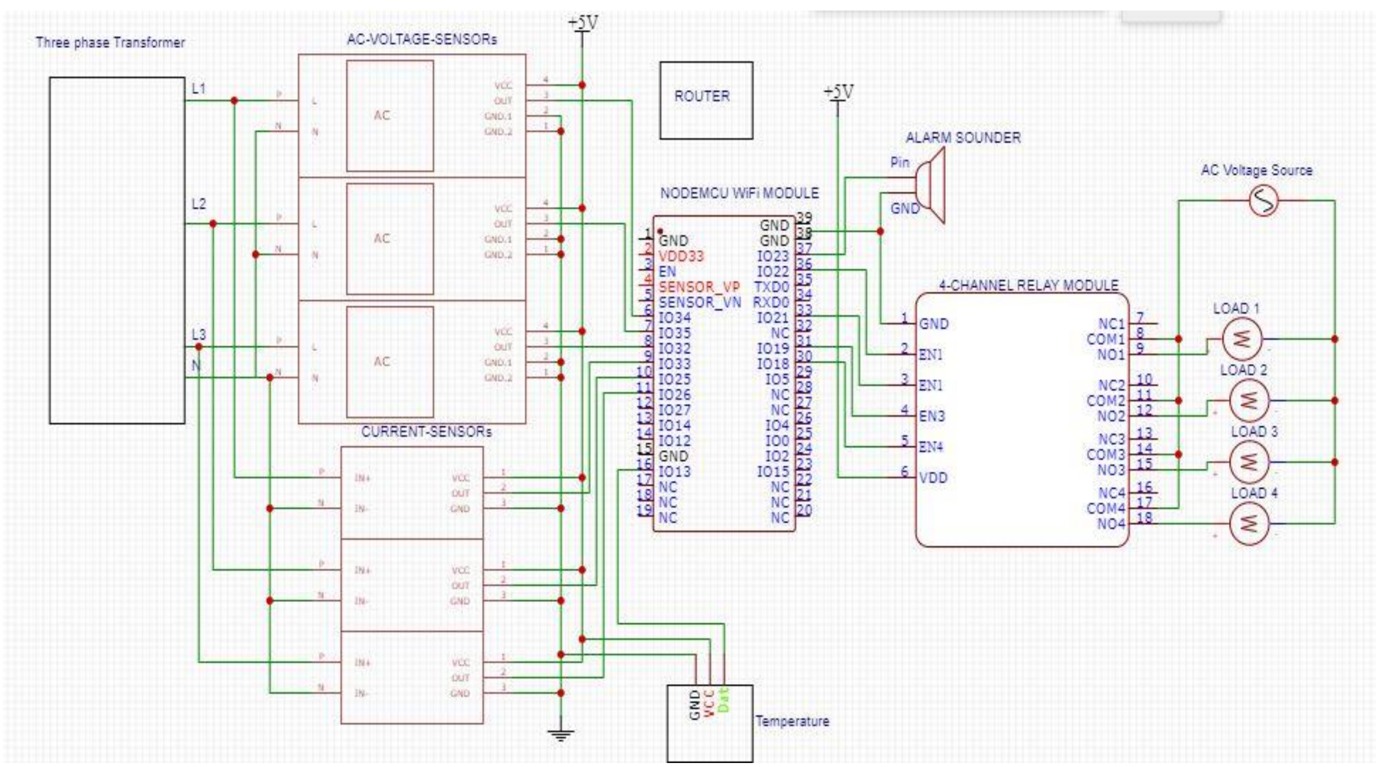

Figure 6: Circuit diagram of the three-phase system

4.3 Electronic Components

4.3.1 NodeMCU

As per Figure 6NodeMCU is built based on the mature ESP8266 technology to take advantage of the abundant resources available on the web. NodeMCU has ESP-12-based serial WiFi integrated on board to provide GPIO, PWM, ADC, I2C and 1-WIRE resources at your fingertips, built-in USB-TTL serial with super reliable industrial strength CH340 for superior stability on all supported platforms. ESP8266 is a highly integrated chip designed for the needs of a new connected world. It offers a complete and self-contained Wi-Fi networking solution, allowing it to either host the application or offload all Wi-Fi networking functions from another application processor. ESP8266 has powerful onboard processing and storage capabilities that allow it to be integrated with the sensors and other application-specific devices through its GPIOs with minimal development up-front and minimal loading during run-time. Its high degree of on-chip integration allows for minimal external circuitry, and the entire solution, including front-end module, is designed to occupy minimal PCB area.

***Table 1:* Technical specification**

Parts	Specification
Microcontroller	ESP-8266 32-bit
NodeMCU Size	49mm x 26mm
Carrier Board Size	n/a
0.9" (22.86mm)	1.1" (27.94mm)
Clock Speed	80 MHz
USB to Serial	CP2102
USB Connector	Micro USB
Operating Voltage	3.3V
Input Voltage	4.5V-10V
Flash Memory/SRAM	4 MB / 64 KB
Digital I/O Pins	11
Analog In Pins	1
ADC Range	0-3.3V
UART/SPI/I2C	1 / 1 / 1

WiFi Built-In	802.11 b/g/n
Temperature Range	-40C to 125C

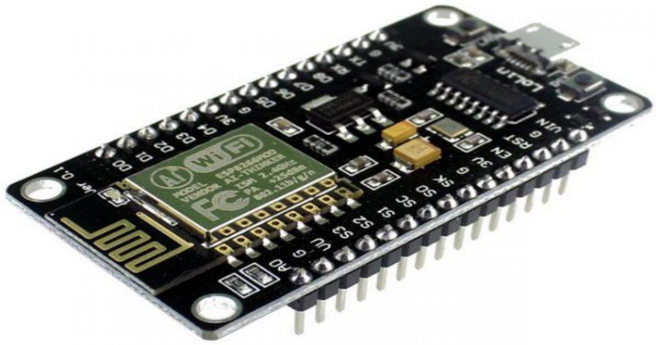

Figure 16: WiFi module (NodeMcu)

4.3.2 Current sensors:

Sensing and controlling current flow is a fundamental requirement in a wide variety of applications including, over-current protection circuits, battery chargers, switching mode power supplies, digital watt meters, programmable current sources, etc. This ACS721 current module is based on an ACS712 sensor, which can accurately detect AC or DC. The maximum AC or DC that can be detected can reach 30A, and the present current signal can be read via the analog I / O port of the Microcontroller and the Figure 7 shows current sensor.

Technical specification

- The module can measure the positive and negative 30 amps
- Pin 5V power supply, on-board power indicator
- Using gold-plated circuit boards
- No test current through the output voltage is VCC / 2

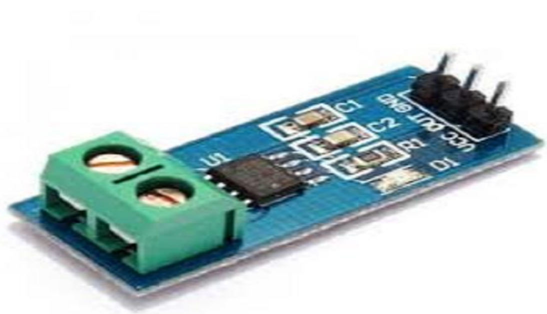

Figure 7: current sensor

4.3.3 Voltage sensors

A01B is an active output single-phase AC voltage transformer module has an onboard precision op amp circuit, the signal for precise sampling and appropriate compensation function, It used an measure the 250V AC voltage corresponding to the analog output that can be adjusted. Note that the output signal is a sine wave and it can be used for AD sampling Figure 8.

Technical specification

- Rated Input Current: 2mA
- Retardation (rated input): “20′ (input 2mA, sampling resistance 100Ω)
- The linear range: 0 ~ 1000V
- Isolation voltage: 4000V
- Operating temperature: -40 ° C- + 70 ° C
- Supply voltage (VCC): 5-30V

Figure 8: voltage sensor

4.3.4 Relay

Figure 9 describes DC Power relay which is electromechanical switching device that can control the automatic isolate loads and transformer.

Figure 9: Relay module

1. **Buzzer**: this buzzer module and you'll be rewarded with a loud noise when there is emergency fault on transformer (Figure 10).

Figure 10: Buzzer

4.4 Layers of the system design architecture

The below figure 16 shows clearly different stages of our system architecture.

4.4.1 Perception/Sensing Layer

The first layer of any IoT system involves "things" or endpoint devices that serve as a conduit between the physical and the digital worlds. Perception refers to the physical layer, which includes sensors and actuators that are capable of collecting, accepting, and processing data over the network. With our prototype, the physical layer is created by actuators like relay module and buzzer as well as sensors like the temperature, current and voltage sensor monitors the temperature, current and voltage of industrial transformer ensuring that all of our transformer are healthy.

4.4.2 Network Layer

Describe the general flow of data through the application. Data gathered by the sensor devices must be sent and processed. The network layer performs that function. It enables connections and communication between these devices and other servers, smart devices, and network devices. Additionally, it manages all device data transfers. With our prototype, through the use of a WiFi module and an application programming interface, sensed data for temperature, current and voltage are sent from the perception layer to the processing layer for communication purposes.

4.4.3 Processing Layer

This layer aids in storing, processing, and analyzing data that comes from the network layer. Data analysis and storage allow for actions to be done at the application layer for decision-making.

4.4.4 Application Layer

User interaction takes place at the application layer, which delivers application-specific services to the user. Various applications services have been developed specifically by making join of services in between and data visualization. In this project, the Internet of Things was used and enable industrial transformer to be smart and we developed our own application programming interface to help the user to visualize and be able to make a decision ().

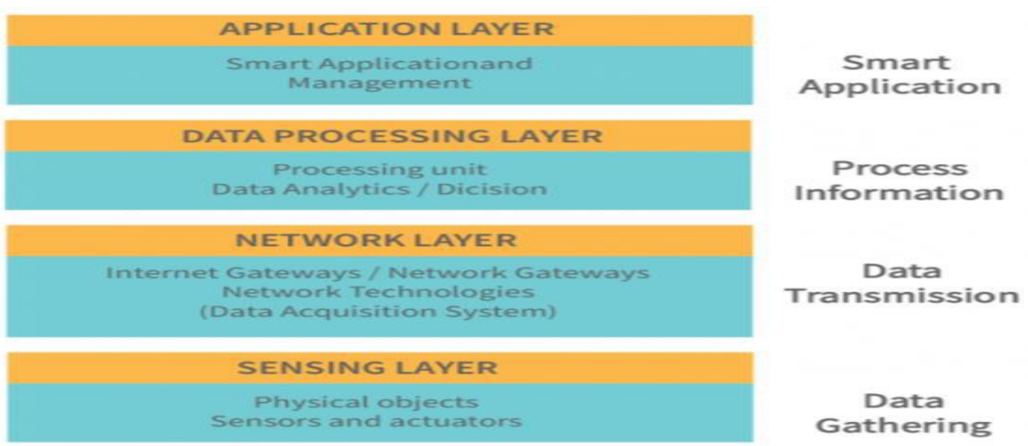

Figure 11: System design architecture

4.5 System functionality

This section provides an illustration of a set of instructions for carrying out specific tasks, from sensing to reporting. It explains how to quickly deliver real-time information to the interested parties and how to monitor the industrial transformer using the Internet of Things.

A text message notice is provided to the involved parties to help them make decisions when current, voltage and temperature are detected and reach the threshold values.

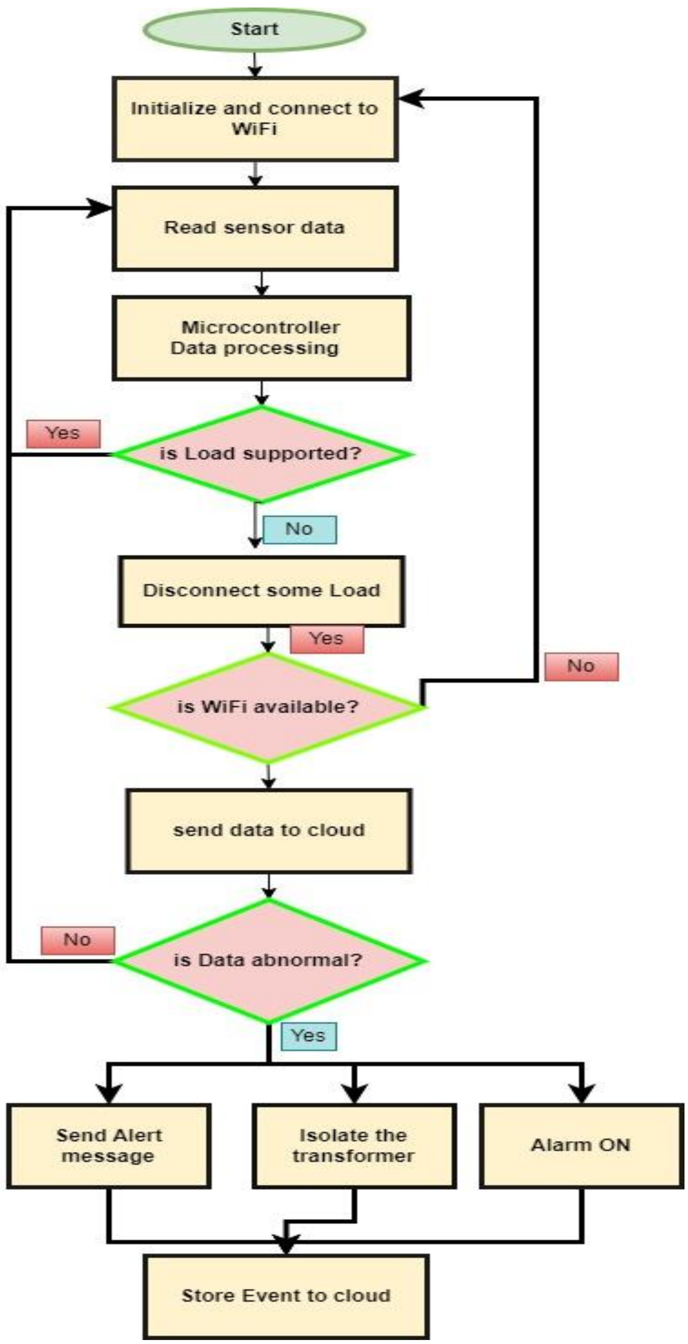

Figure 12 System flow Chart

The above Figure 12 indicates an algorithm using flowchart about IoT-Loads priority and Predictive Maintenance for Industrial Transformers system using IoT in this thesis. The system functionality continues its tasks of detecting

Chapter 5 Results and discussion

5.1 Introduction

This chapter describes the implementation of the System Prototype, finally displaying the results of the device and rationalization of findings and the graphs of effects. The prototype of the system hardware is built using NodeMCU, Current sensor, voltage sensor, and temperature sensor as the main components.

To build a prototype of the system, the interconnection of all components was done systematically and end by a successful IoT device that was used to collect data from the transformer and upload it to webbased application.

Furthermore, a web-based application called Transformer was developed using PHP, HTML, CSS, and Java Script as the user interface for information visualization, the database for transformer parameters data storage, and transformer remote monitoring.

5.2 System simulation

Sensors, actuators and gateway devices are connected to NodeMCU, and the programs' codes were uploaded into the real NodeMCU board hardware using Arduino IDE software as shown figure below.

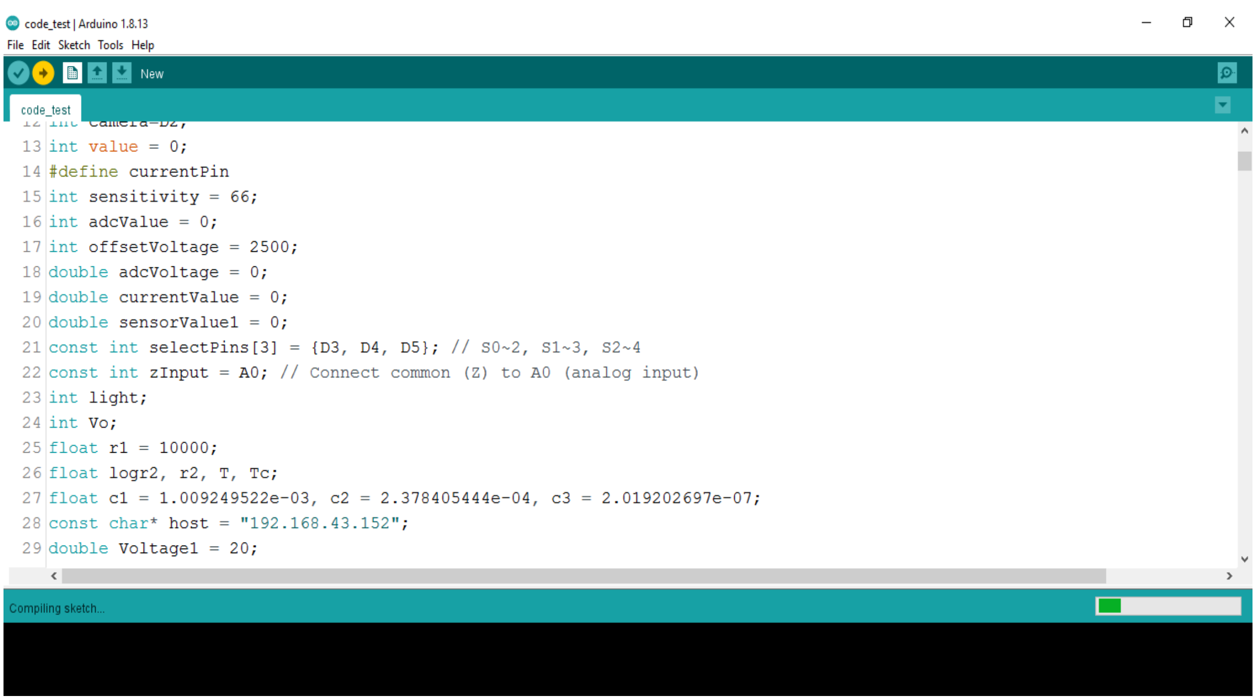

Figure 13Updoing code in NodeMcu using Arduino IDE.

5.2 System Visualization

The developed web-based application called transformer control is used to visualize the transformer information which helps the administrator and the operator to get real-time information about the operations of the transformer. the visualized information is represented in a number of the dashboard and by the graph, which shows the real-time data of the transformer.

The figure below shows the system dashboard which indicates the menu, report button, daily data and graphs menu (Figure 14).

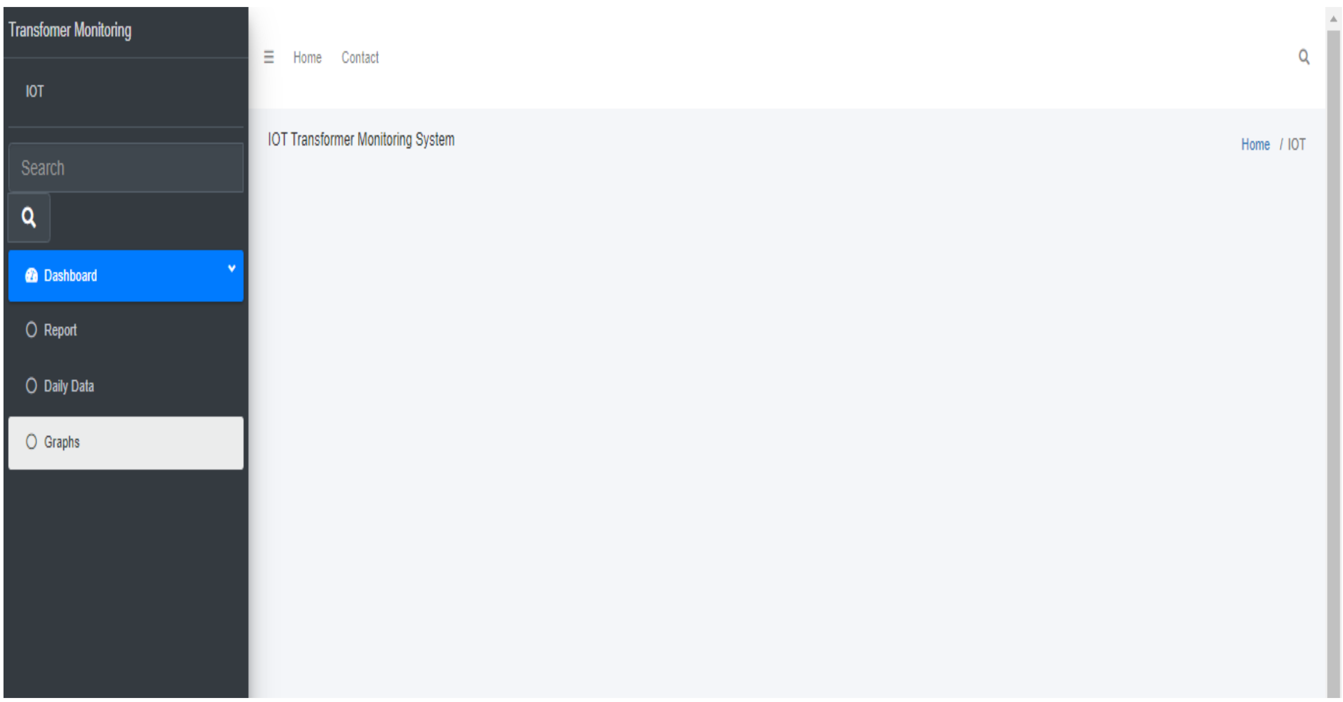

Figure 14 System dashboard.

The figure below shows system-collected data in tabular form.

Table 2: System data in tabular form.

Serial Number	Line1 A	Line2A	Line3 A	Line1 V	Line2 V	Line3 V	Temperature
Sn00200334xb046	0.8	0.8	0.8	225	223	220	36
Sn00200334xb047	0.8	0.8	0.8	225	223	220	36
Sn00200334xb048	0.8	0.8	0.8	225	223	220	36
Sn00200334xb049	0.8	0.8	0.8	225	223	220	36
Sn00200334xb050	0.8	0.8	0.8	225	223	220	36
Sn00200334xb051	0.8	0.8	0.8	225	223	220	36
Sn00200334xb052	0.8	0.8	0.8	224	224	224	36
Sn00200334xb052	0.8	0.8	0.8	224	224	224	36

5.3 Application Testing

The research utilized both unit testing and integration testing methodologies to ensure that the prototype device functioned as per the user requirements. The results of test were presented in a scorecard in table layout as illustrated in table below.

Table 3: Application Test of devices

S/NO	FUNCTION	EXPECTED RESULT	SCORE
Voltage sensor one	Measure voltage of Line one of transformer	Accurately measure the Voltage of Line one of transformer	Pass
Voltage sensor two	Measure voltage of Line two of transformer	Accurately measure the Voltage of Line two of transformer	Pass
Voltage sensor three	Measure voltage of Line three of transformer	Accurately measure the Voltage of Line three of transformer	Pass
Current sensor one	Measure the current of Line one of the transformer	Accurately measure the current of Line one of the transformer	Pass
Current sensor two	Measure the Current of Line two of the transformer	Accurately measure the Current of Line two of the transformer	Pass
Current sensor three	Measure the Current of Line three of the transformer	Accurately measure the Current of Line three of the transformer	Pass
Temperature sensor	Measure the Temperature of the transformer	Accurately measure the of transformer	Pass

In this section, we shall discuss the findings of the implementation of the IoT monitoring and control system of industrial transformers. The model was designed and developed based on the real data. the model focused on three transformer parameters namely: the voltage, current and temperature. The literature review detailed the importance of these quality aspects. industrial transformer needs to be in a normal condition where operated voltage, current and temperature don't exceed that on the name plate and you can make load priority. It was revealed that the changes in either of the parameters affect transformer and connected devices hence the need for constant monitoring. The purpose of the model was to enable the industries and Research Stations using the model to take proactive action and make maintenance plan when the thresholds for the various parameters are achieved in order avoid bad operating, damage to the transformer or connected devices. The respondents were given an opportunity to use the model and later gave their opinions on the various aspects of the model design. The subsequent sections shall discuss the aspects of the model that were tested and the respondents' feedback. Line charts were the researcher's instrument of data presentation. The following data is collected and discussed their meaning and plotted to the graph.

- **Voltage Results**

The first major role of our system is that it can load priority the voltage of the industrial transformer. By monitoring the voltage, you can protect the overvoltage of our transformer and also protect the machine. When overvoltage or undervoltage happens the system has ability to shut down transformer for safety purpose as well save that action occurring to our system for more analysis and reports.

For normal condition voltage are in range of 210-240V between phase and neutral of transformer as shown in figure below

Table 4: Normal voltage

Serial Number	Line1 A	Line2A	Line3 A	Line1 V	Line2 V	Line3 V	Temperature
Sn00200334xb046	0.8	0.8	0.8	225	223	220	36
Sn00200334xb047	0.8	0.8	0.8	225	223	220	36
Sn00200334xb048	0.8	0.8	0.8	225	223	220	36
Sn00200334xb049	0.8	0.8	0.8	225	223	220	36

Sn00200334xb050	0.8	0.8	0.8	225	223	220	36
Sn00200334xb051	0.8	0.8	0.8	225	223	220	36
Sn00200334xb052	0.8	0.8	0.8	224	224	224	36
Sn00200334xb052	0.8	0.8	0.8	224	224	224	36

For abnormal condition voltage is above 240V between phase and neutral (Table 4) , the systeme in this case where abnormal (see Table 5).

Table 5: Abnormal voltage

Serial Number	Line1 A	Line2A	Line3 A	Line1 V	Line2 V	Line3 V	Temperature
Sn00200334xb046	0.8	0.8	0.8	284	284	284	36
Sn00200334xb047	0.8	0.8	0.8	284	284	284	36
Sn00200334xb048	0.8	0.8	0.8	284	284	284	36
Sn00200334xb049	0.8	0.8	0.8	284	284	284	36
Sn00200334xb050	0.8	0.8	0.8	284	284	284	36
Sn00200334xb051	0.8	0.8	0.8	284	284	284	36
Sn00200334xb052	0.8	0.8	0.8	280	280	280	36
Sn00200334xb052	0.8	0.8	0.8	280	280	280	36

- **Current Results**

The system can detect the overcurrent of industrial transformer when it is overloaded and isolate some loads. By monitoring the overloaded, you can protect overload you can protect our transformer from overheat protect. Only loads of high priority will remain operation. In this case the normal condition is set to be 25A as shown in Table 6 and Table 7.

Table 6: Normal current

Serial Number	Line1 A	Line2A	Line3 A	Line1 V	Line2 V	Line3 V	Temperature
Sn00200334xb046	10	10	10	224	224	224	36
Sn00200334xb047	10	10	10	224	224	224	36
Sn00200334xb048	9	10	10	224	224	224	36
Sn00200334xb049	9	10	10	224	224	224	36
Sn00200334xb050	9	10	10.8	224	224	224	36
Sn00200334xb051	9	10	10.8	224	224	224	36
Sn00200334xb052	9	10	10.8	224	224	224	36
Sn00200334xb052	9	10	10.8	225	224	224	36

Table 7: Abnormal current

Serial Number	Line1 A	Line2A	Line3 A	Line1 V	Line2 V	Line3 V	Temperature
Sn00200334xb046	25	28	28	225	224	224	68
Sn00200334xb047	25	28	28	225	224	224	68

Sn00200334xb048	25	28	28	225	224	224	67
Sn00200334xb049	25	28	28	225	224	224	70
Sn00200334xb050	25	28	27	225	224	224	50
Sn00200334xb051	25	28	27	225	224	224	50
Sn00200334xb052	26	28	27	225	224	224	50
Sn00200334xb053	26	28	27	225	224	224	50
Sn00200334xb054	26	27	27	225	224	224	50
Sn00200334xb052	26	27	27	225	224	224	50

- **Temperature results**

When there is an excess flow of current, the temperature will be increased and temperature sensor will detect it and send signal to the system. By monitoring the over overheat of the transformer the system is having the capacity of isolating the entire load. In this case the abnormal temperatures set to be above 40 ^{o}C (belowTable 8 and Table 9).

Table 8:Normal temperature

Serial Number	Line1 A	Line2A	Line3 A	Line1 V	Line2 V	Line3 V	Temperature
Sn00200334xb046	10	10	10	224	224	224	36
Sn00200334xb047	10	10	10	224	224	224	36

Sn00200334xb048	9	10	10	224	224	224	36
Sn00200334xb049	9	10	10	224	224	224	37
Sn00200334xb050	9	10	10	224	224	224	37
Sn00200334xb051	9	10	10	224	224	224	38
Sn00200334xb052	9	10	10	224	224	224	38
Sn00200334xb053	9	10	10.8	224	224	224	38
Sn00200334xb054	9	10	10.8	224	224	224	38
Sn00200334xb052	9	10	10.8	225	224	224	38

Table 9: Abnormal temperature

Serial Number	Line1 A	Line2A	Line3 A	Line1 V	Line2 V	Line3 V	Temperature
Sn00200334xb046	25	28	28	225	224	224	68
Sn00200334xb047	25	28	28	225	224	224	68
Sn00200334xb048	25	28	28	225	224	224	67
Sn00200334xb049	25	28	28	225	224	224	70
Sn00200334xb050	25	28	28	225	224	224	50
Sn00200334xb051	25	28	27	225	224	224	50

Sn00200334xb052	26	28	27	225	224	224	50
Sn00200334xb053	26	28	27	225	224	224	50
Sn00200334xb054	26	2727	27	225	224	224	50
Sn00200334xb052	26	10	27	225	224	224	50

Chapter 6 Conclusion and recommendation

6.1 Conclusion

In addressing the pervasive electricity supply issues within developing industrial sectors, particularly frequent power outages and neglect of industrial transformer maintenance, a critical solution emerges through modern technology integration, specifically the Internet of Things (IoT). The absence of remote mechanisms in many industrial transformers leads to prolonged outages and machinery damage. To counter this, an affordable IoT system utilizing machine learning, temperature, voltage, and current sensors, along with a microcontroller, has been developed. This autonomous system detects vital technical parameters and safeguards transformers by employing power relays to switch off machines during overload scenarios. Additionally, it offers remote control for transformer management and alerts during anomalies. Powered by a rechargeable battery, it ensures functionality in areas with unreliable power infrastructure. This innovative approach significantly advances the resolution of electricity supply challenges in developing industries, enhancing transformer reliability, prioritizing loads, and facilitating predictive maintenance, thus fostering a more stable and efficient industrial ecosystem, notably benefiting regions like Rwanda.

6.2 Recommendation

Introducing an IoT-based Loads priority and Predictive Maintenance system for Industrial Transformers into Rwanda's industrial framework demands a strategic approach. Commencing with a localized pilot project in select industrial zones will illuminate the system's feasibility within Rwanda's specific operational environment. Collaborative partnerships with local stakeholders, including power utilities and regulatory bodies, are paramount to ensure alignment with existing infrastructure and regulations. Customizing the system to suit Rwanda's unique conditions, such as prevalent power fluctuations and commonly used industrial equipment, is crucial for its effectiveness. Empowering local technicians and engineers through comprehensive training programs will foster self-reliance in managing and sustaining the technology. Moreover, prioritizing affordability and accessibility, alongside advocating for supportive regulations and continuous technological upgrades, will ensure the long-term success and widespread adoption of this IoT solution, catalyzing improved electricity stability and industrial efficiency across Rwandan industries.

Bibliography

[1] J. A. Sabate, "Condition Monitoring of Power Transformers Using Fiber-Optic Sensors," *Sensors Journal*, vol. 16, pp. 8440-8447, Dec. 2016.

[2] V. P. Tofoli, C. M. Pereira, S. S. Costa, J. P. Oliveira, and L. A. Petry, "Short-term electric load forecasting using metaheuristic-based artificial neural networks," *2018 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1-8, 2018.

[3] Y. Zhang, Z. Lin, C. Zeng and W. Li, "Research on Power Transformer Condition Monitoring and Fault Diagnosis System Based on the Internet of Things," *2017 36th Chinese Control Conference (CCC)*, pp. 5341-5346, 2017.

[4] S. J. Choi, J. S. Choi, K. S. Eom and J. W. Park, "Smart demand response in distribution system using distributed automation system," *2016 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON*, pp. 1-4, 2016.

[5] S.V. Kulkarni and S.A, "Transformer Engineering," *Design, Technology, and Diagnostics*, 2018.

[6] James H. Harlow, "Electric Power Transformer Engineering," 2016.

[7] P. Kiameh, *Transformer Maintenance*, 2014.

[8] Kumar, R., & El-Sayed, M. M. E. A., "Predictive maintenance of electrical transformers using IoT," *Journal of Electrical Systems and Information Technology*, vol. 6, pp. 1-11, 2019.

[9] Hancke, G. P., et al, "IoT-based predictive maintenance for power transformers," *IEEE Transactions on Industrial Informatics*, vol. 13(1, pp. 335-341, 2017.

[10] Jena, S. S., & Panda, S, "An IoT-based transformer predictive maintenance system using machine learning," *Journal of Electrical Systems and Information Technology*, vol. 6(2), pp. 1-13, 2019.

[11] Joshi, V. R., & Bhadane, V. N, "Smart transformer monitoring system using IoT," *International Journal of Electrical and Computer Engineering*, vol. 8(6, pp. 4062-4070, 2018.

[12] Jadhav, P. S., et al, "IoT-based transformer monitoring and control system," *Journal of Electrical Systems and Information Technology*, vol. 7(1), pp. 1-10, 2020.

[13] K. Kumar et al., "Remote Transformer Faults Analyzing System using IoT," vol. 10, 2021.

[14] Kwarteng, S. Kwaku Okrah, B. Asante, P. Amanor Bediako, and P. Aquesi Adom, "Design and Construction of an IoT Based Distribution Transformer Condition Monitoring System," 2021.

[15] S. M. Dusmez, M. J. Hekimoglu, and H. Hekimoglu, "Reliability improvement in power distribution systems by using remote monitoring and maintenance," *2018 7th International Conference on Renewable Energy Research and Applications (ICRERA),* pp. 1422-1425, 2018.

Appendix

#	serial Number	Line 1 A	Line 2 A	Line 3 A	Line 1 V	Line 2 V	Line 3 V	Temperature	Due Time	
45	sn0020034xbO46	3.2	3.8	3.2	225	223	220	2022-10-07 12:02:26.442673	36	
46	sn0020034xbO46	3.2	3.8	3.2	225	223	220	2022-10-07 12:02:29.549113	36	
47	sn0020034xbO46	3.2	3	3.2	225	223	220	2022-10-07 12:02:37.995176	36	
48	sn0020034xbO46	0.8	0.8	0.8	225	223	220	2022-10-07 12:04:12.607057	36	
49	sn0020034xbO46	0.8	0.8	0.8	225	223	220	2022-10-07 12:04:30.309985	36	
50	sn0020034xbO46	0.8	0.8	0.8	225	223	220	2022-10-07 12:04:36.226031	36	
51	sn0020034xbO46	0.8	0.8	0.8	225	223	220	2022-10-07 12:04:40.962686	36	

IOT Engeer Emmauel

48	sn0020034xbO46	0.8	0.8	0.8	225	223	220	36	2022-10-07 12:04:12.607057	
49	sn0020034xbO46	0.8	0.8	0.8	225	223	220	36	2022-10-07 12:04:30.309985	
50	sn0020034xbO46	0.8	0.8	0.8	225	223	220	36	2022-10-07 12:04:36.226031	
51	sn0020034xbO46	0.8	0.8	0.8	225	223	220	36	2022-10-07 12:04:40.962686	
52	sn0020034xbO46	0.8	0.8	0.8	225	223	220	36	2022-10-07 18:24:37.852928	
53	sn0020034xbO46	0.8	0.8	0.8	225	223	220	36	2022-10-08 15:34:19.884838	
54	sn0020034xbO46	0.8	0.8	0.8	224	224	224	36	2022-10-08 15:38:00.667406	
55	sn0020034xbO46	0.8	0.8	0.8	224	224	224	36	2022-10-08 15:38:18.252900	

57	sn0020034xb046	0.8	0.8	0.8	284	284	284	36	2022-10-08 15:41:18.836207	
58	sn0020034xb046	0.8	0.8	0.8	284	284	284	36	2022-10-08 15:41:23.652327	
59	sn0020034xb046	0.8	0.8	0.8	284	284	284	36	2022-10-08 15:41:32.980598	
60	sn0020034xb046	0.8	0.8	0.8	284	284	284	36	2022-10-08 15:41:40.050193	
61	sn0020034xb046	0.8	0.8	0.8	284	284	284	36	2022-10-08 15:41:44.121660	
62	sn0020034xb046	0.8	0.8	0.8	284	284	284	36	2022-10-08 15:42:02.466330	
63	sn0020034xb046	0.8	0.8	0.8	280	280	280	36	2022-10-08 15:42:12.330103	
64	sn0020034xb046	0.8	0.8	0.8	280	280	280	36	2022-10-08 15:42:16.332239	

72	sn0020034xb046	25	28	288	225	224	224	68	2022-10-08 15:45:15.779173	
73	sn0020034xb046	25	28	28	225	224	224	68	2022-10-08 15:45:46.298824	
74	sn0020034xb046	25	28	28	225	224	224	67	2022-10-08 15:53:21.424037	
75	sn0020034xb046	25	28	28	225	224	224	70	2022-10-08 15:55:33.072640	
76	sn0020034xb046	25	28	28	225	224	224	50	2022-10-08 15:55:39.120649	
77	sn0020034xb046	25	28	27	225	224	224	50	2022-10-08 15:55:43.017453	
78	sn0020034xb046	26	28	27	225	224	224	50	2022-10-08 15:56:15.810321	
79	sn0020034xb046	26	28	27	225	224	224	50	2022-10-08 15:56:19.027277	
80	sn0020034xb046	26	27	27	225	224	224	50	2022-10-08 15:56:26.310027	
81	sn0020034xb046	26	27	27	225	224	224	50	2022-10-08 15:56:40.592398	
#	**serial Number**	**Line 1 A**	**Line 2 A**	**Line 3 A**	**Line 1 V**	**Line 2 V**	**Line 3 V**	**Temperature**	**Due Time**	

#	serial Number	Line 1 A	Line 2 A	Line 3 A	Line 1 V	Line 2 V	Line 3 V	Temperature	Due Time	
65	sn0020034xbO46	10	10	10	224	224	224	36	2022-10-08 15:43:41.045536	
66	sn0020034xbO46	10	10	10	224	224	224	36	2022-10-08 15:43:41.858892	
67	sn0020034xbO46	9	10	10	224	224	224	36	2022-10-08 15:43:55.013203	
68	sn0020034xbO46	9	10	10	224	224	224	37	2022-10-08 15:44:03.031861	
69	sn0020034xbO46	9	10	10.8	224	224	224	37	2022-10-08 15:44:14.839958	
70	sn0020034xbO46	9	10	10.8	224	224	224	38	2022-10-08 15:44:26.507087	
71	sn0020034xbO46	9	10	10.8	225	224	224	38	2022-10-08 15:44:40.996916	
72	sn0020034xbO46	25	28	288	225	224	224	68	2022-10-08 15:45:15.779173	
73	sn0020034xbO46	25	28	28	225	224	224	68	2022-10-08 15:45:46.298824	
74	sn0020034xbO46	25	28	28	225	224	224	67	2022-10-08 15:53:21.424037	
75	sn0020034xbO46	25	28	28	225	224	224	70	2022-10-08 15:55:33.072640	
76	sn0020034xbO46	25	28	28	225	224	224	50	2022-10-08 15:55:39.120649	
77	sn0020034xbO46	25	28	27	225	224	224	50	2022-10-08 15:55:43.017453	
78	sn0020034xbO46	26	28	27	225	224	224	50	2022-10-08 15:56:15.810321	
79	sn0020034xbO46	26	28	27	225	224	224	50	2022-10-08 15:56:19.027277	
80	sn0020034xbO46	26	27	27	225	224	224	50	2022-10-08 15:56:26.310027	
81	sn0020034xbO46	26	27	27	225	224	224	50	2022-10-08 15:56:40.592398	

System results in tabular form

Hardware system code

```
#include <ESP8266WiFi.h>

#include <SPI.h>

#include <Wire.h>

#include <SoftwareSerial.h>

SoftwareSerial wifiSerial(D7,D8);

#define WLAN_SSID       "VRT technology belongs to us"

#define WLAN_PASS       "VRT@0931"

WiFiClient client;

#define voltageSensor

int relay=D6;

int value = 0;

#define currentPin

int sensitivity = 66;

int adcValue = 0;

int offsetVoltage = 2500;

double adcVoltage = 0;

double currentValue = 0;

double sensorValue1 = 0;

const int selectPins[3] = {D3, D4, D5}; // S0~2, S1~3, S2~4

const int zInput = A0; // Connect common (Z) to A0 (analog input)

int light;
```

```
int Vo;

float r1 = 10000;

float logr2, r2, T, Tc;

float c1 = 1.009249522e-03, c2 = 2.378405444e-04, c3 = 2.019202697e-07;

const char* host = "192.168.43.152";

double Voltage1 = 20;

double VRMS1 = 0;

double AmpsRMS1 = 0;

int mVperAmp1 = 185;//use 185 for 5A

double Voltage2 = 0;

double VRMS2 = 0;

double AmpsRMS2 = 0;

int mVperAmp2 = 185;//use 185 for 5A

double Voltage3 = 0;

double VRMS3 = 0;

double AmpsRMS3 = 0;

int mVperAmp3 = 185;//use 185 for 5A

int inputValue;

int crosscount = 0;

int climb_flag = 0;

int val[100];

int max_v = 0;

double VmaxD = 0;
```

```
double VeffD = 0;

double Veff = 0;

int powe=100;

int Voltage = 0;

int buzzer=D1;

//...................int crosscount = 0;

int climb_flag2 = 0;

int val2[100];

int max_v2 = 0;

double VmaxD2 = 0;

double VeffD2 = 0;

double Veff2 = 0;

int Voltage4 = 0;

//....................

int crosscount3 = 0;

int climb_flag3 = 0;

int val3[100];

int max_v3 = 0;

double VmaxD3 = 0;

double VeffD3 = 0;

double Veff3 = 0;

int Voltage5 = 0;

void setup() {
```

```
wifiSerial.begin(9600);
Serial.begin(115200);
pinMode(relay,OUTPUT);
pinMode(buzzer,OUTPUT);
for (int i = 0; i < 3; i++){
pinMode(selectPins[i], OUTPUT);
digitalWrite(selectPins[i], HIGH);
Serial.println(); Serial.println();
Serial.print("Connecting to ");
Serial.println(WLAN_SSID);
WiFi.begin(WLAN_SSID, WLAN_PASS);
while (WiFi.status() != WL_CONNECTED) {
//delay(500);
Serial.print(".");}
Serial.println();
Serial.println("WiFi connected");
Serial.println("IP address: "); Serial.println(WiFi.localIP());}}
void loop(){
for (byte pin = 0; pin <= 7; pin++){
for (int i = 0; i < 3; i++) {
digitalWrite(selectPins[i], pin & (1 << i) ? HIGH : LOW);}
inputValue = analogRead(zInput);
if (pin == 4) {
```

```
Voltage1 = getVPP1();

VRMS1 = (Voltage1/2.0) *0.707;

AmpsRMS1 = ((VRMS1 * 1000)/mVperAmp1);

Serial.print("sensor1:  ");

Serial.println(AmpsRMS1);

Serial.print("inputValue1:  ");

Serial.println(inputValue);   }

if (pin == 5) {

Voltage2 = getVPP2();

VRMS2 = (Voltage2/2.0) *0.707;

AmpsRMS2 = ((VRMS2 * 1000)/mVperAmp1);

Serial.print("sensor2:  ");

Serial.println(AmpsRMS2);

Serial.print("inputValue2:  ");

Serial.println(inputValue);}

if (pin == 7) {

Voltage3 = getVPP3();

VRMS3 = (Voltage3/2.0) *0.707;

AmpsRMS3 = ((VRMS3 * 1000)/mVperAmp1);

Serial.print("sensor3:  ");

Serial.println(AmpsRMS3);

Serial.print("inputValue3:  ");

Serial.println(inputValue);}
```

```
if (pin == 0) {

for ( int i = 0; i < 100; i++ ) {

sensorValue1 = analogRead(zInput);

if (analogRead(zInput) > 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)-476 ;}

else {Veff = 0;}

Serial.print("Voltage1: ");

Serial.println(Voltage);

VmaxD = 0;

delay(400);

if(Veff<0){Voltage=0;}

else{Voltage=Veff;}
```

```
Serial.print("inputValue4:  ");
Serial.println(inputValue);}
if (pin == 1) {
for ( int i = 0; i < 100; i++ ) {
sensorValue1 = analogRead(zInput);
if (analogRead(zInput) > 511) {
val2[i] = sensorValue1;}
else {val2[i] = 0;}
delay(1);}
max_v2 = 0;
for ( int i = 0; i < 100; i++ ){
if ( val2[i] > max_v2 ){max_v2 = val2[i];}
val2[i] = 0;}
if (max_v2 != 0) {
VmaxD2 = max_v2;
VeffD2 = VmaxD2 / sqrt(2);
Veff2 = (((VeffD2 - 420.76) / -90.24) * -210.2)-437 ;}
else {Veff2 = 0;}
Serial.print("Voltage2: ");
Serial.println(Voltage4);
VmaxD2 = 0;
//delay(400);
if(Veff2<0){
```

```
Voltage4=0;}
else{Voltage4=Veff2;}
//delay(1000);
Serial.print("inputValue4:  ");
Serial.println(inputValue);}
if (pin == 3) {
for ( int i = 0; i < 100; i++ ) {
sensorValue1 = analogRead(zInput);
if (analogRead(zInput) > 511) {
val3[i] = sensorValue1;}
else {val3[i] = 0;}
delay(1);}
max_v3 = 0;
for ( int i = 0; i < 100; i++ )
{
if ( val3[i] > max_v3 ){max_v3 = val3[i];}
val3[i] = 0;}
if (max_v3 != 0) {
VmaxD3 = max_v3;
VeffD3 = VmaxD3 / sqrt(2);
Veff3 = (((VeffD3 - 420.76) / -90.24) * -210.2)-476;}
else {Veff3 = 0;}
Serial.print("Voltage3: ");
```

```
Serial.println(Voltage5);

VmaxD3 = 0;

//delay(400);

if(Veff3<0){

Voltage5=0;}

else{Voltage5=Veff3;}

//delay(1000);

Serial.print("inputValue4:  ");

Serial.println(inputValue);}}

if(powe==10){

SendMessage();

Serial.println("okok");

digitalWrite(relay,HIGH);

digitalWrite(buzzer,HIGH);

}

else{  Serial.println("nono");

digitalWrite(relay,LOW);

digitalWrite(buzzer,LOW);

  }

// displayInfo();

gprscode();

//delay(2000);}

void displayInfo(){
```

```
Serial.print("Connecting to ");
Serial.println( host);
const int httpPort = 80;
if (!client.connect(host, httpPort)) {
Serial.println("connection Failed");
return;}
String url = "/pollution/postData.php?";
  url += "gas1=";
  url += "9";
  url += "&gas2=";
  url += "7";
  url += "&speed=";
  url += "120";
  url += "&device_id=";
  url += "G3";
  url += "&device_idkk=";
  url += "G3";
  url += "&device_idbb=";
  url += "G3";
  Serial.print("Requesting URL :");
  Serial.println(url);
  client.print(String("GET /") + url + "HTTP/1.0 \r\n" +
               "HOST :" + host + "\r\n" +
```

```
       "Connection: close \r\n\r\n");
 unsigned long timeout = millis();
 while (client.available() == 0) {
  if (millis() - timeout > 5000) {
   Serial.println(">>>Client Timeout");
   client.stop();
   return;
  }
  while (client.available()) {
   String line = client.readStringUntil('\r');
   Serial.print(line);
  }
  Serial.println();
  Serial.println("Closing Connection");
  delay(5000);
 }
 delay(3000); //3000 = 3 seconds
 Serial.println();
}
float getVPP1(){//this will caculate the peak to peak of sensor1
float result1;
 int readValue1;        //value read from the sensor1
 int maxValue1 = 0;      // store max value here
```

```
  int minValue1 = 1024;          // store min value here
   uint32_t start_time = millis();
   while((millis()-start_time) < 1000) //sample for 1 Sec
   { readValue1 = analogRead(zInput);
       if (readValue1 > maxValue1) {
           maxValue1 = readValue1; }
       if (readValue1 < minValue1)  {
           minValue1 = readValue1;} }
   result1 = ((maxValue1 - minValue1) * 5.0)/1024.0;
   return result1;}

   float getVPP2(){//this will caculate the peak to peak of sensor1
float result2;
  int readValue2;             //value read from the sensor1
  int maxValue2 = 0;          // store max value here
  int minValue2 = 1024;          // store min value here
   uint32_t start_time = millis();
   while((millis()-start_time) < 1000) //sample for 1 Sec
   { readValue2 = analogRead(zInput);
       if (readValue2 > maxValue2) {
           maxValue2 = readValue2; }
       if (readValue2 < minValue2)  {
           minValue2 = readValue2;} }
```

```
  result2 = ((maxValue2 - minValue2) * 5.0)/1024.0;
  return result2;}
  float getVPP3(){//this will caculate the peak to peak of sensor1
float result3;
 int readValue3;             //value read from the sensor1
 int maxValue3 = 0;          // store max value here
 int minValue3 = 1024;          // store min value here
  uint32_t start_time = millis();
  while((millis()-start_time) < 1000) //sample for 1 Sec
  { readValue3 = analogRead(zInput);
      if (readValue3 > maxValue3) {
          maxValue3 = readValue3; }
      if (readValue3 < minValue3)  {
          minValue3 = readValue3;} }
  result3 = ((maxValue3 - minValue3) * 5.0)/1024.0;
  return result3;}
  void SendMessage()
{
 wifiSerial.println("AT+CMGF=1"); //Sets the GSM Module in Text Mode
 delay(1000);  // Delay of 1000 milli seconds or 1 second
 wifiSerial.println("AT+CMGS=\"0785278085\"\r"); // Replace x with mobile
number
 delay(1000);
```

```
  wifiSerial.println("Transiformer has abnormal condition");// The SMS text you want
to send

  delay(100);

 wifiSerial.println((char)26);// ASCII code of CTRL+Z

  delay(1000);

}

void gprscode()

  {

      wifiSerial.println("AT+CREG?");

      delay(100);

      wifiSerial.println("AT+SAPBR=3,1,\"CONTYPE\",\"GPRS\"");

      delay(1000);//2000

      wifiSerial.println("AT+SAPBR=3,1,\"APN\",\"MTN.internet\"");

      delay(1000);//2000

      wifiSerial.println("AT+SAPBR=1,1");

      delay(1000);//2000

      wifiSerial.println("AT+HTTPINIT");

      delay(1000);//2000
//ttps://transfomer.hss.rw/postData.php?current1=98&voltage1=98&current2=98&vo
ltage2=98&current3=98&voltage3=98&temperature=6


wifiSerial.print("AT+HTTPPARA=\"URL\",\"transfomer.hss.rw/postData.php?");

      //Serial.print("AT+HTTPPARA=\"URL\",\"cross-
country.000webhostapp.com/gprs/remote/vehiclereceivedata.php?");
```

```
//Serial.print("AT+HTTPPARA=\"URL\",\"cross-country.000webhostapp.com/all/card.php?");

 wifiSerial.print("current1=");

 //serialN=&lamptype=&light=&volt=

wifiSerial.print("3");

wifiSerial.print("&");

wifiSerial.print("current2=");

wifiSerial.print("3");

wifiSerial.print("&");

wifiSerial.print("current3=");

wifiSerial.print("3");

wifiSerial.print("&");

wifiSerial.print("voltage1=");

wifiSerial.print(Voltage);

  wifiSerial.print("&");

wifiSerial.print("voltage2=");

wifiSerial.print(Voltage4);

  wifiSerial.print("&");

wifiSerial.print("voltage3=");

wifiSerial.print(Voltage5);

  wifiSerial.print("&");

wifiSerial.print("temperature=");

wifiSerial.print(Voltage);
```

```
 wifiSerial.println("\"");
delay(1000);//2000
// set http action type 0 = GET, 1 = POST, 2 = HEAD
 wifiSerial.println("AT+HTTPACTION=0");
delay(1000);//5000
// toSerial();
// read server response
 wifiSerial.println("AT+HTTPREAD");
delay(500);//
// toSerial()
 wifiSerial.println("");
 wifiSerial.println("AT+HTTPTERM");
// toSerial();
// delay(300);
 wifiSerial.println("");
delay(1000);//10000
wifiSerial.println("AT+CIPSHUT");

}
```