Auckland University of Technology

ENEL602 Technical Report - Detailing the Design, Assembly and Testing of the Robot and Starger Projects

John Poirier

A technical report submitted to Auckland University of Technology
in partial fulfilment of the requirements for the degree of
Bachelor of Engineering Technology

June 2025

School of Engineering, Computer and Mathematical Sciences
Faculty of Design and Creative Technologies
Auckland University of Technology
New Zealand

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the qualification of any other degree or diploma of a university or other institution of higher learning.

John Poirier

Signature of student

Abstract

An abstract for a technical report provides a concise summary of the report's contents. It typically includes the following elements:

1. Background: Briefly explains the context and importance of the topic being investigated or the problem being addressed.

2. Objective(s): States the main goals or objectives of the study, indicating what the report aims to achieve or solve.

3. Methodology: Describes the methods and techniques used to conduct the research or project, including any experiments, data collection, or analysis.

4. Results: Summarises the key findings and outcomes of the study, highlighting the main conclusions or insights derived from the analysis.

5. Significance: Discusses the significance or implications of the results in relation to the broader field or industry, addressing the potential impact or application of the research.

6. Conclusion: Provides a brief summary of the overall report, reaffirming the main points and emphasising the value of the research or project.

The abstract is typically written in a concise manner, using clear and specific language to ensure that readers can quickly understand the core aspects of the report without having to delve into the full document.

Contents

List of Tables

List of Figures

Acknowledgements

I would like to express my gratitude to *Michael Bunn and the AUT Technicians*.

Abbreviations

SMD Surface Mount Device.
LiDAR Light Detection and Ranging.
AV Autonomous Vehicle
CTC Clear Timer on Compare Match

Chapter 1

Introduction

This technical report is compiled as a requirement for completing the academic requirements for ENEL602 Semester One, 2025. It is intended to provide students with the foundation required to produce high-quality reports for laboratory work and final year projects.

1.1 Scope

This Report includes a description of the process to design and assemble each of the assigned projects within the course. Specifically:

i. **Starger Project - Rectified Power Supply**

ii. Starger Project - Analogue and Digital Tone Generator Circuit

iii. Starger Project - Robot Starter Circuit

iv. Robot Project - Line-Following Robot

The format required for each chapter of the report is provided, allowing the student to enter just the required information to complete each section. Such information includes descriptions of the design, experimental method, results and conclusions.

1.2 Contributions

The persons contributing to this report are listed below:

Contributor 1 *John Poirier*

1.3 Report Structure

The report is structured as follows:

Chapter 2 is the chapter that describes the process for collecting and processing of data using a "Reading Grid".

Chapter 3 overviews the design process for the Analogue and Digital Tone Generator Starger Project. This chapter specifically highlights any challenges and learnings from this phase of the course.

Chapter 4 overviews the assembly process and testing phases for the Analogue and Digital Tone Generator "Starger" Project. This chapter specifically highlights any challenges and learnings from this phase of the course.

Chapter 5 overviews the assembly process and testing phases for the Analogue and Digital Tone Generator "Starger" Project. This chapter specifically highlights any challenges and learnings from this phase of the course.

Chapter 6 overviews the assembly process and testing phases for the Robot Project. This chapter specifically highlights any challenges and learnings from this phase of the course.

Chapter 7 provides concluding remarks, and summarises the work carried out in this report, highlighting any key findings from the Literature Review, and various stages of the project.

Chapter 2

Information Collection and Processing

2.1 Introduction

In this chapter the process of collecting and processing information for a structured literature review to investigate whether the line following method used by the robots is a valid sensing technology for road vehicles. To explore this question, the "funnel" process is utilised in the information collection exercise in Section 2.3 and a case is made that the line following method is could plausibly be an approach utilising the "Reading Grid" and paraphrasing exercise in Section 2.4.

2.2 Information Collection: Overview

Information is collected from the Scopus database using a "funnel" technique that starts with a broad search and is iteratively refined to exclude irrelevant information. However, it is not so constrained as to exclude potentially relevant information. This is important as one does not wish to conflate learning with what one already "knows", as such a conflation is anathema to the pursuit of knowledge. To learn what one does not already know is the raison d'être for a literature review.

A literature review provides an overview of the current “state of knowledge” of a topic, highlighting what is, and is not discussed in academic literature.

A structured literature review is a method used to demonstrate that the information presented is an accurate representation of what has been studied, highlighting any gaps that may exist. A structured literature review utilises the results obtained from a search of an academic literature database using carefully constructed search strings.

The process of constructing search strings is demonstrated in the following sections, highlighting how the iterative approach allows for increased filtering to narrow results whilst ensuring that potentially relevant results are not excluded.

2.3 Information Collection: Search Strings

This section seeks to examine what technologies are presently used by vehicles for sensing and their capabilities. With an initial search, it is a good idea to begin with an assumption that one has zero knowledge of the topic being researched.

In this case, to establish the case for line-following as a suitable vehicle sensing technology for road vehicles, a good place to start is with Review articles. Review articles aggregate existing research an providing an overview of potential technologies to investigate further. An example search for review articles and selected results provide an initial insight into what vehicle sensing technologies are presently available.

2.3.1 Review Article Search Example

Database: Scopus

Search within	Article title, Abstract, Keywords
Search documents	(vehicle AND sensing AND technology) AND NOT (uav OR "Unmanned Aerial Vehicle")
Subject area	Engineering
Document type	Review
Language	English

Some example papers of papers that are relevant that this search turns up:

1. *The Perception System of Intelligent Ground Vehicles in All Weather Conditions: A Systematic Literature Review*

   This paper specifically mentions the following sensor technologies:

   - Ultrasonic
   - Radar
   - LiDAR
   - Camera-Based (Vision Systems)

2. *Sensor Technologies for Intelligent Transportation Systems*

   This paper specifically mentions the following sensor technologies:

   - Radar
   - LiDAR
   - Camera-Based (Vision Systems)

3. *Multi-Object Detection and Tracking, Based on DNN, for Autonomous Vehicles: A Review* This paper specifically mentions the following sensor technologies:

   - Radar
   - LiDAR
   - Camera-Based (Vision Systems)

   This paper also discusses the concept of Sensor Fusion and the use of Deep Neural

Networks (DNN) machine learning algorithms for processing

An example of a paper from the search outside of scope is: *A comprehensive survey on vehicular networking for safe and efficient driving in smart transportation: A focus on systems, protocols, and applications*. As this paper focuses on inter-vehicle networking and supporting systems, with no direct examination of technologies related to vehicle sensing, it is outside the scope of this search.

2.3.2 Initial Search String

This search string was generated as an initial search for researching multi-sensor fusion technology as the vehicle sensing technology proposed to update the robot project: *sensor AND technology OR line-following OR road AND vehicles* between the years 2015-2025.

The search was limited to the following Subject areas:

1. Engineering
2. Computer Science

Number of Papers Returned by Query

This search query generated $26,881$ total results comprised of

- ✓ $13,117$ Conference Papers
- ✓ $10,685$ Journal Articles
- ✓ 965 Reviews
- ✓ $1,051$ Book Chapters
- ✓ 820 Conference Reviews

2.3.3 Final Search String

After refinement the final search string utilised to research multi-sensor fusion technology as the vehicle sensing technology proposed to update the robot project is: *autonomous AND sensor AND vehicle AND (LIMIT-TO (EXACTKEYWORD , "Sensor Fusion")* between the years 2015-2025.

The search was limited to the following Subject areas:

1. Engineering
2. Computer Science

Number of Papers Returned by Query

This search query generated 81 total results comprised of

- ✓ 42 Conference Papers
- ✓ 31 Journal Articles
- ✓ 4 Reviews
- ✓ 3 Book Chapters
- ✓ 0 Conference Reviews

2.4 Information Categorisation: Reading Grid

This section utilises a Reading Grid as a means of initially categorising the five papers selected from the search detailed in the previous section. The results are presented in Table 2.1 on the following page.

Table 2.1: Reading Grid – Categorised results from Section 2.3.3

Source →	*autonomous AND sensor AND vehicle AND (LIMIT-TO (EXACTKEYWORD , "Sensor Fusion")*				
Key Points ↓	Lei et al. (2024) [1]	Dai et al. (2024) [2]	Park et al. (2024) [3]	Vinoth et al. (2024) [4]	Jamuna et al. (2024) [5]
Problem	Growing demand for convenience, increasing labour costs, and low efficiency of existing solutions	Limiations in single-sensor systems, Autonomous vehicle reliance on environmental precision	Current solution struggles in low-light conditions due to RGB camera's reliance on ambient light	AVs require accurate detection, current solutions have noisy image processing and inaccurate pathing	Visibility issues for large vehicles, lack of automation and visibility for these vehicles
Results	Autonomous delivery vehicle capable of navigating dynamic indoor environments using multi-sensor fusion with reliable perception of objects and smooth pathing	Centre-point distance probability and intersection-over-union matching from LiDAR and camera sensing to improve tracking performance	Combining RGB and thermal imaging using image alignment algorithms and late-fusion techniques improved performance compared to unoptimized iterations	Utilising multi-sensor fusion, image quality enhancement, and multi-segmentation across different fields of view provided improved performance compared to conventional methods	Automatic side lamp activation system using an LDR sensor, improving safety and efficient autonomous operation
Limits	Mapping showed imperfections and required manual adjustments, limited resolution affecting localization precision, unvalidated results in more complex environments	Low-light environments and occlusion still present challenges, requires accurate calibration between sensors	Poor quality thermal imaging, mid-fusion techniques had slower inference speeds, and initial calibration of the alignment algorithm was needed	Energy efficiency is low, and fusion algorithms have latency implications in real-time constraints. Performance in extreme conditions is unverified	Low sensitivity compared to higher-cost solutions, less effective in quickly changing environments
Opportunities	Enhance the system for outdoors, implement higher-level precision and decision making in obstacle avoidance, improve map clarity and automation, broaden application	Enhancing fusion algorithms in varied lighting and traffic conditions, integrating additional sensor types	Generalized alignment algorithm, improved safety in night time driving, deployment in adverse weather	Tuning for real-world application, integration with native vehicle control systems, additional sensor usage, increased efficiency	Affordability provides scalability, integration with other autonomous systems of the vehicle

2.5 Chapter Summary

This chapter demonstrated information collection and categorisation utilising Scopus as the reference database. Section 2.3 discussed the process for collecting relevant information beginning with review articles to understand the topics currently the subject of research.Subsections 2.3.2 and 2.3.3 presented a summarised overview of how the results are improved utilising the “funnel” approach, ensuring relevant information is not missed. Section 2.4 introduced the “Reading Grid” as an example of a method for initial categorisation of information obtained from literature found from the respective search strings.

Chapter 3

Starger Project Design

3.1 Introduction

This chapter describes the process used to design the Analogue and Digital Tone Generator for the “Starger” Project – a portmanteau of Robot STARter and battery charGER – beginning with an overview of Altium 365. This discussion also overviews the process to create projects, the benefits of implementing version control, one challenge presented by Altium 365 allowing collaboration by default and how comments and tasks can be used to provide feedback. The remainder of the chapter details the role schematic and PCB documents play in realising an electronic circuit, along with the schematic and PCB documents generated during the design process.

3.2 Altium 365

In this section the aspects of Altium 365 that contribute to the design process are included under each of the following subsections

3.2.1 Project Creation

To create a project in Altium 365 it is recommended to create the project using the *web* interface rather than with the Altium *desktop* application. Using the *web* interface allows the project to be created in an existing folder rather than a new folder, which is the case when a project is created using the *desktop* application.

3.2.2 Version Control

Version control is *a means to document and store versions of a program or project.*

Some benefits of using it are *being able to return to previous versions, having a timeline of changes to the versions, easy documentation, and often times version control can be used for collaboration.*

3.2.3 Collaboration

Collaboration is allowed by default to those within AUT's workgroup. However, this is not always desired, as others can alter or copy work without permission of the project creator. To limit this behaviour folder and project level options can be changed.

3.2.4 Feedback - Comments and Tasks

To correct mistakes found in the design the lecturer is able to provide direct feedback to the student using comments and tasks. The difference between a comment and task is:

Comments Are suggestions with no specific course of action required.

Tasks Require an action to be completed and can be assigned to a role or specific person.

3.3 Starger Project Design Process

In this section, the following is discussed that explains the process to design the Tone Generator using Altium Designer. In subsection 3.3.1 the function of the schematic diagram is explained, along with the key functions of the power supply. Subsection 3.3.2 explains how the PCB relates to the schematic and fabrication.

3.3.1 Schematic Design

Schematic diagrams provide *logical* connections between the components, which allows the circuit to be understood and without this understanding it would not be possible to make the PCB document. Included in this section are the schematic diagrams the author created in Altium Designer during course and an explanation of each of the key functions.

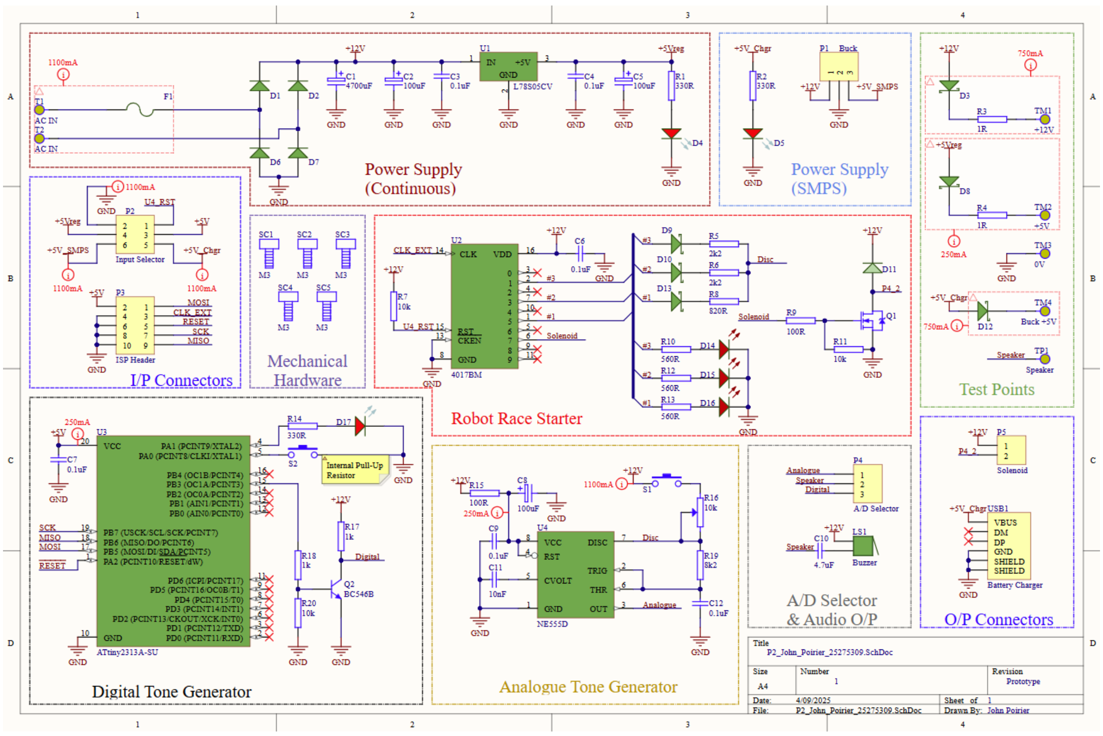

Figure 3.1: Starger Project Schematic drawn by John Poirier

Starger Schematic Power Supply Key Functions

Circuit Protection The components that protect the circuit by sacrificing themselves are *F1*, *R3* and *R4*. D3, D8 and D12 do not perform that function, instead they protect the circuit from *reverse currents*

±12VDC Supply The components responsible for providing the $\pm 12VDC$ supply are *D1*, *D2*, *D6*, *D7* and *C1*. As this is *unregulated* the voltage *does* change with variations in load current.

5VDC Supply The part number for the *Linear Voltage Regulator* (LVR) which provides the *regulated* supply which means the voltage *does not* change with variations in load current. **5VDC** is *L78S05CV*

Starger Schematic Tone Generator Key Functions

Circuit Protection The purpose of diodes D3, D8 and D12 are to *protect the circuit from, and prevent, reverse currents*

Analogue Clock Generation The *NE555D* is responsible for generating the analogue tones and is configured as *an Astable* multi-vibrator. It can be triggered by pressing *S1* or via the *U4 RST* NET in the Robot Race Starter, provided by the *4017BM* IC.

Digital Clock Generation The *ATtiny2313A* IC is responsible for generating the digital clock signal. It works by *comfiguring an internal counter on the ATtiny2313A to count up from zero and automatically reset when it matches a predefined value from the OCR1A. Every time this match occurs, the timer toggle pin PB3, which corresponds to OC1A, which produces a square wave. Using a prescaler and compare value defined by the user, a stable output frequency can be obtained to drive other components.*

Piezo Buzzer The *Buzzer* is responsible for converting the *transverse* to audible *longitudinal waves*. The Piezo-electric effect works by *applying physical stress on a material to produce an electric voltage across the materials surface, or vice versa. This can be used to make a material vibrate at a specific frequency by applying a voltage to it.*

3.3.2 PCB Design

PCB diagrams provide the *Physical* connections between the components, which allows the circuit to be fabricated. Included in this section is the PCB diagram the author created in Altium Designer during course.

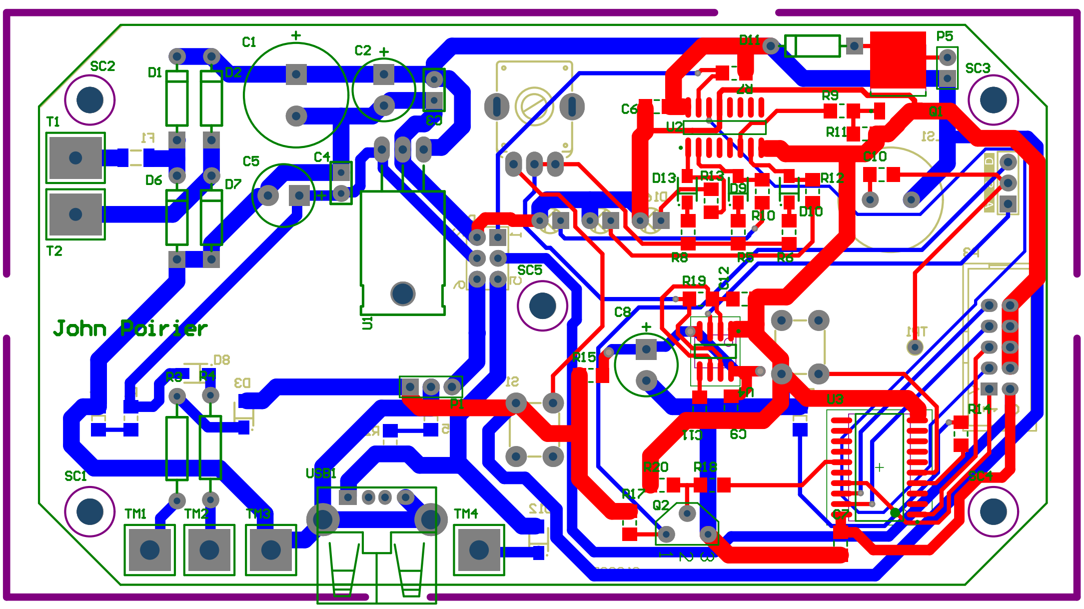

Figure 3.2: Starger PCB drawn by John Poirier

3.4 Discussion

In this section the challenges and successes and learnings the author experienced during the design process are presented.

3.4.1 Challenges

This subsection lists the challenges experienced during the design process of the Starger Project and how they were overcome.

i. *In the CLPS room, only top layer tracks were able to be used. This posed significant complications when designing the PCB compared to if both top and bottom layer tracks were able to be used. Clever planning and efficient placing of components must be used to overcome this challenge.*

ii. *Due to design limits, such as limited numbers of vias, specific track widths, and track-track and track-pad clearance rules, placement of compponents and routing of tracks must be very intentional. Planning the placement of comonents ahead and using the schematic as a reference of what connections will need to be made was necessary.*

3.4.2 Successes and Learnings

This subsection lists the highlights and learning the author experienced during the design process of the Starger Project.

i. *When designing a PCB, there are several rules of thumb that should be followed when possible. These include no right angles in tracks (except for buses), entering square pads orthogonally to the pad, using the most efficient routing wherever possible, keeping filtering capacitors as close to the power input pins of ICs as* *possible, and many other design techniques to be considered when designing any PCB.*

ii. *Rules of manufacturing for PCBs were learned, and considering these rules will remain important for any future PCB design. These rules include elements such as the width of the cutting tool, the layers of the board, and which layers components need to go on, especially SMD components.*

iii. *During design of the PCB, the final design was under the maximum number of allowed vias for the design.*

3.5 Chapter Summary

In this chapter, topics related to the design processes for the Analogue and Digital Tone Generator Starger Project were presented. In particular the following were highlighted:

- The use of Altium 365, the Altium Web Interface, and the Altium Desktop Application to create, manage, and control a project
- The value and use of version control
- The design process of the Starger project, including the schematic design and the PCB design
- The function of particular components within the circuit
- Challenges, successes, and learnings throughout the Starger project

Chapter 4

Starger Project Assembly

4.1 Introduction

This chapter describes the process undertaken to assemble the Starger Project. It explains the hazards present and precautions required when using the project room. The soldering process, including hand and SMD soldering using the pick and place machine.

4.2 Project Room Safety

This section describes the types of hazards that are present in the Project Room in each of the three areas, which are *the soldering stations, the drill press station, and the pick-and-place machines station*.

Soldering Station Hazards Key hazards are *extremely high heat, fumes, and static damage to components while soldering*

Soldering Station Precautions To minimise risks associated with those hazards these precautions are to be adhered to:

- *Be aware of where the soldering iron is at all times, and whenever the iron is not being used, make sure it is safely placed in the holder*

- *When soldering, do not directly breathe in any fumes if avoidable*
- *When soldering components, especially ICs, wear an anti-static strap to prevent damaging the ICs through static electricity*

Drill Press Station Hazards Key hazards are *fast spinning components, sharp components, and shrapnel.*

Dril Press Station Precautions To minimise risks associated with those hazards these precautions are to be adhered to:

- *Keep hands, fingers, hair, and any other body parts clear of the machine*
- *Do not wear any baggy or loose clothing that could get caught in the spinning drill*
- *Wear eye protection to keep your eyes safe from shrapnel*
- *Use a clamp to keep the component being drilled steady during drilling*

Pick-and-Place Station Hazards Key hazards are *toxic materials and high heat*

Pick-and-Place Station Precautions To minimise risks associated with those hazards these precautions are to be adhered to:

- *Wear proper safety gloves when working with the toxic materials, and only use safe materials to clean any toxic materials*
- *Dispose of any rubbish that has come in contact with toxic materials properly and in their proper disposal area*
- *After baking a PCB, let the PCB cool before removing it from the oven*

4.3 Starger Assembly Process

In this section, the following is discussed that explains the process to assemble the Starger Project.

4.3.1 Box Preparation

To prepare the supplied plastic box for use the first step was to *drill* the *holes* according to the diagram in Fig 4.1

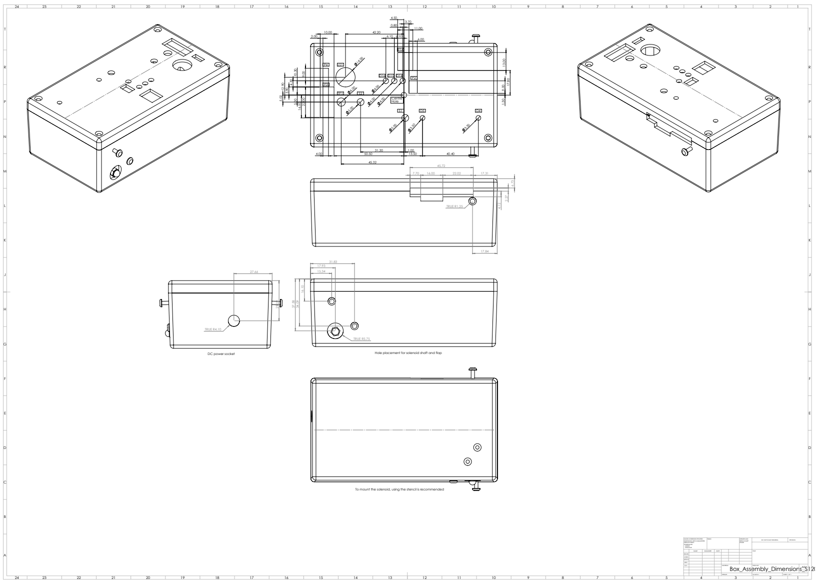

Figure 4.1: Starger Box drilling instructions from supplied in project instructions

4.3.2 PCB Assembly

When assembling the PCB it was recommended that the *high-profile components* be left to last because of their *size and height*. Instead low-profile components were soldered in first, taking care to note the correct *orientation* for polarised components, such as *diodes, ICs, and electrolytic capacitors*.

The following provides the steps used to solder each component into the PCB:

1. *Start by properly cleaning and tinning the soldering iron*
2. *For SMD components:*
   (a) *Choose one side that has the most SMD components, or the most complex SMD components (typically the side with ICs on it) and use the Pick-and-Place machine to solder this side of the PCB*

(b) *Use the Pick-and-Place machine to apply solder paste to the pads of the SMD components, placing enough solder paste to fully cover the pad when the components is placed. Then use the machine to place the components, taking careful consideration of the orientation of any polarised components*

(c) *Bake the PCB in the oven for the proper cycle, and let the PCB cool after the cycle before removing from the oven*

(d) *For the remaining SMD components, for each component, melt solder onto one pad of the PCB for that component, then heat that solder with the soldering iron while placing the component onto the pad*

(e) *Solder the other pad onto the PCB for the component*

3. *For Through-Hole Components:*

(a) *Place the component in the through holes on the PCB for the designated component, paying mind to which side of the PCB the component should be on and orientation for polarised components*

(b) *Make sure the component is securely in the holes, so that when the PCB is flipped over for soldering, the component does not fall out*

(c) *Solder the pads of the PCB to the component on the side opposite of the component*

(d) *Cut the excess wire from the component that is remaining after the solder*

4.3.3 Final Assembly and Functional Testing

Upon completion of the PCB assembly the next step was to *assemble the PCB with the Starger box*. The final step was to functionally test the power supply. The key functions tested and expected results are:

Power Supply Resistance Between Power and Ground Measured using a *multimeter*. A successful result is *a very high value, in the realm of 12-14 MOhms*

LED Indicator Tested by *connecting the Starger box to power*. A successful result is *the Indicator LED turning on.*

Voltage Output Measured using a *multimeter*. A successful result is *stable DC output around 5V.*

4.4 Discussion

In this section the challenges and successes and learnings the author experienced during the assembly process are presented.

4.4.1 Challenges

This subsection lists the challenges experienced during the assembly process of the Starger Project and how they were overcome.

i. *Ensuring that the measurements for drilling the holes were properly aligned proved more difficult than expected. After drilling, many holes were not properly aligned. To fix this, files and a deburring tool were used to augment the shape and size of the holes to accommodate the necessary components.*

ii. *The assembly of the PCB provided relatively little challenges, however, on the first attempt of using the Pick-and-Place machine, too much solder paste was applied to the pads. When the components were placed on the pads, the solder paste spread out and came into contact with one another, causing shorts. This was caught before putting the PCB in the oven, and the components were removed, the solder paste was cleaned from the PCB, and the process was restarted.*

4.4.2 Successes and Learnings

This subsection lists the highlights and learning the author experienced during the assembly process for the Starger Project.

i. *Proper use of the pick-and-place machine was learned through this project, and signs of proper solder paste application were understood.*

ii. *Drilling techniques were learned through this project, as well as proper safety while using drilling tools and all other tools used during this project.*

4.5 Chapter Summary

In this chapter, topics related to the assembly, fault-finding and testing processes for the Starger Project were presented. In particular the following were highlighted:

1. *Project Room components, hazards, and safety components*
2. *Starger Box preparation and assembly*
3. *Starger PCB assembly*
4. *Final assembly and functionality testing of the power supply, LED indicator, and voltage output*
5. *Challenges, successes, and learnings throughout the project*

Chapter 5

Starger Testing and Simulation

5.1 Introduction

This chapter describes the process undertaken to examine the characteristics of the power supply and tone generation for the "Starger" project. The behaviour of the regulated and unregulated outputs of the power supply under load are examined through measurement and simulation in Altium. The analogue clock source is compared with an Altium simulation and then with the digital output of the microcontroller to examine the characteristics of the tone generator.

5.2 Power Supply Background Theory

The $\pm 12VDC$ power supply is configured as a *full*-wave rectifier. Rectified power supplies, such as the *full*-wave rectifier in this design are simple to implement. However, they have limited ability to maintain a constant output voltage and are referred to as "unregulated" power supplies.

The effect described in Eq. 5.1 explains why the output *voltage* of unregulated power supplies changes along with the change in load *current*, as an increase in load *current* also *increases* the *peak-to-peak voltage* resulting in *a reduction* in average output *voltage*.

$$V_{ripple} = \frac{I_{load}}{F_{ripple} \cdot C} \tag{5.1}$$

The $5V\,DC$ output is provided by a *Linear Voltage Regulator* (LVR), which provides a "regulated" output that should not experience the same variance in output *voltage* as "unregulated" power supplies.

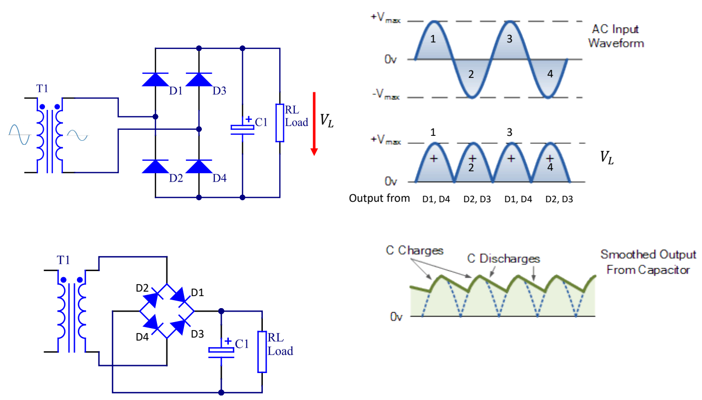

Figure 5.1: AC and unregulated DC waveforms with smoothing capacitor

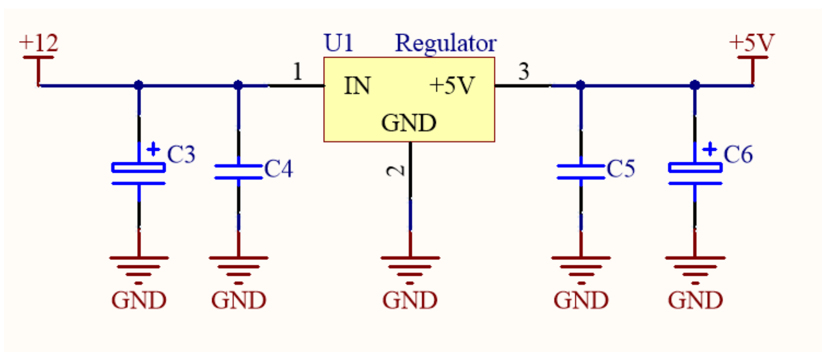

Figure 5.2: Regulated supply circuit

5.3 Power Supply Lab Testing Process

In this section, the process to test the behaviour of the tone generator power supply using laboratory test equipment is explained.

5.3.1 Equipment

The following equipment was used in the load testing for the tone generator's power supply.

- *Oscilloscope*
- *Multimeter*
- *Variable Electronic DC Load*

5.3.2 Method Overview

To verify the behaviour of the regulated and unregulated power supply outputs a testing circuit connected according to Fig 5.3 is used. A variable load resistor is connected in series with the supply and the voltage waveforms displayed on the oscilloscope connected in parallel. The oscilloscope is also used to measure and record the RMS voltage and peak-to-peak voltage for the $+12VDC$ supply. The measurements start at no load to $100mA$ in $10mA$ steps. The changes to the voltage level are observed and explained, with the results presented in table and graph formats. The process is repeated for the $+5VDC$ measuring from no load to $200mA$ in $20mA$ steps. The differences in the results obtained for the $+12V$ and $+5V$ are contrasted and explained in the context of DC ripple with methods to reduce it.

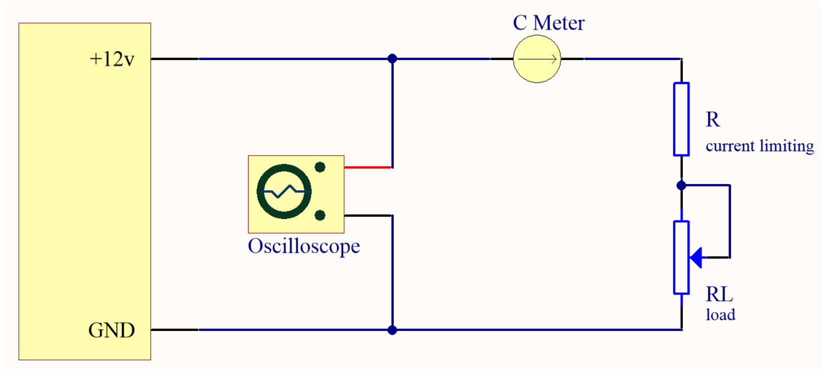

Figure 5.3: Power supply load testing circuit

5.3.3 Detailed Method

The step-by-step process used to set up and measure the waveforms is described below

Step 1 *Collect the necessary equipment to carry out the experimentation. The equipment used for this experiment includes the Starger, an oscilloscope, a multimeter, and a variable electronic DC load.*

Step 2 *Using the Starger and the proper equipment, construct the test circuit shown in Figure 5.3.*

Step 3 *Calibrate the oscilloscope and set it to measure the necessary values. Set the attenuation to 1x, as the voltage ripple is small, and connect the oscilloscope probes to terminal TM1 on the Starger, which is the 12V unregulated test point, and terminal TM3 on the Starger, which is the ground test point.*

Step 4 *The oscilloscope is then used to measure the voltage ripple and RMS output voltage of the 12V unregulated power supply as the load current was increased from 0mA to 100mA in steps of 10mA. The multimeter is used to measure the current through the circuit, which is used to set the variable electronic DC load* *properly for the desired current. Pictures of the measurements and waveforms are taken, and these picture and result are reported.*

Step 5 *To test the 5V regulated power supply, move the oscilloscope probe from TM1 to TM2, which is the test point for the 5V regulated power supply, while keeping the ground probe of the oscilloscope on TM3. Measuring the same values with the oscilloscope and still measuring the current with the multimeter to properly set the variable electronic DC load, the load current is increased from 0mA to 200mA in steps of 20mA. Pictures of the measurements and waveforms are taken, and these pictures and results are reported.*

5.3.4 Results - Raw

The follow includes the raw results captured from the screen of the oscilloscope

Table 5.1: 12V Unregulated Power Supply

Current (mA)	**Output Voltage (RMS) (V)**	**Output Voltage Ripple (Peak to Peak) (mV)**
0	12.5	240
10	12.4	360
20	12.3	360
40	12.2	360
50	12.1	480
60	12.0	360
70	12.0	480
80	11.9	480
90	11.8	480
100	11.8	480
200	11.2	600

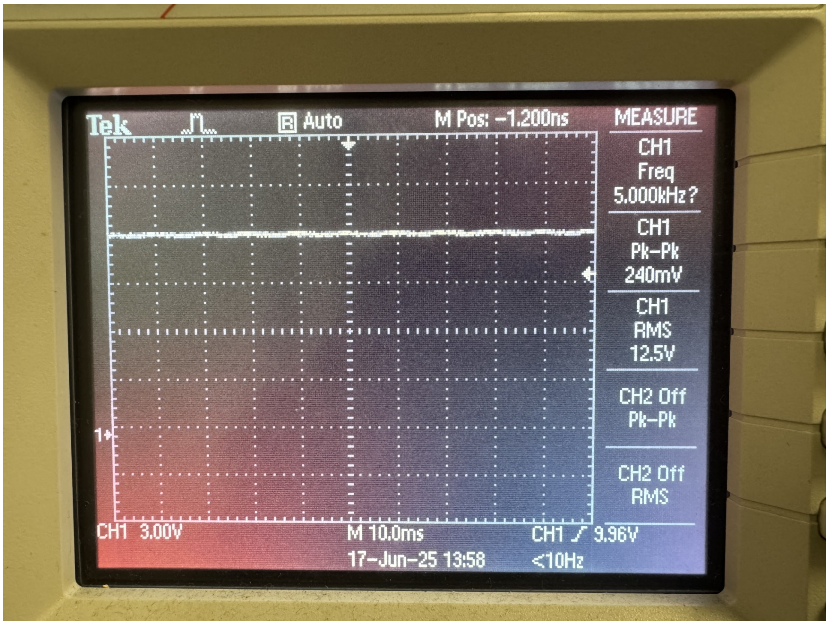

Figure 5.4: Power supply load testing result $12V$ at 0mA

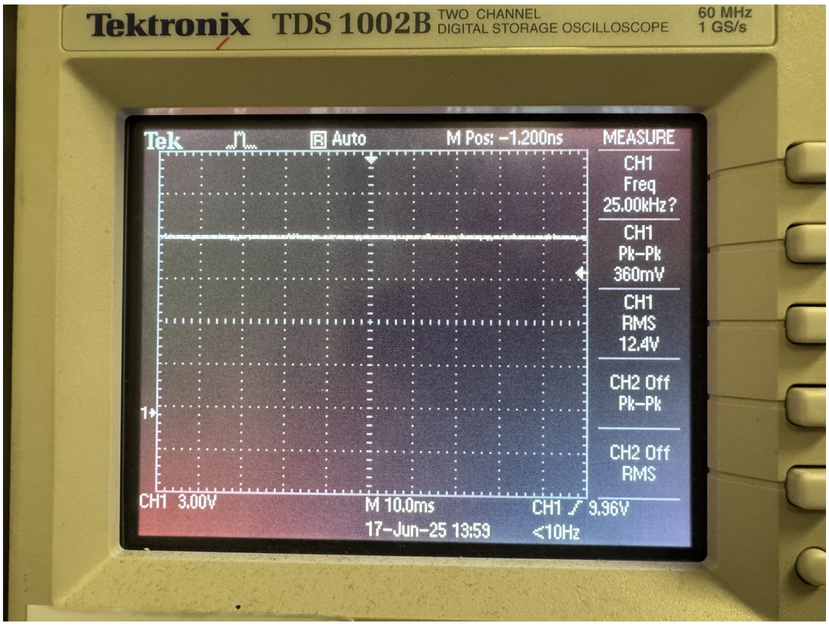

Figure 5.5: Power supply load testing result $12V$ at 10mA

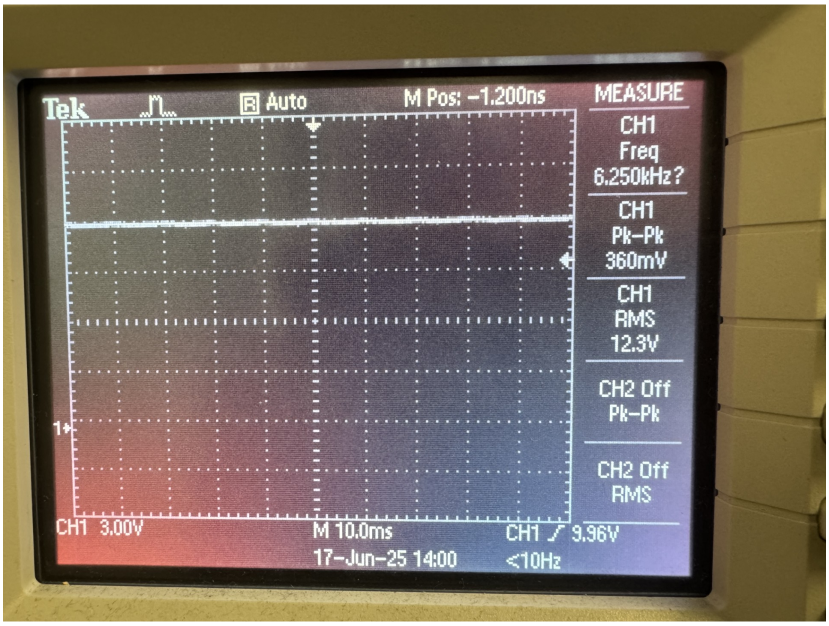

Figure 5.6: Power supply load testing result $12V$ at 20mA

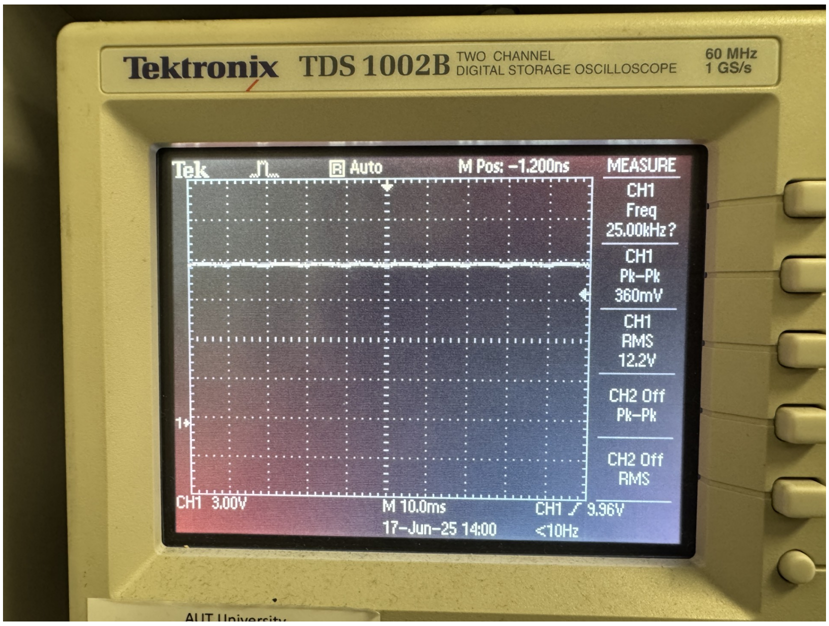

Figure 5.7: Power supply load testing result $12V$ at 40mA

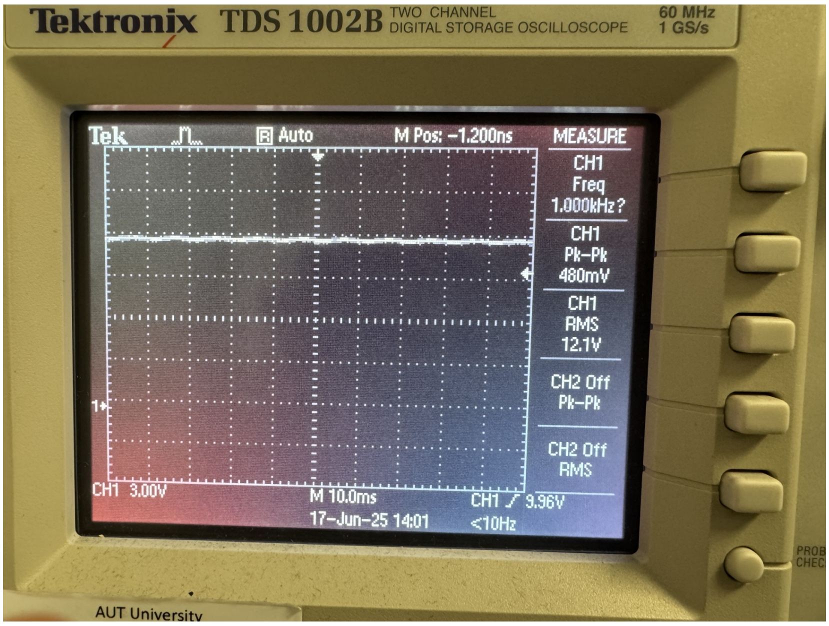

Figure 5.8: Power supply load testing result $12V$ at 50mA

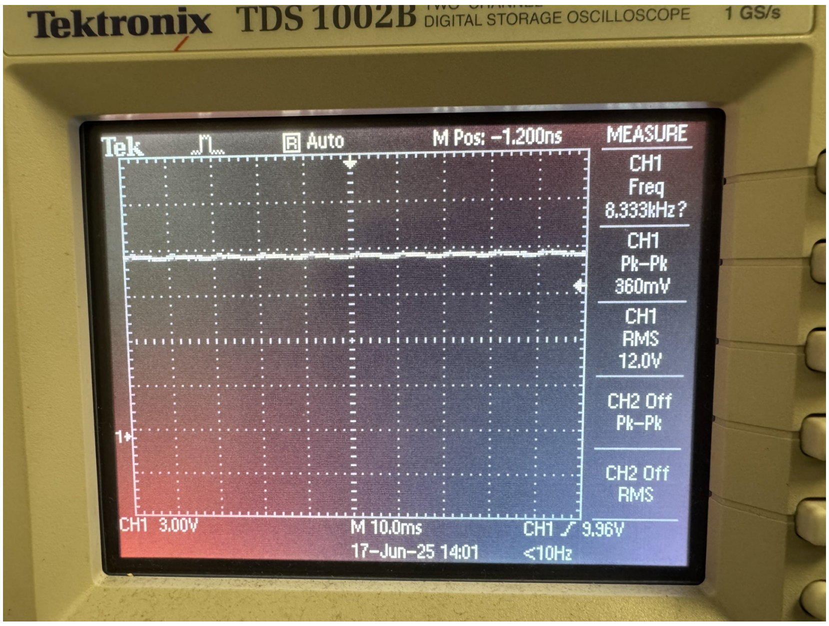

Figure 5.9: Power supply load testing result $12V$ at 60mA

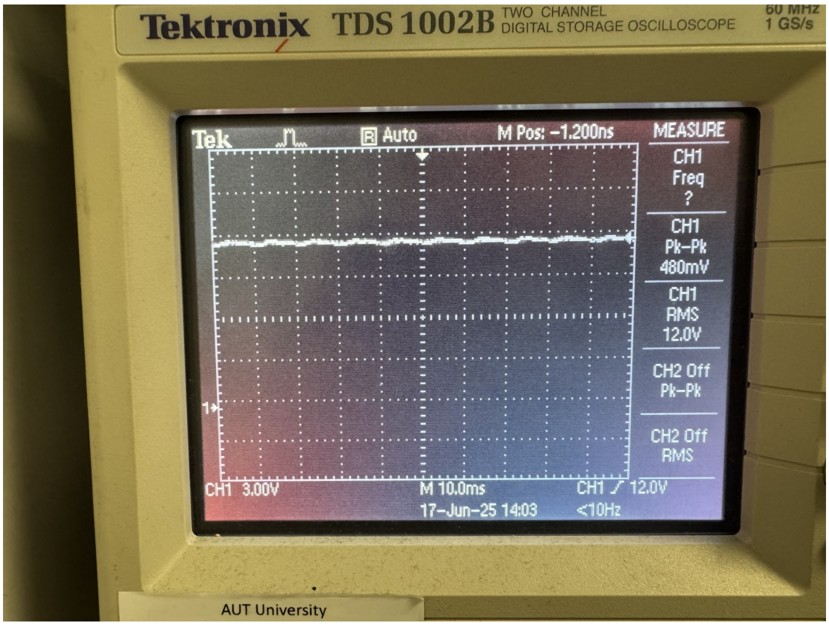

Figure 5.10: Power supply load testing result $12V$ at 70mA

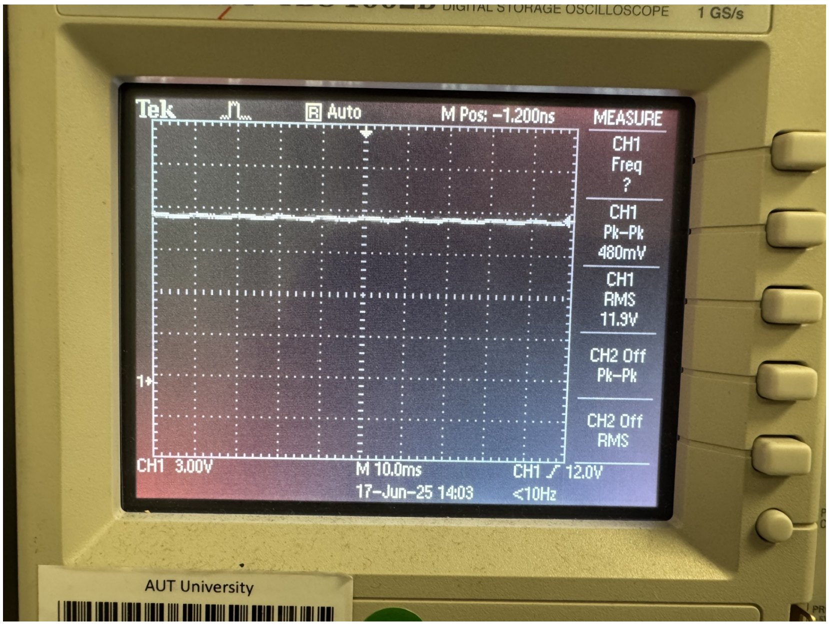

Figure 5.11: Power supply load testing result $12V$ at 80mA

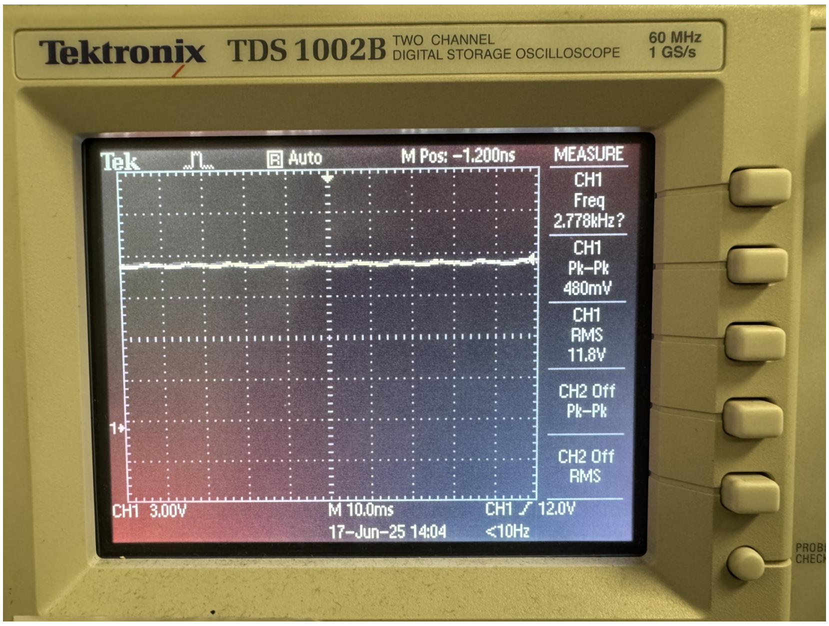

Figure 5.12: Power supply load testing result $12V$ at 90mA

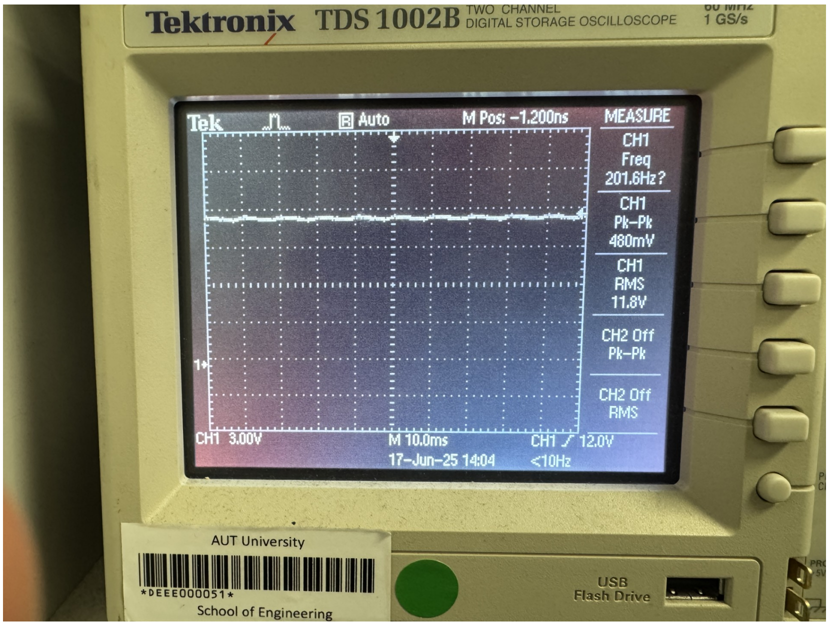

Figure 5.13: Power supply load testing result $12V$ at 100mA

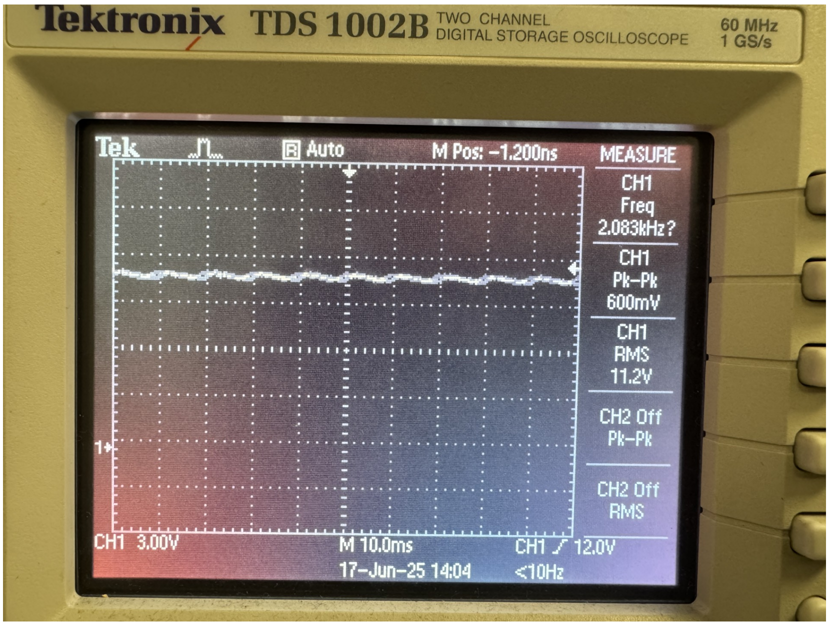

Figure 5.14: Power supply load testing result $12V$ at 200mA

Table 5.2: 5V Regulated Power Supply

Current (mA)	Output Voltage (RMS) (V)	Output Voltage Ripple (Peak to Peak) (mV)
0	4.86	89.6
20	4.78	112
40	4.73	112
60	4.69	134
80	4.66	157
100	4.61	134
120	4.58	134
140	4.56	134
160	4.54	134
180	4.52	134
200	4.49	134

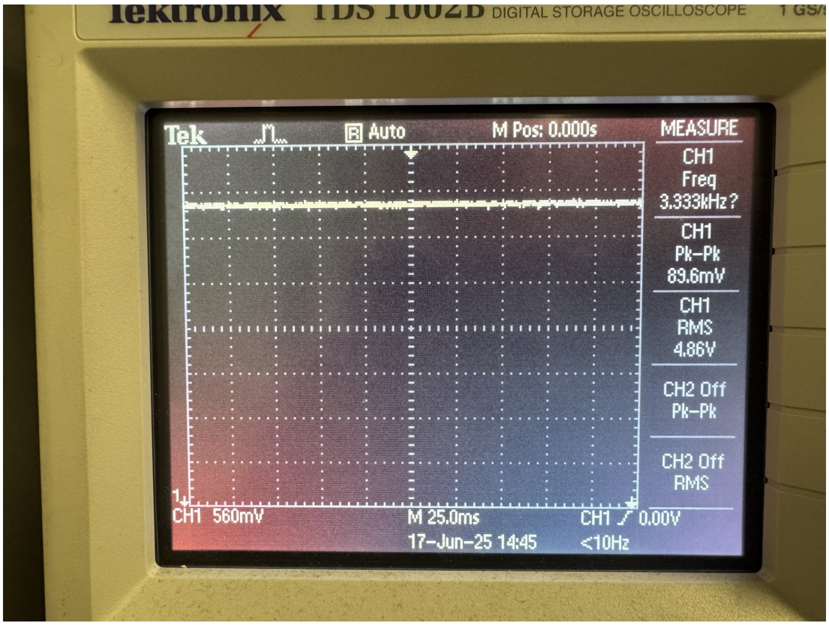

Figure 5.15: Power supply load testing result $5V$ at 0mA

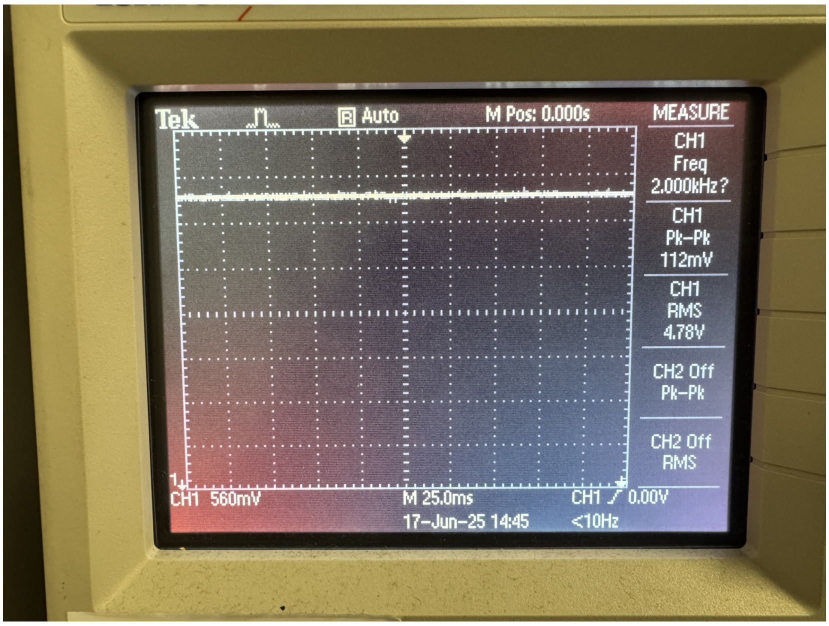

Figure 5.16: Power supply load testing result $5V$ at 20mA

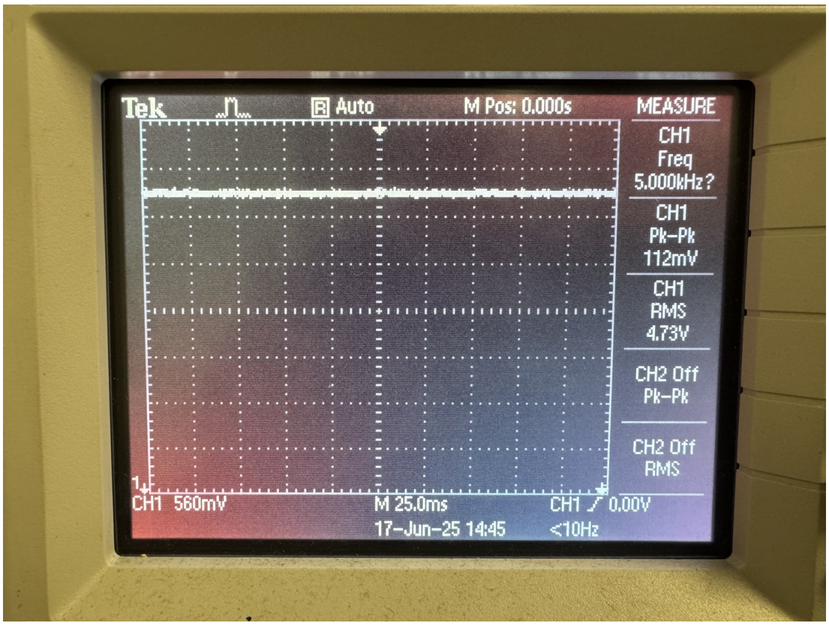

Figure 5.17: Power supply load testing result $5V$ at 40mA

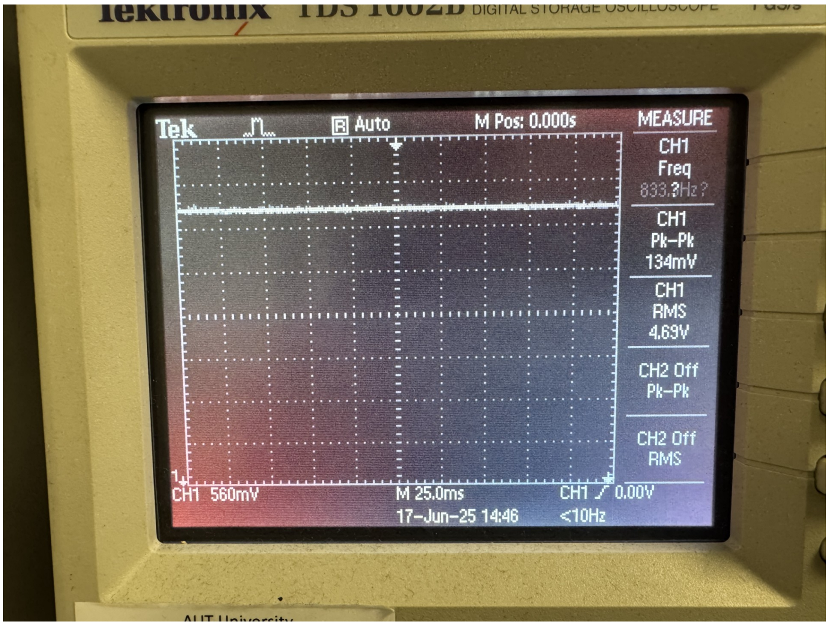

Figure 5.18: Power supply load testing result $5V$ at 60mA

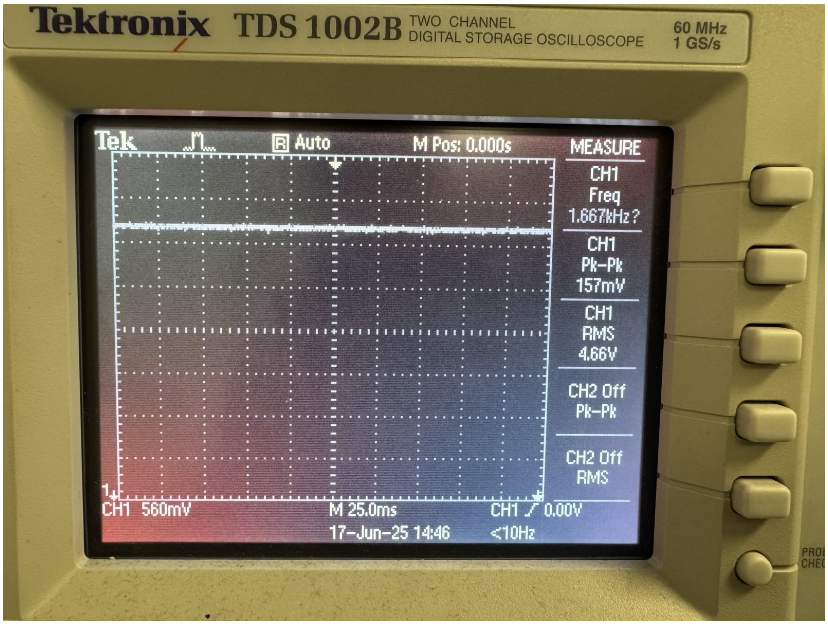

Figure 5.19: Power supply load testing result $5V$ at 80mA

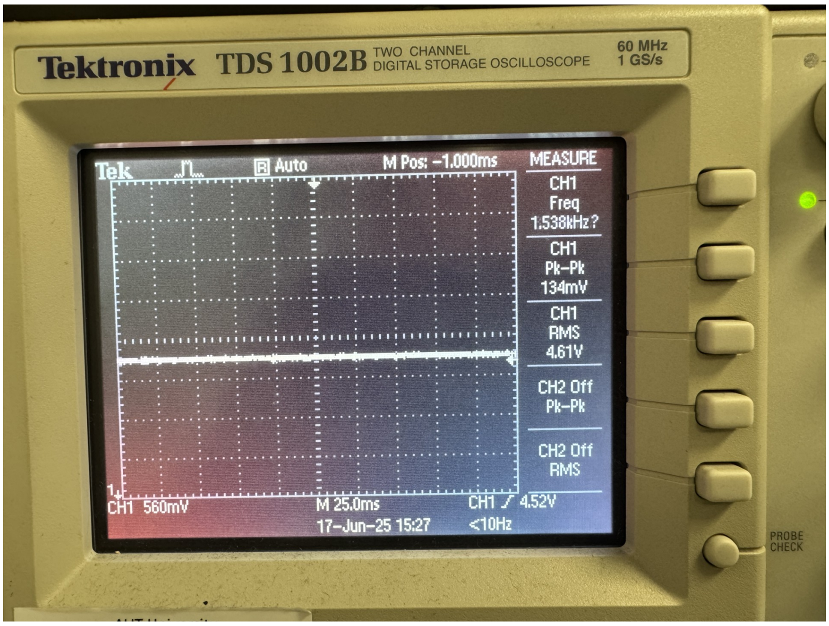

Figure 5.20: Power supply load testing result $5V$ at 100mA

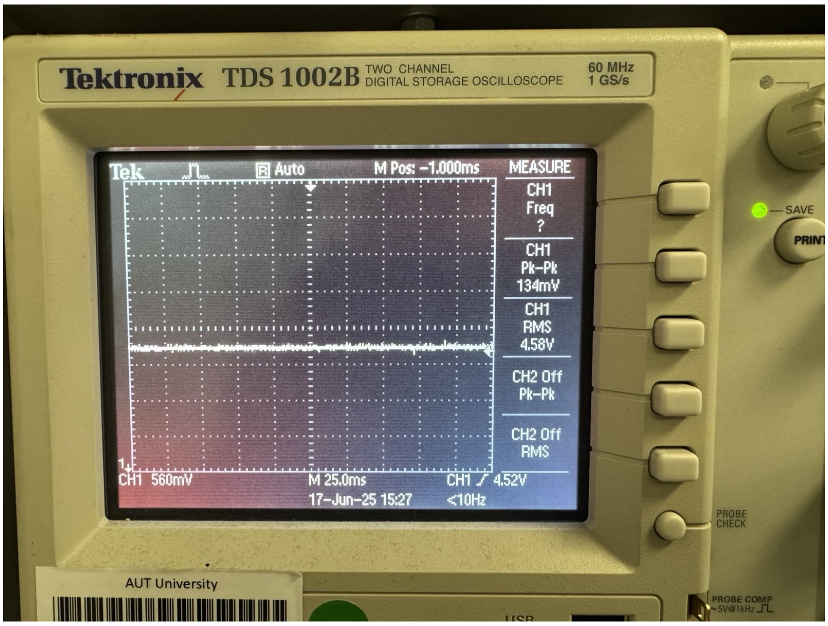

Figure 5.21: Power supply load testing result $5V$ at 120mA

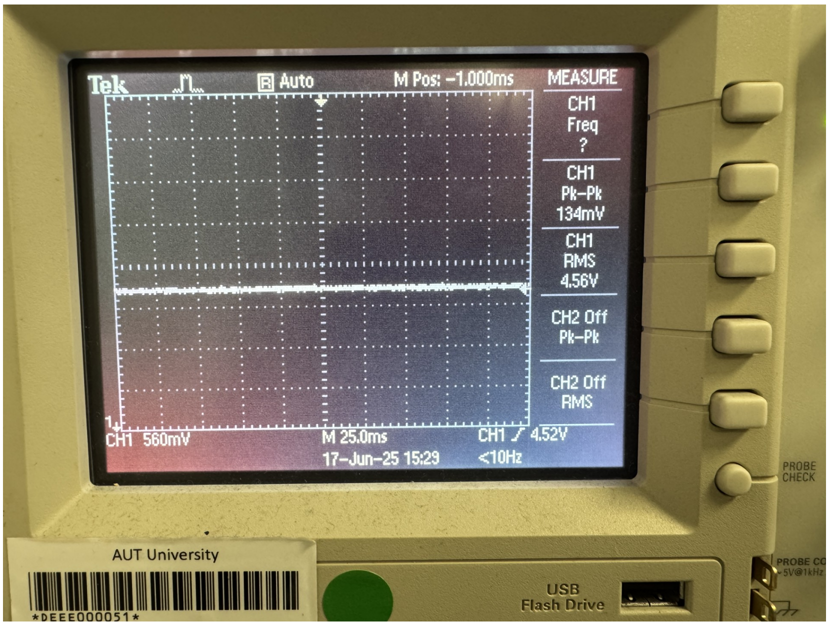

Figure 5.22: Power supply load testing result $5V$ at 140mA

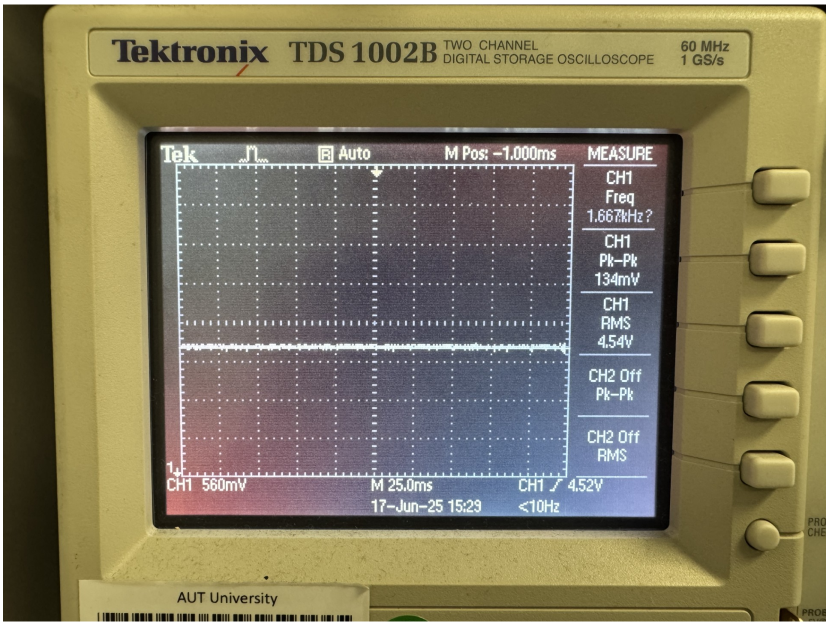

Figure 5.23: Power supply load testing result $5V$ at 160mA

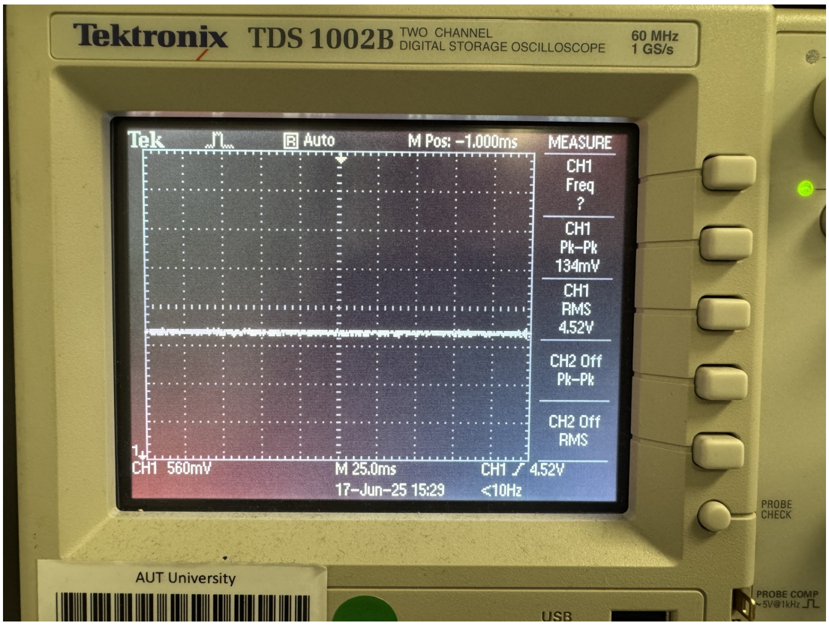

Figure 5.24: Power supply load testing result $5V$ at 180mA

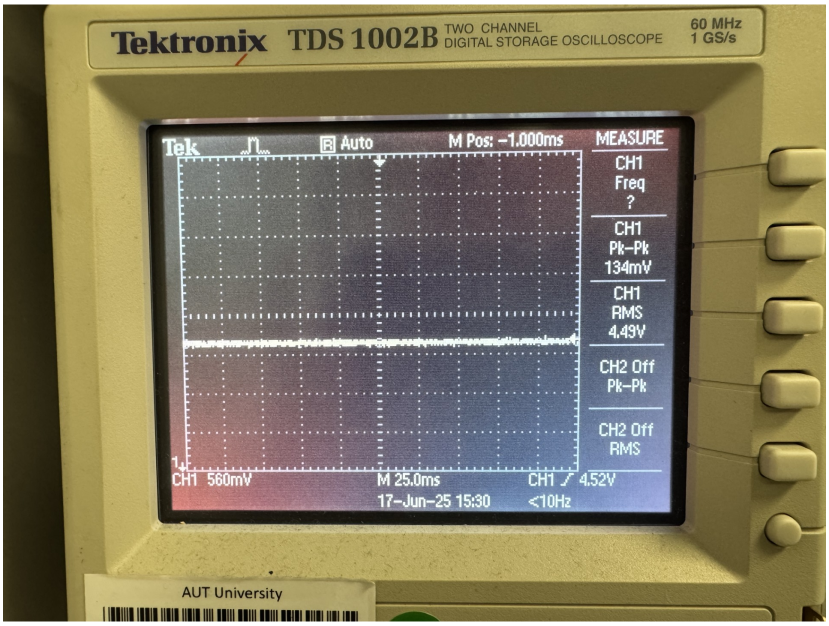

Figure 5.25: Power supply load testing result $5V$ at 200mA

5.3.5 Discussion

Comparing results in Tables 5.1 and 5.2, as well as Figures 5.4-5.25, comparisons and observations can be made between the 12V unregulated power supply and the 5V regulated power supply. For both power supplies, as the load current increases, the output voltage decreases. This is consistent with equation 5.1. Similarly, as the load current increases, the output voltage ripple generally increases, though this is less apparent due to difficulties measuring the peak-to-peak voltage. An important difference between the 12V supply and the 5V supply is that the output voltage ripple of the 5V supply is much less significant than that of the 12V supply. This is due to the regulation in the 5V supply, whereas the 12V supply is unregulated and varies more with the changing load current.

5.4 Power Supply Simulation

In this section, the process to simulate the behaviour of Power supply using Altium Designer is explained.

5.4.1 Simulation Schematic

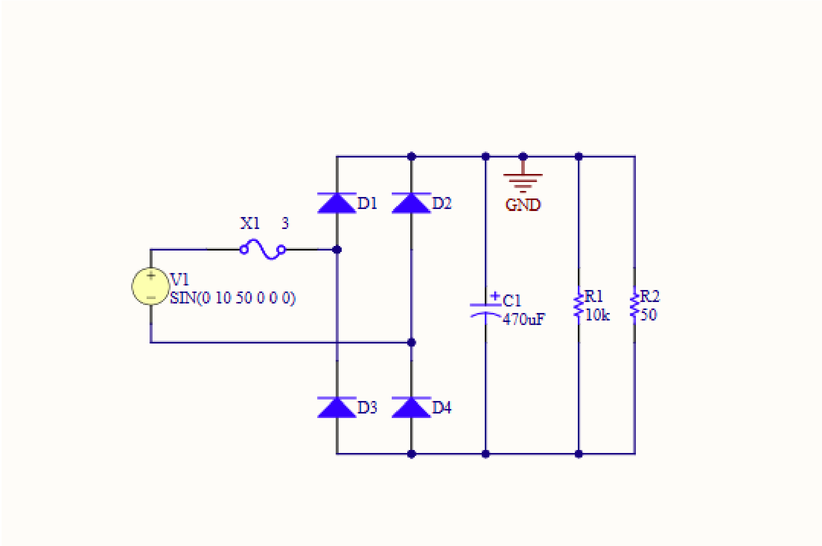

Figure 5.26: Unregulated Power Supply Simulation Schematic

5.4.2 Simulation Results

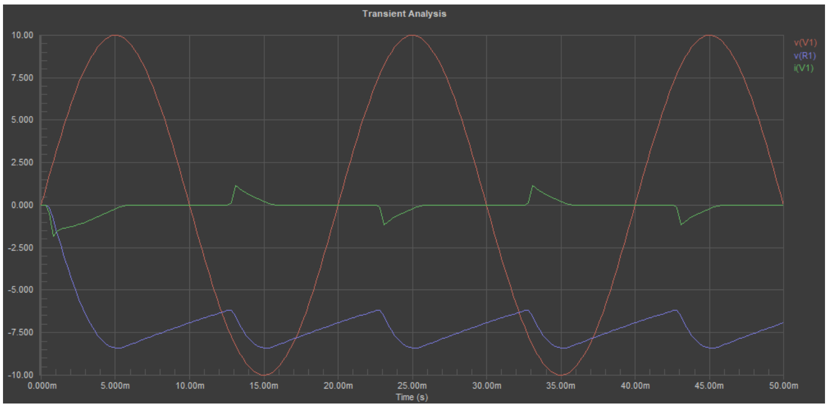

Figure 5.27: Unregulated Power Supply Simulation Output with 50 Ohm Load

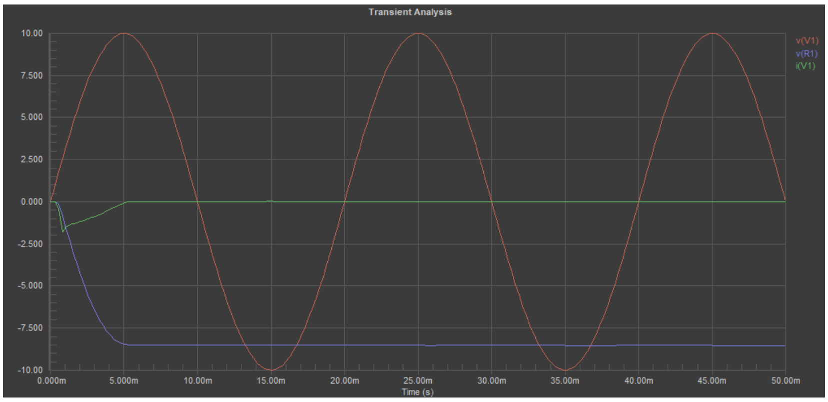

Figure 5.28: Unregulated Power Supply Simulation Output with 50K Ohm Load

5.4.3 Discussion

Comparing the results of simulation to that of the experimental results, it can be observed that the output ripple is similar, as expected. At a low load resistance, which correlates to a high load current for the experimental results, the output voltage ripple is relatively high. When the load resistance is increased, correlating to the lower load current in the experimental results, the output voltage ripple is much less significant.

5.5 Tone Generator Background Theory

In this section how the PWM waveform is generated by the NE555 is described, along with the maximum and minimum potentiometer (R16) values to produce the frequency range $180Hz$, to $800Hz$.

For generation of the digital clock signal for the LED chasing circuit button debouncing is explained along with and how it can be prevented in code and how the internal pull-up resistor is enabled for sensing button S1. Finally, depending on the coding choice of the author, either:

- *The timer is configured in CTC mode to generate square waves for the buzzer. It* *toggles the OC1A pin (PB3) on each compare match, producing a tone. The compare value, OCR1A, is calculated based on the target frequency and a prescaler. This toggling creates a 50 percent duty cycle square wave that drives the buzzer at the correct tone.*

5.6 Tone Generator Lab Testing Process

In this section, the process to measure the accuracy of the tone generator's frequencies using laboratory test equipment is explained. Such information includes how the resistance and PWM waveform is measured using the potentiometer values calculated in Section 5.5 to create the $523 - 1047Hz$ in the circuit. These measured values are compared to the calculated values for both analogue and digital clock sources with the results presented in tables.

5.6.1 Equipment Used

The following equipment was used in the frequency testing for the tone generator.

- *Oscilloscope*
- *Multimeter*

5.6.2 Detailed Method

The step-by-step process used to set up and measure the clock frequencies is described below

Step 1 *First, gather the necessary equipment. This includes the Starger project, an oscilloscope, an oscilloscope probe, a multimeter, and multimeter probes.*

Step 2 *To first test the digital tone generator, a code was created to make the buzzer on the starger step through the desired frequencies of 523, 587, 659, 698, 784, 880, 988, and 1047 Hz.*

Step 3 *Attach the oscilloscope probe to terminals TP1 to measure the buzzer waveform, and TM4 to ground the probe.*

Step 4 *Using the oscilloscope, measure the output frequency of each signal through the buzzer and compare these measured values with the set values described in Step 2. The measured results are reported.*

Step 5 *To test the analogue tone generator, keep the same measurement set up with the oscilloscope, and use the multimeter to measure the resistance of the potentiometer.*

Step 6 *Setting the potentiometer resistance to specified values, in this case 4k, 8k, and 11k Ohms were used, and measure the frequency of the waveform at the buzzer with those potentiometer resistance values. The measured results are reported.*

5.6.3 Results-Raw

Table 5.3: Digital Tone Generation: Set vs Measured Frequencies

Set Frequency (Hz)	Measured Frequency (Hz)
523	517.7
587	585.1
659	654.4
698	696.1
784	776.4
880	877.2
988	984.3
1047	1043

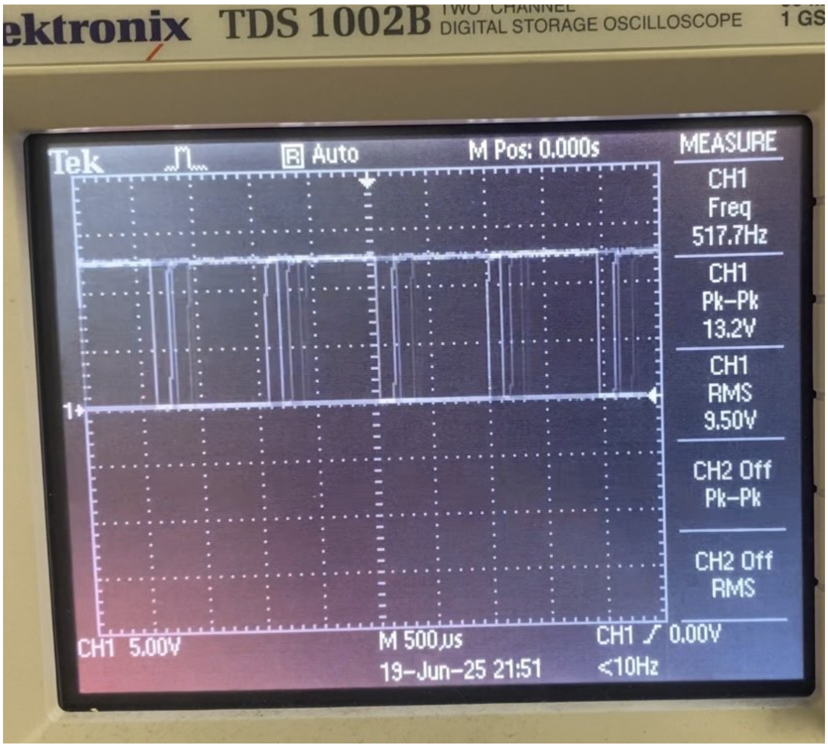

Figure 5.29: Expected frequency $523Hz$ with measured frequency $517.7Hz$

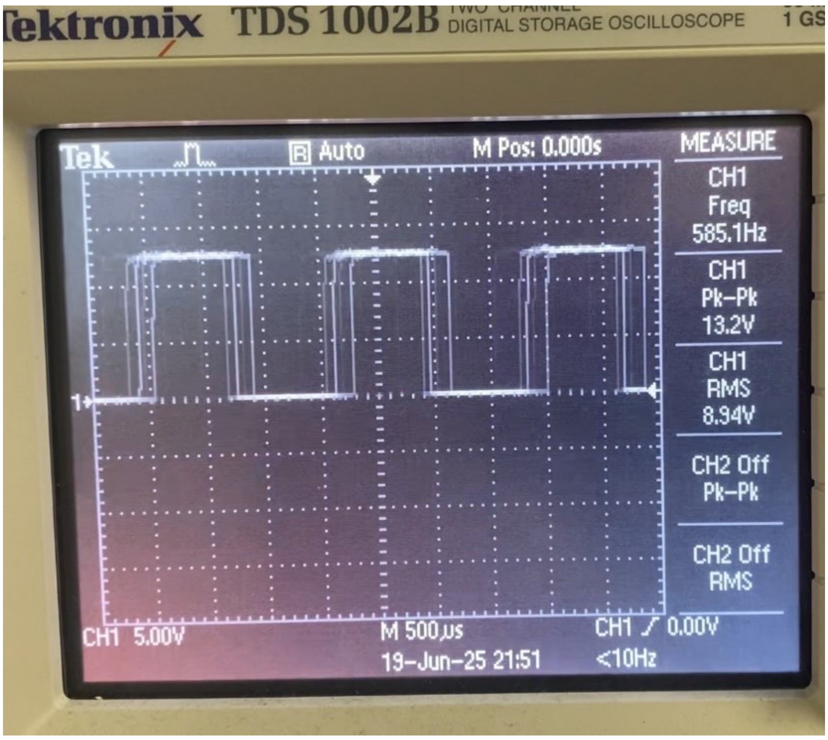

Figure 5.30: Expected frequency $587Hz$ with measured frequency $585.1Hz$

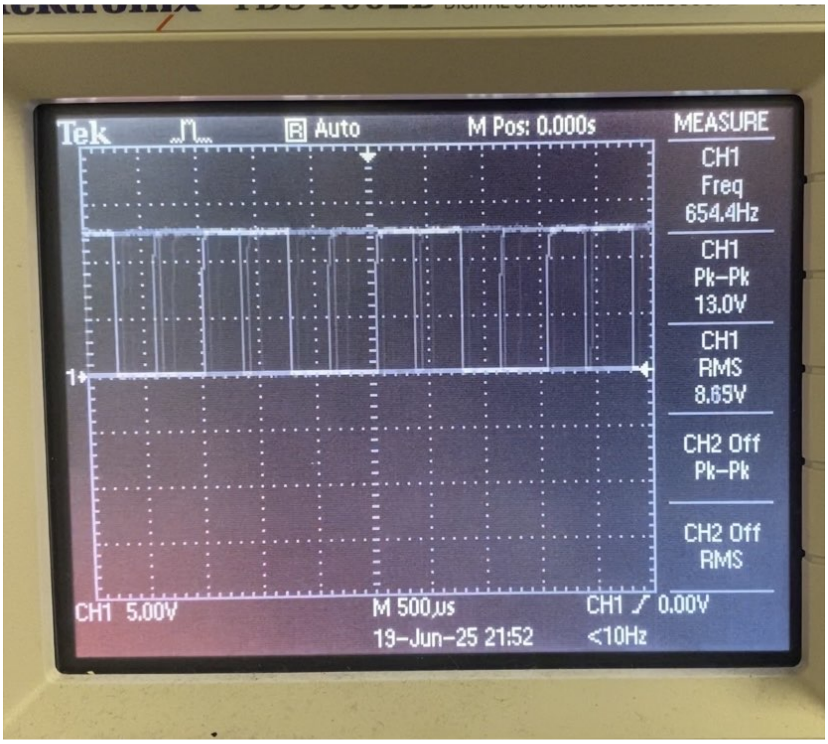

Figure 5.31: Expected frequency $659Hz$ with measured frequency $654.4Hz$

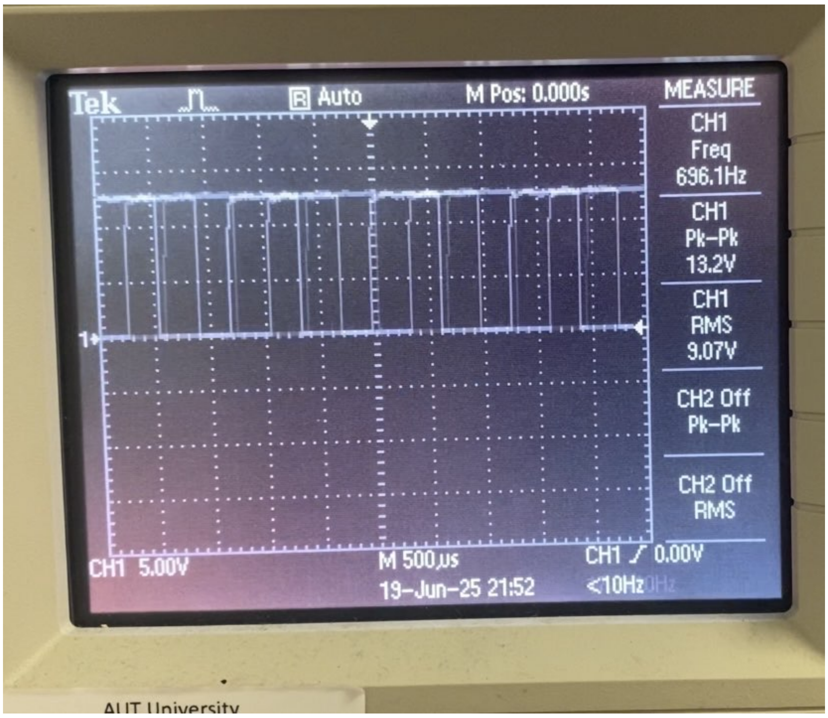

Figure 5.32: Expected frequency $698Hz$ with measured frequency $696.1Hz$

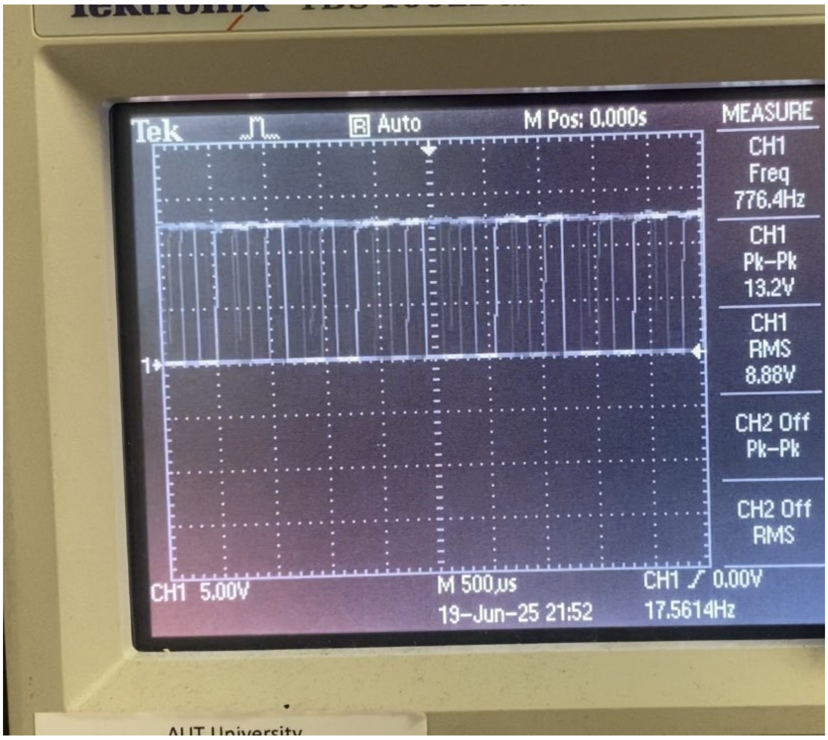

Figure 5.33: Expected frequency $784Hz$ with measured frequency $776.4Hz$

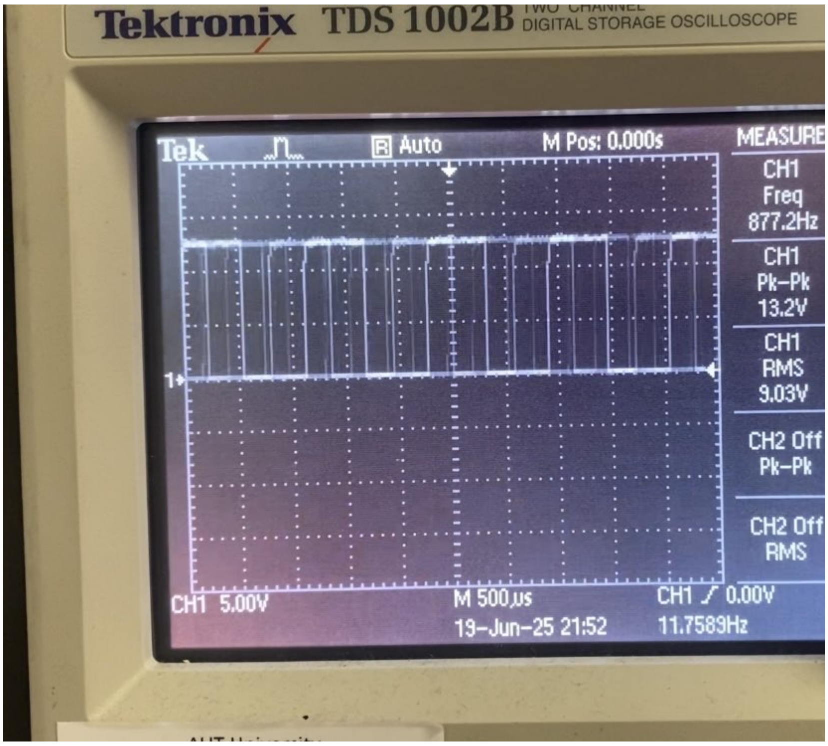

Figure 5.34: Expected frequency $880Hz$ with measured frequency $877.2Hz$

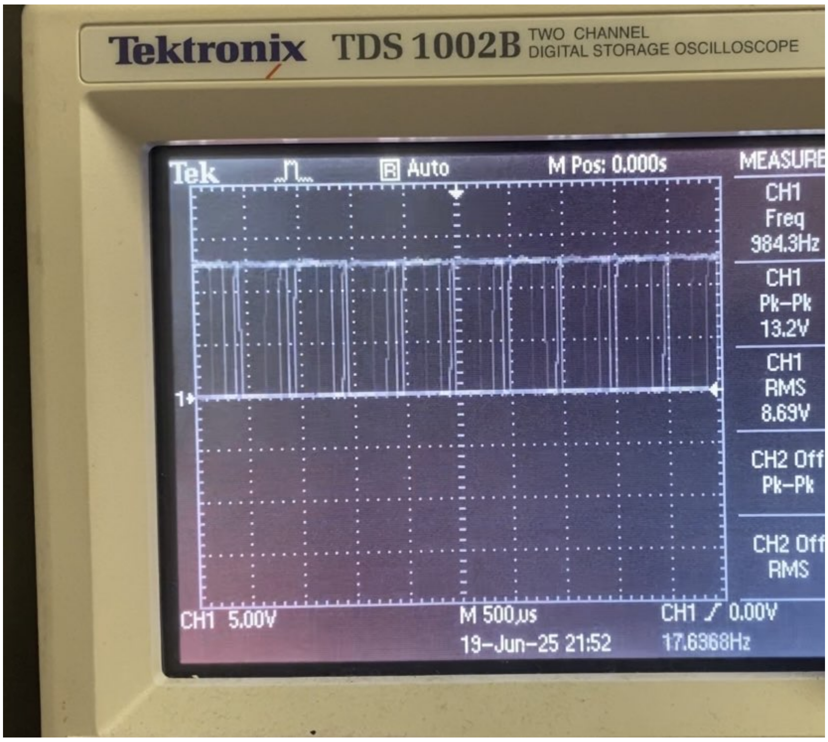

Figure 5.35: Expected frequency $988Hz$ with measured frequency $984.3Hz$

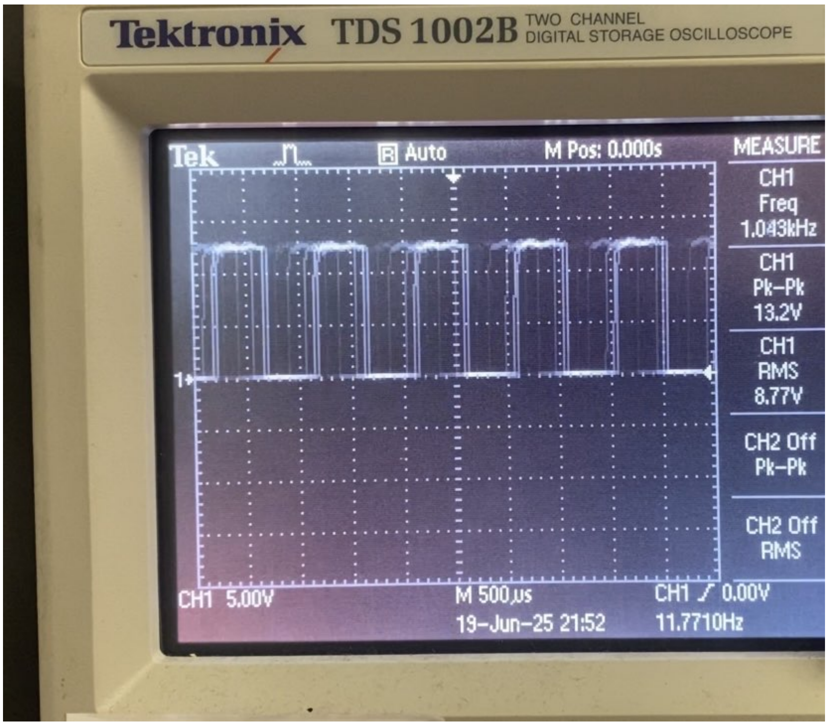

Figure 5.36: Expected frequency $1047Hz$ with measured frequency $1043Hz$

Table 5.4: Analogue Tone Generation: Resistance vs Measured Frequency

Set Potentiometer Resistance (Ohms)	Measured Frequency (Hz)
4000	556.2
8000	396.8
11000	181.2

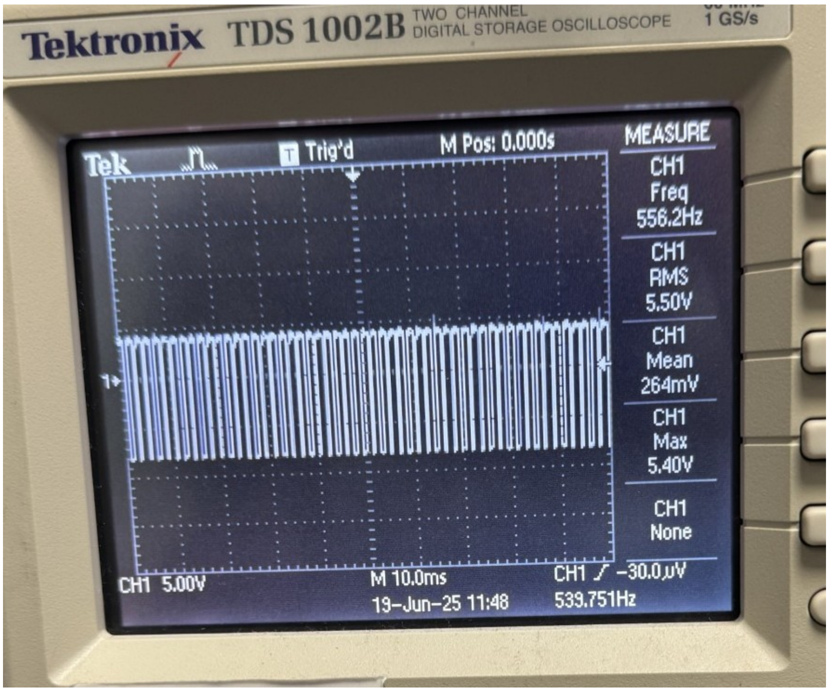

Figure 5.37: Analogue Frequency of $556.2Hz$ with measured resistance of $4kOhms$

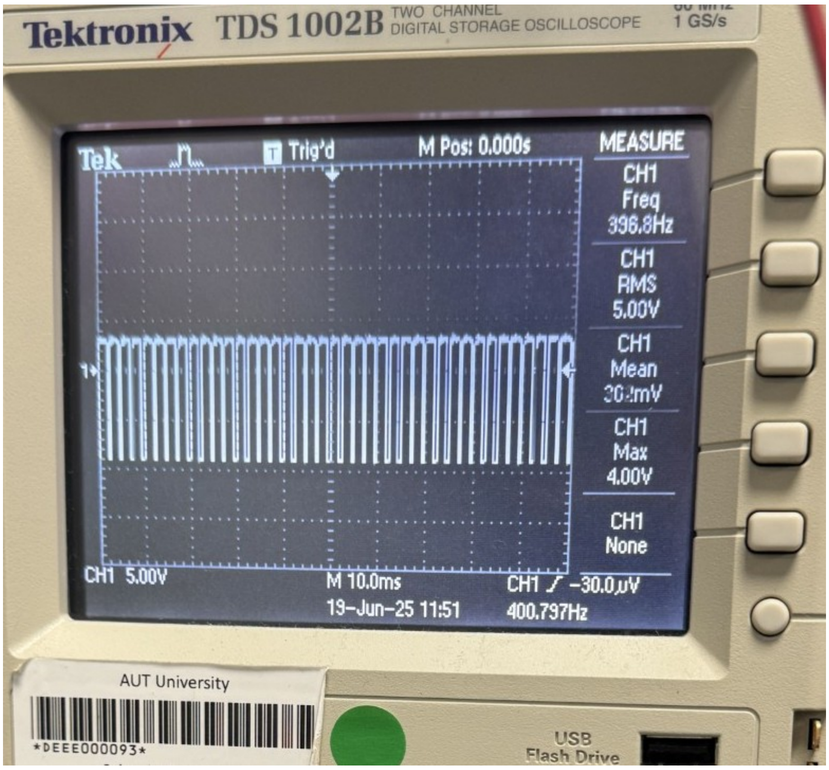

Figure 5.38: Analogue Frequency of $556.2Hz$ with measured resistance of $8kOhms$

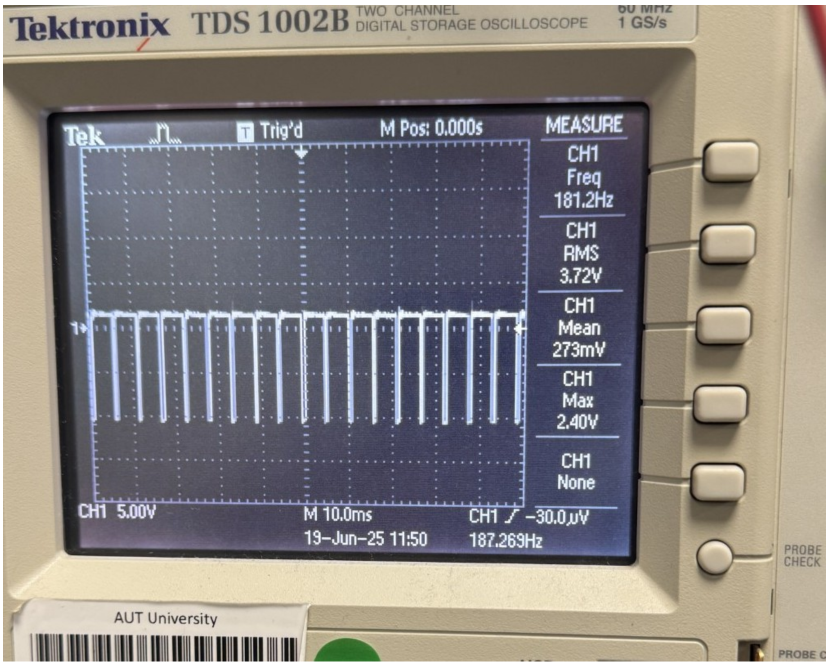

Figure 5.39: Analogue Frequency of $556.2Hz$ with measured resistance of $11kOhms$

5.6.4 Discussion

Observing the results of the digital tone generation experimentation,it can be seen that the code used, provided in Appendix A, accurately sets the frequency of the buzzer. All of the measured frequencies are within a reasonable level of error from the set frequency. Observing the results of the analogue tone generation experimentation, it can be seen that as the resistance of the potentiometer increases, the measured frequency of the buzzer decreases. This can be compared with the simulation results in section 5.7.

5.7 Simulation

In this section, the process to simulate the behaviour of the 555 Timer for the Tone Generator using Altium Designer is explained.

5.7.1 Simulation Schematic

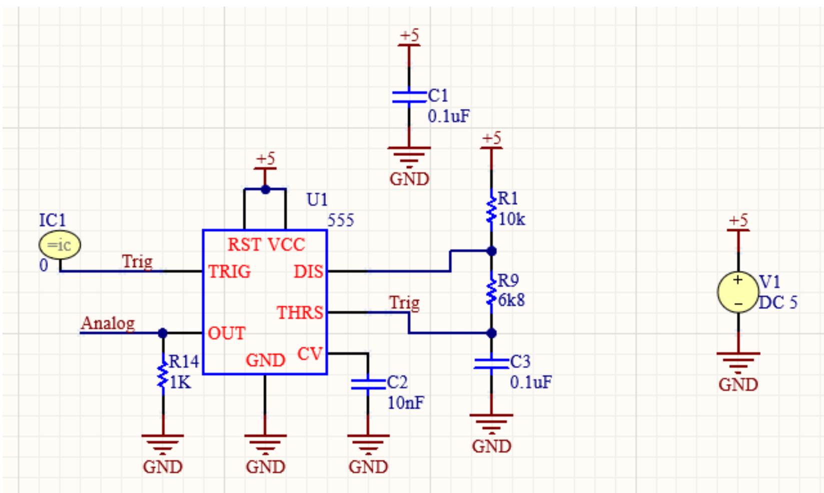

Figure 5.40: 555 Timer Simulation Schematic

5.7.2 Simulation Results

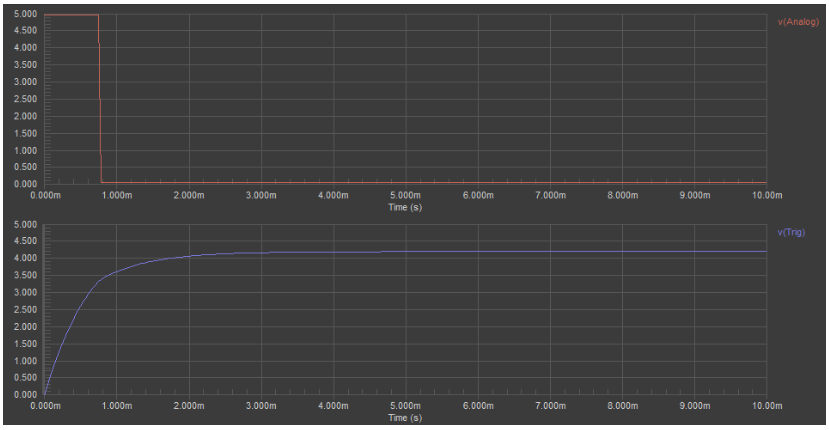

Figure 5.41: 555 Timer Simulation Output Diagram (R1 = 1 Ohms)

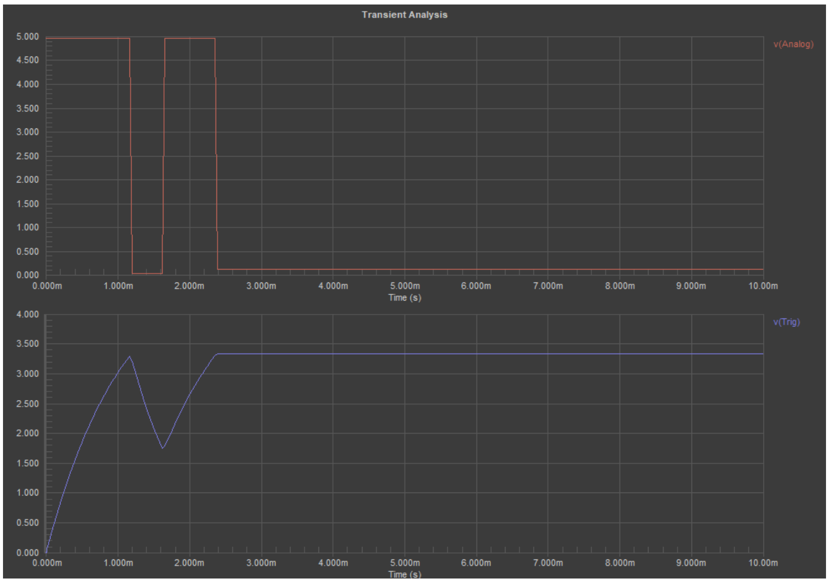

Figure 5.42: 555 Timer Simulation Output Diagram (R1 = 4k Ohms)

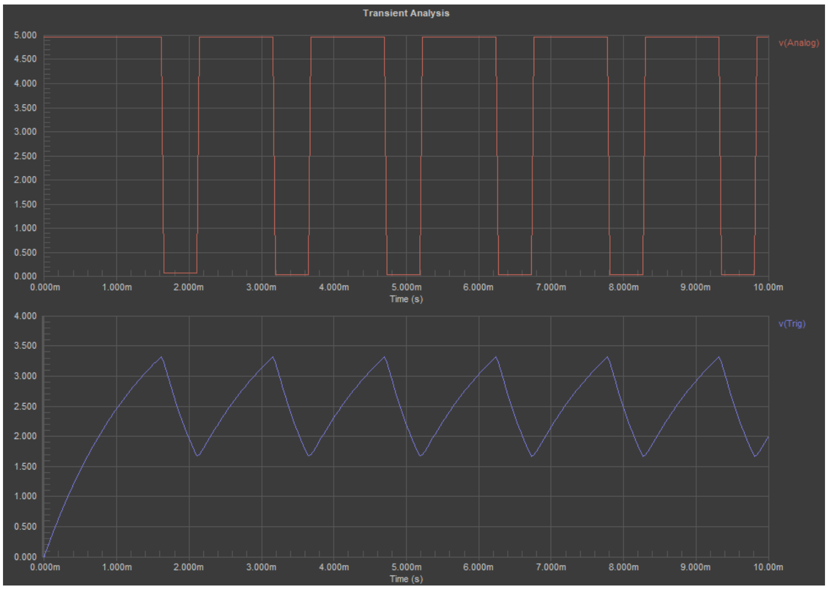

Figure 5.43: 555 Timer Simulation Output Diagram (R1 = 8k Ohms)

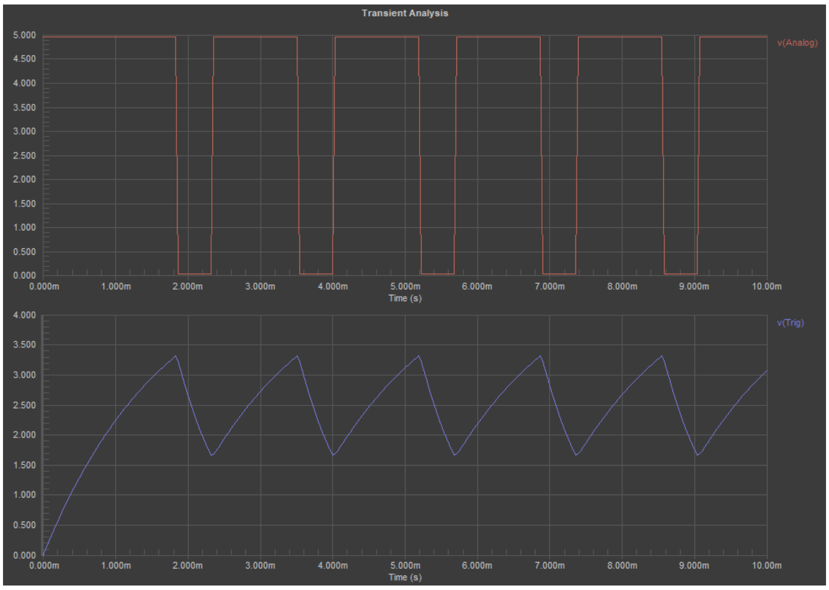

Figure 5.44: 555 Timer Simulation Output Diagram (R1 = 10k Ohms)

5.7.3 Discussion

Observing the results of the simulations, comparisons can be drawn to the experimentation results. In both the simulation and experimentation, as the resistance of the potentiometer increased, the frequency of the buzzer signal decreased. Also observable *is that at potentiometer resistances that are too low, the circuit is not able to successfully produce a square wave signal to the buzzer. This was observed during experimentation as well, which is why the results begin at a potentiometer value of 4k Ohms.*

5.8 Chapter Summary

In this chapter, topics related to the lab testing processes and simulations for the regulated and unregulated power supply and tone generation functions of the Analogue and Digital Tone Generator “Starger” Project were presented. In particular the following were highlighted:

1. *The background theory of the power supply used in the Starger project*
2. *The equipment and methods used to test the power supply*
3. *The simulation results of the power supply*
4. *The background theory of the tone generator*
5. *The equipment and methods used to test the tone generator*
6. *The simulation results of the tone generator*

Chapter 6

Robot Design Process, Testing and Microcontroller Code

6.1 Introduction

This chapter overviews the design and final assembly of the line-following robot, by including the schematic and PCB drawn by the author and pictures of the final assembled robot. The microcontroller code for the robot can be found in Appendix B.

6.2 Robot Schematic

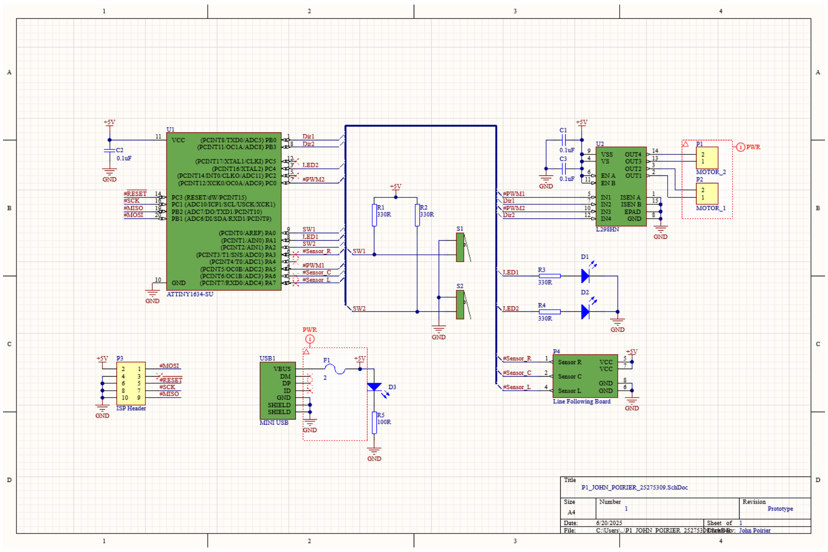

Figure 6.1: Robot Schematic drawn by John Poirier

6.3 Robot PCB

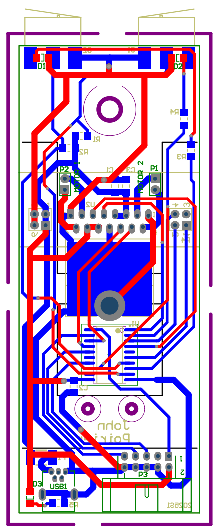

Figure 6.2: Robot PCB drawn by John Poirier

6.3.1 Design Process

The design of the PCB for the line following robot followed the schematic provided in lab. The PCB dimensions were provided in lab, as well as the placement for a select few specific components. From this, the remainder of the components were strategically placed, and the connections necessary were made. The number of vias on the PCB was below the imposed limit, and the PCB was successfully designed.

6.4 Robot Assembly

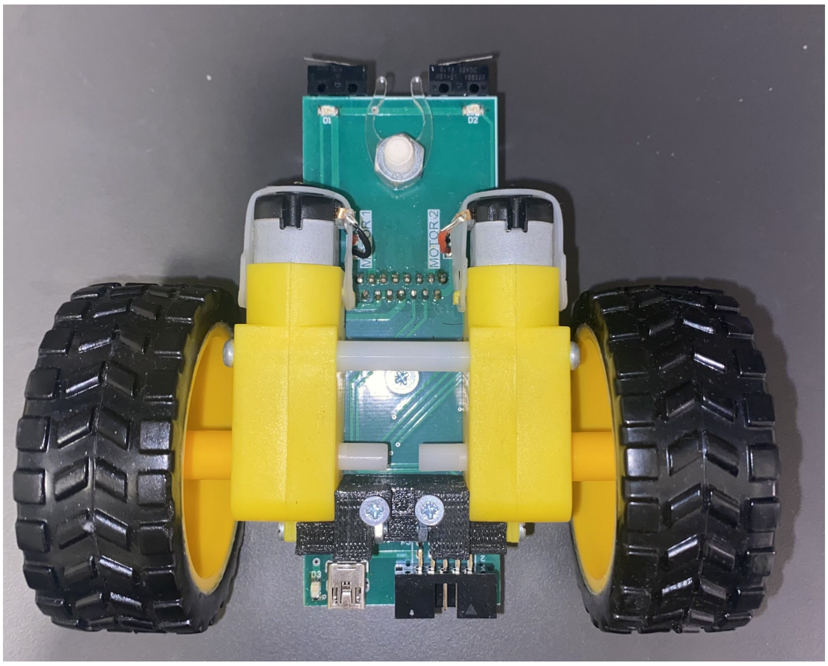

Figure 6.3: Robot assembled by John Poirier

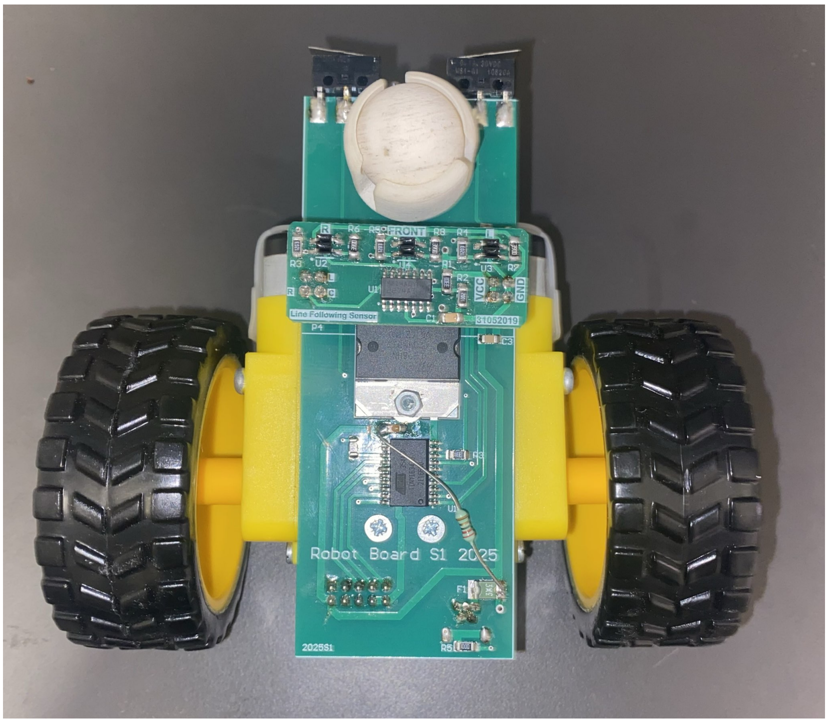

Figure 6.4: Robot assembled by John Poirier

6.4.1 Assembly Process

First, the PCB was assembled. This included using the pick-and-place machine to solder on the majority of the SMD components, and then soldering the remaining components by hand. Once the PCB was assembled, including both the primary PCB and the sensor module PCB, the motors were connected to the PCB. With the motors connected properly, the 3D printed spacer and body of the robot was attached to the PCB along with the motors, the 3D printed front wheel was screwed onto the PCB, and the wheels were attached to the motors. A battery pack holder was attached to the 3D printed body and spacer, used to power the robot.

6.5 Robot Code

Finally, code was developed to automate the line-following robot. To begin, code was developed to test the functionality of the motors. Through trial and error, the specific tuning of the motors was found to make the robot drive in a straight line. With this successful, code was developed to test the functionality of the sensors and the bumpers. Once the sensors and bumpers were functional, code was developed to have the robot commit specific actions based on the input from the sensors and bumpers. The code was created so that, depending on which sensors were sensing the black line, the robot would adjust accordingly to keep on track with the line. This code is shown in Appendix B. The truth table for the functionality of the robot is shown below in table 6.1. To note, when a sensor reads 0, it is reading the line, and when it reads 1, it is not reading the line. Also, CCW correlates to a forward direction, and CW correlates to a backwards direction due to the polarity of the motors connected to the PCB.

Table 6.1: Line Sensor Input and Motor Output Truth Table

Line Sensor			Motor			
Right	Center	Left	R Speed	R Direction	L Speed	L Direction
0	0	0	95%	CCW	100%	CCW
0	1	0	95%	CCW	100%	CCW
0	0	1	75%	CCW	100%	CCW
0	1	1	0%	CCW	100%	CW
1	0	0	95%	CCW	75%	CCW
1	1	0	95%	CCW	0%	CCW
1	1	1	48%	CW	50%	CW

6.5.1 Fault-Finding

During the development of the line-following robot and the code associated with it, several issues arose. Initially, once the robot was fully assembled, there were significant troubles in uploading code to the robot. When the IC programming device, the AVR Dragon, was connected to the robot, it would lose power and not be recognized by MicrochipStudio. The issue turned out to be the battery pack being used. It was not able to supply sufficient stable power to the robot with the dragon connected. Once the battery pack was switched out, the issue was resolved. Additionally, there was an issue with the capacitor used for the filtering of the power input to the ATtiny1634. The capacitance value turned out to be too small, which caused the robot to freeze in place and stop reading the sensor inputs. This was resolved by switching the capacitor for a higher value capacitance, and bridging the connection between the fuse and the capacitor with a current limiting resistor. The code provided fewer complications, and was successfully implemented with few adjustments.

6.6 Chapter Summary

In this chapter, topics related to the design, assembly, fault-finding and testing processes for the Robot project were presented. The microcontroller code, along with a description of how the code works in conjunction with the hardware to enable the robot to follow the line. In particular the following key points were highlighted:

1. *The design of the robot schematic*
2. *The design of the robot PCB*
3. *The process of designing the line-following robot PCB*
4. *The final assembly of the robot*

5. *The process of assembling the line-following robot*

6. *The process of developing the code associated with the line-following robot*

7. *The fault-finding process for the physical assembly the robot as well as the code development*

Chapter 7

Conclusion

This report examined the design, assembly and testing for the Robot and Analogue and Digital Tone Generator “Starger” Projects. In particular:

1. *The scope of the projects*
2. *Research of technology used in real world autonomous vehicles*
3. *The Starger project design, including the use of Altium 365 and Desktop application, the schematic and PCB design, and challenges and successes of the project*
4. *The Starger project assembly, including safety within the project room, the assembly process, the PCB assembly and steps to successful soldering, the assembly of the Starger box, the final assembly of the Starger project, and challenges and successes of the project*
5. *Starger testing and simulation, including theory on the power supply and tone generator, experimental testing of the power supply and tone generator and the equipment and methods used, as well as the results for both the experimental testing and the simulation*

6. *The robot design process and testing, including the schematic design, the PCB design, the final assembly, and the microcontroller code used in the project*

References

[1] R. Lei, L. Tang, J. Guo, J. Sun, and Q. Bu, “Autonomous delivery vehicle system based on multi-sensor data fusion,” 2024, Conference paper, p. 24 – 30.

[2] Z. Dai, Z. Guan, Q. Chen, Y. Xu, and F. Sun, “Enhanced object detection in autonomous vehicles through lidar—camera sensor fusion,” *World Electric Vehicle Journal*, vol. 15, no. 7, 2024.

[3] J. Park, B. K. Thota, and K. Somashekar, “Sensor-fused nighttime system for enhanced pedestrian detection in adas and autonomous vehicles,” *Sensors*, vol. 24, no. 14, 2024.

[4] K. Vinoth and P. Sasikumar, “Multi-sensor fusion and segmentation for autonomous vehicle multi-object tracking using deep q networks,” *Scientific Reports*, vol. 14, no. 1, 2024.

[5] P. Jamuna, K. Kavin Kumar, A. Murugesan, G. Karthikeyan, S. Dineshkumar, and M. Sangeetha, “Intelligent automation in long vehicles through ldr sensor technology for accident prevention,” 2024, Conference paper.

Appendix A

Starger Project - Microcontroller code

```
//initial definitions
#define F_CPU 8000000UL
#include <avr/io.h>
#include <util/delay.h>

// Pin definitions
#define BUTTON_PIN    PA0   // S1 is connected to PA0
#define BUZZER_PIN    PB3   // OC1A output for square wave
#define LED_PIN       PA1   // LED active-low

#define FREQ_COUNT 8 //number of frequencies to cycle

const uint16_t frequency[FREQ_COUNT] = {
  523, 587, 659, 698, 784, 880, 988, 1047 // Frequencies list (Hz)
};

// Set up pin directions and initial states
void setup(void) {
  DDRA |= (1 << LED_PIN);          // LED output
  DDRB |= (1 << BUZZER_PIN);       // Buzzer output (OC1A)
```

```
  DDRA &= ~(1 << BUTTON_PIN);      // Button input
  PORTA |= (1 << BUTTON_PIN);      // Enable pull-up

  PORTA &= ~(1 << LED_PIN);        // LED OFF
}

// Stop Timer1 and turn off buzzer
void stop_beep() {
  TCCR1A = 0;
  TCCR1B = 0;
  OCR1A = 0;
  TCNT1 = 0;
  PORTB &= ~(1 << BUZZER_PIN); // Ensure pin is low
}

// Start Timer1 with prescaler and OCR1A for given frequency
void start_beep(uint16_t freq) {
  stop_beep(); // Reset
  _delay_ms(1);

  //pre-scaling
  uint32_t ocr_val = 0;
  uint8_t cs_bits = 0;
  uint8_t found = 0;

  const struct {
    uint16_t prescale;
    uint8_t cs_bits;
  } options[] = {
    {1,     (1 << CS10)},
    {8,     (1 << CS11)},
    {64,    (1 << CS11) | (1 << CS10)},
```

```
    {256,    (1 << CS12)},
    {1024,   (1 << CS12) | (1 << CS10)}
  };

  for (uint8_t i = 0; i < sizeof(options)/sizeof(options[0]); i++) {
    ocr_val = (F_CPU / (2UL * options[i].prescale * freq)) - 1;
    if (ocr_val <= 65535) {
      cs_bits = options[i].cs_bits;
      found = 1;
      break;
    }
  }

  if (found) {
    OCR1A = (uint16_t)ocr_val;
    TCNT1 = 0;

    TCCR1A = (1 << COM1A0);          // Toggle OC1A on Compare Match
    TCCR1B = (1 << WGM12) | cs_bits; // CTC mode, pre-scaler set
  } else {
    stop_beep(); // Out of range
  }
}

int main(void) {
  setup();

  while (1) {
    // Wait for button press
    if (!(PINA & (1 << BUTTON_PIN))) {
      _delay_ms(5); // De-bounce
      if (!(PINA & (1 << BUTTON_PIN))) {
```

```
        for (uint8_t i = 0; i < FREQ_COUNT; i++) {
          start_beep(frequency[i]*8.107); //scalar found through
           ↪ testing

          PORTA |= (1 << LED_PIN);     // Turn LED ON
          _delay_ms(300);              // Beep duration

          stop_beep();
          PORTA &= ~(1 << LED_PIN);    // Turn LED OFF

          _delay_ms(100);              // Pause between tones
        }

        // Wait for button release
        while (!(PINA & (1 << BUTTON_PIN)));
      }
    }
  }
}
```

Appendix B

Robot Project - Microcontroller code

```
//initial definitions
#define F_CPU 8000000UL
#include <avr/io.h>
#include <util/delay.h>

/* Output defines */
#define ledLeftOn PORTA |= (1 << 1)
#define ledLeftOff PORTA &= ~(1 << 1)
#define ledRightOn PORTC |= (1 << 4)
#define ledRightOff PORTC &= ~(1 << 4)

#define LeftControlCW PORTB |= (1 << 0)
#define LeftControlCCW PORTB &= ~(1 << 0)
#define RightControlCW PORTB |= (1 << 3)
#define RightControlCCW PORTB &= ~(1 << 3)

/* Input defines */
#define leftBumper (!(PINA & (1 << 2)))  // PA2
#define rightBumper (!(PINA & (1 << 0))) // PA0
#define leftSensor (PINA & (1 << 7))     // PA7
#define centreSensor (PINA & (1 << 6))   // PA6
```

```
#define rightSensor (PINA & (1 << 3))    // PA3

//Direction definitions
#define CCW 0
#define CW 1

//Movement functions
#define moveForward() \
SetRightMotorSpeedandDirection(95, CCW); \ //tuned to go straight
SetLeftMotorSpeedandDirection(100, CCW); \
ledLeftOff; ledRightOff;

#define stopAndBackLeft() \
SetLeftMotorSpeedandDirection(45, CW); \
SetRightMotorSpeedandDirection(80, CW); \
_delay_ms(750);

#define stopAndBackRight() \
SetLeftMotorSpeedandDirection(80, CW); \
SetRightMotorSpeedandDirection(45, CW); \
_delay_ms(750);

#define rightPivot() \
SetRightMotorSpeedandDirection(0, CCW); \
SetLeftMotorSpeedandDirection(100, CCW); \
ledLeftOff; ledRightOn;

#define leftPivot() \
SetRightMotorSpeedandDirection(95, CCW); \
SetLeftMotorSpeedandDirection(0, CCW); \
ledLeftOn; ledRightOff;
```

```
#define rightAdjust() \
SetRightMotorSpeedandDirection(75, CCW); \
SetLeftMotorSpeedandDirection(100, CCW); \
ledLeftOff; ledRightOn;

#define leftAdjust() \
SetRightMotorSpeedandDirection(95, CCW); \
SetLeftMotorSpeedandDirection(75, CCW); \
ledLeftOn; ledRightOff;

#define backUp() \
SetRightMotorSpeedandDirection(48, CW); \
SetLeftMotorSpeedandDirection(50, CW); \
ledLeftOn; ledRightOn;

//initialize motors
void setup(void);
void SetLeftMotorSpeedandDirection(unsigned char Speed, char
 ↪  Direction);
void SetRightMotorSpeedandDirection(unsigned char Speed, char
 ↪  Direction);

int main(void)
{
  setup();

  while (1)
  {
    //Handle bumper input
    if (leftBumper && !rightBumper) {
      stopAndBackRight();
      continue;
```

```
    }
    else if (rightBumper && !leftBumper) {
      stopAndBackLeft();
      continue;
    }

    /* Read and convert sensor input to active-high logic (1 = line
     ↪  seen) */
    unsigned char sensorPattern = 0;
    if (!rightSensor)  sensorPattern |= (1 << 2); // Right
    if (!centreSensor) sensorPattern |= (1 << 1); // Centre
    if (!leftSensor)   sensorPattern |= (1 << 0); // Left

    switch (sensorPattern)
    {
      case 0b000: // All on line
      case 0b010: // Only Left an Right sensor (shouldn't happen)
      moveForward();
      break;

      case 0b001: // Centre & Right on line
      rightAdjust();
      break;

      case 0b011: // Only Right
      rightPivot();
      break;

      case 0b100: // Centre & Left on line
      leftAdjust();
      break;
```

```
      case 0b110: // Only Left
      leftPivot();
      break;

      case 0b111: // No line sensed
      backUp();

      break;

      case 0b101: // Only center sees line
      moveForward();
      break;

      default: //if breaks, go forward and hope
      moveForward();
      break;
    }
  }
}

void setup(void)
{
  DDRA = 0b00100010; //set PORTA according to the schematic
  DDRB = 0b00001001; //set PORTB according to the schematic
  DDRC = 0b00010001; //set PORTC according to the schematic
  /* Timer setup for Motors*/
  TCCR0A = (1 << WGM01) | (1 << WGM00); // fast pwm channel A and B,
   ↪ Top = 0xFF;
  TCCR0B = (1 << CS02) | (1 << CS00); // prescaling = /1024
  OCR0A = 0; // Right motor off
  OCR0B = 0; // Left motor off
  PORTA |= (1 << 1);
```

```
  PORTC |= (1 << 4);
}

//set up left motor control
void SetLeftMotorSpeedandDirection(unsigned char Speed, char
 ↪  Direction)
{
  if (Speed > 100) Speed = 100;

  if (Speed == 0)
  {
    TCCR0A &= ~((1 << COM0B1) | (1 << COM0B0));
    LeftControlCCW;
    OCR0B = 0;
  }
  else
  {
    OCR0B = (Speed * 255) / 100;

    if (Direction == CCW)
    {
      TCCR0A |= (1 << COM0B1);
      TCCR0A &= ~(1 << COM0B0);
      LeftControlCCW;
    }
    else
    {
      TCCR0A |= (1 << COM0B1) | (1 << COM0B0);
      LeftControlCW;
    }
  }
}
```

```
//Set up right motor control
void SetRightMotorSpeedandDirection(unsigned char Speed, char
 ↪  Direction)
{
  if (Speed > 100) Speed = 100;

  if (Speed == 0)
  {
    TCCR0A &= ~((1 << COM0A1) | (1 << COM0A0));
    RightControlCCW;
    OCR0A = 0;
  }
  else
  {
    OCR0A = (Speed * 255) / 100;

    if (Direction == CCW)
    {
      TCCR0A |= (1 << COM0A1) | (1 << COM0A0);
      RightControlCW;
    }
    else
    {
      TCCR0A |= (1 << COM0A1);
      TCCR0A &= ~(1 << COM0A0);
      RightControlCCW;
    }
  }
}
```