2023-08

Brix and alcohol content monitoring using wireless sensor network

Willa, Victor

NM-AIST

BRIX AND ALCOHOL CONTENT MONITORING USING WIRELESS SENSOR NETWORK

Victor Willa

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

Arusha, Tanzania

August, 2023

ABSTRACT

The fermentation process plays a vital role in the production of wine and beer by converting Brix (Sugar) into alcohol. Consequently, monitoring this fermentation is crucial for breweries to ensure the quality of their products. This project's main objective was to enhance alcohol quality monitoring the by-products of fermentation, namely Brix (Sugar concentration) and alcohol levels. Each stage of fermentation results in varying Brix and alcohol percentages by volume. To achieve this, a system utilizing wireless communication protocols was proposed. Sensor nodes were strategically placed to collect data, which was then transmitted to a centralization station for monitoring and visualization. The sampling technique used was non-probability purposive, as it allowed the project to gather essential information from knowledgeable personnel in the field of study, contributing to a deeper understanding of the problem. The implementation of an IoT (Internet of Things) and Wireless Sensor Network solution proved to be highly advantageous for the brewery industry. This approach facilitated the seamless transfer of real-world fermentation processes into the digital realm, enabling optimization of these processes. Through this project study a novel and automated method with commendable accuracy was developed for estimating Brix and alcohol content during fermentation process. This innovative solution promises to improve the overall quality of alcohol production and enhance the efficiency of monitoring and control in brewery.

DECLARATION

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

Victor Willa
_____
Name of Candidate   **Signature**   **Date**

The above declaration is confirmed by:

Prof. Anael Sam
_____
Name of Supervisor 1   **Signature**   **Date**

Dr. Devotha Nyambo
_____
Name of Supervisor 2   **Signature**   **Date**

COPYRIGHT

CERTIFICATION

The undersigned certify that they have read and hereby recommend for acceptance by The Nelson Mandela African Institution of Science and Technology, a project report titled *"Brix and Alcohol content monitoring using Wireless Sensor Network"* in partial fulfillment of the requirements for the degree of Master of Science in Embedded and Mobile Systems of the Nelson Mandela African Institution of Science and Technology.

Prof. Anael Sam

| **Name of Supervisor 1** | **Signature** | **Date** |

Dr. Devotha Nyambo

| **Name of Supervisor 2** | **Signature** | **Date** |

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

API   Application Programming Interface

DFD   Data Flow Diagram

DHCP   Dynamic Host Controll Protocoll

ICT   Information Communication Technology

IDE   Integrated Development Environment

IOT   Internet of Things

IP   Ingress Protection

IP   Internet Protocol

LIPO   Lithium Polymer

LTD   Limited

MAC   Media Access Controller

MVC   Model View Controller

PC   Personal Computer

PCB   Printed Circuit Board

SG   Specific Gravity

WIFI   Wireless Fidelity

WSN   Wireless Sensor Network

CHAPTER ONE

INTRODUCTION

1.1   Background of the Problem

Fermentation is widely used process in alcohol production. Banana juice as one example contain various components, including sugar, vitamin C, brix. These elements contribute to the characteristics of the quality of the final product (Akubor & Ukwuru, 2003). The beverage industries conduct regularly monitoring of alcohol and sugar concentration standards required for the resulting wine. As a result, the fermentation process must be closely monitored and checked at frequent intervals to ensure that the final product meets the expected levels of quality and adheres to the specified standards.

Fermentation is a biochemical phenomenon where in yeast converts sugars in solutions into alcohol and carbon dioxide. This vital process plays a fundamental role in the production of beverages like wine and beer (Maicas, 2020). Due to these fermentation plays central role in beer production hence monitoring entire activities are essential to breweries.

Brix is a technical term that denotes the percentage by weight of sugar solids present in a liquid. It serves as a measure to express the sugar content within a solution, particularly in fruits and vegetables. By indicating the estimated sugar content, Brix becomes a valuable indicator for consumer ratings as it directly influences sweetness of the product (Kleinhenz & Bumgarner, 2013).

Internet of Things (IoT) based application and WSN architecture are adapted to facilitate monitoring these processes during fermentation of wine to increase accuracy, reliability and traceability (Botha *et al.,* 2019). In general, a Wireless Sensor Network (WSN) comprises numerous sensor nodes that can communicate with one another by generating signals. Each of these sensor nodes operates under resource limitations. Once deployed, these sensor nodes autonomously manage their organization and utilize their network infrastructure. Additionally, these sensor devices can respond to request from a monitoring tool, executing specific instructions or furnishing information as needed. The WSN forms a network of interconnected devices, known as nodes, which collaboratively collect and transmit data through wireless communication, enabling efficient information gathering and monitoring. (Buratti *et al.,* 2009). A single node WSN as performedin this research suffers from scalability.

The use of optical instruments which use refractive index which is then converted into brix scale through the use of digital handheld devices has been popular used (Buczinski & Vandeweerd, 2016). Using Wireless Sensor Network these could be easily implemented on brew industries since only sensor node reading is required and the result is known.

In this work a device for estimating alcohol and brix content was developed. The present method requires the use of sophisticated and expensive equipment's through the use of alcohol and brix refractometer also hydrometer where by these devices are not practical in a largescale and are expensive (Kanyathare & Peiponen, 2018).

1.2   Statement of the Problem

Determination of alcohol content in factories especially at Raha beverage company is essential due to regulations methods and quality. Currently hand-held refractometer is used but strongly depends on expertise of the operator and human accuracy. Brix is calculated primarily using gravimetric and refractive methods such as the hydrometer, westphal balance, and Pycnometer. The hydrometer is by far the most popular of these. However, due to human error, these calculation methods are both inefficient and unreliable. (Dongare *et al.,* 2015). Manually collection of fermentation parameters from varioustanks is tiresome and operator can easily forget or make mistakes in updating tables.

Traditional brix and alcohol content monitoring is done at single storage tank at a time which makes difficult representation of data (Lopes, 2015). One of the important parameter to control in Beer fermentation is Sugar concentration (Brix) (Grassi *et al.,* 2014) which is critical parameter in producing quality beverage.

Apart from this other factor include the following:

(i)   The existing manual process is neither reliable nor efficiency.

(ii)   Measurement is done through hand held refractometer and one may misread due to fatigue.

(iii)   The data collection is time consuming.

1.3   Rationale of the Study

During fermentation process the required threshold of Sugar Concentration (Brix) in Beverage products should be 5-15. Fermentation process is heavily dependent on alcohol and Sugar concentration (Brix), different levels of Sugar concentration produce different flavor and aroma (Lu *et al.,* 2018). Various properties must be daily monitored such as sugar content, PH, alcohol content during the fermentation stage before the wine is harvested.

Figure 1 shows the relationship and changes between Brix level and alcohol percent by volume with respect to time.
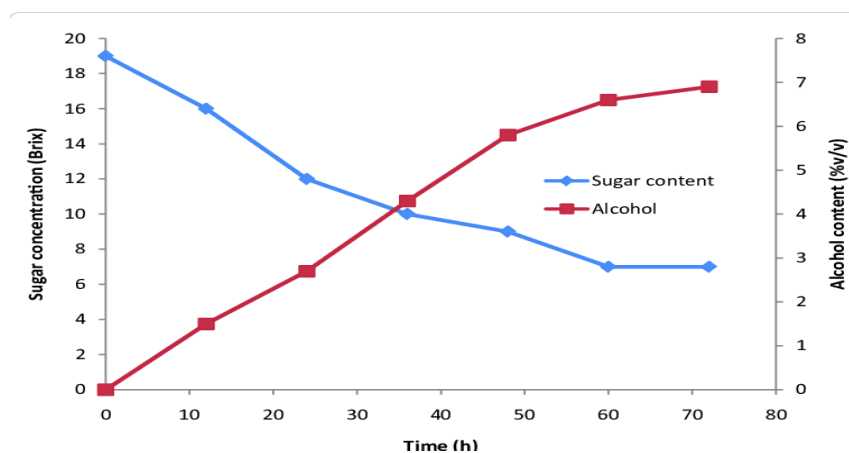
Figure 1:   Changes in sugar concentration (brix) and alcohol concentration verses time

1.4   Objectives of the Study

1.4.1   General Objective

To develop a WSN architecture for monitoring of Brix (sugar concentration) and alcohol concentration at Raha Beverage Company Arusha Tanzania.

1.4.2   Specific Objectives

The study aimed to achieve the following specific objectives:

(i)   To review the requirements for developing WSN for monitoring of Brix (sugar concentration) and alcohol concentration.

(ii)   To develop a monitoring system for brix and alcohol.

(iii)   To deploy the system.

1.5   Research Questions

The study intended to answer the following questions:

(i)   What attributes and methods used in the existing Brix and Alcohol content monitoring are important to contribute to quality wine/alcohol.

(ii)   How could the Brix and alcohol monitoring system using WSN be developed to measure wine parameters and ensure real time monitoring.

(iii)   How could the functional requirements and efficiency be determined and developed when the system have been deployed.

1.6   Significance of the Study

This study provides a reliable way of monitoring Fermentation parameters in Wine Industries through application of IoT & WSN architecture, as IoT technology is expected to grow to 36 billion connected devices (Vasconcelos *et al.,* 2017). Wine parameters are considered vital to measure and monitor.

Software based methods are becoming of special interest as they require conventional instrumentation to support them (Parihar, 2019). Hence literature insists on adaption for automated methods for monitoring wine parameters.

Monitoring and Management is very crucial as studies have revealed that fermentation parameters play important role on the final product of wine and beer hence monitoring entire activities is vital. To produce consistently high product processes control of wine fermentation is an important factor (Claus & Mojsov, 2018).

According to Consumer International report (2016), globally consumers are already users of connected devices. Through IoT and WSN it adds on by providing infrastructure of overcoming the need of human involvement that the interaction is solely based on the internet and the connected objects.

1.7   Delineation of the Study

Fermentation process by-product include Brix, PH, Carbon dioxide Alcohol percentage by volume. This project is intended to monitor the most important fermentation parameters which

contribute to quality alcohol which are Brix and Alcohol percentage by volume rather than the entire by product of fermentation. This project mainly monitors two fermentation parameters. By monitoring it focus on utilizing the use of low cost and automated transmission where it focuses on use of wireless technology to transmit data to a centralized monitoring. The visualization tool is the Web API for monitoring and documentation.

CHAPTER TWO

LITERATURE REVIEW

2.1   The IoT Architecture for Wine Monitoring

The monitoring of key critical parameters is intricately tied to wine production. The Internet of Things (IoT) has witnessed widespread adoption across various industries, including automotive, agriculture, and healthcare. It is anticipated that the number of IoT devices will surge to 36 billion by the year 2020 (Vasconcelos *et al.,* 2017)

The IoT architecture has been applied in wine sector and research indicates the potential for implementing a sensor network to measure the essential by product of fermentation during wine production (Jiménez-Márquez *et al.,* 2013).

The paper presents an IoT architecture for monitoring wine focusing on critical parameters like wine pH, wine temperature and level. However, it acknowledges limitations as wine fermentation crucially requires monitoring alcohol percentage by volume, brix concentration and carbon dioxide emission. To address these this paper uses standalone buoy which has source of power and exhibit low consumption which makes it suitable for fermentation process, similar approaches has been successfully in many domains such as oceanography (Hsieh *et al.,* 2017), the plastic material in which the material is made is water proof and have floating properties. This study introduced an affordable buoy with integrated electronic components to be placed at the bottom of tanks enabling enhanced monitoring and efficiency in wine production. The novel IoT architecture employs connectionless communication making it suitable for remote monitoring in challenging environments. This cost-effective system can be implemented across various wines companies to improve production effectiveness.

2.2   The IoT Architecture for Monitoring Wine Fermentation Process of Debina Variety Semi Sparkling Wine

The advent of the internet has ushered in the era of the Internet of Things (IoT), connecting various devices. To facilitate the seamless transfer of real-time sensor data, different objects are employed for data exchange. This literature review explores an integrated IoT-based system designed for monitoring and measuring wine parameters. The system incorporates interconnected sensors and actuators, adding multiple dimensions to real-time wine parameter

monitoring. While fermentation is a crucial process in wine production, monitoring holds equal significance for ensuring successful outcomes.

In this paper, the author presents an IoT based integrated system for monitoring poor quality wine which may be caused by stuck fermentation can be reduced through monitoring (Urtubia *et al.,* 2004). Recent progress in monitoring systems has resulted in diverse designs of wine sensors. In this study, the author adopts a low-cost microcontroller that utilizes Bluetooth for communication. Nonetheless Bluetooth comes with drawbacks, such as short-range communication and reduced security. The methodology employed in this research is the Smart Barrel architecture of the Wine Smart Barrel System, which involves wine stored in vessels. However, it is worth noting that this methodology can be expensive to implement and adapt.

Previous research has showcased the significance of a fermentation monitoring system due to its notable increase. It has become evident that daily monitoring of wine is crucial pre-harvest requirement, Moreover, this system has led to the development of a central station designed to analyze and monitor the gathered information effectively (Jiménez-Márquez *et al.,* 2013).

2.3　Sensor Module for Monitoring Wine Fermentation Wine Fermentation Process

Sensor network have become popular and find applications in many areas include agriculture, healthcare (Angelkov & Bande, 2018). Several studies have proposed system to develop on wine monitoring activities. This is one of the literatures that report on deploying ofdeveloped sensor architecture system aimed at managing and controlling wine production activities. In this research paper the author has developed a sensor network for monitoring temperature, wine acidity, pH, alcohol and carbon dioxide gas.

The research study introduces the use of PIC16F877A microcontroller as its main component of the system. The system uses connection-oriented architecture to send sensor data to the remote server. Through this method it is difficult to send data as the fermented must contains liquids. This research paper introduced a monitoring fermentation process and send sensor data through wired communication this offers disadvantage.

CHAPTER THREE

MATERIALS AND METHODS

3.1   Study Area

The Raha Beverage Company, situated in the northern region of Tanzania, Arusha, undertook this project. The company primarily focuses on the production of alcoholic beverages using locally sourced bananas abundantly grown in this part of region of Tanzania. Among its prominent products are Raha Gold and Raha Poa, both of which have gained significantly popularity among the local population and market. These alcoholic beverages have earned widespread recognition due to their affordability and high-quality standards, enabling them to be distributed and sold across the entire country of Tanzania.

3.2   Sampling Design

For this project a specific sample size was deliberately selected to address the challenges related to obtaining Brix and Alcohol control data at Raha Beverage Company. The sampling technique was non-probability purposive sampling these was chosen because it allowed the collection of crucial information from the most relevant personnel in the field of area of interest. The selection of participants was based primarily on their expertise, knowledge, and understanding of the subject matter ensuring valuable insights to contribute to a comprehensive understanding of the problem at hand (Etikan, 2016). Furthermore, Journal articles, Books and Websites covering Brix and Alcohol were used to improve the study.

Table 1:   The number of people who took part in the Raha Beverage Company survey

Department	Respondent
Management	7
Lab Technicians	4
ICT	3
Fermentation department	4
Quality Assurance	5

3.3 Data Collection

To find relevant information for designing the monitoring of brix and alcohol content using wireless sensor network. The approach of collecting data was used. The initial step dedicated obtain the available details related on brix and alcohol monitoring and the challenges encountered on the process, the following procedure was to utilize the informationobtained in the procedure done before to design test and deploy the system.

The data was obtained from the study area between November 30th and December 31st 2020,in addition to the existing information on brix and alcohol monitoring issues faced using conventional monitoring. The following are the various data collection methods that were used:

(i)   Personal Interview

In this method, structured organized interview was used by asking questions to officials through one-on-one session, Lab analyst, Lab Engineer, technical staff in Fermentation department. The Interview was meant for officials to provide information onhow monitoring activities are done.

(ii)   Content Analysis

Data was obtained using this approach by analyzing literature related to brix and alcohol content monitoring from various sources. Brix and alcohol screening werestudied using books, research papers, journals.

3.4   Reliability and Validity of Data

Various scales are used to indicate the level of sugar content of solutions (Ball, 2006). Due to the importance of sugar concentration in wines and beer factory there are various scales that are commonly available to express sugar content.

(i)   Degree Brix

A scale commonly used to measure the percent of sugar content. Every degree brix is approximately 0.0128 SG units.

(ii)   Degree Baume

Is a scale of relative density that is developed that is developed from Specific gravity

Sugar to Potential Alcohol Chart

Brix	Baumé	°Oechsle	Specific Gravity	Potential Alcohol
14	7.8		1.0568	7.6
15	8.3		1.0611	8.2
16	8.9	65	1.0654	8.8
17	9.4	69.8	1.0698	9.5
18	10.0	74.1	1.0741	10.1
19	10.6	78.5	1.0785	10.8
20	11.1	83	1.0830	11.5
21	11.7	87.4	1.0875	12.2
22	12.2	91.9	1.0920	12.9
23	12.8	96.5	1.0965	13.6
24	13.3	101	1.1011	14.4
25	13.9	105.6	1.1057	15.1
26	14.4	110.3	1.1103	15.9
27	15	114.9	1.1150	16.7
28	15.6	119.6	1.1197	17.5

Figure 2:   Concentration scale for sugar solutions

The more fermentable sugar the more amount of alcohol present in a liquid 1.000 SG is equivalent to O degree Brix. These scales are used by winery Brix scales start at 0 SG startsat 1.

Using gravity scale the SG of the solution in fermentation will be measured the more alcohol the lower the readings as shown in Fig. 2.

3.5   System Implementation

This section describes the methods used to build a Brix and Alcohol tracking system using a Wireless Sensor Network System. It covers the methods for selecting sensors, as well as PHP web application creation. The following subsections show preprocessed data that was sent to the cloud via Web API.

Wireless Sensor Network (WSN) can be defined as a network of sensor node that comprehend their environment, These nodes play a crucial role in the network by transmitting data to the cloud through an HTTP API connected to a web client. Additionally each sensor node is equipped with a power module consisting of a battery source and a microcontroller with a Wi-Fi module.

3.6   System Design

3.6.1   Block Diagram Architecture

The brix and alcohol monitoring system based on wireless sensor network (WSN) draws its architecture inspirations from the Internet of Things (IoT) framework. It encompasses data collection, analysis and storage. As its core, there is a standalone microcontroller ESP32 integrated with a gyroscope capable of interpreting motion and measuring liquid density to determine brix and alcohol content in the solution.

The system is complemented by application software, enabling real-time visualization of alcohol and brix parameters, facilitating the generation of detailed reports. Figure 3 depicts the block diagram illustrating the entire setup for brix and alcohol monitoring using WSN.

The block diagram for brix and alcohol content comprises two key components, the data transmission segment and the embedded board featuring the ESP32 microcontroller and the gyroscope along with the accelerometer serving as an actuator. Together this architecture enables accurate monitoring of Brix and Alcohol levels in the solution with the help of WSN technology.
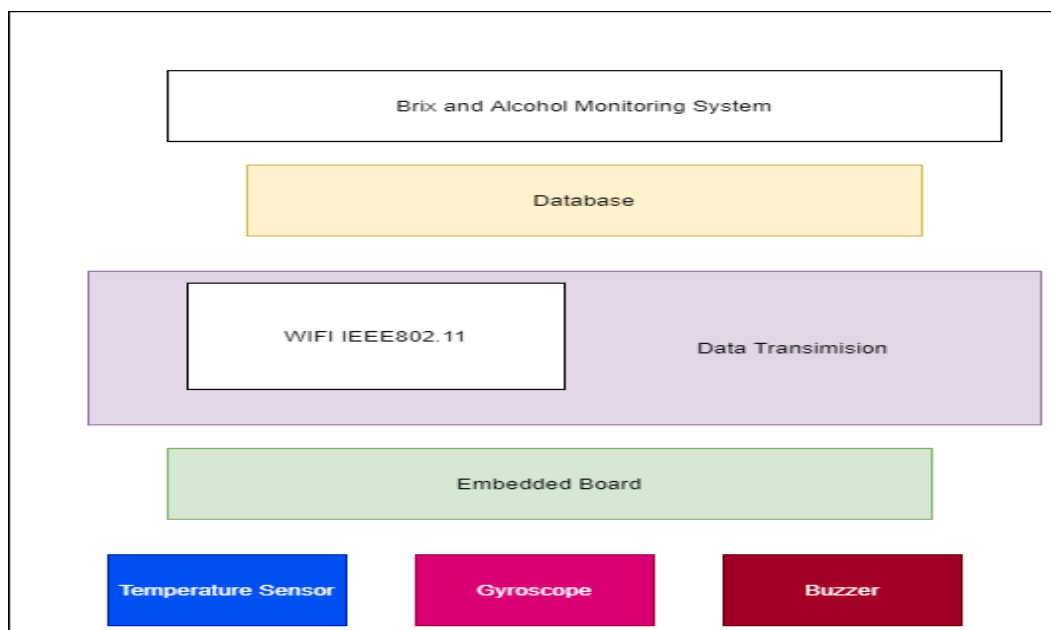
Figure 3:   Block diagram

The brew monitoring Web API allows only Brew Admin to register system users while only the registered user can login to the system. This process ensures authorized access to web API.

3.6.2 Fritzing

Is an open source tool for building electronic prototypes, fritzing allows to display schematic which helps to focus with complicated circuit (Knörig *et al.,* 2009). For visuallypresenting the circuit fritzing was used, the software enables to produce prototype which can be transition transferred to a PCB layout. The circuit view was reflected on a PCB.
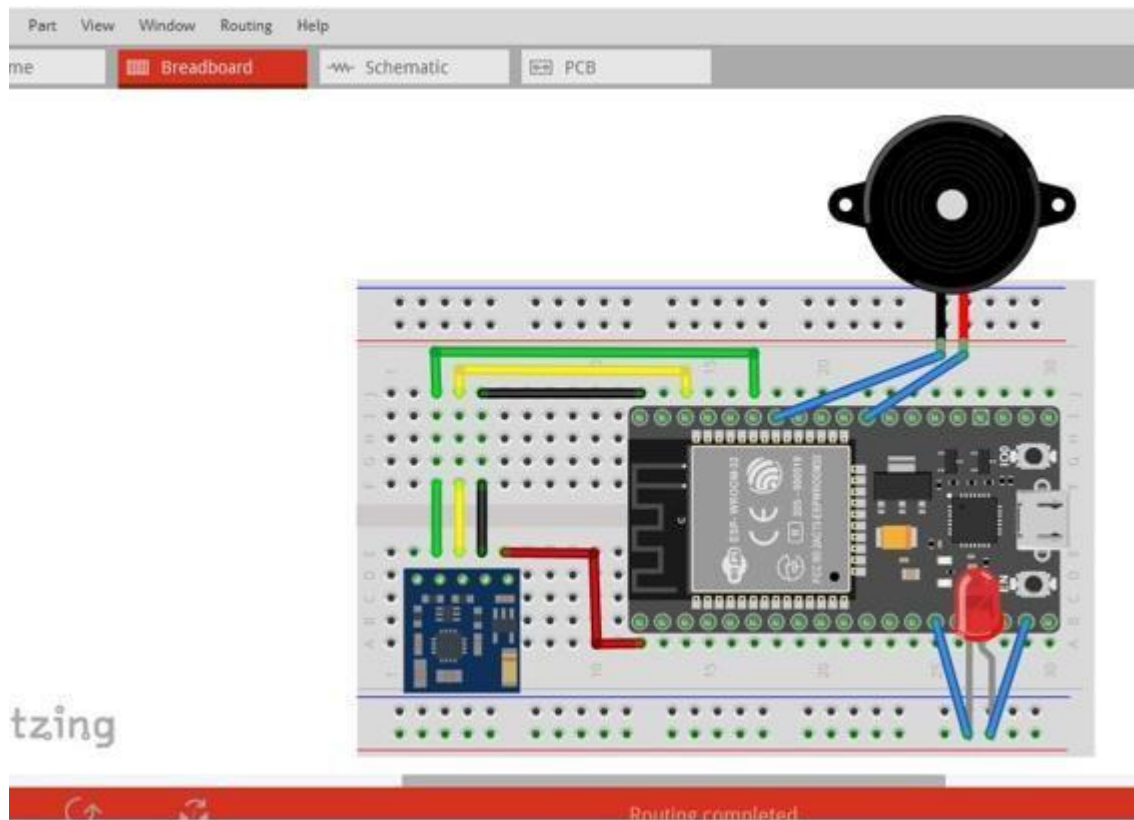
Figure 4:   Fritzing Schematic circuit

3.6.3 Designing the Printed Circuit Board

To achieve a compact and professional appearance for the electric circuit a specially designed PCB (Printed Circuit Board) specially designed was utilized. The PCB incorporates a combination of electronic chips, capacitors and semiconductors strategically mounted to establish seamless electrical connectivity among the components (LaDou, 2006). The PCB manufacturing process involves several key steps namely drilling, image transfer and electroplating. First holes are drilled into the PCB to create interconnections for various components. These holes are subsequently plated to enhance their conductivity. Once the PCB is prepared all the components are soldered and connected to the board to complete the assembly the process.
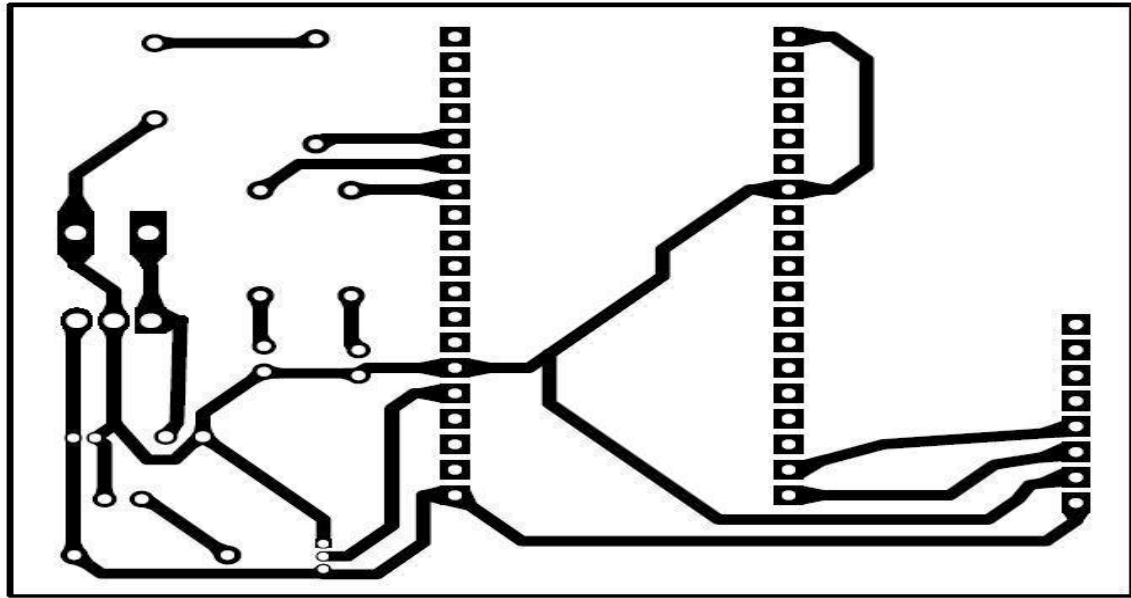
Figure 5:   Printed circuit board diagram

The number of conducting layers in PCB offers minimum trace spacing while maintaining electrical connection (Shamkhalichenar *et al.,* 2020). To fit the placement of electrical components PCB becomes an evolving solution.

3.7   System Requirements

3.7.1   Software Requirements

(i)   The PHP Storm

During development of the Brew Monitoring tool PHP Storm IDE was used for writing code. The PHP Storm is a Python application development tool. It has a lot of advantages over other editors. It supports multiple OS, facilitate code completion and documentation (Grigorev, 2014).

Brew admin and monitoring personnel has ability to monitor the ongoing fermentation process through web application named "Brew Monitoring Tool" developed by PHP, PHP isa scripting tool for web application that allows dynamic interaction of web application, its such intuitive (Onlerawaju, 2015).

3.7.2 Hardware Requirements

(i)   NodeMCU

The ESP 32 is an advanced microcontroller that comes with integrated Wi-Fi eliminating the need for any additional chipsets, making it a perfect choice for IoT applications. This feature rich MCU not only boasts integrated Wi-Fi but also offers Bluetooth connectivity making it high versatile for various applications. The ESP32 utilizes built-in communication standards within the chip of the development board eliminating the requirement for external modules for communication purposes (Foltýnek *et al.,* 2019)

Figure 6:   The ESP32 WROM

(ii)   Accelerometer Sensor

The MPU6050 gyroscope computes the rotational change of the object along XYZ plane. This gyroscopic data represents the object's angular position changes over time defining its orientation and overall movement.
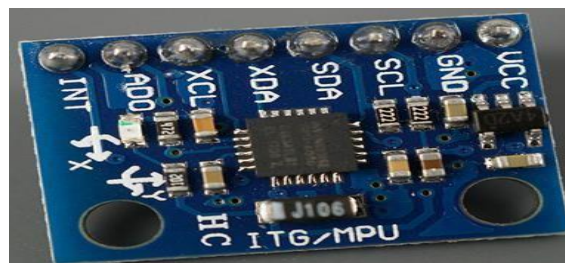
Figure 7:   Accelerometer sensor

The accelerometer is responsible for measuring an object's acceleration, detecting static forces like gravity. For a stationary object, the Z axis acceleration corresponds to the gravitational force while the other axes should register zero acceleration. By utilizing the accelerometer data one can compute movement angles using trigonometry.

Using these results, we then can calibrate the obtained results from the sensor to get more accurate information about the sensor orientation.

(iii)   Router

A device which allows communication between local home network. A router connects different communication devices include modem, fiber cables and many other devices. Through a router network cable connect via interface card in wired network. Ip address assigned to a router is a private IP address this means network devices are only accessible within a local area network. The IP address assigned to a router provides gateway to deviceson the same network. Several network devices can connect to a router.

Figure 8:   Dlink Router

Figure 8 shows the setup and configuration of the sensor node as a webserver and a Dlink router to connect into the local area network

Figure 9:   Configuring DHCP static IP to router

(iv)   Sensor

The hardware devices serve as the source of information for the Web API. Specifically, the ESP32 functions as a webserver facilitating the connection between software and hardware to handle data retrieval and maintenance for web clients. In this project the web clients used was desktop PC. This communication model follows the client-server architecture paradigm where the ESP32 web server acts as the server and the desktop PC acts as the client. Figure 9 illustrates the HTTP communication between the ESP32 web server and the client.
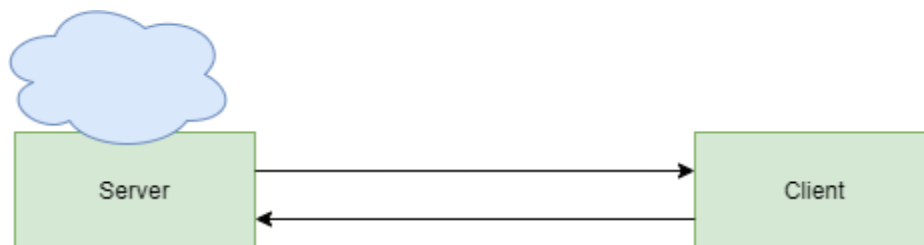
Figure 10:   The HTTP Protocol

The protocol used was HTTP this type of protocol involves communication between client and server.

(v)   WiFi Mode Operation of ESP32

The ESP32 can basically operate in various type of WIFI mode. It can either be from the following:

(i)   STA Station Mode

(ii)   Soft Access Mode AP

(iii)   Station + Soft AP mode

In station mode as used in this project the ESP32 will connect to an existing WIFI network. To create ESP32 as a web server the mode used was station mode SSID of the router and respectively password was connected to an ESP32. For ESP32 to connect to a router and send data it a web API it must be on the same network with the web client its Mac address and IP address must be configured to a router.

Figure 11: **Configuration of IP address and MAC address to a router**

(vi) Sensor selection

On choosing sensors different knowledge about the type of application is essential also the required operating condition, requirement is essential to create informed and accurate sensor selection. The choice of the sensors does strongly depend on the targeted application.

Performance of accelerometers can be summarized in terms of Range R and Sensing frequency (JShieh, 2001). For proper selection of sensors main sensor characteristics have been shown in Table 2.

Table 2: **Main characteristics of Sensors**

Aspects	Description
Range	The electrical output signal range is determined by calculating the disparity between its maximum and minimum values
Resolution	Ability to detect changes in the measurement of quantity
Sensing Frequency	As accelerometers are used for vibration sensing frequency is important
Accuracy	The closeness of the measured value to the exact value
Size	Include dimension of sensors

3.7.3 Data Publishing

The HTTP API helps interacting with your data in an efficient and real time way. A protocol usedto communicate with internet resources for each action requested. The HTTP is a standard requestresponse protocol. These is an IoT devices which uses internet to send and receive information based on needs and requirements.

(i) Data transmission

In the system, transfer of sensor measurements from the sensor node placed on the solution to a remote database station was facilitated using ESP32 WIFI features. It operates on 2.4GHZ frequency band this is a perfect application for IoT since its versatile and robust as well it's a low power chip.

The Esp32 implements on TCP/IP protocol this means it can operate on two modes as access point and Wi-Fi station which enables it to speak to WIFI routers and servers.

(ii) Coding of the ESP32

Programming of the controller is done the sketch is uploaded through the Arduino Ide platform which is an open electronic platform for the development support on microcontroller. Several libraries are used in the Arduino Ide includes Adafruit Esp 32 sensor library and Adafruit MPU 6050 library, communication Port libraries, TCP libraries to enable network connectivity.

Figure 12:   The esp32 sketch

3.7.4   Floating Buoy

From Logic and science for a sensor node to float it requires to be less dense than the solution, For measurements to be sent to web API the sensor node being insulated is placed inside the tank. Sensor node was used for monitoring and producing the status of the solution, the installation is simple to perform and is less costly and its design should is well suitable insulated and proofing and allow easy replaceable power supply system (Albadalejo, 2012).

The sensor buoy is supported with energy supply mechanism, to ensure stability and floating the mechanical structure was designed to be light. The sensor buoy has the following functions:

(i)   Connection

(ii)   Regular synchronization with the central station through data request/response.

(iii)   Data acquisition and data transmission

The floating material is IP68 ratings and casing is watertight made of plastic material.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Results and Discussion from Study Area and Related Document Review

Understanding the current brix and alcohol control system and identifying the necessary components for proper technology used to meet the organization's needs the use of questionnaires with Raha Beverage fermentation department officials was conducted. The key relevant information helped in the parameters measured.

4.1.1 Monitored Parameters Monitored During Fermentation Process

In a whole fermentation process involves various key parameters which are taking place include brix level, alcohol, carbon dioxide which is the byproduct of fermentation, pH level and Temperature. The mentioned parameters have direct effect and contribute much to fermentation process. This study found that among all parameters the brix and alcohol have great impact to quality wine.
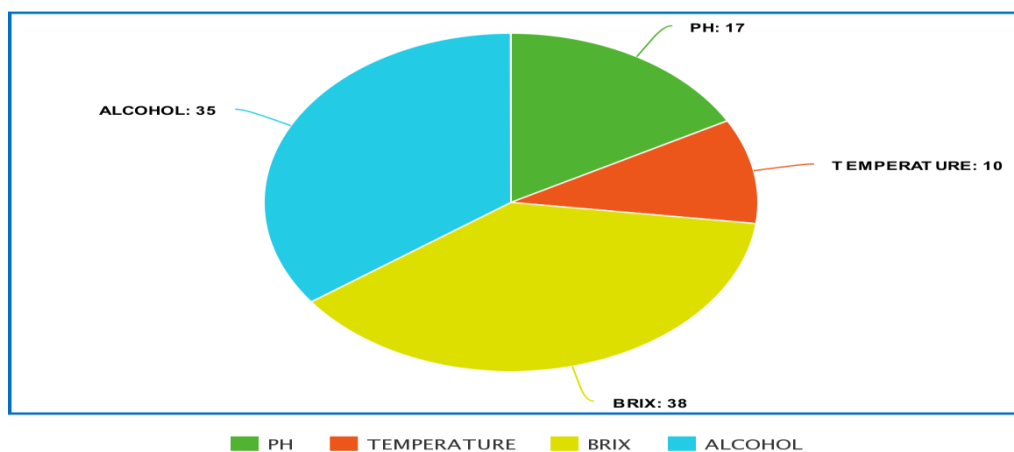
Figure 13:   Distribution of fermentation parameters as monitored at Raha Beverage company

4.2 Results of System Design

4.2.1 Web Based

The web monitoring tool was intended to provide administration purposes register required users allow user authentication, and display of information monitored. For these to be done user and server-side application were developed using Symfony frame work this is a Python

web framework which is easier to learn, it comes with many tools and classes and also easy to install (TimBowler & Bancer, 2009). Symphony, like other MVC frameworks, is made up of three layers: View, model, and controller, each of which receives a request and responds appropriately. The View is in rensponsible of displaying information to the user. The Model is in charge of the application's business logic, while the Controller is in charge of user interaction and connection.
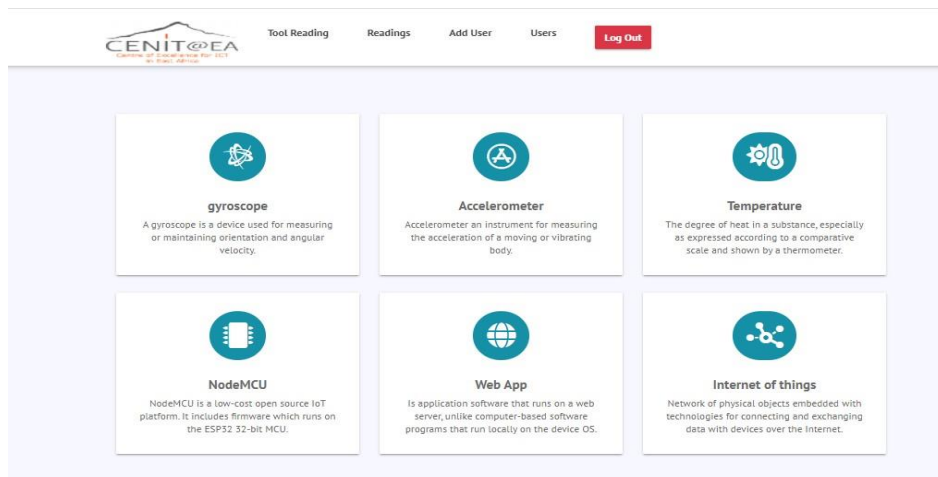
Figure 14:   Monitoring tool

(i)   User Authentication

Figure 13 shows the login interface which facilitate registered user to login into the system by requiring to fill their respective login credentials. By using this index page registered users will have access into the system and monitor the fermentation parameters.
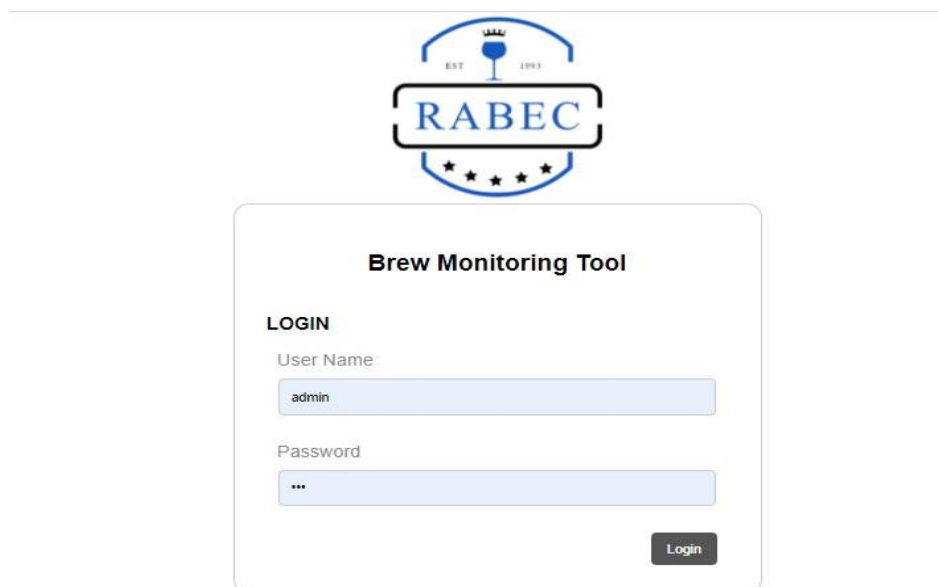
Figure 15:   User authentication

The primary objective was to see if users can access this web application and able to login their credential.

Table 3:   User authentication Test

Check	Input	Output	Outcomes
Login module	Username and password	Tool monitoring page	SUCCESS
Logout module	Click Logout button	Index Page	SUCCESS

(ii)   Manage accounts (Register User and Login)

Only Brew Admin can register system users with the Brew Monitoring Web API, and only the registered user can login to the system. This procedure ensures that only authorized personal gain access to the web API.
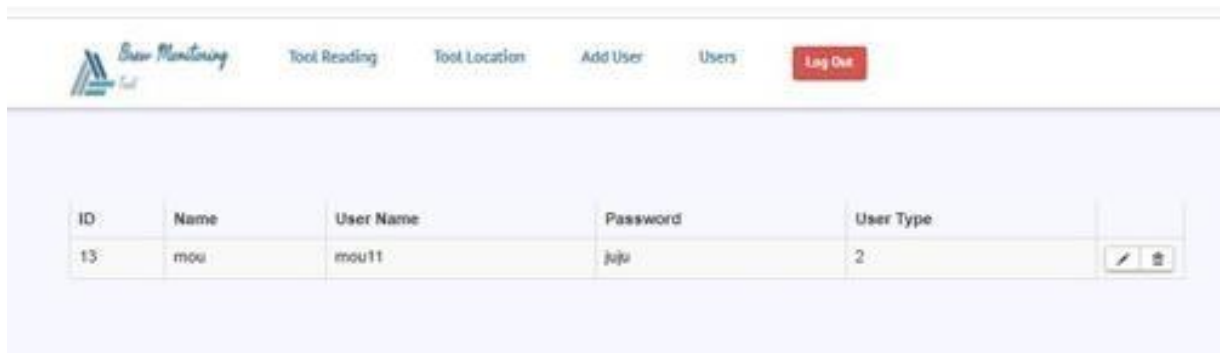
Figure 16:   Registering user through WebAPI

(iii)   Web Application System Unit Testing

A variety of unit tests were conducted in the various modules, data acquisition, and report generation units to ensure that the Brew Monitoring Web Application was fully functional. The results for the appropriate test were obtained as stated in the following subsection.

(iv)   System Administration Testing

The aim was to see if Brew admin could monitor user accounts. The system performed as planned, as shown in Table 4.

Table 4: System administration testing

Test	Input	Output	Results
Index page	Enter credentials	Field administration page	SUCCESS
Admin page	Register users	Client form page	SUCCESS
Admin page	Delete users	User group page	SUCCESS

4.2.2 System Modeling

In the development of the Monitoring of Brix and Alcohol using Wireless Sensor network. Modeling was employed to show the process functionality. The process involved in the architecture development of the system was examined its operational, its relationship with other processes and if it produces the output required. In the study of Brix and Alcohol monitoring the activities used modelling the system which include:

(i) Schematic Diagram

The schematic diagram is a representation of an embedded circuit in pictorial form. It depicts the electrical connections between various electronic components. In the context of designing a sensing architecture, the schematic diagram was used. as the sensing unit which involved sensors, printed circuit board mounted with components. Figure 3 was designed using easyEDA web based electronic design tool. The choice of using easyEDA circuit design tool the choice of is due to the fact it is web based which allow to edit schematic diagram and lay out PCB without download.
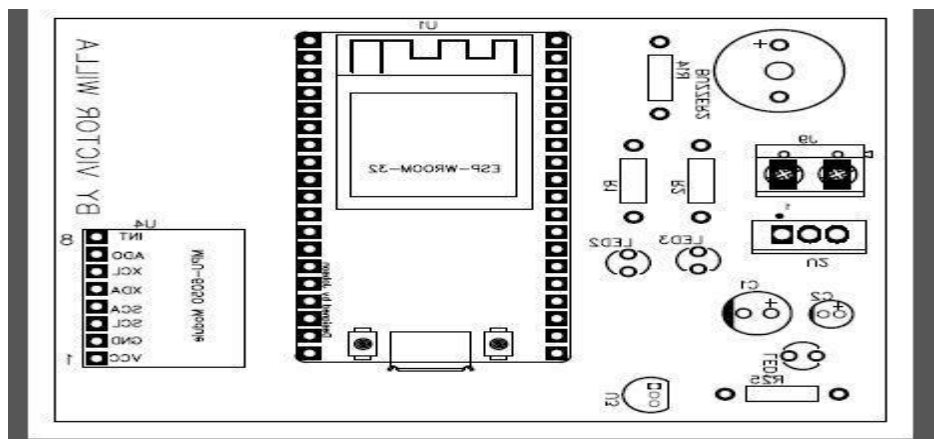
Figure 17: The PCB layout

4.3   Use Case Modeling of the System

4.3.1   Use Case Diagram

A use case diagram serves as a graphical depiction of a system's operations from the user viewpoint. It visually represents the interactions between actors and the system to fulfill functional requirements. Consequently, for this study the decision was made to utilize the use case diagram as a valuable design tool. Its primary purpose lies in effectively and accurately capturing the system's functional specifications. The use case approach is widely recognized and employed in modeling engineering functional requirements due to its popularity and effectiveness (Firesmith, 2003). Use case is a popular modeling approach.

In order to function properly the use case component are described in Table 5.

Table 5:   **Use case components**

Component	Descriptions
System boundary	Is referred as rectangle whereby it is used to distinguish environmental components of the system
Relationship	It shows connection between user and system
Extend	It involves one use case indirectly incorporates the behaviorof other
Actors	It includes people, automated system which has ability to interact with a system
Use Case	The sequence of actions taken by users to get the required output

The use case diagram in Fig. 18 depicts interactions in the Brew Monitoring WebAPI framework, which has three actors called Brew Official, Brew Official Admin, and Sensor node.
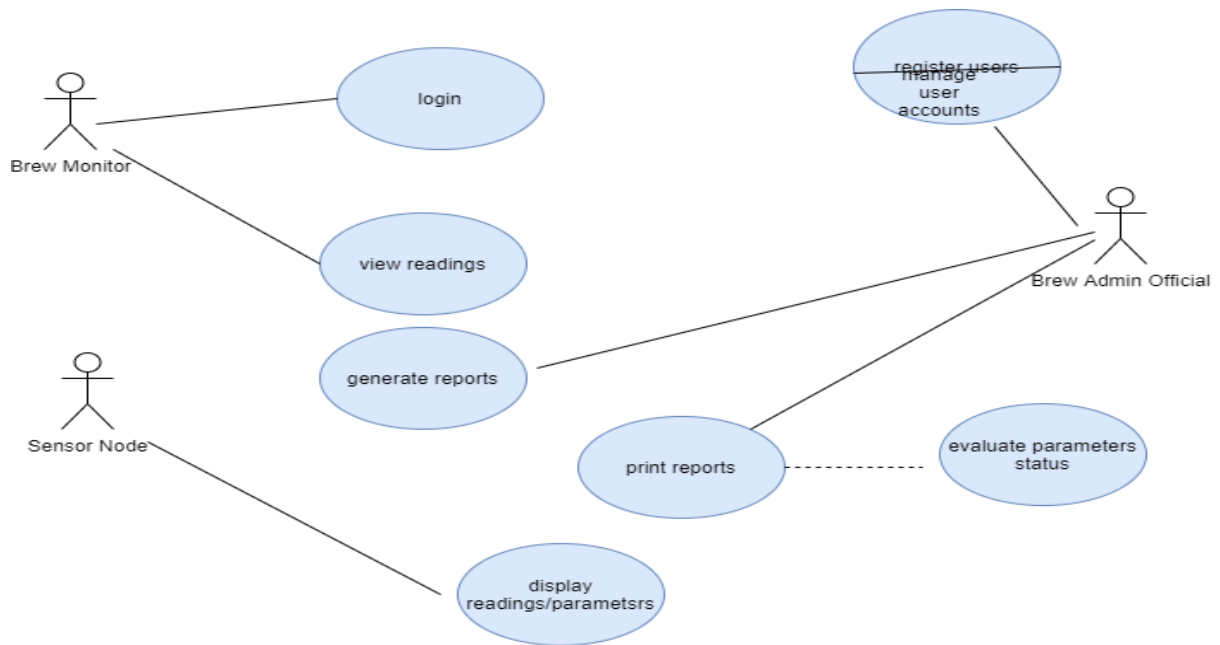
Figure 18:   Use Case diagram

4.3.2   Data Flow Diagram

Data flow diagram (DFD) is the way to visualize flow of information in a system. It clear analyzes system requirements. DFD is significant in the design and implementation of information system (Bruza & Weide, 1989). DFD are widely to model flow of information in a system (Larsen *et al.,* 1994). Figure 19 presents the Data Flow Diagram (DFD) that was purposely used in the development of Brew Monitoring Web API system was developed to facilitate transfer of information between sensor node and the system. It was also basic building block for developing the Brew monitoring tool at Raha Beverage Company.
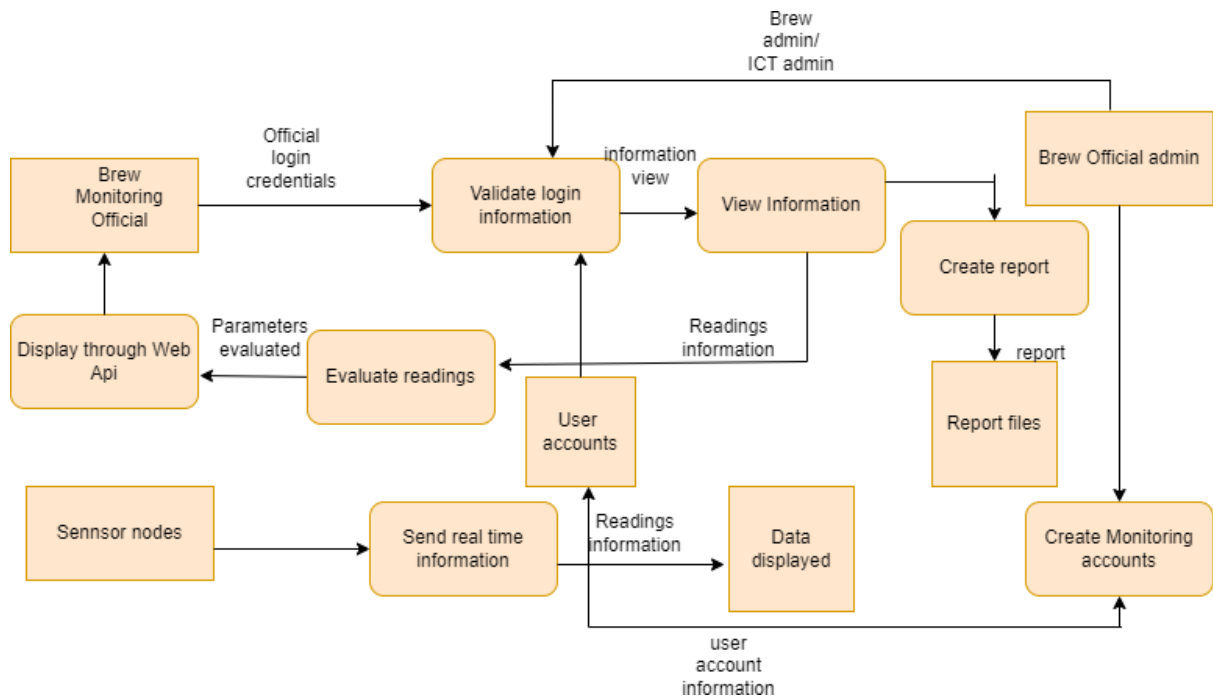
Figure 19:   Data flow diagram

4.4   Alcohol potential and Estimating % Alcohol by Volume

Brix and alcohol content play an important part in the quality of the wine-making process. Ensuring these parameters are carefully maintained throughout the fermentation process is absolutely essential.

When fermentation process turns whole sugar into alcohol the SG scale measures the concentration of sugar or brix level that has taken place, alcohol content is estimated that the greater value sweeter wine lower SG dryer the wine.

Figure 20:   Readings from the monitoring tool

The Specific Gravity SG serves as a measure to assess the potential alcohol by volume and Brix in a given context (Harry *et al.,* 2019). It is a dimensionless value that signifies the relative gravity at various stage of fermentation indicating the amount of fermentable sugar or potential alcohol percentage. Each stage of the process produces distinct outcomes. The Figure 19 shows the Brix% is 5% with a corresponding SG of 1.065.

Figure 21:   Graphs displayed from readings

4.4.1 Challenges of Existing Brix and Alcohol Monitoring System at Raha Beverage Company

From the information obtained from the Raha Beverage Fermentation staff and information encountered during traditionally monitoring activity exposed that, existing methods and process of monitoring are time consuming and less effective.

4.5 Design of a Monitoring System at Raha Beverage Company

Before the design of the system data was obtained from mixed research methodology include interviews, questionnaires and literature reviews which influenced greatly to depict clear constraints that should be incorporated in the developed system. To achieve this the system was analyzed as follows.

4.5.1 Functional and Non-Functional Requirements

Functional and Non-functional specifications are the two types of requirements (Eckhardt, 2016). The functions that the machine must execute without limitations are known as functional requirements (Alsaeh & Haron, 2016). It offers an overview of the system's actions. Non-functional criteria are properties that the final product must have and characterize non-behavioral characteristics.

Table 6: **Functional requirement of the system**

Functional Requirements	Description
The system shall automatically monitor brix and alcohol	Real time and updated data provided to the system
Brew admin is supposed to login into the system	Only registered officials shall login into the system
The system shall render real time data	Display of monitored parameters
Registering Staff, monitoring personnel, delete user,	Workers, officials, and users are all registered in the system

Table 7: Non-functional requirement of the system

Non-Functional Requirements	Description
Security	The system can only be accessed and used by registered users.
Accessibility	Any time the system is accessed, it must bepresent

4.5.2 System Modelling

The design of a Brix and Alcohol monitoring system using WSN relied upon principles of designing a low-cost system, efficient and relatively easy to use. With using a low cost and cheap sensing unit this has been achieved. Scalability was achieved by involving the potential of scaling system. The system's also provided ability to data integrity and confidentiality.

(i) Reading and calibrating data from sensors

A sketch written in C language is used to extract sensor data from the ESP32 WROM 32 I2C pins. ESp32 includes the embedded C programming language. The ESp32 pyserial library was imported into the software to allow reading signals from MPU 6050 channels in order to retrieve data I2C communication pins. I2c was created to facilitate communication between circuit boards. To Calibrate data requires finding the relationship between the tilt angle and the monitoring parameters provided from the formulas, for this reason The brix reading was calculated using the mathematical relationship shown in equation (1) as it is express the use of tilt angles in gravity data to identify the monitored readings.

$$gravity = a*tilt*tilt+b*tilt*c \qquad\qquad Eq.\ 1$$

Where gravity measured from the tilt angles of the accelerometer are computed.

4.5.3 Design Architecture

The device design architecture consists of a sensor node and a web application system for monitoring and analyzing of data. The definitions of the system and the block diagram is shown in Fig. 21.

(i) System's Conceptual Framework

For to address the challenges at Raha Beverage Company regarding conducting and monitoring

fermentation process which includes lack of automated monitoring tools, absence of quality and reliable data of the information collected from the barrel tanks, and involvementof experts in reading data. In relational to the mentioned challenges the design of Brix and Alcohol Monitoring using WSN conceptualized the following stages:

(a)   Sensor devices will be mounted on fermentation tanks to monitor fermentation parameters such as brix and alcohol. Via the controller, data will be stored, analyzed, and processed.

(b)   In the first step, data will be collected and transmitted using a local area network.

(c)   Data collected from the sensor node be monitored at real time

(d)   Brew official's ability to monitor, view, and record status of the monitoredparameters.

Figure 22 shows the system performed as expected to retrieve sensor values from the sensor node.
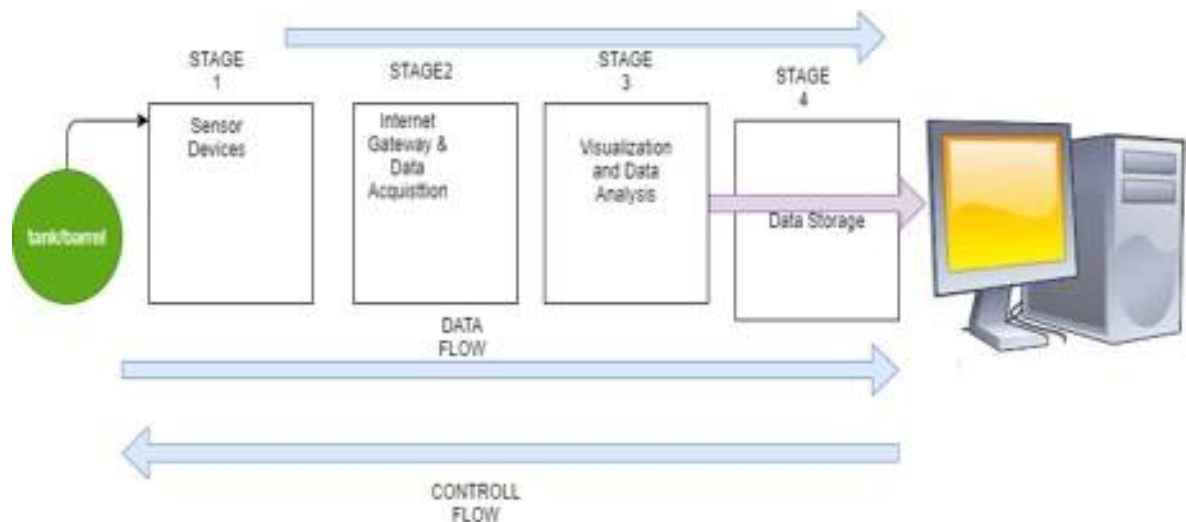
Figure 22:   Flow of information from sensor node to monitoring

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1   Conclusion

The alarm about monitoring fermentation parameters at Raha Beverage Company Limited is high due to increase of demand and need of producing quality Wine. Due to following regulation and standards has led to increasing importance of managing and monitoring Wine fermentation process. In factories and production companies where there is substantial need of monitoring these parameters. However, the main challenges facing the Fermentation authorities at Raha Beverage Company include absence of modern and automated equipment to measure and monitor fermentation by products parameters timely, accurately and reliably.

From the findings and later results obtained from this study conducted at Raha Beverage Company Staff indicated that the present monitoring system are ineffective as it requires much time to get results, more over implementing the monitoring process incurs considerable expenses in terms of both human and financial resources. Additionally, the findings indicated that the Brix and Alcohol readings were susceptible to subjectivity when manually taken by humans. In light of these challenges and considering the advancement in existing literature and sensor technology, this study proposes the development of a monitoring tool that provides real time information of evaluating readings. The innovative approach aims to address limitations of traditional methods and enhance accuracy and efficiency of the monitoring process.

The research centered on using wireless sensor technology to increase data flow from the sensor node to the central station. Different from the existing monitoring system the new developed system ensures real time monitoring of the parameters through visualization by Web API it also stores and manages the readings for records.

Based on the findings, it was clear that advancements in wireless communications and sensor technology could help Raha Beverage Company improve the quality of its fermentation monitoring systems.

5.2   Recommendations

The expressed views from the Raha Beverage Company Ltd officials resulted the outcome of the improved Monitoring of Brix and Alcohol system that was developed to face the existing challenges and recommend the following:

(i)   The use of Brix and Alcohol monitoring using Wireless Sensor Network could be adoptedby Raha fermentation official at fermentation department to facilitate reliable collection and monitoring of key fermentation parameters.

(ii)   Current fermentation monitoring methods have been shown to be ineffective and unreliable. As a result, the Raha Management and Team should set aside funds and budget todeploy ICT-related systems to manage the company's efficiency, as it has proven to be helpful in decision-making and reducing challenges.

(iii)   By taking advantages of WSN monitoring parameters can be gathered from the sensor node located several distances apart and be easily monitored.

This study sought to design an enhanced Brix and Alcohol monitoring system utilizing existing literature and sensor technology. The ultimate goal was to overcome the current challenges in monitoring of fermentation process.

The research focused on utilizing wireless technology to allow transmission of information from the sensor node to the Web API. The developed system consists of a sensor node and a web-based monitoring tool. The contribution has been achieved through:

(i)   Design architecture solution for Brix and Alcohol monitoring at Raha Beverage Company.

(ii)   The use of cost-effective electronics device to build a technological solution which is cheap and relatively easy to use.

(iii)   The developed monitoring system ensure reliability, high performance and scalability to system users and personnel responsible for keeping and maintaining these records.

REFERENCES

Akubor, P. I., & Ukwuru, M. U. (2003). Functional properties and biscuit making potential of soybean and cassava flour blends. *Plant Foods for Human Nutrition*, *58*(3), 1-12. https://doi.org/10.1023/b:qual.0000040344.93438.df.

Angelkov, D., & Bande, C. M. (2018). Sensor module for monitoring wine fermentation process. *Lecture Notes in Electrical Engineering*, *428*, 253–262.

Ball, D. W. (2006). Concentration scales for sugar solutions. *Journal of Chemical Education*, *83*(10), 1489. https://doi.org/10.1021/ed083p1489.

Botha, E., Malekian, R., & Ijiga, O. E. (2019). IoT in Agriculture: Enhanced throughput in South African farming applications. *2019 IEEE 2nd Wireless Africa Conference,* 1-5. https://doi.org/10.1109/AFRICA.2019.8843423.

Bruza, P. D., & Van der Weide, T. (1989). *The semantics of data flow diagrams*. Nijmegen TR: University of Nijmegen, Department of Informatics, Faculty of Mathematics and Informatics.

Buczinski, S., & Vandeweerd, J. M. (2016). Diagnostic accuracy of refractometry for assessing bovine colostrum quality: A systematic review and meta-analysis. *Journal of Dairy Science*, *99*(9), 7381-7394. https://doi.org/10.3168/jds.2016-10955.

Buratti, C., Conti, A., Dardari, D., & Verdone, R. (2009). An overview on wireless sensor networks technology and evolution. *Sensors*, *9*(9), 6869-6896.

Dongare, M. L., Buchade, P. B., & Shaligram, A. D. (2015). Refractive index based optical Brix measurement technique with equilateral angle prism for sugar and Allied Industries. *Optik*, *126*(20). https://doi.org/10.1016/j.ijleo.2015.05.137.

Etikan, I. (2016). Comparison of Convenience Sampling and Purposive Sampling. *American Journal of Theoretical and Applied Statistics*, *5*(1), 1-4.

Firesmith, D. G. (2003). Security use cases. *Journal of Object Technology, 2*(3), 53-64. https://doi.org/10.5381/jot.2003.2.3.c6.

Foltýnek, P., Babiuch, M., & Šuránek, P. (2019). Measurement and data processing from

Internet of Things modules by dual-core application using ESP32 board. *Measurement and Control*, *52*(7-8), 970-984. https://doi.org/10.1177/0020294019857748.

Grassi, S., Amigo, J. M., Lyndgaard, C. B., Foschino, R., & Casiraghi, E. (2014). Beer fermentation: Monitoring of process parameters by FT-NIR and multivariate data analysis. *Food Chemistry*, *155*. https://doi.org/10.1016/j.foodchem.2014.01.060.

Harry, M. F., Carly, Z. S. D., & Jong, M. N. (2019). Sorghum Coffee–Lactose Stout Production and Its Physico-Chemical Characterization. *Beverages*, *5*(1), 20.

Hsieh, C. Y., Ma, Y. F., Shie, S. C., Her, W. H., Chang, H. I., Yang, Y. J., & Wei, C. L. (2017). The NTU buoy for typhoon observation. Part 2: Field tests. *Oceans 2017 Aberdeen*. https://doi.org/10.1109/OCEANSE.2017.8084821.

Jiménez-Márquez, F., Vázquez, J., Úbeda, J., & Sánchez-Rojas, J. L. (2013). Low-cost and portable refractive optoelectronic device for measuring wine fermentation kinetics. *Sensors and Actuators, B: Chemical*, *178*. https://doi.org/10.1016/j.snb.2012.12.091.

Kanyathare, B., & Peiponen, K. E. (2018). Hand-held refractometer-based measurement and excess permittivity analysis method for detection of diesel oils adulterated by Kerosene in field conditions. *Sensors (Switzerland)*, *18*(5), 1551.

Kleinhenz, M. D., & Bumgarner, N. R. (2013). Using Brix as an indicator of vegetable quality. *Linking measured values to crop management. Fact Sheet. Agriculture and Natural Resources. The Ohio State University, Columbus, OH*.

Knörig, A., Wettach, R., & Cohen, J. (2009). Fritzing: A tool for advancing electronic prototyping for designers. *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction, TEI'09*. https://doi.org/10.1145/1517664.1517735.

Larsen, P. G., Plat, N., & Toetenel, H. (1994). A formal semantics of data flow diagrams. *Formal aspects of Computing*, *6*(6), 586-606.

Lu, Y., Chan, L. J., Li, X., & Liu, S. Q. (2018). Effects of sugar concentration on mango wine composition fermented by Saccharomyces cerevisiae MERIT. ferm. *International Journal of Food Science & Technology*, *53*(1), 199-208.

Maicas, S. (2020). The role of yeasts in fermentation processes. *Microorganisms*, *8*(8), 1142.

Parihar, Y. S. (2019). Internet of Things and Nodemcu: A review of use of Nodemcu ESP8266 in IoT products. *Journal of Emerging Technologies and Innovative Research*, *6*(6), 1085.

Shamkhalichenar, H., Bueche, C. J., & Choi, J. W. (2020). Printed circuit board (PCB) technology for electrochemical sensors and sensing platforms. *Biosensors*, *10*(11), 159.

Urtubia, A., Ricardo Pérez-Correa, J., Meurens, M., & Agosin, E. (2004). Monitoring large scale wine fermentations with infrared spectroscopy. *Talanta*, *64*(3), 778-784. https://doi.org/10.1016/j.talanta.2004.04.005.

Vasconcelos, F., Figueiredo, L., Almeida, A., & Ferreira, J. C. (2017). Smart sensor network: With Bluetooth low energy and CAN-BUS. *2017 IEEE International Conference on Service Operations and Logistics, and Informatics,* 217-223.

Appendix 1:   The PHP Code to SignUp in to the System

```
<!DOCTYPE html>

<html>

<head>

<title>Register</title>

<link rel="stylesheet" type="text/css" href="style1.css">

</head>

<body>

<form action="signup-check.php" method="post">

<h2>Register User</h2>

<?php if (isset($_GET['error'])) { ?>

                                    <p class="error"><?php echo $_GET['error']; ?></p>

<?php } ?>

<?php if (isset($_GET['success'])) { ?>

                                    <p class="success"><?php echo $_GET['success']; ?></p>

<?php } ?>

<label>Name</label>

<?php if (isset($_GET['name'])) { ?>

<input type="text" name="name" placeholder="Name"

value="<?php echo $_GET['name']; ?>"><br>

<?php }else{ ?>

<input type="text" name="name" placeholder="Name"><br>
```

```php
<?php }?>
```

```html
<label>User Name</label>
```

```php
<?php if (isset($_GET['uname'])) { ?>
```

```html
<input type="text" name="uname"

    placeholder="User Name"

    value="<?php echo $_GET['uname']; ?>"><br>
```

```php
<?php }else{ ?>
```

```html
<input type="text" name="uname"

    placeholder="User Name"><br>
```

```php
<?php }?>
```

```html
<label>Password</label>
 <input type="password" name="password" placeholder="Password"><br>


<label>Re Password</label>
<input type="password" name="re_password" placeholder="Re_Password"><br>


<button type="submit">Register</button>
<button type="button" onclick="window.location.href='home.php';">Cancel</button>

</form>
</body>
</html>
```

Appendix 2:   The PHP Code to Login into The System

```
<!DOCTYPE html>

<html>

<head>

<title>LOGIN</title>

<link rel="stylesheet" type="text/css" href="style1.css">

</head>

<body style="background-color:white;">

<center><img width="200" height="200" src="assets/images/logo.jpg" ></center>

    <form style="margin-bottom: 10%;"  class="col-12 col-6 col-4"  action="login.php"
method="post">

<h2>Brew Monitoring Tool</h2>

<h3>LOGIN</h3>

<?php if (isset($_GET['error'])) { ?>

                <p class="error"><?php echo $_GET['error']; ?></p>

<?php } ?>

<label>User Name</label>

            <input type="text" name="uname" placeholder="User Name"><br>

<label>Password</label>

<input type="password" name="password" placeholder="Password"><br>

<button type="submit">Login</button>

</form>

</body>

</html>
```

Appendix 3:   The PHP, MYSQL Code to Connect database

```php
<?php session_start();

include ('db_conn.php');

$connect = mysqli_connect("localhost", "root", "", "brew");

$query = "SELECT * FROM users WHERE user_type = 2 ORDER BY id DESC";

$result = mysqli_query($connect, $query);

?>
```

```html
<html>

<head>

<title>users</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<link rel="shortcut icon" href="assets/images/fav.jpg">

<link rel="stylesheet" href="assets/css/bootstrap.min.css">

<link rel="stylesheet" href="assets/css/fontawsom-all.min.css">

<link rel="stylesheet" href="assets/plugins/slider/css/owl.carousel.min.css">

<link rel="stylesheet" href="assets/plugins/slider/css/owl.theme.default.css">

<link rel="stylesheet" type="text/css" href="assets/css/style.css" />

<link rel="stylesheet" href="https://pro.fontawesome.com/releases/v5.8.0/css/all.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/jquery.min.js"></script>

<linkrel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" />

        <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
```

```html
<script src="jquery.tabledit.min.js"></script>

</head>

<body>

<header id="menu-jk">

<div id="nav-head" class="header-nav">

                              <div class="container">

                                <div class="row">

<div class="col-lg-2 col-md-3 no-padding col-sm-12 nav-img">

              <a    href="home.php">  <img   width="50"   height="70"
                  style="color:red;"src="assets/images/papa3.jpg" alt="NM-AIST
LOGO"></a>


              <a data-toggle="collapse" data-target="#menu" href="#menu" ><i class="fas d-
block d-md-none small-menu fa-bars"></i></a>

</div>

<div id="menu" class="col-lg-8 col-md-9 d-none d-md-block nav-item">

<ul>

<li><a href="tool.php">Tool Reading</a></li>

<li><a href="location.php">Tool Location</a>

<?php if($_SESSION['utype'] == 1): ?>

<li><a href="signup.php">Add User</a></li>

<li><a href="users.php">Users</a></li>

<?php endif; ?>

<li><a class="btn btn-danger" href="logout.php">Log Out</a></li>

</ul>

</div></div></div></div>
```

```html
</header>

<div class="container">

<br /><br /><br />

<div class="table-responsive">

<h3 align="center"></h3><br />

<table id="editable_table" class="table table-bordered table-striped">

<thead>

<tr>

<th>ID</th>

<th>Name</th>

<th>User Name</th>

<th>Password</th>

<th>User Type</th>

</tr>

</thead>

<tbody>

<?php

while($row = mysqli_fetch_array($result))

{

echo '

<tr>

<td>'.$row["id"].'</td>

<td>'.$row["name"].'</td>

<td>'.$row["user_name"].'</td>
```

```
<td>'.$row["password"].'</td>

<td>'.$row["user_type"].'</td>

</tr>

}

</tbody>

</table>

</div></div>

</body>

</html>

<script>

$(document).ready(function(){

$('#editable_table').Tabledit({url:'action.php',

columns:{ identifier:[0, "id"],editable:[]

},

restoreButton:false, onSuccess:function(data, textStatus, jqXHR)

{

if(data.action == 'delete')

{

$('#'+data.id).remove();

}

}

});

});

</script>
```

Appendix 4: The Esp Code to Send data and display on the WebApi

```cpp
#include <AsyncEventSource.h>

#include <AsyncJson.h> #include <SPIFFSEditor.h> #include <WebHandlerImpl.h>

#include <ESPAsyncWebServer.h> #include <WebAuthentication.h> #include

<AsyncWebSynchronization.h>#include <AsyncWebSocket.h>

#include <WebResponseImpl.h>#include <StringArray.h>

   #include <Uri.h> #include <SPIFFS.h> #include <Wire.h> #include <ESPmDNS.h>

#include <Uri.h> #include <AsyncTCP.h>#include "esp_adc_cal.h"#include <regex>

#include "SGdata.h"

String version = "1.0";

const char* ServerName = "Brew Mnitoring";


#define SensorReadings 144

#define NumOfSensors      2

#define NumOfEvents 4#define noRefresh false #define Refresh  true

#define ON             true

#define OFF            false#define RelayPIN           0

#define LEDPIN     5

#define RelayReverse     false

define simulating       OFF

typedef struct {float temp = 0;

  float specificGravity = 0;float  tiltAngle = 0;

  } sensordatatype;
```

```
String        Received_Data[10];

String        SensorReading[NumOfSensors][6]; int SensorReadingPointer[NumOfSensors];

sensordatatype sensordata[NumOfSensors][SensorReadings];
struct settings {String DoW;

String Start[NumOfEvents];String Stop[NumOfEvents];String temp[NumOfEvents];

};


//String        Received_Data[10];

//String        SensorReading[NumOfSensors][6];String        DataFile = "params.txt";


String        Time_str, DoW_str;settings      Timer[7];

//int           SensorReadingPointer[NumOfSensors];float      Hysteresis      = 0.2;

const String legendColour = "black";const String titleColour        = "purple";

const String backgrndColour = "gainsboro";const String data1Colour        = "red";

const String data2Colour      = "orange";

const char* ssid        = "chickentree"; const char* password  = "12345678";

const char* Timezone  = "GMT0BST,M3.5.0/01,M10.5.0/02";

// Example time zones

//const char* Timezone = "MET-1METDST,M3.5.0/01,M10.5.0/02";

//const char* Timezone = "CET-1CEST,M3.5.0,M10.5.0/3";

//const char* Timezone = "EST-2METDST,M3.5.0/01,M10.5.0/02";

//const char* Timezone = "EST5EDT,M3.2.0,M11.1.0";

//const char* Timezone = "CST6CDT,M3.2.0,M11.1.0";

//const char* Timezone = "MST7MDT,M4.1.0,M10.5.0";
```

```
//const char* Timezone = "NZST-12NZDT,M9.5.0,M4.1.0/3";

//const char* Timezone = "EET-2EEST,M3.5.5/0,M10.5.5/0";

//const char* Timezone = "ACST-9:30ACDT,M10.1.0,M4.1.0/3":

// System values

String sitetitle          = "Brew Monitoring";String Year           = "2021";

float  temp               = 0; float  specificGravity        = 0;float  tiltAngle          = 0;

float  TargetTemp         = 21;

int     FrostTemp         = 5;

float  ManOverrideTemp         = 21;float  MaxTemperature = 28;


bool   ManualOverride  = false;int     EarlyStart       = 0;

String RelayState          = "OFF";

String TimerState          = "OFF";String Units       = "M";


String webpage             = "";

int     TimerCheckDuration = 5000;int       LastReadingDuration  = 1;

int     LastTimerSwitchCheck = 0;int       LastReadingCheck     = 0;

float  LastTemperature        = 0;int   UnixTime       = 0;


AsyncWebServer server(80);


static size_t content_len;

#define PD 3 // degrees of polynomial, 3 is good

extern const char* angle_gravity;

// polynomial data

static int N = 0; // no. of data-points
```

```cpp
static const int pdMax = 6; // maximum degrees of polynomialstatic double coEff[pdMax]; //
polynomial coefficients

static double B[pdMax+1][pdMax+2]; // the normal augmented matrix

static char hubURL[30]; static char jsonMessage[100];static bool stayAwake = true;static
uint32_t awakeTime;

static esp_adc_cal_characteristics_t *adc_chars; // holds ADC characteristics static const
adc_atten_t ADCatten = ADC_ATTEN_DB_11; // attenuation levelstatic const adc_unit_t
ADCunit = ADC_UNIT_1; // using ADC1

static const adc_bits_width_t ADCbits = ADC_WIDTH_BIT_11; // ADC bit resolution


#define uS_TO_S_FACTOR 1000000 // Conversion factor for micro seconds to seconds

#define DEFAULT_VREF 1100 // if eFuse or two point not available on old ESPs


// MPU6050 definitions

#define MPU6050_ADDR 0x68 // MPU6050 I2C address if AD0 grounded, 0x69 if pulled
high

#define SENS_2G (32768.0/2.0) // divider for 2G sensitivity reading#define
ACCEL_BYTES 6 // 2 bytes per axis

#define CONFIG 0x1A

#define ACCEL_CONFIG 0x1C #define ACCEL_XOUT_H 0x3B#define PWR_MGMT_1
0x6B #define NO_VAL 0xFF


//WebServer server(80);

byte writeMPU6050(byte regAddr, byte regVal, bool busRelease = true){
Wire.beginTransmission(MPU6050_ADDR);
if (regAddr != NO_VAL) Wire.write(regAddr);if (regVal != NO_VAL) Wire.write(regVal);
```

```cpp
    return Wire.endTransmission(busRelease);

    }

    bool setupMPU6050() {

    // check MPU6050 is available

    Wire.begin();

    if (byte err = writeMPU6050(NO_VAL, NO_VAL)) {

    printf("MPU6050 not found at %#04x with error %u\n", MPU6050_ADDR, err);return false;

    } else {

    writeMPU6050(CONFIG, 0x00); // set full range writeMPU6050(PWR_MGMT_1, 0x01); //

    wakeup the sensor

    return true;

    }

    }

    void StartWiFi() {

    //Serial.print("\r\nConnecting to: "); Serial.println(String(ssid));IPAddress dns(8, 8, 8, 8); //

    Use Google as DNS WiFi.disconnect();

    WiFi.mode(WIFI_STA);          // switch off APWiFi.setAutoConnect(true);

    WiFi.setAutoReconnect(true); WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {Serial.print(".");

    delay(5000);

    }

    Serial.println("\nWiFi connected at: " + WiFi.localIP().toString());

    }

    String WiFiSignal() {
```

```cpp
float Signal = WiFi.RSSI();

Signal = 90 / 40.0 * Signal + 212.5;if (Signal > 100) Signal = 100; return " " + String(Signal,

0) + "%";

}


void append_HTML_header(bool refreshMode) { webpage  = "<!DOCTYPE html><html

lang='en'>";webpage += "<head>";

webpage += "<title>" + sitetitle + "</title>";webpage += "<meta charset='UTF-8'>";

  if (refreshMode) webpage += "<meta http-equiv='refresh' content='5'>"; // each time
refresh

webpage += "<script src=\"https://code.jquery.com/jquery-3.2.1.min.js\"></script>";webpage

+= "<style>";

  webpage += "body                       {width:68em;margin-left:auto;margin-
right:auto;font-family:Arial,Helvetica,sans-serif;font-size:14px;color:blue;background-
color:#e1e1ff;text- align:center;}";

webpage += ".centre          {margin-left:auto;margin-right:auto;}";

webpage += "h2              {margin-top:0.3em;margin-bottom:0.3em;font-size:1.4em;}";

webpage += "h3              {margin-top:0.3em;margin-bottom:0.3em;font-size:1.2em;}";

webpage += "h4              {margin-top:0.3em;margin-bottom:0.3em;font-size:0.8em;}";

webpage += ".on            {color: red;}";

webpage += ".off           {color: limegreen;}";

webpage += ".topnav          {overflow: hidden;background-color:lightcyan;}";

  webpage += ".topnav a       {float:left;color:blue;text-align:center;padding:1em
1.14em;text-decoration:none;font-size:1.3em;}";

webpage += ".topnav a:hover {background-color:deepskyblue;color:white;}";webpage +=
```

".topnav a.active {background-color:lightblue;color:blue;}";

  webpage += "table tr, td   {padding:0.2em 0.5em 0.2em 0.5em;font-size:1.0em;font-family:Arial,Helvetica,sans-serif;}";

  webpage   +=   "col:first-child {background:lightcyan}col:nth-child(2){background:#CCC}col:nth-child(8){background:#CCC}";

webpage += "tr:first-child {background:lightcyan}";

webpage += ".large   {font-size:1.8em;padding:0;margin:0}"; webpage += ".medium {font-size:1.4em;padding:0;margin:0}";webpage += ".ps {font-size:0.7em;padding:0;margin:0}";

webpage += "#outer   {width:100%;display:flex;justify-content:center;}"; webpage += "footer   {padding:0.08em;background-color:cyan;font-size:1.1em;}";

  webpage += ".numberCircle   {border-radius:50%;width:2.7em;height:2.7em;border:0.11emsolid blue;padding:0.2em;color:blue;text-align:center;font-size:3em;";

webpage += "   display:inline-flex;justify-content:center;align-items:center;}";

webpage += ".wifi   {padding:3px;position:relative;top:1em;left:0.36em;}";

  webpage += ".wifi, .wifi:before {display:inline-block;border:9px double transparent;border-top-color:currentColor;border-radius:50%;}";

webpage += ".wifi:before   {content:'';width:0;height:0;}";webpage += "</style></head>";

webpage += "<body>";

webpage += "<div class='topnav'>";
webpage += "Status"; webpage += "Graphs";

webpage += "<div class='wifi'/></div>" + WiFiSignal() + "";webpage += "</div>
";

```
    }


void append_HTML_footer() {webpage += "<footer>";

 webpage += "<p class='medium'>Brew Monitoring</p>";

 webpage += "<p class='ps'><i>Copyright &copy; Victor Willa " + String(Year) + " V"
+ version + "</i></p>"; webpage += "</footer>";

 webpage += "</body></html>";

 }


void CheckTimerEvent() { String TimeNow; UpdateTargetTemperature();

 TimeNow      = ConvertUnixTime(UnixTime);TimerState = "OFF";

 if (EarlyStart > 0) {

 TimeNow = ConvertUnixTime(UnixTime + EarlyStart * 60);

 }

 if (ManualOverride == ON) {
TargetTemp = ManOverrideTemp;ControlHeating();

 }

 for (byte dow = 0; dow < 7; dow++) {

 for (byte p = 0; p < NumOfEvents; p++) {


    if (String(dow) == DoW_str && (TimeNow >= Timer[dow].Start[p] && TimeNow <=
Timer[dow].Stop[p] && Timer[dow].Start[p] != ""))

 {

 TimerState = "ON"; ControlHeating(); ManualOverride = OFF;

 }

 }
```

```
    }

  CheckAndSetFrostTemperature();

  }

  void ControlHeating() {

if (temp < (TargetTemp - Hysteresis)) {ActuateHeating(ON);

  }

if (temp > (TargetTemp + Hysteresis)) {ActuateHeating(OFF);

  }
if (temp > MaxTemperature) {ActuateHeating(OFF);

  }

  }

void UpdateTargetTemperature() {String TimeNow;

  TimeNow = ConvertUnixTime(UnixTime);for (byte dow = 0; dow < 7; dow++) {

  for (byte p = 0; p < NumOfEvents; p++) {

      if (String(dow) == DoW_str && (TimeNow >= Timer[dow].Start[p] && TimeNow <
  Timer[dow].Stop[p]))

  {

  TargetTemp = Timer[dow].temp[p].toFloat();

  }

  }

  }

  if (ManualOverride == ON) TargetTemp = ManOverrideTemp;

  // Serial.println("Target Temperature = " + String(TargetTemp, 1) + "°");

  }
```

```
void CheckAndSetFrostTemperature() {

if (TimerState == "OFF" && ManualOverride == OFF) {if (temp < (FrostTemp - Hysteresis))

 { ActuateHeating(ON);

 //  Serial.println("Frost protection actuated...");

 }

if (temp > (FrostTemp + Hysteresis)) {ActuateHeating(OFF);

 }

 }

 }


void ActuateHeating(bool demand) {pinMode(RelayPIN, OUTPUT); pinMode(LEDPIN,

 OUTPUT);

if (demand) { RelayState = "ON";if (RelayReverse) {

 digitalWrite(RelayPIN, LOW);

 }

 else

 {

 digitalWrite(RelayPIN, HIGH);

 }

 digitalWrite(LEDPIN, LOW);

 // Serial.println("Thermostat ON");

 }

 else
```

```
{

RelayState = "OFF";if (RelayReverse) {

digitalWrite(RelayPIN, HIGH);

}

else

{

digitalWrite(RelayPIN, LOW);

}

digitalWrite(LEDPIN, HIGH);

//Serial.println("Thermostat OFF");

}

}

void SetupDeviceName(const char *DeviceName) {

if (MDNS.begin(DeviceName)) { // The name that will identify your device on the network

//Serial.println("mDNS responder started");

//Serial.print("Device name: ");

// Serial.println(DeviceName);

MDNS.addService("n8i-mlp", "tcp", 23); // Add service

}

else

Serial.println("Error setting up MDNS responder");

}

//##############################################################################
###############
```

```
void SetupSystem() { Serial.begin(115200);delay(200);

 // Serial.println(__FILE__);

 //Serial.println("Starting...");

 }



boolean SetupTime() { configTime(0, 0, "time.nist.gov");

 setenv("TZ", Timezone, 1);tzset();

 delay(200);

 bool TimeStatus = UpdateLocalTime();return TimeStatus;

 }



boolean UpdateLocalTime() {struct tm timeinfo;

 time_t now;

 char  time_output[30];
while (!getLocalTime(&timeinfo, 15000)) {return false;

 }

 time(&now);
 UnixTime = now;



 strftime(time_output, sizeof(time_output), "%H:%M", &timeinfo);Time_str = time_output;

 strftime(time_output, sizeof(time_output), "%w", &timeinfo);DoW_str  = time_output;

 return true;

 }



String ConvertUnixTime(int unix_time) {time_t tm = unix_time;
```

```
struct tm *now_tm = localtime(&tm);char output[40];

strftime(output, sizeof(output), "%H:%M", now_tm);return output;

}


void StartSPIFFS() {

//Serial.println("Starting SPIFFS");boolean SPIFFS_Status; SPIFFS_Status =

SPIFFS.begin(); if (SPIFFS_Status == false)

{ // Most likely SPIFFS has not yet been formated, so do so

// Serial.println("Formatting SPIFFS (it may take some time)...");
SPIFFS.begin(true); // Now format SPIFFS

File datafile = SPIFFS.open("/" + DataFile, "r");if (!datafile || !datafile.isDirectory()) {

    // Serial.println("SPIFFS failed to start..."); // Nothing more can be done, so delete and then
create another file

SPIFFS.remove("/" + DataFile); // The file is corrupted!!datafile.close();

}

}

else Serial.println("SPIFFS Started successfully...");

}


void Initialise_Array() {

  Timer[0].DoW = "Sun"; Timer[1].DoW = "Mon"; Timer[2].DoW = "Tue"; Timer[3].DoW =
"Wed"; Timer[4].DoW = "Thu"; Timer[5].DoW = "Fri"; Timer[6].DoW = "Sat";

}

void SaveSettings() {

// Serial.println("Getting ready to Save settings...");File dataFile = SPIFFS.open("/" +

DataFile, "w"); if (dataFile) { // Save settings
```

```
// Serial.println("Saving settings..."); for (byte dow = 0; dow < 7; dow++) {

// Serial.println("Day of week = " + String(dow));for (byte p = 0; p < NumOfEvents; p++) {

 dataFile.println(Timer[dow].temp[p]);

 dataFile.println(Timer[dow].Start[p]);dataFile.println(Timer[dow].Stop[p]);

    // Serial.println("Period: " + String(p) + " " + Timer[dow].temp[p] + " frpm: " +
Timer[dow].Start[p] + " to: " + Timer[dow].Stop[p]);

 }

 }

 dataFile.println(Hysteresis, 1);

 dataFile.println(FrostTemp, 1);dataFile.println(EarlyStart);

 //Serial.println("Saved Hysteresis : " + String(Hysteresis));

 //Serial.println("Saved Frost Temp : " + String(FrostTemp));

 //Serial.println("Saved EarlyStart : " + String(EarlyStart));dataFile.close();

 // Serial.println("Settings saved...");

 }

 }


void RecoverSettings() {String Entry;

 //Serial.println("Reading settings...");

 File dataFile = SPIFFS.open("/" + DataFile, "r");if (dataFile) { // if the file is available, read it

 //Serial.println("Recovering settings...");while (dataFile.available()) {

 for (byte dow = 0; dow < 7; dow++) {
// Serial.println("Day of week = " + String(dow));for (byte p = 0; p < NumOfEvents; p++) {

 Timer[dow].temp[p] = dataFile.readStringUntil('\n'); Timer[dow].temp[p].trim();

 Timer[dow].Start[p] = dataFile.readStringUntil('\n'); Timer[dow].Start[p].trim();
```

Timer[dow].Stop[p] = dataFile.readStringUntil('\n'); Timer[dow].Stop[p].trim();

// Serial.println("Period: " + String(p) + " " + Timer[dow].temp[p] + " frpm: " + Timer[dow].Start[p] + " to: " + Timer[dow].Stop[p]);

}

}

Entry = dataFile.readStringUntil('\n'); Entry.trim(); Hysteresis = Entry.toFloat();Entry =

dataFile.readStringUntil('\n'); Entry.trim(); FrostTemp = Entry.toInt(); Entry =

dataFile.readStringUntil('\n'); Entry.trim(); EarlyStart = Entry.toInt();

// Serial.println("Recovered Hysteresis : " + String(Hysteresis));

//Serial.println("Recovered Frost Temp : " + String(FrostTemp));

//Serial.println("Recovered EarlyStart : " + String(EarlyStart));dataFile.close();

// Serial.println("Settings recovered...");

}

}

}

void Graphs() { append_HTML_header(Refresh);

webpage += "<h2>Sensor Node Readings</h2>";

webpage += "<script type='text/javascript' src='https://www.gstatic.com/charts/loader.js'></script>";

webpage += "<script type='text/javascript'>";

webpage += "google.charts.load('current', {'packages':['corechart']});";

webpage += "google.charts.setOnLoadCallback(drawGraphT1);"; // Pre-load function names for Temperature graphs

webpage += "google.charts.setOnLoadCallback(drawGraphH1);"; // Pre-load function names for Humidity graphs

```
AddGraph(1, "GraphT", "Temperature", "TS", "°C", "red",  "chart_div");

AddGraph(1, "GraphH", "SpecificGravity",      "HS", "%", "blue", "chart_div");webpage

+= "</script>";

webpage += "<div id='outer'>";webpage += "<table>"; webpage += "<tr>";

webpage += " <td><div id='chart_divTS1' style='width:50%'></div></td>"; webpage += "

<td><div id='chart_divHS1' style='width:50%'></div></td>"; webpage += "</tr>";

webpage += "</table>";webpage += "<br>"; webpage += "</div>";


append_HTML_footer();

}
String PreLoadChartData(byte Channel, String Type) {writeMPU6050(ACCEL_XOUT_H,

NO_VAL, false);

  Wire.requestFrom(MPU6050_ADDR, ACCEL_BYTES+2); // read 3 axis accelerometer &
temperature

int16_t rawX = Wire.read() << 8 | Wire.read(); int16_t rawY = Wire.read() << 8 |

Wire.read(); int16_t rawZ = Wire.read() << 8 | Wire.read(); int16_t rawTemp = Wire.read()

<< 8 | Wire.read();

// each axis G force value, straight down is 1.0 if stationarydouble gX = (double)(rawX /

SENS_2G);

double gY = (double)(rawY / SENS_2G);double gZ = (double)(rawZ / SENS_2G);

//printf("gX : %0.2f\n", gX); // pitch

// calculate gravity from all 3 axes to eliminate roll double gXYZ =

sqrt(pow(gX,2)+pow(gY,2)+pow(gZ,2));

//printf("gXYZ : %0.2f\n", gXYZ);


// axis used for pitch is whichever is linear to PETling length
```

```
// generally this will be the X axis

// get tilt angle (pitch) wrt to horizontal and convert to degreesdouble ratio = gX/gXYZ;

  float    tiltAngle  =  (float)((ratio  <  0.5)  ?  90-fabs(asin(ratio)*RAD_TO_DEG)
          :fabs(acos(ratio)*RAD_TO_DEG));

// printf("tiltAngle: %0.1f\n", tiltAngle);


// calculate SG using polynomialfloat specificGravity = coEff[0];

for (int i=1; i<=PD; i++) specificGravity += coEff[i]*pow(tiltAngle,i);

//printf("specificGravity : %0.4f\n", specificGravity);

float temp = (rawTemp / 340.0) + 36.53; // degrees celsius

// printf("Temp : %0.1fC\n", temp);


byte r = 0; String Data = "";do {

if (Type == "Temperature") {

    Data += "[" + String(r) + "," + String(sensordata[Channel][r].temp, 1) + "," + String(temp,
1) + "],";

}

else

{

    Data += "[" + String(r) + "," + String(sensordata[Channel][r].temp, 1) + "," +
String(specificGravity, 1) + "],";

} r++;

} while (r < temp);Data  += "]"; return Data;

}
```

```
void AddGraph(byte Channel, String Type, String Title, String GraphType, String Units, String
Colour, String Div) {

String Data = PreLoadChartData(Channel, Title);


webpage += "function draw" + Type + String(Channel) + "() {";if (Type == "GraphT") {

    webpage += " var data = google.visualization.arrayToDataTable(" + String("[['Hour', 'Ferm
T°', 'Temp°'],") + Data + ");";

}
else

    webpage += " var data = google.visualization.arrayToDataTable(" + String("[['Hour', 'SG',
'],") + Data + ");";

webpage += " var options = {"; webpage += " title: '" + Title + "',";webpage += "

titleFontSize: 14,";

webpage += " backgroundColor: '" + backgrndColour + "',"; webpage += " legendTextStyle:

{ color: '" + legendColour + "' },";webpage += " titleTextStyle: { color: '" + titleColour + "'

},"; webpage += " hAxis: {color: '#FFF'},";

webpage += " vAxis: {color: '#FFF', title: '" + Units + "'},";webpage += " curveType:

'function',";

webpage += " pointSize: 1,";webpage += " lineWidth: 1,";webpage += " width: 450,";

webpage += " height: 280,";

webpage += " colors:['" + Colour + (Type == "GraphT" ? "', 'orange" : "") + "'],";webpage

+= " legend: { position: 'right' }";

webpage += " };";

webpage += " var chart = new google.visualization.LineChart(document.getElementById('"
+ Div + GraphType + String(Channel) + "'));";webpage += " chart.draw(data, options);";
```

```
webpage += " };";

}

void Homepage() {

calculateSG();

}

void calculateSG() { writeMPU6050(ACCEL_XOUT_H, NO_VAL, false);

  Wire.requestFrom(MPU6050_ADDR, ACCEL_BYTES+2); // read 3 axis accelerometer &
  temperature

int16_t rawX = Wire.read() << 8 | Wire.read(); int16_t rawY = Wire.read() << 8 |

Wire.read(); int16_t rawZ = Wire.read() << 8 | Wire.read(); int16_t rawTemp = Wire.read()

<< 8 | Wire.read();

// each axis G force value, straight down is 1.0 if stationarydouble gX = (double)(rawX /

SENS_2G);

double gY = (double)(rawY / SENS_2G);double gZ = (double)(rawZ / SENS_2G);printf("gX

: %0.2f\n", gX); // pitch

// calculate gravity from all 3 axes to eliminate roll
double gXYZ = sqrt(pow(gX,2)+pow(gY,2)+pow(gZ,2));printf("gXYZ : %0.2f\n", gXYZ);



// axis used for pitch is whichever is linear to PETling length

// generally this will be the X axis

// get tilt angle (pitch) wrt to horizontal and convert to degreesdouble ratio = gX/gXYZ;

  float    tiltAngle   =   (float)((ratio   <   0.5)   ?   90-fabs(asin(ratio)*RAD_TO_DEG)
          :fabs(acos(ratio)*RAD_TO_DEG));

printf("tiltAngle: %0.1f\n", tiltAngle);
```

```cpp
// calculate SG using polynomialfloat specificGravity = coEff[0];

for (int i=1; i<=PD; i++) specificGravity += coEff[i]*pow(tiltAngle,i);printf("specificGravity

: %0.4f\n", specificGravity);

float temp = (rawTemp / 340.0) + 36.53; // degrees celsiusprintf("Temp : %0.1fC\n", temp);

server.on("/", HTTP_GET, [](AsyncWebServerRequest * request) {request-

>redirect("/homepage");                    // Go to home page

});

server.on("/homepage", HTTP_GET, [](AsyncWebServerRequest * request) {Homepage();

request->send(200, "text/html", webpage);

});

// Set handler for '/graphs'

server.on("/graphs", HTTP_GET, [](AsyncWebServerRequest * request) {Graphs();

request->send(200, "text/html", webpage);

});

// Set handler for '/handletimer' inputs

server.on("/handletimer", HTTP_GET, [](AsyncWebServerRequest * request) {for (byte dow

= 0; dow < 7; dow++) {

for (byte p = 0; p < 4; p++) {

Timer[dow].temp[p] = request->arg(String(dow) + "." + String(p) + ".temp");

Timer[dow].Start[p] = request->arg(String(dow) + "." + String(p) + ".Start");

Timer[dow].Stop[p]  = request->arg(String(dow) + "." + String(p) + ".Stop");

}

}

SaveSettings();
```

```cpp
    request->redirect("/homepage");

  });
server.on("/handlesetup", HTTP_GET, [](AsyncWebServerRequest * request) {if (request-

  >hasArg("hysteresis")) {

  String numArg = request->arg("hysteresis");Hysteresis = numArg.toFloat();

  }

  if (request->hasArg("frosttemp")) {

  String numArg = request->arg("frosttemp");FrostTemp      = numArg.toFloat();

  }

  if (request->hasArg("earlystart")) {

  String numArg = request->arg("earlystart");EarlyStart      = numArg.toInt();

  }

  if (request->hasArg("manualoverride")) {

  String stringArg = request->arg("manualoverride");

  if (stringArg == "ON") ManualOverride = true; else ManualOverride = false;

  }

  if (request->hasArg("manualoverridetemp")) {

  String numArg = request->arg("manualoverridetemp");ManOverrideTemp =

  numArg.toFloat();

  }

  SaveSettings();

  request->redirect("/homepage");

  });
```

```
server.begin();

// Get current sensor values

ActuateHeating(OFF);                                    // Switch heating OFF

                        // Get current sensor values LastTimerSwitchCheck = millis() +

TimerCheckDuration;

ActuateHeating(OFF);


LastTimerSwitchCheck = millis() + TimerCheckDuration;


append_HTML_header(Refresh);

webpage += "<h2>Brew Monitoring</h2><br>";

 webpage += "<div class='numberCircle'><span class=" + String((RelayState == "ON" ?
"'on'>" : "'off'>")) + String(temp, 1) + "&deg;</span></div><br><br><br>";

webpage += "<table class='centre'>";webpage += "<tr>";

webpage += "<td>Temperature</td>"; webpage += "<td>SpecificGravity</td>";webpage +=

"<td>TiltAngle</td>";


if (ManualOverride) {

webpage += "<td>ManualOverride</td>";

}

webpage += "</tr>";webpage += "<tr>";

webpage += "<td class='large'>" + String(temp, 1)           + "&deg; C</td>"; webpage +=

"<td class='large'>" + String(specificGravity, 1)           + "</td>";webpage += "<td

class='large'>" + String(tiltAngle, 1) + "&deg;</td>";

if (ManualOverride) {
```

```cpp
    webpage += "<td class='large'>" + String(ManualOverride ? "ON" : "OFF") + "</td>";

    }

    webpage += "</tr>"; webpage += "</table>"; webpage += "<br>"; append_HTML_footer();

    }

    size_t extractPairVals(const char* pairValStr, const char* itemSep, const char* pairSep,
    std::vector<double> &pairVals) {

    // populate numeric array from paired numeric values in stringchar* pairStart =

    strdup(pairValStr);

    char* pairEnd = strchr(pairStart, pairSep[0]);int i = 0;

    while (pairEnd != NULL) {

    *pairEnd++ = '\0'; // replace pair separator with string terminatorif (strlen(pairStart)) {

    char* pairItem = strtok(pairStart, itemSep); // get pairwhile (pairItem != NULL) {

    pairVals.push_back (atof(pairItem));

    pairItem = strtok(NULL, itemSep); // next item in pair
    } i++;

    }

    pairStart = pairEnd; // point to start of next pair

    pairEnd = strchr(pairStart, pairSep[0]); // point to end of next pair

    }

    return i; // number of pairs

    }
static void start_handler(AsyncWebServerRequest *request) {puts("User requested deep

    sleep");

    request->send(200, "text/html", " ");delay(1000); esp_deep_sleep_start();

    }
```

```cpp
static void reset_handler(AsyncWebServerRequest *request) {puts("User requested reset");

  request->send(200, "text/html", " ");delay(1000);

  ESP.restart();

  }
  void generatePolynomial() {

    // Fit a polynomial curve to a given set of data points using the Least Squares Approximation
    Method

  // and return the polynomial coefficientsint i,j,k;

  // read in data point pairs static std::vector<double> dp;

  N = extractPairVals(angle_gravity, " ", "\n", dp);


  double X[2*PD+1] = {0.0};

  for (i=0; i<=2*PD; i++) for (j=0; j<N; j++) X[i] += pow(dp[j*2],i);


  // rhs

  double Y[PD+1] = {0.0};

  for (i=0; i<=PD; i++) for (j=0; j<N; j++) Y[i] += pow(dp[j*2],i)*dp[(j*2)+1];for (i=0;

  i<=PD; i++) for (j=0; j<=PD; j++) B[i][j] = X[i+j];

  for (i=0; i<=PD; i++) B[i][PD+1] = Y[i];


  // gauss elimination for (i=0; i<PD; i++) {

  // Partial Pivoting

  for (k=i+1; k<PD+1; k++) {

  // If diagonal element(absolute value) is smaller than any of the terms below it
  if (fabs(B[i][i]) < fabs(B[k][i])) {

  // Swap the rows

  for (j=0; j<PD+2; j++) {double temp = B[i][j]; B[i][j] = B[k][j];
```

```
    B[k][j] = temp;

    }

    }

    }

for (k=i+1; k<PD+1; k++) { double term = B[k][i]/B[i][i];

 for(j=0; j<PD+2; j++) B[k][j] = B[k][j]-term*B[i][j];

    }

    }


    // Back-substitution for (i=PD; i>=0; i--) {

    coEff[i] = B[i][PD+1];

    for (j=i+1; j<PD+1; j++) coEff[i] -= B[i][j]*coEff[j];coEff[i] /= B[i][i];

    }

    printf("\nPolynomial coefficients: ");

    for (i=0; i<=PD; i++) printf("%u:%0.9f ",i, coEff[i]);
    puts("");

    }

    void setup() { SetupSystem(); StartWiFi(); SetupTime(); StartSPIFFS();

    Initialise_Array(); RecoverSettings(); SetupDeviceName(ServerName);calculateSG();

    setupMPU6050(); generatePolynomial();

    }


    void loop() {

    static bool blinking = true;blinking = !blinking; delay(1000);

     calculate  SG()
```

Appendix 5:   Poster Presentation