IOT BASED SYSTEM FOR MONITORING AND CONTROL DATA CENTER TEMPERATURE AT HABARI NODE COMPANY

Innocent Ciza

A Project Report Submitted in Partial Fulfillment of the Requirements of the Award the Degree of Master of Science in Embedded and Mobile Systems of the Nelson Mandela African Institution of Science and Technology

Arusha, Tanzania

August, 2022

ABSTRACT

The Internet of Things (IoT) Based Systems have improved the way of life of companies as well as individuals. Through interconnected objects via intranet and Internet, IoT technologies enhance the productivities of businesses by reducing the wastage of time, reducing the expenses of the businesses by its low-cost property. The IoT is empowered by a system of detection, actuation, control, and cloud storage. IoT technologies should be applied to monitor the change of the value of valuables like temperature, voltage, gas and liquid level, etc.; it helps to get the record of the change in the value of the valuables detected. The proposed system was developed to monitor and control data center temperature through ESP32 board, DHT22 module, relays and ThingSpeak cloud. To significantly reduce the power consumption of the cooling devices, the system uses a cooling device backup which can be controlled by a microcontroller through a smartphone over the Internet. Moreover, the system assists the data center manager by automatically switching on the backup in case of the failure of one working cooling device. However, an email address and SMS are sent to the data center manager at any time when there is a collected value that is not in the normal temperature range. As a result, the project enhances the productivity of the company by reducing the cost expended in the data center infrastructure, and by reducing the activities performed by a data center manager.

DECLARATION

I, Innocent Ciza, do hereby declare to the Senate of the Nelson Mandela African Institution of Science and Technology that this project report is my original work and that it has neither been submitted nor being concurrently submitted for a degree award in any other institution.

Innocent Ciza		05.08.2022
Name of Candidate	**Signature**	**Date**

The above declaration is confirmed by:

Dr. Ramadhani Sinde		7th August 2022
Name of Supervisor 1	**Signature**	**Date**

Prof. Shubi Kaijage		7-8-2022
Name of Supervisor 2	**Signature**	**Date**

COPYRIGHT

CERTIFICATION

The undersigned certify that they have read and hereby recommend for acceptance by The Nelson Mandela African Institution of Science and Technology, a project report titled **"*IoT based system for monitoring and control Data center temperature at Habari Node company*"** in partial fulfillment of the requirements for the degree of Master of Science in Embedded and Mobile Systems, Embedded Systems specialty of the Nelson Mandela African Institution of Science and Technology.

Dr. Ramadhani Sinde		7th August 2022
Name of Supervisor 1	**Signature**	**Date**

Prof. Shubi Kaijage		7-8-2022
Name of Supervisor 2	**Signature**	**Date**

ACKNOWLEDGMENTS

First and foremost, I express my heartfelt gratitude to God for being a sanctuary, support, and assistance that never fails in times of need. I can't stop myself from thanking the Almighty for his boundless love, compassion, and protection during our student careers, for keeping me alive, and for providing me with the ability, strength, and willingness to complete this project.

My sincere thanks go to the Centre of Excellence for Information and Communication Technologies in East Africa (CENIT@EA) for providing financial assistance and sponsorship for my master's degree studies.

I would like to express my deepest gratitude to my academic project supervisors, Dr. Ramadhani Sinde and Prof. Shubi Kaijage, for their time, technical assistance, and advice during all phases of the project. They have been nice and helpful people throughout my academic career. Their excellent collaboration boosted my confidence and inspired me to work not only hard but also intelligently.

I am grateful to Eng. Lomayani S. Laizer, Technical Manager of Habari Node PLC and my host company supervisor, for his exceptional collaboration and assistance, which was a significant contribution to my work.

Finally, I would want to give special thanks to my Pastor Saimon Kaaya for his continuous sympathy, spiritual support, and prayers. My thanks may be found in the book for my classmates and colleagues from NM-AIST, as well as others whose names are not included, who contributed to the completion of this work.

May God's peace fill you to overflowing.

DEDICATION

This work is dedicated to my magnificent parents, Mr. Ngendahoruri François and Mme Ntiruvakure Eugénie, for their unwavering support and encouragement throughout my academic career.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS AND SYMBOLS

°C	Degree Celsius
AC	Air Conditioner
AI	Artificial Intelligent
API	Application Programming Interface
CCTV	Closed-Circuit Television
DC	Direct Current
DHCP	Dynamic Host Configuration Protocol
ESP	Espressif
GPIO	General Purpose Input/Output
HTML	Hypertext Markup Language
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTP	Hypertext Transfer Protocol
HVAC	Heating, Ventilation, and Air Conditioning
I2C	Inter-Integrated Circuit
ICT	Information and Communication Technologies
IDE	Integrated Development Environment
IoT	Internet of Things
IP	Internet Protocol
ISP	Internet Service Provider
IT	Information Technology
LAN	Local Area Network
MAC	Media Access Control
NAT	Network Address Translation

PC	Personal Computer
PCB	Printed Circuit Board
PHP	Hypertext Preprocessor
PLC	Public Limited Company
RAC	Real Application Cluster
RAM	Random Access Memory
SDLC	System Development Life Cycle
SMS	Short Message Service
SPI	Serial Peripheral Interface
SSID	Service Set Identifier
UART	Universal Asynchronous Receiver-Transmitter
URL	Uniform Resource Locator
Wi-Fi	Wireless Fidelity
WSN	Wireless Sensor Network

CHAPTER ONE

INTRODUCTION

1.1 Background of the Problem

A data center is a physical facility that organizations use to house their information and software (Stryer, 2010). A data center is a computer facility where a large number of computer systems are joined together to form a network, allowing them to communicate with one another as well as connect to the outside world (Sarkar, 2010). It is a critical part of company infrastructure that provides support and security of data of the company. In any data center, key areas such as reliability, security, flexibility, scalability, and emergency backup should be considered when assessing its quality (Collins, 2016).

In cloud computing, maintaining customer confidence and satisfaction and minimizing revenue losses requires delivering highly accessible and dependable services (Mesbahi *et al.*, 2018; Meza *et al.*, 2018). It depends on the power source of a data center that can be enabled by generators as a backup power supply once there is a fluctuation of the main power supply. It can also depend on Heating, Ventilation, and Air Conditioning (HVAC) systems; that is mostly used to optimize air condition in data centers. Data center equipment generates a considerable amount of heat in a tiny space because, every Watt of electrical power utilized by a system is lost to the environment as heat (Bhatia, 2019). The ambient temperature would rise until the heat is dissipated, surpassing design specifications and causing electronic device failure. Therefore, to keep all services and data in a secured manner, it is evident that one data center could have at least two working cooling devices unless it might go to downtime. It can handicap the standard functionality of the data center appliances when there is no early action done.

The data center must also guarantee both physical security and cybersecurity; There are measurements of safety such as automated doors through face recognition, password, and/or fingerprint. The use of Closed-Circuit Television (CCTV) cameras is necessary for protecting data from physical threats such as theft, destruction, and unauthorized access. These security measurements aim to guarantee the company to its clients the availability of their services by keeping the data center resilient (Courtemanche, 2015).

Often, a backup system is highly recommended for the data center to respond to all types of emergencies (Niemann *et al.*, 2013). In the case of a disaster or tragedy, a data center should continue operating and remain unaffected. In hindsight, to maintain company continuity in the case of a disaster, a strong emergency backup system should include fire detection and suppression, power generators, redundancy systems. Furthermore, there is a need of using a cooling device backup and a monitoring system of temperature and humidity since most of the servers are pretty sensitive to temperature and humidity (Li & Kandlikar, 2015; Woods, 2010).

To make their cooling infrastructure even more efficient, several data centers are adding advanced Artificial Intelligent systems. Static electricity may be created by dust accumulation, and when it discharges, it can harm electronic components. Dust or other debris can obstruct airflow, causing servers to generate more heat and strain the cooling system. One of the essential routine chores that a data center manager must undertake is keeping airways and the space surrounding servers clean (Sampera, 2019).

Nowadays internet of things (IoT) has improved the way data center cooling system works (Rahman *et al.,* 2020). It can help remote monitor temperature and humidity inside the room and regularly notify the data center managers about the actual temperature and humidity values (Saha & Majumdar, 2017). Internet of Things (Iot) technology has been enabled by the ubiquity of the Internet and cloud storage platforms instead of the traditional databases.

1.2 Statement of the Problem

Habari Node Public Limited Company (Habari Node PLC) is a Tanzanian Internet Service Provider with headquarters in Arusha. It offers basic Internet services as well as a variety of additional ICT-based business solutions. Habari Node PLC provides a variety of services such as hosting, cloud computing, tower space, and data center services. Also, the company provides services from small to large enterprises, national and international organizations, non-governmental organizations, and government institutions.

Regardless of the potential of Habari Node PLC as one of the leading service providers, its data centers cooling systems are done through old fashion Windows Air Conditioners (AC) (Fig. 7) and Split Air Conditioners (Fig. 8). These devices actively participate together in cooling the air inside the data center room. Unfortunately, the AC and its split have to operate all the time without stopping until they fail. Because of that, the cooling devices are likely to fail after three years of operation. Moreover, the cooling devices are probably misbehaving 2 to 3 times a month due to the electricity fluctuation and hardware failure; this is also caused by either the higher or lower voltage provided by the power source.

Because of the unpredictability of ability and the lack of a Data Historian, a data center manager must configure at least four cooling devices to function in parallel to assist each other when one fails. Cooling systems use a lot of energy, and one of the devices may consume a lot of energy each month. If there are a lot of them, they consume a lot of electricity. In case temperature value is below the threshold, the data center manager open door of the server room to let the outside hot air enter the room; which is not healthy for a server room since extern environment might contain dust and gases harmful to the servers.

To ensure that everything is good in the data center, the Information Technology (IT) manager needs to keep track of the real temperature in the room. The data center manager constantly enters the data center room to check whether the data center equipment is still working perfectly. Due to the cold, it is not suitable for human beings to stay inside the data center for a long time. A temperature monitoring device also sends the current temperature level daily to the data center manager through e-mail. If the temperature level reaches 24 ˚C, the data center manager is notified. The same notification is pushed when the temperature level in the room goes away from the range of 22 ˚C-25 ˚C. However, this is not enough for data security since it is not easy to respond quickly to put data center temperature at its normal range. Moreover, there is no dashboard to visualize the change in temperature over a certain time. To be often present in the data center room is not health-friendly because of the coldness.

To resolve these listed challenges of lacking automation in cooling systems, Internet of things-based temperature controlling system was designed and implemented. It would allow the IT manager to remotely control the temperature of the data center, and fix issues without a physical presence in the data center. The IT manager can visualize the temperature level and take action through cloud-based software and web User Interface. Using the software dashboard, the IT manager can switch a facility ON or OFF depending on the current monitored temperature. This system also contributes to the region's development, especially in the IT domain, by assisting data center managers in their responsibilities.

1.3 Rationale of the Study

Internet of things technology has significantly and positively impacted the lives of society in all sectors such as healthcare, security, education (Association, 2014). Nowadays, any information system should be enhanced by using an IoT solution. The more information systems there are, the larger and better the infrastructure for information storage is necessary. Data centers do not just need a normal cooling system to protect equipment against the heat, but also need to monitor and control the level of temperature so that the manager may take a quick decision before the excess of temperature value to keep the server environment at a good air condition. Data center managers need to remotely visualize the monitored value of temperature through a dashboard. Security of server room is always implemented since the failure of cooling devices may risk the destruction of the equipment inside a data server. Replacement and backup devices in a cooling system are among key factors to fight against the failure and then to keep the normal functionality of servers. Therefore, this study aimed to develop an IoT-based system for monitoring and controlling data center temperature to provide features such as; prevent downtime of servers, monitoring of temperature, and control of temperature, a cloud-based dashboard for data visualization.

1.4 Objectives

1.4.1 General Objective

The main objective is to develop an IoT-based system for monitoring and controlling data center temperature at Habari Node Company.

1.4.2 Specific Objectives

(i) To review and analyze the requirements for developing an IoT-based system for temperature monitoring and control in a data center.

(ii) To design and develop an IoT-based system for the data center temperature monitoring and control.

(iii) To validate the developed system.

1.5 Research Questions

(i) What are the requirements for developing an IoT-based system for monitoring and controlling the data center's temperature?

(ii) How to design and develop an IoT-based system for the data center temperature monitoring and control?

(iii) How to ensure the developed system meets the specified requirement?

1.6 Significance of the Study

The proposed system helps the data center manager remotely visualize the monitored temperature level in the data center from a remote area and gets an alert notification of the temperature value. It undoubtedly decreases the stress of the data center manager. In fact, through the microcontroller ESP WROOM 32 and cloud API, the sensory data are sent to the cloud. Then the data center manager can visualize and be notified in real-time. The data center manager can switch on and/or off the cooling devices remotely. It helps the data center manager avoid often entering the server room and then contribute to protecting his/her health again any inconvenience due to the coldness of the data center.

The cooling devices in the data center play a significant role in generating humidity and pulling temperature from inside the room (Dunlap & Rasmussen, 2006). Therefore, the use of backup devices significantly enhances the security of the data center equipment that may be caused by the downtime of one or more cooling devices. The auto-switching system of cooling devices contributes to the protection of the equipment. The remote switching of cooling devices contributes to the extension of their lifespan since they are likely to fail after three years of their constant operation.

Furthermore, when the company explains its core values to its customers, they get more confidence in the company. The solution also helps save money that would otherwise be spent on purchasing new tools or repairing broken ones since with the backup mechanism, the chances of the devices failing are low.

1.7 Delineation of the Study

This study aimed to develop an IoT-based system for monitoring and controlling data center temperature at Habari Node PLC company to provide features such as real-time temperature notification, equipment failure prevention, temperature tracking, a virtual remotely accessible dashboard to display recent and current temperature values, and remote and automated control system over cooling devices. As a result, the prototype incorporates a temperature monitoring system, customizable air cooling and notification systems, data storage system via a cloud-based platform, and a web server to provide a low-cost solution.

CHAPTER TWO

LITERATURE REVIEW

2.1 Heating, Ventilation, and Air Conditioning Systems

Heating, ventilation, and air conditioning (HVAC) is a critical part of facility mechanical systems that provide thermal comfort and indoor air quality to users (Seyam, 2018).

Heating, ventilation, and air conditioning (HVAC) systems are mostly used in indoor and outdoor places where there is a necessity of maintaining the environment at a certain temperature to protect equipment and/or people from freezing or excessively high temperatures. HVAC systems are mostly applied in air conditioning server rooms (Fig. 1), telecom shelters, laboratories, and many other IT infrastructures.

Figure 1: Real application cluster (RAC) of server room

2.1.1 Heating, Ventilation, and Air Conditioning Components

Heating, ventilation, and air conditioning (HVAC) systems are mainly composed of three units as heating, ventilation, and air conditioning units.

(i) Heating Unit

Common heating systems, which are used to provide warmth in interior areas, including boilers, furnaces, and heat pumps. In most situations, hot water, steam, or airflow is used to disperse heat throughout the structure. Heat is often dispersed throughout the structure using radiators that are fixed on the walls or set in the floor in the case of water and steam.

The Heating Unit can intervene when the HVAC system is intended to be used in a location where the temperature is significantly low.

(ii) Ventilation Unit

The practice of removing and replacing the air inside a place to achieve good indoor air quality is known as ventilation. This includes, but is not limited to, the replenishment of oxygen as well as the removal of moisture, smells, smoke, dust, and carbon dioxide. Extractor fans and recirculation fans are examples of mechanical ventilation systems; movable windows, louvers, and trickling vents are examples of passive ventilation methods.

(iii) Air Conditioning Unit

An air conditioning unit cools and controls the humidity in a structure. In most situations, outside air is pulled inside the structure, where heat is dissipated using cooling refrigerants. Dehumidification, commonly known as "air drying," happens during the cooling process.

2.1.2 Common Architecture of Cooling Systems

Because cooling system infrastructure must execute continuous cooling operations, a variable temperature range map is required. To best match the optimal air condition of the room, cooling device installation principles must be strictly followed to the server orientation.

(i) Cooling system with concealed ceiling units

A Cooling system with concealed ceiling units participates in cooling the air inside any enclosed place by accumulating the heat from between server ranks. The accumulated heat dissipates through the concealed ceiling unit (Fig. 2). Then it goes through the tunnel that congest it to be pushed again inside the server room to the other side of the server racks. The congested air is shared through the opposite walls.

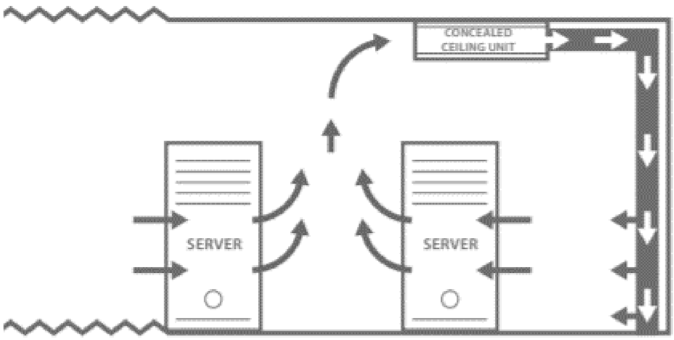


Figure 2: Cooling system with concealed ceiling units

(ii) Cooling system with Hot-isle and Cold-isle set-up

By gathering heat from between server ranks, a Cooling system with Hot-isle and Cold-isle set-up helps to chill the air within any confined space. As seen in Fig. 3, the collected heat is dissipated through the concealed ceiling unit. It then passes via a congested tunnel before being pushed back into the server room to the other side of the server racks. In this case, congested air passes through the underground tunnel and then is dissipated to the server rank to the opposite sides.

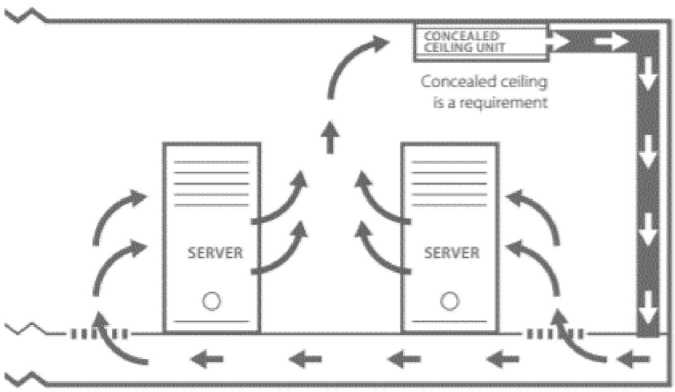


Figure 3: Cooling system with hot-isle and cold-isle set-up

(iii) Cooling system with concealed ceiling units

By gathering heat from the top of server ranks, a cooling system with concealed ceiling units helps to chill the air within an enclosed room. As seen in Fig. 4, the collected heat is dissipated through the disguised ceiling unit. It then passes via a congested tunnel before being pushed back into the server room to the bottom of the server racks.

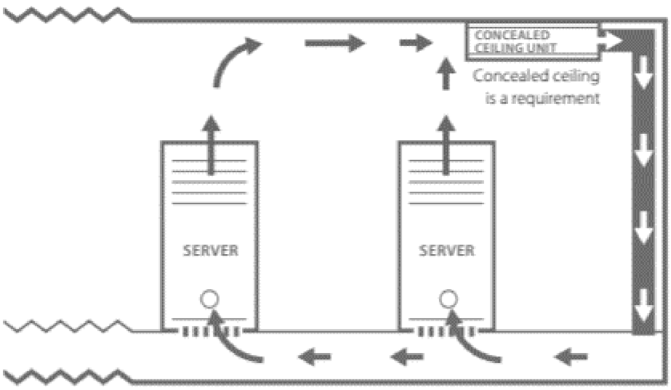


Figure 4: Cooling system with concealed ceiling units

(iv) Cooling system with ceiling suspended units

These units are built by ceiling suspended units that are hung to the roof of the server room as opposed to each other so that they can provide the side to pull heat from the server rack and dissipate the congested air on the other side of the server racks (Fig. 5).

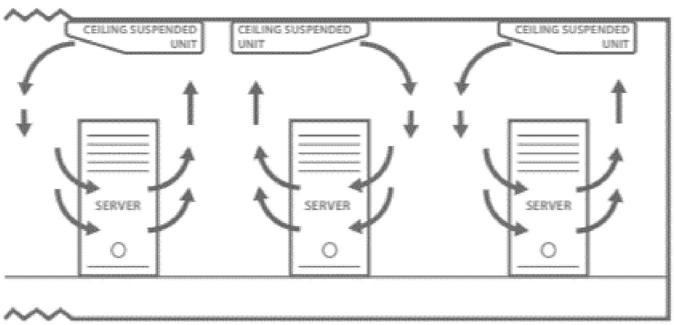


Figure 5: Cooling system with ceiling suspended units

(v) Cooling system with wall mounted units

These units are built by wall-mounted units that are hung to the roof and the wall of the server room. They are placed in the way that they pull heat from the server rack in the same direction and dissipate the congested air on the other side of the server racks (Fig. 6).

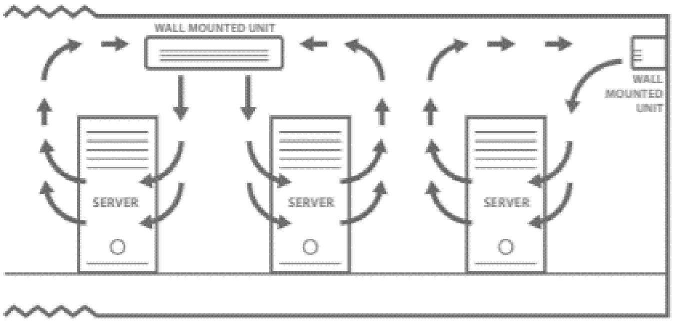


Figure 6: Cooling system with wall mounted units

Cooling devices in a cooling system depend on the size of the data center and how servers are placed inside the server room. A data center manager must apply a corresponding Cooling System to the data center.

2.1.3 Common HVAC System Types

Window and split AC are commonly HVAC types since they are typically less costly and cheaper to operate.

A window air conditioner (Fig. 7) is a single unit with all of its components enclosed inside. It ejects heat out of its outdoor side and blows cool air into the room on the indoor side. As the name suggests, it is installed in a window or by making a hole in the wall. Such air conditioners have a filter that slides out so that they can be cleaned regularly for full AC efficiency. These air conditioners have controls on the unit & may also come with a remote.

Figure 7: Window air conditioner

A split air conditioner consists of an outdoor unit and an indoor unit. The outdoor unit is installed on or near the exterior wall of the room that you wish to cool. This unit houses the compressor, condenser coil, and expansion coil or capillary tubing. The sleek-looking indoor unit contains a cooling coil, a long blower, and an air filter.

Figure 8: A split air conditioner

A type of HVAC system used at Habari Node is one of the local HVAC systems; Habari Node Data center uses an air conditioning system that involves Window Air Conditioner and Split AC to maintain the temperature of the normal level (from 22 ˚C to 25 ˚C).

2.1.4 Disadvantages of Traditional HVAC systems

The most inevitable issue with HVACs is that they use a significant amount of energy to operate. When they are running, they use a lot of electricity. The reason for the requirement for a control system is to limit the number of devices that run at the same time to save money on energy usage (Controllers & About, 2021).

It is difficult to do dependable air conditioner maintenance without a monitoring system of how they work. Either change the cooling unit that is still operational or wait till they fail. A monitoring system is required to detect misbehavior or failure of cooling systems and then to fight against Inadequate maintenance and unexpected failures.

It is not feasible to access or configure it remotely; thus, it is advised to set up a system that allows for remote access and configuration of cooling devices via an IoT system.

2.1.5 Smart Air Conditioner

A smart air conditioner is nothing but a mini-split, window, or portable air conditioner smart improved by an IoT system; A smart AC can be remotely controlled via a smartphone app which gives it a few to save energy.

However, Smart Air Conditioners are not commonly used since they are more expensive than other types of air conditioners and their features depend on the availability of internet connection. Furthermore, they are not easily maintainable in case of need.

Therefore, they can be replaced by the common air conditioners that are equipped with IoT monitoring and regulating systems; these features aid to keep track of temperature and provide an interface to visualize the monitored value. An actuation unity such as warning unit, notification unit, command unit can eventually be applied to enhance the features of IoT-based system for air conditioning.

2.2 Temperature Monitoring Systems

Traditional temperature monitoring systems operate almost in the same way as IoT-based temperature monitoring systems. They can sense data from the environment and send them to the intended email and they can be programmed to send a notification when the temperature data has reached a certain value. The sensor probe2 (Fig. 9) is widely used in temperature monitoring systems even though it has some limitations such as a poor notification system, lack of storage and dashboard.

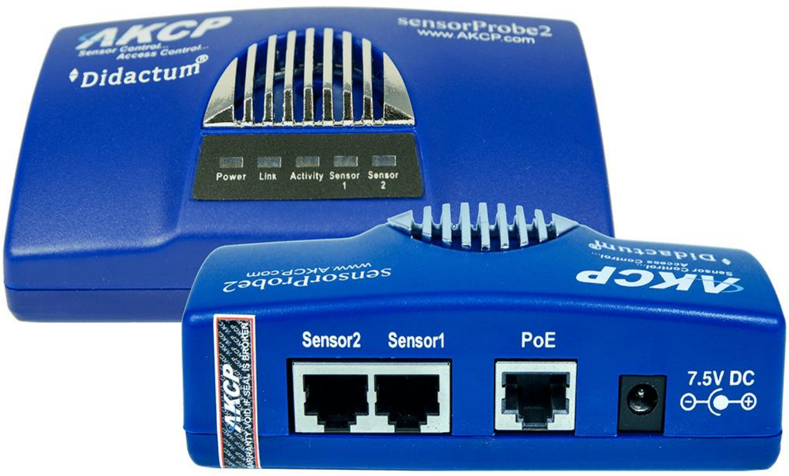


Figure 9: Temperature sensor Probe2

Table 1 shows a list of famous temperature monitoring systems and devices such as AKCP sensorProbe2, Thermo-Humidity Logger ZN-THX11-SA, Type K Thermocouple and IoT System equipped by temperature sensor.

Table 1: Temperature monitoring systems

Name	**Sensed Data**	**Range**	**Accuracy**	**Notification**
AKCP sensorProbe2	Temperature and Humidity	Temp: Min. -35°C - Max.80°C Humidity: Min. 20% - Max. 80%	-	2 Email addresses
Thermo-Humidity Logger	Temperature and Humidity	-25 to 60^{o}C	-	Alarm output

ZN-THX11-SA				
Type K Thermocouple	Temperature	–270 to 1260ºC	Standard: +/- 2.2⁰C or +/- .75%	-
IoT System (DHT22 Sensor)	Temperature and Humidity	Temperature -40 to 80°C humidity 0-100%	Humidity 2-5% Temperature ±0.5°C	Alarm Output SMS mails

2.3 Internet of Things Based Temperature Monitoring System Applied in HVAC Systems

Traditional HVAC systems are not efficient and are likely to fail; they don't have a dashboard to display detected temperature and can't keep a record of recent temperature. Therefore, it is impossible to predict how temperature would reach a certain level in near future due to the lack of records (Controllers & About, 2021).

A temperature monitoring and control system using IoT is a system that includes a temperature detector, an actuator to process the detected temperature value, and storage to store and retrieve the monitored temperature value; the storage can be a local server or a cloud-based platform. IoT-based temperature monitoring systems are more precise and have many additional benefits such as presenting temperature values in real-time, being economical, and being easy to maintain. Cooling units can be activated or deactivated using actuators based on the most recent and current value of the monitored temperature.

As a result, there are several advantages for adopting IoT in HVAC systems:

(i) Expanding HVAC equipment lifetime by limiting the total number of devices that work together.

(ii) Minimizing the amount of money being spent as a result of increased electricity usage.

(iii) Reducing Unexpected HVAC Failures.

(iv) Having a central dashboard to visualize temperature variations.

(v) Providing remote access to and control over the cooling equipment over the Internet.

Figure 10 shows the detail of components of modern HVAC.

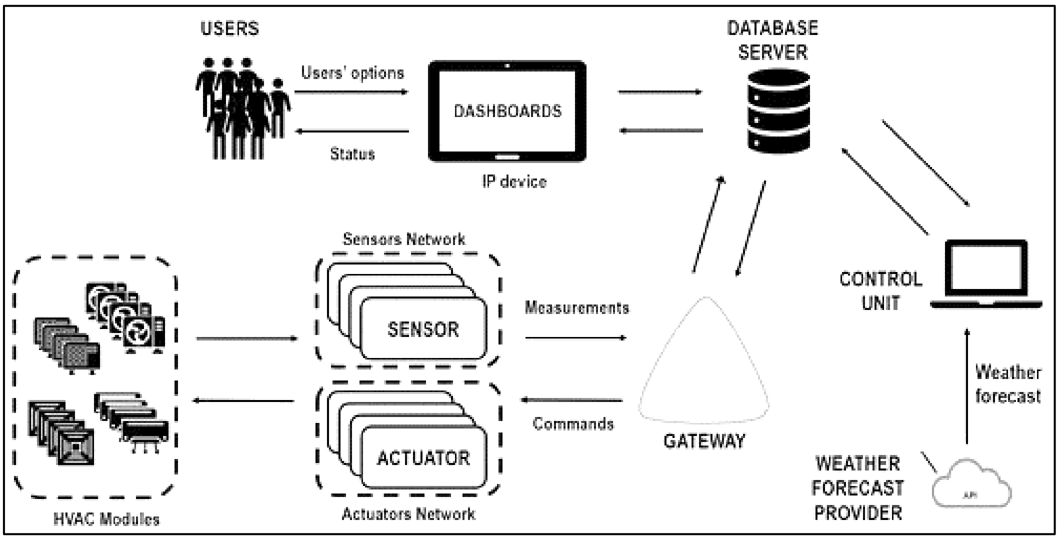


Figure 10: Modern HVAC architecture

2.4 Definition of Terms of IoT System and Data Center

(i) Cloud Storage API

An Application Program Interface that links a local application to a cloud-based storage system, allowing a user to input data and later edit it.

(ii) Computer Room Air Conditioning Unit

A computer room air conditioning unit (CRAC) is a device that monitors and controls the temperature, air distribution, and humidity levels in a network room or data center.

(iii) Computer Room Air Handler

The computer room air handler (CRAH) is a device that is commonly used in data centers to cope with the heat generated by the equipment.

(iv) Computerized Maintenance Management System

A computerized maintenance management system (CMMS) is a piece of software that assists operations and maintenance personnel in identifying and tracking the progress of maintenance jobs as well as the availability of replacement components.

(v) Critical Infrastructure

Critical infrastructure is a critical infrastructure is a collection of systems, networks, and assets that are so vital that their ongoing functioning is necessary to maintain the security of a particular nation, its economy, and the health and/or safety of the general public.

(vi) Data Center Resiliency

Data center resiliency is the capacity of a server, network, storage system, or entire data center to continue running despite equipment failure, power loss, or other disturbance.

(vii) Data Historian

Data historian is a software application that records the data generated by computer activities.

(viii) Immutable Infrastructure

Immutable infrastructure is a method of managing IT services and software deployments in which components are replaced rather than changed.

(ix) Intelligent Power Management

An intelligent power management (IPM) is a hardware and software combination that optimizes the distribution and usage of electrical power in computer systems and data centers.

2.5 Related Works

Different researches have been carried to tackle the issue of temperature monitoring systems in a critical environment where the temperature is the main parameter to consider for an optimal environment such as a data center. The following are the most relevant researches in that field.

Rahman *et al.* (2020) developed a system to monitor the temperature and humidity of a data center. The work intended to determine the relationship and difference between temperature and humidity. This system had been implemented in different locations of temperature and humidity measurements at the data center of Politeknik Muadzam Shah. The system based on the IoT platform was able to detect extreme changes in temperature and humidity and automatically send a notification to IT personnel via e-mail, Short Messaging Service (SMS), and a mobile device could push notifications for further action.

Medina-Santiago *et al.* (2020) developed a system that was able to monitor, in real-time, the temperature and humidity in the data center located in the TECNM-Villahermosa, south-southeast of Mexico. Through IoT systems, Graphs, and the analysis of the data, the system was able to check whether the temperature and humidity levels inside the data center lie in normal range according to the International Computer Room Experts Association (ICREA) and the American Society of Heating.

The simple all-in-one system for monitoring the data center environment and the network was designed to enhance traditional data environment monitoring systems. The detected and monitored variables were temperature, humidity, location, motion, smoke, water, and voltage variation. Besides that, there was a system for notification through e-mail and mobile networks. Data visualization was used to display the data and handle the user requests (Ramphela *et al.*, 2020).

A system of monitoring the temperature inside a laboratory rearing unit that experiments with the life of insects impacted by temperature variation was developed. The system regularly and automatically does backups of sensory data in both local and online server storages based on raspberry pi. The system also sent an alarm and displayed a dashboard report of all rearing units in an HTML file. The system was chip and easy to implement in extensive laboratories (Rebaudo & Benoist, 2019). The systems for monitoring temperature are critical, especially in such an environment where life mainly depends on the temperature.

Saha and Majumdar (2017) implemented a Wireless Sensor Network (WSN) based monitoring system to monitor temperature and humidity. This system was based on ESP8266, Wi-Fi Rooter, temperature and humidity sensors. The prototype could sense both temperature and humidity, but only the temperature is taken into consideration. The system had a Real-time online dashboard to display the monitored temperature and an interface to define SMS and e-mail alerts on specific sensory values. It was possible due to cloud configuration for both sensors and e-mail alerts. The ESP8266 board is connected to the Wi-Fi router using an SSID code and password. It allowed the system to establish a connection with the cloud platform using HTTP Protocol effectively.

Sari *et al.* (2019) built a system capable of collecting temperature data using the DS1621 sensor. The system would communicate temperature data to the administrator through an

SMS gateway to alert the data center manager if the server reached its upper limit temperature, allowing to take immediate action to remedy the issue.

Two years later, the solution was improved by monitoring both temperature and electricity; the program would then transmit an order to automatically shut down the server. This would improve server upkeep and performance, particularly when the administrator or data center operator is not available (Sari *et al.*, 2021).

A system to remotely monitor the temperature and humidity of the server room was designed using Lattepanda and ThingSpeak. The study's goal was to create a versatile and simple-to-use hardware and software solution that would assist data center managers in avoiding the chilly impact of constantly entering the server room to verify temperature and humidity levels (Nasution *et al.*, 2019).

Alvan *et al.* (2019) created a system that monitors temperature and humidity and sends notifications and commands via the Telegram app. The system used a raspberry pi as the sensing module's mainboard and an Arduino as the actuation module's mainboard; if a temperature value falls below 18 °C or rises above 27 °C, the system instructs Arduino to reduce or raise the temperature of the Air Conditioner and sends a notification to Telegram. The system seeks to offer an easy approach to manage the ideal temperature of a saver room to prevent critical room hardware from damage caused by excessive temperatures and also a waste of power caused by too low temperatures.

An IoT-based system was developed to monitor the temperature and humidity of indoor rooms, particularly data centers; the system aimed to gain a better understanding of temperature and humidity and has significantly reduced the hassle of constructing complex storage to capture and analyze temperature humidity data. The system can send temperature and humidity measurements every minute and broadcast a warning message to Telegram if the temperature is not between 18 and 27 degrees Celsius. The alert will also be activated if the humidity level is less than 40% and more than 55% (Shamang *et al.*, 2020).

To sum up all of those works, the temperature monitoring systems have been done to enhance the safety and security in the environment where the temperature is the main parameter. However, there is no powerful automation to regulate the temperature in case a suitable range is passed by. It is critical to have an automated system for monitoring and adjusting the temperature. In reality, there is a requirement for full control over cooling equipment; by automatically receiving temperature values as input, the system may then command the devices to turn on or off based on the temperature data obtained.

CHAPTER THREE

MATERIALS AND METHODS

3.1 Methodology

The project must have well-defined methodologies to be successful. From a broad grasp of the project to a thorough comprehension of the project, feedback from others who worked on similar projects must be taken. Before being provided to the final user, the prototype should be thoroughly tested.

For continuous iteration of development and testing of changeable prototypes, the agile development approach is used. Agile methodology is iterative and incremental development with requirements that may be altered as needed in response to customer requests *(Sharma et al.*, 2012). This project qualifies to apply this method since there is a continuous improvement at each system level (Fig. 11). It also adjusts quickly to modifications and gives more control over the project. Finally, it raises the visibility of project performance (Fireteanu, 2020).

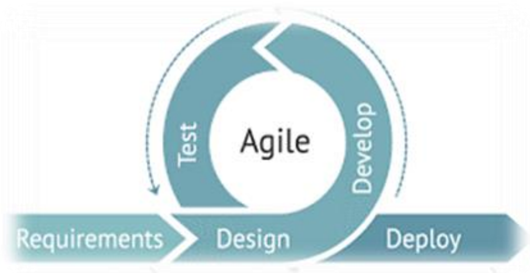


Figure 11: Agile system development

3.2 Study Area and Scope of the project

The project was carried out at Habari Node PLC in Arusha, Tanzania (Appendix 1), and focused on data center air cooling technologies. Data collection from the present air conditioning system and a literature study of relevant works were done to develop the new air conditioning system. As a result, having a remote temperature monitoring and auto-switching cooling system for detecting the temperature level in the data center aids in monitoring and controlling the current temperature in real-time. Moreover, if the data center temperature exceeds a certain threshold, either the system or the data center manager takes immediate action.

3.3 Data Collection

The project involves research qualitative approaches; in fact, data collection was done through interviews (Appendix 2) and roundtable discussions with the data center manager at Habari Node Company. This approach leads to improving the existing system and developing a friendly-user system for the Data center manager.

Therefore, gathering information about the data center, temperature monitoring systems, and cooling system help name all of the excellent software and hardware required for developing a Temperature monitoring and controlling system. Also, secondary data collection from literature review on various electronic components and manufacturer datasheets was used to understand all requirements for creating a temperature monitoring and controlling system. Thus, all supporting materials and tools for completing the project were obtained by analyzing the collected information.

3.4 System Requirements

The development and implementation of the prototype require both hardware and software requirements. It means that the system needs specific hardware components and other software resources.

3.4.1 Hardware Requirements

The hardware tools include selecting appropriate hardware equipment for prototype development, computer, a mobile device and other materials.

The proposed system requires temperature monitoring, automated and remote-control cooling, and a notification system. Therefore, the prototype requires Microcontroller ESP-WROOM32, DHT22 temperature sensor, Electrical Programmable relays, Fans that stand as ACs, 12Volts Solar Battery, Mobile phone, resistors, jumper wires, breadboard, and a rooter.

During the system development, a laptop computer was used to power and program the ESP-WROOM32 board. A mobile device and rooter were used as hotspots to share the internet connection (Wi-Fi) with the ESP WROOM board. Thus, ESP32 sends temperature sensory data to the cloud to enable data visualization and processing. The mobile device also facilitates remote data visualization and cooling devices control respectively through cloud-based mobile applications and ESP web applications. Following are the main components for prototyping the project.

(i) Microcontroller ESP-WROOM32

The Microcontroller ESP-WROOM32, or simply known as ESP32 chip, can perform as a complete standalone system. The chip includes integrated Wi-Fi, Bluetooth, and other communication interfaces such as SPI and I2C/UART (Folty´nek *et al.,* 2019). There are some advantages of this microcontroller over others; the chip size, the number of pins, embedded Wi-Fi.

The board version used to build the prototype on a breadboard is ESP32 DOIT DEVKIT V1 and the model of 30 pins (Fig. 12).

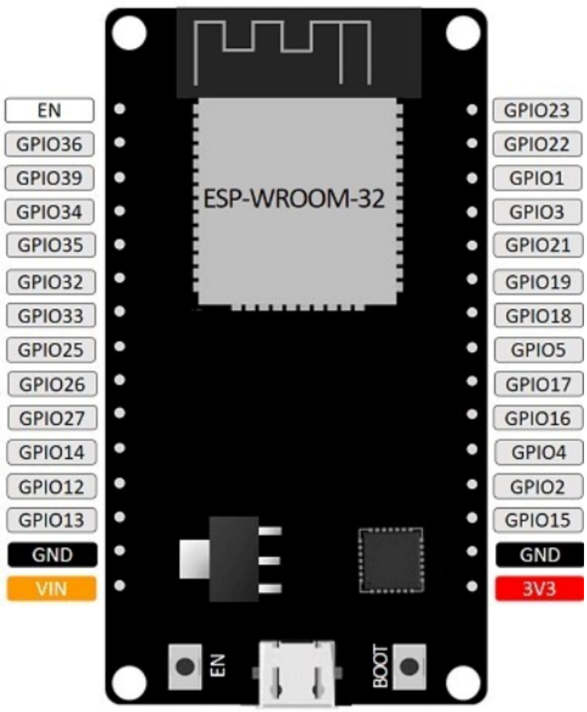


Figure 12: ESP32 DOIT DEVKIT V1 Board version with 30 GPIOs Pinouts

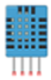
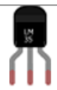
Table 2: ESP32 DEVKIT V1 DOIT specifications

Number of cores	2 (dual core)
Wi-Fi	2.4 GHz up to 150 Mbits/s
Bluetooth	BLE (Bluetooth Low Energy) and legacy Bluetooth
Architecture	32 bits
Clock frequency	Up to 240 MHz
RAM	512 KB
Pins	30 or 36 (depends on the model)
Peripherals	Capacitive touch, ADC (Analog to Digital Converter), DAC (Digital to Analog Converter), I2C (Inter-Integrated Circuit), UART (Universal Asynchronous Receiver/Transmitter), CAN 2.0 (Controller Area Network), SPI (Serial Peripheral Interface), I2S (Integrated Inter-IC Sound), RMI (Reduced Media-Independent Interface), PWM (Pulse Width Modulation), and more.

(ii) Digital Humidity and Temperature Sensor

Temperature sensors that work with Arduino, ESP32, ESP8266, and other development boards are many (Santos, 2016). In Table 3, there is a comparison between 6 widely used temperature sensors such as DHT11, DHT22, LM35, DS18B20, BME280, and BMP180 (Santos, 2019).

Table 3: Temperature sensors comparison table

						
Sensor	**DHT11**	**DHT22 (AM2302)**	**LM35**	**DS18B20**	**BME280**	**BMP180**
Measures	Temperature Humidity	Temperature Humidity	Temperature	Temperature	Temperature Humidity Pressure	Temperature Pressure
Communication protocol	One-wire	One-wire	Analog	One-wire	I2C SPI	I2C
Supply voltage	3 to 5.5V DC	3 to 6V DC	4 to 30 V DC	3 to 5.5V DC	1.7 to 3.6V (for the chip) 3.3 to 5V for the board	1.8 to 3.6V (for the chip) 3.3 to 5V for the board
Temperature range	0 to 50ºC	-40 to 80ºC	-55 to 150ºC	-55 to 125ºC	-40 to 85ºC	0 to 65ºC
Accuracy	+/- 2ºC (at 0 to 50ºC)	+/- 0.5ºC (at -40 to 80ºC)	+/-0.5ºC (at 25ºC)	+/-0.5ºC (at -10 to 85ºC)	+/-0.5ºC (at 25ºC)	+/-0.5ºC (at 25ºC)
Support (Arduino IDE)	Adafruit DHT Library Adafruit Unified Sensor Library	Adafruit DHT Library Adafruit Unified Sensor Library	analogRead()	DallasTemperature OneWire	Adafruit BME280 library Adafruit Unified Sensor Library	Adafruit BME085 Adafruit Unified Sensor Library

The digital humidity temperature (DHT22) is a more precise and efficient temperature sensor than other sensors (Table 3). This sensor has a 0-100% humidity reading with a 2-5 percent accuracy and can measure temperatures ranging from -40 to 80 degrees Celsius. (Alvan *et al.*, 2019; Lady-ada, 2020). According to the properties and accuracy, DHT22 is the best temperature sensor to be used in this project. According to the best performance of DHT22 sensor and how big the data center room is, one DHT22 sensor was used to get the accurate temperature data to save to the cloud.

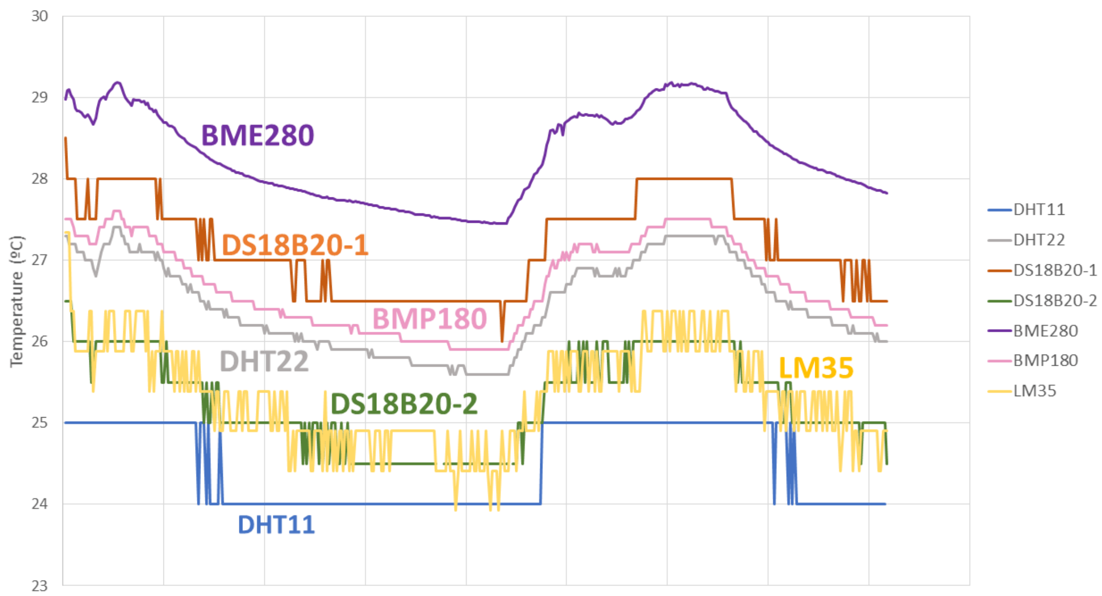


Figure 13: Testing temperature sensors charts (https://randomnerdtutorials.com)

(iii) Electrical Relays

Relays open and close connections in another circuit to regulate one electrical circuit. Relays are programmable switches that interfere with the circuit to make it interactive for users who can control them remotely. They use electromechanical or electrical means to open and shut circuits.

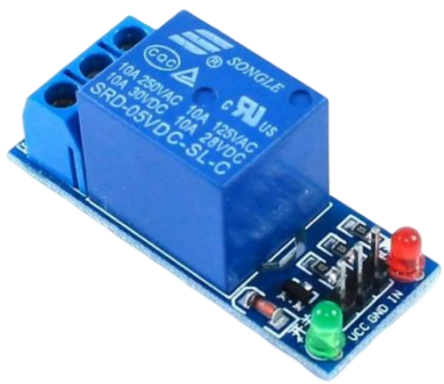


Figure 14: Single channel relay module

3.4.2 Software Requirements

The software requirements for the proposed system are Fritzing for the designing of the temperature monitoring and control system, Arduino IDE with all of the required libraries for Microcontroller ESP-WROOM32 programming, the cloud platforms (software and APIs) to facilitate data storage, visualization, Sublime text to edit HTML script to perform a remote control of the relays and PHP script to send Email and SMS notifications (Appendix 6).

ThingSpeak is the cloud plate-form that was used to store sensory data. ESP webserver library was used to host a web page to remote control the relays connected to the cooling devices. ThingSpeak mobile app (Thing View Free) was used for visualization of temperature data in real-time.

3.5 System Design

3.5.1 System Overview

The proposed system design comprises four parts; the first part is the temperature monitoring system; a sensing system enables this through DHT22 sensor and Microcontroller ESP-WROOM32. The sensory data is sent and stored in the cloud. The second part of the proposed system is the data processing that uses Microcontroller ESP-WROOM32 to process the detected temperature to command the relays in case the sensed temperature is out of the normal range. The third part of the data visualization is done by the mobile application. And the last part is the remote control of the relays, which is done through the mobile application to send the signal to the Microcontroller ESP-WROOM32.

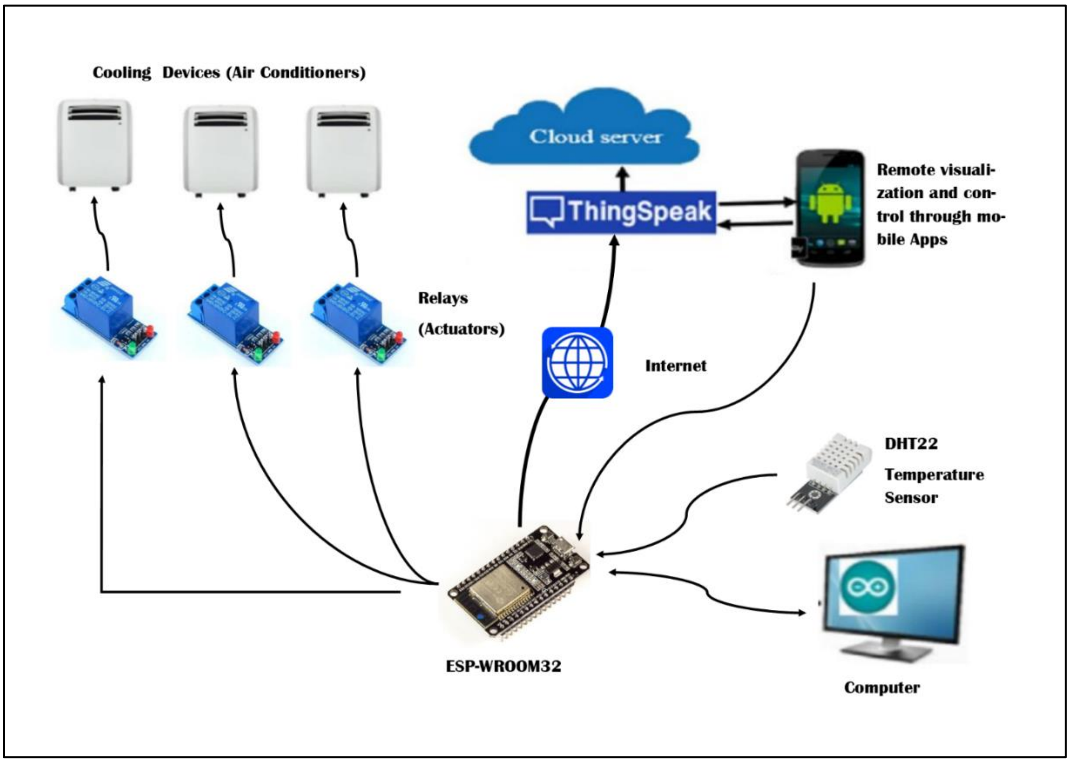


Figure 15: Prototype system architecture

As evident in the Fig. 15, the prototype has an ESP-WROOM32 board as a central board to which different components are connected. A DHT22 sensor is used to detect the temperature in the environment effectively. Three relays and three 12V-Fans are used to stand for the cooling system. 12V- battery is used to power the circuit. Finally, the prototype needs cloud storage and a mobile phone to access the stored data over the Internet from a remote location. The mobile phone also is used to remotely switch on and off the relays so that the user comes to control the cooling system. This scenario is summarized in the Fig. 16.

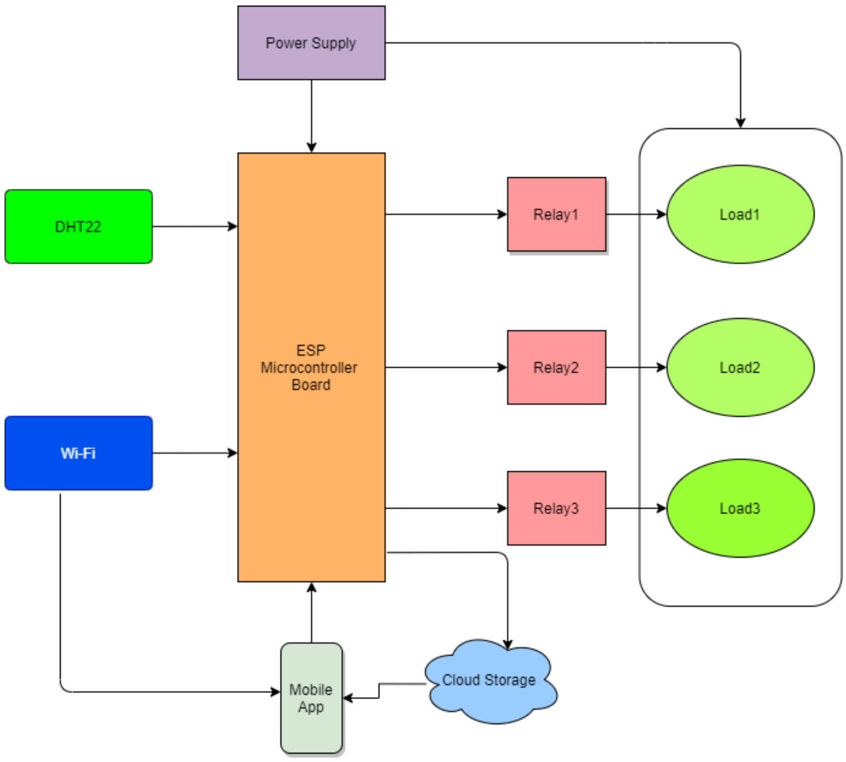


Figure 16: Block diagram of the system

3.5.2 System Assembling

All parts of the system once assembled must work regarding Fig. 17; a temperature monitoring system begins the operation and then the cooling and notification system reacts according to the sensed data. The sensed data is always sent to the cloud for visualization and analysis.

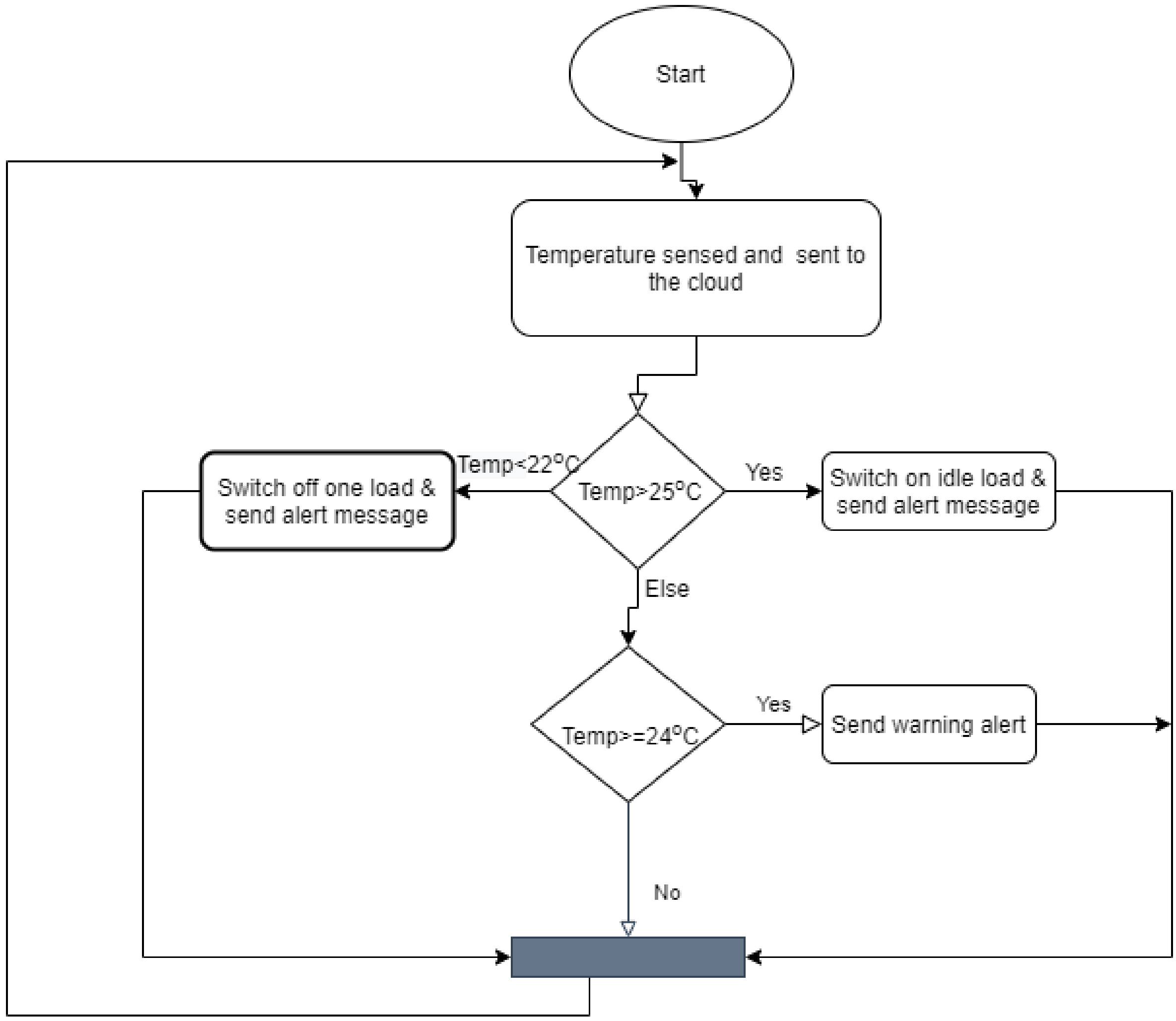


Figure 17: The flowchart diagram of the system

3.5.3 Temperature Monitoring System

The proposed Temperature Monitoring system was developed by using the DHT22 temperature and humidity sensor. The current detected temperature value is sent to the cloud database to be remotely visualized. The e-mail and SMS notifications from the cloud platform are regularly done pushed by the webserver script (Fig. 18).

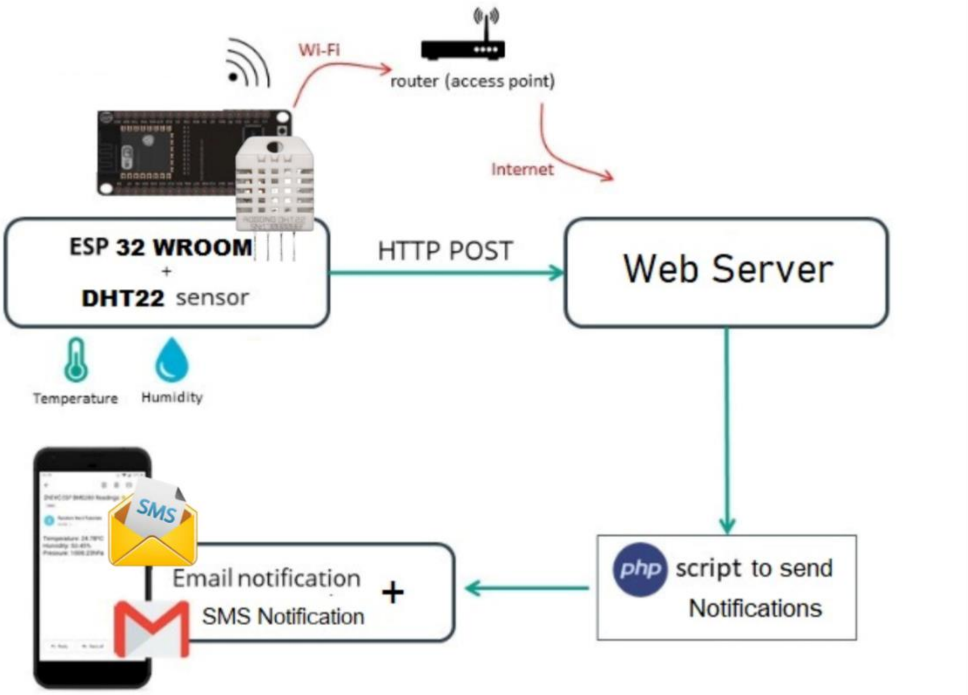


Figure 18: Temperature monitoring and notification system

The temperature sensed is also sent to the cloud ThingSpeak (Appendix 4) over the internet by the means of the cloud API (Fig. 19).

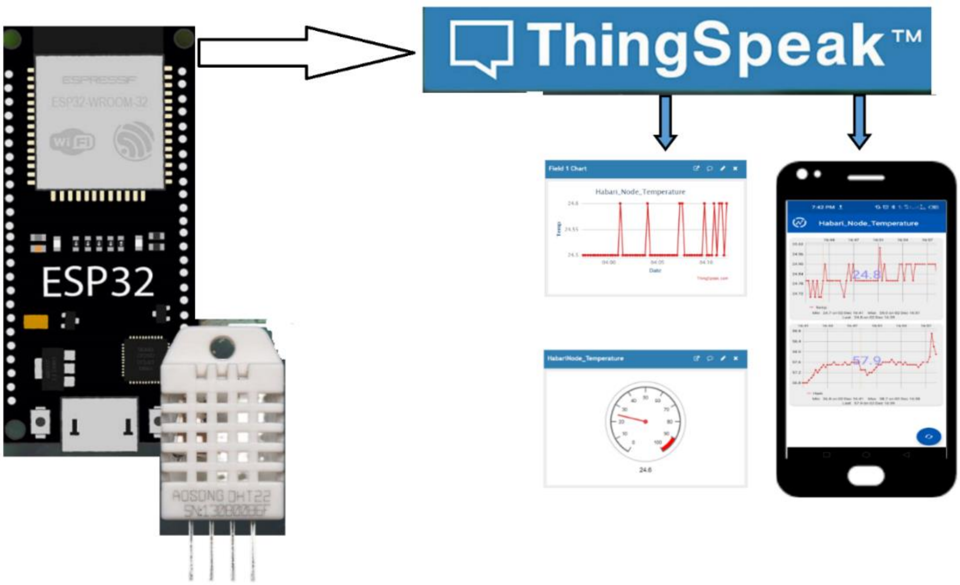


Figure 19: Temperature sensing and monitoring system

3.5.4 Automated cooling System

Backup cooling devices were used to intervene if the current operating cooling devices cannot maintain the normal range of temperature. However, when the temperature level returns to its normal range, one cooling device is switched off automatically through the signal sent from the microcontroller ESP-WROOM32. The auto-switching system (Appendix 5) for cooling devices is done through actuation triggered by either the failure of one of the devices or the signal sent from the microcontroller.

Here is the sketch to auto-switch the relays when the temperature is out of the normal range.

```
    // call autoSwitch function to operate when temp >25 or Temp<22

    if(t >25.0 || t < 22.0){

    autoSwitch(t,23,22,21);

    }
```

The function autoSwitch() only operates when the temperature t is above 25 ˚C or below 22 ˚C to automatically switch ON or OFF the cooling devices controlled by the relays connected to ESP pin 21, 22, and 23.

Figure 20 displays the scenario of the backup and auto-switch system.

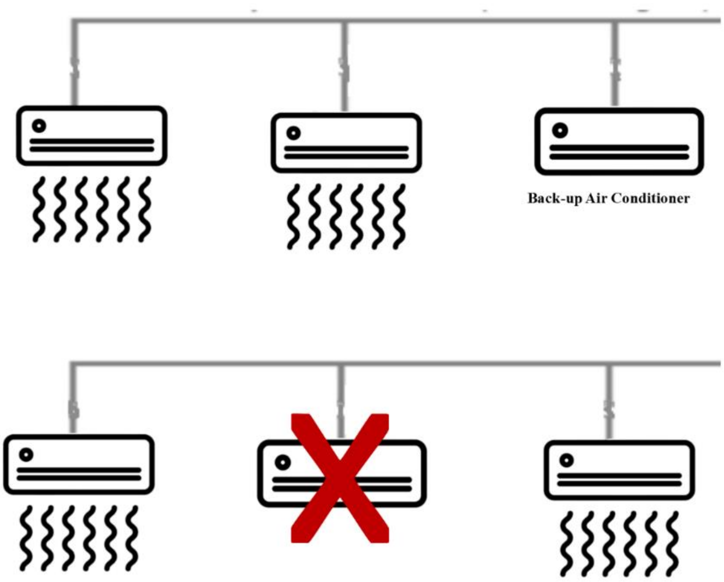


Figure 20: Cooling unit backup operation

When a unit fails, a security mechanism is triggered in which the standby backup unit takes over and an alert message is automatically transmitted to the data center manager.

3.5.5 Remote Control of Cooling Devices

The remote control of cooling devices works as an automated cooling system; the only difference is that the signal to switch on or off the relays is done manually through a mobile application. It is to assist the automated system in choosing a cooling device that can or cannot work. Furthermore, it helps the data center to over controlling the cooling.

To access an ESP32 webserver from outside its local network for controlling devices remotely involve getting your ISP to provide your router a static IP address, then configuring port forwarding or tunneling to send traffic on a certain port to the ESP32's IP.

(i) Setting ESP Static IP Address

To set a static IP address for the ESP board involves configuring the rooter to always assign the same IP address when it is requested. The requirement to configure the rooter for Static IP Address is the IP Address within the range of DHCP port reservation and MAC address of the ESP board. A static IP address is picked up from the range of IP addresses that can be assigned to clients connected to the rooter and the MAC address of the ESP board can be got from the output of the Arduino sketch.

```
// Print ESP32 Local IP Address
Serial.println("ESP connected to WiFi with IP Address : ");
Serial.println(WiFi.localIP());

// Get ESP MAC Address
Serial.print("ESP Board MAC Address:  ");
Serial.println(WiFi.macAddress());
```

Figure 21: Arduino sketch to get ESP board IP and MAC Addresses

```
ESP connected to WiFi with IP Address :
192.168.0.103
ESP Board MAC Address:  08:3A:F2:AB:65:80
```

Figure 22: Output of Arduino sketch to get IP and MAC Addresses

The IP address 192.168.0.103 and MAC address 08:3A:F2:AB:65:80 are then used to configure the IP address of the rooter as shown in Figure 23.

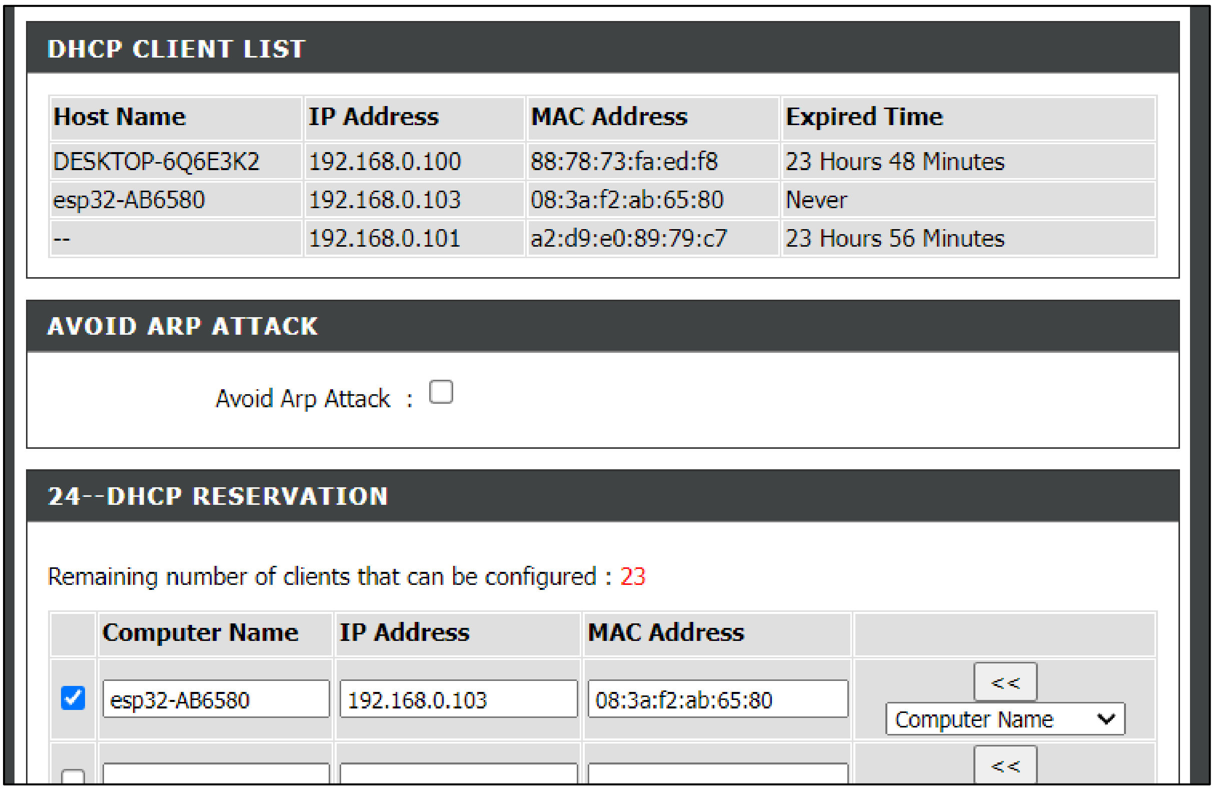


Figure 23: Static IP address reservation for ESP Board on D-Link Rooter

When an IP address is reserved for an ESP board, it will never neither expire nor be given to any other device. Then the ESP32 board will always get 192.168.0.103 as its IP address from the same rooter.

(ii) Port Tunneling

The IP address 192.168.0.103 is used to access the ESP webserver from the LAN (from any device connected to the same rooter) by typing it into the device's browser. When accessing the ESP web server from outside the LAN, however, port forwarding or tunneling must be set.

Because port tunneling is more secure than port forward, it was used in the prototype. Several systems can do port tunneling, but the prototype was built using a free version of Ngrok, which exposes local networked services behind NATs and firewalls to the public internet through a secure tunnel.

The Tunnel Auth-token from the Ngrok platform, the static IP Address, and the port via which the ESP's website is accessible are the criteria for configuring port tunneling. The following instruction is run in the CMD window of the Ngrok to get a public URL (Fig. 24).

```
ngrok tcp IP_ADDRESS:8888 --authtoken TUNNEL_AUTHTOKEN
```

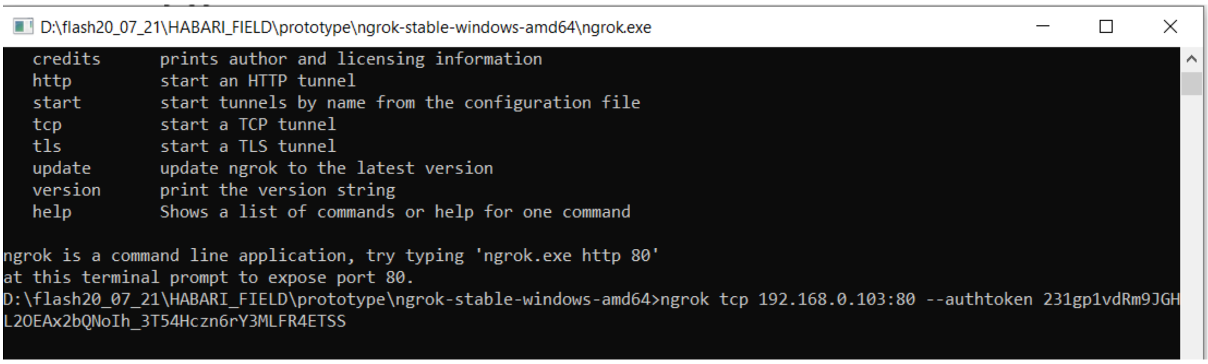


Figure 24: Port tunneling configuration in Ngrok window

When the instruction of the Figure 24, was run, it generated a URL that can be used to reach the ESP web server from any device that is connected to the internet.

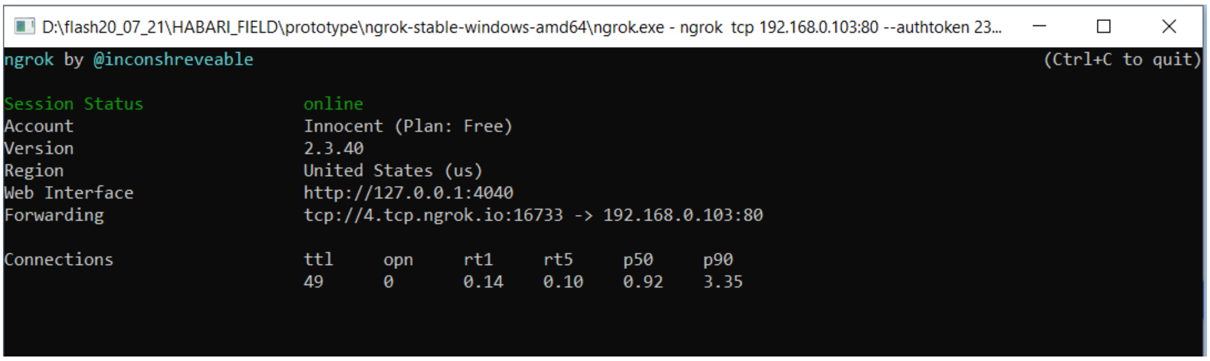


Figure 25: Port tunneling trough Ngrok

In Fig. 25, http://4.tcp.ngrok.io:16733 is a public URL that helps access the ESP webserver from any device connected to the internet regarding Figure 26.

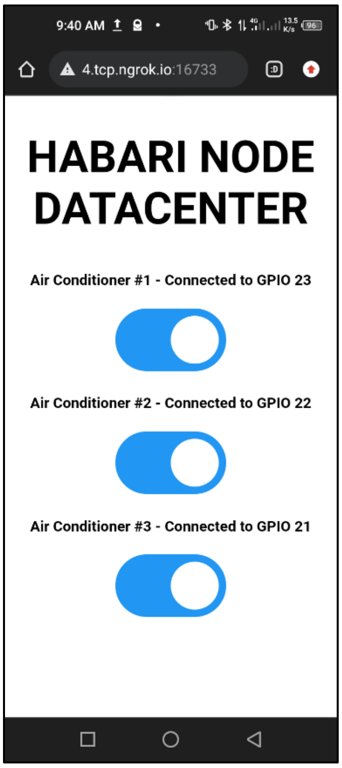


Figure 26: ESP accessed through Ngrok URL

3.6 System Implementation

An Arduino IDE sketch was written in embedded C programming language program the temperature monitoring system and activate and deactivate relays using an ESP-WROOM32 microcontroller board. The cloud APIs facilitate the communication between the ESP-WROOM32 microcontroller board and the cloud for data storage.

3.6.1 Connecting DHT22 to ESP Board

To connect DHT22 sensor to the board of ESP require to understand the pins of the two devices (Fig. 27); in fact, DHT22 sensor need to run under a voltage between 3.3 V and 5 V. It requires to be connected to the ESP board pin will accept input data (Table 4).

Figure 27: Digital humidity and temperature sensor 22 pinouts

Table 4: Digital humidity and temperature sensor 22 and ESP board connection

PIN	Function	How to connect to ESP board
1	VCC	3.3V
2	Data	GPIO 14
3	NULL(NC)	Not connected
4	GND	GND

3.6.2 Connecting Relays to ESP Board

Each connection on the relay module has three sockets: common (COM), normally closed (NC), and normally open (NO). The current to be regulated is connected to the COM socket (Fig. 28).

When the relay is meant to be closed by default, the NC (Normally Closed) configuration is utilized. Unless you transmit a signal from the ESP32 to the relay module to open the circuit and interrupt the current flow, the NC and COM pins are connected, which means the current is flowing.

Furthermore, the NO (Normally Open) configuration is the absolute opposite of the NC: there is no connection between the NO and COM pins, therefore the circuit is broken unless the ESP32 sends a signal to close it.

In the prototype, NO configuration was used to facilitate the test of the system. The ESP has to send a high signal to all relays to allow the current to flow.

Figure 28: Pin configuration of a single channel relay module

In the prototype, the NO is directly connected to the load (VCC of the fan) to control while the COM socket is connected to the main voltage (Solar battery). Therefore, the GND of the Fan is connected to the GND of the battery. The GND, VCC, and SIG of all relays are connected to the ESP board as it is indicated in Table 5.

Table 5: Relays' Pins connected to the ESP board

Relay Pins	How to connect to ESP board
VCC	5 V
SIG (Relay1)	GPIO 21
SIG (Relay2)	GPIO 22
SIG (Relay3)	GPIO 23
GND	GND

3.6.3 Implementation of the Overall System

Figure 29 shows the overall system design formed by an ESP32 board, 3 relays, and a DHT22 temperature sensor. Connections between them are done regarding Table 4 and Table 5.

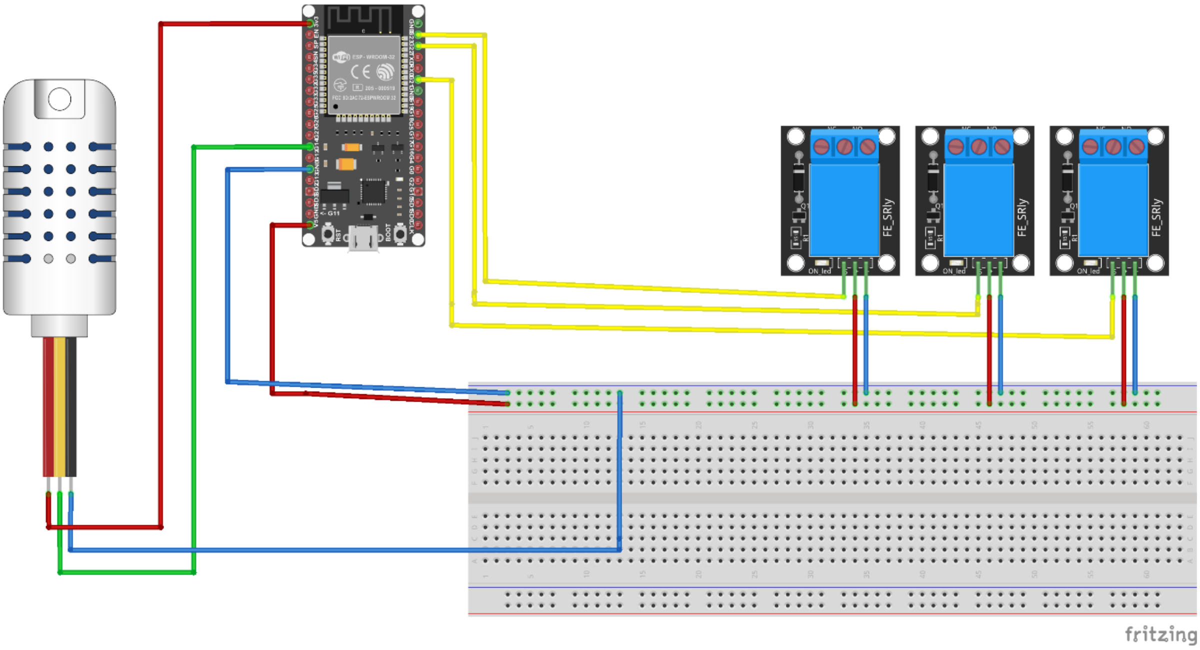


Figure 29: Prototype circuit diagram

The prototype was implemented on both breadboard (Fig. 30) and PCB (Fig. 31) to check whether the system can work perfectly in case of a DC and Alternative Current system. However, the voltage to control in really life is around 200 V, therefore the implementation of the prototype on the PCB brought much conviction of the usability of the system.

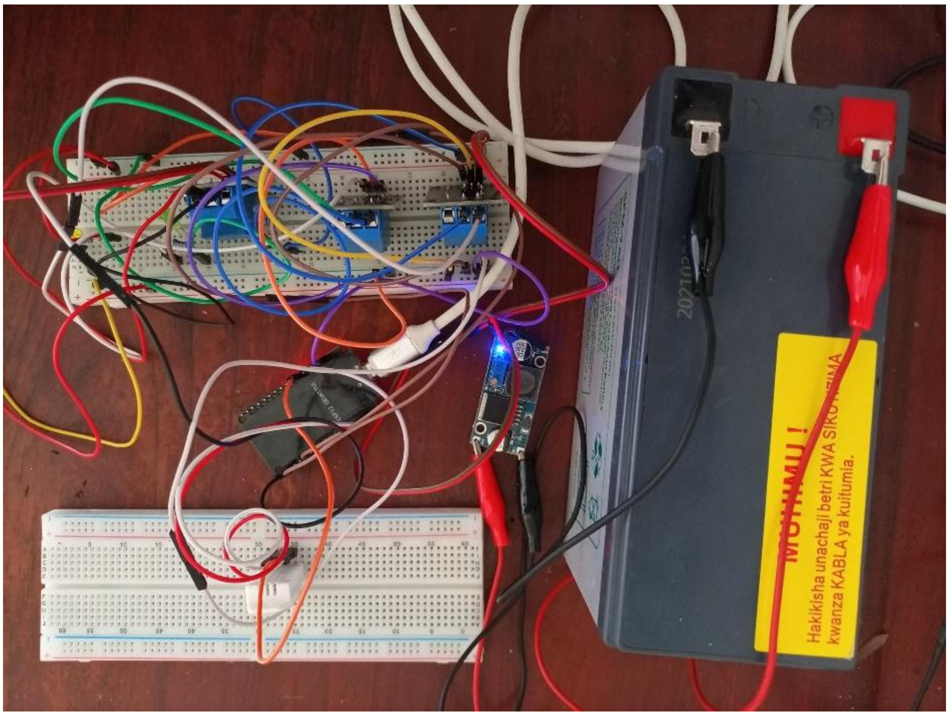


Figure 30: Prototype implemented on breadboards

Figure 31: Prototype implemented on PCB

3.7 System Testing

The proposed system was tested at Habari Node Plc in Arusha, where the company has a central data center. Both unity and integration testing were required to ensure that the system works as expected by the end-user.

3.7.1 Unit Testing

All parts of the system are supposed to be tested separately; the unit testing detects and corrects misbehavior once integrated into the whole system. The test starts from the temperature monitoring system to determine whether the sensed data from the data center room is sent and stored to the cloud. The process checks whether the stored data can be visualized from a remote place and through a mobile application. The testing proceeds with the notification via e-mail if the temperature level approaches the limit range. Finally, the test of both automated and remote control of the relays must be done to determine if they are working correctly.

3.7.2 Integration Testing

Each part of the system is integrated into the whole system for integration testing to make sure that the entire system can accurately work. The whole system was tested for many days to ensure that the system operates without interruption. The temperature monitoring system was checked whether the sudden temperature increase triggers the notification. It also tests whether the cooling device system automatically switches on the idle cooling device. The remote control of the cooling device was done to determine the ability of the system to accept the signal sent from the ESP web server (Appendix 3) through the mobile web application.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Overview

In this chapter, the outputs of the developed system are thoroughly discussed concerning the project objectives. The goal was to create a cost-effective automated air conditioner system for temperature monitoring and control from afar. The temperature monitoring system was able to detect temperature and communicate the measured value to the ThingSpeak cloud using the DHT22 sensor and ESp32 microcontroller board. Figure 38 shows how the ESP32 board might manage the cooling system remotely using a mobile device. When there was an observation of the temperature value overflowing, the notification system was able to send a mail and an SMS notice in real-time.

4.2 Temperature Monitoring System

The temperature monitoring system is composed of the ESP32 board, the DHT22 temperature sensor (Figure 32), and the ThingSpeak cloud storage.

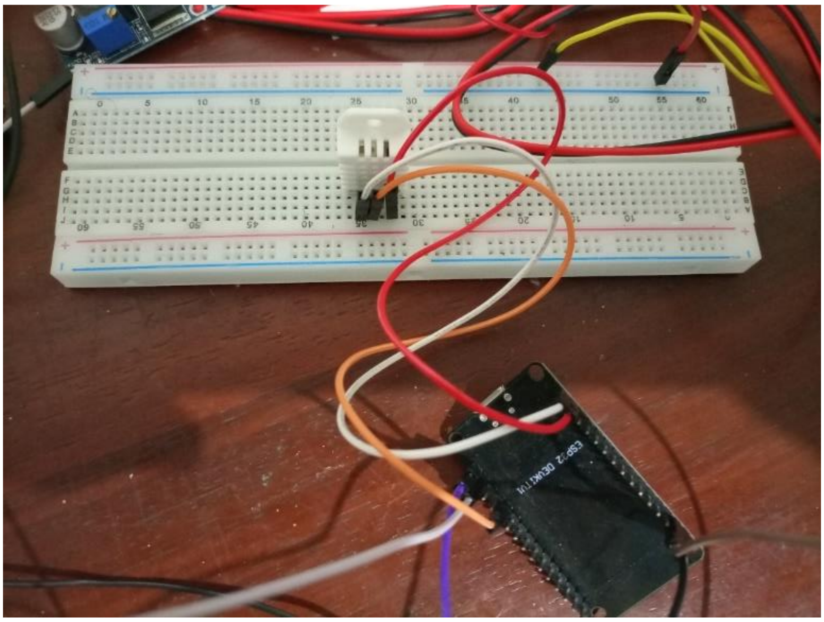


Figure 32: Temperature monitoring system

The temperature monitoring system was able to accurately detect the value of temperature in the environment and save it to the cloud (Fig. 33).

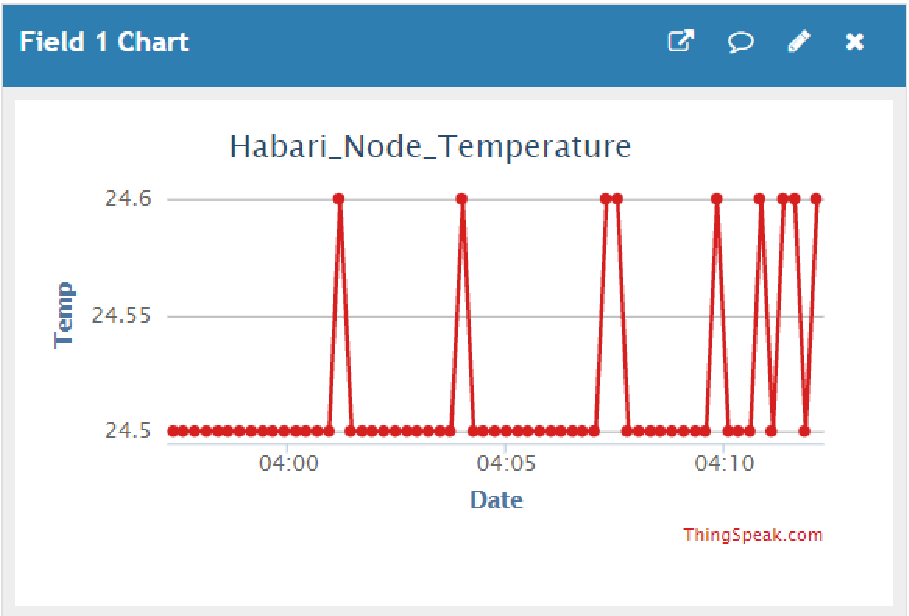


Figure 33: ThingSpeak chart of the temperature monitored value

Therefore, a gadget was set in the cloud to help end-user to see the variation of temperature in real-time. However, the last monitored value of temperature is displayed along with the gadget for much accuracy as shown in Fig. 34.

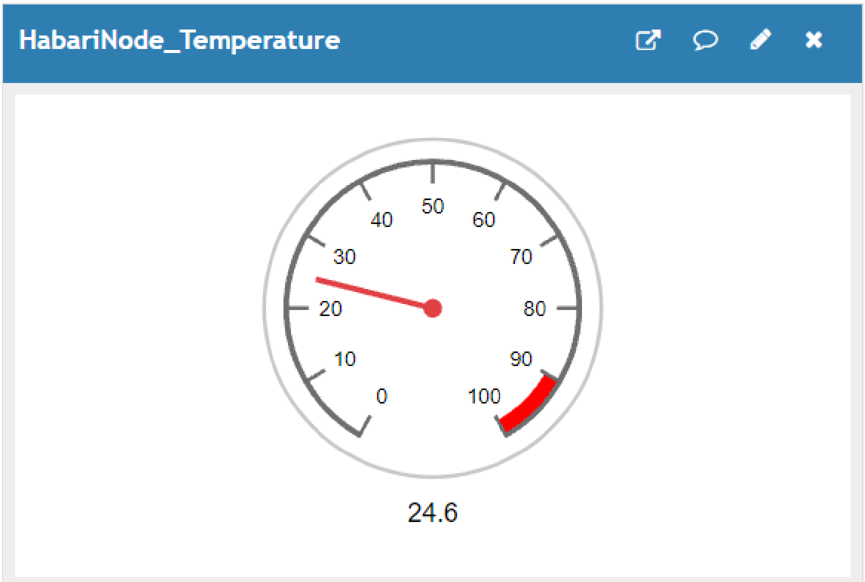


Figure 34: ThingSpeak gadget of the temperature monitored value

In contrast with the existing system to monitor temperature, the developed system was able to save data and to provide to the end-user an interface to visualize the current and recent value of temperature.

4.3 Notification System

A webserver was used to monitor the value and provide notifications when a weird value was found. The monitored value is sent by the ESP32 board to both the cloud and the website at the same time. If a threshold is surpassed (either below 22 ˚C or above 24 ˚C), an email is sent to the IT personnel.

The system has two kinds of notification; on one hand, there is an SMS notification sent to the mobile number of the Data center manager (Fig. 35).

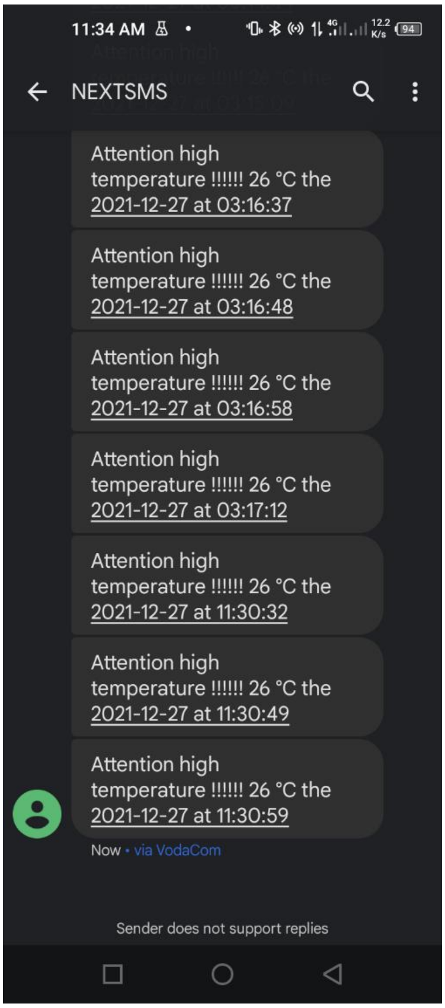


Figure 35: Short message service (SMS) notification

On the other hand, there is a mail sent to the data center manager just to compliment the SMS notification to maximize the transmission of the information to the end-user of the system (Fig. 36).

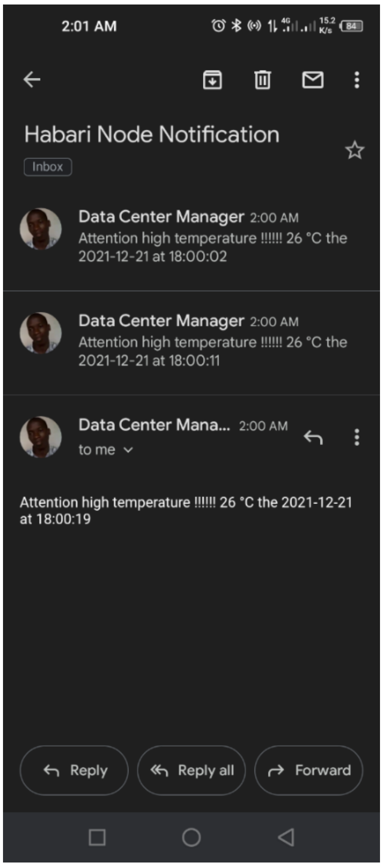


Figure 36: Mail notification

As indicated in Table 6, the new system powered by backups utilized would significantly contribute to saving the expenses used to run the current four cooling devices at the same time. Only three devices are used while one of them stands as a backup to intervene when it is needed. Therefore, it can reduce the cost spend on the data center facility to half since the idle cooling device would work rarely or when the data center wants to interchange it with a working one.

Table 6: Comparison between the estimated cost to run four devices and two devices

Window and Split AC	1 hour (kW)	1 day (24 h) (kW)	1 month (30days) (kW)
1 Device	2.2	52.8	1,584
4 Devices	8.8	211.2	6,336

2 Devices	4.4	105.6	3,168
Energy saved (Using 2 devices at a go instead of 4)	4.4	105.6	3,168

4.4 Data Processing and Visualization

4.4.1 ThingSpeak for Temperature Storage and Visualization

The temperature data from the data center was processed using an ESP 32 microcontroller board and a DHT22 sensor, as described in Appendix 2. As a result, whether through a web interface or on a mobile phone, the cloud provides an excellent interface for displaying the monitored value. The newest entry overlaps the plotted data on the mobile app interface, which displays all of the entries of the monitored value. In addition, Figure 37 shows the lowest and maximum values as well as the date when they were taken.

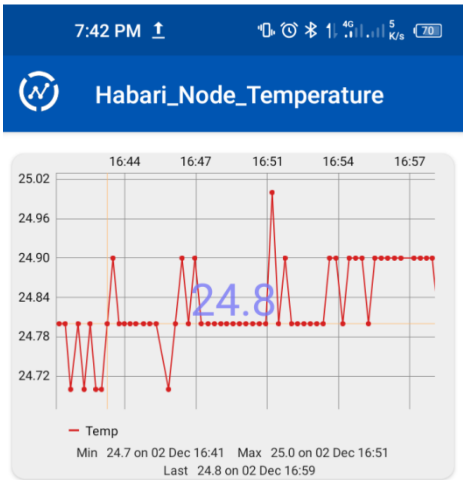


Figure 37: Mobile view of the ThingSpeak channel

4.4.2 ESP Web Application

The web application that allows remote control of cooling units was made possible because of the ESP webserver. The virtual push-button may then be switched on or off based on the signal sent through the ESP32 Room Board's web server.

Through the IP address supplied to the ESP board, the application may be accessed using any browser on a PC or a smartphone (Fig. 38).

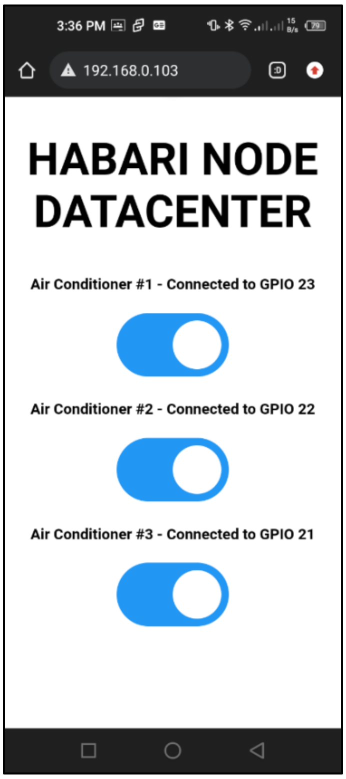


Figure 38: A user interface to control ESP board pins

The virtual toggle buttons (Fig. 39) might transmit a high or low signal to the board pin attached to the cooling device. They then send the input signal to the ESP board to control the related device connected directly to the pins of the ESP board.

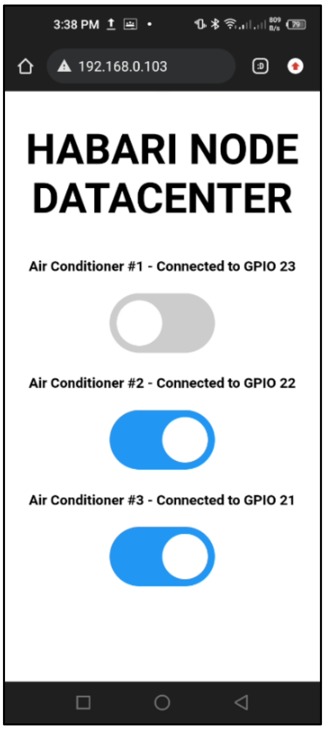


Figure 39: Toggle button in high signal to switch off the AC1

4.5 System Validation

The end-users were guaranteed the system's stability and efficacy due to the created system testing. The system was put to the test multiple times using the prototypes that had been created, and it was continually enhanced to meet the project's objectives. Unit and integrated testing were completed, and Habari Node Company's existing temperature monitoring equipment was used to compare the accuracy of the developed system and the existing system. The temperature data were recorded and compared to the temperature value identified by the devised system.

According to the system requirements, the temperature monitoring system, alerting system, and cooling system were tested and validated. The temperature data was sent to the cloud and visualized using a mobile app. The system prototype was presented to and tested by Habari Node Company's Technical Manager to ensure that all of the system's functions performed as planned.

The built system allowed for real-time and remote temperature monitoring of the Habari Node data center, which was a significant improvement over the current system at Habari Node. The data center manager was able to keep track of the temperature of the data center by receiving notifications in the form of messages and emails when the temperature rises above the usual range.

Table 7: User system acceptance testing

Testing output	Strong agree	Agree	Disagree
The developed temperature monitoring system is user friendly	**x**		
The measured temperature data is accurate		**x**	
The plot to visualize monitored values though think speak mobile app is readable	**x**		
The web server to remotely control cooling devices is responding in real-time	**x**		
The notification mail and SMS are important for the security of the Data center	**x**		
The auto-switching cooling system enhances the lifespan of the equipment		**x**	
The system would significantly help the	**x**		

company to reduce the cost of the power

The proposed system is relevant, affordable, usable, and meets the expectation of the user. The data center manager's experience at Habari Node has been substantially improved due to a real-time temperature monitoring system and a remote auto-switching cooling system that keeps the data center in normal air condition. Habari Node is eager to implement the designed system because it is inexpensive and functional.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The project aims to improve the temperature monitoring and control system of Habari Node Company. Habari node Company uses a common method of temperature monitoring that has many weaknesses, including the lack of storing the detected value, then the luck of the dashboard, and the use of a large number of cooling devices (Window Air Conditioner) that work together to help each other in the event of failure, resulting in high energy consumption. All of the project's goals were met; the temperature monitoring system, which included cloud storage and a dashboard, as well as the remote management of the cooling system through relays, the ESP Microcontroller Board, and the ESP web server, all worked as intended. Finally, to fill the holes of the current notification system, a notification system through SMS and email was established.

As a result of the project, an IoT system to monitor and regulate the HVAC in a data center has been developed. The measured temperature is stored in the cloud, which allows the end-user to view the monitored temperature remotely. The ESP web server gives the end-user a way to communicate with the ESP-WROOM32 microcontroller board remotely. It aids in the control of the data center's air conditioning. If the temperature falls outside of its standard range, the webserver sends an e-mail and SMS to the data center manager. Furthermore, the sensed temperature value is analyzed to determine whether the cooling devices can be turned on or off by the relays.

The project would considerably extend the life of cooling devices (Split and Windows AC) by allowing each of them to stay inactive for a short period. It also minimizes the cost of acquiring cooling equipment as well as their energy usage because only three devices (rather than four) may function at the same time, whereas two are continually operating at the same time. Temperature monitoring in real-time would aid in the prevention and identification of equipment failure.

If this system is implemented, it will allow for convenient real-time, low cost, and remote monitoring of temperature levels in all Habari Node Company's Data centers, hence facilitating server room cooling systems.

5.2 Recommendations

The use of IoT to monitor and control server temperature is advised in critical infrastructure like a data center, where Computer Room Air Conditioning is essential and a computerized maintenance system is important. It aids data centers in constructing an immutable infrastructure and becoming more resilient as a result. The collection of temperature data via a temperature monitoring system would result in a data Historian in cloud storage; hence, the implementation of a machine learning algorithm should aid in the forecast of temperature variation in the server room.

The concept can serve as a model for other areas (such as agriculture, healthcare, education, and others) that need the same solution to monitor and control related parameters due to its effectiveness in dealing with the challenges of monitoring and controlling temperature; it implies that any type of electrical appliance can be controlled through the internet instead of using a manual switch.

Furthermore, due to the project's time limits, the built temperature monitoring system can only be utilized to identify temperature changes that act as a parameter to indicate cooling system misbehavior. The project should then be upgraded by including humidity as co-parameter of temperature and sensors to detect any cooling device breakdown.

To collect temperature data from all Habari Node data centers, nodes must be installed to acquire data from all rooms, then all data must be brought together to be visualized and evaluated. As a result, a web software should be created to plot all Habari Node Data center dashboards in one spot. Furthermore, involving database in that a web-based software would help to improve the system by allowing for additional customization and management of users and Data centers.

Maintenance of the electrical devices and cooling devices is strongly recommended in order to resolve the problem relating to the cooling equipment' frequent misbehavior. The entire data center is at risk if the situation remains.

REFERENCES

Alvan, P. U., Aziz, A., Winarno, & Harjito, B. (2019). Server Room Temperature & Humidity Monitoring Based on Internet of Thing (IoT). *Journal of Physics: Conference Series*, *1306*(1). https://doi.org/10.1088/1742-6596/1306/1/012030

Association, G. (2014). Understanding the Internet of Things (IoT). *GSMA Connected Living, 1*(July), 13.

Controllers, S., & About, L. (2021). HVAC Management. 2.

Courtemanche, M. (2015). Data center terminology that will get you hired interview.

Dunlap, K., & Rasmussen, N. (2006). The Advantages of Row and Rack- oriented Cooling Architectures for Data Centers. *APC White Paper*, 1–19.

Fireteanu, V. V. (2020). Agile Methodology Advantages when delivering Internet of Things projects. In *2020 12th International Conference on Electronics, Computers and Artificial Intelligence*, 1-5. https://doi.org/10.1109/ECAI50035.2020.9223172

Folty´nek, P. F., Babiuch, M., & Uránek, P. S. (2019). Measurement and data processing from Internet of Things modules by dual-core application using ESP32 board. *Measurement and Control*, *52*(8), 970–984. https://doi.org/10.1177/0020294019857748

Lady. (2020). DHT11, DHT22 and AM2302 Sensors. https://learn.adafruit.com/dht

Li, Z., & Kandlikar, S. G. (2015). Current status and future trends in data-center cooling technologies. *Heat Transfer Engineering, 36*(6), 523–538.

Medina-Santiago, A., Azucena, A. D. P., Gomez-Zea, J. M., Jesus-Magana, J. A., De La Luz Valdez-Ramos, M., Sosa-Silva, E., & Falcon-Perez, F. (2020). Adaptive Model IoT for Monitoring in Data Centers. *IEEE Access, 8,* 5622–5634.

Mesbahi, M. R., Rahmani, A. M., & Hosseinzadeh, M. (2018). Reliability and high availability in cloud computing environments: A reference roadmap. *Human-Centric Computing and Information Sciences*, *8*(1), 1–31. https://doi.org/10.1186/s13673-018-0143-8

Meza, J., Xu, T., Veeraraghavan, K., & Mutlu, O. (2018). A large-scale study of data center network reliability. *Proceedings of the Internet Measurement Conference 2018*, 393-407. https://doi.org/TBA

Nasution, T. H., Muchtar, M. A., Seniman, S., & Siregar, I. (2019). Monitoring temperature and humidity of server room using Lattepanda and ThingSpeak. *Journal of Physics: Conference Series, 1235*(1), 012068. IOP Publishing.

Niemann, J., Bean, J., & Avelar, V. (2013). Economizer Modes of Data Center Cooling Systems. *White Paper, 132*(2013), 11–12.

Rahman, R. A., Hashim, U. R., & Ahmad, S. (2020). IoT based temperature and humidity monitoring framework. *Bulletin of Electrical Engineering and Informatics, 9*(1), 229–237. https://doi.org/10.11591/eei.v9i1.1557

Ramphela, M. K. J., Owolawi, P. A., Mapayi, T., & Aiyetoro, G. (2020). Internet of Things (IoT) integrated data center infrastructure monitoring system. *International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems, IcABCD 2020 - Proceedings,* 0–5.

Rebaudo, F., & Benoist, R. (2019). Low-cost automatic temperature monitoring system with alerts for laboratory rearing units. *MethodsX, 6*(May), 2127–2133.

Rui Santos. (2016). 9 Arduino Compatible Temperature Sensors | Random Nerd Tutorials. https://randomnerdtutorials.com/9-arduino-compatible-temperature-sensors-for-your-electronics-projects/

Saha, S., & Majumdar, A. (2017). Data Centre Temperature Monitoring with ESP8266 Based Wireless Sensor Network and Cloud Based Dashboard with Real Time Alert System. *International Devices for Integrated Circuit*, 307–310.

Sampera, E. (2019). The Server Room Explained and the Importance of Security.

https://www.vxchnge.com/blog/server-room-security-importance

Santos, S. (2019). DHT11 vs DHT22 vs LM35 vs DS18B20 vs BME280 vs BMP180 | Random Nerd Tutorials. https://randomnerdtutorials.com/dht11-vs-dht22-vs-lm35-vs-ds18b20-vs-bme280-vs-bmp180/

Sari, M., & Sadikin, N. (2019). Server Temperature Monitoring System Using Web Based Censor and SMS Gateway. *Journal of Physics: Conference Series, 1361*(1), 012026. IOP Publishing. https://doi.org/10.1088/1742-6596/1361/1/012026

Sari, M., Sadikin, N., & Fauzan, M. (2021). Server Electricity and Temperature Monitoring By Using Access Point, Lm 35 Censor And Java Programming Language To Check Electricity Parameter. In *Journal of Physics: Conference Series* (Vol. 1830, No. 1, p. 012015). IOP Publishing. https://doi.org/10.1088/1742-6596/1830/1/012015

Sarkar, C. (2010). Data Center Network Design. February, 1–9.

Seyam, S. (2018). Types of HVAC systems. *HVAC System*, 49-66.

Shamang, K. J., Chukwuma-Uchegbu, M. I., & Sa'id El-nafaty, A. (2020). Indoor Temperature and Humidity Monitoring System. *Journal of Science Engineering Technology and Management*, *02*(05), 50–58. https://doi.org/10.46820/jsetm.2020.2503

Sharma, S., Sarkar, D., & Gupta, D. (2012). Agile Processes and Methodologies: A Conceptual Study. *International Journal on Computer Science & Engineering*, *4*(5), 892–898. http://search.ebscohost.com/login.aspx?direct=true&db=aph&AN=82397457&site=ehost-live

Stryer, P. (2010). Understanding Data Centers and Cloud Computing. *2010 Global Knowledge Training LLC*, 1–7. www.globalknowledge..com

Tom Collins. (2016). 7 Key Points to Consider When Choosing a Data Center. https://www.atlantech.net/blog/7-key-points-to-consider-when-choosing-a-data-center

Woods, A. (2010). Cooling the data center. Queue, 8(3), 1–10.

APPENDICES

Appendix 1: Acceptance Letter

HABARI NODE

Our Ref:

Date: 26th April, 2021

Email :sales@habari.co.tz

Phone: +255 411 200 900

Mr. Innocent Ciza

Postal address:1215 Arusha.

P.o Box 447

NM-AIST

Arusha -Tanzania

Dear Mr. Innocent,

Re:Offer of Internship at Habari Node Company.

This is in response to your internship application dated March 16th, 2021 at Habari Node Company, Arusha.

Your internships have been accepted by the company for a period of six months, and you will be assigned to the IT department under the supervision of the Technical Manager.This internship will begin on Jun 08, 2021 and will end on December 31, 2021.

However, you should be aware that Habari Node Company may or may not provide financial assistance or allowances to interns. This is subject to discussion.In the meantime, interns will be responsible for their own maintenance throughout the duration of the internship.

Interns should conduct their relevant industrial Supervisor if they have any concerns or need clarification.

Your Sincerely

HABARI NODE PLC P.O. BOX 1215 ARUSHA - TANZANIA

Eng. Lomayani Solomon

Technical Manager, Habari Node Company Arusha.

Appendix 2: Interview Guide

IoT based system for monitoring and control Datacenter temperature at Habari Node Campany

Questionnaire

I. Background information

1. How many datacenters does Habari Node Plc have?

2. Which are the major security of the datacenter at Habari Node?

3. What is the normal range of temperature inside the datacenter?

4. How does window AC work?

5. Do you have IT company of which the data are stored into your servers?

6. How much (in term of percentage) does the company invest in Datacenter infrastructure management)

II. Problem statement and justification

7. Does the company have cooling device replacements?

If yes, how many are they?

8. How much (in term of percentage) do they pay for reparation of failed cooling device comparing to buying a new one?

9. How does the current system of temperature monitor system works?

10. How often it sends notifications

11. What kind of notification message does it give?

12. How do you fix issue when you observe that the temperature level has reached the normal range?

13. How often does the temperature level go beyond the limit of the normal range?

14. Do you have someone who is in charge of assisting the data-center manager and then therefore could help fix the issue when the data-center manager is not around?

15. How many times the cooling devices fails?

16. What is the cause of the failure of the cooling devices?

Note: All questions are asked for the purpose of research to input the project and then their answers are not intended to be used for any other purpose.

Appendix 3: ESP Web Server Code

```
const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE HTML><html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <style>
    html {font-family: Arial; display: inline-block; text-align: center;}
    h2 {font-size: 3.0rem;}
    p {font-size: 3.0rem;}
    body {max-width: 600px; margin:0px auto; padding-bottom: 25px;}
    .switch {position: relative; display: inline-block; width: 120px;
     height: 68px}
    .switch input {display: none}
    .slider {position: absolute; top: 0; left: 0; right: 0; bottom: 0;
     background-color: #ccc; border-radius: 34px}
    .slider:before {position: absolute; content: ""; height: 52px;

     width: 52px; left: 8px; bottom: 8px; background-color: #fff;
     -webkit-transition: .4s; transition: .4s; border-radius: 68px}
    input:checked+.slider {background-color: #2196F3}
    input:checked+.slider:before {-webkit-transform: translateX(52px);
      -ms-transform: translateX(52px); transform: translateX(52px)}
  </style>
</head>
<body>
  <h2>HABARI NODE DATACENTER</h2>
  %BUTTONPLACEHOLDER%
<script>function toggleCheckbox(element) {
  var xhr = new XMLHttpRequest();
  if(element.checked){ xhr.open("GET", "/update?relay="+element.id
    +"&state=1", true); }
  else { xhr.open("GET", "/update?relay="+element.id+"&state=0"
    , true); }
  xhr.send();
}</script>
</body>
</html>
)rawliteral";
```

Appendix 4: Arduino Sketch to Send Temperature Data to ThingSpeak

```
    void send_to_cloud(){
    float h = dht.readHumidity(); // Reading Temperature form DHT
    float t = dht.readTemperature(); // Reading Humidity form DHT

     if (isnan(h) || isnan(t)) {
     Serial.println("Failed to read from DHT sensor!");
     return;
     }
     ThingSpeak.setField(1, t);
     ThingSpeak.setField(2, h);
    int Data = ThingSpeak.writeFields(1574895, myWriteAPIKey);
   if(Data == 200){
     Serial.println("Channel updated successfully!");
   }
   else{
     Serial.println("Problem updating channel.
     HTTP error code " + String(Data));
   }
   delay(6000);
 }
```

Appendix 5: Arduino Sketch for Relay Auto-Switching

```
void autoSwitch(float temp,int relay1,int relay2,int relay3){

 if(temp>25.0){

  digitalWrite(relay1, HIGH);
  digitalWrite(relay2, HIGH);
  digitalWrite(relay3, HIGH);

  }else if(temp<22.0){

    digitalWrite(random_relay(), LOW);

    }

  }

  int random_relay(){

    return random(21, 23);

}
```

Appendix 6: PHP Script to Send Email and SMS

Script to send email

```php
    public function index()
    {

      $api_key = $temp = $hum =  "";

     if($_SERVER["REQUEST_METHOD"] == "POST") {


        $temp =$this->test_input($_POST["value1"]);
        $hum =$this->test_input($_POST["value2"]);


         if ($temp >= 24 || $temp < 22) { // send message only when temp is less than 22 or greater than or equal 24


            //print_r($users_Manager);exit();
            $from_email = "cibokabafu@gmail.com";
            $this->load->library('email');


            $this ->email->from($from_email, "Data Center Manager");
            $this ->email ->to('icizakaba@gmail.com');
            $this ->email->subject('Habari Node Notification');
            $this ->email->message($SMS_NOTIF);

            $this->email->send();

            echo "Mail Sent";

```

Script to send SMS

```php
                // **********sms code begins here*******

        $curl = curl_init();

        $telephone = '255756968493';

        $SMS_NOTIF = $this->Customize_Notif($temp); // Call to the function to create the message

        curl_setopt_array($curl, array(
          CURLOPT_URL => 'https://messaging-service.co.tz/api/sms/v1/text/single',
          CURLOPT_RETURNTRANSFER => true,
          CURLOPT_ENCODING => '',
          CURLOPT_MAXREDIRS => 10,
          CURLOPT_TIMEOUT => 0,
          CURLOPT_FOLLOWLOCATION => true,
          CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
          CURLOPT_CUSTOMREQUEST => 'POST',
          CURLOPT_POSTFIELDS =>'{"from":"NEXTSMS", "to":"'.$telephone.'",  "text": "'.$SMS_NOTIF.'"}',
          CURLOPT_HTTPHEADER => array(
            'Authorization: Basic QmlsbDM2MDpiZWNrZXQjMzYw',
            'Content-Type: application/json',
            'Accept: application/json'
          ),
        ));

        $response = curl_exec($curl);


            }
         }
```

Script to test the inputs and to set notification message

```php
public function test_input($data) {

  $data = trim($data);
  $data = stripslashes($data);
  $data = htmlspecialchars($data);
  return $data;
 }

   public function Customize_Notif($temperature) {

  // Set timezone to Tanzania
      date_default_timezone_set('Africa/Dar_es_Salaam');


      $d = date("Y-m-d");
      $t = date("H:i:s");

   if ($temperature<22) {

      $SMS_NOTIF = 'Temperature is low !!!!!! '.$temperature.' °C the '.$d.' at '.$t;

   } else if ($temperature>25) {

      $SMS_NOTIF = 'Temperature is High !!!!!! '.$temperature.' °C the '.$d.' at '.$t;

   }else if ($temperature>=24){

      $SMS_NOTIF = 'Temperature is turning High !!!!!! '.$temperature.' °C the '.$d.' at '.$t;
   }

   return $SMS_NOTIF;

 }
```

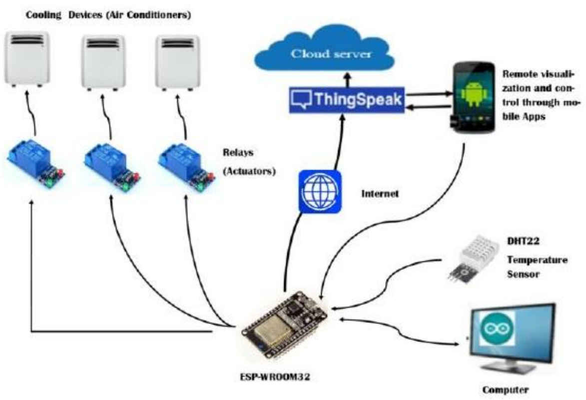
Appendix 7: Poster Presentation

IoT based system for monitoring and control Datacenter temperature at Habari Node Company

1. Innocent Ciza, 2. Dr. Ramadhani Sinde, 3. Prof. Shubi Kaijage

Introduction

In order to provide features like real-time temperature notification, equipment failure prevention, temperature tracking, a virtual remotely accessible dashboard to display recent and current temperature values, and remote and automated control over cooling devices, this study developed an IoT-based system for monitoring and controlling data center temperature at Habari Node PLC company. As a result, in order to provide a low-cost solution, the prototype includes a temperature monitoring system, customized air conditioning and notification systems through a web server, a data storage system via a cloud-based platform.



Problem Statement

Habari node Company uses a common method of monitoring temperature, which has many limitations, such as the luck of storing the detected value, then the luck of the dashboard, and the use of many cooling devices (Window Air Conditioner) that work together to help each other in case of failure, resulting in a high energy consumption. The existing notification system is based on email and has no configurable options.

Tools

Electronic devices	Software and platforms
DHT22 sensor Three AC fans ESP-WROOM32 Relay switches	Arduino IDE ThingSpeak Platform Web Server NextSMS Platform

Results

A cost-effective automated air conditioner system for temperature monitoring and remote control was created. The temperature monitoring system was able to detect temperature and communicate the measured value to the ThingSpeak cloud using the DHT22 sensor and ESP32 microcontroller board. When the temperature value is out of the normal range, the notification system sends a mail and an SMS in real-time.

Conclusion

The project could resolve a number of challenges of the datacenter. really time dashboard for data visualization and remote control of air condition. The reduction from four to three devices reduces the energy consumption and reduces the cost of purchasing cooling. It increases the lifespan of cooling. To collect temperature data from all Habari Node data centers. Finally, the system enhances the notification system by including SMS and allowing for extensive customization.