ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

Snooze-Cruiser: An Autonomous Robotic Alarm System

Team #46

JIZHEN CHEN (jizhenc2@illinois.edu)
ALEX WANG (zw71@illinois.edu)
JIACHEN HU (hu86@illinois.edu)

TA: Jiaming Xu

May 7, 2026

Abstract

Snooze-Cruiser can be described as an autonomous robotic alarm system that ensures user wakefulness, as physical interaction is required to switch it off. The system consists of a differential-drive locomotion mechanism, obstacle detection sensors, and microcontroller-based control, allowing the robot to move within an enclosed space without collisions. When activated, the robot moves in an unpredictable manner while producing an alarm sound until it is physically restrained by the user. The most crucial subsystems include motor control, sensing, and real-time decision-making. Tests conducted show that the Snooze-Cruiser effectively meets the requirements for movement, obstacle avoidance, and user interaction.

Contents

1 Introduction

Many students and working adults rely on alarms to wake up on time for classes, work, meetings, and other daily responsibilities. However, traditional alarm clocks and smartphone alarms often fail to fully wake the user. One major reason is sleep inertia, the temporary period of reduced alertness and impaired decision-making that occurs immediately after waking. During this state, a person may silence an alarm without becoming fully conscious, then return to sleep and oversleep. This problem can lead to missed obligations, reduced productivity, academic or professional consequences, and unnecessary stress.

Most common alarm systems depend almost entirely on sound. Although loud audio can interrupt sleep, it does not necessarily require the user to become physically active. As a result, the user can often stop the alarm while still lying in bed. Some alternative alarm products attempt to solve this problem by requiring puzzle solving or by moving away from the user, but these approaches can still be unreliable, unsafe, or unsuitable for confined indoor spaces. A more effective alarm system should require meaningful physical engagement while also operating safely around furniture, walls, and people.

Snooze-Cruiser was designed to address this problem by combining a conventional alarm with a small autonomous mobile robot. At the programmed alarm time, the system activates both audio output and robot motion. Instead of allowing the user to simply press a button from bed, the robot moves through the room and requires the user to physically interact with it before the alarm stops. This motion-based interaction is intended to increase physical activity and awareness, making it less likely that the user will immediately fall back asleep.

The final Snooze-Cruiser prototype consists of a microcontroller-based control system, a two-wheel mobile chassis, a motor driver and DC motors for movement, a laser distance sensor for obstacle detection, an inertial sensor for user-interaction detection, an OLED display and buttons for user input, an audio output subsystem, and a battery-powered regulation system. The microcontroller coordinates alarm timing, motion control, sensing, user interface behavior, and shutdown logic. During operation, the robot moves autonomously, avoids obstacles in its path, displays relevant alarm information, and produces audible alarm output.

During development, the design evolved from the original proposal. The initial design planned to use multiple distance sensors and lift-based pickup detection. In the final implementation, the team used a single laser sensor because the multiple-sensor configuration introduced tuning difficulty and measurement inconsistency. The stop condition was also changed from lift detection to roll-over detection because it provided more reliable user-interaction detection during testing. These changes improved system stability while preserving the main project objective: the alarm should only stop after deliberate physical interaction from the user.

The completed system was evaluated according to the project's major requirements: accurate alarm activation, synchronized motion and audio output, autonomous movement, obstacle avoidance, reliable user-interaction detection, and safe operation in a confined indoor environment. Final testing showed that the robot could count time accurately, accept user input through the interface, move continuously, avoid obstacles, respond to user interaction, and operate for more than 48 hours on battery power. Therefore, Snooze-Cruiser successfully demonstrates a practical robotic alarm system that improves upon traditional alarms by requiring active physical engagement while maintaining safe indoor operation.

2 Design

2.1 System Overview

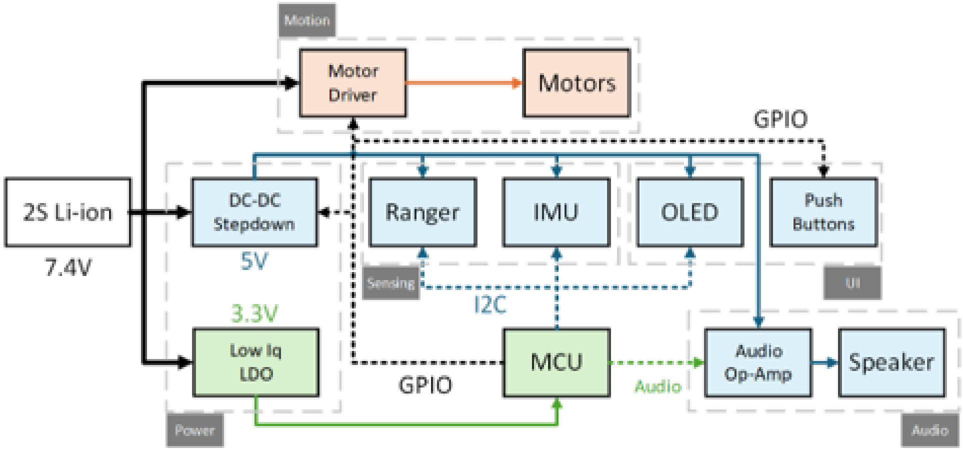


Figure 1: Overall system architecture of the Snooze-Cruiser.

Snooze-Cruiser is organized as a microcontroller-centered robotic alarm system. The final system integrates time counting, user input, display output, motor control, obstacle sensing, inertial user-interaction detection, audio output, and battery-powered operation. As shown in Figure 1, the system is centered around the MCU, which coordinates sensing, motion control, and user interaction. The main purpose of the design is to activate an alarm at the programmed time and force the user to physically interact with the device before the alarm can be disabled.

The system is controlled by an STM32F446RCT6 microcontroller unit (MCU). The MCU coordinates all major subsystems, including the motion subsystem, sensing subsystem, user interface, audio output, and power system. During normal operation, the user sets the alarm time through push buttons, and the current system state is shown on the organic light-emitting diode (OLED) display. When the programmed alarm time is reached, the MCU activates both the audio subsystem and the motion subsystem. The robot then begins moving in the indoor environment while using sensor feedback to detect nearby obstacles. The alarm stops only after the user physically interacts with the robot and triggers the inertial stop condition.

The original design separated the system into six major blocks: motion, sensing, inertial measurement, user interface, audio, and power. The motion block used a motor driver and two direct-current (DC) motors to provide differential-drive motion. The sensing block originally used multiple distance sensors to provide front obstacle detection. The inertial measurement unit (IMU) was used to detect when the user picked up the robot. The user interface consisted of an OLED display and push buttons. The audio block generated the alarm sound, and the power block provided regulated voltage rails from a rechargeable battery source.

Several changes were made during implementation and testing. The original design planned to use three distance sensors to provide wider obstacle-detection coverage. In the final prototype, the system used a single laser distance sensor because the multiple-sensor configuration was difficult to tune and produced inconsistent measurements during integration. The original stop condition was based on lift detection using the IMU. In the final implementation, the stop condition was changed to roll-over detection because it produced a clearer and more reliable inertial signature. The power subsystem also changed during integration. After the original power regulation circuit failed, an external regulator board was used to provide stable voltage rails for the prototype. These changes simplified the final implementation while preserving the main high-level requirements of timed alarm activation, autonomous motion, obstacle response, and physical user interaction.

2.2 Microcontroller and Control Logic

The STM32F446RCT6 microcontroller serves as the central controller for Snooze-Cruiser. This device was selected because it provides sufficient general-purpose input/output (GPIO) pins, timer resources, pulse-width modulation (PWM) capability, and serial communication interfaces for the system. These features allow one controller to manage the display, buttons, motor driver, distance sensor, inertial sensor, and audio output.

The MCU performs four main control tasks. First, it maintains the time-counting logic needed to compare the current time with the user-programmed alarm time. Second, it reads push-button inputs and updates the OLED display so that the user can configure and monitor the alarm. Third, it controls the motor driver using GPIO and PWM signals to produce forward motion and turning behavior. Fourth, it reads sensor data from the laser sensor and inertial sensor to decide when to avoid an obstacle or stop the alarm.

The firmware can be described as a state-based control system. In the idle state, the device displays the current time and waits for user input. In the alarm-setting state, button inputs are used to adjust the programmed alarm time. In the waiting state, the MCU continues counting time until the current time reaches the alarm time. In the active alarm state, the MCU enables both motion and audio output. While the alarm is active, the MCU continuously monitors the laser sensor for obstacle detection and the inertial sensor for the stop condition. Once roll-over detection is triggered, the MCU disables the motors and audio output and returns the system to a stopped or reset state.

This state-based approach was chosen because it separates the major operating modes of the device and reduces the chance of conflicting control actions. For example, button inputs are used for alarm setting before activation, but they are not used as the primary alarm-disable method during active operation. This design choice helps ensure that the user must physically interact with the robot rather than simply pressing a button to silence the alarm.

2.3 Time Counting and User Interface Subsystem

The time counting and user interface subsystem allows the user to set and monitor the alarm. It consists of the internal timing logic of the MCU, an OLED display, and push buttons. The OLED display communicates with the MCU and shows information such as the current time, alarm time, and system state. The push buttons are connected to GPIO pins and allow the user to adjust the alarm time.

The time-counting function is implemented in firmware using the MCU timing resources. The alarm time is stored as a programmed target value. During operation, the MCU compares the current time count with the alarm setting. When the programmed time is reached, the MCU changes the system state from waiting mode to active alarm mode. In active alarm mode, the MCU starts the motion and audio subsystems.

The push-button interface was designed to be simple and reliable. Mechanical buttons can produce several fast transitions when pressed or released, which may cause false input if the signal is read directly. To reduce this problem, the firmware debounces the button input before accepting it as a valid command. This ensures that a single physical press corresponds to one intended input action.

The OLED display provides immediate feedback to the user. Without a display, it would be difficult for the user to confirm the current alarm setting. The display also makes testing easier because the system state and time-counting behavior can be observed directly. This subsystem was therefore important for both final device usability and debugging during development.

2.4 Motion Subsystem

The motion subsystem allows Snooze-Cruiser to move away from the user after the alarm activates. The robot uses a two-wheel differential-drive chassis with a passive caster for stability. The two drive wheels are powered by DC gear motors. By controlling the relative speed and direction of the left and right motors, the robot can move forward, turn left, turn right, or stop.

The MCU cannot directly drive the motors because the motors require more current than a microcontroller pin can safely supply. Therefore, a motor driver is used as the interface between the low-power MCU control signals and the higher-power motor load. The motor driver receives direction and PWM control signals from the MCU and switches the motor current accordingly. PWM control allows the effective motor speed to be adjusted by changing the duty cycle of the signal.

A key design concern for the motion subsystem is the motor stall current. When a DC motor is prevented from rotating, the motor current can rise to its stall-current value. The approximate stall current can be estimated using Ohm's law:

$$I_{\text{stall}} = \frac{V}{R_m} \tag{1}$$

where I_{stall} is the stall current, V is the applied motor voltage, and R_m is the motor winding resistance. For example, if a motor is driven at 5 V and has a winding resistance of 2.5 Ω, the estimated stall current is

$$I_{\text{stall}} = \frac{5 \text{ V}}{2.5 \ \Omega} = 2 \text{ A}. \tag{2}$$

This calculation shows why the motor driver must be selected with sufficient current capacity and why the MCU cannot directly power the motors.

The motion subsystem was designed for moderate indoor movement rather than high speed. Since the device operates in a bedroom, dorm room, or similar confined space, high speed would create unnecessary collision risk. The firmware therefore limits the motor behavior to controlled movement suitable for indoor testing. This supports the project goal of forcing user engagement while maintaining safe operation.

2.5 Obstacle Detection and Navigation Subsystem

The obstacle detection subsystem allows the robot to respond to objects in its path while moving. The original design used three Time-of-Flight or laser distance sensors placed at the front-left, front-center, and front-right positions of the chassis. This arrangement was intended to provide broad forward coverage for obstacle avoidance. In theory, the three-sensor design would allow the MCU to determine whether an obstacle was located to the left, center, or right of the robot and then choose an appropriate turning direction.

During implementation, the multiple-sensor configuration created practical tuning difficulties. The sensors produced inconsistent readings, and the added complexity made the obstacle-avoidance behavior less stable. As a result, the final prototype used a single laser distance sensor. This reduced the complexity of the sensing subsystem and improved repeatability during testing. Although the final single-sensor implementation provides less directional coverage than the original three-sensor plan, it was sufficient for detecting forward obstacles in the final demonstration environment.

The navigation logic is based on a distance threshold. While the alarm is active, the MCU repeatedly reads the laser sensor. If the measured distance is greater than the programmed threshold, the robot continues normal motion. If the measured distance is less than the threshold, the MCU interprets this as an obstacle in the robot's path and adjusts the motor behavior. The adjustment can include stopping briefly, turning, or changing direction before continuing movement.

The relationship between sensor delay, robot speed, and safe stopping margin was considered during design. If the robot moves at velocity v and the control loop has delay t_d, then the robot continues moving for a distance

$$d = vt_d \tag{3}$$

before the controller can respond. For example, if the robot moves at 0.3 m/s and the control delay is 0.05 s, the delay distance is

$$d = (0.3 \text{ m/s})(0.05 \text{ s}) = 0.015 \text{ m} = 1.5 \text{ cm}. \tag{4}$$

Therefore, the obstacle threshold must include enough margin to account for sensor tolerance, control delay, and motor response time. A larger threshold improves safety but may also cause the robot to react earlier than necessary. The final threshold was selected to provide reliable obstacle response in the indoor demonstration setting.

2.6 User-Interaction Detection Subsystem

The user-interaction detection subsystem determines when the alarm should stop. This subsystem is important because the main purpose of Snooze-Cruiser is to prevent passive alarm dismissal. A traditional alarm can be disabled by pressing a button, but Snooze-Cruiser is designed so that the active alarm stops only after deliberate physical interaction with the robot.

The original design used an IMU-based lift detection method. The IMU measures acceleration and angular motion, which can be used to infer when the robot is lifted from the floor. The intended approach was to detect a sudden change in vertical acceleration or orientation when the user picked up the robot. Once this condition was detected, the MCU would stop the motors and audio output.

During testing, lift detection was found to be less reliable than expected. Normal driving vibration, surface irregularities, and small disturbances could produce inertial readings that were difficult to distinguish from a true lift event. This created a risk of false triggering or missed detection. To improve reliability, the final design changed the stop condition to roll-over detection.

Roll-over detection uses the inertial sensor to detect a clear orientation change when the user rolls or flips the robot. This action produces a more distinct sensor signature than simply lifting the robot. In the final implementation, the alarm continues operating until the roll-over condition is detected. Once the condition is detected, the MCU disables the motor outputs and stops the audio output. This final design still satisfies the main intent of the stop requirement because the user must physically reach the robot and manipulate it before the alarm stops.

2.7 Audio Subsystem

The audio subsystem provides the alarm sound that wakes the user. The MCU controls the alarm output and enables the audio signal when the programmed alarm time is reached. The sound output operates at the same time as the motion subsystem, so the device functions as both a conventional audible alarm and a moving robotic alarm.

The original design used an audio amplifier and speaker so that the MCU would not need to directly drive the speaker load. This is necessary because a speaker requires more current and power than a microcontroller output pin can provide. The amplifier increases the signal power and drives the speaker at a usable volume. In the final system behavior, the audio output remains active while the robot is moving and stops only after the inertial user-interaction condition is detected.

The audio subsystem is coordinated with the motion subsystem through firmware. When the alarm state becomes active, the MCU enables both the alarm sound and motor control. When the stop condition is detected, the MCU disables both outputs. This synchronized behavior is important because the device should not continue moving silently after the alarm has been stopped, and it should not continue producing sound after the user has completed the required physical interaction.

2.8 Power Subsystem

The power subsystem supplies the voltages required by the MCU, sensors, display, motor driver, and audio subsystem. The original design used a rechargeable 2S lithium-ion battery pack as the main energy source. A 2S lithium-ion pack provides approximately 7.4 V nominally. Since not all components can operate directly from this voltage, regulation is required to produce lower-voltage rails such as 5 V and 3.3 V.

The 3.3 V rail is required for low-power digital components such as the MCU, sensors, and OLED display. The 5 V rail or battery-level supply is used for higher-power components such as motors and audio circuitry, depending on the specific module requirements. Separating the power domains helps protect the low-voltage logic from the higher current demands and electrical noise produced by the motors.

During integration, the original power regulation circuit failed. This was one of the main hardware challenges encountered during the project. To continue development and complete the final prototype, the team replaced the failed regulation stage with an external regulator board. This external board provided stable supply voltages for the rest of the system. Although this change reduced the level of integration compared with the original PCB design, it preserved the intended function of the power subsystem and allowed the final prototype to operate reliably.

Battery operation was an important design requirement because the robot must move freely without being connected to a wired power supply. The final prototype demonstrated sufficient battery life for practical alarm operation, with testing showing operation for more than 48 hours. This result indicates that the power subsystem provided enough energy capacity for the time-counting, display, sensing, motion, and alarm functions during the tested operating conditions.

2.9 Physical Integration

The final prototype was built on a small two-wheel robotic car chassis. The chassis provides the mechanical base for the motors, wheels, caster, battery, control electronics, sensors, display, and wiring. The use of a purchased chassis reduced mechanical construction time and allowed the team to focus on electrical integration, embedded control, and system testing.

The control electronics and modules were mounted on or near the chassis so that the robot could move as a self-contained device. The OLED display and buttons were placed where the user could access them for alarm setup. The distance sensor was positioned toward the front of the robot so that it could detect obstacles in the direction of travel. The inertial sensor was mounted so that orientation changes could be detected during roll-over interaction. The battery and power regulation hardware were secured to reduce movement during operation.

Hardware integration required several adjustments during development. Some modules were replaced, and wiring was modified to improve stability. These changes were necessary because the final prototype had to operate as a complete moving system, not only as separate breadboard-tested subsystems. The final physical design successfully integrated motion, sensing, alarm output, time counting, display, button control, and battery power into a working robotic alarm prototype.

The completed Snooze-Cruiser prototype is shown in Figure 2, illustrating the integration of all subsystems into a single functional device.

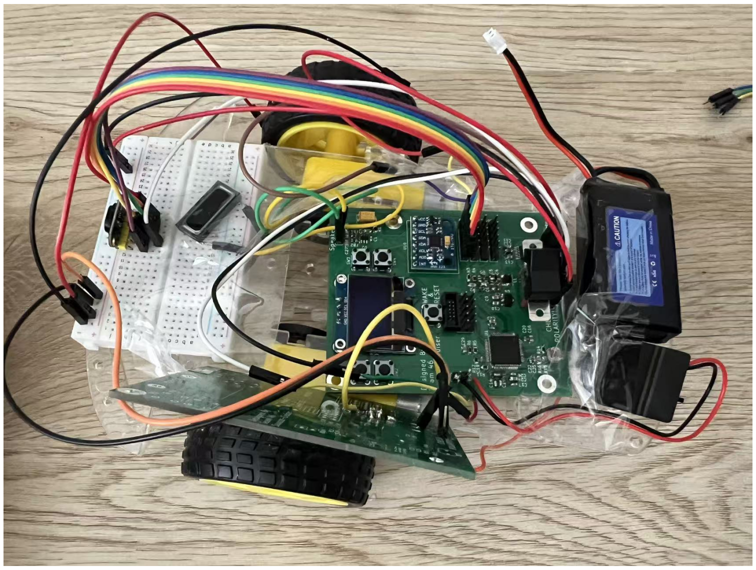


Figure 2: Final Snooze-Cruiser prototype showing integrated hardware components.

2.10 Design Summary

The final Snooze-Cruiser design combines an audible alarm with autonomous robotic motion to require physical user interaction. The MCU coordinates time counting, user input, display output, motor control, obstacle sensing, inertial stop detection, and audio output. The motion subsystem allows the robot to move through an indoor environment, while the laser distance sensor provides forward obstacle detection. The inertial stop condition ensures that the alarm is disabled only after the user physically rolls over the robot. The power subsystem supports untethered operation using a rechargeable battery and regulated voltage rails.

The final implementation differed from the original design in several important ways. The obstacle detection subsystem was simplified from multiple sensors to one laser sensor, the stop condition was changed from lift detection to roll-over detection, and the failed onboard power regulation was replaced by an external regulator board. These changes were made in response to integration and testing challenges. The resulting prototype preserved the main project goal and produced a more stable final demonstration system.

3 Verification

3.1 Testing Overview

The Snooze-Cruiser was tested in a controlled indoor environment to evaluate system performance across all major subsystems, including motion, sensing, timing, and power. A combination of stopwatch measurements, video recordings, and instrument-based measurements (oscilloscope, multimeter, and power supply) were used to verify system behavior. Each requirement was tested multiple times to ensure repeatability and consistency of results.

3.2 Key Results

The system successfully met or exceeded all major functional requirements. A summary of key verification results is shown in Table 1.

Subsystem	**Requirement**	**Result**
Motion Subsystem	Continuous operation $>$ 5 minutes	Operated for over 20 minutes
Obstacle Avoidance	Minimal collisions in indoor environment	Negligible collisions observed during testing
IMU / Stop Condition	Response time $<$ 5 seconds	Response time $\approx 3 \pm 0.75$ seconds
Time Counting	Accuracy $<$ 2 seconds	Error within 0.1 seconds
Battery Life	Operation duration $>$ 12 hours	Operated for over 48 hours

Table 1: Summary of System Verification Results

3.3 Discussion of Requirements

The motion subsystem demonstrated reliable operation, exceeding the required continuous runtime by operating for over 20 minutes without interruption. Obstacle avoidance performance was validated through repeated trials in indoor environments, with minimal collisions observed, indicating effective sensor integration and control logic.

The stop condition, implemented using IMU-based detection, consistently responded within the required time frame, with an average response time of approximately 3 seconds. Timing accuracy was also verified, with measured error well below the specified 2-second threshold.

Power system performance exceeded expectations, with the device operating for over 48 hours under normal conditions, significantly surpassing the minimum requirement of 12 hours. Voltage regulation and power stability were also verified across operating conditions.

3.4 Summary

Overall, the Snooze-Cruiser met all specified requirements across motion, sensing, timing, and power subsystems. The system demonstrated stable and reliable operation under repeated testing conditions.

The complete Requirement and Verification (R&V) table is provided in Appendix A.

4 Cost Analysis

4.1 Labor Cost

The labor cost is estimated assuming each team member is an entry-level Electrical and Computer Engineering graduate earning \$42 per hour. Each member contributed approximately 120 hours to the project. The total labor cost is calculated using the standard formula:

$$\text{Cost} = (\text{Hourly Rate}) \times (\text{Hours}) \times 2.5 \tag{5}$$

The factor of 2.5 accounts for overhead and additional costs associated with engineering labor.

The cost per team member is:

$$42 \times 120 \times 2.5 = 12{,}600 \tag{6}$$

For three team members, the total labor cost is:

$$3 \times 12{,}600 = 37{,}800 \tag{7}$$

4.2 Parts Cost

The components used in the Snooze-Cruiser are categorized according to system functionality.

4.2.1 Central MCU Subsystem

- STM32F446RCT6 microcontroller — \$7
- Crystal oscillator — \$10
- SMD resistors and capacitors — \$10

4.2.2 Motion Subsystem

- N20 DC gear motors (2 pcs) — \$25
- Motor driver DRV8833 — \$2.61
- Robot car chassis kit — \$13

4.2.3 Sensor Subsystem

- Time-of-Flight sensor VL53L1X — \$29.95

4.2.4 Audio Subsystem

- Audio amplifier PAM8301AAF — \$0.42
- Speaker — \$5

4.2.5 Power Subsystem

- Li-ion battery pack — \$15
- Battery charger MCP73844 — \$1.76
- Power regulation components — \$10

4.2.6 Supporting Components

- Wiring, connectors, headers, terminal blocks — \$10
- Switches and buttons for UI — \$5
- LEDs and indicator components — \$3
- Mechanical fasteners, standoffs, mounting hardware — \$5
- Miscellaneous prototyping supplies — \$10

4.2.7 PCB and Manufacturing

- Custom PCB fabrication — \$40
- Assembly materials — \$10

$$\text{Total Parts Cost} = 229.74 \tag{8}$$

4.3 Total Project Cost

The total project cost is the sum of labor and parts costs:

$$\text{Total Cost} = 37{,}800 + 229.74 = 38{,}029.74 \tag{9}$$

5 Safety, Ethics, and Broader Impact

The Snooze-Cruiser is developed considering relevant engineering guidelines to ensure safety, effectiveness, and reliable operation. In particular, its function is based on a low-voltage battery, reducing the risk of electrical hazards. Additional safeguards include current limiting and regulated power supply. Mechanical safety is ensured through limitations on motion, including speed limits, obstacle detection, and automatic motor deactivation when the device is lifted.

The device is intended for indoor operation. Potential hazards, such as use near stairs or elevated surfaces, are mitigated through firmware constraints and user guidelines.

From an ethical perspective, the project complies with the IEEE Code of Ethics by prioritizing the safety, health, and welfare of the public, and by emphasizing honesty and integrity in engineering practice [1]. Similarly, the ACM Code of Ethics highlights the responsibility to avoid harm and consider the societal impacts of technology [2]. In this context, the Snooze-Cruiser does not include cameras or microphones, thereby protecting user privacy, and all testing results are reported truthfully.

In terms of broader impact, the device addresses the limitations of conventional alarm systems by requiring physical interaction to deactivate the alarm, which may improve wakefulness and daily productivity. The system is constructed using relatively low-cost components, making it suitable for commercial production. Additionally, the use of rechargeable batteries reduces environmental waste.

The device is designed to comply with FCC regulations for unintentional radiators under 47 CFR Part 15 [3].

Overall, the project incorporates appropriate safety considerations and adheres to established ethical and engineering standards.

6 Conclusion

The Snooze-Cruiser project successfully demonstrates the design and implementation of an autonomous robotic alarm system that requires physical user interaction to deactivate. The system integrates multiple subsystems, including motion control, obstacle detection, timing logic, audio output, and user interface components, into a cohesive and functional embedded system.

Through testing and verification, the system met all major functional requirements. Reliable time counting and accurate user input handling were achieved, ensuring consistent alarm behavior. The motion subsystem demonstrated stable and continuous operation, while the obstacle avoidance system effectively minimized collisions in typical indoor environments. In addition, the pickup detection mechanism responded within the required time constraints, and the system exceeded expectations in battery performance, operating significantly longer than the specified minimum duration.

During development, several engineering challenges were encountered. Power regulation issues required redesign of the power delivery approach, and initial IMU-based detection proved unreliable, leading to the adoption of a more robust flip-based detection method. Sensor integration and hardware stability also required iterative refinement. These challenges highlight the importance of practical debugging, system-level thinking, and iterative design in embedded systems development.

Feedback from the demonstration phase was incorporated by refining hardware integration and improving system stability.

Overall, the Snooze-Cruiser fulfills its intended purpose as an interactive alarm system while demonstrating key engineering principles such as modular design, system integration, and real-time decision making. The project reflects a complete engineering workflow from initial design to final implementation and validation, and emphasizes the importance of teamwork and iterative problem-solving in achieving a reliable and functional system.

Future improvements may include enhanced localization capabilities for more intelligent navigation, improved energy efficiency to further extend battery life, and more advanced user interaction features to increase usability and customization.

References

[1] IEEE. “IEEE Code of Ethics.” IEEE, 2020. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html.

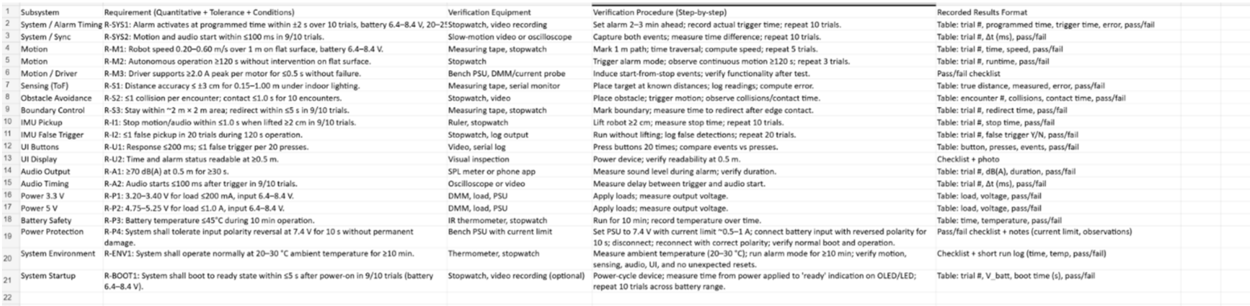
[2] Association for Computing Machinery. “ACM Code of Ethics and Professional Conduct.” ACM, 2018. [Online]. Available: https://www.acm.org/code-of-ethics.

[3] Federal Communications Commission. “47 CFR Part 15 — Radio Frequency Devices.” Code of Federal Regulations, 2023. [Online]. Available: https://www.ecfr.gov/current/title-47/part-15.

Appendix A Appendix

A.1 Requirement and Verification Table

The complete Requirement and Verification (R&V) table is shown in Figure 3.



1	Subsystem	Requirement (Quantitative + Tolerance + Conditions)	Verification Equipment	Verification Procedure (Step-by-step)	Recorded Results Format
2	System / Alarm Timing	R-SYS1: Alarm activates at programmed time within ±2 s over 10 trials, battery 6.4–8.4 V, 20–2	Stopwatch, video recording	Set alarm 2–3 min ahead; record actual trigger time; repeat 10 trials.	Table: trial #, programmed time, trigger time, error, pass/fail
3	System / Sync	R-SYS2: Motion and audio start within ≤100 ms in 9/10 trials.	Slow-motion video or oscilloscope	Capture both events; measure time difference; repeat 10 trials.	Table: trial #, Δt (ms), pass/fail
4	Motion	R-M1: Robot speed 0.20–0.60 m/s over 1 m on flat surface, battery 6.4–8.4 V.	Measuring tape, stopwatch	Mark 1 m path; time traversal; compute speed; repeat 5 trials.	Table: trial #, time, speed, pass/fail
5	Motion	R-M2: Autonomous operation ≥120 s without intervention on flat surface.	Stopwatch	Trigger alarm mode; observe continuous motion ≥120 s; repeat 3 trials.	Table: trial #, runtime, pass/fail
6	Motion / Driver	R-M3: Driver supports ≥2.0 A peak per motor for ≤0.5 s without failure.	Bench PSU, DMM/current probe	Induce start-from-stop events; verify functionality after test.	Pass/fail checklist
7	Sensing (ToF)	R-S1: Distance accuracy ≤ ±3 cm for 0.15–1.00 m under indoor lighting.	Measuring tape, serial monitor	Place target at known distances; log readings; compute error.	Table: true distance, measured, error, pass/fail
8	Obstacle Avoidance	R-S2: ≤1 collision per encounter; contact ≤1.0 s for 10 encounters.	Stopwatch, video	Place obstacle; trigger motion; observe collisions/contact time.	Table: encounter #, collisions, contact time, pass/fail
9	Boundary Control	R-S3: Stay within ~2 m × 2 m area; redirect within ≤5 s in 9/10 trials.	Measuring tape, stopwatch	Mark boundary; measure time to redirect after edge contact.	Table: trial #, redirect time, pass/fail
10	IMU Pickup	R-I1: Stop motion/audio within ≤1.0 s when lifted ≥2 cm in 9/10 trials.	Ruler, stopwatch	Lift robot ≥2 cm; measure stop time; repeat 10 trials.	Table: trial #, stop time, pass/fail
11	IMU False Trigger	R-I2: ≤1 false pickup in 20 trials during 120 s operation.	Stopwatch, log output	Run without lifting; log false detections; repeat 20 trials.	Table: trial #, false trigger Y/N, pass/fail
12	UI Buttons	R-U1: Response ≤200 ms; ≤1 false trigger per 20 presses.	Video, serial log	Press buttons 20 times; compare events vs presses.	Table: button, presses, events, pass/fail
13	UI Display	R-U2: Time and alarm status readable at ≥0.5 m.	Visual inspection	Power device; verify readability at 0.5 m.	Checklist + photo
14	Audio Output	R-A1: ≥70 dB(A) at 0.5 m for ≥30 s.	SPL meter or phone app	Measure sound level during alarm; verify duration.	Table: trial #, dB(A), duration, pass/fail
15	Audio Timing	R-A2: Audio starts ≤100 ms after trigger in 9/10 trials.	Oscilloscope or video	Measure delay between trigger and audio start.	Table: trial #, Δt (ms), pass/fail
16	Power 3.3 V	R-P1: 3.20–3.40 V for load ≤200 mA, input 6.4–8.4 V.	DMM, load, PSU	Apply loads; measure output voltage.	Table: load, voltage, pass/fail
17	Power 5 V	R-P2: 4.75–5.25 V for load ≤1.0 A, input 6.4–8.4 V.	DMM, load, PSU	Apply loads; measure output voltage.	Table: load, voltage, pass/fail
18	Battery Safety	R-P3: Battery temperature ≤45°C during 10 min operation.	IR thermometer, stopwatch	Run for 10 min; record temperature over time.	Table: time, temperature, pass/fail
19	Power Protection	R-P4: System shall tolerate input polarity reversal at 7.4 V for 10 s without permanent damage.	Bench PSU with current limit	Set PSU to 7.4 V with current limit ~0.5–1 A; connect battery input with reversed polarity for 10 s; disconnect; reconnect with correct polarity; verify normal boot and operation.	Pass/fail checklist + notes (current limit, observations)
20	System Environment	R-ENV1: System shall operate normally at 20–30 °C ambient temperature for ≥10 min.	Thermometer, stopwatch	Measure ambient temperature (20–30 °C); run alarm mode for ≥10 min; verify motion, sensing, audio, UI, and no unexpected resets.	Checklist + short run log (time, temp, pass/fail)
21	System Startup	R-BOOT1: System shall boot to ready state within ≤5 s after power-on in 9/10 trials (battery 6.4–8.4 V).	Stopwatch, video recording (optional)	Power-cycle device; measure time from power applied to 'ready' indication on OLED/LED; repeat 10 trials across battery range.	Table: trial #, V_batt, boot time (s), pass/fail
22					

Figure 3: Requirement and Verification Table

A.2 PWM Signal Verification

Motor control in the Snooze-Cruiser is implemented using pulse-width modulation (PWM) signals generated by the microcontroller. The PWM signals control motor speed by varying the duty cycle.

Figure 4: Oscilloscope measurement of PWM signals used for motor control.