Team 34 Final Report

University of Illinois Urbana-Champaign

ECE445 - Spring 2025

Board Buddy

TA: Surya Vasanth

Professor: Michael Oelze

Team Members:

Alfredo Angel (alfredo9)

Gabe Valles (gvall4)

Louie Conn (lewisc2)

Abstract

The subsequent report will document the ECE445 Final Project for Team 34 per the University of Illinois Electrical and Computer Engineering curriculum. This report will be a cumulative summary of all hardware, software, and firmware aspects of Team 34's Final Project, the Board Buddy. The following will document all related analyses that were required to complete this project.
**Reference Appendix A for any unknown abbreviations and symbols.

Contents

1. Introduction

1.1 Problem

Teachers too often waste precious class time erasing chalkboards and whiteboards. Throughout the team's schooling experience, we have witnessed many teachers and students lose their train of thought during an erasing hiatus. There is a need for an erasing device that can work in parallel with the class, ensuring no moment is wasted in any learning environment.

The team learned that in the Electrical and Computer Engineering Building, custodians spend hours throughout the week erasing boards for classes the following day. This not only wastes the time of custodians who have much more important tasks to undertake, but also wastes the money of the university, as it pays for a task that can be automated.

1.2 Solution

The team has proposed a multi-fold device solution to these problems. Dubbed the Board Buddy, the device will act as an automatic erasing apparatus attachable to any whiteboard or chalkboard in the Electrical and Computer Engineering Building.

The Board Buddy will be integrated into a user-friendly application that will allow the device to operate wirelessly. This application will allow remote activation, saving the custodians hours of nightly work. The device will also allow for immediate or timed activation upon user input so that teachers can seamlessly transition from topic to topic on a busy board.

1.3 Visual Aid

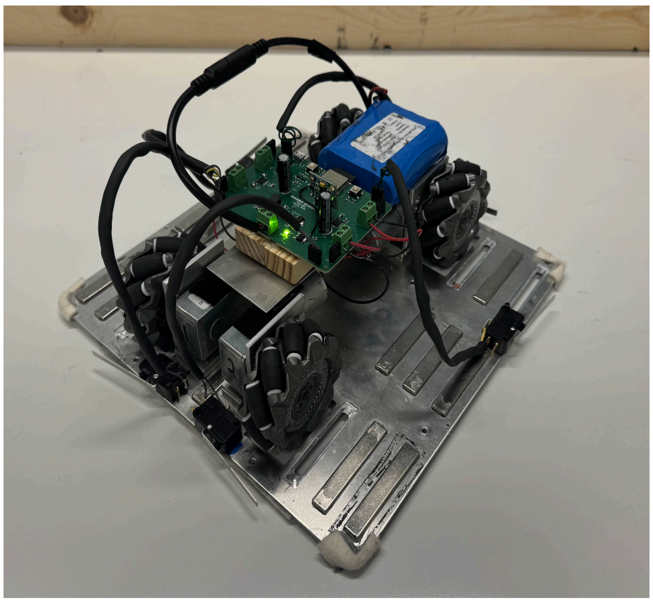

Figure 1 : BoardBuddy Final Design

1.4 High-Level Requirements

High-Level Requirements	
1	The first quantitative characteristic is that this device must erase the majority of the residue in a single pass.
2	The next quantitative characteristic is that it must pass through a typical ECEB educational board (four feet x eight feet) in under two minutes.
3	The final quantitative characteristic is that it must have app integration that will allow for remote and timed activation.

Table 1 : High Level Requirements

2. Design

2.1 Design Overview

2.1.1 Block Diagrams

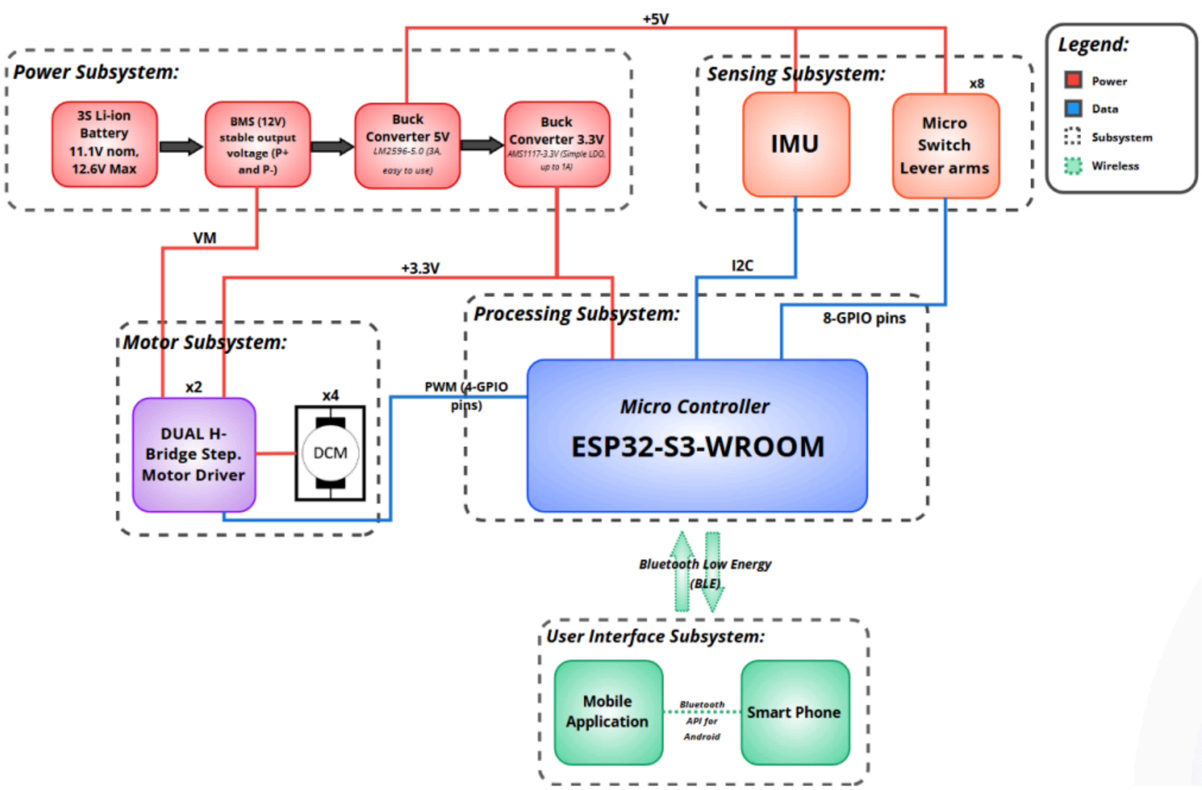

Figure 2 : Hardware Block Diagram

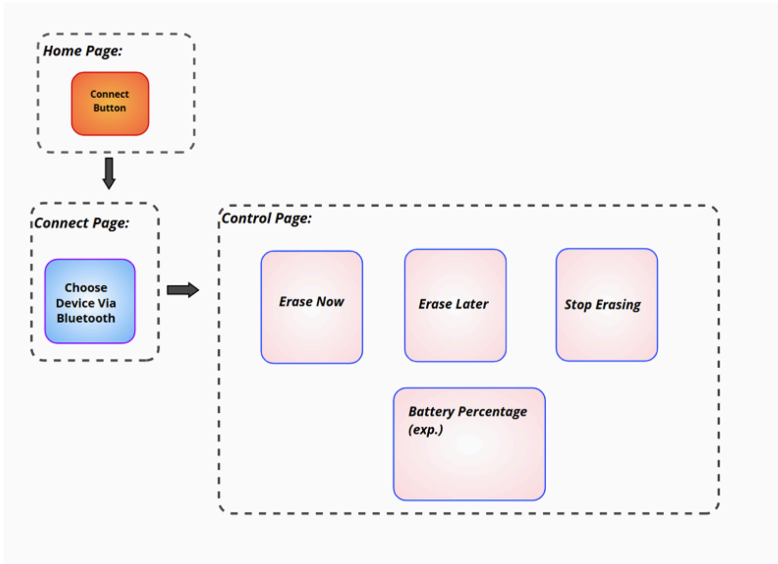

Figure 3 : Application Page Transition Diagram

2.1.2 Power Subsystem

The power subsystem is necessary to provide power to the PCB and any external components (such as our motors). The power subsystem for the BoardBuddy device ensures stable and efficient energy delivery to all electronic components. It includes a rechargeable lithium-ion battery as the primary power source. The subsystem supports charging, with onboard voltage regulators (e.g., buck converters/LDO) to supply 3.3V and 5V rails as required. Low-power design considerations and power-saving modes are integrated to maximize battery life while maintaining reliable performance in portable and active environments.

2.1.3 Motor Subsystem

The motor subsystem enables BoardBuddy's movement and cleaning functionality. It consists of compact DC gear motors for reliable torque and speed control, along with dual H-bridge motor drivers to regulate direction and power delivery. The system is designed for smooth, bidirectional control and integrates PWM signals from the microcontroller to modulate motor speed. Built-in current limiting and thermal protection features in the drivers ensure safe and efficient operation across a variety of surfaces.

2.1.4 Sensing Subsystem

The sensing subsystem provides the BoardBuddy with the physical and positional feedback required for accurate control and effective cleaning of an educational board. It incorporates an Inertial Measurement Unit (IMU) as well as 5 Micro-Switch Lever Arms.

2.1.5 Processing Subsystem

The processing subsystem is a multi-faceted subsystem to integrate the sensing subsystem and the motor subsystem signals in traversal. It also has the purpose of transferring signals between the user interface subsystem through BLE to the microcontroller. Finally, it also transfers data from the power subsystem to be displayed on the user interface.

2.1.6 User Interface Subsystem

The user interface subsystem pertains to the Android application and its functionality. Through BLE, this application will allow for remote activation of the device, in either an instant or timed manner. This subsystem has various user-oriented aspects such as instant erasing, scheduled erasing, the hard stop of erasing, and a display to show the battery level of the device.

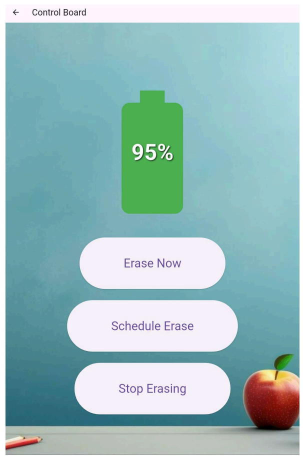

Figure 4 : Application Control Page

2.2 Design Details

2.2.1 Schematic / PCB

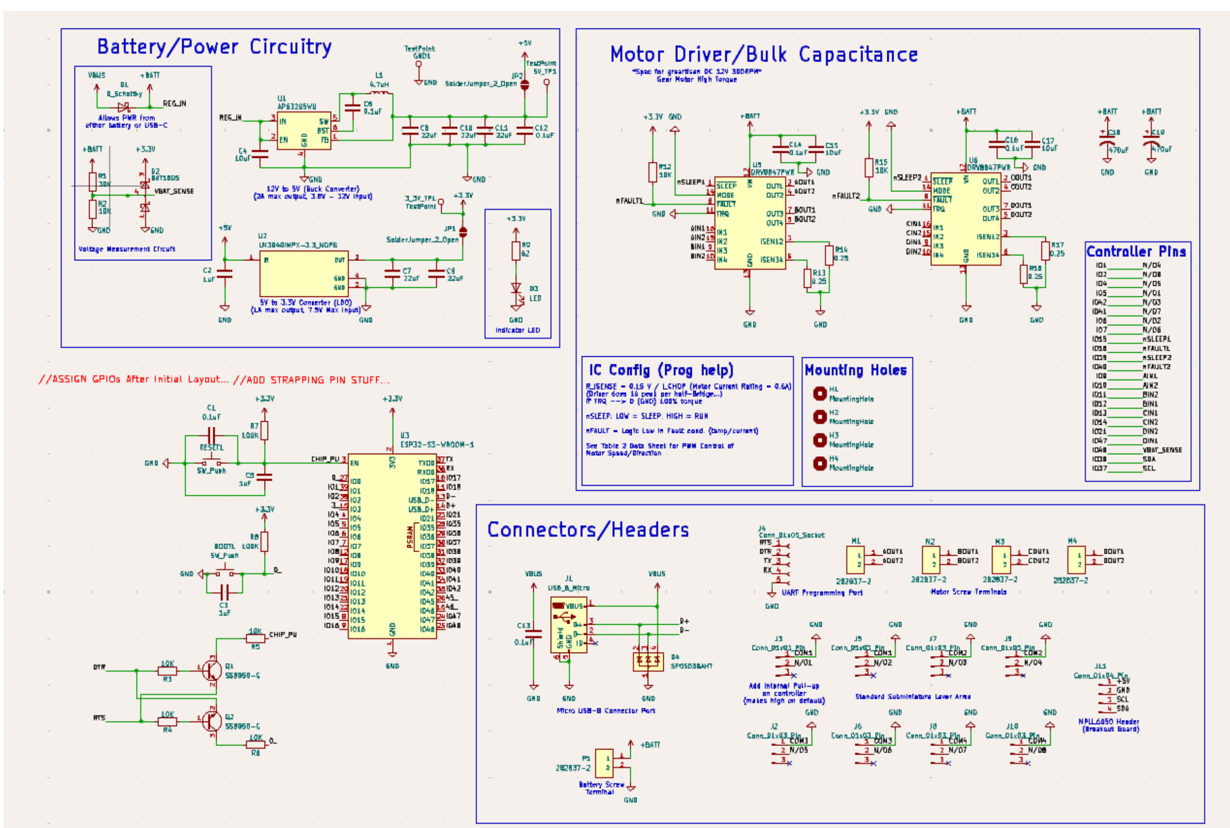

Figure 5 : Schematic

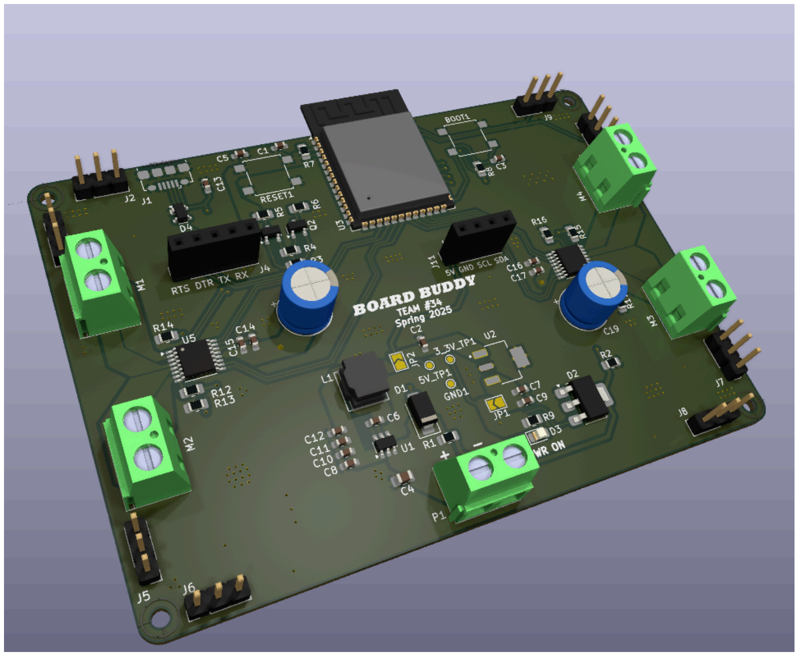

Figure 6 : PCB

2.2.2 Power Subsystem

The first part of the power subsystem was our battery. Our final design employed the Seasider 12V 5000mAh Rechargeable Lithium Battery Pack. Each battery has a built-in protection board to prevent the battery from overcharging, overdischarging, over-voltage, and short-circuits. The battery has a voltage range from 11.3 to 12.6 V and an output of 5A max. The next stage of the power circuitry was a 12V to 5V buck converter and a 5V to 3.3V LDO. Our logic level circuitry peaks out around 750-900 mA, so we required ICs that could handle 2A or more.

The AP63205 buck converter and the LM3940-3.3 LDO. The AP63205 was chosen for its high-efficiency step-down conversion from the battery's ~12V input to a stable 5V rail, capable of delivering up to 2A of current, ideal for high-power peripherals. For components requiring 3.3V, such as the microcontroller and sensors, we specced the LM3940-3.3 LDO. Although it's a linear regulator and less efficient than a switching regulator, it provides low-noise output and sufficient current for low-power digital circuits, making it well-suited for sensitive electronics. Thermal performance, dropout voltage, and quiescent current were also considered during selection to ensure safe and reliable operation across all load conditions.

2.2.3 Motor Subsystem

Motors:

For our application, we used a DC 12V 200 RPM High Torque Turbo Worm Geared Motor rated at 2A stall current, 20W power, and capable of producing up to 10 kg·cm of torque. This motor was selected due to its compact form factor, high torque output, and integrated worm gear, which provides self-locking capabilities *ideal for resisting back-driving when stationary.*

Torque and Load Capacity: The rated torque of the motor is 10 kg·cm, which we used to estimate the maximum pulling force the motor can apply at the wheel.

We used 3-inch diameter wheels, which gives a wheel radius of:

r = 3 in / 2 = 1.5 in = 3.81 cm (2.2.3.1)

To calculate the maximum force at the wheel perimeter (tangential force), we used:

Torque (τ) = Force (F) × Radius ® (2.2.3.2)

$$F = \tau / r$$

We obtain:

$$F = 10\ \text{kg·cm} / 3.81\ \text{cm} \approx 2.625\ \text{kg} \approx 25.76\ \text{N} \qquad (2.2.3.3)$$

To express this pulling force as an equivalent weight under standard gravity

$$Mass_{equiv} = \frac{25.76\ N}{9.81\ m/s^2} \approx 2.63\ kg \approx 5.8\ lb \qquad (2.2.3.4)$$

However, as a safety and thermal constraint, we applied current limiting at 0.5 A, which is one-fourth of the rated stall current. Assuming linear scaling, the *effective pulling capacity per motor* is reduced accordingly:

$$Effective\ Force \approx \frac{1}{4} * 5.8\ lb = 1.45\ lb \qquad (2.2.3.5)$$

With four motors operating in parallel and assuming equal loading and no significant slippage, the system was expected to sustain a total drawbar pull of approximately 5.8 lb under nominal conditions.

Drivers:

We selected the DRV8847 dual H-bridge motor driver from Texas Instruments to control our worm-geared DC motors. This chip is highly integrated and well-suited for our application due to the following reasons: Supports dual brushed DC motors with current outputs up to 2 A RMS, meeting our motor's full stall current spec. Built-in current regulation allows us to limit motor current in software using an external sense resistor and internal PWM chopping. Wide voltage range (4.5 V to 18 V) is compatible with our 12 V motor supply.

The operating voltage range of the driver (4.5 V to 18 V) aligns with our system's 12 V supply rail. To ensure voltage stability during transient events such as motor startup or stall, we calculated and included bulk capacitance on the motor power rail. Based on typical capacitor sizing guidance for transient suppression, we employed the following expression:

$$C_{bulk} = \frac{I_{motor} \bullet t_{rise}}{\Delta V} \tag{2.2.3.6}$$

Where:

- I_{motor} = 2A (Stall Current)
- t_{rise} = 1 ms (estimated transient duration)
- ΔV = 2V (maximum allowable voltage drop)

Substituting values into (2.2.3.6):

$$C_{bulk} = \frac{2\,A \bullet 0.001\,s}{2\,V} = 0.001\,F = 1000\,\mu F$$

We implemented this capacitance as two parallel 470 µF electrolytic capacitors, providing sufficient energy buffering capacity to reduce line droop and improve driver performance.

2.2.4 Sensing Subsystem

This subsystem was designed in order to provide the BoardBuddy with responsive feedback based on both positional awareness and orientation correction.

MPU 6050 IMU: This was the IMU we selected due to its compact form, low cost, and dual-function sensing (gyroscope and accelerometer). While drift is a common issue and challenge that comes with low-cost MEMS gyroscopes, we believed that our software bias and periodic re-zeroing via board-edge contact would minimize the long-term error, as well as our trying to implement different types of filters to reduce the amount of drift that is present in the IMU.

Micro-Switch Lever Arms: These are mechanical switches that were placed around the BoardBuddy chassis; these switches reliably provided us with triggers when the edge of the board was reached. Each switch was validated with multimeter continuity testing in order to ensure the functionality of each limit switch. The state of the limit switch was communicated via BLE for external debugging, and its logic was key in the operation of the motion sequences of the BoardBuddy device, including the front/back contact which causes for the yaw angle to re-zero, the right contact to initialize the start of the homing movement, also completing the homing movement once the left limit switch comes to contact with the left edge of the board.

By combining these two parts of the sensing subsystem, it enabled reactive, reliable decision making, which is essential for autonomous board cleaning where precision and safety are critical.

2.2.5 Processing Subsystem

The subsystem has three major aspects: limit switch to motor subsystem data transfer, IMU to motor subsystem data transfer, and microcontroller to user interface subsystem data transfer. All the subsequent data transfers occurred with a 100 ms updating period

The limit switch data transfer uses I/O GPIO (specifically GPIO7, GPIO4, GPIO6, GPIO41, and GPIO1) to receive edge detection actuation signals to determine the movement of the device. With signals being sent to COM and N/O set to ground, these switches were used as input pull-up signals set to a low value. Upon actuation, a high signal was received and transferred to the microcontroller. From here, this signal was used to traverse the device appropriately.

The IMU uses I2C data transfer to account for the gyroscope and accelerometer data of our device's position. Through various filtering techniques, this signal was sent using SDA (GPIO38) and SCL (GPIO37) to the microcontroller to account for off-axis drift for appropriate intervals.

The microcontroller user interface integration used BLE data transfer with a latency of less than 10 ms up to a tested 20 feet. String data transfer from the app to the device was either delayed for a scheduled interval or instant, based on user input. Real-time voltage data was received by the microcontroller through A/DC transfer (GPIO8) from the power subsystem. This string data was relayed through BLE to the user interface subsystem to display the battery percentage of the device.

2.2.6 User-Interface Subsystem

This subsystem was an Android-based application for the control and monitoring of the device. While the focus of this application was Android devices, software dependencies were included so that it may also run on Chrome, Windows desktop, and Edge. This application was made using Dart programming through a Flutter app-building framework. This application controls and monitors the device through BLE using the processing framework as outlined in the previous Section 2.2.5.

This application uses the page/state transition, as seen in Figure 3. The BLE device interface uses a multi-faceted framework of assurance in device connection. Only with the assurance of Service and Characteristic UUIDs, along with the device name, will the application connect to the device.

3. Design Justification and Alternatives

3.1 Design Justification and Tolerance Analysis

The following design justifications and analyses are intended to ensure the requirements and verifications of the device design as outlined in Appendix B.

3.1.1 Power Subsystem

Voltage measurements were recorded at key nodes of the power subsystem using a laboratory-grade digital multimeter (DMM). The measured values were as follows:

- Battery terminal: 12.623 V
- Buck converter output: 4.962 V
- Low dropout (LDO) regulator output: 3.291 V

These values confirm that all subsystem voltages fall within acceptable tolerances for their respective loads. No modifications to the power regulation architecture were required based on these measurements.

A higher capacity lithium-ion battery was selected in the final design revision to accommodate increased power demands, particularly due to the use of upgraded motors with elevated current requirements. The DRV8847 motor driver was configured with a hardware-enforced current limit of 0.5 A per half-bridge. Since the system employs four motors, each driven by a separate half-bridge, the total peak current draw from the motor subsystem is:

$$I_{total} = 0.5\,A \times 4 = 2.0\,A \tag{3.1.1.1}$$

Accounting for additional current draw from logic, sensors, and inefficiencies, the peak board-level current was measured at 2.7438 A. The estimated instantaneous power consumption under full load is:

$$P = V \times I = 12\,V \times 2.7438\,A = 32.93\,W \tag{3.1.1.2}$$

The selected battery has a rated capacity of 5 Ah at 12 V, corresponding to an energy capacity of:

$$E = 12\,V \times 5\,Ah = 60\,Wh \tag{3.1.1.3}$$

An ideal runtime estimate (ignoring non-idealities such as efficiency losses) can be obtained by:

$$t_{ideal} = \frac{60\,Wh}{32.93\,W} \approx 1.83\,h \tag{3.1.1.3}$$

Empirical testing of the final system yielded a runtime of 2 h and 22 min (2.367 h) under nominal conditions. This discrepancy is attributed to the test conditions: the device was operated unloaded (suspended on a support structure), reducing actual mechanical power demands. In addition, the ESP32 and other control subsystems may not have reached their maximum current draw during typical operation.

Despite the simplified conditions, this result significantly exceeded the design requirement of 30 minutes of continuous runtime, demonstrating effective power management and confirming that battery sizing was sufficient for the intended application.

3.1.2 Motor Subsystem

Motors allowed for stable, omnidirectional movement. We were able to both transverse and erase the board (horizontal) with no issue. Drivetrain can be slightly upgraded (see section 3.2), but PWM signals from the microcontroller as well as the H-bridge output, worked as expected.

Figure 7: DRV8847 Output (80% Duty Cycle)

Figure 8: ESP32 Signal Output (80% Duty Cycle)

The left image shows the DRV8847 motor output pin voltage, and the right image shows the microcontroller output. As one can see (Fig. 8), with the duty cycle set to 80%, we do indeed get a pin output with a + duty cycle of ~80 %. On the driver side (left image), we also get a duty cycle of ~80 percent. We take the delta of the two cursors (x1-x2) and get 780 µs. As the period is 1ms, we do indeed satisfy our variable speed requirement.

3.1.3 Sensing Subsystem

The processing subsystem has two main aspects: The Inertial Measurement Unit and the Micro-Switch Lever Arm.

The MPU6050 IMU is a six-axis IMU, which was mounted centrally on the BoardBuddy chassis, which measures linear acceleration (three-axis) and angular velocity (three-axis). Only the Z-axis angular velocity (angleZ) is used in real-time yaw calculations to account for drift correction. The raw sensor values are read via I2C on GPIO pins 38 (SDA) and 37 (SCL), which are then converted to degrees per second using the appropriate scale based on our MPU6050. This enables the BoardBuddy to detect off-axis deviation and apply PWM-based differential motor correction, which allows for the BoardBuddy to remain aligned during its traversal across the educational board. To compute our angleZ from the gyroscope data, we used an approximation to get us a value which we deemed to be accurate enough to get a reading on angleZ.

$$\Delta\theta = w * \Delta t \tag{3.1.3.1}$$

Where w was the angular velocity in degrees per second, which was further corrected by including a calibration routine in order to store a bias offset, which was then subtracted from real-time readings in order to reduce drifts, and Δt was time steps in seconds.

$$w = \frac{Gz}{131} - gyroZBias \tag{3.1.3.2}$$

This equation shows the calibrated angular velocity, which uses 200 samples of gyroZBias while the BoardBuddy device is stationary, in order to compute and store a bias offset, which is then subtracted from every real-time reading in order to give us a more accurate angleZ.

Once we have an angle reading of angleZ, we enable a tuning threshold of 1.5° in either direction. If the yaw angle exceeds this threshold, an additional PWM was applied to the correct motors opposite of the drift directions in order to fix the tilt off axis.

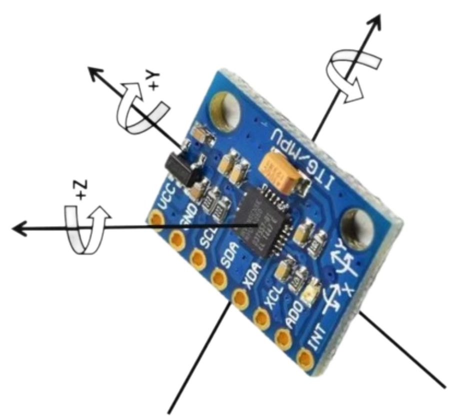

Figure 9 : MPU6050 Orientation

if (angleZ <= -TOLERANCE) {adj3 += CORRECTION_INCREMENT; adj4 += CORRECTION_INCREMENT;}

else if (angleZ >= TOLERANCE) {adj1 += CORRECTION_INCREMENT; adj2 += CORRECTION_INCREMENT;}

This was implemented in the code as such, this correction loop enabled the BoardBuddy to maintain a stable forward and reverse trajectory across the entire educational board and should account for any motor mismatches that are inherent due to the tolerance variation present on each motor.

The Micro-Switch Lever Arms provide the ESP32 with digital signals via GPIO interrupts. They are positioned at the front, back, left, and right sides of the chassis, which then actuate when the device reaches the edge of the board. The limit switches are configured with internal pull-ups and are read using the "digitalRead()" logic. The front and back limit switches are used not only to signal the start of right movement but also to reset angleZ to zero degrees in order to prevent long-term gyro drift from the IMU. As the BoardBuddy device traverses, it follows a simple S-Pattern traversal until the right limit switch (LIM_SWITCH6) is actuated. It waits for the back limit switch to be triggered in order to begin its homing routine, moving back to its original position on the educational board.

3.1.4 Processing Subsystem

The processing subsystem had many different components that had to be considered while handling data transmission. Key components such as the microcontroller and its GPIO capabilities had to have certain functionality to allow our design to work as intended.

In choosing a microcontroller, we had to ensure that BLE and WIFI were available, as some microcontrollers only had Bluetooth capabilities. This microcontroller also had to have a surplus of GPIO as our system had eight limit switches, four bidirectional motors, one IMU, and other niche components such as battery level output.

It was key to choose a microcontroller with BLE and antenna as not to use a surplus of energy during its simple interfacing with the user interface subsystem. As the goal of our edge detecting actuators was to read high and low signals, we had to ensure that the microcontroller had at least eight I/O GPIO. While the IMU communicates using SDA and SCL, we had to ensure that this microcontroller also had these serial data capabilities. Finally, we had to ensure that our chosen microcontroller had at least one ADC GPIO to ensure that the analog data of our voltage level was properly communicated to the application.

In making this subsystem, there was one key tolerance that we had to analyze. This tolerance was the capabilities of the BLE connection while being surrounded by motors and magnets. A simple connection test proved that the EMI from both the motors and the magnets did not have a significant effect on the connection up to a tested 20 feet. Moving the PCB as far from the magnets and motors as we could, this connection stayed stable and never interfered with our design.

In choosing our microcontroller and designing our firmware, we had to ensure a few capabilities were included:

- BLE latency less than 10ms
- Firmware signals are updated every 100ms

These capabilities were met by reference to the microcontroller datasheet along with the sleep functions included in the firmware.

3.1.5 User-Interface Subsystem

There were multiple things that had to be considered while implementing the user-interfacing subsystem. The most important aspect of this subsystem was the focus of what devices can run the application. Using a Flutter app building framework with Dart programming allowed us to make an application capable of running on Windows Desktop, Android, Microsoft

Edge, and Chrome. This was chosen as certain dependencies could be added to include these other ways to run the application, compared to other programming techniques (such as Android Studio) that would have blocked us from using more than a single type of device.

In making this subsystem, certain functionality had to be chosen to ensure the safety and "user-friendliness" of the design. Initially, we had planned to make a design that could activate the device instantly, activate the device at a certain time, stop the device, and display battery voltage. Other than this, we had initially planned to make the application capable of keeping a set of 10 erasing times to run throughout the week. While this is true, this aspect of its functionality had to be taken out due to complications with SRAM data being overwritten. Although this aspect of functionality was taken out in early design considerations, all other aspects of this application were met.

The BLE connection of the device to the application also had to be fervently considered. As to not divulge any sensitive information, the BLE connection was based on the name of the BoardBuddy device and its service/characteristic UUIDs. This was chosen rather than divulging any information about the application device itself (such as its name or MAC address). Only through meeting all of these signifiers would the application connect to the device. The specific signifiers for our device can be referenced below:

- Device Name : Room4070 BoardBuddy
- Service UUID : "4fafc201-1fb5-459e-8fcc-c5c9c331914b"
- Characteristic UUID : "beb5483e-36e1-4688-b7f5-ea07361b26a8"

The battery level of the device is shown as a percentage at the top of the control page, as seen in Figure 4. This was calculated using a 10-second running average to not produce a noisy signal. This calculation can be seen below.

$$Battery\ Percentage \ = \ \frac{Average\ Reading - 11.1}{1.5} \times 100\% \qquad (3.1.5.1)$$

In order to successfully activate this device at instant and scheduled times, an activation signal is sent from the application to the device. This activation signal is either sent instantly or delayed until a user-scheduled time. Doing this allowed for the real-time alteration of activation time. This was chosen rather than delaying the signal in the microcontroller itself, as programming resets would overwrite any previously scheduled times.

3.2 Design Alternatives

In the process of final functional tests, it was concluded that running the device vertically on a board was impossible with our current implementation. For one, we found that the sheet metal chassis was on the upper end of our calculated tolerance analysis, total device weight being ~5.6 lbs. The motors simply did not have the torque to fight both gravity due to the weight of the device and the magnetic normal force into the board. Secondly, with the sheet metal, we needed very fine tolerances in order to make good contact with the board (for reference, some of our wheels weren't even in contact with the board).

For future revisions of this device, there are available motors on the market with similar current /rpm ratings to our current motors, which can span from 20kg-cm to 40 kg-cm of torque, double and even quadruple the torque we have at the moment. Unfortunately, these motors are very costly (~$18-20 per motor), which is what deterred us from this decision in the first place, however, this addition would work.

Secondly, another material, such as PLA would have been a better selection for the chassis. In the prototyping process, we could have adjusted wheel positions/magnet heights without the need of machining, not to mention the weight saving based on the endless possible in-fill options. Both of these additions would have made vertical traversal possible.

Initially, we used the MPU6050, a 6-DOF IMU that includes a 3-axis accelerometer and 3-axis gyroscope, to estimate the orientation of our robot. However, during testing, we encountered noticeable angle drift over time, especially during long runs or when the robot moved slowly or stayed in place. This drift is a common limitation of gyroscope-based systems due to bias instability and integration error. We attempted to correct this by implementing both complementary and Kalman filters, which improved short-term stability but failed to eliminate long-term drift entirely. In hindsight, a more robust solution would have been to use an IMU with an integrated magnetometer, such as the MPU9250 or BNO055. The magnetometer would have provided an absolute heading reference with respect to the Earth's magnetic field, allowing for better drift correction in yaw estimation and more reliable long-term orientation tracking. This upgrade would have significantly improved the robot's ability to maintain accurate directional control during extended operation.

Finally, to improve the functionality of the user interface subsystem, it would be pertinent to solve our previous issues with SRAM data being overwritten. Solving this issue would allow us to save a log of continuous erasing times, meeting our requirement. If this issue persists, it may be useful to include a separate memory drive to save these erasing logs. After completing the final presentation, we found a new importance of this application having functionality over multiple devices. Due to this, the team has found it important that in the future we make applications on both Swift and Dart programming, to allow for both IOS and Android implementation.

4. Costs and Schedule

4.1 Cost Analysis

4.1.1 Bill of Materials

The Bill of Materials (BOM) for both prototyping and mass production of the BoardBuddy device is detailed in Appendix C. These figures include the finalized component selection and cost of all components, which have been verified through Digi-Key and vendor invoices.

Prototype Creation Cost: The total cost for the initial construction of the BoardBuddy device prototype came out to $173.51, which includes mechanical, electrical, and structural components. This price reflects individual parts sourcing, which includes shipping premiums, non-bulk rates, and surplus part ordering for rework.

Mass Production Estimates (100 Units): For mass production, the total costs drop significantly to $3,325.00 or $33.25 per device. This is a huge reduction in unit cost due to the bulk pricing discounts, shared PCB fabrication, and reduced shipping and setup fees.

4.1.2 Labor

The table below presents our labor analysis, detailing the total hours contributed to the project. It includes both internal labor from our team and external support from resources like the machine shop. This breakdown helps illustrate the time investment required for various project tasks. By capturing all sources of labor, we can better assess project scope and resource allocation for possible future work.

Labor Analysis				
	Hourly Wage	Hours to Complete	Calculation	Total
Louie	$45.00	132	$45.00 * 2.5 * 132 hrs	$14,850.00
Gabe	$45.00	110	$45.00 * 2.5 * 110 hrs	$12,375.00
Alfredo	$45.00	121	$45.00 * 2.5 * 121 hrs	$13,612.50
Machine Shop	~$25.00	5	$25.00 * 2.5 * 5hrs	$312.50
Total	NA	367	NA	$41,150.00

Table 2 : Labor Analysis

4.2 Schedule

Team 34 : February 2025				
	Week 1	Week 2	Week 3	Week4
Louie	-Initial Discussion(3 hrs) -Follow Up Discussion (2 hrs)	-Project Proposal (6 hrs) -Follow Up Discussion (2 hrs)	-Follow Up Discussion (2 hrs) -App Revisions (8	-ESP App Integration (9 hrs)

	-CAD (5 hrs)	-App Research (5 hrs) -App Coding (4 hrs)	hrs) -App Coding (4 hrs)	-Demo Preparation (4 hrs)
Alfredo	-Initial Discussion (3 hrs) -Follow Up Discussion (2 hrs)	-Project Proposal (6 hrs) -Follow Up Discussion (2 hrs) -BLE Connection Research (4 hrs)	-Follow Up Discussion (2 hrs) -Initial ESP Coding Research (5 hrs) -BLE Connection Between App and ESP (5 hrs)	-Firmware Programmin g (4 hrs) -Demo Preparation (4 hrs)
Gabe	-Initial Discussion (3 hrs) -Follow Up Discussion (2 hrs)	-Project Proposal (6 hrs) -Follow Up Discussion (2 hrs) -Initial PCB component Research (6 hrs)	-Follow Up Discussion (2 hrs) -Initial PCB component Research (6 hrs)	-Final PCB component Selection (5 hrs) -Demo Preparation (4 hrs)

Table 3 : February Schedule

Team 34 : March 2025				
	Week 1	Week 2	Week 3	Week4
Louie	-Team Discussion (2 hrs) -Demo Preparation (4 hrs) -Design Document (6 hrs)	-Demo Follow Up Discussion (3 hrs) -App Revisions (10 hrs)	Spring Break (NA)	Spring Break (NA) -Programming Prep (IMU) (4 hrs)
Alfredo	-Team Discussion (2 hrs) -Demo Preparation (8 hrs) -Design Document (4 hrs)	-Demo Follow Up Discussion (3 hrs) -Firmware Programming (IMU) (5 hrs)	Spring Break (NA)	-Motor Driver Firmware Programming (6 hrs)
Gabe	-Team Discussion (2 hrs) -PCB Schematic	-PCB Layout (10 hrs)	Spring Break (NA)	-PCB Layout Rev 2 (5 hrs) - PCB Soldering

	Design (7 hrs) -Design Document (6 hrs)			(9 hrs)

Table 4 : March Schedule

Team 34 : April 2025				
	Week 1	Week 2	Week 3	Week4
Louie	-Programming Prep (IMU) (7 hrs) -Programming Assistance (5 hrs) -Machinery Fixing (1 hr)	-Programming Assistance (10 hrs) -Machinery Fixing (1 hr)	-Machinery Fixing (5 hrs) -Programming Assistance (10 hrs) -Final Demo Prep (2 hrs)	-Programming Touch Up (1 hr) -Final Presentation Prep (2 hrs)
Alfredo	-Machinery Fixing (1 hr) -Firmware Programming (10 hrs)	-Firmware Programming (15 hrs)	-Firmware Programming (12 hrs) -Final Demo Prep (2 hrs) -App Development Assistance (6 hrs)	-Firmware Programming (3 hrs) -Final Presentation Prep (2 hrs)
Gabe	-GeneralTeam Assistance (4 hrs) - PCB Soldering (9 hrs) (Rev 2 board)	-General Team Assistance (6 hrs)	- General Team Assistance (6 hrs) -Final Demo Prep (2 hrs)	-Final Presentation Prep (2 hrs)

Table 5 : April Schedule

Team 34 : May 2025				
	Week 1	Week 2	Week 3	Week4
Louie	-Final Report (6 hrs)	NA	NA	NA
Alfredo	-Final Report (5 hrs)	NA	NA	NA

Gabe	-Final Report (6 hrs)	NA	NA	NA

Table 6 : May Schedule

5. Ethics and Safety

IEEE Code of Ethics (IEEE Code of Ethics, 2025)

[I.1] Regarding IEEE Code I.1, the team must ensure the safety and privacy of the public. To do this, we must take into account the privacy of the device application so we do not release sensitive information. Furthermore, regarding the safety of the public, we must take into account the magnetic aspect of the project. We must also ensure that the magnet is not strong enough to damage electronics or fall off an educational board.

[I.5] According to the IEEE Code I.5, we must ensure that we state realistic claims of our device's capability while properly crediting the work of others. To do this, we will reach out to the course assistance networks as often as we need while recording honest data to be published with our work. In publishing this work, we will ensure that references are always given to any party that assisted.

ACM Code of Ethics (ACM Code of Ethics, 2025)

[1.6] A particular ACM Code of Ethics, Code 1.6, will be of significant importance in this project. As this device will have Bluetooth capabilities, we must ensure that the application connection respects the privacy of the user. We must be sure to use only the minimum amount of personal information to ensure the application works as desired.

University of Illinois at Urbana-Champaign Student Code of Ethics (Student Code, 2025)

[1.4] Article 1, Part 4, Academic Integrity, is a code that the team strives to uphold. In this project, the same as with many projects before, we must uphold academic integrity. We must ensure that we accredit any affiliated parties or released works that have assisted us.

[Stanford] A large number of Neodymium Magnets are being activated while this product is in use. Documentation for the safety of use of these magnets is provided in the references. The team will follow these guidelines to ensure the safe creation and use of the product.

[Harvard] A Lithium-Ion Battery is being charged and discharged while this product is in use. Documentation for the safety of use of these batteries is provided in the references. The team will follow these guidelines to ensure the safe creation and use of the product.

6. Conclusion

The BoardBuddy project successfully demonstrated a viable solution to an autonomous educational board cleaning device. By integrating motion control, inertial sensing, and Bluetooth connection, which allows for operation through a user-friendly application. The final prototype device achieved traversal on a flat whiteboard, with responsive edge detection and consistent board coverage.

From a marketing point of view, the device seems to be very appealing at a mass production cost of $35 per unit; the BoardBuddy device could reduce the operational costs of the custodial staff at the Electrical-Computer Engineering Building. We interviewed various custodians and on average, seven and a half hours are spent only on cleaning educational boards throughout the ECEB. If the system is successfully implemented across the ECEB, the system will pay for itself in under a week. Clearly offering value to the institution that is constantly seeking efficiency and automation.

Several technical components of this project functioned especially well, including the BLE-enabled Android user interface that offered reliable remote and scheduled activations, and the properly interfaced signals through ESP32 to make the IMU and limit switches interact with the motors. As a group, we learned the importance of embedded systems integrations and real-world sensor limitations

References

"324 DigiKey." *DigiKey Electronics*, www.digikey.com.br/en/products/detail/adafruit-industries-llc/324/5022791?gQT=1. Accessed 12 Feb. 2025.

"ACM Code of Ethics." *Code of Ethics*, www.acm.org/code-of-ethics/. Accessed 8 Feb. 2025.

"DRV8421." *DRV8421 Data Sheet, Product Information and Support | TI.Com*, www.ti.com/product/DRV8421. Accessed 12 Feb. 2025.

Esp32 Series, www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. Accessed 13 Feb. 2025.

Harvard Environmental Health & Safety. Laboratory Safety Guideline: Lithium Ion Batteries. Harvard University, n.d., https://www.ehs.harvard.edu/sites/default/files/lab_safety_guideline_lithium_ion_batteries.pdf. Accessed 5 Mar. 2025.

IEEE - IEEE Code of Ethics, www.ieee.org/about/corporate/governance/p7-8.html. Accessed 8 Feb. 2025.

"MPU-6000 and MPU-6050 Product Specification Revision ..." *MPU-6000 and MPU-6050*, invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf. Accessed 13 Feb. 2025.

Appendix A: Abbreviations

Unit or Term	Abbreviation Symbol
Acceleration	a
Ampere	A
Ampere-hour	Ah
Analog to Digital Converter	ADC
Application	App
Application Programming Interface	API
Association for Computing Machinery	ACM
Bill of Materials	BOM
Bluetooth Low Energy	BLE
Battery Management System	BMS
Common	COM
Degrees of Freedom	DOF
Direct Current	DC
Discontinuous Conduction Mode	DCM
Electrical and Computer Engineering	ECE
Electrical and Computer Engineering Building	ECEB
ElectroMagnetic Interference	EMI
Espressif Systems Platform	ESP
Force	F
General Purpose Input/Output	GPIO
Ground Terminal	GND
Gram	g
Hertz	Hz
Input/Output	I/O
Institute of Electrical and Electronics Engineers	IEEE
Integrated Circuit	IC
Inter-Integrated Circuit	I2C
Internal Measurement Unit	IMU
Kilo-Hertz	kHz
Low Dropout Regulator	LDO

Unit or Term	Abbreviation Symbol
Mass	m
Mechanical Engineering	ME
Media Access Control	MAC
Micro-Ampere	µA
Micro-Fahrrad	µF
Milli-Ampere	mA
Millimeter	mm
Milli-Second	ms
Newton	N
Normally Closed	N/C
Normally Open	N/O
Not Applicable	NA
PolyLactic Acid	PLA
Pound	lbs
Printed Circuit Board	PCB
Pulse Width Modulation	PWM
Radio Frequency	RF
Static Random Access Memory	SRAM
Teaching Assistant	TA
Universally Unique Identifier	UUID
University of Illinois Urbana - Champaign	UIUC
United States Dollar	USD
Revolutions per Minute	RPM
Revolutions per Second	RPS
Serial Clock Line	SCL
Serial Data Line	SDL
Voltage	V
Voltage at Motor	VM
Watt	W
Watt-hours	Wh
Wireless Fidelity	WIFI

Appendix B: Requirements and Verifications

Requirement	***Verification***	***Verification Status***
Power Subsystem	**Power Subsystem**	**Y or N**
R1: Provide a stable output of 12V +/- 0.1V (1400 mA) (for motors).	V1: Ensure 12 Volts on the Power Rail by verifying with a multimeter underload and confirming its 12±0.1V when motors are running. (+VBATT)	Y
R2: Provide a stable output of 5V +/- 0.1V (3.8 mA) (IMU and limit switch)	V2: Ensure 5 Volts on Power Rail by verifying with a multimeter underload and confirming its 5±0.1V when motors are running. (+5.5V_TP)	Y
R3: Provide a stable output of 3.3V +/- 0.1V (740 mA) (Microcontroller and H-Driver ICs)	V3: Ensure 3.3 Volts on Power Rail by verifying with a multimeter underload and confirming its 3.3±0.1V when motors are running. (+3.3V_TP)	Y
Sensing Subsystem	**Sensing Subsystem**	
R1: The IMU must provide x and y orientation data with an accuracy of ±2°.	V1: Place the IMU on a calibrated surface and tilt at known angles in order to verify that the orientation is accurate to ±2°. (GPIO 38, 37)	Y
R2: The system must update orientation estimates at a frequency of at least 1 Hz.	V2: Be able to log data from the IMU in order to confirm that updates of orientation data are updating at least ≥1 Hz.	Y
R3: The IMU must detect gradual changes in motion (e.g., off-axis drift) and trigger an appropriate response within 200 ms.	V3: Introduce drift off-axis control and confirm that the system is triggered within 200ms.	Y
R4:	V4:	Y

The IMU data must be processed using a sensor fusion algorithm (e.g., complementary filter, Kalman filter) to reduce drift.	Implement an algorithm in order to show reduction in drift.	
R5: Each limit switch must detect contact within 10 ms of impact.	V5: Test each limit switch by actuating and make sure that the data is received by the ESP 32 within 10ms. (GPIO 1,2,4,5,42,41,6,7)	Y
R6: The system must stop movement within 50 ms of a limit switch activation.	V6: Verify the system halts motor movement within 50 ms of a limit switch activation.	Y
Processing Subsystem	**Processing Subsystem**	
R1: ESP32 should take in sensor readings/inputs and provide the correct output to motors (within the time restrictions outlined in Section 2.3.2).	V1: Simulate sensor inputs and confirm ESP 32 produces correct PWM output to the motors. (GPIO 9,10,11,12,13,14,21,47).	Y
R2: The UI provided by the configured app should communicate with the ESP32 as outlined in this document, with a latency of less than 10ms.	V2: Use the Flutter app and send commands to the ESP 32, making sure to confirm that connection latency is within 10ms.	Y
R3: The flash storage or shared preferences should store at least 10 scheduled erasing events.	V3: Be able to schedule and confirm that 10 events are stored in the ESP 32 flash memory or shared preferences with the app, and are retrieved properly.	N
Motor Subsystem	**Motor Subsystem**	
R1: Should be able to adjust speed via ESP32 direction based on actuation of lever arms and IMU data.	V1: Use PWM test signals to control motor speed and direction, and be able to detect lever arm and IMU data.	Y
R2: Needs to be able to support an omnidirectional movement	V2: Verify that motors can perform omnidirectional movement.	Y

across the board.		
R3: To pass the high-level requirement, the device needs to be able to move at 4.8 inches per second.	V3: Place the device on an educational board and measure travel speed to confirm it reaches at least 4.8 inches per second.	Y
User Interface Subsystem	**User Interface Subsystem**	
R1: A user interface must be created that allows for the board to be scheduled and erased.	V1: Verify that the app allows users to schedule and erase the board through user-friendly interactions.	Y
R2: Must be able to store a history log of erasings in the ESP flash drive or shared preferences.	V2: Schedule multiple board erasing events through the user interface and confirm all events are correctly saved in ESP32 flash storage or shared preferences. Retrieve and display the saved schedule and history log to verify accuracy.	N
R3: Be able to communicate with the ESP-32 module through Bluetooth.	V3: Pair the user interface with the ESP32 via Bluetooth and confirm stable communication.	Y
R4: Real-Time Status Changes involving battery life or charging status of the device, determining if the device is currently being used, which would allow us to turn off the device automatically.	V4: Simulate different device states (charging, low battery, active erasing, idle) and confirm the user interface reflects these status changes in real time.	Y

Appendix C: Bill of Materials

Bill of Materials for Project Prototyping				
Quantity	Part Number	Manufacturer Part Number	Description	Extended Price USD
4	SW415-ND	B3S-1000	SWITCH TACTILE SPST-NO 0.05A 24V	2.28
2	296-53425-1-ND	DRV8847PWR	2A DUAL H-BRIDGE MOTOR DRIVER	3.74
2	LM3940IMPX-3.3/NOPBCT-ND	LM3940IMPX-3.3/NOPB	IC REG LINEAR 3.3V 1A SOT-223-4	3.26
2	AP63205WU-7DICT-ND	AP63205WU-7	IC REG BUCK 5V 2A TSOT23-6	2.76
2	311-62ARCT-ND	RC0805JR-0762RL	RES 62 OHM 5% 1/8W 0805	0.20
4	CSR0805FKR250CT-ND	CSR0805FKR250	RES 0.25 OHM 1% 1/4W 0805	1.12
10	RMCF0805FT30K0CT-ND	RMCF0805FT30K0	RES 30K OHM 1% 1/8W 0805	0.25
5	A113320-ND	282837-2	TERM BLK 2P SIDE ENT 5.08MM PCB	5.60
2	SRN6045TA-4R7MCT-ND	SRN6045TA-4R7M	FIXED IND 4.7UH 4.5A 26 MOHM SMD	0.88
2	S6103-ND	PPTC051LFBN-RC	CONN HDR 5POS 0.1 TIN PCB	0.84
2	3147-B1701UYG-20D000114U1930CT-ND	B1701UYG-20D000114U1930	LED YLW-GRN DIFFUSED 0805 SMD	0.20
2	1727-5451-1-ND	BAT160S,115	DIODE ARRAY SCHOT 60V 1A SOT-223	1.62
2	P15392CT-ND	EEU-FR1V471LB	CAP ALUM 470UF 20% 35V RADIAL TH	1.34
10	1276-1274-1-ND	CL10A226MP8NUNE	CAP CER 22UF 10V X5R 0603	0.59
6	C0603C104J4RACTU	C0603C104J4RACTU	0.1 µF ±5% 16V Ceramic Capacitor X7R 0603 (1608 Metric)	0.72
3	CL10B105KA8N	CL10B105KA8NNNC	1 µF ±10% 25V Ceramic Capacitor X7R	0.18

	NNC		0603 (1608 Metric)	
1	641-1707-2-ND	CDBA540-HF	DIODE SCHOTTKY 40V 5A DO214AC	0.14241
1	1965-ESP32-WROOM-32-N4DKR-ND	ESP32-WROOM-32-N4	RF TXRX MOD BLUETOOTH WIFI SMD	6.56
1	Case	NA	Aluminum Base	5.00
4	Omni Wheels	NA	Omnidirectional Wheels	30.00
4	DC Motors	Greartisan DC 12V 300RPM	Greartisan DC 12V 300RPM Gear Motor High Torque Electric Micro Speed Reduction Geared Motor Eccentric Output Shaft 37mm Diameter Gearbox	60.54
30	Neodymium Magnets	MIN CI Super Strong Neodymium Magnet Bar	MIN CI Super Strong Neodymium Magnet Bar, 40 X 10 X 3 mm Powerful Rare Earth Magnets Strip Heavy Duty	24.50
1	Lithium-Ion Battery	Lithium-Ion Battery	Lithium-Ion Battery	15.00
5	PCBs	PCB	PCBWay	5.00
TOTAL	TOTAL	TOTAL	TOTAL	$173.51

Bill of Materials for Mass Production (100 Devices)				
	Part Number	**Manufacturer Part Number**	**Description**	**Extended Price USD**
	SW415-ND	B3S-1000	SWITCH TACTILE SPST-NO 0.05A 24V	30.00
	296-53425-1-ND	DRV8847PWR	2A DUAL H-BRIDGE MOTOR DRIVER	30.00
	LM3940IMPX-3.3/NOPBCT-ND	LM3940IMPX-3.3/NOPB	IC REG LINEAR 3.3V 1A SOT-223-4	30.00
	AP63205WU-7DICT-ND	AP63205WU-7	IC REG BUCK 5V 2A TSOT23-6	30.00
	311-62ARCT-ND	RC0805JR-0762RL	RES 62 OHM 5% 1/8W 0805	30.00
	CSR0805FKR250CT-ND	CSR0805FKR250	RES 0.25 OHM 1% 1/4W 0805	30.00

	RMCF0805FT30K0CT-ND	RMCF0805FT30K0	RES 30K OHM 1% 1/8W 0805	30.00
	A113320-ND	282837-2	TERM BLK 2P SIDE ENT 5.08MM PCB	150.00
	SRN6045TA-4R7MCT-ND	SRN6045TA-4R7M	FIXED IND 4.7UH 4.5A 26 MOHM SMD	40.00
	S6103-ND	PPTC051LFBN-RC	CONN HDR 5POS 0.1 TIN PCB	30.00
	3147-B1701UYG-20D000114U1930CT-ND	B1701UYG-20D000114U1930	LED YLW-GRN DIFFUSED 0805 SMD	30.00
	1727-5451-1-ND	BAT160S,115	DIODE ARRAY SCHOT 60V 1A SOT-223	30.00
	P15392CT-ND	EEU-FR1V471LB	CAP ALUM 470UF 20% 35V RADIAL TH	30.00
	1276-1274-1-ND	CL10A226MP8NUNE	CAP CER 22UF 10V X5R 0603	30.00
	C0603C104J4RACTU	C0603C104J4RACTU	0.1 µF ±5% 16V Ceramic Capacitor X7R 0603 (1608 Metric)	30.00
	CL10B105KA8NNNC	CL10B105KA8NNNC	1 µF ±10% 25V Ceramic Capacitor X7R 0603 (1608 Metric)	35.00
	641-1707-2-ND	CDBA540-HF	DIODE SCHOTTKY 40V 5A DO214AC	30.00
	1965-ESP32-WROOM-32-N4DKR-ND	ESP32-WROOM-32-N4	RF TXRX MOD BLUETOOTH WIFI SMD	750.00
	Case	NA	Aluminum Base	50.00
	Omni Wheels	NA	Omnidirectional Wheels	500.00
	DC Motors	Greartisan DC 12V 300RPM	Greartisan DC 12V 300RPM Gear Motor High Torque Electric Micro Speed Reduction Geared Motor Eccentric Output Shaft 37mm Diameter Gearbox	1000.00
	Neodymium Magnets	MIN CI Super Strong Neodymium Magnet Bar	MIN CI Super Strong Neodymium Magnet Bar, 40 X 10 X 3 mm Powerful Rare Earth Magnets Strip Heavy Duty	300.00

	PCBs	PCB	PCBWay	80.00
TOTAL	TOTAL	TOTAL	TOTAL	3,325.00
TOTAL PER DEVICE	TOTAL PER DEVICE	TOTAL PER DEVICE	TOTAL PER DEVICE	$33.25