UNIVERSITY OF NAIROBI

FACULTY OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING

AUTOMATED ELECTROLYSIS SYSTEM

PROJECT INDEX: PRJ 098

SUBMITTED BY: DONATUS OMONDI OCHIENG

REGISTRATION NUMBER: F17/1798/2019

SUPERVISOR(S): DR DAVIES SEGERA

EXAMINER: DR ING WILFRED MWEMA

This project is submitted in partial fulfilment of the requirement for the award of the Degree of Bachelor of Science in Electrical and Electronic Engineering at the University of Nairobi

SUBMITTED ON: 25TH JUNE 2024

DECLARATION OF ORIGINALITY

1. I understand what plagiarism is and I am aware of the university policy in this regard.
2. I declare that this final year project report is my original work and has not been submitted elsewhere for examination, award of a degree or publication. Where other people's work or my own work has been used, this has properly been acknowledged and referenced in accordance with the University of Nairobi's requirements.
3. I have not sought or used the services of any professional agencies to produce this work.
4. I have not allowed, and shall not allow anyone to copy my work with the intention of passing it off as his/her own work.
5. I understand that any false claim in respect of this work shall result in disciplinary action, in accordance with University anti-plagiarism policy.

Signature: …………..………………………

Date: …………………………………

CERTIFICATION

This report has been submitted to the Department of Electrical and Information Engineering, University of Nairobi with my approval as supervisor:

Signature: …………..………………………

Date: …………………………………

DEDICATION

I would like to dedicate this project to my parents, siblings, friends, and mentors, all of whom have significantly influenced my journey. Their unwavering support, encouragement, and invaluable guidance have been instrumental in helping me navigate the challenges and milestones of this engineering project. I am deeply grateful for their belief in my abilities and for always being there to offer advice, motivation, and a listening ear. This accomplishment would not have been possible without their contributions, and I am truly thankful for their presence in my life.

ACKNOWLEDGEMENTS

ABSTRACT

An automated electrolysis system is a sophisticated technology designed to streamline and optimise the process of electrolysis, which involves the use of electrical energy to drive a non-spontaneous chemical reaction. This system is engineered to autonomously manage and control various parameters essential for efficient electrolysis, including voltage, current, temperature, and electrolyte concentration. By integrating advanced sensors and control algorithms, the system ensures precise monitoring and adjustment of these parameters in real-time, leading to enhanced efficiency, safety, and reliability. The automation reduces the need for manual intervention, thereby minimising human error and operational costs.

The core of an automated electrolysis system lies in its intelligent control unit, which employs a combination of hardware and software components to execute the electrolysis process with high precision. The control unit is equipped with a microcontroller that processes data from sensors monitoring the electrolysis cell. Advanced software algorithms analyse this data to optimise the electrolysis conditions continuously. Furthermore, the system can be programmed to handle different types of electrolysis reactions, whether for hydrogen and oxygen production, metal plating, or water purification, making it highly versatile and adaptable to various industrial applications.

In addition to operational efficiency, an automated electrolysis system offers significant improvements in safety and sustainability. The system can detect anomalies such as gas leaks, overheating, or short circuits, triggering automatic shutdown procedures to prevent accidents. Enhanced control over the electrolysis process also leads to more efficient use of energy and raw materials, reducing the environmental footprint. By automating the electrolysis process, industries can achieve consistent product quality, increased throughput and reduced operational costs, thereby contributing to more sustainable and economically viable manufacturing practices.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

PCB…………………………………………….Printed Circuit Board

AEL…………………………………………….Alkaline Electrolyser

PEM…………………………………………….Proton Exchange Membrane

SOE…………………………………………….Solid Oxide Electrolyser

SMR…………………………………………….Steam Methane Reforming

POX…………………………………………….Partial Oxidation

VOC…………………………………………….Volatile Organic Compounds

MOS…………………………………………….Metal Oxide Semiconductor

NDIR…………………………………………….Non-dispersive Infrared

RTD…………………………………………..Resistance Temperature Detector

UART…………………………………………...Universal Asynchronous Receive Transmit

I2C…………………………………………….Inter-Intergrated Circuit

SPI…………………………………………….Serial Peripheral Interface

GSM…………………………………………….Global System for Mobile Communications

GPRS…………………………………………….General Packet Radio Service

Wi-Fi…………………………………………….Wireless Fidelity

OTA…………………………………………..Over The Air Update

1 CHAPTER ONE - INTRODUCTION

1.1 BACKGROUND

Electrolysis has been a fundamental chemical process used for over a century, pivotal in various industrial applications such as hydrogen and oxygen production, metal extraction, and electroplating. Electrolysis involves the use of an electrical current to drive a chemical reaction that otherwise would not occur spontaneously. This process typically requires careful manual control to maintain optimal conditions, including the regulation of voltage, current, and electrolyte composition.

1.2 PROBLEM STATEMENT

The increasing global demand for hydrogen and oxygen, driven by their critical roles in various industries such as energy, transportation, and healthcare, presents a significant challenge. Traditional methods of producing these gases often rely on fossil fuels, leading to environmental degradation and contributing to greenhouse gas emissions. As industries seek cleaner and more sustainable energy sources, the need for an efficient, eco-friendly method to produce hydrogen and oxygen becomes paramount. This demand necessitates innovative solutions that not only meet industrial needs but also align with environmental sustainability goals.

1.3 JUSTIFICATION

The implementation of an automated electrolysis system is justified by its potential to revolutionize efficiency, safety, and overall performance in industrial electrolysis applications. Automation ensures precise control over critical process parameters, significantly reducing the likelihood of human error and enhancing the consistency and quality of the output. This precision also leads to optimal use of energy and raw materials, contributing to cost savings and environmental sustainability. Furthermore, the advanced monitoring capabilities of an automated system enhance safety by enabling immediate detection and response to anomalies, thereby preventing accidents and equipment damage. The flexibility of automated systems allows for easy adaptation to different electrolysis processes, making them highly versatile and scalable for various industrial

needs. Ultimately, the shift towards automation in electrolysis not only addresses the current challenges but also sets a new standard for future technological advancements in the field.

1.4 OBJECTIVES

1.4.1 MAIN OBJECTIVE

To develop an automated electrolysis system that enhances efficiency, safety, and adaptability for diverse industrial applications with key focus on hydrogen and oxygen production.

1.4.2 SPECIFIC OBJECTIVES

1. To detect gas flowrate, concentration and temperature realised by the electrolysis system
2. To detect gas pressure in a gas reservoir
3. To design pcb incorporating hardware used to automate the electrolysis system
4. To develop firmware to achieve automation of electrolysis operation

1.5 SCOPE

The automated electrolysis system aims to achieve control of the electrolysis process by measuring the key parameter realised in the process which is the gas. Key gas parameters include flowrate, concentration, temperature and pressure. Using these parameters the system will be able to discerne whether to continue the electrolysis process or to hault the process until certain conditions and thresholds are achieved.

Good operation without manual interaction ensures safety of personnel since production of oxygen and hydrogen in very large quantities poses a lot of risk to life.

However, there are still unavoidable manual interactions in this first prototype which can be catered for with further improvements in the control system

2 CHAPTER TWO – LITERATURE REVIEW

2.1 ELECTROLYSIS PROCESS

Electrolysis is a process that uses electrical energy to drive a non-spontaneous chemical reaction. This technique involves the passage of an electric current through an electrolyte to cause a chemical change. The electrolyte, a substance containing free ions, typically dissolves in water, enabling the ions to move freely. When electrical energy is supplied through electrodes immersed in the electrolyte, positive ions migrate to the cathode, where reduction occurs, while negative ions move towards the anode, where oxidation occurs. This process results in the decomposition of the electrolyte into its constituent elements or the formation of new substances (Bard, 2001).

Electrolysis can be utilized for various purposes, including the decomposition of water into hydrogen and oxygen, electroplating, and the extraction of metals from ores. The efficiency and effectiveness of electrolysis depend on several factors, including the nature of the electrolyte, the type of electrodes, the applied voltage, and the temperature of the solution (McMurry, 2007).

Faraday's first law states that the mass (m) of a substance deposited or liberated at an electrode is directly proportional to the charge (Q)

$$m \; \alpha \; Q \rightarrow \frac{m}{Q} = Z$$

The constant of proportionality Z is called the electro-chemical equivalent (ECE) of the substance. Thus, the ECE can be defined as the mass of the substance deposited or liberated per unit charge.

Faraday's second law states that the amounts of different substances liberated by the same quantity of electricity passing through the electrolytic solution are proportional to their equivalent weights (E). This is expressed as:

$$m \; \alpha \; E; \quad E = \frac{M}{v}$$

$$\frac{m_1}{m_2} = \frac{M_1/v_1}{M_2/v_2}$$

where:

- m is the mass of the substance liberated at an electrode
- M is the molar mass of the substance
- v the valency number of ions in the substance
- m_1 and m_2 are the masses of substances liberated
- M_1/v_1 and M_2/v_2 are their respective equivalent weights (Bard, 2001).

A v-valent ion will require v electrons for discharge. For x electrons flowing, $\frac{x}{v}$ atoms will be discharged. Thus, the mass discharged can be obtained as

$$m = \frac{xM}{vN_A} = \frac{QM}{eN_Av} = \frac{QM}{vF}$$

where

1. N_A is the Avogadro constant
2. Q = xe is the total charge, equal to the number of electrons (x) times the elementary charge (e)
3. F is the Faraday's constant

Using these concepts we can quantify the amount of mass liberated based on the current flow in the electrolysis system

2.2   OTHER HYDROGEN AND OXYGEN PRODUCTION PROCESSES

Hydrogen and oxygen can be produced using several industrial processes apart from electrolysis. Some of the common methods include:

1. **Steam Methane Reforming (SMR):** This is the most widely used method for hydrogen production, where methane reacts with steam under high pressure to produce hydrogen and carbon monoxide. The carbon monoxide is then reacted with water to produce additional hydrogen and carbon dioxide. While SMR is cost-effective and efficient, it relies on fossil fuels and generates significant CO_2 emissions, making it less environmentally friendly than electrolysis (Kothari, 2008).

2. **Partial Oxidation (POX)**: In this process, hydrocarbons such as natural gas are partially oxidized with oxygen to produce hydrogen and carbon monoxide. POX can handle heavier hydrocarbons compared to SMR but still produces CO2 and other pollutants (Kothari, 2008).

3. **Biomass Gasification**: This method involves converting organic materials into hydrogen, carbon monoxide, and carbon dioxide through high-temperature reactions. Biomass gasification is renewable and can utilize waste materials, but the technology is less mature and requires further development for widespread use (Balat, 2008).

4. **Cryogenic Air Separation**: In cryogenic air separation, atmospheric air is first compressed and cooled to remove impurities like water vapor and carbon dioxide. The air is then further cooled to cryogenic temperatures, causing it to liquefy. This liquefied air is fed into a distillation column where it is separated into its components based on their different boiling points. Oxygen, having a higher boiling point (-183°C) compared to nitrogen (-195.8°C), is collected from the bottom of the column. The separated oxygen is then collected and stored in liquid or gaseous form for industrial and medical use (Perry, 2008).

5. **Pressure Swing Adsorption**: In PSA, compressed air is passed through a bed of adsorbent material, typically zeolites, which preferentially adsorb nitrogen over oxygen at high pressure. This leaves an oxygen-enriched stream in the gas phase. Once the adsorbent material becomes saturated with nitrogen, the pressure is lowered, causing the nitrogen to desorb and be removed from the system. The cycle is repeated, with the adsorbent being regenerated with each cycle.

This process allows for the continuous production of high-purity oxygen, which is then collected and supplied for various applications (Sircar, 2000).

6. **Vacuum Pressure Swing Adsorption**: VPSA operates similarly to PSA but incorporates a vacuum step to enhance desorption efficiency. Compressed air is passed through an adsorbent bed where nitrogen is preferentially adsorbed, leaving an oxygen-enriched stream. When the adsorbent bed becomes saturated, instead of just reducing the pressure to atmospheric levels, a vacuum is applied to further lower the pressure. This vacuum step enhances the desorption of nitrogen, improving the efficiency and purity of the oxygen produced. The cycle of adsorption and vacuum desorption is repeated, and the oxygen is collected for use in industrial, medical, and other applications (Ruthven, 1994).

Comparatively, electrolysis stands out as a cleaner alternative, especially when powered by renewable energy sources, as it produces no direct carbon emissions and results in high-purity hydrogen and oxygen (Kumar, 2019).

2.3 HYDROGEN AND OXYGEN PRODUCTION USING ELECTROLYSIS

Hydrogen and oxygen production via electrolysis involves the decomposition of water (H_2O) into its constituent gases as follows:

$$2H_2O(l) \rightarrow 2H_2(g) + O_2(g)$$
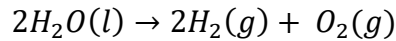
This process occurs in an electrolyser, which consists of an anode and a cathode separated by an electrolyte. When an electric current is applied, the following reactions take place:

At the cathode (reduction):

$$2H_2O + 2e^- \rightarrow H_2 + 2OH^-$$
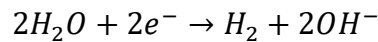
At the anode (oxidation):

$$4OH^- \rightarrow O_2 + 2H_2O + 4e^-$$
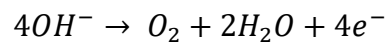
These reactions result in the liberation of hydrogen gas at the cathode and oxygen gas at the anode.

Hydrogen and oxygen production using electrolysis offers several advantages for hydrogen and oxygen production:

1. Environmental Benefits: When powered by renewable energy sources like wind, solar, or hydroelectric power, electrolysis produces hydrogen and oxygen with zero carbon emissions, making it an environmentally sustainable method (Momirlan, 2005).
2. High Purity: Electrolysis produces hydrogen and oxygen of high purity, which is essential for applications in industries such as electronics, pharmaceuticals, and fuel cells (Kumar, 2019).
3. Scalability: Electrolysis systems can be scaled to match demand, from small laboratory setups to large industrial plants, providing flexibility in production capacity (Patel, 2016).
4. Energy Storage: Hydrogen produced through electrolysis can be used as an energy carrier, storing excess renewable energy and providing a solution to the intermittency of renewable energy sources (Kothari, 2008).
5. Decentralised Production: Electrolysis can be implemented locally, reducing the need for extensive transportation and storage infrastructure for hydrogen, thereby enhancing energy security (Balat, 2008).

2.4 TYPES OF ELECTROLYSERS

Electrolysers are devices used to induce water electrolysis, splitting water molecules into hydrogen and oxygen gases. The type of electrolyser used can significantly impact the efficiency, cost, and application suitability of hydrogen production processes. There are different types of electrolysers, including Alkaline Electrolysers (AEL), Proton Exchange Membrane (PEM) Electrolysers, and Solid Oxide Electrolysers (SOE).

2.4.1 ALKALINE ELECTROLYSERS (AEL)

Alkaline electrolysers are the oldest and most established type of electrolyser, typically using an aqueous solution of potassium hydroxide (KOH) or sodium hydroxide (NaOH) as the electrolyte. The cell structure includes a cathode and an anode, separated by a diaphragm to prevent gas mixing.

Advantages include:

1. Mature Technology: Proven track record with long operational life.
2. Lower Catalyst Cost: Use of non-precious metal catalysts such as nickel.
3. Scalability: Suitable for large-scale industrial hydrogen production.

Disadvantages include:

1. Lower Efficiency: Generally lower efficiency compared to PEM electrolysers.
2. Large Footprint: Requires larger space for installation.
3. Long Start-Up Time: Less responsive to variable loads, making it less suitable for intermittent renewable energy sources (Carmo, 2013).

2.4.2 PROTON EXCHANGE MEMBRANE (PEM) ELECTROLYSERS

PEM electrolysers utilize a solid polymer membrane as the electrolyte, which conducts protons from the anode to the cathode and acts as a barrier to prevent gas crossover. The MEA (Membrane Electrode Assembly) is a crucial component comprising the membrane, catalyst layers, and gas diffusion layers.

In PEM electrolysis, water is supplied to the anode side, where it undergoes oxidation to produce oxygen gas, protons, and electrons. The protons migrate through the PEM to the cathode, while the electrons travel through an external circuit to the cathode. At the cathode, protons combine with electrons to form hydrogen gas. The PEM plays a crucial role in allowing only protons to pass through while preventing gas crossover, thereby ensuring high purity of the produced hydrogen (Carmo, 2013; Fiorenza, 2019).

Advantages include:

1. High Efficiency and Purity: PEM electrolyzers can operate at higher current densities, leading to higher efficiency. The solid polymer membrane ensures that the produced hydrogen and oxygen are of high purity, typically exceeding 99.99% (Fiorenza, 2019).
2. Compact Design and Fast Response: The design of PEM electrolyzers is compact, and they can respond quickly to changes in power input, making them suitable for integration with variable renewable energy sources such as solar and wind (Giddey, 2012).
3. High Pressure Operation: PEM electrolyzers can operate at high pressures (up to 70 bar) without the need for mechanical compressors, which simplifies the system design and reduces costs associated with hydrogen compression and storage (Carmo, 2013).
4. Scalability: PEM electrolyzers can be easily scaled from small laboratory setups to large industrial systems, providing flexibility in production capacity to meet various application requirements (Kumar, 2019).
5. Safety: The solid membrane reduces the risk of gas leakage and mixing, enhancing the overall safety of the system (Gahleitner, 2013).

Disadvantages include:

1. High Cost of Materials: The use of noble metal catalysts (e.g., platinum, iridium) and the expensive polymer membrane (e.g., Nafion) contribute significantly to the high cost of PEM electrolyzers. Efforts are ongoing to develop cost-effective and efficient catalysts and membranes (Carmo, 2013).
2. Durability and Lifespan: The durability of PEM electrolyzers is affected by the degradation of the membrane and catalysts, particularly under harsh operating conditions such as high current densities and temperatures. Enhancing the lifespan of these components is crucial for the commercial viability of PEM electrolysis (Kumar, 2019).
3. Water Management: Efficient water management is essential to prevent membrane dehydration or flooding, both of which can impair the performance and longevity of the electrolyzer. Advanced designs and control strategies are needed to optimize water distribution within the cell (Gahleitner, 2013).

4. System Integration: Integrating PEM electrolyzers with renewable energy systems poses challenges related to the variability and intermittency of renewable power sources. Effective strategies for managing power fluctuations and ensuring stable operation are necessary (Fiorenza, 2019).

2.4.3 SOLID OXIDE ELECTROLYSERS (SOE)

Solid oxide electrolysers operate at high temperatures (700-1000°C) using a solid ceramic electrolyte that conducts oxygen ions. They are often used in combination with high-temperature waste heat sources, such as in industrial processes or combined heat and power systems.

Advantages include:

1. High Efficiency: High temperature operation reduces the electrical energy requirement, leading to higher efficiency.
2. Integration with Industrial Processes: Can utilize waste heat from industrial processes, improving overall energy efficiency.
3. Flexibility: Can operate in both electrolysis mode and fuel cell mode, providing versatility in energy systems.

Disadvantages include:

1. High Operating Temperature: Requires materials that can withstand high temperatures, increasing costs and complexity.
2. Thermal Cycling: Subject to thermal stresses that can affect durability and lifespan.
3. Startup Time: Long startup time due to the need to reach high operating temperatures (Laguna-Bercero).

Each type of electrolyser has its unique strengths and weaknesses, making them suitable for different applications.

Alkaline electrolysers are well-suited for large-scale industrial hydrogen production due to their cost-effectiveness and mature technology.

PEM electrolysers, with their high efficiency and quick response times, are ideal for integration with renewable energy sources and applications requiring high-purity hydrogen.

Solid oxide electrolysers, with their potential for very high efficiency and use of waste heat, are promising for industrial integration and combined heat and power applications.

2.5 AUTOMATION OF ELECTROLYSIS

2.5.1 IMPORTANT PARAMETERS OF CONSIDERATION IN AN ELECTROLYSIS SYSTEM

The efficiency and effectiveness of an electrolysis system depends on several critical parameters:

1. Voltage and Current: The applied voltage and current density directly affect the rate of electrolysis and the overall energy efficiency of the process (Bard, 2001).
2. Temperature: The temperature of the electrolyte influences the conductivity and reaction kinetics, with higher temperatures generally increasing efficiency but also posing challenges for material stability (Mills, 2014).
3. Electrolyte Concentration: The concentration of the electrolyte determines the ionic conductivity and the efficiency of the electrolysis process. Optimal concentration levels need to be maintained for effective operation (McMurry, 2007).
4. Electrode Material: The choice of electrode materials affects the overpotential and durability of the electrodes. Common materials include platinum, nickel, and specialized alloys (Holton, 2013).
5. Flow Rate: The flow rate of the electrolyte and the generated gases needs to be managed to ensure efficient gas separation and prevent mixing (Kiros, 2008).

2.5.2 IMPORTANT GAS PARAMETERS OF CONSIDERATION

In hydrogen and oxygen production via electrolysis, critical gas parameters include:

1. Flow Rate: The rate at which hydrogen and oxygen gases are produced must be monitored and controlled to match demand and ensure system stability (Dick, 2008).

2. Concentration: The purity of the produced gases is crucial, especially for applications requiring high-purity hydrogen and oxygen (Kumar, 2019).

3. Temperature: The temperature of the produced gases can impact their storage and transport, as well as the efficiency of downstream processes (Lipták, 2003).

4. Pressure: The pressure of the produced gases affects their storage density and safety. Maintaining optimal pressure levels is essential for efficient gas handling (Lipták, 2003).

The gas parameters mentioned (flow rate, concentration, temperature, and pressure) can change with variations in the electrolysis system's operational conditions:

Flow Rate: Changes with the applied current density and voltage, as higher power inputs generally increase the rate of gas production (Bard, 2001).

Concentration: Can vary with the purity of the water used and the efficiency of the gas separation system within the electrolyzer (Kumar, 2019).

Temperature: Influenced by the operational temperature of the electrolyte and the exothermic nature of the electrolysis reactions.

Pressure: Affected by the design of the gas collection system and the operational settings of the electrolyser, such as whether it is operated under pressurized conditions.

2.6 GAS FLOW RATE SENSORS

Gas flow rate sensors measure the volume or mass of gas passing through a sensor over a specified period. Common principles of operation include thermal, differential pressure, ultrasonic, and Coriolis effects. These are explained below:

1. Thermal Mass Flow Sensors: These sensors operate based on the cooling effect of gas flow over a heated element. The sensor typically consists of two temperature sensors: one heated and one unheated. The gas flow cools the heated sensor, and the temperature difference between the two sensors is proportional to the gas flow rate. Thermal mass flow sensors are widely used due to their accuracy and ability to measure low flow rates (Petersen, 2013).

2. Differential Pressure Flow Sensors: These sensors use a restriction (such as an orifice plate, Venturi tube, or flow nozzle) to create a pressure drop in the gas flow. The differential pressure across the restriction is measured and related to the flow rate using Bernoulli's principle. These sensors are robust and suitable for high flow rates but can introduce pressure losses in the system (Beckwith, 2011).
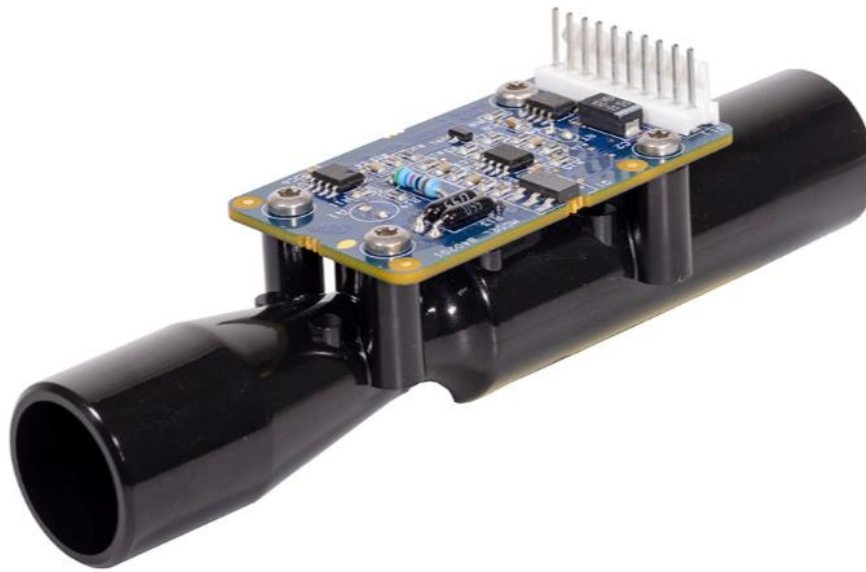
Figure 2.1 Differential Pressure Gas Flowrate Sensor

3. Ultrasonic Flow Sensors: These sensors use ultrasonic waves to measure the flow rate. They work on the transit-time or Doppler effect principle. In transit-time ultrasonic flow sensors, the time difference between upstream and downstream ultrasonic waves is measured and related to the flow rate. Doppler ultrasonic sensors measure the frequency shift of ultrasonic waves reflected by gas particles. These sensors are non-intrusive and suitable for a wide range of gas flows (Baker, 2005).

4. Coriolis Flow Sensors: These sensors measure the mass flow rate by utilizing the Coriolis effect. Gas flowing through a vibrating tube induces changes in the vibration pattern, which are proportional to the mass flow rate. Coriolis flow sensors provide high accuracy and are suitable for measuring various gases, but they are relatively expensive and complex (Santos, 2006).

2.7 GAS CONCENTRATION SENSORS

Gas concentration sensors detect and measure the presence and quantity of specific gas components within a mixture. Common principles of operation include electrochemical, infrared (IR), catalytic bead, and metal-oxide-semiconductor (MOS) techniques. These are explained below:

1. Electrochemical Sensors: These sensors consist of electrodes separated by an electrolyte. When a target gas diffuses through the sensor membrane, it undergoes a chemical reaction at the electrode surfaces, generating an electric current proportional to the gas concentration. Electrochemical sensors are widely used for detecting toxic gases such as carbon monoxide (CO) and hydrogen sulfide (H_2S) due to their high sensitivity and specificity (Stetter, 2008).

2. Infrared (IR) Sensors: IR gas sensors measure gas concentration based on the absorption of infrared light at specific wavelengths corresponding to the gas molecules' vibrational modes. Non-dispersive infrared (NDIR) sensors are commonly used, where an IR source emits light through the gas sample and a detector measures the absorbed light. IR sensors are suitable for detecting hydrocarbons, carbon dioxide (CO_2), and other gases with characteristic absorption bands (Cabrera, 2006)

3. Catalytic Bead Sensors: These sensors detect combustible gases by oxidizing the gas on a heated catalytic bead, causing a temperature rise proportional to the gas concentration. The sensor consists of two beads: an active (catalytic) bead and a reference (non-catalytic) bead. The temperature difference between the beads generates a resistance change, which is measured and related to the gas concentration. Catalytic bead sensors are commonly used for detecting methane (CH_4) and other hydrocarbons (Moseley, 2017)

4. Metal-Oxide-Semiconductor (MOS) Sensors: MOS sensors operate based on the change in electrical resistance of a metal-oxide semiconductor material when exposed to target gases. The gas molecules interact with the surface of the metal oxide, altering its electrical properties. MOS sensors are widely used for detecting volatile organic compounds (VOCs) and other gases due to their sensitivity and low cost (Korotcenkov, 2005).

5. Ultrasonic gas concentration sensors utilise the propagation characteristics of ultrasonic waves through a gas medium to determine the concentration of gas. These sensors are based on the principle that the speed of sound in a gas mixture depends on the composition of the gas. For a binary gas mixture like air (composed primarily of nitrogen and oxygen), the speed of sound v can be related to the oxygen concentration C_{O_2} through the following relationships:

- $M = \left(\gamma * R * \frac{T}{V^2}\right) * 10^3$

  where

  - M is the molecular weight of the gas mixture.
  - γ is the ratio of heat capacity at constant volume to heat capacity at constant pressure.
  - R is a constant of the gas.
  - T is absolute temperature of the gas
  - V speed of sound measured.

- $C_{O_2} = \frac{M - M_{N_2}}{|M_{O_2} - M_{N_2}|} * 100$

  - C_{O_2} is the concentration of oxygen
  - M_{N_2} is the molecular weight of nitrogen
  - M_{O_2} is the molecular weight of oxygen
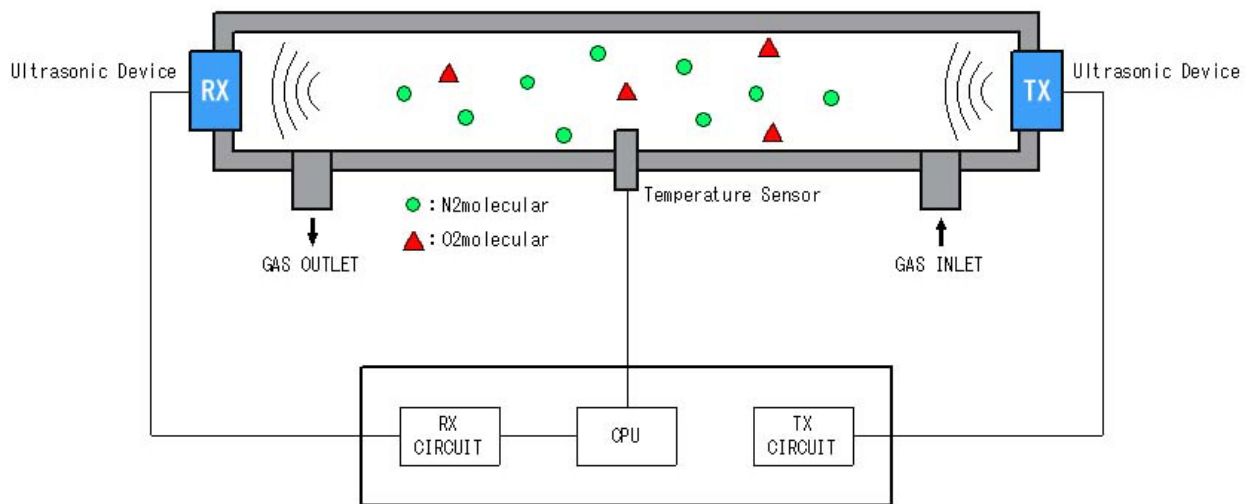
Figure 2.2 Gas concentration measurement by ultrasonic device

The ultrasonic gas concentration sensors can theoretically measure a mixture of any two gases (Daiichi Nekken CO., n.d.)

2.8   GAS TEMPERATURE SENSORS

Gas temperature sensors measure the kinetic energy of gas molecules, which corresponds to the gas temperature. Common principles of operation include resistance temperature detectors (RTDs), thermocouples, and thermistors. These are explained below:

1. Resistance Temperature Detectors (RTDs): RTDs measure temperature based on the change in electrical resistance of a metal (usually platinum) with temperature. The resistance increase is proportional to the temperature rise. RTDs offer high accuracy, stability, and repeatability, making them suitable for precise temperature measurements in gas flows (Bentley, 1998)
2. Thermocouples: Thermocouples consist of two different metal wires joined at one end, creating a junction. The temperature difference between the junction and the other ends generates a voltage (Seebeck effect) proportional to the temperature. Thermocouples are robust, have a wide temperature range, and are widely used in industrial applications, though they are less accurate than RTDs (Lipták, 2003).
3. Thermistors: Thermistors are temperature-sensitive resistors made of semiconductor materials. They exhibit a large change in resistance with temperature, which can be either negative coefficient (NTC) or positive coefficient (PTC). Thermistors are highly sensitive and suitable for measuring small temperature changes in gases but have a limited temperature range and stability compared to RTDs and thermocouples (Sze, 2008).

2.9   GAS PRESSURE SENSORS

Gas pressure sensors measure the force exerted by gas molecules per unit area. Common principles of operation include piezoelectric, capacitive, and strain gauge methods. These are explained below:

1. Piezoelectric Pressure Sensors: These sensors use piezoelectric materials (such as quartz or ceramics) that generate an electric charge when subjected to mechanical stress. The pressure-induced stress creates a charge proportional to the pressure. Piezoelectric sensors are suitable

for dynamic pressure measurements and are widely used in applications requiring high sensitivity and fast response times (Ballato, 1996).

2. Capacitive Pressure Sensors: These sensors consist of two parallel conductive plates separated by a dielectric material. Pressure changes cause a deformation in the plates, altering the capacitance between them. The capacitance change is measured and related to the pressure. Capacitive sensors offer high sensitivity, low power consumption, and stability, making them suitable for various industrial applications (Fraden, 2016).

3. Strain Gauge Pressure Sensors: Strain gauge sensors measure pressure-induced deformation of a diaphragm or other elastic element. The deformation changes the electrical resistance of strain gauges bonded to the element. The resistance change is measured and related to the pressure. Strain gauge sensors are robust, accurate, and widely used in industrial applications (Dally, 1993).

3   CHAPTER THREE - DESIGN

The design of the electrolysis system involved the following main design steps:

1. Hardware design
2. Software design

3.1   HARDWARE DESIGN

The type of electrolyser decided upon was the alkaline electrolyser. This is due to its relatively low cost of realisation.

The electrolysis system consists of four different electrolysis chambers with their respective gas outputs connected to a common collection area. Using different electrolysis chambers as opposed to one big chamber allowed for continuous operation even when one electrolysis chamber came to a fault.

The control system is intended to control the four different alkaline electrolysis chambers. A display for the system was also chosen to be able to inform an operator of the status of the system.

Hardware design involved

1. Selection of hardware components
2. PCB Design

3.1.1   SELECTION OF HARDWARE COMPONENTS

The following hardware components were selected:

1. **STM32F411CEU6 microcontroller unit**
   - This is a 32 bit microcontroller with a clock speed of up to 100MHz. This serves to control other components of the system. This microcontroller has a hardware floating point unit enabling floating point operations without causing a lot of overhead on

microcontroller operation. This microcontroller is equipped with enough interfaces to control other peripherals intended to be used.
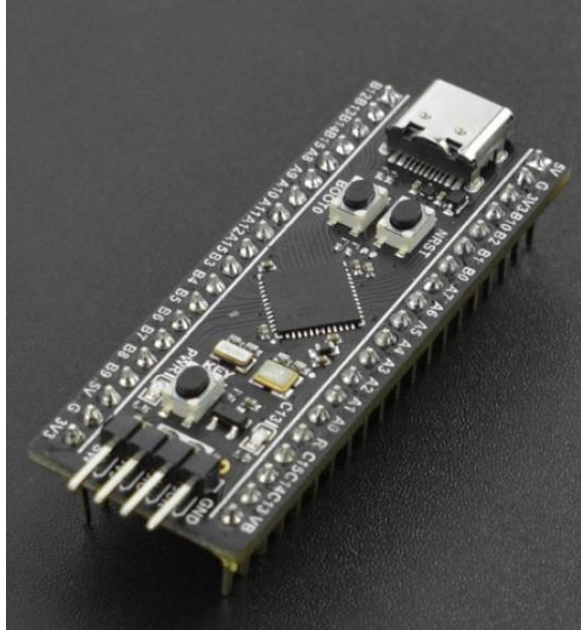
Figure 3.1 STM32F411CEU6 Development board

2. **Relays**

- A set of 4 relays were chosen to control 4 electrolysis chambers and another set of 4 relays chosen to control 4 solenoid valves that allow electrolyte solution into the chamber. These relays were chosen to have a current rating of 10A and a maximum coil voltage of 30VDC.

Figure 3.2 Relay

19

3. **Pressure sensor**

- A silicon strain gauge pressure sensor was chosen incorporating a 4-20ma interface. The 4-20ma interface ensured accurate readings were obtainable from the sensor in the event the sensor and control system were a distance apart. This device has a measuring range of 0 – 10KPa

Figure 3.3 Silicon strain gauge gas pressure sensor

4. **Gas flow rate, concentration and temperature sensor interface**

- A single sensor device capable of measuring flowrate, concentration and temperature was chosen. It has a resistive temperature detector to measure gas temperature. This device uses ultrasonic waves for gas flow rate and concentration measurement. This device communicates over a UART interface with the microcontroller. It has a concentration measuring range of 21% to 95.6% and a resolution of 0.1%. It has a flowrate measuring range of 0 – 10L/min and a resolution of 0.01L/min

Figure 3.4 Ultrasonic gas flowrate, concentration and temperature sensor

5. **Display**

- A display device was chosen with a display controller that communicates over either of two protocols; I2C and SPI, with the microcontroller. The display connected serves to display system operation parameters and inform an operator of the current state of the system.

Figure 3.5 OLED Display

6. **Voltage regulator**

- Two voltage regulators were chosen to supply power at two levels; 5V and 3.3V. 5V was used to power the relays and gas flowrate sensor device. 3.3V was used to facilitate reading of digital voltage levels by the microcontroller since the microcontroller can

handle up to 3.3V at its input. The voltage regulator modules incorporated the lm2596 switching voltage regulator chip capable of supplying up to 3A.

Figure 3.6 LM2596 Voltage Regulator Module

7. **Other elements**

- Other basic elements were included to facilitate connection of these main components to form the control system. These elements are resistors, transistors, leds and a power switch.

3.1.2 PCB DESIGN

The block diagram of the system is shown below:

AUTOMATED ELECTROLYSIS SYSTEM BLOCK DIAGRAM
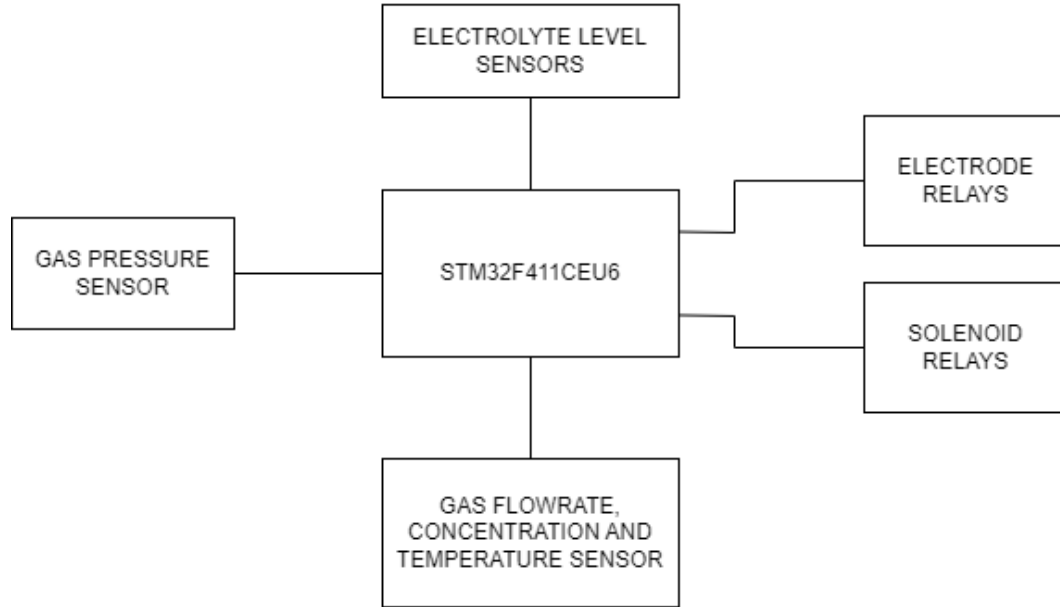
Figure 3.7 Automated electrolysis system block diagram

A schematic covering connection of devices was done as follows:
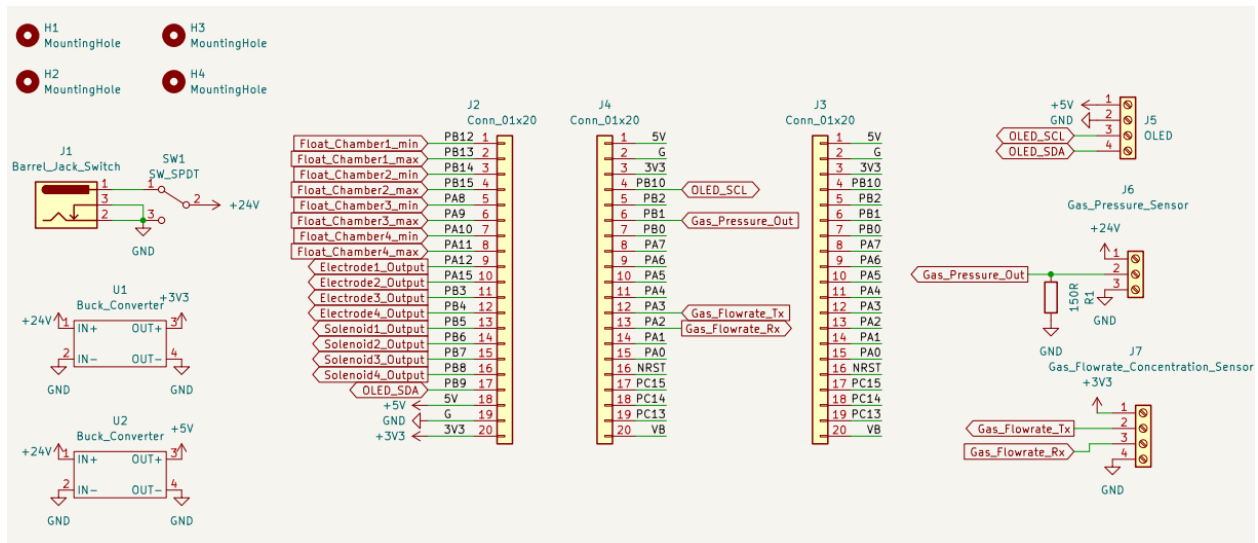
Figure 3.8 General schematic layout
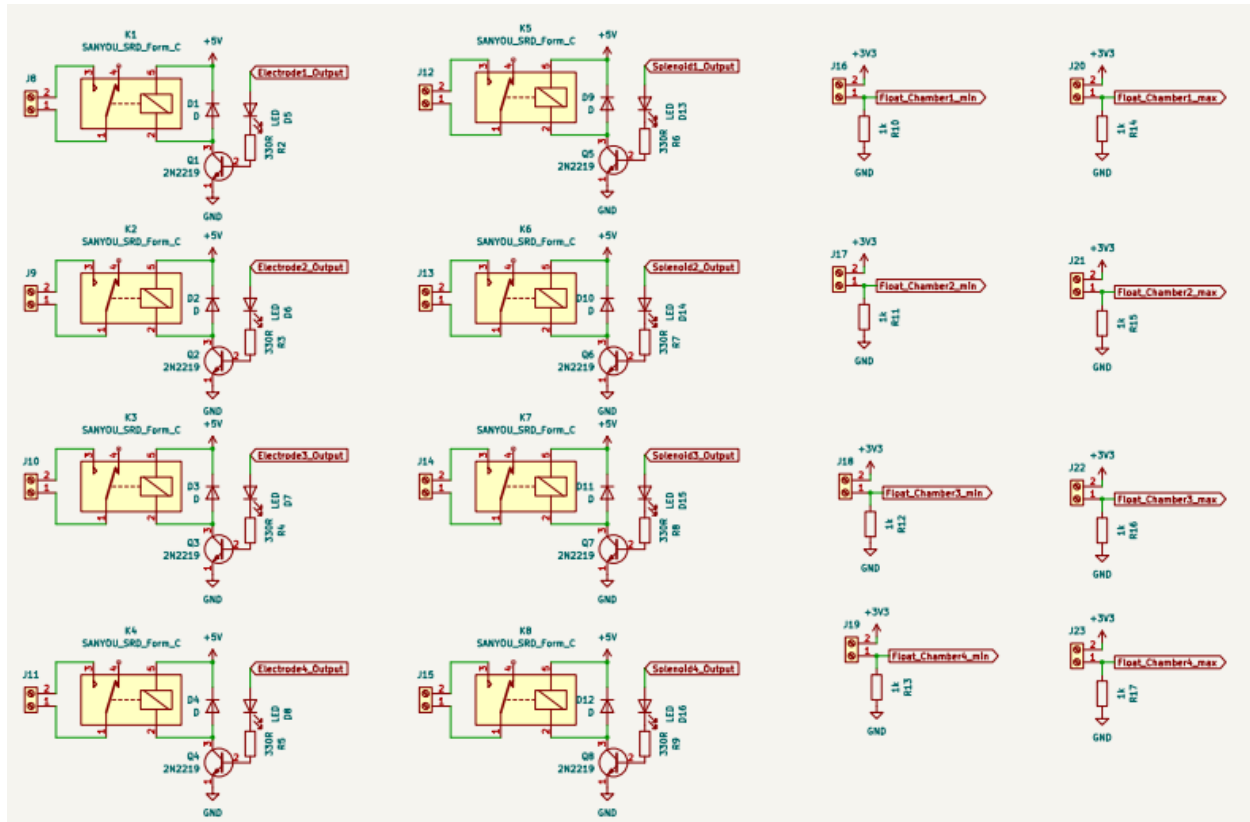
Figure 3.9 Relays and float switches schematic layout

The schematic design incorporated the use to labels so as to make the schematic neat. Breaking down the schematic into pages has also been done to serve the same purpose.

The schematic shows three female header connector pins, 2 of which the STM32F411CEU6 development board will be mounted onto and 1 which will expose unused pins of the microcontroller.

Mounting holes have also been shown to allow for mounting of the final pcb on the standoffs of an enclosure.

Screw terminal blocks interfaces have been used on the schematic for the gas pressure and gas flowrate, concentration and temperature sensor since these devices will not be physically on the devices but mounted elsewhere.

Schematic of relays and float switches has been done on a different page.

The resulting pcb footprint layout was as follows

24
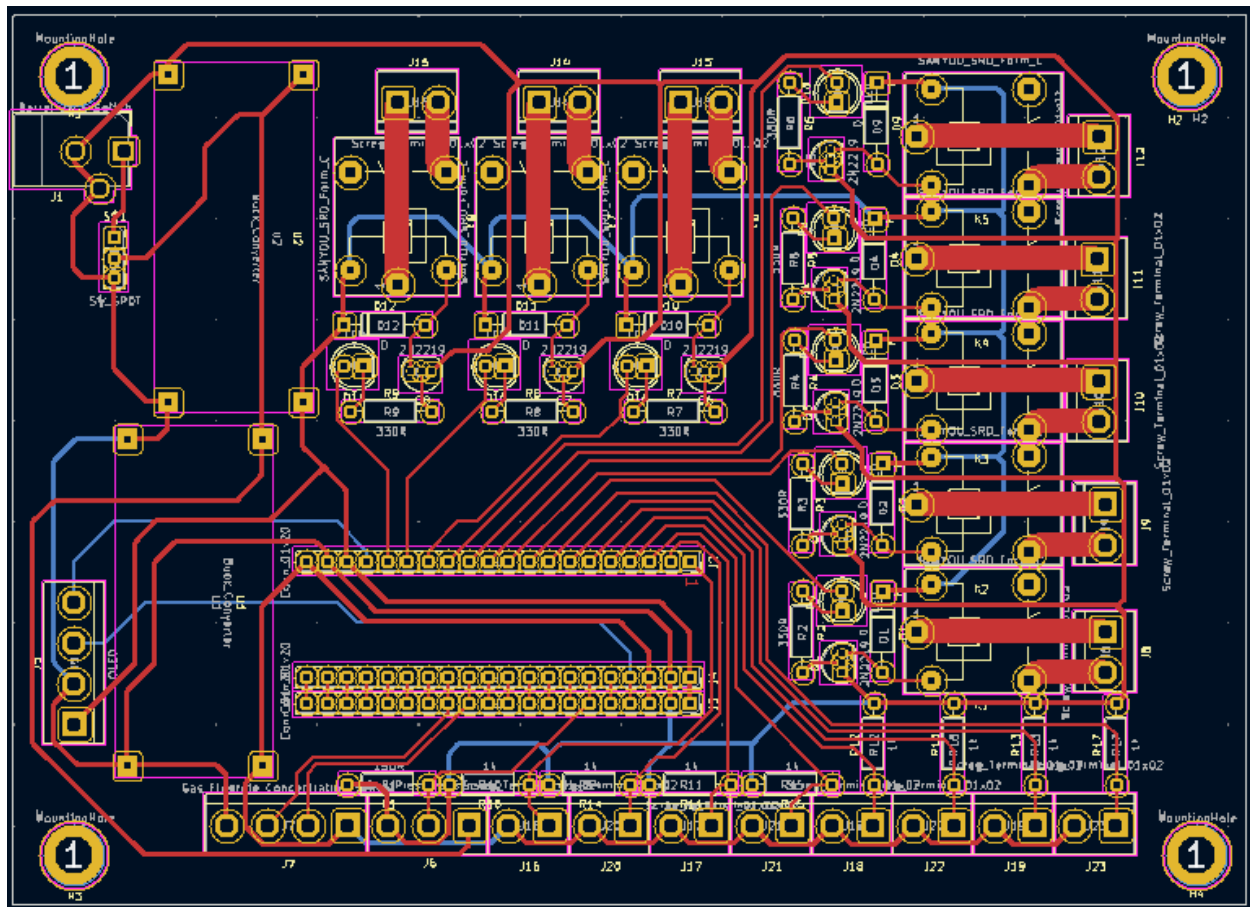
Figure 3.10 PCB Layout

3.2 SOFTWARE DESIGN

The operation of the control system is as follows depicted by the flowchart below

Gas flowrate, concentration, temperature and pressure regulation function


```
START
```

```
gas_flowrate = read_flowrate()
gas_concentration = read_concentration()
gas_temperature = read_temperature()
gas_pressure = read_pressure()
```

```
if gas_pressure < max_gas_pressure
&& gas_pressure >= min_gas_pressure
&& gas_flowrate < max_gas_flowrate
&& gas_concentration >= min_gas_concentration
&& gas_temperature < max_gas_temperature
&& gas_temperature >= min_gas_temperature
```

```
Display system status
```

```
if electrode_status == OFF
```

```
power on electrodes
```

```
if electrode_status == ON
```

```
power off electrodes
```

Figure 3.11 Electrolysis system operation flowchart

26

Automating the electrolysis process involves controlling the continuation or cessation of the process based on specific operational parameters, which include:

1. Electrolyte level in the electrolysis chamber
2. Temperature of the gases
3. Hydrogen and oxygen concentration level
4. Flow rate of the gases
5. Gas pressure in the storage area
6. Rate of change of these parameters

When the control system is activated, it checks if the electrolyte levels in all electrolysis chambers are at their maximum. If any chamber's level is below maximum, the corresponding solenoid valve opens to allow electrolyte to flow from the source. Once the electrolyte reaches the maximum level, float switches send this information to the microcontroller unit, which then closes the solenoid valves.

Next, the system reads all gas parameters from the sensors. It then checks if gas parameters are all within threshold. Gas pressure, concentration and pressure have minimum readings which serve as a differentiation of normal peripheral device operation and fault of the peripheral device.

Upon valid gas parameters the system will check if the electrodes are off and turns them on. Otherwise the system displays the cause of fault and execution loops to the beginning.

Upon invalid gas parameters the system will check if the electrodes are on and turns them off. Otherwise operation loops to the beginning.

The system will display its status on the display and continue execution from the beginning of the loop.

After startup, the system continuously monitors its operation against established safe operation limits. These limits, which can be fine-tuned through stress testing, are essential for ensuring safe operation. The current safe operation limits for the first prototype are:

1. Maximum startup time to achieve optimal parameters: 3 minutes
2. Minimum electrolyte quantity per chamber: determined by float switch position
3. Maximum gas temperature: 70°C
4. Minimum gas concentration after startup: 80%
5. Gas flow rate: 30 L/min (20 L/min hydrogen, 10 L/min oxygen)
6. Maximum gas pressure: 10 bar
7. Maximum allowed rate of change for each parameter on continuous operation: 30%

These limits enable the control system to effectively decide whether to continue or halt the electrolysis process.

4   CHAPTER FOUR – IMPLEMENTATION

The PCB design of the system was etched in the lab.
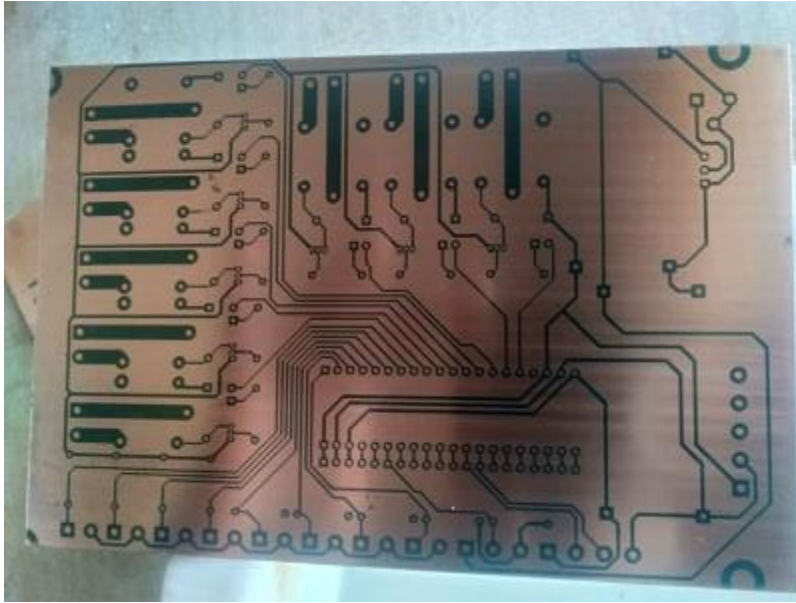
Figure 4.1 PCB After etching process

A portion of the components were soldered onto the board and preliminary tests were done. These test included:

1. Power test
2. Microcontroller boot test and running of test program

Upon valid preliminary test results, full component assembly of the board was done and test with the electrolysis chamber was done. Below are images showing the control system and the electrolysis chambers
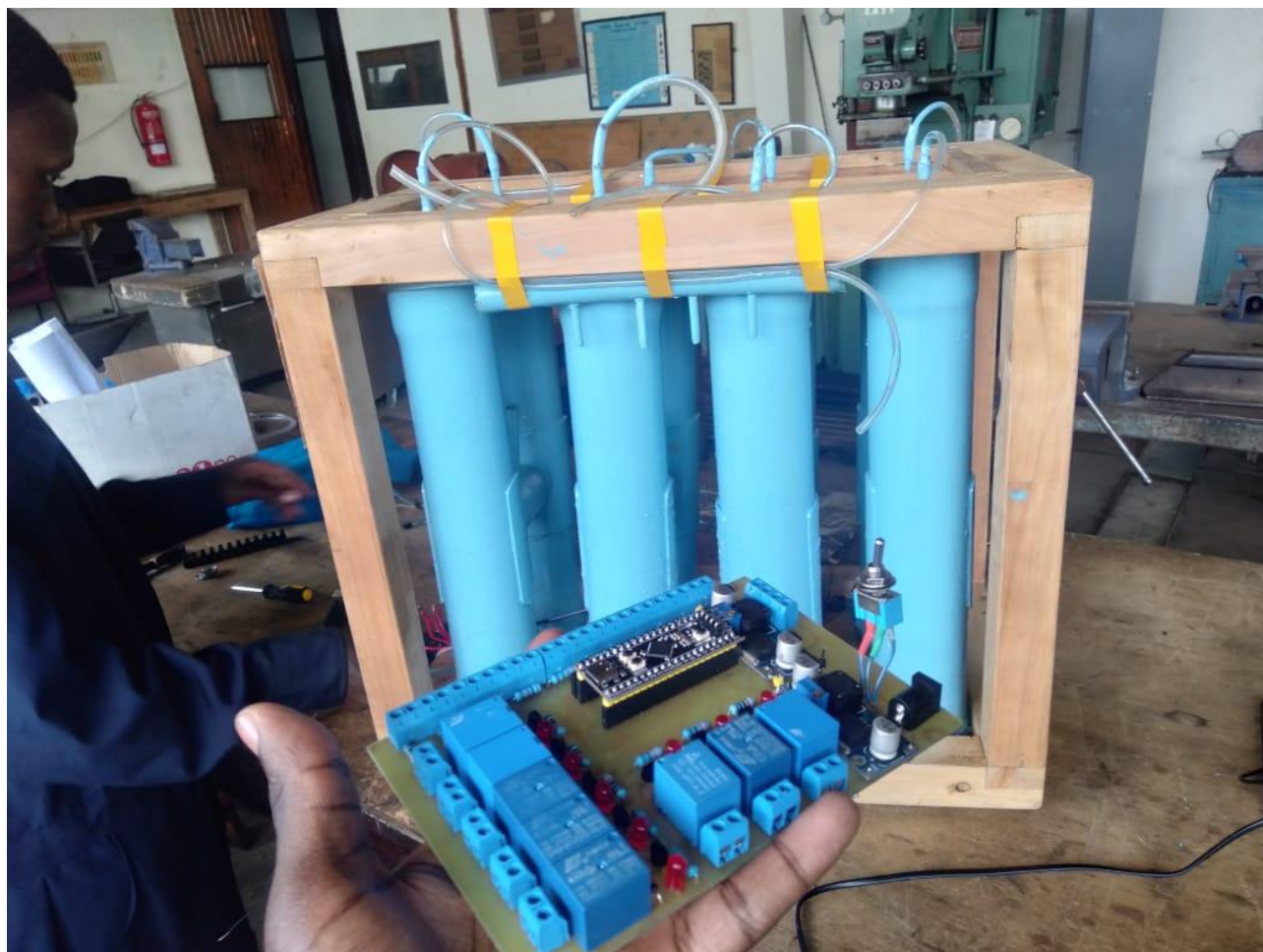
Figure 4.2 Electrolysis control system and the electrolysis chambers

Figure 4.3 Electrolysis system with connection to chamber electrodes

Figure 4.4 Electrolysis system with adjacent power supply used for test

Below is the library that handles gas parameters acquisition as well as control of the electrolysis system

```c
/*
 * gas.c
 *
 *  Created on: Mar 23, 2024
 *      Author: Donatus
 */



#include "main.h"
#include "gas.h"
#include "ssd1306_funcs.h"
#include "stdio.h"
#include "ssd1306.h"
#include <string.h>

int gas_counter = 0;
int gas_counter2 = 0;
uint8_t gasRxBuffer[GASRXBUFSIZE];
uint32_t gasPressureBitBuffer[1];
GAS_t gas1;
char oled_buf2[200];

int check_buf(void)
{
	for (int i = 0; i < GASRXBUFSIZE; i++)
	{
		if ((gasRxBuffer[i] == 0x16) && (gasRxBuffer[i + 1] == 0x09) &&
(gasRxBuffer[i + 2] == 0x01))
			return 1;
		if (i > 0)
			break;
	}

	return 0;
}


void get_oxygen_params(void)
{
	gas_counter = 0;
	gas_counter2 = 3;

	if (check_buf())
	{
		while (gas_counter < 3)
		{
			gas1.gas_params[gas_counter] = (gasRxBuffer[gas_counter2 +
gas_counter] * 256 + gasRxBuffer[(++gas_counter2) + gas_counter])/10.0;
```

```c
                        gas_counter++;
                }
        }

        else
        {
                while (gas_counter < 3)
                {
                        gas1.gas_params[gas_counter] = 0;
                        gas_counter++;
                }
        }
        gas_bit_to_bar();
}


void gas_bit_to_bar(void)
{
        /* pressure sensor output is 4-20mA and takes input of 10-30VDC
         * pressure sensor can measure 0-16bar
         * atmospheric pressure is 1.01325bar
         * 4mA is 1.01325bar and we can say 20mA is 16bar
         * We can assume the pressure-current relationship is linear and use a
formula
         * y = mx + c
         * m = (20-4)/(16-1) = (1/3)[mA/bar]
         * with a defined resistance r, m can be (r/3)[mV/bar]
         * voltage read from the drop across the resistor can be converted to
pressure in bar
         * x = y/m
         * adc is 12bit thus has values from 0 to 4096
         * adc max input voltage is 3.3V thus 0int=0V and 4096int=3.3V
         * adc int to voltage -> voltage = int * 3.3/4096
         * Using a resistance r:
         *     0bar is 4ma*r -> 4ma * 150R = 0.6V
         *     16bar is 20ma*r -> 20ma * 150R = 3V
         */
        HAL_ADC_Start(&hadc1);
        HAL_ADC_PollForConversion(&hadc1, 10);
        *gasPressureBitBuffer = HAL_ADC_GetValue(&hadc1);
        gas1.gas_pressure = ((*gasPressureBitBuffer) * (3.3 / 4096.0)) /
(PRESSURE_R_DROP / 3.0);
}

void power_electrodes(int power_direction, int *electrode_power_status)
{
        HAL_GPIO_WritePin(Electrode1_Output_GPIO_Port, Electrode1_Output_Pin,
power_direction);
        HAL_GPIO_WritePin(Electrode2_Output_GPIO_Port, Electrode2_Output_Pin,
power_direction);
        HAL_GPIO_WritePin(Electrode3_Output_GPIO_Port, Electrode3_Output_Pin,
power_direction);
```

```c
        HAL_GPIO_WritePin(Electrode4_Output_GPIO_Port, Electrode4_Output_Pin,
power_direction);
        *electrode_power_status = 1;
}

void manage_chambers(void)
{
        if (gas1.gas_pressure >= PRESSURE_THRESH && *electrode_power_status)
        {
                // stop operation as gas is full in reservoir
                display_message_overwrite("Pressure limit reached");
                power_electrodes(0, electrode_power_status);
                return;
        }

        if (gas1.gas_concentration < CONCENTRATION_THRESH &&
*electrode_power_status)
        {
                // possible leakage
                display_message_overwrite("Possible GAS leakage");
                power_electrodes(0, electrode_power_status);
                return;
        }

        if (gas1.gas_temperature >= TEMPERATURE_THRESH && *electrode_power_status)
        {
                // possible overheating
                display_message_overwrite("Possible Overheating");
                power_electrodes(0, electrode_power_status);
                return;
        }

        if (gas1.gas_pressure < PRESSURE_THRESH &&
                        gas1.gas_temperature < TEMPERATURE_THRESH &&
                        !(*electrode_power_status))
        {
                // gas parameters within threshold
                display_message_overwrite("Powering Electrodes");
                power_electrodes(1, electrode_power_status);
        }

}

void display_gas_parameters(void)
{
        ssd1306_Fill(Black);
        ssd1306_SetCursor(0, 0);
        memset(oled_buf2, 0, sizeof(oled_buf2));
        sprintf(oled_buf2, "P: %.4fbar", gas1.gas_pressure);
        ssd1306_WriteString(oled_buf2, Font_7x10, White);

        ssd1306_SetCursor(0, 10);
        memset(oled_buf2, 0, sizeof(oled_buf2));
```

```c
        sprintf(oled_buf2, "Conc: %.2f%%", gas1.gas_concentration);
        ssd1306_WriteString(oled_buf2, Font_7x10, White);
        ssd1306_SetCursor(0, 20);
        memset(oled_buf2, 0, sizeof(oled_buf2));
        sprintf(oled_buf2, "Flow: %.2fL/min", gas1.gas_flowrate);
        ssd1306_WriteString(oled_buf2, Font_7x10, White);
        ssd1306_SetCursor(0, 30);
        memset(oled_buf2, 0, sizeof(oled_buf2));
        sprintf(oled_buf2, "Temp: %.2fdegC", gas1.gas_temperature);
        ssd1306_WriteString(oled_buf2, Font_7x10, White);
        ssd1306_UpdateScreen();
}
```

5 CHAPTER FIVE – RESULTS AND ANALYSIS

The following system operations were as expected:

1. Measurement of gas parameters
2. Displaying of system status and cause of system fault whenever fault occurs
3. Powering off and on of the electrode relays based on control thresholds

Gas production peak was under 5ml/min of oxygen and under 10ml/min of hydrogen. This was well below our target of 5L/min of oxygen and 10L/min of hydrogen. As such, this flowrate was below the resolution of the flowrate sensor and we resorted to quantify it in bubbles per minute through over water method of gas collection.

The following table shows amount of electrolyte, input power and resulting flowrate in bubbles per minute

Table 1 Amount of Electrolyte, Input power and resulting Flowrate

Amount of electrolyte (g)	Voltage(V)	Current(A)	Flowrate(Bubbles per minute)
10	20	6	4
20	20	6.2	6
30	20	6.2	7
40	20	6.4	9
50	20	6.5	11
60	20	6.7	12
70	20	7.1	16

The following table shows the increase in oxygen concentration over time from starting of the electrolysis process

Time(minutes)	Voltage(V)	Current(A)	Oxygen Concentration(%)
1	24	8	25
3	24	8.4	34
6	24	9.0	48
9	24	9.7	62
12	24	10.2	83
15	24	11.1	95
18	24	11.6	95

Figure 5.1 Gas Concentration after 5 minutes
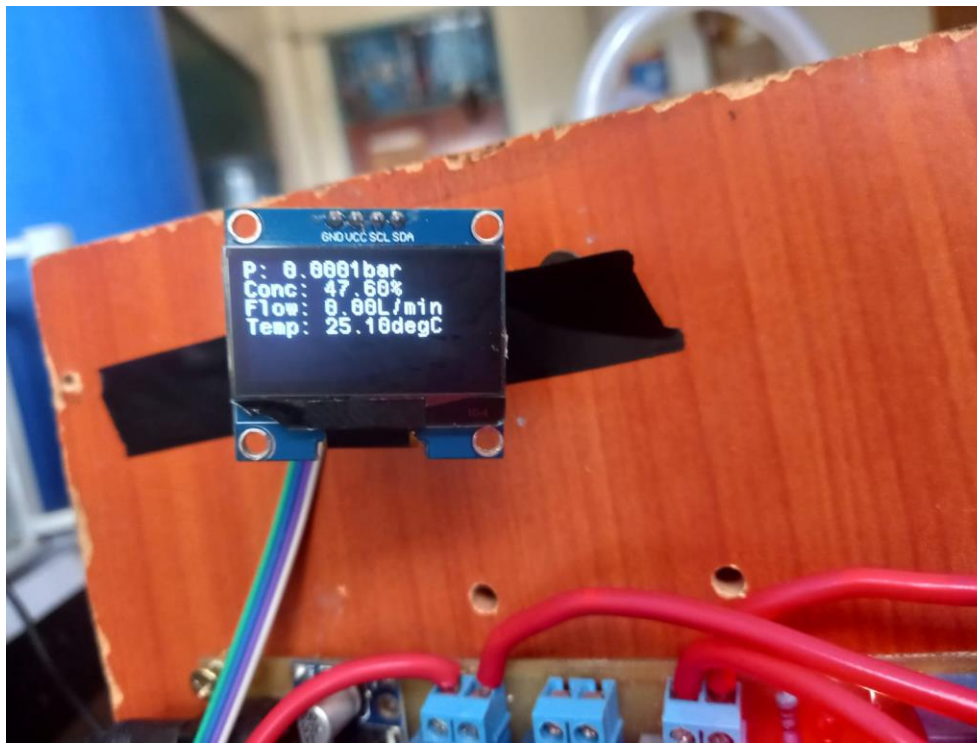
Figure 5.2 Gas Concentration after 20 minutes

Oxygen gas concentration measurement was up to 95.60%.

The electrolysis chambers experienced a lot of electrolyte and gas leaks leading to inaccurate quantification of results.

Furthermore, the electrode separation was large leading to high power consumption of the system.

Gas pressure readings were small due to low gas flowrate.

6 CHAPTER SIX – CONCLUSION AND RECOMMENDATIONS

Design and implementation of an automated control system was a success.

The system was able to measure gas parameters and control the system based on readings obtained from the gas sensors.

The system was able to display its state to inform an operator of the state of the system.

However, it was noted that the electrolysis chamber design was a key determiner of the success of the project in obtaining target results and should be heavily focused on in future revisions of this project

The major chamber design aspect to be corrected was noted to be making the electrode spacing as small as possible

Current areas of improvement of the current system include:

1. Board design

The system as it is does not realise the full potential of the microcontroller unit it uses. This is because of the following:

- Interfaces of the microcontroller unit that can perform complex operations have been used up by components that require basic operation. Relays and float switches require basic operation and such an operation can be delegated to a simple IC which can be controlled by the main microcontroller. A good board design will cater for such. With such an improvement more gas pressure sensors and gas flow rate sensor interfaces can be operated by the microcontroller unit
- Currently the only way the board can inform a plant operator is by the use of the display. This is not the best way of conveying information. Systems that provide automation functions are best equipped with components that allow connectivity through GSM, GPRS or Wi-Fi. A good board design will cater for such.

2. Over the Air Updates (OTA)

Currently, the system lacks a GSM/GPRS or Wi-Fi module that allows connectivity. This is a major drawback as it makes such a system lack over the air updates.

When provisioning thousands of such systems, firmware improvements of the systems will not be possible manually as the systems will be in different locations. Furthermore, manual intervention also will pose the system to risk due to errors made by manual operations. A good board design will cater for such.

REFERENCES

Baker, R. C. (2005). *Flow Measurement Handbook: Industrial Designs, Operating Principles, Performance, and Applications.* Cambridge University Press.

Balat, M. (2008). Potential importance of hydrogen as a future solution to environmental and transportation problems. *International Journal of Hydrogen Energy, 33*(15), 4013-4029.

Ballato, A. (1996). *Piezoelectricity: Evolution and Future of a Technology.* Springer.

Bard, A. J. (2001). *Electrochemical Methods: Fundamentals and Applications.* Wiley.

Beckwith, T. G. (2011). *Mechanical Measurements.* Pearson.

Bentley, R. E. (1998). *Handbook of Temperature Measurement.* Springer.

Cabrera, S. &. (2006). Principles and applications of non-dispersive infrared (NDIR) gas sensors. *Sensors, 6*(6), 567-592.

Carmo, M. F. (2013). A comprehensive review on PEM water electrolysis. *International Journal of Hydrogen Energy, 38*(12), 4901-4934.

Daiichi Nekken CO., L. (n.d.). *Ultrasonic gas analyzer*. Retrieved from The principle of the measurement by the ultrasonic gas analyser: https://www.us-analyzer.com/english/#:~:text=Ultrasonic%20waves%20pass%20through%20the,two%20known%20gases%20is%20found

Dally, J. W. (1993). *Dally, J. W., & Riley, W. F.* McGraw-Hill.

Dick, C. (2008). Measurement of gas concentration by infrared absorption spectroscopy. *Journal of Applied Spectroscopy, 75*(4), 550-558.

Fiorenza, G. S. (2019). Hydrogen production: state of the art and future perspective. *International Journal of Hydrogen Energy, 44*(23), 11744-11773.

Fraden, J. (2016). *Handbook of Modern Sensors: Physics, Designs, and Applications.* Springer.

Gahleitner, G. (2013). Hydrogen from renewable electricity: An international review of power-to-gas pilot plants for stationary applications. *International Journal of Hydrogen Energy, 38*(5), 2039-2061.

Giddey, S. B. (2012). Renewable hydrogen for sustainable energy systems. *International Journal of Hydrogen Energy, 38*(2), 1742-1749.

Holton, O. T. (2013). The role of platinum in proton exchange membrane fuel cells evaluation of platinum's unique properties for use in both the anode and cathode of a proton exchange membrane fuel cell. In O. T. Holton, *Journal of Power Sources* (pp. 222,122-134).

Kalantar-Zadeh, K. &. (2008). Nanotechnology-enabled sensors: Advances and applications. In K. &. Kalantar-Zadeh, *Nanotechnology-enabled sensors* (pp. 1363-1370). ACS Nano.

Khayati, G. R. (2013). Industrial hydrogen production processes and their comparison. *International Journal of Hydrogen Energy, 38*(12), 5233-5237.

Kiros, Y. &. (2008). Electrolysis of water with a new membrane cell using bipolar electrodes. *International Journal of Hydrogen Energy, 33*(2), 221-226.

Korotcenkov, G. (2005). Metal oxides for solid-state gas sensors: What determines our choice? *Materials Science and Engineering: B, 139*(1), 1-23.

Kothari, R. B. (2008). Comparison of environmental and economic aspects of various hydrogen production methods. *Renewable and Sustainable Energy Reviews, 12*(2), 553-563.

Kumar, S. S. (2019). Hydrogen production by PEM water electrolysis – A review. *Materials Science for Energy Technologies, 2*(3), 442-454.

Laguna-Bercero, M. A. (n.d.). Recent advances in high temperature electrolysis using solid oxide fuel cells: A review. *Journal of Power Sources, 203*, 4-16.

Lipták, B. G. (2003). *Instrument Engineers' Handbook: Process Measurement and Analysis.* CRC Press.

McMurry, J. E. (2007). *Chemistry.* Pearson Prentice Hall.

Mills, G. &. (2014). The effects of temperature on the performance of water electrolyzers. *Journal of Electroanalytical Chemistry, 731*, 102-108.

Momirlan, M. &. (2005). The properties of hydrogen as fuel for tomorrow in sustainable energy system for a cleaner planet. *International Journal of Hydrogen Energy, 30*(7), 795-802.

Moseley, P. T. (2017). Solid state gas sensors. *Measurement Science and Technology, 8*(3), 223-237.

Patel, Y. K. (2016). Recent advances in the hydrogen production by water electrolysis. *Journal of Renewable and Sustainable Energy, 8*(3), 702.

Perry, R. &. (2008). *Perry's Chemical Engineers' Handbook.* McGraw-Hill.

Petersen, J. E. (2013). *Handbook of Mass Measurement.* CRC Press.

Ruthven, D. F. (1994). *Pressure Swing Adsorption.* VCH Publishers.

Santos, M. B. (2006). Comparative analysis of the accuracy of Coriolis flow meters. *Flow Measurement and Instrumentation, 17*(1), 1-8.

Sircar, S. &. (2000). Pressure Swing Adsorption Technology for Hydrogen Production. *Gas Separation & Purification*, 77-85.

Stetter, J. R. (2008). Amperometric gas sensors: A review. *Chemical Reviews, 108*(2), 352-366.

Sze, S. M. (2008). *Semiconductor Sensors.* Wiley.

APPENDICES

APPENDIX A – COST ANALYSIS

Table 3 Bill Of Quantites

Item Number	Item Name	Quantity	Unit Price	Total (KES)
1	2N2219 BJT Transistor	8	5	40
2	330 Ohm 0.5W resistor	8	5	40
3	SPDT Relay	8	50	400
4	1x02 Screw Terminal Block	16	20	320
5	Barrel Jack Connector	1	30	30
6	Red LEDs	8	5	40
7	1k Ohm 0.5W resistor	8	5	40
8	1x20 Female Pinheader	3	40	120
9	150 Ohm 0.5W resistor	1	5	5
10	1x03 Female Pinheader	1	40	40
11	Zener diode 13V breakdown	8	10	80
12	LM2596 Voltage regulator module	2	200	400
13	1x04 Screw Terminal Block	2	30	60
14	1x03 Screw Terminal Block	1	30	30
15	Gas Pressure Sensor	2	3100	6200
16	Gas Flowrate, Concentration & Temperature Sensor	2	6300	12600
17	Float Switch	8	1000	8000
18	Solenoid Valve	8	1200	9600
19	STM32F411CEU6 Development Board	1	1500	1500

20	ST-Link V2 Programmer	1	700	700
21	Presensitised PCB	1	550	550
22	2.5mm^2 connecting wire	30	30	900
23	OLED Screen 1.54"	1	1000	1000
Total				42695

APPENDIX B – CODE

MAIN.C

```c
/* USER CODE BEGIN Header */
/**
  ******************************************************************************
  * @file           : main.c
  * @brief          : Main program body
  ******************************************************************************
  * @attention
  *
  * Copyright (c) 2024 STMicroelectronics.
  * All rights reserved.
  *
  * This software is licensed under terms that can be found in the LICENSE file
  * in the root directory of this software component.
  * If no LICENSE file comes with this software, it is provided AS-IS.
  *
  ******************************************************************************
  */
/* USER CODE END Header */
/* Includes ------------------------------------------------------------------*/
#include "main.h"

/* Private includes ----------------------------------------------------------*/
/* USER CODE BEGIN Includes */
#include "stdio.h"
#include "ssd1306.h"
#include "ssd1306_funcs.h"
/* GAS CODE BEGIN Includes */
#include "gas.h"
#include <string.h>
/* GAS CODE END Includes */

/* water CODE BEGIN Includes */
#include "water.h"
/* water CODE END Includes */
```

45

```c
/* USER CODE END Includes */

/* Private typedef -----------------------------------------------------------*/
/* USER CODE BEGIN PTD */

/* USER CODE END PTD */

/* Private define ------------------------------------------------------------*/
/* USER CODE BEGIN PD */

/* USER CODE END PD */

/* Private macro -------------------------------------------------------------*/
/* USER CODE BEGIN PM */

/* USER CODE END PM */

/* Private variables ---------------------------------------------------------*/
ADC_HandleTypeDef hadc1;

I2C_HandleTypeDef hi2c2;

SPI_HandleTypeDef hspi1;

UART_HandleTypeDef huart2;
DMA_HandleTypeDef hdma_usart2_rx;

/* USER CODE BEGIN PV */

/* GAS CODE BEGIN PV */

/* GAS CODE END PV */

/* USER CODE END PV */

/* Private function prototypes -----------------------------------------------*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_DMA_Init(void);
static void MX_USART2_UART_Init(void);
static void MX_ADC1_Init(void);
static void MX_I2C2_Init(void);
static void MX_SPI1_Init(void);
/* USER CODE BEGIN PFP */

/* GAS CODE BEGIN PFP */
void HAL_UARTEx_RxEventCallback(UART_HandleTypeDef *huart, uint16_t buf_size);
/* GAS CODE END PFP */

/* USER CODE END PFP */

/* Private user code ---------------------------------------------------------*/
```

```c
/* USER CODE BEGIN 0 */

/* GAS CODE BEGIN 0 */
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
       if (huart->Instance == USART2)
       {
               HAL_UART_Receive_DMA(huart, gasRxBuffer, GASRXBUFSIZE);
               __HAL_DMA_DISABLE_IT(&hdma_usart2_rx, DMA_IT_HT);
       }
}
/* GAS CODE END 0 */

/* USER CODE END 0 */

/**
  * @brief  The application entry point.
  * @retval int
  */
int main(void)
{

  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */

  /* MCU Configuration--------------------------------------------------------*/

  /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
  HAL_Init();

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */

  /* Configure the system clock */
  SystemClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_DMA_Init();
  MX_USART2_UART_Init();
  MX_ADC1_Init();
  MX_I2C2_Init();
  MX_SPI1_Init();
  /* USER CODE BEGIN 2 */
       HAL_Delay(1500);
       ssd1306_Init();
       /* GAS CODE BEGIN 2 */
```

```c
        HAL_UART_Receive_DMA(&huart2, gasRxBuffer, GASRXBUFSIZE);
        __HAL_DMA_DISABLE_IT(&hdma_usart2_rx, DMA_IT_HT);
        /* GAS CODE END 2 */
        /* WATER CODE BEGIN 2 */
//      waterInitialization();
        power_electrodes(1, electrode_power_status);
        /* WATER CODE END 2 */
  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
      while (1)
      {
            /* GAS CODE BEGIN WHILE */
            waterManagement();
            get_oxygen_params();
            manage_chambers();
            display_gas_parameters();
            HAL_Delay(100);
            /* GAS CODE END WHILE */

    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
      }
  /* USER CODE END 3 */
}

/**
  * @brief System Clock Configuration
  * @retval None
  */
void SystemClock_Config(void)
{
  RCC_OscInitTypeDef RCC_OscInitStruct = {0};
  RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

  /** Configure the main internal regulator output voltage
  */
  __HAL_RCC_PWR_CLK_ENABLE();
  __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);

  /** Initializes the RCC Oscillators according to the specified parameters
  * in the RCC_OscInitTypeDef structure.
  */
  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
  RCC_OscInitStruct.HSEState = RCC_HSE_ON;
  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
  RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
  RCC_OscInitStruct.PLL.PLLM = 25;
  RCC_OscInitStruct.PLL.PLLN = 192;
  RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
  RCC_OscInitStruct.PLL.PLLQ = 4;
```

```c
  if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
  {
    Error_Handler();
  }

  /** Initializes the CPU, AHB and APB buses clocks
  */
  RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                              |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
  RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
  RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
  RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
  RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

  if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_3) != HAL_OK)
  {
    Error_Handler();
  }
}

/**
  * @brief ADC1 Initialization Function
  * @param None
  * @retval None
  */
static void MX_ADC1_Init(void)
{

  /* USER CODE BEGIN ADC1_Init 0 */

  /* USER CODE END ADC1_Init 0 */

  ADC_ChannelConfTypeDef sConfig = {0};

  /* USER CODE BEGIN ADC1_Init 1 */

  /* USER CODE END ADC1_Init 1 */

  /** Configure the global features of the ADC (Clock, Resolution, Data Alignment
and number of conversion)
  */
  hadc1.Instance = ADC1;
  hadc1.Init.ClockPrescaler = ADC_CLOCK_SYNC_PCLK_DIV4;
  hadc1.Init.Resolution = ADC_RESOLUTION_12B;
  hadc1.Init.ScanConvMode = DISABLE;
  hadc1.Init.ContinuousConvMode = DISABLE;
  hadc1.Init.DiscontinuousConvMode = DISABLE;
  hadc1.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_NONE;
  hadc1.Init.ExternalTrigConv = ADC_SOFTWARE_START;
  hadc1.Init.DataAlign = ADC_DATAALIGN_RIGHT;
  hadc1.Init.NbrOfConversion = 1;
  hadc1.Init.DMAContinuousRequests = DISABLE;
  hadc1.Init.EOCSelection = ADC_EOC_SINGLE_CONV;
```

```c
  if (HAL_ADC_Init(&hadc1) != HAL_OK)
  {
    Error_Handler();
  }

  /** Configure for the selected ADC regular channel its corresponding rank in the
sequencer and its sample time.
  */
  sConfig.Channel = ADC_CHANNEL_9;
  sConfig.Rank = 1;
  sConfig.SamplingTime = ADC_SAMPLETIME_3CYCLES;
  if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK)
  {
    Error_Handler();
  }
  /* USER CODE BEGIN ADC1_Init 2 */

  /* USER CODE END ADC1_Init 2 */

}

/**
  * @brief I2C2 Initialization Function
  * @param None
  * @retval None
  */
static void MX_I2C2_Init(void)
{

  /* USER CODE BEGIN I2C2_Init 0 */

  /* USER CODE END I2C2_Init 0 */

  /* USER CODE BEGIN I2C2_Init 1 */

  /* USER CODE END I2C2_Init 1 */
  hi2c2.Instance = I2C2;
  hi2c2.Init.ClockSpeed = 400000;
  hi2c2.Init.DutyCycle = I2C_DUTYCYCLE_2;
  hi2c2.Init.OwnAddress1 = 0;
  hi2c2.Init.AddressingMode = I2C_ADDRESSINGMODE_7BIT;
  hi2c2.Init.DualAddressMode = I2C_DUALADDRESS_DISABLE;
  hi2c2.Init.OwnAddress2 = 0;
  hi2c2.Init.GeneralCallMode = I2C_GENERALCALL_DISABLE;
  hi2c2.Init.NoStretchMode = I2C_NOSTRETCH_DISABLE;
  if (HAL_I2C_Init(&hi2c2) != HAL_OK)
  {
    Error_Handler();
  }
  /* USER CODE BEGIN I2C2_Init 2 */

  /* USER CODE END I2C2_Init 2 */
```

```c
}

/**
  * @brief SPI1 Initialization Function
  * @param None
  * @retval None
  */
static void MX_SPI1_Init(void)
{

  /* USER CODE BEGIN SPI1_Init 0 */

  /* USER CODE END SPI1_Init 0 */

  /* USER CODE BEGIN SPI1_Init 1 */

  /* USER CODE END SPI1_Init 1 */
  /* SPI1 parameter configuration*/
  hspi1.Instance = SPI1;
  hspi1.Init.Mode = SPI_MODE_SLAVE;
  hspi1.Init.Direction = SPI_DIRECTION_2LINES;
  hspi1.Init.DataSize = SPI_DATASIZE_8BIT;
  hspi1.Init.CLKPolarity = SPI_POLARITY_LOW;
  hspi1.Init.CLKPhase = SPI_PHASE_1EDGE;
  hspi1.Init.NSS = SPI_NSS_HARD_INPUT;
  hspi1.Init.FirstBit = SPI_FIRSTBIT_MSB;
  hspi1.Init.TIMode = SPI_TIMODE_DISABLE;
  hspi1.Init.CRCCalculation = SPI_CRCCALCULATION_DISABLE;
  hspi1.Init.CRCPolynomial = 10;
  if (HAL_SPI_Init(&hspi1) != HAL_OK)
  {
    Error_Handler();
  }
  /* USER CODE BEGIN SPI1_Init 2 */

  /* USER CODE END SPI1_Init 2 */

}

/**
  * @brief USART2 Initialization Function
  * @param None
  * @retval None
  */
static void MX_USART2_UART_Init(void)
{

  /* USER CODE BEGIN USART2_Init 0 */

  /* USER CODE END USART2_Init 0 */

  /* USER CODE BEGIN USART2_Init 1 */
```

```c
  /* USER CODE END USART2_Init 1 */
  huart2.Instance = USART2;
  huart2.Init.BaudRate = 9600;
  huart2.Init.WordLength = UART_WORDLENGTH_8B;
  huart2.Init.StopBits = UART_STOPBITS_1;
  huart2.Init.Parity = UART_PARITY_NONE;
  huart2.Init.Mode = UART_MODE_TX_RX;
  huart2.Init.HwFlowCtl = UART_HWCONTROL_NONE;
  huart2.Init.OverSampling = UART_OVERSAMPLING_16;
  if (HAL_UART_Init(&huart2) != HAL_OK)
  {
    Error_Handler();
  }
  /* USER CODE BEGIN USART2_Init 2 */

  /* USER CODE END USART2_Init 2 */

}

/**
  * Enable DMA controller clock
  */
static void MX_DMA_Init(void)
{

  /* DMA controller clock enable */
  __HAL_RCC_DMA1_CLK_ENABLE();

  /* DMA interrupt init */
  /* DMA1_Stream5_IRQn interrupt configuration */
  HAL_NVIC_SetPriority(DMA1_Stream5_IRQn, 0, 0);
  HAL_NVIC_EnableIRQ(DMA1_Stream5_IRQn);

}

/**
  * @brief GPIO Initialization Function
  * @param None
  * @retval None
  */
static void MX_GPIO_Init(void)
{
  GPIO_InitTypeDef GPIO_InitStruct = {0};
/* USER CODE BEGIN MX_GPIO_Init_1 */
/* USER CODE END MX_GPIO_Init_1 */

  /* GPIO Ports Clock Enable */
  __HAL_RCC_GPIOH_CLK_ENABLE();
  __HAL_RCC_GPIOA_CLK_ENABLE();
  __HAL_RCC_GPIOB_CLK_ENABLE();

  /*Configure GPIO pin Output Level */
```

```
  HAL_GPIO_WritePin(GPIOA, Electrode1_Output_Pin|Electrode2_Output_Pin,
GPIO_PIN_RESET);

  /*Configure GPIO pin Output Level */
  HAL_GPIO_WritePin(GPIOB,
Electrode3_Output_Pin|Electrode4_Output_Pin|Solenoid1_Output_Pin|Solenoid2_Output_P
in
                        |Solenoid3_Output_Pin|Solenoid4_Output_Pin,
GPIO_PIN_RESET);

  /*Configure GPIO pins : Float_Chamber1_min_Pin Float_Chamber1_max_Pin
Float_Chamber2_min_Pin Float_Chamber2_max_Pin */
  GPIO_InitStruct.Pin =
Float_Chamber1_min_Pin|Float_Chamber1_max_Pin|Float_Chamber2_min_Pin|Float_Chamber2
_max_Pin;
  GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

  /*Configure GPIO pins : Float_Chamber3_min_Pin Float_Chamber3_max_Pin
Float_Chamber4_min_Pin Float_Chamber4_max_Pin */
  GPIO_InitStruct.Pin =
Float_Chamber3_min_Pin|Float_Chamber3_max_Pin|Float_Chamber4_min_Pin|Float_Chamber4
_max_Pin;
  GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

  /*Configure GPIO pins : Electrode1_Output_Pin Electrode2_Output_Pin */
  GPIO_InitStruct.Pin = Electrode1_Output_Pin|Electrode2_Output_Pin;
  GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

  /*Configure GPIO pins : Electrode3_Output_Pin Electrode4_Output_Pin
Solenoid1_Output_Pin Solenoid2_Output_Pin
                        Solenoid3_Output_Pin Solenoid4_Output_Pin */
  GPIO_InitStruct.Pin =
Electrode3_Output_Pin|Electrode4_Output_Pin|Solenoid1_Output_Pin|Solenoid2_Output_P
in
                        |Solenoid3_Output_Pin|Solenoid4_Output_Pin;
  GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

/* USER CODE BEGIN MX_GPIO_Init_2 */
/* USER CODE END MX_GPIO_Init_2 */
}

/* USER CODE BEGIN 4 */
```

53

```
/* USER CODE END 4 */

/**
  * @brief  This function is executed in case of error occurrence.
  * @retval None
  */
void Error_Handler(void)
{
  /* USER CODE BEGIN Error_Handler_Debug */
      /* User can add his own implementation to report the HAL error return state
*/
      __disable_irq();
      while (1)
      {
      }
  /* USER CODE END Error_Handler_Debug */
}

#ifdef  USE_FULL_ASSERT
/**
  * @brief  Reports the name of the source file and the source line number
  *         where the assert_param error has occurred.
  * @param  file: pointer to the source file name
  * @param  line: assert_param error line source number
  * @retval None
  */
void assert_failed(uint8_t *file, uint32_t line)
{
  /* USER CODE BEGIN 6 */
      /* User can add his own implementation to report the file name and line
number,
    ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
  /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */
```

APPENDIX C – GITHUB REPOSITORY

https://github.com/Donatussss/Final_year_project_undergrad