ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

Bicycle Radar Array

Team #91

RAHUL NAYAK
(rn8@illinois.edu)
CHARLIE WANG
(cgwang3@illinois.edu)
NATHAN ZHU
(nyzhu2@illinois.edu)

TA: Frey Zhao

May 6, 2026

Abstract

Cycling in dense environments poses safety risks due to interactions between cyclists and pedestrians, with existing solutions relying on manual alerts and rider reaction time. This project presents a radar-based bicycle safety system that provides automated hazard detection and adaptive alerts. Multiple millimeter-wave (mmWave) radar sensors measure distance and velocity of nearby objects, while an ESP32 processes this data to estimate collision risk. The system generates multi-stage auditory alerts and operates on a portable power subsystem using an 18650 battery with integrated regulation and protection circuitry. Testing shows reliable detection of moving human targets within approximately 3.5 meters, with reduced accuracy at greater distances. The system successfully integrates sensing, processing, and alerting into a compact platform. Limitations include decreased long-range detection performance and limited alert flexibility due to the use of an active buzzer.

Contents

1 Introduction **1**
- 1.1 Problem . . . 1
- 1.2 Solution . . . 1
- 1.3 High Level Requirements . . . 1
- 1.4 Block Diagram . . . 1

2 Design **3**
- 2.1 Power . . . 3
  - 2.1.1 Buck-Boost Converter . . . 3
  - 2.1.2 Voltage Regulator . . . 4
  - 2.1.3 USB-C and Charging . . . 4
  - 2.1.4 Battery Management System . . . 4
- 2.2 Radar . . . 5
- 2.3 Microcontroller . . . 6
- 2.4 User I/O . . . 7
- 2.5 PCB . . . 8

3 Design Verification **9**
- 3.1 Power Subsystem . . . 9
- 3.2 Radar Subsystem . . . 10
- 3.3 Microcontroller Subsection . . . 11
- 3.4 User I/O Subsection . . . 12

4 Costs **15**
- 4.1 Parts . . . 15
- 4.2 Labor . . . 15
- 4.3 Total . . . 16

5 Conclusion **17**
- 5.1 Accomplishments . . . 17
- 5.2 Limitations . . . 17
- 5.3 Ethical Considerations . . . 17
- 5.4 Future Work . . . 17

References **19**

Appendix A PCB Design **20**

Appendix B RV Tables **21**

Appendix C Microcontroller Testing **23**

1 Introduction

1.1 Problem

Cycling has long been an essential pillar of urban and academic mobility. For students navigating sprawling university campuses and professionals commuting through dense city centers, the bicycle represents an efficient, sustainable, and vital mode of transport. However, as these environments become increasingly congested, the shared infrastructure between cyclists and pedestrians is becoming a site of growing conflict.

However, this system has many flaws. A cyclist must manually identify a hazard, process the risk, and activate a mechanical bell, hoping that the pedestrian reacts in time, making the entire sequence reactive rather than proactive. This requires a quick reaction time from the cyclist, leaving little time for error, especially when pedestrians and riders alike are increasingly being distracted by mobile devices. Furthermore, manual detection is significantly compromised in low-visibility conditions, such as night-time commuting or heavy rain, where the visual range of a rider is reduced.

1.2 Solution

To address this issue, our group has created an automated alert system, completely removing the need for the cyclist to identify a hazard, and ring the bell.

1.3 High Level Requirements

Our three high level requirements are:

- The system must achieve a minimum detection rate of 90% for moving hazards within a 3.5 meter range.
- The total elapsed time from the initial sensor detection of an object to the activation of the audible alert must not exceed 200 milliseconds to ensure the cyclist has sufficient time to react.
- The power management subsystem must provide a minimum of 8 hours of continuous operation on a single charge of the 18650 battery, ensuring the device remains active for a standard commuting or delivery shift.

1.4 Block Diagram

As shown in Figure 1, the system is composed of four primary subsystems: Power, Radar, Microcontroller, and User Input/Output (I/O). These subsystems interact to detect nearby hazards and provide alerts to the rider.

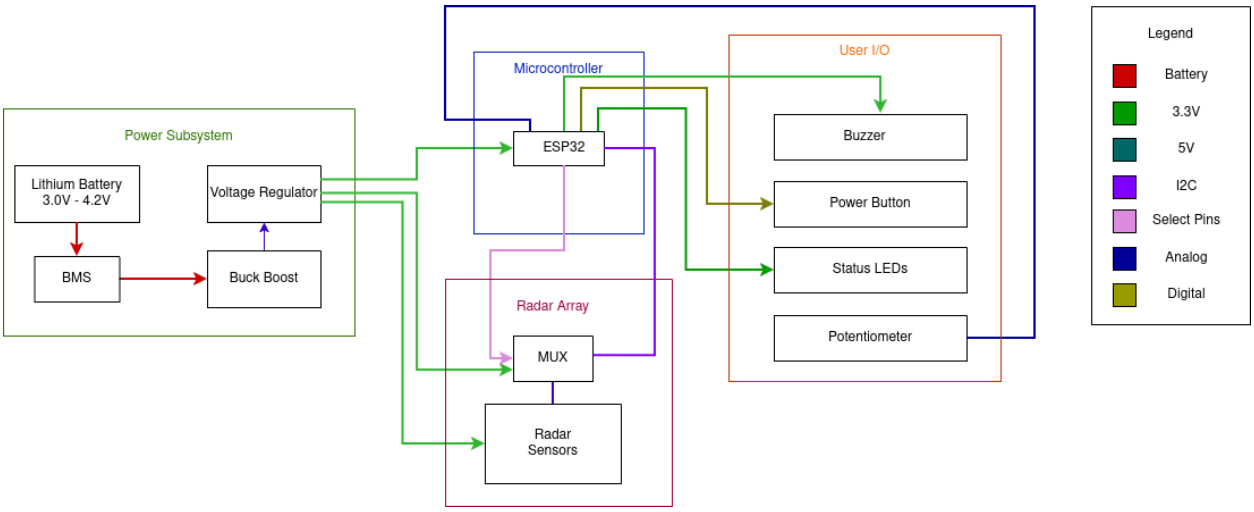

Figure 1: Block Diagram

The radar subsystem continuously monitors the environment and provides distance and velocity data for detected objects. This data is transmitted to the microcontroller subsystem, which serves as the central processing unit of the system. The microcontroller analyzes incoming sensor data and determines the level of collision risk.

Based on this analysis, the microcontroller drives the user I/O subsystem, which provides feedback to the rider through auditory alerts and visual indicators. The power subsystem supplies regulated voltage to all components, ensuring stable operation across varying load conditions.

This high-level architecture enables real-time hazard detection and response while maintaining a modular design for ease of development and testing.

2 Design

2.1 Power

2.1.1 Buck-Boost Converter

In order to generate a stable $5\,\mathrm{V}$ supply from the 18650 Lithium-ion (Li-ion) battery, whose voltage varies between approximately $3.3\,\mathrm{V}$ and $4.3\,\mathrm{V}$, the TPS63060 buck-boost converter was selected. The output voltage of the TPS63060 is set using a resistor divider connected to the feedback (FB) pin. According to its datasheet, the feedback reference voltage V_{FB} is $0.5\,\mathrm{V}$ [1]. The relationship between the output voltage and the resistor values is given by the standard feedback equation:

$$V_{OUT} = V_{FB}\left(1 + \frac{R_1}{R_2}\right) \tag{1}$$

For this design, resistor values of $R_1 = 900\,\mathrm{k\Omega}$ and $R_2 = 100\,\mathrm{k\Omega}$ were selected to produce a regulated output of $5\,\mathrm{V}$. These values satisfy the required ratio while maintaining relatively high resistance to minimize power loss through the divider network.

The remaining component values, including the inductor and capacitors, were selected based on manufacturer recommendations to ensure stable operation across the expected input voltage range and load conditions, as shown in Figure 2.

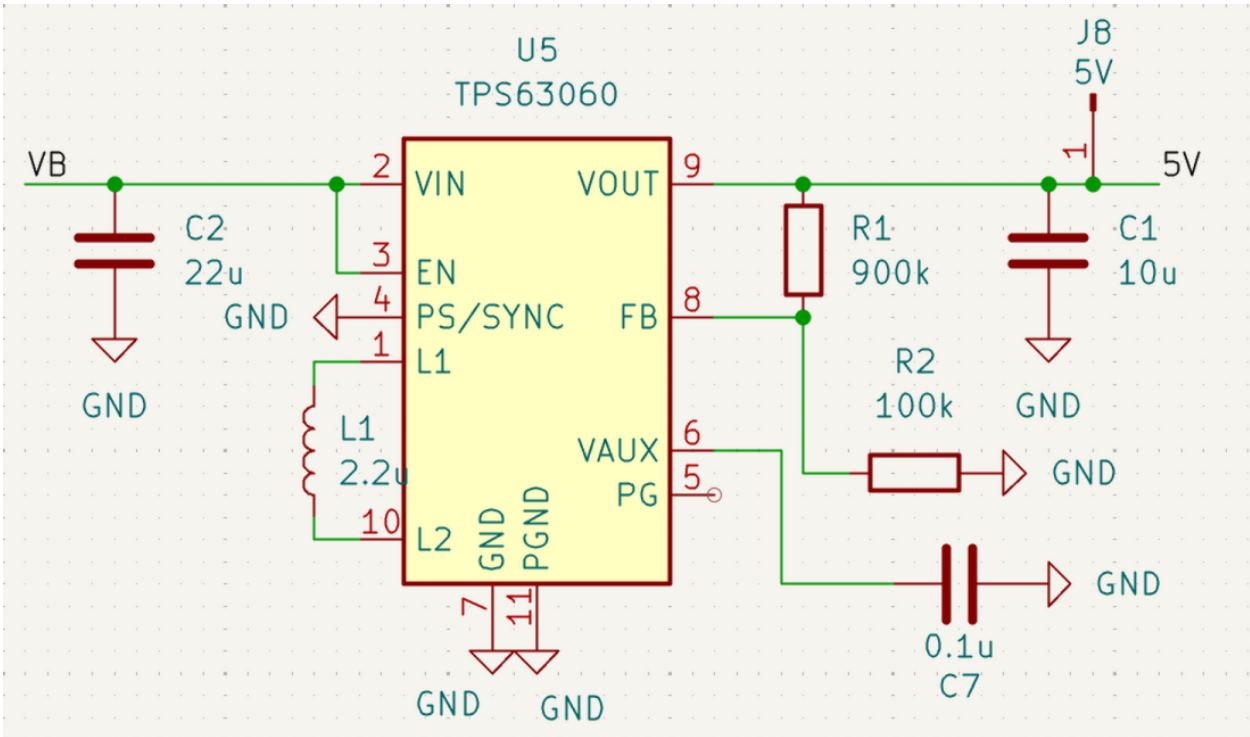

Figure 2: Buck-Boost Converter Schematic

2.1.2 Voltage Regulator

The AMS1117-3.3's purpose is to ensure that the ESP32 is able to receive a constant and stable $3.3\,\mathrm{V}$ input voltage. Due to its relatively high dropout voltage of $1.2\,\mathrm{V}$, it is one of the reasons the $5\,\mathrm{V}$ boost is required, as the 18650 battery cannot reliably provide the required input voltage for a long duration [2] [3]. The input and output capacitors follow the recommendations given in the datasheet.

2.1.3 USB-C and Charging

The MCP73831 battery charge management controller was selected for its integrated Li-ion charging capabilities, with the charge current configured via an external programming resistor to match the battery specifications [4]. As shown in Figure 3, the USB-C interface is configured in a power-only mode using $5.1\,\mathrm{k\Omega}$ pull-down resistors on the CC lines to correctly advertise the device as a power sink and enable $5\,\mathrm{V}$ delivery from the source [5].

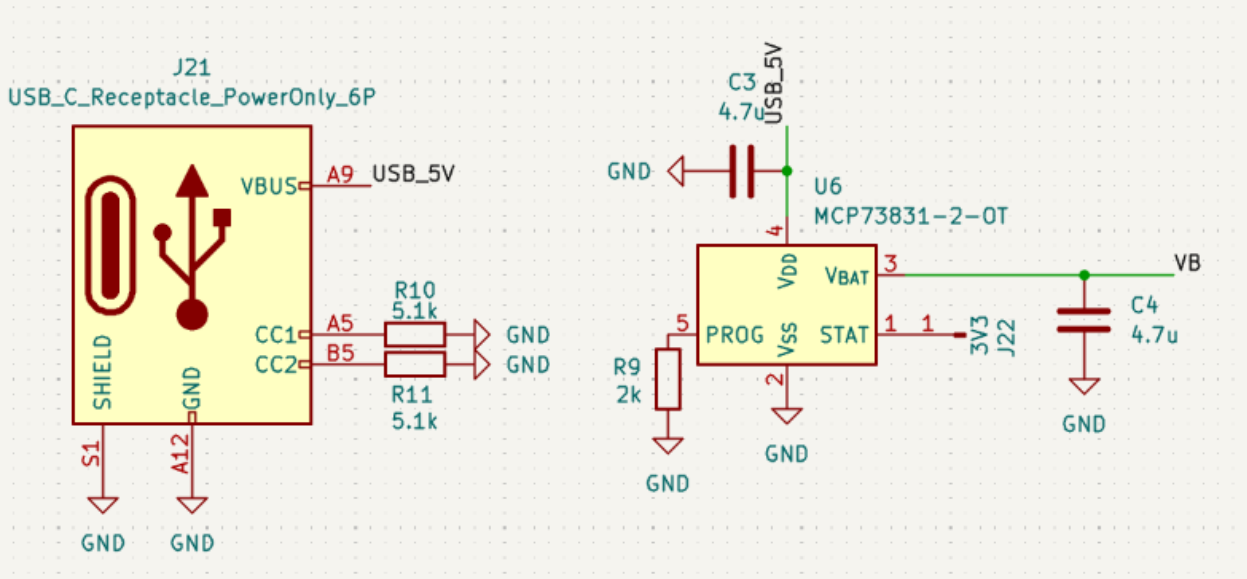

Figure 3: USB-C and Charging Schematics

2.1.4 Battery Management System

To ensure safe operation of the 18650 Li-ion battery, a battery management system (BMS) was implemented using the DW01A protection IC. The DW01A monitors the battery voltage and current, and controls the gate of two external MOSFETs to regulate the connection between the battery and the system. This dual MOSFET configuration, as shown in Figure 4, enables bidirectional control, allowing both charging and discharging currents under normal conditions while disconnecting the battery when fault conditions are detected.

Under overcharge or overdischarge conditions, the DW01A disables the MOSFETs to isolate the battery, preventing damage and extending battery lifespan. Similarly, in the event of an overcurrent condition or a short-circuit, the protection circuit rapidly disconnects the load to ensure safe operation [6].

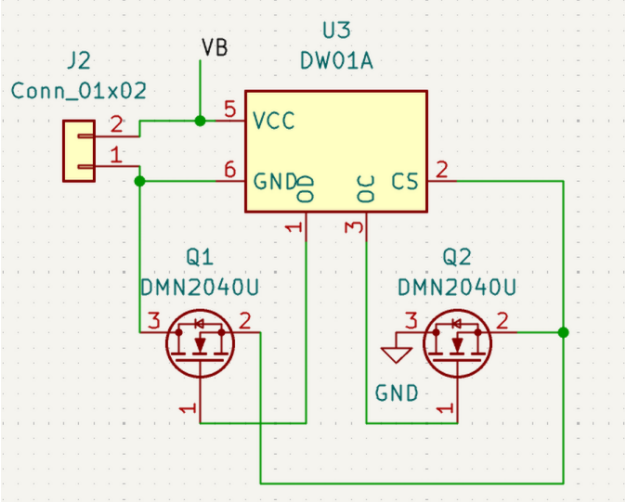

Figure 4: Battery Management Schematic

2.2 Radar

The radar subsystem is responsible for detecting nearby objects and providing distance and velocity data to the system. This is achieved using multiple millimeter-wave (mmWave) radar sensors, which enable reliable detection of moving objects in a variety of environmental conditions, including low visibility scenarios where vision-based systems may be ineffective.

The original design had us using three sensors, in which we would detect specific regions pertaining to each sensor, with a left, center, and right region. Our algorithm would attempt to determine the direction of motion of the object based on which sensors were picking it up to use in our collision detection.

Each radar sensor communicates over the Inter-Integrated Circuit (I^2C) protocol; however, the sensors support only two I^2C addresses, creating an address conflict when using three or more devices on the same bus. To resolve this, we planned to use a TCA9548A I^2C multiplexer (MUX), as shown in Figure 5. The MUX allows the ESP32 to selectively enable one sensor at a time by switching between independent I^2C channels, effectively isolating each sensor and preventing address conflicts.

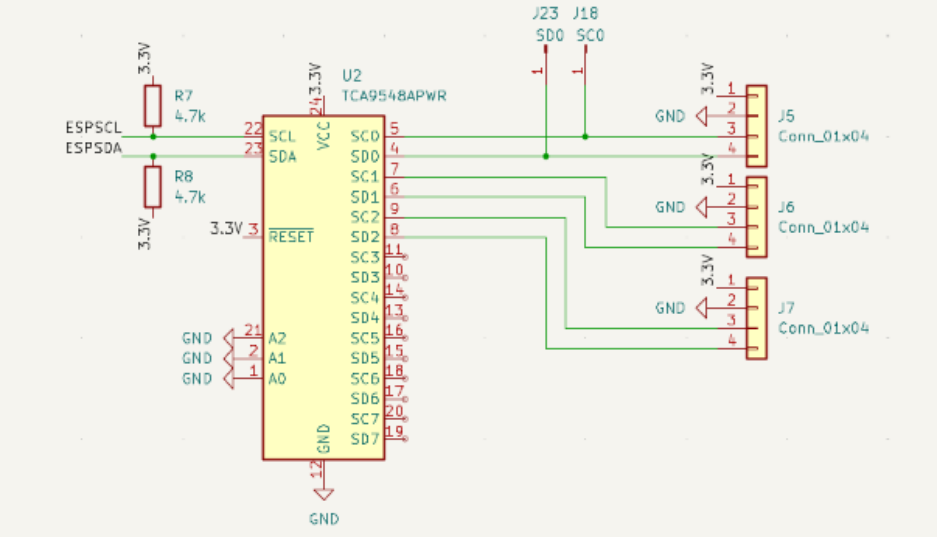

Figure 5: Radar Schematic

Pull-up resistors of $4.7\,\mathrm{k\Omega}$ were used on the Serial Data (SDA) and Serial Clock (SCL) lines to ensure proper I^{2}C communication at $3.3\,\mathrm{V}$ logic levels. Each sensor is connected to a separate downstream channel of the MUX, allowing the microcontroller to sequentially query each sensor and collect data.

This multiplexer was ultimately not used in the final implementation, as the system achieved the desired performance using only two sensors. This configuration provided sufficient field-of-view coverage and position estimation accuracy, eliminating the need for additional I^{2}C channel expansion.

Originally, the system was designed to use the universal asynchronous receiver-transmitter (UART) protocol in conjunction with a multiplexer to interface multiple sensors with the microcontroller. However, during testing, the UART interface on the sensors proved unreliable, producing inconsistent or invalid data outputs. As a result, the design was transitioned to the aforementioned I^{2}C-based communication scheme, which provided stable and consistent data transmission.

2.3 Microcontroller

The ESP32-S3 microcontroller serves as the central processing unit of the system, responsible for coordinating sensor data acquisition, processing incoming measurements, and generating appropriate user alerts. The microcontroller interfaces with the radar sensors over I^{2}C and communicates with the user I/O components, including the buzzer, LEDs, and input controls.

The primary functionality of the microcontroller is the implementation of a hazard detection algorithm based on distance and velocity data from the radar sensors. Sensor readings are collected sequentially and processed in real time to estimate the likelihood of a potential collision. Rather than relying solely on distance thresholds, the system

incorporates velocity data to prioritize objects moving toward the rider, allowing for more intelligent filtering of irrelevant detections such as stationary objects, since objects with near-zero velocity generally do not pose an immediate collision risk despite being nearby.

A simplified time-to-collision (TTC) approach was used to determine the urgency of a detected object. By combining measured distance and relative velocity, the system estimates how quickly an object is approaching and assigns it to one of three risk levels (i.e. low, medium, high). These risk levels are then mapped to different alert patterns in the buzzer output.

The algorithm was designed to operate within a fixed execution time constraint, ensuring that all necessary sensor data could be processed within a $50\,\mathrm{ms}$ loop cycle. This real-time requirement influenced the choice of a relatively lightweight computation method rather than more complex tracking or filtering algorithms.

In order to estimate an object's position, distance measurements from the two radar sensors were combined using a simplified trilateration-based approach. Each sensor provides a scalar distance measurement to the detected object, which can be interpreted geometrically as a circle centered at the sensor location with a radius equal to the measured distance. The potential position of the object therefore lies along the circumference of this circle for each sensor.

By considering both sensors simultaneously, the system approximates the object's location as the intersection region of the two circles. Since only two sensors are used, this does not yield a unique solution in all cases; instead, the algorithm selects the most physically reasonable position based on the expected direction of motion and the sensor field of view. In practice, this results in a sufficient approximation of the object's lateral and forward position relative to the bicycle.

This estimated position is then used in conjunction with velocity measurements to evaluate collision risk and determine the appropriate alert level, as shown in Figure **??**. Velocity measurements are additionally filtered to reject abrupt discontinuities likely caused by transient reflections or measurement noise.

2.4 User I/O

The User I/O subsystem provides the interface between the system and the rider, enabling both system feedback and user control. This subsystem consists of an audible buzzer, status light emitting diodes (LED), a power button, and a potentiometer for sensitivity adjustment.

The primary output mechanism is the buzzer, which alerts the rider to potential hazards detected by the system. Based on the risk level determined by the microcontroller, the system generates distinct alert patterns, including intermittent beeps and continuous tones, to indicate increasing levels of urgency. This allows the rider to quickly interpret the severity of a potential collision without requiring visual attention. During the initial

design, a passive buzzer was considered to allow for more flexible control over the output signal. However, this approach was ultimately not pursued, as varying volume alone was not expected to provide sufficiently distinct feedback across different environmental conditions, where lower-volume alerts could be masked by ambient noise. Instead, the design focused on varying the tempo of the alert signal through beep frequency, which provides more perceptually distinct and reliable cues to the rider. This approach enables clearer differentiation between risk levels without relying on precise control of sound amplitude.

Visual feedback is provided through status LEDs, which indicate system states such as power and operation status. User input is handled through a push-button used to control system power or enable/disable operation, as well as a potentiometer that allows the rider to adjust detection sensitivity. The potentiometer modifies the internal threshold values, enabling the system to adapt to different environments, such as low-traffic or high-density areas.

The design of the User I/O subsystem prioritizes simplicity and minimal rider distraction. By relying primarily on auditory alerts and simple controls, the system allows the rider to maintain focus on the surrounding environment while still receiving timely hazard notifications.

2.5 PCB

Our printed circuit board (PCB) was designed based on the overall schematic, with a layout organized into four distinct regions corresponding to each subsystem. This partitioning was intentionally designed to simplify debugging and allow for easier isolation and testing of individual subsystems.

The board is divided into four quadrants: the power subsystem in the top left, the microcontroller in the top right, the radar subsystem in the bottom left, and the user I/O subsystem in the bottom right, as shown in Figure A.1 (Appendix A). This structured layout also helps reduce interference between subsystems by separating high-current power circuitry from more sensitive signal-processing components.

An exception to this organization is the USB-C receptacle, located within the radar subsystem region. This placement was chosen during a later design iteration due to space constraints after it was omitted in an earlier revision, so the device was positioned in the nearest available area without significantly impacting system performance.

3 Design Verification

3.1 Power Subsystem

The power subsystem consists of multiple interconnected components and was therefore verified by testing each component individually rather than as a fully integrated system. The corresponding requirements and verification table is shown in Table B.1 (Appendix B).

Due to manufacturing and assembly issues, the buck-boost converter was not able to be successfully verified. During PCB assembly, inconsistent solder paste application resulted in poor electrical connections, preventing proper operation of the converter. Additionally, a footprint mismatch caused by a mistake in the package variant further contributed to unreliable connectivity. An inductor with a slightly higher inductance than the recommended value was also used during assembly, due to lack of proper components and time constraints. While this would not necessarily prevent the converter from operating, it may have altered the converter's ripple current, transient response, and overall regulation behavior. Because this deviation occurred alongside soldering and footprint issues, it introduced additional uncertainty when diagnosing the failed buck-boost stage. This limitation prevented verification of the $5\,\mathrm{V}$ rail and impacted overall power subsystem performance. Despite the failed buck-boost stage, system operation remained possible because all components could operate directly from the regulated $3.3\,\mathrm{V}$ supply.

The voltage regulator was successfully tested by measuring its output using a multimeter. A voltage of $3.242\,\mathrm{V}$ was recorded, which falls within the acceptable tolerance of $\pm 0.1\,\mathrm{V}$, as can be seen in Figure 6. This means that, even with diminished longevity, our system is still operating under safe voltage levels.

Figure 6: Regulator Output Voltage

The total current draw of the system was measured to be approximately $0.237\,\mathrm{A}$ during full load as shown in Figure 7. Using a $2200\,\mathrm{mA\,h}$ ($2.2\,\mathrm{A\,h}$) 18650 battery, the expected

battery life can be estimated by dividing the battery capacity by the average current draw. This yields an approximate runtime of 9.3 hours under continuous operation, exceeding our requirement of 8 hours.

$$t = \frac{C}{I} = \frac{2.2\,\mathrm{A\,h}}{0.237\,\mathrm{A}} \approx 9.28\,\mathrm{h} \tag{2}$$

This estimate assumes ideal battery capacity utilization and approximately constant current consumption. Actual runtime may be reduced due to converter inefficiencies, battery aging, and transient current spikes during operation.

Figure 7: Total Current Output

The BMS also exhibited issues due to improper implementation. A required resistor for the current sensing pin was omitted, preventing the DW01A protection IC from correctly detecting overcurrent conditions. As a result, the protection circuitry was unable to function as intended.

Because of this, the BMS was effectively bypassed, leaving the battery unprotected against conditions such as over-discharge and excessive current draw. This introduces significant safety and reliability concerns, as the system could operate outside of the battery's safe limits without any automatic shut-off or protection. This issue does not directly prevent the battery from charging, as the MCP73831 charge controller operates independently of the protection circuit. However, the absence of a functional BMS removes critical safety features during both charging and discharging.

Overall, the power subsystem only partially satisfied its requirements due to failure of the buck-boost converter and BMS protection circuitry.

3.2 Radar Subsystem

The radar subsystem relies primarily on the performance of the selected sensors, and verification focused on evaluating detection accuracy and reliability. The corresponding requirements and verification table is shown in Table B.2 (Appendix B).

While the original architecture proposed three radar sensors and an I^2C multiplexer, the final implementation used two sensors without the multiplexer after testing demonstrated sufficient field-of-view coverage and positional estimation accuracy. As shown in Figure 8, the radar sensors demonstrated reliable detection performance up to approximately $3.5\,\mathrm{m}$. Across 50 walking trials at each tested distance, where a subject approached the sensor at a normal walking speed under consistent lighting and environmental conditions, the detection rate remained above the required threshold through $3.5\,\mathrm{m}$. Beyond this range, reliability decreased significantly, with detection performance dropping at $4.0\,\mathrm{m}$ and becoming inconsistent at $4.5\,\mathrm{m}$. These results indicate that the radar subsystem satisfies the intended detection requirement within the target range, but performance becomes unreliable at longer distances. The detections observed at $4.5\,\mathrm{m}$ were inconsistent and were likely caused by sensor noise or intermittent false detections rather than stable target detection.

Overall, the radar subsystem successfully satisfied the minimum detection-range requirement of $3.5\,\mathrm{m}$.

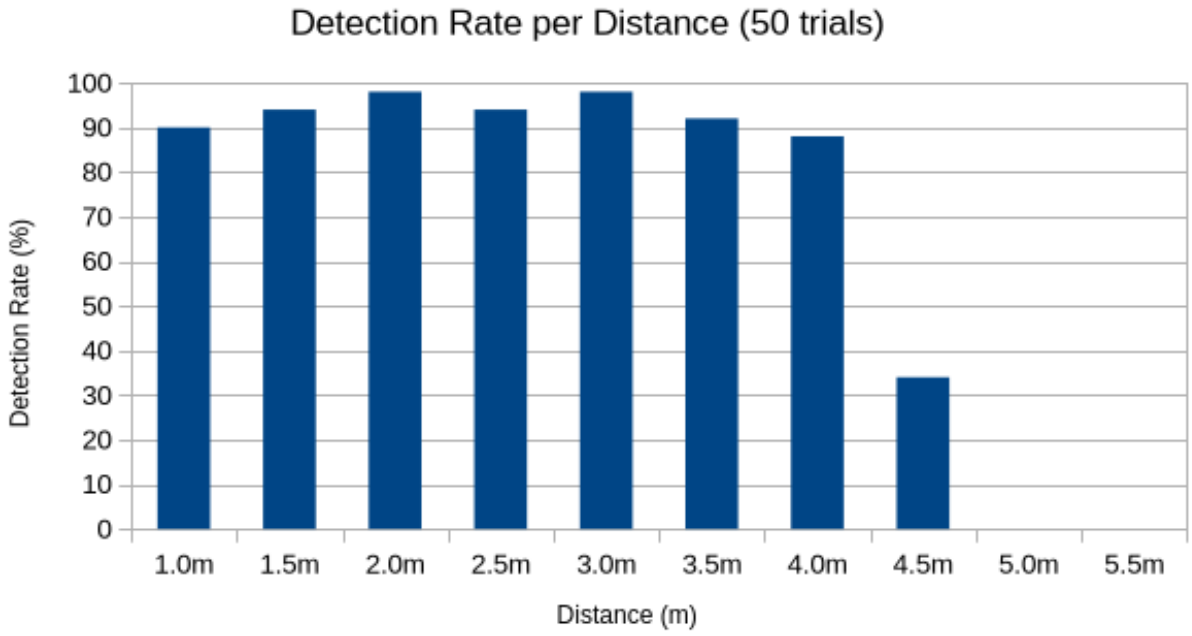

Figure 8: Radar Detection Over 50 Trials

3.3 Microcontroller Subsection

The microcontroller subsystem was primarily verified through software-based testing and real-time data observation. Proper I^2C communication with the radar sensors was confirmed by monitoring data output through the serial interface, ensuring that valid

distance and velocity measurements were consistently received. The corresponding requirements and verification table are shown in Table B.3 (Appendix B).

To further validate system functionality, sensor data was streamed and visualized using an external Python script, allowing real-time plotting of both distance and estimated position. These plots, shown in Figure 9, demonstrate continuous and responsive data acquisition, as well as the system's ability to process and interpret sensor inputs, including our trilateration algorithm. This approach provided both qualitative and quantitative confirmation that the microcontroller was correctly interfacing with the sensors and executing the required data processing tasks.

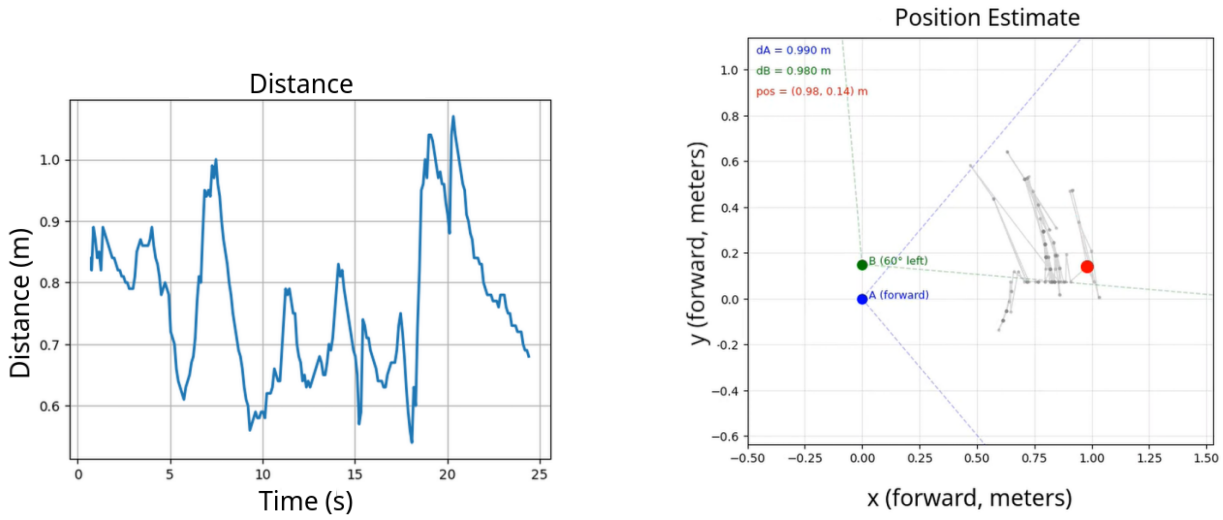

Figure 9: Python Graphs of Radar Output

To evaluate the execution time of the algorithm, timestamps were inserted at the beginning and end of each processing loop to measure the loop duration. The total execution time was found to be less than $1\,\mathrm{ms}$ per iteration, as shown in Figure C.1 (Appendix C).

Overall, the microcontroller subsystem successfully satisfied the timing requirement of algorithm completion in under $50\,\mathrm{ms}$ and demonstrated successful communication with the sensors and distinct buzzer output patterns.

3.4 User I/O Subsection

The User I/O subsystem was verified by individually testing each input and output component to ensure correct functionality and integration with the microcontroller. The corresponding requirements and verification table is shown in Table B.4 (Appendix B).

The buzzer was tested by triggering each risk level within the system and observing the corresponding alert patterns. Distinct auditory outputs were successfully generated for low, medium, and high risk levels, implemented as slow beeping, rapid beeping, and a

continuous tone, respectively. These patterns were clearly distinguishable and provided intuitive feedback. Additionally, the volume of the buzzer was around $75\,\mathrm{dB}$ shown in Figure 10, which is similar to a typical bicycle bell [7].

Figure 10: Buzzer Volume

The LED indicators were verified by observing their response to system state changes. The LEDs correctly reflected power and operational status, providing visual confirmation of system activity.

The push-button input was tested by repeatedly toggling the system state and verifying the corresponding response through LED indication and overall system behavior. Software debouncing was also implemented to eliminate unintended rapid toggling caused by mechanical noise. As a result, the button produced consistent and reliable state changes when pressed and held for a short period of time.

The potentiometer was verified by measuring its analog output using the microcontroller's Analog-to-Digital Converter (ADC) while adjusting its position. The readings changed smoothly across the expected range, and the corresponding sensitivity thresholds within the system were successfully modified, demonstrating proper functionality as seen in Figure 11.

Overall, all aspects of user I/O were successfully demonstrated.

```
scale=1.95  close=0.98  med=1.95  far=3.90  d=0.810  latency=0ms
scale=1.94  close=0.97  med=1.94  far=3.88  d=0.810  latency=0ms
scale=1.93  close=0.97  med=1.93  far=3.86  d=0.810  latency=0ms
scale=1.88  close=0.94  med=1.88  far=3.75  d=0.820  latency=0ms
scale=1.86  close=0.93  med=1.86  far=3.73  d=0.820  latency=0ms
scale=1.85  close=0.92  med=1.85  far=3.70  d=0.820  latency=0ms
scale=1.78  close=0.89  med=1.78  far=3.57  d=0.820  latency=0ms
scale=1.77  close=0.89  med=1.77  far=3.54  d=0.820  latency=0ms
scale=1.76  close=0.88  med=1.76  far=3.51  d=0.820  latency=0ms
scale=1.75  close=0.87  med=1.75  far=3.49  d=0.820  latency=0ms
scale=1.68  close=0.84  med=1.68  far=3.36  d=0.820  latency=0ms
scale=1.66  close=0.83  med=1.66  far=3.32  d=0.820  latency=0ms
scale=1.64  close=0.82  med=1.64  far=3.29  d=0.820  latency=0ms
scale=1.57  close=0.79  med=1.57  far=3.15  d=0.820  latency=0ms
scale=1.57  close=0.79  med=1.57  far=3.14  d=0.820  latency=0ms
scale=1.56  close=0.78  med=1.56  far=3.12  d=0.820  latency=0ms
scale=1.54  close=0.77  med=1.54  far=3.07  d=0.820  latency=0ms
```

Figure 11: Potentiometer Scaling

4 Costs

4.1 Parts

While most parts were ordered through our provided funds, many late purchases were made with our own money. SMD components were not considered for pricing. The individual and total costs, using bulk values, are as shown in Table 1.

Table 1: Parts Costs

Part	Manufacturer	Retail Cost ($)	Bulk Purchase Cost ($)	Quantity	Actual Cost ($)
HP-090403 Polycase Enclosure	Polycase	18.05	18.05	1	18.05
SEN0610 mmWave Sensor	DFRobot	12.90	12.90	3	38.70
ESP32-S3-WROOM-1	Espressif Systems	5.66	4.05	1	4.05
AMS1117-3.3	EVVO	0.29	0.13	1	0.13
DW01A	Shenzhen	0.10	0.01	1	0.01
DMN2040U	Diodes Incorporated	0.51	0.08	2	0.16
2N2219	Central Semiconductor Corp	3.47	1.03	1	1.03
TPS63060	Texas Instruments	3.22	1.85	1	1.85
TCA9548A	Texas Instruments	1.36	0.70	1	0.70
USB4135-GF-A	GCT	0.68	0.44	1	0.44
SFM-27	Handson Technology	1.39	0.79	1	0.79
1505 Pushbutton	Adafruit	0.95	0.95	1	0.95
18650 Battery	Adafruit	9.95	9.95	1	9.95
RV4NAYSD253A Potentiometer	Precision Electronics Corporation	22.44	13.31	1	13.31
HLMP3301 LED	Everlight Electronics Co Ltd	0.24	0.06	5	0.30
Total					**90.42**

Additionally, there were a few parts that were purchased for prototyping and testing that were not be required for actual manufacturing, as shown in Table 2.

Table 2: Parts Costs

Part	**Cost ($)**	**Quantity**	**Actual Cost ($)**
ESP32-S3 DevBoard	15.00	1	15.00
Bicycle	40.00	1	40.00
Screwdriver Kit	14.99	1	14.99
Total			**69.99**

4.2 Labor

Using the equation provided in the final report guidelines,

$$C = r \cdot h \cdot 2.5 \tag{3}$$

where r is the ideal hourly rate and h is the number of hours worked, the total cost per team member is calculated as shown in Table 3. We can estimate the average salary of an engineering student to be around $40 per hour [8].

Table 3: Labor Costs

Team Member	**$/hr**	**Hours Worked per Week**	**Weeks Worked**	**Total Cost ($)**
Rahul Nayak	40	6	14	8400
Charlie Wang	40	6	14	8400
Nathan Zhu	40	6	14	8400
Total				**25200**

Using this same salary for the machine shop workers, and with the relative simplicity of our project's design, we can estimate a 2 hour completion time for production, providing us with an assembly cost of $\$40 \times 2 = \$80$.

4.3 Total

Adding all of our costs together, and assuming we are able to get bulk pricing on all components, we are left with a total cost of $\$90.42 + \$69.99 + \$25200 + \$80 = \$25440.41$.

5 Conclusion

5.1 Accomplishments

The final system successfully integrated radar sensing, real-time processing, and user feedback into a cohesive platform capable of detecting nearby moving objects and generating hazard alerts with sub-millisecond processing latency. The system achieved reliable object detection within approximately $3.5\,\mathrm{m}$ and generated appropriate alerts in real time, satisfying the primary functional requirements of the project despite several hardware implementation challenges.

5.2 Limitations

The original design required a higher intermediate voltage to support proper regulation down to $3.3\,\mathrm{V}$, as linear regulators require sufficient headroom between input and output voltages. However, a reliable $5\,\mathrm{V}$ rail was not successfully implemented during testing. This reduced the effective operating lifetime of the system, since the battery voltage could no longer be efficiently regulated across the full discharge range of the 18650 cell.

The selected SEN0610 sensors exhibited intermittent false detections, likely caused by multipath reflections and environmental clutter inherent to low-cost mmWave sensing systems. This introduced uncertainty in detection reliability and overall system performance. Alternative sensing solutions or improved filtering algorithms may be required to achieve more consistent and accurate detection results.

5.3 Ethical Considerations

This project is designed in strict accordance with IEEE Code of Ethics 1.1, which mandates holding the safety and privacy of the public paramount [9]. The use of mmWave radars, as opposed to cameras, ensures that the privacy of the rider and pedestrians is maintained, and all data is processed locally on board in real time, with no capturing of identifiable information.

Additionally, a significant ethical consideration for a project of this nature is the potential for automation complacency. There is a risk that cyclists, knowing that they have an automated alarm, may become less vigilant or more likely to engage in distracted riding. To mitigate this, the system is designed as an assistive warning device rather than an autonomous safety system, ensuring that the rider remains the primary decision maker.

5.4 Future Work

The biggest improvement would come from switching to improved sensors. LiDAR offers higher accuracy and improved range, which could significantly improve object detection and positioning. However, this comes with tradeoffs, as LiDAR performance may

degrade in adverse weather conditions such as rain or fog, and is generally more expensive. Improving the detection and position algorithm would also significantly improve positional accuracy and hazard classification.

Another area for improvement is reducing the overall footprint of the device. The current design is limited by the size of the enclosure and PCB. A more compact PCB layout and optimized enclosure design would improve usability and allow for easier mounting and a better cycling experience.

Finally, the addition of a secondary feedback mechanism, such as a vibration motor mounted on the handlebars, could improve rider awareness. This haptic feedback would provide a non-auditory alert, which is particularly useful in noisy environments. Replacing the buzzer with a more natural bicycle bell sound would also help with pattern recognition, as it is a widely recognized and intuitive sound.

References

[1] TI. "TPS6306x High Input Voltage, Buck-Boost Converter With 2-A Switch Current," Accessed: Feb. 18, 2026. [Online]. Available: https://www.ti.com/lit/ds/symlink/tps63060.pdf.

[2] DigiKey. "AMS1117 1A LDO Voltage Regulator Datasheet," Accessed: Feb. 18, 2026. [Online]. Available: https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/5011/AMS1117.pdf.

[3] Adafruit. "Polymer Li-ion Battery Technology Specification," Accessed: Feb. 10, 2026. [Online]. Available: https://cdn-shop.adafruit.com/product-files/1781/C2253_-_ICR18650_2200mAh_3.7V_with_PCM_20140728_APPROVED_8.18.pdf.

[4] Microchip. "Miniature Single-Cell, Fully Integrated Li-Ion, Li-Polymer Charge Management Controllers," Accessed: Mar. 10, 2026. [Online]. Available: https://ww1.microchip.com/downloads/aemDocuments/documents/APID/ProductDocuments/DataSheets/MCP73831-Family-Data-Sheet-DS20001984H.pdf.

[5] S. Sky. "UJ20-C-H-G-MSMT-FT-P16-TR-67," Accessed: Mar. 8, 2026. [Online]. Available: https://www.sameskydevices.com/product/resource/uj20-c-h-g-msmt-ft-p16-tr-67.pdf.

[6] P. Semi. "UJ20-C-H-G-MSMT-FT-P16-TR-67," Accessed: Mar. 24, 2026. [Online]. Available: http://www.pingjingsemi.com/UploadFile/pdf/DW01A.pdf.

[7] P. Minarik. "8 Best Bike Bells Actually Bought, Tested, and Ranked," Cyclists Hub, Accessed: Feb. 10, 2026. [Online]. Available: https://www.cyclistshub.com/best-bike-bells.

[8] G. E. O. of Marketing and Communications. "Salary Averages," Accessed: Feb. 8, 2026. [Online]. Available: https://ece.illinois.edu/admissions/why-ece/salary-averages.

[9] IEEE. "IEEE Code of Ethics," Accessed: Feb. 12, 2026. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html.

Appendix A PCB Design

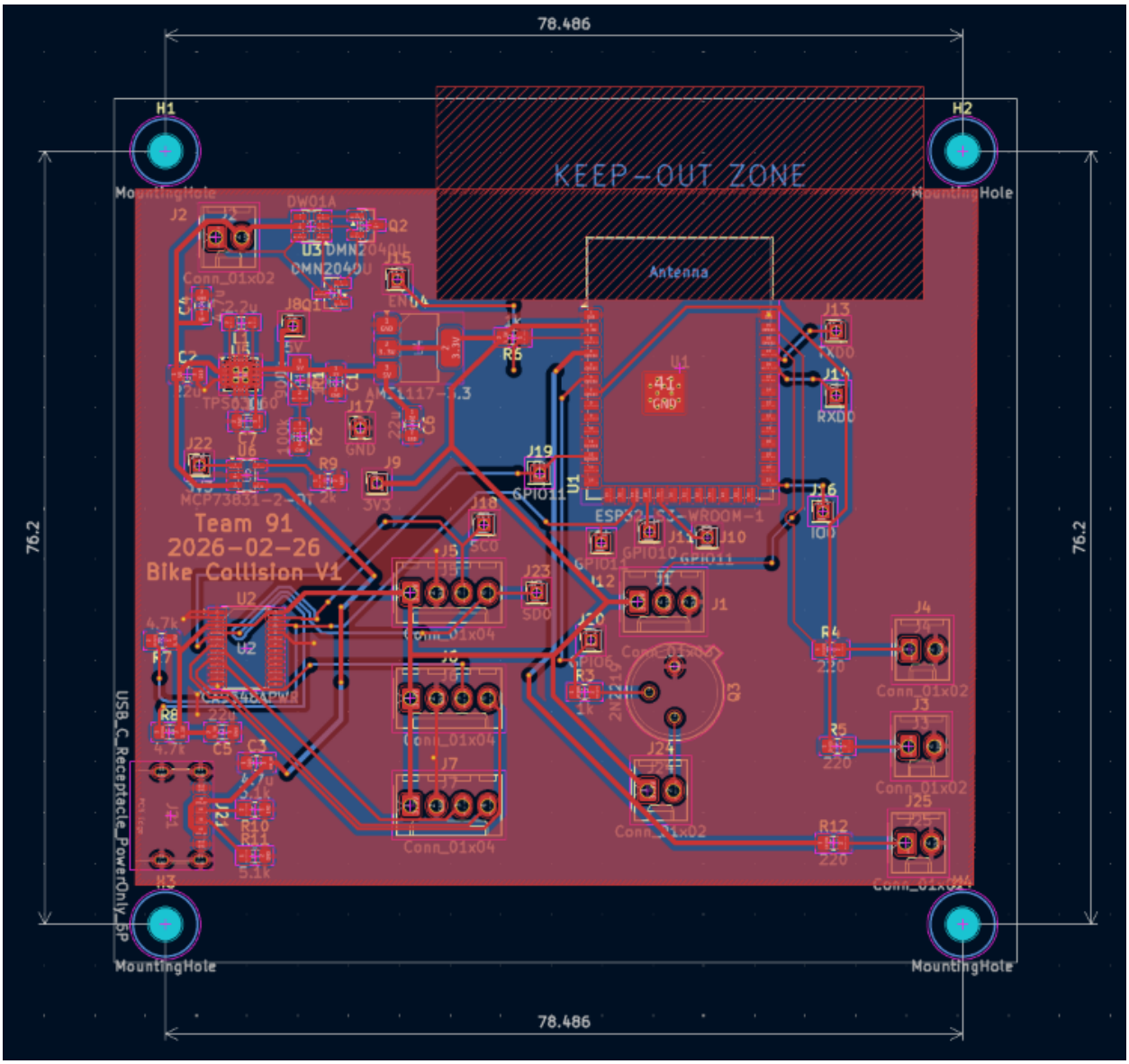

Figure A.1: PCB Design

Appendix B RV Tables

Table B.1: Power Requirements and Verification

Requirement	**Verification**	**Status**
The power subsystem shall provide a stable 3.3V $\pm$ 0.1V rail to the ESP32 and logic-level components under expected load.	Power the board using the 18650 battery or bench supply. Measure the 3.3V rail with a DMM under no-load and full-load conditions. Verify voltage remains within 3.2V–3.4V.	Verified
The buck-boost converter shall provide a stable 5V $\pm$ 0.25V rail across the expected battery range.	Use a current-limited bench supply to emulate battery voltages from approximately 3.0V to 4.2V. Measure the 5V rail under load and verify it remains within tolerance.	Unverified
The DW01A protection circuit shall disconnect the battery output when cell voltage drops below the over-discharge threshold.	Connect a variable DC supply to the battery terminals and slowly decrease voltage while monitoring load voltage. Record the cutoff voltage where the system output disconnects.	Unverified

Table B.2: Radar Requirements and Verification

Requirement	**Verification**	**Status**
Each SEN0610 sensor must accurately detect moving human targets at a distance of 3.5 m $\pm$ 0.5 m.	Place the sensors on a flat surface, measure out 3.5 meters, and check for a reading.	Verified
Each sensor has a false positive rate of less than 15%.	Over the course of 60 seconds, measure how many times a detection has been falsely recorded.	Unverified

Table B.3: Microcontroller Requirements and Verification

Requirement	**Verification**	**Status**
The ESP32-S3 must complete one full cycle of the Time-To-Collision (TTC) algorithm for all sensors in $\leq$ 50 ms.	Insert "timestamp" code at the start and end of the TTC main loop. Run the system and print the delta to the Serial Monitor. Verify the loop time stays consistently below 50 ms over a 5-minute test period.	Verified
The ESP32-S3 shall successfully communicate with each radar sensor through I2C.	Verify all connected sensors return readable data without bus conflicts.	Verified
The ESP32-S3 shall generate distinct buzzer control patterns for Low, Medium, and High risk states.	Trigger each risk state manually or through simulated sensor input. Observe to verify different beeping intervals for each state.	Verified

Table B.4: User I/O Requirements and Verification

Requirement	**Verification**	**Status**
Potentiometer is able to change distance thresholds by a linear amount.	Print threshold values to serial and turn the potentiometer.	Verified
Button is able to turn system on/off.	Make the system detect a presence within a distance threshold to create noise. Press the button to turn off the system and verify the buzzer stops.	Verified
The system must vary the buzzer's beeping frequency to provide at least three distinct auditory patterns corresponding to "Low," "Medium," and "High" risk levels (e.g., 0.8, 2, and 6 beeps per second).	Trigger each risk level and measure/observe buzzer timing. Verify Low, Medium, and High alerts are distinguishable (slow beeps, fast beeps, continuous tone).	Verified
LEDs show power state.	Power on system and verify LEDs match power state.	Verified

Appendix C Microcontroller Testing

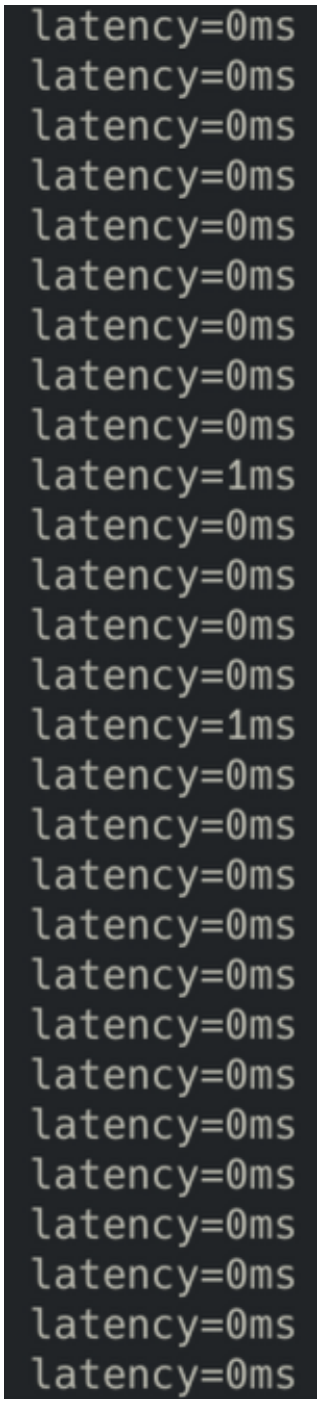

Figure C.1: Algorithm Latency