Instituto Politécnico de Tomar

Escola Superior de Tecnologia de Tomar

OTIMIZAÇÃO E MONITORIZAÇÃO DE PROTÓTIPO PARA PRODUÇÃO DE H_2 A PARTIR DE ENERGIA RENOVÁVEL

OPTIMIZATION AND MONITORING OF A PROTOTYPE FOR H_2 PRODUCTION FROM RENEWABLE ENERGY

Projeto

Vasim Tana

Mestrado em Engenharia Eletrotécnica

Tomar/Novembro/2024

Instituto Politécnico de Tomar

Escola Superior de Tecnologia de Tomar

Vasim Tana

OTIMIZAÇÃO E MONITORIZAÇÃO DE PROTÓTIPO PARA PRODUÇÃO DE H_2 A PARTIR DE ENERGIA RENOVÁVEL

OPTIMIZATION AND MONITORING OF A PROTOTYPE FOR H_2 PRODUCTION FROM RENEWABLE ENERGY

Projeto

Orientado por:

Prof. Paulo Coelho, Instituto Politécnico de Tomar
Prof. Henrique Pinho, Instituto Politécnico de Tomar

Projeto apresentada ao Instituto Politécnico de Tomar para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Eletrotécnica

ABSTRACT

This report presents the development and implementation of an electrolysis-based hydrogen production prototype powered by renewable energy sources, specifically solar energy. The prototype integrates a Proton Exchange Membrane (PEM) electrolyzer, control system, and real-time monitoring to optimize hydrogen production as well as ensuring safety and efficiency. The system includes an ESP32 microcontroller for control, a DC/DC buck converter for voltage regulation, and sensors for real-time data monitoring. Data storage and visualization are achieved by integrating InfluxDB and Grafana. The system permits the production of hydrogen either with or without control through a user-friendly IoT dashboard. This project shows a laboratory scale and sustainable approach to green hydrogen production, thus focusing on key challenges like process optimization and user interface. Although the system operates on an open-loop control and lacks hydrogen storage unit, this report identifies potential enhancements for future development, including closed-loop systems and hydrogen storage solutions.

Key words: Green Hydrogen, Renewable Energy, PEM water electrolysis, Control System, IoT Interface, DC/DC Converter

RESUMO

Este relatório apresenta o desenvolvimento e implementação de um protótipo de produção de hidrogénio uma produção de hidrogénio baseada em eletrólise alimentado por fontes de energia renováveis, nomeadamente energia solar. O protótipo integra um eletrolisador de membrana de troca de protões (PEM), sistema de controlo e monitorização em tempo real para otimizar a produção de hidrogénio, além de garantir segurança e eficiência. O sistema inclui um microcontrolador ESP32 para controlo, um conversor buck DC/DC para regulação de tensão e sensores para monitorização de dados em tempo real. O sistema permite produzir hidrogénio com ou sem controlo através de um painel IoT fácil de utilizar. Este projeto mostra uma abordagem sustentável e à escala laboratorial para a produção de hidrogénio verde, concentrando-se assim em desafios-chave como a otimização de processos e a interface do utilizador. Embora o sistema tenha controlo e não disponha de uma unidade de armazenamento de hidrogénio, este relatório identifica potenciais melhorias para desenvolvimento futuro, incluindo sistemas de controlo em malha fechada e soluções de armazenamento de hidrogénio.

Palavras-chave: Hidrogénio Verde, Energia Renovável, Eletrólise de Água PEM, Sistema de Controlo, Interface IoT, Conversor DC/DC

ACKNOWLEDGMENTS

As I have achieved my goals and objectives for this project, I wish to express my gratitude to both of my supervisors, Prof. Paulo Coelho, and Prof. Henrique Pinho for their guidance throughout the development of the project and I'd like to express my appreciation for their experience that was essential for achieving this work.

I would also like to thank the Polytechnic University of Tomar for providing the space and logistic support to conclude this project. I shall also point out the support of the FCT (Fundação para a Ciência e a Tecnologia) through the H2-REnWaste internal project under the grant UIDP/05567/2020 to the IPT's Smart Cities Research Center (Ci2).

I shall thank my teachers, friends, and colleges for their advice and suggestions along the path and I'd like to express my appreciation for the support of Prof. Francisco Nunes, Prof. Carlos Ferreira, Prof. José Fernandes and Eng. Pedro Neves who significantly helped with their feedback related to various critical points in the present project. Finally, my extreme gratitude to my family who was the source of motivation and support during my educational career abroad regardless of the long distance.

INDEX

INDEX OF FIGURES

INDEX OF TABLES

ACRONYMS

ACV	Adjustable Check Valve
ADC	Analog-to-Digital Converter
AE	Alkaline water Electrolysis
API	Application Programming Interface
ATSM	American Society for Testing and Materials
Ci2	Smart Cities Research Center
CO	Carbon Monoxide
CO_2	Carbon Dioxide
CPU	Central Processing Unit
DAC	Digital-to-Analog Converter
DC	Direct Current
DSP	Digital Signal Processor
FB	Feedback
GPIO	General Purpose Input/Output
HE	Heat Exchanger
HTTP	Hypertext Transfer Protocol
I2C	Inter-Integrated Circuit (I2C Protocol)
IDE	Integrated Development Environment
IoT	Internet of Things
IPT	Polytechnic University of Tomar
IR	Infrared
LCD	Liquid Crystal Display
LPG	Liquefied Petroleum Gas
MEC	Microbial Electrolysis Cells
MPPT	Maximum Power Point Tracking
NC	Normally Closed (electrical contacts)
NO	Normally Open (electrical contacts)
PCB	Printed Circuit Board
PEM	Proton Exchange Membrane
PS	Pressure Sensor
PV	Photovoltaic
PWM	Pulse Width Modulation
RP	Resin Patron
SOE	Solid Oxide Electrolysis
USB	Universal Serial Bus
Wi-Fi	Wireless Fidelity

UNITS

A	Ampere	Electric current (A)
Ah	Ampere-hour	Electric charge (Ah)
bar	Bar	Pressure (1 bar)
℃	Degrees Celsius	Temperature (°C)
$I_{MP\ P}$	Current at Maximum Power Point	Amperes (A)
I_{SC}	Short Circuit Current	Amperes (A)
I_{CC}	Charge Current	Amperes (A)
I_{DC}	Discharge Current	Amperes (A)
ml	Milliliter	Volume (mL)
ms	Millisecond	Time (ms)
NmL	Normal milliliters	Volume (mL)
ppm	Parts Per Million	Concentration (ppm)
t_{elap}	Elapsed Time	Time
t_{req}	Requested Time	Time
V	Voltage	Volts (V)
V_{MPP}	Voltage at Maximum Power Point	Volts (V)
V_{OC}	Open-Circuit Voltage	Volts (V)
V_{DC}	Nominal Voltage	Volts (V)
V_{MSI}	Maximum Solar Input Voltage	Volts (V)
V_{SV}	System Voltage	Volts (V)
V_{in}	Input Voltage	Volts (V)
V_{out}	Output Voltage	Volts (V)
V_{DAC}	Digital-to-Analog Converter Voltage	Volts (V)
$V_{DAC\ H}$	High DAC Voltage	Volts (V)
$V_{DAC\ L}$	Low DAC Voltage	Volts (V)
$V_{OUT\ H}$	High Output Voltage	Volts (V)
$V_{OUT\ L}$	Low Output Voltage	Volts (V)
V_{Ref}	Reference Voltage	Volts (V)
V_{req}	Requested Voltage	Volts (V)
W	Watt	Power (W)

1. INTRODUCTION

This report was formulated within the framework of the final project as a part of curriculum, in the second year of the master's degree course of Electrical Engineering, offered by the Polytechnic University of Tomar (IPT). The present project was conducted in collaboration with Smart Cities Research Center - Ci2, with the objective of optimization and monitoring of an integrated laboratory demonstration prototype designed and implemented to produce hydrogen from green energy (namely using photovoltaic energy). The prototype designed in the present work shall make it possible to understand the entire hydrogen production cycle from green energy sources.

Within the scope of the present project the aim is not only optimize the hydrogen production process but also provide a monitoring system for user convenience. The main objectives are:

- Optimizing the hydrogen production process by integration of sensors for monitoring;
- Control the hydrogen production process using an ESP32 microcontroller and enable remote operation through wireless connectivity;
- Enable data logging into database and develop a user-friendly interface allowing monitorization of the process.

1.1. Overview

Nowadays the increasing demand for sustainable energy solutions is one of the most significant challenges. As concerns about the environmental impact of fossil fuels and climate change are growing, renewable energy sources have gained relevance as a promising alternative to meet energy needs [1]. Hydrogen is particularly worthy as it is a good energy carrier which has the potential revolutionize the energy sector. Alongside being a fuel that can generate electricity in fuel cells, hydrogen also serves as a key component in various industrial processes which offers a carbon-free alternative when produced using renewable energy [2]. However, for hydrogen to be an energy source used

on a massive scale, it is necessary to improve the production methods that are effective, safe, and environmentally friendly.

Hydrogen generation from renewable sources, such as electrolysis, is attracting a lot of attention. However, such technology is still in the early phase and is costly compared to hydrogen generation from fossil fuels [3]. Electrolysis is a technique in which a chemical reaction is driven by a direct electric current between electrodes through a chemical solution with an objective to separate the elements. Water electrolysis results in the breaking of the chemical bonds present in water, thus separating oxygen and hydrogen, providing hydrogen gas which later can be used as clean fuel [4]. When the electricity used comes from renewable sources like solar or wind energy, the entire production cycle becomes a "green" process, with a final product namely green hydrogen, contributing to the global goal of reducing greenhouse gas emissions, figure 1 [5]. However, green hydrogen production faces various challenges, such as the associated cost from storage and transportation as well as expensive materials used to make the electrolyzer and the technology is under continuous development [6].

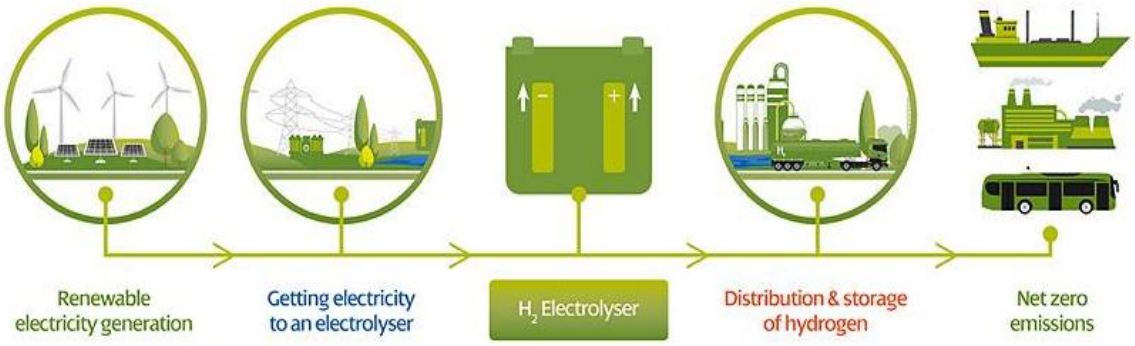


Figure 1 - Green hydrogen flow [5]

1.2. Objectives

This project focuses on designing and developing an optimization and monitoring system of a hydrogen production prototype powered entirely by renewable energy. The aim is to design a system that not only produces hydrogen efficiently but also integrates monitoring and control for optimal operation. The designed system consists mainly of three units: the power unit, the control unit, and the hydrogen production unit. The hydrogen production in

the system is achieved through a proton exchange membrane (PEM) electrolyzer. The PEM electrolyzer is supplied with electric current from the power unit, consisting of a photovoltaic (PV) solar panel ensuring that the energy input is renewable, batteries and a solar charge controller. The current supplied by the PV panel is regulated through a solar charge controller which manages the power system by charging the batteries and supplying the electrolyzer. The hydrogen production process itself is optimized and controlled through the control unit which consists of an ESP32 microcontroller and a DC/DC buck converter.

A key feature of this project is the integration of real-time monitoring as well as process control through a user-friendly interface. The system continuously tracks important parameters such as temperature, voltage, current, and gas leakage through various sensors. This ensures safe operation and provides data for optimizing the production process. The data is stored in a time-series database on InfluxDB and the history is visualized using Grafana. Additionally, the system takes advantage of the ESP32 microcontroller, such the dual core CPU, Wi-Fi connectivity, and compact size as well as the possibility of programming in FreeRTOS ensuring real-time operation. The user interface dashboard is implemented through Arduino IoT cloud, which is an online platform for coding, deploying, and monitoring IoT projects with support for various connections and devices. The hydrogen production can run on either manual or automated mode of operation allowing users to define specific hydrogen production volume or run the process continuously, as well as switching on/off the system and button to start/stop the operation.

Overall, this project is a practical application of green hydrogen production on the laboratory scale. By combining renewable energy with PEM electrolysis technology, it demonstrates the potential of a sustainable energy system. Through continuous development, creative design, and commitment to environmental responsibility, this project aims to contribute to exploring and development of such systems to take part in the global effort to combat climate change and build greener, more sustainable energy sources.

1.3. Report Structure

This report is divided into six chapters, namely: Introduction, Literature Review, System Design, System Monitoring, System Optimization, and Conclusion. Chapter one, introduction, a brief overview of the framework of the project and the main objectives. In addition, it points out the aim, outlines the purpose and motivation for developing this project. Chapter two, literature review, provides a general review about hydrogen production methods, water electrolysis and PEM technology as well as an overview of a state of art device design for laboratory on demand hydrogen production, explaining the design, functioning and applications of the device. Chapter three, system units and design, provides a detailed description of the design of the hydrogen production system and the compatibility of the components, as well as a description of the implementation of the control. Chapter four, system monitoring, presents the implemented sensors for monitoring the process, in real-time, and all the associated operations. Chapter five, system optimization, presents a detailed explanation about the control of the process, which is done through open-loop control, and all associated operations. Chapter six, conclusions, this chapter provides a comprehensive overview of the work conducted and the results obtained.

2. LITERATURE REVIEW

2.1. Hydrogen Production

Hydrogen as a fuel source dates back around two centuries. Its potential as a fuel was first discovered in the early 19th century, and since then, it has undergone many developments and has been used in various applications. Later on, during the mid-20th century, hydrogen started draw attention and it played an important role such as being used as rocket fuel, especially during the Apollo moon missions [7]. During this period, the generation of electricity by combining hydrogen and oxygen in fuel cells was also developed. This technology was first developed to be applied in space missions initially and later it found application in vehicles as well.

Recently, hydrogen is being considered as a clean fuel in various industries and it has gained attention as a green energy source. Furthermore, due to global warming and greenhouse gas emissions, the focus is not only using hydrogen but also producing it in an eco-friendly method such as using renewable sources of electricity in water electrolysis. Hydrogen produced from such methods is referred to as green hydrogen [8]. To obtain hydrogen, it must be separated from other elements such as oxygen in water which is a very common method of producing hydrogen and applied in various industrial production facilities. Figure 2 shows each of the main methods and the percentage of worldwide production of hydrogen. In general, there are non-renewable and renewable methods of hydrogen production which branch into various methods, however the main ones, that fall under non-renewable, are steam reforming of natural gas, oil reforming, or coal gasification with the largest percentage. Moreover, a small percentage of 4%, which falls under the renewable, is produced through water electrolysis and other renewable sources [9].

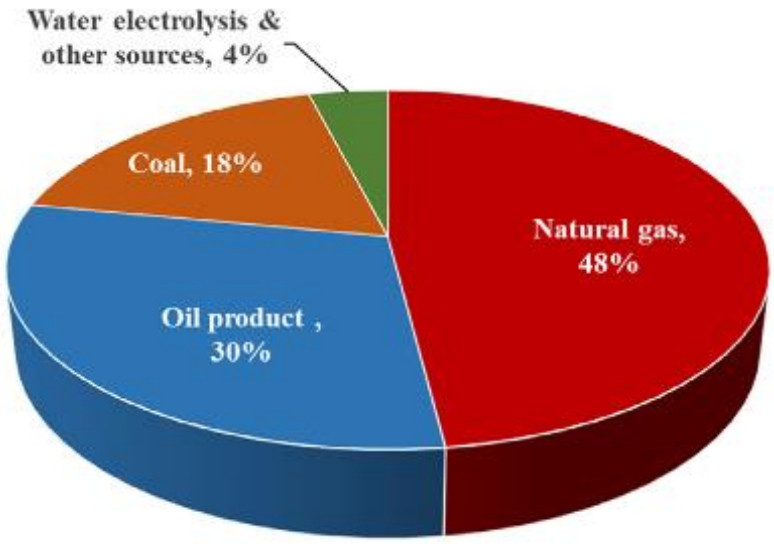


Figure 2 – Most common hydrogen production methods [9]

The generation of hydrogen from renewable sources, such as electrolysis powered by photovoltaic cells, is attracting a lot of attention. However, such technological advances are still in the early phases and are costly compared to hydrogen generation from fossil fuels [3]. This is due to the high costs associated with hydrogen generation and storage where both are the biggest obstacles to the widespread of this technology in a variety of industries. Figure 3 shows some of the most common hydrogen production methods, the source of energy used, and the cost of the process. As a clean and sustainable new energy source, hydrogen is gradually developing as an effective alternative to fossil fuels, contributing to the control of greenhouse gas emissions and global temperature change. Thus, as the demand in recent years has been gradually increasing many doors for promising application have opened, such as pharmaceuticals, aviation & maritime and especially in the energy sector where it can be used as fuel [10].

Method		Cost ($/kg H_2)
Fossil fuels to hydrogen	SMR	2.08
	SMR with CCS	2.27
	CG	1.34
	CG with CCS	1.63
	ATR with CCS	1.48
Biomass to Hydrogen	Biomass gasification	1.77–2.77
	Direct biophotolysis	2.13
	Indirect biophotolysis	1.42
	Photo fermentation	2.83
	Dark fermentation	2.57–6.98
Electrolysis	Grid electrolysis	5.73–8.54
	PV electrolysis	5.78–23.27
	Wind electrolysis	5.27–9.37
	Nuclear electrolysis	3.56–7.00
	High-temperature electrolysis (SOEL)	2.89–6.03
Thermolysis & thermochemical cycles	Nuclear thermolysis	2.17–2.63
	Solar thermolysis	7.98–8.40
	S–I cycle	1.99–14.85
	Cu–Cl cycle	1.71–14.20
	Ca–Br cycle	7.06
	Mg–Cl cycle	3.67

Figure 3 - Cost of hydrogen production through various methods [9]

Each of the methods that are used today to produce hydrogen has its own advantages and disadvantages. However, many factors are considered during an industrial production process and as the world aims at cleaner energy and reducing greenhouse effects, hydrogen is an example with a big potential and one of the most promising alternative fuels for the future due to capacity of storing high quality energy.

Figure 4 shows some of the main hydrogen production methods with respect to their advantages and disadvantages, as well as their efficiency. It is notable how electrolysis has high efficiency and the possibility of establishing a zero emission of greenhouse gases when compared to the most applied method, Steam reforming, which has high efficiency. However, its largest disadvantage is that the by product is CO and CO_2.

Hydrogen production Method	Advantages	Disadvantages	Efficiency
Steam Reforming	Developed technology & Existing infrastructure	Produced CO, CO_2 Unstable supply	74–85
Partial Oxidation	Established technology	Along with H_2 Production, produced heavy oils and petroleum coke	60–75
Auto thermal Reforming	Well established technology & Existing infrastructure	Produced CO_2 as a byproduct, use of fossil fuels.	60–75
Bio photolysis	Consumed CO_2, Produced O_2 as a byproduct, working under mild conditions.	Low yields of H_2, sunlight needed, large reactor required, O_2 sensitivity, high cost of material.	10–11
Dark Fermentation	Simple method, H_2 produced without light, no limitation O_2, CO_2-neutral, involves to waste recycling	Fatty acids elimination, low yields of H_2, low efficiency, necessity of huge volume of reactor	60–80
Photo Fermentation	Involves to waste water recycling, used different organic waste waters, CO_2-neutral.	low efficiency, Low H_2 production rate, sunlight required, necessity of huge volume of reactor, O_2-sensitivity	0.1
Gasification	Abundant, cheap feedstock and neutral CO_2.	Fluctuating H_2 yields because of feedstock impurities, seasonal availability and formation of tar.	30–40
Pyrolysis	Abundant, cheap feedstock and CO_2-neutral.	Tar formation, fluctuating H_2 amount because of feedstock impurities and seasonal availability	35–50
Thermolysis	Clean and sustainable, O_2-byproduct, copious feedstock	High capital costs, Elements toxicity, corrosion problems.	20–45
Photolysis	O_2 as byproduct, abundant feedstock, No emissions.	Low efficiency, non-effective photocatalytic material, Requires sunlight.	0.06
Electrolysis	Established technology Zero emission Existing infrastructure O_2 as byproduct	Storage and Transportation problem.	60–80

Figure 4 - Advantages & disadvantages of hydrogen production methods [11]

2.2. Water Electrolysis

Electrolysis is a technique in which a chemical reaction is driven by a direct electric current between electrodes through a chemical solution with an objective to separate the elements. Water electrolysis results in the breaking of the chemical bonds present in water, thus separating the oxygen and hydrogen, providing hydrogen gas which later can be used as clean fuel [12]. The technique itself is around 200 years old technology where in 1800, J.W.Ritter demonstrated the principles of electrolysis by experiment while later in the same period waters' decomposition into hydrogen and oxygen using electrolysis was presented in England by William Nicholson and Anthony Carlise. After 90 years, in 1890, the French military constructed the first electrolysis unit for hydrogen production which was used to fuel airships [13].

Nowadays, water electrolysis is a highly discussed topic and seen as a promising technology for green hydrogen production, considering using renewable energy to power the process, for both commercial and industrial use due to its advantages such as high cell efficiency and greater hydrogen production rate with high purity. There are four methods where this can be done, which are: i) Alkaline water electrolysis (AE); ii) Solid oxide electrolysis (SOE), also known as high temperature steam electrolysis; iii) Proton exchange membrane electrolysis (PEM); and iv) Microbial electrolysis cells (MEC) [11].

	AEL	PEM	SOEL
Electrolyte	NaOH/KOH(aq)	Polymer(s)	YSZ(s)
Charge carriers	OH^-	H^+	O^{2-}
Electrode material	Ni and Ni alloys	Platinum group metals	Cermet and doped metal composites
Temperature	60–90 °C	50–90 °C	500–1000 °C
Pressure	2–10 bar	15–30 bar	less than 30 bar
Cell voltage	1.8–2.4 V	1.8–2.2 V	0.95–1.3 V
Current density	0.2–0.5 A/cm^2	1–2 A/cm^2	0.3–1 A/cm^2
Efficiency	62–82%	67–84%	81–86%
Stack lifetime (maximum)	90000 h	40000 h	40000 h
System lifetime	20–30 yr	10–20 yr	–
Hydrogen production (maximum)	760Nm^3/h	30Nm^3/h	–
Cost (by 2050)	~\$600/kW$_{ch}$	~\$750/kW$_{ch}$	~\$200/kW$_{ch}$
Annual degradation	2–4%	2–4%	17%
Need a diaphragm	✓	×	×
Commercial or not	✓	✓	×
Advantages	Long lifetime Low cost High maturity Large stack size	Higher current density Compact system design Rapid response Dynamic advantages	High energy efficiency Less electricity consumption Feasible reversible operation Low expected cost Combination with other technologies
Disadvantages	Low current density Corrosive electrolyte Mixing of gases	Need noble metal materials Short lifetime High membrane cost	Severe environment Unstable electrodes Sealing issues

Figure 5 - Water electrolysis technologies [9]

The Microbial electrolysis cells technology is still under development and having several challenges towards hydrogen production rate, high current densities, and efficiency as well as complicated design that needs to be addressed before commercialization of this technology [14]. On the other hand, Alkaline water electrolysis, Solid oxide electrolysis, and Polymer electrolyte membrane electrolysis, figure 5, are getting a lot of attention since these methods have been in use for the past century. However, the focus now is to depend more on renewable energy as the power source for these processes rather than non-renewable fuels. Figure 6 shows the different reactions that take place in the four methods mentioned. It is noticeable that the bases of all the methods are identical, however, the variation lays in the design of the cell, type of electrolyte used, function temperature, and current density, as well as costs [11][15].

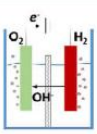
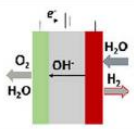
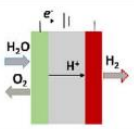
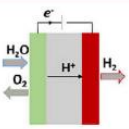
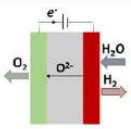
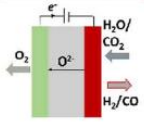
	Low Temperature Electrolysis			High Temperature Electrolysis		
	Alkaline (OH^-) electrolysis		Proton Exchange (H^+) electrolysis	Oxygen ion(O^{2-}) electrolysis		
	Liquid	Polymer Electrolyte Membrane		Solid Oxide Electrolysis (SOE)		
	Conventional	Solid alkaline	H^+ - PEM	H^+ - SOE	O^{2-} - SOE	Co-electrolysis
Operation principles						
Charge carrier	OH^-	OH^-	H^+	H^+	O^{2-}	O^{2-}
Temperature	20-80°C	20-200°C	20-200°C	500-1000°C	500-1000°C	750-900°C
Electrolyte	liquid	solid (polymeric)	solid (polymeric)	solid (ceramic)	solid (ceramic)	solid (ceramic)
Anodic Reaction (OER)	$4OH^- \rightarrow 2H_2O + O_2 + 4e^-$	$4OH^- \rightarrow 2H_2O + O_2 + 4e^-$	$2H_2O \rightarrow 4H^+ + O_2 + 4e^-$	$2H_2O \rightarrow 4H^+ + 4e^- + O_2$	$O^{2-} \rightarrow {}^1/_2O_2 + 2e^-$	$O^{2-} \rightarrow {}^1/_2O_2 + 2e^-$
Anodes	Ni > Co > Fe (oxides) Perovskites: $Ba_{0.5}Sr_{0.5}Co_{0.8}Fe_{0.2}O_{3-\delta}$, $LaCoO_3$	Ni-based	IrO_2, RuO_2, $Ir_xRu_{1-x}O_2$ Supports: TiO_2, ITO, TiC	Perovskites with protonic-electronic conductivity	$La_xSr_{1-x}MnO_3$ + Y-Stabilized ZrO_2 (LSM-YSZ)	$La_xSr_{1-x}MnO_3$ + Y-Stabilized ZrO_2 (LSM-YSZ)
Cathodic Reaction (HER)	$2H_2O + 4e^- \rightarrow 4OH^- + 2H_2$	$2H_2O + 4e^- \rightarrow 4OH^- + 2H_2$	$4H^+ + 4e^- \rightarrow 2H_2$	$4H^+ + 4e^- \rightarrow 2H_2$	$H_2O + 2e^- \rightarrow H_2 + O^{2-}$	$H_2O + 2e^- \rightarrow H_2 + O^{2-}$ $CO_2 + 2e^- \rightarrow CO + O^{2-}$
Cathodes	Ni alloys	Ni, Ni-Fe, $NiFe_2O_4$	Pt/C MoS_2	Ni-cermets	Ni-YSZ Subst. $LaCrO_3$	Ni-YSZ perovskites
Efficiency	59-70%		65-82%	up to 100%	up to 100%	-
Applicability	commercial	laboratory scale	near-term commercialization	laboratory scale	demonstration	laboratory scale
Advantages	low capital cost, relatively stable, mature technology	combination of alkaline and H^+-PEM electrolysis	compact design, fast response/start-up, high-purity H_2	enhanced kinetics, thermodynamics: lower energy demands, low capital cost		+ direct production of syngas
Disadvantages	corrosive electrolyte, gas permeation, slow dynamics	low OH^- conductivity in polymeric membranes	high cost polymeric membranes; acidic: noble metals	mechanically unstable electrodes (cracking), safety issues: improper sealing		
Challenges	Improve durability/reliability; and Oxygen Evolution	Improve electrolyte	Reduce noble-metal utilization	microstructural changes in the electrodes: delamination, blocking of TPBs, passivation		C deposition, microstructural change electrodes

Figure 6 - Electolysis methods charateristics [15]

In water electrolysis, water is considered as the reactant which dissociates into hydrogen and oxygen once it undergoes the influence of a direct electrical current. Typically, a water electrolysis unit consists of a cathode, anode, and an electrolyte in between. In the first three methods, the electrolyte can be made of aqueous solution containing ions, a proton exchange membrane, or an oxygen ion exchange membrane. Once direct electric current is applied at the anode, it travels through the electrolyte to the cathode causing an electrochemical reaction which results in hydrogen production [16]. In the context of this project, the PEM electrolysis is further discussed.

2.3. PEM technology

The first electrolyzer based on solid polymer electrolyte concept was developed in the 1960s by General Electric [17]. This concept is also referred to as polymer electrolyte membrane water electrolysis. The PEM is mainly responsible for providing high proton conductivity and low gas crossover, allowing it to produce hydrogen with high purity, in a compact system. It is also capable of operating at current above 2 A/cm^2, which in turn

reduces the operation cost associated to the process. Inside the compact system of the electrolyzer, a proton exchange membrane is combined with negatively and positively charged electrodes, a cathode, and an anode respectively, figure 7. When water is pumped to the anode it gets split into oxygen, hydrogen ions/protons and electrons. Protons are then conducted through the membrane to the cathode where they reunite with the electrons [18]. The produced hydrogen can then be stored in various forms or directly consumed by a fuel cell to produce electricity.

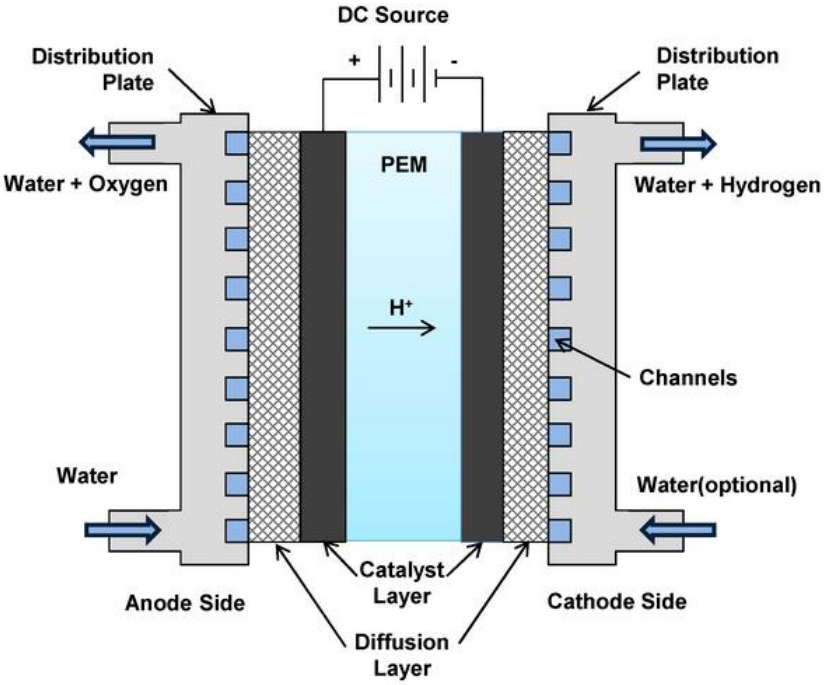


Figure 7 - PEM electrolyzer cell cross section [19]

Fuel cells, figure 8, are devices designed to electrochemically convert chemical energy into electricity which are regarded as the next generation power electric power devices due to their efficiency and low emissions. PEM fuel cells are a class of fuel cells that operate at considerably low temperatures between -40ºC up to 120ºC and, nowadays, are being further developed for various applications due to the simplicity of operation [20]. Fuel cells possess various applications and are continuously developed.

Fuel cell type (type of electrolyte)	Operating temperature (°C)	Applications	Electrical power range (kW)	Electrical efficiency (%)
Proton exchange membrane	60–110	Mobile, portable, low power generation	0.01–250	40–55
Alkaline	70–130	Space, military, mobile	0.1–50	50–70
Direct methanol	60–120	Portable, mobile	0.001–100	40
Phosphoric acid	175–210	Medium- to large-scale power and CHP	50–1000	40–45
Molten carbonate	550–650	Large-scale power generation	200–100,000	50–60
Solid oxide	500–1000	Medium- to large-scale power and CHP, vehicle auxiliary power units, off-grid power and micro-CHP	0.5–2000	40–72

Figure 8 - Fuel cell types of overview [21]

As the present project aims to develop and implement laboratory demonstration unit to produce hydrogen, a PEM fuel cell will be further discussed, whereas it has been chosen for various reasons including: 1) Its low operating temperature, which is suitable for laboratory scale applications. 2) Low power generation capacity, which is controlable on a small scale. 3) Practicallity, where as it is relatively simple to operate. When hydrogen gas is introduced into the PEM fuel cell it is oxidized at the anode and the protons released are exchanged through the membrane to the cathode. As the membrane is not conductive, the electrons released from the reaction are forced to travel through a current collector layer [22]. When the circuit is closed, as a result, an electrical current flow through a load, thus providing electrical power. Figure 9 displays a clear cross section view of the general design and components of a PEM fuel cell. PEM fuel cells can be used in various applications, on both small and large scale, such as: transportation, stationary power generation stations, and fuel cell electric vehicles (FCEV). Moreover, a lot of research and development is taking place as PEM fuel cell are seen as a promising solution for portable electronics and airplane industry, such as: laptops, cell phones, and military radio/communication devices, considering the limitation of electric batteries whereas it is unlikely that they can meet the fast-growing energy demand [23]

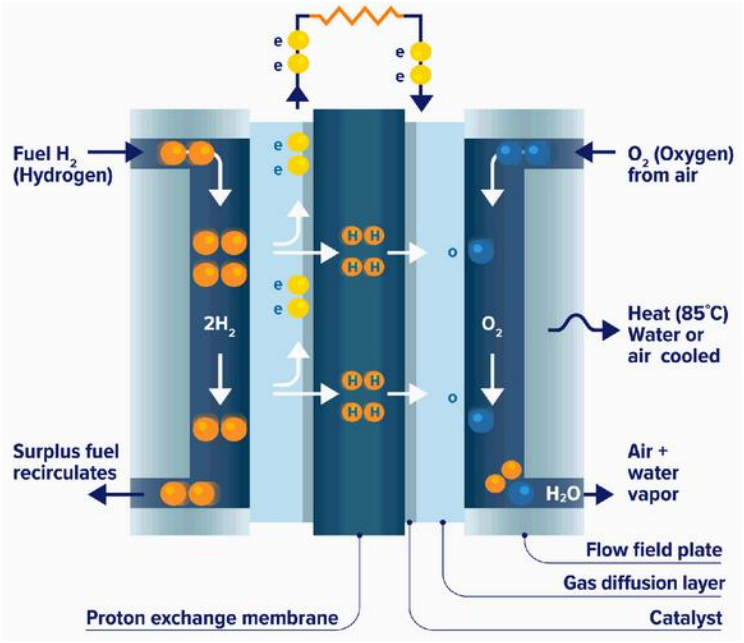


Figure 9 - PEM Fuel cell cross section [24]

2.4. State of Art

This chapter provides an overview of the state of art device for laboratory-scale hydrogen generation, and it covers its innovative design, safety features, performance, and an overview of its application. In addition, a description of the hydrogen production process and the advantages of this compact and user-friendly device ensuring its value as a reliable tool for laboratory hydrogenation processes. The state of art device serves as a reference for comparison of the laboratory scale hydrogen production devices and to better understand the design and process of hydrogen production compared to the prototype developed in this project.

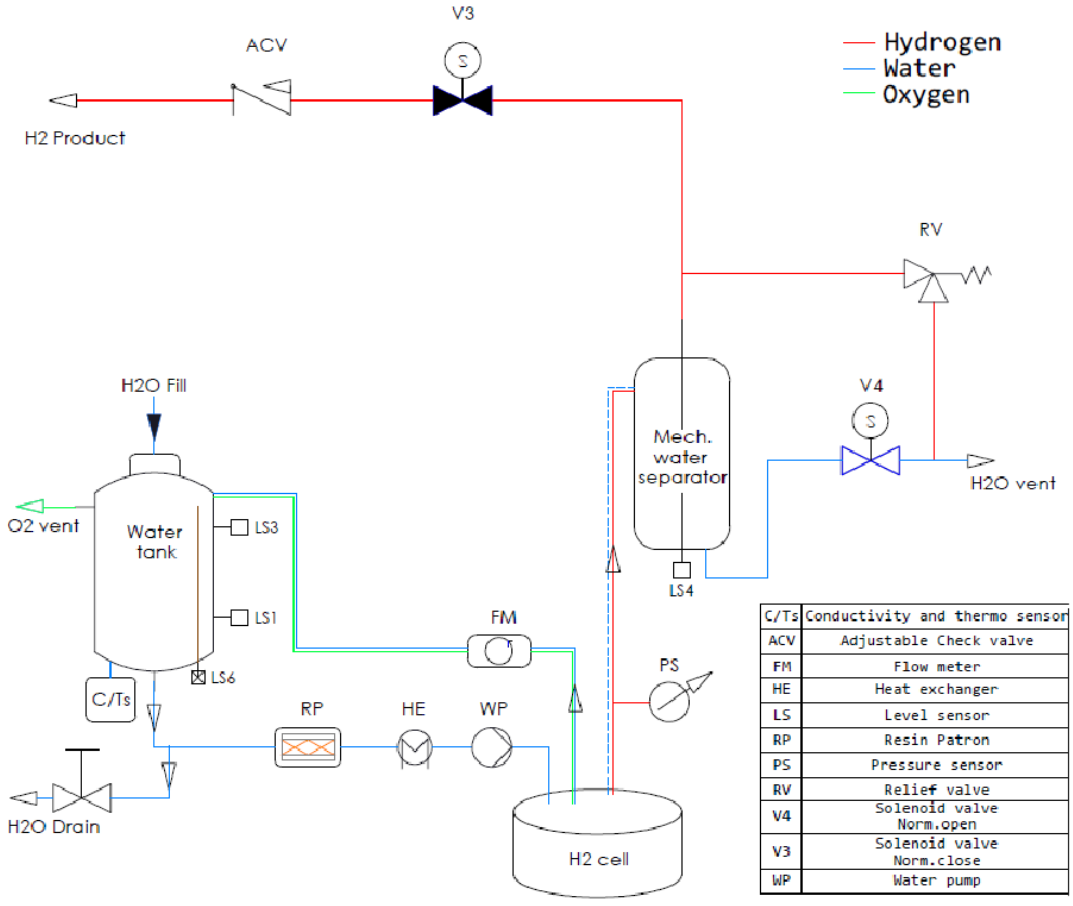
The H-Genie Lite is a significant improvement in the generation of hydrogen on the laboratory scale, being a small, efficient, and safe device when compared to conventional hydrogen cylinders. This device developed by ThalesNano incorporates proton exchange membrane (PEM) electrolysis technology for generating high purity hydrogen gas as needed for chemical synthesis applications.

Figure 10 - H-Genie Lite [25]

2.4.1. Design and Performance

To enhance the safety of the laboratory environment, H-Genie® Lite is designed to generate hydrogen gas from deionized water, eliminating the use of high-pressure hydrogen gas cylinders. To maintain safe operation and ensure leakage does not occur, the

device is built with internal hydrogen leak detectors as well as other safety mechanisms to automatically turn the device off in the event a leak is present. Easy to operate as there is an intuitive touchscreen interface consisting of simplified controls that allow the user to dial in their desired pressure or how long they wish to run the unit. With dimensions of 385 mm (H) x 365 mm (W) x 476 mm (D) and a weight of 33.7 kg, the H-Genie® Lite can be used with ease and is compact enough so that it can easily fit inside the standard laboratory. The device can deliver hydrogen at ambient temperature, flow rates of up to 1200 normal milliliters per minute (NmL/min), and 1 to 50 bars pressure, with a purity of ≥99.9%. It has a water tank capacity of 3 liters which is helpful for extended operation periods and has a consumption level of technically 200 mL of deionized water in an hour, table 1.



C/Ts	Conductivity and thermo sensor
ACV	Adjustable Check valve
FM	Flow meter
HE	Heat exchanger
LS	Level sensor
RP	Resin Patron
PS	Pressure sensor
RV	Relief valve
V4	Solenoid valve Norm.open
V3	Solenoid valve Norm.close
WP	Water pump

Figure 11 - H-Genie Lite H_2 production diagram [25]

The system starts from the water storage tank that contains deionized water which passes through a resin patron (RP) to get rid of impurities. Water is pumped through a heat exchanger (HE) to achieve optimal temperatures before entering the cell. Within the cell, water is electrolyzed which breaks down the water into hydrogen and oxygen. A mechanical water separator separates most of the water from hydrogen, while oxygen is purged from the system. The hydrogen travels through a pressure sensor (PS) for pressure control league, into the product line via an adjustable check valve (ACV). A V3 solenoid valve ensures that hydrogen is operated in a safe and reliable way. Wastewater produced from the process is eliminated through the telling solenoid valve (V4). Water levels in the tank and in the mechanical separator of the system are maintained through level sensors which are safety and monitoring features for efficient and safe operation. This closed-loop system can generate hydrogen with promised purity of 99.9% and at pressures of 50 bars.

2.4.2. Applications and Advantages

H-Genie® Lite is specifically useful in hydrogenation reactions during chemical synthesis. Its ability to provide high-pressure hydrogen on-demand allows chemists to perform experiments because they do not have to use gas cylinders due to its provision of high-pressure hydrogen, which is needed in small quantities. Enhanced by their ability to be used with an assortment of reactors such as batch and flow systems, enhances their versatility in research and development settings.

Compared to traditional hydrogen sources, the H-Genie® Lite offers several advantages. Generating hydrogen on demand and including leak detection sensors helps enhance safety, and thus, the chances of incidents involving hydrogen are limited. On the other hand, operational efficiency and cost effectiveness is also an advantage because on demand capability reduces non-productive time caused by cylinder changeovers and saves on space. The ease of everyday use is combined in the H-Genie Lite with features and improved safety, which makes it easy to use PEM electrolysis technology through a user-friendly, safe, and efficient hydrogen generation system for laboratory applications. As designed, it solves the problems encountered in most hydrogen research in laboratories and is therefore a useful hydrogenation tool for chemists performing hydrogenation and related processes.

Table 1 - H-Genie Lite technical data [25]

Technical Data	
Pressure Range of H-Genie® Lite:	From ambient to 50 bar (gauge)
Hydrogen flow rate range: *	Approximately 1200 NmL/min
Water purity at room temperature:	99.9%
Internal hydrogen volume:	64 mL
Water consumption rate:	~ 200 cm3/h
Water tank capacity:	3L
Recommended environment:	Ventilated laboratory fume hood
Power requirements:	Mains: 100V to 240V AC, 47-63Hz
Power consumption:	max. 600 VA
Unit dimensions (H x W x D):	385 mm x 365 mm x 476 mm

*Performance depends on the age and use time of the cell.

3. SYSTEM DESIGN

3.1. System Devices

The system consists of three primary subsystems that collaboratively enhance the hydrogen production process, resulting in a green hydrogen cycle. These are power, control, and hydrogen production. Each of the components is chosen specifically to be compatible and functional with integrated together.

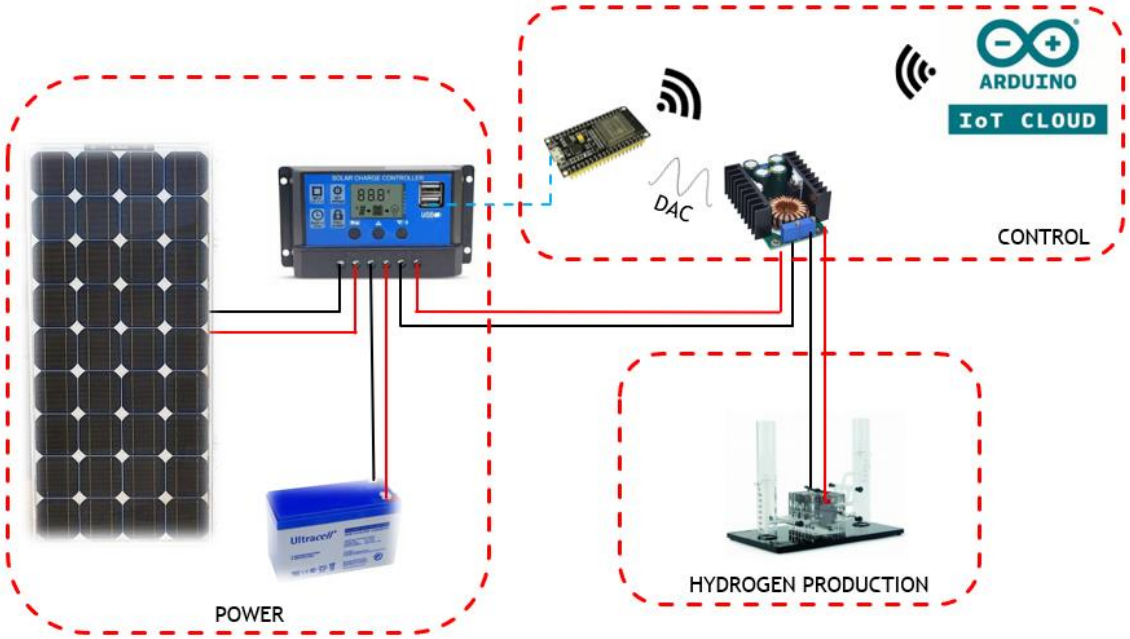


Figure 12 - H_2 production system

3.1.1. Power

This subsystem is responsible for powering the system from solar energy to operate. It consists of a photovoltaic panel, two 12 V batteries connected in series, and a PWM solar charge controller, figure 13. The photovoltaic panel harnesses solar energy, regulated by the solar charge controller, which manages the electrical power supplied to the electrolyzer and charges the batteries. A 185 W PV panel is utilized to supply electrical power to the system. The selected PV panel provides an open circuit voltage of 43.92 VDC and a short circuit current of 5.39 A, which meet the operational parameters required for the electrolyzer to operate at full capacity to produce hydrogen. A PWM solar charge

controller is utilized to manage the solar charge from the PV panel, regulating electricity to the panel and charging the batteries. In solar power applications, an MPPT is typically utilized; however, given that the electrolyzer operates within a voltage range of 0-14 VDC and a maximum power of 56 W, around 30% of the power provided by the PV panel, it is not necessary to ensure the maximum power point of 185W from the PV panel for this application.

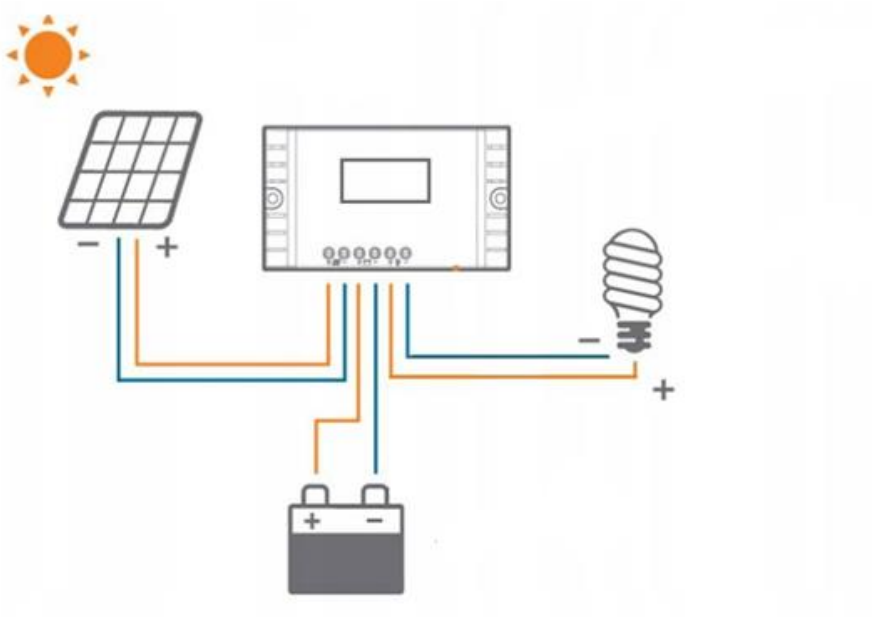


Figure 13 - Power unit layout

The PWM solar charge controller is utilized to control the operation of photovoltaic systems. This device guarantees that the correct charging current characteristics for the batteries, as well as protecting them from excessive discharges or overcharges, which negatively affect battery life. The following features are included:

- **Automatic Voltage Recognition:** The system automatically detects and adjusts to 12V or 24V input voltage, enhancing compatibility and reducing setup complexity.
- **User Interface:** Equipped with a clear LCD display, the device interface is controlled via three buttons, ensuring user-friendly operation and monitoring of system status.
- **3 phase PWM charging:** this provides stable and efficient charging for solar energy applications such as the one in this project.

- **Battery level control:** the device manages the battery level through discharge stop and discharge reconnect limits as well as built in timer activation for night/day only operation.
- **2 USB port for auxiliary application:** built in 5V/2A USB ports for auxiliary use, such in this project, powering the ESP32.

To complete the power unit, two 12 V/7 Ah sealed lead batteries are utilized. The batteries are connected in series to provide 24 V which powers the system, figure 14.

Figure 14 - Solar charge controller & Batteries

3.1.2. Control

The control unit consists of the ESP32, a microcontroller, which is the control core of the system and a DC/DC XL4016 buck converter. The model of the microcontroller used is ESP32-WROOM-32D, which is a powerful module with Wi-Fi and Bluetooth capability, designed for a wide range of IoT applications. It offers a comprehensive set of features for building interconnected devices. With dual-core processing, ample memory and several integrated peripherals, it provides versatility and efficiency in the development of various IoT and sensor/actuator projects. Alongside controlling the input voltage into the electrolyzer, the microcontroller is responsible for sending the data read from sensor to the InfluxDB database over Wi-Fi, as well as displaying real-time feedback from the sensors and the process to monitor the system.

3.1.2.1. ESP32 microcontroller

ESP32 is capable of functioning reliably in industrial environments, with an operating temperature ranging from –40°C to +125°C, figure 15. Powered by advanced calibration circuitries, it can dynamically remove external circuit imperfections and adapt to changes in external conditions. In addition, it can be employed for mobile devices, wearable electronics and IoT applications, whereas it achieves ultra-low power consumption with a combination of several types of proprietary software.

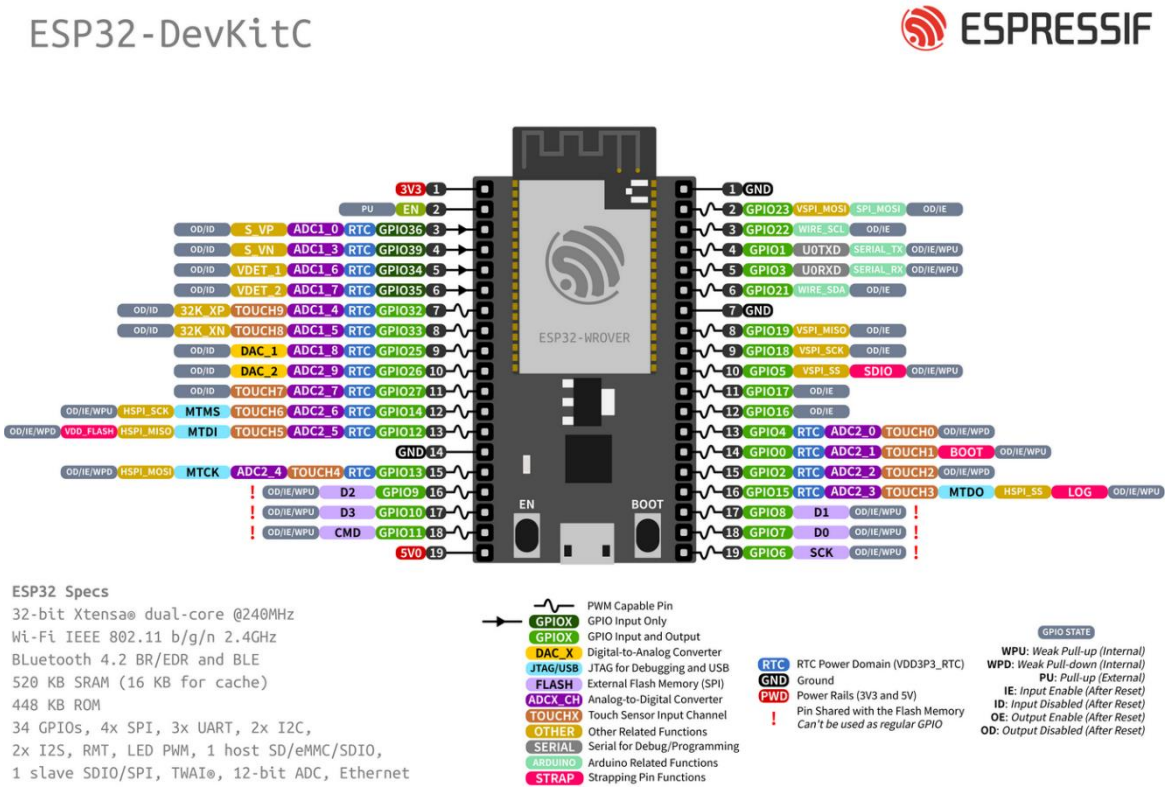


Figure 15 - ESP32 layout [26]

Moreover, the ESP32 includes features, such as fine-grained clock gating, various power modes and dynamic power scaling. It is also highly integrated with in-built antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power management modules. Regarding IoT applications it adds priceless functionality and versatility with minimal Printed Circuit Board (PCB) requirements. ESP32 can interface with other systems to provide Wi-Fi and Bluetooth functionality through its SPI / SDIO or I2C / UART interfaces [26].

3.1.2.2. Arduino IoT Cloud

The programing is done on Arduino IoT Cloud platform. Arduino IoT Cloud allows the configuration, programing and connection to IoT devices through one dashboard. It is designed for IoT applications and projects. Arduino IoT Cloud provides online IDE, which facilitates the possibility of programing from anywhere, and dashboard creation with various widgets which are connected to variables associated to applications on the microcontroller connected to it. The default option for programming on microcontrollers such as ESP32 and to connect to the Arduino Cloud is by using the C++ language. The configuration and connection between the microcontroller and the Arduino Cloud is supported by the ArduinoIoTCloud library & Arduino_ConnectionHandler libraries.

The Arduino IoT Cloud, figure 16, has a lot of new features including the OTA (over-the-air) feature. Quite literally, this feature enables the user to upload programs to your Arduino boards without needing to connect them physically. This means that, once the board has been connected to a WiFi network and is set for OTA use, there will no longer be a necessity to plug the board into the PC to upload new sketches on the board. All subsequent operations will be done through over-the-air technology.

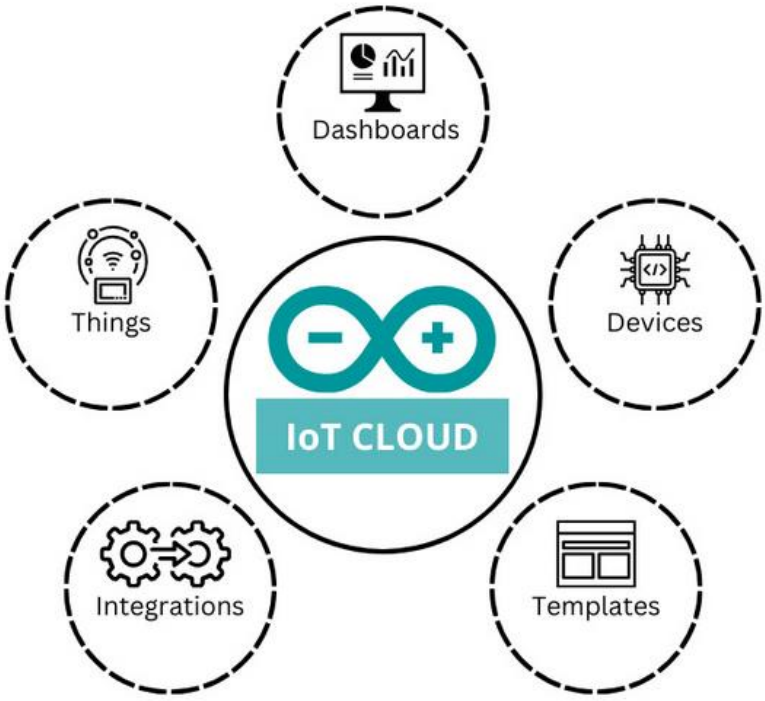


Figure 16 - Arduino IoT Cloud structure

Through the mobile application, which is called Arduino IoT Remote, it is then possible to control how certain dashboards will look like and how they are used in the Arduino Cloud. Also, using the app, interaction with such hardware as the internal phone's sensors could be performed, e.g. GPS data, light sensor, IMU etc.

Whenever a Thing is created in the Arduino Cloud, figure 17, it automatically starts generating a set of files that will handle the configurations, credentials & connection:

- <sketchname>.ino: the main sketch file, where the code in C++ is written
- thingProperties.h: the main configuration file, including all necessary properties for the sketch.
- arduino_secrets.h: the credentials file (for API key, Wi-Fi network etc.) where the network SSID and password are configured alongside other essential access configurations.

The "Thing" is a virtual copy of the hardware application. It is where the variables that synchronize between the cloud and the board are within the scope of this project it is named H2_Thing. The variable types available to be created are:

- Boolean
- String
- Float
- Integer

It is also possible to associate a variable permission whether to "Read & Write" or to "Read Only", as well as a variable update policy which can be configured to be either "On Change" or "Periodically". It is also possible to synchronize the same variable with various "Things" to be able to interconnect various IoT applications.

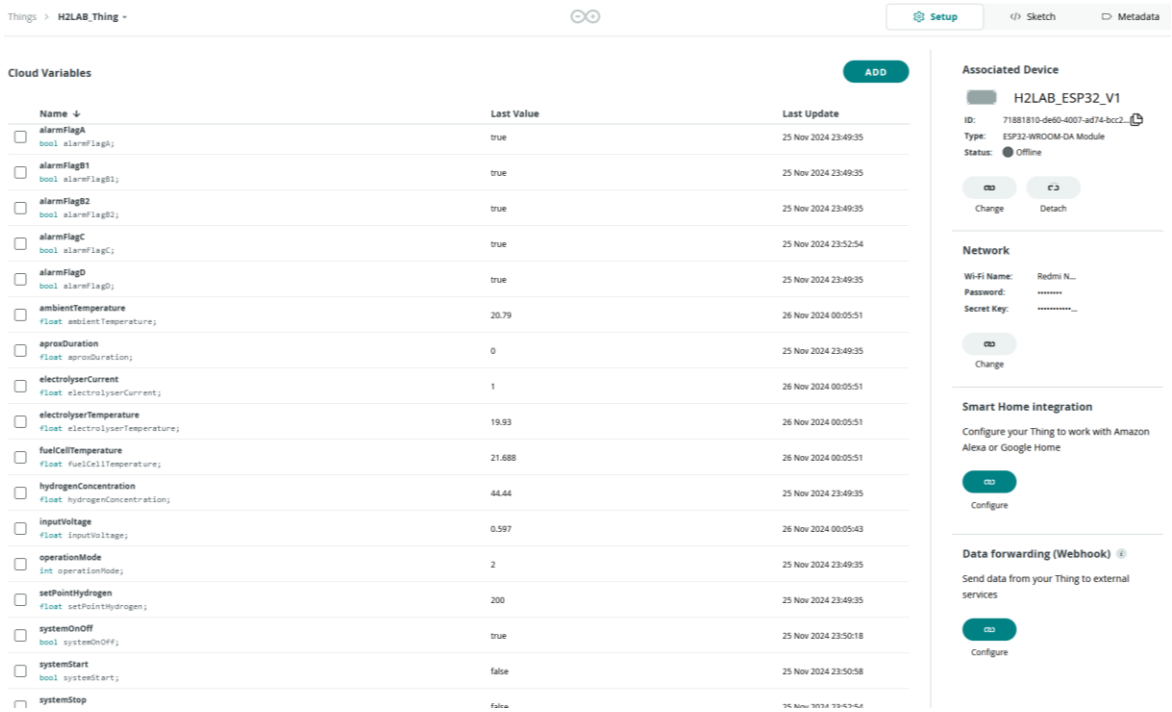


Figure 17 - Arduino IoT Cloud Thing panel

3.1.2.3. DC/DC Buck Converter

The DC/DC XL4016 buck converter is a 180 KHz fixed frequency PWM buck (step-down) DC/DC converter, capable of driving an 8 A load, and maximum 280 W with high efficiency, low ripple and excellent line and load regulation. The module consists of a XL4016 voltage regulator and a few other active (semiconductors) and passive components. It has a 7805 5V which regulates input voltage for the LM358 chip that operates as a feedback voltage comparator. TL431 operates as a shunt regulator, it is utilized as a positive voltage reference to the comparator circuit. The circuit also includes dual Schottky diode which serves the purpose of a voltage rectifier and on account of its high speed characteristic is essential for the efficiency of the converter. The converter circuit has input and output electrolytic capacitors that function to filter out unwanted ripples from the unregulated and switching section of the circuit and to also accumulate electric charge. The output voltage is adjustable between 1.25-35 V. In the scope of this project, the output voltage of the converter is controlled through digital-analog signal (DAC) from the ESP32 rather than the potentiometer. This will be discussed in more detail in chapter 6.

Figure 18 - DC/DC Xl4016 Buck converter

3.1.3. Hydrogen Production

3.1.3.1. E207 Electrolyzer

For the proper functioning of the E207 electrolyzer, figure 19, as part of the hydrogen production unit project, some operating parameters should be taken into consideration. One of these parameters is the operating medium, which must be distilled water. The distilled water used in the hydrogen production process of the E207 electrolyzer should have an electrical conductivity of less than 2uS/cm, as specified in the user manual provided by the manufacturer, H-TEC Education. This is mainly because the proton exchange membrane in the cell serves as both an electrolyte and a physical barrier between the anode and cathode. The membrane is designed to conduct protons (H^+ ions) from the anode to the cathode while preventing the passage of gases (hydrogen and oxygen) and electrons [27].

Figure 19 -E207 PEM Electrolyzer

The E207 electrolyzer consists of a seven-cell electrolysis stack, a hydrogen storage tank and an oxygen storage tank mounted on a baseplate. The seven individual cells, 16 cm^2 area, of the electrolysis stack are electrically connected in series, figure 19. Hydrogen and oxygen are removed via the tubes. The technical specifications are displayed in annex A. The water required for electrolysis is supplied from the gas storage through distribution pipe. The oxygen generated escapes through the water tank and the hydrogen produced is collected through a connecting pipe and is delivered to the hydrogen outlet. The E207 PEM

electrolyzer is capable of production hydrogen at a rate of 230 ml/min being supplied with voltage above 12.85 VDC. The hydrogen production rate is directly proportional to the input voltage. Figure 21 provides the relation between both values from previous tests that on the equipment.

Alongside the electrolyzer, an external storage cylinder is utilized. The storage tank for hydrogen or oxygen can hold up to 80 ml of gas, figure 20. By filling the storage with 80 ml of distilled water and connecting it to the hydrogen output storage cylinder, it acts as temporary storage for hydrogen gas. As the gas enters the lower tank it starts displacing the water to the upper tank, thus being stored at ambient pressure.

Figure 20 - External storage cylinder

TEST 1 RESULTS					
Voltage input(V)	Current consume(A)	Avg Current(A)	Time(s)	Time(min)	H2 produced(ml)
5.000	0.000	0.000	0	0	0.000
6.000	0.000	0.000	0	0	0.000
7.000	0.000	0.000	0	0	0.000
8.000	0.000	0.000	0	0	0.000
9.000	0.000	0.000	0	0	0.000
9.500	0.000	0.000	0	0	0.000
9.750	0.001	0.001	0	0	0.000
10.000	0.006	0.006	0	0	0.000
10.250	0.051	0.051	0	0	0.001
10.500	0.160	0.159	252	04m12s	25.000
	0.159		••	••	50.000
	0.158				
10.750	0.400	0.399	102	01m42s	25.000
	0.399		••	••	50.000
	0.398				
11.000	0.745	0.744	36	00m36s	25.000
	0.744		••	••	50.000
	0.742				
11.250	1.154	1.153	31	00m36s	25.000
	1.153		••	••	50.000
	1.152				
11.500	1.599	1.597	20	00m20s	25.000
	1.597		44	00m44s	50.000
	1.596				
11.750	2.080	2.079	14	00m14s	25.000
	2.079		35	00m35s	50.000
	2.078				
12.000	2.616	2.611	11	00m11s	25.000
	2.610		28	00m28s	50.000
	2.607				
12.250	3.117	3.114	10	00m10s	25.000
	3.114		22	00m22s	50.000
	3.111				
12.500	3.641	3.639	8	00m08s	25.000
	3.639		17	00m17s	50.000
	3.636				
12.750	4.200	4.198	7	00m07s	25.000
	4.198		15	00m15s	50.000
	4.196				

Figure 21 - E207 PEM Electrolyzer voltage & production rate relation

3.1.3.2. Water Conductivity

Operating the electrolyzer using water with an electrical conductivity of greater than or equal to 2 uS/cm will result in irreparable damage to the device. Therefore, taking this into consideration, during the preparation of the equipment to test and before adding any distilled water to the water tanks of the electrolyzer, the water to be used for this project was tested. The distilled water used in this project is obtained from the water purification system, WaterBlue L2 purification system, installed at the chemical engineering faculty at IPT. The purification system is ideal for everyday pure water laboratory tasks, providing ATSM type ll water for general chemistry applications requiring low ionic content and bacteria free water.

Conductivity testing was performed on a sample of distilled water using a PC 50 VioLab multimeter, figure 22, a device used to measure water pH, Redox, conductivity TDS, and temperature. The initial results obtained were of conductivity greater than 2uS/cm. After replacing the purification system's filters and analyzing a new water sample. The measured electrical conductivity was less than 2uS/cm, which is within the acceptable limits.

Figure 22 -Water conductivity test

3.1.4. H-12 Fuel Cell

Since the system does not include a storage unit for produced hydrogen, a PEM fuel cell is integrated into the system to consume hydrogen and complete the cycle. A fuel cell is a device that converts hydrogen and oxygen into water and electricity. The H-12 Fuel Cell stack includes a plurality of plate cells arranged along an axis generally parallel to cell thickness with electrically conductive separator plates between each pair of cells.

Figure 23 - H12 Fuel Cell

To obtain a deeper understanding about the fuel cell and it capacity a few tests were conducted following the manufacturer's instructions. To minimize the presence of contaminants, a filter was placed between the electrolyzer, the hydrogen outlet, and at the fuel cell stack, hydrogen gas input. To obtain higher efficiency, it is recommended to connect a pin to the hydrogen gas output at the fuel cell stack, however, it is suggested to remove the pin for 1 second every 10 minutes to purge the water from the stack.

A total of three tests were undertaken to obtain an average dataset by comparing the outcomes of each individual test. To conduct the experiment, an ammeter and a voltmeter were connected between the load and the fuel cell stack to read and record the values, annex C. Since the H-12 fuel cell stack operates at max power of 12 W at a current level of 1,50 A therefore, to simulate the load a combination of ceramic resistors that could handle this amount of current should was used. Table 2 is generated by combining the results obtained from the three tests conducted. In addition, a voltage-current curve is outlined in figure 24. It is quite notable that the power-current curve matches the curve outlined by the manufacturer in the user manual for the H-12 Fuel Cell Stack.

Table 2 - Fuel cell test results

Load (ohms)	H_2 flow rate(ml/min)	H_2 volume in cylinder(ml)	Fuel cell output voltage(V)	Fuel cell current(A)	Power(W)
7.7	230	35.67	8.96	1.053	9.43
8.1	230	43.33	9.01	0.993	8.95
11.0	230	47.00	9.23	0.783	7.23
13.3	230	39.17	9.30	0.647	6.02
23.0	230	38.33	9.57	0.407	3.89
44.5	230	45.33	9.76	0.247	2.41
77.5	230	45.00	9.98	0.133	1.33
100.9	230	41.33	10.03	0.100	1.00
open circuit	230	49.67	10.30	0.000	0.00

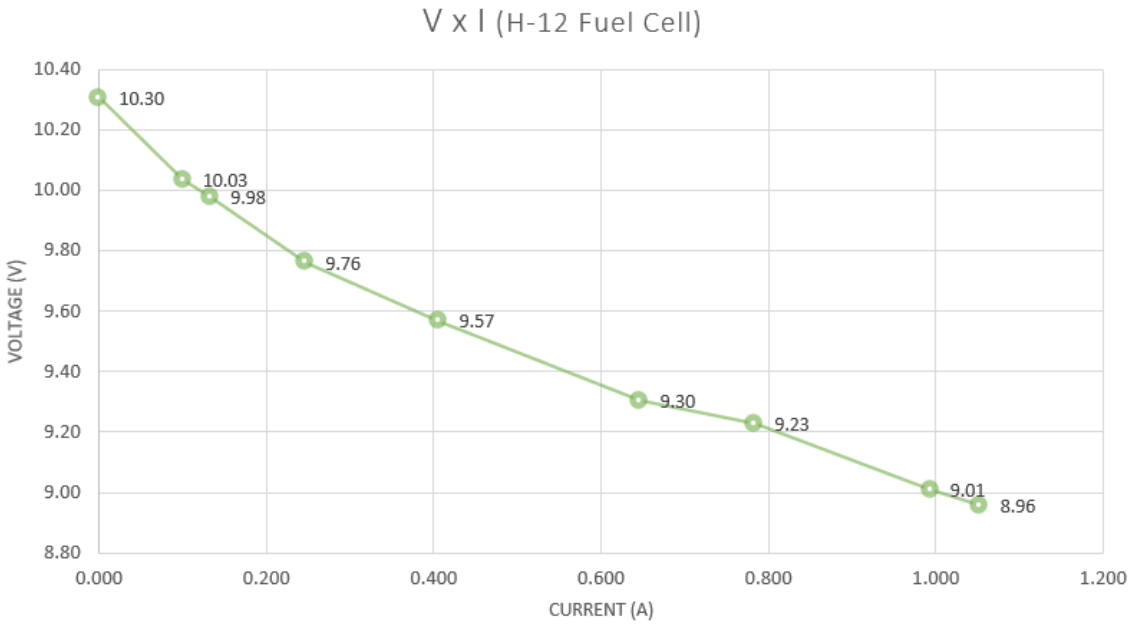


Figure 24 - Fuel cell Voltage vs Current curve

3.2. System Design

The system, figure 25, aims to produce hydrogen in an effective manner by employing a PEM electrolyzer which is powered by solar panels and batteries. A solar charge controller regulates the 24 V output which then passes through a DC/DC buck converter to target the input needed by the electrolyzer. The operations are controlled via an ESP32 microcontroller in connection with the Arduino IoT Cloud. Emergency shutdowns are among the safety provisions.

The system is controlled by ESP32 through Arduino IoT Cloud application. The user may switch on/off, define the setpoint (H_2 volume in ml). Once these parameters are defined the user may start the process. There are two operation modes, with control and without control, which the user can choose from. The main difference between them is that with control, the production duration is calculated based on the defined production rate and once the production period is over the production stops. However, without control, production is continuous until it is manually stopped. Besides controlling production, the ESP32 microcontroller performs other tasks, such as monitoring the system in real-time, sending data read from the sensors to InfluxDB database, and performing an emergency shutdown to protect the equipment in case of any abnormal conditions take place.

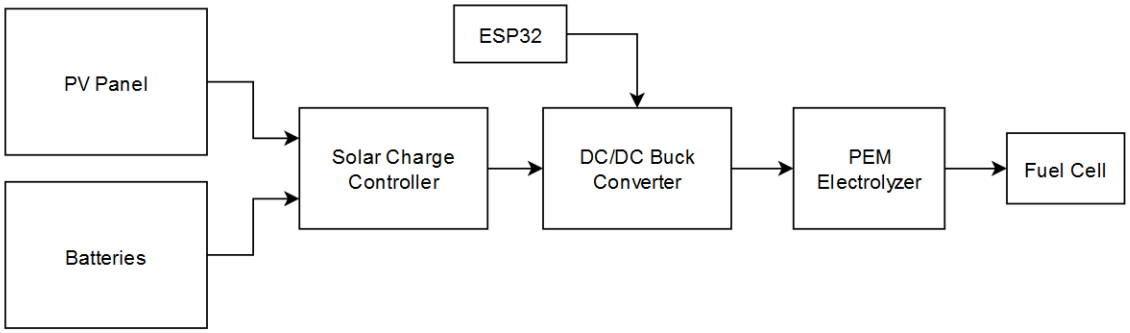


Figure 25 - Block diagram of the system

3.2.1. Hardware

The hardware components of the system have been carefully selected to ensure efficient operation, table 3. The PEM electrolyzer is the primary component for hydrogen production which operates within specific conditions and requirements. To meet these requirements, the system includes a DC/DC buck converter, a solar charge controller, PV panel and batteries.

- Photovoltaic (PV) Panel: The PV panel acts as the main energy input in the system. At the voltage of 36.50 V and current of 5.07 A, the power available at the maximum power point is 185 W. The maximum open circuit voltage is 43.95 V and the max series current is 5.39 A. The solar charge controller maintains this output to ensure that the electrolyzer supplied with sufficient power.

- Solar Charge Controller: A photovoltaic panel charge controller is an essential device for managing the non-constant current output from the PV panel. It functions as a constant 24 VDC source for the DC/DC buck converter. The controller works with 12 V and 24 V battery systems automatically recognizing the arrangement and capable of handling loads of 20 A. The controller has an LCD screen that allows controlling it and defining various setup and parameters such as: choosing battery types, setting day/night timers, battery charge and discharge parameters.

- Batteries: The system is equipped with sealed lead (Pb) batteries with two batteries of 12 V each that are connected in series to make a total of 24 V with a total capacity of 7 Ah. The batteries serve the purpose of energy storage which provides energy during cloudy periods or lack of solar energy.

- DC/DC Buck Converter: The DC/DC buck converter reduces the 24 VDC output voltage from the solar charge controller to the voltage level required for the PEM electrolyzer. It has an input voltage range of 7-40 volts and an output voltage range between 1.25-35 volts, 1.25 being the feedback maintained by the XL4016 as a

reference voltage internally, with the maximum output power being 280 watts. The converter can handle load up to 8 A.

- PEM Electrolyzer: being the most important component, it operates within an input voltage range of 0-14 VDC and a maximum current of 4.4 A, with a maximum hydrogen production rate of 230ml/min.

Table 3 - Techical specifications of the equipment

Equipment	**Technical specifications**	
PV panel	36.50 V_{MPP}	43.95 V_{OC}
	5.07 A (I_{MPP})	5.39 A (I_{SC})
	185 W	24 V_{DC}
Solar Charge Controller	<50 V_{MSI}	20 A (I_{CC})
	12/24 V_{SV} (Auto)	20 A (I_{DC})
Batteries	24 V (x2 12 V in series)	7 Ah
DC/DC Buck converter	7-40 V_{in}	8 A
	1.25-35 V_{out}	280 W
PEM electrolyzer	0-14 V_{in}	4.4 A

In addition, a PCB, figure 26, is developed on KICAD to ensure a compact design with input, output, power supply, and ground connectors. The PCB helps maintain a more reliable connection, which can be a source of noise, interference, and faulty connections.

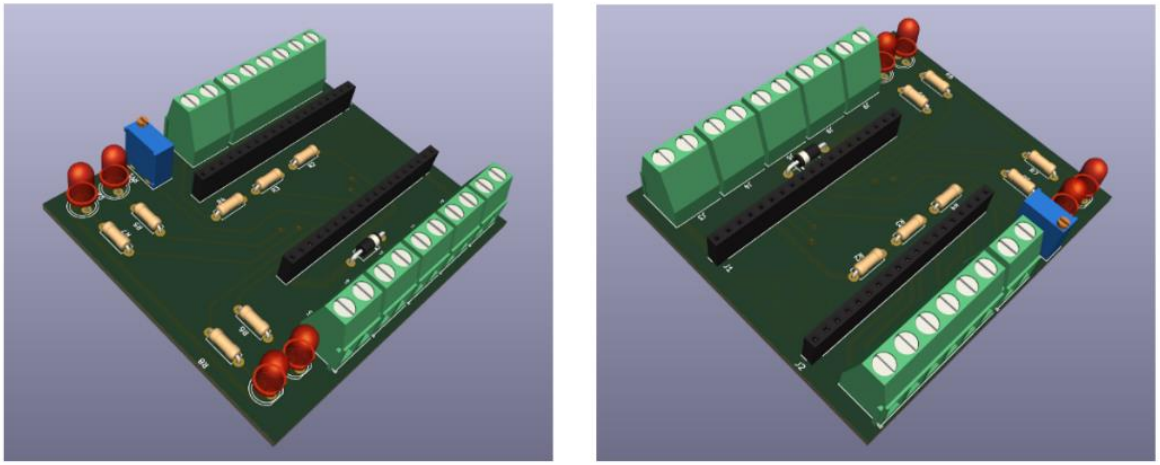


Figure 26 - PCB Board

The small and compact design of the PCB includes seven inputs accounting for various sensors used, six used and one as an auxiliary input pin, a poti-trimmer functioning as a variable resistor, represented as R3, and four LEDs for the status of the system, with the following designations:

- GREEN: System is ON/OFF (when switched on and off, respectively)
- RED: Alarm condition set (to be verified on the dashboard)
- YELLOW: System is initiating (performing voltage ramp)
- BLUE: Hydrogen production in action

In addition, since the MQ8 sensor signal is 5 V, a voltage divider is added to the circuit, which lays under the ESP32 board, to scale down the signal from the sensor to 3.3 V. The value of the resistance used in the voltage divider is 1k ohm for all three resistors, presented on the circuit, figure 27, as R2, R3, and R4. A 1N4001 diode is also integrated to protect the ESP32 from the discharge of the solenoid valve, as it behaves as an electromagnetic coil, since the valve is powered by 3.3 V from the ESP32's internal voltage regulator. In figure 27, the position of the ESP32 board is marked in blue. (see annex B)

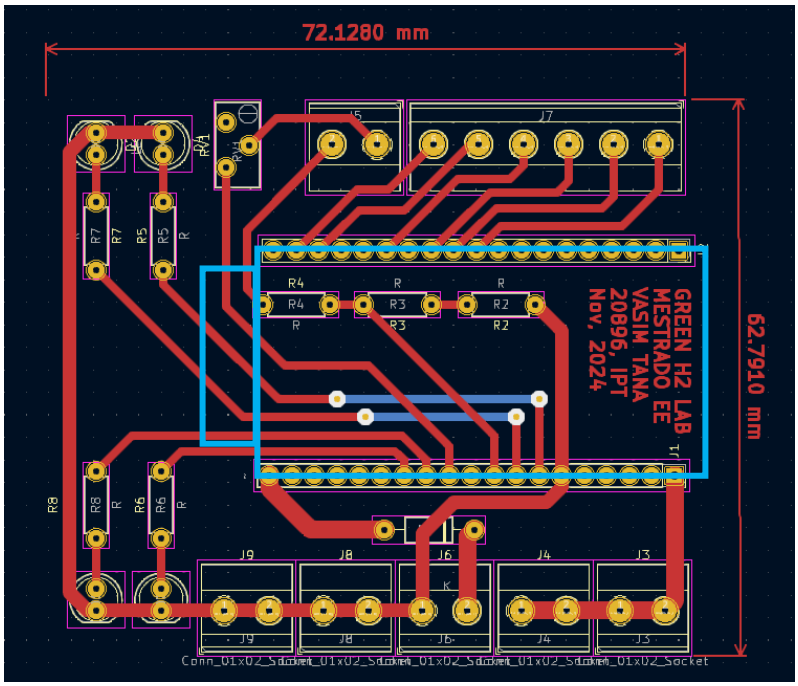


Figure 27 - PCB conncetions layout

3.2.2. Software

Arduino IoT Cloud is used for the user interface and system control, figure 28. The components of the Arduino IoT Cloud dashboard include an on/off switch, a box where the user can enter the required amount of produced hydrogen volume, in ml, a production monitor, and buttons to manually start and stop operations.

The first stage of the program is initialization, where software and hardware variables are set up, communication protocols for sensors (such as I2C one wire), FreeRTOS tasks, queues, and semaphores are created and configured. There are three main tasks: controlTask, monitoringTask and databaseTask, alongside various function that perform auxiliary operations. The monitoringTask and databaseTask are executed simultaneously on core 0, as by default it is primarily used for Wi-Fi and Bluetooth communication and are synchronized using semaphores to ensure dynamic execution, while the controlTask is executed on core 1.

The controlTask handles the main logic concerning the operation of the PEM electrolyzer, which works in two modes: With-Control and Without-Control. With control mode applies when a user inputs the required volume of hydrogen for this mode in the IoT Cloud dashboard. In this mode, the task computes the time required for production depending on the volume specified and supervises a timer using checkTimer() function for operational accuracy. voltageRamp() function gradualy transitions to the target voltage level. The input for the buck converter is regulated by the dac_output_voltage(), an Arduino function for the DAC pins, which ensures that the electrolyser receives a voltage within its operation limits. After the timer has finished, the process will end, and the system will be reset.

Without-control mode, the electrolyser will operate until it is manually stopped from the IoT dashboard. The functions voltageRamp() and dac_output_voltage() are executed as in the With-Control mode, except there is no timer that restricts the operational period. Production process stops when with user intervention in this case. Once the relevant mode logic is executed, the controlTask unblocks the semaphore via the call xSemaphoreGive() and enters a delay state by calling xTaskDelay() of 100 ms to permit dynamic execution.

Regarding data monitoring, the monitoringTask is set to read the sensors and checks for alarms condition. If an event raised an alarm, such as high temperature or a peak in current, it would set a flag to the occurrence to trigger system reset. After all the monitoring is done, the task releases the semaphore and delays its execution, ensuring efficient CPU utilization. On the other hand, the databaseTask is executed every 1 minute, where it writes

the values read from the sensors to a database to keep track of historical events of the system during each production cycle.

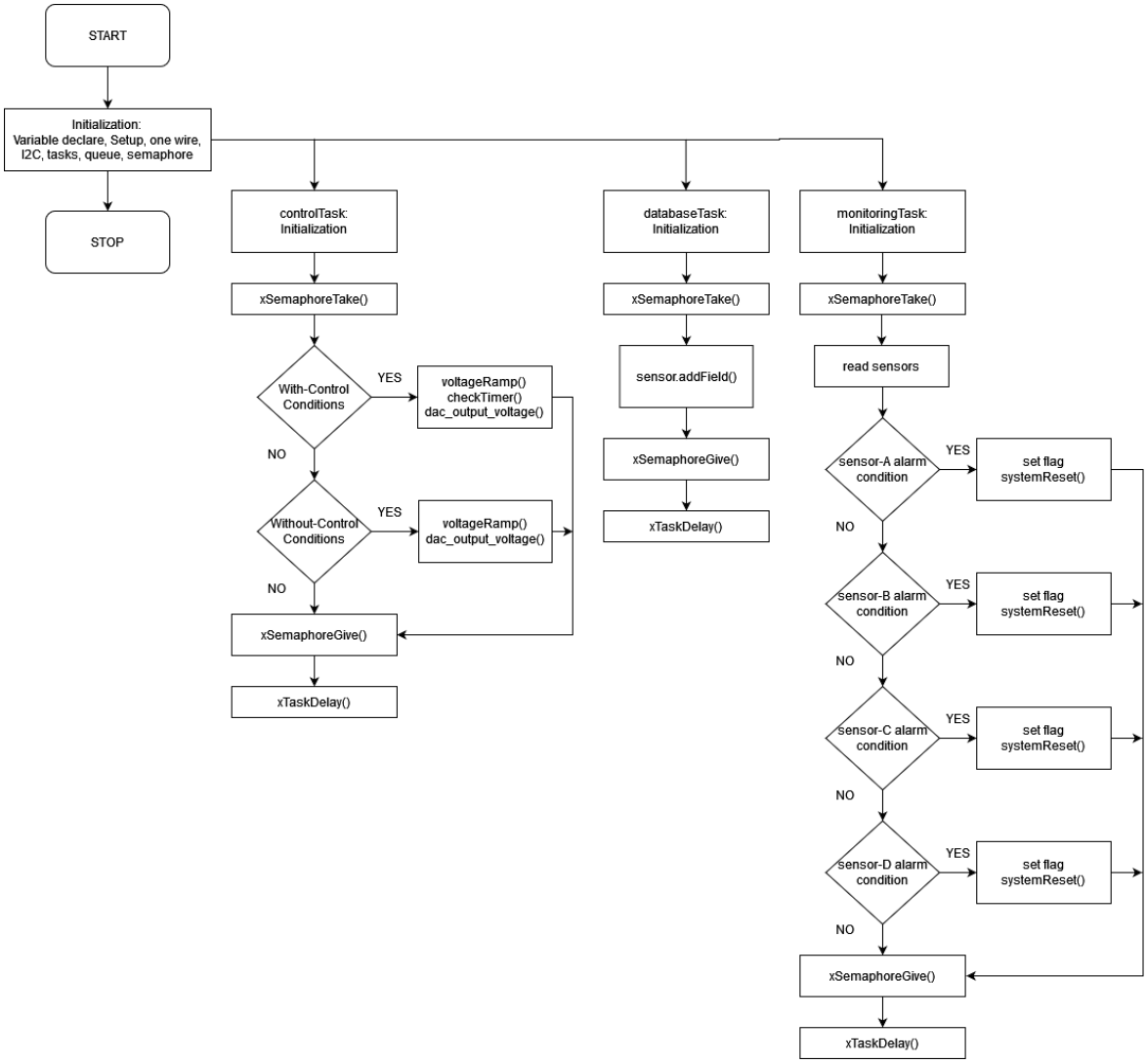


Figure 28 - FreeRTOS flow diagram (Annex D)

3.2.2.1. Libraries

The code includes various libraries to function properly and be able to operate the complex structure that consists of different sensors, protocols, and functions. The following libraries are employed to support the code and provide essential functionalities:

- thingProperties.h: it is used for handling IoT properties for Arduino IoT Cloud, such as managing Thing Descriptions and interacting with IoT platform.
- Arduino.h: The core library for Arduino/ESP32 development. It provids fundamental functions for programming and controlling Arduino/ESP32 boards.
- DFRobot_INA219.h: A library for interfacing with the INA219 sensor module, which is used for measuring current, voltage, and power in the system.
- DallasTemperature.h: A library for interfacing with Dallas Semiconductor's DS18B20 one-wire temperature sensors. It also includes the OneWire.h library for one-wire communication.
- DFRobot_MLX90614.h: A library for the MLX90614 infrared temperature sensor, used for non-contact temperature measurements. It also includes the Wire.h library for I2C communication.
- driver/dac.h: A specific library for handling digital-to-analog conversion (DAC) on compatible devices. On ESP32, DAC pins are by default GPIO 25 and 26.
- InfluxDbClient.h: A library for communicating with InfluxDB, it is used for logging for data logging from IoT devices to database bucket.
- InfluxDbCloud.h: A specialized library for functioning with InfluxDB's cloud platform, it allows IoT devices to send data to InfluxDB hosted services.
- WifiMulti.h: is a library to manage multiple Wi-Fi networks specially for ESP32. However, the code is configured to recognize the microcontroller connected and assign the proper library. In case of Esp 8266 is used, ESP8266WifiMulti.h will be included.

3.2.2.2. FreeRTOS Tasks

To obtain and process data in real-time, the code is written in FreeRTOS, a real-time operating system for microcontrollers and small microprocessors. All tasks run continuously in an infinite loop and use FreeRTOS features for multitasking. In scope of this application, semaphores were applied for such an aim. The code runs three tasks mainly, and they are:

- controlTask: is responsible for controlling the system's operation based on defined conditions. The task calls ArduinoCloud.update() function which ensures the system remains synchronized. Following, the system status, on/off, operation mode, and production status is validated. If the chosen mode is with control, then the internal timer check will be performed to control production. Otherwise, production is continuous until the user terminates the process. In both cases, if all conditions are valid, the voltage ramp is performed as it is essential for running the electrolyzer safely. This task is pinned to core 1 as it is dedicated to control using various functions.

- databaseTask: is responsible for sending data from sensors over to InfluxDB based on predefined time period. The task calls two functions, sensor.clearFields and sensor.addField, which clears the object field and adds the new data read, respectively, to InfluxDB bucket. This action is only performed when the system is validated to be on, and the production process is running. This task is pinned to core 0, as it is dedicated to send data over Wi-Fi to InfluxDB since core 0 is defined by default to run communication related operations.

- monitoringTask: is responsible for reading the sensor and writing the data into the associated variables to display them on the dashboard. In addition, the emergency system shutdown is integrated within this task by continuously checking for safety conditions and act upon by terminating the process and indicating the alarm on the dashboard. This task is pinned to core 0, as it is dedicated to send data over Wi-Fi to Arduino IoT Clod since core 0 is defined by default to run communication related operations.

3.2.2.3. Functions

To support the task in performing, four functions are integrated for auxiliary operations related to control, these functions are called within the tasks and/or other functions once or various times depending on their application.

- setTimer: is called once at the beginning of the process within the rampVoltage function. It is responsible for capturing the staring instance of the production and indicate the beginning of the production be setting a flag. In addition, it calculates the required production time, in seconds, for a defined hydrogen volume. This function is utilized in the with control mode.

- checkTimer: is responsible for checking the elapsed time since the start if production. This is done by comparing the elapsed time to the required time for production, calculated by setTimer. Once the condition is true, the production process is terminated alongside a system reset.

- systemReset: is responsible for resetting the system to initial conditions. This function is called in various situations, such as: manual stop, automatic stop, emergency shut down, and switching on/off the system.

- voltageRamp: is responsible for initiating the hydrogen production by gradually increasing the input voltage into the electrolyzer until reaching maximum production rate, 230 ml/min at around 13 V. This function is discussed in further detail in chapter 6, section 6.2. hydrogen production control.

4. SYSTEM MONITORING

For the system to function properly and achieve the aim of the project, various sensors and actuators were integrated. In total, there are four sensors and three actuators. The main objective of the sensors is to monitor the system's status, such as temperature, input voltage, current, and hydrogen leak.

4.1. Sensors and Actuators

4.1.1. Sensor A

DS18B20 is a digital temperature sensor that communicates using one-wire protocol, providing temperature in degrees Celsius with 9–12-bit precision. Its temperature range is between -55ºC to 125ºC. Each sensor has an associated 64-bit serial number - allowing many sensors to be used on a data bus. This component is the cornerstone of many data logging and temperature control projects. This sensor is designed to monitor the temperature of the fuel cell in the system, figure 29.

Figure 29 - Fuel Cell temperature sensor

4.1.2. Sensor B

INA219 Sensor is a digital wattmeter of a high-resolution, high-precision, large-scale measuring module that can measure the voltage, current and power of various electronic modules and electrical equipment within 0-26V and maximum current of 8A. It

communicates over I2C protocol, and the maximum relative error is less than ±0.2% (simple manual calibration is required before use). It can be used to evaluate the energy consumption or battery life of solar energy systems, batteries, motors, controllers or electronic modules. This sensor is designed to monitor the input voltage from the source and the current consumption by the electrolyzer in the system, figure 30.

In the Digital wattmeter, the voltage measurement does not require calibration. However, the current measurement must be calibrated. If calibration is not carried out, the relative error of the maximum current measurement is around 3%. If a single-point linear calibration is carried out using a high-precision multimeter or an electronic load, the linear error of the system can be effectively eliminated and the maximum relative error can be ±0.2%.

The following steps, indicated by the manufacturer, were followed to calibrate the sensor, as it was not possible to use an electronic load:

1. Connect the ESP32, the multimeter (switch to ammeter) and the load (motor or resistance). It is recommended that the power consumption of the load is not less than 100 mA.
2. Upload a calibration code provided by the manufacturer.
3. Change the value of the variable "float ina219Reading_mA = 1000;" according to the readings of the serial port print "Current" and "float extMeterReading_mA = 1000;" according to the current readings of the multimeter.
4. Reload the code.
5. Calibration completed.

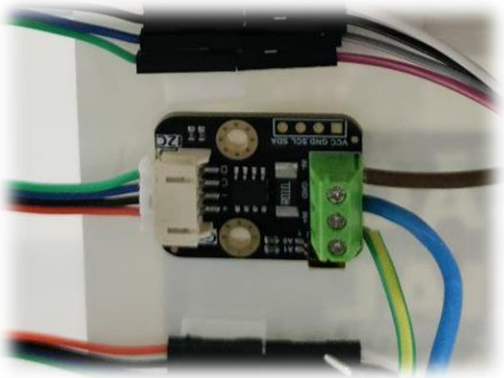


Figure 30 - Digital wattmeter

4.1.3. Sensor C

MLX90614 IR is a sensor that measures surface temperature by detecting infrared radiation energy and wavelength distribution. The infrared temperature probe consists of an optical system, a photoelectric detector, an amplifier, a signal processing module and an output module. It communicates over I2C protocol. This sensor is designed to monitor the temperature of the electrolyzer in the system, figure 31.

After being processed by the amplifier and the signal processing circuit, calibrated according to the algorithm and the target's emissivity, the signal is converted into the target's temperature value. MLX90614 is self-calibrating and has a low-noise amplifier integrated into the signal processing chip. The chip itself is a 17-bit ADC and DSP device, providing accurate and reliable results.

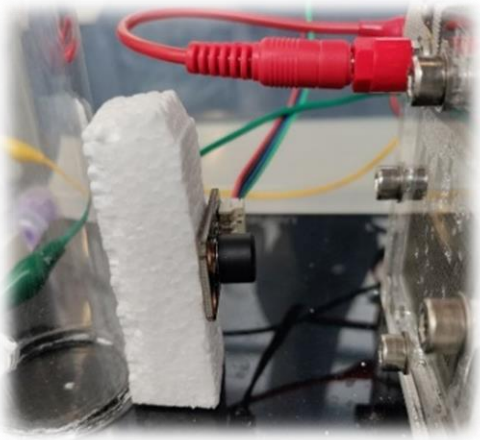


Figure 31 - Electrolyzer temperature sensor

4.1.4. Sensor D

MQ-8 Sensor is a hydrogen gas sensor that is highly sensitive to hydrogen gas and less sensitive to alcohol, LPG and cooking fumes. The sensitivity can be adjusted via the potentiometer. The output signal is proportional to the gas concentration. This sensor is used to monitor the leakage of hydrogen gas in the system and to obtain an approximate concentration of the amount of hydrogen that has escaped in ppm (parts per million).

The sensor consists of aluminum oxide ($Al_2O_{3)}$ ceramic micro tube, a sensitive layer of tin dioxide ($SnO_{2)}$, figure 32, a measuring electrode and a heater fixed to a crust made of plastic and stainless-steel mesh. The heater provides the necessary working conditions for the sensitive components to operate. The MQ-8 involved has 6 pins, 4 of which are used to obtain signals and the other 2 are used to supply heating current to the resistance.

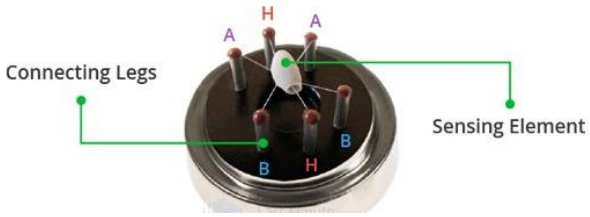


Figure 32 - MQ8 hydrogen sensor [28]

For this sensor to function properly, a sequence of calibrations had been done. The MQ-8 gas sensor depends on two resistance values to measure the concentration of hydrogen gas, figure 33. These values are the resistance in clean air, without hydrogen, and resistance in presence of hydrogen, Ro and Rs respectively. The calibration starts by setting the base resistance value (Ro) in clean air (Temp: 21 ºC and Hum: 65%). The sensor's analog output is read repeatedly and the average value of Vout is calculated to be over a total of 100 samples. This average value is then used to calculate Rs. Rs is divided by a clean air factor, Rs/Ro=70, to obtain Ro. The clean air factor, around 70, is determined from the sensor's datasheet. Once Ro has been obtained, it serves as a reference for measuring gas concentrations. Calibration ensures accurate gas measurements, considering variations in sensor sensitivity and environmental conditions.

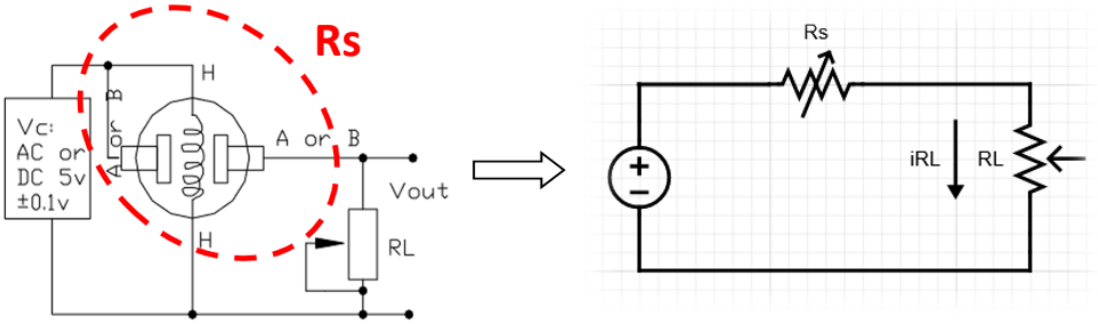


Figure 33 - MQ8 hydrogen sensor circuit

By applying ohms law and inserting equation 2 into equation 1, equation 3 is obtained for calculating the value of Rs. To avoid errors, the Arduino UNO was used to calibrate the MQ8 sensor, since the GPIO pins of the ESP32's ADC are limited to 3.3 V signal. However, Arduino can receive analog input signals of up to 5 V.

$$V = I.R \ \ (ohms\ law)$$

$$V_s = V_{R_L} = I_{R_L}.R_L \tag{1}$$

$$I_{R_L} = \frac{V_{cc}}{R_s + R_L} \tag{2}$$

Rearranging equation 2 into 1 and solve for Rs, equation 3 is obtained:

$$V_{RL} = \frac{V_{cc}}{R_s + R_L}.R_L$$

$$V_{R_L}.R_s + V_{R_L}.R_L = V_{cc}.R_L$$

$$V_{R_L}.R_s = V_{cc}.R_L - V_{R_L}.R_L$$

$$R_s = \frac{V_{cc}.R_L}{V_{R_L}} - \frac{V_{R_L}.R_L}{V_{R_L}}$$

$$R_s = \frac{V_{cc}.R_L}{V_{R_L}} - R_L \tag{3}$$

Where considering a bit resolution for the Arduino is 10, thus:

$$ADC\ range: 1 - 2^{10} = 0 - 1023$$

$$R_L = 10000$$

$$V_{cc} = 5$$

$$V_{R_L} = V_{out} = \frac{V_{cc}.ADC_{input}}{1023} \tag{4}$$

Equation 4 can be used to calculate the input signal from the MQ8 sensor which is employed to calculate the instant approximate value of Rs thus determining the ratio and

the approximate concentration. For confirming the calibration and how accurate it is, a comparison with done between a theoretical and practical values. The maximum and minimum Rs values were calculated by assuming output signal of 0.83 V, for no hydrogen presence, and 4.89 V, for high hydrogen presence, which corresponds to very low and very high hydrogen concentration, respectively.

$$R_{s_{max}} = \frac{V_{cc}.R_L}{V_{R_L}} - R_L = \frac{5.00*10{,}000.00}{0.83} - 10{,}000.00 \cong 50{,}240.96\ \Omega(ADC = 170)$$

$$R_{s_{min}} = \frac{V_{cc}.R_L}{V_{R_L}} - R_L = \frac{5.00*10{,}000.00}{4.89} - 10{,}000.00 \cong 224.95\ \Omega(ADC = 1000)$$

For obtaining the real Ro value, it is important to mention that the MQ8 sensor was calibrated following the datasheet instruction and under the following conditions: on 25/01/2023, during a sunny day with a temperature of 22 ºC and a humidity of 60%, figure 34, the MQ8 sensor was calibrated in air. It is important to note that the sensor was calibrated in the shade to prevent direct contact with the sun from increasing the sensor's temperature. Using equation 5, the value of R0 can be obtained as following:

$$\frac{R_s}{R_0} = 70 \tag{5}$$

$$@temp: 21ºC\ 21\%\ Oxygen\ and\ 65\%\ humdity,$$

$$since\ R_0 = \frac{R_s}{70} = 717.728$$

$$R_{s\ open\ air} = 50507.17\ \Omega(real\ value) \qquad for\ ADC\ input\ signal\ 169\ \ (open\ air)$$

Perfect calibration conditions should be carried out with a calibration kit or in isolated and controlled environment. However, due to the lack of such equipment, the MQ8 sensor was calibrated in this way, which is also possible. However, it does not provide exact hydrogen concentration values. The hydrogen concentration provided after this calibration is an approximate value obtained using the power regression method. This concentration values obtained are sufficient for detecting hydrogen presence, thus, can be trusted to indicate hydrogen gas leakage.

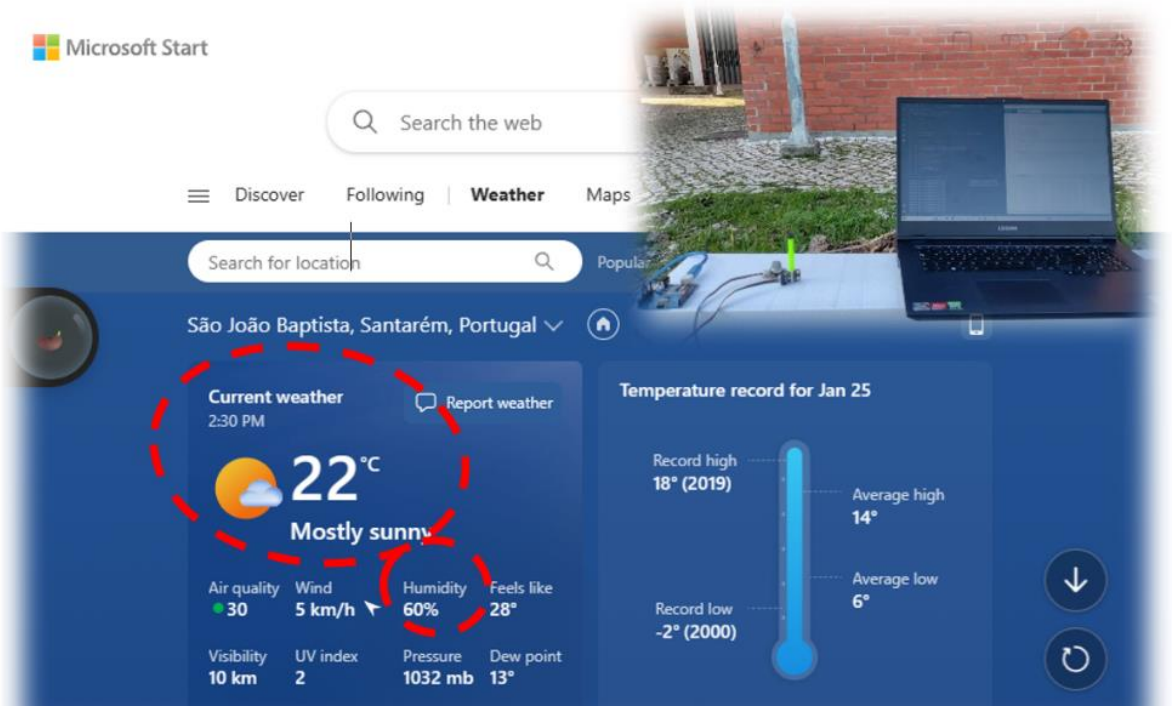


Figure 34 - MQ8 hydrogen sensor calibration (open air)

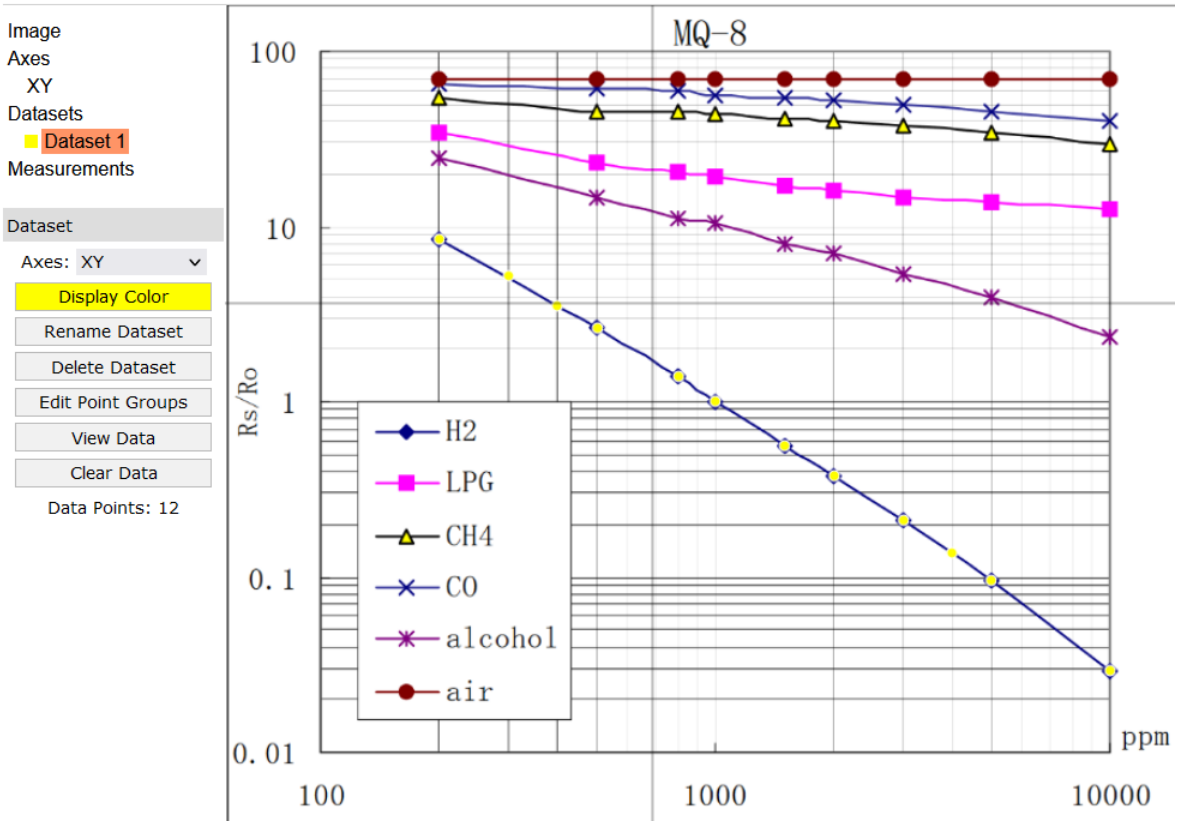


Figure 35- MQ8 hydrogen sensor graph (Rs/R0 vs ppm)

To calculate hydrogen approximate concentration based on the change in resistance, Rs, of the MQ8 sensor, a non-linear regression model is applied. The power regression model is used to calculate the approximate H_2 concentration in ppm. The model was obtained using the Excel formula and data from the MQ8 sensor data sheet. Table 4 is constructed using the online tool "WebPlotDigitizer" to obtain the coordinates of the MQ8 concentration graph, figure 35, which can be found on the datasheet.

Table 4 - MQ8 hydrogen sensor (Rs/R0 vs ppm) values

ppm	Rs/Ro
300	5.1163
400	3.5057
500	2.6176
800	1.4101
1000	1.0000
1500	0.5575
2000	0.3691
3000	0.2094
4000	0.1362
5000	0.0950
10000	0.0290

By alternating the values of concentration (in ppm) on the y-axis and rate of Rs/R0 on the x-axis, we can then apply the power regression model in excel and obtain the values of the constants “a” and “b”. This model will provide approximate concentration values (in ppm) based on the data obtained from the MQ8 data sheet.

R^2 represents the coefficient of determination, which is a statistical measure of the degree to which the independent variables (x) explain the variability of the dependent variable (y) in the model. By applying the Rs/R0 values obtained from the datasheet to the power regression model obtained, we can generate an approximate concentration of hydrogen gas

(in ppm) figure 36. Table 5 displays the variation in the values obtained between -4.6% and 3.5% for values closer to the inflection point. The model can therefore be relied upon to obtain an approximate value within +/- 5% of the real value.

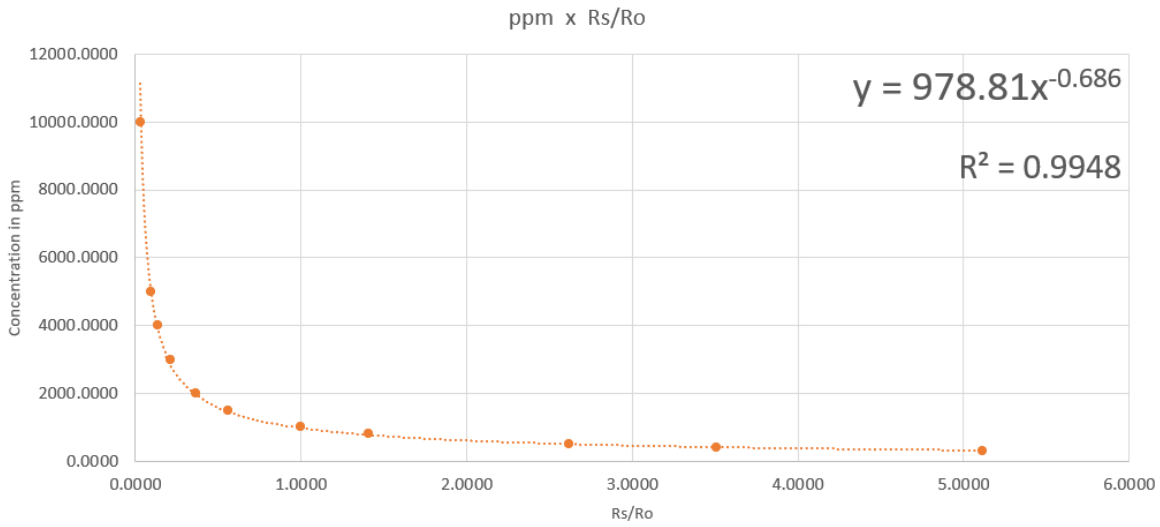


Figure 36 - MQ8 hydrogen sensor power regression model

Table 5 - Comparison of power regression model and datasheet values

Rs/Ro (Datasheet)	ppm (Datasheet)	ppm ($y = 978.81 \times x^{-0.686}$)	Difference %
5.1163	300	391.4149	+6.5
3.5057	400	413.9825	+3.5
2.6176	500	505.8384	+1.2
1.4101	800	773.2329	-3.3
1.0000	1000	978.8100	-2.1
0.5575	1500	1461.3630	-2.6
0.3691	2000	1939.2052	-3.0
0.2094	3000	2861.3044	-4.6
0.1362	4000	3841.9273	-4.0
0.0950	5000	4921.0367	-1.6
0.0290	10000	11099.2955	+11.0

4.1.5. Relays

The relay module is designed to be used with development boards such as ESP32. It provides the ability to switch high voltage to power devices from a single digital pin. Two of these relays are implemented into the system. The first is used to control the exit valve between the electrolyzer and the external storage tank and the second is for protect on the positive output terminal of the DC/DC buck converter that supplies the electrolyser with power. The positive terminal of the power supply is connected via the NO (Normally Open) pin and the negative terminal via the COM (common) pin of the relay. In this way, the circuit is open and when the relay actuates the connected device is powered.

Figure 37 - Relay model

4.1.6. Actuator

The solenoid valve is a normally closed electric solenoid bidirectional gas valve. It functions as an outlet valve between the electrolyzer and the external storage tank. It operates with 3 VDC and its main objective is to control the release of hydrogen gas into the fuel cell stack. The valve is controlled to open once the electrolyzer has a stable input voltage and the hydrogen production is at maximum rate.

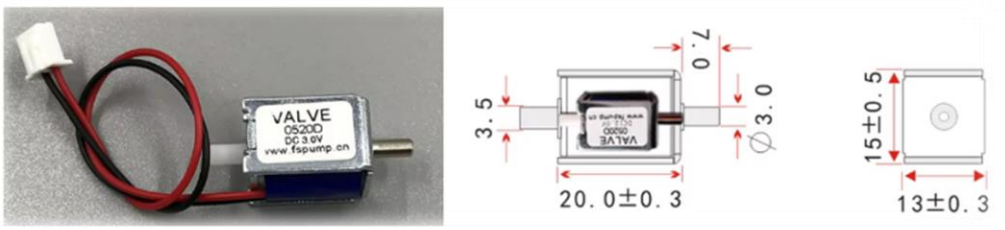


Figure 38 - Solenoid Valve

4.2. Data Storage

All the data obtained from the sensors is stored in a time series database on InfluxDB. InfluxDB is an open-source time series database optimized for storing and querying large volumes of data from sensors and IoT devices. It is designed to handle high volumes of real-time data which in turn allows for efficient data storage over time, providing a solid basis for analysis.

To store data on InfluxDB, the database is configured by creating a bucket. In InfluxDB, buckets are containers in which data is stored, and they define the data storage period. Each bucket has a specific API Token attributed. The API Token is an authentication key used to access the InfluxDB database securely. It is essential for integrating external applications with InfluxDB, allowing applications to write and read data programmatically. Once the setup is done, API Token, network, password and relative configuration is done on the Esp32, IoT device, and with configuration done on both sides it is possible to write data into the bucket through HTTP over Wi-Fi securely.

Figure 39 - InfluxDB data bucket

4.3. Data Visualization

For historical data monitorization, Grafana is configured to use InfluxDB as its data source, which allows for dynamic queries and visualizations based on the stored data. InfluxDB provides the time series data that is read via queries on Grafana, and these queries are used in the dashboards in Grafana. The Grafana recognizes data buckets from InfluxDB through API Tokens. Grafana is an open-source analysis and monitoring platform that allows users to create interactive and dynamic dashboards.

On a dashboard, the user can set up various panels that display different metrics and sensor readings, such as voltage, current, H_2 concentration and temperatures. For this project, a single dashboard is created to monitor the data stored in the database for defined periods of time range. Grafana has various refresh rates starting from 5s.

Figure 40 - Grafana history dashboard

4.4. Data Monitoring

Real-time data monitorization is achieved through a dashboard on Arduino IoT Cloud. As mentioned previously, Arduino IoT Cloud communicates directly, over Wi-Fi, with the ESP32 through variables that are synchronized in the sketch. The dashboard is interactive and allows controlling by switching it on/off and starting and stopping the hydrogen production, as well as introducing the hydrogen volume set point parameter and choosing whether the process is with or without control. The dashboard is updated in real-time and directly communicates with ESP32 over Wi-Fi through various variables. The dashboard is accessible on web browser and App, figures 43 and 44.

```
/*
  Sketch generated by the Arduino IoT Cloud Thing "Untitled 2"
  https://create.arduino.cc/cloud/things/2c58dc18-5135-45b8-8e96-6d013c3f6a08

  Arduino IoT Cloud Variables description

  The following variables are automatically generated and updated when changes are made to the Thing

  float ambientTemperature;
  float aproxDuration;
  float electrolyserCurrent;
  float electrolyserTemperature;
  float fuelCellTemperature;
  float hydrogenConcentration;
  float inputVoltage;
  float setPointHydrogen;
  int operationMode;
  bool alarmFlagA;
  bool alarmFlagB1;
  bool alarmFlagB2;
  bool alarmFlagC;
  bool alarmFlagD;
  bool systemOnOff;
  bool systemStart;
  bool systemStop;

  Variables which are marked as READ/WRITE in the Cloud Thing will also have functions
  which are called when their values are changed from the Dashboard.
  These functions are generated with the Thing and added at the end of this sketch.
*/
```

Figure 41 - Sketch variables on Arduino IoT

Variables on Arduino IoT Cloud, figure 42, can be configured to update “On change” or “Periodically”. The variables created for this project are all configured to update on change. The variables can also be configured as “READ/WRITE”, which are variable that can be read and written into, or “READ ONLY”, which are variable that can only be read and not modified during the process.

```
36   void initProperties(){
37
38     ArduinoCloud.setBoardId(DEVICE_LOGIN_NAME);
39     ArduinoCloud.setSecretDeviceKey(DEVICE_KEY);
40     ArduinoCloud.addProperty(ambientTemperature, READ, ON_CHANGE, NULL);
41     ArduinoCloud.addProperty(aproxDuration, READ, ON_CHANGE, NULL);
42     ArduinoCloud.addProperty(electrolyserCurrent, READ, ON_CHANGE, NULL);
43     ArduinoCloud.addProperty(electrolyserTemperature, READ, ON_CHANGE, NULL);
44     ArduinoCloud.addProperty(fuelCellTemperature, READ, ON_CHANGE, NULL);
45     ArduinoCloud.addProperty(hydrogenConcentration, READ, ON_CHANGE, NULL);
46     ArduinoCloud.addProperty(inputVoltage, READ, ON_CHANGE, NULL);
47     ArduinoCloud.addProperty(setPointHydrogen, READWRITE, ON_CHANGE, onSetPointHydrogenChange);
48     ArduinoCloud.addProperty(operationMode, READWRITE, ON_CHANGE, onOperationModeChange);
49     ArduinoCloud.addProperty(alarmFlagA, READ, ON_CHANGE, NULL);
50     ArduinoCloud.addProperty(alarmFlagB1, READ, ON_CHANGE, NULL);
51     ArduinoCloud.addProperty(alarmFlagB2, READ, ON_CHANGE, NULL);
52     ArduinoCloud.addProperty(alarmFlagC, READ, ON_CHANGE, NULL);
53     ArduinoCloud.addProperty(alarmFlagD, READ, ON_CHANGE, NULL);
54     ArduinoCloud.addProperty(systemOnOff, READWRITE, ON_CHANGE, onSystemOnOffChange);
55     ArduinoCloud.addProperty(systemStart, READWRITE, ON_CHANGE, onSystemStartChange);
56     ArduinoCloud.addProperty(systemStop, READWRITE, ON_CHANGE, onSystemStopChange);
57
58   }
59
```

Figure 42 - Variables intiation in thingProperties.h libarary

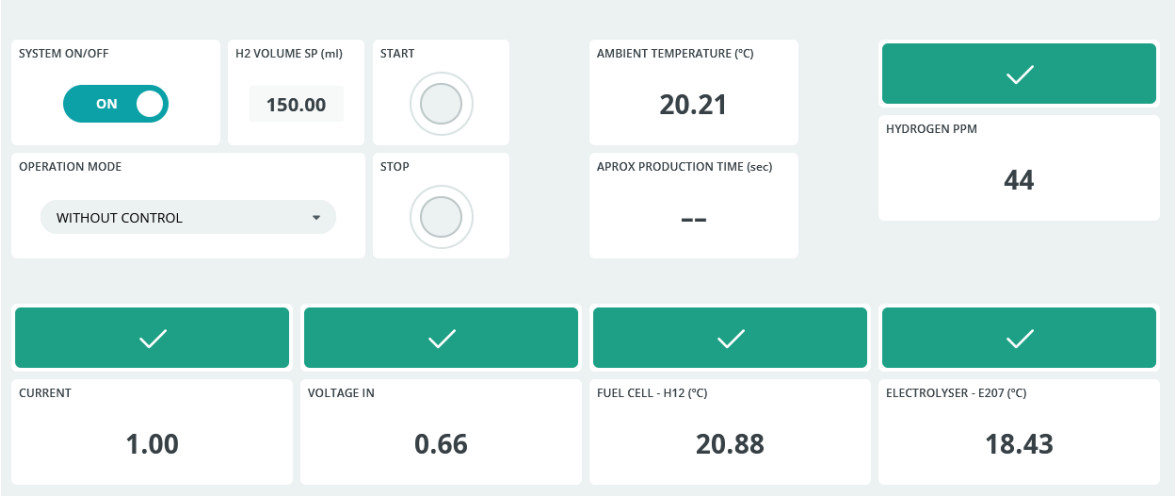


Figure 43 - Control dashboard

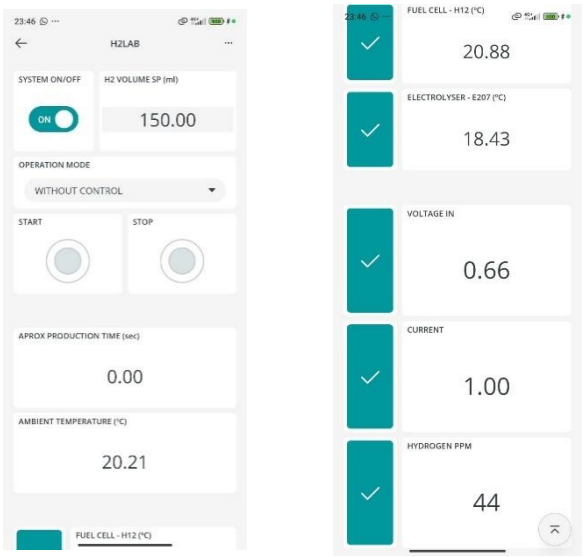


Figure 44 - Control dashboard (App)

5. SYSTEM OPTIMIZATION

A user-friendly interface alongside two main controls, namely: input voltage control and hydrogen production control were developed into the system and has also contributed to its optimization. A direct relationship between the hydrogen generation rate and the input voltage, as presented in figure 21 (see chapter 4). Therefore, it is crucial to regulate input voltage within the operation limits of the electrolyzer. This is accomplished via the DC/DC Buck converter. In addition, controlling hydrogen production is crucial for system optimization and is accomplished by user feedback regarding the necessary quantity. Both cycles operate together to achieve the desired results.

5.1. Voltage control

5.1.1. DAC signal control

Since the voltage provided by the solar charge controller is 24 V, the DC/DC buck converter is responsible for stepping it down to 0-14 V. The XL4016 buck converter used has an adjustable output voltage powering the electrolyzer and it is possible to obtain any desired value between 1.25 V and 35 V. The converter regulates the output voltage by maintaining a reference voltage of 1.25 V at the feedback pin, figure 45.

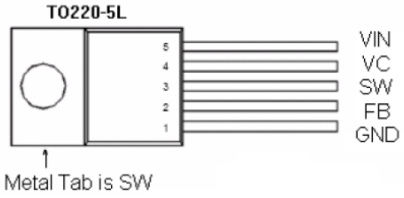


Figure 45- XL4016 pin layout

This is achieved by adjusting the potentiometer, which acts as a variable resistor of 10000 ohm, which makes it possible to control the voltage drop across itself. The output voltage is the sum of the feedback reference voltage, and the voltage across the potentiometer.

To optimize the system and be able to dynamically control the production of hydrogen, it is important to be able to control the output voltage. This is achieved by varying DAC signals, which are introduced into the feedback node [29]. The ESP32 has two built-in DAC channels, allowing you to generate analog output voltages from a digital signal. The digital value between 0–255, 8-bit resolution, is converted to a corresponding analog signal between 0-3.3 V. However, in practice, the maximum output voltage of the ESP32 DAC (pins 25 and/or 26) is usually lower than 3.3V due to voltage drop and internal circuitry limitations. Regarding the lower limit, it is set to 1 V as it has been observed with values lower than 1 V, the voltage control doesn't function. This is mainly due to the limitation of DAC pin current, max 20 mA, sourcing on the ESP32 microcontroller. When the DAC voltage is lower than 1 V, it pulls the feedback pin voltage lower than the reference value.

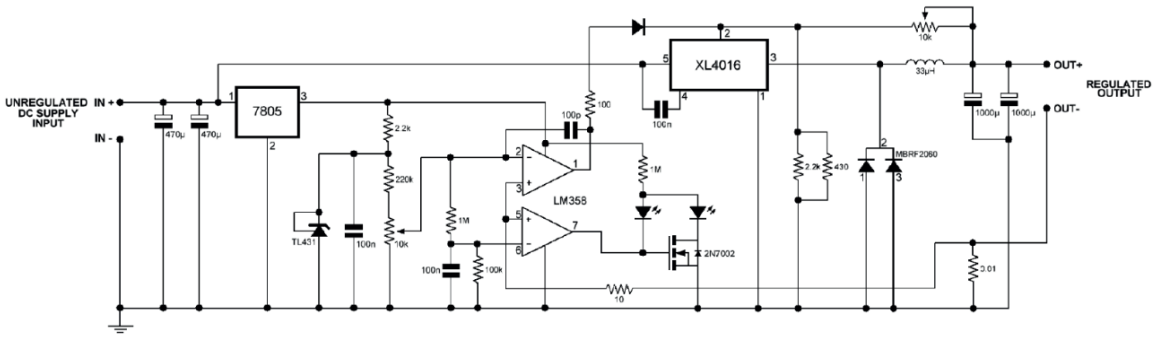


Figure 46 - DC/DC Buck converter circuit

By testing various lower and upper limits of the DAC voltage, the control is functional within the range of 1-3 V. With an analog voltage signal from the microcontroller at the feedback node of the converter, any output voltage, within the output range, is attainable instead of a fixed value that must be adjusted manually. Figure 47 lays out the FB node circuit, it is possible to obtain the output voltage based on the values of the reference voltage and the DAC voltage.

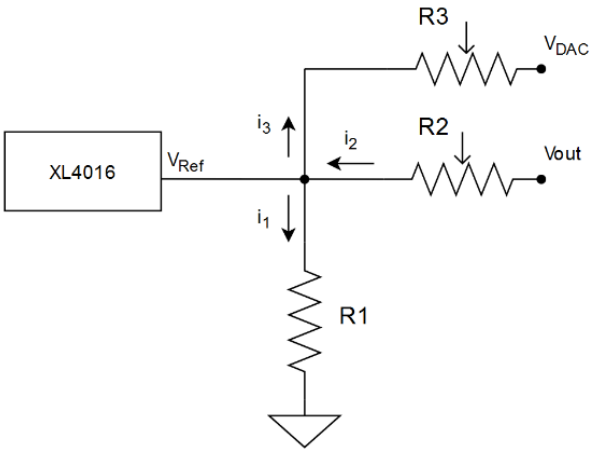


Figure 47 - Feedback node

Analyzing the node and applying Kirchhoff's law, the output voltage can be defined by giving the following:

$$i_2 = i_1 + i_3 \tag{6}$$

$$i_1 = \frac{V_{Ref}}{R1} \tag{7}$$

$$i_3 = \frac{(V_{Ref} - V_{DAC})}{R3} \tag{8}$$

$$V_{out} = V_{Ref} + i_2 R2 \tag{9}$$

Replacing equations 6, 7, and 8 into equation 9, the output voltage can be given as:

$$V_{out} = V_{Ref} + \left(\frac{V_{Ref}}{R1} + \frac{V_{Ref}}{R1}\right) R2$$

$$V_{out} = V_{Ref}\left(1 + \frac{R2}{R1} + \frac{R2}{R3}\right) - \frac{R2}{R3} V_{DAC} \tag{10}$$

Having that R2 and R3 are unknown, their values depend on the value of the reference voltage as well as the DAC voltage and pretended output voltage ranges (minimum and maximum). From the circuit of the XL4016 converter, figure 46, the value of R1 can be obtained as 359 ohms which can be calculated from two parallel resistors of 2200 ohms and 430 ohms. R2 and R3 are potentiometers that function as a variable resistor and their values are calculated using the following equations 11 and 12 [29].

$$R2 = \frac{R1}{-V_{Ref}\left(\frac{V_{DAC_L} - V_{DAC_H}}{(V_{out_L} - V_{out_H} + V_{DAC_L} - V_{DAC_H}) \times V_{Ref} - (V_{DAC_L} \times V_{out_L}) + (V_{DAC_H} \times V_{out_H})}\right)} \quad (11)$$

$$R3 = R2 \times R1 \times \left(\frac{V_{DAC_L} - V_{Ref}}{(R1 \times V_{Ref}) + (R2 \times V_{Ref}) - (R1 \times V_{out_L})}\right) \quad (12)$$

Table 6 - FB node resistors

Resistor	Value (ohms)
R1	359
R2	3204
R3	502

Since all three resistors values, table 6, V_{out} and V_{DAC} ranges are known, it is possible to demonstrate the converter's output voltage control graphically, figure 48. By achieving this, it is possible to dynamically control the output voltage and supply the PEM electrolyzer with a range of 0-14 VDC. Table 7 presents a comparison between the theoretical and real values of V_{out} and V_{DAC} as well as the corresponding steps, between 0-255, to generate the DAC signal.

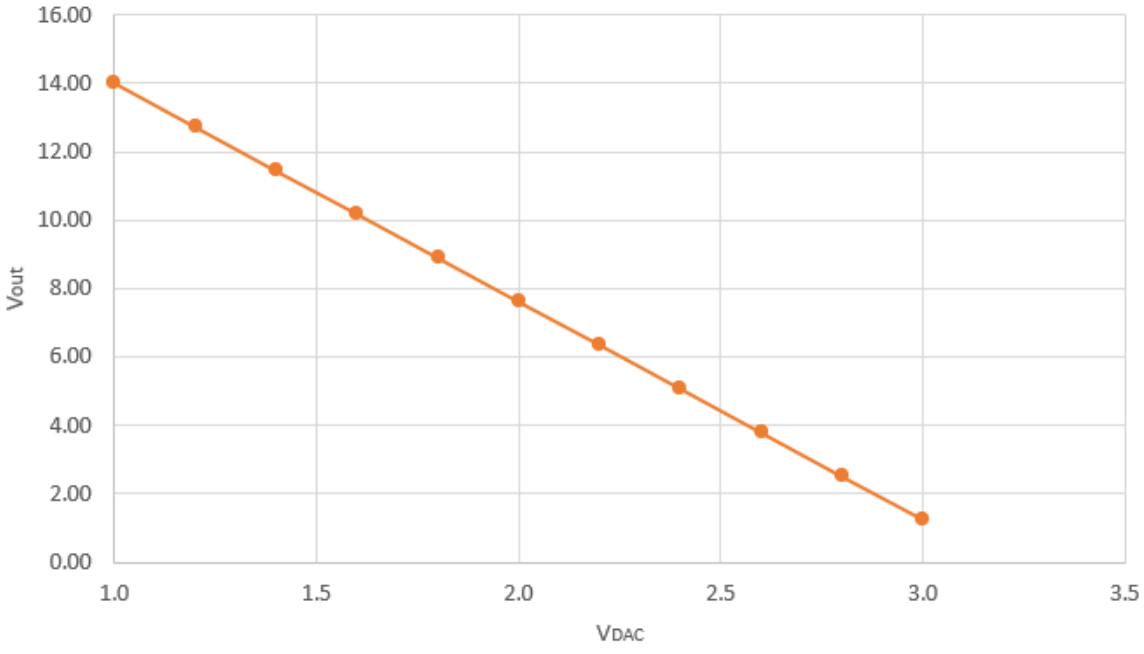


Figure 48 Vout vs DAC signal graph

Table 7 - Vout vs DAC signal values

Step (8-bit resolution)	Theorical		Real	
	V_{DAC}	V_{out}	V_{DAC}	V_{out}
95	1.0	14.00	1.05	13.94
110	1.2	12.73	1.35	12.54
125	1.4	11.45	1.53	11.13
140	1.6	10.17	1.73	9.41
155	1.8	8.90	1.91	8.60
170	2.0	7.62	2.09	7.91
185	2.2	6.34	2.17	6.33
200	2.4	5.07	2.46	4.82
215	2.6	3.79	2.63	3.26
230	2.8	2.51	2.82	2.48
245	3.0	1.24	3.01	1.30

5.1.2. Voltage Ramp

For the electrolyzer to function, it demands a certain amount of energy to initiate the electrochemical reaction. At the beginning, this energy demand can cause current peaks. This is due to the interface between the electrode and electrolyte, which forms a double-layer capacitance. When voltage is applied, this capacitance must be charged, which leads to an initial peak in current.

To protect the cells from potential damage, a voltage ramp is implemented. This ramp gradually increases the voltage, ensuring a smooth transition and avoiding sudden peaks. As well as the voltage ramp is applied in both controlled and uncontrolled modes and is managed through the voltageRamp() function.

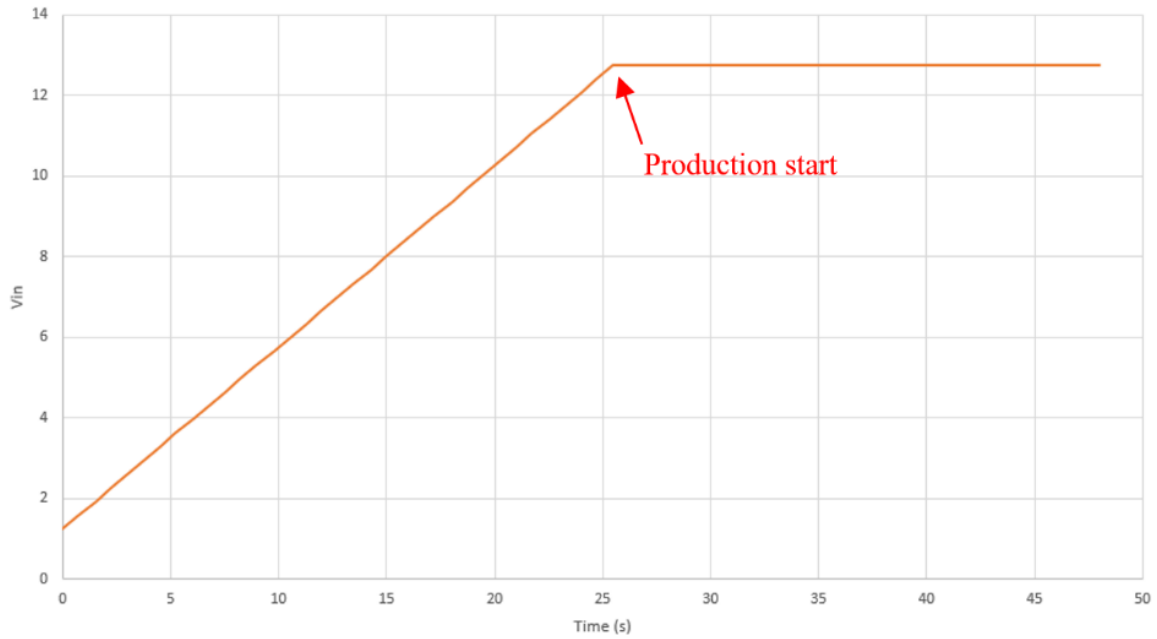


Figure 49 - Voltage ramp graph

In addition, some modifications to the circuit of the DC/DC converter were necessary to accomplish the DAC signal control over the output voltage. A part of the circuit of the converter is an LM385 comparator which serves as a protection for the converter by limiting the current. It provides a reference voltage which is compared against the voltage across the current-sensing resistor (0.01 ohms). When the sensed voltage exceeds the reference, the LM358 reduces the feedback voltage, limiting the current through XL4016.

To disable this mechanism from the circuit, the diode connected to the LM385 to the FB pin of the XL4016 has been removed, figure 50, thus interrupting the LM385 feedback.

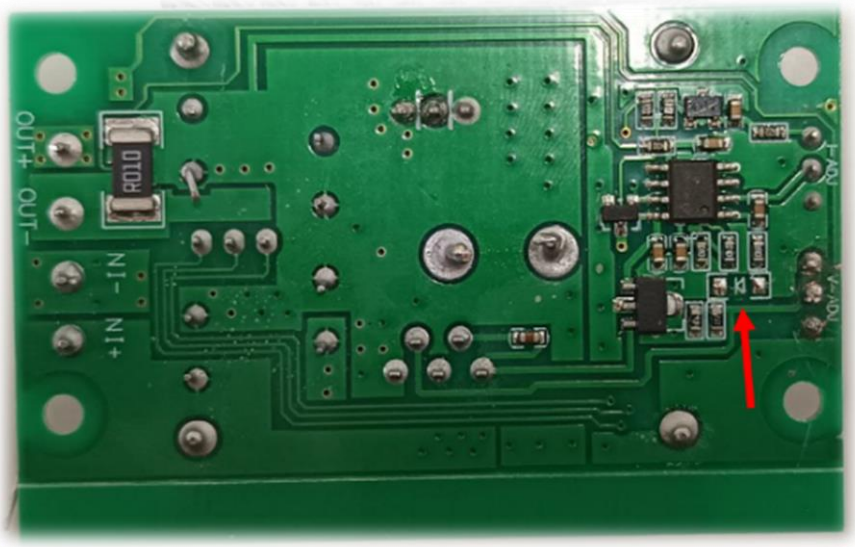


Figure 50 - DC/DC Converter back view

5.2. Hydrogen production control

The hydrogen production control is based on the volume of hydrogen defined by the user. In industrial systems, the amount produced is measured using a hydrogen gas flowmeter which continuously gives feedback to the control loop. However, due to the small scale of the system, the use of a flowmeter is not possible as there is no such flowmeter for small scale projects. As a solution, an open-loop control, figure 51, is based on time periods where the production rate per minute is known from previous tests is applied. By knowing the production rate and the required volume, it is possible to calculate the necessary time to produce the desired volume of hydrogen gas.

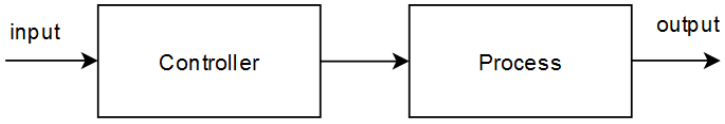


Figure 51 - Control process

This open-loop control is achieved using the internal clock of the microcontroller, ESP32, where the exact instant of the beginning of production is recorded and continuously compared to the calculated necessary period. Within this control, once the duration of the production reaches the calculated duration, the production is terminated. This is achieved through the checkTimer() function within the controlTask(). The checkTimer function is designed to monitor the progress of a timer and act when the specified time duration is reached. It operates by first checking the timer flag and then compares the elapsed time to the required duration, once the elapsed time exceeds the required duration the system is reset, and the production process is terminated.

As mentioned previously, the hydrogen production rate is known as 230ml/min. Thus, the volume of hydrogen to be produced within a specific period can be given by equation 13.

$$V(t) = R * t_{elap} \tag{13}$$

Where:

- V(t): Volume of hydrogen produced at time t.
- R: Known production rate (ml/min).
- T_{elap}: Time elapsed since the beginning of production (min).

For the desired volume of hydrogen required, by rearranging equation 13 the required production duration can be obtained by:

$$t_{req} = \frac{V_{req}}{R} \tag{14}$$

Where:

- V_{req}: Hydrogen volume setpoint by user.
- R: Known production rate (ml/min).
- t_{req}: calculated duration to produce V_{req} (min).

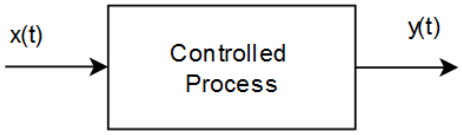


Figure 52 - Open loop controller process

Since the transfer function relates the input denoted as *x(t)*, duration of production, and the output denoted as *y(t)*, volume of hydrogen, , through the controlled process, figure 52, then the mathematical representation in the time domain and Laplace transforms it can be given by equation 15 and 16, respectively.

$$y(t) = g(t) * x(t) \tag{15}$$

$\mathscr{L}$: $$Y(s) = G(s) \cdot X(s) \tag{16}$$

Considering *g(t)* is the impulse of the controller process.

T.F: $$\frac{Y(s)}{X(s)} = G(s) \tag{17}$$

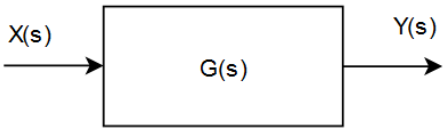


Figure 53 - open loop controller (Laplace)

Key Characteristics of the Open-Loop System:

- No Feedback: The system directly applies the input *x(t)* to the process without measuring or correcting the output *y(t)*.
- Control Action: The behavior of *y(t)* is determined only by *x(t)* and the dynamics of the process represented by *G(s)*.

In conclusion, the open-loop control system for hydrogen production developed provides a simple, however effective method for managing the volume of hydrogen produced. By depending on predefined production rates and durations calculated from user defined volume, the system eliminates the need for real-time feedback, known as closed-loop control, which simplifies the design and operation. Although the lack of feedback limits the system, the simplicity and practicality of this approach makes it a good solution for controlled hydrogen production for the equipment used within the scope of this project.

6. CONCLUSION

As hydrogen is considered a clean and sustainable alternative to fossil fuels and various technologies are being developed to produce hydrogen through renewable sources like water electrolysis powered by solar energy, namely referred to as green hydrogen. The focus of this project is to design and develop an optimization and monitoring system of a hydrogen production prototype powered photovoltaic energy. The integration of a PEM electrolyzer with power and control units consisting of PV panel, batteries, and solar charge controller and ESP32 microcontroller, respectively, permits the demonstration of a green hydrogen production process on a laboratory scale.

Within the scope of the project, creating a user-friendly system that is effectively capable to use solar energy to power a PEM electrolyzer and produce hydrogen from it, which has potential applications in the future energy systems. As it is not possible to integrate a hydrogen flowmeter into the system due to its small scale, an open-loop control was applied, by controlling the production duration, through the internal timer of the Esp32 microcontroller. For the DC/DC buck converter, the DAC signal control of the ESP32 is implemented to control the PEM electrolyzer by gradually increasing the input voltage until the maximum production rate point. This ensures efficient hydrogen production at a rate of 230 ml/min. Moreover, the real-time data acquisition and monitoring system implemented using Arduino IoT Cloud made it possible for the system to have a user friendly and an efficient interface to control and monitor the hydrogen production process. By including sensors for temperature and hydrogen leakage detection, the system can perform emergency shutdown to protect the equipment from any damage. This ensures that the setup is suitable for real-world application.

Despite the various aims achieved in this project, certain limitations were encountered, which provide opportunities for future improvements. The system operates with an open-loop control system, which is functional, yet it is limited to pre-calculated output and lacks the adaptability of a closed-loop feedback system. Future development could integrate closed-loop controls to dynamically adjust the production rate on demand in response to real-time feedback, which enhances efficiency. Additionally, the lack of hydrogen gas flowmeters at the laboratory scale has limited the system from having production volume measurements.

Another area for future development is in integrating hydrogen storage unit. The current setup focuses on production and monitoring, adding a compressor or a storage solution would make the system more practical. There is also space for the integration of more sensors, such as water quality, conductivity, and pressure, which could also be incorporated to improve both operational control and system efficiency.

In conclusion, this project demonstrates the integration of renewable energy sources, control systems, and PEM hydrogen production technology, incorporated together within a functional and safe laboratory-scale prototype. While certain limitations are obvious, the system represents a significant step toward scalable, sustainable hydrogen production solutions. By discussing these limitations and expanding the system's capabilities, the project offers a foundation for future development in green hydrogen technology.

REFERENCES

[1] Hassan, Q., Viktor, P., Al-Musawi, T. J., Ali, B. M., Algburi, S., Alzoubi, H. M., ... & Jaszczur, M. (2024). The renewable energy role in the global energy Transformations. Renewable Energy Focus, 48, 100545.

[2] Ramachandran, R., & Menon, R. K. (1998). An overview of industrial uses of hydrogen. International journal of hydrogen energy, 23(7), 593-598.

[3] Qazi, U. Y. (2022). Future of Hydrogen as an Alternative Fuel for Next-Generation Industrial Applications; Challenges and Expected Opportunities. Energies, 15(13), 4741

[4] Kalamaras, C. M., & Efstathiou, A. M. (2013). Hydrogen production technologies: current state and future developments. In Conference papers in science (Vol. 2013). Hindawi

[5] Mitchell, S. (2021, February 11). ScottishPower sets sights on green hydrogen revolution. Ethical Marketing News. https://ethicalmarketingnews.com/scottishpower-sets-sights-on-green-hydrogen-revolution

[6] Kumar, S. S., & Lim, H. (2022). An overview of water electrolysis technologies for green hydrogen production. Energy reports, 8, 13793-13813.

[7] Sloop, J. L. (1978). Liquid hydrogen as a propulsion fuel, 1945-1959 (Vol. 4404). Scientific and Technical Information Office, National Aeronautics and Space Administration.

[8] Qureshi, F., Yusuf, M., Ibrahim, H., Kamyab, H., Chelliapan, S., Pham, C. Q., & Vo, D. V. N. (2023). Contemporary avenues of the Hydrogen industry: Opportunities and challenges in the eco-friendly approach. Environmental Research, 115963

[9] Ji, M., & Wang, J. (2021). Review and comparison of various hydrogen production methods based on costs and life cycle impact assessment indicators. International Journal of Hydrogen Energy, 46(78), 38612-38635

[10] Okolie, J. A., Patra, B. R., Mukherjee, A., Nanda, S., Dalai, A. K., & Kozinski, J. A. (2021). Futuristic applications of hydrogen in energy, biorefining, aerospace, pharmaceuticals, and metallurgy. International Journal of Hydrogen Energy, 46(13), 8885-8905

[11] Kumar, S. S., & Himabindu, V. (2019). Hydrogen production by PEM water electrolysis–A review. Materials Science for Energy Technologies, 2(3), 442-454

[12] Kalamaras, C. M., & Efstathiou, A. M. (2013). Hydrogen production technologies: current state and future developments. In Conference papers in science (Vol. 2013). Hindawi

[13] Smolinka, T., Bergmann, H., Garche, J., & Kusnezoff, M. (2022). The history of water electrolysis from its beginnings to the present. In Electrochemical Power Sources: Fundamentals, Systems, and Applications (pp. 83-164). Elsevier

[14] Rousseau, R., Etcheverry, L., Roubaud, E., Basséguy, R., Délia, M. L., & Bergel, A. (2020). Microbial electrolysis cell (MEC): Strengths, weaknesses and research needs from electrochemical engineering standpoint. Applied Energy, 257, 113938

[15] Chi, J., & Yu, H. (2018). Water electrolysis based on renewable energy for hydrogen production. Chinese Journal of Catalysis, 39(3), 390-394

[16] Eyvaz, M. (Ed.). (2018). Advances In Hydrogen Generation Technologies. BoD–Books on Demand

[17] Russell, J. H., Nuttall, L. J., & Fickett, A. P. (1973). Hydrogen generation by solid polymer electrolyte water electrolysis. Am. Chem. Soc. Div. Fuel Chem. Prepr, 18, 24-40

[18] Carmo, M., Fritz, D. L., Mergel, J., & Stolten, D. (2013). A comprehensive review on PEM water electrolysis. International journal of hydrogen energy, 38(12), 4901-4934 [19]

[19] Titanium Fiber Felt – gas diffusion layer material. (n.d.). https://www.sintered-mesh.com/technology/gdl-material-requirements-pemec-hydrogen-production.html

[20] Wang, Y., Pang, Y., Xu, H., Martinez, A., & Chen, K. S. (2022). hydrogen infrastructure development–a review. Energy & Environmental Science, 15(6), 2288-2328

[21] Edwards, P. P., Kuznetsov, V. L., David, W. I., & Brandon, N. P. (2008). Hydrogen and fuel cells: Towards a sustainable energy future. Energy policy, 36(12), 4356-4362

[22] Litster, S., & McLean, G. J. J. O. P. S. (2004). PEM fuel cell electrodes. Journal of power sources, 130(1-2), 61-76

[23] Wang, Y., Diaz, D. F. R., Chen, K. S., Wang, Z., & Adroher, X. C. (2020). Materials, technological status, and fundamentals of PEM fuel cells–a review. Materials today, 32, 178-203

[24] ALL ABOUT FUEL CELLS HOW DO THEY WORK. (n.d.). Copyright 2020 - 2022 Climate Change Trust and Universal Smart Batteries. https://www.fuelcellscars.com/Fuel_Cells_All_About_How_Do_They_Work.htm

[25] ThalesNano Inc. (2024, August 29). H-Genie Lite hydrogen generator. ThalesNano. https://thalesnano.com/products-and-services/h-genie-lite/

[26] *ESP32-DevKitC-1—Arduino ESP32 latest documentation*. (n.d.). https://docs.espressif.com/projects/arduino-esp32/en/latest/boards/ESP32-DevKitC-1.html

[27] Gao, J., Dong, X., Tian, Q., & He, Y. (2023). Carbon nanotubes reinforced proton exchange membranes in fuel cells: An overview. International Journal of Hydrogen Energy, 48(8), 3216-3231.

[28] arduinokitproject.com. (2024, June 21). Interface MQ2 Gas/Smoke Sensor with Arduino: Step-by-Step Guide. ARDUINOKIT PROJECT. https://arduinokitproject.com/mq2-gas-senser-arduino-tutorial/

[29] Glaser, C. Methods of output-voltage adjustment for DC/DC converters. Analog Design Journal SLYT777 Q, 3, 2019.

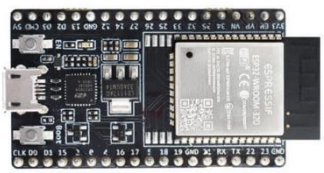
ANNEX

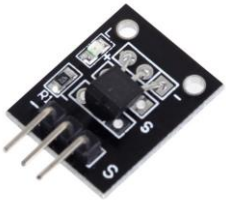
Annex A

	**Item**	**Quantity**	**Price**
	E207 Electrolyzer 230	1	€ 500.0
	Links	Website: HERE User manual: HERE	
	Company	QUINTECH, Germany	
	About		
	The Electrolyzer 230 is a seven-cell PEM electrolyser stack with water storage tank mounted on a black base plate, to produce hydrogen. There are seven cells within the Electrolyzer 230, each with an electrode area of 16 cm^2.		

	**Item**	**Quantity**	**Price**
	A153 Storage 80	1	€ 65.00
	Links	Website: HERE	
	Company	QUINTECH, Germany	
	About		
	The Storage 80 is a gas storage tank for hydrogen and oxygen that can hold up to 80 mL of gas.		

	**Item**	**Quantity**	**Price**
	H-12 PEM FC System 12 W	1	€ 420.00
	Links	Website: HERE User manual: HERE	
	Company	QUINTECH, Germany	
	About		
	The H-12 PEM FC System is a fuel cell stack of 13 cells with a rated power of 12W. The fuel cell comes with an integrated fan and casing and uses H2 and Air to produce electricity.		

	**Item**	**Quantity**	**Price**
	ESP32	1	€ 9.80
	Links	Website: - Datasheet: HERE	
	Company	ESPRESSIF SYS.	
	About		
	ESP32-WROOM-32 is a powerful, generic Wi-Fi + Bluetooth LE MCU module that targets 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.		



Item	Quantity	Price
DS18B20 Sensor	1	€ 3.90
Links	Website: HERE Datasheet: HERE	
Company	DALLAS semiconductor	
About		
The DS18B20 Digital Thermometer provides 9 to 12-bit(configurable) temperature readings which indicate the temperature of the device.		

Item	Quantity	Price
Wattímetro digital Sensor	1	€ 7.95
Links	Website: HERE Datasheet: HERE	
Company	TEXAS INSTRUMENTS	
About		
Gravity: I2C Digital Wattmeter is a high-resolution, high-precision, large-scale measurement module that can measure the voltage, current, and power of various electronic modules and electrical equipment within 26V 8A.		

Item	Quantity	Price
MLX90614 Sensor	1	€ 18.90
Links	Website: HERE Datasheet: HERE	
Company	MELEXIS	
About		
The MLX90614 is an Infra-Red thermometer for noncontact temperature measurements. Both the IR sensitive thermopile detector chip and the signal conditioning ASSP are integrated in the same TO-39 can.		

Item	Quantity	Price
MQ8 Sensor	1	€ 9.80
Links	Website: HERE Datasheet: HERE	
Company	HANWEI ELETRONICS	
About		
Hydrogen gas sensor. It has high sensitivity to Hydrogen (H2) and less sensitivity to alcohol, LPG,cooking fumes. Hydrogen gas is highly flammable and will burn in air at a very wide range of concentrations.		

Item	Quantity	Price
Relay	1	-
Links	Website: - Datasheet: HERE	
Company	SONGLE RELAY	
About		
It is a relay for 5V signal control that can be applied and used in areas requiring various contact control. It can receive a 5V signal and control the power supply up to 250V (10A). It is a compact, sealed 5-pin type, and can control NO and NC types.		

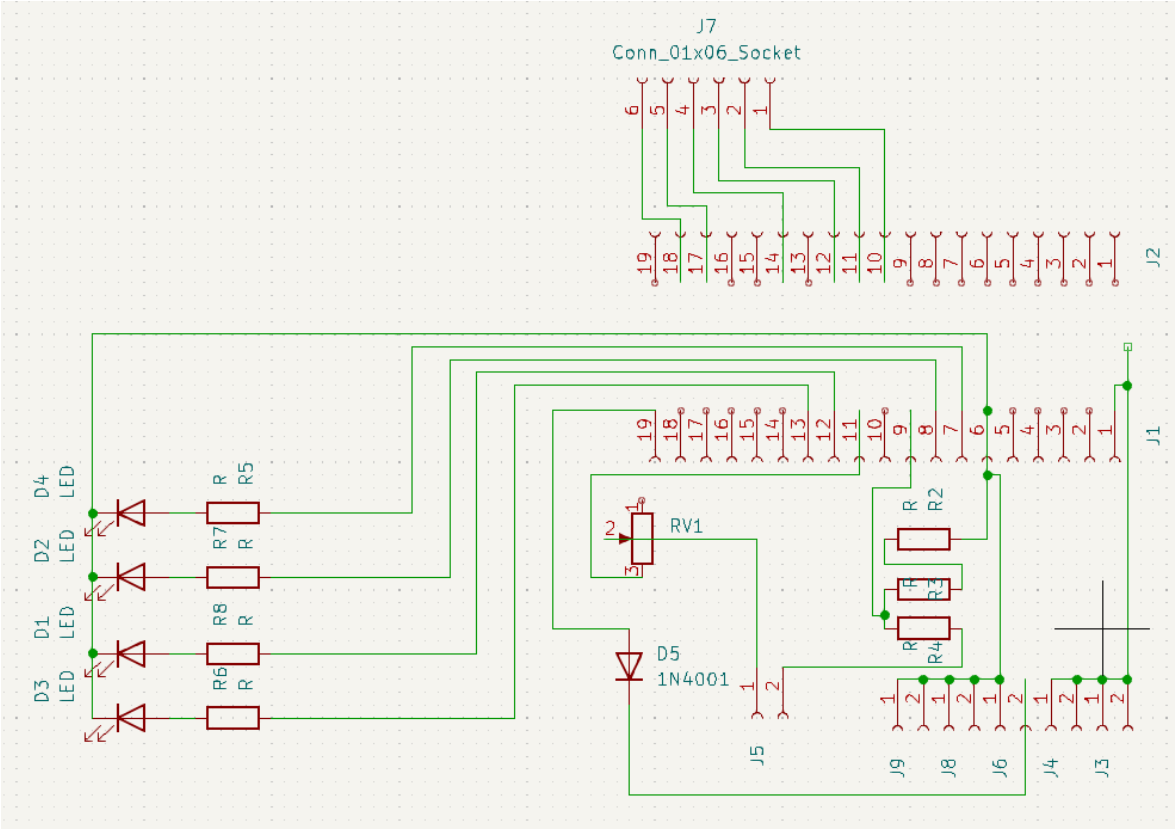
Item	Quantity	Price
Air valve	1	€ 3.45
Links	Website: HERE Datasheet: -	
Company	FSPUMP	
About		
Small electric mini solenoid valve, normally closed, bidirectional gas air valve, N/C, DC 3V,		

Item	Quantity	Price
SCC-20A Solar charge controller	1	€ 14.12
Links	Website: HERE Datasheet: HERE	
Company	-	
About		
A PWM solar charge controller regulate current from PV panel at an output of12/24 VDC. It has various configurations and can manage electric battery storage.		

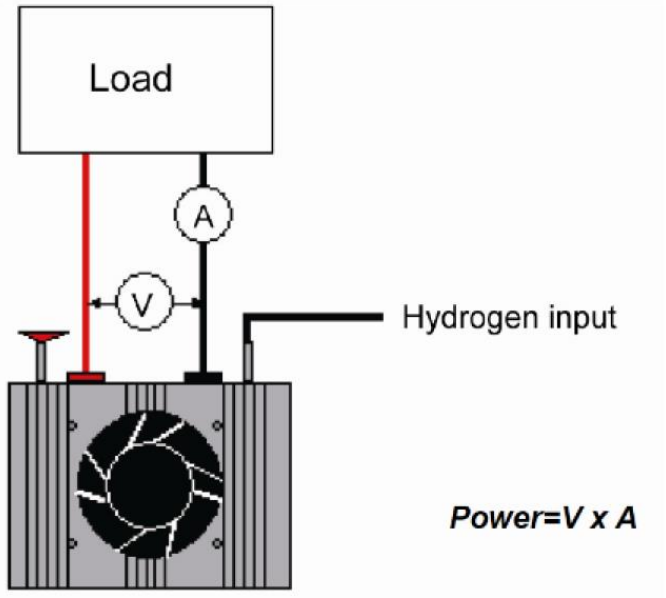
Item	Quantity	Price
DC/DC XL4016 Converter	1	€ 9.90
Links	Website: HERE Datasheet: HERE	
Company	AZ-Delivery	
About		
The XL4016 is a 180 KHz fixed frequency PWM buck (step-down) DC/DC converter, capable of driving an 8A load with high efficiency, with 1.25-35 Vout.		

Item	Quantity	Price
PV panel	1	€ 81.78
Links	Website: HERE Datasheet: HERE	
Company	Xunzel	
About		
A 185W high efficiency photovoltaic panel for 24V systems for charging and maintaining batteries in off-grid systems		

Annex B


J7
Conn_01x06_Socket
J2
J1
D4
LED
D2
LED
D1
LED
D3
LED
R
R5
R7
R8
R6
RV1
R2
R3
R4
D5
1N4001
J5
J9
J8
J6
J4
J3

Annex C



*Fuel cell connection diagram

Annex D

```
/*
  Sketch generated by the Arduino IoT Cloud Thing "Untitled 2"
  https://create.arduino.cc/cloud/things/2c58dc18-5135-45b8-8e96-6d013c3f6a08

  Arduino IoT Cloud Variables description

  The following variables are automatically generated and updated when changes are made to the
Thing

  float ambientTemperature;
  float aproxDuration;
  float electrolyserCurrent;
  float electrolyserTemperature;
  float fuelCellTemperature;
  float hydrogenConcentration;
  float inputVoltage;
  float setPointHydrogen;
  int operationMode;
  bool alarmFlagA;
  bool alarmFlagB1;
  bool alarmFlagB2;
  bool alarmFlagC;
  bool alarmFlagD;
  bool systemOnOff;
  bool systemStart;
  bool systemStop;

  Variables which are marked as READ/WRITE in the Cloud Thing will also have functions
  which are called when their values are changed from the Dashboard.
  These functions are generated with the Thing and added at the end of this sketch.
*/

#include "thingProperties.h"
#include <Arduino.h>
#include <DFRobot_INA219.h>
#include <DallasTemperature.h>      // Includes OneWire.h library
#include <DFRobot_MLX90614.h>       // Includes Wire.h library
#include <driver/dac.h>
#include <InfluxDbClient.h>
#include <InfluxDbCloud.h>

#if defined(ESP32)
  #include <WiFiMulti.h>
  WiFiMulti wifiMulti;
  #define DEVICE "ESP32"
```

```
  #elif defined(ESP8266)
  #include <ESP8266WiFiMulti.h>
  ESP8266WiFiMulti wifiMulti;
  #define DEVICE "ESP8266"
  #endif



/*
  InfluxDB configuration: WiFi SECRET_SSID & SECRET_OPTIONAL_PASS defined from Arudion IoT Cloud
  URL, TOKEN, ORG, BUCK are obtained from InfluxDB
  Time zone info defined based on location
*/
#define WIFI_SSID "Redmi Note 13 5G"
#define WIFI_PASSWORD "from1to8"
#define INFLUXDB_URL "https://eu-central-1-1.aws.cloud2.influxdata.com"
#define INFLUXDB_TOKEN "JSGw96scLlfVp3JaU9-
yMevyqumbsEupcW62DG83vUuFMam8KbnGYFokHxv_yherNNje_CSrgEahi6EvKIna9Q=="
#define INFLUXDB_ORG "71b8b8322903983e"
#define INFLUXDB_BUCKET "H2LAB"
#define TZ_INFO "UTC0"

// Hardware pin assignments
#define sensor_A_Input          19      // Input GPIO for 18B20 Temperature sensor (sensor_A)
#define sensor_D_Input          18      // Input GPIO for MQ8 Hydrogen sensor (sensor_D)
#define actuator_A_Output       23      // Output GPIO for valve activiation relay
#define actuator_B_Output        5      // Output GPIO for electrolyzer input relay
#define LED_ONOFF               15
#define LED_Startup             16
#define LED_Production           0
#define LED_Alarme               4

// Calibration constants
#define ina219Reading_mA       130      // Calibration constant for INA219 power sensor (sensor_B)
#define extMeterReading_mA     132      // Calibration constant for INA219 power sensor (sensor_B)

// ADC and bit resolution
#define ADC_voltage            3.3
#define Bit_Resolution        4095

// Constants used in calculations, better declared with `const` for type safety
const float RL = 10000.0;                     // RL value = 10k ohms for MQ8 sensor
const float Vcc = 5.0;                        // Vcc = 5.0V for MQ8 sensor
const float a = 978.81;                       // Power regression constant 'a' for MQ8
const float b = -0.686;                       // Power regression constant 'b' for MQ8
const float R0 = 721.531;                     // Base resistance in clean air for MQ8
```

```
// Sensor thresholds, using `const` for clear data type definition
const float sensor_A_Cond = 50.0;              // Fuel Cell operating temperature limit (°C)
const float sensor_B_Cond1 = 14.0;             // Electrolyzer permissible voltage (V)
const float sensor_B_Cond2 = 4.4;              // Electrolyzer permissible current (mA)
const float sensor_C_Cond = 50.0;              // Electrolyzer permissible temperature (°C)
const float sensor_D_Cond = 200.0;             // Hydrogen gas leak detection threshold (ppm)

// Define constants for production rate in ml/min and corresponding voltage levels
const float ProductionRate = 230.0;            // Maximum production rate in ml/min

// Boolean flags for program status tracking
bool timerStatus = false;                      // Tracks whether the timer is active or inactive
bool productionStatus = false;                 // Indicates if production is currently active

// Variables for controlling and tracking system operation parameters
int Step = 245;                                // step for DAC signal in 8bit resolution
int StepAux = 245;                             // Auxiliary step for DAC signal in 8bit resolution
float requiredTime = 0.0;                      // Required operating time (in seconds) for a
process
float SensorTemp = 0.0;
float SensorVolt = 0.0;
float SensorAmp = 0.0;
float SensorTempAmb = 0.0;
float SensorTempObj = 0.0;
float ppm_H2 = 0.0;

unsigned long timerVal = 0;                        // Timer value for tracking elapsed time (in
milliseconds)
unsigned long requiredTimeInMillis = 0;            // Required operating time (in milliseconds) for
process timing

xSemaphoreHandle xMutex = NULL;                    // Declare xMutex hanlde as NULL

// FreeRTOS task handles for managing concurrent tasks in the system
TaskHandle_t controlTaskHandle;                    // Task handle for the main control task
TaskHandle_t databaseTaskHandle;                   // Task handle for managing database operations
TaskHandle_t monitoringTaskHandle;                 // Task handle for actuator_A control, associated
with the relay module

// Forward declaration of task functions for FreeRTOS tasks
void controlTask(void *parameter);                 // Task function for system control operations
void databaseTask(void *parameter);                // Task function for handling database operations
void monitoringTask(void *pvParameter);            // Task function for actuator_A, managing relay
and temperature sensor

// Forward declaration of functions
void checkTimer();                                 // Checks and updates the timer status
```

```
void systemReset();                                 // Resets the system to its initial state
void setTimer();                                    // Sets the timer the control
void voltageRamp();                                 // Handles voltage ramp-up sequence
void databaseWrite();

//
OneWire sensor_A_Ref(sensor_A_Input);                                       // Create a OneWire
instance for communication with the 18B20 temperature sensor (sensor_A)
DallasTemperature sensor_A_Trigger(&sensor_A_Ref);                          // Link the OneWire
reference to the DallasTemperature instance for reading sensor_A data
DFRobot_INA219_IIC sensor_B_Trigger(&Wire, INA219_I2C_ADDRESS1);            // Instantiate an I2C
communication object for the INA219 sensor (sensor_B)
DFRobot_MLX90614_I2C sensor_C_Trigger;                                      // Instantiate an I2C
communication object for the MLX90614 temperature sensor (sensor_C)
InfluxDBClient client(INFLUXDB_URL, INFLUXDB_ORG, INFLUXDB_BUCKET, INFLUXDB_TOKEN,
InfluxDbCloud2CACert);    // Declare an InfluxDB client instance, configured with cloud certificate
for secure connection
Point sensor("sensor");                                                     // Create a data point
for writing sensor measurements to InfluxDB

void setup() {

  Serial.begin(9600);                               // Initialize serial and wait for port to
open:
  delay(1500);                                      // This delay gives the chance to wait for a
Serial Monitor without blocking if none is found
  initProperties();                                 // Defined in thingProperties.h
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);  // Connect to Arduino IoT Cloud
  WiFi.mode(WIFI_STA);                              // Setup wifi
  wifiMulti.addAP(WIFI_SSID, WIFI_PASSWORD);        // Access wifi
  // Connetion to wifi progress
  //Serial.print("Connecting to wifi");
  while (wifiMulti.run() != WL_CONNECTED) {
    //Serial.print(".");
    delay(100);
  }
  // Synchronize time with NTP servers and validate connection to InfluxDB
  Serial.println();
  timeSync(TZ_INFO, "pool.ntp.org", "time.nis.gov");
  // Check server connection
  if (client.validateConnection()) {
    Serial.print("Connected to InfluxDB: ");
    Serial.println(client.getServerUrl());
  } else {
    Serial.print("InfluxDB connection failed: ");
    Serial.println(client.getLastErrorMessage());
  }
```

```
  pinMode(sensor_A_Input, INPUT);
  pinMode(sensor_D_Input, INPUT);
  pinMode(actuator_A_Output, OUTPUT);
  pinMode(actuator_B_Output, OUTPUT);
  pinMode(LED_ONOFF, OUTPUT);                // Set GPIO as output pin
  pinMode(LED_Startup, OUTPUT);
  pinMode(LED_Production, OUTPUT);           // Set GPIO as output pin
  pinMode(LED_Alarme, OUTPUT);               // Set GPIO as output pin

  digitalWrite(actuator_A_Output, HIGH);     // Activate relay
  digitalWrite(actuator_B_Output, LOW);      // Activate relay

  dac_output_enable(DAC_CHANNEL_1);
  dac_output_voltage(DAC_CHANNEL_1, Step);
  delay(2000);

  alarmFlagA = true;
  alarmFlagB1 = true;
  alarmFlagB2 = true;
  alarmFlagC = true;
  alarmFlagD = true;

  // Initialize I2C communication
  sensor_B_Trigger.begin();                                          // Start I2C
communication for the INA219 sensor (sensor_B)
  sensor_B_Trigger.linearCalibrate(ina219Reading_mA, extMeterReading_mA);  // Perform internal
calibration for the INA219 sensor using provided reference values

  // Add tags to the data point for InfluxDB
  sensor.addTag("device", DEVICE);                                   // Add a tag for the
device identifier
  sensor.addTag("SSID", WiFi.SSID());                                // Add a tag for the
current WiFi SSID

  //  Set the debug message level for network and IoT Cloud connection status
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();

  xMutex = xSemaphoreCreateMutex();          // Create xMutex

  // Create FreeRTOS tasks
  xTaskCreatePinnedToCore(
    controlTask,                           // Function to implement the task
    "Check Timer Task",                    // Name of the task
    8000,                                  // Stack size in words
    NULL,                                  // Task input parameter
```

```
    1,                                      // Priority of the task
    &controlTaskHandle,                     // Task handle
    1                                       // Core 1 (primarily for application-level tasks,
including custom user-defined tasks)
  );
  xTaskCreatePinnedToCore(
    databaseTask,                           // Function to implement the task
    "Database Task",                        // Name of the task
    10000,                                  // Stack size in words
    NULL,                                   // Task input parameter
    1,                                      // Priority of the task
    &databaseTaskHandle,                    // Task handle
    0                                       // Core 0 (Primarily reserved for system-level tasks,
Wi-Fi, and Bluetooth stacks)
  );
  xTaskCreatePinnedToCore(
    monitoringTask,                         // Function to implement the task
    "Monitoring Task",                      // Name of the task
    3000,                                   // Stack size in words
    NULL,                                   // Task input parameter
    0,                                      // Priority of the task
    &monitoringTaskHandle,                  // Task handle
    0);                                     // Core 0 (Primarily reserved for system-level tasks,
Wi-Fi, and Bluetooth stacks)

}

void loop() {
  //Empty as we are using FreeRTOS
}

// Task responsible for controlling the process based on system conditions
void controlTask(void *parameter) {
  while (true) {
    if (xSemaphoreTake(xMutex, portMAX_DELAY) == pdTRUE) {
      ArduinoCloud.update();                                    // Update the Arduino Cloud
connection

      // Check system status and operation mode to control DAC output signal
      if (systemOnOff && operationMode == 1 && productionStatus) {
        voltageRamp();                                          // Perform voltage ramp-up
for operation mode 1
        checkTimer();                                           // Check the timer for
elapsed time
        dac_output_voltage(DAC_CHANNEL_1, StepAux);
      } else if (systemOnOff && operationMode == 2 && productionStatus) {
```

```
        voltageRamp();                                          // Perform voltage ramp-up
for operation mode 2
        dac_output_voltage(DAC_CHANNEL_1, StepAux);
      }
      xSemaphoreGive(xMutex);                                   // Release the mutex after
updates
    }
    vTaskDelay(pdMS_TO_TICKS(100));                             // Delay for 100
milliseconds
  }
}

// Task for sending data to a database
void databaseTask(void *parameter) {
  while (true) {
    if ( xSemaphoreTake( xMutex, portMAX_DELAY == pdTRUE) ) {
      if ( systemOnOff && productionStatus ) {
        sensor.clearFields();                                    // Clear fields for
reusing the point. Tags will remain the same as set above.
        // Store measured value into point
        sensor.addField("Sensor A", SensorTemp);
        sensor.addField("Sensor B1", SensorVolt);
        sensor.addField("Sensor B2", SensorAmp);
        sensor.addField("Sensor C1", SensorTempAmb);
        sensor.addField("Sensor C2", SensorTempObj);
        sensor.addField("Sensor D", ppm_H2);
        // Print what are we exactly writing
        //Serial.print("Writing: ");
        //Serial.println(sensor.toLineProtocol());
        // Check WiFi connection and reconnect if needed
        if (wifiMulti.run() != WL_CONNECTED) {
          Serial.println("Wifi connection lost");
          }
        // Write point
        if (!client.writePoint(sensor)) {
          Serial.print("InfluxDB write failed: ");
          Serial.println(client.getLastErrorMessage());
        }
      }
      xSemaphoreGive ( xMutex );
    }
    vTaskDelay(pdMS_TO_TICKS(1000));  // Delay to avoid hogging CPU
  }
}

void monitoringTask(void * pvParameter) {
   while (true) {
```

```
    if ( xSemaphoreTake( xMutex, portMAX_DELAY == pdTRUE) ) {
      // SENSOR A READ
       sensor_A_Trigger.requestTemperatures();                    // Call
sensors.requestTemperatures() to issue a global temperature request
       SensorTemp = sensor_A_Trigger.getTempCByIndex(0);          // Get the temperature in Celsius
at index 0(first and only 18B20 sensor on one wire connection)
      fuelCellTemperature = SensorTemp;

      // Condition validation to shutdown system
      if ( SensorTemp >= sensor_A_Cond ){
        //Serial.println("\nSHUT DOWN ISSUE: Fuel-Cell H12 (High Temperature)");
        alarmFlagA = false;
        systemReset();
      }
      // SENSOR B READ
       SensorVolt = sensor_B_Trigger.getBusVoltage_V();       // Get voltage reading in Volts
       SensorAmp = sensor_B_Trigger.getCurrent_mA();          // Get current reading in mAmp
        inputVoltage = SensorVolt;
        electrolyserCurrent = SensorAmp;

      // Condition validation to shutdown system
      if ( SensorVolt >= sensor_B_Cond1 ){
        //Serial.println("\nSHUT DOWN ISSUE: Electrolyzer-E207 (Max input voltage exceeded) ");
        alarmFlagB1 = false;
        systemReset();
      }
      else if ( SensorAmp >= sensor_B_Cond2 )
      {
        //Serial.println("\nSHUT DOWN ISSUE: Electrolyzer-E207 (Max current exceeded) ");
        alarmFlagB2 = false;
        systemReset();
      }

      // SENSOR C READ
      SensorTempAmb = sensor_C_Trigger.getAmbientTempCelsius();    // Get ambient temperature in
Celsius
      SensorTempObj = sensor_C_Trigger.getObjectTempCelsius();     // Get object temperature in
Celsius
      ambientTemperature = SensorTempAmb;
      electrolyserTemperature = SensorTempObj;

      // Condition validation to shutdown system
      if ( SensorTempObj >= sensor_C_Cond ){
        //Serial.println("\nSHUT DOWN ISSUE: Electrolyzer-E207 (High Temperature)");
        alarmFlagC = false;
        systemReset();
      }
```

```
      // SENSOR D READ
      // float ADC_MQ8 = analogRead(sensor_D_Input);              // 12bits on ESP32 > max=4095 //
10bits on Arduino UNO > max=1023
      // float Vs = (ADC_MQ8 * ADC_voltage)/Bit_Resolution;       // Obtain Vs on scale of 3V3
      // Vs = (Vs * Vcc)/ADC_voltage;                            // Converting Vs back to 5V scale
      // float Rs = ((Vcc * RL) / Vs) - RL;                      // Calculate Rs, Varying MQ8
resistance with H2 concentration
      // float ratio_RsR0 = Rs/R0;                               // Calculate Rs/R0 ratio
      // float ppm_H2 = a * pow(ratio_RsR0, b);                  // Calculate concentration in ppm
using power regression model y=a.x^-b
      ppm_H2 = ((random() % 1000) / 1.11);                       // generate random value since
sensor are not connected
      hydrogenConcentration = 44.44;

      // Only values in the range "100 <= ppm <= 10000" are admitted and printed on serial monitor
      if ( ppm_H2 <= sensor_D_Cond ) {
        //Serial.println("H2 concentration is out of detectable range ( < 100 ppm )");
      } else if ( ppm_H2 > sensor_D_Cond  ){
        //Serial.println("\nSHUT DOWN ISSUE: Hydrogen Gas Leakage detected");
        alarmFlagD = false;
        systemReset();
      }

      xSemaphoreGive ( xMutex );
    }

    vTaskDelay(pdMS_TO_TICKS(150));  // Delay task for 1s before the next iteration of the task
  }
}
/*
  Since SystemOnOff is READ_WRITE variable, onSystemOnOffChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onSystemOnOffChange()  {
  if( systemOnOff )  {
    digitalWrite(LED_ONOFF, HIGH);
  } else if ( !systemOnOff ){
    systemReset();
    digitalWrite(LED_ONOFF, LOW);
  }
}
/*
  Since OperationMode is READ_WRITE variable, onOperationModeChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onOperationModeChange()  {
```

```
}
/*
  Since SetPointHydrogen is READ_WRITE variable, onSetPointHydrogenChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onSetPointHydrogenChange()  {
}
/*
  Since SystemStart is READ_WRITE variable, onSystemStartChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onSystemStartChange()  {
  if ( systemOnOff ) {
  Step = 95;
  productionStatus = true;
  digitalWrite(LED_Startup, HIGH);
  digitalWrite(actuator_B_Output, HIGH);
  }
}
/*
  Since SystemStop is READ_WRITE variable, onSystemStopChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onSystemStopChange()  {
  alarmFlagA = true;
  alarmFlagB1 = true;
  alarmFlagB2 = true;
  alarmFlagC = true;
  alarmFlagD = true;
  systemReset();
}
/*
  Function to start the timer for time based control feedback
*/
void setTimer() {
  timerVal = millis();                                    // Capture the starting time
  timerStatus = true;                                     // Set the timer status to active

  requiredTime = setPointHydrogen / ProductionRate;       // Time in minutes
  requiredTimeInMillis = requiredTime * 60000;            // 60,000 ms in 1 minute
  aproxDuration = requiredTime * 60;                      // Time in seconds

}
/*
  Function to check the timer for time based control feedback
*/
void checkTimer() {
```

```
  if ( timerStatus ) {
    unsigned long elapsedTime = millis() - timerVal;          // Calculate the elapsed time in
milliseconds
    if (elapsedTime >= requiredTimeInMillis) {
      systemReset();                                          // Resets the system to its initial
state
    }
  }
}
/*
  Function to reset system variables and states
*/
void systemReset() {
  Step = 245;                                             // Reset duty cycle to zero
  dac_output_voltage(DAC_CHANNEL_1, Step);
  StepAux = 245;                                          // Reset auxiliary duty cycle to zero
  productionStatus = false;                                  // Indicate that the production
process is complete
  timerStatus = false;                                       // Deactivate the timer
  aproxDuration = 0.0;                                       // Reset approximate duration to zero
  requiredTimeInMillis = 0;                                  // Reset required time in
milliseconds to zero
  //digitalWrite(LED_ONOFF, LOW);                            // Yellow LED
  digitalWrite(LED_Startup, LOW);                            // Yellow LED
  digitalWrite(LED_Production, LOW);                         // Yellow LED
  digitalWrite(actuator_A_Output, HIGH);                                  // Set GPIO 23
to LOW (turn off the associated output)
  digitalWrite(actuator_B_Output, LOW);                                   // Set GPIO 5
to LOW (turn off the associated output)

}
/*
  Function to gradually increase the duty cycle to create a voltage ramp
effect
*/
void voltageRamp() {
 while ( Step < StepAux) {
    dac_output_voltage(DAC_CHANNEL_1, StepAux);
    StepAux = StepAux - 5;                                                // Increment
the auxiliary duty cycle
    vTaskDelay(pdMS_TO_TICKS(750));                                       // Wait for
750 milliseconds before the next increment
    if ( StepAux <= Step ) {

      if ( operationMode == 1 ) {
        setTimer();
      }
```

```
      digitalWrite(LED_Startup, LOW);
      digitalWrite(LED_Production, HIGH);
      digitalWrite(actuator_A_Output, LOW);                    // Set GPIO
18 to HIGH (turn on the associated output)
    }
  }
}
```

Instituto Politécnico de Tomar

www.ipt.pt