



UNIVERSITEIT•STELLENBOSCH•UNIVERSITY  
jou kennisvennoot • your knowledge partner

# Smart Braai Monitoring and Notification System

Estiaan Botha

21337985

Report submitted in partial fulfilment of the requirements of the module  
Project (E) 448 for the degree Baccalaureus in Engineering in the Department of  
Electrical and Electronic Engineering at Stellenbosch University.

Supervisor: Dr R. P. Theart

June 2025

# **Acknowledgements**

I would like to express my heartfelt gratitude to my parents for their unwavering support and encouragement throughout the years. Their belief in me has been the foundation of my journey. A sincere thank you to my supervisor, Dr R. P. Theart, for his guidance, insight, and support throughout the course of this project and the writing of this report. To my friends, and especially my girlfriend, thank you for standing by me and believing in me—especially during times when I struggled to believe in myself. Your support meant more than words can say.



UNIVERSITEIT•STELLENBOSCH•UNIVERSITY  
jou kennisvennoot • your knowledge partner

## Plagiaatverklaring / Plagiarism Declaration

1. Plagiaat is die oorneem en gebruik van die idees, materiaal en ander intellektuele eiendom van ander persone asof dit jou eie werk is.

*Plagiarism is the use of ideas, material and other intellectual property of another's work and to present it as my own.*

2. Ek erken dat die pleeg van plagiaat 'n strafbare oortreding is aangesien dit 'n vorm van diefstal is.

*I agree that plagiarism is a punishable offence because it constitutes theft.*

3. Ek verstaan ook dat direkte vertalings plagiaat is.

*I also understand that direct translations are plagiarism.*

4. Dienooreenkomsdig is alle aanhalings en bydraes vanuit enige bron (ingesluit die internet) volledig verwys (erken). Ek erken dat die woordelikse aanhaal van teks sonder aanhalingstekens (selfs al word die bron volledig erken) plagiaat is.

*Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism*

5. Ek verklaar dat die werk in hierdie skryfstuk vervat, behalwe waar anders aangedui, my eie oorspronklike werk is en dat ek dit nie vantevore in die geheel of gedeeltelik ingehandig het vir bepunting in hierdie module/werkstuk of 'n ander module/werkstuk nie.

*I declare that the work contained in this assignment, except where otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.*

21337985 Studentenommer / Student number	 Handtekening / Signature
EM Botha Voorletters en van / Initials and surname	03 June 2025 Datum / Date

# **Abstract**

## **English**

Cooking food over fire has always been a part of human history, and in South Africa, especially, it is rooted deep in our culture with a colourful history, coining the Afrikaans term ‘braai’. A problem many people face with braaing is the inconsistency thereof, relying heavily on intuition and experience. The aim of this project is to develop a wireless monitoring system that gives real-time temperature data, follows a cooking profile based on the type of food, and alerts the user when to turn the grid and when the food reaches a desired cooking level.

Grid-level temperature probes measure whether the heat distribution along the grid is suitable, while additional probes measure the internal temperature of the food.

## **Afrikaans**

Om kos oor vuur gaan te maak was nog altyd deel van die menslike geskiedenis, en in Suid-Afrika is dit veral diep gewortel in ons kultuur met ’n ryk gesiedenis, en het tot die Afrikaanse term ‘braai’ geleid. ’n Probleem wat baie mense met braai het is die inkonsekwendheid daarvan omdat dit baie staatmaak op intuisie en ervaring. Die doel van hierdie projek is om ’n draadlose moniteringstelsel te ontwikkel wat intydse temperatuur metings verskaf, ’n kos kook profiel volg gebaseer op die tipe kos, en vir die gebruiker in kennis stel wanneer die rooster gedraai moet word en wanneer die kos die korrekte gaarheidvlak bereik.

Rooster-vlak temperatuur metings meet of die hitte verspreiding oor die rooster gepas is, terwyl addisionele metings van die interne kos temperatuur ook gemeet word.

# Contents

<b>Declaration</b>	ii
<b>Abstract</b>	iii
<b>List of Figures</b>	vii
<b>List of Tables</b>	viii
<b>Nomenclature</b>	ix
<b>1. Introduction</b>	1
1.1. Background . . . . .	1
1.2. Problem statement . . . . .	1
1.3. Project objectives . . . . .	1
1.4. Project scope . . . . .	2
1.5. Contents of report . . . . .	2
<b>2. Literature Review</b>	4
2.1. Related work . . . . .	4
2.1.1. Wired probe system . . . . .	4
2.1.2. Wireless probe system . . . . .	5
2.1.3. Conclusion . . . . .	6
2.2. Temperatures related to braaiing . . . . .	6
2.2.1. Range . . . . .	6
2.2.2. Accuracy . . . . .	7
2.3. Temperature sensors . . . . .	7
2.3.1. Resistance temperature detector . . . . .	7
2.3.2. Thermistor . . . . .	8
2.3.3. Thermocouple . . . . .	8
2.3.4. Semiconductor-based temperature sensors . . . . .	9
2.3.5. Comparison and selection . . . . .	9
2.4. Chapter summary . . . . .	10
<b>3. System Design</b>	11
3.1. System Overview . . . . .	11

3.2. Hardware subsystems . . . . .	12
3.2.1. Microcontroller . . . . .	12
3.2.2. Power Management . . . . .	13
3.2.3. Temperature Sensing . . . . .	14
3.2.4. Accelerometer . . . . .	15
3.2.5. Human-Machine Interaction (HMI) . . . . .	15
3.2.6. Storage . . . . .	16
3.2.7. USB-C . . . . .	16
3.2.8. PCB Design . . . . .	17
3.3. Software implementation . . . . .	17
3.4. Summary . . . . .	17
<b>4. Detailed Hardware Design</b>	<b>18</b>
4.1. Microcontroller Unit (MCU) . . . . .	18
4.2. Power Management . . . . .	20
4.3. Human-Machine Interaction . . . . .	22
4.4. Temperature Sensing . . . . .	23
4.5. Motion Detection . . . . .	25
4.6. SD Card Interface . . . . .	25
4.7. Power budget . . . . .	26
4.7.1. Battery size selection . . . . .	27
4.8. Pinout Summary . . . . .	27
4.9. PCB Stack-up and Assembly . . . . .	27
4.10. Summary . . . . .	28
<b>5. Detailed Software Design</b>	<b>29</b>
5.1. System Architecture . . . . .	29
5.2. Global Structures . . . . .	29
5.3. Startup and Setup Sequence . . . . .	30
5.4. Task Scheduling . . . . .	30
5.5. Web Interface Backend . . . . .	30
5.6. Temperature and Sequence Logic . . . . .	31
5.7. Data Logging . . . . .	31
5.8. Client Interface . . . . .	31
<b>6. Results</b>	<b>33</b>
6.1. Hardware subsystem verification . . . . .	33
6.1.1. Temperature measurement accuracy . . . . .	33
6.1.2. Accelerometer orientation detection . . . . .	33
6.1.3. Battery Voltage Readout . . . . .	34

6.1.4. Voltage Rails Verification . . . . .	34
6.1.5. Total Board Current Consumption . . . . .	34
6.1.6. LED and Buzzer Functional Test . . . . .	34
6.1.7. SD Card Insert Detection . . . . .	35
6.2. Software-Hardware Integration Testing . . . . .	35
6.2.1. Web Interface Data Retrieval . . . . .	35
6.2.2. Sequence Execution Test . . . . .	35
6.2.3. Temperature Logging to SD Card . . . . .	36
6.2.4. Task Scheduler Operation . . . . .	37
6.3. Full Test Sequence Verification . . . . .	37
6.4. Task Scheduler Operation . . . . .	38
6.5. Summary . . . . .	38
<b>7. Summary and Conclusion</b>	<b>39</b>
7.1. Summary of Results . . . . .	39
7.2. Discussion of Limitations . . . . .	39
7.3. Future Improvements . . . . .	40
7.4. Conclusion . . . . .	40
<b>Bibliography</b>	<b>41</b>
<b>A. Planning</b>	<b>46</b>
<b>B. Outcomes compliance</b>	<b>47</b>
<b>C. Combined Schematic</b>	<b>49</b>

# List of Figures

2.1. FireBoard 2 Thermometer . . . . .	4
2.2. MEATER Pro XL Thermometer . . . . .	5
2.3. Comparison of food internal temperature guidelines from different sources.	7
3.1. System-level block diagram. . . . .	11
4.1. Microcontroller connectivity and supporting circuits. . . . .	19
4.2. USB-C differential pair trace length matching. . . . .	20
4.3. Power management circuit diagrams including USB input, battery charging, and 3.3V regulation. . . . .	22
4.4. HMI circuit diagrams. . . . .	23
4.5. Temperature sensors circuit diagrams. . . . .	24
4.6. Accelerometer circuit diagram. . . . .	25
4.7. SD card holder circuit diagram. . . . .	26
4.8. SD card length matched data paths. . . . .	26
4.9. Comparison between the 3D render and the actual assembled PCB. . . . .	28
5.1. Software architecture overview. . . . .	29
5.2. Web Interaction Flow Between Client and ESP32 . . . . .	31
5.3. Web GUI for Monitoring and Configuration . . . . .	32
6.1. Data retrieval serial monitor and WebSocket console output. . . . .	35
6.2. Sequence execution test confirmation. . . . .	36
6.3. Temperature vs. time plot from SD card log, showing probes transitioning from ice water to boiling water and back to ice water. . . . .	36
C.1. The full schematic of the system. . . . .	49

# List of Tables

2.1. Relative advantages and disadvantages of RTDs, thermistors, thermocouples and IC sensors . . . . .	10
3.1. Summary of Microcontroller Family Strengths. . . . .	12
3.2. Comparison of possible ESP32 modules. . . . .	13
3.3. Comparison of thermocouple interface ICs. . . . .	15
3.4. Comparison of accelerometer options. . . . .	15
3.5. Comparison of piezo buzzer options. . . . .	16
4.1. Voltage divider configuration and validation for MCP9600 I <sup>2</sup> C address selection . . . . .	24
4.2. Maximum current consumption of system components . . . . .	27
4.3. MCU pin assignments to sensors and interfaces . . . . .	27
6.1. Temperature accuracy test results. . . . .	33
6.2. Accelerometer Rotation Test Results . . . . .	34
6.3. Battery Voltage Calibration . . . . .	34
6.4. Voltage Rail Measurements . . . . .	34
6.5. System Power Consumption . . . . .	34
6.6. Output Peripheral Verification . . . . .	35
6.7. Task Scheduler Timing . . . . .	37
6.8. Task Scheduler Timing . . . . .	38
A.1. Project Planning Schedule. . . . .	46
B.1. Graduate Attribute Compliance (Part 1) . . . . .	47
B.2. Graduate Attribute Compliance (Part 2) . . . . .	48

# Nomenclature

## Variables and functions

$V_{\text{ADDR}}$	Address configuration voltage for MCP9600
$V_{\text{FB}}$	Feedback voltage reference of TPS785-Q1 LDO
$R_{\text{PROG}}$	Programming resistor value to set battery charge current
$I_{\text{CHG}}$	Battery charge current

## **Acronyms and abbreviations**

ADC	Analog-to-Digital Converter
AP	Access Point
ESP32	Espressif Systems' Microcontroller Family
GUI	Graphical User Interface
HMI	Human-Machine Interface
IC	Integrated Circuit
I2C	Inter-Integrated Circuit communication protocol
IP	Internet Protocol
LED	Light-Emitting Diode
LDO	Low Dropout Regulator
PCB	Printed Circuit Board
PoC	Proof of Concept
PWM	Pulse Width Modulation
RTD	Resistance Temperature Detector
SD	Secure Digital (memory card format)
SDIO	Secure Digital Input Output
SPL	Sound Pressure Level
SPI	Serial Peripheral Interface
USB	Universal Serial Bus
Wi-Fi	Wireless Fidelity

# **Chapter 1**

## **Introduction**

### **1.1. Background**

A big part of South African cooking culture is the braai, which is cooking food over coals or an open flame. Though this is a popular cooking method, being able to follow precise instructions is complicated by the inconsistent and uncontrollable nature of a fire and its subsequent coals. This is especially difficult for people new to braaiing, as it relies heavily on intuition and experience. Under- or overcooked food due to poor heat distribution, or burning one side due to improperly timed grid turns are common problems everyone experiences.

As technology has improved most aspects of life, it is surprising to see the process of braaiing unassisted by modern technology. Apart from mechanical tools to improve the braai experience, there are no smart devices to assist users with the braai process, save for using a thermometer only after the meat has been cooked.

### **1.2. Problem statement**

The braaiing process lacks observable data and is largely intuition-driven, which, for both beginners and those more experienced, can lead to uneven cooking due to multiple factors. A solution is required that can give accurate real-time readings of the food's internal temperature and the grid-level temperature. Based on the type of food and the desired level of cooking, the device should notify the user when to turn the grid and when the food is ready.

### **1.3. Project objectives**

This project aims to find a solution through a proof of concept that can:

1. Accurately measure the internal temperature of multiple food items simultaneously.
2. Determine whether the heat distribution is even across the grid.

3. Notify the user when the grid needs to be turned and when a food item has reached its desired level of cooking.
4. Wirelessly connect to a smart device to display real-time information.
5. Offer the user a selection of presets for various food items and their desired level of cooking, if applicable.

## 1.4. Project scope

The project aims to design, create, and test a proof of concept that can complete the objectives mentioned in Section 1.3. The proof of concept tests technical feasibility, so boundaries are set on areas that would be covered if the project evolves to the prototype phase of development. The proof of concept boundaries for this project are:

- The system will be tested with wood coals as a heat source for the braai, though charcoal would also be a viable option.
- Although the system's use case is outdoors and in contact with food, the system will not be tested for waterproof capabilities and only the temperature probe material will need to be of food-grade compliant material.
- The functionality of cooking profiles will be tested using three types of meat, with the possibility of adding a broader food variety in the future.
- Although machine learning and adaptive algorithms would improve the system logic, for this project the logic will be manually tuned.
- Wireless control will focus on functionality, so the graphical user interface (GUI) will only display what is necessary for testing. Advancements like smartphone app integration and the use of a server to download cooking profiles can be integrated in the future.

## 1.5. Contents of report

**Chapter 1** The background information on the project is stated along with the scope, objectives, and contents of the entire report.

**Chapter 2** A deeper investigation into current products that achieve a similar goal, and a discussion on the temperature ranges and accuracy expected for the project.

**Chapter 3** This chapter deals with the design of the system, from concept and features needed, to high level explanations of required components and their functions.

**Chapter 4** Detailed design choices of hardware elements and how they work together are discussed in this chapter.

**Chapter 5** This chapter contains a detailed design of how the software of the system flows, along with the functions used to get the system working.

**Chapter 6** Testing of the system as a whole is discussed in this section, along with how the tests will work and their results.

**Chapter 7** A summary of the project as a whole is covered in this section, along with how the concept can be improved and what features could be added in case it becomes a viable product.

# Chapter 2

## Literature Review

### 2.1. Related work

Looking at current smart thermometers with a similar objective, the available solutions can be divided into wired probe and wireless probe systems. These solutions, however, do not meet all of the objectives this project sets out to achieve. An example product is looked at for each available system, and its method and shortcomings are discussed.

Since one of the main objectives of this project is to measure multiple internal temperatures simultaneously, the related systems discussed here must feature multiple probes. They should also have wireless connectivity to a smart device for real-time measurements.

#### 2.1.1. Wired probe system

In finding an example for the wired probe system, the following two articles were referenced: *The Best Bluetooth BBQ Thermometers for 2025* [1], and *We Put 21 Wireless Grill Thermometers to the Test—These 6 Were the Best* [2]. Both articles assessed the capabilities of multiple smart thermometer devices, and in each, the FireBoard 2 [3] was one of the wired solutions that stood out.



**Figure 2.1:** FireBoard 2 Thermometer. Reproduced from [3].

The FireBoard 2, seen in Figure 2.1, has six channels for external thermistor temperature probes, which can be configured as either food or ambient probes. There is a variant of

the FireBoard 2, the FireBoard 2 Pro [4], that features three input channels designed for thermocouple probes. In both variants, the probes are connected to the hub, which in turn connects to your smart device via WiFi and Bluetooth 4.0 connectivity.

An ambient measurement would be used to assess the cooking temperature and to automate an optional drive fan system. Food probes, on the other hand, monitor the meat's internal temperature and alert the user when it has reached the desired cooking level and that it should be taken off the heat. For each food probe, the user can select the type of meat being monitored and what level of cooking is required. The specifications in the user guide [5], the measurement range is  $-70$  to  $400^{\circ}\text{C}$  with an accuracy of  $\pm 0.4^{\circ}\text{C}$ .

According to the articles and product user reviews, the FireBoard 2 performs perfectly in grilling and smoking environments. The combination of precise temperature readings and detailed data tracking makes it the perfect tool to have next to the cooking station. However, this fixed placement of the system becomes a downfall in the case of a braai, where the grid is periodically turned.

The turning of the grid could cause the cables to tangle around the grid handle and get damaged, or the hub could be displaced. Another issue is that the ambient temperature measurement assumes a relatively uniform cooking environment and does not account for uneven heat distribution across the grid.

### 2.1.2. Wireless probe system

Looking at the second article mentioned above [2], along with the article *Our Favorite Meat Thermometers Take the Guesswork Out of Cooking* [6], the MEATER Pro XL [7] emerges as a good example for wireless probe systems and can be seen in Figure 2.2. This product features four fully wireless stainless steel probes, able to measure internal and ambient temperature.



**Figure 2.2:** MEATER Pro XL Thermometer. Reproduced from [7].

Each probe has five sensors for the internal temperature and one sensor for ambient measurement. Five sensors, along the neck of the probe, help find the true internal temperature by taking the lowest of the measurements. The ambient sensor, located in

the square end of the probe, calculates the approximate cooking time based on the type of meat selected. The internal sensors can withstand a maximum of 105°C, while the ambient sensor can handle up to 538°C, both boasting an accuracy of  $\pm 0.3^\circ\text{C}$ .

The system, by default, alerts the user when to remove food from heat and when you have five minutes left on your cook. For users desiring more control, the MEATER app [8] allows the user to set four custom alerts based on temperature readings. The app also allows users to select the type of meat for each probe and to see real-time temperature readings. In case the app cannot be used, the base station includes an OLED display with touch control which the user can use to operate the system.

Having wireless probes avoids the cable management problem stated in Section 2.1.1. Although the probes excel at giving accurate temperature measurements, the ambient sensors on their own don't fully represent the heat distribution along the cooking surface. In a braai environment, which is generally more open-spaced, this could cause uneven cooking consistency.

### **2.1.3. Conclusion**

Both wired and wireless probe systems discussed above offer reliable solutions for monitoring food and ambient temperatures, however they fall short when applied to a braai environment. For a wired system, such as the FireBoard 2, the cables going from the probes to the hub along with the periodical turning of the grid will cause issues. Although the wireless probes don't have this problem, neither system uses the ambient measurements to detect uneven heat distribution along the cooking surface.

## **2.2. Temperatures related to braaiing**

For the scope of this project there are two temperature ranges that need to be analyzed, namely ambient and food temperatures. These are required to know the range and accuracy of our measurement capabilities.

### **2.2.1. Range**

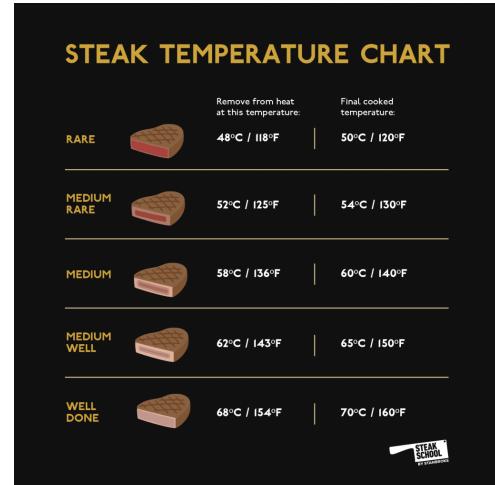
The measurement range is determined by the maximum temperature a wood fire can reach. According to City Fire Protection, an independent fire protection company operating in the United Kingdom (UK), the flame from a log fire is roughly 600°C [9]. This is corroborated by an article from Target Fire Protection, another commercial fire safety and protection company in the UK, which also states that red flames can range from 525°C to 1000°C [10].

## 2.2.2. Accuracy

To find the accuracy required for this project, the internal temperature of different meats at their various cooking levels needs to be examined. These temperatures are acquired from two sources: a guide published by Weber [11] and an article from Steak School by Stanbroke [12].

INTERNAL MEAT TEMPERATURE GUIDE		
Red Meat	Rare	49°C
	Medium Rare	54°C
	Medium	60°C
	Medium Well	66°C
	Well Done	68°C
	Low and Slow (Sliced)	88-91°C
	Low and Slow (Pulled)	93-95°C
Pork	Medium	63°C
	Medium Well	68°C
	Low and Slow (Sliced)	88-91°C
	Low and Slow (Pulled)	93-95°C
Ham, raw		71°C
Ham, fully cooked (to reheat)		60°C
Poultry	Well Done	74°C
Minced Meat / Sausage	Well Done	68°C
Fish	Medium	57°C

(a) Food internal temperatures as recommended by Weber. Reproduced from [11].



(b) Steak internal temperatures from Steak School by Stanbroke. Reproduced from [12].

**Figure 2.3:** Comparison of food internal temperature guidelines from different sources.

Excerpts from both sources, seen in Figure 2.3, show very similar temperatures for the various cooking levels of steak, with Weber also including internal temperatures for other meats. From these sources, the smallest change in cooking levels is 2°C, meaning a measurement accuracy of  $\pm 1^\circ\text{C}$  would be the ideal.

## 2.3. Temperature sensors

There are a variety of temperature sensor types that exist, each with their own benefits, drawbacks, and working principles. This section looks at four of the most common temperature sensor types and for each discusses how they work, the operating temperature ranges, their accuracy, circuit complexity (including additional components required), cost and power consumption. The information in this section is adapted from this source [13].

### 2.3.1. Resistance temperature detector

Resistance temperature detectors (RTDs) have a sensor (metal film or coil) that changes resistance in response to temperature, with an increase in temperature causing an increase in resistance [14]. The relationship between temperature and resistance is nearly linear,

and common practice is digitizing the signal and using a lookup table to correct errors [15]. RTDs have a wide operating range of roughly  $-250^{\circ}\text{C}$  to  $+750^{\circ}\text{C}$  [15]. The accuracy of an RTD changes with temperature and this change depends on the RTD's classification, with Class A and Class B being the most common by International Standard IEC 751 [16]. The accuracy of Class A RTDs are  $\pm(0.15 + 0.002T)^{\circ}\text{C}$ , and for Class B its  $\pm(0.30 + 0.005T)^{\circ}\text{C}$ , with  $T$  being temperature [16].

RTDs are passive components and require an excitation current to measure, increasing the power consumption of the circuit. They also require external components like amplifiers and ADCs (analog-to-digital converters) which could impose more errors. With RTDs alone already being more expensive than other temperature sensors, the cost for an entire temperature sensing circuit is high [15].

### **2.3.2. Thermistor**

Thermistors come in two types, the most common being negative temperature coefficient (NTC) thermistors [17]. They are similar to RTDs in that a change in temperature causes a change in resistance, but the resistance decreases as temperature increases. A standard thermistor's operating range is from  $-40^{\circ}\text{C}$  to  $+150^{\circ}\text{C}$  [15]. They have a very high accuracy of roughly  $\pm 0.2^{\circ}\text{C}$ , but it can be as low as  $\pm 1.5^{\circ}\text{C}$  [17]. Thermistors are quite inexpensive, however they require multiple components such as reference and discrete resistors as well as a comparator which increases the circuit complexity [15]. Another disadvantage they have is higher power consumption [18].

### **2.3.3. Thermocouple**

Thermocouples consist of two dissimilar material metal wires electrically bonded together at one end, called the measurement (or hot) junction, while the opposite end, called the reference (or cold) junction, connects to a measurement system [19]. For each wire, a voltage difference is produced across its length when there is a temperature difference between the two ends. This is called the Seebeck effect and the voltage is called the Seebeck voltage [19]. If the leads at the cold junction are kept at the same known temperature, the dissimilar metals produce different Seebeck voltages which can be used to determine the temperature at the hot junction. There are many different types of thermocouple types, each utilizing different combinations of dissimilar materials, the most common being the K-type made from a chromel-Alumel combination.

Thermocouples don't require an excitation current but the voltage they produce is very small (tens of microvolts per degree Celsius) so an amplifier is required, along with cold junction compensation for accurate readings [15]. The operating range varies depending on the thermocouple type, but most can handle ranges of  $-200^{\circ}\text{C}$  to  $+1750^{\circ}\text{C}$  [20]. Accuracy depends on thermocouple type and the standard followed by the manufacturer. The

International Electrotechnical Commission (IEC) standard has three tolerance classes for each thermocouple type. For K-type, the class 1 tolerance has a maximum temperature error of the larger of either  $\pm 1.5^\circ\text{C}$  or  $\pm(0.004T)^\circ\text{C}$ , with  $T$  being temperature [21]. Since thermocouples are self-powered, the power consumption of the circuit depends on the amplifier and cold junction compensation used. They are also quite inexpensive [21].

### 2.3.4. Semiconductor-based temperature sensors

These type of temperature sensors are mostly incorporated into integrated circuits (ICs) [20], meaning they are surface mount components on a printed circuit board (PCB). They work with diodes producing temperature sensitive voltages according to temperature changes. They typically have a working range of  $-55^\circ\text{C}$  to  $+150^\circ\text{C}$  with an accuracy varying between  $\pm 1^\circ\text{C}$  and  $\pm 5^\circ\text{C}$  [22]. Since they are incorporated into ICs they rarely require additional components (unless an ADC is not built in), have low power consumption and have low device cost [15].

### 2.3.5. Comparison and selection

The characteristics of the sensors discussed above are summarized in Table 2.1. For this project, in order to avoid increased complexity and cost, one sensor type will be used for measuring both internal food and cooking surface temperatures. Therefore, the chosen sensor must be affordable, able to handle temperatures as stated in Section 2.2.1, and measure the internal temperature of food with sufficient accuracy. Section 2.2.2 outlines an ideal numerical accuracy based on the smallest difference in cooking levels from both sources, but it is important to note that internal food temperatures are not an exact science. This is evident by the difference in target temperatures for cooking levels of steak seen in Figures 2.3a and 2.3b. Consequently, for this proof of concept, the sensor should at minimum offer enough resolution to clearly distinguish between most adjacent cooking levels.

RTDs are very accurate, however they are expensive and could damage under harsh temperature conditions. While thermistors are inexpensive, they are not suitable to handle the temperature ranges required. The same applies to semiconductor-based sensors, along with them being PCB mount devices and won't be able to measure food internal temperature.

Thermocouples are the best choice since they offer a wide operating range and are inexpensive. Although they have a lower accuracy compared to RTDs and thermistors, their accuracy will still suffice for the goal of this project. The sensitivity criteria in Table 2.1 is slightly misleading as it relates to the amplitude of change in output signal per degree Celsius rather than overall performance. While the larger resistance changes of RTDs and thermistors would be advantageous, careful hardware design can reduce

noise and enable reliable temperature measurement. Based on the temperature ranges and accuracy they offer, K-type thermocouples are selected for the project [21].

**Table 2.1:** Relative advantages and disadvantages of RTDs, thermistors, thermocouples and IC sensors. Adapted from [15].

Criteria	RTD	Thermistor	Thermocouple	IC sensor
Temperature range	$-250^{\circ}C$ to $+750^{\circ}C$	$-40^{\circ}C$ to $+150^{\circ}C$	$-267^{\circ}C$ to $+2316^{\circ}C$	$-55^{\circ}C$ to $+200^{\circ}C$
Accuracy	Best	Depends on calibration	Good	Good
Linearity	Best	Depends on type	Good	Best
Sensitivity	Good	Depends on type	Worst	Good
Circuitry	Less	Depends on type	Complex	Simplest
Power consumption	High	High	Low to high	Lowest
Cost	Medium to high	Low to high	Medium to high	Low to medium

## 2.4. Chapter summary

This chapter looked at existing smart temperature measurement devices, temperature values relevant to braaiing, and different temperature sensor options to determine which would work best for this project. The information gathered here is used to justify design choices made in the following chapters.

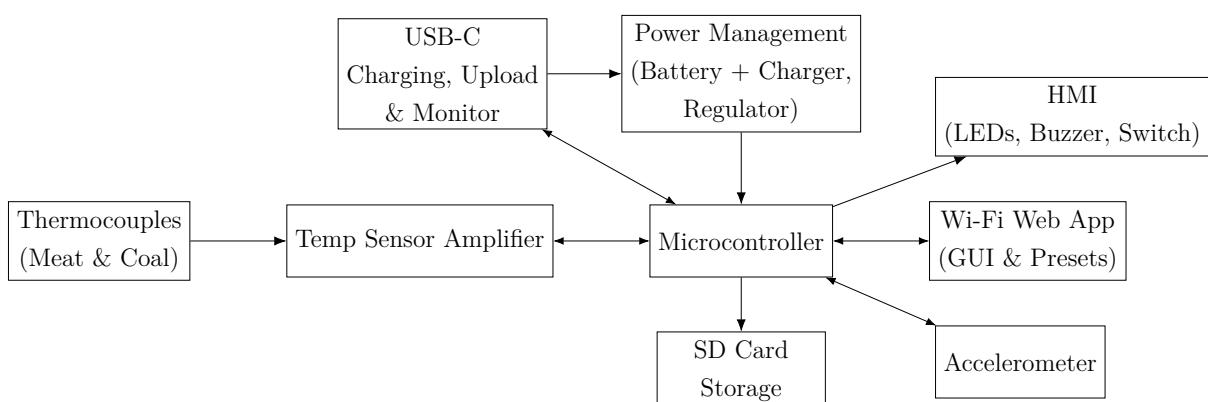
# Chapter 3

## System Design

This project aims to create a smart monitoring and notification system to automate the braai sequence, save for turning the grid. Since the system is a proof of concept, the design choices are made regarding simplicity and affordability. This chapter outlines the high-level design choices made regarding hardware component selection and software implementation.

### 3.1. System Overview

The block diagram, seen in Figure 3.1, shows the respective blocks required for the system. The microcontroller communicates and processes data from sensor handling circuits such as the temperature sensor amplifier and accelerometer blocks and uses the human machine interaction (HMI) circuits and Wi-Fi web app to relay relevant information. The Wi-Fi web app is also used as the control mechanism for the system. An SD card is included to store data to be used for testing and debugging. A USB-C connector is used to upload software to the microcontroller, monitor serial communications for debugging and testing, and to deliver power to the charging circuit in the power management block. Power management also includes a regulator used to power the entire system. To ensure affordability and avoid additional components needed, the components selected in the sections hereunder should all work with the same operating voltage.



**Figure 3.1:** System-level block diagram.

## 3.2. Hardware subsystems

### 3.2.1. Microcontroller

In order to choose a microcontroller for this project, a suitable microcontroller family must first be selected. To do that, a comparative analysis of five widely used microcontroller families - AVR, 8052, PIC, STM32, and ESP32 - is adapted from this source [23]. Table 3.1 summarizes the key features and advantages of each microcontroller family.

**Table 3.1:** Summary of Microcontroller Family Strengths. Adapted from [23]

Microcontroller Family	Key Features and Advantages
AVR	Ideal for applications requiring low power consumption and high performance, with a wide range of peripherals.
8052	Suitable for simple embedded systems due to its simplicity, ease of use, and low cost.
PIC	Offers a wide range of peripherals, making it suitable for a variety of applications.
ESP32	Versatile microcontroller with integrated Wi-Fi and Bluetooth, making it ideal for IoT applications.
STM32	Provides high integration, strong performance, and broad peripheral support—suitable for complex applications.

The ESP32 stood out as the most suitable choice for this system primarily due to its affordability and native support of Wi-Fi, which is essential for wireless communication with the GUI. The ESP32 supports multiple programming frameworks, allowing advanced development through ESP-IDF.

To select a specific ESP32 microcontroller, the following capabilities are required:

- Wi-Fi: 2.4GHz 802.11 b/g/n/ax
- Maximum temperature: 105°C
- Antenna: PCB
- Peripherals: SDIO Slave, I<sup>2</sup>C

Inputting these capabilities into the Espressif product selector [24], the possible microcontroller options are found as shown in Table 3.2. The ESP32-WROOM-32E-H4 and ESP32-MINI-1-H4 have faster clock rates and larger read-only memory (ROM) and static random-access memory (SRAM), however they only support up to Wi-Fi 4 (802.11n) standard. Both ESP32-C6 series microcontrollers support Wi-Fi 6 (802.11ax) standard which has improved network efficiency, uses less power and increased range [25].

**Table 3.2:** Comparison of possible ESP32 modules [24].

Part Number	Flash	SRAM	ROM	WiFi	Bluetooth	Max Freq	Temp Range	GPIOs
ESP32-WROOM-32E-H4	4 MB	520 KB	448 KB	802.11b/g/n	v4.2	240 MHz	-40–105°C	26
ESP32-MINI-1-H4	4 MB	520 KB	448 KB	802.11b/g/n	v4.2	240 MHz	-40–105°C	28
ESP32-C6-MINI-1-H4	4 MB	512 KB	320 KB	802.11b/g/n/ax	v5.3	160 MHz	-40–105°C	22
ESP32-C6-MINI-1-H8	8 MB	512 KB	320 KB	802.11b/g/n/ax	v5.3	160 MHz	-40–105°C	22

Of the ESP32-C6 series, the MINI-1-H8 does offer double the amount of flash memory compared to the MINI-1-H4, but due to availability constraints [26] the MINI-1-H4 is selected. This microcontroller has an operating voltage of 3 to 3.6 V, with 3.3 V typical [27]. Following this, the subsystems below must also have the same voltage requirements.

### 3.2.2. Power Management

The power management subsystem comprises of the battery, battery charger circuit and voltage regulator circuit. For the battery charger and voltage regulator circuits, the SparkFun Qwiic ESP32-C6 Pocket Development Board [28] is used to influence the component selection as it offers reliable components proven to work with the ESP32-C6 series microcontroller. These components are evaluated in the following sections to determine if they are suitable for the system.

#### Battery and battery charger

For the battery charger circuit, the MCP73831 charge management controller is selected. Other options such as the BQ21040 are also viable and offer a higher charge current, but the MCP73831 is ultimately chosen due to its mass availability and much smaller leakage current. This device comes in two packages, and for this project the 5-Lead, SOT-23 package [29] is chosen due to its smaller size, cheaper cost and being more readily available [30].

For the battery, a lithium polymer (Li-Po) battery is chosen over a lithium-ion (Li-Ion) battery since it is better suited for the compact design required for the system. A silver flat pack Li-Po battery with a nominal voltage of 3.7 V [31] is chosen for the project. The capacity of the required battery will be selected in Section 4.7 based on the power budget.

#### Voltage regulator

Since the difference between the 3.7 V input and required 3.3 V output voltage is so small, a linear regulator will not suffice [32]. Therefore, a low-dropout (LDO) regulator must be used. The SparkFun ESP32-C6 Development Board uses a RT9080-33GJ5 [28] which has a maximum output current of 600mA [33]. Since our system includes peripherals such as a buzzer, RGB LEDs (Red, Green, and Blue Light-Emitting Diodes), Wi-Fi communication

and an SD card, this maximum current might not accommodate higher instantaneous current draws.

For this project the TPS785-Q1 is chosen as the LDO regulator since it provides a ultra-low dropout of 315 mV (max) at 1 A when outputting 3.3 V [34]. The TPS785-Q1 thermal pad is electrically connected to the ground pin which, when connected to a large-area ground plane, offers excellent thermal performance by distributing the heat.

### **3.2.3. Temperature Sensing**

The temperature sensing subsystem comprises of five K-type thermocouples, thermocouple connectors, and, as discussed in Section 2.3.3, a thermocouple amplifying circuit. The amplifier circuits communicate with the microcontroller, sending the relevant temperature information.

#### **K-type thermocouple**

Since the thermocouple sensors will be subject to the temperatures discussed in Section 2.2.1, the probe and cable will need to have adequate insulation. Two options were compared, namely the Pro Series Thermocouple Food Probe from FireBoard [35] and the COM1706 Stainless Steel K-type Thermocouple from Pimoroni. These thermocouples are selected due to their cables having fiberglass insulation as well as steel braided outer sheath. Although the COM1706 thermocouple doesn't have a penetrative tip like its FireBoard counterpart, it is selected based on its affordability.

#### **Thermocouple connector**

In order to connect the thermocouples to the PCB, the system requires K-type female connectors. The XE-1139-001 [36] is selected as it offers additional sunken thru-holes which can be used to mount the connectors to a encapsulating body, improving its durability. Due to the thermocouples chosen being terminated with fork terminals, additional male connectors are needed to connect to the board. The XE-1384-001 connectors [37] are chosen as they are from the same brand as the female connectors, ensuring compatibility.

#### **Thermocouple amplifier**

For the thermocouple amplifier circuit, three ICs were considered namely the MAX31856MUD+, MCP9600T-E/MX and MCP9601T-E/MX. Table 3.3 compares these options to one another. The MAX31856MUD+ has the highest precision, built-in cold-junction compensation and fault detection capability, however, it only supports SPI [38], which would require 8 wires to communicate with the five ICs required. The MCP9601 includes additional open- and short-circuit fault detection compared to the MCP9600 [39],

however for the proof-of-concept this is not essential. The MCP9600 offers the best balance of precision, I<sup>2</sup>C capability, and multi-address compatibility [39]. I<sup>2</sup>C is beneficial since it allows for communication with five ICs using only two wires. The MCP9600 is chosen as thermocouple amplifier.

**Table 3.3:** Comparison of thermocouple interface ICs.

Parameter	MCP9600	MCP9601	MAX31856
Interface	I <sup>2</sup> C & SPI	I <sup>2</sup> C & SPI	SPI only
Accuracy (Typ./Max.)	±0.5°C / ±1.5°C	±0.5°C / ±1.5°C	±0.7°C / ±1.5°C
Cold Junction Compensation	Integrated	Integrated	Integrated
Open/Short Circuit Detection	No	Yes	Yes
Additional Notes	Proven, low-cost, multi-address I <sup>2</sup> C support	Adds fault detection features	High-end, but I <sup>2</sup> C unavailable

### 3.2.4. Accelerometer

An accelerometer is added to further automate the system by detecting when the grid, and by extension the printed circuit board (PCB), is flipped over. This removes any input required from the user to indicate that the grid has been turned over. Three accelerometers are selected and compared to one another to find the best option. The comparisons can be seen in Table 3.4. Although the KX122-1037 offers very high bandwidth [40] and the LIS2DE12TR has ultra-low power consumption [41], they are not necessary for a subsystem meant to only detect changes in z-axis orientation. The LIS3DHTR is selected as it has enough bandwidth for z-axis flip detection and its current consumption in high-resolution mode is relatively equal to that of the LIS2DE12TR [42].

**Table 3.4:** Comparison of accelerometer options.

Parameter	LIS3DHTR [42]	LIS2DE12TR [41]	KX122-1037 [40]
Interface	I <sup>2</sup> C / SPI	I <sup>2</sup> C / SPI	I <sup>2</sup> C / SPI
Selectable Range	±2/4/8/16 g	±2/4/8/16 g	±2/4/8 g
Bandwidth	0.5 Hz – 625 Hz	0.5 Hz – 2.69 kHz	6.25 Hz – 12.8 kHz
Supply Voltage	1.71 V – 3.6 V	1.71 V – 3.6 V	1.71 V – 3.6 V
Max Current Consumption @ 50 Hz	11 µA	6 µA	145 µA
Special Features	High-resolution mode	Compact, ultra-low-power	High bandwidth, robust specs

### 3.2.5. Human-Machine Interaction (HMI)

The human-machine interaction subsystem is used to turn the system on and off, and to provide the system with additional communication capabilities apart from the web app. It includes a rocker switch, RGB LED and buzzer. For the rocker switch, used to turn the system on and off, the 500ASSP1M6QE single pole double throw (SPDT) slide switch [43] is chosen as its right angle mounting offers mechanical robustness and ease of access, while its 3 A current rating ensures electrical reliability and thermal safety.

While the web application serves as the primary interface for monitoring and notification, the RGB LED and buzzer provide additional visual and auditory feedback when necessary,

like when the grid needs to be turned or when the cooking cycle is completed. They are also used during system startup to communicate the local web URL the user should use to access the system web app. A RGB LED is selected over a single colour LED as it can use different colours to identify different digits in the URL. It can also be used to communicate information regarding specific probes or system states. This avoids using an LED screen which would increase cost and power consumption.

For the LED a 5mm common cathode RGB LED is selected [44]. For the buzzer a piezoelectric buzzer is chosen over a magnetic buzzer since its current draw is much lower and it has a higher sound pressure level (SPL) [45]. Table 3.5 shows a comparison of the buzzer options considered. The buzzer options were selected to have a 60dB SPL at a rated voltage of 3V . The PS1240P02CT3 is selected since it has the smallest diameter and height [46].

**Table 3.5:** Comparison of piezo buzzer options.

Parameter	PS1240P02CT3	PS1740P02CE	PS1240P02BT
Sound Pressure Level	60 dB @ 3V, 10cm	60 dB @ 3V, 10cm	60 dB @ 3V, 10cm
Diameter	12.2 mm	17.0 mm	12.2 mm
Height (Max)	4.6 mm	4.9 mm	7.8 mm
Notes	Slim profile	Larger footprint	Taller profile

### 3.2.6. Storage

An SD card is used to log session data which can be used for testing and performance evaluation. To minimize the physical footprint, the storage system will use a microSD card and card slot. The MEM2075-00-140-01-A microSD card connector is chosen as it has a push-push insertion and removal method, and it includes a normally open switch [47] used to detect whether a microSD card is present.

### 3.2.7. USB-C

A USB-C port is used to support firmware uploads, monitor serial communication and deliver power for battery charging. The USB-C connector, similar to the rocker switch chosen in Section 3.2.5, should mount at a right-angle to offer mechanical robustness and ease of access. This project will make use of the UJ20-C-H-G-SMT-TR USB Type C receptacle [48]. It offers a sufficient amount of 10,000 life cycles, ensuring longevity of the system.

### **3.2.8. PCB Design**

The final design uses a 6-layer PCB, manufactured by JLCPCB. A 2-layer PCB was initially considered for the system to reduce cost. However, as the design progressed, it became clear that the combination of limited board space, signal routing complexity, and the need for effective grounding and power distribution made a 2-layer layout impractical. Additionally, a JLCPCB 6-layer manufacturing promotion offering \$30 USD off (approximately R570) made 6-layer boards financially viable without exceeding the budget. This allowed for a more robust design, with improved signal integrity, thermal performance, and overall layout flexibility. PCB specific details are discussed in Section 4.9.

## **3.3. Software implementation**

The systems software consists of firmware running on the ESP32-C6 microcontroller and a web-based graphical user interface (GUI) accessible over Wi-Fi. The firmware is written in C++ using the Arduino framework, and is responsible for sensor polling, SD card logging, buzzer and LED control, flip detection logic, and managing HTTP/WebSocket communication with the frontend. Detailed software design is fully discussed in Chapter 5.

## **3.4. Summary**

This chapter covered the system overview and discussed the subsystems required. For each subsystem, the possible component choices were compared to each other to choose whichever would work best for the system. The following section discusses the detailed hardware design required for each of the components.

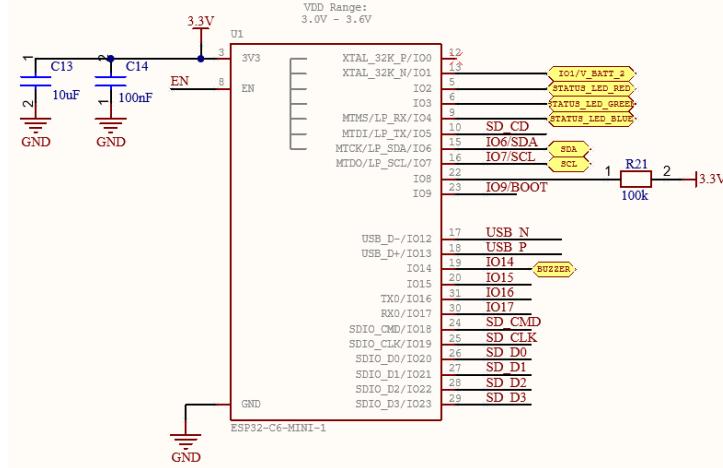
# **Chapter 4**

## **Detailed Hardware Design**

This chapter outlines the detailed hardware design of the components selected in Chapter 3. All component circuits are designed based on information available in the datasheet or following evaluation board reference designs. The system is implemented on a 6-layer stack PCB to avoid routing complexity and ensure a compact design.

### **4.1. Microcontroller Unit (MCU)**

The ESP32-C6-MINI-1-H4 module manages all the system functions. It communicates with sensing circuits using I<sup>2</sup>C, writes to the SD card through SDIO interface, and controls the HMI circuits through GPIOs. The ESP32 and supporting circuitry can be seen in Figure 4.1.



(a) ESP32-C6 pin configuration and connections.

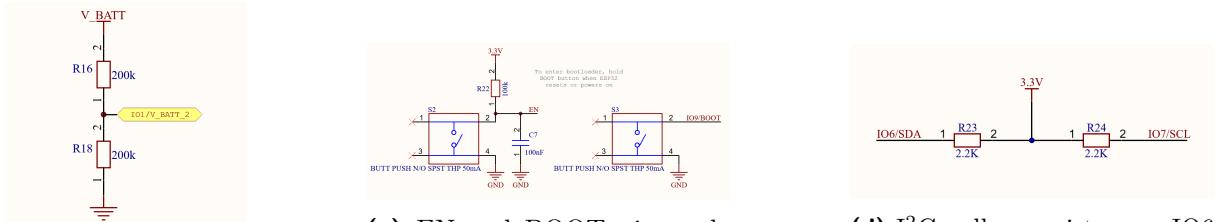


Figure 4.1: Microcontroller connectivity and supporting circuits.

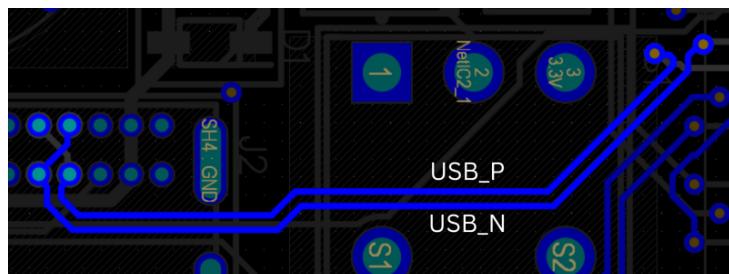
The following pin configuration and allocation decisions were made based on the guidelines and electrical characteristics detailed in the MCUs datasheet [27]:

- **Boot Mode Control:** IO8 is pulled high using a 100 kΩ resistor. IO9 is internally pulled up and connected to an external SPST (Single-Pole Single-Throw) push-button. Holding this button low during reset places the MCU in Download Boot mode.
- **SD Card Interface:** IO18 to IO23 are dedicated to the SDIO interface (CMD, CLK, D0–D3). Pull-up resistors and length matching is discussed in the SD card design.
- **I<sup>2</sup>C Bus:** IO6 (SDA) and IO7 (SCL) are used to communicate with the MCP9600 temperature sensors and LIS3DHTR accelerometer. 2.2 kΩ pull-up resistors are placed near the microcontroller.
- **Battery Voltage Sensing:** IO1 is an ADC-enabled pin used to measure battery voltage via a resistor divider.
- **USB Communication:** IO12 and IO13 are used as USB\_D- and USB\_D+, respectively, for serial communication via USB. This connection is also used to upload firmware onto the ESP32.

- **SD Card Detection:** IO5 is connected to the card detect switch in the microSD socket and pulled up to 3.3 V via a  $100\text{ k}\Omega$  resistor.
- **RGB LED:** The RGB LED pins are driven by IO2 (Red), IO3 (Green), and IO4 (Blue).
- **Buzzer Control:** IO14 is used to drive a buzzer through a transistor switch.
- **Enable Pin:** The EN pin is tied high through a resistor but also connected to a SPST push-button, which pulls the EN pin low when pressed to shut down the system.
- **Bypass Capacitors:**  $100\text{ nF}$  and  $10\mu\text{F}$  capacitors are placed as close as possible to the 3.3 V power pin of the ESP32 to suppress voltage ripple and high-frequency noise.

## 4.2. Power Management

The power management subsystem features USB-C input, battery charging using the MCP73831, and 3.3 V voltage regulation using the TPS785-Q1. The respective circuit diagrams are shown in Figure 4.3. Power enters the system through a USB-C connector, providing 5 V to the MCP73831 charge controller. To enable proper power negotiation, the USB-C CC1 and CC2 configuration channel pins are each pulled to ground using  $5.1\text{ k}\Omega$  resistors. Without these resistors, power will not be provided by standard USB-C sources unless a USB-A to USB-C cable is used. Ignoring one of the resistors limits functionality to a single plug orientation [49]. The pins used to program the MCU, namely USB\_P and USB\_N seen in Figure 4.3a, are a differential pair [50], meaning their traces need to be length matched and maintain consistent impedance to minimize skew and signal reflections [51]. The tracks for this differential pair is shown in Figure 4.2 and was implemented on Altium using differential pair tuning.



**Figure 4.2:** USB-C differential pair trace length matching.

The layout of the battery charge circuit, seen in Figure 4.3b, is acquired from the typical application circuit in the MCP73831 datasheet [29]. The  $10\mu\text{F}$  capacitors are

used to maintain good alternating current (AC) stability and provide compensation when there is no battery load. The  $4.7\text{k}\Omega$  resistor is used to limit the current through the LED. The resistor connected to the PROG pin of the MCP73831 is used to scale the charge current. The MCP73831 can provide a charge current up to 500 mA but to ensure thermal safety and reduce stress on the battery, a target charge current of 400 mA is selected. The charge current  $I_{\text{CHG}}$  is set using Equation 4.1 [29]. Using a charge current of 400 mA, the calculated  $R_{\text{PROG}}$  value is  $2.5\text{k}\Omega$ . In practice, a  $2.2\text{k}\Omega$  resistor was used, resulting in a slightly higher current of approximately 454.5 mA.

$$I_{\text{CHG}} = \frac{1000}{R_{\text{PROG}}} \quad (4.1)$$

The TPS785-Q1 LDO regulator has an adjustable output voltage. The circuit diagram in Figure 4.3c is obtained from the datasheet [34]. Its output voltage  $V_{\text{OUT}}$  is set using the following formula found in the datasheet [34]:

$$V_{\text{OUT}} = V_{\text{FB}} \times \left(1 + \frac{R_1}{R_2}\right) + I_{\text{FB}} \times R_1 \quad (4.2)$$

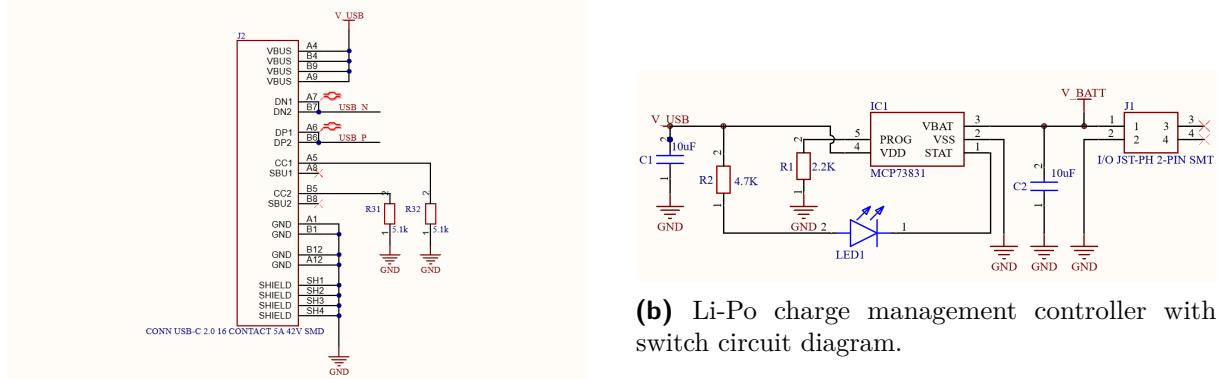
Where:

- $V_{\text{FB}}$  is the reference voltage (typically 1.2 V),
- $I_{\text{FB}}$  is the feedback pin bias current (typically  $0.01\text{\mu A}$ ),
- $R_1$  and  $R_2$  are the upper and lower resistors in the voltage divider.

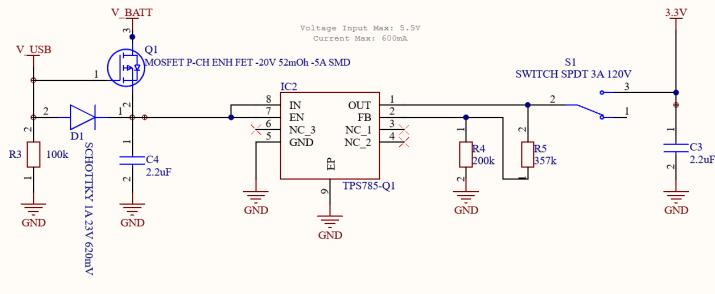
For an output voltage of 3.3 V, and choosing  $R_2$  to be  $200\text{k}\Omega$ , Equation 4.2 can be solved to find  $R_1$  equal to

$$R_1 = 349.4\text{k}\Omega.$$

In practice  $R_1$  is chosen to be  $357\text{k}\Omega$ , and using Equation 4.2 again the output voltage is equal to 3.342 V which is within tolerance for the system. Also seen in Figure 4.3c is a P-channel MOSFET circuit. This circuit enables automatic switching between USB power and battery output to the 3.3 V LDO regulator and is sourced from the SparkFun ESP32-C6 development board [28].



**(a)** USB-C male connector circuit diagram.



**(c)** 3.3V LDO voltage regulator circuit diagram.

**Figure 4.3:** Power management circuit diagrams including USB input, battery charging, and 3.3V regulation.

## 4.3. Human-Machine Interaction

The HMI includes the RGB LED, buzzer and rocker switch. The rocker switch circuit is included in the LDO voltage regulator circuit diagram shown in Figure 4.3c. The RGB LED circuit diagram is shown in Figure 4.4a along with the power LED circuit. The RGB LED is connected in a common cathode configuration with resistors added to limit the current. Each LED channel in an RGB package has a different forward voltage based on its colour [52]:

- Red:  $V_{F,\text{Red}} \approx 1.6\text{to}2\text{ V}$
  - Green:  $V_{F,\text{Green}} \approx 1.9\text{to}4\text{ V}$
  - Blue:  $V_{F,\text{Blue}} \approx 2.5\text{to}3.7\text{ V}$

The ESP32-C6 GPIO has a high-level output voltage of 3.3 V [27]. If the current limiting resistors are set to  $R = 1\text{ k}\Omega$ , the current through each color channel can be

estimated as:

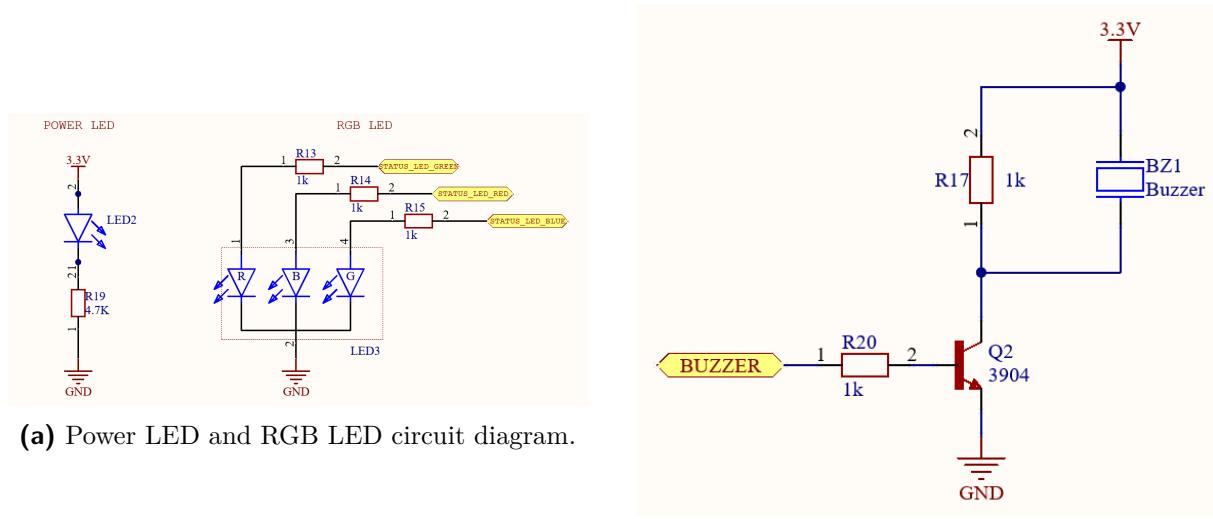
$$I_{\text{Red}} = \frac{3.3 \text{ V} - 1.8 \text{ V}}{1 \text{ k}\Omega} = 1.5 \text{ mA}$$

$$I_{\text{Green}} = \frac{3.3 \text{ V} - 2.95 \text{ V}}{1 \text{ k}\Omega} = 0.35 \text{ mA}$$

$$I_{\text{Blue}} = \frac{3.3 \text{ V} - 3.1 \text{ V}}{1 \text{ k}\Omega} = 0.2 \text{ mA}$$

These currents fall within the ESP32-C6's safe GPIO limitations and give sufficient brightness for status indication.

The buzzer circuit shown in Figure 4.4b is sourced from the recommended operating circuit in the PS1240P02CT3 datasheet [46]. A 3904 NPN bipolar transistor is used as it is very affordable and readily available [53].



**Figure 4.4:** HMI circuit diagrams.

## 4.4. Temperature Sensing

Temperature measurements are performed using five MCP9600 thermocouple amplifier ICs. These ICs communicate with the microcontroller over the I<sup>2</sup>C bus. Each MCP9600 is configured with a unique address to allow multi-probe support on the same bus. Each individual circuit layout, seen in Figure 4.5a, is based on the Adafruit MCP9600 I<sup>2</sup>C Thermocouple Amplifier breakout board [54], with the addition of a 100 nF bypass capacitor at the V<sub>DD</sub> pin to minimize supply voltage noise.

The unique I<sup>2</sup>C addresses are configured using the voltage level applied to the ADDR pin of the MCP9600. The voltage applied determines the last three bits of the 7-bit address, with the IC offering eight address selection levels (1100xxx) as seen in input/output pin DC characteristics table in the datasheet [39]. Equation 4.3 shows the formula for the

typical voltage level for each address, while Equations 4.4 and 4.5 give the low and high voltage limits.

$$V_{\text{ADDR\_TYP}} = \frac{\text{Address} \times V_{\text{DD}}}{8} + \frac{V_{\text{DD}}}{16} \quad (4.3)$$

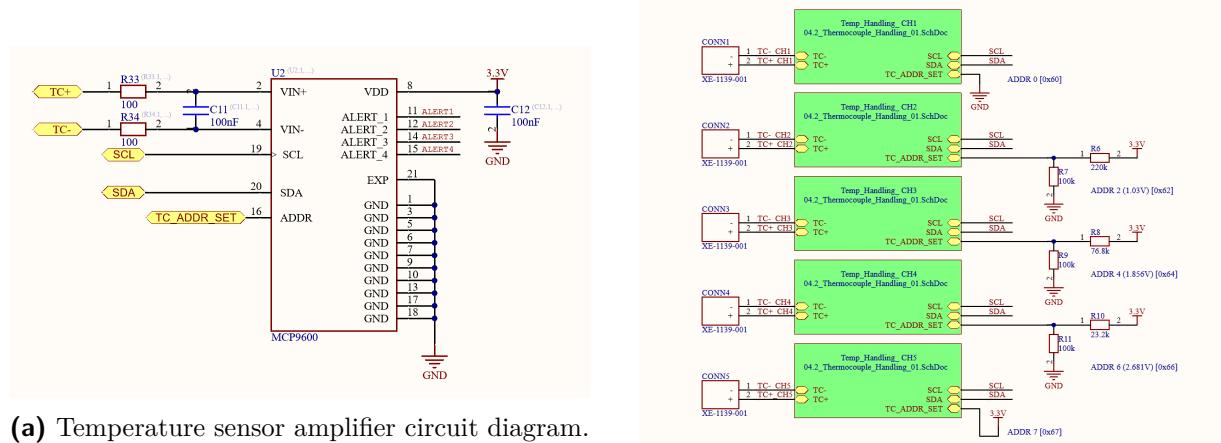
$$V_{\text{ADDR\_L}} = V_{\text{ADDR\_TYP}} - \frac{V_{\text{DD}}}{32} \quad (4.4)$$

$$V_{\text{ADDR\_H}} = V_{\text{ADDR\_TYP}} + \frac{V_{\text{DD}}}{32} \quad (4.5)$$

Using a voltage divider and choosing a  $100\text{ k}\Omega$  for the bottom resistor, we need to calculate the individual resistors required to configure the addresses for each IC. Selecting addresses 0, 2, 4, 6 and 7, and using Equations 4.3, 4.4 and 4.5, we find the resistors as shown in Table 4.1. The addressing circuits are shown in Figure 4.5b.

**Table 4.1:** Voltage divider configuration and validation for MCP9600 I<sup>2</sup>C address selection

Address	Hex	R <sub>top</sub> (kΩ)	R <sub>bottom</sub> (kΩ)	Calculated V <sub>ADDR</sub> (V)	V <sub>ADDR_L</sub> (V)	V <sub>ADDR_H</sub> (V)
0	0x60	0 (GND)	—	0.00	N/A	N/A
2	0x62	220	100	1.03	0.93	1.13
4	0x64	76.8	100	1.86	1.76	1.96
6	0x66	23.2	100	2.68	2.58	2.78
7	0x67	— (VDD)	0	3.30	N/A	N/A

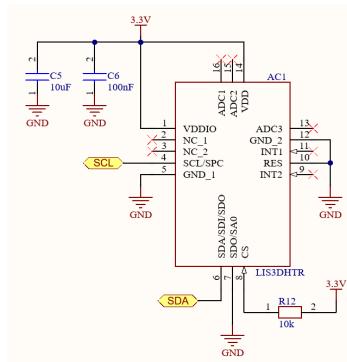


**Figure 4.5:** Temperature sensors circuit diagrams.

Regarding PCB layout considerations, the MCP9600 datasheet recommends nine thermal vias be placed underneath the exposed pad below the IC. This was implemented when creating the PCB.

## 4.5. Motion Detection

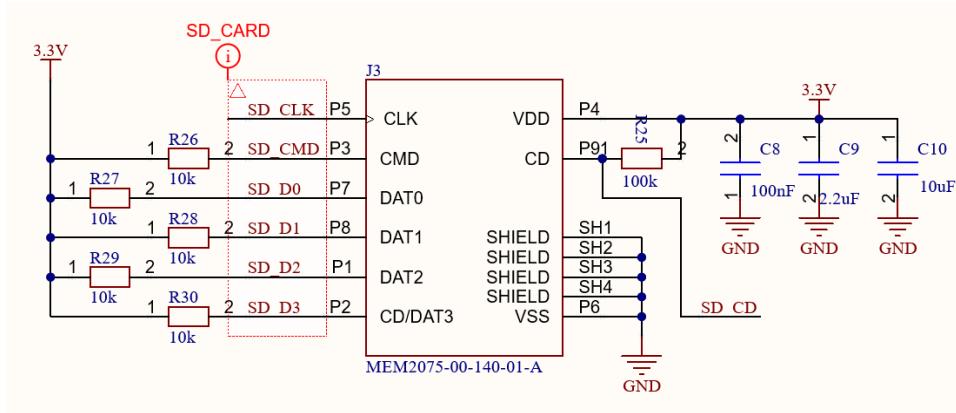
The LIS3DHTR accelerometer circuit diagram seen in Figure 4.6 is sourced from the application hints chapter of the datasheet [42]. It states that the  $10\ \mu\text{F}$  and  $100\ \mu\text{F}$  power supply decoupling capacitors should be placed as close to pin 14, the  $\text{V}_{\text{DD}}$  supply voltage pin, as possible. The chapter also states that the ADC1, ADC2 and ADC3 pins can be left floating if not used. In order to enable I<sup>2</sup>C communication instead of SPI, pin number 8 labeled CS should be tied high. This is done using a  $10\ \text{k}\Omega$  resistor. The SDO/SA0 pin, pin number 7, can be used to select between the two available I<sup>2</sup>C addresses (001100x). Since the system is only using one accelerometer, and both LIS3DHTR addresses aren't similar to any of the MCP9600 addresses mentioned in Section 4.4, the SDO/SA0 pin is tied low.



**Figure 4.6:** Accelerometer circuit diagram.

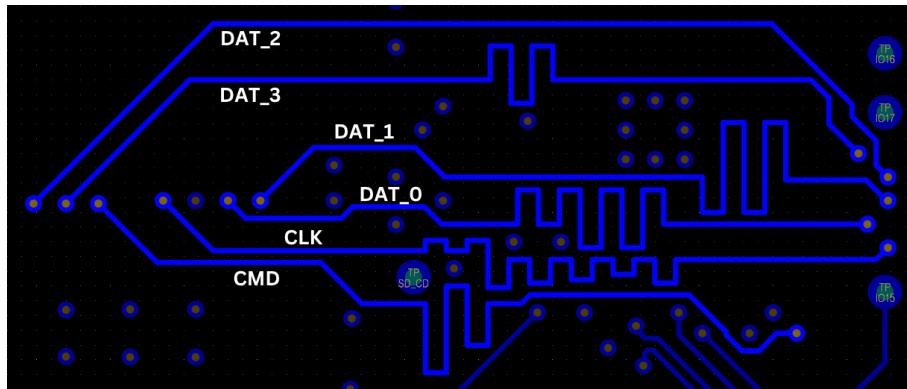
## 4.6. SD Card Interface

The SDIO interface is used for SD card access. Following the guide by Espressif on the SDIO Card Slave Driver it states that the CMD and DATA lines (DAT0-DAT3) should be pulled up by  $10\ \text{k}\Omega$  to  $90\ \text{k}\Omega$  resistors, whether in 1-bit SD, 4-bit SD or SPI mode. The card detect pin CD is pulled high using a  $100\ \text{k}\Omega$  resistor. When a card is inserted, the CD pin is pulled to ground [47]. A combination of 100 nF, 2.2  $\mu\text{F}$  and 10  $\mu\text{F}$  decoupling capacitors is used at the SD card's supply pin. These capacitors ensure stable supply voltage across a wide frequency range and minimize dips in the voltage when initialization or write operations. The final circuit can be seen in Figure 4.7.



**Figure 4.7:** SD card holder circuit diagram.

An important aspect, in order to avoid glitches or data corruption, when writing to an SD card is to ensure the data signal paths are length matched to within a tenth of a millimeter, and the clock signal must be one millimeter longer [55]. These tracks must also be surrounded by a good ground plane with an adequate number of vias to avoid crosstalk between the signals. This length matched circuit, connecting the ESP32 to the SD card holder, is shown in Figure 4.8.



**Figure 4.8:** SD card length matched data paths.

## 4.7. Power budget

Table 4.2 below outlines the maximum current consumption of each significant component used in the system. The current consumption values are sourced from the relevant component datasheets based on the working operation as discussed in the sections above.

**Table 4.2:** Maximum current consumption of system components

Component	Part Number / Type	Max Current (mA)
Microcontroller	ESP32-C6-MINI-1	251
Temperature Sensor (x5)	MCP9600	$5 \times 1.5 = 7.5$
Accelerometer	LIS3DHTR	0.13
microSD Card	MEM2075-00-140-01-A	100
RGB LED (all ON, worst case)	Red + Green + Blue	$1.5 + 0.35 + 0.2 = 2.05$
Buzzer	Piezo buzzer	5
3.3V Regulator	TPS785-Q1	0.045
Miscellaneous + Margin	—	50
<b>Total</b>		<b>416.7</b>

#### 4.7.1. Battery size selection

The system's estimated maximum current draw is calculated as 417 mA in Table 4.2. For three hours of continuous operation, this results in a required capacity of  $417 \text{ mA} \times 3 \text{ h} = 1251 \text{ mAh}$ .

A 2000 mAh Li-Po battery is chosen to cover regulator losses, battery aging, and peak current needs. This ensures consistent performance and increases operating life beyond the minimum need.

### 4.8. Pinout Summary

**Table 4.3:** MCU pin assignments to sensors and interfaces

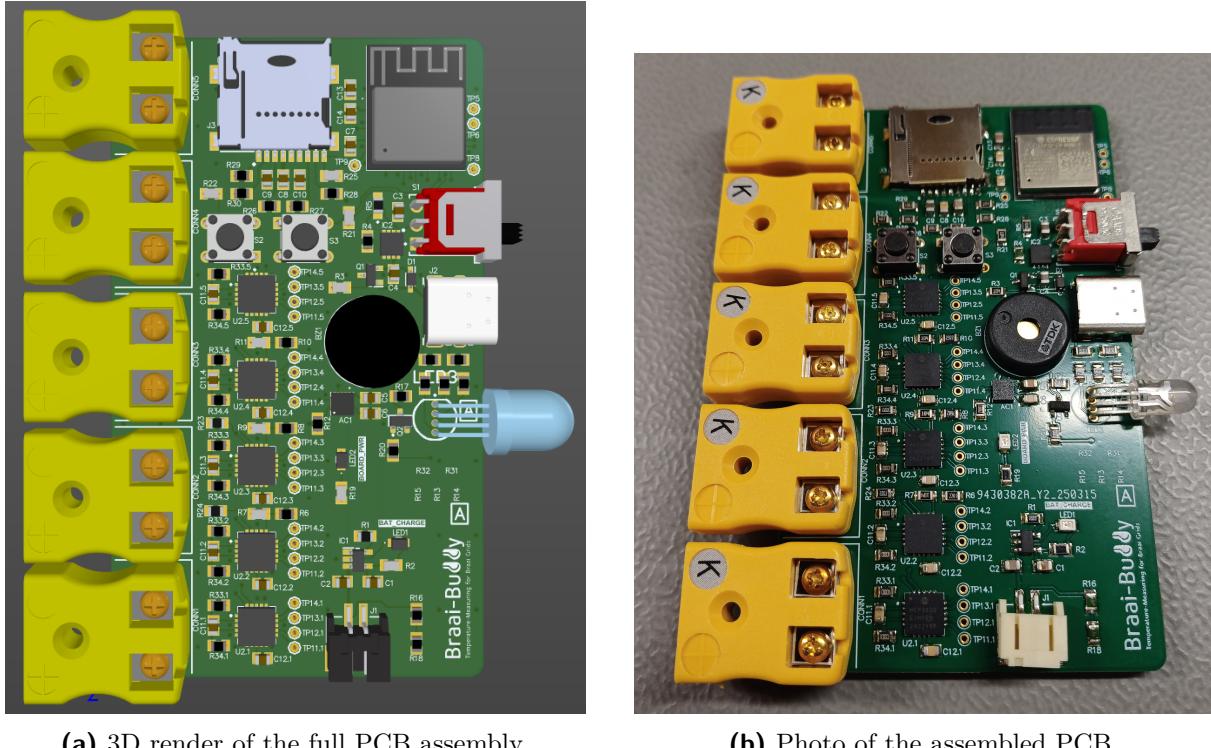
Signal	ESP32 Pin	Connected Device
IO6 / SDA	IO6	MCP9600, LIS3DHTR
IO7 / SCL	IO7	MCP9600, LIS3DHTR
IO2	IO2	RGB LED (Red)
IO3	IO3	RGB LED (Green)
IO4	IO4	RGB LED (Blue)
IO14	IO14	Buzzer
IO18 / SDIO CMD	IO18	SD Card
IO19 / SDIO CLK	IO19	SD Card
IO20 / SDIO D0	IO20	SD Card
IO21 / SDIO D1	IO21	SD Card
IO22 / SDIO D2	IO22	SD Card
IO23 / SDIO D3	IO23	SD Card
IO5 (GPIO)	IO5	SD_CD (detect)
EN	EN	Enable (Power control)
IO13	IO13	USB+
IO12	IO12	USB-

### 4.9. PCB Stack-up and Assembly

A 6-layer PCB was selected to support signal integrity and thermal distribution:

- Layer 1: Signal
- Layer 2: Ground
- Layer 3: Power
- Layer 4: Signal
- Layer 5: Ground
- Layer 6: Signal

The board was manufactured by JLCPCB. Ordering a fully populated board was initially considered for simplicity, but manual assembly was ultimately required due to unavailability of certain components. A 3D render of the PCB assembly and a photo of the actual assembled PCB can be seen in Figure 4.9.



**Figure 4.9:** Comparison between the 3D render and the actual assembled PCB.

## 4.10. Summary

This chapter focused on the detailed hardware design, outlining the circuit-level considerations required to operate and integrate each component effectively. The individual circuits were combined into a complete system and implemented on a custom-designed and assembled PCB.

# Chapter 5

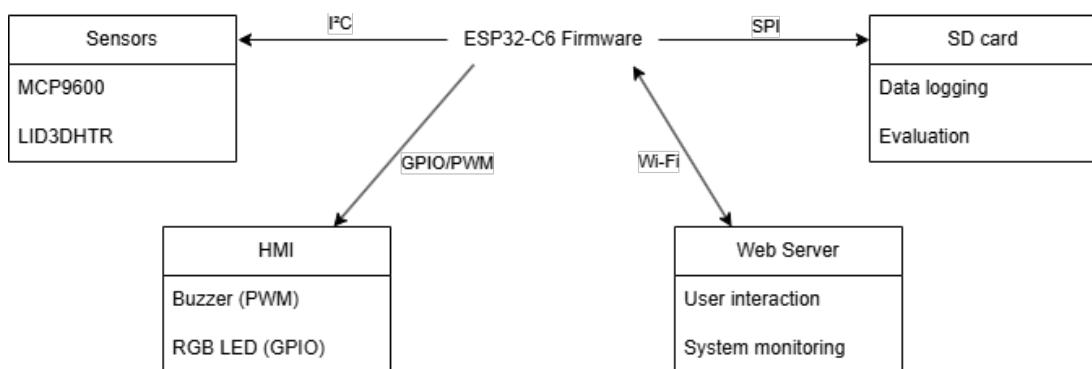
## Detailed Software Design

This chapter provides a detailed overview of the software architecture and operation of the system. The code executes on the ESP32-C6, interfacing with hardware components and serving a web-based GUI to clients via Wi-Fi. The software comprises firmware routines in C++ and embedded web content in HTML, CSS, and JavaScript.

### 5.1. System Architecture

The system software is structured around three main layers:

- **Hardware abstraction and drivers:** Interface to sensors (temperature, accelerometer) and peripherals (buzzer, RGB LED, SD card).
- **Task scheduler and control logic:** Handles periodic sensor polling, sequence execution, and communication management.
- **Web interface backend:** Facilitates user interaction via a local network.



**Figure 5.1:** Software architecture overview.

### 5.2. Global Structures

`Globals.h` defines shared structs for sensor readings and system status:

- `SensorData` contains temperature, orientation, and battery voltage readings.
- `WebSystemState` stores probe logs and sequence status for GUI communication.

## 5.3. Startup and Setup Sequence

On boot, the system:

1. Initializes serial debugging.
2. Sets up I<sup>2</sup>C peripherals: 5x MCP9600 thermocouple sensors and LIS3DHTR accelerometer.
3. Initializes RGB LEDs and buzzer pins.
4. Mounts SPIFFS and SD card filesystem.
5. Starts a WiFi AP.
6. Configures hardware timers for task scheduling.
7. Starts the IP address communication handling.

## 5.4. Task Scheduling

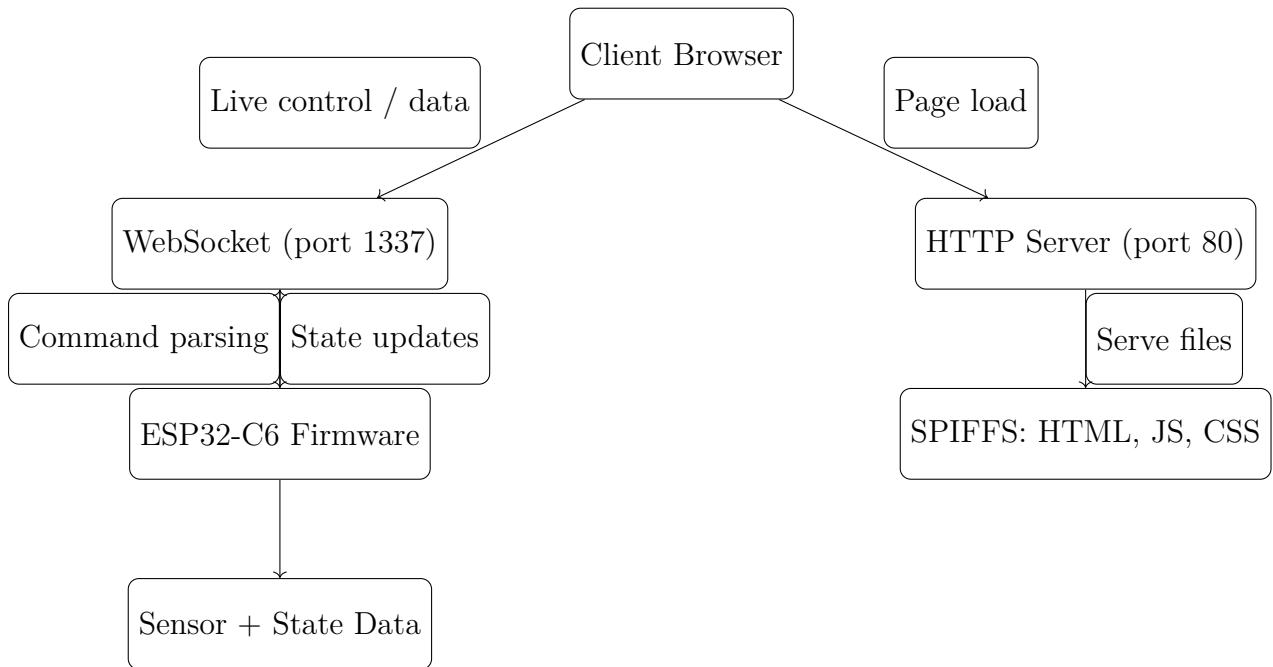
A 1ms timer interrupt sets flags for:

- 1Hz: Reads temperatures, logs sequences.
- 20Hz: Reads accelerometer.
- 100Hz: Handles IP LED/buzzer sequence.
- 2s: Reads the battery voltage.

## 5.5. Web Interface Backend

The backend uses `ESPAsyncWebServer` and `WebSocketsServer`. It:

- Serves HTML files for the GUI.
- Handles dynamic parameter exchange (temperature, sequences).
- Responds to probe test selection and grid turn timing.



**Figure 5.2:** Web Interaction Flow Between Client and ESP32

## 5.6. Temperature and Sequence Logic

Each probe channel can execute a test sequence. Tests are defined per-channel via the web UI. On sequence start:

1. The selected sequence (e.g., Boil Test) is saved.
2. At 1Hz, temperatures are logged to SD.
3. Results are pushed to the webpage.

## 5.7. Data Logging

Data is logged to `/temp_log.csv` on the SD card. First-time boot clears the file and adds headers. Data includes timestamp and all five probe readings.

## 5.8. Client Interface

`index.html` and `NetworkConnection.html` are served to clients. The UI allows:

- Connecting to local WiFi.
- Initiating tests.
- Monitoring sensor values and plots.

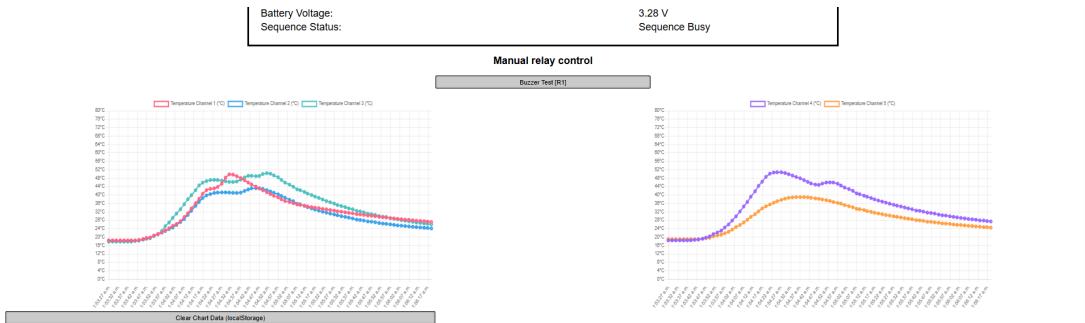
WiFi Status: Connected to Espressif WiFi

Forget WiFi network Connect to a WiFi network

### Braai Buddy

Probe 1:	Select Test
Probe 2:	Select Test
Probe 3:	Select Test
Probe 4:	Select Test
Probe 5:	Select Test
Grid Turning Time: 180 seconds	<input type="range"/>
<input type="button" value="Start Sequence"/>	
Temperature Sensor Channel 1: 18.50000°C Channel 1 Sequence:	
Temperature Sensor Channel 2: 18.562500°C Channel 2 Sequence:	
Temperature Sensor Channel 3: 18.25000°C Channel 3 Sequence:	
Temperature Sensor Channel 4: 18.75000°C Channel 4 Sequence:	
Temperature Sensor Channel 5: 18.875000°C Channel 5 Sequence:	
Accelerometer X Reading: 0.000000 Accelerometer Y Reading: 0.037500 Accelerometer Z Reading: 0.987500 Buzzer Status: N.A. Battery Voltage: 3.28 V Sequence Status: Sequence Busy	

(a) Top of GUI (Test selection and status).



(b) Bottom of GUI (Graphing and control).

**Figure 5.3:** Web GUI for Monitoring and Configuration

# Chapter 6

## Results

This chapter outlines the tests performed to verify both the hardware and software integration of the system. It includes defined tests, pass/fail criteria, and tables and screenshots of the test results.

### 6.1. Hardware subsystem verification

#### 6.1.1. Temperature measurement accuracy

The MCP9600 thermocouple amplifiers are tested by immersing the thermocouples in boiling and ice water.

**Table 6.1:** Temperature accuracy test results.

Probe	Expected Temp (°C)	Measured Temp (°C)	Max Error (°C)
CH1	100 / 0	99.81 / 0.38	0.38
CH2	100 / 0	99.25 / 0.25	0.75
CH3	100 / 0	100.12 / 0.44	0.44
CH4	100 / 0	99.62 / 0.88	0.88
CH5	100 / 0	99.19 / 0.69	0.81

#### 6.1.2. Accelerometer orientation detection

The LIS3DHTR accelerometer was tested by rotating the board to isolate each axis. Initial readings while the board was flat were approximately  $(X, Y, Z) = (0, 0, 1)$ , consistent with gravity acting on the Z-axis. Rotation around the Y-axis causes a change in the X-axis reading, rotation around the X-axis affects the Y-axis reading, and rotating around either the X or Y axis changes the Z-axis reading. A change of approximately 1 g was expected on the active axis. Results are shown in Table 6.2.

**Table 6.2:** Accelerometer Rotation Test Results

Rotation Axis	Expected Axis Change (g)	Measured (X, Y, Z)	Pass/Fail
Y (tests X)	~1g on X	(1.01, 0.025, -0.05)	Pass
X (tests Y)	~1g on Y	(-0.06, 1.03, 0.04)	Pass
X or Y (tests Z)	~1g on Z	(-0.96, 0.05, -0.01)	Pass

### 6.1.3. Battery Voltage Readout

Verify ADC-calculated voltage level from resistor divider.

**Table 6.3:** Battery Voltage Calibration

Measured Voltage (Multimeter)	ADC Reported Voltage	Error (%)
4.12	3.23	21.6

### 6.1.4. Voltage Rails Verification

Voltage levels on the 3.3V rail are verified at the supply pins for several components.

**Table 6.4:** Voltage Rail Measurements

Test Point	Expected Voltage (V)	Measured Voltage (V)
3.3V Regulator Output	3.3	3.37
ESP32 VDD Pin	3.3	3.37
MCP9600 VDD Pin	3.3	3.37
LIS3DHTR VDDIO Pin	3.3	3.37

### 6.1.5. Total Board Current Consumption

The average current drawn from the battery is measured under various system loads.

**Table 6.5:** System Power Consumption

Condition	Current Draw (mA)	Estimated Battery Life (2000mAh)
Idle (WiFi off, sensors idle)	30	66.7 hours
Full Sequence Active (Logging + WiFi + Sensors)	170	11.7 hours

### 6.1.6. LED and Buzzer Functional Test

The HMI circuits are tested by implementing code that sequentially toggles each individual RGB LED pin, all RGB LED pins together (white light) and then plays a 1 second 3 kHz tone through the buzzer.

**Table 6.6:** Output Peripheral Verification

Peripheral	Expected Behaviour	Observed Behaviour
Red LED	Lights when triggered	Visibly bright in daylight
Green LED	Lights when triggered	Visibly bright in daylight
Blue LED	Lights when triggered	Visibly bright in daylight
White LED (all RGB pins high)	Lights when triggered	Visibly bright in daylight
Buzzer	Beep tone played	Audible from adjacent room (broken line of sight)

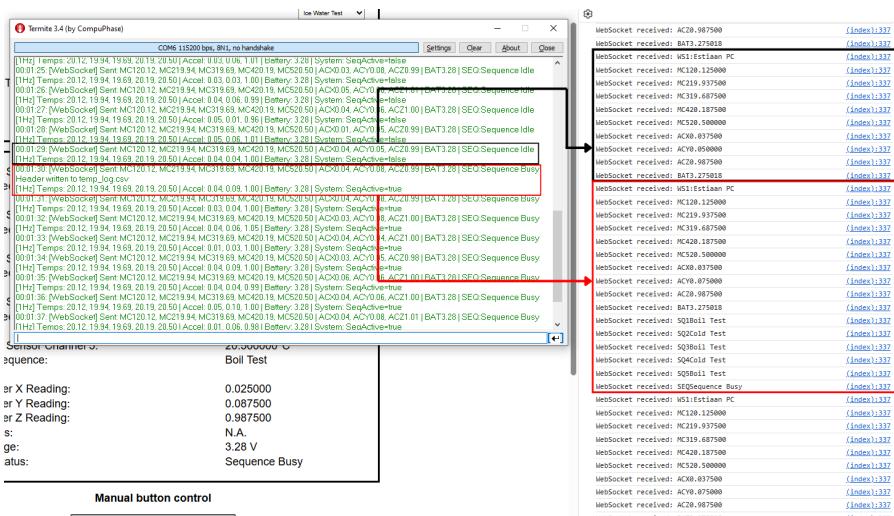
### 6.1.7. SD Card Insert Detection

To verify the SD card insert detection feature, the card detect pin was measured using a multimeter both with and without an SD card inserted. In both cases, the pin remained at ground level, indicating no change in logic state. This confirms that the hardware-based card detect circuit is not functioning as intended.

## 6.2. Software-Hardware Integration Testing

### 6.2.1. Web Interface Data Retrieval

Figure 6.1 shows that the serial monitor and WebSocket console confirm consistent 1 Hz data retrieval from the ESP32 to the web app. The serial monitor appends timestamps, which verify the update rate, while matched values (e.g., accelerometer readings) confirm real-time synchronization.

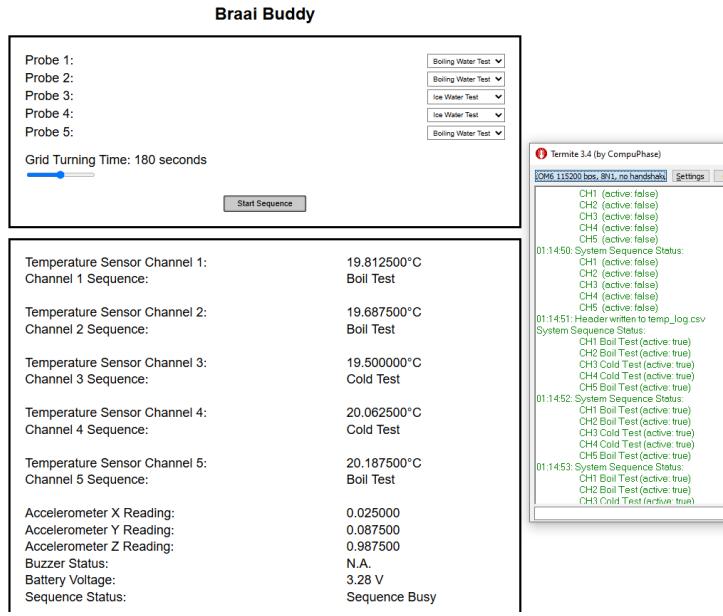


**Figure 6.1:** Data retrieval serial monitor and WebSocket console output.

### 6.2.2. Sequence Execution Test

The sequence was triggered via the web GUI by selecting actions for each probe and pressing “Start Sequence”. As shown in Figure 6.2, the GUI updates and serial monitor

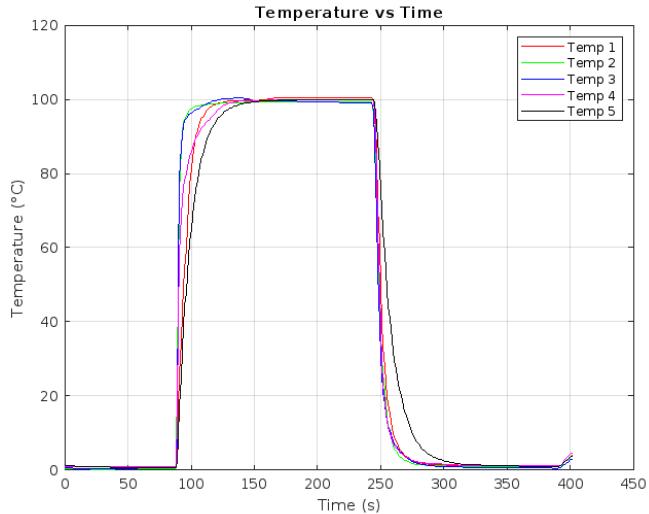
output confirm that the selected sequences were correctly received and executed by the system.



**Figure 6.2:** Sequence execution test confirmation.

### 6.2.3. Temperature Logging to SD Card

Temperature logging was verified by moving the probes from ice water to boiling water and back. As shown in Figure 6.3, all five channels captured the expected rise and fall in temperature, confirming accurate logging and proper system response.



**Figure 6.3:** Temperature vs. time plot from SD card log, showing probes transitioning from ice water to boiling water and back to ice water.

#### 6.2.4. Task Scheduler Operation

Confirm task flags are operating at expected frequencies by having ESP32 output a unique message over the serial monitor when the respective scheduled tasks are called.

**Table 6.7:** Task Scheduler Timing

Task	Expected Frequency	Serial Confirmation (Yes/No)
0.5Hz Batt Update	0.5 Hz	Yes
1Hz Temp Update	1 Hz	Yes
20Hz Accel Update	20 Hz	Yes
100Hz IP LED/Buzzer	100 Hz	Yes

### 6.3. Full Test Sequence Verification

A comprehensive end-to-end test, referred to as the “Boil and Freeze” sequence, was conducted to validate the system’s integrated hardware and software functionality under real operating conditions. This test sequence verifies temperature sensing, orientation detection, output signaling, sequence handling, and SD card data logging.

1. All thermocouple probes were first submerged in boiling water (approximately 100°C). The system continuously monitored probe temperatures until all were within a configured high-temperature threshold.
2. Once this condition was satisfied, the system activated both the buzzer and RGB LED to signal that the device must be flipped over.
3. Upon detecting a Z-axis inversion via the onboard accelerometer, the system automatically deactivated the buzzer and LED, indicating successful orientation change and progressing to the next phase.
4. The probes were then submerged in ice water (approximately 0°C), and temperature monitoring resumed until all probes fell within the configured low-temperature threshold.
5. Once the freezing condition was met, the buzzer and RGB LED were again triggered to instruct a second board flip.
6. Detection of the second orientation change via the accelerometer marked the completion of the test sequence.

Throughout the sequence, all relevant sensor readings, orientation changes, and system status indicators were logged to the onboard SD card for post-analysis. The successful

execution of this procedure confirms correct system coordination, reliability of sensor readings, and robustness of the sequence logic.

## 6.4. Task Scheduler Operation

Confirm task flags are operating at expected frequencies by having ESP32 output a unique message over the serial monitor when the respective scheduled tasks are called.

**Table 6.8:** Task Scheduler Timing

Task	Expected Frequency	Serial Confirmation (Yes/No)
0.5Hz Batt Update	0.5 Hz	Yes
1Hz Temp Update	1 Hz	Yes
20Hz Accel Update	20 Hz	Yes
100Hz IP LED/Buzzer	100 Hz	Yes

## 6.5. Summary

The above tests cover all interfaces and logic blocks of the system. Any deviations from expected values will be used to guide future system tuning and debugging.

# Chapter 7

## Summary and Conclusion

This project set out to develop a smart braai monitoring and notification system, capable of real-time temperature logging, user alerts, and web-based interaction. The completed proof-of-concept system integrated all critical hardware subsystems successfully and demonstrated reliable performance under testing.

### 7.1. Summary of Results

All hardware components performed as expected, with full functional verification of the following:

- **Temperature sensing:** Five MCP9600 thermocouple amplifier circuits accurately read data from K-type thermocouples. These readings were stable and in line with expected values during boiling and freezing tests.
- **Accelerometer:** The LIS3DHTR successfully detected grid flips based on Z-axis orientation changes, which were used as triggers in the test sequence.
- **Power system:** Battery voltage readings and rail verification confirmed the power subsystem delivered stable 3.3 V to all logic components, with current draw within budget for a minimum 2-hour battery life.
- **HMI outputs:** RGB LEDs and buzzer were correctly driven by GPIOs, allowing for user alerts through visual and audio feedback.
- **Data logging:** The SD card interface operated reliably using the ESP32-C6's SDIO peripheral, with temperature data successfully recorded at 1 Hz intervals.
- **Web interface:** Real-time probe data and sequence configuration were accessible via a WiFi-hosted HTML dashboard using WebSockets.

### 7.2. Discussion of Limitations

Although the hardware functioned as intended, time constraints prevented full implementation of the cooking profile system. The sequence logic currently supports manual testing

modes (e.g., Boil Test, Cold Test), but lacks:

- Automatic profile progression based on food type
- Preset management via the web interface
- Notifications based on cooking levels

These features were scoped for a future iteration. Regardless, the current implementation forms a solid foundation for extending sequence logic and improving user interaction.

### **7.3. Future Improvements**

To evolve the prototype into a product-ready system, the following enhancements are suggested:

- Implementation of temperature profiles for different meats with automatic progression
- User configuration of profile parameters and alert thresholds
- Cloud-based data sync or remote monitoring capabilities
- Waterproofing and heat shielding for outdoor reliability

The successful validation of the hardware enables confidence in continuing with software expansion.

### **7.4. Conclusion**

The system successfully meets all core hardware and integration objectives. It demonstrated real-time data logging, user interaction through web and HMI interfaces, and robust operation during the boil-freeze test. While full cooking profile automation was not completed, the platform is extensible and ready for further development.

# Bibliography

- [1] J. Clements, “The best bluetooth bbq thermometers for 2025,” <https://www.smokedbbqsource.com/best-bluetooth-bbq-thermometers/>, November 2023.
- [2] R. Bilow, “We put 21 wireless grill thermometers to the test—these 6 were the best,” <https://www.serioouseats.com/best-wireless-grill-thermometers-7568246>, May 2025.
- [3] F. Labs, “Fireboard 2,” <https://www.fireboard.com/shop/fireboard-2/>.
- [4] ——, “Fireboard 2 pro,” <https://www.fireboard.com/shop/fireboard-2-pro/>, February 2025.
- [5] ——, *FBX2 International Digital User Guide*, FireBoard Labs, Kansas City, MO, USA, May 2020. [Online]. Available: [https://s3-us-west-2.amazonaws.com/fireboard-wordpress-media/wp-content/uploads/2020/06/10124305/FBX2-International-Digital-User-Guide\\_20200522.pdf](https://s3-us-west-2.amazonaws.com/fireboard-wordpress-media/wp-content/uploads/2020/06/10124305/FBX2-International-Digital-User-Guide_20200522.pdf)
- [6] Allrecipes, “Our favorite meat thermometers take the guesswork out of cooking,” <https://www.allrecipes.com/longform/best-meat-thermometers/>, April 2025.
- [7] MEATER, “Meater pro xl,” <https://store-us.meater.com/products/meater-pro-xl>, February 2025.
- [8] A. Labs, “Meater® smart meat thermometer,” [https://play.google.com/store/apps/details?id=com.apptionlabs.meater.app&hl=en\\_ZA](https://play.google.com/store/apps/details?id=com.apptionlabs.meater.app&hl=en_ZA), March 2025.
- [9] City Fire Protection, “The temperature of fire,” <https://www.cityfire.co.uk/news/the-temperature-of-fire/>, May 2025.
- [10] Target Fire Protection, “What is the temperature of fire?” <https://www.target-fire.co.uk/resource-centre/what-is-the-temperature-of-fire/>, May 2025.
- [11] Weber-Stephen Products LLC, “2023 generic bbq and temperature guide,” <https://www.weber.com/cms-remote-assets/2023-Generic-bbq-and-temp-guide-R2-09-23.pdf>, September 2023.
- [12] Stanbroke, “The only steak temperature chart you’ll need,” <https://steakschool.com/learn/steak-temperature-chart/>, 2021.
- [13] Ametherm, “4 most common types of temperature sensor,” <https://www.ametherm.com/blog/thermistors/temperature-sensor-types>, December 2017.

- [14] TE Connectivity, “Understanding rtds,” <https://www.te.com/en/products/sensors/temperature-sensors/resources/understanding-rtds.html>, 2016, accessed: May 28, 2025.
- [15] M. Soltero, “What are you sensing? pros and cons of four temperature sensor types,” <https://www.ti.com/lit/ta/ssztca2/ssztca2.pdf>, August 2015, accessed: May 25, 2025.
- [16] T. E. H. Corporation, “True rtd accuracy – what you need to know!” <https://www.tempco.com/Tempco/Blog/True-RTD-Accuracy--What-You-Need-to-Know.htm>, September 2022, [Online].
- [17] Dwyer Instruments, Inc. (2025) Thermistor. Accessed: 2025-05-29. [Online]. Available: <https://www.dwyeromega.com/en-us/resources/thermistor>
- [18] M. Soltero, “What are you sensing? pros and cons of four temperature sensor types,” Texas Instruments, Tech. Rep., August 2015, accessed: 2025-05-29. [Online]. Available: <https://www.ti.com/lit/ta/ssztc25/ssztc25.pdf>
- [19] R. DeRose. (2021, Jun.) Thermocouple basics: Using the seebeck effect for temperature measurement. All About Circuits. Accessed: 2025-05-29. [Online]. Available: <https://www.allaboutcircuits.com/technical-articles/thermocouple-basics-using-the-seebeck-effect-for-temperature-measurement/>
- [20] Atlas Scientific, “4 types of temperature sensors,” <https://atlas-scientific.com/blog/types-of-temperature-sensors/>, 2024, accessed: May 28, 2025.
- [21] E. Holman, “A basic guide to thermocouple measurements,” Texas Instruments, Tech. Rep. SBAA274A, Apr. 2022, accessed: 2025-05-29. [Online]. Available: <https://www.ti.com/lit/an/sbaa274a/sbaa274a.pdf>
- [22] C. Massiot. (2022, Mar.) Types of temperature sensors and their working principles. AZoSensors. Accessed: 2025-05-29. [Online]. Available: <https://www.azosensors.com/article.aspx?ArticleID=2852>
- [23] M. Samiullah, M. Z. Irfan, and A. Rafique, “Microcontrollers: A comprehensive overview and comparative analysis of diverse types,” Preprint, Sep. 2023. [Online]. Available: <https://doi.org/10.31224/3228>
- [24] Espressif Systems. (2025) Product selector. Accessed: 2025-05-29. [Online]. Available: <https://products.espressif.com/#/product-selector?names=>
- [25] P. Gralla. (2023, Dec.) Wireless standards: 802.11a, 802.11b/g/n, and 802.11ac. Lifewire. Accessed: 2025-05-29. [Online]. Available: <https://www.lifewire.com/wireless-standards-802-11a-802-11b-g-n-and-802-11ac-816553>

- [26] Digi-Key Electronics. (2025) Rf transceiver modules and modems. Accessed: 2025-05-29. [Online]. Available: <https://www.digikey.co.za/en/products/filter/rf-transceiver-modules-and-modems/872?s=N4IgTCBcDaIKYGcAOBmMBaAxgNnQWwEsA7AkAXQF8g>
- [27] Espressif Systems. (2024, Feb.) Esp32-c6-mini-1/mini-1u datasheet. Accessed: 2025-05-29. [Online]. Available: [https://www.espressif.com/sites/default/files/documentation/esp32-c6-mini-1\\_mini-1u\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-c6-mini-1_mini-1u_datasheet_en.pdf)
- [28] SparkFun Electronics. (2024) Sparkfun qwiic pocket development board - esp32-c6. Accessed: 2025-05-29. [Online]. Available: <https://www.sparkfun.com/sparkfun-qwiic-pocket-development-board-esp32-c6.html>
- [29] Microchip Technology Inc. (2023, Jan.) Mcp73831/2 li-ion/li-polymer charge management controller. Revision H. [Online]. Available: <https://ww1.microchip.com/downloads/en/DeviceDoc/MCP73831-Family-Data-Sheet-DS20001984H.pdf>
- [30] Digi-Key Electronics. (2025) Battery chargers. Accessed: 2025-05-29. [Online]. Available: <https://www.digikey.co.za/en/products/filter/battery-chargers/781?s=N4IgTCBcDaILIGEAKB2AzADjQRhAXQF8g>
- [31] SparkFun Electronics. (2020) Battery technologies. Accessed: 2025-05-29. [Online]. Available: <https://learn.sparkfun.com/tutorials/battery-technologies/all>
- [32] Monolithic Power Systems. (2024) Types of voltage regulators. Accessed: 2025-05-29. [Online]. Available: <https://www.monolithicpower.com/en/learning/resources/voltage-regulator-types>
- [33] Richtek Technology Corp. (2015) Rt9080-33gj5: 600ma low dropout voltage regulator. Accessed: 2025-05-29. [Online]. Available: <https://datasheet.octopart.com/RT9080-33GJ5-Richtek-datasheet-157196975.pdf>
- [34] Texas Instruments. (2021, Apr.) Tps785-q1: 500-ma, low-iq, low-dropout regulator for automotive applications. Revision E. [Online]. Available: <https://www.ti.com/lit/ds/symlink/tps785-q1.pdf>
- [35] FireBoard Labs. (2025) Type-k food probe. Accessed: 2025-05-29. [Online]. Available: <https://www.fireboard.com/shop/type-k-food-probe/>
- [36] Labfacility Ltd. (2022) Pcb mounting thermocouple connector socket am-k-pcb type k ansi. Accessed: 2025-05-29. [Online]. Available: <https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/6465/XE-1139-001.pdf>

- [37] Digi-Key Electronics. (2025) Xe-1384-001 – miniature thermocouple connector plug bm-k-m type k bs. Accessed: 2025-05-29. [Online]. Available: <https://www.digikey.com/en/products/detail/labfacility-ltd/XE-1384-001/25605900?&s=N4IgTCBcDaIBoFEC0BGAzADgCxIAy5RAF0BfIA>
- [38] Analog Devices, Inc. (2016, Dec.) Max31856 thermocouple-to-digital converter. Revision 4. [Online]. Available: <https://www.analog.com/media/en/technical-documentation/data-sheets/MAX31856.pdf>
- [39] Microchip Technology Inc. (2023, Jul.) Mcp960x/mcp96l0x/mcp96rl0x thermocouple emf to °c converter with linearization. Revision F. [Online]. Available: <https://ww1.microchip.com/downloads/en/DeviceDoc/MCP960X-L0X-RL0X-Data-Sheet-20005426F.pdf>
- [40] ROHM Semiconductor. (2021, Nov.) Kx122-1037 specifications. Revision 6.0. [Online]. Available: <https://fscdn.rohm.com/kionix/en/datasheet/KX122-1037-Specifications-Rev-6.0.pdf>
- [41] STMicroelectronics. (2023, Jun.) Lis2de12: Mems digital output motion sensor. Revision 9. [Online]. Available: <https://www.st.com/resource/en/datasheet/lis2de12.pdf>
- [42] ——. (2022, Mar.) Lis3dh: Mems motion sensor: ultra-low-power high-performance 3-axis ‘nano’ accelerometer. Revision 8. [Online]. Available: <https://www.st.com/resource/en/datasheet/lis3dh.pdf>
- [43] Anderson Power Products. (2022) Spec pak® sealed connector series – 2 position inline. [Online]. Available: <https://configured-product-images.s3.amazonaws.com/Datasheets/500A.pdf>
- [44] Robotics.org.za. (2025) Rgb led 5mm common cathode (10 pack). [Online]. Available: <https://www.robotics.org.za/RGB-LED-05-CC>
- [45] ISL Products International Ltd. (2022) Piezo buzzers vs. magnetic buzzers: What’s the difference? [Online]. Available: <https://islproducts.com/design-note/piezo-buzzers-vs-magnetic-buzzers/>
- [46] TDK Corporation. (2025) Ps1240p02ct3 – piezo buzzer. [Online]. Available: [https://product.tdk.com/en/search/sw\\_piezo/sw\\_piezo/piezo-buzzer/info?part\\_no=PS1240P02CT3](https://product.tdk.com/en/search/sw_piezo/sw_piezo/piezo-buzzer/info?part_no=PS1240P02CT3)
- [47] Global Connector Technology. (2022) Mem2075: Microsd connector – push-push, smt. Revision A4. [Online]. Available: <https://gct.co/files/specs/mem2075-spec.pdf>

- [48] Samesky Devices. (2023) Uj20-c-h-g-smt-tr: Usb type-c receptacle connector. [Online]. Available: <https://www.sameskydevices.com/product/resource/uj20-c-h-g-smt-tr.pdf>
- [49] A. Picugins. (2023) All about usb-c: Example circuits. Accessed: 2025-05-28. [Online]. Available: <https://hackaday.com/2023/08/07/all-about-usb-c-example-circuits/>
- [50] Wikipedia. (2025, May) Usb-c. [Online]. Available: <https://en.wikipedia.org/wiki/USB-C>
- [51] JLCPCB. (2023) Differential pair in pcb design. [Online]. Available: <https://jlcpcb.com/blog/pcb-basics-differential-pair-in-pcb-design>
- [52] CircuitBread. (2021) The forward voltages of different leds. [Online]. Available: <https://www.circuitbread.com/ee-faq/the-forward-voltages-of-different-leds>
- [53] Digi-Key Electronics. (2025) Mmbt3904 – bipolar (bjt) transistor npn 40 v 200 ma 625 mw surface mount sot-23. [Online]. Available: <https://www.digikey.co.za/en/products/detail/diotec-semiconductor/MMBT3904/13163698>
- [54] Adafruit Industries. (2023) Adafruit mcp9600 i<sup>2</sup>c thermocouple amplifier. [Online]. Available: <https://learn.adafruit.com/adafruit-mcp9600-i2c-thermocouple-amplifier/downloads>
- [55] Acme Systems, “How to design the microsd circuitry,” [https://www.acmesystems.it/pcb\\_microsd](https://www.acmesystems.it/pcb_microsd), 2025.

# Appendix A

## Planning

**Table A.1:** Project Planning Schedule.

Month	Day	Planned Work
February	16	Research on related work and identification of system capabilities
February	20	System overview completed and components selected for sourcing
February	26	Components finalized to begin PCB layout design
March	9	PCB layout finalized and ordered; components ordered
March	18	Software flow and GUI layout finalized
March	22	PCB assembled
March	29	Software for individual components initialized and verified
April	4	Timers and flags implemented to enable synchronized component operation
April	9	Sequential SD card storage of sensor data implemented
April	13	Wi-Fi connection and WebSocket communication established; sensor values displayed on GUI
April	23	Temperature data graphed in GUI
May	2	GUI control triggers sequences on ESP32; system retains state across page reloads
May	9	Full system sequences tested and results captured
May	12	Report outline, introduction, and background sections completed
May	18	System design, detailed hardware, and detailed software sections completed
May	24	Test procedures, results, and conclusion sections finalized
May	30	Supervisor feedback incorporated into report
June	3	Final report handed in

# Appendix B

## Outcomes compliance

**Table B.1:** Graduate Attribute Compliance (Part 1)

ECSA Graduate Attribute	How Achieved in This Project
GA 1: Problem Solving – Identify, formulate, analyse and solve complex engineering problems creatively and innovatively.	System design (Section 3), Detailed hardware (Section 4) and software (Section 5) development involved solving the unique challenge of automating braai monitoring. Specific examples include heat distribution mapping and grid-turn alert logic (Section 6.3).
GA 2: Application of Scientific and Engineering Knowledge – Apply knowledge of mathematics, natural sciences, engineering fundamentals and an engineering speciality.	Selection and application of temperature sensors and voltage dividers (Sections 3.2.3 and 4.4) required analysis using engineering principles and electrical theory.
GA 3: Engineering Design – Perform creative, procedural and non-procedural design and synthesis of components, systems, engineering works, products or processes.	The system was designed to solve a real-world problem through an innovative multi-probe monitoring solution (Sections 3.1, 4.1–4.9), balancing cost, manufacturability, and functionality.
GA 4: Investigations, Experiments and Data Analysis – Demonstrate competence to design and conduct investigations and experiments.	Full system validation using the “Boil and Freeze” test and temperature accuracy testing (Sections 6.1.1 and 6.3) demonstrates effective experimentation and analysis.

**Table B.2:** Graduate Attribute Compliance (Part 2)

ECSA Graduate Attribute	How Achieved in This Project
GA 5: Engineering Methods, Skills and Tools, including Information Technology – Demonstrate competence to use appropriate engineering methods, skills and tools, including those based on information technology.	Used C++ and HTML/JS for firmware and GUI (Section 5), Altium for PCB design (Section 4.9), and analytical tools to match layout and interface specifications (Figures 4.2, 4.8).
GA 6: Professional and Technical Communication – Demonstrate competence to communicate effectively, both orally and in writing, with engineering audiences and the community at large.	A structured, professionally written report documents all system aspects. Visuals such as flowcharts, graphs, and diagrams support communication (Figures 5.2, 6.3, etc.).
GA 8: Individual Work – Demonstrate competence to work effectively as an individual.	The entire project and report writing was carried out independently, along with researching background information and system-level design and implementation across hardware, firmware, and frontend design.
GA 9: Independent Learning Ability – Demonstrate competence to engage in independent learning through well-developed learning skills.	Topics such as thermocouple circuitry, SDIO interfacing, and embedded web design were self-studied and successfully applied (Sections 2.3, 4.6, 5.5).

# Appendix C

## Combined Schematic

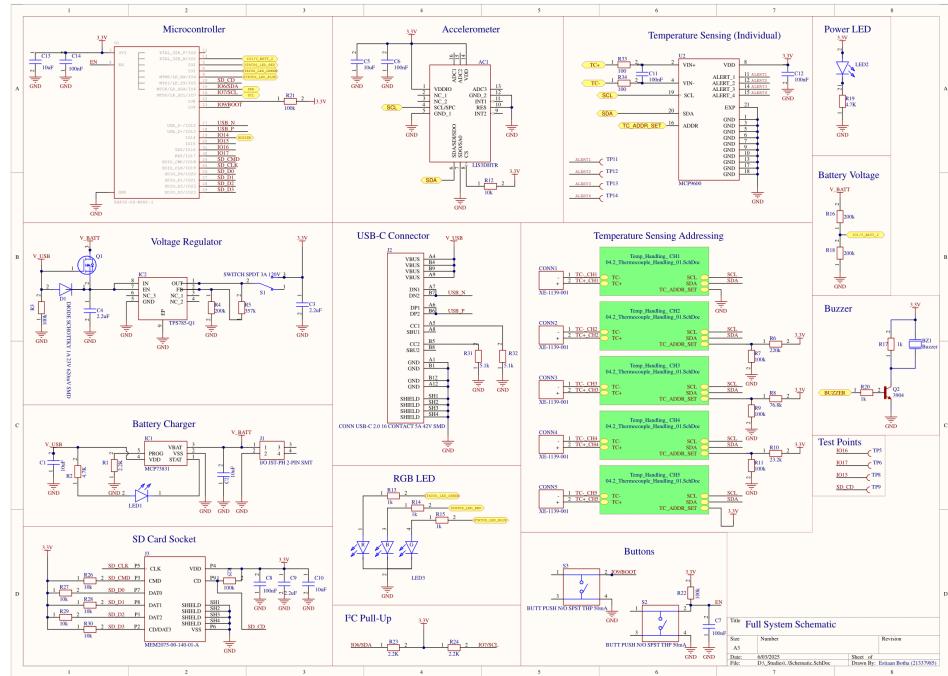


Figure C.1: The full schematic of the system.