UNIVERSITY of RWANDA

ACEIoT

Website: www.aceiot.ur.ac.rw
Mail: aceiot@ur.ac.rw

COLLEGE OF SCIENCE AND TECHNOLOGY

"DESIGN OF AN INTELLIGENT CONTROL SYSTEM FOR SMALL-SCALE POULTRY FARMS"

MASTER'S DISSERTATION

Submitted in partial fulfilment of the requirements for the award of

MASTERS OF SCIENCE DEGREE IN INTERNET OF THINGS – EMBEDDED COMPUTING SYSTEMS

Submitted by

WILLIE BRIAN KASAKULA (REF.NO: 219013718)

Under the Supervision of

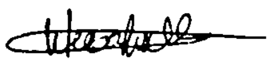
Dr Omar Gatera
Dr Didacienne Mukanyiligira

23rd NOVEMBER 2020

DECLERATION

I Willie Brian Kasakula hereby declare that this thesis is an original report of my research, has been composed sorely by myself and has not been submitted, in whole or in part, for any other degree or qualification to any other University. Except where stated by reference or acknowledgement, the work presented is entirely my own.

Name: **Willie Brian Kasakula**



Signature:

Date: 23rd November 2020

BONAFIDE CERTIFICATE

BONAFIDE CERTIFICATE

This is to certify that the project entitled "Design of an Intelligent Control System for Small-Scale Poultry Farms" is a record of original work done by Willie Brian Kasakula with reg number 219013718 in partial fulfilment of the requirement for the award of masters of sciences in Internet of Things in College of Science and Technology, University of Rwanda, Academic year 2021.

This work has been submitted under the guidance of Dr Omar GATERA and Dr Didacienne Mukanyiligira.

Main Supervisor:

Dr Omar GATERA

Signature:

Co-Supervisor:

Dr Didacienne Mukanyiligira

Signature:

The Head of Masters and Trainings

Names: Dr Didacienne Mukanyiligira

Signature:

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisors, Dr Omar Gatera and Dr Didacienne Mukanyiligira, whose support guided me throughout the study leading to this thesis document. They replied to my emails in time and provided me with useful insights that led to the successful execution of the projects.

I also wish to acknowledge my colleagues from the University of Malawi – Chancellor College, who I worked with during my internship where the idea for this study came from. Their ideas and support paved a way for me and for that I am grateful.

Most importantly, I extend my appreciation to Dumisani, Cliff, Kambombo, and all my friends and family. Their support was invaluable and they helped in all the ways they could to make this journey as easy as possible.

ABSTRACT

The world has been experiencing rapid growth in the poultry sector and this growth is projected to continue in the foreseeable future. The per capita consumption of poultry products is increasing rapidly and it is putting stress on the poultry farmers especially in low-income countries. This is so because low-income countries depend on small-scale farmers for their poultry products but small-scale farmers fail to afford the inclusion of technology which would help them to improve their production the same way it improves production for medium to large scale farmers. With these small-scale farmers in mind, it is, therefore, a requirement that technology is developed to ensure improved production while maintaining low-cost input as a requirement for small-scale production. This study focused on using new methods in the development of an embedded control system for poultry farms. Firstly, temperature and humidity data were collected to be used to determine their distribution in a room. It was found that in a well-ventilated room, the temperature and humidity have an equal distribution such that one sensor is enough to monitor both these parameters. This result was then used to extend to rooms with one heater, another with more than one heater. The circle packing algorithm was then used to propose a way to arrange the heaters across a poultry house. Based on this arrangement, a sensor placement algorithm was then proposed which enables the monitoring of the poultry farm in zones such that a system malfunction in one zone can be pinpointed and not be let to affect the system performance of the whole farm.

Having proposed the sensor placement model, the way actuators are controlled was also examined. A hypothesis was proposed to say that if the relationship between temperature and humidity was considered during their control, an algorithm would be designed to leverage the relationship and achieve control of the said parameters using less electrical energy. An algorithm was then designed and simulated. A comparison was done against a conventional algorithm that does not consider the temperature-humidity relationship. It showed that using the proposed algorithm, the actuators controlled the respective parameters in less time hence using less electrical energy. The actuators also operated in short bursts when using the conventional algorithm as compared to the proposed algorithm. The hypothesis was them proved to be true.

After these two algorithms were developed, an embedded system was designed using the algorithms. It was designed to have multiple sensor and actuators for each of the monitored environmental parameters. It was also designed with multiple communication channels to be able to send data to the cloud even when one of the communication channels is broken.

This embedded system was then coupled with the cloud to provide real-time monitoring of the system performance and data storage for later use. The cloud had an API, Database and a web data dashboard. The embedded system sent data to the cloud through this API and the API persisted the data to the database. Simultaneously, the data was also sent to the front-end dashboard through the same MQTT broker to which the dashboard was subscribed to. Users were provided with the ability to monitor the poultry farms in real-time. The users also use the same dashboard application to set configurations for the embedded system as well as to manually control some actuators as an override to the automatic operation of the system.

The study achieved all its objectives by doing the above discussed but there is still much more potential for further research in the field of poultry farm control systems

LIST OF ACRONYMS

AMQP Advanced Message Queuing Protocol

API Application Programming Interface

BLE Bluetooth Low Energy

CoAP Constrained Application Protocol

CSV Comma-Separated Values

DBMS Database Management System

DHT Digital Humidity and Temperature

GPIO General Purpose Input/Output

GPS Global Positioning System

GSM Global System for Mobile communication

HTTP Hypertext Transfer Protocol

HVAC Heating, Ventilation and Air Conditioning

I/O Input/Output

IoT Internet of Things

JSON JavaScript Object Notation

LCD Liquid Crystal Diode

LED Light Emitting Diode

LoRa Long Range

MCU Microcontroller Unit

MQTT Message Queuing Telemetry Transport

NFC Near-Field Communication

RAM Random Access Memory

REST Representational State Transfer

RFID Radio Frequency Identification

RTOS Real-Time Operating System

SBC Single-Board Computer

SQL Structured Query Language

THI Temperature-Humidity Index

LIST OF FIGURES

Figure 1: Tunnel Ventilation....7
Figure 2: Message Size vs Overhead....9
Figure 3: Power Consumption vs Resource Requirement....9
Figure 4: Bandwidth vs Latency....10
Figure 5: Reliability/QoS vs Interoperability....10
Figure 6: Security vs Provisioning....10
Figure 7: M2M/IoT Usage vs Standardization....10
Figure 8: Room Setup....15
Figure 9: NodeMCU....16
Figure 10: DHT Sensor....18
Figure 11: Embedded System Flow Chart....21
Figure 12: Data Collection System Architecture....24
Figure 13: Data Representation....25
Figure 14: Hexagonal Lattice Circle Packing....30
Figure 15: Sensor Placement Algorithm....32
Figure 16: Login Screen....47
Figure 17: Registration Screen....48
Figure 18: Home Screen (Dashboard)....48
Figure 19: Animal Breed Configurations....49
Figure 20: Editing Configurations....50
Figure 21: Deleting Configurations....50
Figure 22: Sensor - Actuator Mapping....51
Figure 23: Entity Relationship Diagram....55
Figure 24: Poultry Farm Control System Architecture....56
Figure 25: Poultry Farm Controller Architecture....57
Figure 26: DHT 22 Sensors....60
Figure 27: MQ-135 Air Quality Sensors....60
Figure 28: Ultrasonic Sensor....61
Figure 29: Waterflow Sensor and Water Valve....61
Figure 30: Exhaust Fan....62
Figure 31: Servo Motor....63
Figure 32: Relay Modules....64
Figure 33: Pin Extender....64
Figure 34: Temperature Data....67
Figure 35: Temperature Ranges....68
Figure 36: Humidity Data....69
Figure 37: Humidity Ranges....69
Figure 38: Temperature Correlation....70
Figure 39: Humidity Correlation....71
Figure 40: Sensor Arrangement....73
Figure 41: Modified Temperature Range....74
Figure 42: Modified Temperature Data....74
Figure 43: Modified Humidity Data....75

Figure 44: Modified Humidity Ranges......75

LIST OF TABLES

Table 1: Database Schema....20
Table 3: Initial Project Budget....44
Table 4: Final Budget....45
Table 5: Project Timetable....45
Table 6: Temperature Differences....68
Table 7: Temperature Differences....70
Table 8: Simulation Results....76

TABLE OF CONTENTS

Contents

"DESIGN OF AN INTELLIGENT CONTROL SYSTEM FOR SMALL-SCALE POULTRY FARMS" i

DECLERATION i

BONAFIDE CERTIFICATE ii

ACKNOWLEDGEMENTS iii

ABSTRACT i

LIST OF ACRONYMS ii

LIST OF FIGURES iii

LIST OF TABLES v

TABLE OF CONTENTS vi

1. INTRODUCTION 1

1.1 Background and Motivation 1

1.2 Problem Statement 2

1.3 Study Objectives 2

1.4 Hypotheses 3

1.4.1 General Hypothesis 3

1.4.2 Specific Hypotheses 3

1.5 Study Scope 3

1.6 Significance of the study 4

2. LITERATURE REVIEW 5

3. RESEARCH METHODOLOGY 13

3.1 Introduction 13

3.2 Sensor Placement 14

3.2.1. Data Collection 14

3.2.2. Overall Architecture 24

3.2.3. Data Analysis 24

3.2.4. Introducing Heaters 27

3.2.5. Different Farm Scenarios Based on Heaters 29

3.3 Actuator Control Algorithm 33

3.3.1. Algorithm Design 33

3.3.2. Algorithm Simulation 36

3.4 Conclusion 39
4. SYSTEM ANALYSIS AND DESIGN..... 40
4.1. Systems Planning..... 40
4.1.1. System Components 40
4.1.2. System Feasibility 42
Arduplane Airspeed Sensor 44
4.2. Systems Design 47
4.2.1. User Interface Design 47
4.2.2. Data Design 52
4.2.3. Architecture Design..... 56
4.3. Systems Implementation..... 58
4.3.1. Application Development..... 58
5. RESULTS AND DISCUSSION 67
5.1. Data Representation 67
5.1.1. Temperature..... 67
5.1.2. Humidity..... 69
5.2. Correlation 70
5.3. Introduction of Heaters..... 71
5.3.1. No Heaters..... 72
5.3.2. One Heater..... 72
5.3.3. N Heaters..... 72
5.4. Actuator Control Algorithm..... 73
5.4.1. Simulation..... 76
6. CONCLUSIONS AND RECOMMENDATIONS..... 77
6.1. Conclusion 77
6.2. Recommendations..... 77
References 78

1. INTRODUCTION

1.1 Background and Motivation

Poultry refers to birds such as chicken, turkeys and ducks, that are bred for their meat and eggs [1]. Poultry birds are also bred for their feather waste which is mixed with the soil and it plays the role of nitrogen fertilizer [2]. It contains high amounts of organic matter (35.9%) and nitrogen (4%) in addition to other major nutrient elements. The demand for poultry meat and eggs is increasing and expected to continue increasing due to population growth and the increasing individual consumption [3]. The per capita growth is slightly higher in developing countries than in developed countries and the market for poultry meat is projected to increase regardless of region or income level.

However, a study in a developing country (Pakistan) [4] showed that the average cost of production affects sales price which further affects the supply of poultry products significantly. In light of the issues discussed above, the field of poultry science is growing with a rapid pace and following are the currently trending topics of research in poultry science:

The risk mitigation of diseases in poultry is one of the most important topics in poultry science. Poultry birds suffer diseases from nutritional and metabolic causes but the most fatal ones are those caused by infectious agents [5] which harm the profitability of commercial operations. Dealing with these diseases is essential

Monitoring and control of the rearing environment is another hot topic in poultry science. The Temperature-Humidity Index determines the heat stress level in livestock [6]. Heat stress causes reduced bodyweight of the birds and causes a drop in the quantity of the eggs and quality of the eggshells. An increased level of ammonia causes respiratory diseases [7] and poor lighting affects food intake of the birds while the same light during the resting periods prevents the birds from resting properly [8]. Controlling the rearing environmental to prevent these unfavourable climatic conditions ensures better welfare of the birds which reduces mortality and increases production.

Another focal point in poultry science is in nutrition and food technology. Much work has been done in increasing ingredient utilization by the birds (feed processing and particle size) [9], adding enzymes to improve ingredient digestibility and nutrient absorption [10] which greatly increases production.

Numerous reports over the past decades have shown that genetic modification in poultry also has a high potential of improving poultry production by improving both the egg and tissue nutrient content [11]. This is achieved through the creation of gene-targeted insertions and deletions in avian species.

This study, therefore, falls in the field of Monitoring and Control of the rearing environment. Specifically, the research will use tools from embedded computing and artificial intelligence. An embedded control system for the rearing environment will be developed to monitor and control the climate in a poultry farm. The knowledge that this research intends to add to the body of knowledge is on how to achieve real-time monitoring of temperature and humidity and new methods on how to intelligently control the same for a more efficient electrical energy consumption. Knowledge will also be added in the modelling of sensor placement in a poultry farm to effectively monitor the environment in the poultry farm by reducing the number of sensors used while maximizing the coverage area of the sensors.

1.2 Problem Statement

The rapid growth of the poultry sector in developing countries, where small-scale farmers fail to meet the demands for poultry products, has introduced opportunities as much as it has brought problems to the farmers. Fortunately, advances in technology have shown to have the potential to help the farmers in increasing the supply to meet the growing demand. Among these many technological solutions are control systems for the rearing environment. They work by sensing the climatic conditions in the poultry farm through their sensors and with the help of actuators, they control the environment to make it more conducive for the desired growth of the poultry birds [8] [12] [13]. It has been proven that these systems increase production significantly and help in the reduction of many health-related problems in the birds. The data they collect through the sensors is then sent to the cloud which analyses the data to draw useful insights.

The main problem posed by these control systems is in their cost. They have high upfront and operational costs such that it becomes economically infeasible to small-scale farmers to invest in them. It is, therefore, necessary that new methods of developing and operating these control systems be developed such that they become feasible for small-scale farmers to use to increase their production. This study focuses on two main aspects of these control systems to increase efficiency.

i. Developing a sensor positioning model to increase monitoring efficiency which can then increase the overall efficiency of the system

ii. Developing an actuator operation model to control the environmental parameters while using less electrical energy than the current systems.

Most of the operating costs of the system come from the cost of the electrical energy used by the actuators, i.e. heaters. If their operation can be made more efficient, there is a possibility that the poultry farm control system, overall, can have significantly reduced costs and be made available for use by small-scale farmers.

This study aims at solving the above-stated problem and below is a detailed report on how these problems were approached and the results of the solutions that were taken.

1.3 Study Objectives

General Objective

The main objective of this study is to develop an environment control system for poultry farms, that will always keep the environment in the farm optimal for the maximum possible product while keeping the costs low.

Particularly, the study has the following specific objectives:

Specific Objectives

a. To design a control system for a poultry farm environment.
b. To model the optimal sensor placement for a poultry farm.
c. To model the operation schedule of actuators to reduce energy consumption
d. To design a web-based data dashboard for farm data access and analysis.

1.4 Hypotheses

1.4.1 General Hypothesis

The general hypothesis of the study is as follows:

A Poultry Farm Control System can be designed with intelligence to make it more effective and efficient for small-scale farmers.

This hypothesis was split into more specific but measurable hypotheses.

1.4.2 Specific Hypotheses

a. A sensor placement model can be developed to position sensors in poultry farm such that they can monitor every section of the farm individually and the farm as a whole to reduce monitoring errors.
b. If the relationship between temperature and humidity is considered by the actuator control algorithm, the control of the humidity and temperature actuators would be more efficient.

If the above stated specific hypotheses are proved, the results can be used in the design of a poultry farm control system and hence the general objective of the study will have been proved.

1.5 Study Scope

Numerous poultry farm control systems have been developed in literature and they cover many aspects of the design and implementation of it. This particular study, however, aims at designing an intelligent poultry farm control system, that is mainly tailored for small-scale poultry farmers. The control system will mainly focus on monitoring and controlling the environment in the poultry farm i.e. temperature, humidity and light. The study prototype to be developed in this study will not include other aspects of poultry farm control system such as automatic feeding even though these aspects will still be discussed.

The study will put much focus on the sensor placement, actuator controlling, and remote monitoring of the poultry farm environment through a web dashboard. It will be conducted in partial fulfilment of the requirements for the award of a Master's of Science Degree in Internet of Things - Embedded Computing Systems. It is to be conducted from March 2020 to November 2020.

The topics and theories to be discussed will range from algorithm design and analysis, networks, electronic designs, internet of things and embedded computing. Some specific details from these broader terms will be used to achieve the objectives of this study.

1.6 Significance of the study

The main goal of this research is to make efficient a technology that was rather only feasible for use by large scale farmers who contribute only a small portion to the poultry need of the general population. This technology, if adopted, has the potential to increase the production of small-scale poultry farming significantly such that the ever-increasing demands for poultry products would be met.

Poultry products are consumed and/or used by people from all over the world regardless of their economic status. Therefore, if the objectives of this study are to be met, it will affect people from all over the world by improving their access to poultry products. It is also the case that the increase in the supply of poultry products may lead to a reduction of their cost such that the nutrients poultry products have to offer will be made available to people easily.

2. LITERATURE REVIEW

The World Organization for Animal Health defined animal welfare as follows [14]: "An animal is in a good state of welfare if (as indicated by scientific evidence) it is healthy, comfortable, well-nourished, safe, able to express innate behaviour, and if it is not suffering from unpleasant states such as pain, fear and distress". For long term sustainability of commercial poultry farming, farmers need to consider the overall welfare of the birds. Many factors need to be considered when addressing issues to do with the welfare of the poultry. These include the suitability of the social environments, stocking density, thermal stress, environmental deterioration, or the difficulty in accessing the necessary resources for the welfare of the birds. In light of these factors, it becomes important to assess the welfare of the birds to get information on what needs to be controlled or minimized and how to achieve it. Assessing the birds' welfare, however, becomes difficult when dealing with large poultry flocks [15]. With this problem in mind, new technologies have emerged in attempts to address it. These include sensors for environmental monitoring, movement, or physiological parameters; imaging technologies to detect feather pecking; infrared technologies to evaluate the thermoregulatory features and metabolism changes that are directly linked with health, management problems and overall welfare of the birds. Different sensors were used to achieve the monitoring of the birds' welfare [15]. These were:

a. Environmental Sensors

   The environment in which the birds live in plays a vital role in the welfare of the birds and hence the overall profitability of the poultry farming business. This includes the temperature, relative humidity, and the levels of noxious gases like carbon dioxide and ammonia which can affect growth, feed conversion and immunological response [16] [17]. Thus, monitoring these environmental parameters is a step closer to monitoring the welfare of the birds on the poultry farm.

b. Acoustic Sensors

   Bio acoustics studies the sounds emitted by organisms since they carry a biological significance. Birds use acoustic communication as stress indicators, alarm signalling and even for their social interactions [18]. Using vocalization frequency, Zimmerman et al. [19] were able to detect the feeding behaviours of the birds i.e. feeding motivation and frustration. Sound analyses have also been used to estimate the thermal comfort of the birds and to reduce the hatching window. Using acoustic sensors, therefore, one can be able to monitor the welfare of the birds and get information on where to make improvements and how.

c. Movement Sensors

   As per the definition of animal welfare by the World Organization for Animal Health, as stated above, an animal is in a good state of welfare if it can express innate behaviour [14]. Free movement by the birds is an innate behaviour, however, the conditions in the rearing environment of the birds may hinder the free movement in the birds due to health conditions, space availability and high density among others [20]. To monitor the birds' movement, Daigle et al. used a wireless sensor attached to the birds' bodies in combination with a Global Positioning System (GPS) to study how movement and behaviour of the birds are related.

d. Sensors for Health Status Detection

Temperature sensors alone mounted on the birds are not enough to detect if a bird has avian influenza or not. Temperature sensors in collaboration with accelerometers were used to detect both the temperature and the activity of the birds [21] [22]. If the activity of a bird happens to decrease, it is assumed to be in an abnormal state. This way, chickens with the highly pathogenic avian influenza were detected even before their temperature began to rise. The problem with this method is that it requires mounting sensors on every bird and that proves to be very expensive for large flocks.

Furthermore, Image technology was also analysed in the same paper [15] as follows:

a. Image Analysis
   Image analysis was used to assess the activity of broiler chickens with different gait scores [23]. The gait score system is an estimate of locomotion deficiency and it is based on the visual judgement of the ability of a broiler chicken to walk on a known surface [24]. The gait score can show the bird's activity and hence its welfare.
b. Infrared Thermal Imaging
   It is important to prevent heat stress in the birds for their welfare since it affects their behaviour, feeding and most of their physiological processes. Infrared thermal imaging can be used to measure the temperature of the birds by showing the body images with superficial temperature distribution. Thermal imaging was also used to determine the optimal air velocity for the thermoregulation of broilers while maintaining adequate temperature and relative humidity in the poultry farm [25].

The technologies discussed above focus on the welfare of the birds which is proved to affect the overall profitability of poultry farming. Most of the technologies discussed require mounting sensors on the individual birds which introduce extra expenses. Some of them focus on monitoring the environment when the birds live in since it affects the welfare of the birds directly as well. This research has its main focus on monitoring the environment as well. Below is a discussion on why it is important to monitor the rearing environment of the birds.

The thermal comfort of the birds is determined by the Temperature–Humidity Index [26]. It is a linear combination of the dry-bulb temperature and the wet-bulb temperature. The Temperature-Humidity Index determines the heat stress level in livestock [6]. Heat stress is a condition where the bird's body is under stress from overheating and it fails to thermoregulate. It causes heat-related illnesses and its symptoms range from dizziness, limited movement and drowsiness. Heat stress causes reduced bodyweight of the birds and causes a drop in the quantity of the eggs and quality of the eggshells. An increased level of ammonia in a poultry farm also causes respiratory diseases [17] and this can increase mortality in the birds in addition to contributing to the poor growth of the remaining birds. Poor lighting is another issue that affects food intake of the birds while the same light during the resting periods prevents the birds from resting properly [18]. The light, therefore, has a significant effect on the general welfare of the birds.

Controlling the rearing environmental to prevent these unfavourable climatic conditions avoids losses in poultry farming. There have many approaches in controlling the environment in poultry farms. Controlling the environment ensures that the birds live in a comfortable environment for their optimal growth.

Temperature

The temperature in poultry farms can be higher than required, at which point cooling may be required and it may also be lower than required at which point heating is required. Many areas experience daily fluctuation of cases where either heating or cooling is required. Different approaches to heating and cooling have been used.

1. Heating
   Heating can be achieved differently but the main types of heaters used in poultry farms are space heaters and radiant brooders [27]. Space heaters, radiant heaters or a combination of both can be used. Space heaters work by heating the air in the poultry farm while radiant heaters may heat the air to some level but they mainly work by heating the birds and the floor directly. It was discussed that when the emphasis in on chick comfort up to about two weeks of age, radiant heating is very important while for older broilers, space heating works since the check would have learned to manoeuvre in search for a comfortable temperature [27]. Space heaters, however, are more economical than the radiant heaters. The location of the heaters and the sensors to control these heaters becomes an issue in poultry farms. The common practice is to distribute the heaters evenly in the room to provide nearly equal heat to all parts of the room. Fans are also coupled with the heaters to help in distributing the hot air in the room. It also is shown that vertical stratification occurs where hot air is concentrated near the ceiling and not at the birds' level. Ceiling fans are used to overcome this. This study, therefore, tries to leverage the low cost of space heaters while providing more efficient heating by providing a better positioning of the heaters and sensors and the use of fans.

2. Cooling
   As discussed earlier, when the temperature is higher than required, the birds are more susceptible to heat stress which can drastically affect production in the birds. To mitigate this problem, there is a need for providing methods to reduce the temperature to the required level. In poultry farms, cooling is typically done in one of two methods which are tunnel ventilation and evaporative cooling [28]. Tunnel ventilation is a type of exhaust systems. It works by drawing air from one end of the poultry house to the other with the use of exhaust fans [29].

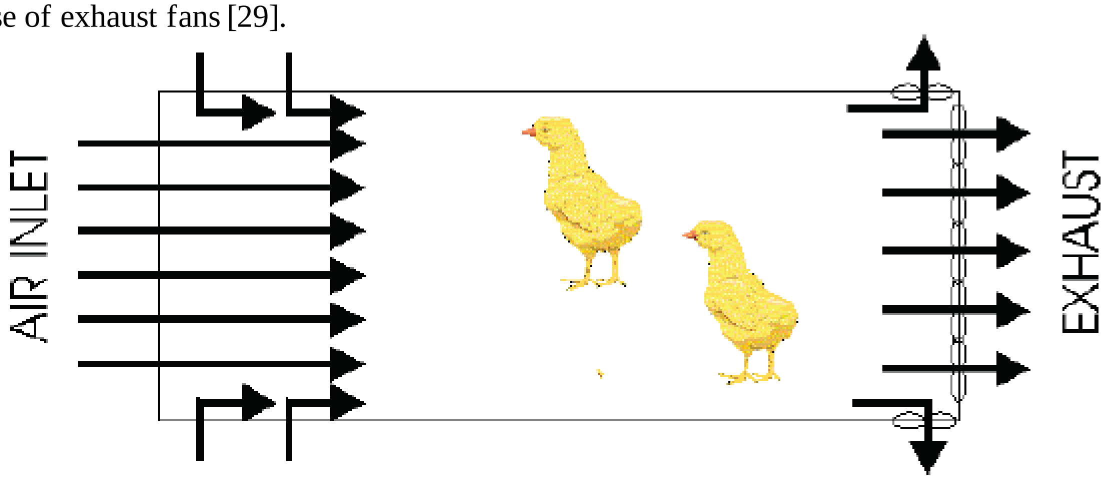


Figure 1: Tunnel Ventilation

The air moves over the birds' bodies and heat is removed from the birds making the bird feel cooler. The rapid movement of air as caused by tunnel ventilation results in even more cooling of the birds' bodies but it does not work effectively when the temperature is very high. At that point, evaporative cooling is required.
Evaporative cooling works by evaporating water into the air hence removing heat from the air and decreasing its temperature. There is two commonly used evaporative cooling system in poultry housing [30]. These are:

- Pad and fan system
- Fogging or misting system

Pad and fan systems work by using exhaust fans to draw air into the house through a moist porous pad. As the air is passing through the moist pad, the water evaporates off the pad using the heat from the air hence reducing the temperature of the air. Fogging or misting systems work by spraying fine water droplets into the air. As these water droplets float in the air, they evaporate using the heat from the air hence reducing the air temperature. To improve the efficiency of these systems, there is a need to maximize the amount of air which comes into contact with the water droplets.

In this study, the evaporative cooling was used placed at the air inlet hence removing the effort of modelling cooler positioning in the poultry house.

Heating and cooling of the poultry house as explained above can only work if there is knowledge of what the current temperature is at any point in time. This works hand in hand with the humidity, which calls for humidification or dehumidification. The information on the current temperature and humidity is provided by sensors which can read these accurately. This can be achieved in different ways. The study also focused on the monitoring of these environmental parameters in real-time and remotely to provide farmers with the necessary information whenever needed.

A sensor is a device, module or subsystem that detects states or changes of a physical phenomenon in its environment and transform it into an electrical signal which can be read by other electronics [31]. There are different ways to classify sensors. They may be classified based on physical laws or convenient distinguishing properties [32]:

Active and passive sensors: If a sensor requires external power to operate, then it is an active sensor. Active sensors are also called parametric sensors since their output is a function of a parameter i.e. resistance. A sensor that functions with its own generated electric signal is called a passive sensor. They are also called self-generating sensors.

Contact and Non-contact sensors: A contact sensor requires physical contact with the stimulus e.g. temperature sensor while a non-contact sensor requires no physical contact with the stimulus e.g. infrared thermometer.

Sensors may also be classified based on the broad area of detection: Electric sensors, magnetic, electromagnetic, acoustic, chemical, optical, heat, mechanical, radiation, biological etc. They may also be classified based on physical laws: Photoelectric, magnetoelectric, thermoelectric, photoconductive, photomagnetic, thermomagnetic, thermooptic, electrochemical, magnetoresistive, photoelastic etc.

Extending the concept of a sensor is the smart sensor. A smart sensor is a sensor with the added functionality of signal conditioning, embedded algorithms and digital interfaces to pre-process and format the data it collects using built-in compute resources before passing it on [33]. An example of a smart sensor is the ultrasonic sensor. It can be used to measure the distance from the sensor to an object with high accuracy. It emits a high-frequency sound which reflects and goes back to the sensor when it hits an object [34]. Since the speed of the sound is known precisely, the sensor calculates the distance by multiplying the speed with half the time it took the sound to travel from the sensor and back. This is the pre-processing that makes the ultrasonic sensor a smart sensor.

When data is collected from the sensors, it is sometimes required to be sent to another device. It may be another IoT device, a gateway, the cloud or an offsite database. Depending on the proximity of the other device, different methods, techniques and channels of communication are employed. For short-range communication, several alternatives exist; (a) 6LoWpAN which is an IP-based standard internetworking protocol that can connect directly to another IP network without intermediate entities like translation gateways and proxies [35], (b) ZigBee, (c) BLE also known as Bluetooth Smart which is designed for short-range, low bandwidth and low latency, (d) RFID and (e) NFC.

For long-distance communication, several options exist as well; (a) Cellular technology, which is a great fit for applications that have a high capacity power source, require operation over long distances and need high data throughput. It takes advantage of GSM/3G/4G and now 5G. (b) SigFox, which is a low power wireless communication technology. It can transport small amounts of data for up to 50 kilometres. Unlike cellular technology, SigFox is designed to handle low data speeds but can run on a small battery. (c) LoRa, a low-power and low-bitrate wireless communications technology, offers a long range of 2-3 km in urban areas and 3-5 km in rural areas. The data rate can reach to 50Kbps [36] and it uses license-free radio frequency bands which makes it accessible to anyone free of charge.

Using the above-discussed technologies, data is transferred from one device to another using standard protocol. The four most used messaging protocols for IoT are CoAP, MQTT, AMQP and HTTP. HTTP uses a request/response architecture, MQTT uses the publish/subscribe architecture while CoAP and AMQP use both.

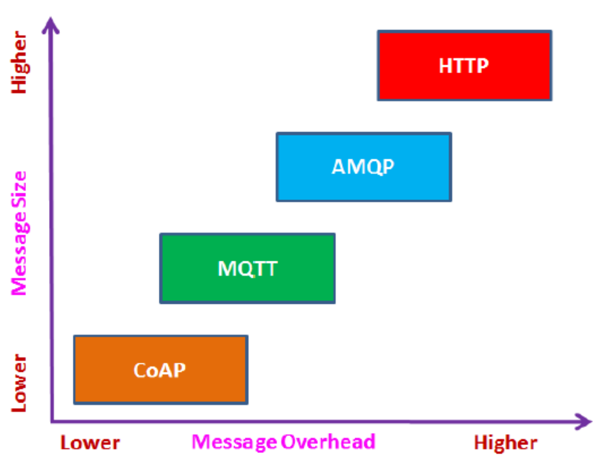


Figure 2: Message Size vs Overhead

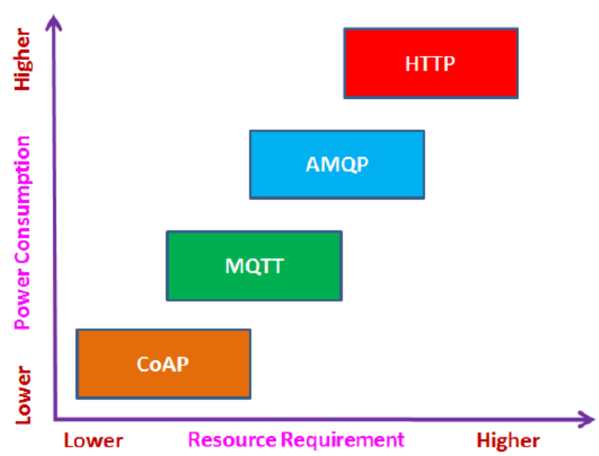


Figure 3: Power Consumption vs Resource Requirement

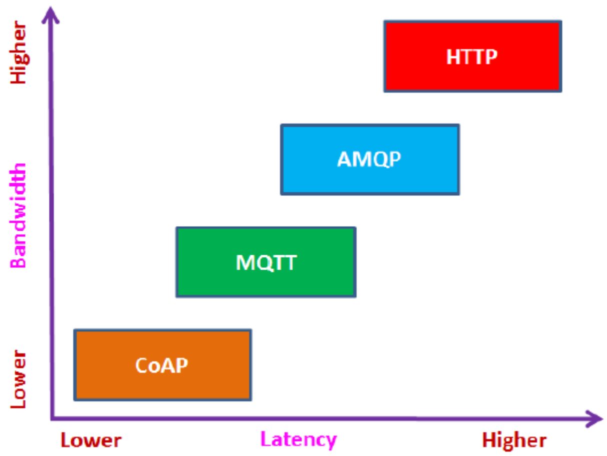


Figure 4: Bandwidth vs Latency

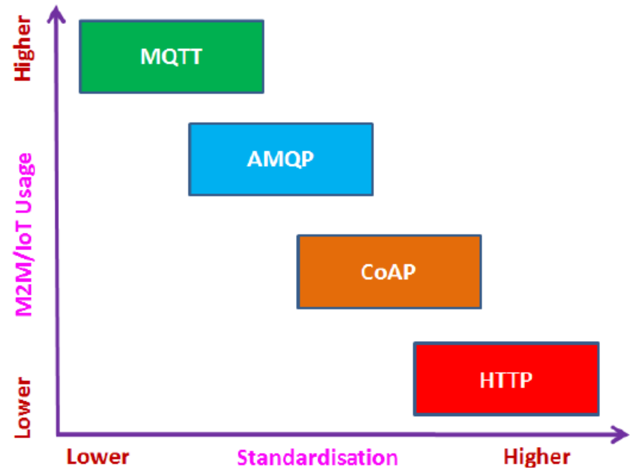


Figure 7: M2M/IoT Usage vs Standardization

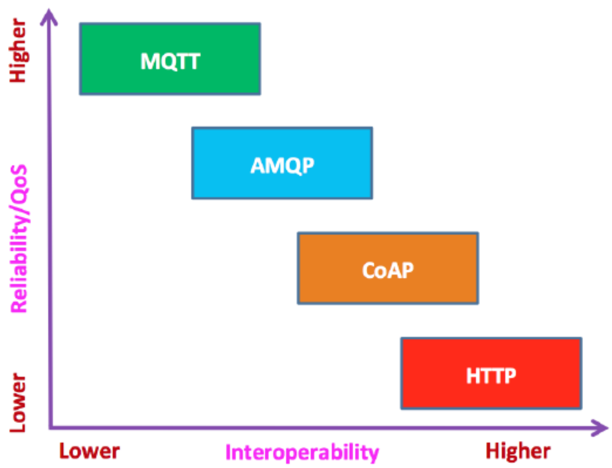


Figure 5: Reliability/QoS vs Interoperability

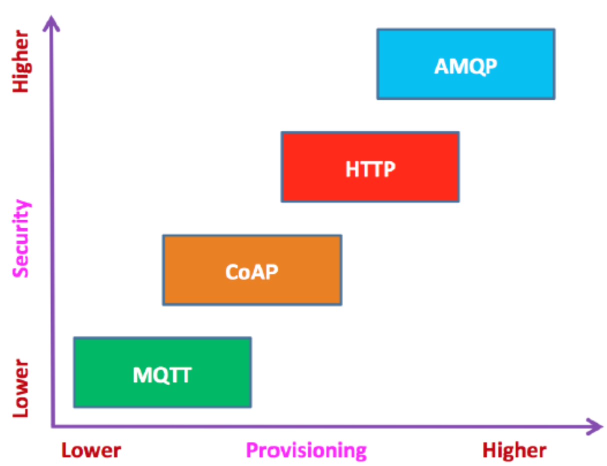


Figure 6: Security vs Provisioning

From Figure 2, Figure 3 and Figure 4 it is clear that CoAP is the most favourable, followed by MQTT [37]. However, MQTT has a better latency from Figure 4**Error! Reference source not found.** making it more suitable for real-time communication. From Figure 7 it is also clear that MQTT is the most widely used in IoT hence making it the most favourable for real-time monitoring in IoT devices. The main difference between CoAP and MQTT is that CoAP is mainly designed for one-to-one communication while MQTT is designed for many-to-many communication. This makes MQTT even more favourable for use in a wireless sensor network.

Having talked about the communication technologies and protocols to use when sending sensor data, a question that still prevails is where to position the sensors and how many of them to use to effectively and efficiently monitor the environment in the poultry farm. Sang-Yeon Lee et al. proposed a brute force method of finding the optimal locations of a specified number of sensors in a poultry house by trying out all different combinations of sensor setups and finding the one that is most representative of the average temperature [38]. Gradient descent was also used in another study to optimize the positioning of sensors in a general case [39]. These methods did not consider other sources of the physical phenomenon being monitored i.e. heaters. This study also looks at the optimization of sensor location depending on the location of the actuators affecting the monitored phenomenon. The information from these sensors is what is used to operate the actuators to control the respective parameters.

Heating and cooling use up 40% of the global energy use and there is potential to save up to 10% of this energy by using more efficient technologies [40]. The potential improvements include a better thermal performance of the buildings and the schedules of operation. Zhou Yu et al. looked at improving the energy efficiency of these HVAC systems through by improving the operational schedules [40]. It was shown that this method can save up to 20% of the energy on heating. What was not considered is the effect of humidity on heating and cooling which, as a hypothesis in this study, can save up more energy. This study, therefore, looks at developing an algorithm which, on top of scheduling the operation of the heating equipment, it can also use humidifiers and dehumidifiers in the control of temperature whenever it is predicted to be more efficient.

Some of the components to be used in this research are as explained below:

Actuators

After being detected by the sensors, it may sometimes be a requirement that the physical phenomena be acted upon as required by the specific application. An IoT actuator is the right device for that. IoT Actuators operate in the reverse direction of sensors [41]. It takes an electrical signal as input and the outcome is a physical activity which is reflected in its operating environment. There are three types of actuators and these are Electrical, Hydraulic (Use fluid to actuate motion) and pneumatic (Use compressed air to actuate motion) actuators. Electrical actuators are the most commonly used among the three [42]. Some of the physical phenomena that the actuators control, have interaction among each other. Temperature, humidity and air quality interact such that altering one of these conditions results in an alteration in the other phenomena as well [43]. This introduces the need for models which consider the interactions between different phenomena and how to operate actuators to control these phenomena with the utmost ease.

In Heating, Ventilation and Air Conditioning (HVAC) systems, the actuators usually have a high power-rating. That means they consume a high amount of electrical energy per unit time of their operation. To reduce the energy consumption of these high-power devices, it is important to limit the amount of time they spend operating. An energy scheduling algorithm that reduces the consumed energy in a particular time interval was proposed and it showed gains that can be achieved in both energy and cost [44]. This research proposes a scheduling model which takes into account the interaction between different physical phenomena i.e. Humidity, Temperature and Air quality in a poultry farm.

Controllers

For embedded systems, sensors and actuators need to be connected to a central device to function. This central device needs to have a processor to instruct the actuators to operate and to process the data collected by the sensors. Most applications use microcontrollers for this function. A microcontroller is a chip containing a small CPU (Microprocessor), memory, and basic I/O functions [45]. Generally, a microcontroller is loaded with a dedicated program which performs a specific task [46]. This operation mode is called "bare metal" since the program runs without an operating system and talks directly to the hardware. In more complex systems, a high-end microcontroller can run a specialized Real-Time Operating System (RTOS) which talks to the hardware on behalf of the programs and controls the scheduling of tasks in the system. Extending the concept of a microcontroller is the single-board computer (SBC). An SBC is a step up from microcontroller which allows you to attach peripheral devices like keyboards, mice and screens [47]. They offer more memory and processing power and some even contain graphic chips making them suitable for complex machine learning applications. Some applications require these single-board computers for faster prototyping and their ease to work with. Other applications require some special piece of hardware i.e. graphics card, which is not present in most off-the-shelf microcontrollers hence the need to work with single-board computers.

Gateways

In most IoT applications there is a need for various devices to have a connection to the cloud. These devices have diverse communication interfaces. Some devices lack the resources to connect to the internet and there is a need for an extra device to sit in between the internet and these devices to enable communication and for more reliability [48]. These devices are called gateways. They support different methods of device connection and consolidate data from disparate sources and interfaces then bring them to the internet hence mitigating the great variety and diversity of devices.

Embedded Intelligence

The data collected from the sensors have several uses. Depending on the application, some of the data is analysed in the cloud where insights are drawn and presented to the users. Sometimes the data must be analysed on-site, in real-time, for immediate action by the device. The processes of including this intelligence in the IoT device is called embedded intelligence. It enables the device with the ability to reflect on its operation performance, usage load and its environment, to enhance the product performance and lifetime [49]. In a poultry farm control system, this would be useful to detect anomalies in the system's performance and alert the system's users on when to intervene and when to perform system maintenance. It may also be useful in the control of actuators to make the system more efficient.

3. RESEARCH METHODOLOGY

This section of the thesis document serves the purpose of clarifying the methods employed in the conducted research and why the methods were favoured over their alternatives. It shows how these methods were used to achieve the objectives of the research, the obstacles in collecting and analysing data, as well as the impacts of unexpected challenges during the research.

3.1 Introduction

Keeping poultry birds in a controlled microclimate has proved to increase production in both small-scale and large-scale poultry farming. For small-scale farmers, however, operating the control systems for the microclimate introduces huge financial constraints due to the upfront costs, operating costs and maintenance costs of the system. Several methods have been employed in trying to reduce these costs and this research sought to design an intelligent control system for poultry farms in a way that it is efficient in its operation hence reducing the costs of operation and maintenance. The hypothesis was that, by achieving the objectives outlined in the study, a poultry farm control system suitable to small-scale farmers would have been designed due to its low operation and maintenance costs. This section presents how this research problem was solved.

Each specific objective of the research addressed a problem that when solved and combined with the solutions from the other specific objectives, the main aim of this research would have been achieved. The specific objectives were:

a. To design an embedded control system for a poultry farm environment.
b. To model the sensor placement for a poultry farm.
c. To model the operation schedule of actuators to reduce energy consumption
d. To design a web-based data dashboard for farm data access and analysis.

This section describes how each of the above-stated objectives was achieved, describing thoroughly the steps taken and how the parts connect to achieve the main objective, where the whole is greater than the sum of its parts.

3.2 Sensor Placement

In smart poultry farms, the need for cooling or heating the farm is typically determined from the readings from the mounted temperature sensors. It is not clear on how temperature is distributed in the farm to determine where to heat, where to cool and where not to act. The same thing applies to the humidity distribution which is shown to have a relationship with temperature. This problem introduces another problem of determining the number of sensors to deploy on a farm and where to position them. Currently, most poultry farms use one sensor each for temperature and humidity positioned at the centre of the farm. The temperature and humidity at the centre of the farm are assumed to be universal for the whole farm but that has not been proved to be true. If the assumption is wrong, the worst-case scenario would be heating, cooling, humidifying or dehumidifying zones in the farm, that have the right temperature and/or humidity. That would result in a wastage of the energy used by the actuators and general ineffectiveness of the HVAC system. This objective was aimed at finding the strategic positioning of temperature sensors in a poultry farm depending on its size and the position of heaters.

To determine where a temperature sensor needs to be placed and how many of them to deploy in a particular poultry farm, it is necessary to know how temperature is distributed on the farm. The plan on how to achieve this objective was as outlined below:

a. Find the temperature distribution in a small room with no artificial heat sources
b. Examine temperature distribution in a room with artificial heat sources
c. Determine locations for the artificial heat sources
d. Model sensor placement dependent on the temperature distribution considering both natural and artificial heat sources.

3.2.1. Data Collection

This study was conducted during the tough time of COVID-19 when too many measures were put in place restricting access to movement and laboratory access. It was a time when getting access to working equipment was almost impossible. This study was therefore conducted from a small room which was considered to have the same characteristics of a poultry farm save for the absence of the poultry birds. Poultry birds have varying space requirements depending on the species and breed. The Food and Agriculture Organization of the United Nations (FAO) recommends that broiler chickens be put in groups of 4 to 5 birds per square meter. There is no standard size for a poultry farm and any size is recommended for as long as it follows the recommended number of birds per unit area. [50]. The room from which the study was conducted was 3 meters long, 3 meters wide and 4 meters high (3m × 3m × 4m). That accounted for 9 square meters of floor area and according to the FAO requirements stated previously, this room would house 36 to 45 broiler chickens. This was relatively small compared to other poultry farms in literature which can get up to 1386 square meters (110m × 12.6m) [51]. Much as there were not many choices for the sizes of rooms to conduct this study from, the small room would still be the preferred choice for the study so that the temperature distribution from a small area could be studied and then scaled up to consider bigger poultry farms. Since the aim was to find the natural temperature distribution in the room, there was no need to manipulate any variable hence this was merely an observational study.

The collection of primary data was chosen over secondary data for the following reasons:

a. The search for secondary data to match the requirements for this study was not successful.

b. When using primary data, it was possible to control what variables were to be collected and the format in which they were to be stored.

Room Setup

To correctly check how the temperature was distributed, the room was horizontally divided into four equal zones. The zones were 1.5m × 1.5m each as shown below:

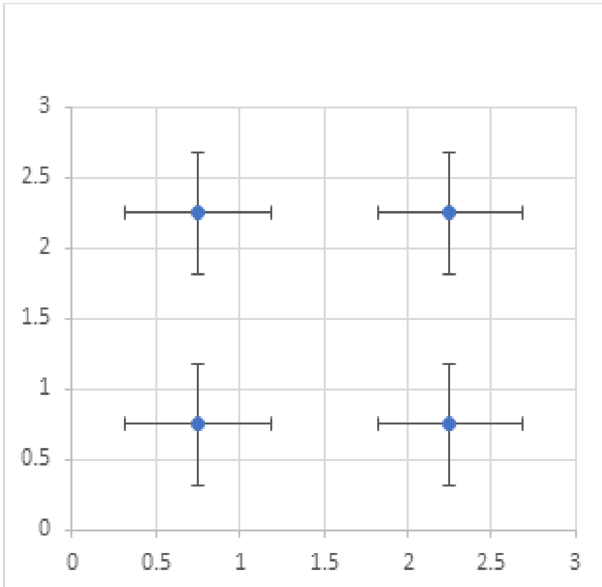


Figure 8: Room Setup

The centre of each of the zones had a temperature and humidity sensor as shown in the figure above. The room was set up in this way to collect the temperature and humidity data for each of the zones and compare among the zones to find their difference and, therefore, determine the temperature distribution of the room.

Having the room set up in this way, a device was designed to connect with the sensors, request for the temperature and humidity data, format the data in the required way, then upload the data to a server where they were to be stored for later analysis to finally determine what the temperature distribution was like. The following section discusses the data collection device.

Data Collection Device

The first decision affecting constraint met during the design of the device was the financial constraint. There were no funds available for the research, therefore, the device had to be designed using available resources in such a way that the extra resources to be purchased were minimal. This section provides details on what resources were used, why they were used, any alternatives that could have been used and a comparison between what was used and its alternatives.

Microcontroller Unit

This is the part of the sensor node that is responsible for reading data from the sensors, processing the data and communicating with other devices in the network using a connected networking module. To collect the required data, a NodeMCU V2 microcontroller was used. NodeMCU (as shown in the figure below) is a development kit based on ESP8266 [52].

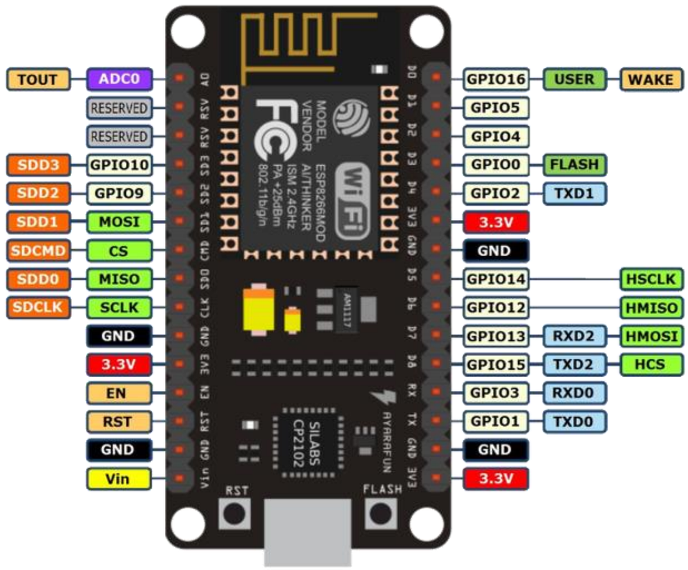


Figure 9: NodeMCU

ESP8266 is a Wi-Fi Microcontroller Unit which includes a 32-bit processor and it is integrated with the full TCP/IP stack [53]. It has a power-saving architecture and it is highly durable due to its wide range of operating temperature. The high range of operating temperature made it suitable for this study since the purpose was to read the temperature in different ranges. The integrated Wi-Fi capabilities made it the right choice since most of the readily available resources for this study had Wi-Fi capabilities. The NodeMCU development kit can be programmed using various development environments and programming languages. It also has a large community of developers which makes it easy for a developer to find solutions when they run into problems during prototyping. Since the prototype required four temperature sensors and four humidity sensors (8 sensors in total), the ten GPIO pins provided by the NodeMCU were more than enough. It will be shown in the next section that the prototype ended up using only four pins for all the temperature and humidity sensors.

It has to be stated, however, that the NodeMCU was not the only microcontroller that could be chosen to achieve the task. Some of the alternatives are discussed next:

a. Arduino Uno Wi-Fi
   Arduino Uno Wi-Fi is an Internet of Things development board based on the ATmega328P microcontroller unit and it is integrated with the ESP8266WiFi module for its Wi-Fi capabilities [54]. In effect, this development board has two microcontroller units where one is used for accessing sensor data and processing them while the other one is used for establishing a network connection, then communicating the data from the first microcontroller unit to other devices in the network. This introduces complexity in the system and redundancy which is not available in the NodeMCU since it performs all the functions from a single microcontroller unit.

b. Raspberry Pi
   Raspberry Pi is a single-board computer the size of a credit card. There are many versions of Raspberry Pi but in this review, Raspberry Pi 3 B+ is focused on. It is recommended to run the Raspberry Pi with the Raspbian Operating System which is a Linux distribution

based on Debian. However, there are several other operating systems which can run on the computer. It has a Quad-Core 1.2GHz Broadcom BCM2837 64bit CPU, 1GB RAM, Bluetooth, Wi-Fi, 40 GPIO pins, 4 USB ports and a 100 based Ethernet port. With these specifications, the Raspberry Pi is far more superior to the NodeMCU but this superiority comes at a cost. The Raspberry Pi is more expensive than the NodeMCU and it uses more power. While the Raspberry Pi could accomplish the task just as the NodeMCU did, the operation would have cost more without any effect on the results.

With the above comparisons, NodeMCU was the most suitable development board to use for the task of data collection.

Sensors

To gather the temperature and humidity data, there had to be sensors. A sensor is a device with the ability to detect a physical phenomenon then turn it into an electrical signal. There are many choices for temperature sensors to use with the NodeMCU. In the same way, there are many choices for the humidity sensor. The DHT AM2302 was used for data collection in this study. DHT AM2302 is a digital temperature and humidity sensor [55]. It is made of three parts, a capacitive humidity sensor, a thermistor and an Analog to digital converter that outputs a digital signal with temperature and humidity values. This sensor was favoured over the alternatives since it bundles a temperature and humidity sensor in one package hence requiring only one GPIO pin to get both values. That reduces the number of devices to power and makes the device less clouded with connections. The NodeMCU, on the other hand, has only one analog pin, therefore, to have four similar sensors collecting the temperature data, they needed to be digital, else, there would be the need of having either an analog to digital converter to convert the signal to digital, or a multiplexer to connect all the sensors to one analog pin. This means having the sensors as analog, would introduce the need of having extra components to the data collection device, which would increase the complexity of the device and defeat the initial plan of designing a device using minimal resources. The downside of the DHT AM2302 is that it is slow. It records data at a frequency of 0.5 Hz (once every 2 seconds). For this data collection task, however, it was appropriate to use it since the data was sampled once every 15 seconds as will be discussed.

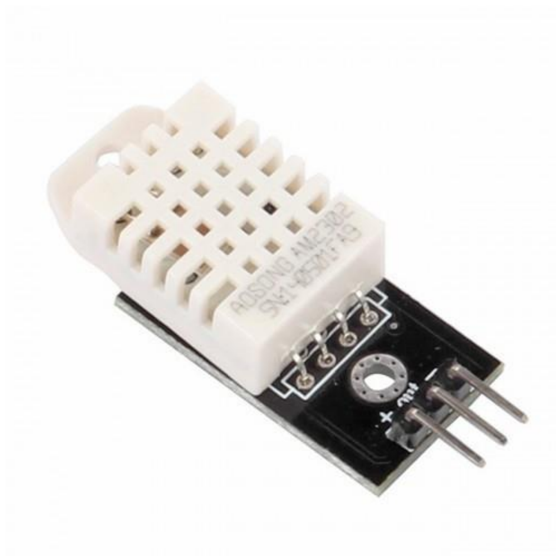


Figure 10: DHT Sensor

Server

After collecting the temperature and humidity data from the sensors, the microcontroller had to send the data to another device for storage. For this task, there had to be a webserver to handle the network traffic and a database to store the data. There were several options for the webserver to use but the two most apparent according to their popularity were the Apache and Nginx (Pronounced as Engine X) servers. The main difference between Apache and Nginx is in their design architecture. Apache runs every process in its thread while Nginx is event-driven and handles every request in the same thread. These servers are both open source and in today's web, most applications use both these servers in complement to each other. This data collection task was not heavy since the sensor node was sending data once every 15 seconds and it was a one to one connection. Therefore, there was no difference to the task's perspective in using any one of these two web servers. Apache Server was used in this task mainly because it was readily available and already set up.

Database

The web server had the task of handling web traffic then passing it to the respective programs to process the data from the requests. This data had to be persisted in somewhere for later usage. The data could be persisted in a flat file or a database. The requirements for this task needed multiple users to be accessing the data simultaneously i.e. The data collection device when persisting the data and the end-users to monitor the progress of data collection in real-time from the server. Flat files are not designed to handle multiple user access to the data therefore, a database was the apparent choice henceforth. For the task of persisting the data, there were several options for a database management system. A database management system is software for storing and retrieving data from a database. It includes several programs that help users create, read, update and delete data from a database by instructing the operating system to do so as per users' requests. The main decision, however, was on the type of database management system to use. There are several types of database management systems:

a. Hierarchical Database
   The hierarchical database model is a data model where data is arranged in a tree-like structure. The data is represented using a parent-child relationship and a parent may have many children while a child can only have one parent. It is best to use the hierarchical database if the data can be represented in a tree-structure and high performance is of paramount importance.

b. Network Database
   The network database model also allows data to be modelled in a hierarchical structure. The main difference with the Hierarchical database model is that the network model allows for a child to have many parents. In this model, entities are represented as a graph and they can be accessed using many paths. It is best for applications where the entities have many-to-many relationships.

c. Relational Database
   The relational database model represents data as a set of relations which are tables and every row represents a collection of related data values. These tables are related to each other using special keys. The relational database model is by far the most widely used database model hence it has a large community of developers solving any problem faced by other developers.

d. Object-Oriented Database
   The Object-Oriented Database model represents data as objects which are instances of classes. It stores both the member values and the operations of the objects. It allows the definition of different types of relationships between two or more objects. It is best to use when there is a business need for high performance on very complex data. This model of databases, however, is not popular among developers. That introduces a hurdle to developers when debugging during development since information on bugs in this model is not widely available.

The Relational database model was chosen over its alternatives for this task mainly for its simplicity and the simplicity of the current task which required neither complex data structure nor high levels of performance. The MariaDB relational database was used to persist the data. It is an open-source fork of the popular MySQL database. The MariaDB database was chosen over the MySQL database since MySQL uses some proprietary code in its enterprise edition while MariaDB is completely open-source.

The database schema was as below:

Table 1: Database Schema

Field	Type	Null	Key	Default	Extra
id	bigint(20) Unsigned	NO	Primary	Null	auto_increment
temp1	double(8, 2)	NO		0.00	
hum1	double(8, 2)	NO		0.00	
temp2	double(8, 2)	NO		0.00	
hum2	double(8, 2)	NO		0.00	
temp3	double(8, 2)	NO		0.00	
hum3	double(8, 2)	NO		0.00	
temp4	double(8, 2)	NO		0.00	
hum4	double(8, 2)	NO		0.00	
collected_time	timestamp	NO		Currrent_Timestamp	

The database only had the above relation and the data from all the sensors were inserted at once with the same collected time for easy comparison during analysis.

Embedded Software

To be able to read the data collected from the sensors, format them and upload them to the webserver for persisting in the database, there had to be an embedded software instructing the microcontroller to perform the task with clearly detailed procedures on how to perform them. There were several options for the computer language to use when programming the NodeMCU just as there are many options for the development environment to use. The most common go-to languages for programming NodeMCU are Lua, C and C++. C has been popular in embedded software development for a long time for its high performance since it is low-level. Lua on the other hand is also widely used due to its fast learning curve and efficient memory usage. C++, however, takes all the advantages of C, adds control over memory allocation, and introduces object-oriented programming, templates and generic programming. The downside of C++ is its slow compilation. C++ was chosen for the task of developing embedded software for data collection. This was mainly due to a large number of libraries to work with and the object-oriented abilities which made troubleshooting easier due to its modular nature. The embedded system was designed as below.

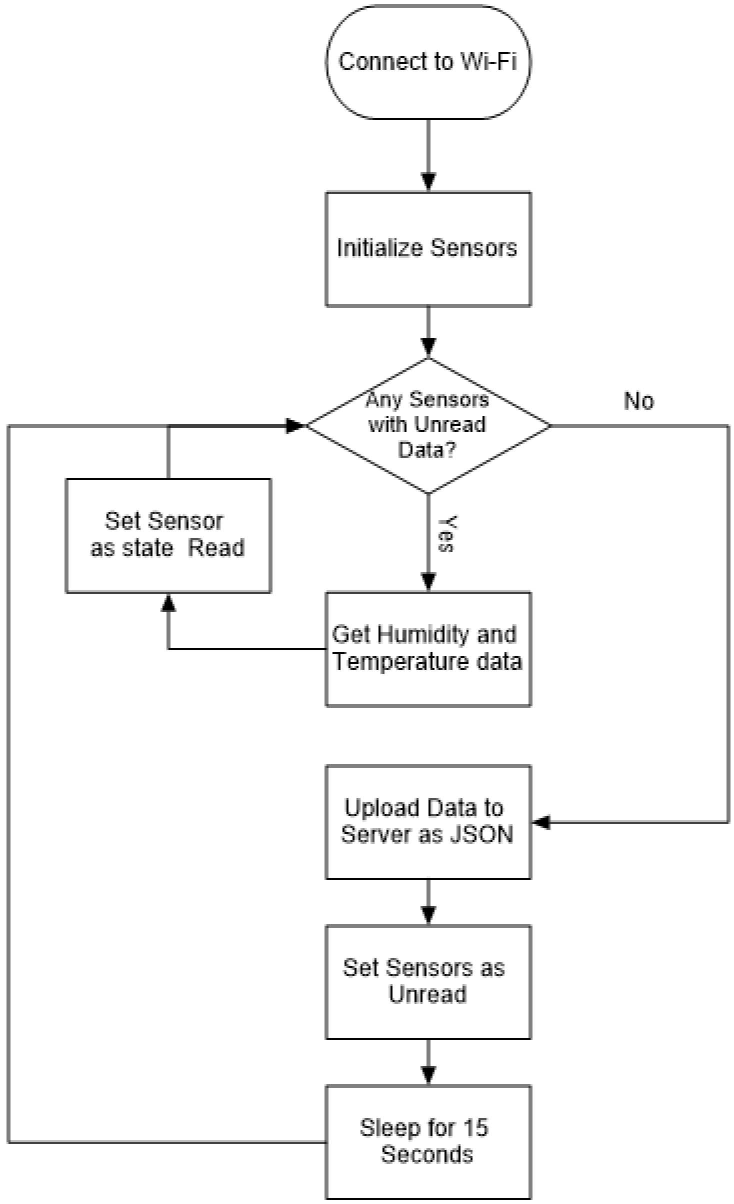


Figure 11: Embedded System Flow Chart

Application Programming Interface

An application programming interface is a software intermediary that allows two or more applications to communicate with each other [56]. It defines the conventions to follow, the data formats to be used, the kinds of calls or requests allowed and how to make them. There are several types of APIs depending on their architectural styles. Below are some of the types of APIs [57]:

a. REST

REST is an acronym for Representational State Transfer. It is an architecture that separates the concerns of the API consumer from the API provider by exposing its functionality through the built-in commands of the underlying network protocol. The client and server apps are completely decoupled in this architecture such that a client or server program would change without affecting the other and all they have to conform to is the data format.

It is a stateless protocol and communication are done by the client initiating a request and the server replying with a response.

b. RPC

RPC is an acronym for Remote Procedure Calls. They involve clients invoking procedures in other systems by name. They are protocol-agnostic, meaning they have the potential to be supported on many protocols by they lose some of the benefits of using native local protocols. The examples of APIs that use the RPC architectural pattern are SOAP, JSON-RPC and XML-RPC. The main drawback of this protocol is that it requires the calling of procedures by name hence introducing high coupling which introduces problems when one of the programs has to be changed. RPC is best for APIs that are mostly doing actions while REST APIs are best for CRUD operations on data.

c. Event-Driven/Streaming

Also referred to as evented, real-time, asynchronous, streaming or push architectures, Event-driven APIs act on events instead of waiting for requests from an API consumer. Clients can subscribe to events and when they occur, the API responds to the clients. They are best used for real-time applications where the occurrence of events cannot be predicted and responses are required the moment events occur.

For this task, a RESTful API was used since the main operations required were persisting data, and reading data which are merely CRUD operations. There was no direct need for real-time requesting of the data as this data collection was collected locally and the server console was available for checking if the data collection process was proceeding as required.

The following were the API resource endpoints exposed:

a. Create Resource

```
Post /api/v1/data HTTP/1.1


data= {
"temp1": 25.5,
"hum1": 60.3,
"temp2": 25.5,
"hum2": 66.5,
"temp3": 27.5,
"hum3": 61.5,
"temp4": 25.8,
"hum4": 65.9,
}
```

The create resource endpoint accepts a JSON (JavaScript Object Notation) string with the values for all the sensors bundled together with the request with a content type of "application/json". The API calls an SQL query to insert the data to the database, extracting all the variables from the JSON string then passing them to the query. The database automatically creates an ID (Primary Key) for the data and adds a collected_time column to correspond to the time of insertion to the database.

b. List Resource

GET /api/v1/data HTTP/1.1


```
{
        data: [{
"id": 1
"temp1": 25.5,
"hum1": 60.3,
"temp2": 25.5,
"hum2": 66.5,
"temp3": 27.5,
"hum3": 61.5,
"temp4": 25.8,
"hum4": 65.9,
"collected_time": "2020-06-10 21:37:35"
},
{
"id": 2
"temp1": 25.5,
"hum1": 60.4,
"temp2": 25.5,
"hum2": 66.5,
"temp3": 27.5,
"hum3": 61.4,
"temp4": 25.8,
"hum4": 65.8,
"collected_time": "2020-06-10 21:37:50"
}
]
}
```


The resource listing endpoint calls an SQL query to extract all the rows from the database table and formats the data in a JSON string containing an array of other JSON strings which represent the data from each row of the database table. The data are then returned to the client with a content type of "application/json". This endpoint was provided solely for the purpose of extracting the data from the database for data analysis after finishing the data collection process.

3.2.2. Overall Architecture

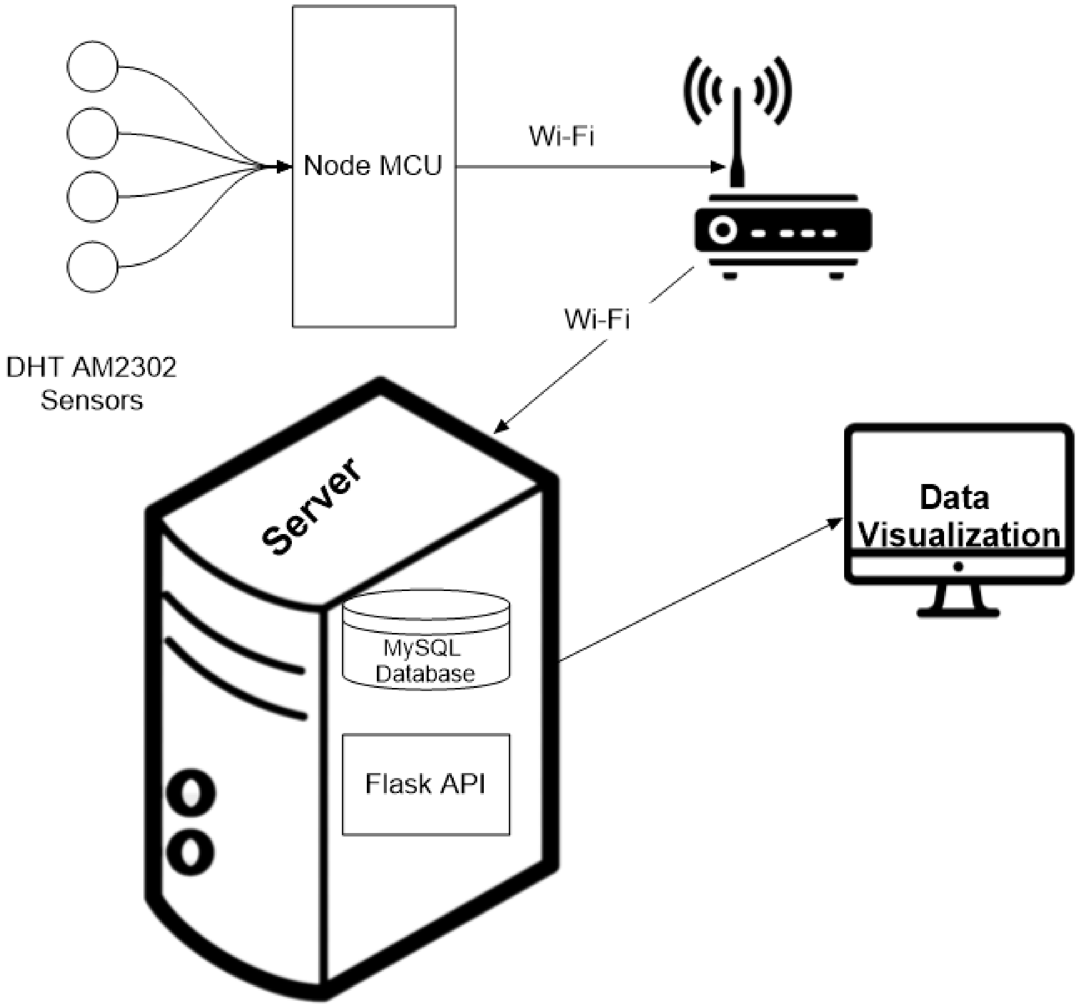


Figure 12: Data Collection System Architecture

3.2.3. Data Analysis

The data was exported from the database in a CSV (Comma Separated Values) format ready for analysis. The data had 24134 rows, each of which had 10 columns. Below is the representation of the data with the first five rows where *temp1* and *hum1* represented temperature from sensor 1 and humidity from sensor 2 respectively:

	id	temp1	hum1	temp2	hum2	temp3	hum3	temp4	hum4	collected_time
0	1	25.5	60.3	25.5	66.5	25.7	61.5	25.8	65.9	2020-06-10 21:37:35
1	2	25.5	60.3	25.6	66.6	25.7	61.6	25.8	65.9	2020-06-10 21:37:50
2	3	25.5	60.3	25.5	66.5	25.7	61.6	25.8	66.0	2020-06-10 21:38:05
3	4	25.5	60.4	25.5	66.5	25.7	61.6	25.8	66.0	2020-06-10 21:38:20
4	5	25.5	60.4	25.6	66.6	25.7	61.6	25.8	66.0	2020-06-10 21:38:35

Figure 13: Data Representation

The main aim of the data analysis was to examine how the data from the four zones compared with each other. To achieve that, numerical methods were to be applied since the data was in numbers and the anticipated result was the difference in temperature and/or humidity from zone to zone which can be represented numerically.

Data Preparation

Before starting to compare the data from the zones, it was necessary to prepare the data to make sure there were no missing values or any data that can affect the correctness of the final results of the analysis. Looking back to the data collection, the data from all the four sensors were uploaded to the server at the same time but there was a possibility of sending the data while one or more of the sensors were not working. In such a scenario, the data collection device was programmed to send zero values. If these values were left like that during the analysis of the data, it would have been indirectly assumed that the temperature and/or relative humidity of the zone where the sensor was deployed were 0℃ and 0% respectively. These are called missing values and they needed to be taken care of if the results of the analysis were to be accurate.

To analyse the data, there were many options for environments and packages that could achieve the same results. Some of the options for the packages were: SPSS, STATA, SAS, Python, R, MATLAB, and Microsoft Excel. Each of these has its pros and cons. According to the nature of this study, it was necessary to use a package that provides an easy way of creating custom functions which would suit the nature of the data and the required results. The obvious choices considering that point were the R, MATLAB, SAS and Python.

R is a popular open-source environment that can be extended by other packages at will. It has very high library support and provides easy ways to visualize data. On the other hand, it has a very steep learning curve and it requires powerful hardware whereas the computer used for this task was not powerful.

MATLAB is a powerful environment that has elegant matrix support and provides great data visualization tools. It has many packages for various analysis use cases. The power of MATLAB comes at a cost since it is very expensive. MATLAB also suffers incomplete support of its statistics packages.

SAS is another environment that provides a different programming language for data analysis. It provides a fast integration of new statistical methods and has very stable and reliable routines which can handle large datasets very well. SAS is also well documented. The downside of SAS is that it uses an outdated programming language and it is very expensive.

Python is a full general-purpose programming language that is very popular due to its use in data science, web development and its wide use in deep learning and machine learning applications. It has many professional development environments, huge community support and supports unit tests and easy debugging due to its being a mature programming language. The downside of python is that it has a very high entry bar due to its being a full programming language. That did not introduce a problem in this study due to the programming experience available to the data analyst. Considering the comparisons made above, python programming language was chosen and among its many data analysis libraries, Pandas was used due to its focus on tabulated and time-series data.

To mitigate the effect of missing values, all the rows with a value of zero were truncated from the data. Only two rows were removed by the operation such that the remaining useful data were 24132 rows. The next step was to check for the maximum and minimum values for each sensor to get a clear picture of what the range of the data was. It was important to get the range of the data for a better understanding of how spread the data were. The same value of the maximum difference among sensors would be interpreted differently in data with different ranges. In data with a high range, the tolerance of data distribution to be considered as an insignificant difference would be larger since the values in a high range data have more degrees of freedom which increases the differences.

Correlation

The next step was to check how the sensor data from the different zones correlated with each other. To do this, the data from the exported CSV file were loaded in a Pandas DataFrame. Since what was needed was the temperature distribution and the humidity distribution, it was necessary to compute the correlation of the temperature values of all the sensors separately from the correlation of the humidity values of all the sensors. The columns were renamed to show the sensor numbers instead of “temp1” or “hum1” as it was in the collected data. This was done since labelling using the sensor number was more intuitive to the readers of the document than the initial column names.

Correlation is the measure of how two variables are associated [58]. If two variables are correlated, the change in the magnitude of one variable is associated with the change in the magnitude of the other variable either in the same direction (Positive correlation) or in the opposite direction (Negative correlation). The term correlation is mostly used to mean the linear relationship between two continuous variables expressed as the Pearson product-moment correlation. The Pearson correlation is mostly used in normally distributed data. The temperature and humidity data in this study had a normal distribution and that called for the use of the Pearson correlation coefficient. It is expressed as a number in the scale of -1 to 1, where -1 represents a 100% negative association of the variables, +1 represents a 100% positive correlation and a value of 0 represents the absence of any linear or monotonic association between the variables. Using the Pearson correlation coefficient as explained, all the temperature sensor values were compared against each other and so were the humidity sensor values. The coefficient matrix was obtained and more on it will be described in the Results Section of this document. This coefficient matrix showed that the temperature and humidity in the small room with no heaters was almost always equal in all zones and that the differences were insignificant as far as poultry farms are concerned. More on this will be discussed in the Results and Discussions section.

3.2.4. Introducing Heaters

The previous data was collected in a small room with no artificial heat sources. It showed that the temperature of the room was always uniform. It has to be noted, however, that poultry farms are designed in different sizes and most importantly, some poultry farms require artificial heat to maintain the recommended temperature on the farm. This knowledge introduced the need for finding the temperature distribution in a bigger room and the temperature distribution in a room with heaters. If the general temperature distribution was to be found, a general sensor placement algorithm could be developed which could be used to deploy sensors on poultry farms regardless of the farm size and number of heaters present.

To understand how to deploy sensors in a poultry farm with heaters, it was necessary to understand how heaters work, and the way they affect the temperature and humidity. This was achieved by first determining the amount of heat required to raise or drop the temperature of the farm. Below is a step by step calculation of the required heat.

$$c_x = a + bT + cT^2 + dT^3 \text{ [59]} \qquad (1)$$

Where

T is the temperature of x in Kelvins
c_x is the specific heat capacity of x.
a, b, c and d are constants dependent on x [60]
x represents air or water vapour

For air:
$a = 28.11$
$b = 0.1967 \times 10^{-2}$
$c = 0.4802 \times 10^{-5}$
$d = -1.966 \times 10^{-9}$

For Water Vapour:
$a = 32.24$
$b = 0.1923 \times 10^{-2}$
$c = 1.055 \times 10^{-5}$
$d = -3.595 \times 10^{-9}$

Once the specific heat capacities of water vapour and air are found they can be used to find the heat capacity of the moist air as follows:

$$C_{moist\ air} = c_{air} + (c_{water\ vapor} \times q) \quad (2)$$

Where
$C_{moist\ air}$ is the heat capacity of moist air,
c_{air} is the specific heat capacity of dry air,
$c_{water\ vapor}$ is the specific heat capacity of water vapour.
q is the specific humidity of the moist air calculated as follows:

$$q = 0.662 \times e°(T) \times RH / p \text{ [61]} \quad (3)$$

Where
RH is the relative humidity
p is the air pressure
$e°(T)$ is the saturation vapour pressure at temperature T (°C)

$$e°(T) = 6.1121 \times e^{(18.678 - \frac{T}{234.5})(\frac{T}{257.14+T})} \text{ [62]} \quad (4)$$

To calculate the energy required to heat a room, the mass of the air must be known

$$M_{moist\ air} = V \times \rho_{moist\ air} \quad (5)$$

Where
$M_{moist\ air}$ is the mass of the air,
V is the volume of the room,
$\rho_{moist\ air}$ is the density of the moist air.

$$\rho_{moist\ air} = \frac{p}{R_a \times T} \times \frac{1+q}{1+(q \times {}^{R_w}/_{R_a})} \text{ [63]} \quad (6)$$

Where
R_a is the individual gas constant of air,
R_w is the individual gas constant of water vapour,
T is the temperature in Kelvins.

Finally, we have everything needed to calculate the energy required to raise the temperature by 1°C.

$$E = C_{moist\ air} * M_{moist\ air} \quad (7)$$

To raise the temperature by k°C the heat required is:

$$H = kE \quad (8)$$

3.2.5. Different Farm Scenarios Based on Heaters

To evaluate the effect of heaters when deployed on farms in general, it became a requirement to evaluate all possible scenarios of poultry farms with respect to the heaters. This evaluation held for any arbitrary farm size. Below are the different scenarios as evaluated during the study:

No Heater

This case is similar to the experimented case where the data was collected. It represented all the poultry farms, regardless of size, that have no artificial heat sources deployed and the temperature is controlled from natural sources. To keep the evaluation controlled, an assumption was made that the farm is closed such that no section of the farm is lit with solar radiation since that could affect the temperature in such a section.

In this scenario, the results from the previous experiment were to be used to know the distribution of temperature and humidity and, hence, determine the number of sensors to deploy on the farm. The effect of the walls, roof and air circulation was then examined and the location to deploy sensors were proposed.

1 Heater

Most of the available poultry farms are located in areas where the environment is not always in the recommended ranges for the optimal growth of the birds. That is the primary reason why this research focused on the design of a poultry farm control system which will work to keep these environmental conditions in the optimal ranges. Heaters are one of the most important actuators when controlling the temperature on a poultry farm. Some farms only have a single heater for controlling the temperature on the farm. This may be due to the size of the farm, or the economic constraints which may not allow for more heaters. To make sure that the performance of the heater is optimized, it is best to place the heater at a place where it controls as much space on the farm as possible. As described in the previous section, this is the centre of the room and away from any ventilation holes and places of poor air circulation. Since the heater changes the temperature of the air in the room, it was also important to evaluate the effect of other objects that affect the temperature of the room. Since the temperature from outside the farm can affect the temperature inside the farm through the ventilation holes and the walls, it was also right to consider the ventilation holes and outside interfacing walls as the source of temperature change in addition to the heaters. Effectively, the true reflection of the temperature in the room was then found considering the three factors and that was the best location to position the sensors.

N Heaters

The deployment of N heaters (where N > 1) in a farm is more complex than the two previously discussed scenarios. This is mainly due to the introduction of several variables that affect where the heaters have to be placed for a good performance of the control system. In this study, the first factor to be considered was the way heat spreads. In a closed room with open air-circulation, the heat energy spreads out from the heater to the surrounding areas in a spherical way. Since all the heaters were deployed at the same height, it was only the horizontal plane that became variable with respect to the heat spreading model and had to be evaluated carefully. With that in mind, the heat spreading model was then remodelled to be circular (2 dimensional) on the horizontal plane and the vertical component was completely disregarded. This model was purposefully applied to optimize the placement of the heaters using the circle packing algorithm.

Circle Packing Problem

The circle packing problem is a notable mathematical problem in the field of computational geometry [64]. It studies the arrangement of circles of either equal or unequal radii in a boundary in such a way that the circles should not overlap and should cover as much area in the boundary as possible. There are different optimized arrangements of a limited number of circles in other circles, rectangles, triangles and squares. The problem has been around since 1985 and it has been mapped in different fields such as numerical analysis, complex analysis, product packaging and deployment of networking devices. Below is the densest of all plane circle packings in a rectangle which is a hexagonal lattice of the bee's honeycomb.

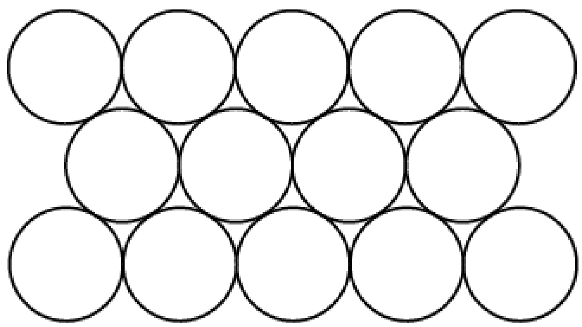


Figure 14: Hexagonal Lattice Circle Packing

In this study, it was shown how circle packing can be used to deploy more than one heater on a farm. To put it in perspective, the vertical component of the spherical volume affected by the heat from the heater can be disregarded such that the affected area can be perceived as a circular area on the horizontal plane. These abstract circles can then be packed in a boundary representing the horizontal dimensions of the poultry farm. The circles are made to be of the same radii to represent the equal power of the heaters. The problem can then be represented mathematically as:

Place $n \geq 2$ circles in a rectangle such that the common radius of the circles is maximal [65].

The best-known solutions to the problem can be found at [66]. If heaters in a farm are deployed in such arrangements, at the circle centres, then the heaters will have been arranged in the best-known way.

Sensor Placement Algorithm

After the heaters are placed optimally on the farm, the sensors can then be placed strategically. Firstly, the effect of each heater was considered individually. When deployed, each heater will control the temperature represented by the circle in the circle packing algorithm. If any heater is malfunctioning, it is the same area that will be affected, therefore, there needs to be a way of monitoring the temperature of each of these circular areas. If all the circular areas of the farm are in the required temperature ranges, it can be generalized that the whole farm is in the recommended temperature range since the circle packing algorithm guarantees that the largest possible area is covered by these circular areas. When the heater is in operation, there is more heat closer to the heater and it decreases the further away from the heater it gets. Since the circular areas touch each other, the tangent of the circles gets the least heat from each heater on the centres of the touching circles. In effect, however, it gets heat from several heaters therefore it detects the temperature of several circular areas. To get the temperature of one specific circular area, it is important to get the mean of the temperature as recorded by several sensors deployed on the circumference of the circle. This information was used to propose an algorithm that can be used to place sensors in this scenario for effective temperature monitoring of the circles as shown below:

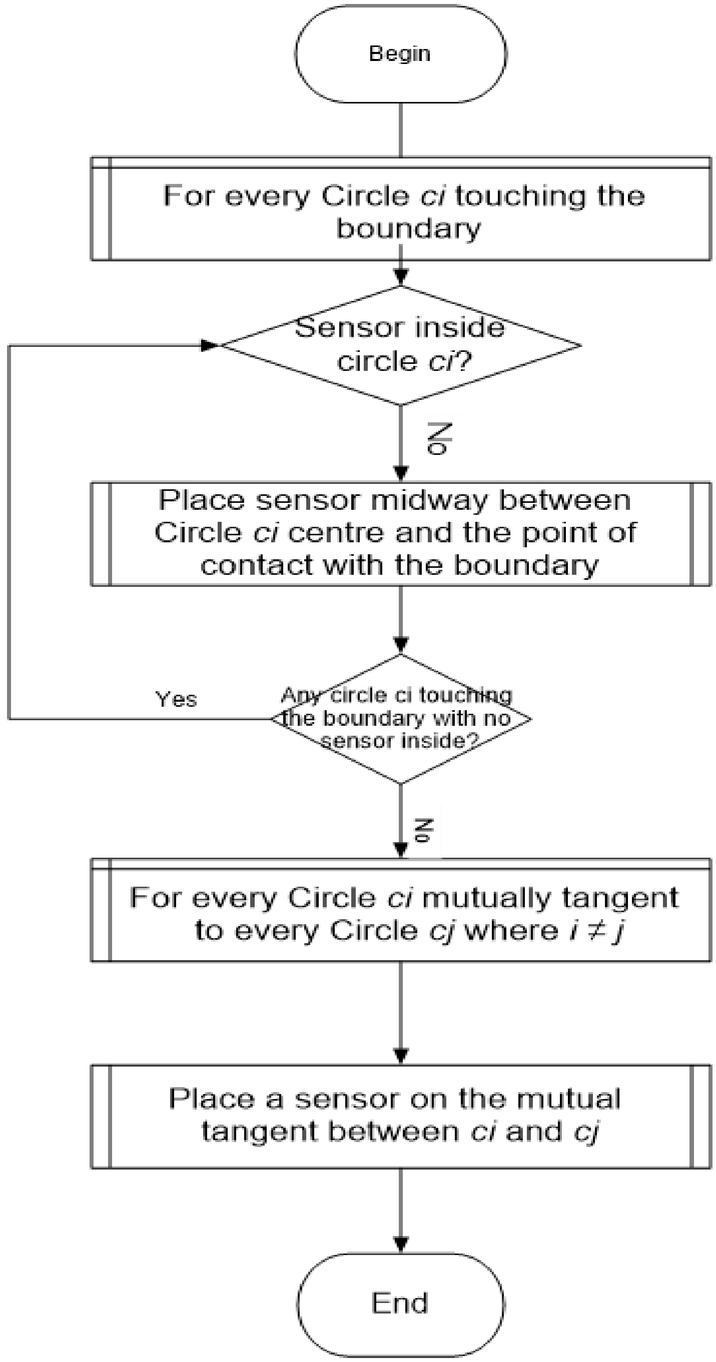


Figure 15: Sensor Placement Algorithm

The resulting sensor setup generated by this algorithm is presented in the Results and Discussion section of this document.

3.3 Actuator Control Algorithm

During the operation of the designed control system, there will arise situations where the temperature is higher or lower than the recommended ranges for the optimal growth of the birds. It is the same case with the relative humidity which will be higher or lower in some moments. It is this problem that called for the design of the environmental control system in the first place. The most common method of operation of the current environmental control systems is to turn heaters on when the temperature is low and to turn cooling equipment on when the temperature is high. With no regard to the temperature, every time the relative humidity is too low, the humidifier is turned on and when the humidity is higher than required, the dehumidifier is employed to operate. Much as this method works, it has to be noted that the amount of heat required to raise the temperature of a room is dependent on several factors, one of them being the amount of moisture in the air. The relative humidity of a room is also a function of the air temperature in the room. With this information, it seems it would make sense to consider the humidity when controlling the temperature and to consider the temperature when controlling the humidity of the room. The hypothesis drawn from this was, if the relationship between temperature and humidity was considered when controlling them, the controlling efficiency would be increased.

To test the hypothesis, an experiment was designed to design an algorithm that considers the relationship between temperature and humidity when controlling them then compares it with the legacy algorithm that controls everything separately. These algorithms were to be compared using simulations and have their results compared for efficiency. Below is how the experiment was done.

3.3.1. Algorithm Design

Environmental States

To design the algorithm, there needed to be a consideration of all the possible states of the environment to determine how the actuators will be controlled in each of those scenarios. The designed algorithm used psychrometric calculations (not to be confused with *psychometric*) to predict the resulting temperature and humidity when an action is taken on the environment since a change in one of the parameters may result in the change of the other. Below are the possible environmental states and an explanation on each.

1. Low Temperature and High Humidity

In this scenario, both heating and dehumidification seem to be required. However, the order in which these two actions are performed can affect the efficiency of the whole process. Raising the air temperature without adding extra water vapour results in lower relative humidity. This is because the air can hold more moisture at higher temperatures. It is possible to predict, through psychrometric calculations, the resulting relative humidity after the temperature is raised to the required level, which will determine whether to humidify, dehumidify or take no action on the humidity. If the resulting humidity is projected to be higher than recommended, it is best to dehumidify before heating since the extra water vapour would require extra heating load.

2. Low Temperature and Low Humidity

In this scenario, it is known that there is no extra water vapour to heat in the air and the relative humidity will drop even further after heating. It is also known that heating dry air requires less heat energy than heating moist air. Harnessing this knowledge, it is best to heat the air to the recommended temperature then humidify it to the recommended relative humidity afterwards.

3. Low Temperature and Recommended Humidity

Using an algorithm that does not consider the interaction between temperature and humidity, it would seem like only heating is required in this scenario. If we look forward, however, increasing the temperature to the recommended level will reduce the relative humidity. The air may then have either recommended or low humidity. Either way, it is still best to heat the air first then raise the humidity if need be.

4. High Temperature and High Humidity

In this scenario, cooling needs to come after dehumidification to avoid extra cooling load from extra moisture. However, it is best to determine how much to drop the humidity by considering the rise in relative humidity that will come after cooling. It is possible to calculate the absolute humidity required in the air to keep the relative humidity in the recommended range after cooling. The relative humidity can then be dropped to the level where the absolute humidity is as would be required after the process. Cooling can follow and that may avoid the cooling of extra water vapour.

5. High Temperature and Low Humidity

Using the algorithm that does not consider the interaction between temperature and humidity, it seems the required actions are Cooling and humidifying. Cooling, however, raises the relative humidity so the humidity after cooling may either be low, high or in the recommended range. It is best to predict what the relative humidity after cooling will be. If the relative humidity will be higher, it needs to be dropped before heating until the available moisture is projected to result in a recommended relative humidity. Otherwise, heating can proceed then humidification can follow later if need be.

6. High Temperature and Recommended Humidity

In this scenario, cooling is required but it will raise the humidity. We can calculate what the resultant relative humidity after cooling will be and if it is to be higher than the recommended value, dehumidification is required before cooling to reduce the cooling load.

7. Recommended Temperature and High Humidity

In this scenario, it is best to dehumidify the air to the recommended range. Dehumidification, however, may release heat, but it is negligible. In the case that the heat released from the dehumidification raises the temperature to above the recommended range, the system would turn itself into either scenario 0 or scenario 0.

8. Recommended Temperature and Low Humidity

In this scenario, it is best to humidify the air to the recommended range. Humidification may cool the air so if the temperature ever drops to levels below the recommended range, the system will turn itself to either scenario 0 or scenario 0.

9. Recommended Temperature and Recommended Humidity

This the desired state of the system and all the actions taken during temperature and humidity control aim at achieving this state. No action, therefore, is required when the system is in this state.

The system being controlled will always be switching from state to state based on the outside conditions. This algorithm considers that dynamic property of the system. Below is the pseudocode of the system representing the above-described scenarios.

Algorithm Pseudocode

Input: current dry bulb temperature, relative humidity, recommended temperature range and recommended relative humidity range.

Output: Action Instruction (Heat, Cool, Humidify or Dehumidify)

If temperature < recommended temperature range **then**

Calculate the resultant relative humidity after raising the temperature

If the resultant humidity after raising temperature > recommended humidity range **then**

Return "Dehumidify"

Else

Return "Heat"

End if

End if

If temperature > recommended temperature range **then**

Calculate the relative humidity after dropping the temperature

If the humidity after dropping temperature > recommended humidity range **then**

Return "Dehumidify"

Else

Return "Cool"

End if

End if

If temperature is in the recommended range **then**

if humidity > recommended humidity range **then**

Return "Dehumidify"

Else If humidity < recommended humidity range **then**

Return “Humidify”

Else

Return “Do Nothing”

End if

End if

Effectively, the algorithm checks the temperature first and if it is not in the recommended range, it projects what the humidity will be when the temperature is controlled. If the humidity is projected to be higher than the required range, dehumidification takes place, otherwise, the temperature can be controlled. If the temperature is in the required range, the humidity is then checked and controlled accordingly.

Humidity and temperature calculation are done using the psychrometric chart. It is available as a library in Python (Used in this study), C, C#, VBA/Excel and R [59].

3.3.2. Algorithm Simulation

Having designed the algorithm and having it developed using python, it had to be compared against the legacy algorithm, also designed in python. The legacy algorithm works by checking the temperature, then sending an instruction to control it accordingly. In the same iteration, it checks the relative humidity and sends an instruction to control it accordingly. This method disregards the effect of controlling the temperature on the humidity and vice versa.

Due to financial constraints and the effect of COVID-19 on the availability of resources, the comparison between the two algorithms was just simulated. The simulation worked by taking temperature and humidity data from a room then running the algorithm on the data. The simulation was receiving the instructions from the algorithm then simulating heating, cooling, humidifying and dehumidifying. In a nutshell, there needed to be data with enough scenarios requiring heating, cooling, humidifying and dehumidifying. The data collected in the previous experiment (Sensor Positioning) was lacking one main characteristic of our needs, which was the availability of enough scenarios where heating or cooling was required. To achieve this property, the data was systematically modified as described below:

Data Preparation

The data from the Sensor Positioning experiment had a small range (23.1℃ to 27.8℃) and this would not provide enough simulation scenarios since the data did not require heating or cooling in most cases. To achieve this, the data had to be modified to increase the spread such that the lower end data had even lower values and the upper-end data had even higher values. In effect, this would create more scenarios with the temperature being out of the required ranges.

To achieve this, every value was updated by adding it with its Z-Score. A Z-Score (Standard Score) of a value is the number of standard deviations by which a value is away from the mean. The Z-Score is positive for values above the mean and negative for the values below the mean. Updating every temperature value with its z-score ensured that the values above the mean get further above the mean and the values below the mean get further below the mean. This effectively increased the spread of the data such that more scenarios requiring heating or cooling were introduced in the data.

The data updating was done as follows:

$$Ti = Ti + (Zi \times k)$$

Where

T_i is the temperature data at position i

$$Zi = (Ti - \mu) / \sigma$$

μ is the mean of the temperature data

σ is the standard deviation of the temperature data

With the modifications done, it was time to run the algorithms through the data using the designed simulation which is explained below:

Simulation

The simulation was designed with some specified considerations. Firstly, it was considered that during heating there would be an equivalent of 1.5KW heating power for every 10 m^2 (150 Watts per m^2). It was also considered that during humidification or dehumidification, there would be an addition or removal of 0.0001 Kg of water per Kg of air respectively. The simulation did per second calculations on the data even though the data was collected per 15 seconds.

The simulation program worked by iterating through the rows of the data and passing the temperature and humidity value from the current row to the algorithm which returned an instruction. The instructions were Heating, Cooling, Humidifying, Dehumidifying or No Action. The returned action was simulated and the resultant heat or moisture was added or subtracted from the data to be used in the next iteration. The next iteration simulated the next seconds which was not always the next row in the data. From the actions instructed by the algorithm, the data was being updated for each second until the iterations hit the next row of data. At that point, the data from that row were compared with the data from the previous row to account for the effects on the data without the actuators. The data was updated accordingly and simulations continued for the next seconds until the upcoming row was reached. Every action instructed by the algorithm was recorded and the amount of time each actuator stayed in active operation was recorded as well for comparison.

Both algorithms were run using the simulation program and their results were recorded. The results of the simulations will be presented in the Results and Discussions section of this document.

3.4 Conclusion

The two main sections discussed in the research methodology set a ground for the design of an intelligent control system for small scale poultry farms. The designed control algorithm introduces the intelligence to project the outcomes of doing certain actions and chooses an action that completes the controlling task efficiently. The sensor positioning algorithm makes the use of resistive heaters (which are cheaper) effective by establishing the right positions to position the heaters and sensors to effectively monitor and heat the environment in the poultry house. The results from these two set the ground for the design of the embedded system to act as the control system sensor node. The sensor node set the ground for the development of the API and data dashboard for the control system and these are discussed in detail in the System Analysis and Design section.

4. SYSTEM ANALYSIS AND DESIGN

4.1. Systems Planning

"If you fail to plan, you are planning to fail" – Benjamin Franklin

Systems planning is the method of analyzing, defining and designing the information architecture of a business [67]. A project is more likely to succeed if the process of systems planning is undertaken carefully such that the feasibility, costs, problems and opportunities are understood. This in turn helps in the drafting of a careful plan and fallback mechanisms in any case of failure of the initial plan. Possible risks and their mitigation strategies are also part of the plan and that reduces the odds of project failure. This part of the document discusses the planning that was conducted during the development of the poultry farm control system.

4.1.1. System Components

A system is a set of related components that produce a certain result [68]. A system may contain different types of components but the main components discussed in this section are Hardware, Software, Data, Processes and Stakeholders (People). The hardware consists of all the physical devices that are used in the system while software consists of all the programs that control the hardware and derives information from data. Data consists of the raw materials that when processed, result in information. Processes refer to the tasks and functions that the system users perform, using the system, to get specific results and/or transform raw data into useful information. Finally, the stakeholders, as stated in this section, refer to all the people that will have direct interaction with the system either by analysing, developing and/or using it to perform tasks. Below is a description of the components that were planned to be used in the system:

Hardware

An embedded system is a computerized system built for a specific purpose [69]. These systems mainly comprise of a microprocessor, memory and peripherals specific for the task at hand. Control systems, on the other hand, use these embedded systems to manage a larger system such that the overall response approximates the commanded behaviour. In this specific control system, the embedded system was used to monitor the environment in a poultry farm then use actuators to control it such that the resulting environmental parameters are optimal for the growth of poultry birds. To achieve such, the following hardware had to be used:

Sensors

To monitor the environment in the poultry farm, embedded systems use these components called sensors. They transform physical phenomena to electrical signals which, when read by the embedded system, can be translated to a value representing the physical phenomena that were read. This system was meant to read temperature and humidity data from the poultry farm before analysing and communicating it.

Actuators

The main purpose of the system was to control the environment in the poultry farm to keep it in the optimal range for the growth of poultry birds. After monitoring the environmental parameters and analysing them, the system needs some sort of mechanism to control the environmental parameter in the case that they are out of the required range. The devices used for this mechanism are called actuators. The actuators required in this system were the ones to increase and decrease air temperature and to humidify and dehumidify the air in the poultry farm.

Microcontroller

A microcontroller is an integrated circuit consisting of a microprocessor, memory and input/output (I/O) peripherals on a single chip. It is used as the brain of most embedded systems. This is the part of the embedded system that processes the sensor data and instructs the actuators to start or stop their operation. Depending on the task at hand, there are various options for microcontrollers with variable processor power, memory and available I/O peripherals. For this system, there was no need for high memory or processor power except for the tasks of formatting sensor data for communication and processing it to find the best way to control the actuators.

Network Modules

One integral part of this system was the communication of the environmental parameters. The system had to send the data it collects from the sensors to a central server where it was to be stored and further analysed. This data is used to detect anomalies in the operation of the embedded system and used to draw insights on the problems and opportunities to tackle in the system operation. To send this data, the embedded system needed to have a networking module that can send the data efficiently and effectively.

Server

A server is a piece of computer hardware or software that provides services for other programs or devices. For this system, the server was required to persist data from the embedded devices and analyse it to draw useful insights. The server was also used to provide a user interface to the embedded system through a web dashboard.

Software

The poultry farm control system consists of both hardware and software to run on it. The software is on several layers of the system. Firstly, the embedded system run software which controls the sensors, actuators and communicates telemetry to a server through the network. In the server, there is software which persists the data in a database, provides a user interface to remotely monitor and control the embedded software and analyses the telemetry to draw insights on the operation of the whole system.

Embedded Software

The poultry farm control system used a low power microcontroller hence the software to run on it had to take into account the hardware constraints the microcontroller introduced. The system has low memory and processing power. Nonetheless, the system had to be hard real-time since the larger system it controls is time-critical. A hard real-time system is the ones that have to always be in sync with the state of the environment they operate in and provide fast response even when imposed with high load. This can be achieved if the load is well managed, the system is programmed with a fast response programming language and the tasks provided to the processor are assigned with priority levels such that the system is able to prioritise the most critical tasks.

Server Software

After the embedded system collects data and controls the environment in the poultry farm, the data is communicated to a server where it is persisted in a database and also analysed for further action. The data is also communicated to the end-users of the system through a dashboard. For these functions to be achieved, there was a need to either build or use off-the-shelf software to run in the server. The software needed were ones to be responsible for communicating with the embedded device and the browsers running the web dashboard, persisting data and data analysis.

Data

Data is an integral part of the poultry farm control system being built in this study. The main process run by the embedded system starts with the collection of temperature and humidity data from the environment then processing the data to decide on the action to be taken. Additionally, any action taken by the actuators needs to be recorded together with the electrical power that these devices use during their operation. This data is what determines the overall performance of the whole system. It is the same data that is communicated to a server and sent to the end-users of the system for remote monitoring of the operation of the poultry farm control system.

Processes

These are the tasks and functions that the system carries out either automatically or as a response to a user's command. The fundamental processes of the poultry farm control system are sensor data collection, calculating the control path of actuators, communicating sensor data with the server and displaying the data to the end-user on a graphical user interface. These processes have many other subprocesses which when linked together provides the full functionality of the system.

People

It was important to understand the stakeholders of the system before diving into its development. The system is mainly targeted to the farmers who run poultry farms and need to use this system to automate the farm processes. These farmers are not expected to have full trust of the automatic operation of the full system hence they are supposed to be provided with a way to monitor the operations of the system remotely. That is why the data collected from the farms are sent to a server and presented to the farmers in a human-readable and user-friendly manner. However, for the system to operate and accommodate as many poultry farms as possible, there is need for another user of the system to be responsible for managing the data of the farmers by registering, editing and/or removing farmers from the system. Another important category of stakeholders of the system is the developers and maintainers of the system. It is expected that the system will get feedback from the users and will need to be improved where problems or opportunities arise. These are the people who will be responsible for that function. For the development of this system, it is integral that all these stakeholder categories are considered.

4.1.2. System Feasibility

A feasibility study is the assessment of the practicality of a proposed project or system. It uses four main yardsticks to measure a proposal: Operational Feasibility, Economic Feasibility, Technical Feasibility and Schedule Feasibility. The questions that these main yardsticks answers are as follows:

i. Is the proposal desirable in an operational sense? Is it a practical approach that will solve a problem or take advantage of an opportunity to achieve business goals?
ii. Is the proposal technically feasible? Are the necessary resources and people available for the project?
iii. Is the project economically desirable? What are the projected savings and costs? Are there any intangible factors involved? Is it a sound investment?
iv. Can the projects be accomplished in an acceptable timeframe?

Once these tests pass, it is safe to conclude that the project is feasible.

Operational Feasibility

The poultry farm control system is intended for use by small-scale poultry farmers who may not have the technical know-how to operate most computer systems. It was shown that farmers' literacy level is a key driver in food production and food security [70]. This system is designed to overcome the need for farmers' education for the effective production of poultry products. It automates most of the farm activities and operates with precision. This increases the number of people with the ability to use this system effectively and increases the likelihood that farmers will adopt it. The system also solves the problem of having to use a farmers' senses, which are not accurate, to measure the environmental parameters in a poultry farm and control them manually. The farmer cannot be on the farm at every point in timer and neither can they be able to accurately monitor and control the environment on the farm. The system tackles both these problems by continuously monitoring the environment through day and night and accurately monitoring and controlling the environment when the need arises.

The projected effects of the system to the farmers will be the need for electricity to run the actuators responsible for controlling the environment. This, however, will be in replacement of the already existing methods of controlling the environment that the farmers use. The advantage is that this system will work more efficiently by operating the actuators precisely and only when required. In addition to that, the system will calculate a more efficient way of operating the actuators to reduce the amount of electrical energy consumed.

Technical Feasibility

The poultry farm control system is designed to have an embedded system in a poultry farm and an offsite server for further data processing, storage and user access. During this study, there was a need to develop a prototype as a proof of concept. For this to happen, several things had to be available. These were the necessary hardware, software and personnel to develop the prototype. The project was conducted on a very stringent budget and therefore it was designed to use easily accessible resources to show the concept.

The embedded system required a microcontroller, sensors, actuators, a communication module and an access point. In the case that any of the sensors and/or actuators are not available, a fallback mechanism was planned to simulate the parameters from the sensors and the effect of the actuators respectively. The required sensors were temperature sensor, humidity sensor, and light sensor. The required actuators were, humidifier, dehumidifier, heater, cooler and light bulbs. On top of the sensors and actuators, the system required a communication module with its respective access point to provide internet access. These were easy to find, especially using Wi-Fi since it was readily available.

The system also required a server to persist data and do further processing of the data. The server also needed to have an internet connection to be able to receive data from the embedded system in addition to receiving data, the server was supposed to serve a web dashboard for users to access the poultry farms' data remotely. Due to the availability of resources, the system was designed to run the server in the cloud since it does not require any hardware investments, it is available 99.9% of the time and it is reliable.

Finally, the system had the required expertise to be developed. The authors of this thesis had the technical know-how and required knowledge sources to take the project to completion. After planning all these out, it was enough to conclude that the system was technically feasible.

Economic Feasibility

For the poultry farm control system to be developed and used, it had to make economic sense. The poultry farm control system was developed to control the environment in a poultry farm for the optimal growth of poultry birds. It is discussed in the literature review of this document that the effect of the environmental parameters on the birds can be harmful to the whole poultry farming enterprise such that the profitability of the business is diminished.

The main issue to be discussed in this section is the projected benefits of the system and whether they outweigh the costs. For such comparisons to be made, there was a need to come up with a budget for the whole poultry farm control system project.

Below is the initial project budget as proposed:

Table 2: Initial Project Budget

Resource/Activity Name	Cost/Use	Quantity	Total Cost
NVIDIA Jetson	$300.00	1	$300.00
Water Heating Element	$18.00	4	$72.00
DHT22	$5.00	10	$50.00
Heating Element	$25.00	2	$50.00
Electric Fan	$7.00	4	$28.00
Humidifier	$35.00	2	$70.00
Servo Motor	$19.00	4	$76.00
Water Level Sensor	$8.00	4	$32.00
DS18B20	$5.00	10	$50.00
MQ135	$5.00	4	$20.00
SEN-10245 Weight Sensor	$22.00	4	$88.00
Waterflow Sensor	$18.00	4	$72.00
Solenoid Valve	$25.00	4	$100.00
Pin Extender (MCP23017)	$5.00	2	$10.00
Arduplane Airspeed Sensor	$5.00	2	$10.00
ADC (MCP3008)	$5.00	4	$20.00
Data Collection	$30.00	2	$60.00
Circuit Board Printing	$15.00	1	$15.00
3D Printing	$20.00	1	$20.00
Field Test	$50.00	1	$50.00
Shipping Costs	$150.00	1	$150.00
Contingency	$150.00	1	$150.00
		Total	$1,493.00

Due to financial problems and unavailability of most of the resources during the COVID-19 period, some of the above-stated resources were removed from the plan and changes were made to the overall project plan so that their functionalities would be simulated.

Table 3: Final Budget

Resource/Activity Name	Cost/Use	Quantity	Total Cost
NodeMCU	$15.00	1	$15.00
Bread Board	$5.00	4	$5.00
DHT22	$40	4	$40
Electric Fan	$7.00	4	$28.00
Wi-Fi Access Point	$20.00	1	$20.00
		Total	$108.00

The table above shows the final project budget after most of the proposed resources were removed. The system to be built in this system is projected to remove the need for a farmer's intervention in controlling the environment in the poultry farm which reduces labour wages. In the long run, this amount is likely projected to outweigh the upfront cost of the system. If the system delivers what it promises, it is also likely that it will reduce the operation costs of controlling the environment hence reducing costs while maximizing production and hence increasing the profit of the poultry business. With all that is light, it was safe to conclude that the proposed poultry farm control system is economically feasible.

Schedule Feasibility

A project may be operationally, technically and economically feasible but if it cannot be delivered in the required timeframe then it is not feasible. It was important to establish a timetable for the project to be able to assess if it could be delivered before the set deadline and to act as a checklist to track the project's progress. Below is the high-level timetable of the project as a Gantt Chart:

Table 4: Project Timetable

Work Package Name/ Definition/Status	**MONTHS**					
	March	**April**	**May**	**June**	**July**	**August**
Preparing Research Design						
Prototype Design						
User Interface						
Data Collection						
Data Analysis						
Compile First Paper						
Develop Embedded System						
Integrate Embedded System and Graphic User Interface						
Compile Second Paper						
Compiling Up the Thesis						

During the planning phase, it was not projected that something as severe as the COVID-19 lockdown could affect a very big part of the schedule. The project was given a go-ahead after concluding that the schedule of the project was feasible. Later when the schedule was significantly affected, two months were added to the project timeframe such that the project remained feasible as far as the schedule was concerned.

The project plan was documented in Microsoft project and the task breakdown was represented in a Gantt Chart for tracking. Reports were being generated at the end of the week to provide an idea on whether to work at the same pace, slow down or work for some extra hours.

4.2. Systems Design

This section discusses the design of the poultry farm control system. The design was done in phases and iteratively. The very first stage of the design process was the user interface design followed by the data design and finalized with the system architecture design.

4.2.1. User Interface Design

The user interface is a very important component of any computer system. It describes how a user interacts with the system and consists of all the hardware, software, screens, menus, functions, output and all the features that affect the two-way communication between a user and the computer system [68]. The user interface is what makes a computer system easy to learn and use. The poultry farm control system considered two types of user interfaces depending on the functions. These were the hardware user interface on the embedded system and the web data dashboard's user interface for monitoring the control system's performance and the administrating the system.

Web Dashboard UI Design

The web dashboard is an important part of the system since it enables the users of the system to monitor the system's performance. It shows all the parameters that are collected from the control system and sent to the web for storage and further analytics. A good design of the web dashboard UI shows the required data to be collected and how the data should be stored. The dashboard shows temperature, humidity and energy consumption data to the users in real-time and shows the actuator operation and energy consumption after further analysis. Below are some of the screen grabs from the web dashboard:

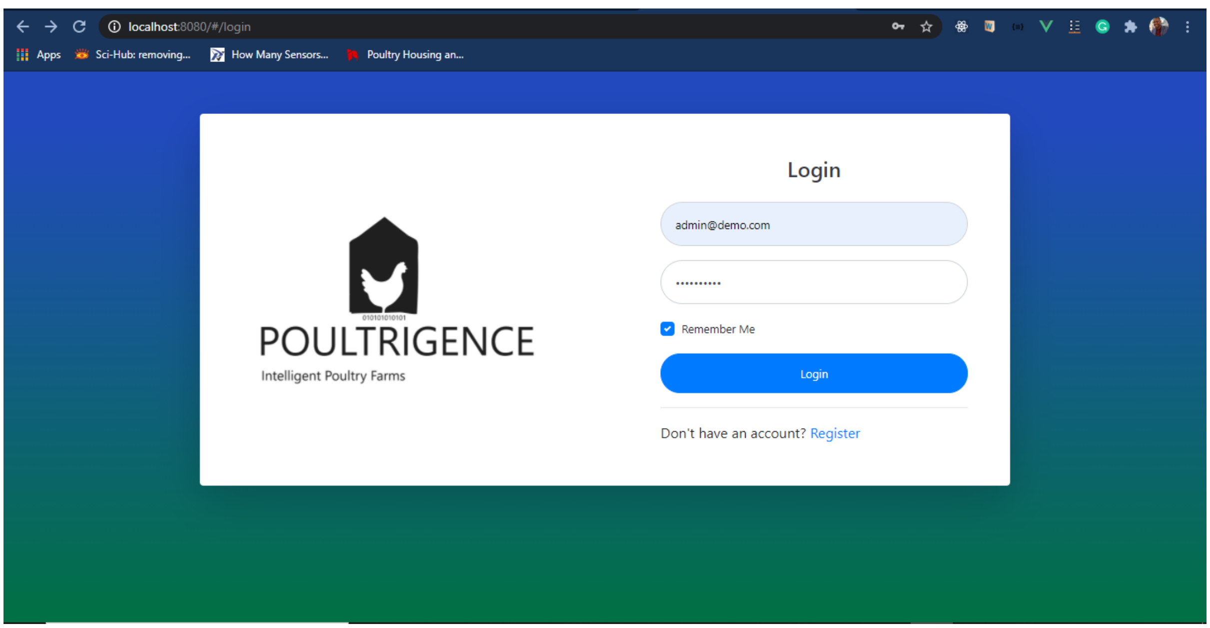


Figure 16: Login Screen

The picture above is the login screen of the poultry farm control system. It requires the email and the respective password as they were set during the user account initiation. It also includes a 'remember me' checkbox to ease authentication the next time the user tries to log into the system. This part is available to smoothen the user experience on the system and not make them bored with having to provide login details every time they have to use the system. It also provides a link to the registration page in case a non-registered user tries to visit the web dashboard.

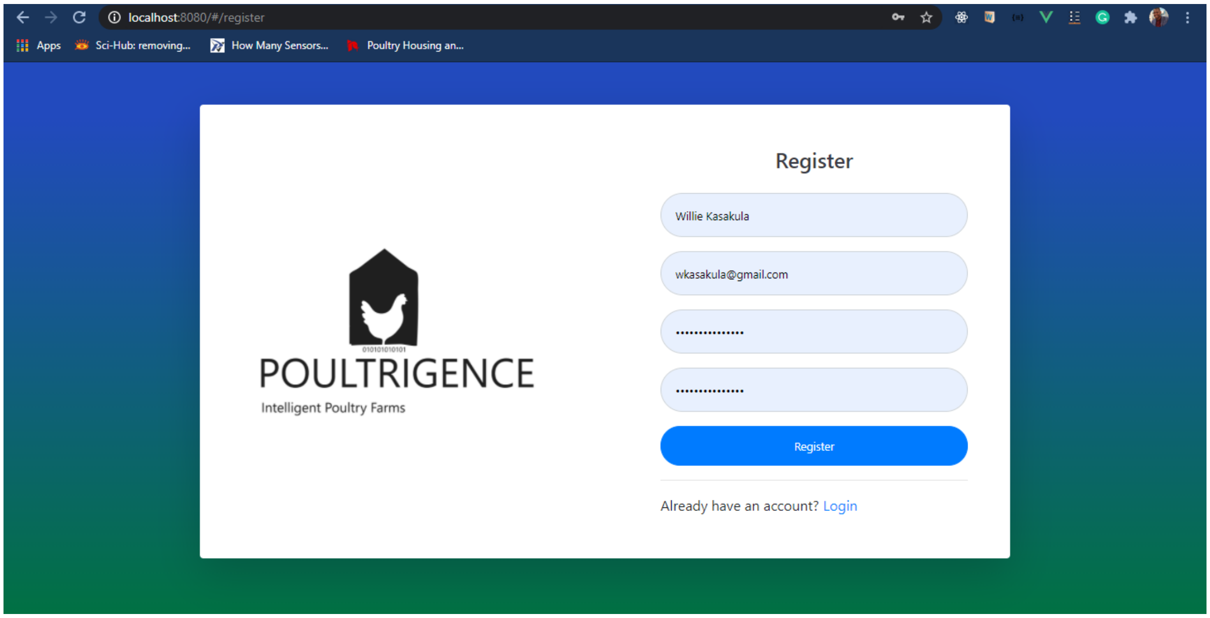


Figure 17: Registration Screen

The screen above shows the registration page that any user of the system can use to create an account. The user needs to provide a name, email address, and a password twice to minimize any chances of mistyping it. The page also provides a link to the login screen in case a user with an account already visits the registration page. The registration page is designed in such a way that after the register button is clicked, the details should be submitted to the system and an account with the details should be created then have the user logged in automatically.

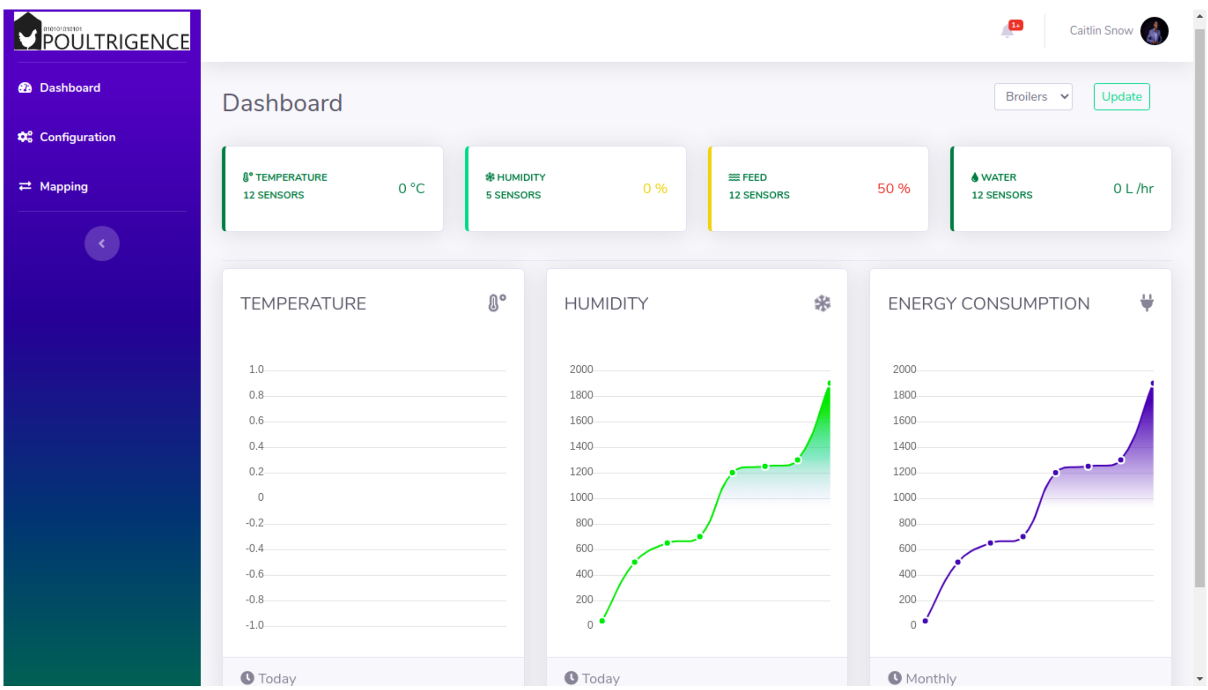


Figure 18: Home Screen (Dashboard)

The homepage is the main component of the dashboard. It shows summaries of all the parameters that are being collected from the poultry farm control system. It shows the number of active sensors for each parameter and the average of the parameters. The dashboard also plots line charts of the said parameters in real-time. This provides a mechanism where a user can know the parameters in their poultry farm without having to physically be on the farm. It also provides transparency in the way the poultry farm is operating. The dashboard also shows the energy consumption of the system derived from the voltage and current of all the sensors and actuators in operation at any point in time. Additionally, the dashboard provides a dropdown list of the different breeds of poultry birds as provided by the system. Once a breed is chosen, the system is supposed to respond by controlling itself to match the thresholds predefined by the selected breed. This change is also supposed to be reflected in the line charts present on the dashboard. On the left side is a navigation panel which provides easy links to other parts of the web system. The top bar provides access to notifications and user profile management in addition to an option to log out of the system.

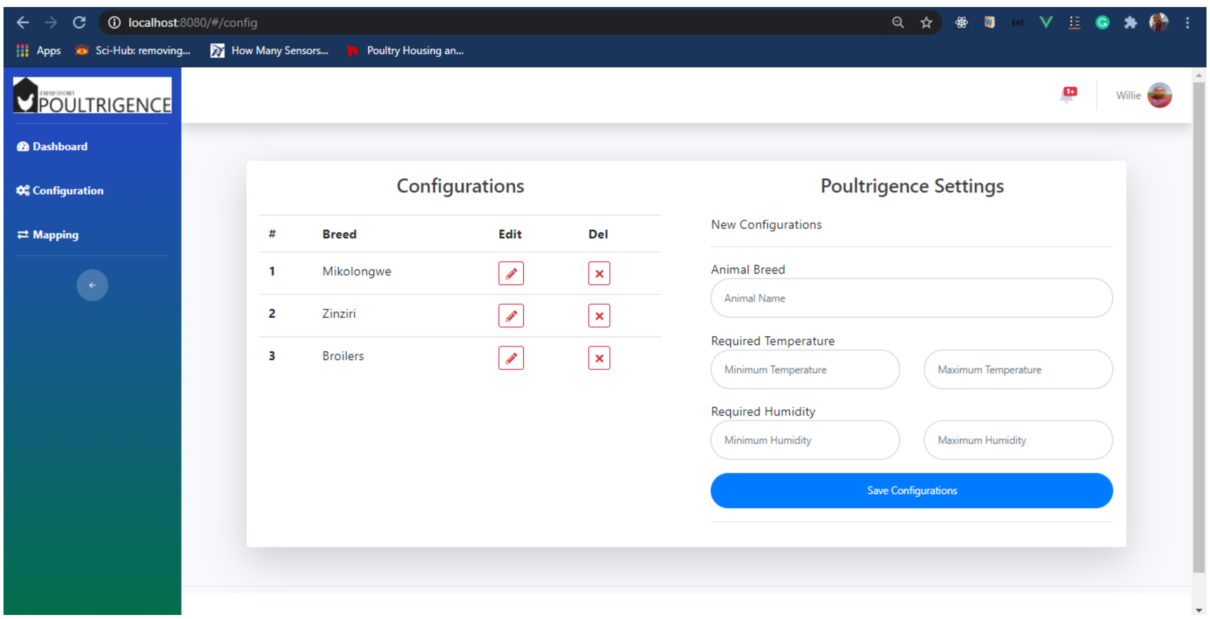


Figure 19: Animal Breed Configurations

The screen above shows the user interface for the configuration of poultry breeds in the poultry farm control system. It collects the breed name, minimum and maximum temperature, as well as the minimum and maximum humidity. The main purpose behind this is, the poultry farm control system is designed to work in any farm with any poultry breed so one the threshold parameters for each breed are configured in the web system, the embedded system in the poultry farm will work accordingly. On top of the preset breed configurations, any user can add extra configurations of their own which will only be available to them. These configurations can be found on a dropdown list on the main dashboard and a user can select which breed they want their poultry farm to be set to at any point in time and it will update in real-time.

On the left is a list of all the available breed configurations in the system and their respective edit and delete button. When the edit button is clicked, the configurations of the selected breed are populated in the respective fields for a user to know the current values then update accordingly as they wish.

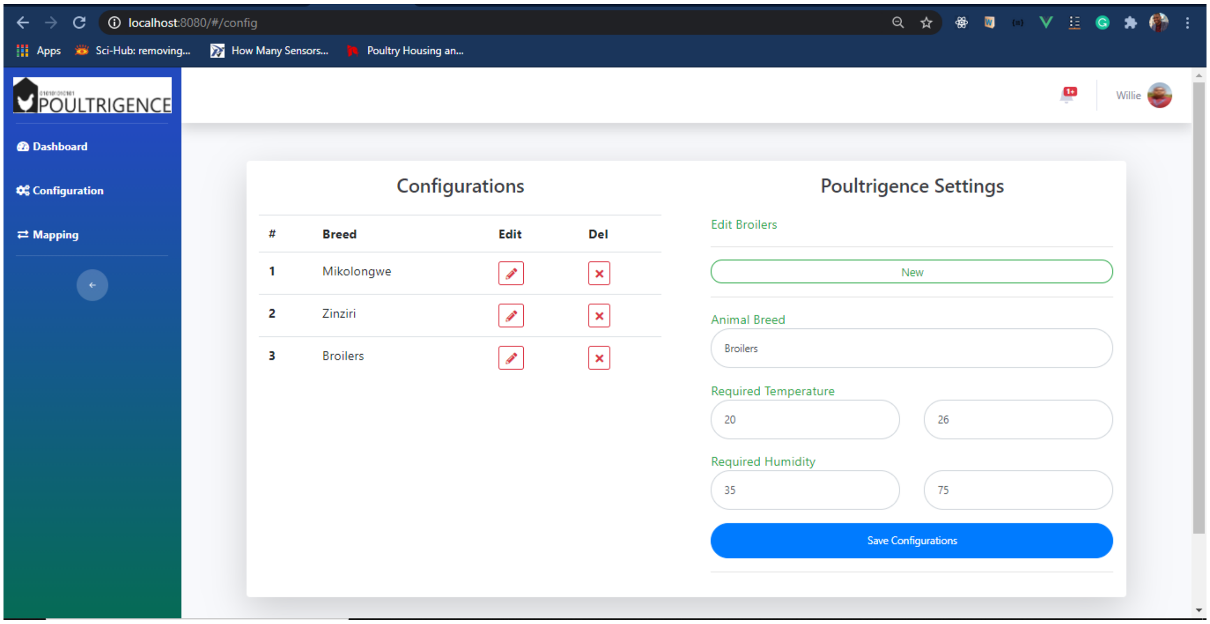


Figure 20: Editing Configurations

Once the user saves the configurations, they are instantly updated in the embedded system in the poultry farm for the parameters in the poultry farm to be controlled as per the new configurations.

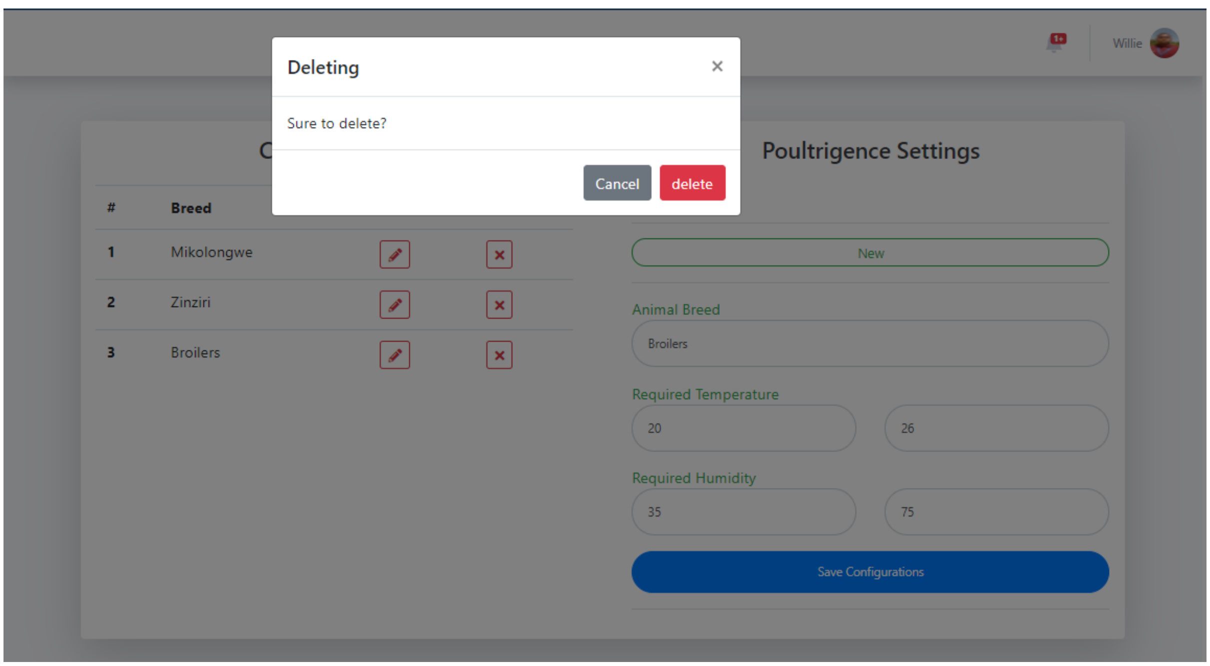


Figure 21: Deleting Configurations

When a configuration is being deleted, the system shows a confirmation box with an intuitive delete and cancel button. This is to avoid deleting configurations mistakenly and making sure that a user only deletes a configuration when they are sure.

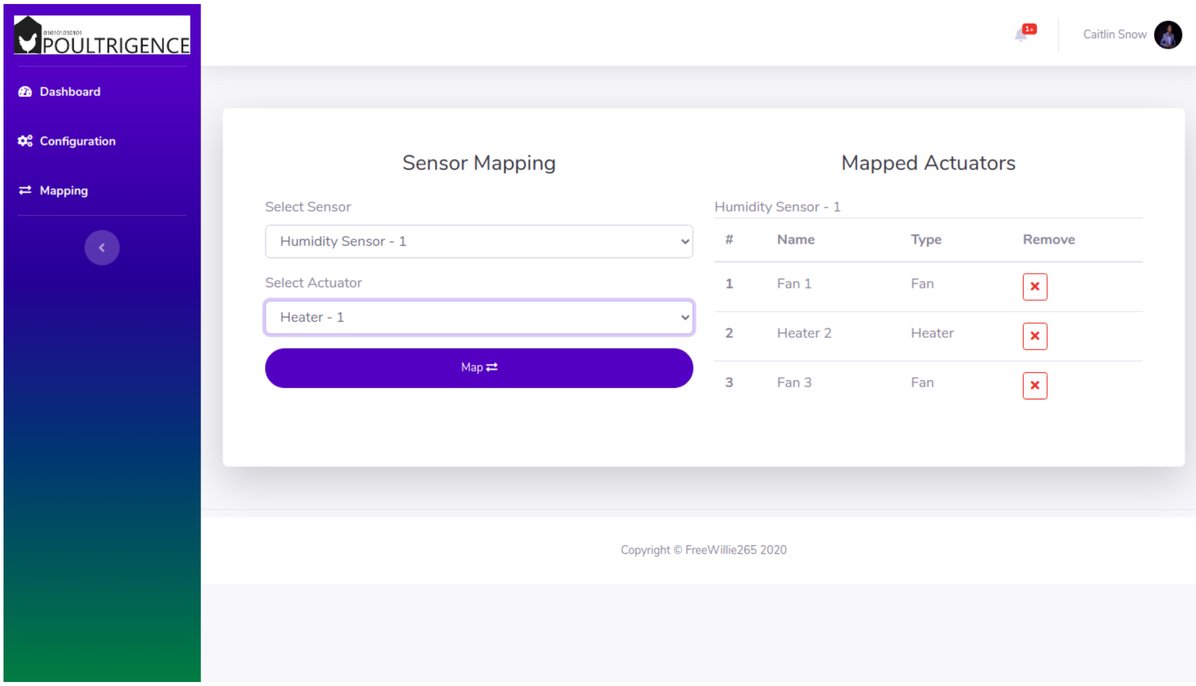


Figure 22: Sensor - Actuator Mapping

The poultry farm control system is designed to have several sensors for each parameter and it is the same case with the actuators for the same parameters. The user interface provides a way to map actuators to respond to some particular sensors. This can be done by selecting a sensor from the dropdown list of available sensors and selecting the respective actuator from the list of available actuators. Once the 'map' button is clicked, the actuator will respond to the parameters from the mapped sensors. On the right of these fields is the list of all the sensor – actuator relationship and a respective button on each relationship to remove it.

Embedded System UI Design

The usability of an embedded system is dependent on how well the user interface of the system is designed. An embedded system's user interface design is done at both the software and hardware layer. The poultry farm control system has most of its functions automated such that the user can only do a limited number of tasks by interacting with the embedded system directly. The user interface was designed using three main principles which were: Visibility, feedback and information availability.

Visibility

Since most of the functions of the poultry farm control system will be automated, there was a need to provide an easy to use way to override the automatic nature of the system. This can be useful when the system is misbehaving due to sensor errors or any other factors. The system is designed with a three-state button on each of the ports where the actuators are connected. The three states are automatic, on and off. The 'on' and 'off' states are used to manually control the respective actuator while the 'automatic' state uses the results of the embedded system's computation to automatically switch the actuators on and off. These buttons are located right next to the port of each actuator to make it intuitive for a user when changing the states of each actuator.

Feedback

As the three-state buttons are being pressed and the states are changing, an LCD at the front of the system shows text confirming the successful change of states and it shows which state the system it has changed to. Next to each button is an LED which shows flashes every time the respective button is pressed to show to the user that the system understood that a button was pressed. The colour of the LED also changes and it shows blue to indicate that the system is in the ‘automatic’ state, green to show that it is in the ‘on’ state and it is off the show that the actuator is in the ‘off’ state. These feedback mechanisms are there to smoothen the user experience with the system rather than keeping the user guessing on what state the system is with no direct way to confirm.

Information Availability

The final principle in the user interface design for the embedded system was the information availability. To make it very intuitive to the user on what is happening, the system had to provide enough information to the user of the system on what is happening. The system was designed with an LCD and LEDs to present the user with information about the state of the system.

Firstly, the system has a power LED to show that the system is powered on. This LED normally shows a green light. In any case that the system is not in the desired state but the system is automatically trying to control it, this LED shows a blue light which has the purpose of notifying the user that the system is not in the desired state but it is in the process of controlling it. This may be a bad value of temperature or humidity in the poultry farm which the system tries to control using the actuators. If the system does not have the actuators to control the parameter or it has spent an unusually long time trying to control the system but it is not changing, the LED shows a red light to notify the user that there is need for external intervention to correct the problem.

To discuss more on the LEDs, the ports when the actuators are connected have an LED each to show the status of the respective actuator. The LED shows blue to indicate that the actuator is in the ‘automatic’ state, ‘green’ to show that it is in the ‘on’ state and it is off when the actuator is in the ‘off’ state. This provides the user with information on which of the actuators is operating and in what state. Additionally, the LED shows red when the system detects a problem or an unusual operation of the respective actuator.

Finally, the LCD at the front of the system is there to provide information to the user. The LCD shows an animation of texts one after another showing the current average of each parameter the system is monitoring, the number of sensors connected for each parameter, the actuators connected and the electrical power being used by the system at each point in time.

The LCD also shows information about the network status of the system. On this, the LCD shows the strength of the signal the same way it is shown on a cellphone. Additionally, it shows if there is a connection to the internet. This is important to let the user know if data is being synchronized with the web system. If the data is not uploaded to the cloud for a while, the system also shows a warning sign to make the information even more visible to the user.

4.2.2. Data Design

The poultry farm control system relies on data to make decisions on how to control the environment. It is the same data that is used to assess the performance of the system which is vital in determining whether maintenance is due or not. For this assessment to happen and for the users of the system to be able to monitor the poultry farm remotely, the data has to be stored in a way that it is easily accessible and makes sense. It is with that reason that the section of data design is written and was done during this study. To begin with, the entities in this system had to be identified for their relationships to be mapped and create a stable design of the data. Below are the entities that were identified.

1. Devices
   Devices, as described in this system, are the sensors and actuators used in this system. The embedded system offers the flexibility to connect multiple sensors and actuators of both the same and different types. The system offers the flexibility of having to configure which actuator responds to the data from which sensor. For this to happen, the data of the sensors and actuators connected to the controller have to be recorded to easily identify them. Other details about these devices enable the system to estimate the electrical power consumed by the actuators and use the results to come up with a sequence of operating the actuators to reduce energy consumption.

2. Animal Types
   Poultry breeds have different requirement for the environment they do best in. The poultry farm control system needs to keep the data about the recommended environmental parameters for each of the breeds that it is to work for. It is designed to be general and work for any breed of poultry as long as the recommended parameters are entered and selected in the system. For this to happen, the data about the breeds have to be stored and, in this system, the data is stored as an entity called animal types.

3. Farms
   The poultry farm control system is designed to be a network of many poultry farms with every farm sending its data to a central location where it is persisted. Users can then remotely monitor the performance of their poultry farms by whether they have one or multiple farms by logging into the system and selecting the specific farm they want to monitor. For this to happen, the system has to keep data of the farms registered to the system and the respective users with the permission to monitor them.

4. Users
   The poultry farm control system is designed with users in mind but these users have to be authenticated and given permissions to only view the part of the system they are supposed to. The users have to be registered to the system and have their poultry farms registered as well or be given administrator privileges to administrate the system. For this to happen, the system has to keep data about the users hence making it one of the most important entities in the system.

5. Temperature
   The temperature sensors in the poultry farm continuously collect temperature data in the poultry farm and for this data to read by the users remotely, the data has to be stored in a server. This data is also kept for later analysis and performance assessment. The temperature entity in the poultry farm control system is designed for those temperature records.

6. Gas

Just as with temperature, the poultry farm control system also keeps records of the amounts of different gases which affect the well-being of the birds on the farm. The entity is designed to keep this data.

7. Humidity
   The humidity entity is there to keep records on the amount of water vapour in the air. This data was proposed to be store as the relative humidity which is represented as the percentage of water in the air as compared to the maximum amount of water vapour that the air can keep at the respective temperature.

Below is the Entity Relationship Model of the system

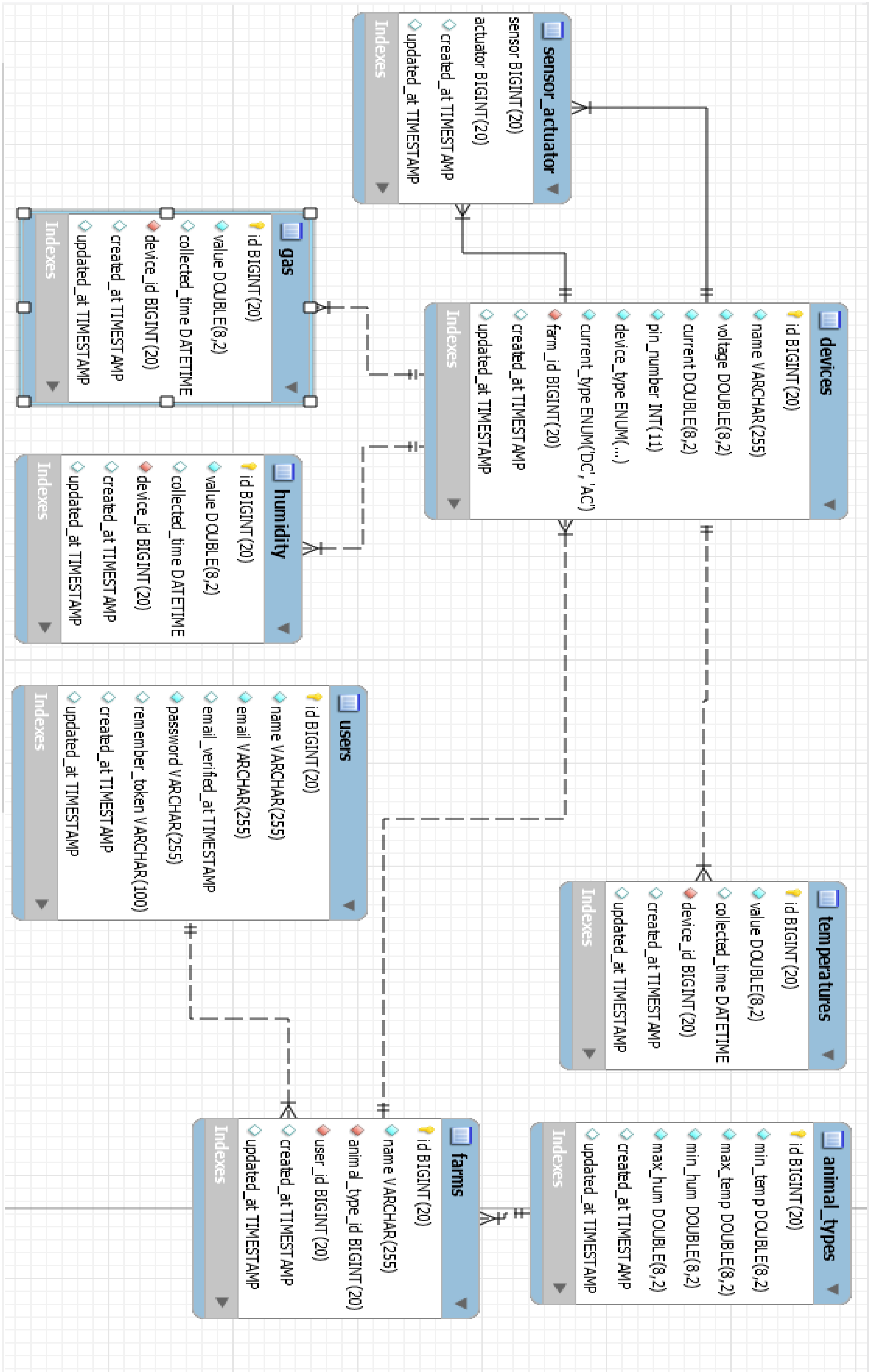


Figure 23: Entity Relationship Diagram

4.2.3. Architecture Design

Having discussed the hardware, software, data, processes and the stakeholders of the system, it led into the design of the web dashboard user interface design, the embedded system user interface design and the data design. All these parts of the system have to be pieced and work together to deliver the requirements of the system. This led to the design of the architecture of the system which would later be implemented into the full poultry farm control system. Below is the architecture of the system.

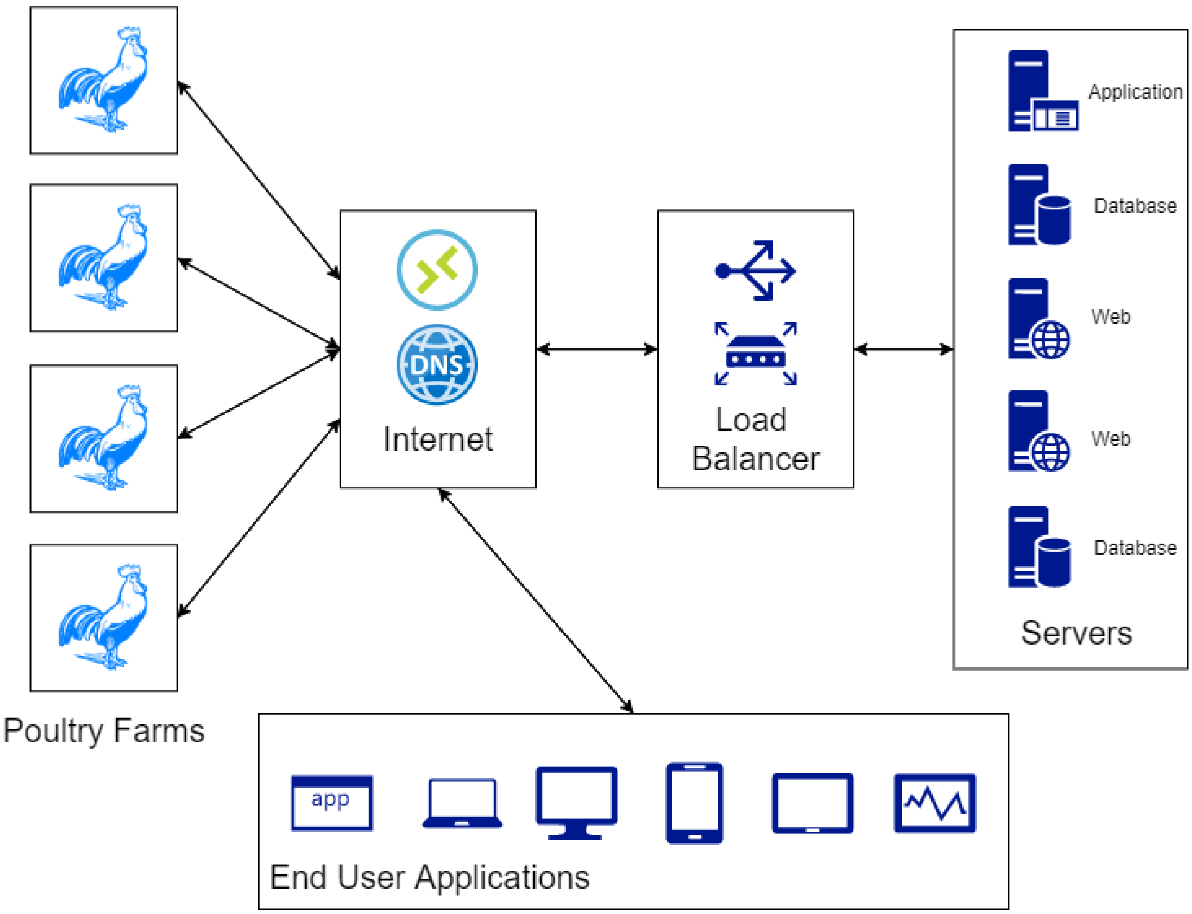


Figure 24: Poultry Farm Control System Architecture

The system is designed to have multiple poultry farms, each with an embedded system to monitor and control the environment in the poultry farm. These poultry farms are connected to the cloud via the internet. The data collected from the poultry farms is directed to a load balancer which directs the traffic to the servers of the system. There are several servers of each type where some are application servers, database servers and web servers. End-users' application for various device types are connected to the internet and have their requests directed to the right servers by the same load balancer to monitor, control and administrate the system.

This architecture decouples the different aspects of the system like end-user applications, website, database and other application to make it easy to maintain the system. This also reduces the load handled by a single device during the whole sequence of processes that occur from when the data is collected from a poultry farm to when it is transformed to information made available to an end-user. This eventually reduces the system's proneness to errors and simplifies the process of debugging problems and adding features to the system whenever an opportunity arises.

Another important aspect of the poultry farm system is the architecture of the embedded system to be installed on the poultry farm to monitor and control the environment on the farm. Below is the architecture of the embedded system of the controller.

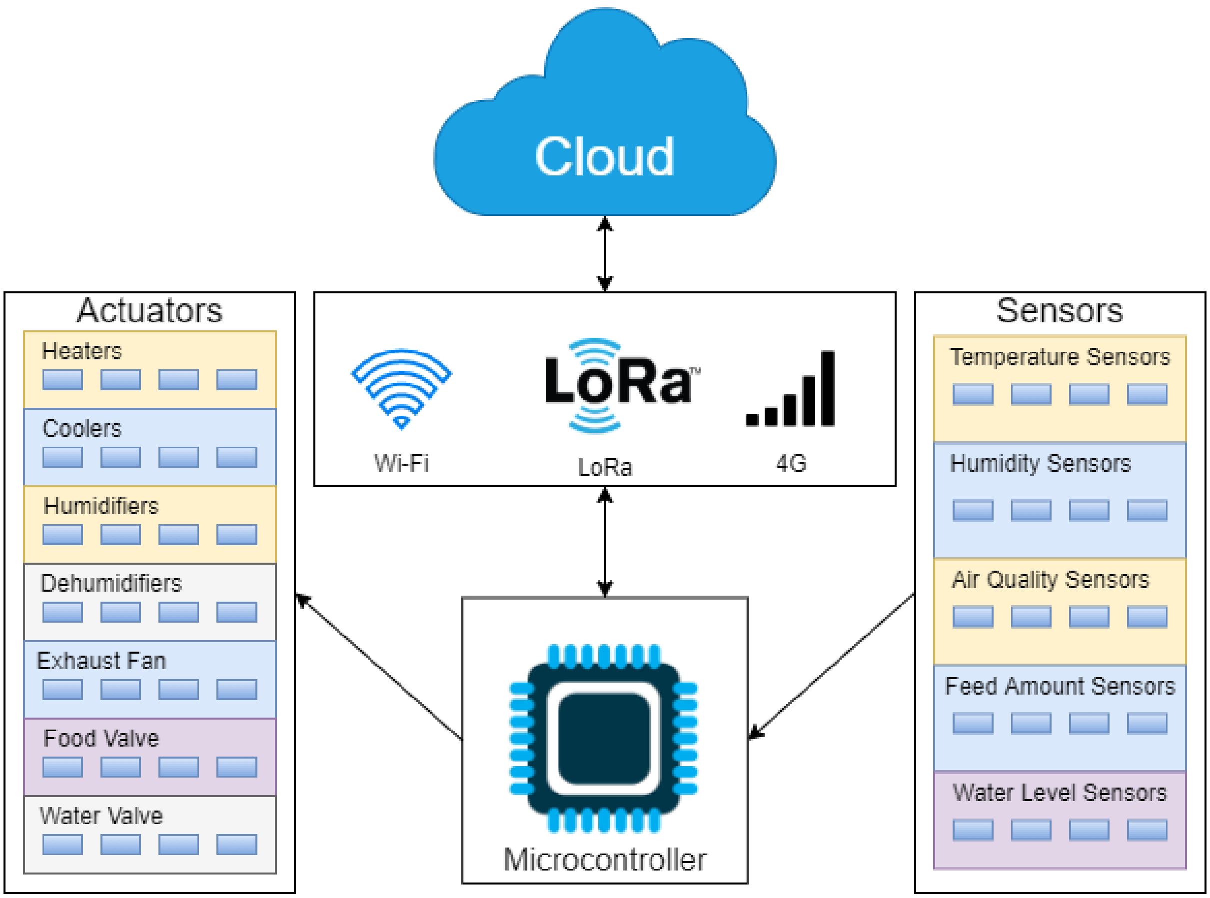


Figure 25: Poultry Farm Controller Architecture

The controller is designed to have connections to several sensors and actuators of various types. In Figure 25, there are four ports for each of the sensors and actuators for the designed system but there can be more ports in the implementation. The microcontroller is the brain of the controller and every component of the controller connects to it. The microcontroller reads data from every sensor connected to the ports available to each section. Since each of the sections is labelled with the type of sensor to be connected to its ports, the microcontroller is programmed to recognize the type of sensor on each port by the number of the port it is connected to.

When the microcontroller analyses the data from the sensors, it determines whether the data is in the required range or not, at which point it may employ the respective actuators to control the parameter in question. The actuators, depending on the port they are connected to, can be dynamically mapped to a specific sensor, or a set of sensors by the end-user. The actuator will then be turned on every time any of the sensors provide a reading which is determined to be out of the recommended range.

The data collected from the sensors and the information about the time when the actuators turn on and off is supposed to be sent to the cloud for storage, analysis and remote monitoring. This data can be sent to the cloud through the internet. For this to happen, there had to be communication modules connected to the microcontroller. The poultry farm controller is designed with Wi-Fi, LoRaWAN and 4G. In normal operation, the system will use LoRaWAN and if it does not work, it will fall back to Wi-Fi and when that fails the system will turn to use 4G. The system will also offer the user with the flexibility of changing the priorities of the communication protocol at which case the system will follow the newly defined sequence when trying to establish a connection with the cloud.

4.3. Systems Implementation

Having reached the point where the systems planning and designs were done and the full specifications and designs of the system were available, implementing the system was next. In the systems implementation stage of the project involved applications development, hardware development, testing and documentation. In this section of the document, the technical details of the implementation of the system are discussed with emphasis on the protocols used and how they are connected to achieve the overall goal of the system.

4.3.1. Application Development

The term 'Application Development' in this document refers to the development of end-user applications which are used to monitor the poultry farms and configure various aspects of the operation of the embedded system. In the scope of this project, a web dashboard was developed as the end-user application. It was developed to have an API to accept data from the embedded system then save it in the database while sending the same data to the frontend application for end-user monitoring. The API also has the responsibility of sending commands to the embedded system for configuration of which actuator responds to which sensor, the thresholds of the sensed parameters and some other system updates.

Database

Having designed the data requirements of the system, implementing it was next. During this stage, the considerations to be made were the type of database to use based on how related the data is, how frequently the database is going to be accessed and how big the database is projected to be. The database system's concurrency and extendibility are considered when choosing the database type based on the aforementioned factors.

There were several options for the database management systems (DBMS) to use for this task. There were options to use a Hierarchical Database, Network Database, Relational Database or Object-Oriented Database. The go-to option was the Relational Database due to its simplicity of representing relationships of data. The designed data architecture for the system represented many relationships among data but it was not so complex as to need the use of a Network DBMS. There were also many options for the specific Relational DBMS to use. The most apparent options were the Oracle Database, MySQL, MariaDB and PostgreSQL. MariaDB was chosen over its alternatives due to its wide use, simplicity and open-source nature. It uses most of the MySQL functionality under the hood since it was forked from MySQL but it does not contain any proprietary code and it is entirely open-source, unlike its predecessor.

Application Programming Interface

The API was used to interface the software intermediaries of the system. The database, parts of the web dashboard and some of the components of the embedded system were interfaced together using a REST API. This API was built in Laravel, a PHP framework built by Taylor Otwell [71]. Firstly, the API exposes endpoints for all of the models that the database has. It provides Create, Read, Update and Delete (CRUD) functionalities for all of the models.

The API also includes functions for the administrator to manage users and other aspects of the system. The endpoints for these functions are restricted by permissions which are validated depending on the authenticated user sending the requests. The API uses two communication protocols:

1. HTTP
   HTTP is a request-response protocol where a connection is established when a client sends a request to the server and the server sends a response back to the client. The connection is killed soon after the response is received. This protocol is very useful for functions that are once-off and do not require a consistent connection. It was used for administrative functions like registering a user, authenticating a user, changing the user's details, configuring the poultry farm's thresholds and getting historical data from the database.

2. MQTT
   MQTT is a publish-subscribe model with a central broker. It provides independence from client to client such that failure on one client does not affect the other clients. It keeps a consistent connection such that data is sent from publishers to subscribers without the headers found in every HTTP request and this reduces overhead. MQTT is great for a client that repeatedly communicate while trying to achieve real-time communication.
   The API subscribed to topics from the broker where it was receiving data published by the embedded systems to achieve real-time communication. The API was also publishing updates to the broker to be received by the embedded systems and others to be received by the embedded systems.

Frontend Application

The frontend applications are a set of software that work together and interface with the users to provide them with methods of interacting with the whole system. The end-user application for this system was built as a web application. It is a single page application built with Vue (JavaScript Framework). The application provides different user interfaces for administrators and farm owners.

It provides interfaces for registering users, authenticating users, configuring thresholds for different poultry breeds and querying historical data. The application interfaces with the API on the other hand using the HTTP protocol to Create, Read, Update and Delete (CRUD) models from the backend system. It provides forms to enter data and lists to view the same data as shown in the *Web Dashboard UI Design*.

For real-time monitoring of the poultry farms, the application connects to the embedded system through an MQTT broker. Both the API and the frontend application are just clients to the broker such that the real-time data is delivered to the end-user regardless of whether the data has reached the API or not.

The frontend application is built as a single page application to make it very responsive and separate from the API. It shows a better performance than multiple page applications which have to load every page from the server every time a user tries a new function.

Embedded Hardware

The embedded system was connected with different sensors and actuators. These have different power requirements and different way of connection. This part of the document shows and explains how different hardware components were connected in the implementation of the embedded system.

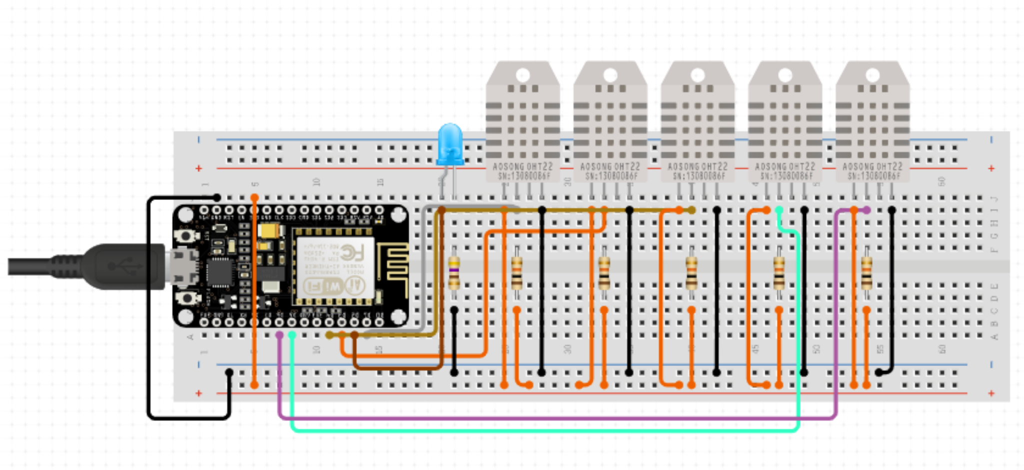


Figure 26: DHT 22 Sensors

The DHT 22 sensors are digital and they have four pins. Only three of these pins were used and these were the ground pin, VCC (+) pin and the data pin. The system is designed to accommodate multiple pins so in the diagram above, five DHT22 sensors are shown. The diagram also shows an LED which was used to provide the end-user with information on whether the whole device is powered on or not.

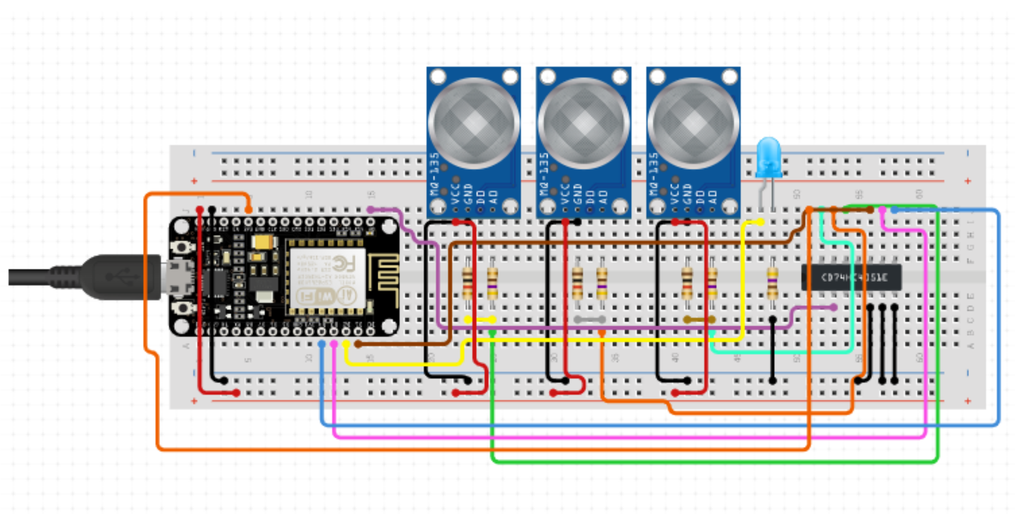


Figure 27: MQ-135 Air Quality Sensors

The diagram above shows MQ-135 Sensors which are used to monitor the quality of the air. These sensors are analog but the NodeMCU only has one analog pin. To accommodate multiple MQ-135 sensors, an analog-to-digital converter had to be used. This chip converts the analog signal from the sensors to a digital signal to be read by the digital pins on the NodeMCU.

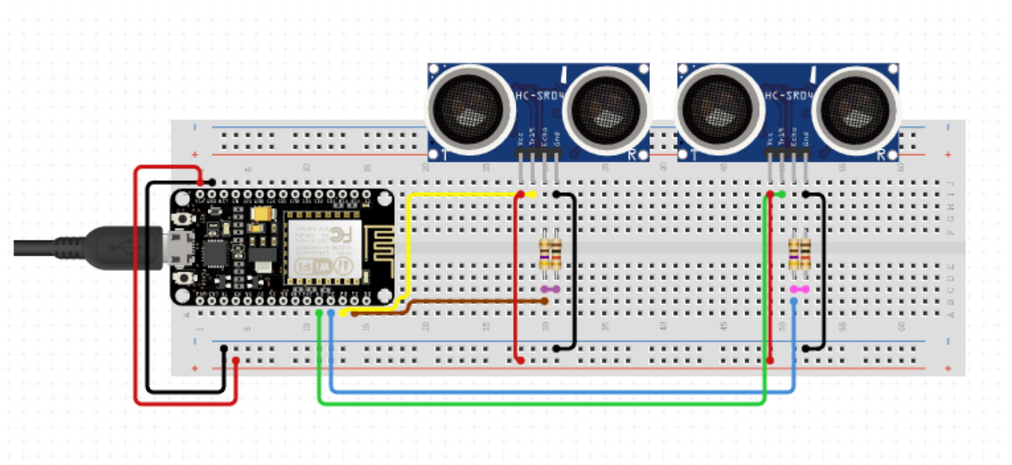


Figure 28: Ultrasonic Sensor

To monitor the level of water in the water drinkers, several methods can be used. In this project, ultrasonic sensors were used and they monitored the distance from the sensor to the surface of the water. This is then used to calculate the volume of water in the drinker. These sensors are connected as shown in the figure above.

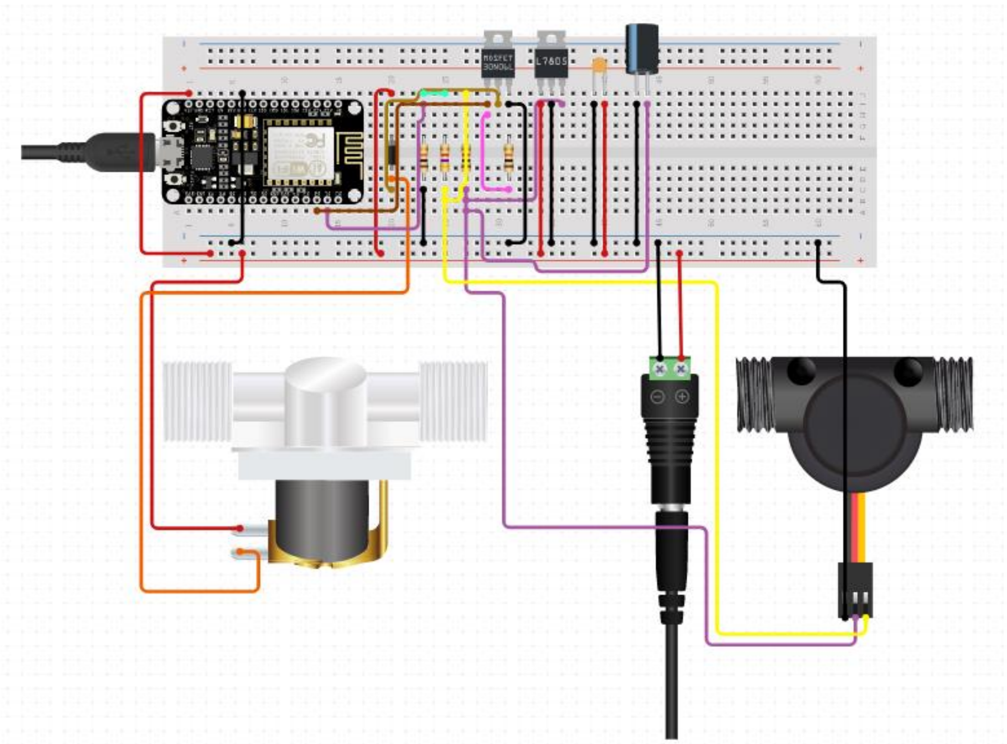


Figure 29: Waterflow Sensor and Water Valve

Drinking water is one of the essentials for the health of poultry birds. To ensure that the birds always have enough water to drink in the water drinkers, the ultrasonic sensor is used to detect the level of the water as discussed earlier. If the water level is too low, the embedded system activates the solenoid valve shown in the picture above to open the water pipes and refill the water drinkers. While refilling the drinkers, the water surface is not so stable such that it becomes difficult to know the amount of water that has been poured in the drinker to know when to stop. To overcome this, a water flow sensor is used to know the amount of water poured in the drinker. Once enough water is filled in the drinker, the system closes the water valve to stop the water flow.

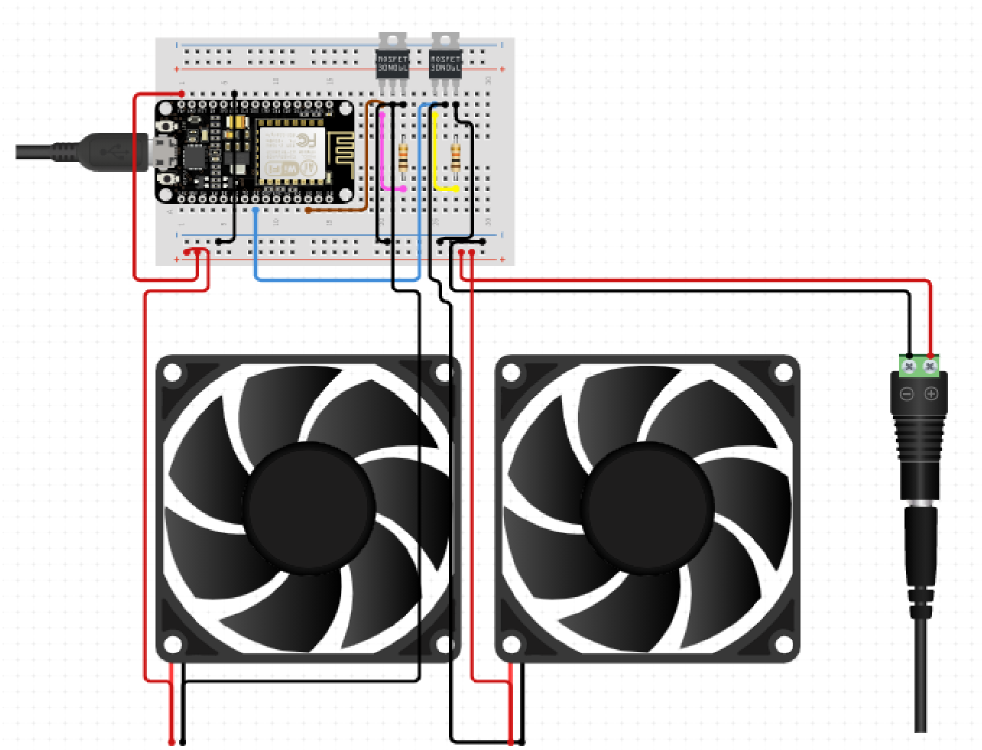


Figure 30: Exhaust Fan

When the air quality sensors detect that the air contains too many hazardous particles, the system activates the exhaust fans. These interface with the outside environment to release the hazardous gases from the room and replace it with fresh air from the outside environment.

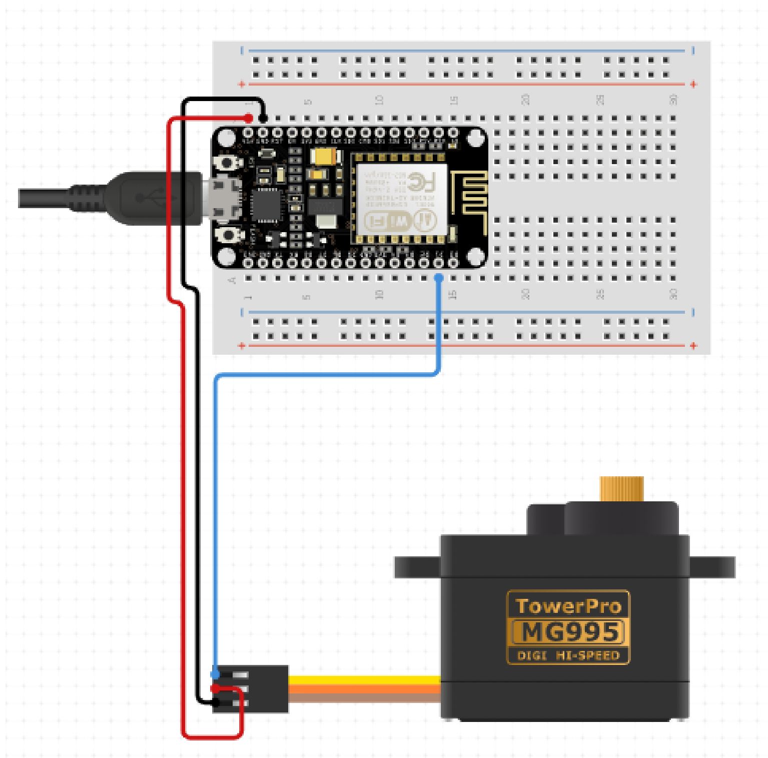


Figure 31: Servo Motor

The servo motor above is used to open and close the feed vault to refill the feeders when the amount of feed is too low. The amount of feed is detected using weight sensors since there is a direct relationship between the amount of feed and its weight. Once the feed amount is too low, the servo motor opens the food vault to refill the feed and when the amount is enough, the servo motor closes the vault.

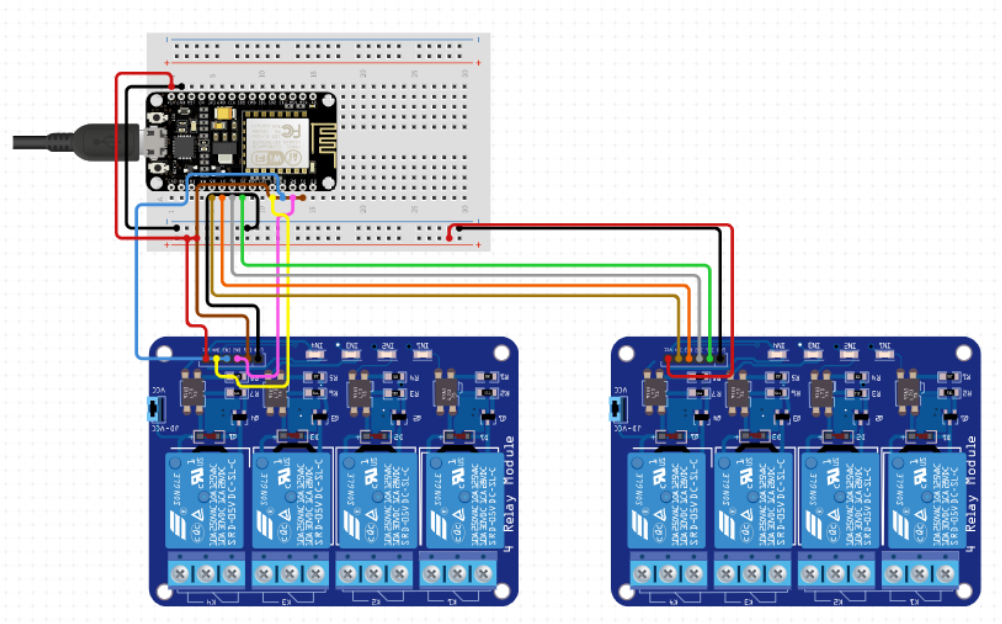


Figure 32: Relay Modules

A relay module is an electrical switch operated by an electromagnet. It is used to switch on/off other devices connected to a separate power source. All actuators, like heaters and humidifiers, that require mains power, are connected to the system using the relay modules. One of the scenarios that use this is when the temperature is too low. The system detects the low temperature using the temperature sensors discussed earlier. The system then activates the pin that is connected to the relay module with a heater connected. The heater heats the room and when the required temperature is reached, the system deactivates the pin connected to the same actuator through the relay. This is how the system operates all the actuators that require mains power.

The NodeMCU has 17 pins and they are not enough to connect all the components the system has at once. For this, a GPIO pin extender was used to increase the number of pins available to connect components to the embedded system. The pin extender is as below.

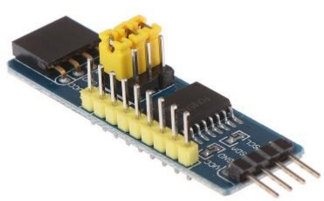


Figure 33: Pin Extender

Embedded Software

The embedded system, having being connected with the necessary sensors, actuators and their respective supporting hardware, required software to drive it and achieve the required functionality. Embedded computing systems, unlike general-purpose computing systems, have limited resources, i.e. memory, storage and computing power. Due to this, the embedded software is designed with so much care on optimizing these limited resources. This section discussed how the embedded software components were implemented in this project.

A. Sensors

The system has different kinds of sensors as shown in the Embedded Hardware section above. In the implementation of the software, each sensor was regarded as a single software object. Each kind of sensors has its class implementing the device-specific details and only providing a few necessary functions to be called from outside the class.

The sensors objects are initialized at device startup detected by detecting the connected pins from the pins reserved for each sensor. The unique address for each pin acts as the ID for the sensor which can later be used in the web dashboard interface to map sensors to actuators.

B. Actuators

Actuators are the easiest part of the embedded system to control. Most of them use an external power source and are connected to the embedded system through means of a power relay module. Each type of actuators has a set of pins reserved for it and in the same way the system is able to detect the pins that are connected and those that are not. This data shows in the web dashboard to be able to map different sensors with the actuators.

The control of actuators however faces a problem when trying to find the best way to control the actuators while using less power. An algorithm was designed in Chapter 3 and this is used every iteration of the program when controlling temperature and humidity. Once an action is determined from the algorithm, the software can activate or deactivate the pin with the respective actuator to control the environment which is the most integral part of the system.

C. Communication

One of the most important parts of the system is the ability to collect data and communicate it with a remote server and remote client application for performance monitoring. The communication part of the system also enables the remote configuration of the embedded system through a web dashboard. To achieve this, the system employs different communication methods to send data to and from the servers and client applications. The system uses Wi-Fi in the current prototype but it provides the flexibility to add more communication protocols such as LoRaWAN and 4G. Below is how the system achieves communications.

i. Telemetry

Sensor data published from the embedded system to the MQTT broker to be received by the server through the API, and the client applications for real-time monitoring by the end-users. Data is collected from each of the sensors as a JSON (JavaScript Object Notation) Object which is supported by most of the known programming languages. This JSON object consists of the sensor value, time of data collection, and the ID of the sensor that collected the data. This data is received by both the server API to persist in the database, and by the end-user application to display to the user on a live data chart.

ii. Updates

The end-user has the liberty to configure the embedded system. However, the embedded system does not have enough user interface functionality to support most of these complex processes. As such, the system uses the web dashboard interface to provide configuration methods. The user can configure which actuator responds to which sensor. This data is delivered to the embedded system as updates and it is provided through MQTT.

5. RESULTS AND DISCUSSION

The research on the design of the poultry farm control system was done as explained thoroughly in the research methodology chapter of this document. The specific objectives of the study were:

a. To design an embedded control system for a poultry farm environment.
b. To model the sensor placement for a poultry farm.
c. To model the operation schedule of actuators to reduce energy consumption
d. To design a web-based data dashboard for farm data access and analysis.

Having explained how the study was conducted to achieve these objectives, this shows and discusses the results that were obtained from the study. It shows the graphical representation of the data and what it means. This is further used to connect the meaning of the data and how it connects to achieve the overall objective of the study.

5.1. Data Representation

Temperature and humidity data were collected from a room with horizontal dimensions of 3m × 3m. This data was collected from four sensors strategically placed at the centre of the four 1.5m × 1.5m zones in the room. The collected data was as follows.

5.1.1. Temperature

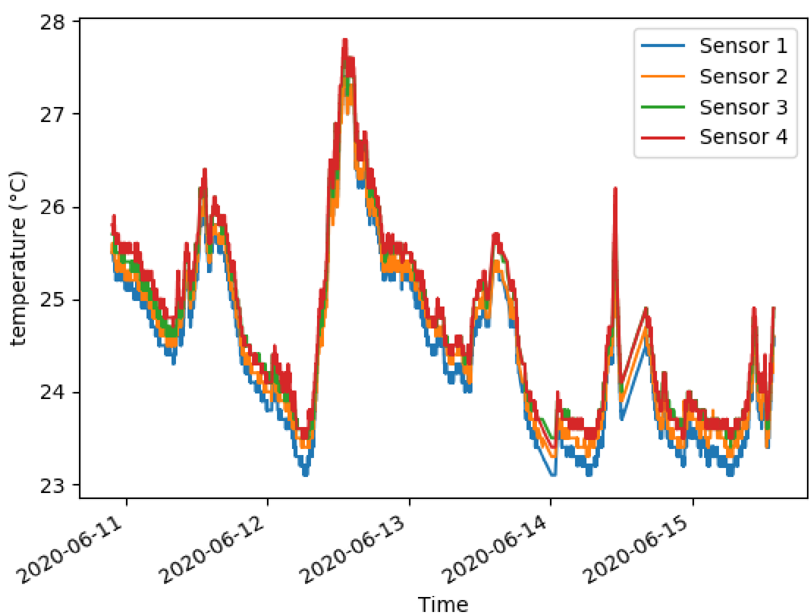


Figure 34: Temperature Data

The above figure shows the changes in temperature represented in the data from the different sensors as collected in the period of the study. The data shows that the temperature was changing as expected i.e. highest around noon and lowest at dawn. The temperature differences among the sensors, however, was very low such that the data from all the sensors had the same shape. This may show a positive correlation of the temperature among the sensors but a more careful analysis was done as shown later in this document.

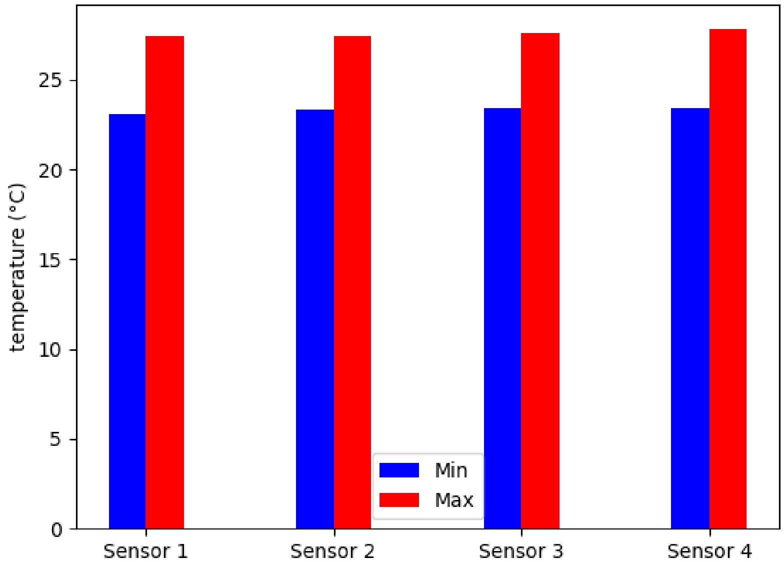


Figure 35: Temperature Ranges

The above image shows the minimum and maximum temperature as collected from the different sensors. From the figure, one can tell that the minimum and maximum values from each of the sensors are roughly the same. A more careful analysis, however, had to be done to get the exact numbers.

Table 5: Temperature Differences

	value	sensors
max difference	0.6	1 & 4
mean max difference	0.32	1 & 4
max difference standard deviation	0.09	2 & 4

The figure shows the differences in the temperature values from sensor to sensor as observed in the collected data. The maximum temperature difference was 0.6 ℃ and that was between sensor 1 and sensor 4. The maximum average difference between any two sensors throughout the study was 0.32 ℃, recorded between sensor 1 and sensor 4. This shows that the temperature in the different zones of the room was almost equal and the differences were negligible as in regards to poultry farms.

5.1.2. Humidity

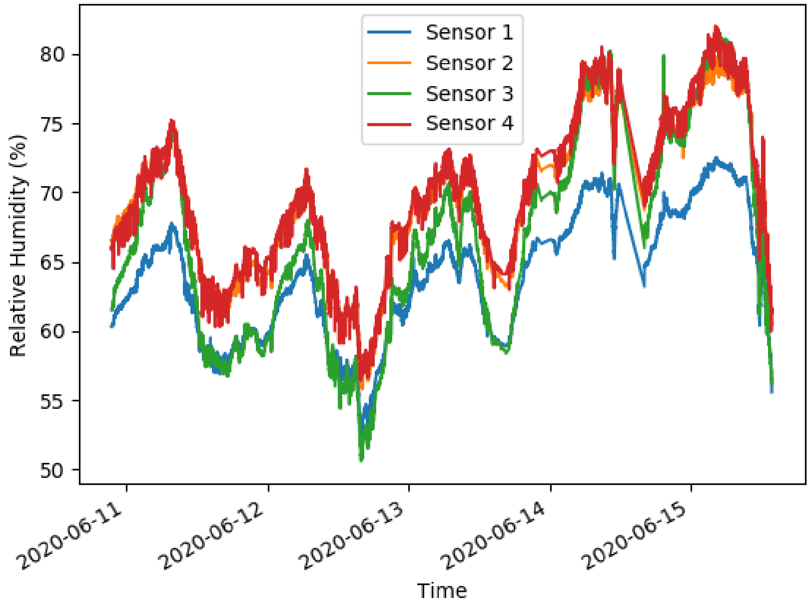


Figure 36: Humidity Data

The figure above shows the changes in the relative humidity throughout the study. The values of humidity, unlike the temperature, dropped during the day and rose at dawn. The data from the different sensors had the same shape but compared to the temperature data, there was a larger difference in the relative humidity among the different sensors.

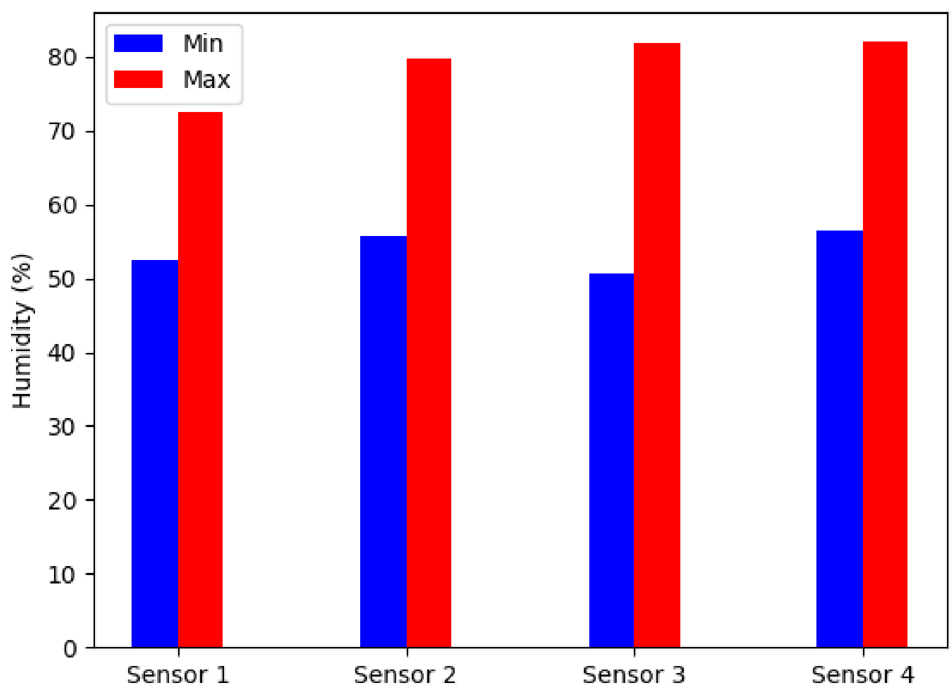


Figure 37: Humidity Ranges

The minimum and maximum values of the relative humidity from the different sensors were as shown in the figure above. The ranges of the humidity data showed more differences among the sensors than the temperature data did.

Table 6: Temperature Differences

	value	sensors
max difference	11.1	1 & 3
mean max difference	6.22	1 & 4
max difference standard deviation	2.69	1 & 3

The maximum relative humidity difference among any two sensors at a point during the period of the study was 11.1 % between sensor 1 and sensor 3. The maximum average relative humidity difference between any two sensors was 6.22% between sensors 1 and 4. The range of the relative humidity as required by the poultry birds is 50% to 80%.

5.2. Correlation

The information about the minimum and maximum differences between the values from the different sensors does not say much about the difference in behaviour among the sensors. It is important to know what happens to the parameters in one zone when the parameters in another zone change. This is called a correlation. It is a measure of how two variables are associated represented as a value from -1 to +1 [58]. A value of -1 represents a 100% negative correlation while a value of +1 represents a 100% positive correlation. Below is a graphical representation of how the values from the different sensors correlated with each other during the study.

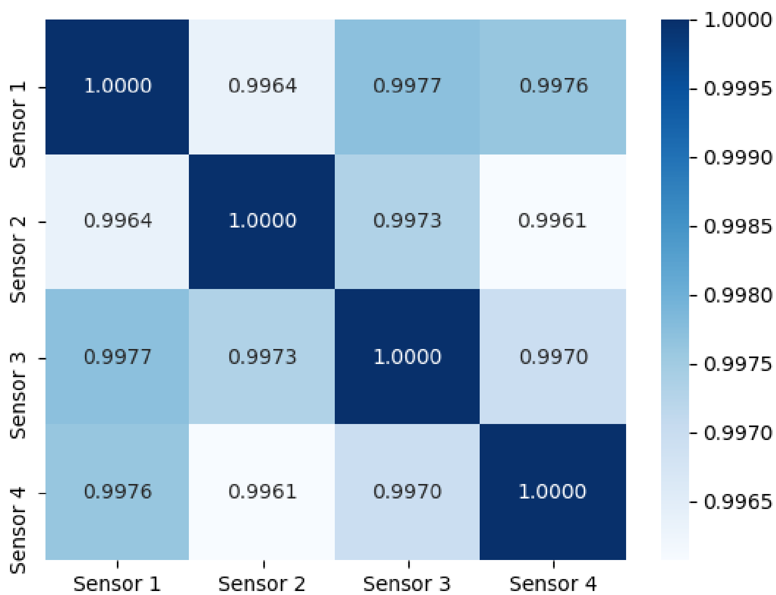


Figure 38: Temperature Correlation

As expected, every sensor has a positive correlation with itself. It would require two different sensors to have completely equal values in the whole dataset to have them correlate 1. It was not the case in this study hence the correlation values of less than 1. A correlation is considered strong if it is of a value greater than 0.7 [72]. The sensors with the least temperature correlation were sensors 2 and 4 with a correlation of 0.9961. This correlation was very strong and it shows that the temperature from all the zones in the room was behaving in the same way.

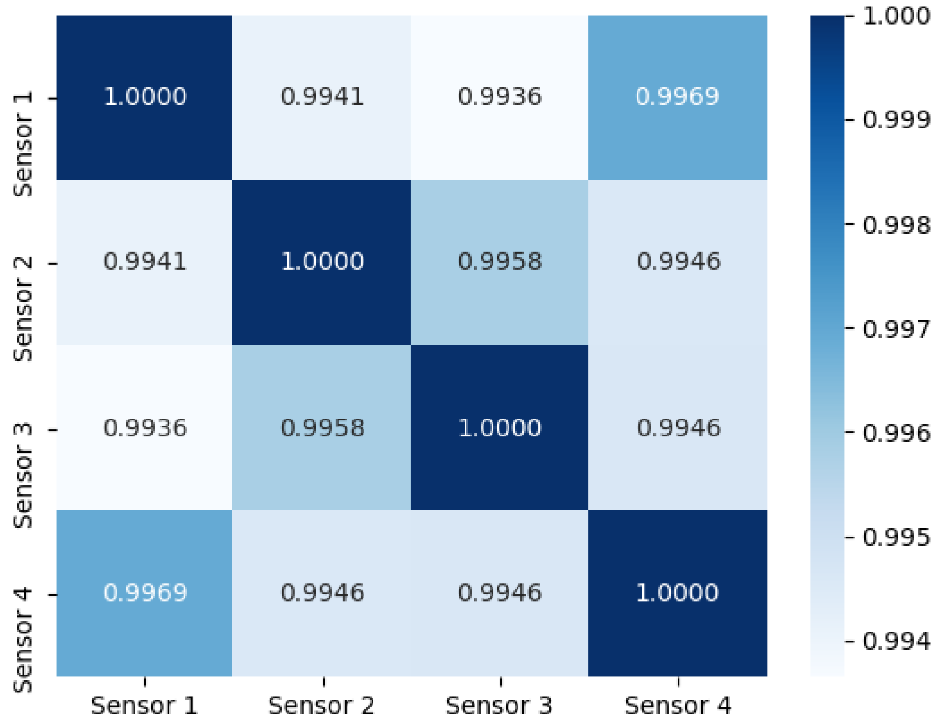


Figure 39: Humidity Correlation

The figure above shows the correlation matrix of the humidity values as collected from the four sensors in the four zones. It shows that the lowest correlation score was 0.9936 between sensor 1 and sensor 4. This also shows that the values of relative humidity from all the zones behaved in the same way.

From the temperature and humidity correlation matrices and the maximum differences in the values as collected from the sensors, it shows that the temperature and humidity from all the four zones had negligible differences with regards to poultry farms. Poultry farms have a recommended range of temperature and relative humidity are 18℃ to 31℃ and 50% to 80% respectively. These account for a tolerance of 13℃ and 30% which is not surpassed by the differences shown in the data.

5.3. Introduction of Heaters

Since the temperature in poultry farm can be out of the recommended ranges, hence the project, it was a requirement to have temperature manipulation devices. For this study, resistive heaters were considered. As explained in the research methodology chapter, three scenarios were considered depending on the number of heaters to be deployed on the farm. These were a poultry farm without a heater, with one heater and another with N heater where $N \geq 2$. In a farm with no heater, it was discussed that the heater is supposed to be placed as close to the centre of the room as possible while being away from outside interfacing walls and ventilation holes. In a more complex scenario where there is more than one heater on the farm, it was proposed that the circle packing algorithm can be used to deploy the heaters. Imaginary circles are packed in the room using the circle packing algorithm and the heaters are placed at the centres of the circles as discussed in the research methodology chapter.

Having deployed the heaters in the room, the next step is to mount the sensors in the farm to start monitoring the environment. The following shows the results of applying the proposed sensor placement algorithm to position the sensors in a poultry farm.

5.3.1. No Heaters

It was discussed that the walls have an effect on the temperature and so does the roof and the ventilation holes. This is so since the walls, roof and ventilation holes are affected by the outside environment. To provide effective monitoring, the sensor has to be placed away from these objects and the best place to fit this constraint is the centre of the farm. Therefore, in a poultry farm with no heaters, the sensors have to be placed on the centre of the farm.

5.3.2. One Heater

It was discussed that in this scenario, the heater would be placed in the centre of the room to increase its efficiency. It also discussed that a for a temperature sensor to provide a true reflection of what the temperature of the room is, it has to be placed away from any heat sources. Additionally, it has to be placed away from walls and ventilation holes. Following these recommendations, it was proposed that it is best to place the sensor midway between the walls of the farm and the heat source. That position provides the average temperature of the farm since it has open air-circulation and the heat spreads out evenly in all directions.

5.3.3. N Heaters

The biggest contribution and application of the proposed algorithm comes to application in this scenario. Having more than one heater requires the deployment of multiple sensors to monitor the temperature from the zones affected by each of the heaters. It was explained in the research methodology chapter on how the circle packing algorithm can be used to deploy the heaters in this scenario. The more pressing issue is how to deploy the sensors in this scenario to achieve efficient monitoring. An algorithm was designed and below is the result of applying this algorithm in a room when heaters are packed as the Hexagonal Lattice shown in the research methodology chapter.

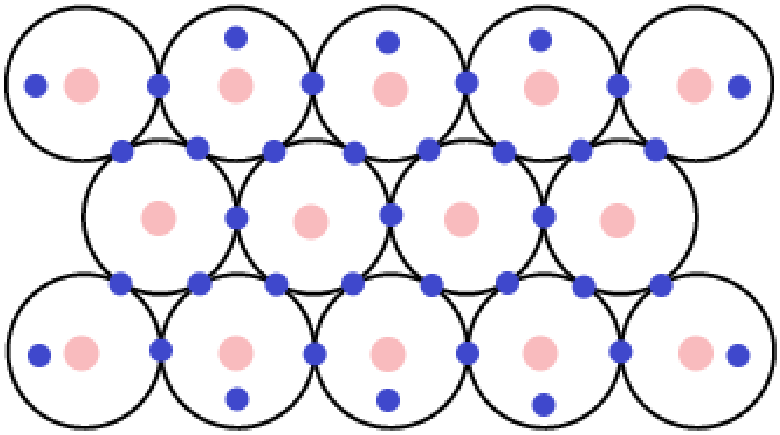


Figure 40: Sensor Arrangement

The above picture shows the results after placing heaters and sensors on a poultry farm. The circles are what are used to decide on the position of the heaters. The red dots represent the position of heaters in the farm and, as shown, these heaters are on the centres of the circles. Using the sensor positioning algorithm shown in the research methodology chapter, the sensors are positioned as the blue dots in Figure 40. The algorithm ensures the placement of sensors on all the common tangents of any two circle that contact each other. The sensors placed on such locations are used to monitor the temperature in the two zones represented by the two circles. Depending on the scenario, each zone may have two or more sensors. The mean temperature of the sensors inside and on the circumference of each zone is what is considered to be the temperature of that zone. This important since it can be used to determine any malfunctioning heaters or sensors easily by comparing with the readings from the neighbouring sensors.

5.4. Actuator Control Algorithm

An algorithm was designed to control temperature and humidity while considering the effect of one on the other. The hypothesis was that, if the relationship between temperature and humidity was to be considered while controlling them, the controlling efficiency would increase. The algorithm was implemented in python and so was the legacy algorithm which controls humidity and temperature without considering the effect of one on the other. An experiment was then designed to compare the efficiency of these two algorithms. To achieve that, the data to run the algorithms had to be modified to introduce more scenarios where heating, cooling, humidification and dehumidification were required. After modifying the data, it was as follows:

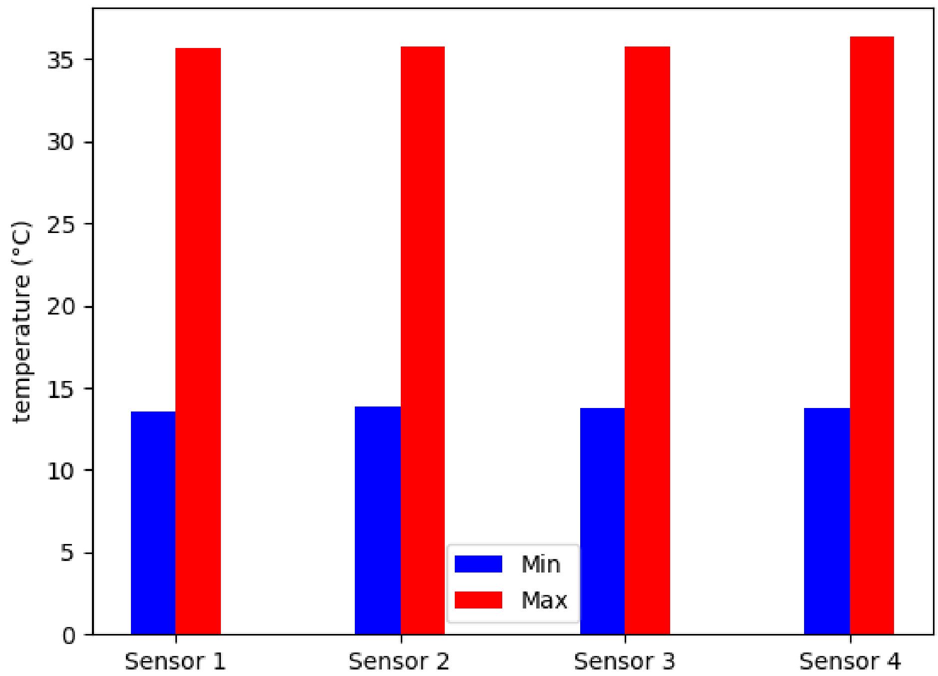


Figure 41: Modified Temperature Range

It is clear from Figure 41 that the range of the temperature data increased from 13.58 ℃ to 36.36 ℃ as compared to Figure 35 where the range was from 23.1 ℃ to 27.8 ℃.

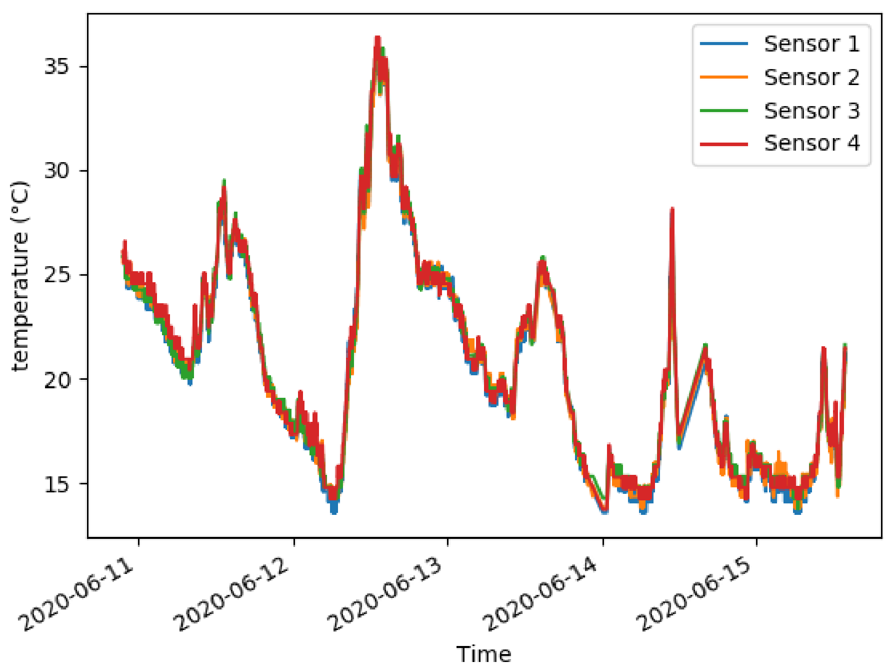


Figure 42: Modified Temperature Data

Figure 42 also shows that the shape of the data did not change but it just got extended when compared with the data shown in Figure 34.

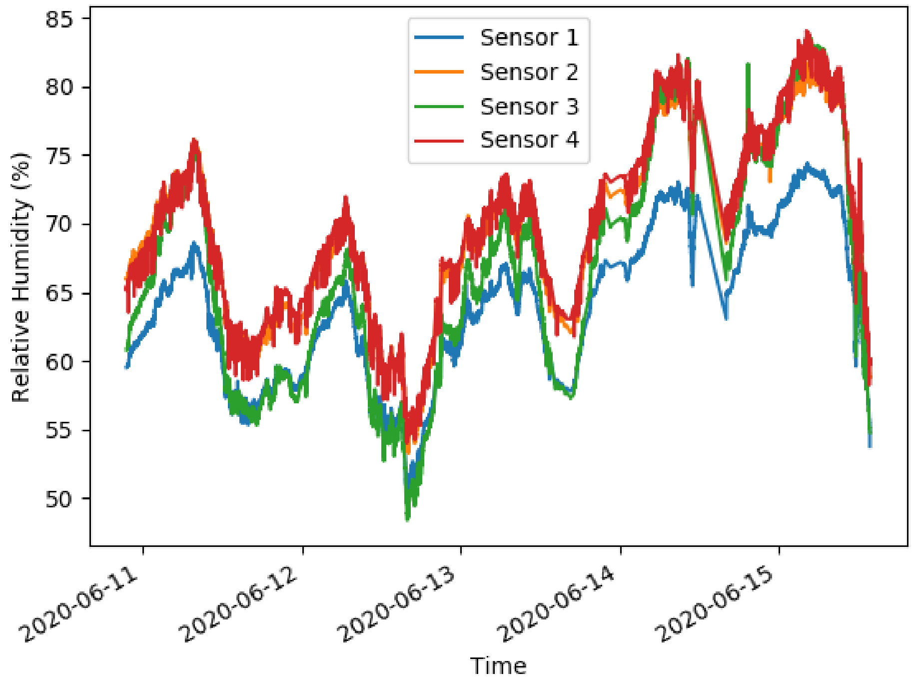


Figure 43: Modified Humidity Data

The shape of the humidity data is also unchanged but only extended when Figure 43 is compared with Figure 36.

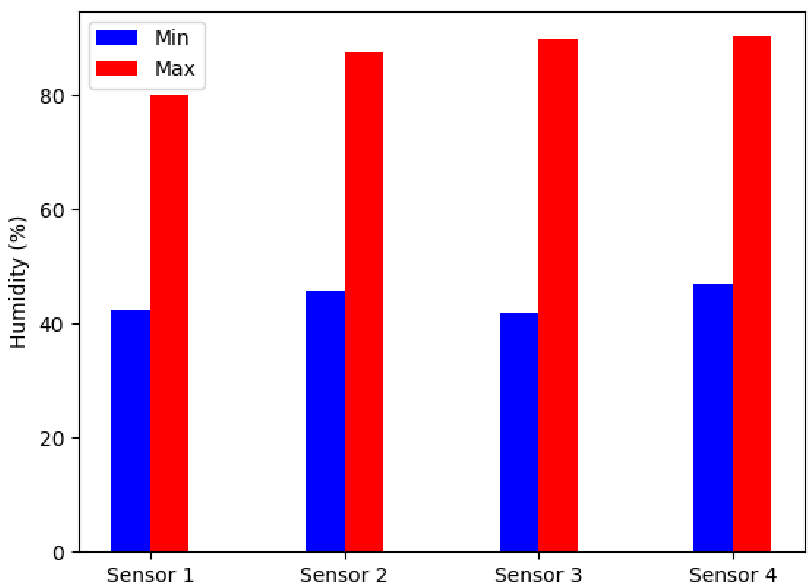


Figure 44: Modified Humidity Ranges

Figure 44 compared to Figure 37 shows that the range of the humidity was also changed. This data provides scenarios where humidification and dehumidification would be required.

5.4.1. Simulation

Having the data prepared for simulation, the two algorithms were simulated as explain in the research methodology chapter and below are the results of the simulations.

Table 7: Simulation Results

	Heater Time (s)	Heater switch Count	Cooler Time (s)	Cooler switch Count	Humidifier Time (s)	Humidifier switch Count	Dehumidifier Time (s)	Dehumidifier switch Count
Legacy Algorithm	654	55	555	44	52	37	94	41
Proposed Algorithm	653	4	553	22	3	2	56	21

The data in Table 7 show the results of simulating the two algorithms on the same data. It shows the number of seconds each actuator stays on and the number of times it is switched on/off. Using the proposed algorithm, the amount of time the heater and cooler stay on was almost the same, with the proposed algorithm having a slightly less running time. However, the runtime of the humidifier and dehumidifier is greatly reduced. The number of times the actuators turn on/off is also reduced greatly when using the proposed algorithm. Rapid on/off switching of the actuators reduces the lifespan of the actuators and consumes more electrical energy used when switching the devices. With this in light, it shows that the proposed algorithm is more efficient than the legacy algorithm.

6. CONCLUSIONS AND RECOMMENDATIONS

6.1.Conclusion

This research aimed at designing an intelligent poultry farm control system for small-scale poultry farms. Using quantitative methods, a sensor placement model was developed based on the location of actuators and it was shown that it can effectively monitor the environment in a poultry farm. An actuator operation algorithm was also designed to control the temperature and humidity in a poultry house while intelligently considering the relationship between temperature and humidity. It was shown that this algorithm reduces the time actuators spend in operation as well as avoid the operation of the actuators in short bursts. This in turn reduces the electrical energy used to control the environment in a poultry farm and poses fewer health problems to the actuators such that they may not require as much maintenance. The sensor placement model and the actuator control algorithm are then used in the design of an embedded system to act as a sensor node in the wireless sensor network of poultry farm control systems. These embedded systems are then connected to a cloud application which contains an Application Programming Interface and a web data-dashboard for real-time monitoring of the control system and the configuration of different parts of the system. As a whole, a poultry farm control system has been developed to be more efficient in the way it monitors the environment hence having a more informed decision on when and where to have which parameter of the environment controlled. This saves the energy which would be used to control zones that were not necessary to be controlled in the farm. The system also uses its embedded intelligence to use less electrical energy when controlling the environment in the poultry farm. Combined, these aspects make the poultry farm control system more efficient and hence more suitable for use by small-scale farmers.

6.2.Recommendations

The research showed that there is so much potential in improving the energy efficiency of poultry farm control systems. An actuator control algorithm was developed based on the known relationships between humidity and temperature. As future research, it would be interesting to develop an algorithm which would have sensors to monitor the energy used by each of the actuators and how effectively to control the respective environmental parameters. This information can then be fed to an algorithm which would update the control algorithm to have more efficient control of the actuators. This would be a self-learning model and it has the potential to produce better results.

Another interesting topic would be the development of the same control system but with a more generic use for any controlled environment i.e. greenhouses and livestock houses in general. This would require intelligence in the system to differentiate the methods of operation depending on the target environment’s requirements. These systems have the potential to improve the agricultural sector in the developing world which relies heavily on small-scale farmers.

References

[1] Cambridge Dictionary, “POULTRY | meaning in the Cambridge English Dictionary,” Cambridge University Press, 2020.

[2] J. Joardar and M. Rahman, “Poultry feather waste management and effects on plant growth,” *International Journal of Recycling of Organic Waste in Agriculture*, vol. 7, pp. 183-188, 2018.

[3] Food and Agriculture Organization, “Gateway to poultry production and processing,” Food and Agriculture Organization, 2020.

[4] A. Ghafoor, H. Badar, M. Hussain and N. Tariq, “An empirical estimation of the factors affecting demand and supply of poultry meat,” *Pakistan Veterinary Journal,* vol. 30, no. 3, pp. 172-174, 2010.

[5] T. Bagust, “Poultry health and diseases in developing countries,” [Online]. Available: http://www.fao.org/3/a-al729e.pdf. [Accessed 5 May 2020].

[6] A. Ranjan, R. Sinha, I. Devi, A. Rahim and S. Tiwari, “Effect of Heat Stress on Poultry Production and their Managemental Approaches,” *International Journal of Current Microbiology and Applied Sciences,* vol. 8, no. 2, pp. 1548-1555, 2019.

[7] I. Kilic and E. Yaslioglu, “Ammonia and carbon dioxide concentrations in a layer house,” *Asian-Australasian Journal of Animal Sciences,* vol. 27, no. 8, pp. 1211-1218, 2014.

[8] G. Corkery, S. Ward, C. Kenny and P. Hemmingway, “Incorporating Smart Sensing Technologies into the Poultry Industry,” *Journal of World's Poultry Research,* vol. 3, no. 4, pp. 106-128, 2013.

[9] A. P. Jr and D. Bruno, “Challenges Facing the Global Poultry Industry to 2020,” in *Australian Poultry Science Symposium*, 2020.

[10] C. W.D., K. A., H. T. and R. P.B., *Animal Feed Science and Technology,* vol. 60, no. 3, pp. 311-319, 1996.

[11] T. Park and J. Y. Han, “Genetic modification of chicken germ cells,” *Annals of the New York Acadamy of Sciences,* vol. 1271, pp. 104-109, 2012.

[12] K. K. Verma, V. Singh, S. L. Gupta, J. Yadav and A. K. Verma, “Environmentally Controlled House - In Poultry Production,” *Poultry Line,* 2014.

[13] B. Ghazal, K. Al-Khatib and K. Chahine, “A poultry farming control system using a ZigBee-based wireless sensor network,” *International Journal of Control and Automation,* vol. 10, no. 9, pp. 191-198, 2017.

[14] OIE (Office International des Epizooties), “Terrestial Animal Health Code,” OIE, Paris, France, 2011.

[15] N. B. Sassi, X. Averós and I. Estevez, "Technology and Poultry Welfare," *Animals,* vol. 6, no. 10, pp. 1-21, 2016.

[16] Y. M. Wang, Q. P. Meng, Y. M. Guo and Y. Z. Wang, "Effect of atmospheric ammonia on growth performance and immunological response of broiler chickens," *Journal of Animal and Veterinary Advances,* vol. 9, no. 22, pp. 2802-2806, 2010.

[17] I. Kilic and E. Yaslioglu, "Ammonia and carbon dioxide concentrations in a layer house," *Asian-Australasian Journal of Animal Sciences,* vol. 27, no. 8, pp. 1211-1218, 2014.

[18] G. Corkery, S. Ward, C. Kenny and P. Hemmingway, "Incorporating Smart Sensing Technologies into the Poultry Industry," *Journal of World's Poultry Research,* vol. 3, no. 4, pp. 106-128, 2013.

[19] P. H. Zimmerman, P. Koene and J. A. Van Hooff, "The vocal expression of feeding motivation and frustration in the domestic laying hen, Gallus gallus domesticus," *Applied Animal Behaviour Science,* vol. 69, no. 4, pp. 265-273, 2000.

[20] E. H. Leone and I. Estevez, "Use of space in the domestic fowl: separating the effects of enclosure size, group size and density," *Animal Behaviour,* vol. 76, no. 5, pp. 1673-1682, 2008.

[21] H. Okada, T. Itoh, K. Suzuki and K. Tsukamoto, "Wireless sensor system for detection of avian influenza," in *IEEE Sensors Conference*, Warwick, 2009.

[22] H. Okada, K. Suzuki, K. Tsukamoto and T. Itoh, "Applicability of Wireless activity sensor network to avian," *Journal of Sensor Technology,* vol. 4, no. 1, pp. 18-23, 2014.

[23] A. Aydin, O. Cangar, S. E. Ozcan, C. Bahr and D. Berckmans, "Application of a fully automatic analysis tool to assess the activity of broiler chickens with different gait scores," *Computers and Electronics in Agriculture,* vol. 73, no. 2, pp. 194-199, 2010.

[24] A. F. Cordeiro, I. A. Nääs and D. D. Salgado, "Field evaluation of broiler gait score using different sampling methods," *Revista Brasileira de Ciencia Avicola,* vol. 11, no. 3, pp. 149-154, 2009.

[25] D. Shinder, M. Rusal, M. Giloh and S. Yahav, "Effect of repetitive acute cold exposures during the last phase of broiler embryogenesis on cold resistance through the life span," *Poultry Science,* vol. 88, no. 3, pp. 636-646, 2009.

[26] H. M. Raza, H. Ashraf, K. Shahzad, M. Sultan, T. Miyazaki, M. Usman, R. R. Shamshiri, Y. Zhou and R. Ahmad, "Investigating applicability of evaporative cooling systems for thermal comfort of poultry birds in Pakistan," *Applied Sciences (Switzerland),* vol. 10, no. 13, 2020.

[27] R. Coffey, "Chapter 8 - Heating Methods and Equipment," in *Poultry Production Manual*, University of Kentucky.

[28] D. A. Alchalabi, "Cooling Poultry Houses Basic Principles of Humidity and Temperature," University of Baghdad, 2015.

[29] R. A. Bucklin, J. P. Jacob, F. B. Mather, J. D. Leary and I. A. Naas, “Tunnel Ventilation of Broiler Houses,” UF/IFAS extension, University of Florida, 2018.

[30] R. Coffey, “Chapter 9: Hot Weather Ventilation,” in *Poultry Production Manual*, Department of Animal and Food Sciences, University of Kentucky.

[31] F. Azzola, “DZone IoT Zone,” 17 April 2018. [Online]. Available: https://dzone.com/articles/introduction-to-iot-sensors. [Accessed 9 May 2020].

[32] P. G. Sinha, “Introduction and Classification of Sensors,” 2017.

[33] S. Gervais-Ducouret, “Next smart sensors generation,” *SAS 2011 - IEEE Sensors Applications Symposium, Proceedings,* pp. 193-196, 2011.

[34] K. G. Panda, D. Agrawal, A. Nshimiyimana and A. Hossain, “Effects of environment on accuracy of ultrasonic sensor operates in millimetre range,” *Perspectives in Science,* vol. 8, no. February 2017, pp. 574-576, 2016.

[35] S. Al-Sarawi, M. Anbar, K. Alieyan and M. Alzubaidi, “Internet of Things (IoT) Communication Protocols: Review,” in *2017 8th International Conference on Information Technology (ICIT)*, 2017.

[36] A. Augustin, J. Yi, T. Clausen and W. M. Townsley, “A study of Lora: Long range & low power networks for the internet of things,” *Sensors (Switzerland),* vol. 16, no. 9, pp. 1-18, 2016.

[37] N. Naik, “Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTPNaik, N. (2017). Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP,” in *IEEE International Symposium on Systems Engineering*, 2017.

[38] S.-y. Lee, I.-b. Lee and G.-y. Park, “Optimal sensor placement for temperature monitoring inside broiler houses,” in *10th International Livestock Environment Symposium (ILES X)*, 2018.

[39] V. Akbarzadeh, J. C. Lévesque, C. Gagné and M. Parizeau, “Efficient sensor placement optimization using gradient descent and probabilistic coverage,” *Sensors (Switzerland),* vol. 14, no. 8, pp. 15525-15552, 2014.

[40] Y. Zhou, A. Bidarmaghz, G. A. Narsilio and L. Aye, “Heating and Cooling Loads of a Poultry Shed in Central Coast, NSW, Australia,” in *World Sustainable Built Environment Conference*, Hong Kong, 2017.

[41] K. Little, “IoT Systems: Sensors and Actuators,” DZone, 6 August 2019. [Online]. Available: https://dzone.com/articles/iot-systems-sensors-and-actuators. [Accessed 9 May 2020].

[42] S. Madakam, R. Ramaswamy and S. Tripathi, “Internet of Things (IoT): A Literature Review,” *Journal of Computer and Communications,* vol. 3, no. 5, pp. 164-173, 2015.

[43] P. Wolkoff, “Indoor air humidity, air quality and health - An overview,” *International Journal of Hygiene and Environmental Health,* vol. 221, no. 3, pp. 376-390, 2018.

[44] J. Serra, D. Pubill, A. Antonopoulos and C. Verikoukis, “Smart HVAC control in IoT: Energy consumption minimization with user comfort constraints,” *Scientific World Journal,* 2014.

[45] A. Trevennor, “A Brief History of Microcontrollers,” in *Practical AVR Microcontrollers*, 2012, pp. 3-11.

[46] Altera Corporation, “Bare-Metal, RTOS, or Linux? Optimize Real-Time Performance with Altera SoCs,” 2014.

[47] A. Gerber and J. Romeo, “Choosing the best hardware for your next project,” 2017.

[48] B. Kang and H. Choo, “An experimental study of a reliable IoT gateway,” *ICT Express,* vol. 4, no. 3, pp. 130-133, 2018.

[49] P. D. Rosero-Montalvo, V. F. L. Batista, E. A. Rosero, E. D. Jaramillo, J. A. Caraguay, J. Pijal-Rojas and D. H. Peluffo-Ord´o˜nez, “Intelligence in Embedded Systems: Overview and Applications,” in *Proceedings of the Future Technologies Conference (FTC)*, vol. 1, 2018, pp. 874-883.

[50] E. Sonaiya and S. Swan, “SMALL-SCALE POULTRY PRODUCTION: CHAPTER 4,” Food and Agriculture Organization of the United Nations, 2004. [Online]. Available: http://www.fao.org/3/y5169e/y5169e05.htm. [Accessed 19 July 2020].

[51] E. Bustamante, E. Guijarro, F. J. García-Diego, S. Balasch, A. Hospitaler and A. G. Torres, “Multisensor system for isotemporal measurements to assess indoor climatic conditions in poultry farms,” *MDPI/Sensors,* vol. 12, no. 5, pp. 5752-5774, 2012.

[52] “NodeMCU Documentation,” GitHub, [Online]. Available: https://nodemcu.readthedocs.io/en/master/. [Accessed 5 July 2020].

[53] ESPRESSIF, “ESP8266,” ESPRESSIF, [Online]. Available: https://www.espressif.com/en/products/socs/esp8266. [Accessed 2020].

[54] Arduino, “Getting Started with the Arduino Uno Wi-Fi,” Arduino, [Online]. Available: https://www.arduino.cc/en/Guide/ArduinoUnoWiFi. [Accessed 2020].

[55] Adafruit, “AM2302 (WIRED DHT22) TEMPERATURE-HUMIDITY SENSOR,” Adafruit, [Online]. Available: https://www.adafruit.com/product/393#:~:text=The%20AM2302%20is%20a%20wired,no%20analog%20input%20pins%20needed).. [Accessed 7 July 2020].

[56] M. Masse, API DESIGN RULEBOOK, O'Reilly Media, Inc, 2011.

[57] MuleSoft, “Types of APIs and how to determine which to build,” Mulesoft, 2020. [Online]. Available: https://www.mulesoft.com/resources/api/types-of-apis. [Accessed August 2020].

[58] P. Schober, C. boer and L. A. Schwarte, “Correlation Coefficients: Appropriate Use and Interpretation,” *Anesthesia & Analgesia,* vol. 126, no. 5, pp. 1763-1768, May 2018.

[59] D. Meyer and D. Thevenard, “PsychroLib: a library of psychrometric functions to calculate thermodynamic properties of air,” *Journal of Open Source Software,* vol. 4, no. 33, p. 1137, 2019.

[60] B. G. Kyle, Chemical and Process Thermodynamics, Englewood Cliffs, NJ: Prentice-Hall, 1984.

[61] I. Dincer and M. A. Rosen, “Exergy Analysis of Psychrometric Processes,” in *Exergy*, 2013, pp. 83-100.

[62] J. Huang, “A simple accurate formula for calculating saturation vapor pressure of water and ice,” *Journal of Applied Meteorology and Climatology,* vol. 57, no. 6, pp. 1265-1272, 2018.

[63] B. H. A. Baloon, “How to Calculate Air Density,” 2020. [Online]. Available: https://www.brisbanehotairballooning.com.au/calculate-air-density/.

[64] M. Hifi and R. M'Hallah, “A literature review on circle and sphere packing problems: Models and methodologies,” *Advances in Operations Research,* vol. 2009, 2009.

[65] P. Szabó, M. Markót, T. Csendes, E. Specht, L. Casado and I. García, New Approaches to Circle Packing in a Square, Springer, 2007.

[66] E.Specht, “The best known packings of equal circles in a rectangle with variable aspect ratio (complete up to N = 500),” 21 June 2018. [Online]. Available: http://www.packomania.com/crc_var/. [Accessed 19 July 2020].

[67] IBM Knowledge Centre, “Planning information systems with Business Functions,” IBM, [Online]. Available: https://www.ibm.com/support/knowledgecenter/SSRA3Z_8.9.1/com.ibm.ima.po_soa/comp/vocab/biz%20funcs/bus_fun_soa_is_plan.html. [Accessed 28 09 2020].

[68] G. B. Shelly and H. J. Rosenblatt, Systems Analysis and Design, vol. 9, Course Technology, Cengage Learning, 2012.

[69] E. White, Making Embedded Systems, O'Reilly Media, Inc, 2011.

[70] S. Iftikhar, H. Amir, Z. Khadim and K. Bilal, “Farmer ’ s Literacy Rate as Key Driver in Food Production and Food Security : An Empirical Appraisal from Punjab , Pakistan,” *European Online Journal of Natural and Social Sciences,* vol. 4, no. 4, pp. 683-690, 2015.

[71] T. Otwell, “Laravel,” Laravel, 2011-2020. [Online]. Available: https://laravel.com/. [Accessed 05 01 2020].

[72] D. Mindrila and P. Balentyne, “Scatterplots and Correlation,” in *The Basic Practice of Statistics* , 2017.