Hochschule Rhein-Waal

H2-Power substituting diesel generator / Wasserstoff ersetzt Dieselgenerator

---

Faculty of Communication and Environment • Interdisciplinary Project • Summer 2024

Lecturers: Omed Abed, Prof. Dr. Irmgard Buder

Supervisor:Dr. André Wenda

Prepared by :
Md Zilani Hossain
Shariar Alam Samir
Tanjir Mehrab

Project Team

Name	Student ID
Daniel Winterstein	30109
David Starcevic	30922
Khawaja Ahsan Elahi	31446
María Verónica Barrera Moreno	29252
Martin Beckers	30478
Md Ferdous Amin	32444
Md Zilani Hossain	32535
Mika Günther	30106
Mohammad-Aminul-Islam Tanvir	31927
Rithik Kumar	31522
Fungai Machingaifa	31065
Shariar Alam Samir	31533
Simon-Eduardo Munoz-Buchwald	30096
Tadiwanashe Dondo	31250
Tanjir Mehrab	31530

Abstract

Author: Tanjir Mehrab

Our project aims to pioneer the development of a prototype hydrogen power generator, a vital step toward advancing sustainable energy solutions. Led by a dedicated team of engineers and researchers, we are meticulously selecting and rigorously verifying each component to ensure their compatibility and efficacy within the system. This careful phase of component selection and validation is crucial for the seamless integration and optimal functionality of the hydrogen power generator prototype.

Hydrogen power, produced through electrolysis using renewable energy, presents a zero-emission alternative to fossil fuels. Our initiative seeks to demonstrate the feasibility and efficiency of hydrogen as a primary energy source, integrating it into existing infrastructures with minimal modifications. We are committed to optimizing safety and efficiency while assessing the environmental impact, particularly reducing greenhouse gas emissions.

Through systematic scrutiny and rigorous testing, we aim to establish a reliable and scalable hydrogen power generator capable of efficient operation under various conditions. Our ultimate goal is to contribute to advancing sustainable energy solutions, support the transition to cleaner power sources, and lay a robust foundation for future experimentation and refinement.

Table of Contents

Introduction

Author: Khawaja Ahsan Elahi

The goal of the project was to develop a replacement power generator that uses hydrogen as an energy source. The focus was on implementing an autonomous island solution that works independently of external power sources. In addition to planning, design, and construction, the safety assessment of the system is also an essential part of the project to ensure safe handling and use of the hydrogen. Adapting the unit for outdoor use presents an additional challenge that requires special attention.

As part of the project, the students work together with OMEXOM GmbH. The students are responsible for checking and selecting suitable system components, such as batteries and inverters. These components not only have to meet the system requirements but also have to be safe to operate. A particular focus is designing the system to work reliably under various outdoor environmental conditions. The finished system will be presented at OMEXOM GmbH in early July to stakeholders from business, science, and politics.

Group members

A total of 15 students from various degree programs worked collectively on this project. Each student contributed their unique expertise and prior knowledge. They collaborated effectively, sharing and expanding their skills. Throughout the project, they learned new concepts and techniques. This combination of diverse knowledge and continuous learning led to the project's success.

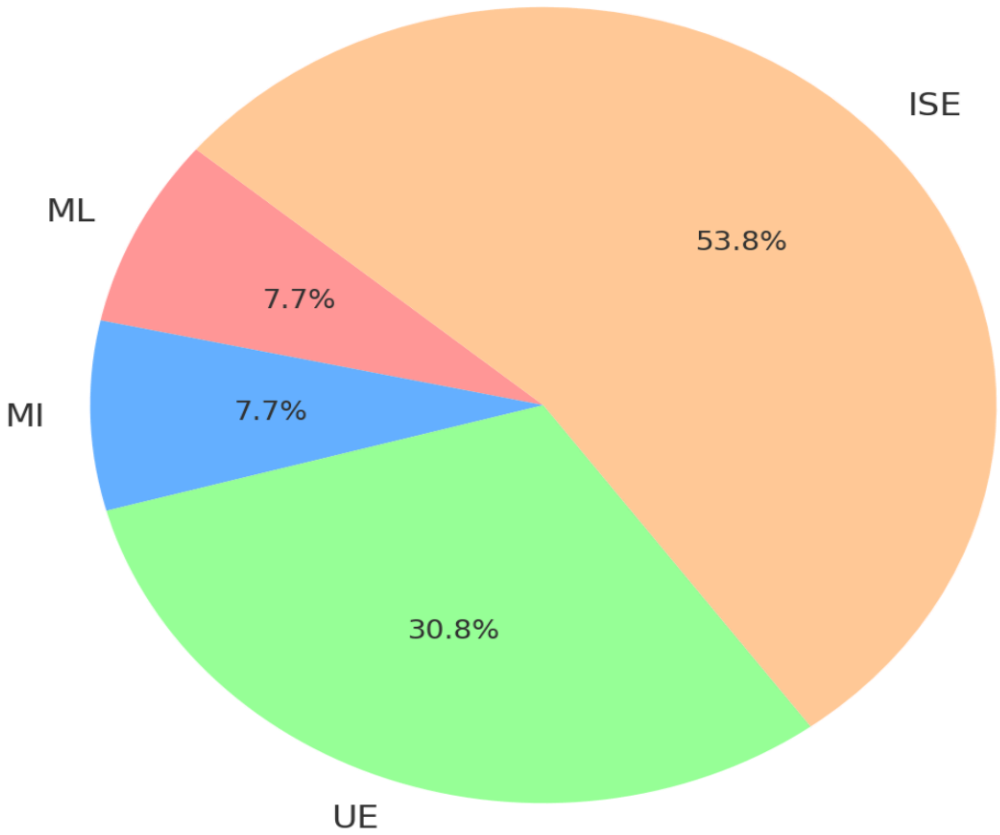

Figure 1.1: Data structure

Source: Own illustration

Project Teams

The project was divided into these teams:

- Components List-Making team: Responsible for making sure all components required are ordered.
- 3D modeling team: Responsible for making an accurate design for our project, which will be used later by the assembly team to create the generator.
- Research team: Conducted Research throughout the project, making sure all the problems were resolved related to the project by providing appropriate solutions.
- IT team for Programming: Developed and implemented software solutions.
- Assembly Team: Constructed the generator from scratch.
- Documentation team: Making the Final Report.
- Proofreading Team: Making sure the Report is structured properly without any mistake

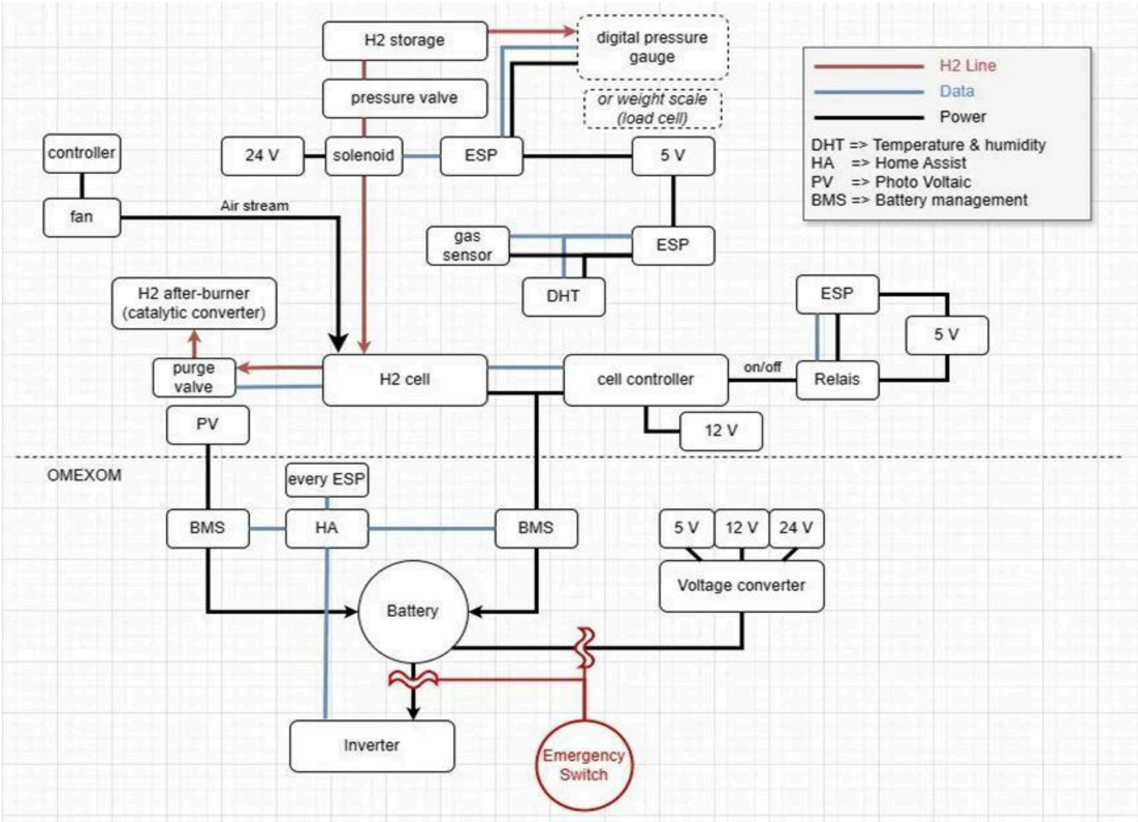

Figure 1.2: Schematic Diagram

Source: Own illustration

Components

- **H2 Storage**: Stores the hydrogen gas.
- **Digital Pressure Gauge**: Monitors the hydrogen pressure.
- **Pressure Valve**: Regulates the flow of hydrogen.
- **Solenoid**: Controls the hydrogen flow.
- **ESP**: Microcontroller units for various control tasks.
- **Gas Sensor**: Detects hydrogen gas leaks.
- **DHT Sensor**: Measures temperature and humidity.
- **Hydrogen Fuel Cell (H2 Cell)**: Converts hydrogen into electrical energy.
- **BMS (Battery Management System)**: Manages the battery's health and performance.
- **HA (Home Assistant)**: System for home automation control.

- **PV (Photovoltaic)**: Solar panels for supplementary power generation.
- **Fan**: Used for cooling the system.
- **Inverter**: Converts DC power from the battery to AC power for home use.
- **Relay**: Switches to control the electrical circuits.
- **Voltage Converter**: Adjusts voltage levels as needed.
- **Emergency Switch**: Shuts down the system in emergencies.

System Flow

- **Hydrogen Storage and Flow**:
  - Hydrogen gas is stored in the H2 storage tank.
  - The digital pressure gauge continuously monitors the pressure of the stored hydrogen.
  - Hydrogen flows through the pressure valve, controlled by a solenoid.
  - A gas sensor detects any leaks of hydrogen gas for safety.

- **Energy Generation**:
  - Hydrogen is fed into the hydrogen fuel cell (H2 cell), where it is converted into electrical energy through a chemical reaction.
  - An H2 after-burner (catalytic converter) handles any residual hydrogen to ensure safe operation.

- **Control and Monitoring**:
  - ESP microcontrollers manage and monitor various components of the system.
  - A cell controller oversees the operations of the hydrogen fuel cell.
  - The DHT sensor provides essential data on temperature and humidity to maintain optimal operating conditions.

  - Relays control the distribution of power within the system.

- **Power Management**:
  - The electrical energy generated by the hydrogen fuel cell charges the battery.
  - Additional energy from photovoltaic (PV) solar panels also charges the battery, supplementing the hydrogen fuel cell.

- The Battery Management System (BMS) ensures the battery operates safely and efficiently.

- **Power Distribution**:
  - A voltage converter adjusts the power to the required levels (5V, 12V).
  - The inverter converts the DC power stored in the battery to AC power for household use.

- **Safety and Automation**:
  - An emergency switch can shut down the system in case of any malfunction.
  - The Home Assistant (HA) integrates with the system for automation and control.
  - Fans are employed to cool various components, maintaining optimal temperatures for efficient operation.

Summary

This diagram outlines a hydrogen fuel cell generator system designed to generate, store, and manage energy. The system ensures efficient energy conversion from hydrogen, integrates safety features, includes monitoring sensors, and employs automation for effective control and utilization, complemented by solar power and managed by various control units and sensors.

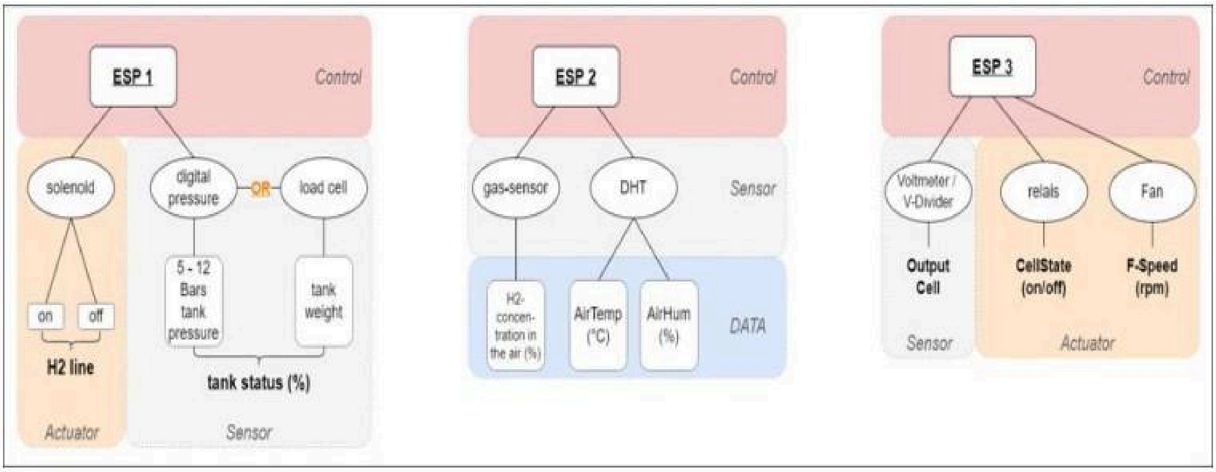

Figure 1.3: Data structure

Source: Own illustration

- **ESP 1**: Controls hydrogen flow and monitors tank status.
- **ESP 2**: Monitors air quality and environmental conditions.
- **ESP 3**: Manages the operational state of the fuel cell and cooling system

Project Requirements by the company

The generator must be watertight, at least against rain, and the cylinder should be separable to allow for the generator and tank to be transported separately. There is a trailer that we may inspect later, and the battery should be replaceable. Omexom aims to integrate the gas sensor on the hardware side, possibly with a redundant sensor. Questions were raised about the load cell in a mobile context and the battery voltage and charging curve. A feed-in socket is to be installed for the UPS function. The use of an MPPT tracker between the fuel cell and voltage regulator was suggested as an alternative to DC-DC converters. Discussion included how the battery can be disconnected in an emergency using load-break switches or circuit breakers.

Sockets are located in the control cabinet, and the interface must be agreed with Omexom. Redundancy in the solenoid and direct contact of the manometer with hydrogen were also discussed. Research into cost-effective measurement options is required, and Omexom uses a blower on-site to discharge hydrogen. The practicality of this in a mobile application, especially regarding safety, was questioned, and researching a catalyzer was proposed.

Additionally, checking for Home Assistant-compatible sockets to monitor/switch them, avoiding WLAN by using wired ESPs, and the mention of the Carlo Gavazzi EM24 meter for mains feed were highlighted.

Brief Insight into Omexom

Omexom, a brand under VINCI Energies, specializes in providing comprehensive solutions in the energy infrastructure sector. Established to support the energy transition, Omexom focuses on generating, transmitting, and distributing electricity. With a global presence, the company operates in over 40 countries and employs around 23,000 professionals.

Omexom is actively working towards advancing hydrogen fuel cell technology, particularly in the realm of power generation. Hydrogen fuel cell generators offer a clean and sustainable alternative to traditional diesel generators, which are commonly used on construction sites and other remote locations.

Overview of the System and Responsibilities

Author: Shariar Alam Samir

The project aimed to develop an autonomous, hydrogen-powered generator suitable for outdoor use. The planning, design, construction, and safety assessment were conducted by students, with support from professors, supervisors, and Omexom workers. The project involved 15 students from various disciplines, including Infotronic Systems Engineering, E-Government, Environment and Energy, Media Communication and Computer Sciences, and Mobility and Logistics.

Communication and Collaboration Platforms

- Discord: Primary communication platform managed by David Starcevic.
- Microsoft Teams: Used for communication with Omexom.
- Google Drive: Created by Daniel-Sebastian Winterstein for file sharing and progress monitoring.

Project Management

- Project Manager: Maria Veronica Barrera Moreno
    - Responsibilities: Managing deadlines, organizing the group, tracking tasks, and assigning roles.

Budget Management

- Budget: No formal budget; materials provided by the company and university, with Omexom purchasing items from a student-prepared list.
- Component List: Led by MD-Ferdous Amin, with purchases managed by Omed Abed.

Task Distribution and Responsibilities

- Weekly Task Tracking: Managed by Mika Nils Günther, who also guided the IT team on Home Assistant and MQTT integration.
- 3D Modeling Team: Led by Fungai Machingaifa, assisted by Tadiwanashe Dondo, Mohammad Aminul Islam, Khawaja Ahsan Elahi, and other team members for assembly.
- Afterburner Development: Initial research was replaced by a functional model created by Omed Abed.
- Hydrogen Pressure Transmitter: Researched and sourced within budget constraints.

Assembly and Testing

- Assembly Team: Tadiwanashe Dondo, Fungai Machingaifa, Mika Nils Günther all other members
    - Responsibilities: Building the generator skeleton, placing main components, and assisting in electronic connections.
- IT Team: Managed the use of a Raspberry Pi and 3 ESPs, integrating Home Assistant and MQTT for system control.

Documentation and Reporting

- Documentation Team: Md-Zilani Hossain, Sharaiar Alam Samir, Tanjir Mehrab.
    - Responsibilities: Preparing the final report with individually assigned chapters.

Hardware and Software Integration

- Hardware: Controlled by a Raspberry Pi coupled with ESPs, managing various functions. Key components included an H2 tank, fuel cell, battery for electricity storage, and inverter for energy utilization.
- Software: Home Assistant as the project brain, alongside MQTT for IoT data networking. Various ESPs were attached for specific purposes.

Project Presentation and Recognition

- Presentation: The completed system was presented at Omexom GmbH on 1st July to stakeholders from business, science, and politics.
- Recognition: The project won the Hochschulpreis 2024 der Wirtschaftsförderung Kreis Kleve.

Results and Future Outlook

- Results: The generator functioned successfully, with all necessary tasks fully implemented.
- Poster Preparation: Handled by Veronica, Tadiwa, Fungai, Mika and Veronica.
- Future Steps: Integrating 4G for complete remote functionality.Project Evaluation and Improvement Areas

- Challenges: Delays in 3D modeling, inefficiencies in task division and communication, late component orders, and insufficient testing.
- Successes: Effective task completion by IT and assembly groups, well-structured documentation, and timely poster preparation

Component Selection

Component List Overview

The components used in our prototype are:

- Fuel Cell
- Hydrate Storage
- MultiPlus
- NF A14 V1
- Router

- UP5000
- Aluminum Profile

Detailed Component Analysis

PTFE Schläuche (30m) (PTFE tubing)

- Details about compatibility: PTFE tubing is suitable for a hydrogen (H2) power generator due to its chemical resistance and high-temperature capabilities. It must meet the pressure, temperature, and chemical compatibility requirements of the system.
- Short Description: PTFE tubing is known for its durability, high-temperature tolerance, and low friction coefficient. It is ideal for conveying various substances, including corrosive chemicals and solvents.

BZ 500W (Power supply with fan)

- Short Description: The "BZ 500W" hydrogen power generator uses electrolysis to split water into hydrogen and oxygen gases, producing 500 watts of electricity. The PTFE tubing must withstand the system's operating pressures and temperatures and be chemically resistant to hydrogen and oxygen gases.

MH Speicher 900L Hydrogen Storage Tank

- Short Description: The MH Speicher 900L is a hydrogen storage tank designed for safe and efficient hydrogen gas containment. Made from high-strength steel or composite materials, it can withstand high pressures. Key safety features include pressure relief valves and rupture discs.

Druckminderer (Pressure Reducer)

- Short Description: The Druckminderer regulates the high pressure of hydrogen gas to levels suitable for fuel cells or other applications. It must be made of materials compatible with hydrogen to prevent corrosion and ensure safety.

NF-A14 v1

- Detailed Specifications: The Noctua NF-A14 PWM is a 140mm cooling fan known for its performance, quiet operation, and longevity. Key features include advanced acoustic optimization, SSO2 bearing, and PWM support for automatic speed control.

Aluminum Profiles (Aluprofile)

- Short Description: Aluminum profiles are used in hydrogen power systems for their strength, lightweight nature, and corrosion resistance. They provide structural support, facilitate heat management, and enhance durability and safety.

Aluminum Profile Connectors (Verbindungsstücke Aluprofile)

- Short Description: Aluminum profile connectors enable the joining of aluminum profiles at various angles and configurations crucial for building the intricate frameworks necessary to mount and secure hydrogen power system components.

Victron MultiPlus-II 48/3000/35-32 GX

- Short Description: The Victron MultiPlus-II 48/3000/35-32 GX is an inverter/charger ideal for hydrogen power systems, capable of converting DC power into AC power and charging battery banks. It includes advanced functionality such as PowerAssist and two AC outputs.

UP5000

- Short Description: The Pylontech U5000 is a Lithium Iron Phosphate (LiFePO4) battery designed for energy storage systems, known for its safety, long cycle life, and performance in solar and renewable energy applications.

Organization of the Tasks

Author: María Verónica Barrera Moreno

Team and their tasks

During the initial stages of the project, specialized teams were formed to handle different aspects of the development process. These teams included:

- **Components List-Making team**: Responsible for compiling and analyzing potential components.
- **3D modeling team**: Responsible for making an accurate design for our project, which will be used later by the assembly team.
- **Research team**: Conducted extensive research on hydrogen technology and related systems.
- **IT team for Programming**: Developed and implemented software solutions.
- **Assembly team:** Constructed the generator.
- **Documentation team:** Was in charge of doing the final report.

Detailed Task Organization

This chapter will explain the tasks in which the project was divided and how they complement each other. The project was supervised by one professor and two assistants. During the first class, an introduction to the project was made and everyone participated in the first sketch. In the second class, a project manager was elected unanimously, the Mobility and Logistics student, Maria Veronica Barrera Moreno. Her tasks were to keep up with deadlines, organize the group as a whole, keep track of tasks to be completed, and assign tasks to people without any.

The tasks to complete were discussed and each student willingly chose which tasks to work on. Mika Nils Günther kept track of which tasks needed to be completed for this project. Also, Mika explained to the IT team how Home Assistant and MQTT work so they could continue on that part on their own.
Mika and Omed worked as supervisors for most of the project where the rest of the team could clarify doubts or check on the correctness of their work by them.
The 3D model team was led by Fungai Machingaifa who explained to the other members how to use Fusion and did the main 3D model. This group needed to create a realistic model to see where to put all the components and get the right measurements to build it later. Later this was corrected by Mika. This model was used to assemble the skeleton of our box, all measures were taken from there. The list of components was led by MD-Ferdous Amin, who submitted the shopping list to Omed Abed, the person in charge of making the final purchase. Research on the afterburner was started in the second class, which wasn't used because Omed created one for the project. The component list was used by the modeling team to accurately download the components for their 3D model.

Due to the use of hydrogen, an adequate pressure gauge was needed. For this purpose, research was done, and the emails were written to companies that gave us a price quote. Finally, one inside our price range was found and bought.
The research team helped the IT team with information from the manual.
After the meeting with Omexom, we had a clear idea of what they expected from us. The company desired more safety features, and we gained knowledge of what part they would assist and which part they would take over. A Microsoft team was created to communicate with the workers from Omexom for any inquiries and exchange of information.

Daniel-Sebastian Winterstein created a Google drive to upload every file so it can be seen by the company and they can have a clear idea of our progress.

At half of our project, a documentation group was created led by Md Zilani Hossain and his teammates Tanjir Mehrab and Shahriar Alam Samir to help with the final report. They interviewed members of each group and wrote down what was done.
A new research group was also created to help the IT team with the ESP connection. This did not lead to the desired outcome, another way was used.
On the 29 of May, the assembly was started by Tadiwanashe Dondo, Fungai Machingaifa, and Maria Veronica Barrera Moreno.
They built the skeleton of the generator based on Fungai's 3D model and with the assistance of Mika, who also made the electronic connection. The main components were placed, like the battery and inverter, after the first two weeks of starting with assembly.

The IT team needed to know the physical components and electronic connections to be aware of what elements they needed to use and what functions needed to be done. The wiring was done by Mika and Omexom and part of the electronic connections was done by the IT team. The IT group used one Raspberry Pi coupled with 3 ESPs, which had control over different functions. On the software side, Homeassist was used and was made the brain of the project together with MQTT, an IoT data network. To this central point of information were different ESPs attached with their own purposes.
The tasks for the modeling of the small components were divided between each team member, except the IT group. Mika, MD-Ferdous, and Mohammad-Aminul-Islam Tanvir did the 3D printing based on their own 3D models. Said people attach these parts to the skeleton. On the last day of delivery, a lot was completed. Finally, the solenoid wasn't used. And the project was successfully delivered to Omexom where it was tested to charge equipment.

After the completion of the delivery, the division of work for the report was done. David organized and helped with the division of tasks. After checking the documentation done by Md-Zilani Hossain and his team, it was decided to change the structure and give everyone at least one chapter to write. The poster also needed to be done, and Verónica, Tadiwa, Fungai, and Mika took responsibility for that.

The machine was completed and is fully functional. The deadlines were not met but the project was completed successfully.

Main Part: Project Tasks

3D-Prints

Author: Khawaja Ahsan Elahi

Mika, Ahsan, Rithik, Fungae, and Tadiwa completed the 3D modeling. Then, the 3D printing team, consisting of Mika, Firdous, and Tanvir, began printing the 3D models one by one. These following method were used to print the 3D models which helped to build the hydrogen fuel cell generator

Exporting Your Model from Fusion

- Open Your Model: Launch Fusion360 and open the 3D Model you want to print.
- Check Model Integrity: Ensure there are no errors in your Model. Use the "Inspect" tool to check for any issues like holes or non-manifold edges.
- Set the Scale: Adjust the scale of your Model if necessary. Ensure it fits within the build volume of your 3D printer.
- Export the Model: Go to File > Export > 3D Model. Choose the file format compatible with your slicer software (typically STL, OBJ, or 3MF).

- STL: Standard format for most 3D printing software.
- OBJ: Includes color and texture information.

Preparing the Model for Printing

- Choose a Slicer Software: The common options include Cura, Prusa Slicer, and Simplify3D. These programs convert your 3D model into a set of instructions for the 3D printer.
- Import the Model: Open your slicer software and import the exported file.
- Check Orientation: Adjust the orientation of the model to minimize the need for support and improve print quality.
- Add Supports: Use the slicer's automatic support generation or manually add supports where necessary.
- Set Print Settings: Adjust print settings such as layer height, infill density, print speed, and temperature.

- Layer Height: Lower values give higher resolution but increase print time.
- Infill Density: Determines the internal structure; higher density means stronger but slower prints.
- Print Speed: Balancing speed and quality is essential; too fast can reduce quality.

Setting Up the 3D Printer

- Prepare the Printer: Ensure the printer is on a stable surface and has been calibrated correctly.
- Load Filament: Insert the filament according to your printer's instructions. Ensure it is the correct type for your model settings.
- Level the Bed: Level the printer bed to ensure proper adhesion and print quality.
- Preheat the Printer: Set the printer to the correct temperature for the filament you are using.

Printing the Model

- Start the Print: Insert the SD card into the printer or send the file via USB. Select the G-code file and start the print.
- Monitor the Print: Keep an eye on the first few layers to ensure they adhere properly. Make adjustments if necessary.
- Troubleshooting: If issues arise (e.g., poor adhesion, stringing, or under-extrusion), pause the print and adjust settings accordingly.

Post-Processing

- Remove the Model: Once printing is complete, carefully remove the model from the print bed.
- Remove Supports: Use tools to carefully remove any supports. Be gentle to avoid damaging the model.
- Sand and Finish: Sand the model if needed to smooth out surfaces.

Main Model Analysis with 3D Model Breakdown

Author: Fungai Machingaifa

The 3d main model was designed with Fusion360 commercial computer-aided design software (CAD) and computer-aided engineering software application(CAE). The use of Fusion360 came in handy for the visual design and accurate placement of the components. We managed to insert the major components and scale the Generator.

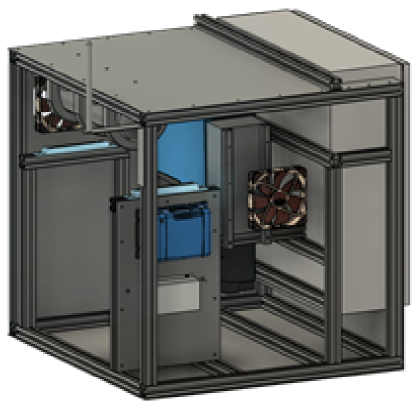

Figure 2.1: Hydrogen Generator 3D model

Source: Own illustration

When designing the Model, the inspiration and goal was to create a mobile, compact hydrogen generator without sacrificing accessibility. We achieved this with the lithium battery stand, which is securely fastened with bolts for stability and is easy to slide out if unfastened. The inverter and fuel cell were strategically placed in the middle compartment with fans on the flank for good airflow for hydrogen leaks and controlled temperature. The control box was on the opposite for good weight distribution and balance. This also facilitates power management and the accessibility of the emergency button.

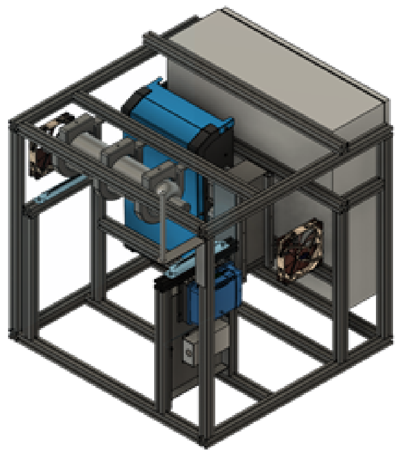

Figure 2.2: Hydrogen Generator 3D model open panels

Source: Own illustration

Given that the goal was to be compact, we had to utilize the space available for smaller components. To maximize this, we mounted this component onto interior side panels. It was perfect for the DC-DC converter and fuel cell controller, as illustrated in Figure 2.3. Well and creative space management ensures all components are securely housed while keeping the generator at the required size.

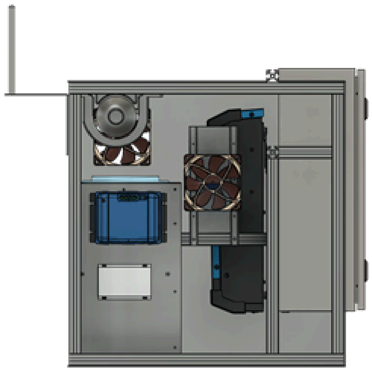

Figure 2.3: Hydrogen Generator 3D model right side view

Source: Own illustration

Figure 2.4: Hydrogen Generator 3D model control box view

Source: Own illustration

To summarize, our 3D design emphasizes compactness, mobility, and accessibility, ensuring that each component is optimally placed for space utilization, effective thermal distribution and functionality, and ease of maintenance. Fusion360 enabled us to create a detailed and precise model, which will guide the assembly and construction of the hydrogen generator.

Research

Author: Mohammad Aminul Islam Tanvir

Introduction

This document explores the redundancy system using Arduino microcontrollers and Raspberry Pi for enhanced reliability and fault tolerance. The system integrates cloud-based monitoring to ensure continuous operation, robust data collection, and real-time accessibility.

Redundant Systems

Redundant systems maintain continuous operation by automatically switching to backup systems upon the failure of the primary system. These systems can either be triggered by faults or run continuously in the background, minimizing latency impact and ensuring faster recovery.

Key Concepts:

Fault Detection: Identifying failures in system components.

Failover Mechanism: Switching operations to a backup system upon fault detection.

Latency Management: Ensuring minimal delay during the switchover from primary to backup systems.

Redundancy System Using Arduino and Raspberry Pi

The redundancy system leverages the strengths of both Arduino microcontrollers and Raspberry Pi. The primary device (Arduino or Raspberry Pi) operates continuously, sending periodic heartbeat messages to the secondary device. If the secondary device does not receive a heartbeat within a specified interval, it assumes the primary has failed and takes over its responsibilities.

Key Components:

Primary Device (Arduino/Raspberry Pi): Sends periodic heartbeat messages.*Secondary Device (Arduino/Raspberry Pi):* Monitors heartbeat messages and takes over if they are not received within the expected timeframe.

Heartbeat Mechanism

The heartbeat mechanism is essential for ensuring fault tolerance. The primary device sends periodic signals (heartbeat messages) to the secondary device. If a heartbeat is not received within a certain period, the secondary device assumes the failure of the primary system and takes over the responsibilities.

Workflow:

1. Initialization: Both devices are configured identically and powered separately.
2. Periodic Heartbeat: The primary device sends heartbeat messages regularly.
3. Heartbeat Monitoring: The secondary device monitors for these messages.
4. Failure Detection: The secondary device takes control if it detects that a heartbeat message has been missed.
5. Failover Activation: The secondary device takes on the functions of the primary device.
6. System Recovery: The primary device resumes operation upon recovery, and the secondary device goes back to standby mode.

Cloud-Based Monitoring System

A cloud-based monitoring system enhances the reliability and scalability of the redundancy system by enabling real-time data collection, storage, analysis, and display from field devices.

System Design:

Field Devices: Equipped with sensors, they connect to the cloud via the Internet to collect data and send it to the cloud platform.

Cloud Platform: Acts as the core for data capturing, storage, and processing. Services such as AWS IoT, Google Cloud IoT, or Microsoft Azure IoT can be used.

User Interface: Web and mobile applications provide real-time data presentation and notifications.

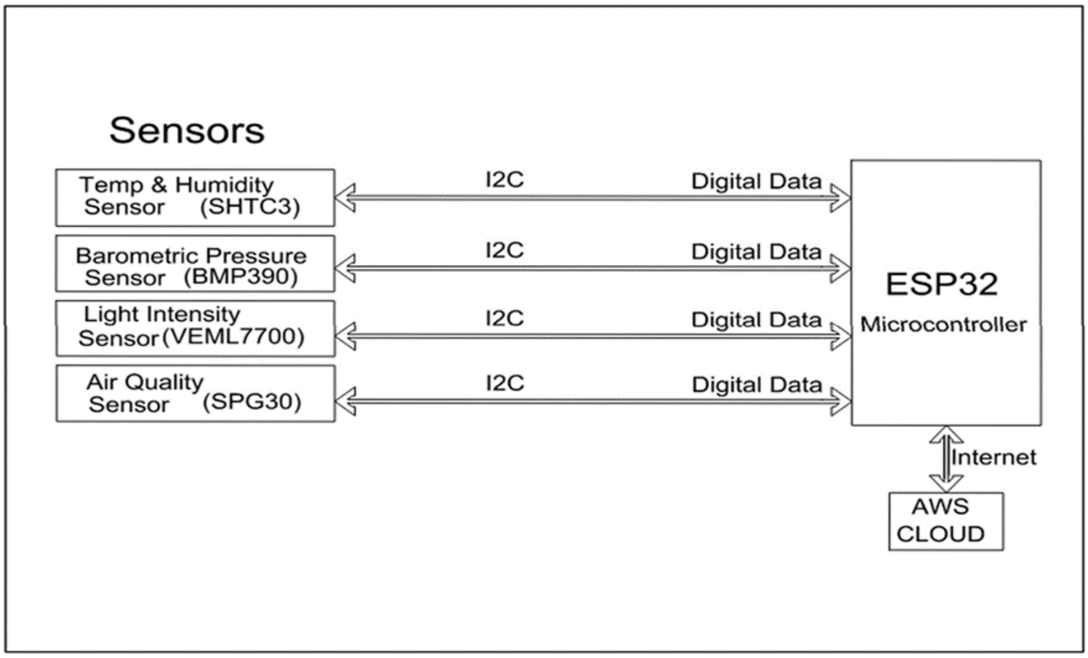

Figure 3.1: System design of field devices

Source: Own illustration

Workflow:

1. Data Collection: Field sensors collect data through various installed sensors.
2. Data Transmission: Data is transmitted to the cloud platform.
3. Data Processing: Cloud services process and analyse the incoming data.
4. Data Storage: Processed data is securely stored in the cloud.
5. Data Visualization: Users access data via web or mobile interfaces.

Technical Background

System Design of Field Devices: Each device (Arduino/Raspberry Pi) is equipped with various sensors (e.g., temperature, humidity, light) and configured to send data to a central system.Master Device System Design: The master device (Raspberry Pi) retrieves data from field devices, processes it, and publishes it to an IoT platform.

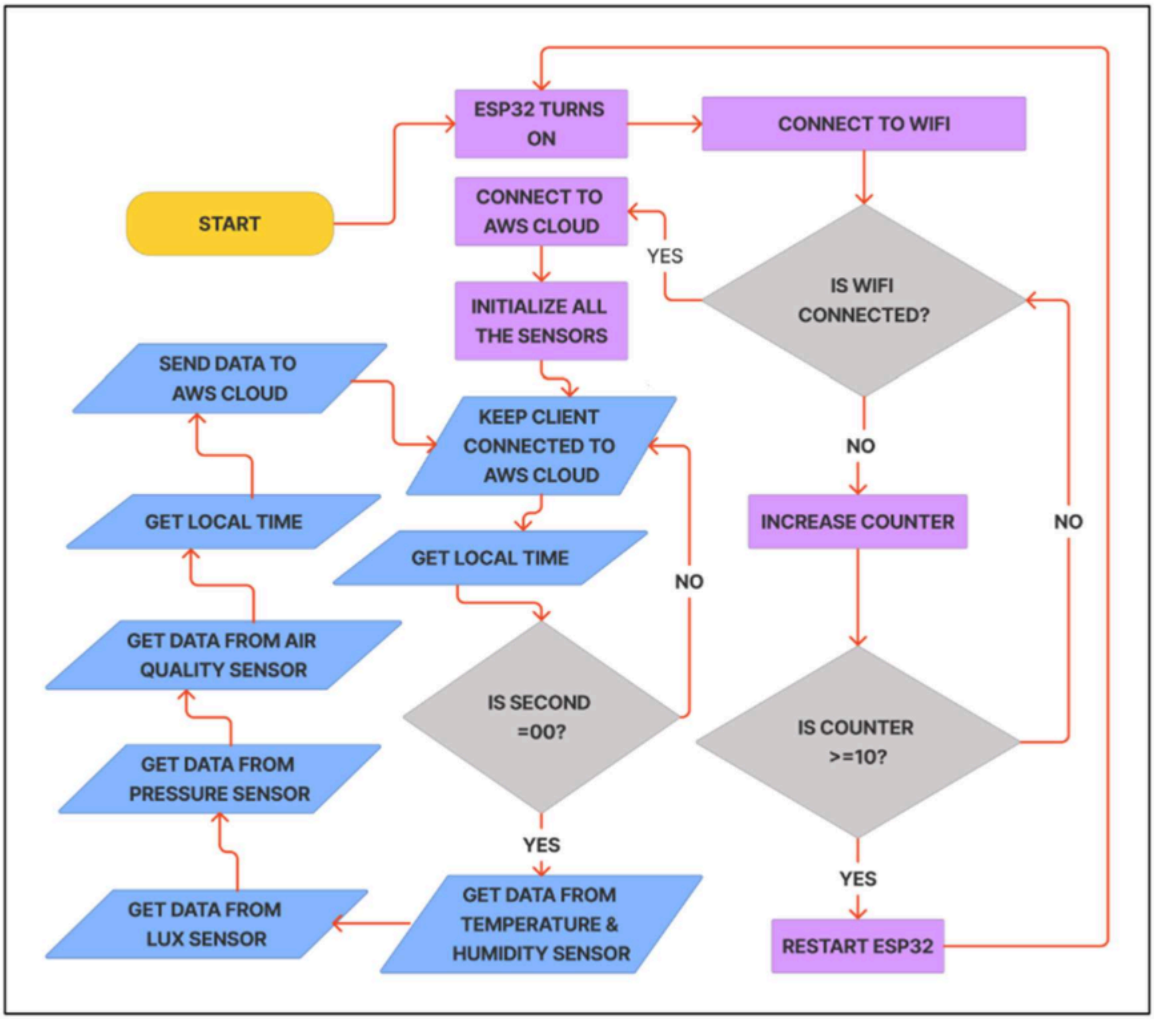

Figure 3.2: Flowchart for ESP32 based field devices

Source: Own illustration

Hardware Selection:

Arduino Uno: Ideal for prototyping and small-scale projects, equipped with sufficient I/O pins for sensor connections.

Raspberry Pi 4 Model B: Equipped with a powerful 64-bit quad-core processor,1.5GHz, 4GB RAM, dual-band wireless LAN, Bluetooth 5.0, gigabit ethernet, and USB 3.0 for external peripherals.

System Implementation

Steps:

1. Setup: Configure both Arduino and Raspberry Pi with identical hardware and software.

2. Heartbeat Transmission: The primary device periodically sends heartbeat messages.

3. Monitoring: The secondary device continuously monitors for heartbeat messages.

4. Failover Procedure: If a heartbeat is missed, the secondary device activates failover mechanisms.

5. Recovery: The primary device reclaims control after recovery.

6. Integration with Cloud: Configure field devices to send data to the cloud.

7. Data Processing: Cloud services process and analyze the data.

8. Visualization: Develop web or mobile applications for real-time data access.

Results

The redundancy system using Arduino and Raspberry Pi ensures reliable data collection, storage, and analysis. Field devices send data periodically to the cloud, which processes and visualizes it effectively. Issues such as sensor problems and code modifications were addressed, ensuring smooth operation with reliable failover mechanisms.

Benefits

Continuous Operation: Ensures uninterrupted operation through failover mechanisms.
Fault Tolerance: Real-time fault detection and mitigation.

System Reliability: Enhanced reliability with backup control.

Remote Monitoring and Maintenance: Allows for remote troubleshooting.
Scalability and Flexibility: Easily scalable to meet evolving needs.

Real-Time Data Access: Cloud integration provides real-time access to data from anywhere.

Conclusion

Implementing a redundancy system using Arduino and Raspberry Pi with cloud-based monitoring significantly enhances system reliability, fault tolerance, and continuous operation. These systems are particularly beneficial in critical and remote environments, ensuring safety and performance.

Procurement

Author: Md Ferdous Amin

This procurement plan for the “H2-Power Substituting Diesel Generator” was carried out in a way that would allow buying all the needed components at the lowest possible cost while guaranteeing high quality at the same time. Here is the breakdown of how procurement was done, with brief descriptions of the items procured.

Procurement Strategy

Our procurement strategy was centered around the following key principles:

1. Vendor Selection:
- It meant that multiple suppliers were identified for each component to keep the suppliers’ price competition going.
- Identified the cheapest supplier for each of the items to be purchased without negating the quality of the products.

2. Cost Management:
- Always kept the project costs at their lowest by looking for cheap suppliers while ensuring that they were quality suppliers.
- Made sure that all parts/members and subassemblies complied with the established design and quality parameters.

3. Logistics:
- Ensured all the components arrived on time within the set timeline of the project.
- Minimizing the wastage of all products received through proper management of Inventory.

Summary of Components Ordered

1. PTFE Schläuche (PTFE Tubing)
- Quantity:
- Description: One that is long-lasting and can withstand high temperatures that can be used for transportation of different materials such as corrosive solutions and solvents.

2. BZ 500W (Power Supply)
- Quantity: 1
- Description: Electrolytic cell, power supply which creates hydrogen and oxygen gasses through the electrolysis of water and provides 500 watts of electrical power.

3. MH Speicher 900L also called Hydrogen Storage Tank
- Quantity: 1
- Description: A pressure vessel for hydrogen storage intended for safe storage and use of hydrogen gas with optimal pressure that the vessel could comfortably contain.

4. Druckminderer (Pressure Reducer)
- Quantity: 1
- Description: An apparatus, which controls high-pressure hydrogen gas for use in fuel cells or any other process in order to make it safe for and avoid corrosion.

Aside from these we also bought Rasberry Pi, ESP, and Aluminium profile. Aluminium profile connector, Relay's. Here is the link to the list that we used to order this item,

List of the components to order.

Implementation

The procurement team was responsible for the following tasks: The procurement team was responsible for the following tasks:

- Carrying out research to establish the right suppliers to approach when searching for more suppliers.
- It involved a comparison of prices and identifying the cheaper options.
- Another strategy relates to products as this deals with the timely acquisition of the necessary components from suppliers.
- Checking on the quality and specification of the received items to confirm they were in good order and met the project standards.

Conclusion

The buying techniques involved in acquiring resources for the "H2-Power Substituting Diesel Generator" project were accomplished with regard to cost, quality, and time factors. Therefore, relying on a careful approach to the choice of the vendors and to the work on the cost estimate and the logistic planning we managed to source all the necessary components required for the development and assembling of the hydrogen power generator prototype.

Software Developing & Logic Implementation

Home Assistant & ESP Home

Author: Martin Beckers

Software Overview

Home Assistant

Home Assistant is an open-source platform for home automation designed to simplify the management and control of smart home devices. The software enables the integration and automation of a wide variety of devices and services that communicate via different protocols and standards. This allows users to centrally control and monitor all their connected devices – from lights and heating systems to security systems and voice assistants.

Home Assistant is known for its flexibility and extensibility. It supports a large number of integrations and can be easily tailored to individual user needs and technical environments. Configuration is typically done through **YAML** files, providing high granularity and control over automations. Additionally, *Home Assistant*offers a user-friendly web interface and mobile apps for easy access and control from anywhere.

Another advantage of *Home Assistant* is its focus on privacy and security. By storing data locally on the user's server, users maintain full control over their personal information.

ESP Home

ESPHome is an open-source software platform specifically designed for programming and managing *ESP8266* and *ESP32* microcontrollers. It allows users to create custom firmware for these microcontrollers to integrate and automate them in smart home applications. With ESPHome, sensors, actuators, and other devices based on ESP chips can be easily configured and operated.

What sets *ESPHome* apart is its user-friendly nature and flexibility. Configuration is done through simple **YAML** files where users define which hardware components are connected and how they should operate. *ESPHome* then handles generating the corresponding firmware and flashing it onto the microcontroller. This makes the device programming and integration process accessible to users with varying technical knowledge.

ESPHome seamlessly integrates with Home Assistant, enabling smooth communication between ESP-based devices and the *Home Assistant* system. This allows users to control and monitor their smart home devices through the *Home Assistant* interface, simplifying setup and management.

Another benefit of *ESPHome* is its support for a wide range of sensors and actuators, including temperature and humidity sensors, relays, LEDs, and many others. This versatility makes *ESPHome* a robust solution for creating custom automations.

Usage in this Project

In this project, *Home Assistant* served as the central platform for automation, handling the integration, control, and monitoring of various components.

ESPHome was utilized for managing the *ESP32* microcontrollers. It provided a straightforward method to configure and deploy firmware for ESP-based devices using **YAML** files. This approach simplified device setup and customization.

The seamless integration between *ESPHome* and *Home Assistant* enabled direct control and monitoring of ESP-based devices via the *Home Assistant* interface. This capability facilitated the creation of complex automation based on data from diverse sensors.

To start the project, *Home Assistant* was first installed on the *Raspberry Pi* using the installation program from https://www.home-assistant.io/installation.

An empty installation of *Home Assistant* was then available. After the installation, there were only. two things left to configure: the user and the access to the internet over wireless network. *ESPHome* was downloaded and installed through the app store integrated in *Home Assistant*. The *ESP*s were then prepared accordingly by connecting them to the *Raspberry Pi* with an USB cable. *ESPHome* established a serial connection to the *ESP* via the USB cable, allowing the initial setup to be loaded onto the *ESP*. Once this was done, the *ESP* could also be addressed wirelessly. This completed the basic setup, and the *ESP* could now be loaded with the configuration files to perform the specific functions required for this project.

MQTT

Author: Simón Eduardo Muñoz Buchwald

What is MQTT?

MQTT stands for “Message Queuing Telemetry Transport” and is a Network System protocol that is primarily designed for electronics that communicate between different devices such as the ones used for the Internet of Things. In this schematic, there is one, in general, more powerful, device as a broker, and in the same network, there can be a full range of different devices like ESPs, temperature sensors, or cameras. All these electronics count as clients and send all the information they produce under the function of MQTT to the broker and also receive from its data. In comparison to a normal network like everybody used to know, the information is targeted from a specific computer in this network to a specific computer. There is on a daily basis no difference to this process but on special occasions. When looking at MQTT the concept of work is completely different because instead of targeting a specific device for messaging information the main function of it is to broadcast all information it produces to every participant in this network. So, every device is theoretically and practically able to react to every message in this network and to filter it. The protocol uses not only messages that are called payloads but also a topic to differ between all. Up to that, it's also able to separate devices into categories which could be rooms, a set of sensors, or logical groups. A topic can be like the following “Temperature Sensor” so everyone listening to this tag would instantly get information and be able to react e.g. shutter the window, or turn on the AC. Since this wouldn’t be an ID and only text it could be various sensors providing updates on the same topic but on different rooms. To separate this and be able to respond depending on the room the sensors are in the tool is to use slashes. For example, a topic could be divided into the following “Living Room/Sensor/Temperature” and “Bed-Room/Sensor/Temperature”. Sometimes it’s necessary to listen to every message on a topic, in this case, there is the # for. “Living Room/Sensors/#” means that every Message from any topic under this will be read and interpreted. A typical case for this is displaying information in a visual dashboard where there is no need to differ sensors. There are no rules for choosing topics except for these basic rules in addition to the topic being case sensitive so “Living room/temp” and “Living Room/Temp” are not the same, but using spaces is allowed in it.

MQTT used in this project

In this fuel cell project, there is a MQTT network in use running on a Raspberry Pi 5 as the broker and the brain. On this device, Home Assistant is running which contains an integrated MQTT program where the configuration and the settings take place. The remaining hardware in the network is 3 ESPs with the function of being clients. On them are all the hardware like sensors, fans, and solenoids where the ESPs function as antennas for them in this concept.

The installation of MQTT on the Home Assistant was already done by installation and only needed an activation. There is also no use in creating additional users since MQTT can import the system's users themselves. This is important to log into the network traffic which is necessary for committing and sending messages. On the side of the ESPs there is the yaml in which must be the following lines written:

The first line represents the component in this case for the protocol. Following are the variables, on which address the ESP can find the broker. This doesn’t have to be a static IP address but can also be “homeassistant.local” including the port which is also used to access the Home Assistant interface. Next, the port number which is 8123 when not changed. The username and password are the ones used for the Home Assistant

Code examples

In our example all variables have the tag !secret which means the secret file in the folder contains the real values.

```
mqtt:
  broker: !secret mqtt_broker
  port: !secret mqtt_port
  username: !secret mqtt_user
  password: !secret mqtt_pass
```

Not common but helpful to know if all devices are only and working properly is the following coding block in which it publishes all 5 seconds under the topic “live/h2fans” the payload “I am working”. This almost identical block is also found in the other two ESPs with the names “live/master” and “live/esp_temp”. Being able of the hash when somebody listening to the topic “live/#” it’s receiving 3 messages every 5 seconds.This is only for error-finding issues and is not necessary in the program flow.

```
interval:
- interval: 5.0s
  then:
        - mqtt.publish:
        topic: live/h2fans
        payload: "h2fans is working!"
```

For example, for publishing data from a sensor or device there is the function “state_topic: NameOfTopic” in which the ESP broadcasts all the values in a specific update interval produced by the object. The following code uses this snippet to send every second the fan speed with the unit “rpm” under the topic of “fan/Out_RPM”.

```
 - platform: pulse_counter
        pin: GPIO12
        name: "Fan Out RPM"
        id: Fan_Out_Rpm
        unit_of_measurement: 'RPM'
        update_interval: 1s
        state_topic: fan/Out_RPM
```

There is also the option to let the ESP do specific things if it receives a topic and even one with a specific payload, which is basically a text string. The following code means that if under the topic “master/switchToggle” and the payload “toggle” a message is received the ESP will publish under the same topic the payload “switch triggered” and also toggle the switch with the id: relay_1, which is the activation switch for the fuel cell.

```
on_message:
  topic: master/switchToggle
  #payload: "toggle"
  then:
          - switch.toggle: relay_1
          - mqtt.publish:
          topic: master/switch
          payload: "switch triggered"
```

Like in the normal yaml there is also in MQTT the possibility to use lambda code, which are like templates with minimal limitations. In our case we used this function to synchronize the speed of the fans with the temperature of the sensors. The DHT Sensor, which stands for digital humidity temperature, sends every 2 seconds under the topic “temp/dht/temp” the measured temperature to the system. With this value the other ESP can decide to correct speed for the fan. The plan was to react to a linear function where 20 degree means 50% speed and 35 degree is 100%.

```
  - platform: dht
          pin: GPIO26
          model: DHT22
          temperature:
          name: "DHT-Temperatur"
          state_topic: temp/dht/temp
          humidity:
          name: "DHT-Humidity"
          state_topic: temp/dht/hum
          update_interval: 2s
```

In the lambda part there are a few codes like std::stof which converts the given value to a float and this value is x.c_str(). X in lambda is the input value which is in this context the temperature from the sensor. Next line is the calculation to divide the temperature by 30 and subtract 0.1667, so the result of this is the linear values we wanted. To handle errors, it won’t accept values under 0 or over 1.0 because these are the boundaries of 0% and 100%, so it will correct them to max or min. In the last two lines the variable will be written to the fans with the ids with the set_level function.

```
on_message:
        - topic: temp/dht/temp
        then:
        - lambda: |-
        float temperature = std::stof(x.c_str());
        float fan_speed = (temperature / 30.0) -
        0.1667;
        if (fan_speed < 0.0) {
        fan_speed = 0.0;
        } else if (fan_speed > 1.0) {
        fan_speed = 1.0;
        }
        id(Fan_In_Pwm)->set_level(fan_speed);
        id(Fan_Out_Pwm)->set_level(fan_speed);
```

Microcontrollers and Sensors as Hardware

Author: David Starcevic

A solid hardware foundation is essential for collecting sufficient data to ensure system safety. Sensors are installed to monitor various parameters, such as temperature, in real time. The collected data is used for automation through microcontrollers.

Raspberry Pi 5

A commonly used mini-computer for such projects forms the central control unit of the system. The Raspberry Pi 5 was chosen for this purpose. This component offers optimal functions for controlling a compact system. Its compatibility with *Home Assistant* and *ESPHome* allows for clear and automated control of the integrated hardware modules.

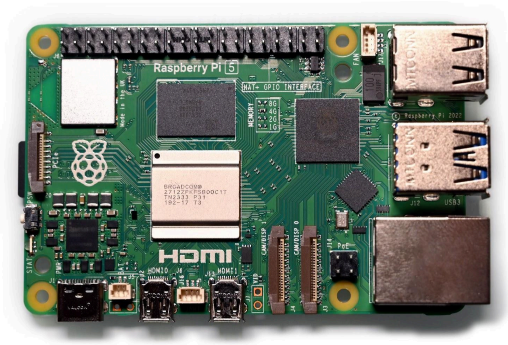

Figure 4.1: Raspberry Pi 5.

Source: SimonWaldherr, “Raspberry Pi 5”, Wikimedia Commons, https://commons.wikimedia.org/wiki/File:Raspberry_Pi_5.jpg, licensed under CC BY 4.0. Accessed July 5, 2024.

ESPs - 32 Bit Microcontrollers

The choice of microcontrollers fell on three ESP32 units due to cost reasons and their numerous functions. In addition to basic functions (computations, physical interfaces), they offer the ability to establish both Bluetooth and Wi-Fi connections (Pereira et al. p.225). This enables wireless connectivity, allowing for a more compact design and the wireless transmission of sensor data to the central unit, the Raspberry Pi 5. This setup has the advantage of placing the Raspberry Pi in an easily accessible control box, while the rest of the technology can be placed directly at the intended locations without complex wiring, requiring only a power supply for the microcontrollers.

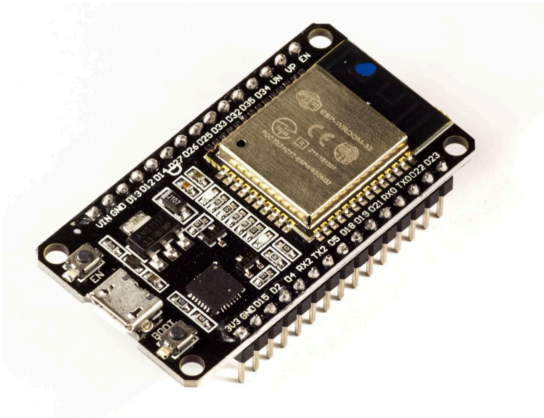

Figure 4.2: ESP32**.**

Source: Ubahnverleih, “ESP32 Espressif ESP-WROOM-32 Dev Board”, Wikimedia Commons, https://commons.wikimedia.org/wiki/File:ESP32_Espressif_ESP-WROOM-32_Dev_Board.jpg, licensed under CC0 1.0 UNIVERSAL. Accessed July 5, 2024.

Hardware Modules

Relays

Two relays are needed to control the fuel cell. These are switches that can be opened and closed with a signal, allowing for functions such as imitating a button press or toggle a signal.

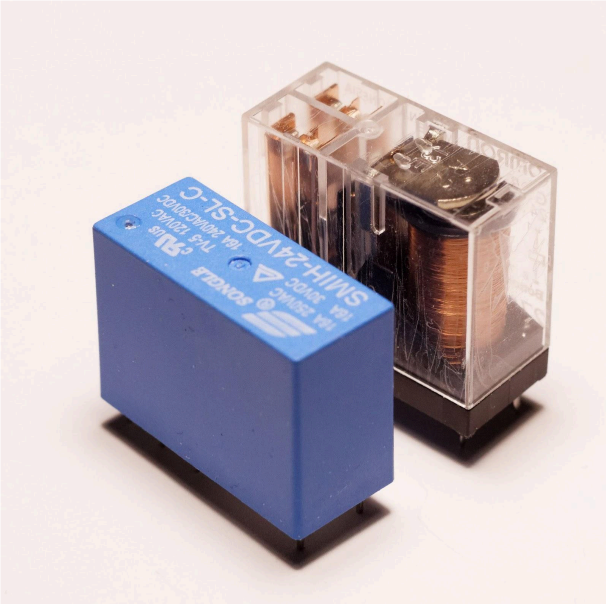

Figure 4.3: Relay

Source: n.d., “Omron G2R (clear), Songle SMIH (blue) and Schrack RT (orange) series relays”, Wikimedia Commons, https://commons.wikimedia.org/wiki/File:Omron_G2R,_Songle_SMIH_series_relays_01.jpg, licensed under CC0 1.0 UNIVERSAL. Accessed July 5, 2024.

Sensors

The system's sensors are limited to the most critical areas. Three sensors measure the temperature and humidity inside the enclosure, the hydrogen concentration, and the temperature of the hydrogen tank.

Hydrogen Sensor (MQ-8)

This sensor measures the hydrogen concentration in the environment and outputs a ppm value (parts per million). Conrad and Kaulbars found out that the threshold for the explosion is 40,000 ppm (4 vol.%) (Conrad & Kaulbars, 1995). Therefore, it is essential for safety reasons not to exceed this limit, and the value must be regularly monitored. The hydrogen sensor is designed to alarm within much lower values, ensuring safe handling of the gas. Since hydrogen rises, the sensor should be mounted as close to the ceiling as possible for representative readings.

Figure 4.4: MQ-8

Source: reichelt elektronik GmbH, “Maxdetect Humidity and Temperature Sensor”, reichelt.de, https://cdn-reichelt.de/bilder/web/artikel_ws/A300/SEN-MQ-8-1-KLEINNEU.jpg. Accessed July 5, 2024.

Temperature and Humidity Sensor (DHT22)

The DHT sensor monitors both the ambient temperature and humidity. It is used to keep track of these values and provides accurate readings despite its low cost. Since warm air rises and the system needs to be checked for heat, the sensor should also be mounted high up.

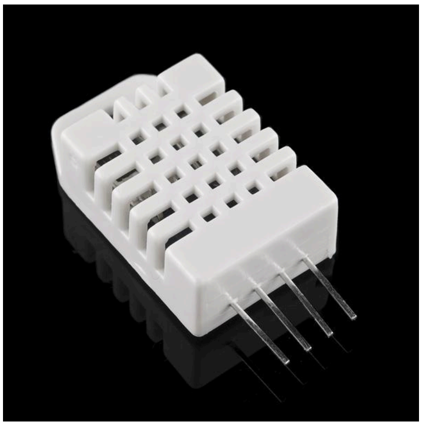

Figure 4.5: DHT22

Source: SparkFun, “Maxdetect Humidity and Temperature Sensor”, Wikimedia Commons, https://commons.wikimedia.org/wiki/File:Maxdetect_Humidity_and_Temperature_Sensor_-_RHT03_10167.jpg, licensed under CC BY 2.0 GENERIC. Accessed July 5, 2024.

Thermocouple (Type K) + MAX6675

The thermocouple is a sensitive, needle-shaped thermometer. Own measurements showed that it provides precise temperature measurements at the tip. It is used to monitor the temperature of the hydrogen tank. The tip of the sensor should therefore be mounted directly to the hydrogen tank. The thermocouple data is interpretable only when connected to a microchip, for which the MAX6675 is used.

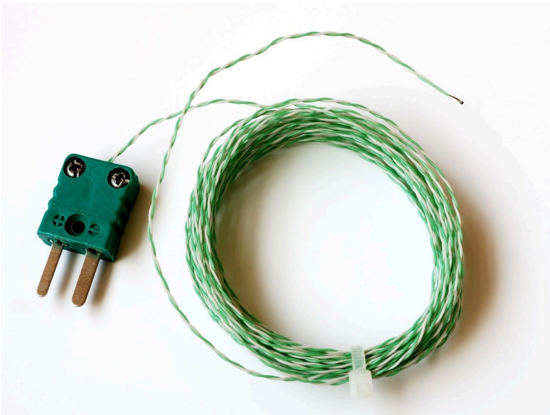

Figure 4.6: Thermocouple (Type K)

Source: Harke, “Thermocouple Type K”, Wikimedia Commons, https://commons.wikimedia.org/wiki/File:Thermocouple_K_(2).jpg, licensed under Public domain. Accessed July 5, 2024.

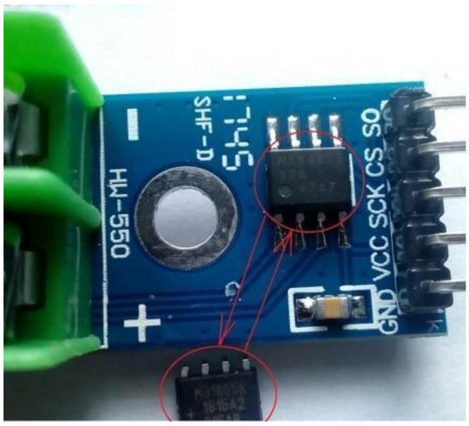

Figure 4.7: MAX6675

Source: Xyz00030280, “MAX6675 and Chip max31855K”, Wikimedia Commons, https://commons.wikimedia.org/wiki/File:MAX6675_and_Chip_max31855K_for_termocouple_measurement.jpg, licensed under CC BY-SA 4.0. Accessed July 5, 2024.

FANS

Two 4-pin fans are installed to ensure adequate air circulation within the enclosure. One fan handles the intake, and the other handles the exhaust. The fans can be regulated using a PWM signal, and their current speed (in rpm) can be output.

Figure 4.8: 4 Pin Fan.

Source: Jacek Halicki, “Corsair LL120 RGB”, Wikimedia Commons, https://commons.wikimedia.org/wiki/File:2023_Corsair_LL120_RGB_(2).jpg, licensed under CC BY-SA 4.0. Accessed July 5, 2024.

System Integration

As previously mentioned, three ESP microcontrollers are installed, each covering a specific area. One ESP handles the two relays that control the fuel cell and is mounted near the fuel cell along with the relays. Another microcontroller is responsible for the thermocouple and the DHT sensor. Since the DHT sensor and the hydrogen tank are mounted on the ceiling of the enclosure and the thermocouple is placed directly at the hydrogen tank, this ESP finds its spot within this area, right below the hydrogen tank. The last microcontroller is connected to the H2 sensor and the system’s two fans. It is mounted on the ceiling between the two fans to ensure accurate H2 sensor measurements while minimizing the distance to the fans.

Wiring and Cable Management

Author: Mika Günther

Good cable management and wiring are important for keeping things organized and safe. Properly handling cables helps reduce clutter, prevent electrical hazards, and make maintenance easier.

Preparation and cabling techniques

KiCad

KiCad is an open-source software suite for electronic design automation (EDA) that simplifies the process of designing and creating printed circuit boards (PCBs). The software allows for the comprehensive integration of various tools and services needed for PCB design, including schematic capture, PCB layout, and 3D visualization. This enables users to efficiently manage and develop all aspects of their electronic design projects in one unified environment.

KiCad is renowned for its flexibility and extensibility. It supports a wide range of component libraries and can be easily customized to meet the specific needs and technical requirements of individual users. Configuration is typically managed through clear and accessible settings, providing users with high granularity and control over their designs. Additionally, *KiCad* offers a user-friendly interface and extensive documentation to facilitate ease of use and quick learning.

Another significant advantage of *KiCad* is its focus on community and collaboration. Being open-source, it encourages contributions from users worldwide, resulting in a continuously improving and up-to-date tool. Moreover, because it is free to use, *KiCad* is accessible to a wide audience, from hobbyists and students to professionals, promoting innovation and learning in the field of electronic design.

Soldering

One way to permanently join components or wires is through a process called soldering. In this process, a tin blend called solder is melted around the connections of wires or components to join them. This ensures a reliable electrical and mechanical bond. Soldering is essential for assembling circuit boards, repairing electronic devices, and creating strong, conductive connections.

Cabling techniques

There are different possibilities for protecting cables in an assembly. For this project, a protection tube was the best option due to its versatility. It protects individual cables from physical damage and prevents them from shifting to places where they don't belong. Additionally, it provides an effective way to route and hide the cables for better aesthetics. Protection tubes are commonly used in industrial, commercial, and residential settings to safeguard electrical and data cables.

The slots in *aluminum profiles* provide a versatile solution for cable management, allowing smaller cables to be neatly hidden within the slots for a clean appearance. For larger cables, cable tie anchors can be used to securely fasten them to the *profiles*. This method ensures that all cables are well-organized and securely held in place, enhancing both the functionality and aesthetics of the installation. By utilizing these slots and anchors, users can achieve a tidy and professional setup, making maintenance and adjustments straightforward.

Cables

2x0.75mm^2 cables were used for power distribution, handling supply voltages of 5V, 12V, and 24V with currents not exceeding 5A. These cables were chosen due to the low voltage and current requirements of our system. For data transmission, we opted for 0.14mm^2 cables, ensuring efficient and reliable communication between components.

Usage in this project

In this project, KiCad was employed to meticulously design the schematics for our ESPs, as detailed in Chapter 2, “Microcontrollers and Sensors as Hardware.” This software enabled us to create precise and comprehensive diagrams that facilitated a clear understanding of the connections and layout.

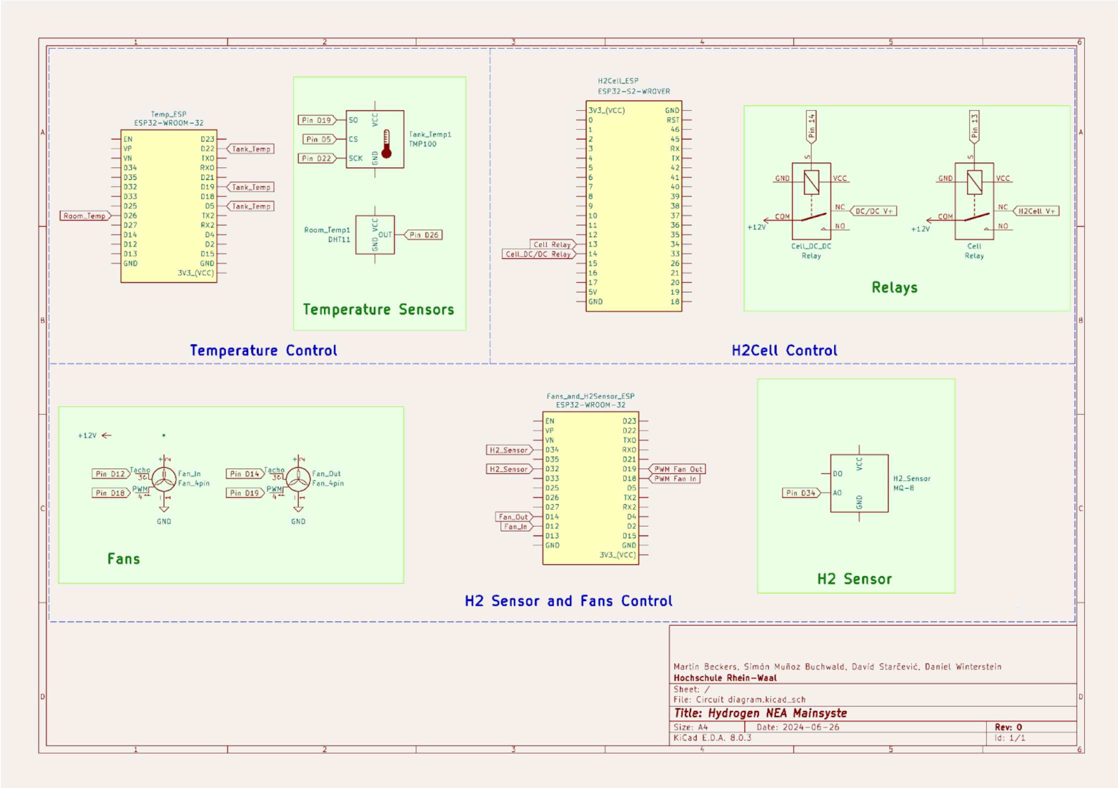

Figure 5.1: Schematic of the 3 ESPs

Source: Martin Beckers, Simón Muñoz Buchwald, David Starčević, Daniel Winterstein; Hochschule Rhein-Waal

For establishing the connections between the ESPs and sensors, as well as linking to the individual power lines, we utilized a soldering iron. This technique ensured that our connections were not only secure but also permanent, providing reliable performance over time.

To protect the wiring from potential environmental hazards and to improve the overall visual appeal of our setup, we implemented protective tubing. Additionally, we strategically utilized the slots in the aluminum profiles to achieve neat and organized cable management. This dual approach not only safeguarded the cables but also contributed to a clean and professional appearance, enhancing both the functionality and aesthetics of our project.

Wiring of the electrical cabinet

The electrical cabinet was done by the trainee's of Omexom, which used "Eplan P8". The Schematic of the electrical cabinet can be seen in the Appendix.

Coding and implementing the logic

Author: Daniel Winterstein

The initial steps and logic implementation for a control system are outlined, focusing on using a *Raspberry Pi 5* and *Home Assistant* to manage microcontrollers, sensors, and fans. Preparatory work involved creating a program flowchart and coordinating with the team to ensure component requirements were understood, facilitating smooth communication and operation. The system utilizes three *ESP32* microcontrollers for efficient wiring and pin allocation, with specific configurations for relays, sensors, and fans. Detailed explanations cover the integration and functioning of the hydrogen sensor, *DHT22* temperature and humidity sensor, and the thermocouple with a *MAX6675* amplifier, all contributing to a dynamic and responsive control system.

First steps and preparation

In a subsequent step, the program flowchart was created. The interrelationships, detailing how components act and react (autonomously or interdependently), are further explained in section 6.3. Solution Architecture.

The preparations described above also included exchanges with the entire team (not just the remaining software developers) about the components to be used for the prototype. This facilitated a better understanding of the emerging requirements from day one of the project - also aiding in the planning of the control implementation for individual elements.

These preparatory steps were time-consuming but necessary and important for further work, as they provided the foundation for constructive, rapid, and successful progress going forward.

Programming Objective / Target of Programming

The goal of the program logic is to ensure smooth communication between the individual components and to provide information about the operation of the system.

This means, for example, that the fans should run at a certain level depending on the temperature and/or hydrogen content inside the generator housing, while the values read from the respective sensors should provide the user with informative data (humidity, temperature inside the housing/at the hydrogen tank, fan speed, status of the relays).

Component control

In this system, three microcontrollers (ESP32) are used. Theoretically, a solution with just one ESP32 would be possible, but an extended variant was chosen here for the sake of wiring (shorter cable distances) and pin allocation (no bottlenecks). The initial division into relays, sensors, and fans was followed, but the final installation in the housing showed that it was more practical to place the H2 sensor together with the fans on a single ESP32. This, however, does not make any difference in the program implementation, so the respective components can continue to be categorized accordingly.

Relays

For system startup, the functionality of the relays is crucial. The program flow should depend on the state of the second relay, which indicates whether the system is operational or not,

- **System off (Relay-2 off):**
  - → *Start the system*
  - → *Relay-1: on*
  - → *Delay: 5 seconds*
  - → *Relay-1: off*
  - → *Delay: 30 seconds*
  - → *Relay-2: on*

- **System on (Relay-2 on):**
  - → *Shut down the system*
  - → *Relay-1: on*
  - → *Delay: 5 seconds*
  - → *Relay-1, Relay-2: off*

Since Relay-1 always behaves the same when it receives a signal (it turns on for 5 seconds), the status of the second relay determines whether the system is started or shut down. A query in the program flow regarding this status (**switch.is_off: relay_2**) provides this information.

A continuous signal for 3 seconds starts or shuts down the fuel cell, with a duration of 5 seconds chosen for margin reasons. After 10 seconds, the system should be fully operational; additionally, a grace period is planned so that the DC/DC converter only starts after 30 seconds and supplies the battery with electricity generated from hydrogen.

Sensors

In total, three sensors were installed in the prototype:

1. **Hydrogen Sensor (MQ-8)**

To determine the air-hydrogen mixture inside the generator housing, a calculation is performed in the program code based on the value transmitted by the sensor. This calculation is composed of:

- **RL:** Load resistor
- **RO:** Sensor resistance in clean air (constant that must be calibrated for the specific sensor)
- **VC:** Sensor supply voltage
- **ADC_MAX:** Maximum ADC value (Analog-to-Digital Converter) of the ESP32.

Using these values and the voltage currently measured at the sensor, the sensor resistance can be determined. From this, a ratio between the current measured resistance and the resistance in clean air can be established. The hydrogen concentration is represented in ppm (Parts Per Million) and calculated using a logarithmic formula based on the specific characteristics of the sensor.

```
float sensor_volt = id(mq8_voltage).state;
float RS = (VC - sensor_volt) * RL / sensor_volt;
float ratio = RS / RO;

float ppm = pow(10, (log10(ratio) - 0.8) / -0.6);
return ppm;
```

Temperature and Humidity Sensor (DHT22)

Compared to the above, a DHT sensor is simpler to handle in the implementation because one can rely on drivers (or driver packages) specifically designed for it in the configuration file, and no calculations are required. Besides this driver, the connected pin, entity names for both temperature and humidity (for external access), and a sampling rate (**update_interval**) are specified.

```
- platform: dht
  pin: GPIO26
  model: DHT22
  temperature:
    name: "DHT-Temperatur"
  humidity:
    name: "DHT-Humidity"
  update_interval: 2s
```

Thermocouple (Type K) + MAX6675

Similar to the previously described DHT sensor, the programming effort for a type K thermocouple is fundamentally straightforward. However, in this case, an amplifier must be interposed between the ESP32 and the thermocouple because the thermocouple itself produces an unamplified signal, akin to a cable or antenna. The MAX6675 used here serves as such an amplifier. In contrast to a DHT sensor, this variant triples the number of pins to be controlled to three (excluding VCC and GND). Therefore, an SPI (Serial Peripheral Interface) must be configured in order to a

1. **“Clock-Pin”** (Pin for synchronizing data transmission) as well as
2. **“MISO-Pin”** (Master In Slave Out), which sends data from an SPI device (Slave) to the microcontroller (Master).

The remaining third pin is assigned a platform, similar to the DHT previously, named "max6675," which provides all necessary drivers and libraries for communication.

However, a cs pin (Chip Select Pin) is selected to inform the MAX6675 when data should be transmitted over the SPI (Serial Peripheral Interface).

```
spi:
  clk_pin: GPIO22
  miso_pin: GPIO19

sensor:
  - platform: max6675
    name: "Thermoelement-Temperatur"
    cs_pin: GPIO5
    update_interval: 1s
```

These three elements are used to monitor critical values, which are also considered in the program flow, but typically intended only for display on the Home Assistant dashboard for informational purposes. However, safety mechanisms have been considered and incorporated.

Fans

The control of the two fans used (supply air and exhaust air) is dynamic, meaning it depends on the values from other sensors. The speed of the fans varies based on the temperature value sent by the *DHT22* sensor. A formula normalizes this temperature to a value between 0 and 1 (representing 0% to 100% of maximum speed). Additionally, a check ensures that the calculated value does not exceed these boundaries. Subsequently, the speed value is transmitted to the fans.

```
float fan_speed = (temperature / 30.0) - 0.1667;

if (fan_speed < 0.0) {
   fan_speed = 0.0;
} else if (fan_speed > 1.0) {
   fan_speed = 1.0;
}
```

```
id(Fan_In_Pwm)->set_level(fan_speed);
id(Fan_Out_Pwm)->set_level(fan_speed);
```

It is entirely possible to assign different values for supply and exhaust air here; however, it is not necessary in the current development phase of the generator. Similarly, this method can also incorporate the values of other sensors described here, as well as potential additional sensors that may be installed and implemented in the future.

Software Development Activities

The IT team focused on developing and implementing the software needed for the hydrogen generator's operation. This included:

- **Source Code Repositories**: Centralized repositories (e.g., GitHub) where all code was stored with version control.
- **Code Documentation**: Inline comments and external documents explaining the purpose and functionality of each code segment.
- **Configuration Files**: Documentation of configuration files for Home Assistant, ESP Home, and MQTT setups.
- **Development Logs**: Daily logs of development progress, challenges, and solutions.

Assembling

Author: Tadiwanashe Dondo

Assembly of the Generator

Assembly started with detailed planning and precise measurements taken from a 3D model, which guided the entire assembly process. These measurements were carefully marked on high-quality aluminum profiles, chosen for their strength and lightweight properties. The entire assembly process took almost 10 hours for the students in the assembly group.

Cutting and Preparing the Aluminum Profiles

Using an electric saw, the aluminum profiles were cut with precision to match the exact dimensions required for the generator's framework. Special care was taken to ensure the cuts were accurate, facilitating a seamless assembly process. After cutting, the edges of the profiles were smoothed to eliminate any sharp edges that could pose a safety hazard or interfere with the assembly.

Assembling the Outer Structure

With the profiles prepared, the next step was to assemble the outer structure of the generator. The aluminum profiles were joined using sturdy brackets and screws, ensuring the frame was both secure and stable. This would serve as the backbone of the generator.

Constructing the Internal Framework

Following the completion of the outer structure, smaller cut aluminum profiles were strategically placed inside the main frame. These internal profiles were crucial for supporting the various components of the generator. The arrangement formed a skeleton that would hold the generator's parts firmly in place.

Installing Internal Components and Wiring

Once the skeleton was established, the internal components and wiring were carefully installed. The components were secured using custom-designed 3D-printed holders, which were created to fit precisely within the framework, providing both support and stability.

Insulating and Sealing the Generator

To protect the generator from environmental factors like dust and moisture, insulation was a key consideration. Side panels were accurately cut to fit the dimensions of the generator, creating a protective barrier against external elements. These panels were attached securely to the sides, with caulk applied generously to the edges to enhance the water-tight seal. This step was crucial in ensuring the generator could operate efficiently in various conditions without the risk of damage from dust or water.

Organization of the project

Author: María Verónica Barrera Moreno

This chapter will describe how the project was organized. The project was supervised by one professor and two assistants. One project manager was elected to keep track of other students' progress. Deadlines and progress were already defined by Omed Abed before the first kickoff meeting. Each team member elected a task in a democratic way. At the first kickoff meeting, it was made clear we needed a way to communicate as a group. The majority voted to use Discord as a communication and information platform, therefore David Starcevic created a server there. Also in the first class expectations and goals were shared. As requested by Prof. Dr. Irmgard Buder, a protocol was made for every class and the first meeting with Omexom.
A budget wasn't considered for this project since some things were provided by the company, some by the university, and the rest by Omexom who bought them from a list done by students within a previously discussed price range allowance.
The organization of this project was facilitated by the project manager, who kept track of deadlines, tasks, and communication between groups.
The tools used to communicate in general were Discord, Microsoft Teams, and personal group meetings, where updates and presentations were made.
Assistant was taken to improve the attendance of team members.
Excel sheets were done regularly to track every new task and the person responsible.
The Gantt chart had the function of tracking progress and accounting for delays. New deadlines were put in the said chart to be realistic with the progress due to difficulties that set the project behind schedule.
The project manager used the democratic style to lead, such as being a team player, being willing to adapt, having a fair mind, and being engaged in the process. As well as seeking diverse opinions and not trying to silence dissenting voices or those that offer a less popular point of view. To best use democratic leadership in an effective way it was needed to ensure people have the necessary skills to succeed, making sure there is plenty of time, giving everyone a chance to contribute, and making expectations clear.
The deadlines were not fulfilled. The 3D model took more time than expected due to multiple corrections and wishes to be more compact. As well minor components were added to a later date. Some of the hard wire was done by Omexom who brought the electric cabinet. The IT part was put to testing on a later date, with the final completion on the delivery day. Communication with the company was lacking, especially because we were not fully aware of their tasks. Components were ordered behind schedule to the point of making a second order list near the end. Testing was not done as many times as wished. The division of groups worked out great. The only problem was that

the group leaders weren't able to equally divide the tasks inside their sub-groups. All groups were able to complete tasks despite not all members doing their part. The IT group was qualified to arrange between themself without having a leader and was the most successful group. Other groups were also successful like the assembly where not all responsibility relied on one person to motivate others. Having a leader inside groups wasn't an efficient way of organizing. The tasks were completed but a lot of burden went to some more than others. The workload wasn’t divided equally and communication on discord was lacking. The project manager kept track and updates of what everyone was doing to improve efficiency but some time was wasted on not using the full potential of each student. Communication, division of workload, keeping up with schedules could have been done better. This knowledge was used to be able to keep with the correct submission deadline of the poster and report. Where division of tasks were done as equally as possible based on the knowledge of each participant and deadlines were put to an earlier date than usual to have plenty of time for corrections. At the end this extra time wasn't used by the proof-reading team. A lot of things could have been done better but it was a good experience on how to handle a group and acknowledge that depending on the people, different ways of organization might be needed.

Documentation

Author: Md Zilani Hossain

Documentation Overview

The documentation of the "H2-Power substituting diesel generator/Wasserstoff ersetzt Dieselgenerator" project serves as a comprehensive record of the entire process, from conceptualization to implementation, testing, and final analysis. The primary purpose of documenting this project is to ensure transparency, facilitate knowledge sharing, support replication, and provide a basis for future improvements. Detailed documentation was essential for keeping track of all activities and milestones achieved during the project.

Why and How the Project is documented

The project was documented to keep track of all activities and milestones. A documentation group was created to help with the final report, interviewing members of each group to record their contributions. This meticulous process includes detailed written records, visual aids, and comprehensive reports covering all aspects of the project, ensuring transparency, knowledge sharing, replication, and compliance with standards. This collaborative effort was crucial for maintaining a comprehensive record of the project's progress and outcomes.

Project Goal

The goal of this project is to create a self-sustaining power generator that utilizes hydrogen as an energy source for a fuel cell, which in turn charges a battery. Additionally, the project aims to implement a remote control system using Wi-Fi and Home Assistant, allowing for efficient and convenient monitoring and management of the generator.

Methods Used for Documentation

The documentation tasks were distributed among team members and specialized teams to ensure a collaborative effort. The methods used include:

1. Kickoff Meetings and Initial Research

Initial Research and Planning: The kickoff meeting was held on April 3rd, where Professor Bruder and Mr. Omed Abed described the project. Following this, specialized teams were formed for research and to create initial sketches of the project. The team then visited Omexom on July 19th to discuss the project further with experts, refining the project goals and gathering insights. Meeting minutes were recorded to capture discussions and decisions.

2. Component Selection and Analysis

Documentation of Component Selection: The component team, led by Md Ferdous Amin, compiled detailed lists of all components with specifications, ensuring clarity and completeness. They also documented supplier information and links to purchase components, facilitating easy procurement and verification. Compatibility studies were conducted to ensure each component's suitability and safety with hydrogen technology, and safety data sheets (SDS) were included to ensure compliance with safety regulations.

3. Design and Modeling

Documentation of Design and Modeling: Fungai Machingaifa led the 3D modeling team, assisted by Tadiwanashe Dondo, Khawaja Ahsan Elahi, Mohammad-Aminul-Islam Tanvir, and Rithik Kumar. The team used Fusion360 to create detailed 3D models and sketches of the prototype design. They also created system architecture diagrams and flowcharts to depict the system's operational flow. Mika Günther provided corrections and final approval for the 3D models. In this whole process, they have used the software Fusion360 for 3D modeling and diagramming tools for creating system architecture diagrams.

4. Software Development and Logic Implementation, Assembly and Integration, Testing and Validation

The software development was managed by Martin Beckers and Simón Eduardo Muñoz Buchwald, who utilized GitHub for version control, and Home Assistant, ESPHome, and MQTT for configuration and integration. They ensured thorough inline code documentation and maintained daily development logs. The assembly and integration tasks were handled by Tadiwanashe Dondo, Fungai Machingaifa, and Mika Günther, who documented the process with step-by-step guides, illustrations, and wiring diagrams using KiCad. The testing and validation phase was led by David Starcevic, with the IT team meticulously recording test results, creating failure analysis reports, and validating system performance against predefined criteria.

5. Presentations

- **Presentation at Omexom (July 1st):** The prototype was presented at Omexom GmbH on July 1st, where a video about Omexom was shown, followed by speeches and a demonstration of the generator. David Starcevic reported that the presentation went well, with everything working as expected, and the audience was impressed.
- **University Presentation (July 15th):** A follow-up presentation was held at the university on July 15th. The IP team presented a poster summarizing the project, highlighting key achievements and future work.

Conclusion

The documentation of the "H2-Power substituting diesel generator/Wasserstoff ersetzt Dieselgenerator" project ensured that every phase was meticulously recorded. This comprehensive documentation serves as a valuable resource for future research and development in the field of hydrogen power generation, supporting transparency, knowledge sharing, replication, and improvement.

Discussion

Author: Rithik Kumar

About Discussion

Converting from traditional diesel generators to hydrogen fuel cell generators is an important step toward reducing carbon emissions and improving fuel efficiency and energy sustainability. This discussion will interpret the key findings of our project, assess their impact in terms of environmental and economic impacts, and highlight the strengths and limitations of our approach. Additionally, we will explore potential future directions for research and technology advances that could further improve the viability of hydrogen fuel cells as a replacement for diesel generators.

Interpretation of Results Efficiency and Performance

Our project demonstrated that hydrogen fuel cell generators are more efficient than diesel generators. Fuel cells operate at around 60% efficiency, significantly higher than the typical 30-40% efficiency of diesel generators. This increased efficiency results in higher energy production per unit of fuel consumed, making hydrogen fuel cells a more efficient power generation option, especially in applications requiring continuous, uninterrupted power output.

In addition, the hydrogen fuel cell system has proven to have high-quality performance characteristics. Unlike diesel generators, which have a noticeable lag during startup and load changes, hydrogen fuel cells provide a quicker response time and more stable operational output. This characteristic makes them suitable for applications where reliability and rapid power availability are critical, such as in backup power systems for data centers or hospitals.

Environmental Impact:

One of the most compelling findings of our project is the substantial reduction in greenhouse gas emissions when using hydrogen fuel cells instead of diesel generators. Diesel engines emit significant amounts of carbon dioxide (CO2), nitrogen oxides (NOx), and particulate matter (PM), all of which contribute to air pollution and climate change. In contrast, hydrogen fuel cells produce only water vapor as a byproduct, resulting in zero direct emissions of CO2 and other pollutants.

The adoption of hydrogen fuel cells could thus play a crucial role in mitigating the environmental impacts associated with fossil fuel-based power generation. For example, replacing diesel generators with hydrogen fuel cells in a mid-sized power application could potentially eliminate several tons of CO2 emissions annually, contributing significantly to achieving carbon reduction targets.

Economic Considerations:

While the initial investment for hydrogen fuel cell systems is currently higher than that for diesel generators, the long-term economic benefits are promising. Hydrogen fuel cells have lower operating and maintenance costs due to fewer moving parts and higher energy efficiency. Over the life of the generator, these savings can offset the initial capital costs.

Furthermore, as hydrogen production and distribution infrastructure develops and becomes efficient On the cost front, Hydrogen fuel prices are expected to decrease, further improving the economic viability of Hydrogen fuel cell systems. Government incentives and subsidies for clean energy technology could also play a central role in making hydrogen fuel cells more competitive with traditional diesel generators.

Implications of the results

Practical implications:

The transition to hydrogen fuel cell generators brings practical benefits in many different fields. For example, in the transportation sector, replacing diesel generators on trains or ships with hydrogen fuel cells could significantly reduce the carbon emissions of these modes of transport. Likewise, in remote areas where grid power is not available, hydrogen fuel cells provide a clean and efficient alternative to diesel generators for off-grid power supply.

In industrial environments, hydrogen fuel cells can be used to provide critical operational power with high reliability and minimal environmental impact. This is particularly valuable in industries such as pharmaceuticals, data centers, and telecommunications, where power reliability and quality are paramount.

Technological and Policy Implications:

Adopting hydrogen fuel cells as a mainstream power generation solution will require advancements in hydrogen production, storage, and distribution technologies. The development of more efficient and cost-effective methods for hydrogen production, such as electrolysis using renewable energy sources, is essential for the sustainable scale-up of this technology.

Furthermore, supportive policies and regulatory frameworks are crucial to foster the growth of the hydrogen economy. Governments can play a significant role by providing incentives for the adoption of hydrogen technologies, investing in hydrogen infrastructure, and setting emission reduction targets that encourage the transition away from fossil fuels.

Strengths and Limitations

Strengths

Our project highlights several strengths of hydrogen fuel cells as an alternative to diesel generators. The primary strength is their superior environmental performance, with zero direct emissions and the potential for significant reductions in greenhouse gas emissions. Additionally, the higher efficiency and better performance characteristics of hydrogen fuel cells make them a viable option for various applications, particularly those requiring reliable and steady power supply.

Limitations

However, the project also identified several limitations. The current high cost of hydrogen fuel cell systems and hydrogen fuel itself is a significant barrier to widespread adoption. The infrastructure for hydrogen production, storage, and distribution is still underdeveloped in many regions, which limits the availability and accessibility of hydrogen fuel.

Moreover, the production of hydrogen, particularly through conventional methods like steam methane reforming, still has a substantial carbon footprint unless it is coupled with carbon capture technologies or produced using renewable energy sources.

Future Directions

Further Research

Further research is needed to address the technical and economic challenges associated with hydrogen fuel cell technology. This includes improving the efficiency and cost-effectiveness of hydrogen production methods, developing more efficient hydrogen storage solutions, and enhancing the durability and reliability of fuel cells under various operating conditions.

Technological Advancements

Advancements in hydrogen production technologies, such as the development of more efficient electrolyzes and the use of alternative renewable energy sources, could significantly reduce the carbon footprint and cost of hydrogen production. Additionally, innovations in fuel cell design and materials can improve their performance and longevity, making them more competitive with traditional power generation technologies.

Policy and Implementation

Governments and industry stakeholders must collaborate to create an enabling environment for the adoption of hydrogen fuel cells. This includes establishing supportive policies, investing in hydrogen infrastructure, and creating market mechanisms that incentivize the transition to clean energy technologies. Public-private partnerships can play an important role in promoting the development and deployment of hydrogen fuel cell systems.

Conclusion

Author: Md Ferdous Amin

The project “H2-Power Substituting Diesel Generator/Wasserstoff ersetzt Diesel generator inasmuch is a laudable step towards developing sources of sustainable and renewable energy systems. The technical experimental work of this multi-disciplinary project along with a group of dedicated students with the help of Industrialists was able to produce a prototype of a hydrogen power generator that is expected to replace the diesel generator. Thus the outcome of the experiment depicts that hydrogen can be a feasible, efficient, and clean source of energy.

Project Results:

1. Technical Innovation: The experiment highlighted the possibility of adapting the current manner of operating by including hydrogen electricity virtually incomprehensible changes. The prototypic hydrogen power generator was created successfully and connected; to the application of new knowledge such as electrolysis, fuel cells, and microcontrollers.

2. Effect on the Environment: Another major achievement of the project is that the cubic meter of greenhouse gases has been cut down. Hydrogen power generators do not emit any direct and tangible byproducts except water vapor is the only byproduct. This brings out the conservation of the environment aspect in the utilization of hydrogen power and the need to decrease carbon emissions from power generation and climate change.

3. Performance and Efficiency: The prototype performed better than the conventional diesel generators with fuel cells of hydrogen being about 60 percent efficient. Consequently, in regards to energy generation per unit of fuel consumption, which results from greater optimized efficiency, hydrogen power is definitely a more cost-effective and environmentally friendly method.

4. Thorough Testing and Validation: This element proved that the Hydrogen power generator was completely safe and dependable through strict testing protocols. In future projects, the documentation of components and their choices and the plan and conduct of system integration and software creation and assembly provided an accessible, sustainable model.

5. Collaboration and Learning: It also ensured interdisciplinarity, by enrolling students from various fields and providing them with hands-on experience with STE. In turn, the success of the project can be attributed to proper management of the project and the application of an interdisciplinary approach.

Future directions:

The project has set the preliminary platform for further research and enhancement in the generation of hydrogen power. Future research could focus on several areas, such as: Some of the proposed solutions included modifying the generator’s design to make it more portable i.e. making it possible to incorporate handles onto the machine.

- Adding more sensors and solenoids into the system to improve the level of automatization, security, and the possibility of monitoring processes online.
- Custom PCBs for a better and more solid structure for the project that should fit some requirements of a specific project.

Exploring the possibilities of developing new, state-of-the-art methods for hydrogen manufacturing to reduce costs and adverse effects even more.

Concluding Words:

In conclusion, the “H2-Power Substituting Diesel Generator” project is a highly valid and promising example of how integrating hydrogen technology in the energy production schemes of the future is quite feasible. The success of the project to develop an affordable, efficient, and environment-friendly power generator shows why further support for related research must continue. Therefore, we are approaching a sustainable path in energy progress that decreases the dependence on petroleum products and reduces the adverse impacts of climate change by creating hydrogen power technologies.

Thus, the development of this project is not only an important contribution to the improvement of theoretical and practical knowledge regarding the efficiency of hydrogen power but also to the creation of the model for other projects that will attempt to make the planet greener and more sustainable in the future. The direction of further research and development in the field of renewable energy sources will inevitably take into account the experience and the achievements made in this project to promote the changes for the better and to support the move towards more sustainable and friendly to the environment future.

Future Work

Author: Mika Günther

1.1. Handle for easier movement

One potential improvement is the addition of a handle to facilitate easier movement of the demonstrator. This modification would enhance the portability and usability of the demonstrator, allowing users to relocate it effortlessly.

1.2. Pressure sensor for measuring residual pressure

Integrating a pressure sensor to measure the residual pressure of the hydrogen tank is a critical enhancement. This sensor will provide real-time data on the pressure levels within the tank, ensuring safer operation and better monitoring capabilities. The data collected from the pressure sensor can be used to estimate the residual runtime of the fuel cell as well as to send notifications for refueling the hydrogen tank.

1.3. Solenoid for controlling the hydrogen flow via home assistant

Incorporating a solenoid valve to control the hydrogen flow managed through Home assistant will significantly improve automation and control. This upgrade will also enhance the efficiency and safety of the system.

1.4. “Hacking” the fuel cell controller for data processing

Another exciting avenue for future work is "hacking" the fuel cell controller to access its sensor data. By doing so, we can process and utilize this data within Home assistant. This will provide comprehensive insights into the fuel cell's performance, enabling more sophisticated monitoring and control. Access to detailed sensor data will also aid in diagnosing issues and optimizing the fuel cell's operation.

1.5. Manufacturing custom PCBs for the sensor assemblies

Manufacturing custom-printed circuit boards (PCBs) for the ESP modules is a significant step forward. Custom PCBs will allow for a more compact, reliable, and efficient design, tailored specifically to the project's needs. This customization can lead to improved performance, reduced interference, and a cleaner overall setup.

Additionally, designing our own PCBs offers the opportunity to integrate all necessary components into a single board, further enhancing the project's robustness and ease of assembly.

1.6. Conclusion

Each of these future enhancements will contribute to the overall improvement and sophistication of the project, ensuring it remains cutting-edge and highly functional. Implementing these features will not only enhance the current capabilities but also pave the way for further innovations and applications.

References

1. Bürkert Fluid Control Systems. “Solenoid Valve Technology.” Solenoid valves control the flow of hydrogen gas in the system activated electronically for precise control. 2016 https://www.burkert.com/. Accessed 21 May 2024.
2. Figaro Engineering Inc. “Hydrogen Gas Sensors.” Sensors used to detect the presence and concentration of hydrogen gas ensuring safety and monitoring performance 2017 https://www.figarosensor.com/. Accessed 21 May 2024.
3. Horizonfuelcell. “Powering The Hydrogen Age.” Hydrogen Fuel Cell 2019 https://www.horizonfuelcell.com/. Accessed 21 May 2024.
4. Parker Hannifin Corp. “Hydrogen Service Pressure Regulators.” Parker Distributor 2018 https://www.parker.com/. Accessed 21 May 2024.
5. PYLONTECH. “Pylon
6. tech. US3000C Lithium Battery.” Lithium Iron Phosphate (LiFePO4) battery with high safety, long cycle life, and efficient energy storage. 2017 https://www.pylontech.com.cn/index.aspx. Accessed 21 May 2024.
7. Raspberry Pi Foundation. “Raspberry Pi 4 Model B Specifications.” A versatile single-board computer used for monitoring, control, and data processing in the hydrogen generator system 2017 https://www.raspberrypi.org/. Accessed 21 May 2024.
8. Rexroth Bosch. “Aluminum Structural Framing Systems.” The Power of Learning: Unlocking Your Potential for Success 2018 https://www.boschrexroth.com/en/dc/. Accessed 21 May 2024.
9. Sandia National Laboratories. “Metal Hydride Hydrogen Storage.” Sandia National Laboratories. 2020 https://www.sandia.gov/. Accessed 21 May 2024.

10. Victron Energy. “MultiPlus-II GX Datasheet.” A multi-functional inverter/charger that manages power supply and battery charging in off-grid and on-grid applications 2021 https://www.victronenergy.com/.
11. https://stackoverflow.com/questions/12894029/arduino-fail-safe-mechanism
12. https://www.reddit.com/r/arduino/comments/2n83jg/is_it_possible_to_have_two_arduinos connected_to/
13. Raspberry Pi and ESP32-Based Smart Sensor Network for IoT Platform Integration and Real-Time Environmental Data Monitoring - Theseus
14. PEREIRA, Gilroy P., Mohamed Z. CHAARI und Fawwad DAROGE, 2023. IoT-Enabled Smart Drip Irrigation System Using ESP32. *Internet of Things (IoT)*. 1 September 2023. Bd. 4, Nr. 3, S. 221–243. DOI 10.3390/iot4030012
15. CONRAD, D. und R. KAULBARS, 1995. Druckabhängigkeit der Explosionsgrenzen von Wasserstoff/Studies on pressure dependence of the explosion limits of hydrogen. Chemie Ingenieur Technik. 1995. Bd. 67, Nr. 2, S. 185–188. DOI 10.1002/cite.330670210
16. MSEd, K. C. (2024, 19. Juni). Is Democratic Leadership the Best Style of Leadership? Verywell Mind. https://www.verywellmind.com/what-is-democratic-leadership-2795315#:~:text=Democratic%20leadership%2C%20also%20known%20as,to%20schools%20to%20the%20government

Declaration of Authorship

We, the project team, hereby declare that we have prepared the present report independently and without unauthorized aids. Information taken from other works or sources, either verbatim or in essence, has been marked and provided with exact source references. Sentences or parts of sentences that have been adopted verbatim are indicated as quotations. The present work has not been submitted for examination anywhere else and has not been published either in whole or in part. Until the results are published by the examination committee, we will retain a copy of this study work and make it accessible if necessary.

Kamp-Lintfort, 29.07.2024