[Durian Antweight Battlebot]

By

[Matthew Jin]

[Tim Fong]

[Ved Tiwari]

Final Report for ECE 445, Senior Design, [Spring 2026]

TA: [Zhuoer Zhang]

[6] [May] 2026]

Project No. 11

Abstract

This report details the design, implementation, and holistic experience of creating this particular Antweight Battlebot. The Durian Antweight Battlebot was designed to compete according to the National Robotics Challenge rulebook. It features a 3D printed assembly, active and passive weaponry, a drivetrain, and the use of an IMU sensor plus ESP32 microcontroller combo. The information provided within this document includes, but is not limited to dimensioning diagrams, 3D model screenshots, block diagrams, financial details, ethics, room for improvement, and technical information (part datasheet/implementation/subsystem). Following the completion of this document, the reader should be able to discern the successes and failures of Group 11's experience. Details about uncertainties and RV tables will be provided toward the bottom of the report to provide further insight into what challenges and triumphs members of this team experienced – individually or collectively.

Contents

1. Introduction

The engineering problem at hand was the design target of an NRC-rulebook abiding Battlebot with the capability of dominating its competitors. To elaborate, the Battlebot must not exceed 2-lb, cannot include metal in its weaponry/armor, cannot have a signal jamming feature, and must include the usage of a microcontroller plus sensor combination. The usefulness of this system can be broken down into its subsystem capabilities: the microcontroller, the mechanical, the electrical, and the safety/reliability. These will be further explored in detail on how they contribute toward the Battlebot's lethality, maneuverability, durability, and combat-related features. The following sections provide in-depth analysis and information regarding the aforementioned subsystems, validation conducted, and cost. Most importantly, the main conclusions (good and bad) of this project experience will be emphasized within the last section.

1.1 Block Diagram

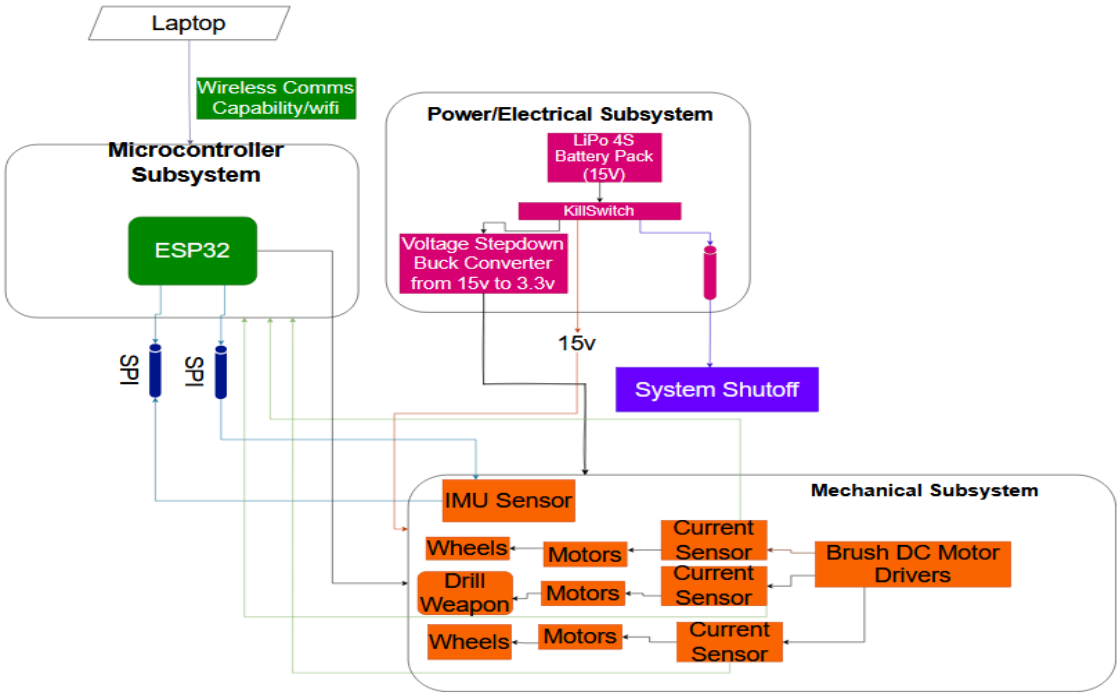

Figure 1: Overarching block diagram that incorporates the internal and external interconnections of the three primary subsystems. Note how the fourth subsystem, safety/reliability, is not explicitly labeled but is instead passively present.

2 Design

The initial design for this Battlebot vastly differed from what the outcome was, particularly from the mechanical perspective. What was a four-wheel drivetrain system was reduced to a two-wheel drivetrains system with a rear skid for weight considerations. What was a lone turtle shell was turned into a shell + centralized chassis layout – what will later be referred to in this document as the "two-castle defense system" (courtesy of Gregg from the ECE machine shop for this terminology). Furthermore, the finalized PCB dimensions and part selection choices changed all four subsystems thoroughly.

2.1 [Mechanical Subsystem]

The mechanical subsystem holds a vastly important purpose in the context of the Battlebot competition. Not only is its functionality integrated intimately with the microcontroller and electrical subsystems, but it is the physical, real-world output of all subsystems. There are four overarching parts to this subsystem. Note that PETG is the 3D printing material used for the shell, chassis, and weaponry. This was chosen due to its mixed features of solid shock absorption capabilities, non-brittle qualities, density (weight consideration), and cost. All dimensioning decisions were made with the fact that the Bambu 3D Printers used within ECEB2070 had some level of dimensioning inaccuracy tolerance (~0.1-0.2mm), This proved especially relevant when designing the inverted lip corners of the chassis halves and the shell+chassis conjoinment window.

2.1.1 [Drivetrain]

To begin with, the drivetrain is the most rudimentary from a functional perspective. A Battlebot cannot hope to compete without mobility, and the best means of attaining this was argued within Group 11. Initially, a 4-wheel drivetrain was proposed due to its versatility and stability. But this proved to be too daunting of a challenge when factoring in the 2-lb weight limit. Instead, a 2-wheel drivetrain with a rear skid was proposed (and accepted by all members). The rear skid was specifically designed as a half-circle rather than a rectangle. This was done deliberately to reduce the surface area of the skid that made contact with the floor, thus reducing the amount of friction (and thus the minimum torque required to overcome this friction) the gearmotors would have to deal with. Several design considerations had already been agreed upon to reach one decision – we would use energy-dense Brushed DC gearmotors that had a 3mm diameter driveshaft. Brushed DC motors were reliable, great from a weight perspective, and were compact enough to fit within a chassis quite easily. Moving onto the wheels, the initial selection was incorrect for our use case. Polyurethane was deemed to be the best material match for the competition setting's floor, and a 3mm diameter axle was necessary in order to be compatible with the gearmotors selected. But the wheels chosen were for an RC place use-case, and did not provide metal wheelhubs that were necessary in fastening the gearmotor securely to the wheels. After several conversations with the machine shop (some specifically regarding the wheels, others regarding the other components of the mechanical subsystem), Gregg offered to custom-make the wheels and metal gearhubs, pictured below. An M.3 screw was used to hold the gearmotor driveshaft in-place.

Figure 2: The polyurethane wheel + metal wheelhub + gearmotor assembly is depicted.

2.1.2 [Chassis]

In order to accommodate for the assembly, accessibility, configurability, and dimensioning requisites of the components that would go inside, the chassis needed to be sized correctly and possess the correct features. Initially, a centralized chassis was not proposed – rather an all-in-one shell+component housing unit was thought of. But this proved to be extremely difficult from a 3D modeling and weight consideration vantage point after conversations with Gregg and fellow teammates. What is reflected in the final design revisions is a two-castle defense system: a centralized chassis + an outer shell. This allowed for the chassis to be printed with a lighter infill percentage (as it was not anticipated to have close proximity to blunt force) compared to the shell. 60% was decided upon among members of Group 11. Furthermore, to account for the aforementioned accessibility and configurability, the shell was split into halves, where inverted lip corners and zip tie terminals were included to allow easy installation/removal. Below depicts the layout of the components that would go on the bottom half of the chassis.

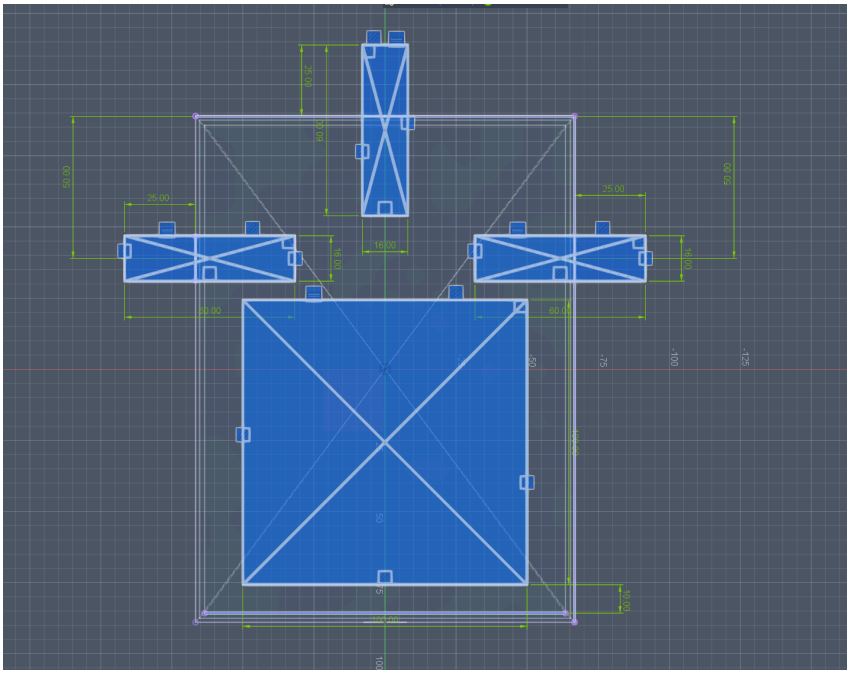

Figure 3: Bottom chassis component layout (Three gearmotors and the PCB).

Several other important functionalities needed to be featured in the chassis design. To make things straightforward when wiring and promote dimensioning feasibility, the battery was to be stored on the upper chassis compartment. This would be done securely through the utilization of a zip tie “hammock”. Not only was this thought to have made it significantly easier to separate high voltage wiring from low voltage wiring, but this provided a way for the battery module to avoid damage incurred by shock absorption. Having the hammock just loose enough to allow for floating wiggle room achieves this. Finally, in order to secure the gearmotors to the chassis walls, an M.2 face mounting pattern was utilized. It was later found that M.2 screws do not thread well at all with 100% infill PETG material, let alone 60%. Superglue needed to be used as a supplement to achieve the stability desired.

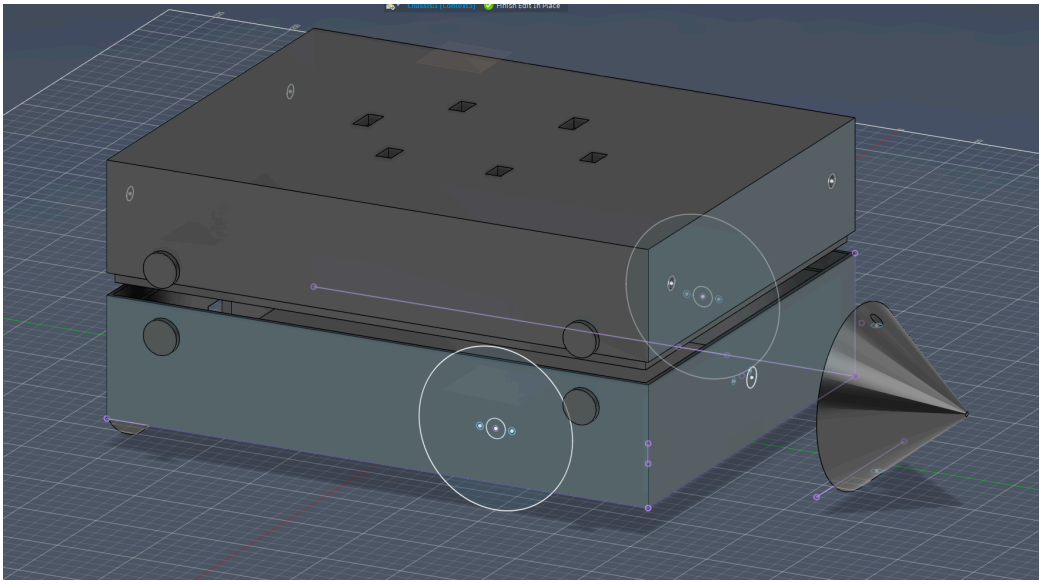

Figure 4: Showcases the features of the chassis (halve conjoinment, M.2 face mounting pattern, hammock cavities).

2.1.3 [Weaponry]

Several weapon types were debated over regarding their feasibility, ease-of-implement, and weight contribution to the overall design. Between hammers, drills, spinners, and flippers, the drill stood out as the best choice for its qualities of being lightweight, straightforward, and easy to design. The primary challenge faced with the drill was its retention methodology. It was known that a 3mm-diameter, 25mm-length gearmotor driveshaft would need to be inserted into the center of the drill's flat-face. But retaining it was an entirely separate issue. Gregg provided valuable insight into how to go about this, and a two-sided screw retention approach was decided on. Essentially, two M.2 screws would go through a 6.35mm pilot (guiding) cavity that was 3D printed. Then, in the ECEB Machine Shop, ~2mm threadholes would be machined into the drill for the M.2 screws to actually clasp. This would "pinch" the gearmotor driveshaft into place, preventing it from releasing under high inertia/bluntforce propagation operating conditions. The volume of this drill was small enough to permit 100% infill. This also slightly improves the threading capabilities of the M.2 screws on PETG filament, which was previously mentioned to be subpar.

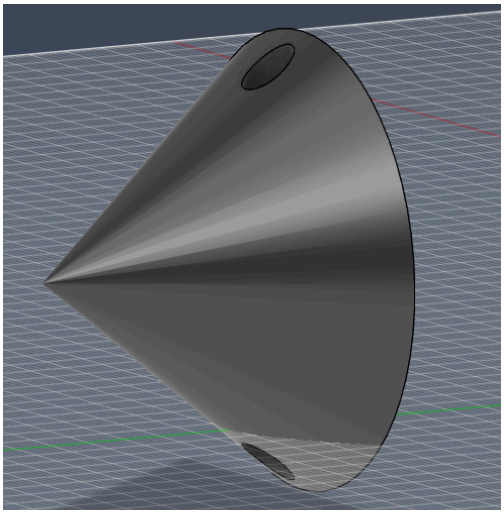

Figure 5: The drill, featuring its 6.35mm pilotholes leading into 2mm threadholes.

2.1.4 [Shell]

The shell wraps up the mechanical subsystem, and it includes one of the most defining features of the Battlebot design. It is a spiked-shell, hence the Durian within the project title. This was done to incorporate some level of deterrence against hammer-type Battlebots: their frequency within our competitors was substantial enough to warrant this.

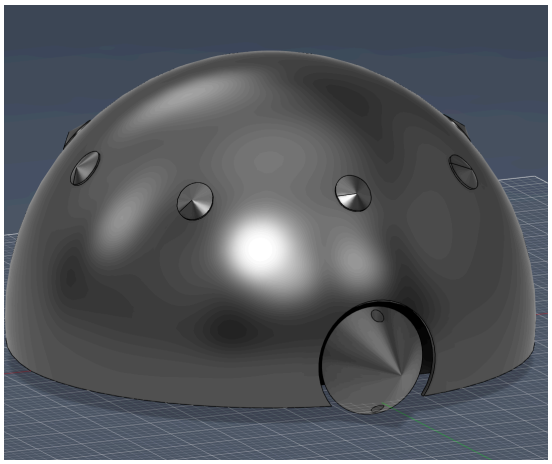

Figure 6: The shell, with a front-facing drill cavity and passive shallow spikes on the dome for hammer deterrence.

Infill was proposed to be 80%, as it was expected to need additional shock absorption capability, while its thickness was set to 2.5mm to avoid being too brittle. However, the shell became difficult to design beyond the dome shape. The shell needed to, one, leave clearance room for the wheels to fit inside, and two, have a way to fasten onto the centralized chassis securely. Depicted below is a cleverly incorporated 3mm thick plate with a window that allows just enough clearance room (~0.2mm) for the roof of the upper chassis to fit through. With that, the mechanical subsystem design process concludes.

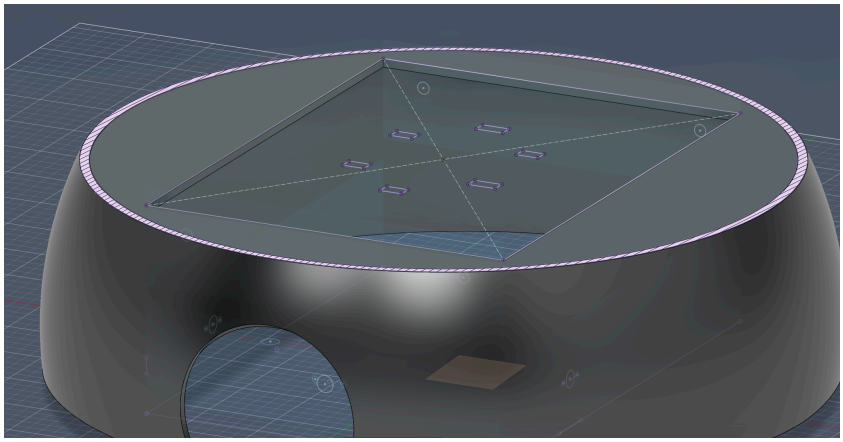

Figure 7: The fastening "window" method between the drill and chassis.

2.2 [Electrical Subsystem]

The electrical subsystem consisted of two step-down voltage converters for a USB connection and the LiPo battery connection. For the USB-C connection, according to a Texas Instruments USB 2.0 product sheet, there should be an expected 5 volts across the VBUS line that is being sent from the laptop the connection is hooked up to [1]. An LDO (AMS1117-3.3) is used to step down the 5V from the host laptop into an appropriate 3.3V to power the ESP32 microcontroller. For the 14.8 V battery, a buck converter (LM2596S-3.3) was used to step down the voltage into a 3.3V to power the microcontroller and motor controller logic sleep lines. There was also a need to add an accessible kill switch that was placed on the connection between the LiPo battery XT30 connector and a current-limiting resistor to ground. Status LEDs connected to current limiting resistors were added to indicate whether these circuits were functioning properly. Ultimately, the PCB failed due to unprotected voltage lines to the ESP32 microcontroller, which was ultimately caused by not protecting the 3.3V line.

One of the main causes was the inrush current supplied from the motor controllers, where possible current spikes could result in too much current that would cause the 3.3V pin on the ESP32 to be overwhelmed and heat up. Another issue was due to not protecting the current-sensing GPIO lines feeding into the microcontroller. The motor drivers feature a CS line that supplies voltages that feed into the ADC pin lines. There was no current-limiting resistor on those lines, which resulted in overvoltages above 3.3V on the GPIO pin lines, causing smoke to come out of the ESP32 microcontroller chip. This resulted in having to switch to a breadboard set up to complete functional objectives. More details will be provided in the Electrical Verification section. PCB layout and overall schematic can be found in Appendix B of the report.

There are concerns over the current draw of the system with respect to the current that is going to be needed from the system overall. Motors will most likely take up most of the current draw coming from the system as a whole. We need the battery to be able to support the electronics of our robot for the full 2 minute duration of a match in the competition. As a result, we want to estimate the total current draw and make a determination on whether the chosen battery will have enough current to supply the whole system at the max current draw each component would have. The buck converter will have a max current draw of 7.5 A[8]. The 3 brushed DC Motor Drivers will have 3.5 A max current draw for a total of 10.5 A of maximum current draw[6]. The IMU sensor has a 100 mA maximum current draw. The ESP32 will have a maximum current draw of 700 mA including the wifi communication[7]. In total this adds up to 18.80 A of total current draw that the battery is facing.

Using the following formula using the battery capacity to get amount of time chosen battery can run:

T in minutes = (850mAH / total peak current draw of Bot) * 60

T in minutes = (850 mAH / 18.80 A) * 60 = 2.71 minutes approximately\

This is assuming all components are running at maximum current draw but can see that this battery can at maximum run all components beyond the 2 minute battle duration in the competition. This estimation is a pessimistic one as a result of assuming all components are running at max current.

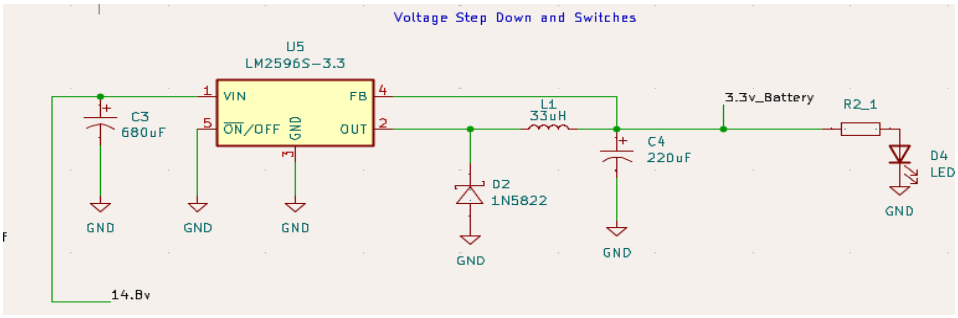

Figure 8: Schematic of Buck Converter 15v to 3.3v Step Down Circuit

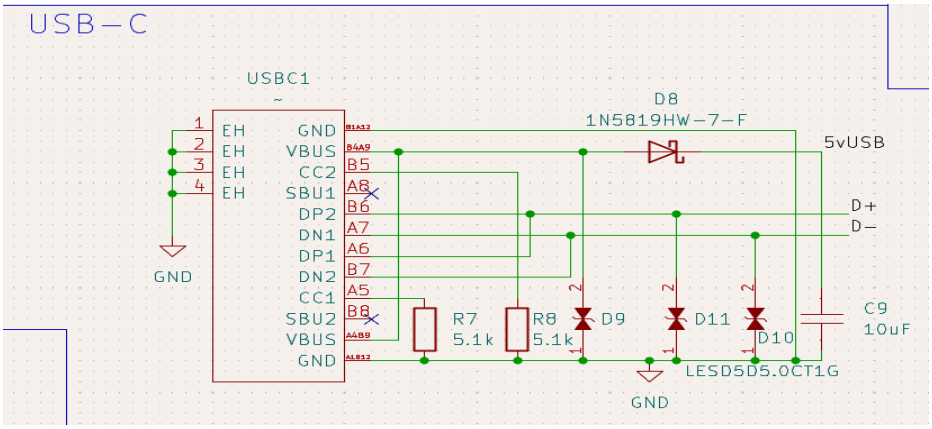

Figure 9: Schematic of USBC Connectors

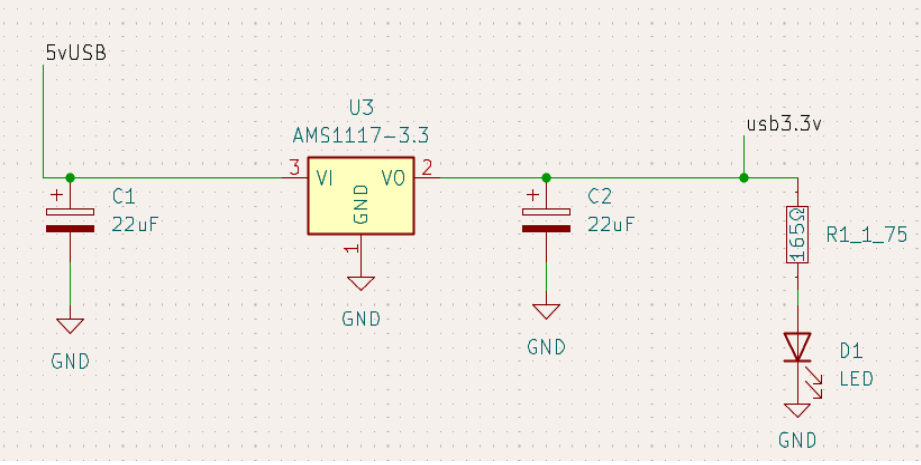

Figure 10: Schematic of LDO 5V to 3.3V Stepdown Circuit

2.3 [Microcontroller Subsystem]

The microcontroller subsystem in this project serves as the key control center of the entire battlebot. The reason for a microcontroller in our project is to facilitate all software and control related behaviors of the system as a whole. The microcontroller is also responsible for relaying the sensor data from the IMU to the GUI that we initialize and interact with through the web-server. This section will break down the technical aspects of the microcontroller subsystem, including pieces of code, communication strategies, etc.

2.3.1 [Technical Software Strategy]

In our system, we settled to use WiFi over Bluetooth (two features native to our ESP32-C6 configuration) as it offered better metrics when considering the tradeoffs between distance/range, power consumption, and reliability. In our system, we deemed the highest importance for our system to be reliable, low latency, and high range. Our LiPo battery that we plan to use virtually eliminates the concern for power consumption as the nature of the ESP32-C6 is quite power efficient at this scale. Our strategy was to use the ESP32 as a websocket (router). This allowed us to host a traditional HTML/JS/CSS GUI that would be familiar with all users. When the user visits the Microcontroller server's default IP address (http://192.168.4.1/), they are greeted with the control center (see figure 9).

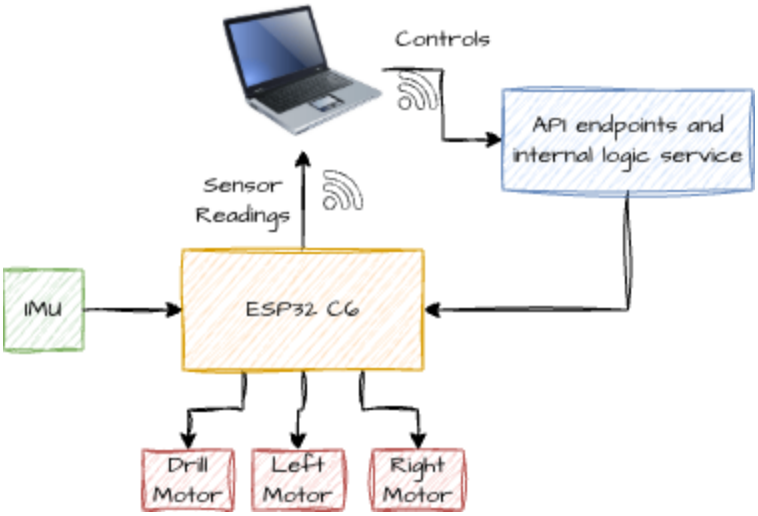

Figure 11: Control flow diagram of microcontroller and communication system

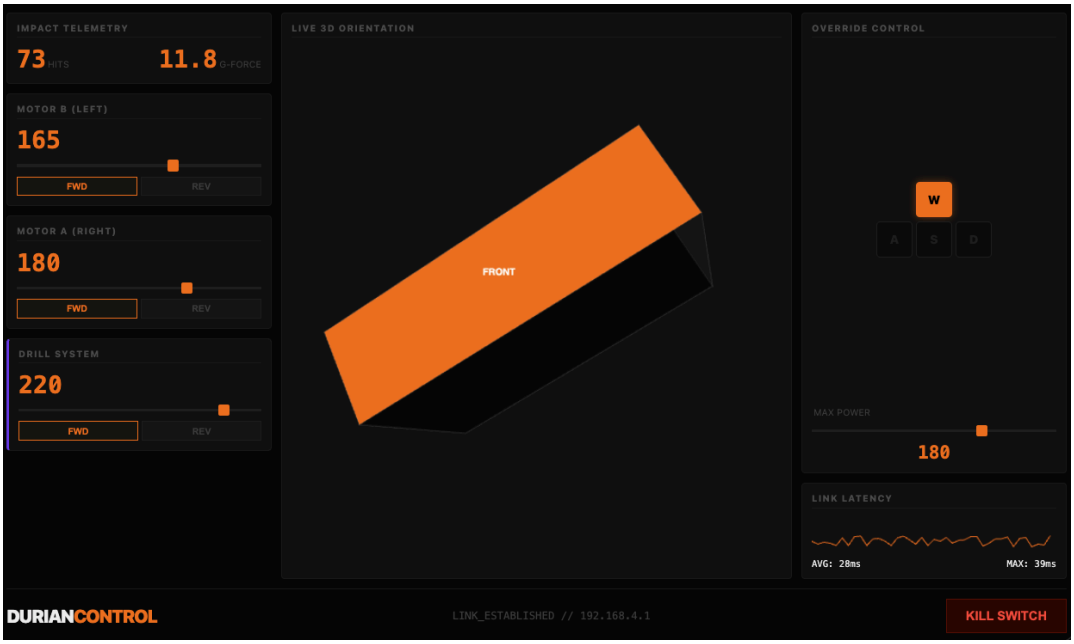

Figure 12: Snippet of control center from the host perspective

2.3.2 [API Layout and motor interactions]

Every action that the user intends to send to /read from the bot is governed by the API suite. This is a collection of API endpoints which change internal state variables, send GPIO signals, or show sensor data. For example, when the user presses the “W” key on the keyboard, an API call to the /drive endpoint is made. On the server side, this triggers a signal through the internal logic to trigger the appropriate GPIO pin(s) to make the respective motors move.

The motors themselves require a PWM signal to match the speed with. In this case, the PWM signal is generated on the server side, and when the user decides to change the speed of a particular motor, they can adjust a slider on figure 9. When the value of the slider updates, a call to the /set API endpoint is made, and the PWM frequency on board is adjusted accordingly.

3. Design Verification

This section encompasses how all subsystems fared when tested in the field to have its features and capabilities evaluated. This section will primarily show imagery and a brief overview of the requisite test to avoid unnecessary redundancy, as the RV table in Appendix A already provides in-depth descriptions.

3.1 [Mechanical Subsystem]

Several facets of the mechanical system were validated through the full-power two minute dyno test, with the results shown below. This was done because the matches are approximately two minutes long.

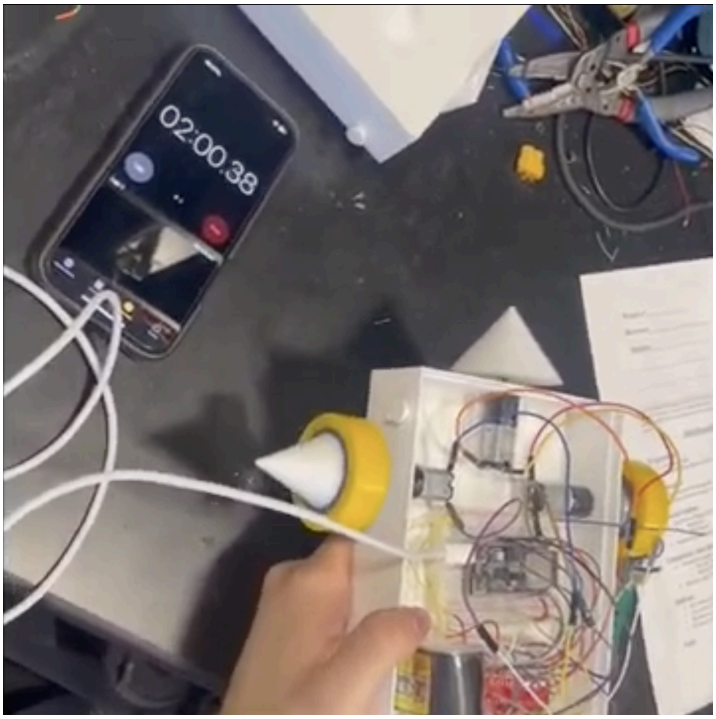

Figure 13: The fastening "window" method between the drill and chassis.

3.2 [Electrical Subsystem]

The killswitch was added on to the side of the robot and could be turned on and off with relative ease by switching up to supply battery voltage and switching down to shutdown robot subsystems. There is no requirement from the operator to do anything but flip the switch that is easily accessible on the bot so it shouldn't take more than a few seconds on average to switch off all physical subsystems. The LDO for the USBC connector read 3.29268v after probing with a multimeter between the ground test point and the LDO output confirming functionality of it within the 10 percent tolerance (image of reading can be found in Appendix C). The buck converter outputted less than a 1v or in some mV range suggesting possible issues regarding the circuitry. The first issue was that not enough current was flowing between the high voltage 15v line and the buck converter due to using 0.2mm trace widths instead of something much wider. Another possible issue is possible soldering mistakes made while soldering the buck converter IC or other SMD components in the circuitry.

For the ADC pins and current sensing, a multimeter was measuring those specific ADC pins on the ESP32 microcontroller, which read 3.3v. This reading suggests that based on the motor driver website with a current sense of 2.5V/A there would be an output current of 1.32 A going into the GPIO pin [6]. The ESP32 s3 WROOM module data sheet the GPIO pins can only handle a maximum of 40mA so this proves the pins were being overloaded with current and overheating the chip[7]. Based on the two minute dyno test the battery was able to withstand a 2 minute run with all motors (Weapon and two wheel motors) running which is the bulk of the current requirements of the battlebot.

A breadboard version of the bot which included motor drivers and using a ESP32 support board to connect everything with a killswitch was used instead for the competition. The breadboard version can be viewed in Appendix D.

3.2 [Microcontroller Subsystem]

The main testpoints for this project are the power consumption and the latency for transmitting signals to and from the WiFi Connection. To test the latency, there is a timer that's initiated and started from the time that an API call is made to the time that a 200 response code is returned back to the host. This timer is then plotted along a time-latency graph so we can measure any visible spikes in latency. A global max timer is also tracked to ensure that we stay consistent with our requirements that were outlined in the beginning of the semester.

4. Costs and Schedule

4.1 Parts

Table 1 Parts Costs

Part	Manufacturer	Retail Cost ($)	Bulk Purchase Cost ($)	Actual Cost ($)
ESP32-S3-WROOM-1	Espressif Systems	3.28	16.4	16.4
ISM330 IMU Sensor	Adafruit	19.95	19.95	19.95
JLCPCB PCB	JLCPCB	15.40	77	77
B0BZ8GQSTB Battery	Tattu	21.50	42.99	42.99
680uF Capacitor	Panasonic Industry	1.29	12.90	12.90
220 uF Capacitor	Nichicon	0.338	3.38	3.38
40V 1A Schottky Diode	Diodes Incorporated	0.20500	2.05	2.05
XT30U-M Battery Connector	LCSC Electronics	0.228	2.28	2.28
3mm BDC Gearmotor	Itgresa Robotics	19.00	57.00	57.00
3.3V Buck Converter IC	Texas Instruments	6.97	34.85	34.85
33UH SMD Inductor	VANXY	0.327	3.27	3.27
3.3V Linear Regulator IC	Shenzhen Slkormicro Semicon Co	0.05400	1.08	1.08
USB-C Jack Connector	Same Sky	0.323	3.23	3.23
Polyurethane Wheels	ECE Machine Shop			
Metal Wheelhubs	ECE Machine Shop			
DRV8876 Gearmotor Drivers	Polulu	8.15	24.45	24.45
Parts Total				**300.83**

4.2 Labor Cost Estimate

An estimated hourly rate of $50 per hour and the following breakdown of estimated time for each division work:

- Mechanical Design (20 hours)
- Electrical Design (30 hours)
- Software Development (20 hours)
- Electrical Manufacturing (7 hours)
- Mechanical Manufacturing (15 hours)

Total labour Cost = Total Time of Labor * $50/hour = $4,600

4.3 Schedule

Week of	Task	Persons
March 6, 2026	Receive requested parts Continue working on the 3D CAD models (chassis, drill, shell).	All members of group
March 13, 2026	Gather microcontroller, and start prototyping code for microcontroller. - Set up websocket - Experiment with WiFi connectivity and study latency thresholds as defined in the requirements Solder PCB components and begin testing PCB. Begin 3D printing copies of the 3D CAD models. Start mechanical tolerance testing.	Ved (Microcontroller) Tim(PCB/Hardware) Matt(Mechanical & Hardware Support)
Spring Break	Get motors and sensors to start collecting data and sending it back to client through WiFi Determine if PCB needs another order and work on ordering another if necessary. Continue mechanical tolerance testing.	Ved (Microcontroller) Tim(PCB/Hardware) Matt(Mechanical & Hardware Support)
March 27, 2026	Design GUI and setup payload skeleton for interpreting keystrokes and sending through wires	Ved (Microcontroller) Tim(PCB/Hardware) Matt(Mechanical & Hardware Support)
April 6, 2026	Progress Demo	Everyone

April 13, 2026	Last Minute Adjustments	Everyone
April 20, 2026	Mock Demo	Everyone
April 27, 2026	Final Demo/	Everyone
May 4, 2026	Work on Final Paper/Competition	Everyone

5. Conclusion

Participation in this project has transformed each and every member of Group 11 into a more refined engineer. This experience included a great deal of learning from numerous facets of engineering, including but not limited to: soldering, PCB fabrication, system integration, software development, and 3D modeling/printing. The following subsections document what was done correctly and incorrectly, along with considerations of ethics and future improvements for this project scope.

5.1 Accomplishments

Throughout the development of this project, our biggest accomplishment was to be able to participate in the battlebot competition as a whole. Given the strict ruleset, there were many difficult rules that we had to comply with to ensure that we weren't disqualified before participation. Furthermore, equally if not more importantly, we learned skills that were quite helpful to learn in a real world project setting including soldering, PCB design, 3D printing/CAD, documentation, and teamwork.

5.2 Uncertainties

For context, before the demo, we had the misfortune to somehow short our microcontroller subsystem when powering the PCB. As we don't have the resources to debug and understand the root of the problem at the specific level, we do have some leading hypotheses that would be worth considering for the sake of understanding and learning from the mistake. The biggest and most probable issue that we think may have occurred is the mistake of leaving the main 3.3V line unprotected. At the time, we were unfamiliar with what's known as inrush current. As discussed earlier, one of the preventative measures that we could've exercised in order to avoid burning our microcontroller is to decouple the 3.3V line with capacitors. The reason for this is because it would effectively soak up all sudden surges of current that's demanded by the motor drivers as a result of inrush current.

5.3 Ethical considerations

Referencing both the IEEE Code of Ethics and the National Robotics Challenge Manual, no breaches of rules have occurred during the numerous phases of this project. Each engineer involved remained committed to meeting the requisites outlined by the project scope. But this was done with careful considerations of competitional safety, fairness, and an effort to earnestly learn throughout this process. Respect for one another was maintained throughout the semester, and features such as the de-energization killswitch and inrush current countermeasures were implemented with safety in mind. Furthermore, the engineering problem tackled within this scope can be extrapolated to reliability systems pertinent to other robotics applications. This includes surgical and/or healthcare companion robotics.

5.4 Future work

Though our demo contained all of the functional features that we tried to outline in the proposal document, the most important improvement we can make with our work is designing a working PCB for our system as a whole. As it stands now, our PCB is missing many essential traces that are essential for the functionality of our microcontroller, along with other issues discussed above that we'd like to take time to debug and understand where we could improve our PCB. This would give us the ability to stray away from using a breadboard to demo key functionality, and rather use a PCB that contains all of the key aspects of our design and idea.

References

[1] J. Bearfield, “MSP Power Supply Products USB Universal Serial Bus.” Available: https://www.ti.com/sc/docs/products/msp/intrface/usb/pwrdist.pdf

[2] *Repeat Robotics Brushed Drive Gearmotor*. ItGresa Products: Robotics and Small Business Support. (2026, May 1). https://itgresa.com/product/repeat-robotics-brushed-drive-gearmotor/

[3] *Contest Manual: The National Robotics Challenge*. (2026). https://www.thenrc.org/contest-manual

[4] IEEE Code of Ethics | IEEE. (n.d.). https://www.ieee.org/about/corporate/governance/p7-8

[5] Dunn, K., & Vker. (n.d.). *)Tattu 850MAh 4S 95C 14.8V R-line LiPo Battery Pack (XT30 Plug*. GensTattu.https://genstattu.com/tattu-r-line-850mah-14-8v-95c-4s1p-lipo-battery-pack-with-xt30-plug.html?srsltid=AfmBOoo1iqVx5if7gNQHki-hlWl6huwLnCxXOvmVmY63FdSSLyldP2J8

[6] “DRV8876 Single Brushed DC Motor Driver Carrier,” *Pololu.com*, 2026. https://www.pololu.com/product/4036

[7] “Espressif Documentation,” *Espressif.com*, 2025. https://documentation.espressif.com/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf

[8] “LM2596 SIMPLE SWITCHER ® Power Converter 150-kHz 3-A Step-Down Voltage Regulator Typical Application LM2596,” 1999. Accessed: May 07, 2026. [Online]. Available: https://www.ti.com/lit/ds/symlink/lm2596.pdf?HQS=dis-dk-null-digikeymode-dsf-pf-null-wwe&ts=1778046211706

Appendix A: Requirement and Verification Table

Table 2 System Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
1. The gearmotors should be able to withstand 2 minutes of nonstop, maximum output conditions. a. Must withstand an ambient room temperature operating condition with +- 5 Celsius tolerance b. Must withstand inertial turbulence	1. A dyno test was run with a two-minute timer to validate this requirement and its subrequirements.	Y
2. The drill and wheels need to be easily replaceable.	2. M.2 and M.3 screws were repeatedly removed/installed using hex keys to attach/detach the drill and wheels to the gearmotor driveshafts.	Y
3. The inner components must be easily accessible for repair/configuration purposes.	3. The chassis halves, joined together using zip ties and a lip conjoinment design, were repeatedly removed and attached with ease. Modifications to the centralized components were easily performable.	Y
4. All system components are designed to operate together in a room-temperature environment with a plus-minus tolerance of 5 Celsius.	4. This was passively tested with the two-minute dyno test within ECEB2072, which was an operating setting slightly above room temperature.	Y
5. A stop switch that can be activated to stop the robot from operating and the robot must be shut off in 60 seconds or less.	5. The easily accessible killswitch placed on the side of the bot made for switching on and off the bot in only a few seconds	Y
6. Voltage Regulator can supply a stable input voltage of 3.3v with a ±2% tolerance for microcontrollers.	6. Placing a multimeter between ground test point on PCB and IC output to read voltage values.	Y
7. Buckconverter can supply a stable input voltage of 3.3v with a ±2% tolerance for microcontroller, IMU, and current sensor voltage input.	7. Placing a multimeter between ground test point and output test point to read voltage values.	N

8. The battery should be able to supply enough voltage for the 2 minute duration of the competition.	8. Running a two minute dyno test with all motors running at maximum speed and measuring battery voltage afterwards.	Y
9. Latency from input to output via the websocket should be under 1 second. With a tolerance of 20ms	9. This can be measured by tracking the elapsed time between sending the payload through the websocket and receiving the acknowledge bit from the microcontroller	Y
10. Memory and CPU power should consistently be well under threshold for microcontrollers. Specifically, RAM usage should be < 2MB with tolerance of 30kb	10. Can track critical system benchmarks like the stack/heap and report back to the host device. Upon exceeding these numbers potentially can reset the system as a whole to reset all cache.	Y
11. Code must have >80% test coverage for system critical functions.	11. Tests can be written using a framework natively supported in C such as μnit. Running the test suite summary yields the percentage of main function code that's covered.	Y, though tests were implementation tests, not integration

Appendix B: PCB Overview and overall Schematic

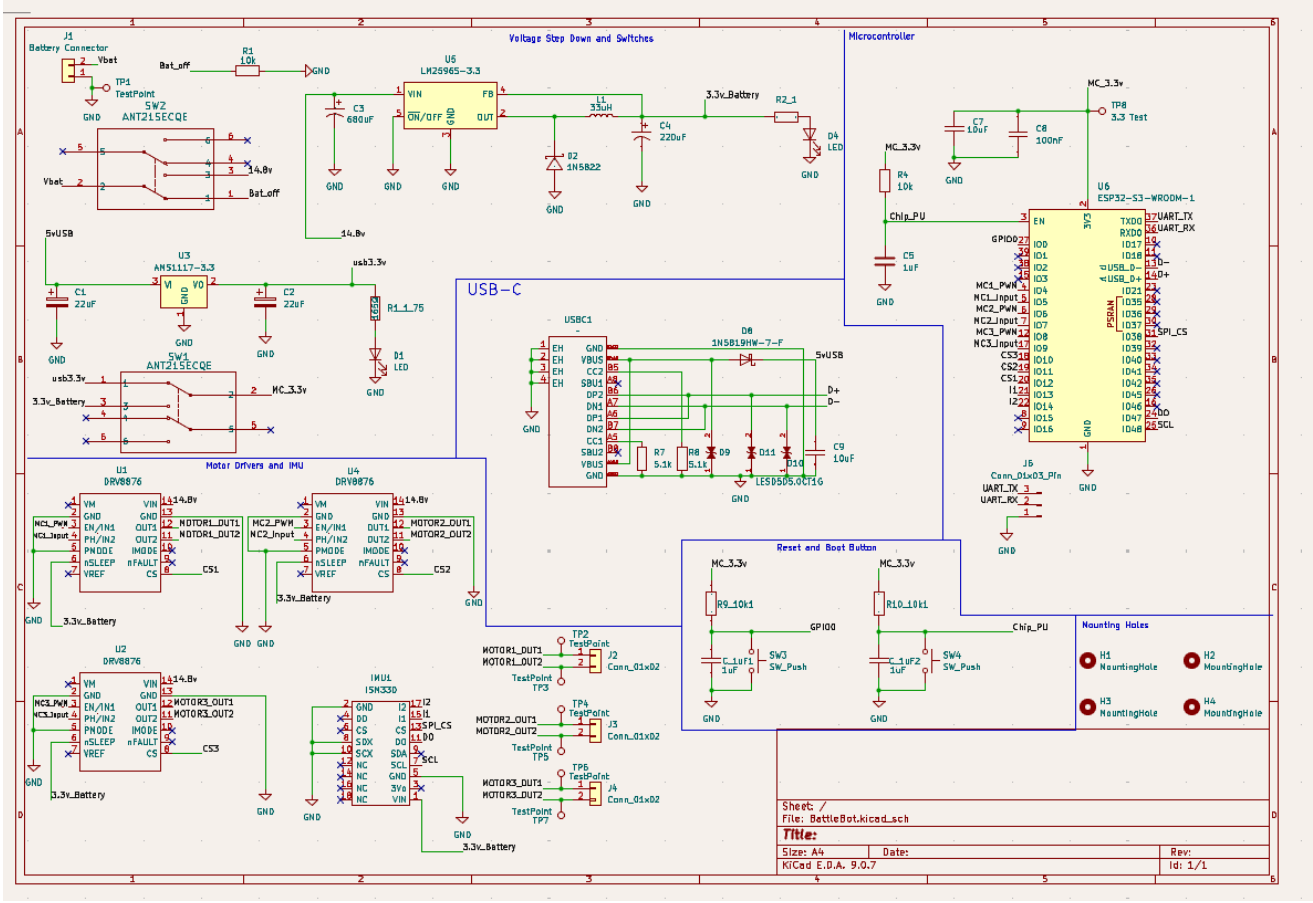

Figure 13: Overall Schematic Overview for PCB

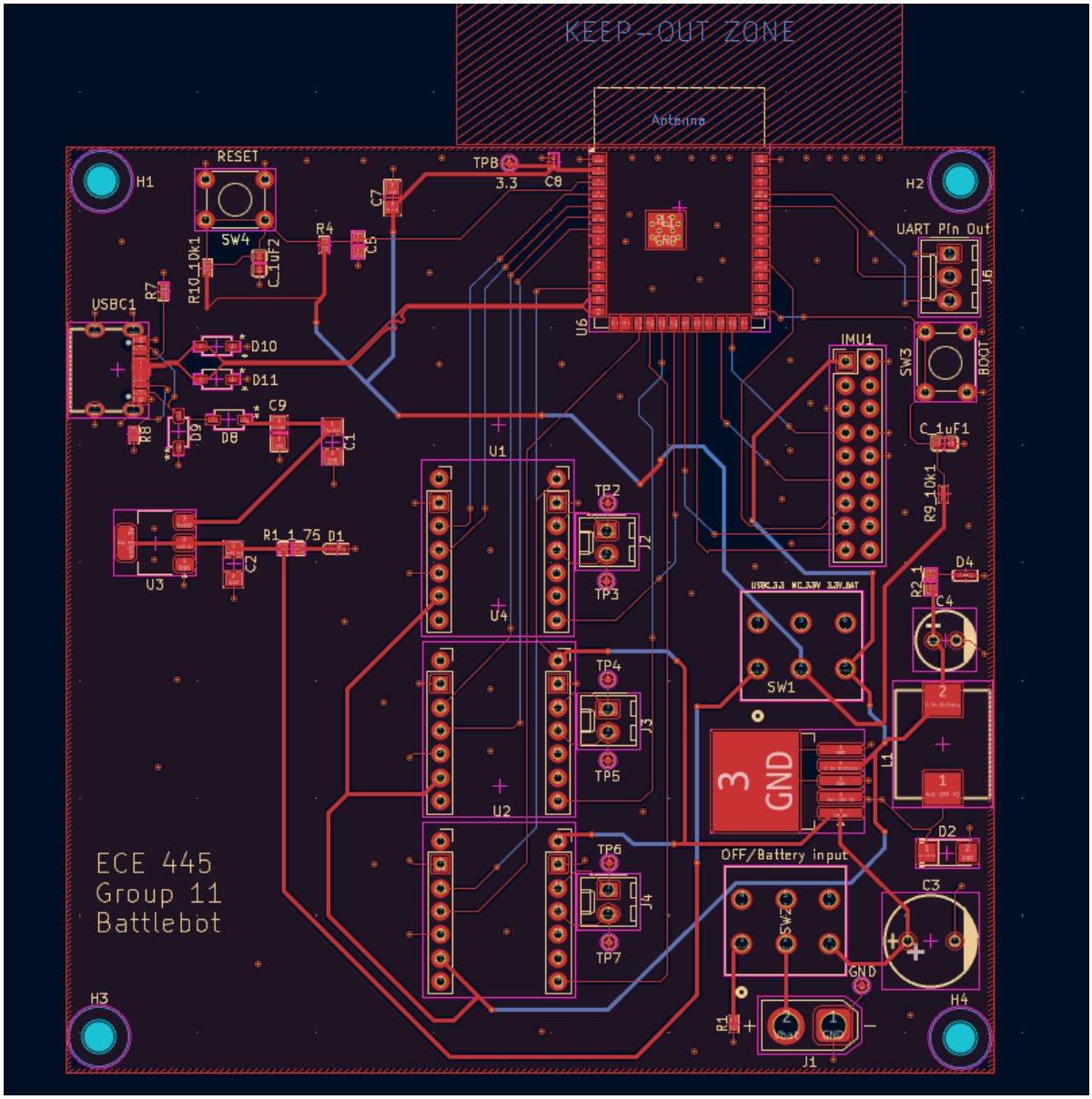

Figure 14: PCB Editor View

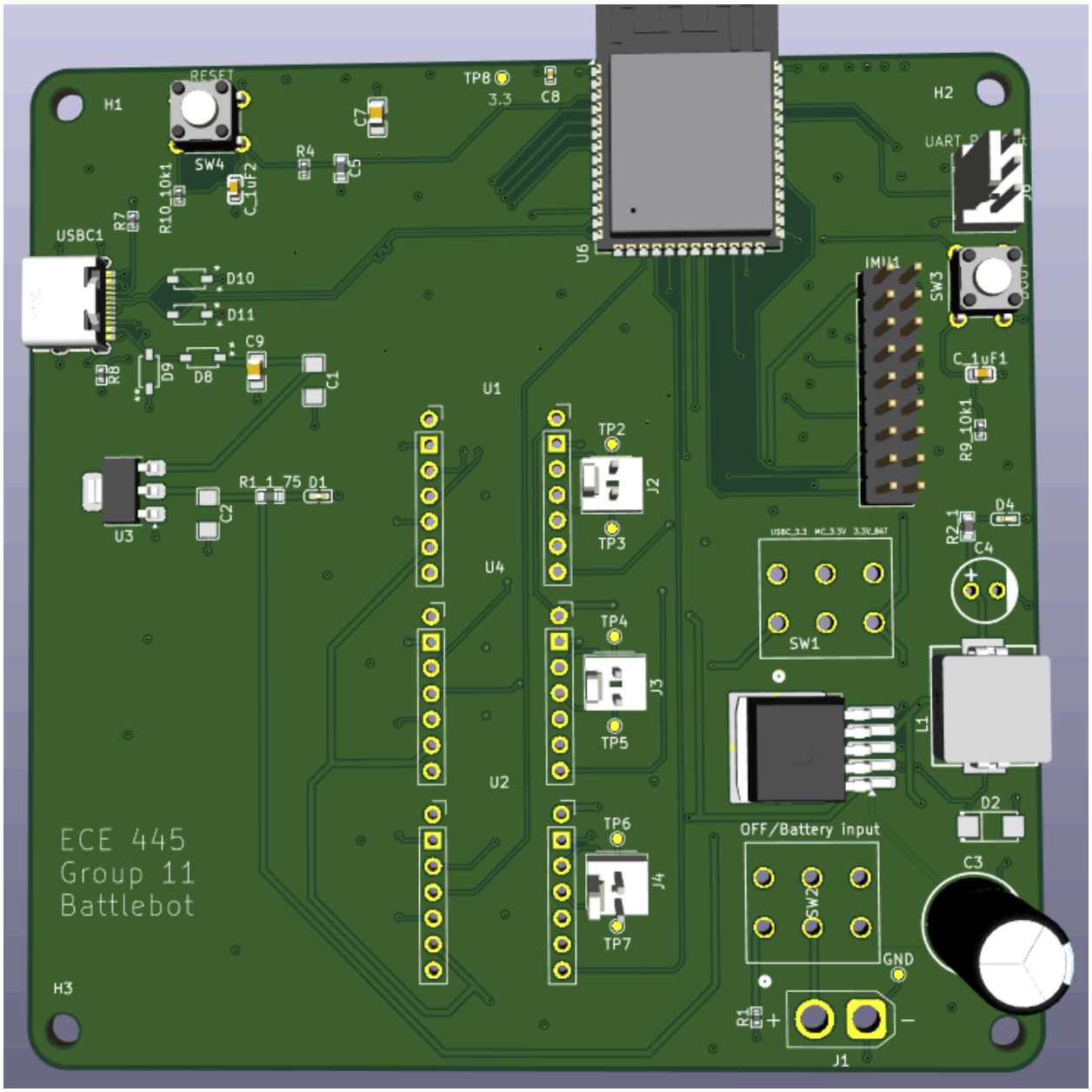

Figure 15: 3D Image of PCB Rendering

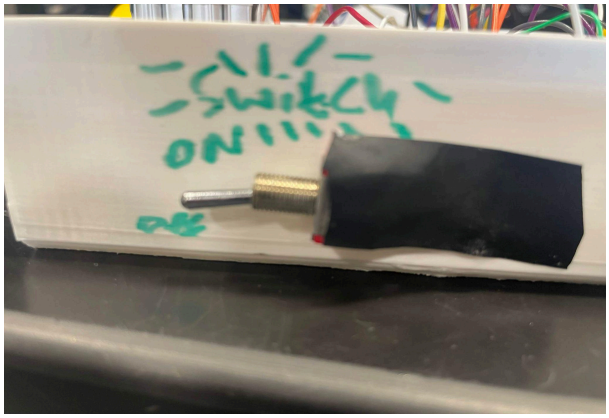

Figure 16: Image of Kill Switch location on side of Bot

Appendix C: Electrical Verification Images

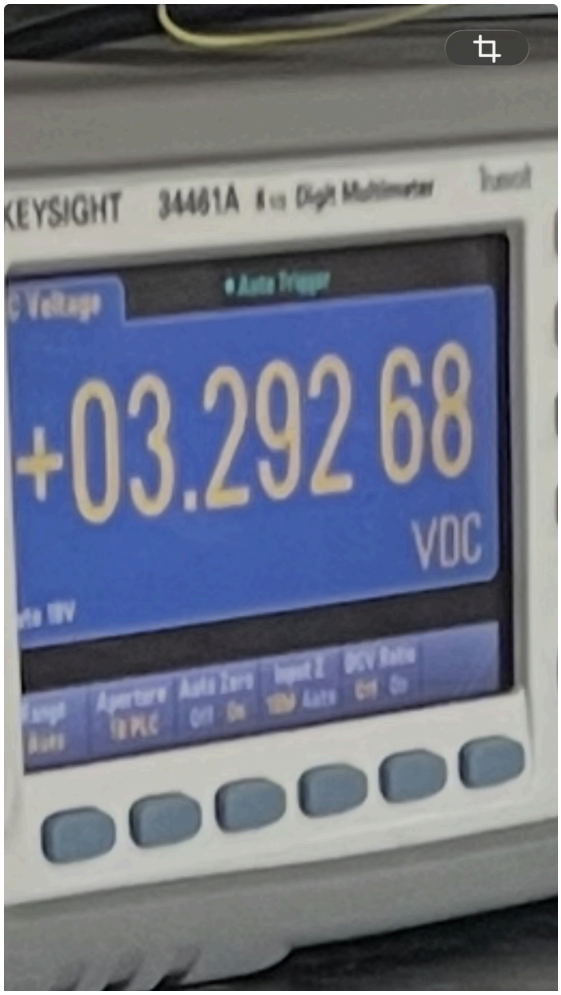

Figure 17: Reading for LDO of the USBC 5V to 3.3V stepdown

Appendix D: Breadboard version of bot

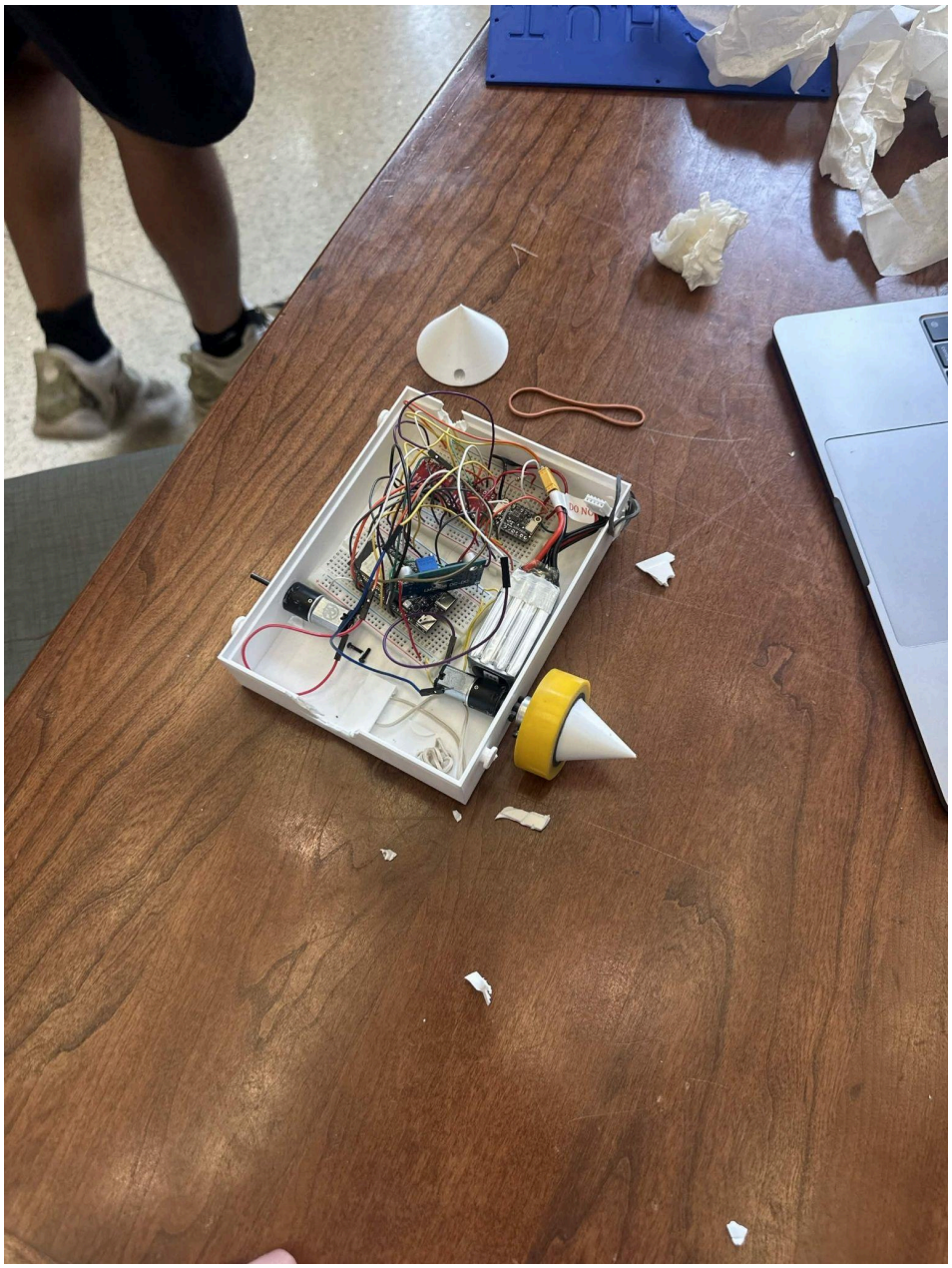

Figure 18: Image of Breadboard Version of Bot