Bindura University of Science Education

FACULTY OF SCIENCE EDUCATION AND ENGINEERING

DEPAERTMENT OF PHYSICS AND ENGINEERING

BY

NAME:	**TICHAONA MUSIKAVANHU**
STUDENT NUMBER:	**B192861B**
PROGRAM:	**B.Sc.ELECTRONIC ENGINEERING (HBScEE)**
COURSE NARRATION:	**FINAL YEAR RESEARCH PROJECT**
COURSE CODE:	**EEN5106-EEN5206**
TITLE:	**REAL TIME FUEL MONITORING AND DATA VISUALISATION**
SUPERVISOR:	**ENGINEER S KOMICH**
DURATION	**2023-2024**

DECLARATION

I, Tichaona Musikavanhu, declare that the contents of this thesis represent my own unaided work, and that the thesis has not previously been submitted in any previous application for a degree. Except where states otherwise by reference or acknowledgement, the content represents my own opinions and not necessarily those of the Bindura University of Science Education

______________________________ Date: ____________

Tichaona Musikavanhu

DISCLAIMER

All the work done in this thesis report belongs to the student alone. With supervision, direction, and support from the university, students were encouraged to take charge of their whole thesis research. Conclusions and recommendations made in this study are not necessarily those of the university or study supervisors.

ACKNOWLEDGEMENTS

I really appreciate Zimbabwe Power Company under ZESA for opportunity they offered me to be one of attachee from November 2022 to August 2023, Eng S Komich, my supervisor, for his invaluable advice and assistance in making this work feasible. Additionally, I would like to thank my friends Darlington Dzama ,Eng. D Panje and Eng. A. Sibanda for all of their encouragement and support. Finally, I want to express my gratitude to God for keeping me alive throughout, as well as to my loving parents for their prayers and sacrifices.

ABSTRACT

The project aims to address the challenges faced by the Zimbabwe Power Company's Hwange Power Station in monitoring and managing fuel consumption. The background highlights the significance of thermal power plants, particularly the Hwange Power Station, in meeting the global demand for electricity. It emphasizes the need for efficient fuel management and the impact of technological advancements on traditional monitoring methods.

Overall, the project seeks to modernize fuel monitoring at the Hwange Power Station, ensuring uninterrupted energy supply, mitigating security risks, and enhancing operational integrity.

Contents

DECLARATION i

DISCLAIMER ii

ACKNOWLEDGEMENTS iii

ABSTRACT iv

CHAPTER 1: INTRODUCTION 1

1.1 OVERVIEW 1

1.2 BACKGROUND 1

1.3 PROBLEM STATEMENT 1

1.4 AIM 1

1.5 OBJECTIVES 2

1.6 JUSTIFICATION OF STUDY 2

CHAPTER 2: LITERATURE REVIEW 3

2.1 INTRODUCTION 3

2.2 EXISTING SYSTEMS 3

2.2.1 Micro-controller-based measurements system 3

2.3 ESP 32 MICROCONTROLLER 5

2.3.1 Overview 5

2.3.2 PIN DEFINITIONS 8

2.3.3 Functional Description 13

2.4 FLOWMETER 15

2.5 DATABASE 17

2.5.1. Introduction 17

2.5.2. Definition of Database 18

2.5.3 Importance of Databases in Modern Society 19

2.6 SD CARD MODULE 20

2.7 POWER SUPPLY 21

CHAPTER 3 DESIGN AND METHODOLOGY 22

3.1 INTRODUCTION 22
3.2PROPOSED PROJECT WORKING PRINCIPLE.. 22
3.3OVERVIEW BLOCK DIAGRAM OF THE PROPOSED SYSTEM 23
3.4 INSTALLING FLOWMETERS IN THE PIPE LINE FLOWCHART 23
3.7 INTERFACING FLOWMETERS TO ESP32 CONTROLLER 25
3.5 CALCULATING TANK LEVELS AND FUEL USED 27
3.6 DATABASE DESIGN.. 28
3.7 DATA ACQUISITION FROM THE ESP32 CONTROLLER INTO THE DATABASE 32
3.8 DESIGN USER-FRIENDLY VISUALISATION TOOLS TO PRESENT FUEL CONSUMPTION TRENDS AND HISTORICAL DATA 33
CHAPTER 4: RESULTS AND ANALYSIS 37
4.1 INTRODUCTION 37
4.2 INSTALLING OF FLOWMETERS IN THE PIPELINE 37
4.3 DATABASE RESULTS.. 40
4.4 DATA VISUALISATION ON WEBSITE RESULTS 40
CHAPTER 5: RECOMMENDATIONS AND CONCLUSIONS 44
5.1: INTRODUCTION 44
5.2 DISCUSSION OF PERFORMANCE 44
5.3 LIMITATIONS OF THE PROJECT PROTOTYPE.. 44
5.4 RECOMMENDATIONS.. 44
5.5 CONCLUSIONS.. 45
REFERENCES 47

CHAPTER 1: INTRODUCTION

1.1 OVERVIEW

In this chapter, the author shares the fundamentals and background that formed the backbone of the work project. A brief summary of the problem's background. This then leads to the problem statement, which comprises the ideal, the realistic, and the solution that brings about the whole project. Thereafter, the study shares the justification for the work done, along with the aims, objectives, and research questions.

1.2 BACKGROUND

Over the past century, thermal power plants have been instrumental in meeting the increasing global demand for electricity. Among these, the Hwange Power Station, operated by the Zimbabwe Power Company, stands as a significant contributor to the region's power supply. Efficient fuel management, encompassing delivery, storage, and consumption processes, is vital for optimal plant performance.

However, as technological advancements continue to reshape the engineering landscape, traditional methods for monitoring and managing fuel consumption at facilities like Hwange Power Station become outdated. In response, new methodologies emerge, integrating cutting-edge engineering solutions with heightened security measures. These innovations offer enhanced accuracy, real-time monitoring capabilities, and improved operational efficiency, ensuring uninterrupted energy supply while mitigating potential security risks.

1.3 PROBLEM STATEMENT

The key problem faced by the Zimbabwe Power Company Hwange Power Station is lack of robust and accurate real-time fuel monitoring system. The existing manual methods results in inaccuracies, delays in data acquisition, and limited insights for optimizing fuel consumption, creating conducive environment for theft and fraud. This include theft during the purchase and delivering of fuel as well as fraud involving the modification of collected data to conceal theft by falsifying the quantity of fuel bought on receipts and filling in volumes on the fuelling report spreadsheet that do not correspond to the actual volume of fuel bought

1.4 AIM

Develop and implement a robust and accurate real-time fuel monitoring system using flowmeters installed at key points, with data storage both on the server and locally on the microcontroller, addressing the challenges of fuel theft and fraud activities by providing a comprehensive solution that enables accurate measurement, real-time data acquisition, secure storage, and efficient analysis of fuel consumption data

1.5 OBJECTIVES

1. Installing flowmeters in series within the fuel supply lines to measure flowrate and calculate total flows accurately

2. Integrating all flowmeters to microcontroller and displaying data on serial monitor for remote monitoring

3. Set up a centralized monitoring system using xammp by creating a MySQL database within XAMP to securely store real-time fuel consumption data from flow sensors and calculated tanks volume

4. Design user-friendly visualisation tools to present fuel consumption trends and historical data

1.6 JUSTIFICATION OF STUDY

Implementing a robust real-time fuel monitoring system at Hwange Power Station, operated by the Zimbabwe Power Company, is critical to safeguarding assets and ensuring operational integrity. By addressing challenges such as fuel theft and fraudulent activities through accurate measurement, secure data storage, and efficient analysis, the project enhances operational efficiency and sustainability, making it a compelling investment opportunity for sponsors.

CHAPTER 2: LITERATURE REVIEW

2.1 INTRODUCTION

According to Justin Paul and Alex Rialp Criado (2020), a literature review article gives a thorough assessment of the literature on a subject, theory, or methodology and synthesizes earlier research to increase the body of knowledge.

In this chapter, the author gives an overview of the current implemented fuel management system, as well as previously related work on the project components.

2.2 EXISTING SYSTEMS

2.2.1 Micro-controller-based measurements system

The old method of measuring fuel tank level using a dipstick, pressure gauge, and manual record keeping had several disadvantages that highlighted the need for real-time monitoring.

The manual method is prone to inaccuracies due to human error in reading the dipstick or pressure gauge, leading to incorrect estimations of fuel levels. It is also a time-consuming process, requiring physical access to the fuel tank, which became impractical when dealing with multiple vehicles or large fuel storage tanks. The limited frequency of measurements resulted in data gaps and made it difficult to track fuel consumption patterns accurately.

Furthermore, manual record keeping did not provide real-time information about fuel levels, making it challenging to respond promptly to changes in fuel consumption or detect fuel leaks or theft. Regular maintenance and calibration of the equipment were also necessary to ensure accuracy, adding to the overall effort.

Real-time monitoring systems, such as electronic fuel level sensors or telematics solutions, addressed these limitations by providing accurate and up-to-date information on fuel levels. These systems allowed for better fuel management, timely refills, and proactive maintenance. Real-time monitoring also enabled the detection of anomalies, such as sudden fuel level drops, indicating potential issues like fuel theft or leaks.

Overall, real-time monitoring improved efficiency, reduced costs, and enhanced safety in fuel management processes by providing accurate and timely information, enabling proactive decision-making, and minimizing the need for manual intervention.

During the early 20th century, fuel consumption measurement in thermal power plants relied on manual methods. Operators manually calculated the amount of fuel consumed based on factors such as fuel flow rate, operating hours, and calorific value [1].

However, this approach was prone to errors and lacked the ability to provide real-time data for effective decision-making.

As technology progressed, the integration of instrumentation systems became prevalent in thermal power plants. In the 1930s to 1950s, various sensors and instruments, including flow meters, pressure transmitters, and temperature sensors, were introduced. These advancements allowed operators to gain better insights into fuel flow rates and temperature profiles, enabling more accurate fuel consumption measurement [2].

The advent of Supervisory Control and Data Acquisition (SCADA) systems in the 1960s to 1970s revolutionized fuel consumption measurement. SCADA systems combined hardware and software components to monitor and control power plant operations. Real-time data acquisition from instrumentation systems provided operators with accurate information about fuel flow rates, temperature, and other relevant parameters. SCADA systems significantly improved the efficiency and accuracy of fuel consumption measurement, enabling more precise control over power plant operations [3].

In the 1980s to 1990s, the widespread adoption of mass flow meters further enhanced fuel consumption measurement in thermal power plants. These devices directly measured the mass flow rate of the fuel, offering precise and real-time data. Mass flow meters were installed at various points in the fuel supply system, including fuel storage tanks, pipelines, and combustion chambers. Their integration improved the accuracy of fuel consumption measurement and facilitated better control over fuel usage [4].

Advancements in technology during the 2000s to 2010s led to the integration of advanced metering and automation systems in thermal power plants. Smart meters, control algorithms, and data analytics were implemented to optimize fuel consumption. These systems monitored fuel flow rates, adjusted combustion processes based on real-time data, and provided valuable insights for operational improvements. The integration of advanced metering and automation played a significant role in enhancing fuel efficiency and reducing environmental impact [5].

The constant progress in fuel consumption measurement techniques and technologies has enabled thermal power plants to accurately monitor and optimize fuel usage. These advancements have led to improved operational efficiency, cost savings, and a reduced environmental footprint, ensuring a greener and more sustainable future for thermal power generation.

2.3 ESP 32 MICROCONTROLLER

2.3.1 Overview

ESP32-WROOM-32D and ESP32-WROOM-32U are powerful, generic Wi-Fi+BT+BLE MCU modules that target a wide variety of applications, ranging from low-power sensor networks to the most demanding tasks, such as voice encoding, music streaming and MP3 decoding. ESP32-WROOM-32U is different from ESP32-WROOM-32D in that ESP32-WROOM-32U integrates a U. FL connector.

Table2.1: ESP32-WROOM-32D vs. ESP32-WROOM-32U

MODULE	ESP32-WROOM-32D	ESP32-WROOM-32U
Core	ESP32-D0WD	ESP32-D0WD
SPI flash	32 Mbits, 3.3V	32 Mbits, 3.3V
Crystal	40 MHz	40 MHz
Antenna	onboard antenna	U.FL connector (which needs to be connected to an external IPEX antenna)
Dimensions	(18±0.2) x (25.5±0.2) x (3.1±0.15) (See Figure 6 for details)	(18±0.1) x (19.2±0.1) x (3.2±0.1) (See Figure 7 for details)

At the core of the two modules are the ESP32-D0WD chip*. The chip embedded is designed to be scalable and adaptive. There are two CPU cores that can be individually controlled, and the CPU clock frequency is adjustable from 80 MHz to 240 MHz The user may power off the CPU and make use of the low-power co-processor to constantly monitor the peripherals for changes or crossing of thresholds. ESP32 integrates a rich set of peripherals, ranging from capacitive touch sensors, Hall sensors, SD card interface, Ethernet, high-speed SPI, UART, I2S and I2C.

The integration of Bluetooth, Bluetooth LE and Wi-Fi ensures that a wide range of applications can be targeted, and that the module is future proof: using Wi-Fi allows a large physical range and direct connection to the internet through a Wi-Fi router, while using Bluetooth allows the user to conveniently connect to the phone or broadcast low energy beacons for its detection. The sleep current of the ESP32 chip is less than 5 μA, making it suitable for battery powered and wearable electronics applications. ESP32 supports a data rate of up to 150 Mbps, and 20.5 dBm output power at the antenna to ensure the widest physical range. As such the chip does offer industry-leading specifications and the best performance for electronic integration, range, power consumption, and connectivity. The operating system chosen for ESP32 is freeRTOS with LwIP; TLS 1.2 with hardware acceleration is built in as well. Secure (encrypted) over the air (OTA) upgrade is also supported, so that developers can continually upgrade their products even after their release.

Table 2.2: ESP32-WROOM-32D and ESP32-WROOM-32U Specifications

Categories	Items	Specifications
Certification	RF Certification	FCC/CE (RED)/IC/TELEC/KCC/SRRC/NCC
	Wi-Fi Certification	Wi-Fi Alliance
	Bluetooth certification	BQB
	Green Certification	REACH/RoHS
Wi-fi	Protocols	802.11 b/g/n (802.11n up to 150 Mbps) A-MPDU and A-MSDU aggregation and 0.4 μs guard interval support
Bluetooth	Protocols	Bluetooth v4.2 BR/EDR and BLE specification
		NZIF receiver with –97 dBm sensitivity Class-1, class-2 and class-3 transmitter
	Radio	CVSD and SBC

	Audio	
Hardware	Module interface	SD card, UART, SPI, SDIO, I2C, LED PWM, Motor PWM, I2S, IR GPIO, capacitive touch sensor, ADC, DA
	On-chip sensor On-board clock Operating voltage/Power supply	Hall sensor 40 MHz crystal 2.7 ~ 3.6V
	Operating current	Average 80mA
	Minimum current delivered by power supply	500mA
	Recommended operating temperature	–40°C ~ +85°C
Software	Wi-Fi mode Wi-Fi Security Encryption	Station/SoftAP/SoftAP+Station/P2P WPA/WPA2/WPA2-Enterprise/WPS AES/RSA/ECC/SHA

	Firmware upgrade	UART Download / OTA (download and write firmware via network or host
	Software development	Supports Cloud Server Development / SDK for custom firmware development
	Network protocols	IPv4, IPv6, SSL, TCP/UDP/HTTP/FTP/MQTT
	User configuration	AT instruction set, cloud server, Android/iOS app

2.3.2 PIN DEFINITIONS

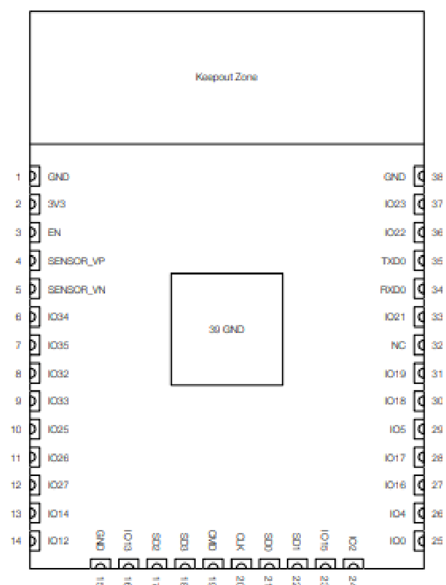


FIGURE2.1 ESP32 PIN LAYOUT

For those looking to add wireless connectivity to embedded projects or to build IoT devices, the ESP32 is arguably the most popular choice. It offers a dual-core option for processor intensive applications, while built-in WiFi and

Bluetooth simplify designs. It also has plenty of I/O, memory, and interoperability to meet the needs of most applications.[9]

Espressif, the company behind the ESP-32 chip, also manufactures modules and development boards. If you intend to design a custom PCB with an ESP32, the ESP32-WROOM-32 module is an excellent choice. Opting the module over the bare chip has several advantages. Notably, the module is already FCC-approved, which means you won't need to take any extra measures to achieve FCC compliance. This simplifies the manufacturing process considerably.[7]

When it comes to custom PCB design, understanding the pinout of the ESP32-WROOM-32 becomes crucial. This article will go over the ESP32-WROOM-32 pinout in detail.

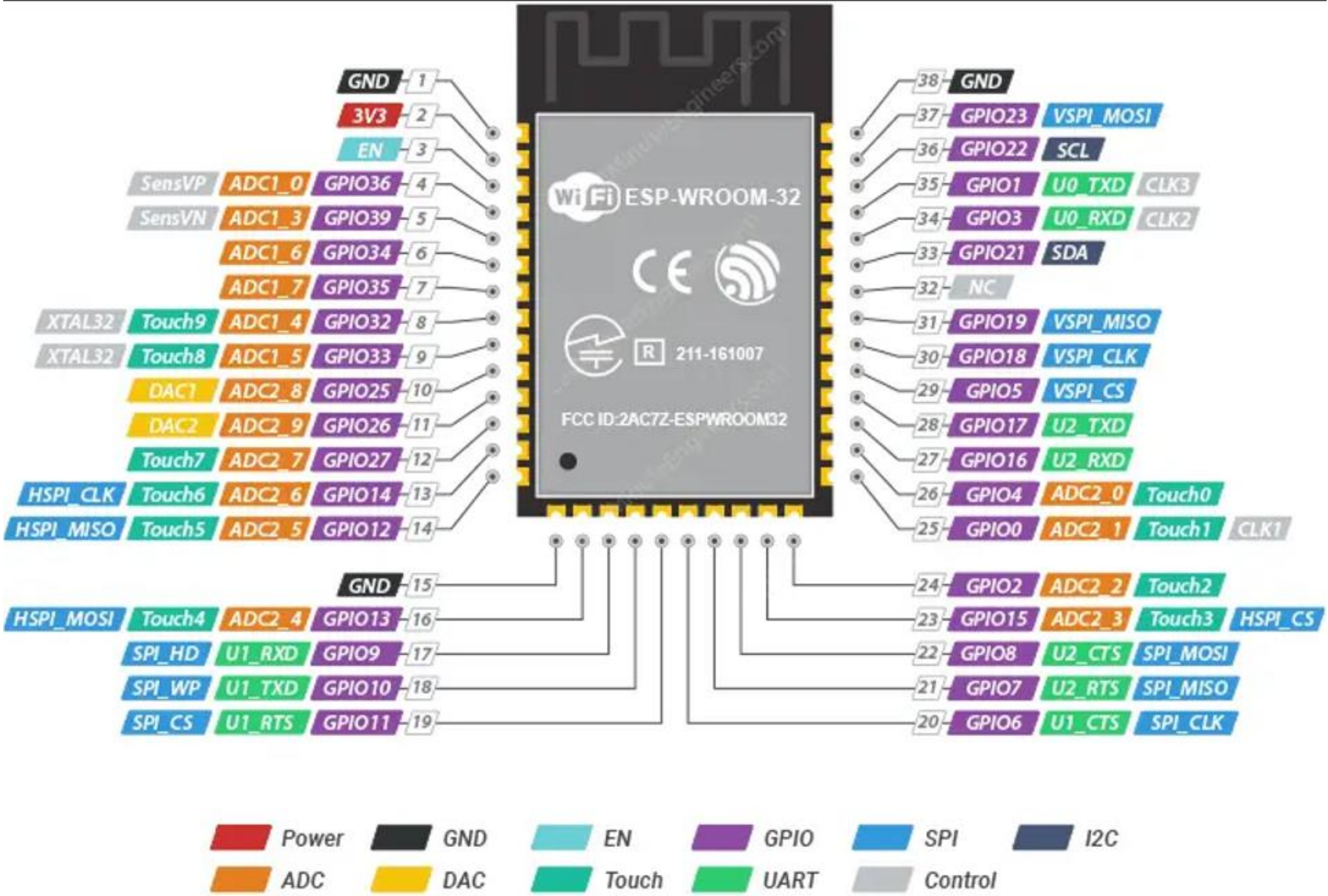


FIGURE2.2 ESP32-WROOM-32 PINOUT [lastminuteengineer.com]

Input Only GPIOs

Pins 4, 5, 6 and 7 cannot be conFigured as outputs. They can be used as digital or analog inputs, or for other purposes. They also lack internal pull-up and pull-down resistors, unlike the other GPIO pins.

ESP32 Interrupt Pins

All GPIOs can be conFigured as interrupts.

ADC Pins

The ESP32-WROOM-32 module has two ADC (Analog to Digital Converter) blocks; ADC1 and ADC2. Each block has multiple channels:

ADC1: contains six channels (labeled as ADC1_CH0 and ADC1_CH3 to ADC1_CH7).

ADC2: contains 10 channels (labeled as ADC2_CH0 to ADC2_CH9).

The resolution of the ADCs on the ESP32 can be conFigured up to 12 bits. This means that the ADC can detect 4096 (2^12) discrete analog levels. This results in a resolution of 3.3 V (operating voltage) / 4096 units, or 0.0008 volts (0.8 mV) per unit.

Warning:

When Wi-Fi is enabled, the ADC2 pins cannot be used. If you need Wi-Fi, consider using the ADC1 pins instead.

DAC Pins

The ESP32-WROOM-32 module contains two 8-bit Digital-to-Analog Converters (DACs). These DACs are useful for converting digital signals into analog voltages.

Touch Pins

The ESP32-WROOM-32 module has ten capacitive touch-enabled GPIOs labeled TOUCH0 through TOUCH9. These pins work by measuring the change in capacitance when a finger or conductive object is near the surface of the pin.

They can be used for various applications, such as touch buttons, touch sliders, or even basic gesture recognition. They can also be used to wake the ESP32 from deep sleep, which is particularly useful in power-sensitive applications.

I2C Pins

The ESP32-WROOM-32 module has two I2C bus interfaces, but no dedicated I2C pins. Instead, it allows for flexible pin assignment, meaning any GPIO pin can be conFigured as I2C SDA (data line) and SCL (clock line).

However, GPIO21 (SDA) and GPIO22 (SCL) are commonly used as the default I2C pins to make it easier for people using existing Arduino code, libraries, and sketches.

SPI Pins

The ESP32-WROOM-32 module features three SPIs (SPI, HSPI, and VSPI). HSPI and VSPI are commonly used for general purposes, while the third one is used for interfacing with the SPI flash memory integrated on the module.

Similar to I2C, the ESP32 allows flexible pin assignment for SPI. This means that any GPIO pin can be conFigured as SPI pins.

UART Pins

The ESP32-WROOM-32 module has three UART interfaces: UART0, UART1, and UART2. These interfaces enable serial communication with various peripherals or for logging and debugging purposes.

Please note that the UART1 pins (GPIO 9 and GPIO 10) are used for interfacing with the SPI flash memory integrated on the module, so you cannot use them. However, you can still use UART1 by bit-banging the UART protocol on other GPIO pins.

Besides the basic TX and RX pins, UART interfaces on the ESP32 also support RTS (Request To Send) and CTS (Clear To Send) for hardware flow control, though these are less commonly used.

PWM Pins

Almost all GPIO pins on the module can be conFigured to generate PWM output.

The PWM on the ESP32 can be conFigured with high resolution, typically up to 16 bits, allowing for fine control over the PWM signals. PWM frequency can also be adjusted, with a typical range from a few Hz to tens of MHz, making it suitable for a wide range of applications, from controlling motors to dimming LEDs.

RTC GPIO Pins

Some GPIOs are routed to the RTC low-power subsystem and are known as RTC GPIOs. These GPIOs can be used for waking up the ESP32 from deep sleep and for interfacing with RTC peripherals.

Strapping Pins

There are five strapping pins on the ESP32: GPIO0, GPIO2, GPIO5, GPIO12, and GPIO15. The state of these pins determines whether the ESP32 enters BOOT mode (to run the program stored in flash memory) or FLASH mode (to upload a program to flash memory) upon power-up.

Be aware that if peripherals are connected to these pins, you might face issues when trying to flash the ESP32 with new firmware. This is because these peripherals could prevent the ESP32 from entering the correct mode**.**

Power Pins

The power pins provide the necessary voltage to the module to operate:

3V3 is the main supply voltage pin. It should be provided with a stable 3.3V power supply.

GND is the ground pin.

Enable Pin

The EN pin is the enable pin for the ESP32. When pulled HIGH, the chip is enabled and operational; when pulled LOW, the chip is disabled.

2.3.3 Functional Description

This chapter describes the modules and functions integrated in ESP32-WROOM-32D and ESP32-WROOM32U.
3.1 CPU and Internal Memory

2.3.3.1 CPU and Internal Memory

ESP32-D0WD contains a dual-core Xtensa® 32-bit LX6 MCU. The internal memory includes:

• 448 kB of ROM for booting and core functions.

• 520 kB of on-chip SRAM for data and instructions.

• 8 kB of SRAM in RTC, which is called RTC FAST Memory and can be used for data storage; it is accessed by the main CPU during RTC Boot from the Deep-sleep mode.

• 8 kB of SRAM in RTC, which is called RTC SLOW Memory and can be accessed by the co-processor during the Deep-sleep mode.

• 1 kbit of eFuse: 256 bits are used for the system (MAC address and chip configuration) and the remaining 768 bits are reserved for customer applications, including flash-encryption and chip-ID.

2.3.3.2 External Flash and SRAM

ESP32 supports multiple external QSPI flash and SRAM chips. More details can be found in Chapter SPI in the ESP32 Technical Reference Manual. ESP32 also supports hardware encryption/decryption based on AES to protect developers' programs and data in flash

ESP32 can access the external QSPI flash and SRAM through high-speed caches

• Up to 16 MB of external flash can be mapped into CPU instruction memory space and read-only memory space simultaneously

– When external flash is mapped into CPU instruction memory space, up to 11 MB+248 KB can be mapped at a time. Note that if more than 3 MB+248 KB are mapped, cache performance will be reduced due to speculative reads by the CPU.

– When external flash is mapped into read-only data memory space, up to 4 MB can be mapped at a time. 8-bit, 16-bit and 32-bit reads are supported.

• External SRAM can be mapped into CPU data memory space. SRAM up to 8 MB is supported and up to 4 MB can be mapped at a time. 8-bit, 16-bit and 32-bit reads and writes are supported.

Both ESP32-WROOM-32D and ESP32-WROOM-32U integrate a 4 MB of external SPI flash. The 4-MB SPI flash can be memory-mapped onto the CPU code space, supporting 8, 16 and 32-bit access. Code execution is supported. The integrated SPI flash is connected to GPIO6, GPIO7, GPIO8, GPIO9, GPIO10 and GPIO11. These six pins cannot be used as regular GPIOs

2.3.3.3 RTC and Low-Power Management

With the use of advanced power-management technologies, ESP32 can switch between different power modes. Power modes

1. Active mode: The chip radio is powered on. The chip can receive, transmit, or listen.
2. Modem-sleep mode: The CPU is operational and the clock is configurable. The Wi-Fi/Bluetooth base band and radio are disabled.
3. Light-sleep mode: The CPU is paused. The RTC memory and RTC peripherals, as well as the ULP co-processor are running. Any wake-up events (MAC, host, RTC timer, or external interrupts) will wake up the chip.
4. Deep-sleep mode: Only RTC memory and RTC peripherals are powered on. Wi-Fi and Bluetooth connection data are stored in the RTC memory. The ULP co-processor is functional.
5. Hibernation mode: The internal 8-MHz oscillator and ULP co-processor are disabled. The RTC recovery memory is powered down. Only one RTC timer on the slow clock and certain RTC GPIOs are active. The RTC timer or the RTC GPIOs can wake up the chip from the Hibernation mode.

2.3.3.4 Peripherals and Sensors

External connections can be made to any GPIO except for GPIOs in the range 6-11. These six GPIOs are connected to the module's integrated SPI flash.

2.4 FLOWMETER

Introduction

Flowmeters are devices used to measure the volumetric or mass flow rate of a fluid, such as a liquid, gas, or steam, flowing through a pipe or channel. Accurate flow measurement is crucial in various industries, including oil and gas, water treatment, manufacturing, and power generation, as it enables efficient process control, resource optimization, and regulatory compliance.

Types of Flowmeters

Flowmeters can be broadly classified into several categories based on their operating principles:

1. Positive Displacement Flowmeters: These meters measure flow by trapping a known volume of fluid and counting the number of times this volume is displaced. Examples include rotary, reciprocating, and nutating disk flowmeters.
2. Velocity Flowmeters: These meters measure the velocity of the fluid and use it to calculate the volumetric flow rate. Examples include turbine, vortex, and electromagnetic flowmeters.
3. Differential Pressure Flowmeters: These meters measure the pressure drop across a restriction in the flow path, such as an orifice plate or a Venturi tube, to determine the flow rate.
4. Mass Flowmeters: These meters directly measure the mass flow rate of a fluid, regardless of changes in temperature, pressure, or fluid properties. Examples include Coriolis and thermal mass flowmeters.
5. Ultrasonic Flowmeters: These meters use the propagation of ultrasonic waves to measure the velocity of the fluid and calculate the volumetric flow rate.

Variable Area Flowmeters: These meters use a variable orifice area to measure the flow, such as in rotameters and variable area flow tubes.

Flowmeter Performance Characteristics

1. Accuracy: The degree of closeness between the measured flow rate and the true flow rate.
2. Repeatability: The ability of the flowmeter to produce consistent readings under the same conditions.
3. Rangeability: The ratio of the maximum to the minimum flow rate that can be measured accurately.
4. Pressure Drop: The difference in pressure between the inlet and outlet of the flowmeter, which can affect the overall system performance.
5. Response Time: The time required for the flowmeter to respond to changes in the flow rate.
6. Immunity to Fluid Properties: The ability of the flowmeter to perform accurately despite changes in fluid properties such as viscosity, density, and temperature.

Factors Affecting Flowmeter Selection

1. Fluid Type: The physical and chemical properties of the fluid, such as viscosity, density, and corrosiveness.
2. Flow Rate Range: The expected minimum and maximum flow rates to be measured.
3. Process Conditions: The operating temperature, pressure, and flow profile of the fluid.
4. Installation Requirements: The available space, straight pipe lengths, and accessibility for maintenance.
5. Accuracy and Repeatability Requirements: The level of precision required for the application.
6. Cost and Maintenance Considerations: The initial cost, operating costs, and ease of maintenance.

Advances in Flowmeter Technology

Flowmeter technology has evolved significantly in recent years, with several advancements:

1. Multiphase Flow Metering: The ability to measure the individual flow rates of different phases (e.g., oil, gas, and water) in a multiphase fluid stream.

2. Wireless and Portable Flowmeters: Flowmeters that can be easily installed and moved without the need for extensive piping modifications.

3. Intelligent Flowmeters: Flowmeters with built-in diagnostics, self-calibration, and remote monitoring capabilities.

4. Miniaturized and Micro-Flowmeters: Flowmeters designed for small-scale applications, such as microfluidic devices and medical equipment.

5. Improved Accuracy and Range ability: Continuous advancements in sensor technology and signal processing algorithms to enhance flowmeter performance.

Conclusion

Flowmeters are essential instruments in various industries, and the selection of the appropriate flowmeter depends on a thorough understanding of the application requirements, fluid properties, and the latest advancements in flowmeter technology. Ongoing research and development in this field are expected to further improve the accuracy, reliability, and versatility of flowmeters, enabling more efficient and sustainable industrial processes.

Figure 2.3: YS201 FLOW SENSOR

The YFS201 water flow sensor is Hall Effect type sensor that is manufactured with good quality material and comes in black colour This Hall Effect sensor that measures water flow has a working voltage of 5 to 18/24V DC. This water flow meter uses the concept of hall effect and has a working flow rate of 1 to 30 Liters/Minute. The YFS201 water flow sensor is designed with 0.78" outer diameter,1/2"of thread and compatible with 1/2" nominal pipe connections. This water flow sensor comes in a small compact size of dimensions: 2.5"x1.4"x1.4"

The sensor comes with three wires

1. Red (5-18/24 DC power)
2. Black(ground)
3. Yellow (hall effect pulse output)

2.5 DATABASE

2.5.1. Introduction

To overcome the hurdles of database, several mechanisms have been suggested by several researchers. Systems like ISAM, B-Trees, etc. are examples of the mechanisms, which provide efficient database access when the data is very large in size and when it is to be updated frequently. All these mechanisms are actually providing support to databases by using one important technique known as Indexing. The importance and strength of the database can be viewed in two different angles by security and privacy. If the database is of good quality then only the management is ensured, thus security. For a good quality database, the basic and the first important thing is that the database is created using the best possible indexing mechanism which can be retrieved very fast in order to handle the excellent output. In order to have good quality database one has to take into consideration the possible threats beforehand. Different types of threats were enlisted in the literature which includes lack of privacy and data sharing, loss of control of the data by donor, possibility of public disclosure of confidential data, loss of control on the part of researchers, and possessed threats such as, possibility of data misuse and the privacy threats associated with patients' mobility.

The concept of database occupies the central position in the fields of computer science and technology. The database itself represents a data block that is logically related in some or the other manner. The database provides a platform upon which various formats of data can be accessed and used effectively. The Database Management System is a software, which effectively handles databases in an organized manner. The difficulty or complexity associated with data management increases as the size of the data increases. In most of the situations database falls short in handling efficiently, especially where the database is huge in size and the volume of update is large.

2.5.2. Definition of Database

Based on the above definitions, the author can conclude that the database is a collection of data, which has been thoroughly organized in a subject application area, which has the necessary order and meaningful relation, and is professionally managed using database system software. A database management system is also said to be a collection of programs that enables users to efficiently create, manage, and access data; the DBMS acts as an interface between the user and the database. The author also stated that “DBMS is the intermediate layer inserted between the operating system and the application system because most of the database-related functions are not provided by the operating system.” DBMS introduces issues such as redundancy, inconsistency, inflexibility, and time-consuming. Redundancy is the same data information appearing in different file sections.

Database and database management system (DBMS) definitions are explained. Databases are huge self-describing collections of integrated records. Databases are used to store and quickly retrieve data so that commercial database systems can be classified by size, application type, programming interface, number of users, concurrency control mechanisms, complexity, and cost. A database is a collection of interrelated data and a database management system (DBMS) is software that manages and controls both the database and the database applications that use the data stored in the database. Computer application programs are a key part of an organization’s information system and the information system consists of computer systems, important programs such as databases, and related manual systems that help management organize, store, and retrieve important information in a systematic and efficient way, so the term database turns attention to the stored data that are pertinent to an organization or its users.[19]

2.5.3 Importance of Databases in Modern Society

In these cases, and in order to provide user satisfaction, databases should have good performance, in terms of time, thanks to computer resources, and the necessary consideration of the user's interest. Unfortunately, as is well known, we do not have infinite resources regarding the computer memory that we must achieve to manage these databases; furthermore, we must take into account other interests, such as ethics and privacy, when we deal with personal interests. As we can see, the problem of managing databases is a potentially sensitive task whose importance can no longer be ignored. The aim of this work is to provide an overview of the most recent and vital robust research in the area of database management, especially from the performance point of view of the database itself, based on several parameters, including the behavior of transactions that access the database of structured designed.[17]

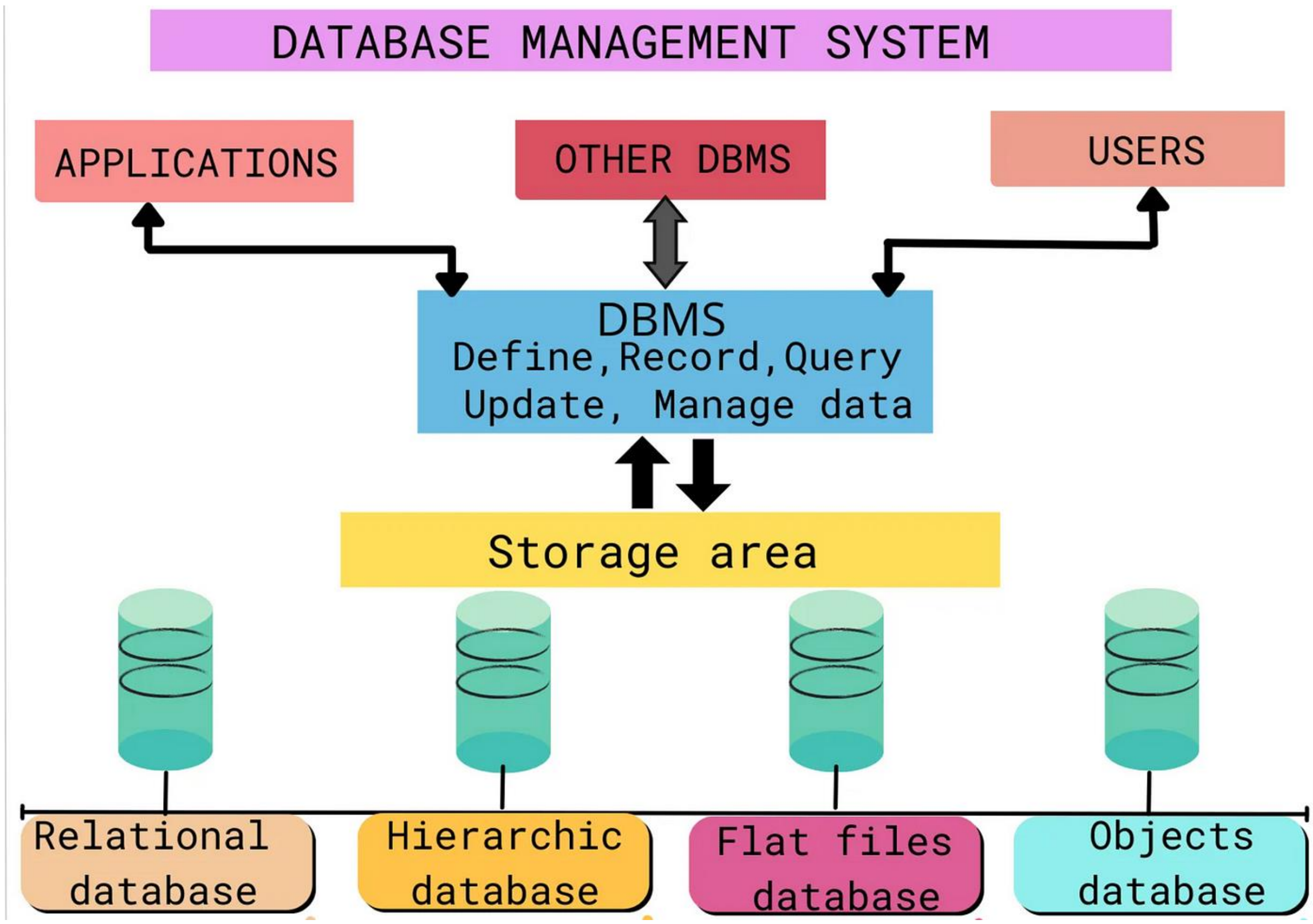


Figure 2.4 DATABASE OVERVIEW

2.6 SD CARD MODULE

The SD card was introduced by SanDisk in 1999 and is used as a portable expanded-memory medium without the cables. SD card module is accessible via pin headers (SMD) enabling users to read/write the SD card with SD card protocol. In a low wire count interface that adapts nicely to the MP3AD4 module. When in the SPI mode, the SPI serial interface holds a CS pin, SCK pin, MOSI pin, and MISO pin. Pin 4 is the SD card present (SCP for existence) signal, and pin 5 is the hardware lock (HWP for locking) its physical apperance and pin layout is shown on Figure 2.5

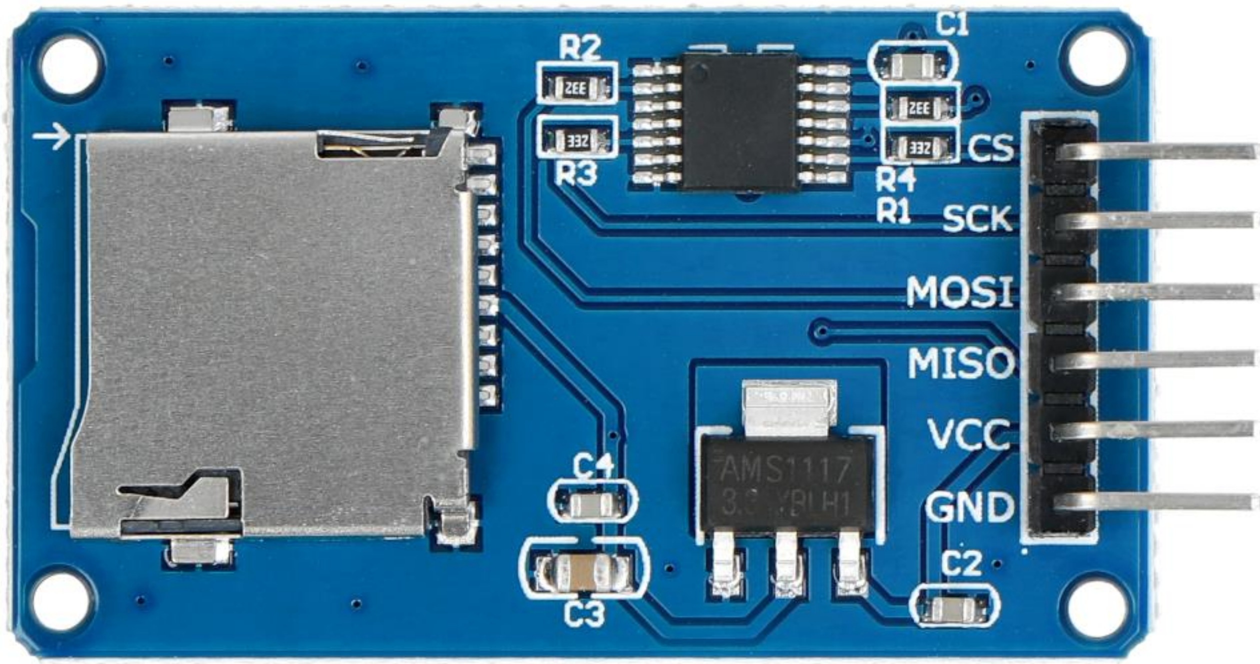


Figure 2.5 SD CARD MODULE

The SPI bus works in the SD card by forcing its bits to appear at the MICK edge – essentially by toggling the SD card's MCI pin faster than MICK, the SD card MCK edge seems to change on the data pin. The SD card is too clever than their tightest implementation of SPI – it needs to have a FIFO of a delay between MICK and data in order to manage the read and write cycles that work at different speeds. After configuration and synchronization but before the SD card's first chip select assertion, it could wait ten idle bits of SSPI – eight "dummy" bits, one for the number of dummy bytes, and a final dummy bit. Devices SD card support data transfers between an analog device and devices wherein, which only supports the SPI protocol require the simplest SD card without code and hardware optimization. When mounting very quick write operations, which are of wealth these respond to conFigure user throughput frequently need to optimize hardware and low-level code access to the SD card. Fashion profiles for modules using the manufacturer-tailored or open-tailored link mode are provided in the following segments, leveraged by fully-static nominal specification example links with a range of popular SD card module tuning options.[14]

7.1 POWER SUPPLY

The power supply may be described as an electrical equipment that provides power to electrical loads. This device's primary job is to convert the electrical current from a source to the precise voltage, frequency, and current required to supply the load. These power sources are sometimes referred to as electric power converters. Some supplies are standalone loads, but others are built within the appliances that they regulate. The power supply circuit is found in a wide range of electrical and electronic devices. Power supply circuits are grouped into several sorts based on the amount of power they provide to circuits or devices. For example, microcontroller-based circuits are typically 5V DC regulated power supply (RPS) circuits that may be created using several methods for converting power from 230V AC to 5V DC. The power supply block diagram is detailed here, as well as the step-by-step conversion of 230V AC to 12V DC.[12]

- The 230V AC is converted to 12v using a step-down converter.
- The bridge rectifier converts alternating current to direct current.
- A capacitor filters the alternating current ripples and supplies them to the voltage regulator.
- Lastly, a voltage regulator limits the voltage to 5V, and a blocking diode is utilized to capture the pulsing waveform.

CHAPTER 3 DESIGN AND METHODOLOGY

3.1 INTRODUCTION

Design research, study or essay, is a creative activity, mediated by tools and objects, that allows to conFigure knowledge using analogy, metaphor, scenario, object and narrative. The design in itself is a thought experiment. The purpose of this book is to transmit, to others, the reflection that has been carried out on the possibilities of the university, and the result is, in the end, a descriptive and prosaic essay, that is, the interpretation of a dimension of reality.[17]

3.2PROPOSED PROJECT WORKING PRINCIPLE

The Fuel Wise Project utilizes flowmeters, which are devices designed to measure the flow rate of a fluid, in this case, fuel. The flowmeters are strategically installed at key points in the fuel management system, including fuel delivery points, storage tanks, and fuel lines connected to the power plant. These flowmeters accurately measure the flow of fuel and provide real-time data.

The data collected from the flowmeters is then transmitted to a centralized data acquisition system. The data acquisition system is responsible for collecting, processing, and storing the data in a structured manner. It ensures that the fuel consumption data is readily available for analysis and decision-making purposes.

To display and visualize the fuel data, a user-friendly interface is developed. The interface provides real-time information on the fuel delivered by trucks, fuel stored in the storage facility, and fuel used by the power plant. It may include visualizations such as charts, graphs, and reports to enable users to monitor the fuel consumption trends and make informed decisions.

The fuel monitoring system is integrated with existing systems in the power plant, such as the plant management system or the SCADA (Supervisory Control and Data Acquisition) system. This integration allows for seamless Integration with Predictive Analytics: Incorporating predictive analytics techniques can enable the system to forecast future fuel consumption trends, optimize fuel usage, and detect potential issues before they occur.

3.3OVERVIEW BLOCK DIAGRAM OF THE PROPOSED SYSTEM

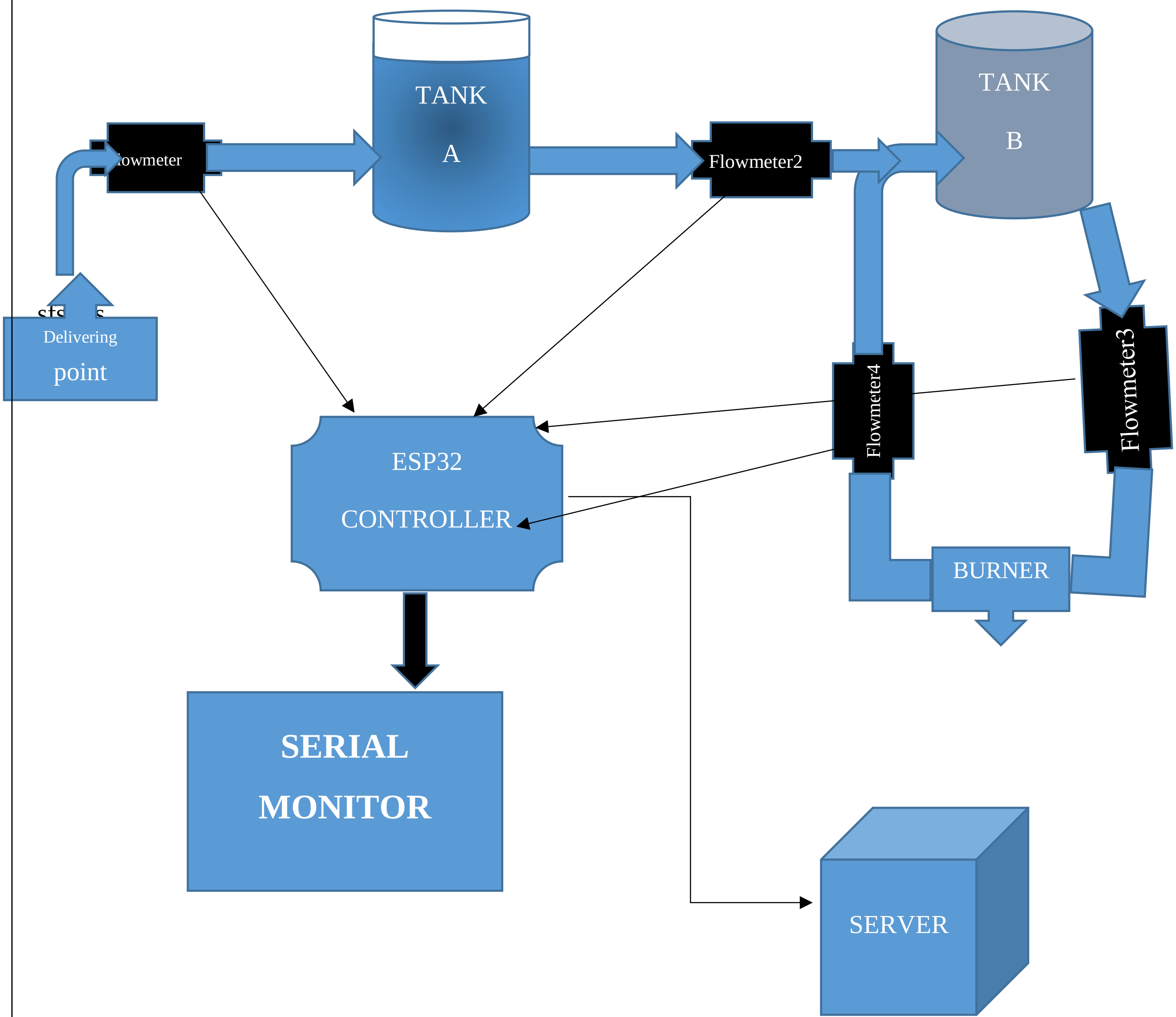


Figure 3.1 OVERVIEW BLOCK DIAGRAM OF THE PROPOSED SYSTEM

Figure 3.1 shows all the key points where flowmeters are going to be installed and the site plan of tanks as it is at ZPC Hwange Power Station. Serial monitor represent the control room where operators will be remotely monitoring power plant .the server is the storage facilities for future auditing and analysis

3.4 INSTALLING FLOWMETERS IN THE PIPE LINE FLOWCHART

Select the proper flowmeter type based on the application requirements, Determine the pipe size and material to ensure the flowmeter will fit properly, Select the flowmeter size and any additional installation requirements, Prepare the pipe location for the flowmeter installation, Install the flowmeter according to the manufacturer's guidelines,

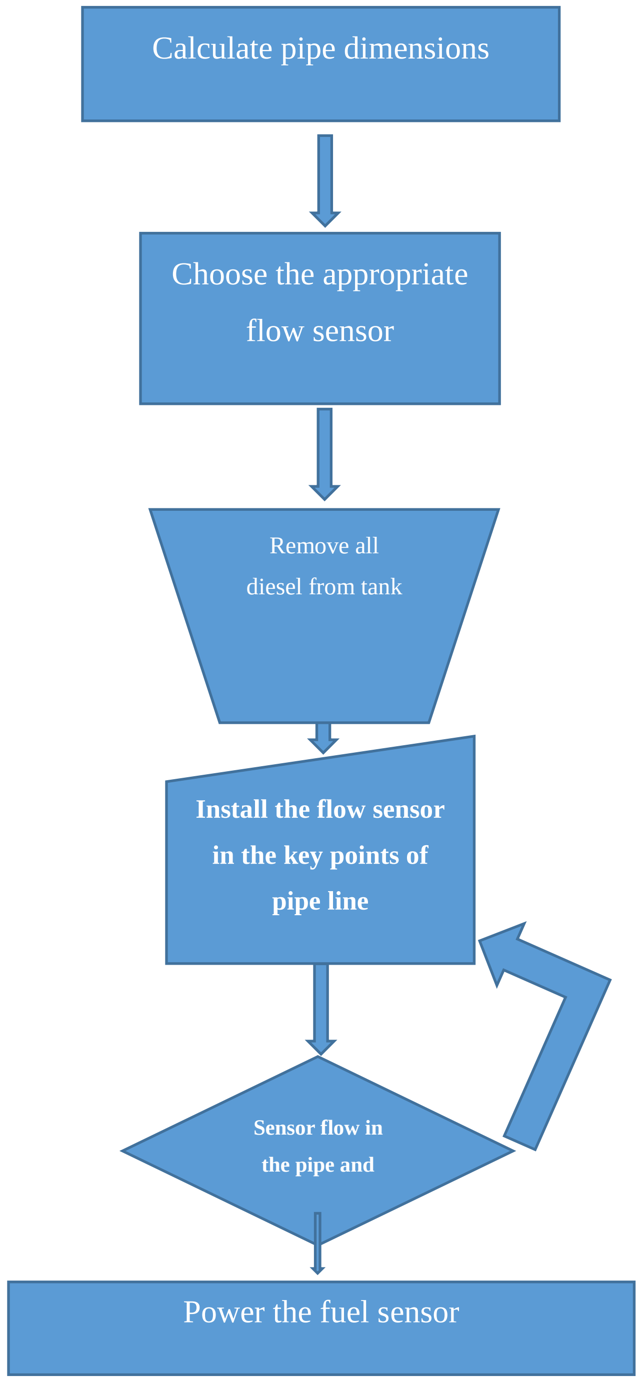


Figure 3.2: FLOWMETERS INSTALLATION

Calibrate and test the flowmeter to ensure proper operation and Document the installation details for future reference.

```
float calibrationFactor1 = 7.5;  // Calibration factor for flow meter 1 (pulses per liter)
float calibrationFactor2 = 7.5;  // Calibration factor for flow meter 2 (pulses per liter)
float calibrationFactor3 = 7.5;  // Calibration factor for flow meter 3 (pulses per liter)
float calibrationFactor4 = 7.5;  // Calibration factor for flow meter 4 (pulses per liter)

const float timeWindow = 1.0;  // Time window (in seconds) for calculating flow rate
const float flowThreshold = 0.1;  // Minimum flow rate threshold to consider the flow meter active

```

Figure 3.2a FLOWMETER CALIBRATION

Calibration on figure 3.2a is done by adjusting the calibration factor from source code of microcontroller as shown on Figure 3.2a

3.7 INTERFACING FLOWMETERS TO ESP32 CONTROLLER

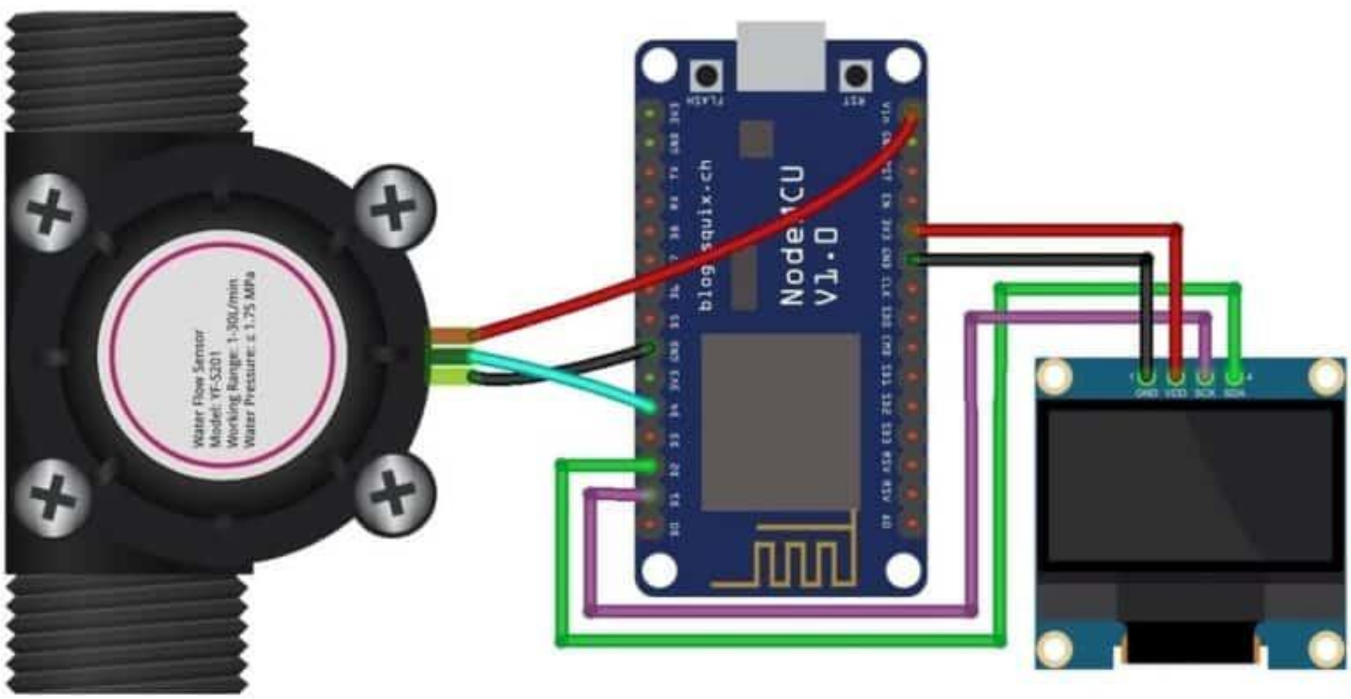


Figure 3.3 CIRCUIT SIMULATION DESIGN ON WOKWI.COM ONLINE SIMULATOR

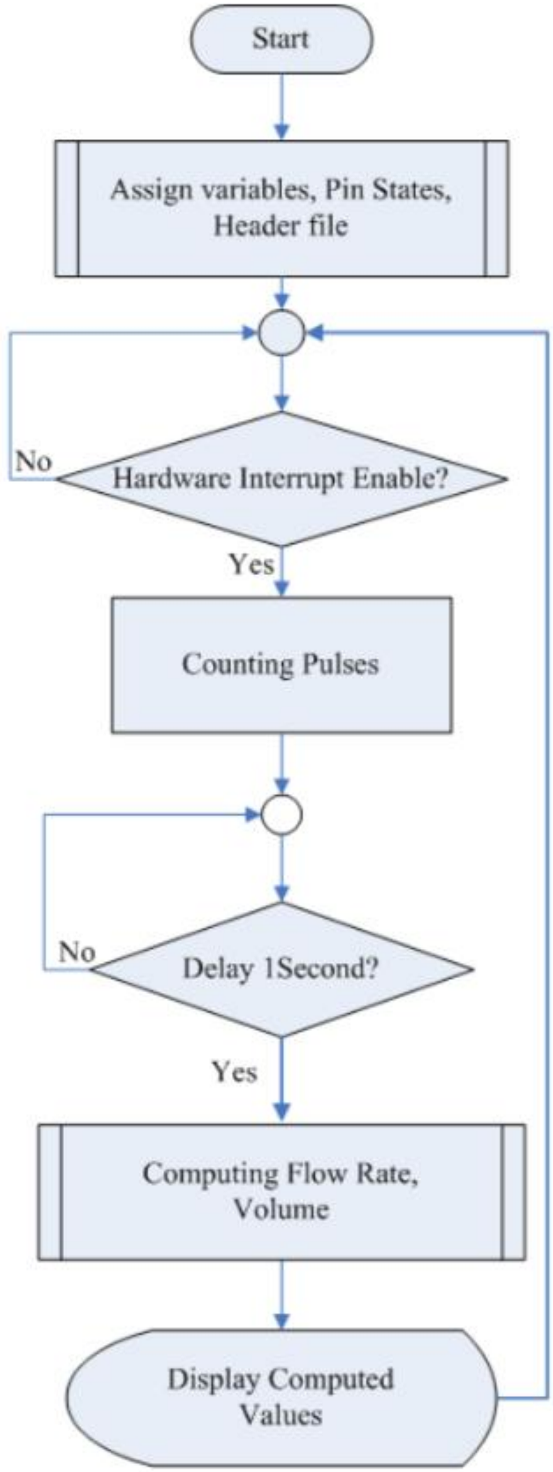


Figure 3.4: INTEGRATING FLOWMETERS TO MICROCONTROLLER AND DISPLAYING DATA ON SERIAL MONITOR

The comprehensive schematic depicted in Figure 3.4 and 3.3 provides the framework for understanding the functionality of the YF-S201 water flow sensor as designed by the author. Utilizing data extracted from its datasheet, the author ascertains the requisite equation and calibration factor. The relationship between pulses and liters is delineated as "pulse=7.5Q", where Q denotes the flow rate measured in liters per minute, thus establishing the calibration factor at 7.5.

The Hall Effect sensor within the YF-S201 emits electrical pulses proportional to the flow of water through the sensor. For every liter of water passing through the sensor in a minute, it generates 450 pulses, thereby establishing the calibration factor as [450/60 = 7.5]. Conversely, each pulse is approximately equivalent to 22.3 milliliters. Through the enumeration of electrical pulses, one can compute both the flow rate and discharged water volume from the sensor.

3.5 CALCULATING TANK LEVELS AND FUEL USED

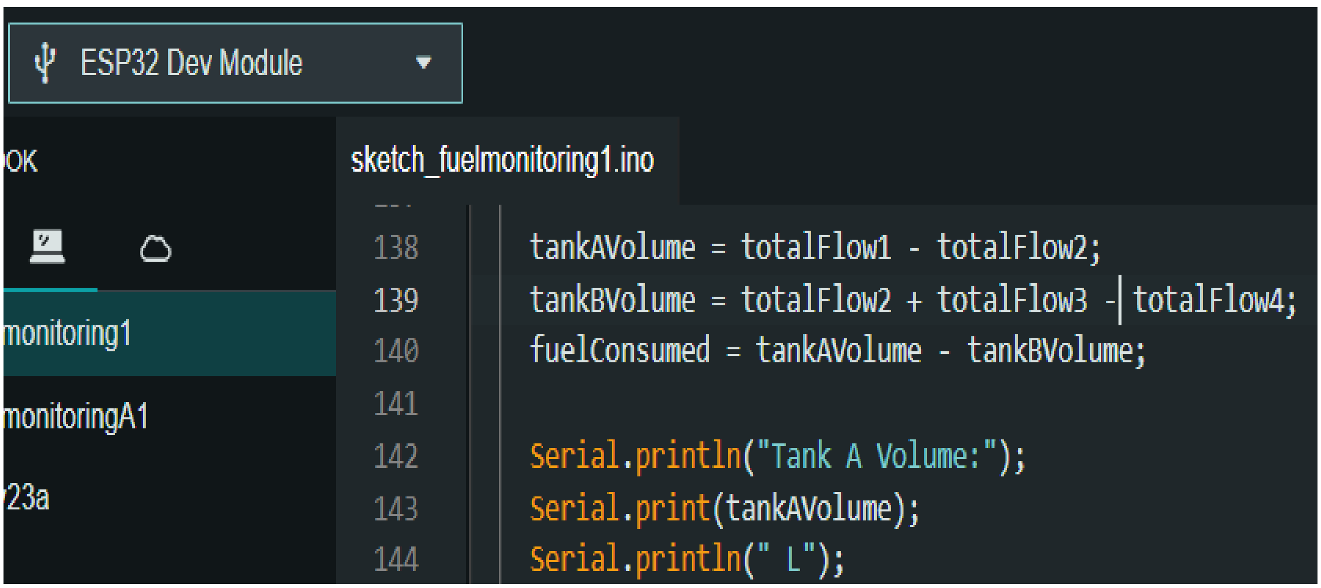


Figure 3.5 TANK LEVEL, FUEL CONSUMED CALCULATIONS

Calculations are done in the source code of microcontroller as shown on Figure 3.5 and the process outlined involves determining the amount of fuel in two tanks and the fuel consumed using readings from flowmeters in a microcontroller system

In the context of Figure 3.1, Tank 1 is positioned between Flowmeters 1 and 2. Consequently, the formula for determining the fuel level in Tank 1 is deduced by subtracting the outflow (total flow measured by Flowmeter 2) from the inflow (total flow measured by Flowmeter 1). Therefore, in engineering terms, the calculation of the fuel level in Tank 1 is based on the principle of mass balance, where the inflow rate is compared to the outflow rate to ascertain the net change in the fuel volume within the tank.

$A = Tf1 - Tf2$……………………………………………………………………………………….(1)

Where:

- A represents the fuel volume in Tank
- ATf1 denotes the total flow recorded by flowmeter 1
- Tf2 represents the total flow recorded by flowmeter 2.

Tank B, situated as the second tank between three flowmeters as illustrated in Figure 3.1, experiences a perpetual flow of fuel. This perpetual flow occurs due to its dual function as a coolant for furnace heaters. The continuous motion of fuel within Tank B is sustained by its role in dissipating heat generated by furnace operations.

$B = Tf2 + Tf3 - Tf4$……………………………………………………………………………..…(2)

$C = A - B$…………………………………………………………………………………………..(3)

Equation (2): Computes the fuel volume in Tank B, considering the continuous movement of fuel and its use as a coolant by furnace heaters. It adds the inflow from flowmeters 2 and 3 and subtracts the outflow from flowmeter 4.

Where:

- B indicates the fuel volume in Tank B
- Tf3 stands for the total flow recorded by flowmeter 3.
- Tf4 denotes the total flow recorded by flowmeter 4.

Equation (3): Determines the fuel consumed by subtracting the fuel volume in Tank B from the fuel volume in Tank A.

Where:

- C represents the fuel consumed.

3.6 DATABASE DESIGN

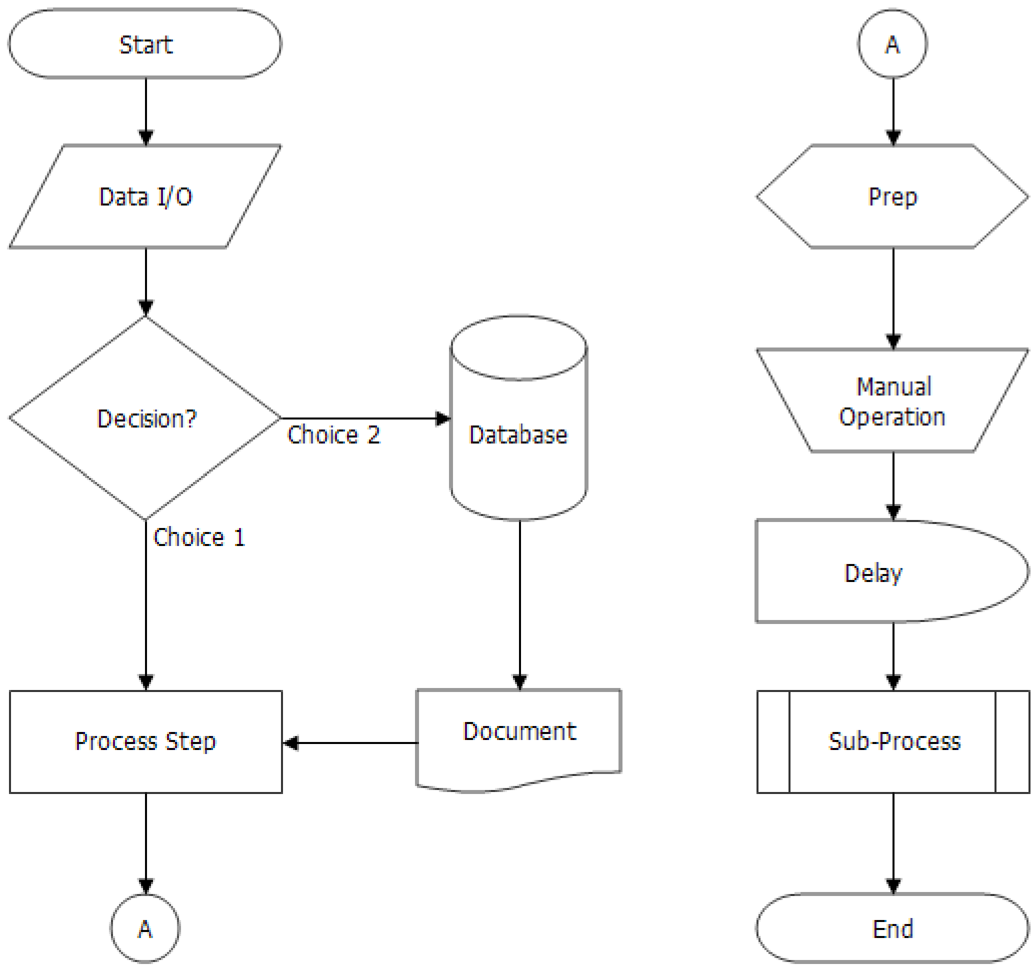


Figure 3.6 DATABASE OPERATION

Figure 3.6 illustrates the workflow of the database design, detailing the sequence of operations upon receiving input data from the ESP32 device. It delineates the steps involved in storing this data, highlighting certain manual processes necessitating human intervention, such as table copying, deletion, or sharing, which are carried out by designated personnel.

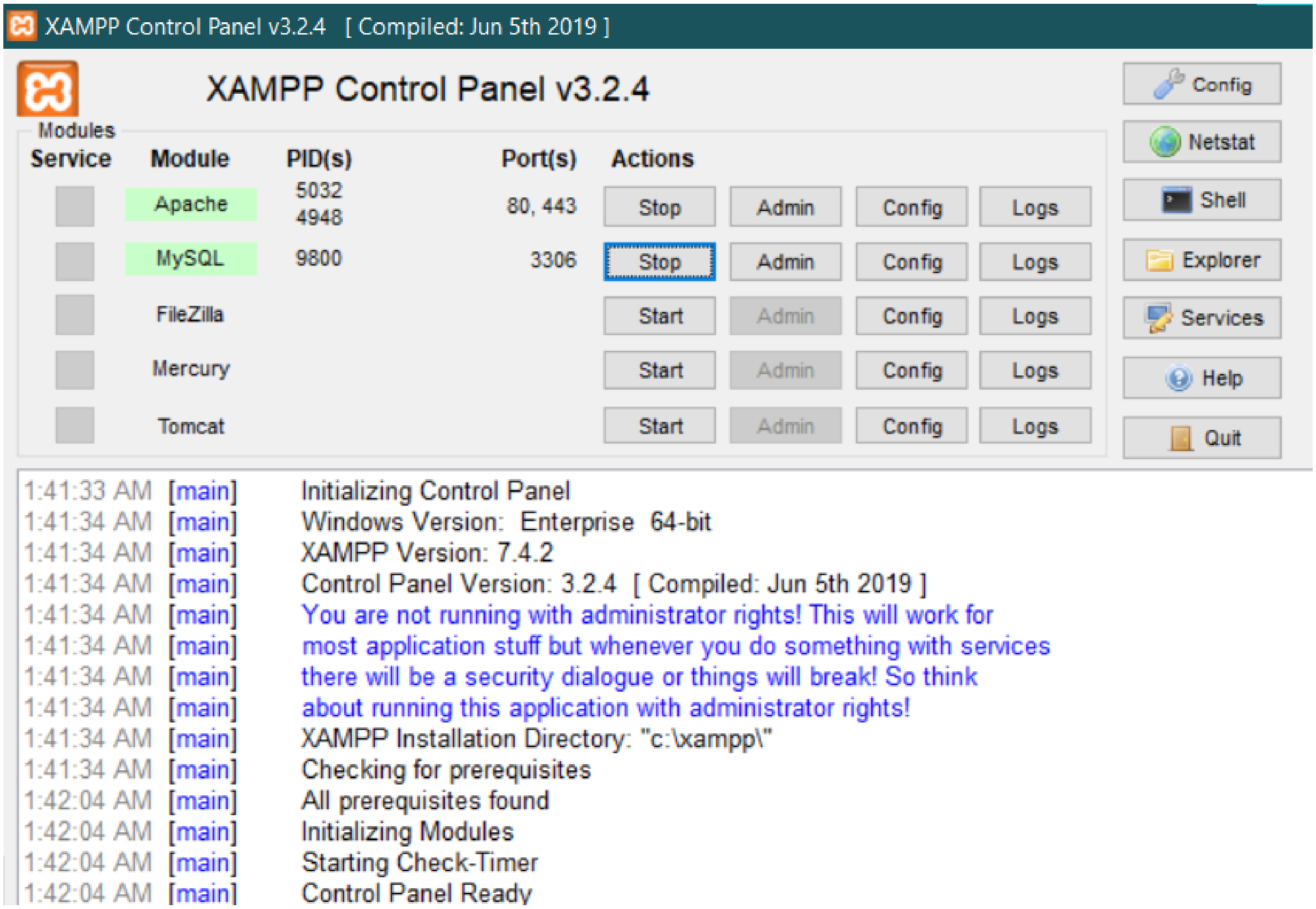
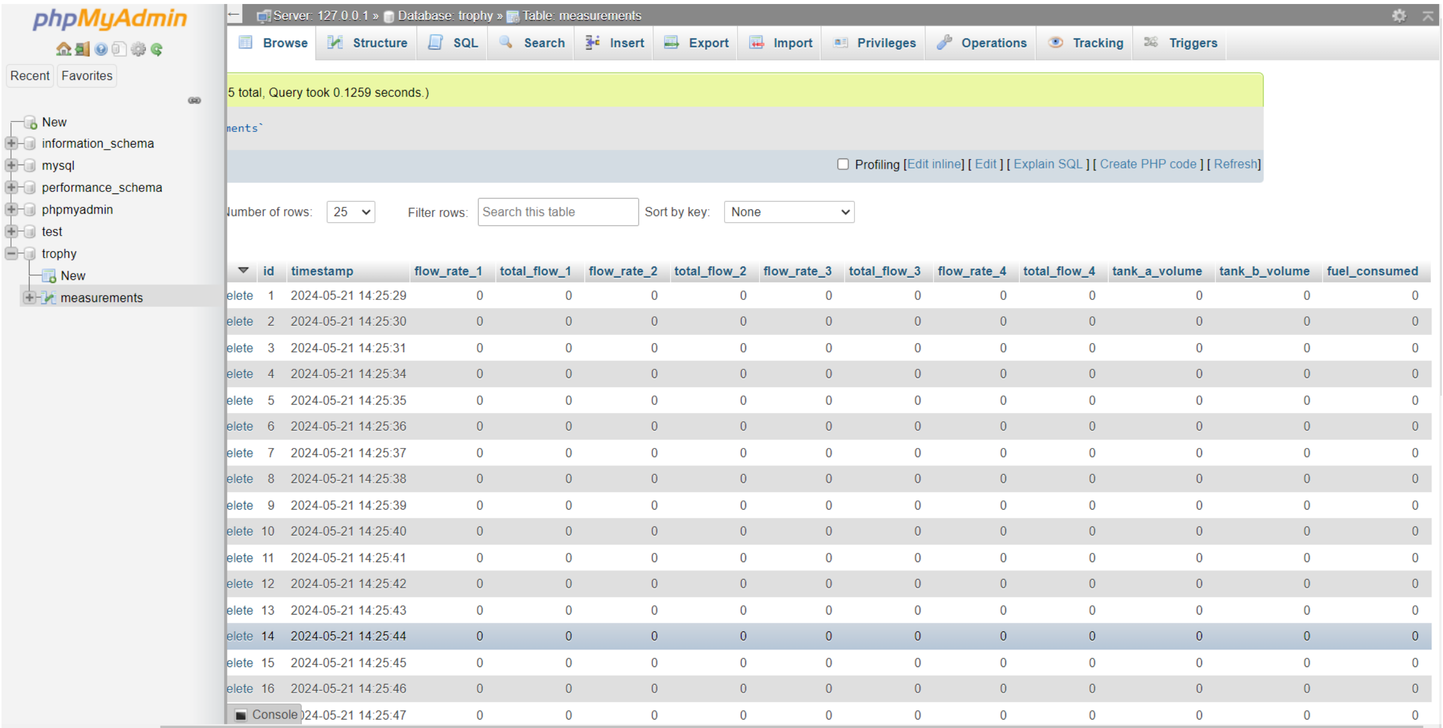


Figure3.7: STARTING APACHE AND MYSQL SERVICES

The first step in database design involves starting Apache and MySQL services as shown on Figure 3.7. Once these services are up and running, you can use tools like phpMyAdmin or MySQL Workbench to create databases, tables, and relationships. With MySQL running, you can begin by defining the structure of your database, including tables, columns, and data types. Then, establish relationships between tables using foreign key constraints to enforce referential integrity. This step lays the foundation for storing and organizing data effectively within the database. Once the database schema is designed, you can populate it with data and begin testing and refining your database design as needed.



	id	timestamp	flow_rate_1	total_flow_1	flow_rate_2	total_flow_2	flow_rate_3	total_flow_3	flow_rate_4	total_flow_4	tank_a_volume	tank_b_volume	fuel_consumed
elete	1	2024-05-21 14:25:29	0	0	0	0	0	0	0	0	0	0	0
elete	2	2024-05-21 14:25:30	0	0	0	0	0	0	0	0	0	0	0
elete	3	2024-05-21 14:25:31	0	0	0	0	0	0	0	0	0	0	0
elete	4	2024-05-21 14:25:34	0	0	0	0	0	0	0	0	0	0	0
elete	5	2024-05-21 14:25:35	0	0	0	0	0	0	0	0	0	0	0
elete	6	2024-05-21 14:25:36	0	0	0	0	0	0	0	0	0	0	0
elete	7	2024-05-21 14:25:37	0	0	0	0	0	0	0	0	0	0	0
elete	8	2024-05-21 14:25:38	0	0	0	0	0	0	0	0	0	0	0
elete	9	2024-05-21 14:25:39	0	0	0	0	0	0	0	0	0	0	0
elete	10	2024-05-21 14:25:40	0	0	0	0	0	0	0	0	0	0	0
elete	11	2024-05-21 14:25:41	0	0	0	0	0	0	0	0	0	0	0
elete	12	2024-05-21 14:25:42	0	0	0	0	0	0	0	0	0	0	0
elete	13	2024-05-21 14:25:43	0	0	0	0	0	0	0	0	0	0	0
elete	14	2024-05-21 14:25:44	0	0	0	0	0	0	0	0	0	0	0
elete	15	2024-05-21 14:25:45	0	0	0	0	0	0	0	0	0	0	0
elete	16	2024-05-21 14:25:46	0	0	0	0	0	0	0	0	0	0	0
		)24-05-21 14:25:47	0	0	0	0	0	0	0	0	0	0	0

Figure 3.8 DATABASE TROPHY WITH TABLE MEASUREMENTS

Figure 3.8 displays the "measurements" table within the "trophy" database, designed to store data transmitted from the ESP32 controller via PHP. The table structure encompasses fields tailored to capture specific metrics pertinent to the controller's operations. These fields include an auto-incrementing "id" serving as the primary key for unique identification of each record and a "timestamp" field set to automatically capture the time of data reception. Additionally, the table comprises fields like "flow_rate_1" to "flow_rate_4" and "total_flow_1" to "total_flow_4" intended to record flow rates and total flow values from multiple sources. Moreover, parameters like "tank a volume" and "tank b volume" are included to monitor the volumes of different tanks, while "fuel consumed" tracks the consumption of fuel. This comprehensive structure facilitates the systematic storage of data transmitted by the ESP32, enabling subsequent retrieval and analysis for various applications.

```
174 void sendDataToServer(float flowRate1, float totalFlow1, float flowRate2, float totalFlow2, float flowRate3, float totalFlow3, float flowRate4, float totalF
175 {
176   HTTPClient http;
177 
178   // Build the data payload
179   String data = "flowRate1=" + String(flowRate1) +
180                 "&totalFlow1=" + String(totalFlow1) +
181                 "&flowRate2=" + String(flowRate2) +
```

Figure 3.9 ESP32 SEND DATA TO SERVER

The function `sendDataToServer` on Figure 3.9 line 78 encapsulates the process of transmitting data from an ESP32 controller to a server using HTTP. It takes as parameters various measurements collected by sensors or devices connected to the controller, such as flow rates, total flows, tank volumes, and fuel consumption. The parameters pass as arguments to the function.

Within the function, an instance of the HTTPClient class is created to facilitate communication with the server. Construct the data payload as a string, incorporating the measured values formatted as key-value pairs. Each key corresponds to a specific parameter, while the values are obtained by converting the respective float variables to strings using the `String () ` function.

This payload construction adheres to the conventions of HTTP requests, facilitating data transmission in a structured manner. By conveying the measurements in this format, the ESP32 controller communicates effectively with the server, enabling the storage and analysis of the collected data for various scientific and engineering applications.

3.7 DATA ACQUISITION FROM THE ESP32 CONTROLLER INTO THE DATABASE

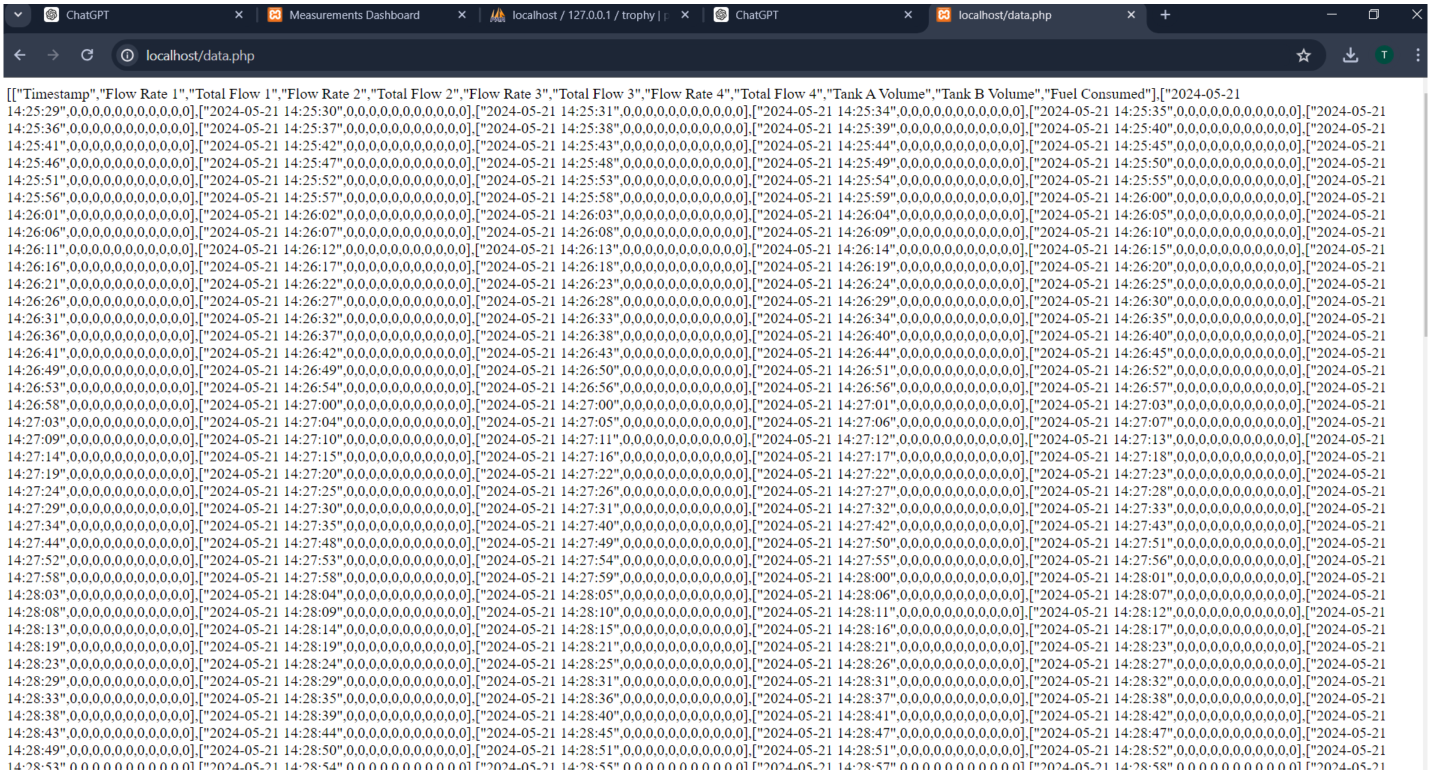
```php
// Database configuration
$servername = "localhost"; // Change this to your database server name if it's different
$username = "root";     // Your MySQL username
$password = "";     // Your MySQL password
$database = "trophy"; // Your database name

// Create connection
$conn = new mysqli($servername, $username, $password, $database);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Get data from ESP32 POST request
$flowRate1 = $_POST["flowRate1"];
$totalFlow1 = $_POST["totalFlow1"];
$flowRate2 = $_POST["flowRate2"];
$totalFlow2 = $_POST["totalFlow2"];
$flowRate3 = $_POST["flowRate3"];
$totalFlow3 = $_POST["totalFlow3"];
$flowRate4 = $_POST["flowRate4"];
$totalFlow4 = $_POST["totalFlow4"];
$tankAVolume = $_POST["tankAVolume"];
$tankBVolume = $_POST["tankBVolume"];
$fuelConsumed = $_POST["fuelConsumed"];

// Insert data into database
$sql = "INSERT INTO measurements (flow_rate_1, total_flow_1, flow_rate_2, total_flow_2, flow_rate_3, total_flow_3, flow_rate_4, total_flow_4, tank_a_volume, tank_b_volu
        VALUES ('$flowRate1', '$totalFlow1', '$flowRate2', '$totalFlow2', '$flowRate3', '$totalFlow3', '$flowRate4', '$totalFlow4', '$tankAVolume', '$tankBVolume', '$fu

if ($conn->query($sql) === TRUE) {
    echo "Data inserted successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
```

Figure 3.10 ESP32 AND DATABASE CONNECTION

This PHP script on Figure 3.10 establishes a connection to a MySQL database, retrieving data sent via an ESP32 POST request. It then constructs an SQL query to insert the received measurements into the "measurements" table. Upon successful insertion, it outputs a confirmation message; otherwise, it reports any encountered errors. Finally, close the database connection to ensure proper resource management. This process facilitates the seamless integration of sensor data acquisition from the ESP32 controller into the database for subsequent analysis and utilization in engineering applications.

3.8 DESIGN USER-FRIENDLY VISUALISATION TOOLS TO PRESENT FUEL CONSUMPTION TRENDS AND HISTORICAL DATA

The objective of this project is to develop user-centric visualization tools for presenting intricate fuel consumption trends and historical data, utilizing a synergy of HTML and PHP components. The integration of HTML's structural mark up with PHP's server-side scripting capabilities will construct dynamic interfaces to facilitate seamless user exploration. Advanced data querying mechanisms encapsulated within PHP scripts such as data.php and database.php, along with efficient visualization rendering techniques facilitated by visualization .php and charts.js, will provide users with an intuitive platform for analysing multifaceted fuel consumption metrics. Additionally, the strategic incorporation of Google Analytics aims to provide comprehensive insights into user behaviour, enabling iterative improvements and informed decision-making in the evolution of the visualization toolset. The project aims to provide stakeholders with a sophisticated toolset based on scientific principles that underpin data visualization and software engineering methodologies.



[["Timestamp","Flow Rate 1","Total Flow 1","Flow Rate 2","Total Flow 2","Flow Rate 3","Total Flow 3","Flow Rate 4","Total Flow 4","Tank A Volume","Tank B Volume","Fuel Consumed"],["2024-05-21
14:25:29",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:25:30",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:25:31",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:25:34",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:25:35",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:25:36",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:25:37",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:25:38",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:25:39",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:25:40",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:25:41",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:25:42",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:25:43",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:25:44",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:25:45",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:25:46",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:25:47",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:25:48",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:25:49",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:25:50",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:25:51",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:25:52",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:25:53",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:25:54",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:25:55",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:25:56",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:25:57",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:25:58",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:25:59",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:00",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:26:01",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:02",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:03",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:04",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:05",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:26:06",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:07",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:08",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:09",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:10",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:26:11",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:12",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:13",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:14",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:15",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:26:16",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:17",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:18",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:19",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:20",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:26:21",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:22",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:23",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:24",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:25",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:26:26",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:27",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:28",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:29",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:30",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:26:31",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:32",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:33",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:34",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:35",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:26:36",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:37",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:38",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:40",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:40",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:26:41",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:42",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:43",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:44",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:45",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:26:49",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:49",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:50",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:51",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:52",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:26:53",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:54",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:56",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:56",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:26:57",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:26:58",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:00",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:00",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:01",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:03",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:27:03",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:04",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:05",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:06",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:07",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:27:09",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:10",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:11",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:12",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:13",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:27:14",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:15",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:16",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:17",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:18",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:27:19",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:20",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:22",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:22",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:23",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:27:24",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:25",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:26",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:27",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:28",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:27:29",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:30",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:31",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:32",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:33",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:27:34",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:35",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:40",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:42",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:43",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:27:44",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:48",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:49",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:50",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:51",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:27:52",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:53",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:54",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:55",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:56",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:27:58",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:58",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:27:59",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:00",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:01",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:28:03",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:04",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:05",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:06",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:07",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:28:08",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:09",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:10",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:11",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:12",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:28:13",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:14",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:15",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:16",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:17",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:28:19",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:19",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:21",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:21",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:23",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:28:23",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:24",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:25",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:26",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:27",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:28:29",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:29",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:31",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:31",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:32",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:28:33",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:35",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:36",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:37",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:38",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:28:38",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:39",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:40",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:41",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:42",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:28:43",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:44",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:45",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:47",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:47",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:28:49",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:50",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:51",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:51",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:52",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21
14:28:53",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:54",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:55",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:57",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21 14:28:58",0,0,0,0,0,0,0,0,0,0,0],["2024-05-21

Figure 3.11 PHP DATA EXTRACCTED FROM DATABASE

Figure 3.11 showcases the dataset extracted from the database, poised for transmission to the web server for the creation of user-friendly visualizations. This crucial juncture embodies a scientific approach to data management, where a bespoke extraction tool in PHP orchestrates the retrieval process with precision and efficiency. Leveraging advanced querying mechanisms and data manipulation functionalities, this extraction tool ensures the fidelity and integrity of the dataset, laying a robust foundation for subsequent analyses. Through meticulous analytics, anomalies are identified and rectified, optimizing the dataset's coherence and reliability. Objectively structured to align seamlessly with the visualization framework, the extracted dataset undergoes pre-processing to streamline its format and enhance interoperability, thus facilitating seamless integration with the web server. The dataset encapsulates a myriad of fuel consumption metrics, spanning temporal trends and categorical distributions, offering stakeholders a comprehensive panorama of consumption dynamics. As the dataset embarks on its journey to the web server, it heralds a transformative era of intuitive visualization experiences, empowering stakeholders to derive actionable insights and make informed decisions in fuel consumption management endeavours.

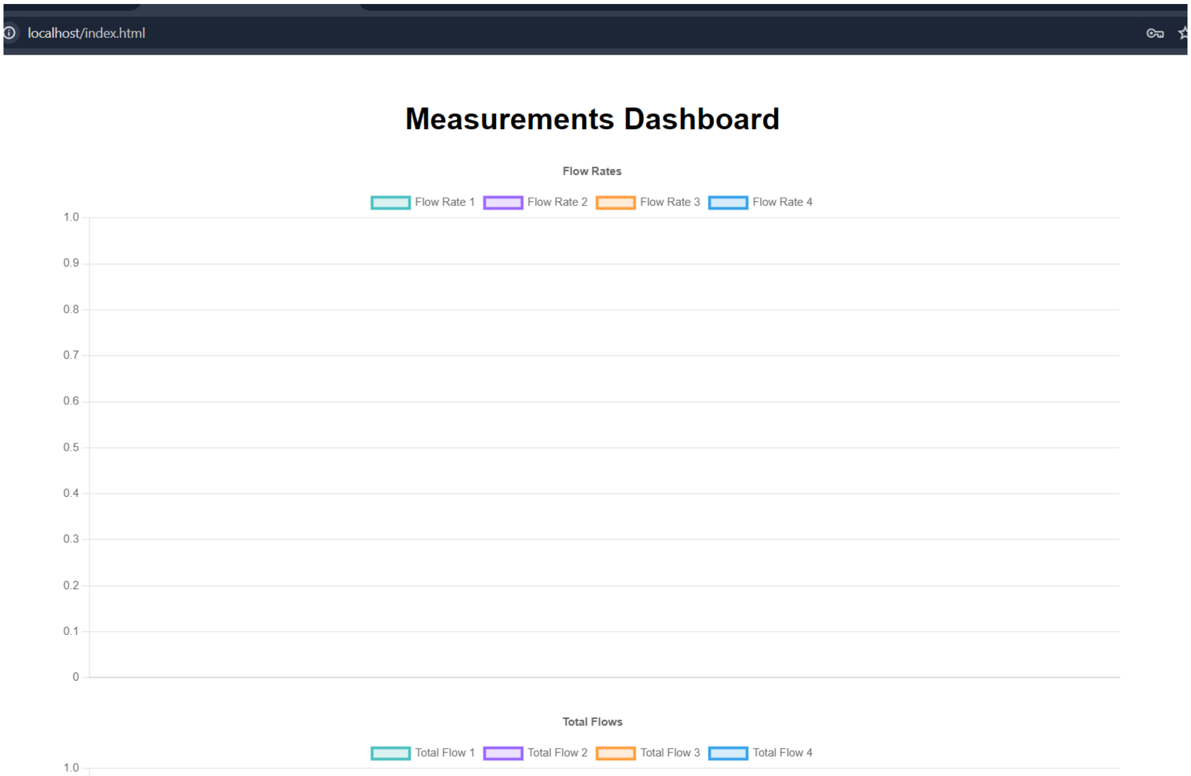


Figure 3.12 WEBSERVER

Figure 3.12 illustrates the HTML-based web server, interfacing seamlessly with PHP, as depicted in Figure 3.11, to generate charts dynamically. This integration optimizes data flow and processing efficiency, enhancing the user experience. Leveraging engineering principles, the web server harnesses real-time data transmission capabilities, facilitating agile visualization updates. The resultant charts clearly and precisely portray intricate fuel consumption trends, empowering stakeholders with actionable insights.

```
3   <head>
4       <title>Measurements Dashboard</title>
5       <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
6       <!-- Google Analytics Tracking Code -->
7       <script async src="https://www.googletagmanager.com/gtag/js?id=UA-XXXXXXXXX-X"></script>
8       <script>
```

FIGURE 3.13 WEBSERVER TO GOOGLE ANALYTIC CONNECTION

In Figure 3.13, line 7 establishes connectivity to Google Analytics, enabling the generation of trend reports.

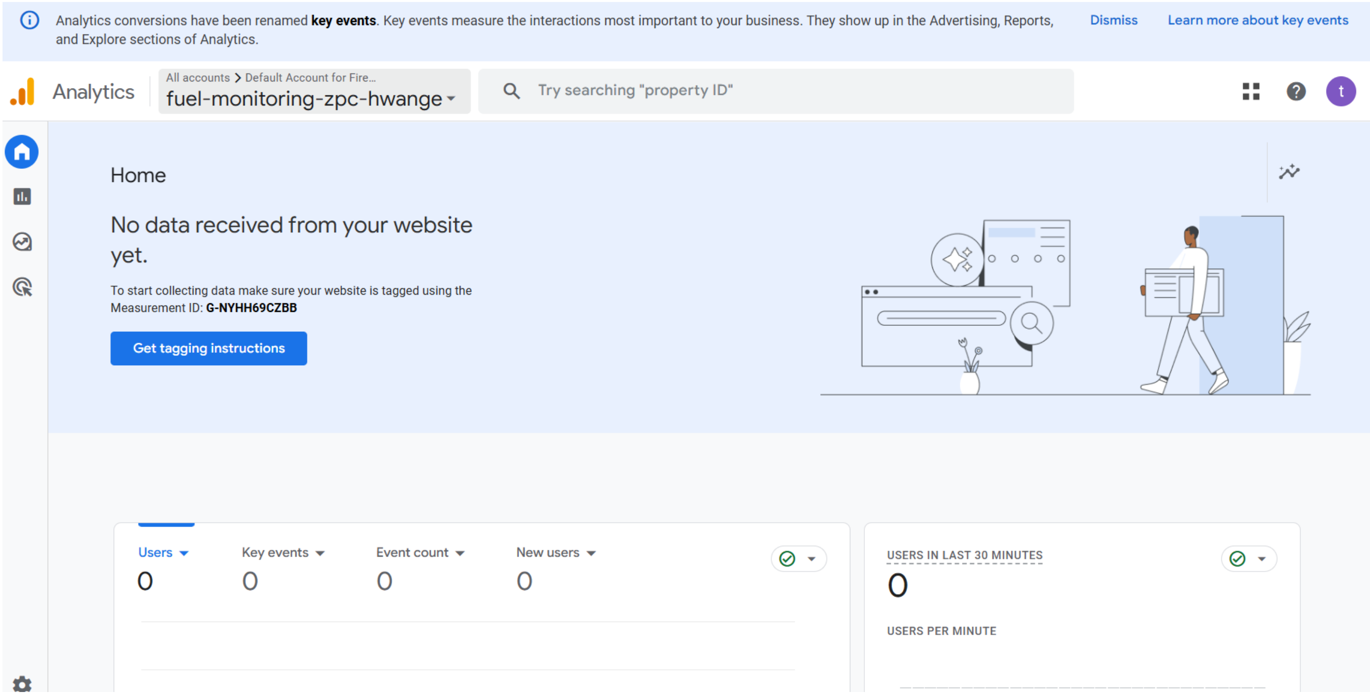


Figure 3.14 DEPICTS THE BESPOKE ANALYTICS PLATFORM ENGINEERED BY GOOGLE

Figure 3.14 depicts the bespoke analytics platform engineered by Google, tailored to efficiently process and analyse data for insightful trend analysis. This system optimizes data handling and facilitates informed decision-making, enhancing operational efficiency.

All objectives outlined in chapter one have been achieved with precision, paving the way for the forthcoming chapter focused on implementation and results. The design endeavours have culminated in comprehensive solutions, meeting the prescribed criteria with precision. The subsequent chapter will delve into the practical execution and outcomes, leveraging the foundational achievements as a springboard for analysis. This progression marks a significant milestone in the engineering process, transitioning from conceptualization to tangible outcomes. With the groundwork firmly established, attention now turns to the tangible manifestation and evaluation of the designed solutions.

CHAPTER 4: RESULTS AND ANALYSIS

4.1 INTRODUCTION

This chapter delves into the outcomes and analyses derived from the pursuit of our objectives aimed at addressing the critical issue faced by the Zimbabwe Power Company Hwange Power Station the lack of a robust and accurate real-time fuel monitoring system.

4.2 INSTALLING OF FLOWMETERS IN THE PIPELINE

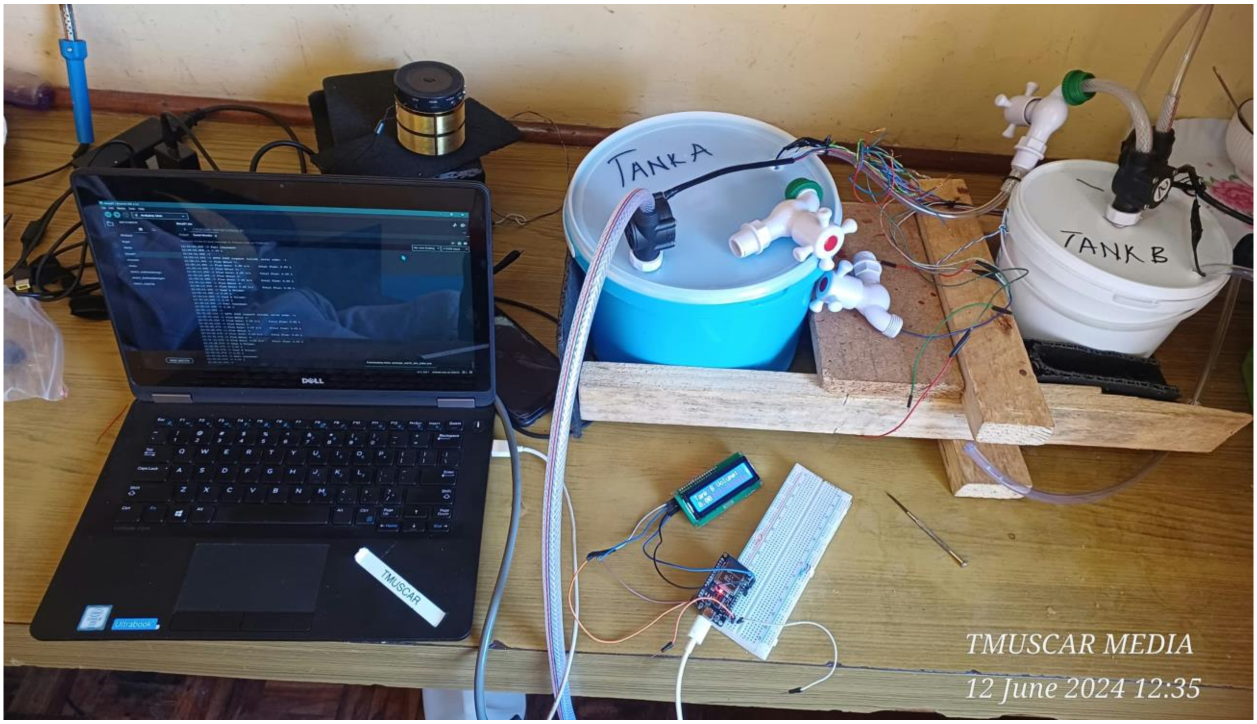


Figure 4.1 OVERALL SYSYTEM WITH INSTALLED FLOWMETERS

Figure 4.1 depicts the proposed system design featuring two tanks, four flowmeters, onsite LCD readings, a serial monitor for the operation room, and pumps for fuel transferring.

4.2 DISPLAYING DATA ON LCD AND SERIAL MONITOR

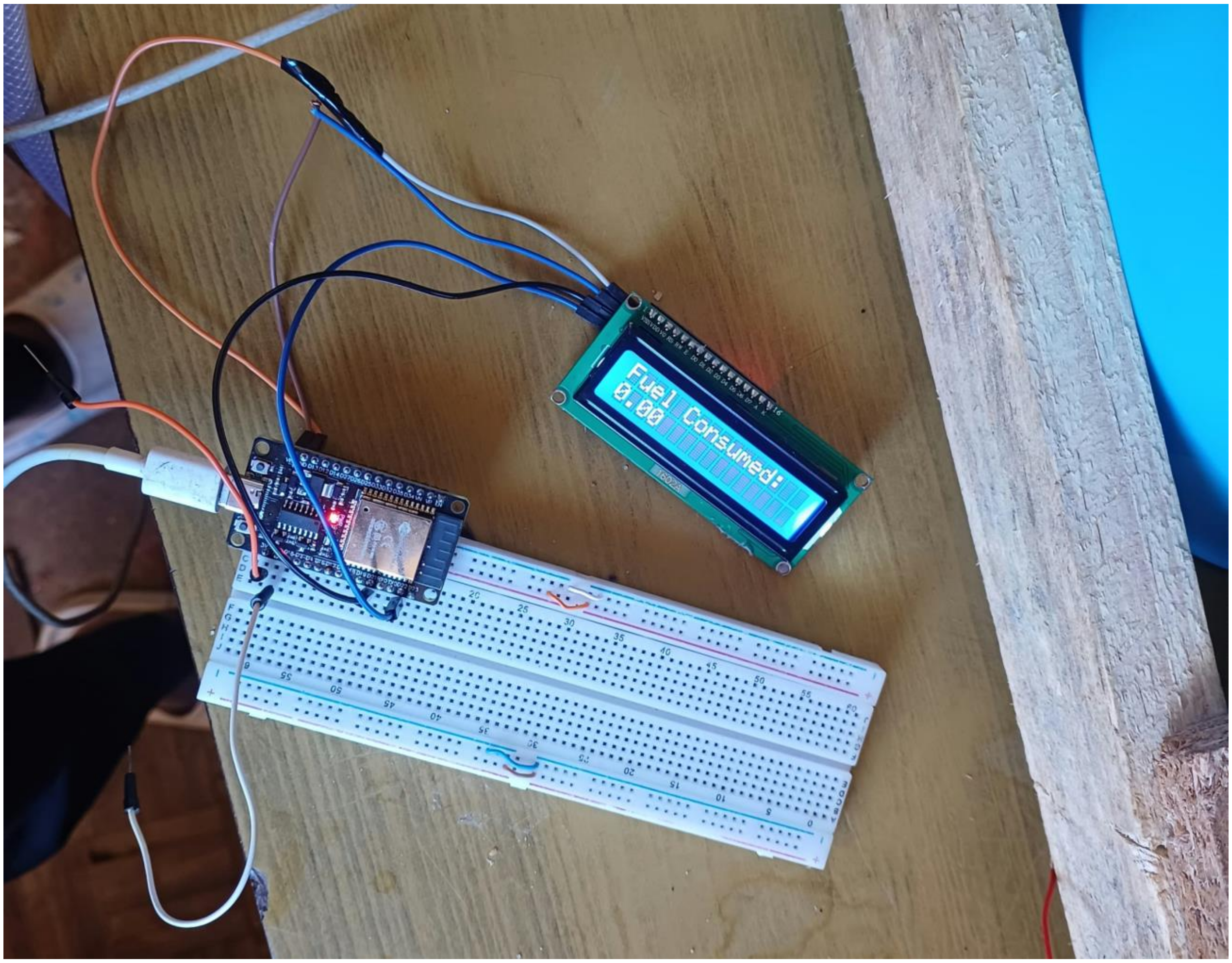


Figure 4.2 LCD DISPLAYING TANK LEVELS

Figure 4.2 illustrates an LCD displaying calculated volumes computed by the ESP32 controller. It will continually show the volume of tank A, tank B, and the amount of fuel consumed.

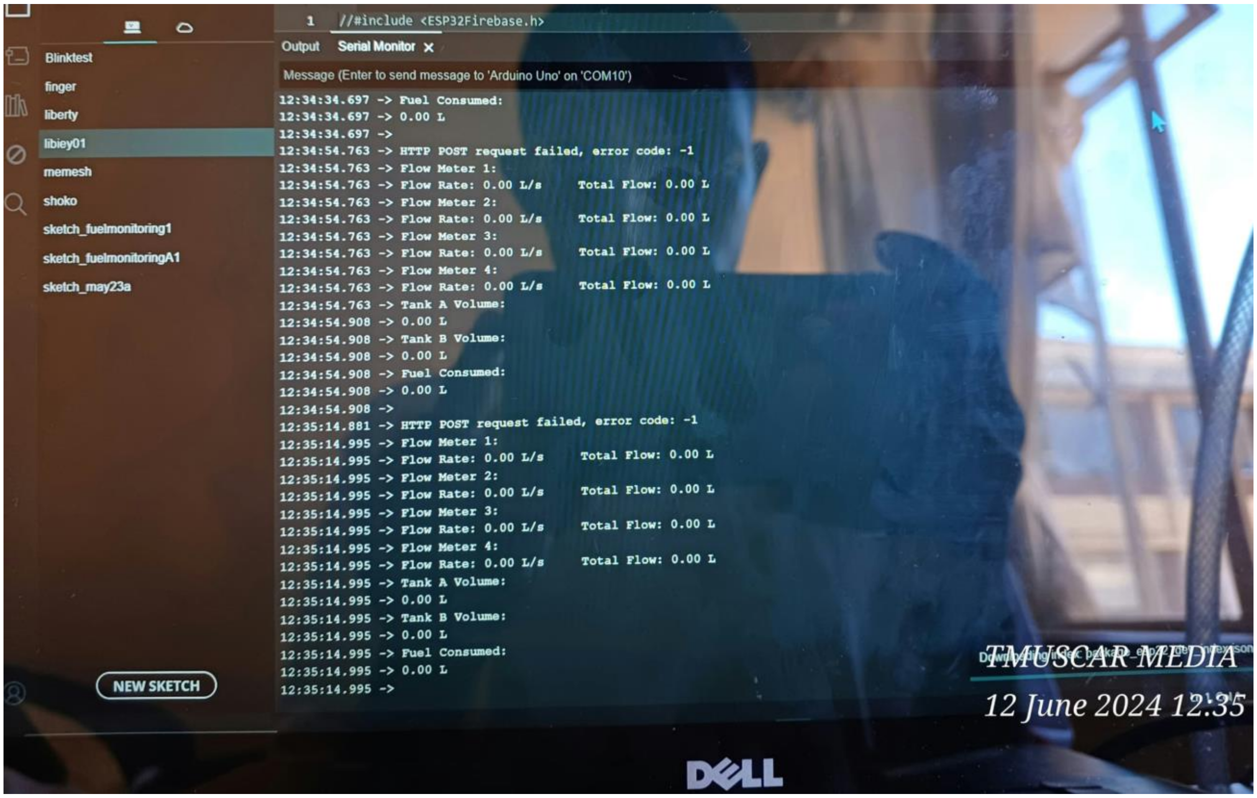
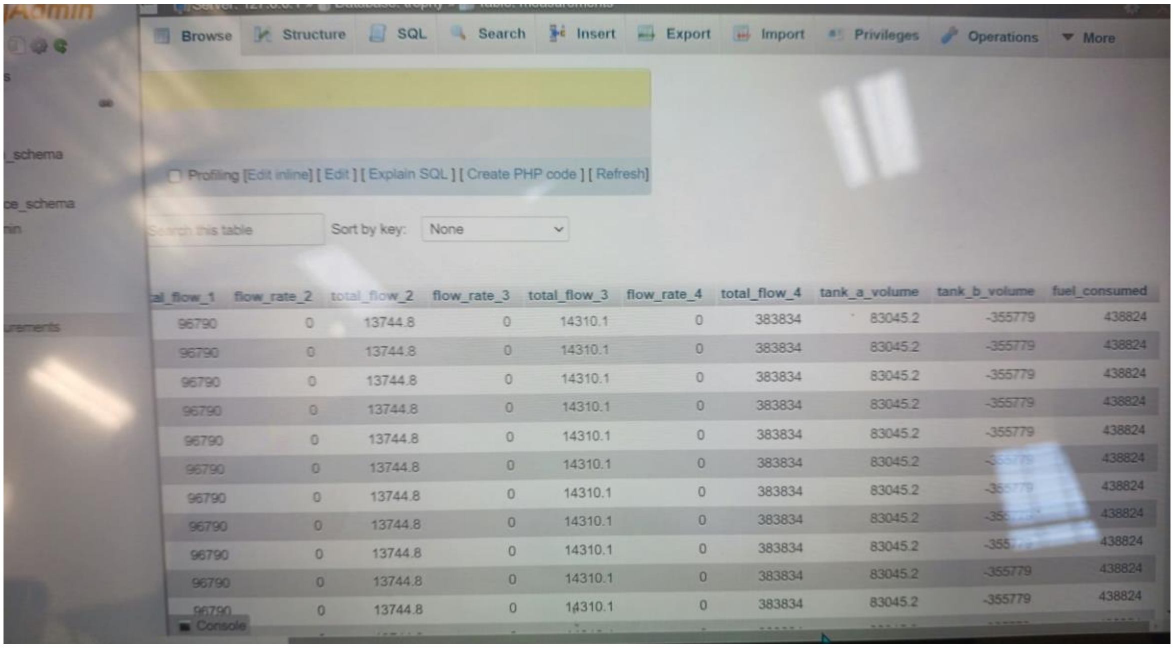


Figure 4.3 SERIAL MONITOR DISPLAY IN THE OPERATION ROOM

Figure 4.2 showcases a serial monitor stationed in the operations room, exhibiting calculated volumes computed by the ESP32 controller. This real-time display continuously presents the volumes of tank A and tank B, along with the consumed fuel quantity. Additionally, it provides the current flow rate of each flowmeter and the cumulative total flow across all flowmeters.

4.3 DATABASE RESULTS



al_flow_1	flow_rate_2	total_flow_2	flow_rate_3	total_flow_3	flow_rate_4	total_flow_4	tank_a_volume	tank_b_volume	fuel_consumed
96790	0	13744.8	0	14310.1	0	383834	83045.2	-355779	438824
96790	0	13744.8	0	14310.1	0	383834	83045.2	-355779	438824
96790	0	13744.8	0	14310.1	0	383834	83045.2	-355779	438824
96790	0	13744.8	0	14310.1	0	383834	83045.2	-355779	438824
96790	0	13744.8	0	14310.1	0	383834	83045.2	-355779	438824
96790	0	13744.8	0	14310.1	0	383834	83045.2	[illegible]	438824
96790	0	13744.8	0	14310.1	0	383834	83045.2	[illegible]	438824
96790	0	13744.8	0	14310.1	0	383834	83045.2	[illegible]	438824
96790	0	13744.8	0	14310.1	0	383834	83045.2	[illegible]	438824
96790	0	13744.8	0	14310.1	0	383834	83045.2	-355779	438824
96790	0	13744.8	0	14310.1	0	383834	83045.2	-355779	438824

Figure 4.4 DATABASE TABLE

Figure 4.4 displays data transmitted via HTTP from the ESP32 controller, which is then stored in the Trophy database's 'Measurements' table. This table includes timestamps and records four flow rates, total flows, tank A level, tank B level, and fuel consumed, as stipulated by Objective 3.

4.4 DATA VISUALISATION ON WEBSITE RESULTS

The following Figures are results for objective 4

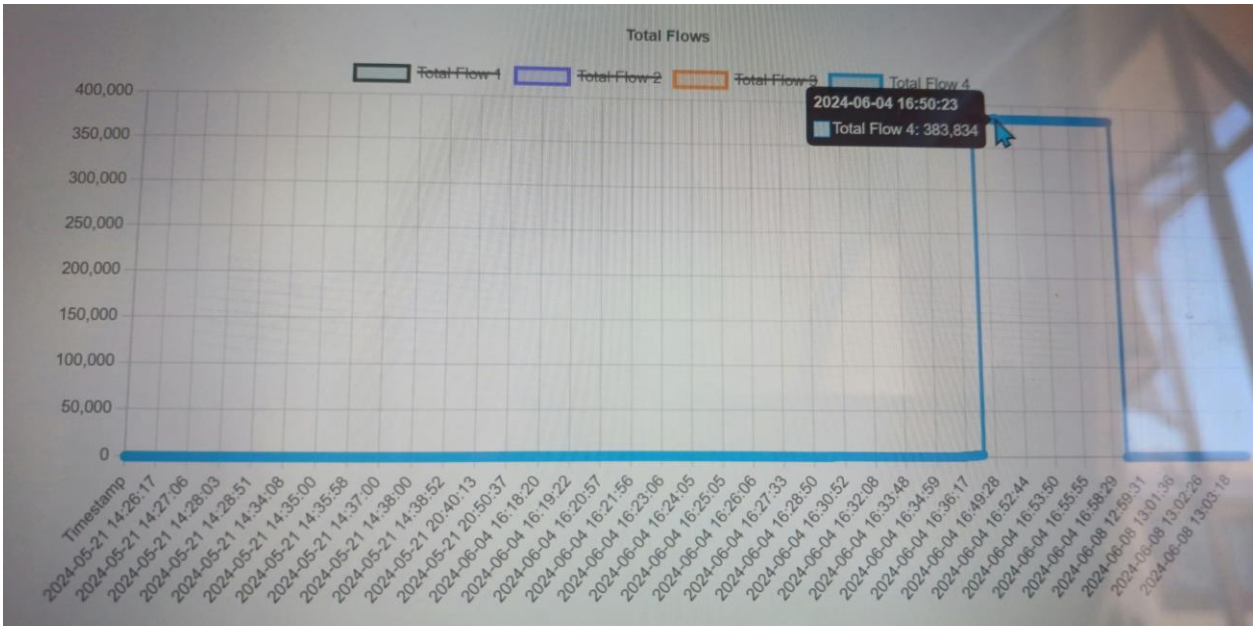


Figure 4.5 TOTAL FLOWS VISUALISATION

Figure 4.5 shows a line graph representing the trends of total flow 4 available on the website. This visualization is instrumental for conducting data analysis, as specified in the objectives.

Figure 4.6 FUEL CONSUMED VISUALISATION

Figure 4.6 exhibits a line graph illustrating the trends of fuel consumed available on the website. This visualization aids in data analysis, in accordance with the objectives previously delineated.

Figure 4.7 TANK VOLUMES VISUALISATION

Figure 4.7 illustrates a line graph presenting the trends of Tank volumes A and B available on the website serveing the purpose of data analysis, aligning with the objectives outlined earlier.

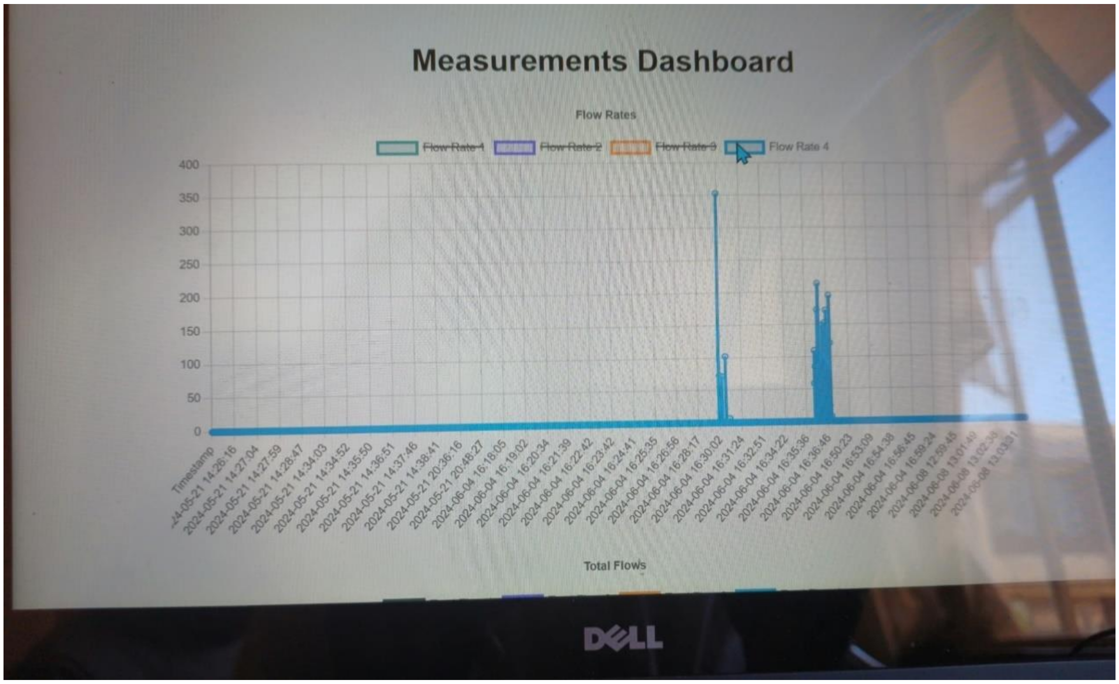
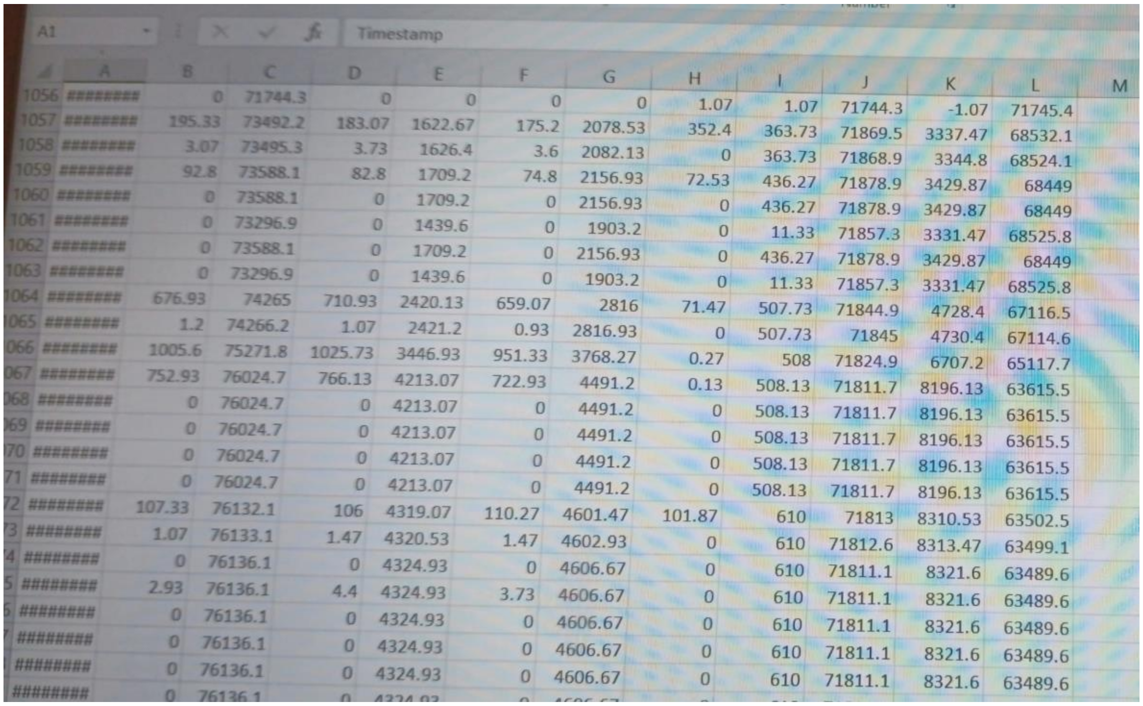


Figure 4.8 FLOWRATES VISUALISAION

Figure 4.8 depicts a line graph showcasing the fluctuating trends of the four flow rates accessible on the website. This graphical representation is instrumental in fulfilling the data analysis objectives previously articulated.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1056	########	0	71744.3	0	0	0	0	1.07	1.07	71744.3	-1.07	71745.4	
1057	########	195.33	73492.2	183.07	1622.67	175.2	2078.53	352.4	363.73	71869.5	3337.47	68532.1	
1058	########	3.07	73495.3	3.73	1626.4	3.6	2082.13	0	363.73	71868.9	3344.8	68524.1	
1059	########	92.8	73588.1	82.8	1709.2	74.8	2156.93	72.53	436.27	71878.9	3429.87	68449	
1060	########	0	73588.1	0	1709.2	0	2156.93	0	436.27	71878.9	3429.87	68449	
1061	########	0	73296.9	0	1439.6	0	1903.2	0	11.33	71857.3	3331.47	68525.8	
1062	########	0	73588.1	0	1709.2	0	2156.93	0	436.27	71878.9	3429.87	68449	
1063	########	0	73296.9	0	1439.6	0	1903.2	0	11.33	71857.3	3331.47	68525.8	
1064	########	676.93	74265	710.93	2420.13	659.07	2816	71.47	507.73	71844.9	4728.4	67116.5	
1065	########	1.2	74266.2	1.07	2421.2	0.93	2816.93	0	507.73	71845	4730.4	67114.6	
066	########	1005.6	75271.8	1025.73	3446.93	951.33	3768.27	0.27	508	71824.9	6707.2	65117.7	
067	########	752.93	76024.7	766.13	4213.07	722.93	4491.2	0.13	508.13	71811.7	8196.13	63615.5	
068	########	0	76024.7	0	4213.07	0	4491.2	0	508.13	71811.7	8196.13	63615.5	
69	########	0	76024.7	0	4213.07	0	4491.2	0	508.13	71811.7	8196.13	63615.5	
70	########	0	76024.7	0	4213.07	0	4491.2	0	508.13	71811.7	8196.13	63615.5	
71	########	0	76024.7	0	4213.07	0	4491.2	0	508.13	71811.7	8196.13	63615.5	
72	########	107.33	76132.1	106	4319.07	110.27	4601.47	101.87	610	71813	8310.53	63502.5	
3	########	1.07	76133.1	1.47	4320.53	1.47	4602.93	0	610	71812.6	8313.47	63499.1	
4	########	0	76136.1	0	4324.93	0	4606.67	0	610	71811.1	8321.6	63489.6	
5	########	2.93	76136.1	4.4	4324.93	3.73	4606.67	0	610	71811.1	8321.6	63489.6	
6	########	0	76136.1	0	4324.93	0	4606.67	0	610	71811.1	8321.6	63489.6	
	########	0	76136.1	0	4324.93	0	4606.67	0	610	71811.1	8321.6	63489.6	
	########	0	76136.1	0	4324.93	0	4606.67	0	610	71811.1	8321.6	63489.6	

Figure 4.9 EXTRACTED DATA FROM VISUALISATION WEBSITE

Figure 4.9 displays an extracted Excel data sheet from the website, presenting various variables along with their respective values. This data serves as a foundation for deeper analysis and informed decision-making processes.

CHAPTER 5: RECOMMENDATIONS AND CONCLUSIONS

5.1: INTRODUCTION

This chapter describes in detail the strength and weakness of the project and the recommendations made for future design projects

5.2 DISCUSSION OF PERFORMANCE

The performance of the developed real-time fuel monitoring system can be evaluated based on several key criteria:

1. **Accuracy:** Assess the accuracy of the flowmeters in measuring fuel consumption against manual records and historical data. Any discrepancies should be identified and addressed promptly.
2. **Reliability:** Evaluate the reliability of data transmission and storage mechanisms. The system should be able to capture real-time data consistently without interruptions or data loss.
3. **Security:** Measure the effectiveness of security measures implemented to protect data integrity and prevent unauthorized access. Regular security audits can help identify vulnerabilities and ensure compliance with data protection standards.
4. **Operational Efficiency:** Monitor the system's impact on operational efficiency, such as reducing fuel wastage, optimizing fuel consumption, and minimizing downtime associated with manual monitoring processes.
5. **User Satisfaction:** Gather feedback from end-users at Hwange Power Station regarding their experience with the new monitoring system. User satisfaction surveys can provide insights into usability, functionality, and areas for improvement.

By addressing these aspects in a comprehensive evaluation, stakeholders can gain a clear understanding of the system's performance and its potential to contribute to the operational objectives of Hwange Power Station.

5.3 LIMITATIONS OF THE PROJECT PROTOTYPE

Despite its benefits, the developed prototype of the real-time fuel monitoring system may have some limitations that need to be acknowledged:

1. **Cost Considerations:** The initial setup cost of installing flowmeters and setting up the monitoring infrastructure can be significant. This might pose financial challenges for widespread adoption, particularly in smaller or less funded power stations.
2. **Technical Challenges:** The effectiveness of the system may be influenced by technical challenges such as signal interference, data transmission errors, or hardware malfunctions. These issues could affect the accuracy and reliability of the monitoring data.
3. **Regulatory Compliance:** Ensure that the monitoring system complies with all relevant regulatory requirements and standards governing fuel management and data security in Zimbabwe.

5.4 RECOMMENDATIONS

Based on the findings and outcomes of the study on developing and implementing a real-time fuel monitoring system at Hwange Power Station, several recommendations can be made to enhance the effectiveness and sustainability of the system:

1. **Continuous Monitoring and Maintenance:** Establish a routine maintenance schedule for the flowmeters and associated hardware to ensure accurate and reliable operation over time. Regular calibration of the flow sensors should also be performed to maintain measurement accuracy.
2. **Integration with Existing Systems:** Explore opportunities to integrate the newly developed monitoring system with other existing operational systems at the power station, such as the plant management system or security systems. This integration can provide a holistic view of operations and enhance overall efficiency.
3. **Training and Capacity Building:** Conduct training programs for the operational staff at Hwange Power Station on the usage and maintenance of the new monitoring system. This will empower them to effectively utilize the system for monitoring, analysis, and decision-making.
4. **Enhanced Security Measures:** Implement additional security protocols to safeguard the integrity of the data collected by the monitoring system. This may include encryption of data transmissions, access control mechanisms, and regular security audits.
5. **Scalability Considerations:** Design the monitoring system with scalability in mind to accommodate future expansions or upgrades of the power station. This includes provisions for adding more flowmeters or integrating with new technologies as they become available.
6. **Feedback Mechanism:** Establish a feedback mechanism involving stakeholders and end-users to continuously improve the system based on operational experiences and evolving requirements.

5.5 CONCLUSIONS

In conclusion, the development and implementation of a real-time fuel monitoring system at Hwange Power Station represent a significant advancement in addressing critical challenges related to fuel management. This project aimed to enhance operational efficiency, mitigate risks associated with fuel theft and fraud, and provide a foundation for sustainable energy management practices. Based on the findings and analysis, several key conclusions can be drawn:

1. **Effectiveness of Real-Time Monitoring:** The implementation of flowmeters and a centralized monitoring system has demonstrated the capability to provide accurate, real-time data on fuel consumption. This has enabled proactive decision-making and improved operational planning.
2. **Impact on Security and Integrity:** By integrating enhanced security measures, including secure data storage and encryption protocols, the system has effectively mitigated risks related to unauthorized access and data tampering.
3. **Operational Efficiency and Cost Savings:** Initial assessments indicate potential cost savings through optimized fuel consumption and reduced losses from theft or inefficiencies in manual monitoring. Continued monitoring and analysis will be necessary to quantify these benefits accurately.
4. **Challenges and Lessons Learned:** The project encountered challenges related to initial costs, technical complexities, and operational adjustments. These challenges underscore the importance of robust planning, stakeholder engagement, and ongoing support for successful implementation.
5. **Future Directions:** Looking ahead, continuous improvement and scalability will be crucial for maximizing the system's long-term benefits. Recommendations for further integration with existing systems, ongoing training, and adaptive management practices will support sustained efficiency gains and operational excellence.

In conclusion, the real-time fuel monitoring system represents a pivotal step towards modernizing operations at Hwange Power Station. By addressing both technical and operational considerations, this initiative not only enhances operational resilience but also sets a precedent for future advancements in energy management within the Zimbabwe Power Company.

REFERENCES

[1] "Fuel Monitoring System based on IoT: Overview and Device," IEEE Xplore, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9817332/. [Accessed: Jun. 17, 2024].

[2] "Design of a real-time ship fuel consumption monitoring system with self-checking function," IEEE Conference Publication, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8047849/. [Accessed: Jun. 17, 2024].

[3] "A Real-Time Motion Based Fuel Monitoring Technique For Vehicle Tracking Systems," IEEE Conference Publication, [Online]. Available: https://ieeexplore.ieee.org/document/9350860/authors. [Accessed: Jun. 17, 2024].

[4] "A System for Integrated Visualization in Oil Exploration and Production," IEEE Computer Graphics and Applications, vol. 36, no. 3, pp. 16-21, May-Jun. 2016, doi: 10.1109/MCG.2016.60.

[5] "A data-driven real time energy management strategy for fuel cell hybrid vehicle in the connected environment," 2022 13th Asian Control Conference (ASCC), [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9828223/. [Accessed: Jun. 17, 2024].

[6] "A big data approach for Fuel Oil Consumption estimation in the context of Weather Routing," [Online]. Available: https://ieeexplore.ieee.org/document/9898181/. [Accessed: Jun. 17, 2024].

[7] "Parallel Implementation and Performance Analysis of a 3D Oil-Water Reservoir Visualization Tool," [Online]. Available: https://ieeexplore.ieee.org/document/6332277/. [Accessed: Jun. 17, 2024].

[8] "Cloud computing and data management," 2016 International Conference on Data and Software Engineering (ICoDSE), pp. 1-6, Nov. 2016, doi: 10.1109/CDAN.2016.7570892.

[9] "Vulnerability Detection and Security Enhancing Using XAMPP," [Online]. Available: https://ieeexplore.ieee.org/document/10346734/. [Accessed: Jun. 17, 2024].

[10] "JavaScript Development Environment for Programming Education," 2019 Seventh International Symposium on Computing and Networking Workshops (CANDARW), pp. 208-214, Dec. 2019, doi: 10.1109/CANDARW.2019.00058.

[11] "Responsive Web Design with HTML5 and CSS: Build future-proof responsive websites using the latest HTML5 and CSS techniques," Packt Publishing, [Online]. Available: https://ieeexplore.ieee.org/document/10162300/. [Accessed: Jun. 17, 2024].

[12] "Web Application Design and Implementation: Apache 2, PHP5, MySQL, JavaScript, and Linux/UNIX," IEEE Xplore, [Online]. Available: https://ieeexplore.ieee.org/book/5989430/. [Accessed: Jun. 17, 2024].

[13] "JavaScript Design Patterns: Deliver fast and efficient code with JavaScript design patterns," IEEE Xplore, [Online]. Available: https://ieeexplore.ieee.org/book/10522570/. [Accessed: Jun. 17, 2024].

[14] ELECTRONOOBS, "ESP32 IOT database with Arduino POST method and PHP," [Online]. Available: https://electronoobs.com/eng_arduino_tut172.php. [Accessed: Jun. 17, 2024].

[15] Rui Santos, "ESP32/ESP8266 Insert Data into MySQL Database using PHP and Arduino IDE," [Online]. Available: https://github.com/RuiSantosdotme/ESP32-ESP8266-PHP-MySQL. [Accessed: Jun. 17, 2024].

[16] Dwi Cakra Dani Elle, "Data Transfer from ESP32 into Database MySQL Using PHP," Medium, May 7, 2023. [Online]. Available: https://medium.com/@dwicakradanielle/data-transfer-from-esp32-into-database-mysql-using-php-3af0d7092a2a. [Accessed: Jun. 17, 2024].

[17] Random Nerd Tutorials, "ESP32/ESP8266 Insert Data into MySQL Database," [Online]. Available: https://randomnerdtutorials.com/esp32-esp8266-mysql-database-php/. [Accessed: Jun. 17, 2024].

[18] Microcontrollers Lab, "ESP32 Insert Data into MySQL Database using PHP and Arduino IDE," [Online]. Available: https://microcontrollerslab.com/esp32-mysql-database-php/. [Accessed: Jun. 17, 2024].

[19] "Design and Implementation of ESP32-Based IoT Devices," Sensors, vol. 23, no. 15, Article 6739, Jul. 2023, doi:10.3390/s23156739.

[20] Random Nerd Tutorials, "ESP32 Pinout Reference: Which GPIO pins should you use?" [Online]. Available: https://randomnerdtutorials.com/esp32-pinout-reference-gpios/. [Accessed: Jun. 17, 2024].

[21] Stack Overflow, "Can you use all the ESP32's GPIO pins when the WiFi is working?" [Online]. Available: https://stackoverflow.com/questions/65125302/can-you-use-all-the-esp32s-gpio-pins-when-the-wifi-is-working. [Accessed: Jun. 17, 2024].

[22] "PERFORMING OPEN SOURCE WIFI MESH SENSOR NETWORK USING ESP32," Journal of Emerging Technologies and Innovative Research (JETIR), vol. 8, no. 6, pp. 14-19, Jun. 2021.