

FACULTY OF ENGINEERING

DYNAMIC IRRIGATION MANAGEMENT

EERI 427

Submitted to: Mr. A.J Alberts

Date: 10/11/2023

Author: Gerrit Coetzer

Student number: 33729859

Abstract

Advanced irrigation management systems are too expensive for the small to medium farmer. Less expensive solutions exist; however, these solutions lack modularity and user-friendliness. Therefore, a gap is created in the agricultural sector for an affordable and dynamic irrigation management system. This project entailed developing a solution to fill this gap.

To ensure the modularity of this solution, protective circuits were incorporated for both the analogue inputs and digital outputs, serving as essential safeguards against potential system damage. This design accommodates 4 analogue inputs and 8 digital outputs, offering users extensive connectivity options for increased flexibility.

A user-friendly web interface, accessible via WiFi communication, enhances efficiency with a communication range extending up to 15 meters. Real-time sensor data is presented to users through this interface, implemented using the WebSocket communication protocol. This system empowers users to define specific analogue thresholds triggering corresponding outputs. Additionally, a comprehensive time scheduling feature is provided for user customization.

With a unit cost of R 1047.19, the anticipated retail price is expected to be in the range of R 2000.00 to R 2500.00. Currently, there are no products available in the agricultural market, at this price point, that provide comparable levels of modularity and efficiency.

Table of Contents

Abstract.....	1
List of figures	5
List of tables.....	7
Declaration of originality.....	8
Chapter 1: Project proposal	9
1.1. Purpose of document.....	9
1.2. Background.....	9
1.2.1 The importance of irrigation in the agricultural sector	9
1.2.2 The challenges of traditional irrigation methods	9
1.2.3 The benefits of automatic irrigation systems	9
1.2.4 Existing irrigation management systems	10
1.3. Problem statement.....	10
1.4. Requirements	10
1.5. Scope definition	10
1.6. Deliverables.....	11
1.7. System context diagram.....	11
1.8. High-level design	11
Chapter 2: Literature study	13
2.1 Introduction.....	13
2.2 User interface	13
2.3 Communication protocol	13
2.4 Microcontroller	14
2.4.1 Built-in versus additional module.....	14
2.4.2 Architecture.....	14
2.4.3 ESP microcontroller	15
2.5 Analogue protection.....	17
2.5.1 Zener diodes.....	17
2.5.2 Filter	17
2.6 Digital output pin protection.....	18
2.6.1 Snubber diode	18
2.6.2 Transistor.....	19
Chapter 3: Preliminary design.....	20
3.1 Introduction.....	20
3.2 Physical architecture	20
3.2.1 High-level physical architecture.....	20

3.2.2	Low-level physical architecture	21
3.3	Functional flow diagram	23
3.3.1	High-level functional flow diagram.....	23
3.3.2	Low-level functional flow diagram	24
3.4	Analysis of alternatives	25
3.5	Firmware flowchart.....	27
Chapter 4: Detail design		29
4.1	Introduction.....	29
4.2	Hardware design.....	29
4.2.1	Power supply	29
4.2.2	Analogue inputs	33
4.2.3	Digital outputs	37
4.2.4	WiFi activation	39
4.2.5	Microcontroller	40
4.2.6	PCB design.....	40
4.3	Firmware design	42
4.4	Software design	45
4.4.1	Analogue Inputs.....	45
4.4.2	Digital Outputs	45
4.4.3	Schedule.....	45
Chapter 5: Implementation.....		46
5.1	Introduction.....	46
5.2	Hardware	46
5.3	Firmware and software	47
5.3.1	Analog inputs	47
5.3.2	Schedule events	48
Chapter 6: Testing		50
6.1	Introduction.....	50
6.2	Voltage regulator	50
6.2.1	Test	50
6.2.2	Method.....	50
6.2.3	Results.....	50
6.3	Digital outputs	52
6.3.1	Test	52
6.3.2	Method.....	52
6.3.3	Results.....	52
6.4	Analogue inputs	53

6.4.1	Test	53
6.4.2	Method.....	53
6.4.3	Results.....	53
6.5	Firmware and software	59
Chapter 7: Improvements and conclusion	60
7.1	Improvements	60
7.2	Conclusion	61
References	62
Appendix A – Project management	64
Risk management	64
Project schedule	64
Second semester schedule	64
Appendix B – Website states	66
Appendix C – Website UI design	68
Navigation menu	68
Analogue inputs	68
Digital outputs	69
Schedule.....	70
Communication	71
Information.....	72
Appendix D – Website implementation	73
Navigation menu	73
Analogue inputs	74
Digital outputs	76
Schedule.....	77
Communication	79
Information.....	80
Appendix E – Bill of Materials (BOM)	81
Hardware	81
Appendix F – Test signals for RC low-pass filter	82

List of figures

Figure 1: System Context Diagram	11
Figure 2: High-Level Architecture	12
Figure 3: RC Filter Circuit Configuration [21]	18
Figure 4: Snubber Diode Circuit Configuration [22]	19
Figure 5: High-Level Physical Architecture	20
Figure 6: Low-level Physical Architecture	21
Figure 7: Microcontroller PA	22
Figure 8: High Level FFD	23
Figure 9: WiFi On FFD Context	24
Figure 10: Setup FFD Context	24
Figure 11: Generate Digital Outputs FFD Context	25
Figure 12: Utility Function for Cost Criterion	25
Figure 13: Utility Function for Risk Criterion	26
Figure 14: Utility Function for GPIO Pins	26
Figure 15: Firmware Flowchart	28
Figure 16: Step-down Regulator Simulation Circuit	31
Figure 17: Output Response for 24V Input	32
Figure 18: Output Response for 12V Input	32
Figure 19: Output Response for 9V Input	32
Figure 20: Step-down Regulator Circuit Design	33
Figure 21: Analogue Protection Simulation	35
Figure 22: Analogue Protection Common Signal	35
Figure 23: Analogue Protection Overvoltage	36
Figure 24: Analogue Protection Without Scaling	36
Figure 25: Analogue Protection With Scaling	36
Figure 26: Analogue Protection Frequency Analysis	37
Figure 27: Analogue Protection Schematic	37
Figure 28: Digital Output Schematic	39
Figure 29: WiFi Activation Schematic	40
Figure 30: Microcontroller Schematic	40
Figure 31: PCB Design Top Layer	41
Figure 32: PCB Design Bottom Layer	41
Figure 33: 3D Model of PCB Design	42
Figure 34: Setup State	42
Figure 35: Main Loop State	43
Figure 36: WebSocket Event Handler State	44
Figure 37: Serve Pages State	44
Figure 38: Grounded Digital Output Circuit	47
Figure 39: Start-Up with No-Load	50
Figure 40: Start-Up with Load	51
Figure 41: Start-Up with Transformer	51
Figure 42: Back-EMF in Digital Output Circuit	52
Figure 43: Measured Analogue Voltage	53
Figure 44: Measured Analogue Voltage with Overvoltage Signal	53
Figure 45: Measured Analogue Signal Without Top Zener	54
Figure 46: Measured Signal with Overvoltage Signal Without Top Zener Diode	54
Figure 47: Measured Signal Without Zener (Mapped)	55
Figure 48: Measured Signal With Top Zener (Mapped)	55
Figure 49: Bit Values (No Mapping)	56

Figure 50: Measured Signal Accuracy Mapping	56
Figure 51: Measured Signal Moving Average Applied	57
Figure 52: Triangular Wave Response	57
Figure 53: Block Wave Response	58
Figure 54: Fourier Transform of 2 kHz Signal.....	58
Figure 55: Fourier Transform of 20 kHz Signal.....	59
Figure 56: Fourier Transform of 200 kHz Signal.....	59
Figure 57: Project Time Schedule	64
Figure 58: Input and Output Page States	66
Figure 59: Communication and Information Page states	67
Figure 60: Schedule Page States.....	67
Figure 61: First Menu Design	68
Figure 62: First Analogue Input Threshold Creation Design	68
Figure 63: First Analogue Input Threshold Submission Design	69
Figure 64: First Digital Output Design	69
Figure 65: First Digital Output Switching Design	70
Figure 66: First Schedule Event Creation Design.....	70
Figure 67: First Schedule Event Submission Design	71
Figure 68: First Communication Page Design	71
Figure 69: First Information Page Design	72
Figure 70: Final Menu Design	73
Figure 71: Final Analogue Input Channel View Design.....	74
Figure 72: Final Analogue Input Threshold Submission Design	75
Figure 73: Final Digital Outputs Page Design.....	76
Figure 74: Final Schedule Event Creation Design	77
Figure 75: Final Schedule Event Submission Design	78
Figure 76: Final Communication Page Design	79
Figure 77: Final Information Page Design	80
Figure 78: Hardware Bill of Materials	81
Figure 79: Filter 2 kHz Test Signal	82
Figure 80: Filter 20 kHz Test Signal	83
Figure 81: Filter 200 kHz Test Signal	83

List of tables

Table 1: Discussion of Functional Units.....	12
Table 2: High-Level PA Functional Unit Descriptions	20
Table 3: High-Level PA Interface Descriptions	21
Table 4: Low-Level PA Functional Unit Description	22
Table 5: Low-Level PA Interface Description.....	22
Table 6: Microcontroller Functional Unit Description	22
Table 7: High-Level FDD Descriptions	23
Table 8: WiFi On FFD Descriptions.....	24
Table 9: Setup FFD Descriptions	24
Table 10: Generate Digital Outputs FFD Descriptions.....	25
Table 11: ESP32-C3-DevKitM-1 MCDM.....	27
Table 12: ESP8684-DevKitM-1	27
Table 13: ESP32-S3-DevKitC-1	27
Table 14: Risk Management	64
Table 15: Second Semester Time Schedule	65

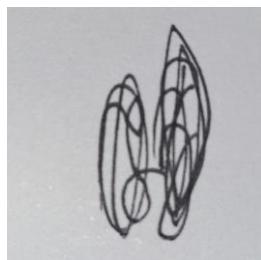
Declaration of originality

I, Gerrit Coetzer, with student number 33729859, declare the following:

This report is my own work, original work. I further declare that:

1. No part of it has been copied from another person;
2. I didn't work with another person on this project and report;
3. I acknowledged all consulted sources in the text and submitted a bibliography;
4. I have indicated all sections where I have used AI-enabled technology;
5. Parts without references are my own ideas, arguments and conclusions.

Signature:



Date: 2023/11/10

Turnitin Percentage: 8 %

Chapter 1: Project proposal

1.1. Purpose of document

This report proposes a dynamic, affordable, and user-friendly irrigation management system in the agricultural sector.

1.2. Background

1.2.1 The importance of irrigation in the agricultural sector

Irrigation is essential to crop growth as it provides plants with the necessary water they need to survive and thrive. In areas where rainfall is insufficient or irregular, irrigation is essential for crop production. Without adequate water, crops may wilt, suffer from stunted growth, or even die [1]. Irrigation ensures that crops receive the necessary water they need to grow, regardless of the weather conditions.

However, it is not enough for farmers to simply provide their crops with water. It is important for farmers to manage water resources effectively to ensure that they are not wasting water, and that the water is being used efficiently. Water is a finite resource, and in many areas, water scarcity is a growing concern. As such, it is important for farmers to use water resources as efficiently as possible to ensure long-term sustainability.

Effective water management in irrigation can lead to a variety of benefits for farmers. It can help increase crop yield and quality, reduce the risk of crop failure due to drought, and decrease the costs of irrigation [2]. Effective water management can also help farmers conserve water resources, which is increasingly important in the face of climate change and growing global population [3].

1.2.2 The challenges of traditional irrigation methods

Overuse of water resources is a major challenge with traditional irrigation methods, particularly in regions with water scarcity [1]. Many farmers tend to use more water than is necessary, leading to water wastage and potential water shortages. In addition, inconsistent water application and poor timing of irrigation can result in uneven crop growth and yield reductions. Traditional irrigation methods often rely on manual labor, which can lead to human error and further aggravate these challenges.

To address these challenges, farmers are increasingly turning to automatic irrigation systems. These methods provide more precise water application and help conserve water resources, while also reducing the cost of irrigation.

1.2.3 The benefits of automatic irrigation systems

Automation can help farmers improve the quality and quantity of their crops by providing consistent and precise application of water. This can lead to better yields and higher-quality produce [2].

Automated irrigation systems can provide water in real-time based on various monitored conditions. This implementation ensures that the water distribution is done efficiently and ultimately results in lower water consumption [3]. Studies were done on automated irrigation technologies in Dookie, Egypt, which resulted in water conservation of up to 38% over that of conventional technologies [4].

1.2.4 Existing irrigation management systems

Netafim offers a wide variety of products for the purpose of irrigation management [5]. Netafim sends a team out to analyze and inspect the situation on the farm. An irrigation system is then designed and implemented. This equipment and service come at a high cost. The high cost is a major barrier to adopting modern irrigation management systems.

Centurion Systems provide a modular and inexpensive system called the G-Ultra. However, this system cannot monitor certain conditions and autonomously manage the irrigation process according to the monitored conditions [6]. This system can only execute instructions on demand from the user.

Agrico offers the Agrico Web Control system which can perform real-time monitoring and automation of irrigation [7]. This system is only applicable to center pivot irrigation systems. The irrigation company, Reinke, offer a similar solution with the same downfall. Reinke offers the RPM Preferred system which is a precision irrigation management system [8]. This system is also only applicable to center-pivot irrigation.

The disadvantages of the irrigation systems mentioned above create a gap in the agricultural sector. Small-to-medium scale farmers have the need for an affordable irrigation management system. A modular system that can manage the irrigation process autonomously will have great benefits for the small-to-medium farmer.

1.3. Problem statement

Currently, advanced irrigation management systems exist, but they are too expensive for small farmers. Therefore, there is a gap in the agricultural sector for an affordable and user-friendly irrigation management system. A need exists for an open-source platform that can be easily configured and interfaced with by a small farmer with basic technical skills. The system should be adaptable to any pump, reservoir, valve, and sensor. The user should be able to input an irrigation schedule and specific conditions that must be monitored by the system. The system must then be able to manage the reservoir levels and irrigation autonomously based on the monitored conditions.

1.4. Requirements

- The proposed solution must be able to adapt to any external component.
- The user must be able to input an irrigation schedule.
- The hardware and software must implement a Real Time Clock (RTC).
- The user must also be able to input conditions that must be monitored.
- The device must be able to manage the irrigation process autonomously according to the schedule and monitored conditions.
- The user must be able to interface with the device.
- The device should take any external component related to irrigation as input or output.
- The device must be easily configurable.
- The device must be affordable.
- The device must have low current consumption.
- The device must have the capability to be powered by a Direct Current (DC) battery.
- The hardware design, firmware, and software must be open source.

1.5. Scope definition

This project will focus on the design and development of an affordable irrigation management system within the agricultural sector. The solution will be able to adapt to any external component related to irrigation. These components will be managed autonomously.

The device will also have the ability to be used outside of the irrigation field. However, this is out of scope for this project. This device will use local networks for communication and will not connect to the internet.

1.6. Deliverables

This project will deliver a device that can manage any irrigation system in the agricultural sector. This device will be able to adapt to any sensor, valve, pump, and reservoir. The device will manage these components autonomously.

A User Control Interface (UCI) will be developed for mobile devices. This will allow the user to interact with the system and configure the system.

1.7. System context diagram

Figure 1 gives the System Context Diagram (SCD) of the proposed solution. This SCD illustrates the interactions between the system of interest and the containing environment.

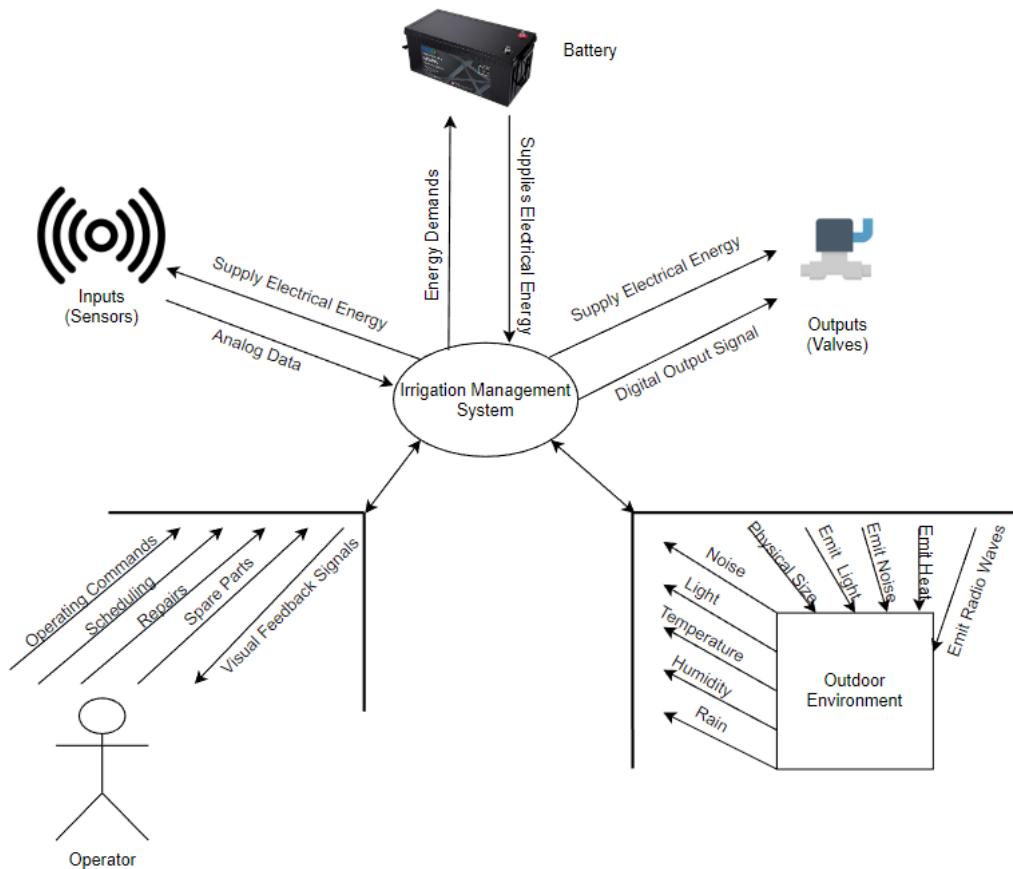


FIGURE 1: SYSTEM CONTEXT DIAGRAM

1.8. High-level design

Figure 2 gives a high-level physical architecture for the irrigation management system. This is a high-level design and may be subjected to change.

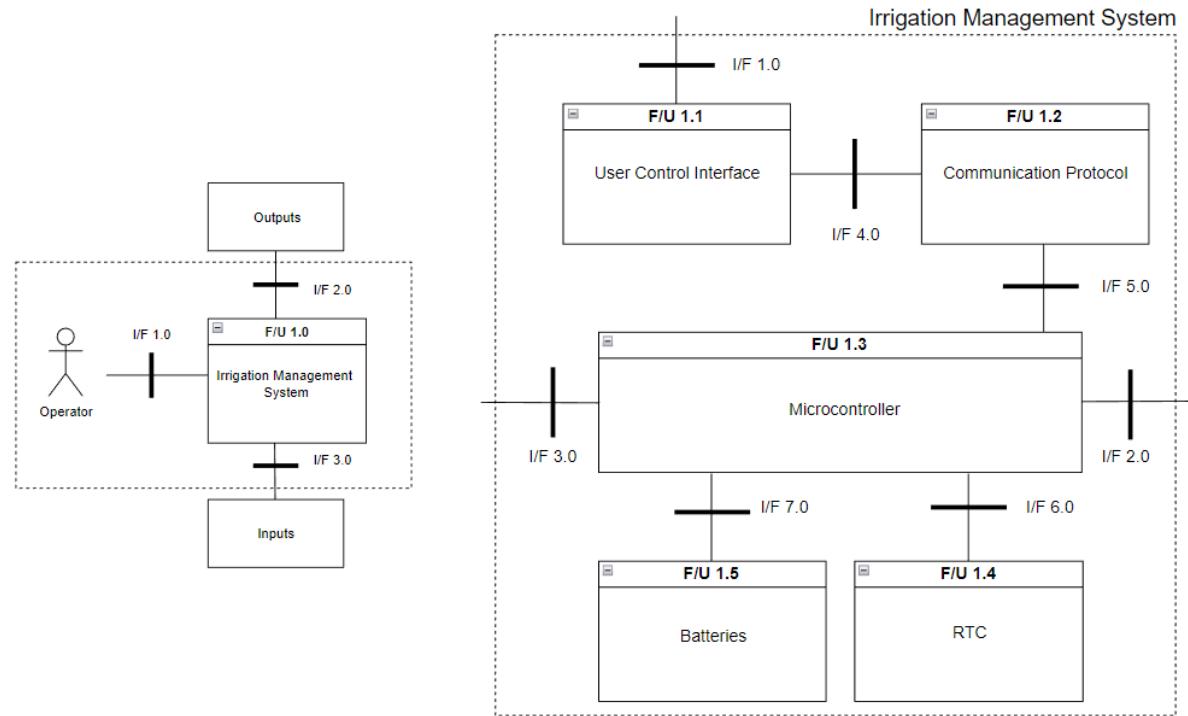


FIGURE 2: HIGH-LEVEL ARCHITECTURE

The functional units within Functional Unit 1.0 are briefly discussed in Table 1.

TABLE 1: DISCUSSION OF FUNCTIONAL UNITS

Functional Unit	Discussion
F/U 1.1 – User Control Interface	The user control interface will enable the user to interact with the system.
F/U 1.2 – Communication Protocol	The communication protocol will be responsible for the transfer of instructions and information between the user control interface and the microcontroller.
F/U 1.3 – Microcontroller	The microcontroller will process the input data and the user instructions. This unit will manage the output signals as well.
F/U 1.4 – RTC	The Real-Time Clock (RTC) will serve as a timer to ensure that every instruction is executed according to the schedule.
F/U 1.5 – Batteries	The batteries will supply the main power to the system.

Chapter 2: Literature study

2.1 Introduction

This chapter will discuss different methods for implementing the various sub-systems of the system. Specifically, the user interface, microcontroller, and communication protocol will be discussed. Additionally, the topic of a protection circuit for the analogue inputs and digital outputs will be examined.

2.2 User interface

The user interface is a crucial component that facilitates user interaction with the system. This allows the user to specify measurements and conditions. The measurements can include values from water level sensors, temperature and humidity sensors, soil moisture sensors etc. The user interface will also enable the user to specify conditions under which a certain action must be performed by the system. For instance, if the soil moisture level falls below a certain threshold, the system must activate the sprinkler system.

The different methods for enabling this interaction between the user and the system will be discussed in this section. It is important to address this topic first, as the choice of user interface component will impact the selection of microcontrollers.

A mobile smart device such as a smartphone or computer, along with a mobile or web application will be the most optimal choice for this application. This decision is supported by the modularity of a mobile interface. A display panel such as a Liquid Crystal Display (LCD) module lacks in modularity. An LCD module is also not user-friendly. In addition, various buttons and potentiometers, and associated input and output pins, will need to be added to the system.

There can also be assumed that the user that can afford this product, already have a smartphone in their possession. Therefore, the product will be less expensive to develop and sell without an additional display panel such as an LCD module. The microcontroller will interface wirelessly with the smartphone, making it more user-friendly.

2.3 Communication protocol

The communication protocol used to communicate between the microcontroller and the smartphone needs to be discussed next. The protocols that will be discussed is Wireless Fidelity (WiFi) and Bluetooth. These are the most common wireless communication protocols between microcontroller and smartphone. This decision will also influence the choice of microcontroller implemented.

WiFi provides a wider communication range in comparison to Bluetooth, which is more suitable for short-range communication [9]. WiFi can be more user-friendly in this context, as it allows users to establish connections and engage with the device from greater distances.

In terms of data-transfer rates, Bluetooth is constrained to lower bandwidth capabilities compared to WiFi [9]. Given the system's requirement to transmit and receive instructions and real-time data to and from the user's smartphone, WiFi emerges as the superior choice. WiFi technology excels in data-intensive applications. Its higher bandwidth capacity allows for reliable transmission of data.

WiFi consumes more current compared to Bluetooth, especially during active data transmission. Bluetooth is designed with low-current consumption in mind. Low current

consumption is essential for this application since the system will most likely be powered by a battery. Therefore, Bluetooth will be more efficient than WiFi in maximising the battery life.

When establishing communication through Bluetooth, the user interface is confined solely to mobile applications. However, opting for WiFi introduces broader possibilities by offering both mobile applications and a web interface as viable options for the user interface. A web-based interface offers versatile interaction capabilities, extending beyond mobile applications to include computer-based access.

In conclusion, WiFi will be used as the communication protocol between the microcontroller and the smartphone, along with a web-based interface. This decision is supported by the higher data transmission rate of the protocol. The problem of high-power consumption can be solved by allowing the user to switch the WiFi on and off.

2.4 Microcontroller

This section will discuss the various microcontrollers that can be implemented with this application. Each microcontroller's advantages and disadvantages will be discussed. First, a microcontroller with built-in WiFi will be compared to a microcontroller with an additional WiFi module. Finally, the different applicable development boards will be discussed.

2.4.1 Built-in versus additional module

Opting for a microcontroller with built-in WiFi means that the WiFi functionality is integrated directly into the microcontroller chip. This integration can minimise Printed Circuit Board (PCB) size and simplify the overall design by reducing the number of components. However, choosing a separate WiFi module allows for the selection of a microcontroller based on other criteria such as current consumption. This approach provides flexibility in terms of choosing the microcontroller which satisfies all the requirements.

Using a microcontroller with built-in WiFi often comes with dedicated software development kits (SDK's). This can streamline development and provide comprehensive support specifically tailored to the integrated WiFi functionality. If a separate WiFi module is chosen, the microcontroller's SDK and the module's SDK will need to be integrated within the program. This will add complexity to the development process.

A microcontroller with built-in WiFi may be less expensive than a microcontroller with an additional WiFi module. For instance, the ESP32-C3-DevKitM-1 development kit, priced at approximately R 170.00, presents a cost-efficient option [10]. Conversely, selecting an Arduino Nano at a cost of around R 300.00 would result in a higher overall expenditure [11]. This is further compounded when incorporating a widely used WiFi module like the MOD-WIFI-ESP8266, which is priced at approximately R 80.00 [12]. Thus, the ESP32 with built-in WiFi offers a more cost-effective solution.

In conclusion, a microcontroller with built-in WiFi will be the best solution. This is supported by the reduced cost and components. The product must be as small as possible to reduce the space occupied in the environment. This can be an essential requirement for users with minimal space.

2.4.2 Architecture

The ESP architecture from Espressif will be the best solution for this application. The following points provides evidence to support the previous statement.

- The ESP development kits have enough processing power to perform the required functions.

- The ESP development boards are small.
- The ESP architecture provide ultra-low power processing. The ESP also features four different sleep modes which reduces the current consumption.
- The ESP development kits are inexpensive.
- The author has experience developing embedded software with the ESP architecture. Therefore, there is low risk involved in the implementation of this architecture.
- The ESP architecture is compatible with the Arduino Integrated Development Environment (IDE). The Arduino IDE have an extensive library of functions to use with the ESP architecture. This will make the implementation of the firmware and software easier.
- The ESP architecture dominates the market when it comes to integrated WiFi microcontrollers. Therefore, a great number of resources are available regarding the implementation of the WiFi capability of the ESP.
- The ESP development boards have through-hole header pins soldered to the breakout board. This allows the board to be implemented with the other circuit components on a single Printed Circuit Board (PCB). This reduces the size of the product.

2.4.3 ESP microcontroller

The microcontrollers that will be discussed are three of the most applicable microcontrollers from Espressif.

2.4.3.1 ESP32-C3-DevKitM-1

The ESP32-C3 series of System-on-Chips (SoCs) is an ultra-low-power and highly-integrate MCU-based solution that supports 2.4 GHz Wi-Fi and Bluetooth Low Energy (Bluetooth LE) [13]. It is built with the RISC-V single-core microprocessor and features a variety of improvements and new features. The ESP32-C3 series has a wide operating temperature range of -40°C to 105°C or -40°C to 85°C, depending on the specific chip. Both ranges make the ESP32-C3 series suitable for use in a variety of environments.

The ESP32-C3-DevKitM-1 features dual-mode Wi-Fi capabilities and supports IEEE 802.11 b/g/n. Additionally, it supports Bluetooth Low Energy (BLE) 5.0. These features make it an ideal choice for wireless communication applications.

The ESP32-C3-DevKitM-1 has an Ultra-Low Power (ULP) co-processor that is designed to handle low-power tasks such as sensor readings, wake-up triggers, and real-time clock functions. The ESP32-C3-DevKitM-1's advanced power-management technology enables the microcontroller to switch between different power modes [13]. These modes include active mode, modem-sleep mode, light-sleep mode, and deep-sleep mode. This feature makes the ESP32-C3-DevKitM-1 an ideal choice for battery-powered applications that require long battery life.

The ESP32-C3-DevKitM-1 has a built-in Real-Time Clock (RTC) that provides accurate time and date information. The RTC slow clock is used for the RTC counter, RTC watchdog and low-power controller. The RTC fast clock is used for RTC peripherals and sensor controllers [13]. The RTC can be used to wake the ESP32-C3-DevKitM-1 from deep sleep mode at specific times, which can help to reduce current consumption.

The ESP32-C3-DevKitM-1 has a variety of peripherals, including Analogue to Digital Converter (ADC). The board has two 12-bit ADC's with up to 6 channels, which can be used to measure analogue signals from sensors [14].

The ESP32-C3-DevKitM-1 has 15 programmable GPIO pins, which provide flexibility for connecting to external devices and sensors [14]. The board also has multiple SPI, I2C, and

UART interfaces, which allow for communication with a variety of devices. This microcontroller also provides a flash memory of 4 MB.

The ESP32-C3-DevKitM-1 supports FreeRTOS, an open-source real-time operating system that allows for efficient multitasking and resource management. This feature makes the ESP32-C3-DevKitM-1 an ideal choice for applications that require real-time response and reliable operation.

The ESP32-C3-DevKitM-1 is cheaper than the other microcontrollers discussed in this literature study. However, one of the limitations of the ESP32-C3-DevKitM-1 is that it has a smaller number of programmable pins compared to other ESP32 boards. This can limit its use in some applications that require a larger number of GPIO pins.

The ESP32-C3-DevKitM-1 is a powerful microcontroller board that offers a variety of features and capabilities. Its operating temperature range, dual-mode Wi-Fi capabilities, ULP co-processor, built-in RTC, and peripherals make it an ideal choice for a variety of applications, especially those that require low power consumption and wireless communication.

2.4.3.2 ESP8684-DevKitM-1

The ESP8684 series of System-on-Chips (SoCs) is a highly integrated, low-power, 2.4 GHz Wi-Fi SoC solution [15]. At the core of this microcontroller is a 32-bit RISC-V Single-Core CPU that operates at up to 120 MHz. This microcontroller is a downgrade to the ESP32-C3-DevKitM-1 in terms of processing power.

The ESP8684-DevKitM-1 has an operating temperature range of -40°C to 105°C. The wide temperature range ensures that the ESP8684-DevKitM-1 can function reliably in extreme environments. The ESP8684-DevKitM-1 has a single-mode WiFi capability, supporting IEEE 802.11 b/g/n standards. This microcontroller also has Bluetooth 5 LE capability.

The ESP8684-DevKitM-1 also has an Ultra-Low Power (ULP) co-processor similar to that of the ESP32-C3-DevKitM-1. The ESP8684-DevKitM-1 has a built-in Real-Time Clock (RTC) identical to that of the ESP32-C3-DevKitM-1. The ESP8684-DevKitM-1 also has a variety of peripherals, including ADC. The board has one 12-bit ADCs with up to 5 channels [16].

The ESP8684-DevKitM-1 has 14 programmable GPIO pins. The ESP8684-DevKitM-1 is a very powerful microcontroller board that have a few advantages such as the wide temperature range.

2.4.3.3 ESP32-S3-DevKitC-1

The ESP32-S3 series is a low-power MCU-based system on a chip (SoC) with integrated 2.4 GHz WiFi and Bluetooth Low Energy (Bluetooth LE) [17]. It consists of high-performance dual-core microprocessor Xtensa 32-bit LX7 that operates at up to 240 MHz. In addition, a low power coprocessor, a WiFi baseband, a Bluetooth LE baseband, Radio Frequency (RF) module, and numerous peripherals.

The ESP32-S3-DevKitC-1 is designed to operate within a temperature range of -40°C to 150°C. This range is wider than that of the ESP8684-DevKitM-1. The microcontroller used in this project must have a wide operating temperature range because the microcontroller will be exposed to extreme weather conditions.

The ESP32-S3-DevKitC-1 also supports 802.11 b/g/n WiFi standards and can act as an access point or station, making it easy to implement the WiFi communication. This microcontroller has an on-board PCB antenna as well.

The ESP32-S3-DevKitC-1 is equipped with a sophisticated Power Management Unit (PMU) that can be customized to activate various power domains within the chip. This feature allows for an optimized trade-off between chip performance, power usage, and the time it takes for the chip to wake up [17]. The ESP32-S3-DevKitC-1 also features an Ultra-Low Power (ULP) co-processor with the same power modes as the ESP32-C3-DevKitM-1.

The ESP32-S3-DevKitC-1 also features a built-in Real-Time Clock (RTC). The ESP32-S3-DevKitC-1 provides an ADC peripheral as well. The ESP32 features two 12-bit SAR ADCs that can sample up to 20 channels. This microcontroller has 45 programmable GPIO pins which is far more than the other microcontroller options [18]. The ESP32-S3-DevKitC-1 also supports the FreeRTOS system.

In conclusion, the ESP32-S3-DevKitC-1 is a very powerful and versatile microcontroller. However, the ESP32-S3-DevKitC-1 is much more expensive than the ESP32-C3-DevKitM-1 and ESP8684-DevKitM-1. The final decision between microcontrollers will be decided with the help of the Multiple-Criteria Decision Matrices (MCDM) analysis.

2.5 Analogue protection

This section will discuss and illustrate how the analogue input pins can be protected from overvoltage. The input signal will also be filtered to ensure accurate data is received.

2.5.1 Zener diodes

Analogue inputs must be protected during the power up and power down stage [19]. Analogue inputs and the Analogue-to-Digital Converter (ADC) can be damaged by signals exceeding the input range specified in the datasheet. This can be caused by the user making faulty connections. The damage caused by exceeding the limits can create serious problems. The ADC will not function properly, or the accuracy of the readings on the pin can decrease.

Zener diodes can be used as voltage regulators or clamps. The reverse breakdown voltage of Zener diodes can be controlled by the careful selection of dimensions and impurities within the silicon [20]. Within this breakdown voltage range, the diode's V/I curve exhibits a flat slope which enables the diode to be implemented as a clamp.

The analogue input can be protected from overvoltage by connecting the cathode of the Zener diode to the input signal. The anode of the Zener diode must be connected to ground. The Zener diode will not conduct any current until the breakdown voltage of the diode is exceeded. The Zener diode will clamp the input signal to an upper limit equal to the breakdown voltage of the Zener diode plus the forward voltage of the diode. The signal will also be clamped at a lower limit of 0 V – the forward voltage of the diode.

The input signal can be clamped closer to ground and breakdown voltage of the diode through the following method. The cathode of the Zener diode must be connected to a constant DC supply equal to the desired upper limit. The signal must be clamped as close as possible to the acceptable input range of the ADC pins.

A resistor in series with the Zener diodes must be implemented to ensure that the current does not exceed the current limit of the Zener diodes. The resistor will also limit the current flowing into the analogue input pin. This resistor will be implemented with a series Resistor Capacitor (RC) filter.

2.5.2 Filter

A low-pass filter attenuates high-frequency components (noise) and passes low-frequency components [20]. It is essential to implement a low-pass filter with this application because

the external environment and connections that the user will make is unknown. The sensor can be a great distance from the system, this will induce noise onto the transmission line. Interference can also be coupled through an external connection [20]. A well-designed low-pass filter will ensure accurate data transmission to the analogue pin.

This low-pass filter can be implemented with a series RC circuit [21]. At zero frequency, the impedance of the capacitor is infinite. Therefore, the capacitor acts as an open circuit. The input and output voltage will be equal. At frequencies higher than zero, the impedance of the capacitor decreases relative to the impedance of the resistor. The source voltage then divides between the resistive impedance and the capacitive impedance. The output voltage will be less than the source voltage. At infinite frequency, the impedance of the capacitor is zero. Therefore, the capacitor acts as a short circuit. The output voltage will then be zero.

The analysis above of how the output voltage changes as a function of frequency indicates that a series RC circuit functions as a low-pass filter. Figure 3 below illustrates the common circuit configuration of a series RC circuit.

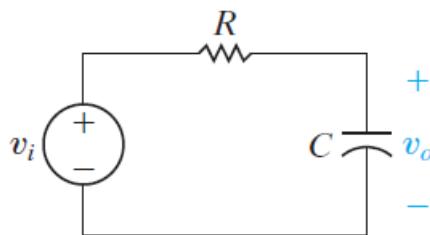


FIGURE 3: RC FILTER CIRCUIT CONFIGURATION [21]

The cut-off frequency of the filter can be calculated with Equation (1) [21]:

$$\omega_c = \frac{1}{RC}, \quad (1)$$

where ω_c represents the cut-off frequency in rad/s . The cut-off frequency in rad/s can be calculated with Equation (2):

$$\omega_c = 2\pi f, \quad (2)$$

where f is the desired cut-off frequency in Hz .

2.6 Digital output pin protection

This section will discuss and illustrate how the digital output pins can be protected from back Electro Magnetic Force (EMF) generated by the relay. Back EMF arises when a relay is deactivated, causing the magnetic field to collapse within the wire coils of the relay [22]. This collapse generates a significant voltage in the relay. As per Lenz's law, the voltage induced by a coil is influenced by the strength of the magnetic field, the number of coils, and the speed at which the field intersects the wire coils. Additionally, lengthy wire connections associated with the relay can contribute to the back EMF voltage. This phenomenon occurs due to the magnetic field dynamics.

2.6.1 Snubber diode

A diode can be connected in parallel with the relay coil to restrict the rate-of-rise of the voltage during switch-off [20]. The diode in this circuit configuration is known as a snubber diode, freewheeling diode, or flyback diode. Figure 4 illustrates a common circuit configuration of a snubber diode.

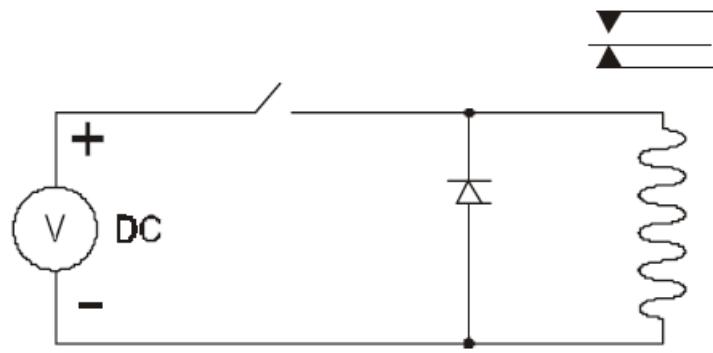


FIGURE 4: SNUBBER DIODE CIRCUIT CONFIGURATION [22]

The snubber diode is connected in parallel with the relay coil in reverse bias configuration. During normal operation, when the relay coil is energized, the snubber diode remains in a non-conductive state. However, when the relay is turned off, the collapsing magnetic field generates a reverse voltage across the coil. At this moment, the snubber diode becomes forward biased and provides a low-resistance path for the current to flow. This allows the induced energy to dissipate harmlessly through the diode, preventing voltage spikes from damaging the circuit or interfering with other components.

2.6.2 Transistor

A NPN transistor can be connected with the snubber circuit configuration. The base must be connected to the digital output pin with a resistor connected in series with the pin and the base. The collector must be connected in series with the parallel snubber circuit connection. The emitter must be connected to ground. This transistor configuration is commonly referred to as a transistor switch.

By using a transistor switch, the microcontroller can easily control the relay's operation through the digital pin. When the digital pin is driven high, it activates the transistor, allowing current to flow through the relay coil and energize it. Conversely, when the digital pin is driven low, the transistor turns off, cutting off the current flow to the relay coil and de-energizing it. This offers precise control over the relay operation.

The resistor in series with the base of the transistor acts as a current-limiting resistor. The value of the current-limiting resistor determines the base current flowing into the transistor. The resistor value can be manipulated to tailor the base current to achieve the desired saturation and cutoff characteristics of the transistor. This allows for optimal control of the relay and efficient operation of the circuit.

Chapter 3: Preliminary design

3.1 Introduction

This chapter will discuss the preliminary design for this project. The Physical Architecture (PA) and Functional Flow Diagrams (FFD) of the system will be derived. An Analysis of Alternatives (AOA) will be done as well.

3.2 Physical architecture

3.2.1 High-level physical architecture

Figure 5 gives the high-level physical architecture of the system. The high-level PA given by Figure 2 was modified to deliver PA below. Figure 2 was modified after more thought and research went into the system.

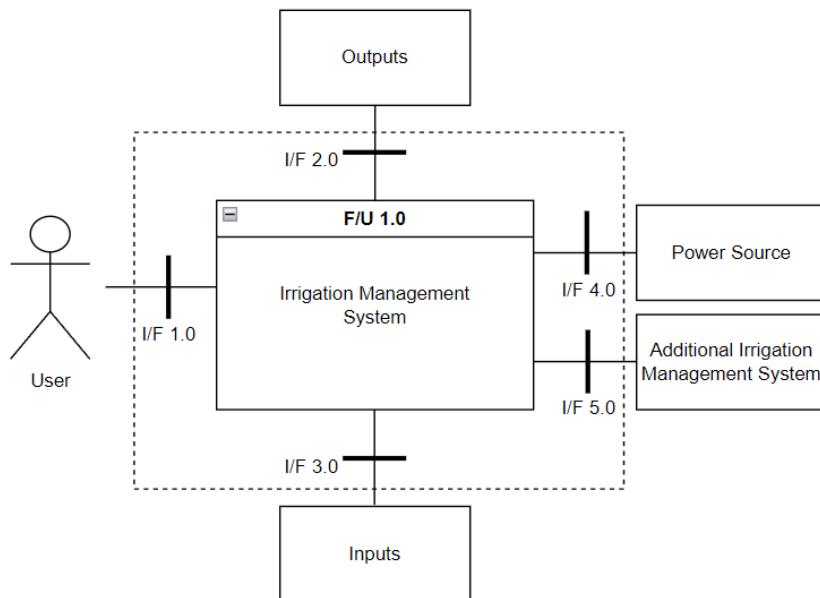


FIGURE 5: HIGH-LEVEL PHYSICAL ARCHITECTURE

The Functional Units (F/U) in Figure 5 are briefly discussed in Table 2.

TABLE 2: HIGH-LEVEL PA FUNCTIONAL UNIT DESCRIPTIONS

Functional Unit	Description
F/U 1.0 – Irrigation Management System	This block represents the main system architecture.
User	The user will interact with the system. The user will specify inputs, outputs, and RTC events.
Inputs	The inputs will be analog inputs from components such as water level sensors, soil moisture sensors, temperature sensors etc.
Outputs	The digital outputs will be connected to components such as solenoid valves.
Power Source	The power source will be an Alternating Current (AC) transformer. The power source

	can also be implemented with a DC battery with any voltage ranging from 9 V to 42 V.
Additional Irrigation Management System	An additional Irrigation Management System can be connected to the system. This will provide the user with more input and output pins.

The Interfaces (I/F) illustrated in Figure 5 are discussed in Table 3.

TABLE 3: HIGH-LEVEL PA INTERFACE DESCRIPTIONS

Interface	Description
I/F 1.0	Visual observations and physical touch.
I/F 2.0	Electrical wire.
I/F 3.0	Electrical wire.
I/F 4.0	Wider diameter electrical wire than I/F 1.0 and 2.0.
I/F 5.0	UART serial communication.

3.2.2 Low-level physical architecture

Figure 6 gives the low-level physical architecture of the system. This architecture was also derived from Figure 2.

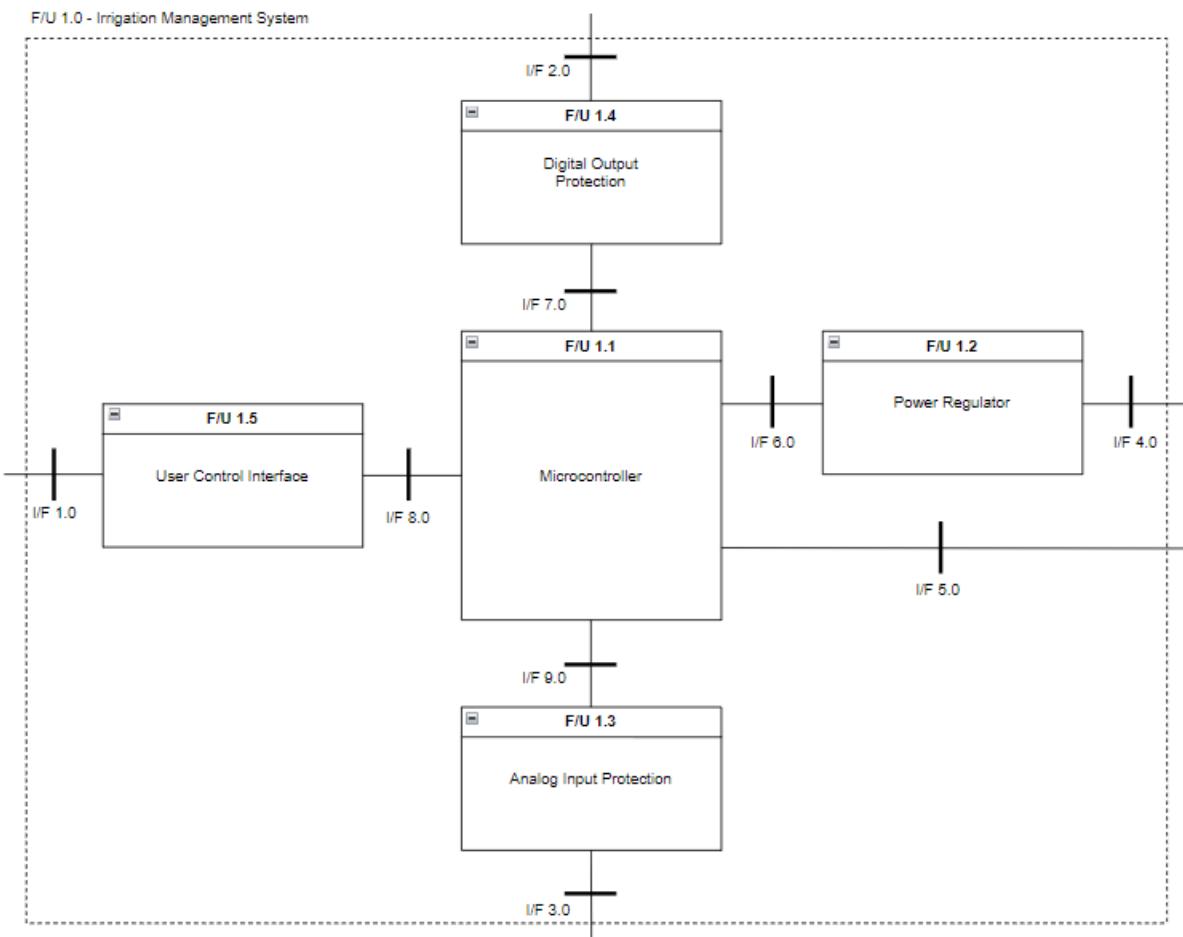


FIGURE 6: LOW-LEVEL PHYSICAL ARCHITECTURE

Table 4 gives a brief description of each F/U illustrated in Figure 6.

TABLE 4: LOW-LEVEL PA FUNCTIONAL UNIT DESCRIPTION

Functional Unit	Description
F/U 1.1 – Microcontroller	The microcontroller will implement the firmware of the system.
F/U 1.2 – Power Regulator	If the system is powered by a transformer, then the power regulator will first rectify the AC to DC. Thereafter, the power regulator will regulate 9 to 42 V DC to 5 V DC.
F/U 1.3 – Analogue Input Protection	This physical architecture protects the analogue inputs from voltage surges and filters the signal.
F/U 1.4 – Digital Output Protection	This physical architecture protects the digital output pin from back-emf generated by the relay when the relay is switched.
F/U 1.5 – User Control Interface	The UCI enables the user to interact with the system. The UCI will be implemented on a smartphone or computer.

Table 5 gives a brief description of each I/F illustrated by Figure 6.

TABLE 5: LOW-LEVEL PA INTERFACE DESCRIPTION

Interface	Description
I/F 6.0	Electrical tracks on PCB board.
I/F 7.0	Electrical tracks on PCB board.
I/F 8.0	Wireless communication via WiFi.
I/F 9.0	Electrical tracks on PCB board.

Figure 7 gives the physical architecture for the microcontroller.

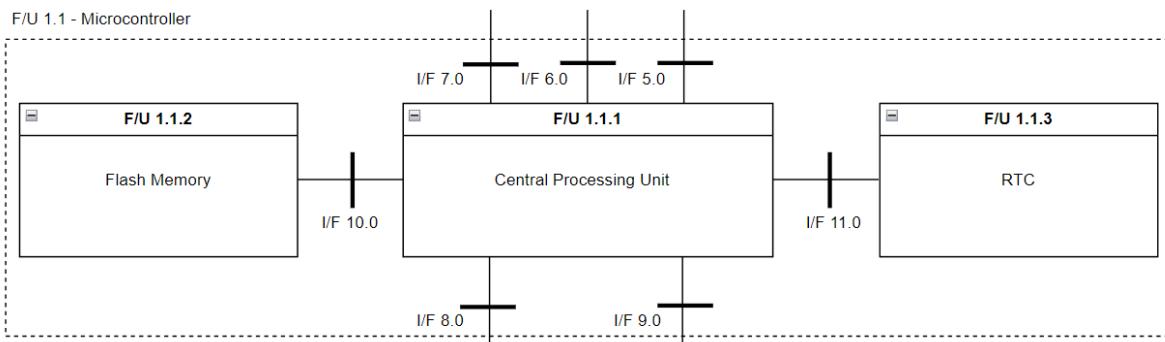


FIGURE 7: MICROCONTROLLER PA

The F/U illustrated in Figure 7 are discussed in Table 6. Interfaces 10.0 and 11.0 are electrical tracks on the PCB board.

TABLE 6: MICROCONTROLLER FUNCTIONAL UNIT DESCRIPTION

Functional Unit	Description
F/U 1.1.1 – CPU	The CPU will control the system according to the firmware.

F/U 1.1.2 – Flash Memory	The flash memory will be responsible for saving the settings made by the user. Therefore, if the power supply is turned off then the system can remember the previous settings.
F/U 1.1.3 – RTC	The RTC will be responsible for triggering events at a scheduled time and date.

3.3 Functional flow diagram

3.3.1 High-level functional flow diagram

The functional flow diagram gives an indication of the logical flow of the system. Figure 8 gives the high-level FFD of the system. Take note, a loop exit block should be added to the diagram. The system should exit the loop when the power is switched off.

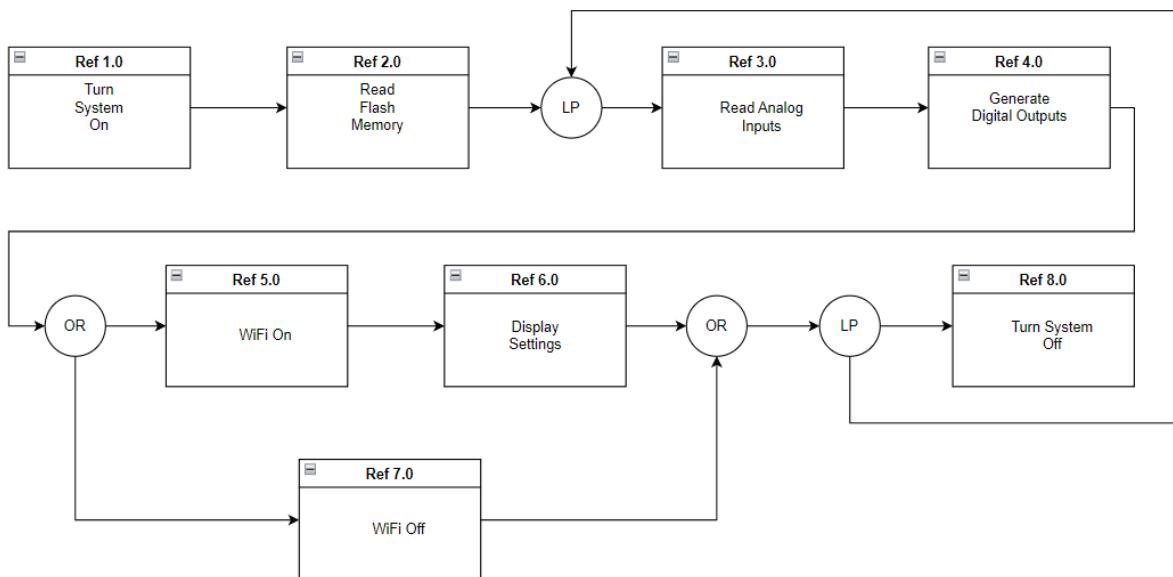


FIGURE 8: HIGH LEVEL FFD

Table 7 gives descriptions of the various blocks displayed in Figure 8.

TABLE 7: HIGH-LEVEL FDD DESCRIPTIONS

Reference Block	Description
Ref 1.0 - Turn System On & Ref 8.0 - Turn System Off	The physical device will be manually switched on or off.
Ref 2.0 - Read Flash Memory	Read the flash memory for previous settings.
Ref 3.0 - Read Analogue Inputs	Read analog input signals if any was defined.
Ref 4.0 - Generate Digital Outputs	Generate digital output signals if any events were defined.
Ref 5.0 - WiFi On & Ref 7.0 - WiFi Off	The user will be able to turn the WiFi on and off manually to save power consumption.
Ref 6.0 - Display Settings	Display the analogue input readings and schedule defined by the user.

3.3.2 Low-level functional flow diagram

The low-level FFD defines the flow of the system in more detail. Figure 9 illustrates the FFD for the WiFi On block from Figure 8.

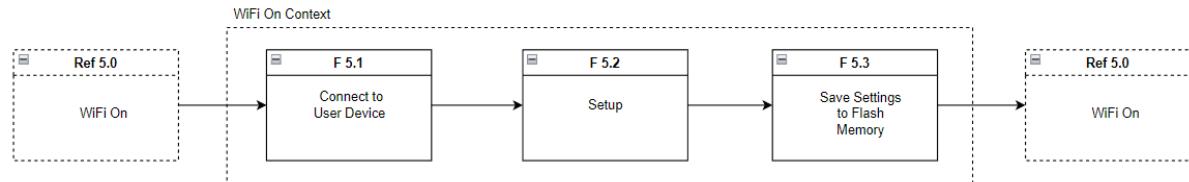


FIGURE 9: WiFi ON FFD CONTEXT

Table 8 gives a brief overview of the blocks illustrated by Figure 9.

TABLE 8: WiFi ON FFD DESCRIPTIONS

Reference Block	Description
F 5.1 – Connect to User Device	The system tries to connect to the user device via WiFi.
F 5.2 - Setup	The user can setup analog inputs and events.
F 5.3 – Save Settings to Flash Memory	After setup, the settings are permanently saved to the flash memory.

The FFD in Figure 9 will be expanded into another layer. The FFD for the Setup block is given by Figure 10.

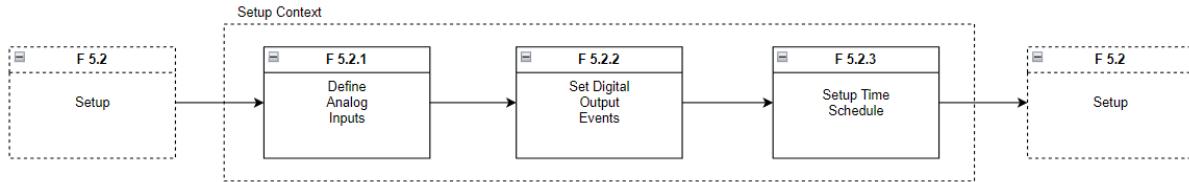


FIGURE 10: SETUP FFD CONTEXT

Table 9 gives a brief overview of the blocks illustrated by Figure 10.

TABLE 9: SETUP FFD DESCRIPTIONS

Reference Block	Description
F 5.2.1 – Define Analogue Inputs	The user will define analogue inputs on certain input pin. The user will be able to give the pin a name, scale and average the values.
F 5.2.2 – Set Digital Output Events	The user will set thresholds according to the readings from the analogue pins. The user will also define an event that will occur when this threshold is reached.
F 5.2.3 – Setup Time Schedule	The user will define events according to a time schedule.

The FFD in Figure 8 will be expanded into another layer. The FFD for the Generate Digital Outputs block is given by Figure 11. Thereafter, descriptions of each block are given by Table 10.

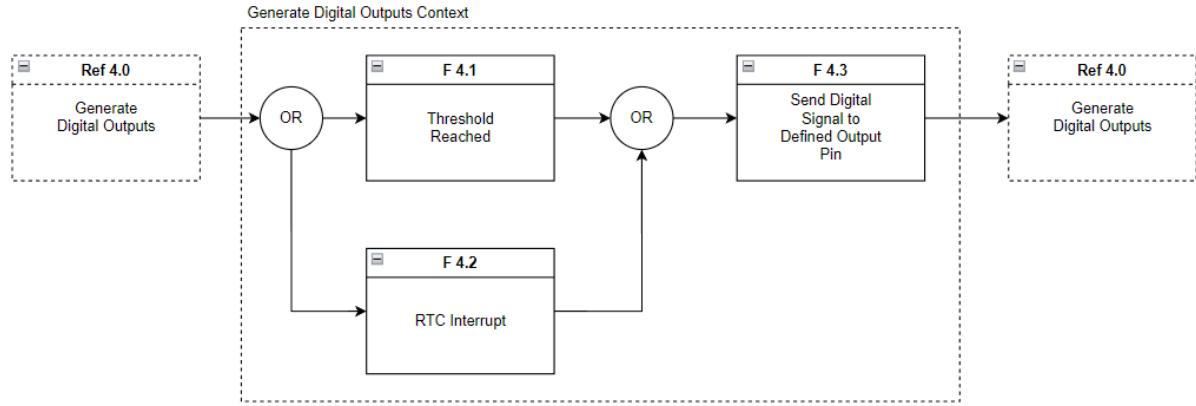


FIGURE 11: GENERATE DIGITAL OUTPUTS FFD CONTEXT

TABLE 10: GENERATE DIGITAL OUTPUTS FFD DESCRIPTIONS

Reference Block	Description
F 4.1 – Threshold Reached	If the threshold on a certain analogue pin is reached.
F 4.2 – RTC Interrupt	The RTC will generate an interrupt if an event must occur at a specific time and date.
F 4.3 – Send Digital Signal to Defined Output Pin	The system will send a digital signal through the defined output pin to the relay.

3.4 Analysis of alternatives

The MCDM design process will be implemented to decide which ESP microcontroller will be the best solution for this application. First, the criteria considered with this analysis will be discussed. The weight of each criterion will be given. Furthermore, the utility function for each criterion will be discussed.

The first criterion is cost. The cost of the microcontroller is an essential criterion since this project aims to make the product as cheap as possible. Therefore, a weight of 0.3 is assigned to the cost. The utility function used for this criterion is represented by Figure 12. The cost is within a range of R0 to R400. The optimal cost is R0, and the absolute maximum cost is R400. The cost in the middle of this range, which is R200, is the average cost.

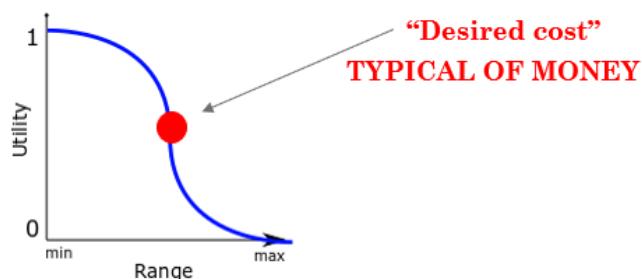


FIGURE 12: UTILITY FUNCTION FOR COST CRITERION

Risk is considered as well. The risk refers to the author's experience with developing firmware with the individual microcontrollers. Each of the microcontrollers considered can be programmed within the Arduino IDE. The author possesses extensive experience with the Arduino IDE. Therefore, the author is equally proficient with each of the microcontrollers under consideration, rendering the risk factor less significant as a criterion. A weight of 0.1 is assigned to the risk.

The utility function used for the risk criterion is given by Figure 13. The risk ranges from 1 to 5. A risk level of 1 represents familiar technology, reapply in familiar context. A risk level of 5 represents unfamiliar technology, applied in an unfamiliar context.

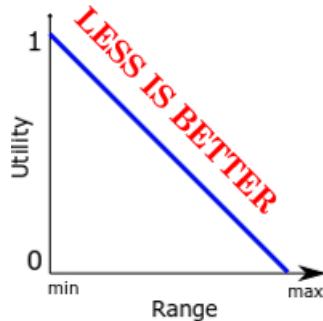


FIGURE 13: UTILITY FUNCTION FOR RISK CRITERION

The peak power consumption whilst in active mode is considered as a criterion. The power consumption is an important criterion because this project aims to minimise the power consumption. A weight of 0.2 is assigned to the power consumption criterion. The utility function used for this criterion is given by Figure 13. The range expands from 300 mA to 400 mA.

The number of programmable General-Purpose Input/Output (GPIO) is defined as a criterion. The number of pins is also important because there must be 8 digital pins and 4 ADC pins available. There must also be a UART transmit and UART receive pin available. Therefore, the minimum number of pins is 14. A weight of 0.2 is assigned to the number of pins.

The utility function is illustrated by Figure 14. The minimum will be 14 and maximum 40. A greater number of pins is desirable because there might be a need for extra pins later in the design and development process.



FIGURE 14: UTILITY FUNCTION FOR GPIO PINS

The size of the development board is considered as a criterion. The size of the breakout board must be as small as possible. This will minimise the total size of the product. In addition, the PCB board will be smaller and therefore less expensive to manufacture. A weight of 0.2 is assigned to the size of the board. The utility function for the size is given by Figure 13. The size will range from 900 mm² to 1600 mm².

The criteria matrices are given below. A utility value between 0 and 1 was given to each criterion according to the score and the utility functions discussed above. The weighed utility was calculated by multiplying the weight with the utility value. Table 11 gives the decision matrix for the ESP32-C3-DevKitM-1 microcontroller.

TABLE 11: ESP32-C3-DevKitM-1 MCDM

ESP32-C3-DevKitM-1				
Criterion	Weight	Score	Utility Value	Weighed Utility
Cost	0,3	R170,00	0,6	0,18
Risk	0,1	1	1	0,1
Power Consumption	0,2	350 mA	0,5	0,1
Number of GPIO pins	0,2	15	0,1	0,02
Size	0,2	988 mm ²	0,9	0,18
	1			0,58

Table 12 gives the decision matrix for the ESP8684-DevKitM-1 microcontroller.

TABLE 12: ESP8684-DEVKITM-1

ESP8684-DevKitM-1				
Criterion	Weight	Score	Utility Value	Weighed Utility
Cost	0,3	R200,00	0,5	0,15
Risk	0,1	1	1	0,1
Power Consumption	0,2	370 mA	0,4	0,08
Number of GPIO pins	0,2	14	0,1	0,02
Size	0,2	988 mm ²	0,9	0,18
	1			0,53

Table 13 gives the decision matrix for the ESP32-S3-DevKitC-1 microcontroller.

TABLE 13: ESP32-S3-DEVKITC-1

ESP32-S3-DevKitC-1				
Criterion	Weight	Score	Utility Value	Weighed Utility
Cost	0,3	R340,00	0,3	0,09
Risk	0,1	1	1	0,1
Power Consumption	0,2	355 mA	0,5	0,1
Number of GPIO pins	0,2	33	0,9	0,18
Size	0,2	1594 mm ²	0,2	0,04
	1			0,51

In conclusion, the ESP32-C3-DevKitM-1 microcontroller has the highest total weighed utility. Therefore, the ESP32-C3-DevKitM-1 is the optimal choice for this application. The ESP8684-DevKitM-1 is a good alternative for this application.

3.5 Firmware flowchart

Figure 15 below illustrates the flow of the firmware. This is the program that will be implemented on the microcontroller. The web-based software for the mobile device will be designed within the detail design of this project.

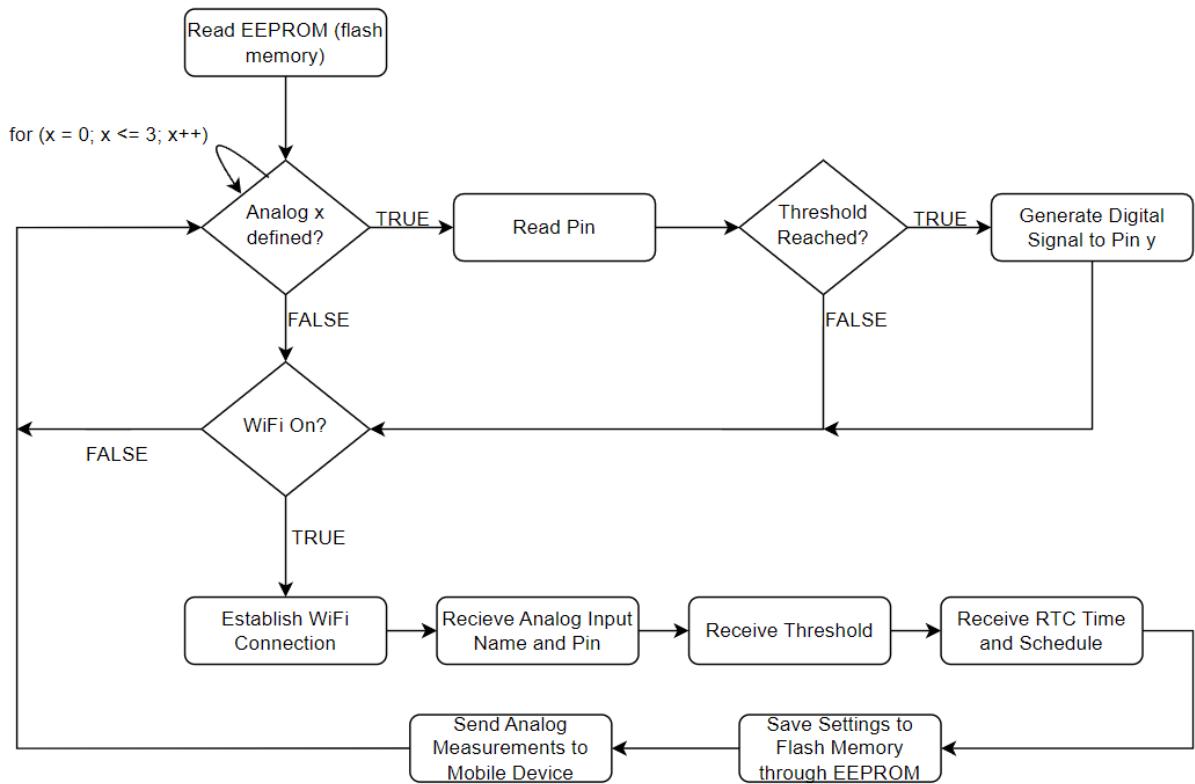


FIGURE 15: FIRMWARE FLOWCHART

Take note that the RTC interrupt is not included in the flow. The RTC will generate an interrupt when the time and date of a specific event is reached. An interrupt function outside of the main loop will be called.

Chapter 4: Detail design

4.1 Introduction

This chapter will discuss the detail design for this project. The detail design will include the hardware, software, and firmware design.

4.2 Hardware design

This section will discuss the detail design for the power supply and protection circuits. The PCB layout will be discussed as well.

4.2.1 Power supply

The LT8610 synchronous step-down regulator from Linear Technology will be implemented as the voltage regulator. The LT8610 is compact and highly efficient. This regulator is also capable of high-speed switching [23].

This regulator is chosen because of the wide input voltage range of 3.4 V to 42 V. This input range is ideal for this application. A model of the LT8610 is also available within the LTSpice simulation software. This simplifies the design process.

The regulator circuitry will be designed according to the LT8610 datasheet's guidelines. These guidelines will be adjusted for this application. The external clock synchronization input (MODE) pin will be connected to ground. This will enable low ripple burst mode operation at low output loads [23]. This mode optimizes efficiency by shutting down all circuitry, between bursts, which controls the output switching. The input supply current is reduced to 1.7 μ A.

A capacitor connected to the output tracking and soft-start (TR/SS) pin programs the output voltage ramp rate during start-up. A 1 nF capacitor will program the ramp rate to approximately 500 μ s. The determination of this capacitor value was derived through an empirical process, as the datasheet does not contain specific guidelines for calculating this capacitance.

The LT8610 employs a constant frequency Pulse-Width Modulation (PWM) architecture, offering programmable frequency adjustment ranging from 200 kHz to 2.2 MHz. This switching frequency is programmed via the connection of a resistor from the RT pin to ground. The choice of operating frequency necessitates a careful balance between factors such as efficiency, component dimensions, and input voltage span. Opting for higher frequency operation facilitates the utilization of smaller inductor and capacitor values, which can be advantageous in terms of component size. However, it comes with certain drawbacks, including reduced efficiency and a narrower input voltage range.

The resistor value is determined by the desired switching frequency. A switching frequency of 1 MHz is chosen. This switching frequency is at the center of the 200 kHz to 2.2 MHz range which makes the trade-offs equal. According to *Table 1. SW Frequency vs R_T Value* in the datasheet, a switching frequency of 1 MHz corresponds to a resistance value of 41.2 k Ω [23].

The switching frequency also determines the minimum input voltage. If the input voltage goes below this minimum voltage, then the LT8610 skips switch cycles resulting in a lower switching frequency than programmed by the RT pin. The minimum input voltage can be calculated with Equation (3):

$$V_{IN(MIN)} = \frac{V_{OUT} + V_{SW(BOT)}}{1 - (f_{SW})(t_{OFF(MIN)})} - V_{SW(BOT)} + V_{SW(TOP)}, \quad (3)$$

where $V_{IN(MIN)}$ is the minimum input voltage, and V_{OUT} is the output voltage. $V_{SW(TOP)}$ and $V_{SW(BOT)}$ are the internal switch drops. f_{SW} is the switching frequency and $t_{OFF(MIN)}$ is the minimum switch off-time.

According to the LT8160 datasheet, the minimum switch off-time at maximum load is 110 ns [23]. The internal switch drops, $V_{SW(TOP)}$ and $V_{SW(BOT)}$, are 0.3 V and 0.15 V respectively. Substituting these values into Equation (3) and calculating the minimum voltage results in:

$$V_{IN(MIN)} = 5.94 \text{ V}$$

The minimum input voltage remains below the required 9 V for this application. Therefore, a switching frequency of 1 MHz is sufficient for this application.

The *EN/UV* pin is connected to the V_{in} pin to disable the shutdown feature of the LT8610. The input of the LT8610 circuit must be bypassed with a ceramic capacitor of X7R or X5R type. This capacitor must be connected to V_{in} and placed as close as possible to the V_{in} and *PGND* pins on the PCB. A 4.7 μF ceramic capacitor is adequate to bypass the LT8610 and will easily handle the ripple current.

A 0.1 μF boost capacitor must be connected between the *BST* and *SW* pin. The *BST* pin is utilized to supply a drive voltage, surpassing the input voltage, to activate the top-side power switch. This capacitor must be placed as close as possible to the LT8610 Integrated Circuit (IC). An appropriate choice for the output inductor can be calculated with Equation (4):

$$L = \frac{V_{OUT} + V_{SW(BOT)}}{f_{SW}} , \quad (4)$$

where f_{SW} is the switching frequency in MHz. Substituting the relevant values, the resulting inductance is obtained as:

$$L = 5.15 \mu\text{H}$$

A standard inductor value of 5 μH is chosen. To mitigate the risk of overheating and ensure optimal efficiency, it is imperative to select an inductor with an RMS current rating exceeding the anticipated maximum output load for the given application. The maximum output load is assumed to be 2.5 A which is the maximum output current of the regulator. In addition, to maintain high efficiency, it is advisable to ensure that the series resistance (DCR) of the inductor does not exceed 40 m Ω . Therefore, an inductor with an RMS rating of 4 A and a DCR of 30 m Ω is chosen.

To optimize output stability and transient response, it is advisable to employ X5R or X7R type capacitors for both the output and feed-forward applications. This selection offers the benefits of reduced output ripple and enhanced transient performance. To further enhance transient response, a higher-value output capacitor is employed, and a feedforward capacitor positioned between the V_{OUT} and *FB* nodes is introduced. A 47 μF ceramic output capacitor is implemented. A 10 μF ceramic capacitor is connected between the V_{OUT} and *FB* nodes.

It is essential to decouple the *INTV_{cc}* pin by connecting it to the ground through a ceramic capacitor with a minimum capacitance of 1 μF and a low Equivalent Series Resistance (ESR). This capacitor should be placed close to the IC. The voltage from the *INTV_{cc}* pin serves as the power supply for the internal power drivers and control circuits.

The datasheet recommends that the *BIAS* pin be connected to V_{OUT} for an output voltage of 3.3 V and above. The internal regulators will draw current from the *BIAS* pin instead of V_{in} .

when *BIAS* is tied to a voltage higher than 3.1 V. Therefore, the *BIAS* pin will be connected to V_{OUT} for this application since the desired output voltage is 5 V.

The output voltage is programmed with a resistor divider circuit between the output voltage and the *FB* pin. The resistor values can be calculated with Equation (5):

$$R_1 = R_2 \left(\frac{V_{OUT}}{0.97 \text{ V}} - 1 \right) \quad (5)$$

Light load efficiency is desired. Therefore, large resistor values will be used for the resistor divider network. R_1 is chosen as 1 MΩ. 5 V is substituted into V_{OUT} and R_2 is calculated as:

$$R_2 = 240694.7891 \Omega$$

Standard resistors with 1% tolerance are recommended to maintain output voltage accuracy. Therefore, R_2 was chosen as 237 kΩ. This resistor divider network yields an output voltage of 5.05 V. The exposed *GND* pad must be soldered to the PCB to lower the thermal resistance.

The LT8610 step-down regulator is simulated with the LTSpice software. Figure 16 gives the circuit design for the step-down regulator.

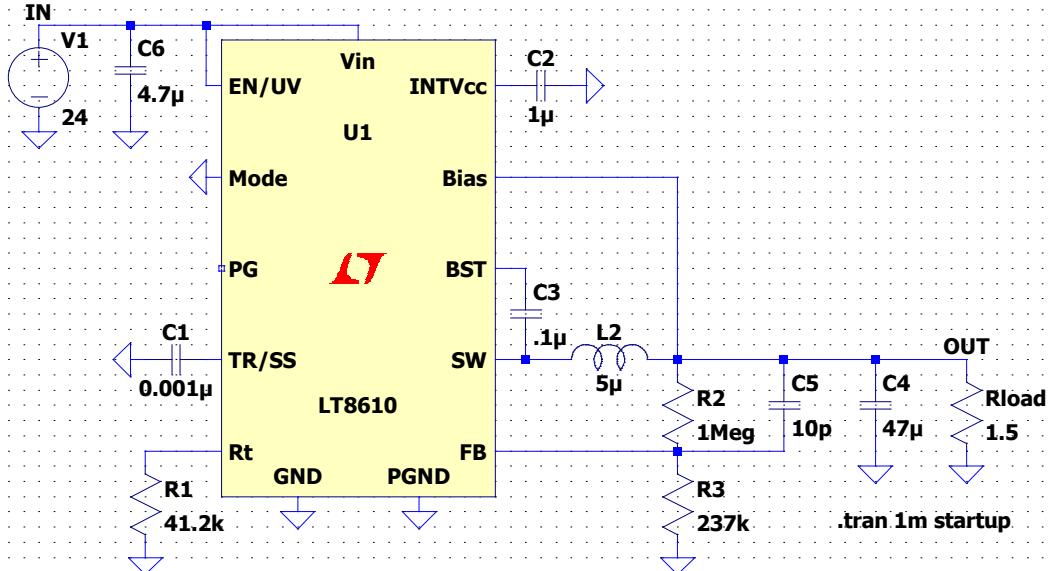


FIGURE 16: STEP-DOWN REGULATOR SIMULATION CIRCUIT

Figure 17 gives the output voltage of the step-down regulator. The simulation indicate that the output voltage remains regulated at 5 V after approximately 500 μs. Figure 18 gives the output voltage for 12 V input. Figure 19 gives the output voltage response for 9 V input. Figure 18 and Figure 19 also indicates constant 5 V regulation. Therefore, the step-down regulator will be appropriate for this application.

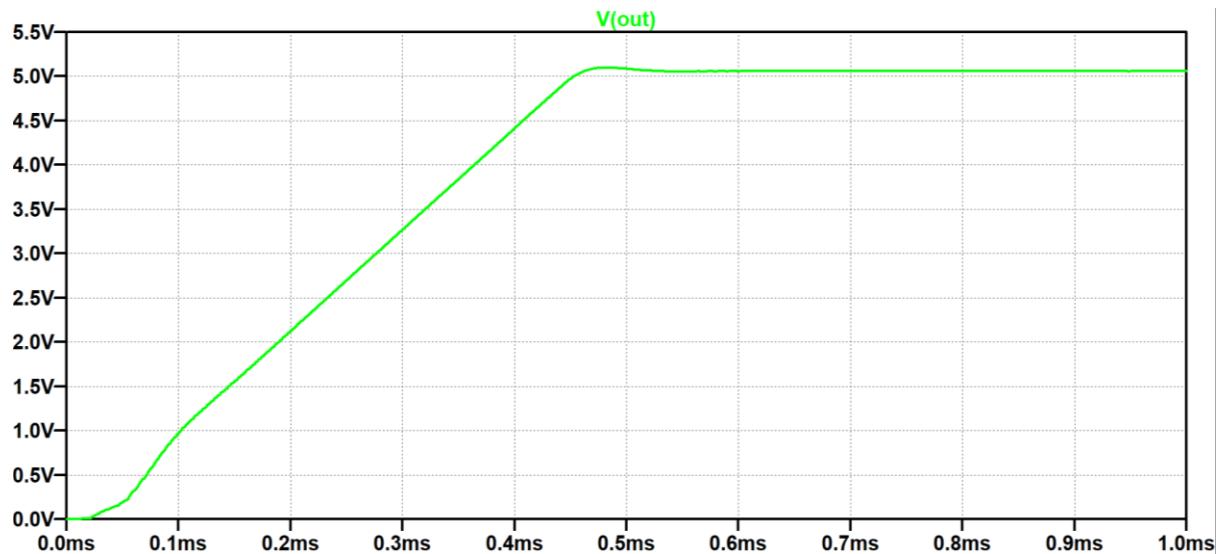


FIGURE 17: OUTPUT RESPONSE FOR 24V INPUT

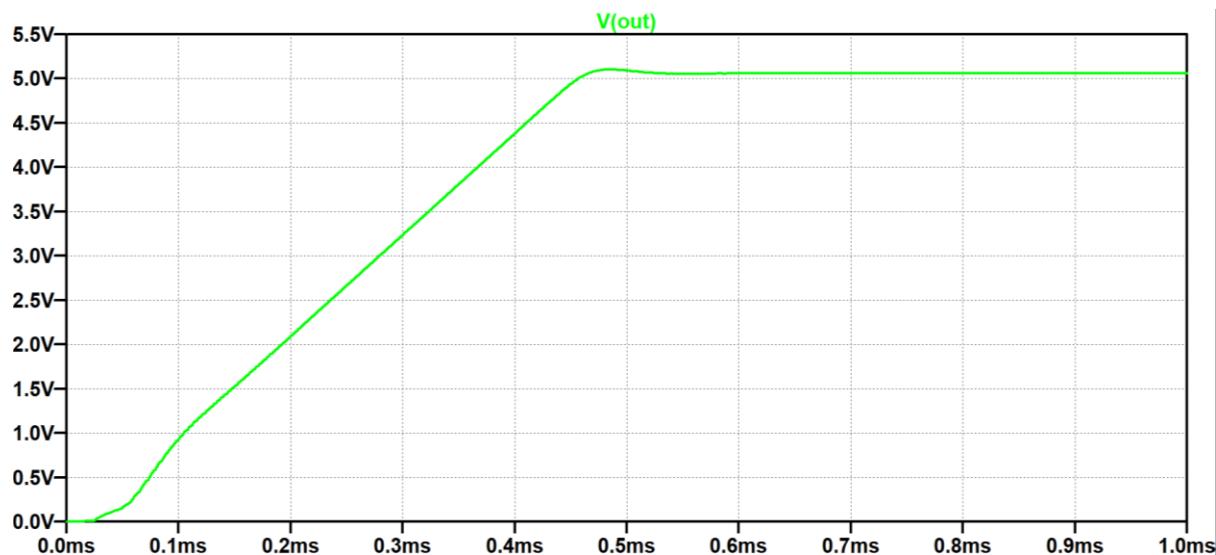


FIGURE 18: OUTPUT RESPONSE FOR 12V INPUT

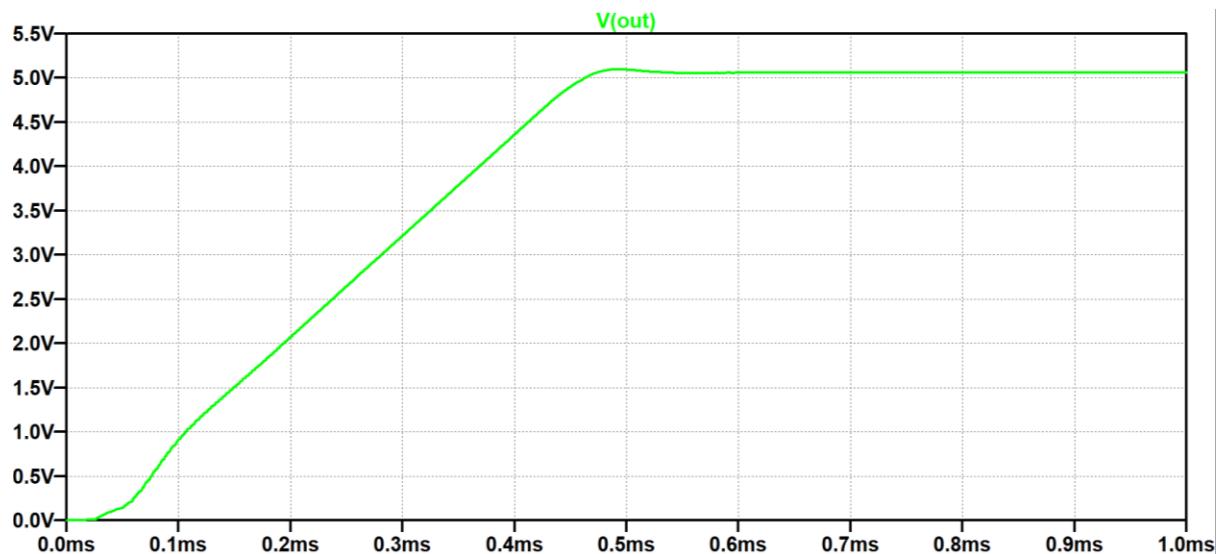


FIGURE 19: OUTPUT RESPONSE FOR 9V INPUT

Figure 20 illustrates the circuit schematic for the LT8610 step-down regulator. The $5VOUT$ power line will provide power to the microcontroller and relays. The $5VFOUT$ power line will provide power to the analogue inputs. This power line is equipped with a 1 A resettable fuse ($F2$) to limit the current that the analog inputs can draw from the voltage regulator. The circuits relative to this project will be designed in EasyEDA. EasyEDA was chosen because specific components can be selected from their extensive component library according to their part number. Furthermore, the footprints for these components will be imported automatically when converting the schematic to the PCB layout.

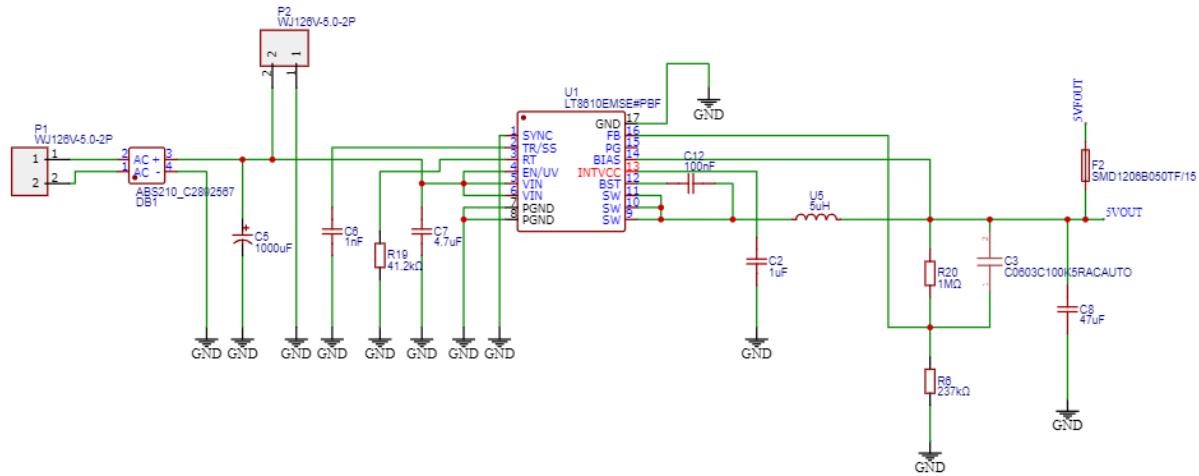


FIGURE 20: STEP-DOWN REGULATOR CIRCUIT DESIGN

A bridge rectifier Integrated Circuit (IC) is also included in the voltage regulator design. This will enable the user to use a transformer to supply power to the system. This IC is labelled as $DB1$ in Figure 20. This IC was chosen because it has a high RMS voltage rating and DC blocking voltage of 700 V and 1000 V respectively. A $1000 \mu F$ aluminium electrolytic capacitor has been incorporated following the bridge rectifier. The objective is to mitigate the ripple effects arising from the rectification process.

4.2.2 Analogue inputs

The selection of the series resistance is deliberate, with a chosen value of $4.64 \text{ k}\Omega$. This specific resistance value serves the purpose of effectively limiting the input current to safeguard both the associated components and the Analogue-to-Digital (ADC) pins.

The desired cut-off frequency of the RC filter is chosen to be 1 kHz . This cut-off frequency will allow the passage of a typical sensor's output signal, characterized by frequencies below 100 Hz . This cut-off frequency is also strategically chosen to filter out any unwanted Radio Frequency (RF) signals. Low Radio Frequency (RF) signals are present within the frequency range starting from 30 kHz and beyond. Consequently, a first-order RC low-pass filter with a 1 kHz cut-off frequency is deemed more than adequate for the intended application. With the series resistance and the cut-off frequency chosen, the capacitor value for the RC filter can be calculated with Equation (1) and Equation (2):

$$\omega_c = \frac{1}{RC}$$

$$2\pi f = \frac{1}{RC}$$

$$2\pi \times (1 \text{ kHz}) = \frac{1}{(4.64 \text{ k}\Omega) \times C}$$

$$C = 34.3 \text{ nF}$$

A standard capacitor value of 33 nF is chosen. This capacitance combined with the series resistance results in a cut-off frequency of 1.039 kHz .

Zener diodes featuring a breakdown voltage of approximately 3.3 V are required for the purpose of clamping the input signal within a range of 0 to 3.3 V . This selection ensures the preservation of the ADC pins, as this breakdown voltage aligns with the operational limits of the ESP32-C3-Mini-1, which ranges from -0.3 V to 3.6 V . Zener diodes with a breakdown voltage range spanning from 3.32 V to 3.53 V have been selected. The lower limit of 3.32 V has been chosen to closely align with the microcontroller's 3.3 V output voltage.

Incorporated into the analogue protection circuit is a voltage divider circuit. This implementation will enable the user to use sensors with an output voltage of either 3.3 V or 5 V . The user will switch between the 3.3 V circuit or the 5 V circuit with the voltage divider. The voltage divider will scale a 5 V input signal to a 3.3 V signal. This will ensure that the full range of the 5 V is measured since the threshold of the clamping circuit is designed to be at 3.3 V .

An additional resistor will be combined with the $4.64 \text{ k}\Omega$ resistor to form the voltage divider. This resistor will be connected to the input line and ground, positioned between the series resistor and filter capacitor. The appropriate resistance can be calculated with Equation (6):

$$V_{OUT} = \frac{V_{IN}R_2}{R_1 + R_2}, \quad (6)$$

where V_{IN} represents the 5 V input signal, R_1 represents the $4.64 \text{ k}\Omega$ resistor, and R_2 the additional resistor. Substituting these values into Equation (6) and 3.3 V into V_{OUT} yields:

$$3.3 = \frac{5 \times R_2}{4.64k + R_2}$$

$$R_2 = 9 \text{ k}\Omega$$

A standard resistance value of $9.09 \text{ k}\Omega$ is chosen. Figure 21 gives the analogue protection circuit as designed with the LTSpice simulation software. The Zener diodes $D1$ and $D2$ have a breakdown voltage of 5.4 V . Therefore, the output voltage of the microcontroller was simulated as 5 V . This breakdown voltage was chosen due to the unavailability of Zener diode models with a breakdown voltage of 3.3 V within LTSpice. The 5.4 V Zener diodes only represents the 3.3 V Zener diodes that will be practically implemented with the PCB. Therefore, the behaviour of the clamping circuit is simulated.

The resistor $R1$ represents the input resistance of the ADC pin. The resistor $R3$ will be disconnected if the sensor has an output voltage of 3.3 V . The user will connect this resistor to ground via a jumper block if the applicable sensor has an output voltage of 5 V .

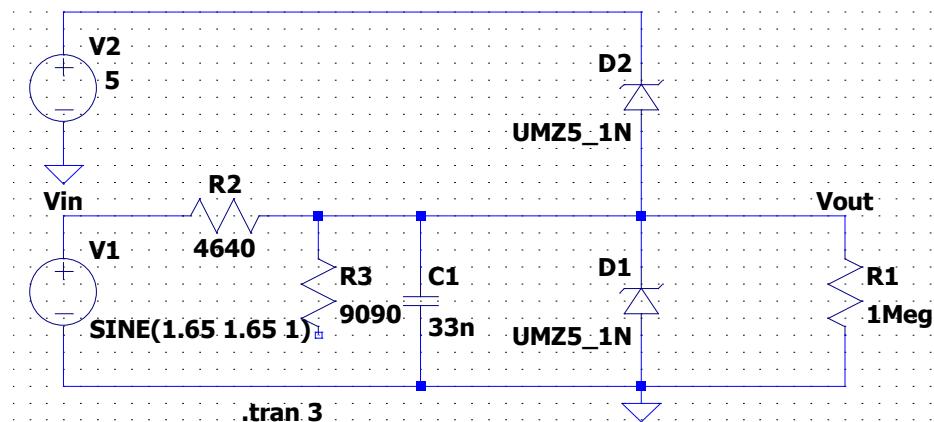


FIGURE 21: ANALOGUE PROTECTION SIMULATION

The voltage response is simulated using an input sine wave characterized by an amplitude of 1.65 V and an offset of 1.65 V. This simulates a typical signal from a sensor ranging from 0 to 3.3 V. Figure 22 gives the result of this simulation. The V_{OUT} signal represents the signal measured by the microcontroller. The V_{IN} signal represents the input signal from the sensor.

Figure 22 illustrates that the signal measured by the ADC exhibits a gain of 300 mV at an input voltage of 0 V. While this gain is deemed negligible, it is acknowledged and can be compensated for programmatically. This region is also non-linear. Therefore, the measured data can be mapped to the correct values with acceptable accuracy.

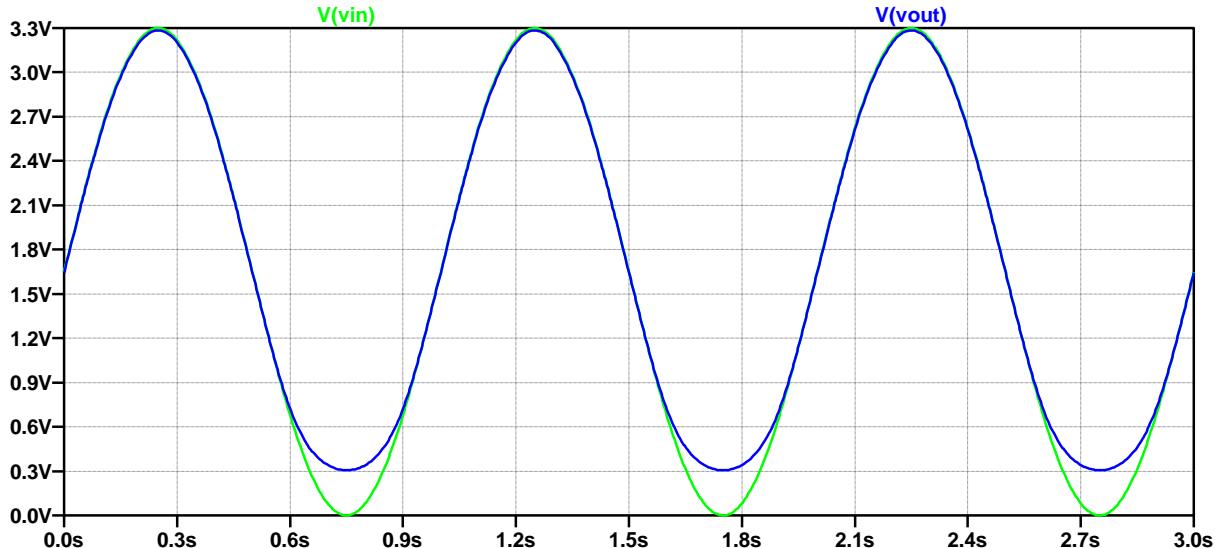


FIGURE 22: ANALOGUE PROTECTION COMMON SIGNAL

Figure 23 illustrates the response of the circuit with an input signal characterized by an amplitude of 10 V. This simulation represents a case where there is an over and undervoltage introduced to the circuit. This response confirms that the circuit accurately clamps the input voltage between 0 V and 5 V.

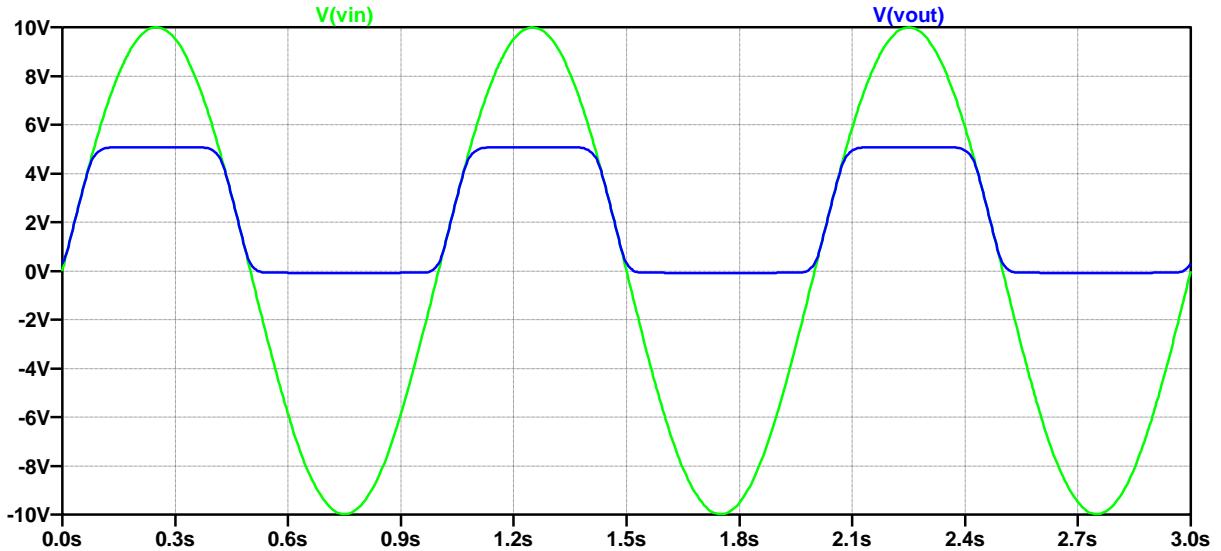


FIGURE 23: ANALOGUE PROTECTION OVERVOLTAGE

Figure 24 gives the response when a constant voltage of 5 V is applied as input to the circuit. This simulation is conducted with the exclusion of the voltage divider resistor.

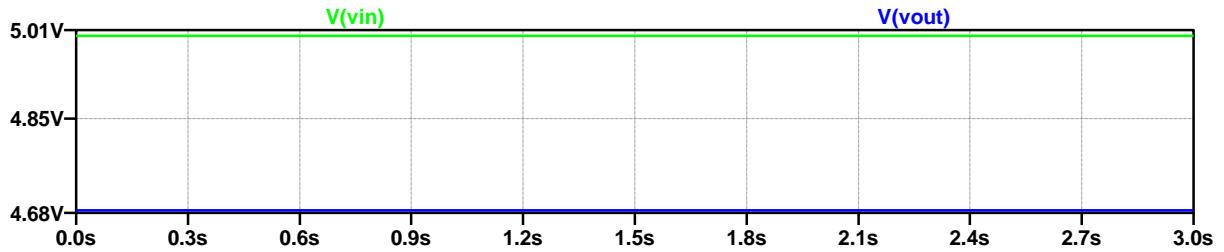


FIGURE 24: ANALOGUE PROTECTION WITHOUT SCALING

Figure 25 gives the result of connecting the voltage divider resistor to ground. This result indicates that the voltage divider accurately scales the 5 V signal to a 3.3 V signal.



FIGURE 25: ANALOGUE PROTECTION WITH SCALING

Figure 26 gives the frequency analysis of the first-order RC filter. This analysis indicate that the cut-off frequency is at 1 kHz. However, the passed signal is attenuated by 7.5 dB. This attenuation is acceptable since this will provide more sensitivity to the measured data. The system will be capable of measuring higher voltages than the specified 3.3 V range without any damage. The measured values will be manipulated in the program to represent higher voltage values.

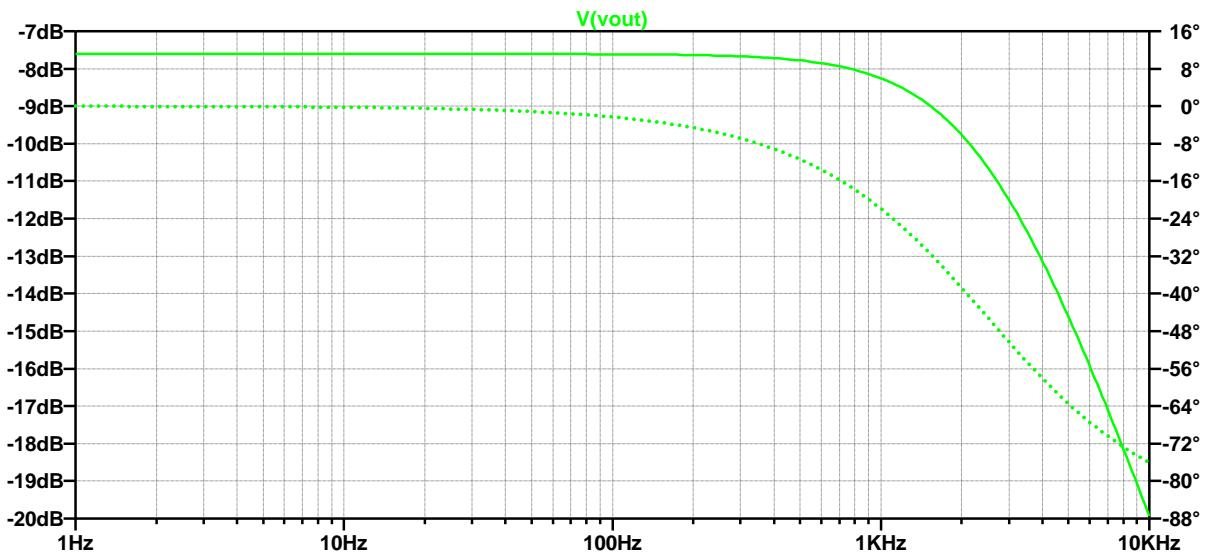


FIGURE 26: ANALOGUE PROTECTION FREQUENCY ANALYSIS

Figure 27 gives the schematic for the analogue protection circuit. There will be four protection circuits implemented for four analogue input pins.

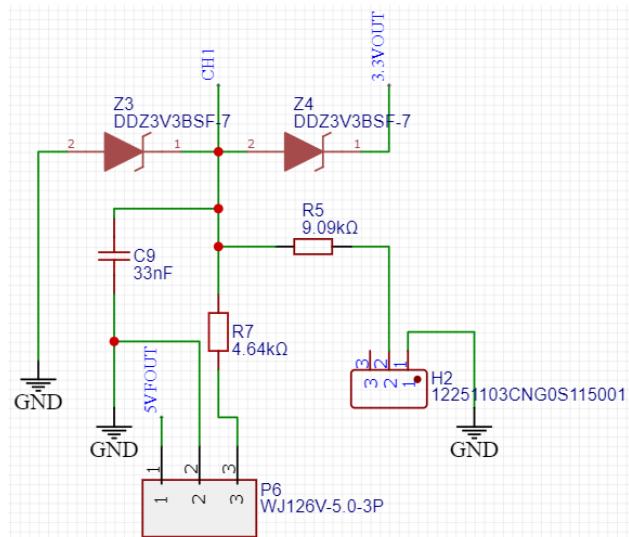


FIGURE 27: ANALOGUE PROTECTION SCHEMATIC

4.2.3 Digital outputs

The initiation of the design process for the digital outputs commenced with the careful selection of a transistor. This transistor requires high voltage and current ratings to effectively manage the back Electromotive Force (back-EMF) generated by the relay. The 2N2222A transistor is identified as the optimal switching transistor for the digital outputs, given its robust specifications.

This transistor features a collector-to-base voltage rating of 75 V, a collector-to-emitter voltage rating of 40 V, and an emitter-to-base voltage rating of 6 V. Additionally, the transistor boasts a continuous collector current rating of 600 mA, rendering it suitable for the intended application. Its widespread availability further contributes to its suitability for integration into the design.

This transistor exhibits a typical base to emitter saturation voltage of 0.6 V and a maximum rating of 1.2 V . The typical output current of a General-Purpose Input/Output (GPIO) pin of the ESP32-C3-DevKitM-1 is 40 mA . The typical output voltage of a GPIO pin of this microcontroller is approximately 3.3 V . The parameters of the transistor, in combination with the parameters of the GPIO pin can be used to calculate the appropriate resistance.

The purpose of this resistance is to ensure that the current of the GPIO pin is appropriately limited. This limitation will ensure that the transistor is properly saturated. This resistance will also prevent any damage to the GPIO pin.

A voltage of 0.7 V at the base of the transistor is desired for saturation. Therefore, the voltage drop over the resistor needs to be 2.6 V . A $1\text{ k}\Omega$ resistor is chosen to be implemented as the base resistor. This resistance results in a base current of 2.6 mA . The transistor has a typical gain of 100. Therefore, the collector current would be 260 mA .

When considering the relay model, the collector current of the transistor needs to be higher than the current needed by the coil. The relay coil is an inductive load, and when it's energized, it requires a certain current to actuate and hold the contacts. If the collector current of the transistor is too close to the relay coil's current requirement, any variations in the relay coil's resistance or the supply voltage could result in the transistor not providing enough current to keep the relay reliably activated. This can lead to unreliable operation.

The selection of a suitable relay necessitates consideration of its capacity to manage high voltages and currents. Additionally, the chosen relay should adhere to the electromechanical category and specifically feature a Single Pole, Double Throw (SPDT) configuration. The SPDT specification is crucial, as it affords the user the flexibility of Normally Open (NO) and Normally Closed (NC) options, enhancing the modularity of the relay system.

The voltage regulator, tailored for this application, will supply the requisite voltage to actuate the relays. This regulator is engineered to proficiently manage the load of the relays. Therefore, the relays need to have a rated voltage of 5 V , since this aligns with the output voltage of the regulator.

A SPDT electromechanical relay from Songle was chosen to be implemented with the transistor. This relay actuates at a voltage of 5 V and its coil current specification is 71.4 mA , which is less than the collector current of the transistor. This relay also has a maximum voltage rating of 250 VAC and a current rating of 15 A , which is ideal for this application.

The selection for the snubber diode, connected in parallel with the relay coil, is determined to be a 1N4007 diode. This diode is chosen based on its notable attributes, including a high maximum DC blocking voltage of 1000 V and a Real Mean Squared (RMS) voltage rating of 700 V . The forward voltage of the diode is not important since the purpose of this diode is to attenuate the back-EMF. This diode is also commonly available.

The relay, integrated with the 2N2222A transistor, the $1\text{ k}\Omega$ resistor, and the 1N4007 diode, underwent testing using a comparable microcontroller to the ESP32-C3-DevKitM-1, which similarly outputs 3.3 V at logic high. The relay switched efficiently between NO and NC. The base voltage was approximately 0.7 V at logic high. The collector current was approximately 70 mA . This deemed the combination of components suitable for this application.

Figure 28 gives the circuit schematic for the digital output protection circuit. There will be eight digital outputs and therefore eight digital protection circuits.

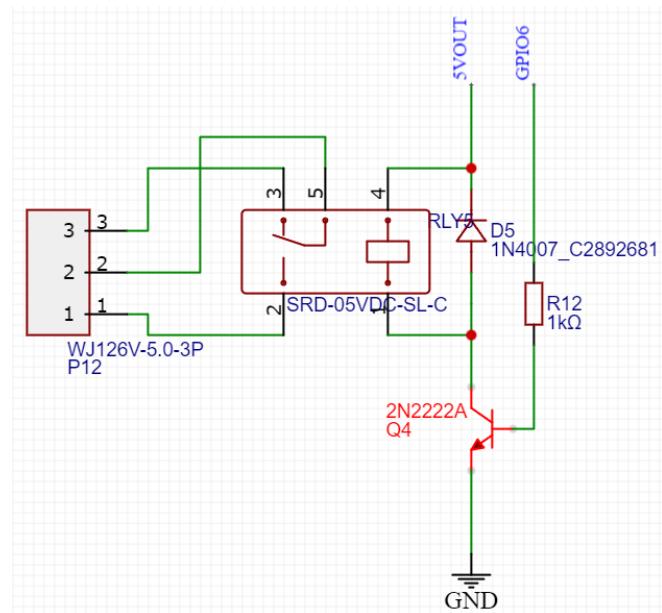


FIGURE 28: DIGITAL OUTPUT SCHEMATIC

4.2.4 WiFi activation

The system will incorporate a functionality allowing the user to selectively enable or disable the WiFi feature on the microcontroller, thereby mitigating the current consumption of the entire system. The deactivation of the WiFi capability results in a substantial reduction current consumption of approximately 100 mA. This feature provides users with the ability to optimize power usage based on operational requirements.

The user will be able to turn the WiFi off through the mobile interface. However, the user will not be able to activate the WiFi through the mobile interface since the connection is interrupted. Therefore, a hardware solution is needed to activate the WiFi again. This can be done with a tactile switch and a digital Input/Output (I/O) pin. Figure 29 gives the circuit schematic for the WiFi activation circuit

When the WiFi is active, the digital GPIO 10 pin will behave as an output and keep the Light-Emitting Diode (LED) high. However, when the user sends the command over the server to disable the WiFi, the system will deactivate the WiFi and the GPIO 10 pin will act as a digital input pin. Thereafter, the system will continuously monitor the logic level of the pin. The switch is connected to the 3.3 V output of the microcontroller. Therefore, when the switch is pressed, the pin will read the logic level as high. The system will activate the WiFi and keep the LED high.

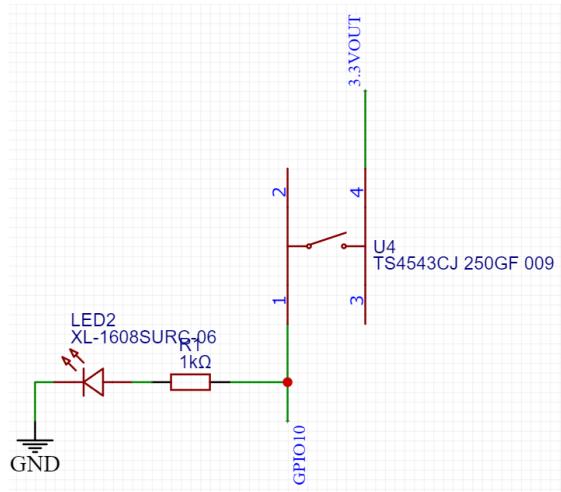


FIGURE 29: WiFi ACTIVATION SCHEMATIC

4.2.5 Microcontroller

Figure 30 gives the circuit schematic for the development board. The R_{21} resistor was implemented to slightly limit the current sunk into the microcontroller when the Zener diode connected to the 3.3 V line of the microcontroller breaks down. The P_4 screw terminal is connected to the UART receive and transmit pins. This terminal allows for the implementation of a master and slave system. This enables the user to connect different units together and program only one unit. The master unit will communicate the user settings to the slave units.

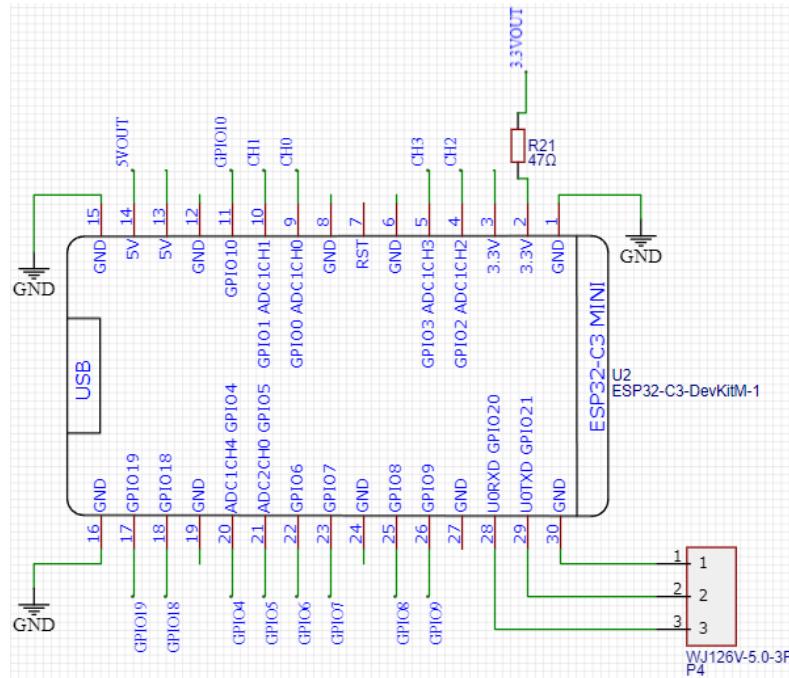


FIGURE 30: MICROCONTROLLER SCHEMATIC

4.2.6 PCB design

Figure 31 depicts the top layer of the PCB design, wherein a dedicated ground plane has been established. The design adheres to a minimum track clearance of 0.152 mm throughout all tracks. The power lines tracks were set to be 1 mm and all the other tracks 0.254 mm. The power lines were designed with thicker tracks to accommodate higher current flow. 90 degree turns in the tracks was avoided as far as possible.

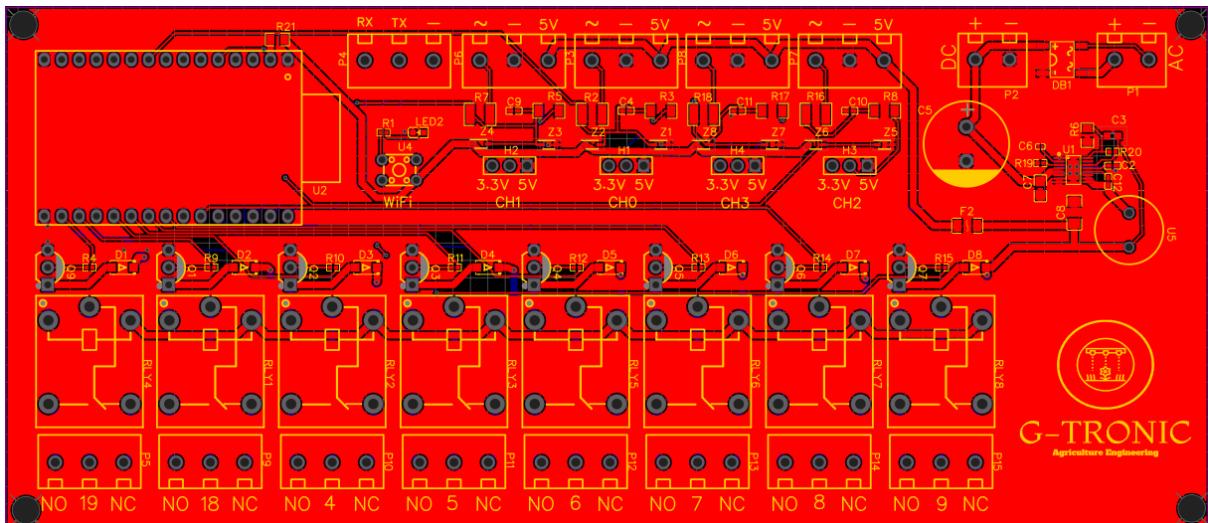


FIGURE 31: PCB DESIGN TOP LAYER

Figure 32 illustrates the bottom layer of the PCB design. The voltage divider of the analogue input circuits was implemented with three pin male headers and two pin jumper blocks. The rightmost pin of the headers is connected to ground. The pin in the middle is connected to the $9.09\text{ k}\Omega$ resistor. The leftmost pin floats. Therefore, if the connected sensor outputs a 3.3 V signal, the jumper block will be connected over the two leftmost pins of the headers. If the sensor outputs a 5 V signal, the jumper block will be connected to the two rightmost pins of the headers. The holes positioned at the corners of the PCB is designed with a diameter of 4 mm . These holes can be used to mount the PCB with standard 4 mm screws.

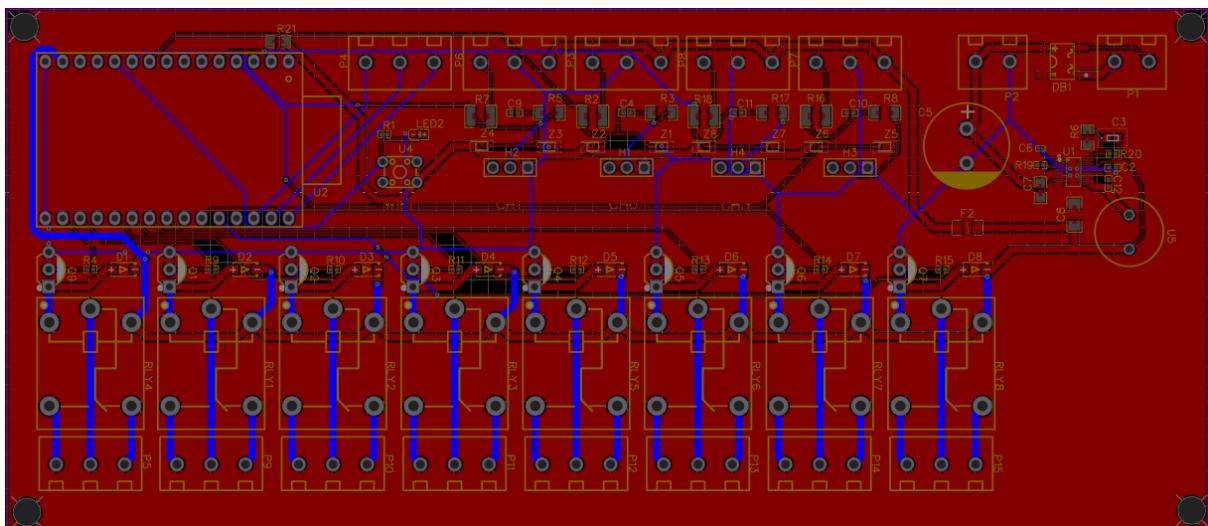


FIGURE 32: PCB DESIGN BOTTOM LAYER

Figure 33 gives the 3D model of the PCB design as generated by EasyEDA. The models for the Zener diodes (Z components) and the $C3$ capacitor are absent because the EasyEDA library did not include the 3D models for these components.

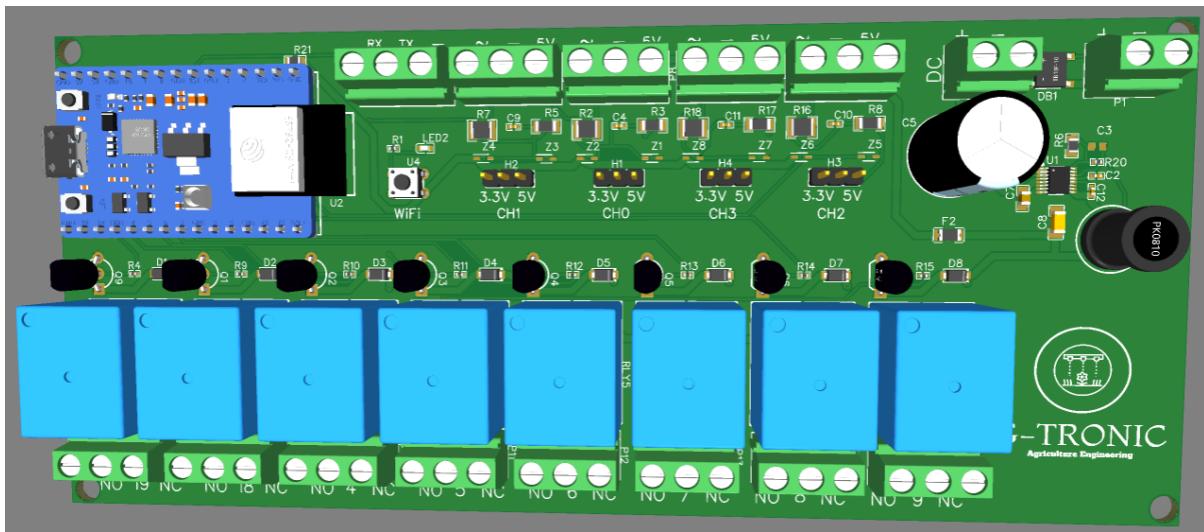


FIGURE 33: 3D MODEL OF PCB DESIGN

4.3 Firmware design

Figure 34 gives the flowchart of the Setup state. This state is the first state to be executed when the microcontroller boots. The Serial Peripheral Interface Flash File System (SPIFFS) is a file system designed for small embedded systems such as microcontrollers. This file system will be used to save the settings configured by the user in the flash memory. This will enable the system to load the previous settings on reset.

The settings will be saved in a JavaScript Object Notation (JSON) file. JSON allows the serialization of data in an efficient manner, making it suitable for representing complex configuration settings with nested parameters. JSON is also a lightweight data format that doesn't introduce excessive overhead in terms of storage space or memory usage. This is especially important in resource-constrained microcontroller environments.

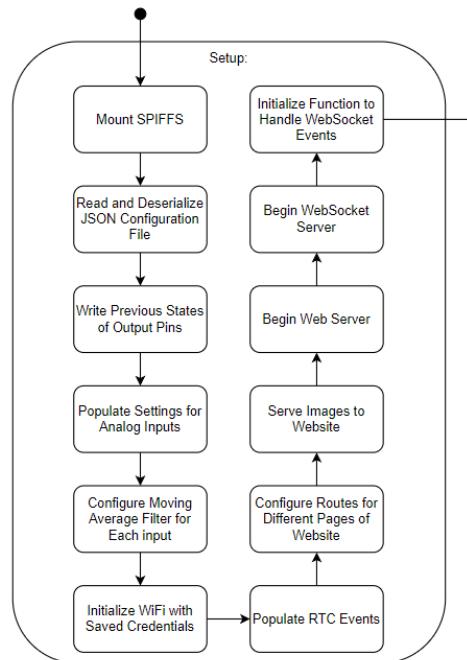


FIGURE 34: SETUP STATE

From the Setup state, the program transfers to the Main Loop state. This state is depicted by Figure 35. All the variables visible in this state will be initialized as zero or empty before the Setup state. The relevant channel's settings will be saved in their own "channelVector" vector. A vector will be used because the size of a vector is adjustable in C++, which is the programming language within the Arduino IDE. Each channel's settings will contain thresholds at which a certain output action must be performed on a defined GPIO pin. Each of the thresholds and outputs of a channel will be saved in a struct within the channel's vector.

The RTC events will be managed in a similar manner. The events and their corresponding information will be saved individually in a struct, within the schedule vector. The conditions paired with an action to reset the relevant variable, are designed to ensure that the message is only sent once when the page is loaded.

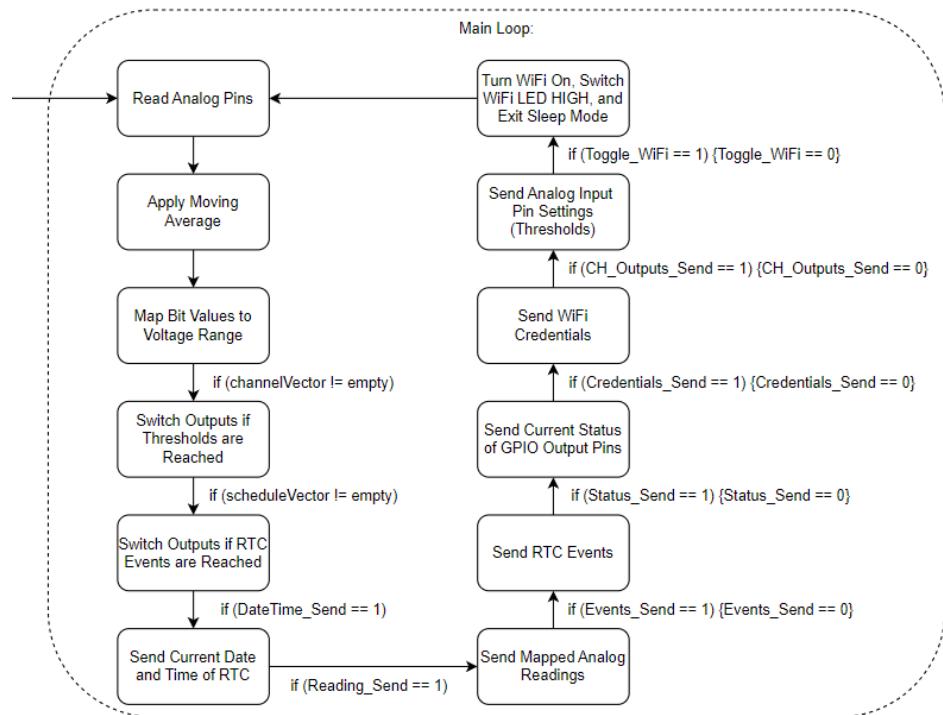


FIGURE 35: MAIN LOOP STATE

Figure 36 outlines the function responsible for managing incoming WebSocket messages. This figure describes the specific instances and conditions under which the variables used in the Main Loop are toggled. The messages processed by this function originate from diverse webpages. The operations and states of these pages are detailed in Appendix B. It is noteworthy that this state is represented such that it will run parallel with the Main Loop. However, in the implementation thereof, the states will not run simultaneously. The WebSocket Event Handler behaves as an interrupt. Therefore, when a message is received through the WebSocket Server, the handler will interrupt the Main Loop's operations.

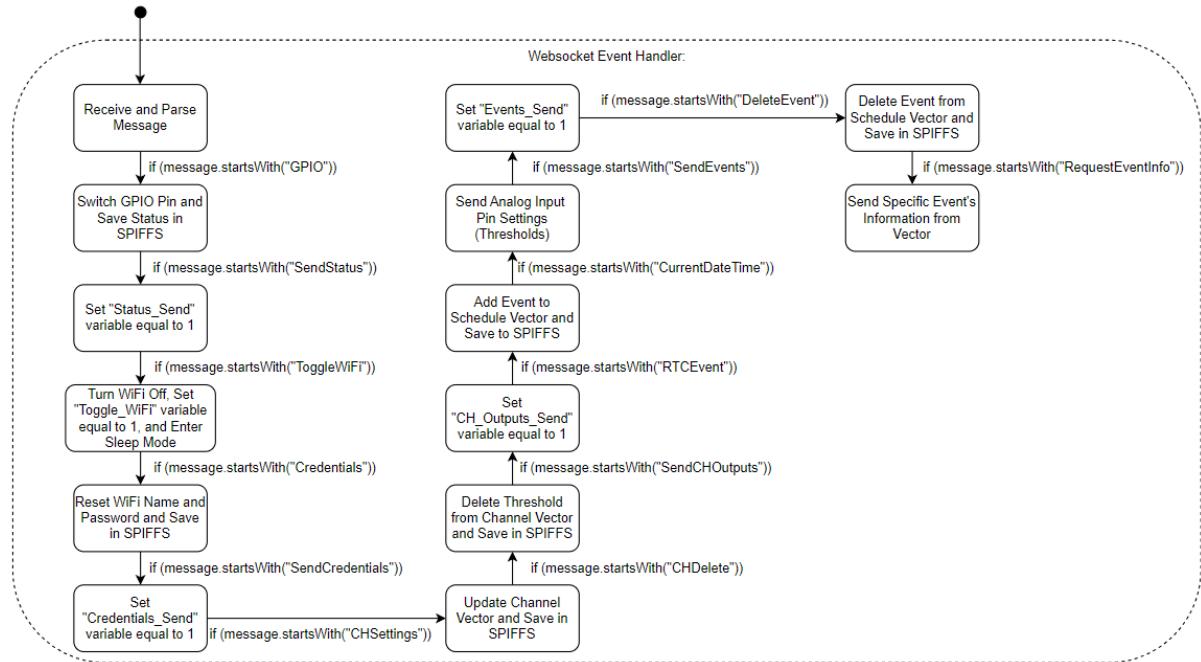


FIGURE 36: WEB SOCKET EVENT HANDLER STATE

Figure 37 indicates how the different pages are served to the browser. This state also illustrates how the “Reading_Send” and “DateTime_Send” variables will be managed. These variables are only toggled during page transitions. This will ensure that the analogue readings, current date, and time are continuously sent over the WebSocket until the relevant page is unloaded. The pages themselves will be constructed using a combination of HTML, CSS, and Javascript. Each page's code will be represented to the browser as a concatenated character sequence within the program memory.

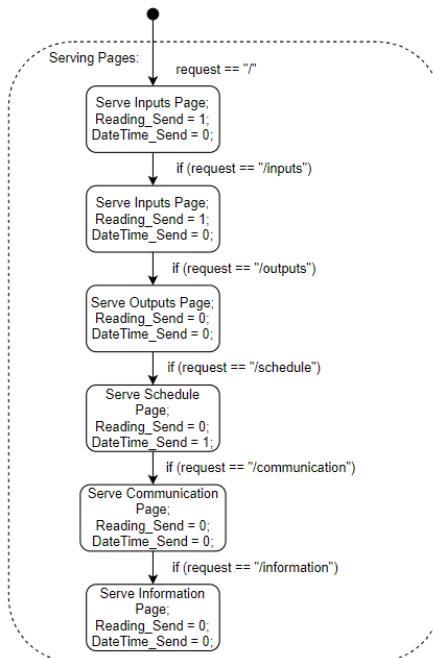


FIGURE 37: SERVE PAGES STATE

The diagrams presented in Appendix B, detailing the operations of the pages, underscore that solely the information, schedule, and outputs pages transmit a message upon the WebSocket

Server's initialization. This was done because the parameters displayed by this page only needs to be received once upon loading. Therefore, the parameters need to be requested only when the WebSocket server has been fully initialized.

The utilization of the "Socket.onopen" command serves to guarantee that the message transmission occurs exclusively when the server attains full initialization. If this command is omitted, and the message is conveyed before the full initialization of the server, then the microcontroller will not receive this message. It is imperative to note that the socket server associated with each page must be closed prior to unloading the page to avoid potential interference with the server established on the subsequent page.

4.4 Software design

This section will discuss the website's User Interface (UI) design and the features associated with it. The page's UI design can be seen in Appendix C.

4.4.1 Analogue Inputs

This page's UI can be seen in the Analogue inputs section within Appendix C. The main page of the analogue inputs will display the four ADC channels. Each channel's readings will be displayed in real-time. When the user clicks on one of the channels, the channel's settings will be displayed.

The user will be able to specify a sensitivity for the channel. This sensitivity value will adjust the window size of the moving average filter. Consequently, entering a higher sensitivity value enhances the stability of the readings, while a lower value has the opposite effect. This dynamic functionality provides users with the means to tailor the performance of the channel to their specific requirements.

Furthermore, the user can add thresholds at which the state of the GPIO pins must be toggled. This threshold is associated with the analogue reading. The user can specify in which state a GPIO pin must remain when the analogue reading is less than, greater than, or equal to a certain value.

4.4.2 Digital Outputs

This page's UI can be seen in the Digital outputs section within Appendix C. This page will allow the user to manually toggle the state of the GPIO. However, the program will keep the GPIO pin in a defined state if there are thresholds configured on the pin. This page will also indicate the status of the GPIO pins.

4.4.3 Schedule

This page's UI can be seen in the Schedule section within Appendix C. This page will manage all the RTC events. Users will have the capability to define the event's name, timing details, duration, and specify the number of days to skip between occurrences. Additionally, the user is prompted to designate the GPIO pin for switching and its corresponding logic level. The pin will be toggled to the opposite state after the specified duration has passed.

Chapter 5: Implementation

5.1 Introduction

This chapter will discuss the implementation process of this project. This will include the acquisition of the various hardware components. This chapter will also discuss the implementation of the firmware and software.

5.2 Hardware

This section will examine the hardware acquisition process, encompassing the construction of the PCB. Furthermore, it will delve into the details of the Bill of Materials (BOM). A thorough analysis will be conducted to calculate the price of a complete unit.

The PCB was constructed with a combination of two different soldering techniques. The through-hole components were soldered with a soldering iron and soldering wire. The surface mount components were soldered with soldering paste and a hot air gun.

All components were soldered by the author excluding the Step-Down Voltage Regulator IC. This chip was professionally soldered by JLCPCB. This decision was made to mitigate the risk of potential human error. The Step-Down Voltage Regulator IC, featuring a ground pad underneath, presents a challenging soldering task. This prompted the outsourcing of this specific soldering operation. The cost of this assembly service is included in the BOM in Appendix E.

All components detailed in Section 4.2 have been incorporated into the BOM, as presented in Appendix E. The BOM provides comprehensive information, including the price per unit, quantity, and total cost for each component. The quantities of each component were chosen to ensure that 10 complete units can be constructed. This was implemented to avoid the potential risk of faulty components. This decision not only safeguards against the prospect of defective parts but also serves to pre-empt additional import costs that might be incurred in the event of a greater quantity requirement.

Included in the BOM, that was not considered in Section 4.2, is female header pins and 10 $k\Omega$ metal film resistors. The acquisition of female header pins was specifically intended for the microcontroller terminals. This was done to avoid soldering the microcontroller directly to the PCB. The implementation of female header pins facilitates enhanced flexibility, allowing for convenient code modifications on the microcontroller or conducting tests without the need for intricate soldering operations.

The acquisition of the 10 $k\Omega$ metal film resistors served the critical purpose of grounding the base of the transistors, a key realization made during the implementation process. This resistance should be large enough to mitigate the risk of creating a voltage divider circuit with the 1 $k\Omega$ base resistance. Therefore, a resistance value of 10 $k\Omega$ was sufficient for this application. Figure 28 in Section 4.2.3 depicts the circuit schematic where the base of the transistor floats. Figure 38 illustrates the addition of the R_{22} resistor to the digital output circuitry.

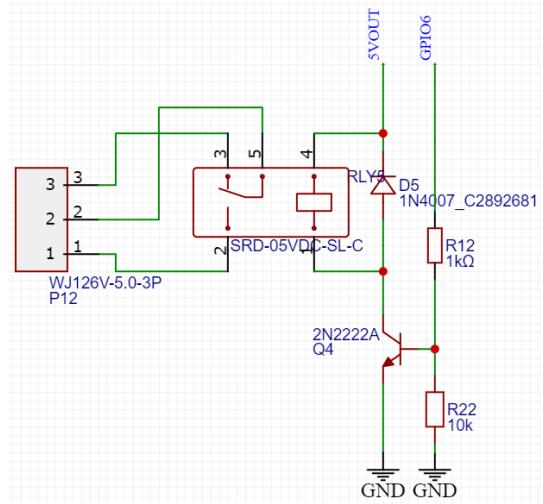


FIGURE 38: GROUNDED DIGITAL OUTPUT CIRCUIT

As observed from the BOM in the Appendix, the aggregate cost of all components, inclusive of shipping, amounts to R 10471.93. Dividing this total by the intended quantity of 10 units, results in a cost per unit of R 1047.19. This cost per unit is acceptable and aligns with the project's affordability requirement. The pricing of R 1047.19 per unit positions this project as a cost-effective solution for irrigation management.

5.3 Firmware and software

Appendix D gives the final implementation of the various pages of the website. The appearance of the website has been improved from the design illustrated in Appendix C. However, the operation and flow of the pages remained similar.

5.3.1 Analog inputs

This section will discuss how the various settings set by the user for each channel are handled within the software and firmware. The code used to switch the outputs at the desired moment will be discussed as well.

A deliberate choice was made to dedicate an individual page to each of the four channel inputs, deviating from the initial detailed design for this page. This decision was motivated by the complexity of displaying each channel's settings on a single page when the channel is selected. The implementation opted for a more streamlined approach, simplifying the code structure by segregating channels into distinct pages.

Each user-generated output block is assigned a unique identifier (ID) to distinguish it within the system. When users submit their settings, this specific ID, along with additional information, is transmitted to the server. Upon reception of this data, the server systematically stores the information in a structure (struct). Thereafter, the server appends the struct into a vector that encompasses all the data pairs associated with that channel. The information saved in the struct for each pair includes the following:

- PairID – This denotes the unique identifier for that pair of data.
- Threshold – This represents the specific analogue voltage at which the output must be toggled.
- Expression – This indicates if the measured analogue voltage should be higher, lower, or equal to the threshold.
- Action – This identifies the logic level of the specific pin when the threshold is reached.
- Output – This variable contains the specific output pin that must be toggled.

When the specific channel's page is loaded by the browser, the server sends all the information within the channel's vector from the SPIFFS to the browser. The browser parses this information and displays the pairs of data to the user. A new unique ID are created, for each new pair, with a number greater than that of the largest ID received by the server.

With every iteration of the main loop, the program processes the vector associated with each channel, given that the vectors are not empty. Thereafter, the program processes each struct within the vector. If the expression variable within the struct is "GT," and the current analogue measurement surpasses the threshold variable in the struct, the pin undergoes a logic level transition. This transition is determined by the action variable specified in the struct. A similar if statement is used if the expression is equal to "LT".

If the expression within the struct equals "EQ," and the absolute difference between the current analogue measurement and the specified threshold is less than 0.1, the pin undergoes a logic level transition based on the action variable in the struct. This tolerance is introduced to mitigate potential interference from noise. In the event of "EQ" and an absolute difference exceeding 0.1, the pin transitions to the opposite logic level as specified by the action variable in the struct.

5.3.2 Schedule events

This section will discuss the operations that transfers and receives the data of each scheduled event. The code used to toggle the outputs at the desired time will be discussed as well.

The event IDs and other data are handled in a similar manner as described in the previous section. The information saved in the struct for each event includes the following:

- EventID – This holds the event's unique identifier.
- Name – This variable contains the name given to the event by the user.
- Time – This holds the RTC time at which an output must be toggled.
- Duration – This indicates the amount of time the output must be kept at a specified logic level.
- EndTime – The end time for toggling the output to the opposite logic state, as specified by the user, is determined by the "Time" and "Duration" variables. These values, received from the user, are parsed into hours and minutes. The hours are converted to minutes by multiplying by 60. The converted minutes, the minutes parsed from the time, along with the user-defined duration, are added together. The resulting total minutes are then used to calculate the new hour and minute values. The hours are obtained by dividing the total minutes by 60, and the minutes are obtained by taking the remainder of that division. Finally, the hours and minutes are combined into the same format as the "Time" variable.
- Date – This variable gives the first date at which the specific output must be toggled.
- Repeat – This variable indicates how many days must be skipped in-between the toggled actions.
- Action – This variable indicates the logic level to which the output must be toggled at the specific time.
- Output – This variable indicates which output pin must be toggled.
- RunEndIf – This variable specifies whether the specific event's output has been toggled at the starting time.
- UpdateDate – This variable specifies whether the specific event has reached its end time.

With every iteration of the main loop, the program processes the schedule vector, if the vector is not empty. Upon entering this if statement, the current RTC time and data are read and stored in temporary variables. Thereafter, the program processes each struct within the schedule vector. If the date and time within the struct equals the current RTC date and time, then the specified output pin are toggled to the level defined by the action variable. However, the pin is only toggled if the “RunEndIf” variable is equal to zero. After the pin was toggled, the “RunEndIf” variable is set to 1.

If the end time and date is equal to the current time and date, and the specific struct’s “RunEndIf” variable is equal to 1, then the output pin is toggled to the opposite level as specified by the action variable. In addition, the “RunEndIf” is set to 0 and the “UpdateDate” variable is set to 1. If the repeat variable is not equal to 0 and the “UpdateDate” variable is equal to 1, the new date is calculated according to the “Repeat” variable.

The new date is calculated by first parsing the starting date into years, months, and days. An array of numbers is created which represents the number of days in each month. The program checks if the current year is a leap year. The current year is a leap year if the year is divisible by 4 but not by 100. The year is also a leap year if the year is divisible by 400. If the year is a leap year, the number of days in February within the array is changed from 28 to 29 days.

Thereafter, a while loop is entered with the condition that the “Repeat” variable must remain greater than 0. The loop operates until all days within the “Repeat” variable have been added. It begins by determining the number of days remaining in the current month, accounting for the current day. If the days to add exceed the remaining days in the month, it advances to the next month, updating the year and adjusting for leap years. If the loop doesn't need to switch to the next month, it simply adds the remaining days to the current day. This process continues until the desired number of days have been added, resulting in a new date representation. The loop efficiently manages month transitions, considering the varying number of days in each month, and ensures accurate date calculations.

Chapter 6: Testing

6.1 Introduction

This section will discuss the tests performed on the hardware, firmware, and software of the system. These tests will demonstrate the functionality of the system.

6.2 Voltage regulator

6.2.1 Test

This section will test the start-up of the voltage regulator. The stability of the voltage regulator will be tested as well. These tests will be performed with no load and with load at 12 V DC input. The voltage regulator will be tested with a 22 VAC transformer as well. In addition, the output of the voltage regulator will be tested at 9 V, 12 V, and 24 V DC input.

6.2.2 Method

The start-up of the voltage regulator will be monitored with an oscilloscope probe connected at the 5 V output supply of the board. Two distinct PCBs were fabricated to assess different scenarios: one with only the voltage regulator soldered onto the board, representing the no-load condition, and another with all components soldered onto the board, representing the loaded scenario.

These configurations will undergo testing with a 12 V input supply from a DC bench power source. The voltage regulator's start-up with the transformer will be evaluated using the fully populated PCB. The stability of the voltage regulator will be assessed by switching the relays on and off simultaneously.

6.2.3 Results

Figure 39 gives the result of the start-up test with no-load.

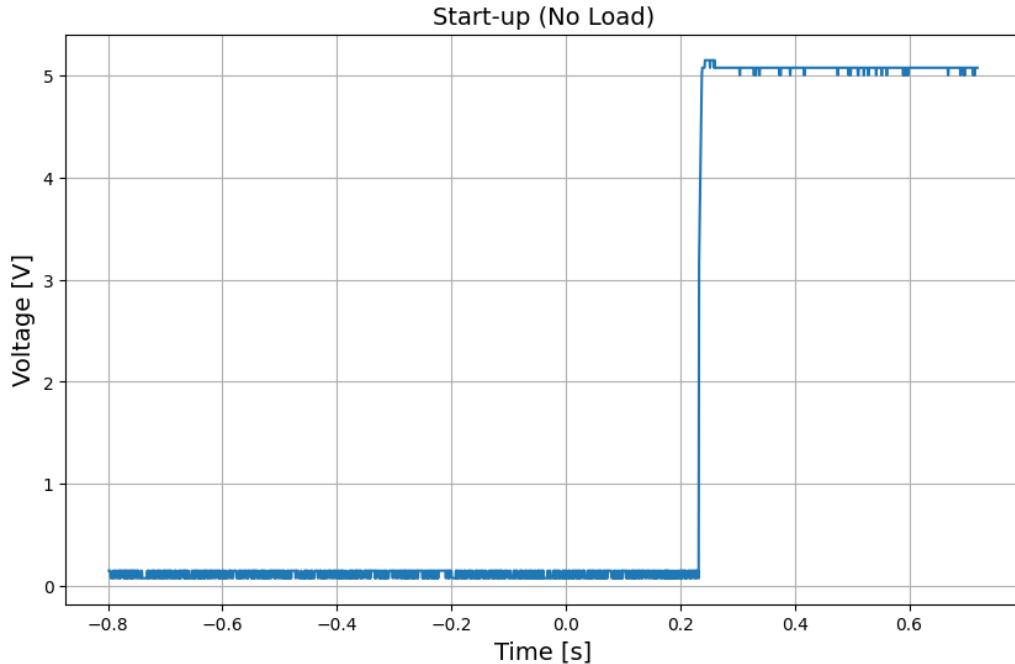


FIGURE 39: START-UP WITH No-LOAD

Figure 40 gives the result of the start-up test with load.

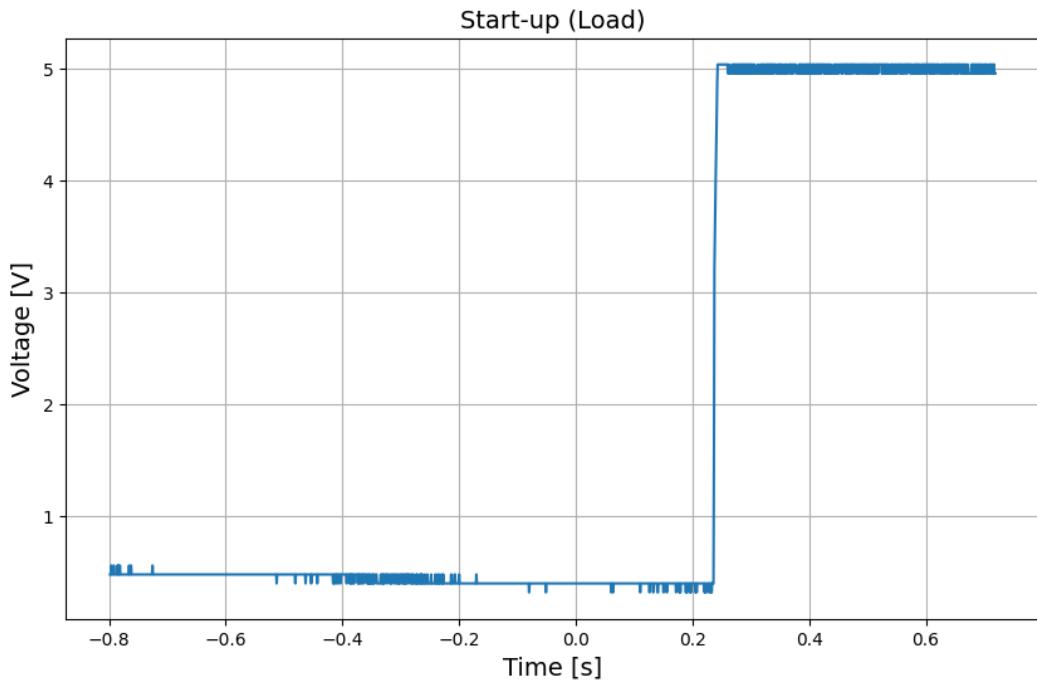


FIGURE 40: START-UP WITH LOAD

Figure 41 gives the result of the start-up test with load and the transformer as input.

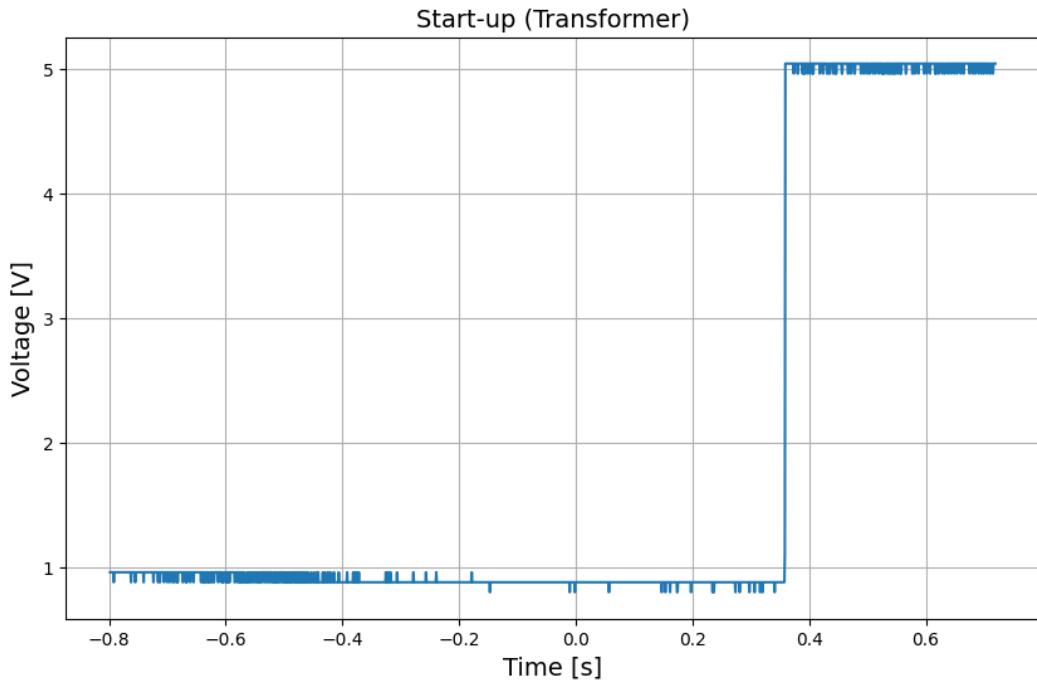


FIGURE 41: START-UP WITH TRANSFORMER

These results indicate that the voltage regulator performs as expected. The start-up of the voltage regulator can be approximated by observing the recorded data itself. The data indicates that the start-up time of the voltage regulator in each case is less than 5 ms. This is sufficient for this application. The figures also indicate that the output remains at 5 V in steady state.

The voltage regulator's output remained stable when the relays were switched simultaneously. The output voltage dropped by approximately 60 mV when all relays were switched on. Furthermore, the output remained stable with 9 V, 12 V, and 24 V DC input.

6.3 Digital outputs

6.3.1 Test

This section will test the stability of the relays and the switching operation. The mitigation of the back-emf generated by the relays will be analysed as well.

6.3.2 Method

The relay stability will be assessed through a systematic switching sequence initiated by the microcontroller. Dedicated code has been developed to sequentially activate each relay with a 500 ms interval, followed by individual deactivation at the same interval. Subsequently, all relays will be activated simultaneously after a 1 s delay, followed by simultaneous deactivation after another 1 s interval. This entire cycle repeats every 2.5 s. The behavior of the relays will be observed and documented.

The back-emf of the relay was tested with the same code. However, an oscilloscope probe was placed at the base of the transistor. Another probe was placed at the collector of the transistor.

6.3.3 Results

The code underwent a 45-minute runtime test on the system, revealing consistent stability in the relay operation without any indications of component failure or excessive heat generation. Figure 42 gives the voltage as measured at the base and the collector of the transistor. This result is indicative of the efficient mitigation of the back-emf through the combination of the diode and the transistor.

A residual 3 V of back-emf is observed in the collector voltage. However, the transistor effectively handles this voltage, signifying a substantial reduction compared to the potential higher voltage in the absence of the snubber diode. The minor back-emf observed in the base voltage, although present, remains at a negligible level, posing no threat of damage to the digital pin.

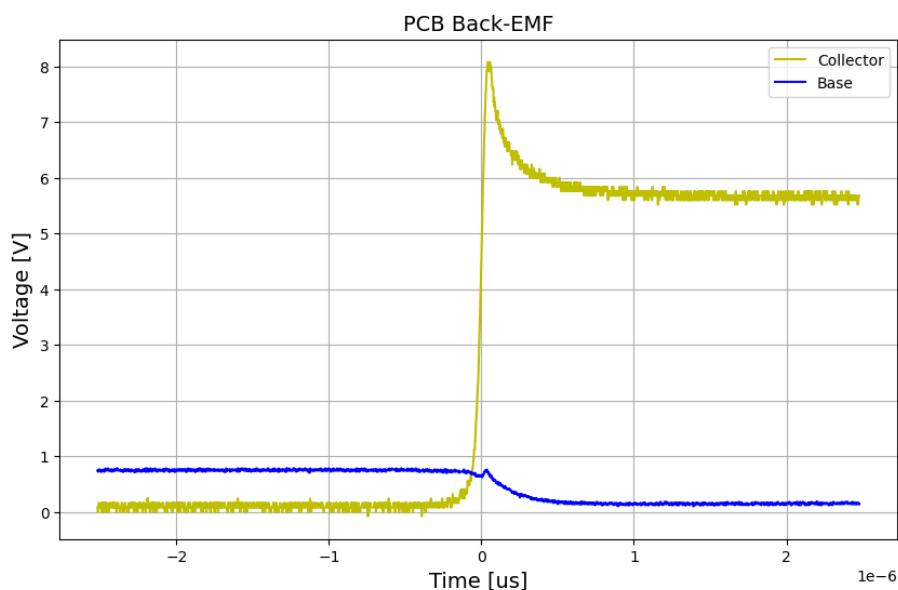


FIGURE 42: BACK-EMF IN DIGITAL OUTPUT CIRCUIT

6.4 Analogue inputs

6.4.1 Test

This section will test the accuracy of the analogue readings. The protection circuit will be evaluated as well. Finally, the RC low-pass filter will be tested.

6.4.2 Method

The accuracy of the readings will be tested by generating a sine wave signal with a signal generator. This signal will have an amplitude of 1.65 V and an offset of 1.65 V . This will simulate a typical output signal of a 3.3 V sensor. The protection circuit will be tested by generating a sine wave signal with a 10 V amplitude and an offset of 0 V . The attenuation of the low-pass filter will be examined at 2 kHz , 20 kHz , and 200 kHz .

6.4.3 Results

Figure 43 gives the result of the accuracy reading of the circuit. The yellow plot represents the input signal generated by the signal generator. The blue plot represents the voltage that would be measured by the ADC. It is evident from this figure that the readings appear to lack accuracy, suggesting that the Zener diodes excessively clamp the signal.

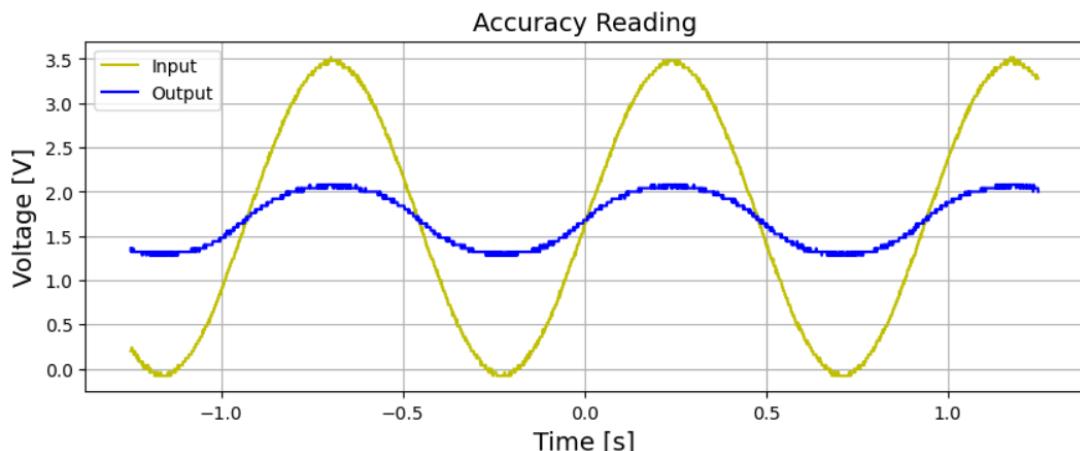


FIGURE 43: MEASURED ANALOGUE VOLTAGE

Figure 44 gives the response of the protection circuit to an overvoltage signal. This response indicates that the voltage signal to the ADC pin does not exceed the range of 0 V to 3.3 V . This behavior will effectively protect the ADC pins from overvoltage.

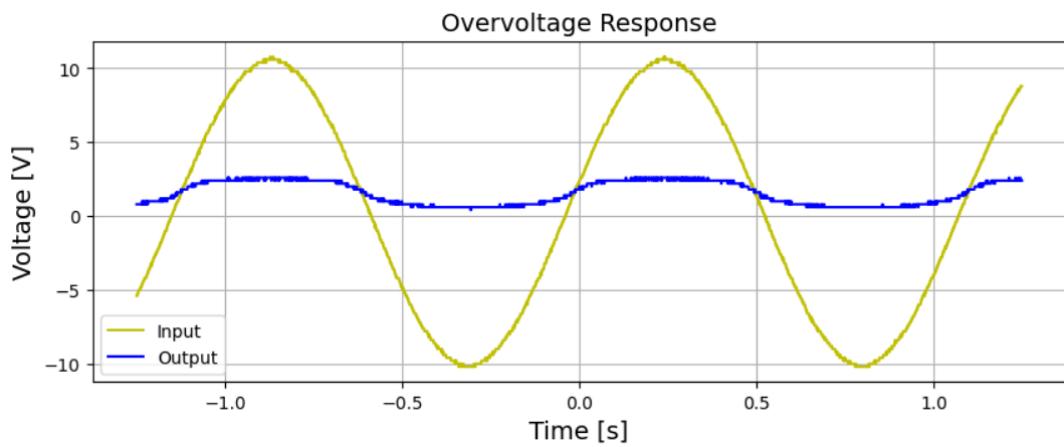


FIGURE 44: MEASURED ANALOGUE VOLTAGE WITH OVERVOLTAGE SIGNAL

The Zener diode connected to the 3.3 V supply was removed. This was done to test if the accuracy of the readings would improve while the clamping behaviour remained within the design requirements. Figure 45 gives the response of the circuit to a typical signal. This result indicates that the readings will be slightly more accurate for the lower voltages.

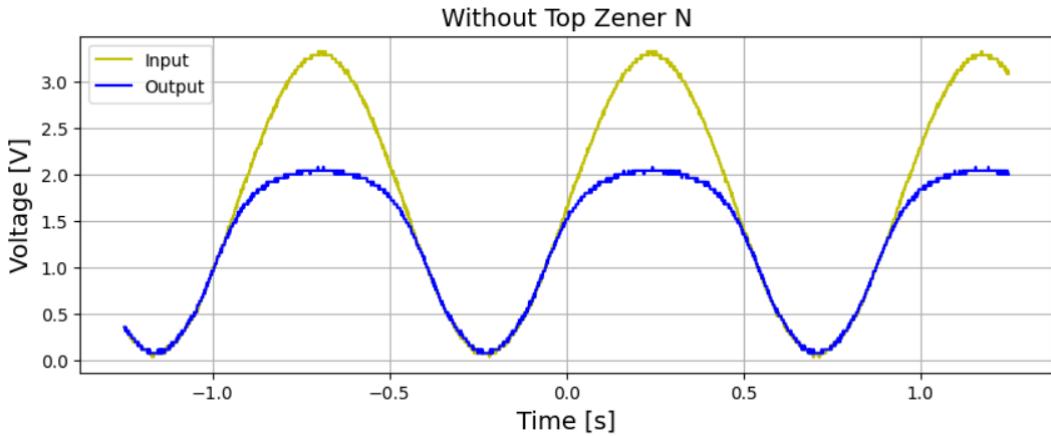


FIGURE 45: MEASURED ANALOGUE SIGNAL WITHOUT TOP ZENER

Figure 46 gives the response of the circuit with an overvoltage signal. This response indicates that the circuit without the top Zener diode clamps the voltage sufficiently. However, the Zener connected clamps the signal closer to 0 V. This indicates that the circuit containing the top Zener will be more appropriate for protection against overvoltage.

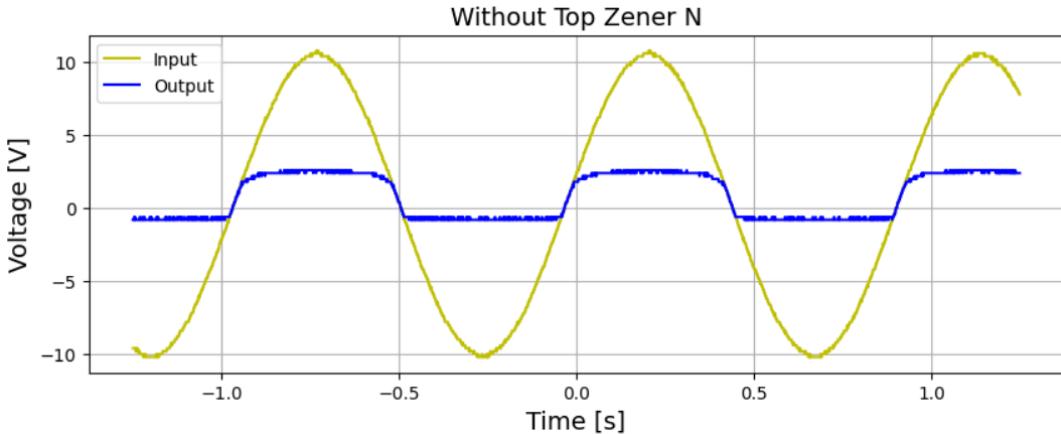


FIGURE 46: MEASURED SIGNAL WITH OVERVOLTAGE SIGNAL WITHOUT TOP ZENER DIODE

In order to make the readings more accurate, the ADC readings are mapped within the code to a range of 0 to 3.3 V. The data obtained from the oscilloscope was linearly mapped within Python to determine which circuit will perform the best. The data was mapped with Equation (7):

$$\text{Mapped Voltage} = \frac{\text{Voltage} - \min(\text{Voltage})}{\max(\text{Voltage}) - \min(\text{Voltage})} \times 3.3 \quad (7)$$

Figure 47 gives the mapped voltage of the signal measured without the Zener diode. The result of this mapping exhibits an error of about 0.5 mV.

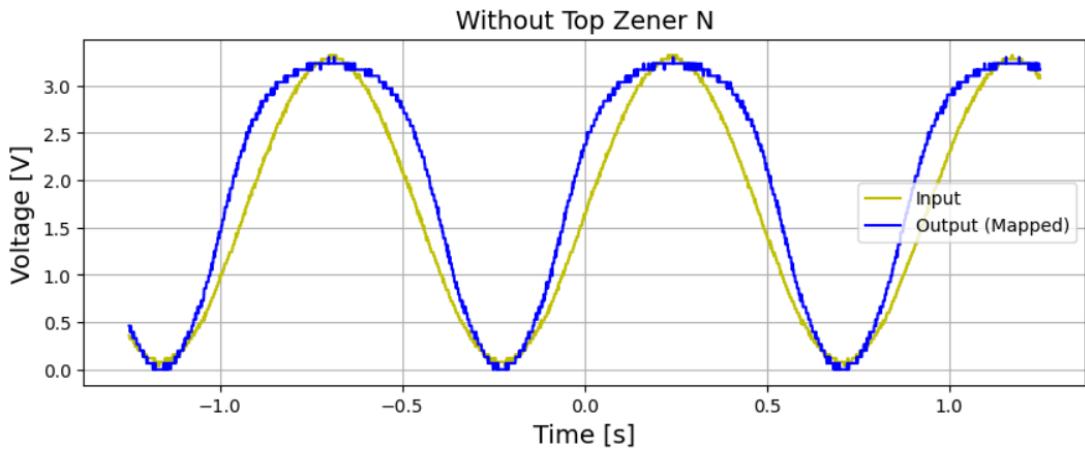


FIGURE 47: MEASURED SIGNAL WITHOUT ZENER (MAPPED)

The voltage data of the signal measured with the top Zener was mapped as well. Figure 48 gives the mapped response of the readings. This response is more accurate than the response without the top Zener. However, the resolution is less desirable. Therefore, a moving average filter was applied to the data to improve the resolution (red signal).

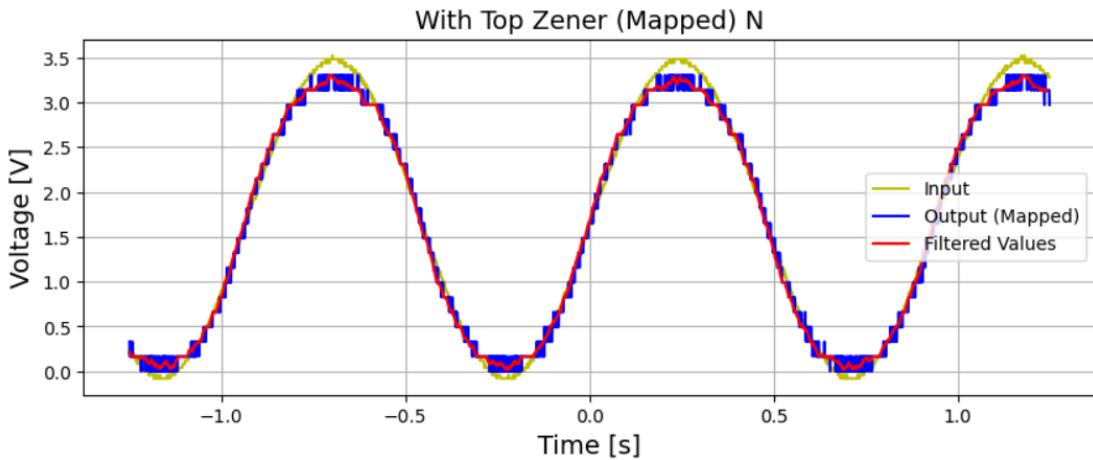


FIGURE 48: MEASURED SIGNAL WITH TOP ZENER (MAPPED)

To test the accuracy of the linear mapping with the measured ADC values, a typical sine wave signal between 0 and 3.3 V was generated. The 12-bit values measured by the microcontroller was exported using the BetterSerialPlotter. The minimum and maximum of this data was determined and the data was mapped with Equation (7) to a range between 0 and 3.3 V. Figure 49 gives the plot of the bit data measured.

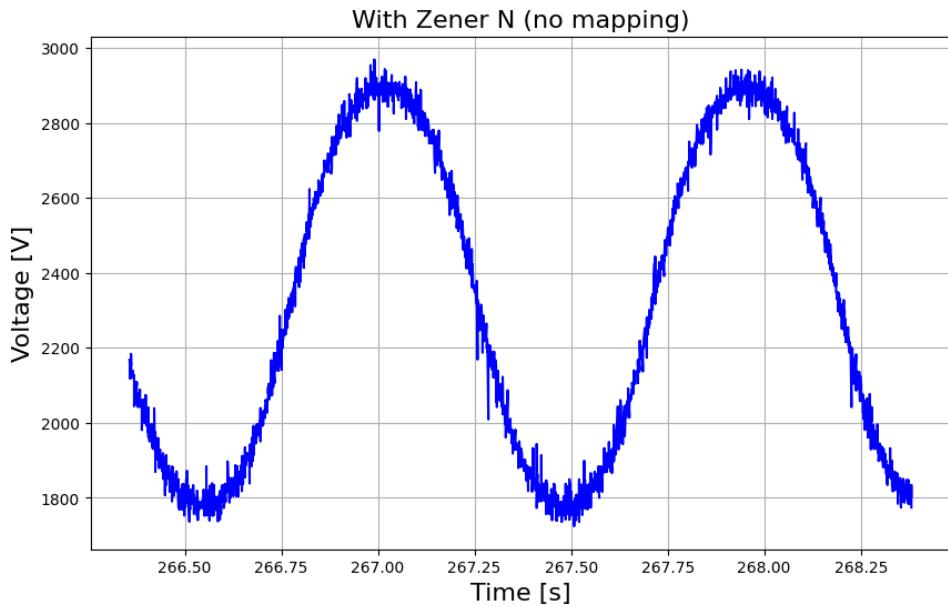


FIGURE 49: BIT VALUES (NO MAPPING)

The signal generated by the signal generator was mapped from a range of 0 - 4095 bits to 0 - 3.3 V. Figure 50 gives the result of the linear mapping. It is clear from this figure that the measured values are accurate to the actual data.

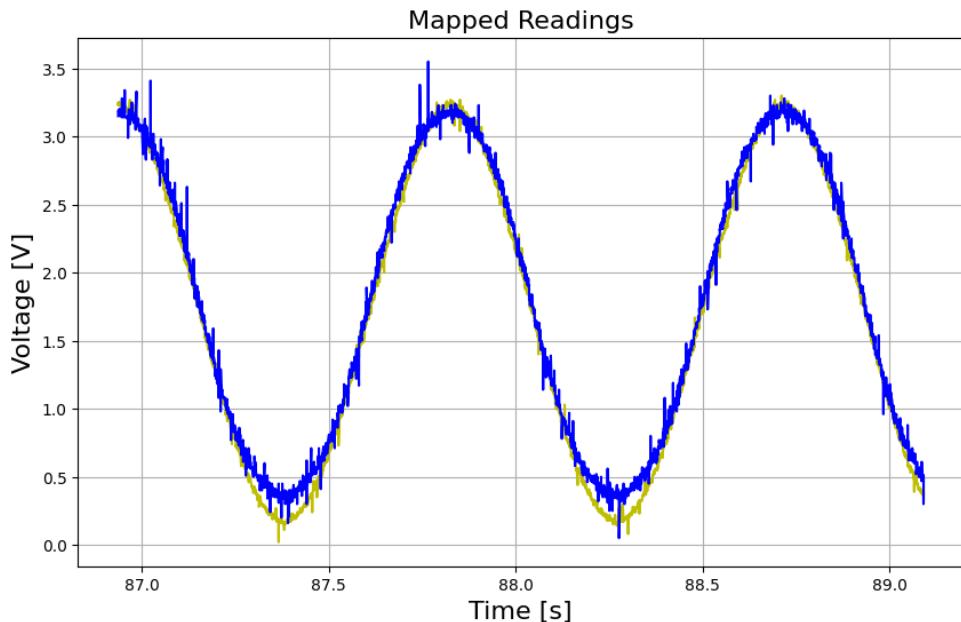


FIGURE 50: MEASURED SIGNAL ACCURACY MAPPING

A moving average filter was applied to the measured values. The moving average was applied with the Equation (8):

$$\text{Moving Average} = \frac{1}{\text{BufferSize}} \sum_{i=0}^{i < \text{BufferSize}} \text{Buffer}[i], \quad (8)$$

where the BufferSize was initialized as 10. The first 10 values in the *Buffer* was initialized as 0. Figure 51 gives the response with the moving average applied. The noise is filtered out and

the resolution is improved. This response also coincides with the simulated response given by Figure 22, where the measured signal is clamped at approximately 300 mV.

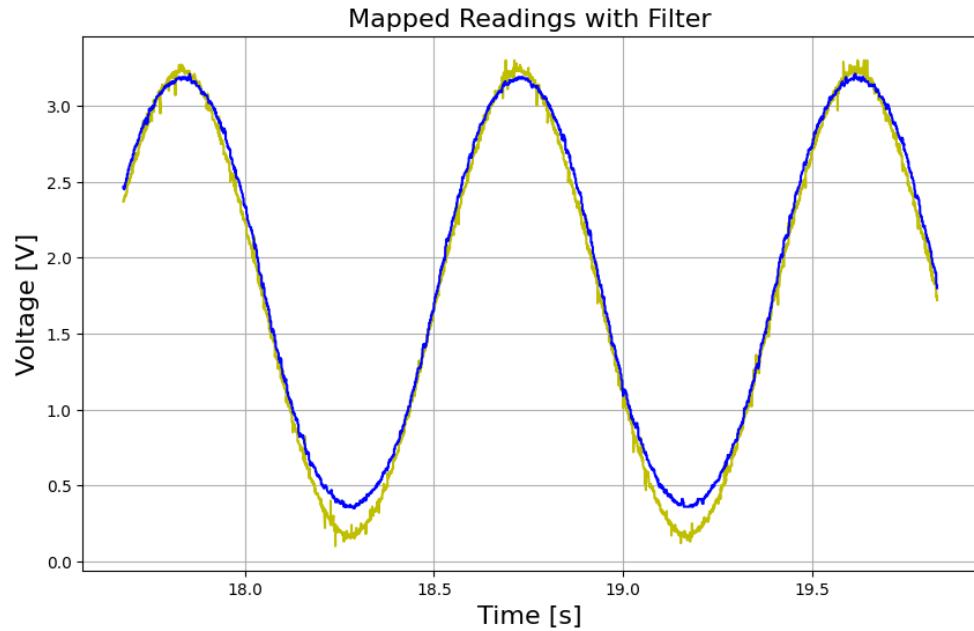


FIGURE 51: MEASURED SIGNAL MOVING AVERAGE APPLIED

This response was tested with a triangular wave as well. Figure 52 illustrates this response.

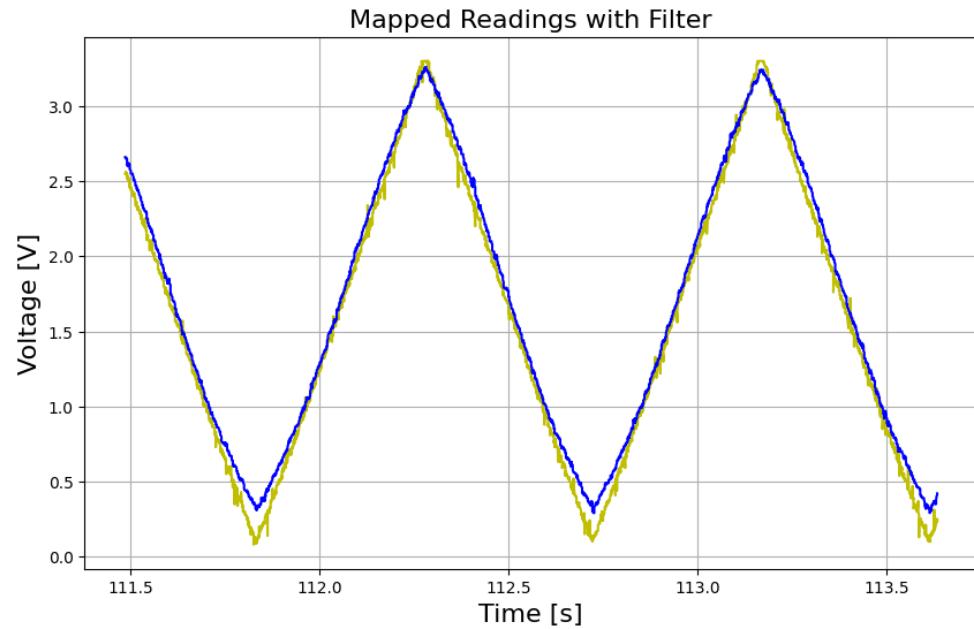


FIGURE 52: TRIANGULAR WAVE RESPONSE

In addition, the response was tested with a block wave. Figure 53 illustrates this response.

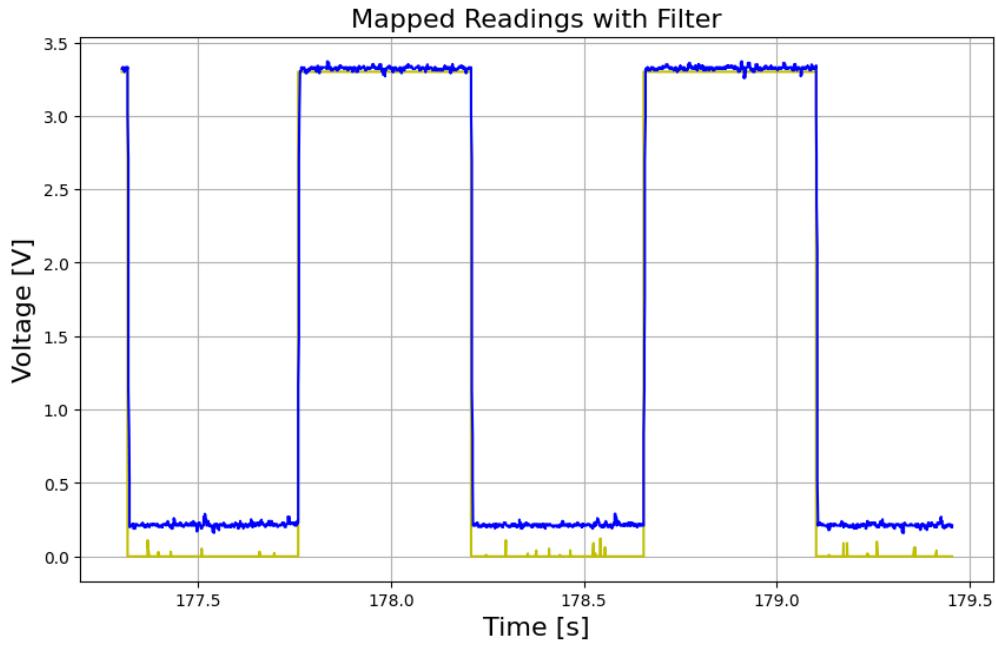


FIGURE 53: BLOCK WAVE RESPONSE

These results indicate that the linear mapping of the ADC values will provide accurate readings to the user in most scenarios. The following three figures give the Fourier transform of the signals generated at 2 kHz, 20 kHz, and 200 kHz. The “Original Signal” in each figure represents the input signal from the signal generator. The “Attenuated Signal” represents the signal passed by the RC filter and the analogue circuit.

These figures provide evidence that the first-order low-pass filter behaves as expected. The signal is substantially attenuated with every decade. Figure 56 illustrates that the passed signal’s magnitude is near zero. This indicates that the RC filter will filter out unwanted high frequencies such as RF signals. Appendix F gives the voltage waveforms of the signals generated for this test.

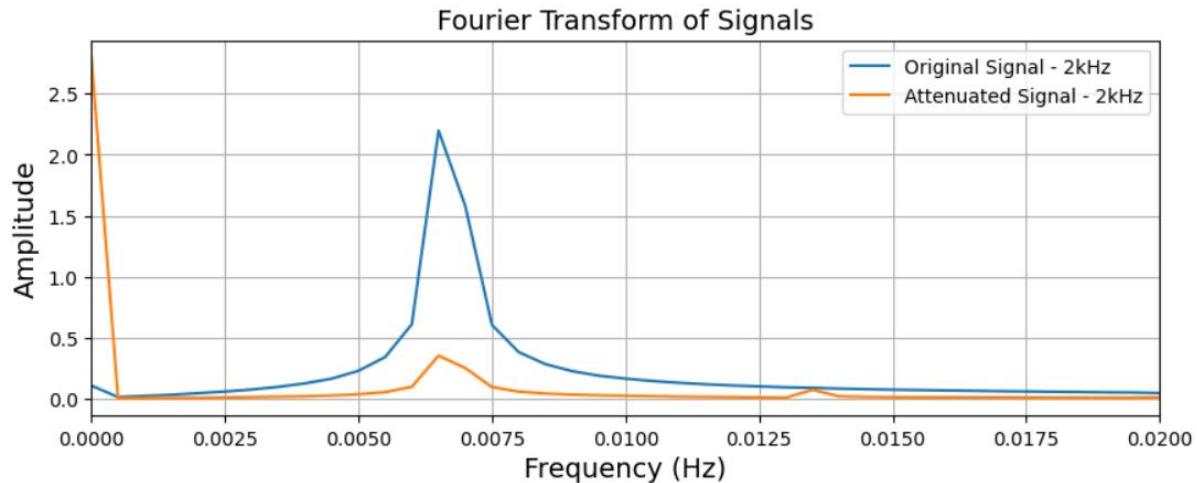


FIGURE 54: FOURIER TRANSFORM OF 2 KHz SIGNAL

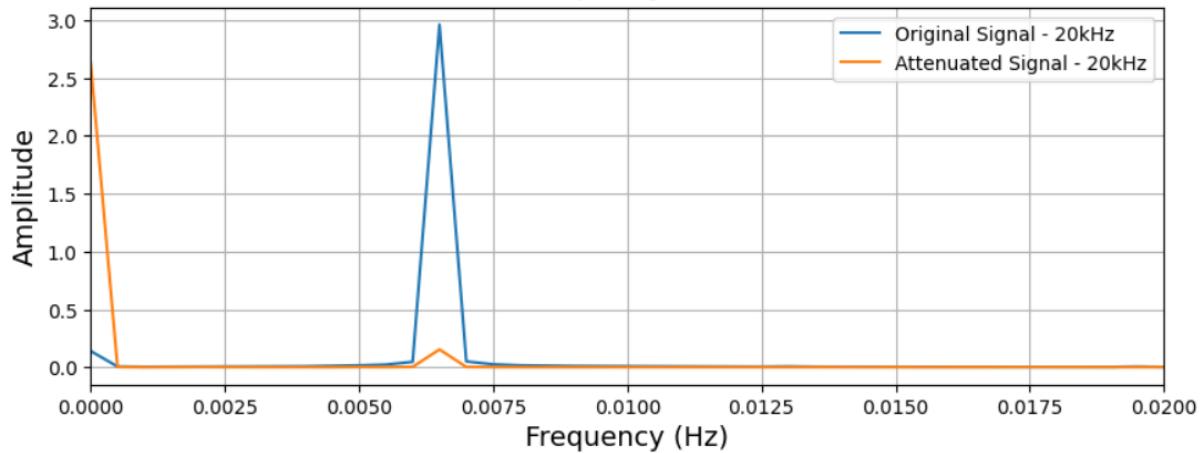


FIGURE 55: FOURIER TRANSFORM OF 20 KHz SIGNAL

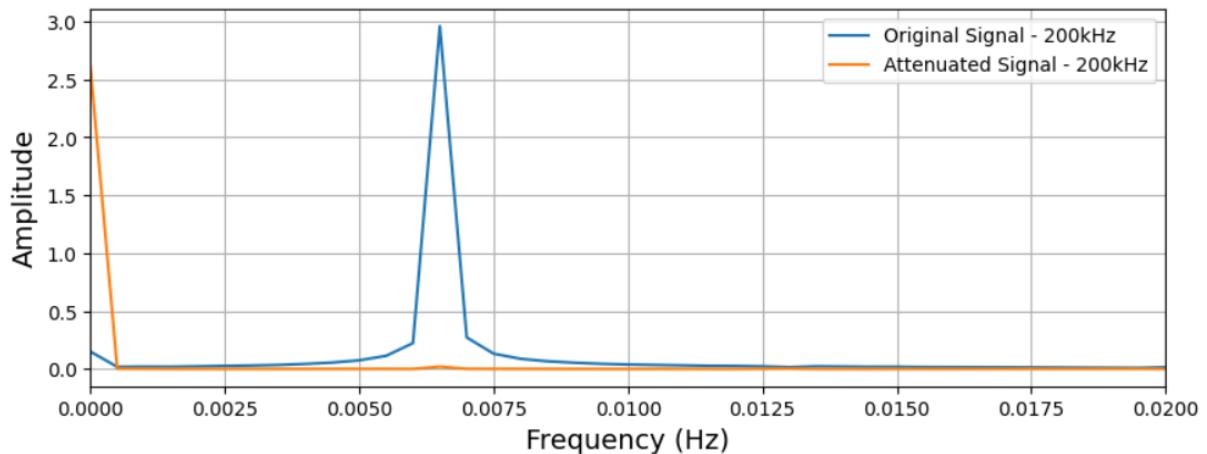


FIGURE 56: FOURIER TRANSFORM OF 200 KHz SIGNAL

6.5 Firmware and software

The firmware and software were tested with tools such as the serial monitor within the Arduino IDE and the alert functions within Javascript. The transfer and saving of the channel and event data was tested by printing the information within each vector to the serial monitor. Through visual inspection, the correct operation of the code was determined. Various hardware solutions were used to test the software and firmware as well. LEDs was used to represent outputs, and potentiometers was used to represent analogue inputs such as sensors.

The channels' operation was tested by creating new thresholds and tuning the potentiometer. The behaviour of the output pin was observed. The RTC events was also tested in a similar manner. The calculation of the new date was verified by printing the new date calculated into the serial monitor. Diverse scenarios, involving adding days within the same month, the next month, the next six months, and the next year, were systematically tested. The operation of saving the data to the SPIFFS was tested by turning the system off and on and verifying if the previous data was present. All conducted tests yielded successful outcomes, affirming the reliability of the implemented functionalities.

Chapter 7: Improvements and conclusion

7.1 Improvements

The following is a list of improvements that will be made to the hardware, firmware, and software:

- Include a ground plane in the bottom layer of the PCB, in addition to the ground plane in the top layer. This will reduce the signal interference in the PCB.
- Add a surface mount $10\ k\Omega$ pull-down resistor to the base of each transistor.
- Replace the through-hole transistors with surface mount transistors. This adjustment will improve the professional appearance of the PCB.
- Modify the analogue protection circuit to clamp the signal closer to the 0 to 3.3 V range. This will provide better resolution and accuracy.
- Adjust the RC low-pass filter to have a steeper roll-off at 1 kHz.
- Implement the deep-sleep mode of the microcontroller to reduce current consumption.
- Integrate the WiFi-toggling circuit with a GPIO pin linked to the RTC peripheral to facilitate waking the microcontroller from deep sleep. Currently, the circuit is connected to GPIO pin 10, which is not associated with the RTC peripheral, hindering the manual wake-up capability from deep sleep.
- Configure the ADCs to wake the microcontroller from deep sleep mode when a user-defined threshold is reached.
- Implement an ESP32 development board with more GPIO pins to avoid using the strapping pins. This will eliminate the requirement to press the reset button on the board before the flashed program is processed. Currently, the strapping pins are pulled low by the pull-down resistors on the base of the transistor. However, they must be pulled high for the microcontroller to boot properly and process the program.
- Add the feature such that the microcontroller can behave as a station and connect to the internet. This will allow the user to modify settings from any location with internet access.
- Implement the master/slave communication system. This system will enable the user to connect units together and then program just the master unit. The master unit will communicate the settings to the slave units. A screw terminal was already implemented for this application which is connected to the UART pins of the microcontroller.
- Make the cost per unit cheaper by ordering components from vendors in South Africa. This will reduce the customs/import duty tax.
- Acquire a weatherproof container for the hardware. A IP67 rated container will be sufficient.
- Add the feature to convert analogue measurements to a user-defined unit.
- Perform tests on the system on a farm with actual agricultural related inputs and outputs for several months.
- Make the website responsive for any screen. As of this moment the layout of the website will only display correctly on mobile smartphones.

7.2 Conclusion

An artifact was developed which satisfies the requirements defined in the problem statement. Positioned as a cost-effective alternative, this product surpasses competing market options in terms of both modularity and features. Therefore, this project will fill the gap created in the agricultural sector for an affordable and dynamic irrigation management system. This project exhibits great potential within the agricultural sector. Furthermore, the modularity of the final deliverable enables it to be implemented with several use cases outside of the agricultural sector. This includes residential applications, such as residential irrigation management.

The open-source design files can be found at the following link:

<https://github.com/GertLC/G-Tronic-Final.git>

References

- [1] V. Dhawan, "Water and Agriculture in India," *Background paper for the South Asia expert panel during the Global Forum for Food and Agriculture*, vol. 28, pp. 80-85, 2017.
- [2] A. Gupta, H. P. Gupta, P. Kumari, R. Mishra, S. Saraswat and T. Dutta, "A Real-Time Precision Agriculture Monitoring System using Mobile Sink in WSN's," IEEE, 2018.
- [3] T. Ojha, S. Misra and N. S. Raghuwanshi, "Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges," *Computer and Electronics in Agriculture*, vol. 118, no. 2015, pp. 66-84, 2015.
- [4] D. Dassanayake, K. Dassanayake, H. Malano, G. Dunn, P. Douglas and J. Langford, "Water saving through smarter irrigation in Australian dairy farming: Use of intelligent irrigation controller and wireless network," Melbourne, 2009.
- [5] Netafim, "Precision Irrigation," [Online]. Available: <https://www.netafim.co.za/precision-Irrigation/>. [Accessed 26 February 2023].
- [6] Centurion Systems, "G-Ultra," [Online]. Available: <https://www.centsys.co.za/g-ultra/>. [Accessed 26 February 2023].
- [7] Agrico, "Agrico Web Control," [Online]. Available: <https://www.agrico.co.za/agrico-web-control/>. [Accessed 26 February 2023].
- [8] Reinke, "RPM Controls," [Online]. Available: <https://www.reinke.com/rpm-controls.html>. [Accessed 26 February 2023].
- [9] S. Dhawan, "Analogy of Promising Wireless Technologies on Different Frequencies: Bluetooth, WiFi, and WiMAX," IEEE, 2007.
- [10] Mouser Electronics, "Espressif Systems ESP32-C3-DevKitM-1," [Online]. Available: <https://www.mouser.co.za/ProductDetail/Espressif-Systems/ESP32-C3-DevKitM-1?qs=sGAEpiMZZMuqBwn8WqcFUj1SFkunHY10w%252BZIYlgtmjYhyFS3YBb2rg%3D%3D>. [Accessed 19 July 2023].
- [11] Mouser Electronics, "Arduino ABX00033," [Online]. Available: <https://www.mouser.co.za/ProductDetail/Arduino/ABX00033?qs=PzGy0jfSMueXyfBfl2XSA%3D%3D>. [Accessed 19 July 2023].
- [12] Mouser, "MOD-WIFI-ESP8266," [Online]. Available: <https://www.mouser.co.za/ProductDetail/Olimex-Ltd/MOD-WIFI-ESP8266?qs=QGk6feVlqMKKopUWzCH4tA%3D%3D>. [Accessed 20 July 2023].
- [13] Espressif Systems, "ESP32-C3 Series Datasheet v1.4," 2020.
- [14] Espressif Systems, "ESP32-C3-MINI-1 & MINI-1U Datasheet v1.3," 2021.
- [15] Espressif Systems, "ESP8684 Series Datasheet v1.2," 2022.
- [16] Espressif Systems, "ESP8684-MINI-1 & ESP8684-MINI-1U Datasheet v0.5," 2021.
- [17] Espressif Systems, "ESP32-S3 Series Datasheet v1.6," 2021.
- [18] Espressif Systems, "ESP32-S3-WROOM-2 Series Datasheet v1.1," 2021.
- [19] Texas Instruments, "ADC Input Protection," 2013.
- [20] T. Williams, *The Circuit Designer's Companion*, Elsevier, 2005.
- [21] J. W. Wilson and S. A. Riedel, *Electric Circuits*, Pearson.
- [22] J. Park, *Practical Embedded Controllers: Design and Troubleshooting with the Motorola 68HC11*, Elsevier, 2003.
- [23] Linear Technology, "LT8610 Liner".

- [24] Mouser Electronics, “NUCLEO-WB55RG,” [Online]. Available: <https://www.mouser.co.za/ProductDetail/STMicroelectronics/NUCLEO-WB55RG?qs=HBWAp0VN4Rhz3AmVigfuHg%3D%3D>. [Accessed 19 July 2023].
- [25] Mouser Electronics, “ESP32-S3-DevKitC-1-N32R8V,” [Online]. Available: <https://www.mouser.co.za/ProductDetail/Espresif-Systems/ESP32-S3-DevKitC-1-N32R8V?qs=sGAEpiMZZMuqBwn8WqcFUipNguezRlc4GzmkN8krWGEfV9T8moekcA%3D%3D>. [Accessed 19 July 2023].
- [26] Espressif, “ESP32-S3-DevKitC-1 Dimensions,” [Online]. Available: https://dl.espressif.com/dl/DXF_ESP32-S3-DevKitC-1_V1_20210312CB.pdf. [Accessed 19 July 2023].
- [27] STMicroelectronics, “STM32WB Nucleo-64 board (MB1641) - User manual,” STMicroelectronics, 2021.

Appendix A – Project management

Risk management

Table 14 identifies the risks involved with this project. This table also gives possible mitigations for the identified risks. The mitigations proposed will be put in place to avoid possible project failure.

TABLE 14: RISK MANAGEMENT

Risk	Possible Mitigation
Schedule Risk – Falling behind on schedule.	Do risk mitigation. Plan for loadshedding. Create a project schedule.
Knowledge Risk – Insufficient knowledge of certain subjects can lead to low quality deliverables.	Do thorough research. Concentrate during development.
Technical Risk – Components can be faulty. Components can malfunction as well.	Order at least two units of each component. Perform sufficient testing procedures. Develop at least two prototypes for demonstration purposes.
Logistic Risk – Hardware components do not arrive on time.	Check the availability of components during the design process. Check the estimated transport time before ordering components.

Project schedule

Figure 57 displays the work schedule for this project. This schedule also displays the milestones for the project. This schedule also gives certain tasks that will be completed throughout this year.

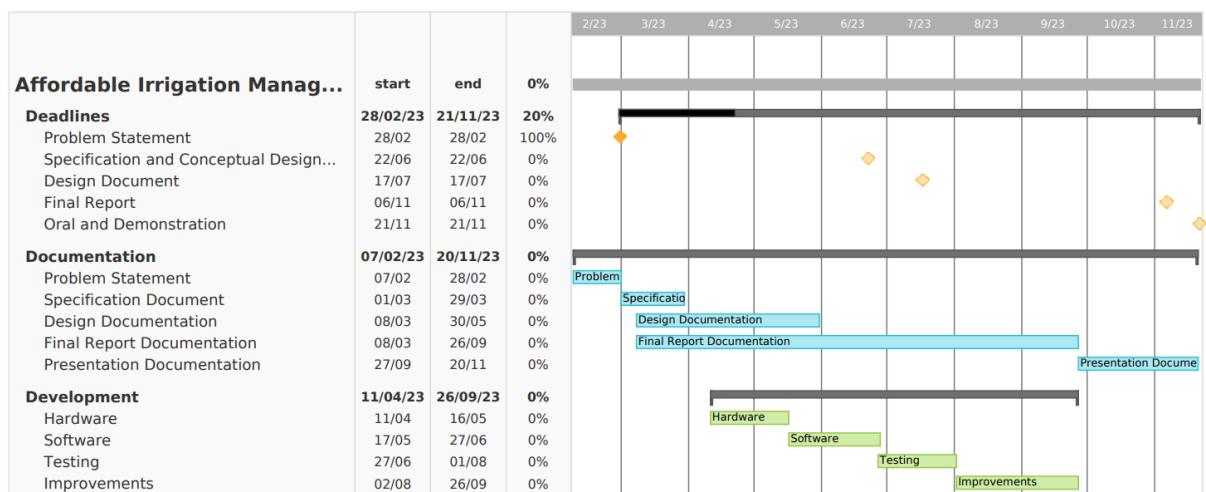


FIGURE 57: PROJECT TIME SCHEDULE

Second semester schedule

Table 15 gives the planned schedule for the second semester.

TABLE 15: SECOND SEMESTER TIME SCHEDULE

Task	Start Date	End Date
Hardware Detailed Design	24/07/2023	30/07/2023
Order Parts	31/07/2023	06/08/2023
Software and Firmware Detailed Design	07/08/2023	13/08/2023
Implement Software and Firmware	14/08/2023	03/09/2023
Construct Hardware	04/09/2023	10/09/2023
Tests and Evaluation	11/09/2023	02/10/2023
Finish Documentation	03/10/2023	15/10/2023
Demonstration Preparation	16/10/2023	01/11/2023

Appendix B – Website states

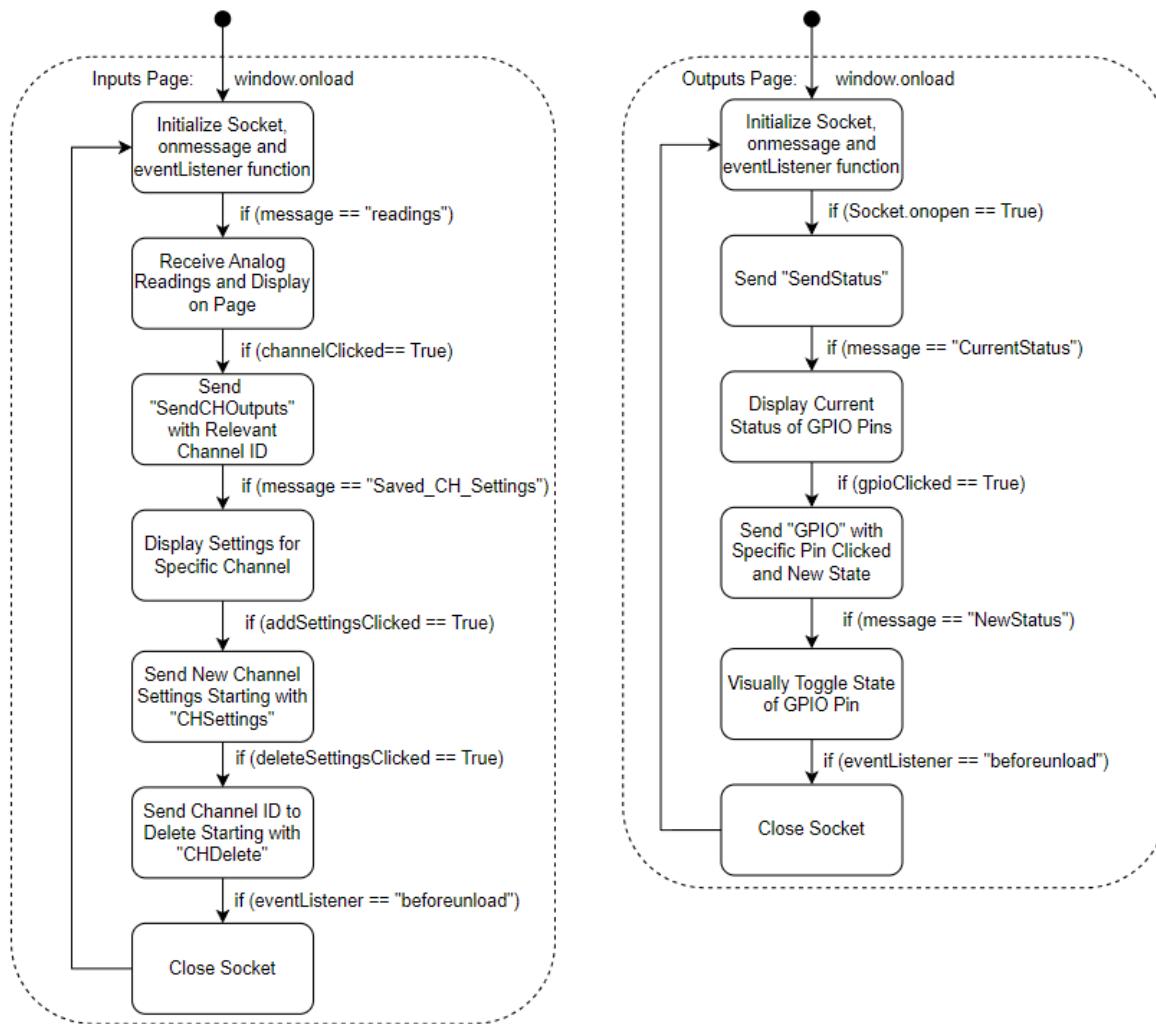


FIGURE 58: INPUT AND OUTPUT PAGE STATES

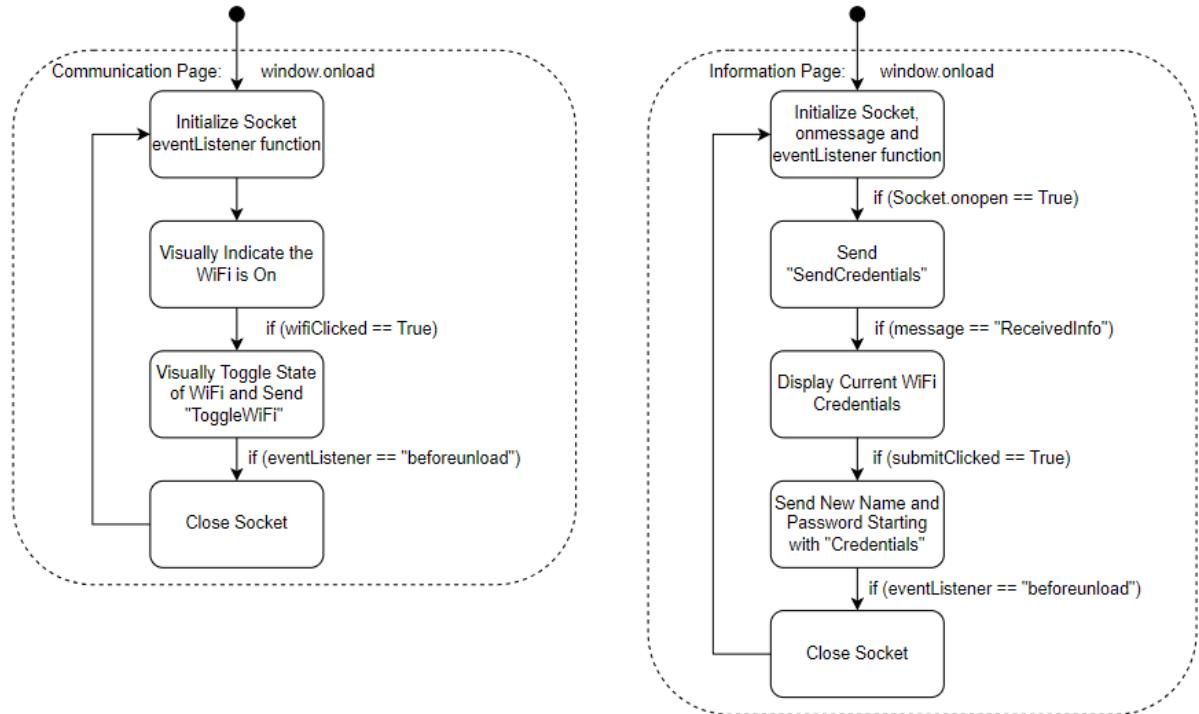


FIGURE 59: COMMUNICATION AND INFORMATION PAGE STATES

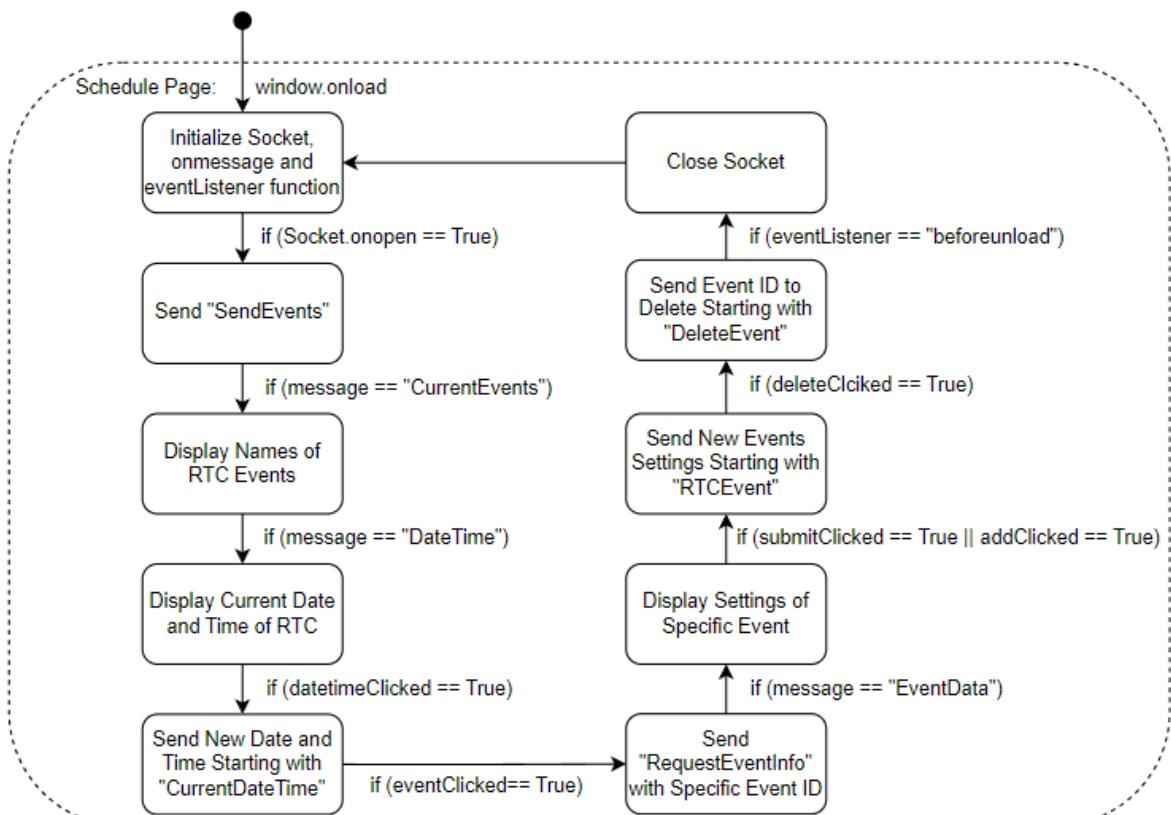


FIGURE 60: SCHEDULE PAGE STATES

Appendix C – Website UI design

Navigation menu

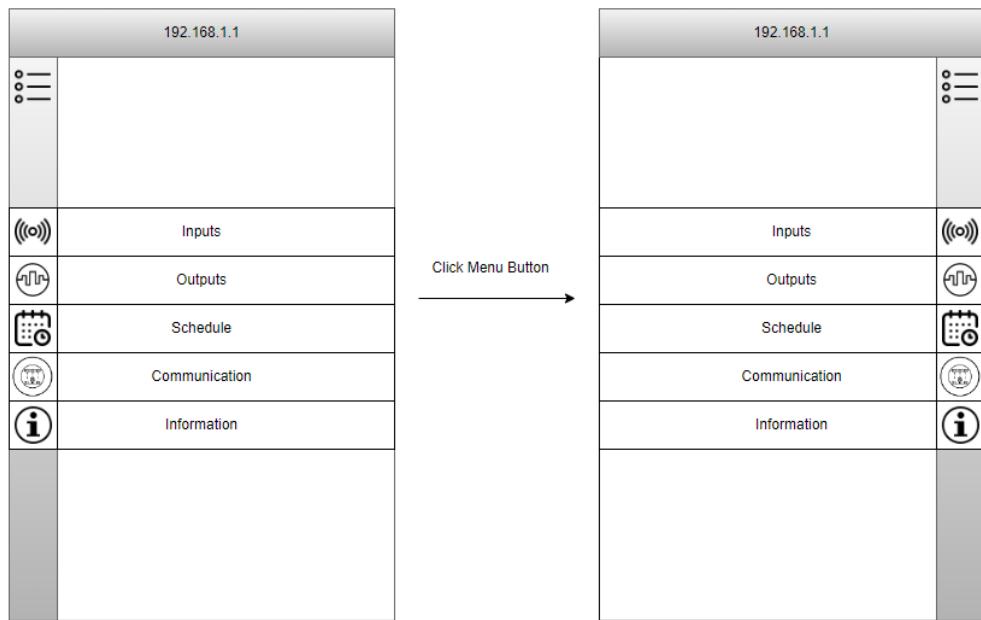


FIGURE 61: FIRST MENU DESIGN

Analogue inputs

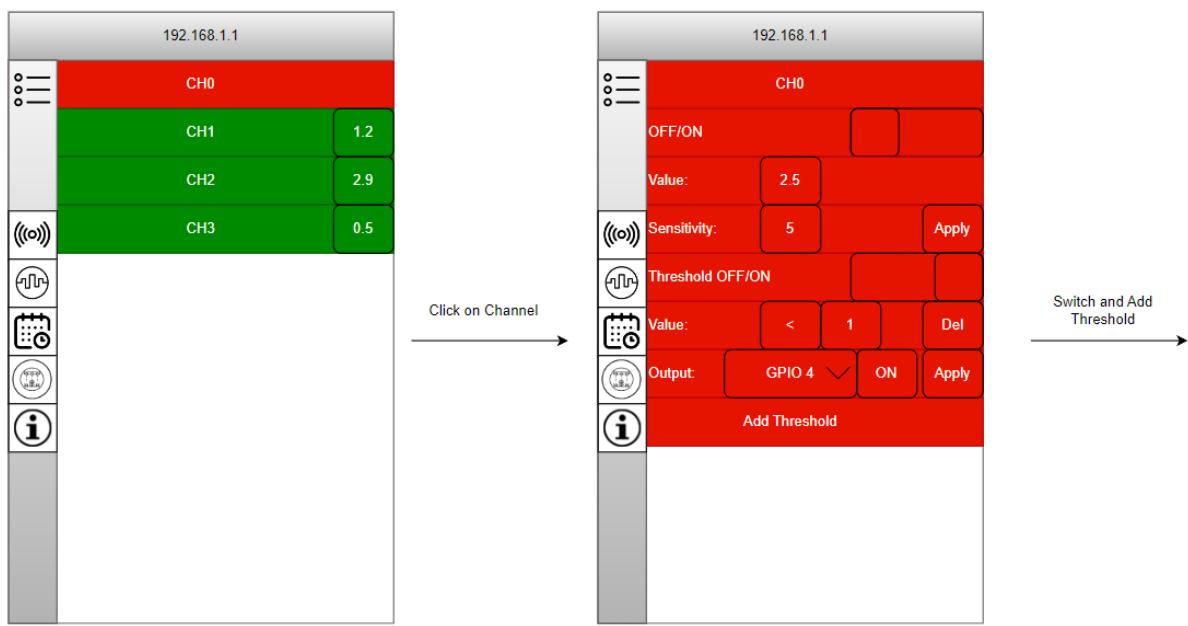


FIGURE 62: FIRST ANALOGUE INPUT THRESHOLD CREATION DESIGN

192.168.1.1

CH0

OFF/ON

Value: 2.5

Sensitivity: 5

Threshold OFF/ON

Value: < 1

Output: GPIO 4 ON

Value: > 3

Output: GPIO 4 OFF

Add Threshold

Fill in Blanks And Click Apply

192.168.1.1

CH0

OFF/ON

Value: 2.5

Sensitivity: 5

Threshold OFF/ON

Value: < 1

Output: GPIO 4 ON

Value: > 3

Output: GPIO 4 OFF

Add Threshold

FIGURE 63: FIRST ANALOGUE INPUT THRESHOLD SUBMISSION DESIGN

Digital outputs

192.168.1.1

GPIO 4

GPIO 5

GPIO 6

GPIO 7

GPIO 8

GPIO 9

GPIO 18

GPIO 19

Click Expand Button

192.168.1.1

GPIO 4

OFF/ON

GPIO 5

GPIO 6

GPIO 7

GPIO 8

GPIO 9

GPIO 18

GPIO 19

Switch

FIGURE 64: FIRST DIGITAL OUTPUT DESIGN

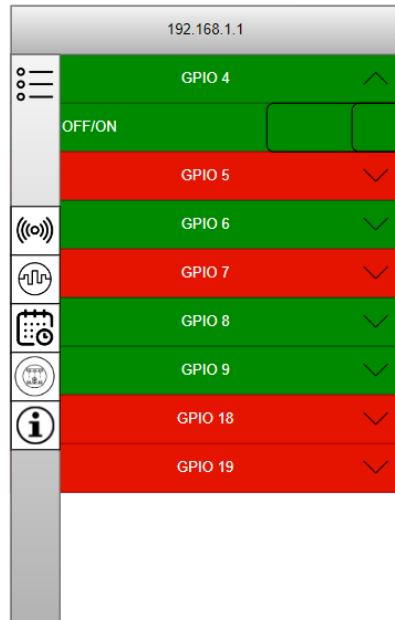


FIGURE 65: FIRST DIGITAL OUTPUT SWITCHING DESIGN

Schedule

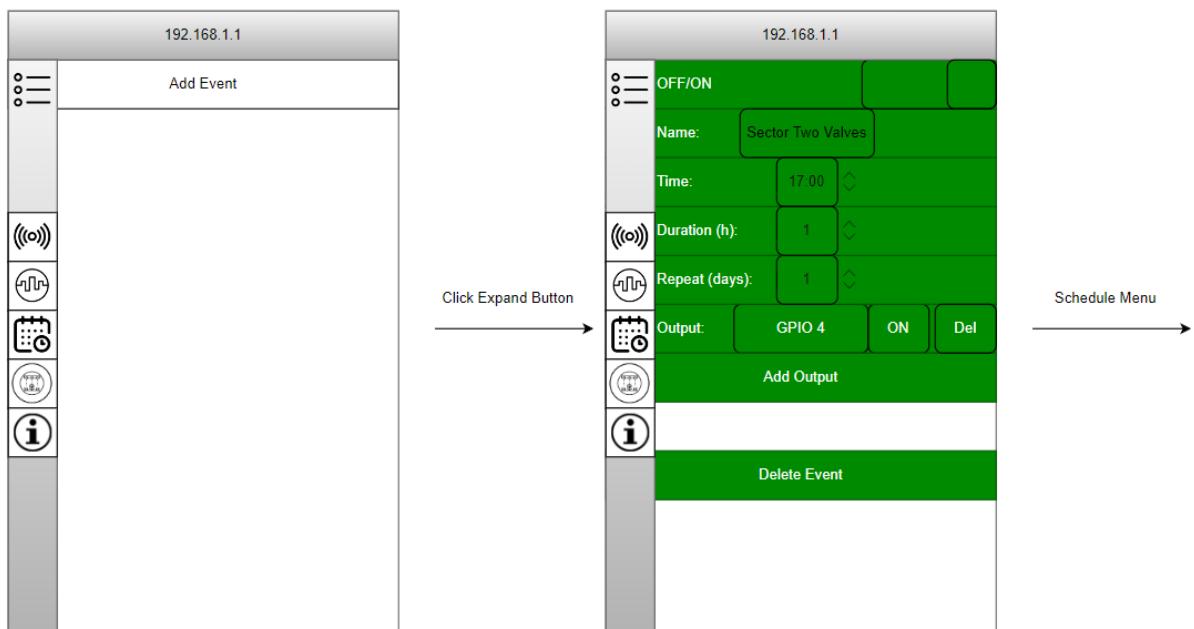


FIGURE 66: FIRST SCHEDULE EVENT CREATION DESIGN

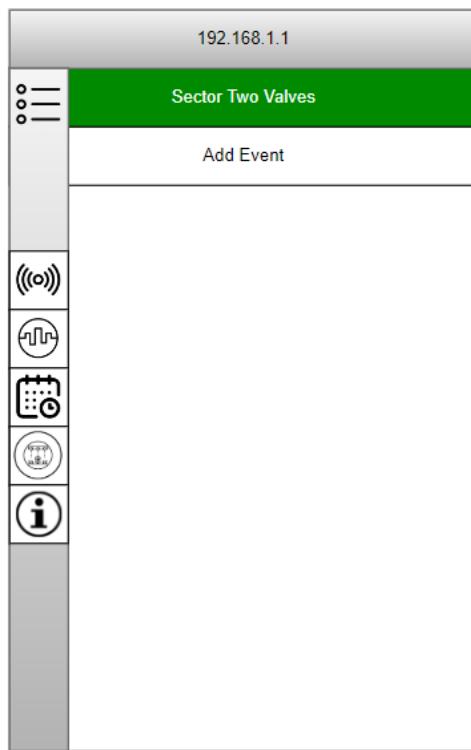


FIGURE 67: FIRST SCHEDULE EVENT SUBMISSION DESIGN

Communication

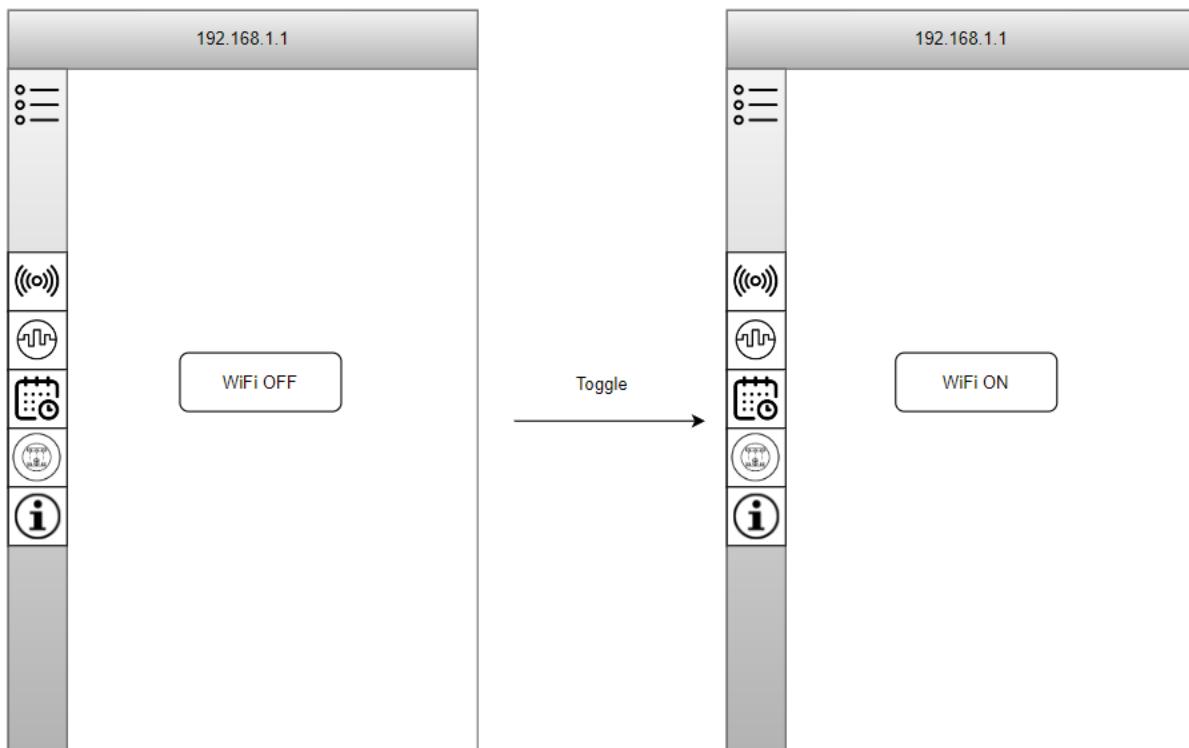


FIGURE 68: FIRST COMMUNICATION PAGE DESIGN

Information

192.168.1.1	
☰	
SSID	Name: ESP32_AP
Lock	Password: 01234567
Apply	
Help	
Info	

FIGURE 69: FIRST INFORMATION PAGE DESIGN

Appendix D – Website implementation

Navigation menu

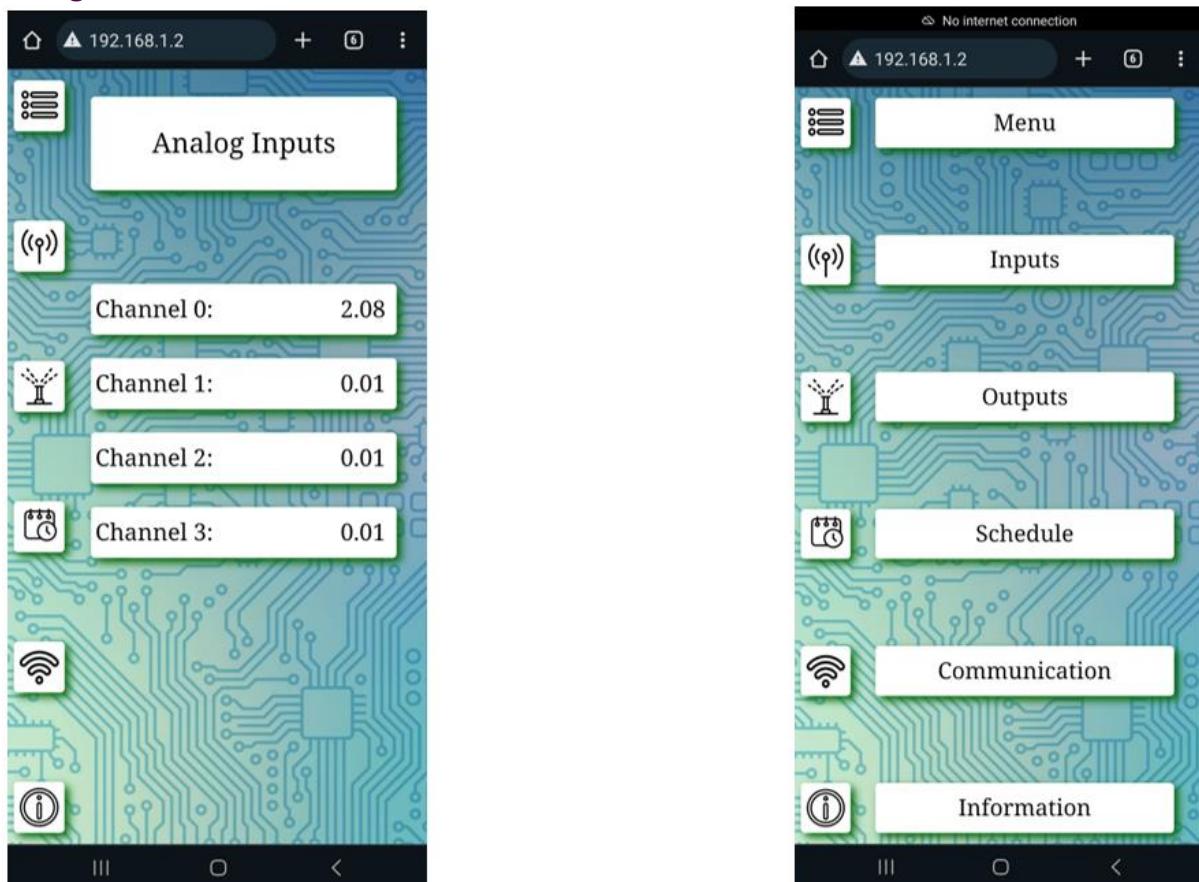


FIGURE 70: FINAL MENU DESIGN

Analogue inputs

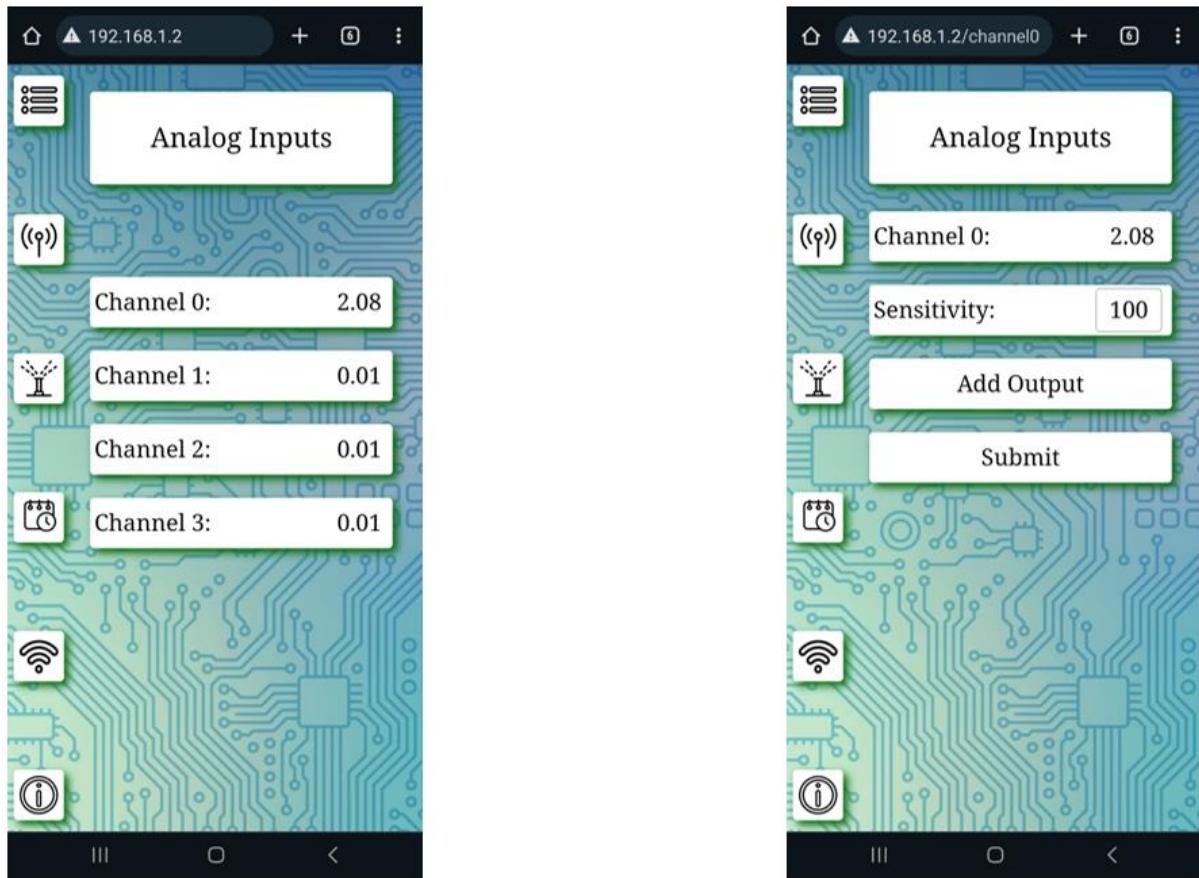


FIGURE 71: FINAL ANALOGUE INPUT CHANNEL VIEW DESIGN

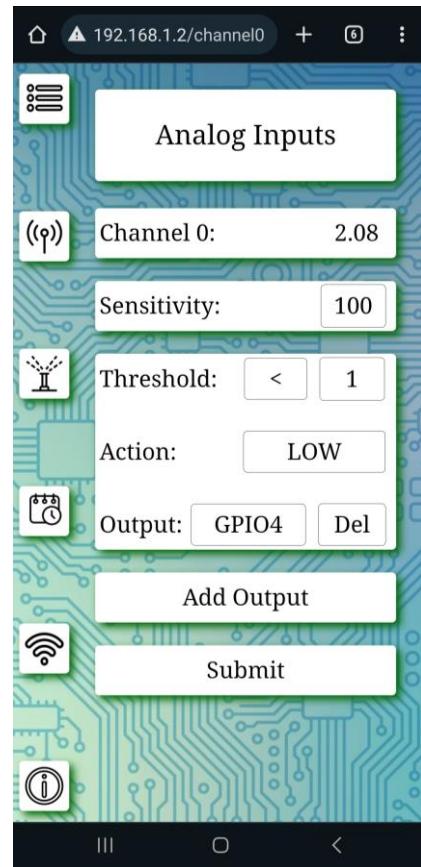


FIGURE 72: FINAL ANALOGUE INPUT THRESHOLD SUBMISSION DESIGN

Digital outputs

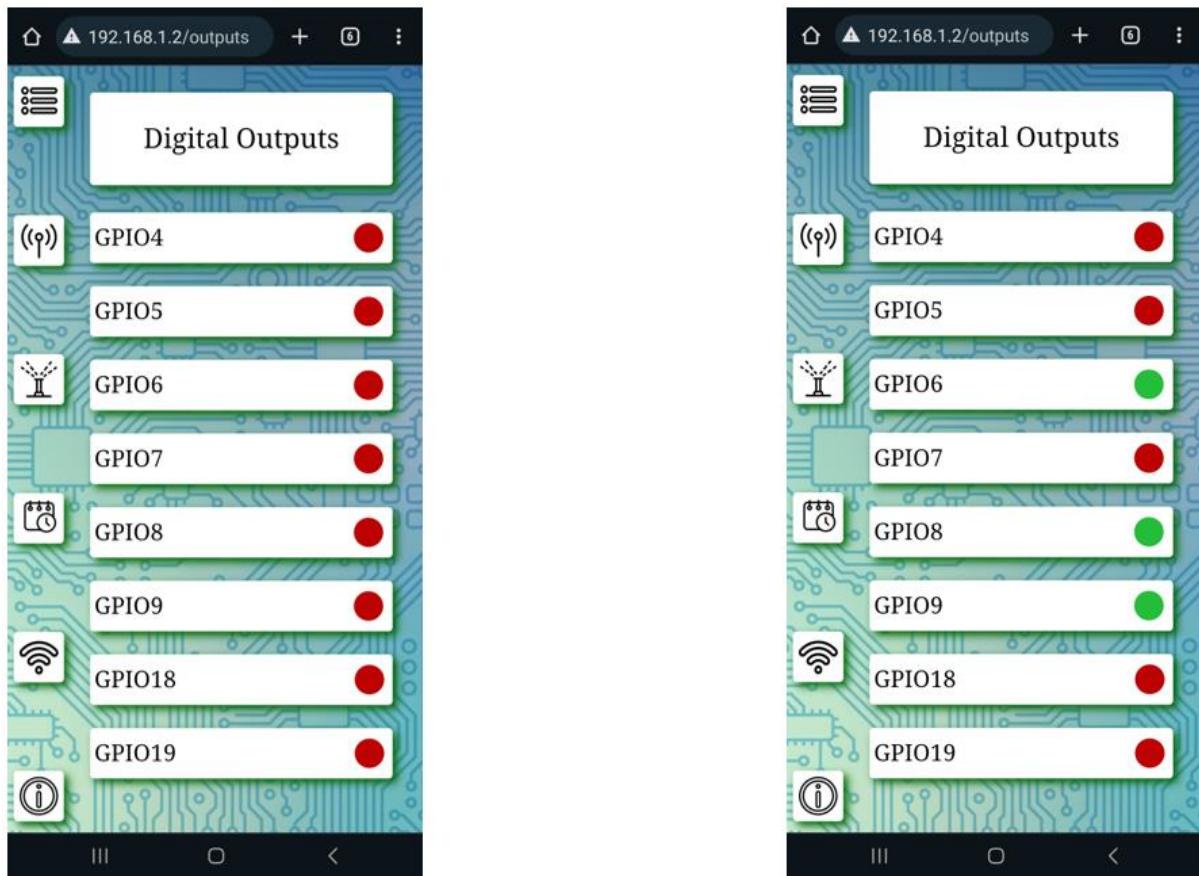


FIGURE 73: FINAL DIGITAL OUTPUTS PAGE DESIGN

Schedule

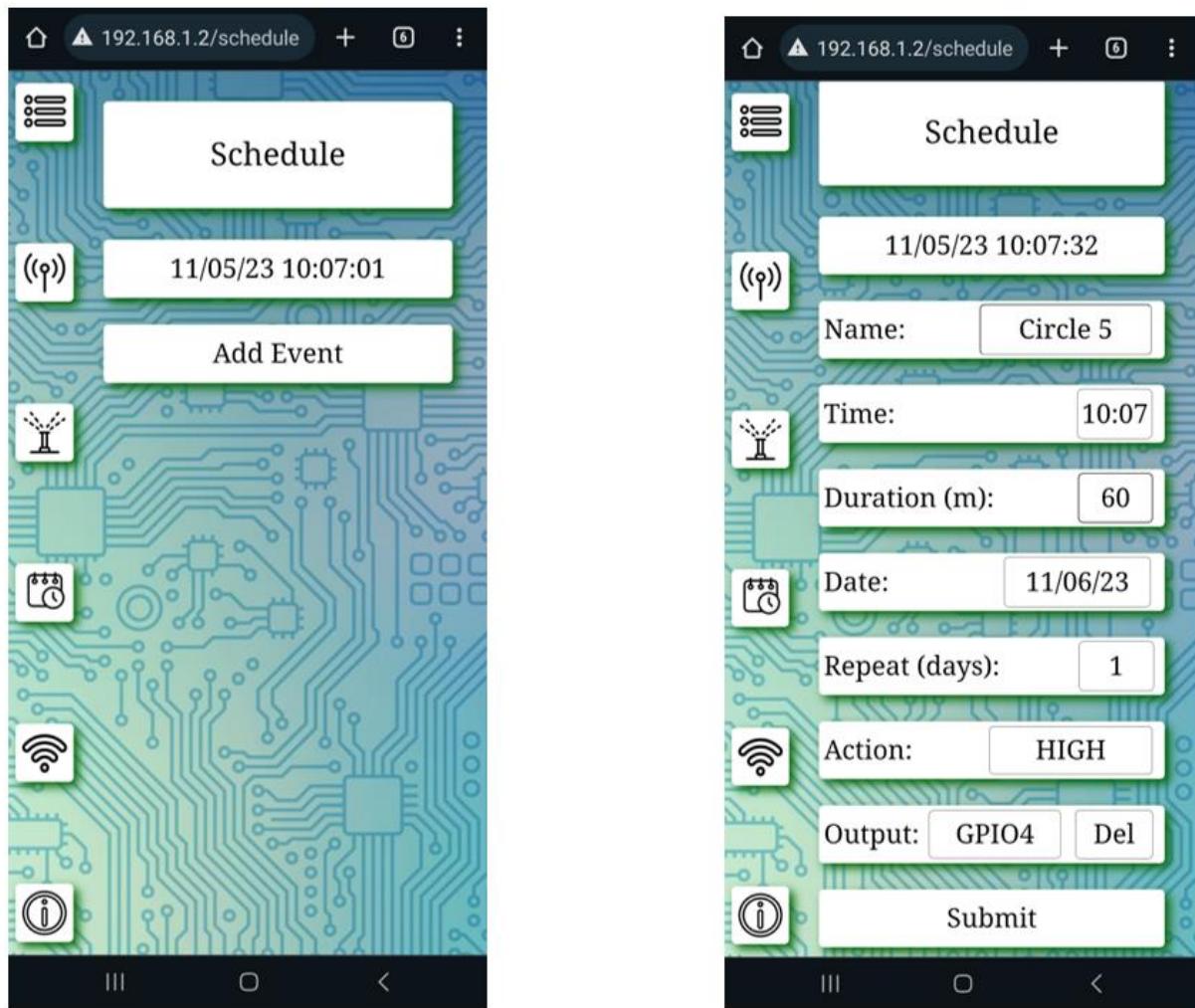


FIGURE 74: FINAL SCHEDULE EVENT CREATION DESIGN

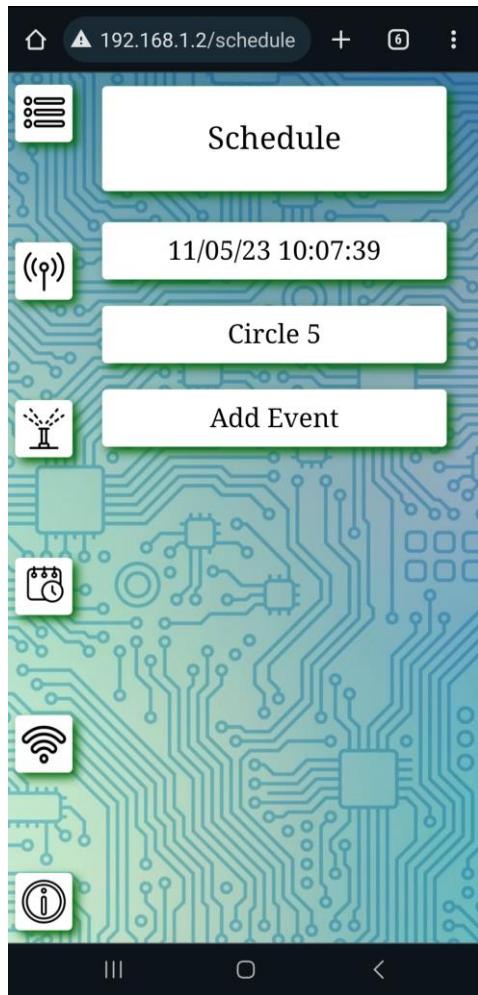


FIGURE 75: FINAL SCHEDULE EVENT SUBMISSION DESIGN

Communication

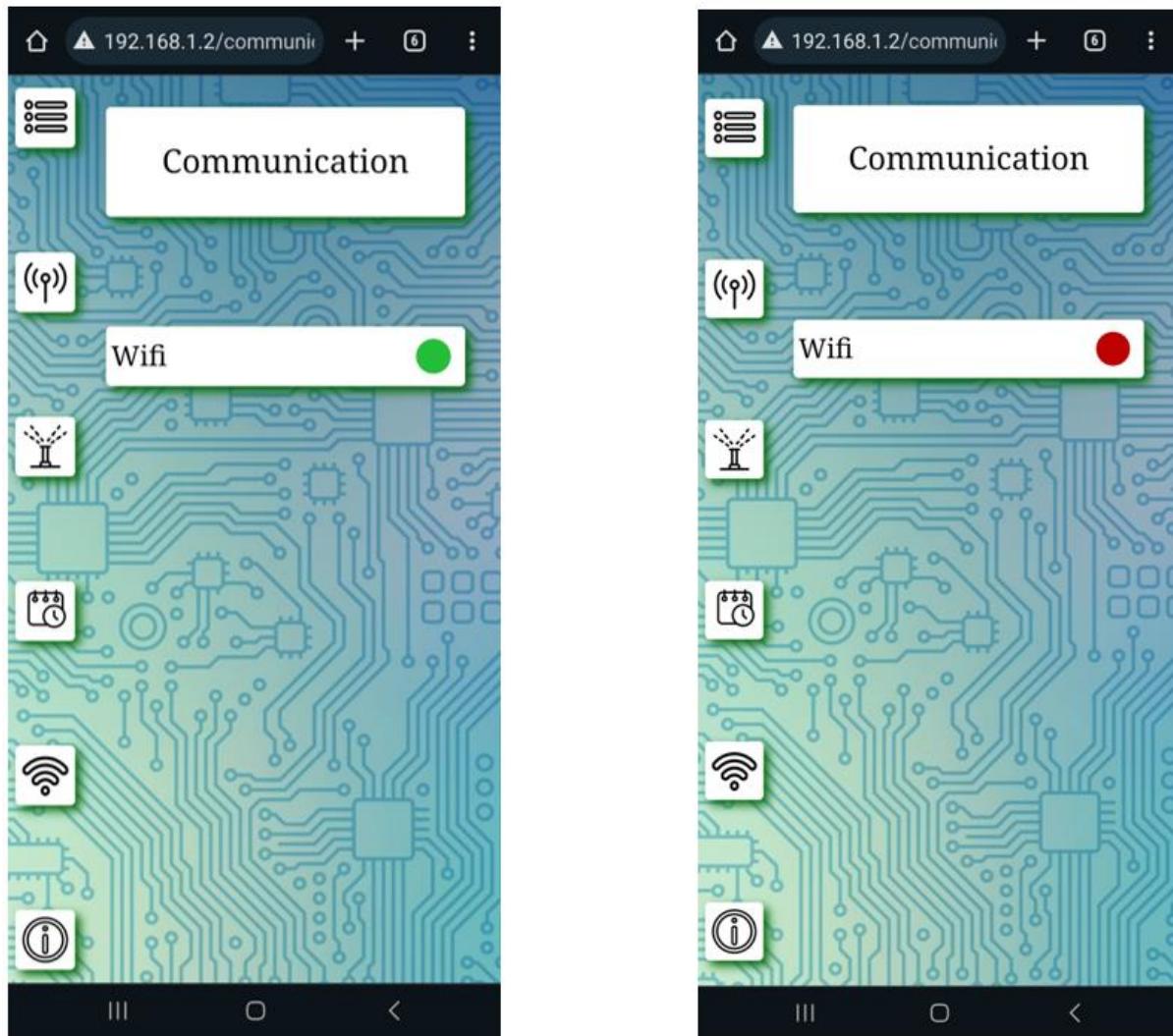


FIGURE 76: FINAL COMMUNICATION PAGE DESIGN

Information

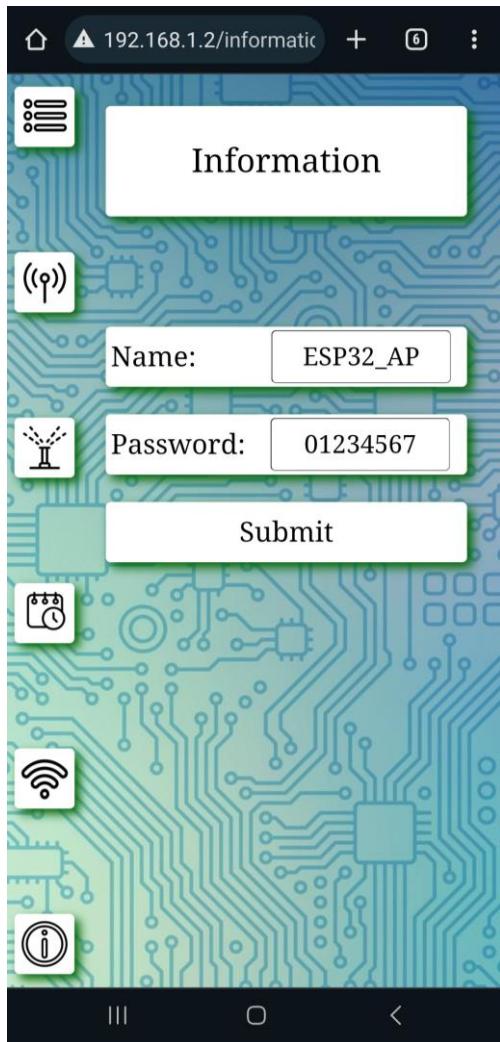


FIGURE 77: FINAL INFORMATION PAGE DESIGN

Appendix E – Bill of Materials (BOM)

Hardware

Vendor	Manufacture Part Number	Manufacturer	Description	Order Qty.	Unit Price [S]	Order Price [S]	Unit Price [R]	Order Price [R]
LCSC	CKPK0810-5uH/L-U3.5	CENKER	5uH ±15% Plugin,D10xL14mm	30	0.064	1.91	1.21	36.11
LCSC	SRD-05VDC-SL-C	Ningbo Songle Relay	5V 1SA SPDT (1 Form C) S Plug	80	0.228	18.21	4.30	344.25
LCSC	ABS210	YONGYUTAI	50A 1V@2A 2A 1kV SOP-4 Brid	20	0.027	0.53	0.50	10.02
LCSC	WJ126V-5.0-2P	Ningbo Kangnex Elec	1x2P 18A -40°~+105° 250V Gre	20	0.068	1.35	1.28	25.52
LCSC	WJ126V-5.0-3P	Ningbo Kangnex Elec	1x3P 18A -40°~+105° 250V Gre	130	0.085	11	1.60	207.95
LCSC	KM108MR050J25RR0VH2FP0	CX(Dongguan Chengxing Elec	1000uF 50V ±20% Plugin,D13xL	10	0.142	1.42	2.69	26.84
LCSC	CC0201KRX5R6BB102	YAGEO	10V 1nF X5R ±10% 0201 Multi	100	0.001	0.13	0.02	2.46
LCSC	CS2012X5R475K500NRE	Samwha Capacitor	50V 4.7uF X5R ±10% 0805 Multi	20	0.020	0.4	0.38	7.56
LCSC	0402X105K100CT	Walsin Tech Corp	10V 1uF X5R ±10% 0402 Multi	100	0.002	0.18	0.03	3.40
LCSC	C0603C100K5RACAUTO	KEMET	50V 10pF X7R ±10% 0603 Multi	50	0.017	0.87	0.33	16.45
LCSC	GRM31CR61A476ME15L	Murata Electronics	10V 47uF X5R ±20% 1206 Multi	10	0.061	0.61	1.16	11.53
LCSC	CC0402KRX5R9BB333	YAGEO	50V 33nF X5R ±10% 0402 Multi	100	0.003	0.27	0.05	5.10
LCSC	CR0402FF4122G	LIZ Elec	62.5mW Thick Film Resistors ±1	100	0.001	0.06	0.01	1.13
LCSC	WF04H1004DTL	Walsin Tech Corp	125mW ±100ppm/? ±0.5% 1MΩ	100	0.004	0.35	0.07	6.62
LCSC	RTT052373FTP	RALEC	125mW Thick Film Resistors ±1	100	0.001	0.14	0.03	2.65
LCSC	AC1210FR-074K64L	YAGEO	500mW Thick Film Resistors ±1	50	0.011	0.53	0.20	10.02
LCSC	1206W4F9091TSE	UNI-ROYAL(Uniroyal Elec)	250mW Thick Film Resistors ±1	50	0.002	0.11	0.04	2.08
LCSC	RTT02102JTH	RALEC	62.5mW Thick Film Resistors ±1	100	0.000	0.04	0.01	0.76
LCSC	SMD1206B050TF/15	Brightking	15V 500mA 100A 1A 1206 Res	20	0.025	0.5	0.47	9.45
LCSC	1N4007	YONGYUTAI	900mA 1.1V@1A 1kV SOD-123	100	0.005	0.54	0.10	10.21
LCSC	2N2222A	Foshan Blue Rocket Elec	40V 625mW 100@150mA,10V	80	0.015	1.22	0.29	23.06
LCSC	T54543CJ 250gf 009	SHOU HAN	No NO 50mA 100MΩ 100000 Ti	50	0.008	0.38	0.14	7.18
LCSC	XL-16085URC-06	XINGLIGHT	Colorless transparency -30°~+40°	100	0.003	0.28	0.05	5.29
LCSC	0805W8F470JT5E	UNI-ROYAL(Uniroyal Elec)	125mW Thick Film Resistors ±1	100	0.002	0.18	0.03	3.40
LCSC	12251103CNG05115001	JILIN	3A Straight Square Pins 2.5mm	50	0.011	0.54	0.20	10.21
LCSC	F254D-02-PT-B	XFCN	Open Top 3A 2.54mm P=2.54m	50	0.012	0.62	0.23	11.72
LCSC	DD23V3BSF-7	Diodes Incorporated	±3% 3.32V~3.53V 500mW 3.43	80	0.037	2.94	0.70	55.58
LCSC	TCC0402X5R104M250AT	CCTC	25V 100nF X5R ±20% 0402 Mu	100	0.001	0.08	0.02	1.51
LCSC		LCSC	Shipping			103.29		1952.18
JLPCB		JLPCB	Printed Circuit Board with Asse	10	9.073	90.73	171.48	1714.80
JLPCB		JLPCB	Shipping			70.61		1334.53
DHL			Import Duty/Tax from LCSC					1805.90
DHL			Import Duty/Tax from JLPCB					479.60
Mantech	MF25-10K-F		Resistor 1/4W RND M/F 1% 10	80			0.30	24.00
Mantech	190881		Socket SiL STR Housed 2.54 40V	5			4.44	22.20
Nuvision Electronics		Espressif	1965-ESP32-C3-DEVKITM-1	10			184.10	1841.00
Nuvision Electronics			Shipping					150.00
Nuvision Electronics			VAT (15%)		2005.00	9.93	310.02	376.46
								10471.93

FIGURE 78: HARDWARE BILL OF MATERIALS

Appendix F – Test signals for RC low-pass filter



FIGURE 79: FILTER 2 kHz TEST SIGNAL



FIGURE 80: FILTER 20 KHz TEST SIGNAL

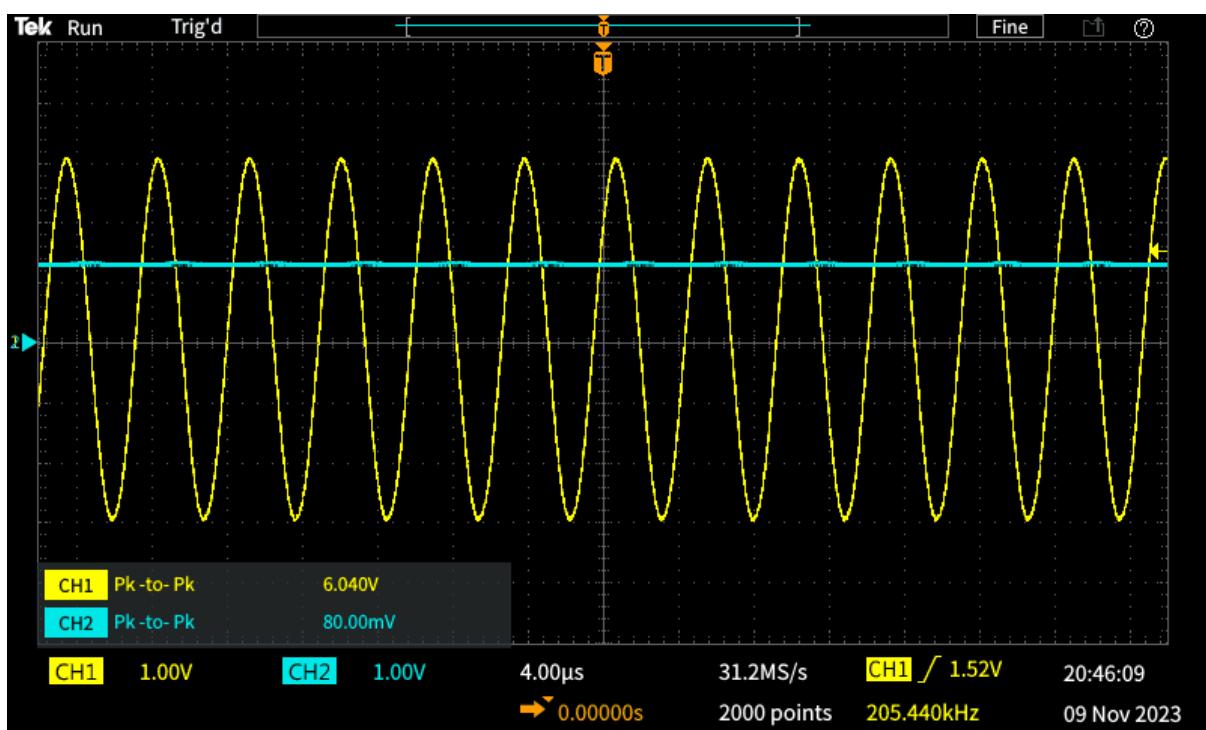


FIGURE 81: FILTER 200 KHz TEST SIGNAL