ECE445
Spring 2025

3D Printed Antweight Battlebot

Team 10
Don Lazatin, Brian Pau, Shashank Sangireddy

TA: Jason Jung

Table of Contents

1. Introduction

1.1 Problem

Combat robotics competitions have experienced a significant increase in popularity in recent years. These competitions offer participants the opportunity to develop and practice engineering, design, and programming skills in an all-hands-on, competitive environment. Professor Gruev's Antweight Battlebot Competition presents a unique challenge where each team must construct a fully functional battlebot with strict design limits and constraints, which are listed below:

1. Battlebot must weigh less than 2 lbs.
2. Battlebot must be 3D printed with the following materials: PET, PETG, ABS, PLA, PLA+.
3. Battlebot must be controlled from the PC via Bluetooth or WiFi.
4. The weapon must be activated using only either motors or pneumatics.
5. Battlebot must have a way of easy manual shutdown and automatic shutdown.
6. Battlebot must adhere to in-competition rules [1].

The goal of this project is to design and implement an Antweight Battlebot that is eligible and capable of competing against other Battlebots in a competition arena environment. More specifically, the Battlebot must be able to disrupt the functionality of competing Battlebots with a fighting tool while ensuring its own functionality. The design of the Battlebot must adhere to the limits and constraints listed above.

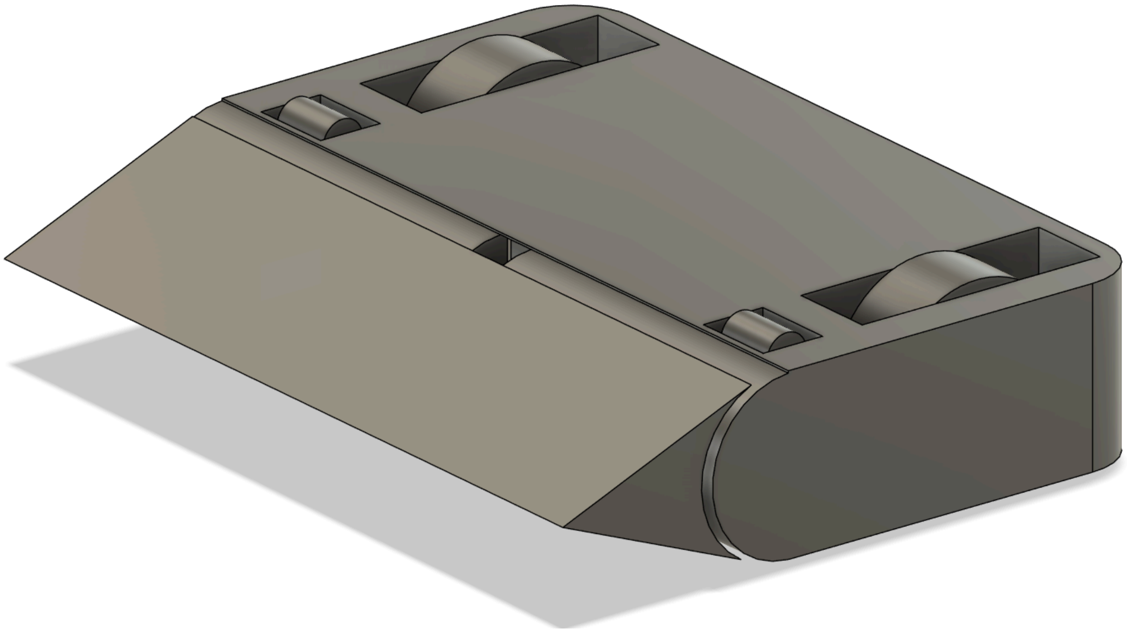
1.2 Solution

Our overall solution to this task is to design and construct a combat-ready battlebot equipped with an opponent-destabilizing wedge weapon, a durable yet lightweight 3D-printed chassis body, and a wireless control system. Our battlebot will be powered by a microcontroller with built-in Bluetooth capabilities, allowing seamless remote operation and communication. The controller we plan on investing in will be the ESP32-S3-WROOM-1. The battlebot's movement system will be controlled via two N20 motors driving the wheels with DRV8231 motor controllers, which have integrated h-bridges, to control the wheel direction.

Our wedge will serve as our primary combat mechanism, actuated by a Micro HDD Servo. This weapon will be designed to lift and destabilize the opponent robots by utilizing its mass and motor-driven activation. To ensure the bot remains functional regardless of its axis orientation, the chassis will be symmetrically structured about its horizontal axis. This will allow our robot to remain functional even if it is flipped. A rechargeable 3S LiPo battery will provide power to all subsystems of the design. In regards to the power distribution of the circuit, we plan on implementing step-down circuitry to regulate voltage throughout the different subsystems as necessary. Our solution will implement multiple kill-switch functions, including both a physical switch as well as a software implementation by disabling the Bluetooth connection, to comply with competition safety requirements and constraints. By integrating mechanical engineering principles with embedded systems and wireless communication, our battlebot will be a competitive and well-engineered entry into Professor Gruev's Antweight Battlebot Competition.

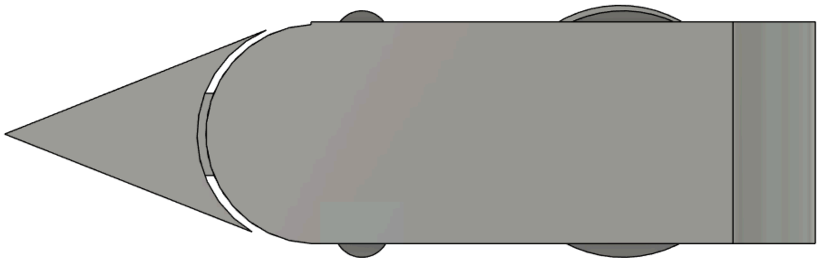
1.3 Visual Aid

The following is an orthographic view of the robot. This depicts the wedge-based weapon as well as the general shape of the chassis that we are aiming to achieve. Also shown are the wheels, with the larger, rear ones being motor activated and the smaller, front ones being free spinning for stability.



Orthographic View of the CAD Model of the Robot

Also depicted below is a side view of the robot. This better puts into context the horizontal symmetry of the robot. If the robot is flipped over, the symmetrical design should still allow the robot to function.



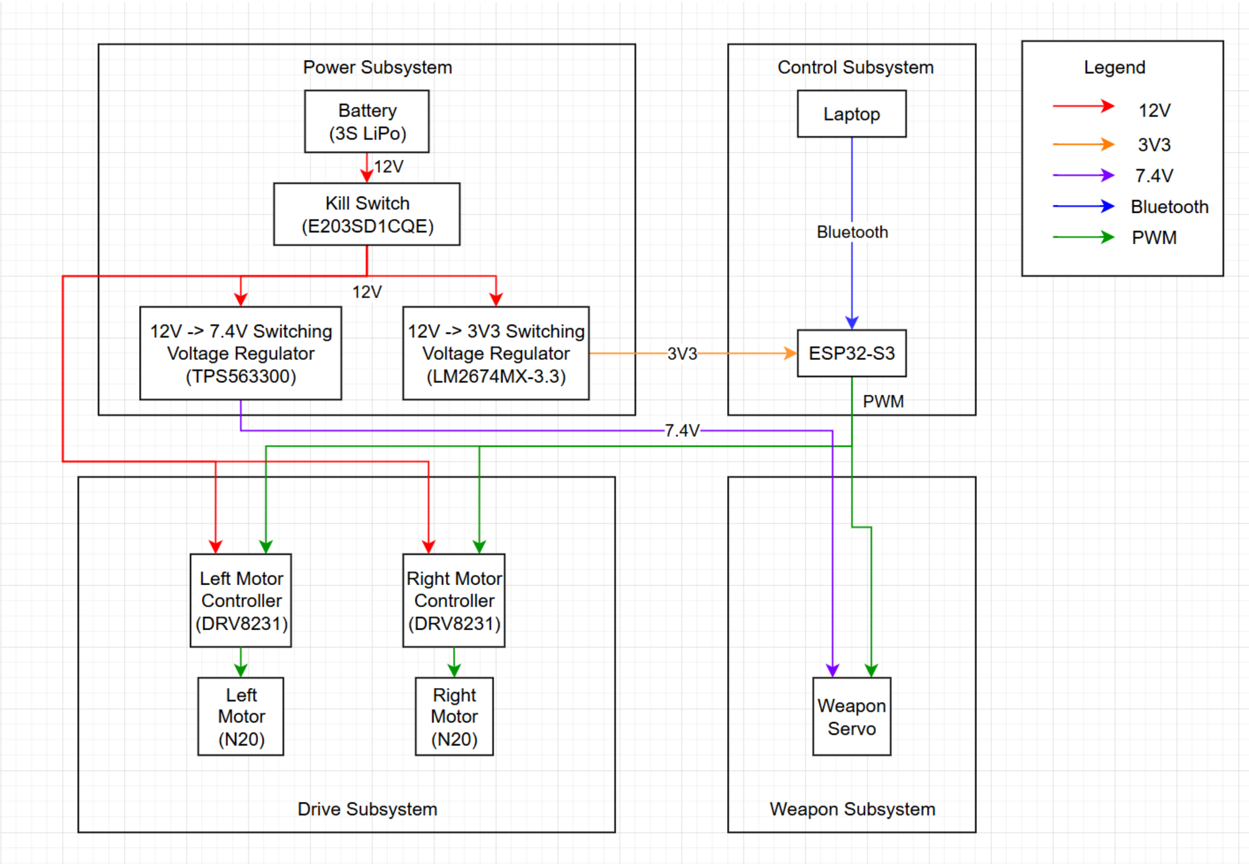
Side View of the CAD Model of the Robot

1.4 High-Level Requirements List

- The battlebot must not exceed a maximum acceleration of 5 m/s^2 to ensure controlled maneuverability across the competition arena.
- The wedge weapon system must be capable of lifting and displacing objects of up to 2 lbs and must return to its original position within 2.5 seconds after activation.
- The battlebot's Bluetooth connection must maintain reliability with the control PC at a range of at least 15 feet, with a command response time of less than 200 milliseconds in order to ensure real-time and precise control.

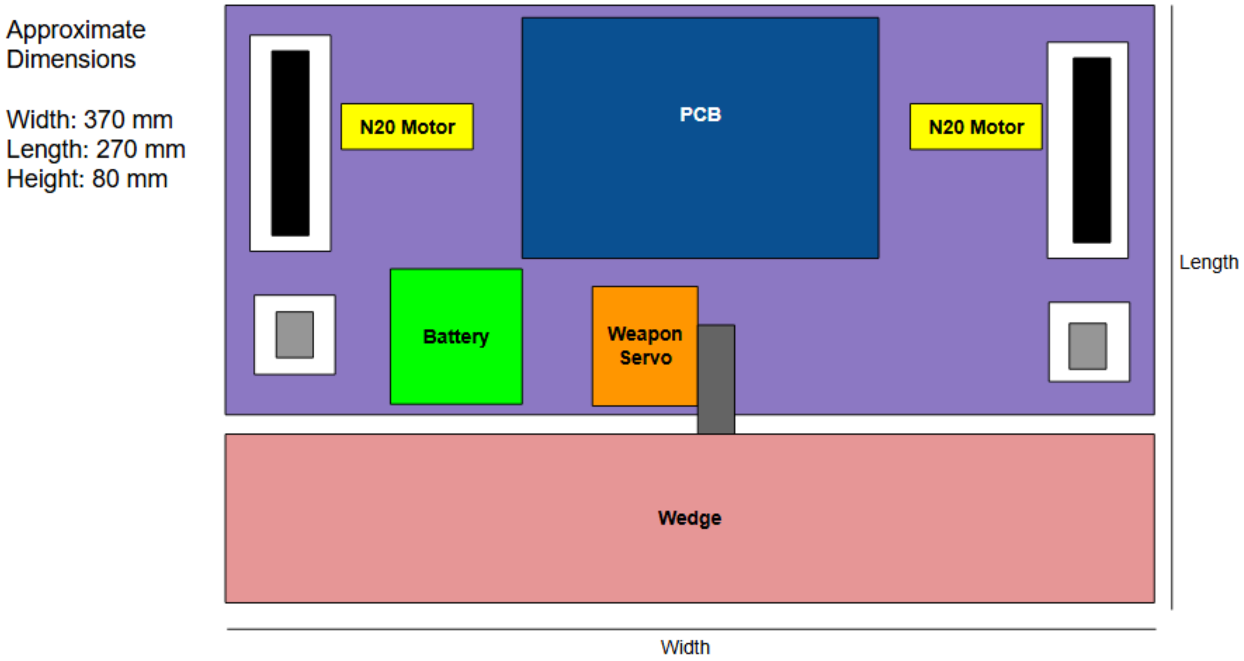
2. Design

2.1 Block Diagram



Our robot consists of 5 total subsystems: power, control, drive, weapon, and chassis. The block diagram displays the interconnections between our different subsystems. The chassis subsystem is not depicted in the block diagram as it is a mechanical component and does not directly interact with any of the other subsystems electrically.

2.2 Physical Design



Above is a basic representation of the internal layout of our battlebot. Colored in light purple is the chassis, which houses all electrical components of our battlebot design. This entire region is to be 3D printed as one part. Colored pink is the fighting tool, which attaches to the front of the chassis by the 3D printed servo arm, colored in dark gray. Due to this arrangement, the servo itself is housed near the front of the chassis. The N20 motors actuate the drivetrain subsystem, and each motor drives one of the larger rear wheels, which are to be covered by a rubber tire tread. In light gray are the set of free-moving wheels, placed near the front of the chassis to prevent dragging across the floor. There are 2 sets of these free-moving wheels, one for the top face and one for the bottom face. All sets of wheels protrude past their respective surfaces such that the chassis never touches the ground plane.

2.3 Subsystem Overview and Requirements

2.3.1 Power Subsystem

Functional Description: The Power Subsystem is responsible for supplying power to the entire robot. This subsystem consists of a 3S 650mAh 75C LiPo battery, a LM2674MX-3.3 switching voltage regulator, and a TPS563300 switching voltage regulator. The LM2674MX-3.3 is used to step down the battery voltage from 12V to 3V3 at 500mA, which is used by the microcontroller. The TPS563300 is used to step down the battery voltage from 12V to 7.4V at a maximum of 3A output current, which is used by the weapon’s servo. This subsystem also includes a physical toggle killswitch between the battery and the rest of the robot. The 3S LiPo battery is nominally 11.1V, but can reach a maximum of 12.6V when fully charged.

Contribution to Overall Design: This subsystem contributes to the overall design by ensuring that the robot is capable of operating for at least the 2-minute competition time. It also ensures that the other components have the necessary power to function. Use of a battery is paramount as the robot needs to be able to move in the competition arena without being hindered by wired connections to a PC or power supply.

Design Decisions:
To prevent the battery from being plugged in incorrectly, we are using an XT30 plug to connect the battery to the PCB. XT30 plugs are manufactured in such a way that it is impossible to plug it in in an incorrect orientation.

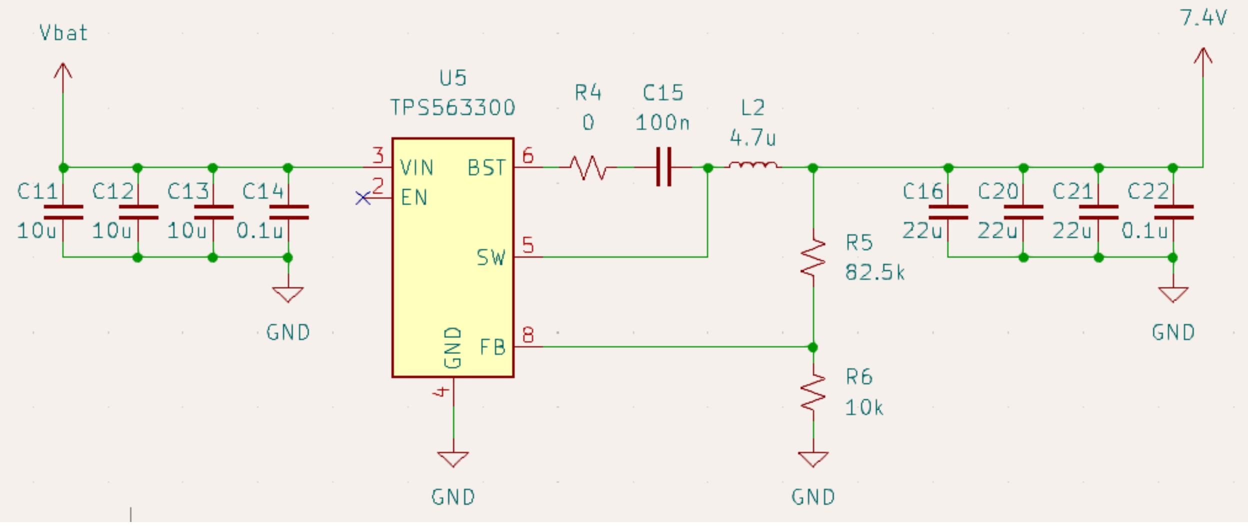
The battery we are using is a 3S LiPo battery with a 650mAh minimum energy capacity and a discharge rate of 75C, with a maximum burst discharge rate of 150C. This means that the battery is able to provide a maximum continuous current of 48.75A with a maximum burst current of 97.5A. Our expected maximum current draw by our robot is 6.7A, well below either of the two limits. A more detailed breakdown can be found in our battery's tolerance analysis, including calculations regarding the battery's expected runtime.

Both regulators we are using, the LM2674MX-3.3 and the TPS563300, are switching regulators. We decided to use switching regulators for the improved heat dissipation. The LM2674MX-3.3 has an efficiency of up to 96% and does not require extra heat sinking. The TPS563300 also has an efficiency of over 90% in our operating range and also has built-in overvoltage, undervoltage, and overcurrent protection.

Since the TPS563300 is an adjustable voltage regulator, the output voltage is set by a voltage divider connected to the feedback pin. The equation is given as follows, with recommended values of $V_{REF} = 0.8V$ and $R_{FBB} = 10k\Omega$:

$$R_{FBT} = \frac{V_{OUT} - V_{REF}}{V_{REF}} \times R_{FBB}$$

We are using this step down converter for the servo, which has an operating range of 4.8 to 8.4V, but gives measurements for current and torque for a 7.4V input. Thus, to achieve a 7.4V output from the TPS563300, we are using an 82.5kΩ resistor as R_{FBT}, as seen in the schematic below.



Circuit Schematic for TPS563300 Voltage Regulator

We are using an E203SD1CQE toggle killswitch between the battery input and the rest of the electronics. This killswitch was chosen for its high current rating (7.5A) and for its easily accessible form factor. The killswitch can be turned on or off to provide a manual way to shut off the robot.

A current sense resistor (R15) has been included for measuring the input current to the system from the battery. It is a 10mΩ resistor rated at 1W. The expected maximum possible current coming into the system is 6.7A, which would mean that the power draw, using $P = I^2R$, would be around 0.45W, well below the resistor's power rating.

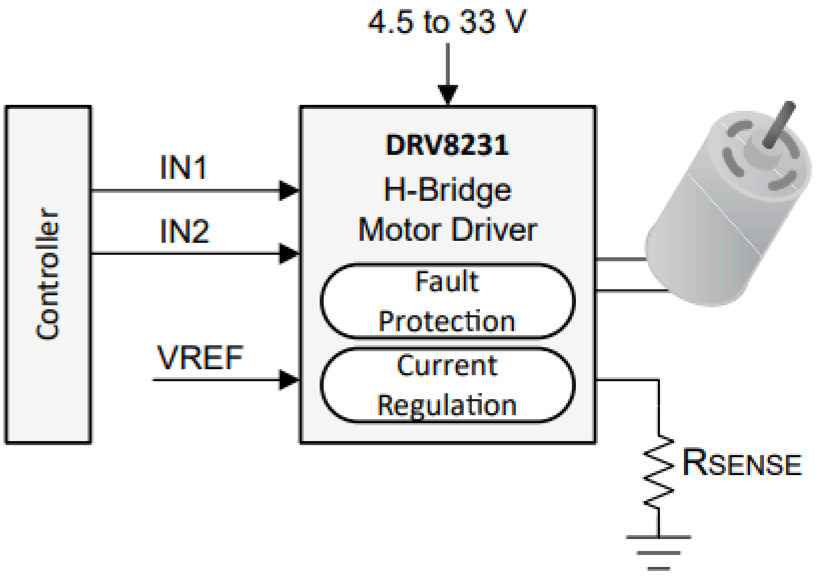
Interfaces: This subsystem interfaces directly with the control, drivetrain, and weapon subsystems as it supplies power to the control circuit and motors. It provides 3V3 and 500mA to our ESP32-S3 microcontroller, 12V to the drivetrain motor controllers, and 7.4V to the servo.

Requirements	**Verification Method**
The battery must be able to supply up to 6.7A at 12V±0.6V.	Test: - To measure the voltage: using a DMM or oscilloscope, probe the battery test point (TP1). - To measure the current: using a DMM, measure the voltage drop across the current sense resistor R15. Then, using its known resistance of 10mΩ, calculate the current using I = V/R.

	The voltage and current measurements should be recorded as singular numerical values and verified that they fall under the required range.
The LM2674MX-3.3 voltage regulator must be able to step down the battery voltage to 3V3±0.3V	Test: To measure the voltage: using a DMM or oscilloscope, probe the ESP32 test point (TP2). The voltage measurement should be recorded as a singular numerical value and verified that it falls under the required range.
The TPS563300 voltage regulator must be able to step down the battery voltage to 7.4±1V	Test: To measure the voltage: using a DMM or oscilloscope, probe the servo test point (TP3). The voltage measurement should be recorded as a singular numerical value and verified that it falls under the required range.
The battery must be able to power the robot for at least 2 minutes	Test: To test the battery lifetime, we will plug the battery into the robot and prepare a stopwatch. We will start the stopwatch and begin simulating a real combat scenario, continuously activating the drivetrain and weapon motors as if we were in competition. This will be recorded as a pass if the robot, under real competition simulation, stays powered and operational for the full 2 minutes. In the case that it does not, it will be recorded as a fail and the time that the robot stops working will be recorded.

2.3.2 Drivetrain Subsystem

Functional Description: The Drivetrain Subsystem is responsible for the battlebot's movement and maneuverability. It utilizes two N20 micro motors each rated for 12V DC operation that provides a no-load speed of approximately 460 RPM and a current draw between 100 mA to 1600 mA. The DRV8231 motor driver chip utilizes an H-bridge mechanism and operates within an input voltage range of 4.5V to 33V and supplies a continuous current of up to 3.7A per motor channel. The driver can receive control signals between 0 and 5.5V from the microcontroller and manages the speed and direction of the motors, which enables precise maneuverability.



Simplified Schematic of DRV8231 H-Bridge Motor Driver

<u>Contribution to Overall Design</u>: This subsystem enables the battlebot to perform movement such as optimal acceleration, deceleration, and turns. All of which are essential for both offensive and defensive maneuvers in combat scenarios. The selection of the lightweight N20 motors contributes to the overall weight efficiency of the design, which ensures our battlebot complies with the 2 lb weight limit restriction.

<u>Interfaces</u>: The Drivetrain Subsystem interfaces with the Power Subsystem and the Control Subsystem. Through the Power Subsystem, this subsystem receives the battery voltage from the power supply to effectively power the drive motors through the motor drivers. Through the Control Subsystem, this subsystem receives PWM (Pulse Width Modulation) signals from the ESP32-S3 microcontroller in order to control motor speed and directional maneuvers.

Requirements	**Verification Method**
Each N20 motor must operate at 12V, providing a speed between 310 RPM to 460 RPM and a pull current between 100 mA to 1600 mA.	<u>Test:</u> To ensure each motor is providing a speed between 310-460 RPM, we will use the equation below. $RPM = (Linear\ Speed * 60) / Wheel\ Circumference$ We note that *Linear Speed* is the maximum velocity that our battlebot is able to achieve. While the motor is rotating the wheel, we will use a

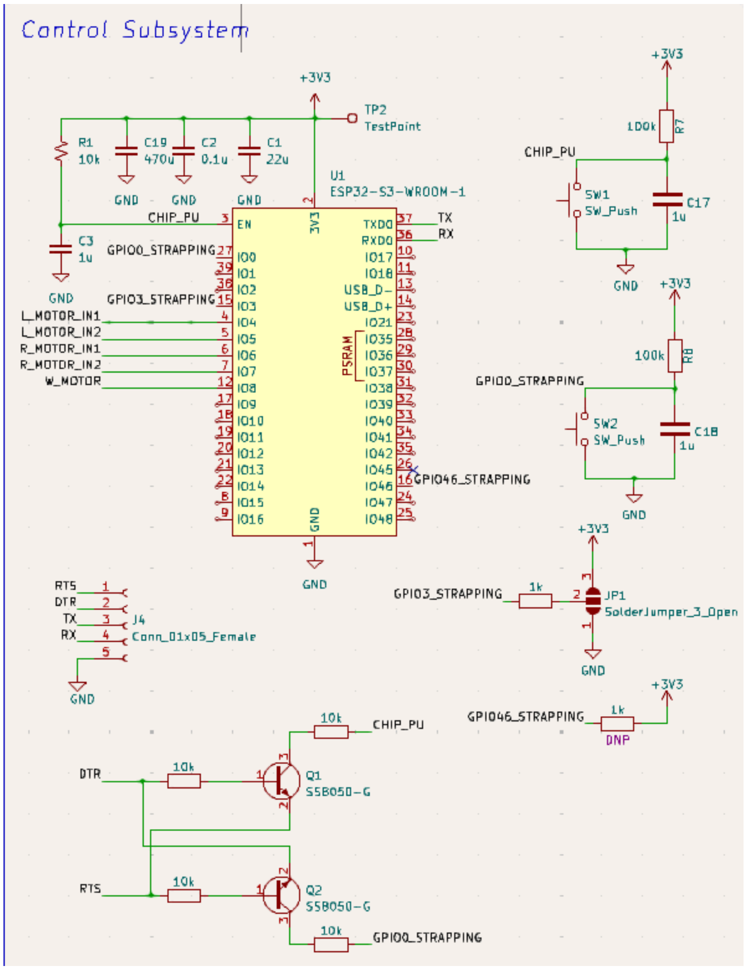
	multimeter to ensure that the current falls within 100-1600 mA under loaded conditions.
Drivetrain subsystem must prevent the battlebot from exceeding a maximum acceleration of 5 m/s^2, ensuring controlled maneuverability across the competition arena.	Test: To measure acceleration, we will run tests in order to verify that the battlebot does not exceed an acceleration of 5 m/s^2. To do this, we will have our battlebot start from rest. We will begin the battlebot's motion at maximum acceleration (full-throttle) and verify the time it takes to pass a marker that is exactly 10 meters away from the starting point. Analysis: Given the data through testing, we will utilize the following kinematic equation to obtain acceleration. We note that v_0 is equal to 0 since we are starting our battlebot at rest and t is the time it takes for our battlebot to cross the 10 meter marker. $s = v_0 t + .5at^2$ $10 = .5at^2$ $a = 20 / t^2$
The subsystem must respond to control inputs within 200 milliseconds to ensure accurate, precise maneuverability.	Test: To test latency, we will send a control signal to the microcontroller. Using an oscilloscope and a timer, we will verify that the time from input to motor response is less ≤ 200ms.

2.3.3 Control Subsystem

Functional Description: The Control Subsystem is responsible for processing user command inputs and translating them into movement commands for the battlebot. This subsystem uses the ESP32-S3-WROOM-1 microcontroller. This controller has a Bluetooth module which will be used to wirelessly communicate with the laptop. We are choosing Bluetooth as a communication protocol to ensure a reliable connection between the laptop and the battlebot. The ESP32-S3-WROOM-1 processes these commands and sends appropriate PWM control signals to the motor driver chips. The Control Subsystem also includes a software-based killswitch: the user can manually disconnect from Bluetooth, which will terminate the connection to the robot, causing it to stop operating.

Contribution to Overall Design: This subsystem enables us to remotely operate the other subsystems of the robot. From the user's perspective, the robot should be able to traverse its environment, activate and reset its combat tool, and manually shut off its functionality in response to a dedicated kill-switch.

Design Decisions: We chose to use the ESP32-S3 microcontroller for its larger flash memory, built-in antenna, and flexibility between Bluetooth and WiFi. The S3 also has two motor control PWM (MCPWM) peripherals with 3 PWM operators each, for a total of 6 signal outputs. This made the S3 appealing to us as our robot is controlled entirely off of PWM signals to the drivetrain motors and weapon servo as seen in the circuit schematic below.



Circuit Schematic for Control Subsystem

Interfaces: The microcontroller will be powered directly by the power subsystem, receiving 3V3 from the voltage regulator. This subsystem also interfaces with the drivetrain and weapon subsystems, providing the control signals to their respective motor controllers to activate the motors.

Requirements	**Verification Method**
The microcontroller must maintain a reliable connection with the control PC at a range of at least 15 feet, with a command response time of less than 200 milliseconds.	Test: - To test the range requirement, we will place the robot at a distance of 15 feet from the control PC. After sending a command from the PC, we will visually evaluate whether the robot behaves as expected. - To test the command response requirement: we will probe the respective test points at our motors using an oscilloscope or the ADALM. We will send a command from the command PC and measure the signal at the motor. Then, using the measurement we can calculate the time difference between when the command was sent and when it was received. The range requirement will be recorded as pass/fail, where if it happens to fail, the test will be repeated in order to find the maximum distance. The command response time will be recorded as a singular numerical value and evaluated against the maximum allowable response time.
The software-based killswitch must deactivate the robot within 1 second.	Test: To test the kill-switch mechanism, we will use an oscilloscope or the ADALM to probe a drivetrain motor test point. We will continuously run the motor and record the control signal. Then, we will disconnect the Bluetooth connection and measure the time it takes for the control signal at the motor to cease, even when a command is still being input from the control PC. This will be recorded as a singular numerical value and evaluated against the maximum allowable time of 1 second.

2.3.4 Weapon Subsystem

Functional Description: The weapon subsystem consists of a Micro HDD Servo with a high power to weight ratio and the physical wedge that is part of the robot's body, located directly in front of the chassis. These two parts work in tandem to act as a high-power lever with wide surface area, fit to lift and displace objects of up to 2 pounds. The weapon motor operates within a 4.8V to 8.4V input voltage range with a current draw of up to 3A. This motor is able to jolt up to 60 degrees within .22 seconds with no load. The weapon motor is controlled by PWM signals directly from the microcontroller.

Contribution to Overall Design: This subsystem enhances the battlebot's offensive capabilities. This subsystem allows for the battlebot to destabilize and lift opposing battlebots of up to 2 lbs. The choice of the wedge weapon maximizes potential to destabilize other opponents while properly managing weight and power consumption.

Interfaces: The weapon subsystem interfaces directly with the power and control subsystems, primarily receiving both power and control signals from the respective sources. The weapon subsystem may also send data back to the control subsystem, specifically the position of the servo arm. This data can be used by the microcontroller to more easily and accurately reset the position of the weapon subsystem for repeated use.

Requirements	Verification Method
The wedge must be capable of lifting and displacing objects of up to 2 lbs.	Test: To test this, we will use a scale to first find an object that weighs at least 2 lbs. We will then verify that the wedge is able to lift this object, unassisted by any other external sources of help. This will be recorded as a pass/fail. If it is a fail, the test will be repeated to find what the maximum weight the wedge can lift is.
The weapon must return to its original position within 2.5 seconds after activation.	Test: Since 2.5 seconds is long enough for the eye to process, we will just be using a timer. We will time how long it takes for the wedge to return to its original position following activation. This will be recorded as a pass/fail. If it is a

	fail, the test will be repeated to find how long it takes for the weapon to return to its original position.
The wedge must not compromise its structure after any actuation, such that this subsystem can be activated repeatedly.	Test: To test this, we will inspect the wedge's physical structure visually and look for any deformities or other such physical defects. Additionally, we will activate the wedge multiple times in succession, to ensure that it can be activated repeatedly. This will be recorded as a pass/fail. If it is a fail, notes will be recorded detailing the type of defect and on what number activation it failed on.

2.3.5 Chassis Subsystem

Functional Description: This subsystem provides the structural base for the battlebot. It holds the main circuit, motors, weaponry, and power source of the battlebot. This subsystem will be 3D printed and constructed using PLA+ filament in order to balance durability and weight management. Its symmetrical design along the horizontal axis ensures continuous operation even if flipped. The square structure of the chassis minimizes weak points, which enhances durability against opposing impacts.

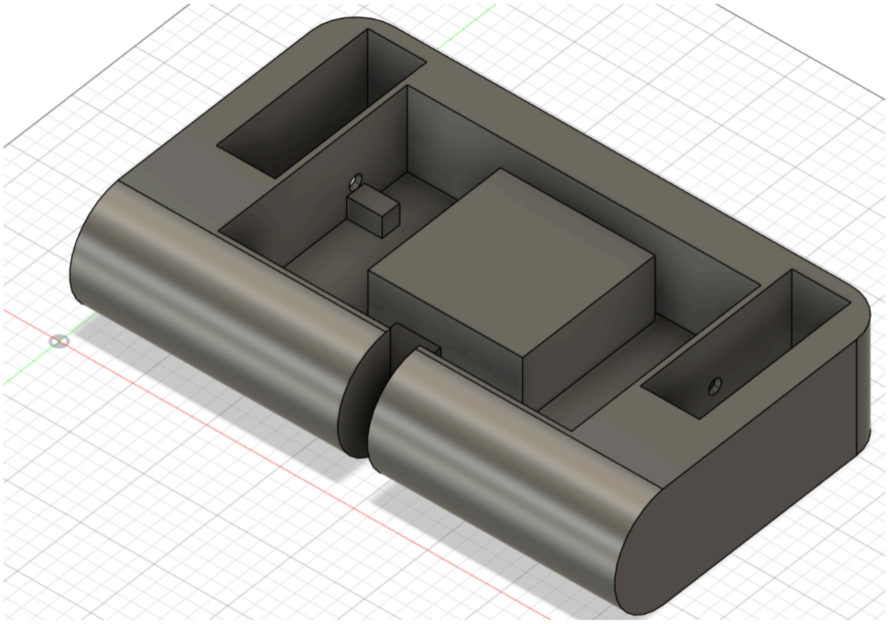
Contribution to Overall Design: The chassis is the robot's main structural frame, housing critical components such as the power supply and central processing unit. The robot's body will be 3D printed using PLA+.

Design Decisions: Pictured below is a CAD model of the internal structure of the chassis. Not shown are the top-facing lid that will secure the electrical components and PCB inside the chassis, and the front-facing wedge that attaches to the actuating servo as the physical component of our battlebot's fighting tool, as well as the wheels.

The drivetrain wheels are protected by the chassis itself, with all wheels being inset into the chassis. The holes in the sides of the body are for the N20 motor shafts to insert through. These holes are centered at the mid level of the main body's height, which would allow for no change in how the robot sits on the ground plane even while upside down.

The actuator for the fighting tool, the high power servo, is seated inside the main body near the front. The "arm piece" connecting our wedge weapon to the servo will need to move through the chassis, which necessitates the shown opening at the front of the body, which extends from the top to the bottom of the main body.

The robot is intentionally designed to be symmetrical along the horizontal axis, such that there is no change in functionality if flipped upside down. Both the front and rear sets of wheels reach past the top and bottom faces, such that the robot does not drag itself across the ground plane while traversing the arena.



Orthographic View of CAD Model of Internal Structure of Chassis

Interfaces: The chassis securely mounts and shields the Drivetrain, Control, Power, and Weaponry Subsystems.

Requirements	**Verification Method**
The chassis must be able to properly house internal components.	Test: To test this, we will visually inspect the fully assembled robot and ensure that no electronics, besides the physical kill switch, are exposed in both an upright and flipped orientation. This will be recorded as a pass/fail.
The robot must maintain functionality in both upright and flipped orientations.	Test: To test this, we will operate the robot in both an upright and then a flipped orientation, ensuring that it is still functional in both orientations.

	This will be recorded as a pass/fail for the upright and flipped orientations individually.

2.4 Tolerance Analysis

<u>Battery Runtime</u>: One of the risks facing our battlebot is ensuring that the robot remains operational for the full 2 minute competition. To ensure this, we must make sure that our battery can sufficiently power our robot for at least 2 minutes.

The ESP32-S3-WROOM-1 microcontroller that we are planning to use draws 500mA at 3V3. The N20 motors for the drivetrain individually draw a maximum of 1600mA, for a combined 3200mA. Lastly, our Micro HDD Servo motor for the weapon draws a maximum of 3000mA. Combined, the absolute maximum current draw of our robot is 6.7A. Thus, our battery should be able to accommodate this for at least 2 minutes, even though the expected current draw will be much less.

$$I_{max} = I_{ESP} + I_{Drive} + I_{Weapon} = 500mA + 3200mA + 3000mA = 6700\, mA$$

The minimum required energy capacity can be calculated as follows:

$$Required\ Capacity = (6.7A) * (1/30\ hours) = 67/300\ Ah \approx 223\ mAh$$

The C rating describes the rate at which the battery can discharge its current. A higher C rating means that the battery can discharge at a quicker rate, safely supplying larger amounts of current than a battery with a lower C rating. It should be noted that a higher C does not necessarily mean the battery has a shorter runtime. The minimum C rating for our can be calculated as follows:

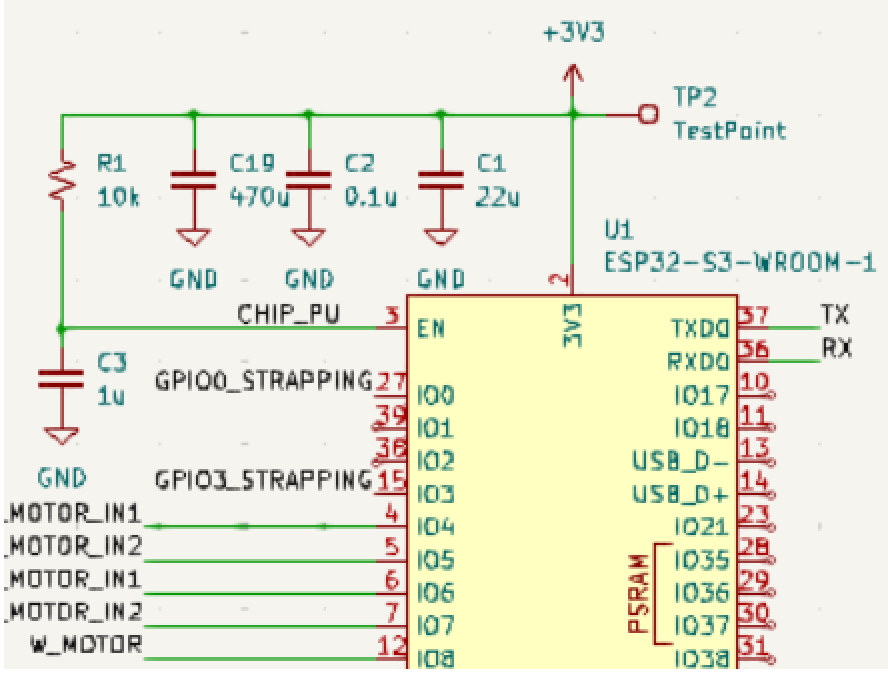
$$Minimum\ C\ Rating = (6.7A) / (67/300\ Ah) = 30C$$

Our chosen battery has a capacity of 650mAh and a C rating of 75C. The capacity of 650mAh means that it should be able to supply 6.7A for about 5.8 minutes, well above the 2 minute competition timeframe. The C rating of 75C means that the battery can safely supply a continuous current draw of 48.75A, which is well above our maximum current draw of 6.7A. These calculations can be seen in the equations below:

$$Runtime = (0.650\ Ah) / (6.7A) \approx .097\ hrs \approx 5.8\ minutes$$

$$Maximum\ Continuous\ Current\ =\ (0.65\ Ah)\ *\ 75C\ =\ 48.75\ A$$

Inrush Current: Another concern is that of inrush current, which are large spikes in current draws when the motors start or stop. This is concerning for our system as it may cause drops in voltages that could negatively affect our electronics, including resetting our microcontroller. To best mitigate this, we've included a large capacitance of 470µF near the voltage input pin to the ESP32 microcontroller. Additionally, the DRV8231 motor controller ICs we are using for the drivetrain motors have built-in current regulation features to help mitigate some of the effects of inrush current as well.



Bulk Capacitance at Voltage Input to ESP32-S3 for Mitigating Inrush Current

Weight:

The last significant risk that our battlebot faces is the physical design constraint of staying under the weight limit of 2 pounds, or 907.185 grams.

Our group intends to follow this constraint by designing our robot's chassis with the required electrical components in mind. Many of the electrical components and ICs that are used do not have listed weights. A very conservative upper limit is used (around 10g) in order to provide a "worst-case" scenario in order to add extra tolerance to our total calculated weight.

The following is a table of each component that shall be mounted onto the robot, and their respective dimensions and weights (in grams):

Component Name	Weight (grams)	Dimensions (mm)
PCB	Undetermined (max 15g)	W 100 mm H 25 mm L 100 mm
Tattu 3s LiPo Battery	59 g	W 31 mm H 16 mm L 58 mm
ESP32-S3-WROOM-1	2.15 g	W 18 mm H 3.1 mm L 25.5 mm
Micro HDD Servo	25 g	W 40 mm H 10 mm L 30 mm
TI DRV8231	Unlisted (max 10g)	W 3.8 mm H 1.7 mm L 4.8 mm
Servocity N20 Gear Motor	9.5 g	W 12 mm H 10 mm L 35 mm
Digikey LM2674MX	Unlisted (max 10g)	W 3.9 mm H 1.75 mm L 4.9 mm
SOIC-8 to DIP-8	Unlisted (max 10g)	W 10.16 mm H 0.46 mm L 10.16 mm
TPS563300DRLR Servo Regulator	Unlisted (max 10g)	W 1.5 mm H 0.6 mm L 2.2 mm
E203D1CQE Kill Switch	Unlisted (max 15g)	W 11.43 mm H 24 mm L 19.05 mm
SK12 Diotec Semiconductor Capture Diode	Unlisted (max 10g)	W 2.7 mm H 2.2 mm L 5 mm

When taking into account the number of parts that are included in the final design of the robot, the combined weight comes out to be:
15 + 59 + 2.15 + 25 + 2(10) + 2(9.5) + 2(10) + 10 + 10 = 180.15 grams

This leaves a margin of:
907.185 - 180.15 = 727.035g or 1.6028 pounds

Therefore, the remainder of the robot's weight, consisting of 3d printing filament (PLA+) and rubber treads for the drive train wheels, must weigh at most 727.035g or 1.6028 pounds.

Due to the nature of 3d printing and other manufacturing inconsistencies, it is possible for the exact weights of each of the components listed above to weigh more than what is expected. For these reasons, our target weight is further lowered to allow for a margin of error before the 2 lb weight limit constraint is failed.

The robot chassis will be adjusted in order to fit under the weight constraint as it is easier and cheaper to cut down on the plastic weight than to change the electrical components. This can be achieved via additional extrusions to remove mass and other changes to the chassis's dimensions. More thorough testing of weights will be done once the robot is fully assembled to ensure the robot is under 2 lbs.

A combined weight of:
180.15g (components) + 710g (plastic + rubber treads) = 890.15g
This adjusted maximum weight allows for approximately a 1.877% margin of error for the robot's weight to stay under the enforced limit. This translates to a margin of 17.035g or 0.0375 lbs.

3. Cost and Schedule

3.1 Component Costs

Control Subsystem Power Subsystem Weaponry Subsystem Drivetrain Subsystem Developmental Components

Component	**Part #**	**Manufacturer**	**Quantity**	**Cost**	**Link**
ESP32-S3-WROOM	ESP32-S3-WROOM-1-N4	Espressif	1	$5.06	Link
E203SD1CQE Kill Switch	E203SD1CQE	C&K	1	$15.99	Link
LM2674-3.3 (Voltage Reg)	LM2674MX-3.3/NOPB	Texas Instruments	2	$2.58	Link

3S 650mAh 75C LiPo battery	TA-75C-650-3S1P-XT30	TATTU	1	$13.80	Link
SK12 Capture Diode	SK12	Diotec Semiconductor	1	$0.19	Link
Micro HDD Servo	EL-3760	ez-robot	1	$22.49	Link
TPS563300DRLR Servo Regulator	TPS563300DRLR	Texas Instruments	1	$0.91	Link
N20 Motors	638126	ServoCity	2	$25.98	Link
DRV8231 (Motor Driver Chip)	DRV8231DDAR	Texas Instruments	2	$2.58	Link
ESP32-S3-DEVKITM-1-N8	ESP32-S3-DEVKITM-1-N8	Espressif	1	$13.30	Link
SOIC-8 TO DIP-8 SMT ADAPTER	PA0001C	Chip Quik Inc.	1	$6.49	Link
3D Printing	N/A	N/A	N/A	$0	N/A

SUM OF ALL COMPONENTS: $109.37

3.2 Cost Analysis

We anticipate working on our Antweight Battlebot for around 8 hours per week over the course of 6 weeks in order to complete our finalized product. We believe that a fair hourly rate is $25 per hour due to the rigorous and tedious aspects that this project involves. Based on this information, we have calculated individual salary and group salary below.

Hours Per Week ***x*** *Pay Per Hour* ***x*** *Total Weeks (Individual Salary)*
$8 * $25 * 6 Weeks = **$1,200**

x *Number of Group Members*
$1,200 * 3 Group Members = **$3,600**

We are estimating to need at most 1-2 labor hours from the ECE machine shop. Although most of our components will either be 3D-printed or ordered online, we may be seeking their assistance for minor mechanical aspects such as motor to wheel connection and guidance on a hinge that will allow for opening and closure of our chassis device. We have estimated this process to be completed within 2 labor hours.

Grand Total (Sum of Costs): $3,600 + $109.37 = **$3,709.37**

3.3 Schedule

Week	**Tasks**
Week of March 10th, 2025	● Order Remainder of Components Needed **(Don)** ● Finalize Codebase for Breadboard Demonstration **(Don & Shashank)** ● Finalize Breadboard Demonstration **(ALL)** ● Finalize 1st Schematic & PCB Design **(ALL)** ● Submit 1st Finalized PCB Order **(Brian)** ● Get Familiar with 3D-Printing Resources **(Shashank)**
Week of March 17th, 2025	● Receive 1st PCB **(ALL)** ● Begin 1st PCB Assembly & Troubleshooting **(ALL)** ○ Solder 1st PCB **(Brian)** ● Request for More Components Needed **(Brian & Shashank)*** ● Order Components Needed **(Don)*** ● Finalize MCU to Laptop Connection **(Don)** ● Begin 3D-Printing Design **(Shashank)**
Week of March 24th, 2025	● Continue 1st PCB Troubleshooting **(Brian)** ● Prepare 2nd PCB Design **(Brian)*** ● Begin Implementation of MCU to External Controller Connection **(Don)** ● Continue 3D-Printing Design & Print **(Shashank)**
Week of March 31st, 2025	● Finalize Verdict on 1st PCB **(ALL)** ● Submit 2nd Finalized PCB Order **(Brian)*** ● Finalize Battlebot Codebase **(Don)** ● Finalize Chassis Design & Final Prints **(Shashank)**
Week of April 7th, 2025	● Receive 2nd PCB **(ALL)*** ● Begin 2nd PCB Assembly **(ALL)*** ● Finalize 2nd PCB Assembly & Soldering **(Brian)*** ● Troubleshoot Code **(Don)** ● Final Print of Chassis **(Shashank)** ● Begin Formation of Battlebot Components **(ALL)**
Week of April 14th, 2025	● Extra Time for Troubleshooting **(ALL)*** ● Finalize Final Presentation **(ALL)** ● Finalize Final Paper **(ALL)**

* *if applicable or if needed*

4. Ethics and Safety

4.1 Ethics

As described by Section I Part 1 of the IEEE Code of Ethics [2], we will always put the safety and health of the public first and disclose any potential risks where appropriate. Combat robots pose an inherent threat to public safety, so we will be responsible for ensuring that the robot is designed, handled, and operated responsibly so as to eliminate any risk to the safety of ourselves and others.

As described by Section I Parts 5 and 6 of the IEEE Code of Ethics [2], we will act responsibly as engineers, making sure to openly accept criticism and feedback and to ensure that we have the proper knowledge to accomplish things safely and correctly. Making a robot is a multi-disciplinary effort that requires knowledge in many different fields and technologies. We will make sure that we are doing the proper research, learning, and asking for help in order to complete our project safely and responsibly.

As described by Sections II and III in their entirety of the IEEE Code of Ethics [2], we will make sure to create and maintain a positive, healthy, and collaborative working environment. This includes avoiding using harmful language and holding each other accountable for our actions. We will strive for open and frequent communication and a willingness to help each other in order to make a positive working experience for all those involved.

4.2 Safety

Combat robots and battlebots are inherently dangerous and require thorough safety guidelines to ensure they do not pose a threat to the public. We will be following the safety regulations outlined by the NRC, specifically regarding antweight battlebots [1]. Additionally, we will also be abiding by the safety guidelines for batteries given by the ECE445 course staff [3] as our robot will be battery-powered. On top of the existing guidelines and rulesets, we will be exercising caution during testing and operation. As our design uses motors and weaponry, we will make sure that they are only operational when being actively powered and controlled by a human operator. Additionally, we will not be operating the robot or its motors outside of a safe competition environment or testing area. Finally, we will be implementing multiple kill-switches that will give the user control to disable the robot at any time as needed.

5. References

[1] National Robotics Challenge, “2025 Contest Manual,” Jan. 06, 2025. https://irp.cdn-website.com/9297868f/files/uploaded/NRCContestRules2025-7677ffcb.pdf (accessed Feb. 10, 2025).

[2] IEEE, “IEEE Code of Ethics,” *ieee.org*, Jun. 2020. https://www.ieee.org/about/corporate/governance/p7-8.html (accessed Feb. 10, 2025)

[3] ECE445 Spring 2016 Course Staff, “Safe Practice for Lead Acid and Lithium Batteries,” Apr. 13, 2016. https://courses.grainger.illinois.edu/ece445/documents/GeneralBatterySafety.pdf (accessed Feb. 10, 2025).

[4] Texas Instruments, “DRV8231 3.7-A Brushed DC Motor Driver with Integrated Current Regulation.,” Nov. 2021. https://www.ti.com/lit/ds/symlink/drv8231.pdf (accessed Mar. 5, 2025).