INVERTIBLE CONTROL ANTWEIGHT BATTLE BOT

By

Ben Goldman

Jack Moran

Design Document for ECE 445, Senior Design, Spring 2026

TA: Haocheng Bill Yang

February 27, 2026

Project No. 13

Abstract

This report presents the design and implementation of an invertible control antweight combat robot. The system uses a 6-axis Inertial Measurement Unit and an ESP32 to provide orientation-aware drive logic. Using a gyroscope and accelerometer, the robot programmatically remaps motor signals to maintain intuitive controls for the operator when the chassis is inverted. The primary weapon is a horizontal spinning bar at the front to deliver powerful blows to the opponents. The 3D-printed chassis will be designed with defense in mind, featuring limited edges, corners, and other vulnerabilities to prevent weapons from damaging or grabbing the robot. At the center, a custom PCB will consolidate power distribution and signal routing, integrating the ESP32 with dedicated motor drivers and voltage regulation circuitry. The design meets all safety regulations and weight class constraints, providing a robust platform that optimizes the operator's tactical efficiency in competitive environments.

Contents

1. Introduction

This project addresses the technical challenge of designing and building a high-impact antweight combat robot that maintains operational control when the platform is inverted. Beyond the immediate engineering application, this project explores the development of orientation-aware control systems to improve user safety and system reliability. This proposal outlines a dual-sided 2lb robot that uses an IMU to drive a feedback loop to automatically remap drive logic to the controller, ensuring simple steering in any orientation. The following sections detail the battle-ready systems, including hardware and software subsystems, to construct a highly capable and powerful antweight combat robot.

1.1 Problem

In the high-stakes environment of antweight combat robotics, the margin for error is incredibly thin. All robots must weigh under 2lb and operate in confined arenas where high-power impacts are common and often result in the robot being flipped or thrown across the enclosure. Many combat robots face a significant tactical disadvantage when inverted. These problems often include having a weapon incapable of operating when inverted, having a drive system that cannot operate inverted, or having a drive system capable of operating inverted but relying on the driver to mentally map inverted controls in real time. For these reasons, many teams choose to build robots with complicated self-righting systems or build defenses to prevent the robot from being flipped in the first place. Any delay in an operator's reaction time or a failure in the mechanical self-righting system often leads to a knockout, as the opponent can take advantage of the moment of vulnerability.

In addition, many teams compromise the reliability of their robots by over-engineering their drive and weapon systems, leading to complications and electronic failures, particularly when under the stress of impacts. Therefore, there is a critical need for a lightweight but robust solution that simplifies the user experience. By removing the cognitive load of navigating inverted controls or commanding a complicated robot, the operator can focus their entire attention on the offensive strategy and evasive maneuvering. This ensures the robot is a constant threat within the arena and has a strong chance of winning regardless of the design of the opponent's robots.

1.2 Solution

Our proposed solution is an invertible control ant-weight combat robot designed for maximum offensive output and intuitive operation. The physical architecture of the device will feature a low-profile, rounded chassis with a primary weapon consisting of a double-sided horizontal spinning bar capable of high-inertia attacks. To ensure the robot remains functional regardless of which side is facing up, the drive system will rely on two recessed wheels that protrude very slightly from both the top and bottom of the chassis. This will allow for a "tank-like" differential drive system. When combined with rounded edges to deflect impacts, the robot will maintain mobility and defense in any orientation without the need for any self-righting arms.

The main innovation lies in the integration of a 6-axis Inertial Measurement Unit and a custom PCB powered by an ESP32 microcontroller. The system uses the IMU's accelerometer and gyroscope data to monitor the robot's orientation in real-time. When the sensor detects that the robot has been flipped, the controls will automatically be inverted via signals sent to the H-bridge motor drivers. The inversion of the control logic will ensure that with the press of the forward input from the laptop, the robot will always move in the direction the weapon is facing. By handling the orientation logic in software, the implementation will achieve a seamless transition that allows for continuous combat effectiveness.

1.3 Visual Aid

The visual representation shown in Figure 1 illustrates the invertible nature of the design. The arrow indicates the robot's wireless control between the ESP32 on the PCB and the operator's laptop. Also, the protrusion of the wheels on both sides allows for ground contact in both the upright and inverted state while also maintaining a very low center of gravity. It is also important to recognize that the top and bottom views are nearly identical, as the direction of rotation for the weapon is the only difference.

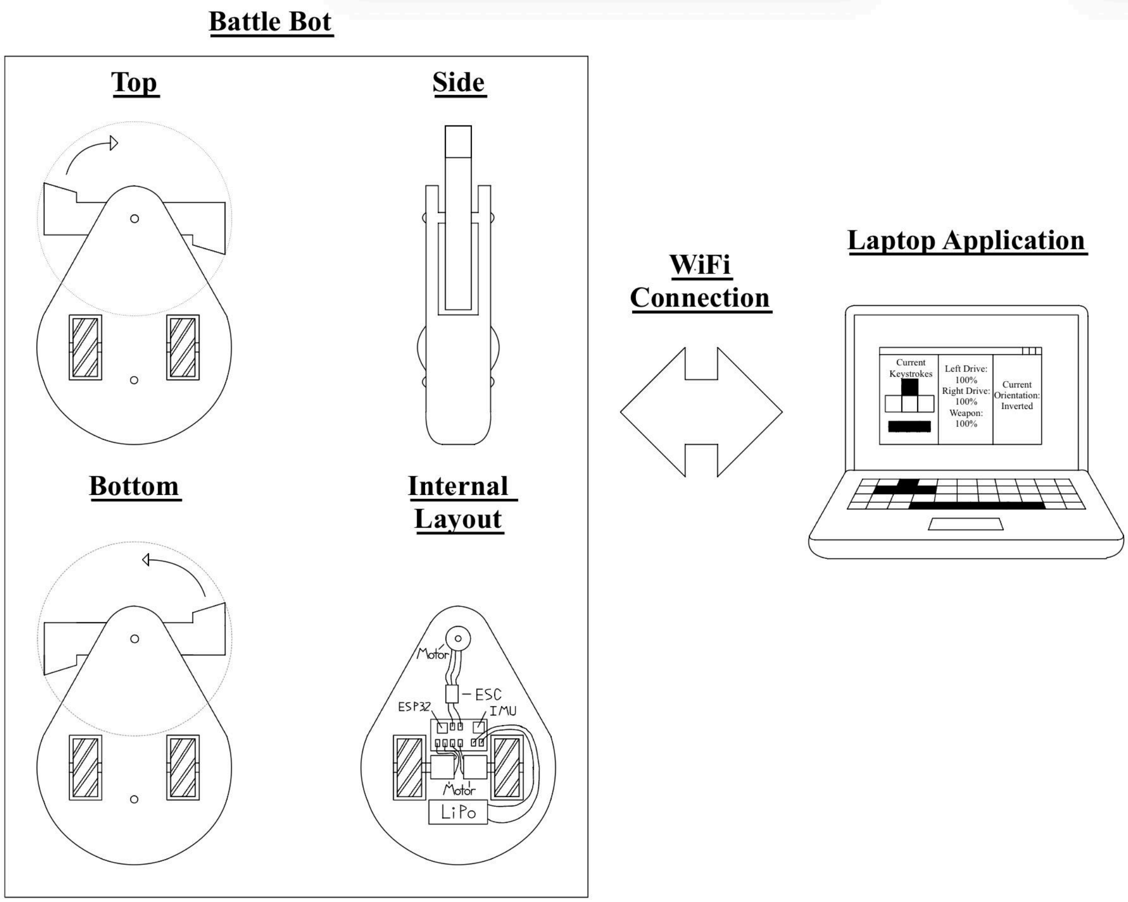

Figure 1: Visual aid of the Invertible Control Antweight Robot. This diagram illustrates the chassis geometry and the wireless connection between the ESP32 and the operator's laptop used to control the robot.

1.4 High-Level Requirements List

To consider our project successful, the antweight combat robot must fulfill the following:

1. The robot must adhere to all of the antweight class specifications, specifically maintaining a total mass less than or equal to 2lb and ensuring that the kinetic weapon system can be deactivated and brought to a complete rotational stop within 60 seconds of a command or failsafe trigger.
2. The onboard IMU and ESP32 control logic must detect a change in vertical orientation and invert the differential drive motor commands within 300 ms, ensuring that the operator perceives no lag in directional control relative to the new orientation.
3. The horizontal spinning bar must reach a combat-effective rotational velocity within 3 seconds of activation and demonstrate structural integrity to resume full speed rotation following a high-energy impact with an opponent or arena obstacle.

2 Design

2.1 Block Diagram

The system architecture of the combat robot is separated into six unique subsystems, as listed in Figure 2. This diagram serves as the structural map for power distribution and signal routing described in the following sections.

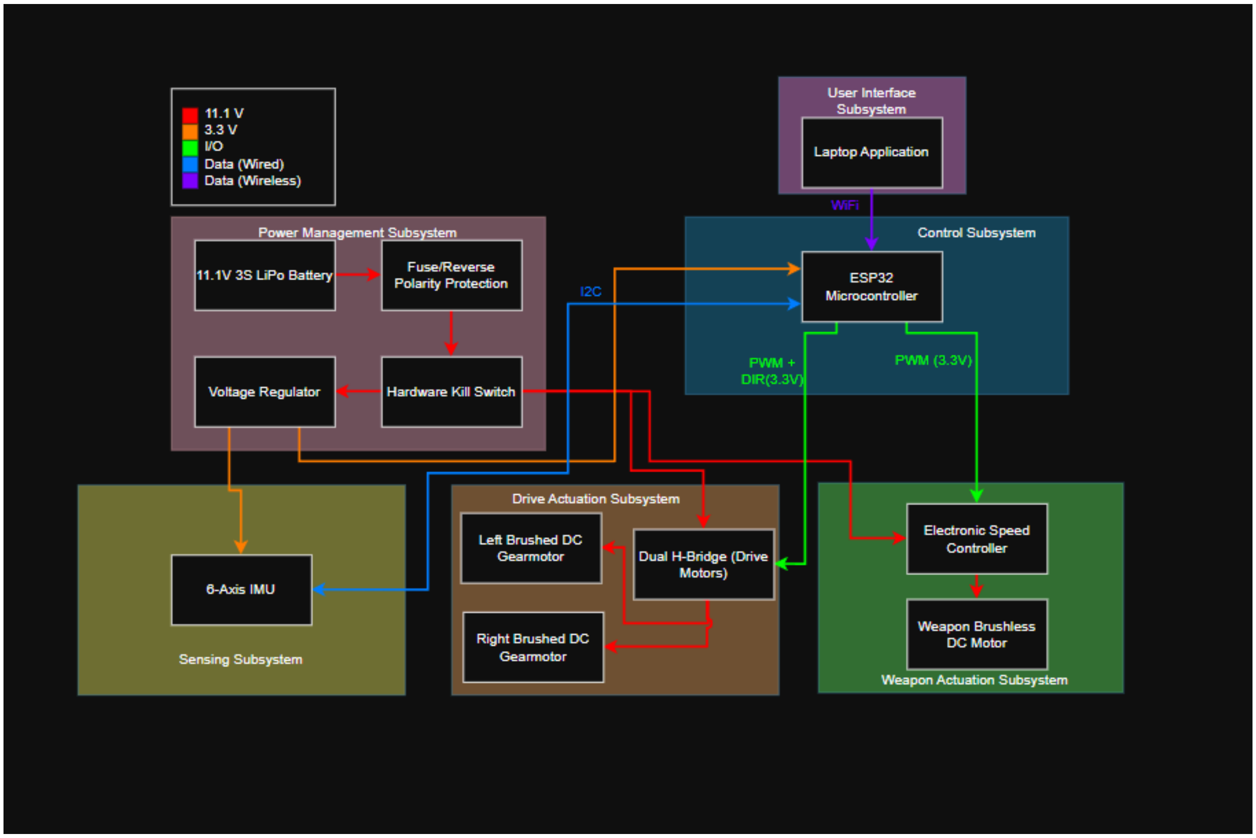

Figure 2: Block Diagram of the Invertible Control Antweight Robot. This diagram illustrates the separation of the robot into six primary subsystems, including the power management, control, sensing, drive actuation, weapon actuation, and user interface subsystems.

2.2 Physical Design

The physical design of the combat robot is optimized for high-speed combat and component protection. Shown in Figure 3, a diagram depicts the physical design of the robot, including the layout of all major internal components. The chassis will be fully 3D printed with defense in mind. This means there will be no sharp corners and all edges will be filleted to prevent opponent weapons from causing damage with impacts or grabbing hold of our robot. In addition, the primary weapon will be 3D printed and mounted at the front of the robot, where it will be directly driven with a high-torque brushless DC motor. This motor will be controlled by an electronic speed controller mounted internally and connected to the PCB. At the center of the system, two brushed DC gear motors are positioned to drive the enclosed wheels, which utilize a tank-style steering system. The robot will be controlled by the centrally located printed circuit board mounted directly to the chassis internally. This PCB will feature an ESP32 as the primary microcontroller responsible for communicating via WiFi with a laptop for control and an IMU to detect inversions using its accelerometer and gyroscope. Finally, the 11.1V 3S LiPo battery will be secured near the back of the robot, where it will be responsible for powering all motors and control systems. All internal components will be secured via integrated mounting standoffs and strong adhesive to prevent mechanical failures during the vibrations and impacts of battle.

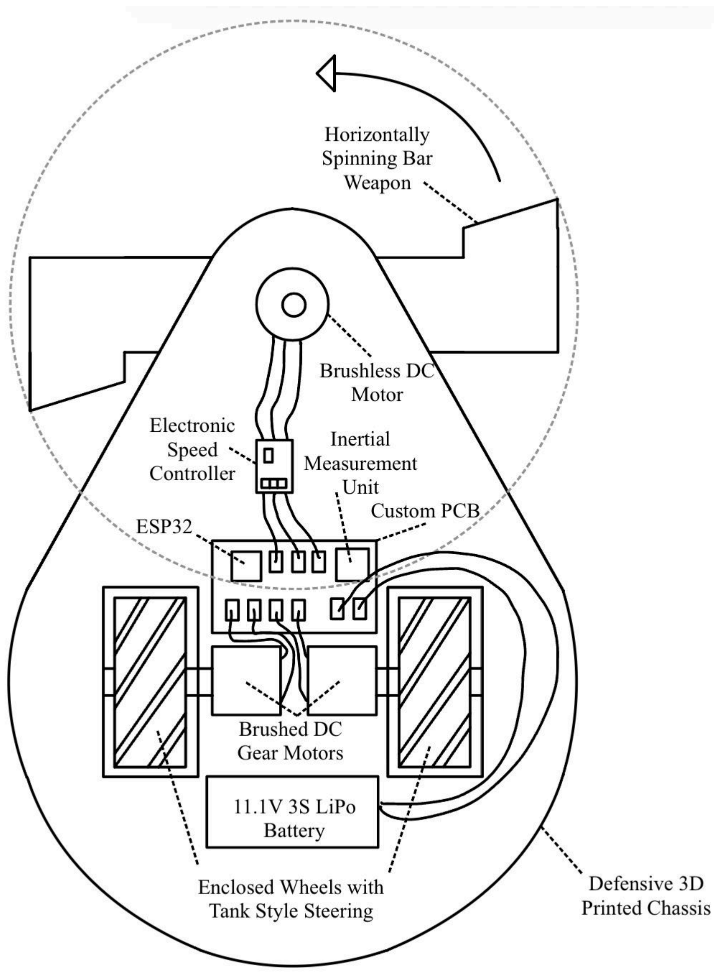

Figure 3: Physical Design of the Invertible Control Antweight Battle Bot. This diagram illustrates the internal layout of all primary components, such as the motors, battery, and custom PCB.

2.3 Subsystem Overview

2.3.1 Power Management Subsystem

The Power Management subsystem provides electrical power to all components of the robot. An 11.1V 3S LiPo battery supplies the high-current loads to the drive motors and the weapon motor, while the voltage regulator decreases the voltage down to 3.3V for the ESP32 and IMU. The subsystem includes a reverse polarity protector and a hardware kill switch to immediately disable all motor outputs. This subsystem interfaces directly with the Control Subsystem, Drive Actuation Subsystem, and Weapon Actuation Subsystem by applying the necessary power to all components.

Figure 4 shows the power regulation implemented within this subsystem. The diagram shows the TPS54302 step-down regulator and configuration used to convert our 11.1V input into a regulated 3.3V rail. The schematic shows the key requirements, including an input capacitor, inductor, output capacitor, and resistor feedback network. The efficiency curve shows how the performance varies across different input and output currents.

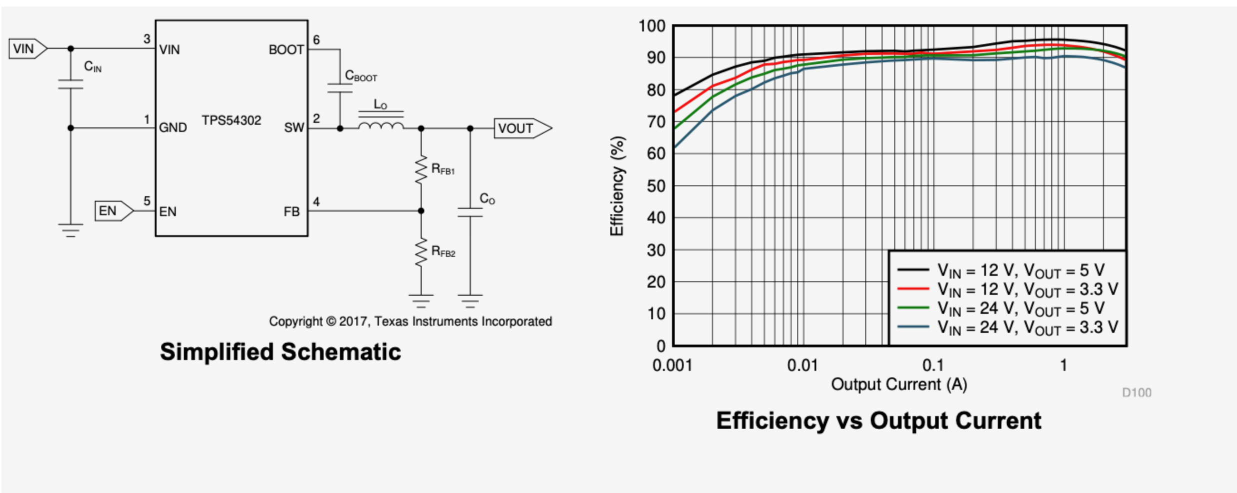

Figure 4: Power Management Subsystem voltage regulation architecture. The figure shows the simplified TPS54302 buck converter schematic. This is used to step down our 11.1V to 3.3V. The efficiency plot shows the regulator performance as a function of output current. This confirms the suitability of the voltage supply to our ESP32 and IMU.

2.3.2 Control Subsystem

The Control Subsystem is centered around the ESP32 microcontroller and is responsible for processing the user inputs and sensor data to generate the motor control signals. It gets wireless commands from a laptop and orientation data from the IMU via WiFi and I2C, respectively. It then generates PWM and directional signals for the dual H-bridge and a PWM signal for the electronic speed controller. Therefore, this subsystem works in tandem with all other subsystems of the robot as the ESP32 is the brains of the operation. This subsystem forms the core logic of the robot and coordinates all of the sensing and communication functions.

2.3.3 Sensing Subsystem

The Sensing Subsystem consists of a 6-axis IMU that provides accelerometer and gyroscope data to determine the orientation of the robot. The IMU communicates with the ESP32 over an I2C interface. This subsystem enables the robot's orientation-aware control logic. It detects when the chassis has been inverted and communicates that information to invert the controls properly. The Sensing Subsystem directly supports the Control Subsystem by supplying the real-time motion and orientation data required for remapping the drive signals on the spot. It is also important to recognize that this subsystem relies on the 3.3V supplied by the Power Management Subsystem.

Figure 5 shows the axis orientation of our 7-axis IMU. Proper alignment of the sensor axes relative to the robot is necessary for orientation and inversion logic. The diagram shows the directions for both linear acceleration and rotational motion.

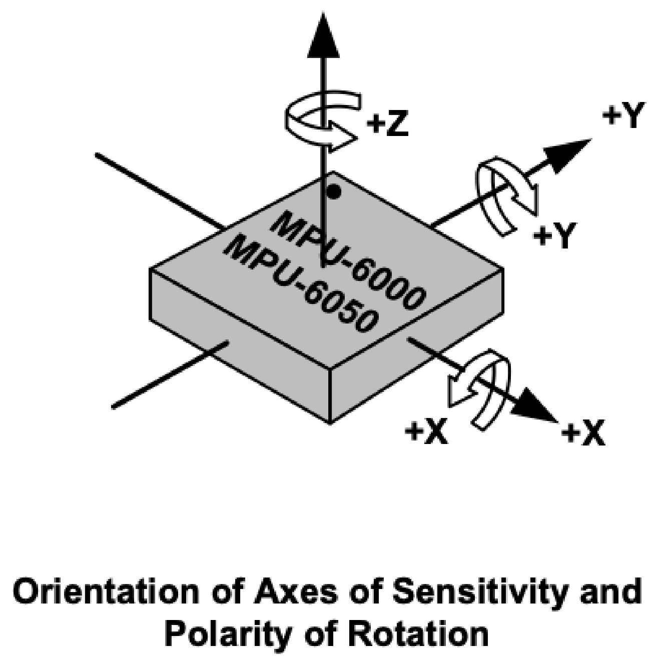

Figure 5: IMU axis orientation and rotation polarity reference. The diagram shows the MPU-6050 frame, indicating the positive X, Y, and Z axis and their directions of rotation. Correct mounting orientation ensures that the accelerometer and gyroscope measurements correspond to the robot's physical frame. This gives us reliable detection for chassis inversion.

2.3.4 Drive Actuation Subsystem

The Drive Actuation Subsystem consists of a dual H-bridge motor driver and two brushed DC gearmotors in a tank-style arrangement. These motors are controlled by the dual H-bridge driver that receives its PWM and direction signals from the ESP32. Interfacing with the Power

Management Subsystem and Control Subsystem, this subsystem converts electrical control signals to mechanical motion to drive the combat robot according to the user's commands.

2.3.5 Weapon Actuation Subsystem

The Weapon Actuation Subsystem consists of an electronic speed controller (ESC) and a brushless DC motor driving the spinning bar. The motor is controlled through the ESC that receives a PWM control signal from the ESP32. Interfacing with the Control Subsystem, this subsystem draws high current from the battery in the Power Management Subsystem and is activated or disabled based on commands from the Control Subsystem.

2.3.6 User Interface Subsystem

The User Interface Subsystem consists of a laptop application that communicates via WiFi with the ESP32. It allows the operator to send drive and weapon commands to the robot during combat. The laptop will utilize its keyboard as input to control the robot during battle in order to move forward, backward, turn, and spin up the primary weapon. This subsystem is the human-machine interface and is directly connected to the Control Subsystem.

2.4 Subsystem Requirements

2.4.1 Power Management Subsystem

The Power Management Subsystem enables the robot to operate safely within the constraints and antweight battlebot rules. It also allows the robot to maintain uninterrupted control logic, so the orientation-aware software and wireless control remain functional during impacts or transients. If this subsystem fails, the robot may become unsafe or uncontrollable.

Block Description:

The subsystem distributes energy from an 11.1V 3S LiPo battery to two power domains:

1. 11.1V_PROT: This is the direct battery rail feeding the drive H-bridge and weapon ESC input. The 11.1V provided must pass through a fuse, hardware kill switch, and reverse polarity protector prior to being distributed.
2. 3V3: This is the regulated power rail that feeds the ESP32, IMU, and logic motor inputs.

This subsystem also implements safety-critical protections:

1. A hardware kill switch that physically removes power to motor paths.
2. Reverse polarity protection to prevent damage from incorrect battery connection.
3. Fuse protection to interrupt fault currents.

Interface Requirements:

1. Battery input connector: 9.0-12.6 V, DC

2. 11.1V_PROT Output: 9.0-12.6 V supplied to H-bridge power input and ESC power input
3. 3V3 Output: 3.3 ± 0.1 V to ESP32, IMU, and any other logic
4. The kill switch must interrupt 11.1V_PROT to the motor supply path, not just the logic power

Table 1: Power Management Subsystem Requirements & Verification

Requirement	Verification Procedure
Battery Operating Range: The subsystem must operate from a 3S LiPo equivalent DC input from 9.0-12.6 V without damage or functionality loss.	**Equipment:** Bench DC supply and multimeter **Steps:** Disconnect the LiPo, connect the bench supply at the input terminals, set the current to a safe value of 1-2A for logic testing, have Vin sweep from 12.6V to 9.0V, and monitor the 3V3 rail for correct voltage. **Record:** Create a small table of Vins and Vouts and make sure that there is no abnormal behavior with the 3V3 line.
High-current rail: With input 9.0-12.6V and the kill switch on, the subsystem must provide 11.1V_PROT as Vin ± 0.20 V at the H-bridge power input and ESC power input. This must have a combined load of up to 10A continuous.	**Equipment:** Bench supply of up to 10A, multimeter, and oscilloscope **Steps:** Apply a Vin of approximately 11V, connect the load to 11.1V_PROT at the same PCB nodes as the H-bridge and ESC, and draw in 2A, 5A, and 10A, measuring the 11.1V_PROT at the load connection. **Record:** Vout stays in 3.2-3.4V across all points and record outputs in a table.
Regulated logic rail: The subsystem must provide 3.3V ± 0.10V for a load current of 0-0.6A and Vin of 9.0-12.6V.	**Equipment:** Bench supply, resistor bank, and multimeter **Steps:** Set Vin to 12.6V, apply loads on 3V3 rail, and send out: 0A, 0.1A, and 0.6A. Measure Vout at the ESP32 and IMU power pins, repeat with lower Vin Voltage. **Record:** Record Vout across all inputs in a table to ensure it stays within 3.2-3.4V.
Transient immunity: During a step load on 3V3 from 0.1 to 0.6A, the 3V3 rail must not dip below 3.10V for more than 5ms.	**Equipment:** Oscilloscope with a probe at the ESP32 3V3 pin and GND, and an electronic load **Steps:** Set Vin to 11.1V, configure the load step from 0.1A to 0.6A at 10Hz, measure 3V3 and the time under 3.10 V.

	Record: Take a picture of the oscilloscope and the minimum voltage duration.
Kill switch power isolation: When the kill switch is activated, the voltage at the H-Bridge and ESC supply pin must fall under 0.5V within 200ms.	**Equipment:** Multimeter or oscilloscope, bench supply or LiPo, and timer **Steps:** Power on the system, kill switch is on, probe the motor supply nodes, turn the kill switch off, and measure decay time until < 0.5V. **Record:** Record time to fall after disconnect over multiple trials to see if it passes consistently.
Reverse polarity protection: With input reversed at -12.0V for 10s, the subsystem must not have permanent damage and function correctly when fixed.	**Equipment:** Bench supply, multimeter, and IR thermometer, if possible **Steps:** Disconnect all motors, set current limit to 0.2-0.5A, apply a -12.0V for 10s, remove power, restore correct polarity, verify regular behavior of 3V3 rail. **Record:** Pass check if the 3V3 rail works correctly and the ESP32 boots correctly.
Fuse protection: The system must include a fuse that interrupts the battery input current when the current exceeds 30A.	**Equipment:** Fuse datasheet and current limit short **Steps:** Document the fuse rating and time-current curve, with the current limited supply, apply a controlled overcurrent validated with the datasheet, and confirm the fuse opens. **Record:** Use the test evidence of the fuse opening to confirm the intended behavior.

2.4.2 Control Subsystem

The control system is responsible for managing the operator commands and the robot operation. It must also issue the motor and weapon controls fast enough to keep the robot controllable and safe. If this block fails, the robot's core behavioral functionalities will cease to exist.

Block Description:

The subsystem consists of the ESP32. The primary responsibilities of the subsystem include:

1. Receive operator commands from the WiFi connection
2. Estimates the orientation from the IMU

3. Applies control logic.
4. Outputs motor control signals to:
    a. H-bridge motor
    b. ESC
5. Implements communication loss failsafe behavior

Interface Requirements:

1. IMU communication: I2C at 3.3V logic
2. 3V3 power input: 3.20-3.40 V from Power Subsystem
3. H-bridge inputs: 3.3V logic-level PWM and DIR signals
4. ESC input: Standard PWM (2ms pulse width and 50Hz)
5. Wireless communication: WiFi protocol supported by the ESP32

Table 2: Control Subsystem Requirements and Verification

Requirements	**Verifications Procedure**
Boot and Readiness: After 3V3 reaches the 3.20 V threshold, the controller must boot and begin generating valid outputs within 5.0s.	**Equipment:** Bench supply, oscilloscope, and serial terminal (USB/UART) **Steps:** Power the system to 3.3V, use the serial log to time the boot, probe a motor-control output pin, and confirm a valid output from the ESP32. **Record:** Screenshot of the “ready” message and a valid scope capture of the output.
IMU update rate: The controller must sample and compute the orientation estimate at 100 Hz or greater.	**Equipment:** Timestamps of the method, GPIO toggle pin, and oscilloscope **Steps:** Add a debug toggle each time an IMU sample is processed and measure the frequency over at least 5 seconds. **Record:** Measure the update rate over time in Hz.
Inversion detection Latency: When the robot goes from upright to inverted (IMU detects g force less than or equal to -0.8g for at least 30ms), the controller gives off an inverted state within 300ms.	**Equipment:** GPIO debug pin, oscilloscope, and rotation fixture **Steps:** Have the PCB in an upright position, add a connection to the GPIO output pin for detecting when inverted, rotate the board to be inverted, compute time difference from inversion to detection, repeat. **Record:** Record a table of latencies, this

	passes if it is detected and changed within 300ms each time.
Correct inverted control mapping: In the inverted state, the drive command will produce forward motion consistent with forward input within 1 control cycle(i.e., 10ms).	**Equipment:** Oscilloscope **Steps:** Force an inverted state via software or manually, apply forward input, and observe the direction pin and PWM responses. **Record:** Show the scope capturing the mapping changes within one update.
ESC command validity: The ESC control signal must be valid across the full range: PWM 1-2ms ± 0.05ms at 50Hz ± 2 Hz (subject to change).	**Equipment:** Oscilloscope **Steps:** Command the weapon throttle across the whole range, measure pulse width and rate at the ESC signal pin. **Record:** Show a correct table of the pulse width and frequency measurements.
Command latency: The latency for the operator input to motor change should be less than or equal to 100ms 95% of the time.	**Equipment:** Oscilloscope and buttons pressed to generate a GPIO signal **Steps:** Create a test where a known input toggles a marker, measure the time from the event to the change at the output, and repeat. **Record:** Show the success rate of the many trials.
Communication loss failsafe: If no valid command is received for more than 250 ms, the controller will set the motor outputs to 0 within 50ms.	**Equipment:** Oscilloscope and interrupt method **Steps:** Run a nonzero drive and weapon motor, cut the link, and measure the time from the last valid log to the settle time. **Record:** Record the fail-safe trigger time, settle time, and possible scope capture.

2.4.3 Sensing Subsystem

The Sensing Subsystem must provide reliable orientation detection to enable the control remapping. Incorrect orientation detection would result in wrong control remapping and cause reversed controls when the robot is inverted.

Block Description:

The subsystem consists of the 6-axis IMU, with an accelerometer and gyroscope, and a hardware routine that:

1. Configures the IMU and samples
2. Converts raw readings to physical units
3. Provides necessary values to the control subsystem
4. Detect invalid sensor states

Interface Requirements:

1. IMU communication: I2C at 3.3V logic
2. Power: 3.20-3.40 V
3. Data outputs to control: Accelerometer vector and gyro vector at 100 Hz or greater (rate can slide)

Table 3: Sensing Subsystem Requirements and Verification

Requirements	**Verification Procedure**
Sample rate: The IMU must provide valid samples to the controller at 100Hz or greater.	**Equipment:** Timestamp log and oscilloscope **Steps:** Toggle the log on each accepted sample and measure the frequency over a period of at least 5 seconds. **Record:** Record the measured sample rate in Hz.
Static gravity accuracy: When the robot is stationary, the measured acceleration magnitude must be 1.00g ± 0.10g in both upright and inverted orientations.	**Equipment:** Serial log and a steady board fixture **Steps:** Hold upright for 5s, record the average acceleration magnitude, and repeat when inverted. **Record:** If the mean for both is within the threshold, then it passes.
Axis sign correctness: In the defined upright orientation, A_z must be greater than 0.80g upright and less than -0.80g inverted.	**Equipment:** Serial log **Steps:** Hold the board upright and record the mean A_z repeat for inverted. **Record:** Record the values to confirm that we are within the threshold range.
Saturation handling: If any accelerometer or gyro axis hits full-scale, the system must flag that event within 20 ms and continue providing the most recent valid data there is.	**Equipment:** Serial log **Steps:** Configure a low full-scale range, induce a quick motion, and confirm saturation with the log. **Record:** Record the saturation event timestamps and the behavior seen.

2.4.4 Drive Actuation Subsystem

The Drive Actuation Subsystem converts the controller's power logic signals into torque at the drive motor. This enables motion and maneuvering. If it cannot deliver the required torque, the robot cannot be driven reliably.

Block Description:

This subsystem includes the dual H-bridge motor driver and the drive DC motors. The operation of this subsystem must:

1. Accepts PWM and direction from the control subsystem
2. Applies 11.1V_PROT to the motor in forward and reverse
3. Limits transient events: brief stalls and startup current/spikes
4. Provides an electrical interface for motor terminal verification

Interface Requirements:

1. Control inputs: 3.3V PWM + DIR at the desired 50Hz
2. Power input: 9.0-12.6V battery
3. Motor output: Two terminals to drive the motor
4. Mechanical output: Wheel/drive linked by speed/torque

Table 4: Drive Actuation Requirements and Verification

Requirements	Verification Procedure
Logic-level compatibility: The H-bridge must interpret control signals as valid logic between 0.8V and 2.0V when driven by a 3.3V Input.	**Equipment:** Multimeter, oscilloscope, and datasheet for reference **Steps:** Measure the GPIO high level at the H-bridge input pin and confirm it is over 0.8V and under 2.0V. **Record:** Measure values and record in between the thresholds.
PWM pass-through: For PWM command duty cycles of 10, 50, and 90%, the motor terminal voltage should be within 10% of the duty * 11.1V_PROT with no load.	**Equipment:** Multimeter, oscilloscope, and bench supply **Steps:** Lift the wheels so there is no load, apply 11.1V_PROT, go through all of the duty cycles, measure the motor waveform, and the average voltage. **Record:** Record the table of duty vs measured average terminal voltage and its percent error.

Bidirectional operation: The subsystem must support forward and reverse rotation. Changing direction will result in direction reversal within 200 ms at a constant duty command of less than 30%.	**Equipment:** Oscilloscope and visual confirmation **Steps:** Set a forward command for 2s at 30% duty, repeat but reverse direction, and observe the motor and direction change time until stable reverse. **Record:** Measure the time in ms and a short video/observation log.
Stall transient survivability: During operation at 50% duty, a forced stall of less than 0.5s must not permanently damage the driver, and the system must resume normal operations afterwards.	**Equipment:** Bench supply, oscilloscope, and IR thermometer is a bonus **Steps:** Command at 50% duty, physically prevent rotation for about 0.5s, release, and verify the motor runs normally. **Record:** Observe and note the shutdown behavior as well as the temperature.
Kill switch dominance: With the kill switch off, the motor voltage must be less than 0.5 V within 200ms regardless of control input state.	**Equipment:** Oscilloscope and kill switch. **Steps:** Run the motor at 30% duty, flip the kill switch off, and measure the time it takes for the voltage to drop to under 0.5 V. **Record:** Record the decay time and waveform of the voltage.

2.4.5 Weapon Actuation Subsystem

The Weapon Actuation Subsystem must provide the primary offensive capability. Failure of the subsystem eliminates the robot's primary objective and offensive mechanism.

Block Description:

This subsystem includes the electronic speed controller and bushless DC motor. Four key components of this subsystem are listed below:

1. The weapon motor must reach operational rotational velocity within 3 seconds of activation.
2. The system must allow a complete shutdown within 60 seconds of a kill command.
3. The ESC and motor must operate from the 9.0-12.6V battery rail.
4. The subsystem must withstand transient current spikes during rapid acceleration and impact.

Interface Requirements:

1. Control input: 3.3V PWM

2. Power input: 11.1V_PROT: 9.0-12.6V
3. Motor output: 3-phase for weapon motor
4. Mechanical output: weapon shaft with speed proxy

Table 5: Weapon Actuation Subsystem Requirements and Verification

Requirements	**Verification Procedure**
Valid throttle interpretation: For control pulse widths of 1.00, 1.50, 2.00 ms (±0.05 ms) at 50 Hz, the ESC must command the respective throttle states.	**Equipment:** Oscilloscope and controller output **Steps:** Drive ESC input with the different pulse widths, confirm that the ESC arms and responds appropriately. **Record:** Record the scope captures of the pulse width and the observation log.
Arming and safe start: Once power is applied, the ESC must arm within 5s and must not start the weapon motor until a throttle command above the minimum is received.	**Equipment:** Stopwatch and oscilloscope **Steps:** Apply power with the minimum throttle, time until the ESC is armed, and verify the motor remains stopped at the minimum throttle. **Record:** Measure the time it takes for the ESC to arm and the confirmation of no unintended spinning.
Spin-up response: From stopped to 80% of the command throttle, the weapon motor must reach a steady speed within 2s.	**Equipment:** High-speed video marking and a stopwatch **Steps:** Start from a stopped position, step up to 80% of throttle, and measure the time to reach a steady speed. **Record:** Record the time to steady speed over at least 5 trials. Make a table of the results.
Stop behavior: When commanded from any throttle to minimum throttle, the weapon motor electrical drive must be removed within 100ms.	**Equipment:** Oscilloscope to probe one motor phase **Steps:** Run the throttle at 60%, cut the link, measure the time until the ESC signal ceases, and the motor drive stops. **Record:** Measure the response timing and include captures across trials.
Continuous current capability: The ESC and wiring path must sustain 15A or greater of continuous current without thermal	**Equipment:** Current measurement and an IR thermometer **Steps:** Load the weapon with current, make

shutdown.	sure it is at least 15A for 60s, and measure the current and the temperatures. **Record:** Record the data in a current and temperature vs time table.
Kill switch: With the kill switch off, the ESC supply pin voltage will fall below 0.5V within 200ms.	**Equipment:** Oscilloscope and kill switch. **Steps:** Run the weapon at 40%, flip the kill switch off, measure the ESC Vin decay time, and confirm the motors stop. **Record:** Record the decay time and observation of the motor.

2.4.6 User Interface Subsystem

The user Interface Subsystem must provide reliable remote operation of drive and weapon systems. Without wireless control and failsafe behavior, the robot cannot be safely or effectively operated.

Block Description:

This subsystem includes the operator-facing input and any on-robot indicators. Proper operations of this subsystem include:

1. Generates drive and weapon commands at a defined update rate
2. Transmits them to a control subsystem
3. Indicates connection and arming status to the operator

Interface Requirements:

1. Commands transmitted: Drive command and weapon throttle
2. Update rate: At least 10 Hz for command refresh
3. Feedback: Link indicator (LED or UI)

Table 6: UI Subsystem Requirements and Verification

Requirements	Verification Procedure
Command update rate: The UI must transmit command updates at least 10 Hz while the operator is commanding motion.	**Equipment:** Receiver counter and timestamps **Steps:** Hold a constant command for 10 seconds, count the number of packets received, and the rate. **Record:** Measure the update rate in Hz.
Command range correctness: The UI must	**Equipment:** Serial log of the received

generate drive commands spanning -100% to +100%, and the weapon throttle from 0% to 100% within 5% for each step.	commands. **Steps:** Sweep UI inputs across a full range, record the min and max as well as the step sizes the controller observes. **Record:** Record a table of the commands and the received values.
Link status indication: The UI must display or indicate "connected" within 3s of a successful connection and "disconnected" within 1s after link loss.	**Equipment:** Stopwatch, UI images, and receiver log timestamps **Steps:** Establish a link and time to connect indication, disable the transmitter, and time to disconnect indication. **Record:** Record the measured times as well as screenshots.
Arming action safety: The UI must require an explicit arming toggle before allowing nonzero weapon throttle.	**Equipment:** UI behavior observation and received command logs **Steps:** Connect without arming, verify weapon commands stay at 0%, arm, command weapon movement, disarm, and make sure it returns to 0%. **Record:** Record the logs of weapon commands and ensure they equal correct timing concerning 0% movement.

2.5 Tolerance Analysis

A primary risk in this design is voltage instability on the 3.3V logic rail. When the weapon motor accelerates, stalls briefly on impact, or when the drive motors change direction quickly, large current spikes can cause a temporary voltage sag. If the regulated 3.3V supply drops excessively, the ESP32 may reset. This could cause a temporary loss of communication and orientation-aware control. Consequently, this would compromise the control remapping needed. We can approximate the drop during a transient using the peak current and load impedance.

$$\Delta V \approx \Delta I \cdot \Delta t \,/\, C + \Delta I \cdot ESR$$

Using the equation derived from Ohm's law as shown above and the fundamental capacitance equation, the first term to the right of the equals sign demonstrates that if the capacitance drops too low, we could see a significant voltage sag. And the right term shows that if the ESR (Equivalent Series Resistance) is too high, we would also receive a voltage sag. To tolerate this, the design will incorporate a low impedance under the bulk capacitance on the 3.3V rail. By additionally making sure that the PCB layout is carefully designed to minimize the shared impedance between the motor current paths, we can ensure that transient current demands do not disrupt the stability of control during combat conditions.

More specifically, the minimum allowed rail during a transient is 3.10V. This gives us an allowed drop of up to 0.20V from the 3.30V stability. We can test the hypothetical droop with different elements. We will assume a ΔI of 0.50A for the logic rail and a Δt of 300us, to stay conservative for the response time.

Table 7: Sample Voltage Sag Estimates

Trial	C	ESR	ΔV_C	ΔV_{ESR}	ΔV total	Under 0.2V
1	220 uF	30 mΩ	0.5 * 300e-6 / 220e-6 = 0.682V	0.015V	0.697V	No
2	470 uF	20 mΩ	0.319V	0.010V	0.329V	No
3	1000 uF	20 mΩ	0.150V	0.010V	0.160V	Yes
4	1500 uF	12 mΩ	0.100V	0.0075V	0.108V	Yes

Table 7 shows that if the regulator response time is within a few hundred microseconds, we need an order of about 1000uF bulk capacitance to stay within the 0.20V droop.

3. Cost and Schedule

3.1 Cost Analysis

The final list of all parts required for purchase in order to complete the final robot with a custom PCB is shown below in Table 8. The total cost of these parts before taxes and fees is $130.36 before shipping. After an approximate 5% shipping cost totaling to $6.52 and 10% tax totaling to $13.04, the total cost of parts is $149.92. Following graduation from ECE at UIUC, we can expect to earn a salary of around $55/hr. Each team member estimates that they will spend an average of 10 hours per week over the course of 15 weeks on the project. Therefore, the cost of labor will be $55/hr x 150hr x 2.5 = $20,625 per team member. Accounting for both team members, the total cost of labor will be $41,250. The final cost for the project will be approximately $41,250 + $149.92 = $41,399.92.

Table 8: Parts List Including Cost

Description	Manufacturer	Part Number	Quantity	Cost	Link
2 pk 860 mAh 80 C 11.1V 3S LiPo Battery	OVONIC	B09CTSCWYM	2	$28.99	Link
Male and Female XT30 Connector Pack	DFRobot	FIT0586	2	$3.80	Link
P-Channel 55V Through Hole MOSFET	Infineon Technologies	IRF4905PBF	1	$2.89	Link
Brushed DC Gearmotor 12V 300 RPM GA12 N20	Anreak	B0D976H39X	2	$15.99	Link
40A Brushless ESC 2-6S UBEC 3A 5V	ReadyToSky	B08HWQ58QX	1	$20.99	Link
ECO II Series 2207 Motor 2400kv	EMAX	789862853527	1	$20.99	Link
11.1V 3S LiPo USB Battery Charger	Blomiky	B082YZ2WRT	1	$9.99	Link
Fuse Board Mount 2A 63VAC 63VDC 1206	Littlefuse Inc.	0466002.NR	1	$1.09	Link
Diode 75 V 150mA Surface Mount Mini MELF	Taiwan Semiconductor Corporation	LL4148 L0G	1	$0.13	Link
47 µF ±20% 10V Ceramic Capacitor	TDK	C2012X5R1	1	$0.70	Link

X5R 0805	Corporation	A476M125AC			
Buck Switching Regulator IC Positive Adjustable 0.596V 1 Output 3A SOT-23-6 Thin	Texas Instruments	TPS54302DDCT	1	$1.85	Link
10 µH Surface Mount Inductor 3.3 A Saturation Current	Coilcraft	XGL5030-103MEC	1	$2.49	Link
45.3 kOhms ±1% 0.25W, 1/4W Chip Resistor 0805	Vishay Dale	CRCW080545K3FKTA	1	$0.15	Link
22 Ohms ±1% 0.125W, 1/8W Chip Resistor 0805	YAGEO	RT0805FRE0722RL	2	$0.20	Link
20V 5A 7.35mm Female Surface Mount, Right Angle Type-C SMD	Korean Hroparts Elec	TYPE-C-31-M-12	1	$0.18	Link
Diode 100 V 150mA Surface Mount SOD-123	Diodes Incorporated	BAT46W-7-F	1	$0.30	Link
0.1 µF ±10% 10V Ceramic Capacitor X7R 0805	KEMET	C0805X104K8RACTU	1	$0.51	Link
Motor Driver Power MOSFET Parallel 24-SSOP	Toshiba Semiconductor and Storage	TB6612FNG,C,8,EL	1	$1.82	Link
100 Ohms ±1% 0.5W, 1/2W Chip Resistor 0805	YAGEO	SR0805FR-47100RL	1	$0.17	Link
4.7 kOhms ±1% 0.125W, 1/8W Chip Resistor 0805	YAGEO	RC0805FR-104K7L	2	$0.20	Link
2200 pF ±1% 50V Ceramic Capacitor C0G, NP0 0805	KEMET	C0805C222F5GACTU	1	$1.93	Link
5PCS MPU-6050 IC Chip in Stock	TDK Corporation	MPU-6050	1	$15.00	Link
				Total:	$130.36

3.2 Schedule

Shown in Table 9 is the division of labor and schedule for work on the combat robot throughout the semester. It is important to note that since our group consists of only two members, we will each have to contribute more time per week to account for a smaller group. For this reason, each member may have multiple tasks per week if they are smaller. In addition, as we near the end of the semester, more work will be spent together as a group to troubleshoot issues and bugs that we will likely face.

Table 9: Schedule for Work Throughout Semester

Timeframe	Task	People
Feb 23 - March 1	1. Finalize subsystem requirements and R&V tables 2. Complete tolerance analysis 3. Finalize the itemized part list 4. Order all parts for the prototype 5. Finish Design Document 6. Finish Design Review presentation **Design Document Due**	1. Ben 2. Ben 3. Jack 4. Jack 5. Everyone 6. Everyone
March 2 - March 8	1. Regulator prototype and transient testing 2. ESP32 firmware base 3. Motor driver bench test 4. IMU orientation logic validation 5. Complete breadboard prototype **Design Review**	1. Ben 2. Jack 3. Ben 4. Jack 5. Everyone
March 9 - March 15	1. Firmware refinement and build out laptop application 2. PCB layout revision (if needed) 3. Simulation of power rail transient events 4. Complete Teamwork Evaluation I **Breadboard Demo Due**	1. Jack 2. Ben 3. Ben 4. Everyone
March 23 - March 29	1. PCB inspection test 2. Power rail bring-up 3. Flash to ESP32 4. Sensor calibration 5. Motor subsystem integration	1. Everyone 2. Ben 3. Jack 4. Jack 5. Everyone
March 30 - April 5	1. Research and test different 3D printing filament types 2. Begin 3D printing, CAD, and prototyping 3. Complete Individual Progress Reports	1. Ben 2. Jack 3. Everyone
April 6 - April 12	1. Weapon motor stall testing 2. Continue chassis prototyping 3. Work on securing hardware in the chassis 4. Fail-safe testing	1. Ben 2. Ben 3. Jack 4. Everyone

	Progress Demo Due	
April 13 - April 19	1. Continue chassis prototyping 2. Optimize system	1. Everyone 2. Everyone
April 20 - April 27	1. Reliability testing 2. Bot reinforcement 3. Final tuning 4. Complete presentation slides 5. Complete video assignment 6. Demo rehearsal **Mock Demo, Mock Presentation, and Video Assignment Due**	1. Everyone 2. Everyone 3. Everyone 4. Ben 5. Jack 6. Everyone
April 27 - May 3	1. Final paper writing 2. Final preparation for demo 3. Final preparation for presentation **Final Demo and Final Presentation Due**	1. Everyone 2. Everyone 3. Everyone
May 3 - May 7	1. Finish final papers 2. Complete lab checkout 3. Turn in lab notebooks **Final Papers and Lab Notebooks Due**	1. Everyone 2. Everyone 3. Everyone

4. Ethics, Safety, and Societal Impact

This project has been designed with careful consideration of societal impact, ethical responsibility, and safety, in accordance with ECE 445 course guidelines and the IEEE Code of Ethics [1]. The antweight combat robot addresses an educational and recreational engineering challenge rather than a public-facing issue. As such, its societal impact is primarily educational in nature. It provides hands-on experience in embedded systems, control logic, power electronics, and mechanical design. The project promotes engineering best practices related to system reliability, safety-conscious design, and responsible use of hazardous pieces within a controlled environment. In accordance with IEEE Code Section 1, Rule 1, we hold paramount the safety of the public by ensuring that all high-energy testing will occur within a containment arena [1].

From a public safety perspective, the robot is intended exclusively for use in supervised lab settings and ant-weight combat competitions. The design complies with established ant-weight competition rules [2], which are specifically structured to mitigate risks associated with high-speed kinetic weapons and high-current electrical systems. Environmental and global impacts are addressed through the responsible procurement of components and commitment to recycling LiPo batteries and electronic waste through campus-approved hazardous material programs. No environmental, economic, or global harm is anticipated, as the project does not involve large-scale autonomous control or production.

From an economic and social standpoint, the project emphasizes responsible engineering while adhering to constraints. The entire system uses commercially available components and avoids reliance on rare or sensitive materials. By designing the bot within the competition class, the project shows how complex systems can be implemented safely and affordably. Socially, this project contributes to the broader robotics community by showing good ethical design in competitive robots. Globally, this product does not introduce any scalable technologies that could cause major harm. Instead, it reinforces ethical design in small-scale systems that align with public welfare principles discussed in the IEEE ethical guidelines [1].

Relevant engineering standards and guidelines have been incorporated into the design process. Electrical systems adhere to standard low-voltage design practices, including proper current limiting, voltage regulation, and protection against reverse polarity and short circuits [3]. Wireless communication and embedded control logic follow IEEE 802.11 standards to ensure signal integrity and minimize interference with other devices [4]. The project aligns with IEEE engineering standards and ethical guidelines by prioritizing safety, transparency in design decisions, and accountability for system behavior [1].

In addition to IEEE standards, the project adheres to general electrical safety practices, including overcurrent protection, insulated wiring, proper connector selection, and controlled battery charging procedures. The mechanical design follows the competition safety standards that define the spin-down and weapon requirements [2]. System implementation follows the industry practices for signal integrity and power control as outlined in manufacturer documentation [3], [4]. These standards are what really guide the design to ensure we follow established engineering norms.

Ethically, the project raises no concerns related to data privacy, surveillance, or misuse, as it does not collect, store, or transmit personal or sensitive information. There is no human or animal testing involved. All design decisions are made with the intent to minimize risk to operators and spectators [5]. This is to keep it consistent with professional engineering responsibility.

While the project is educational in nature, it does involve high-current electrical systems and large amounts of kinetic energy. A combat robot could pose risks if used outside of regulated environments. So this system is intentionally designed without autonomous capabilities. This robot strictly requires human control for activation and control. We have also introduced motor shutdown routines and communication-loss safeguards since the proposal phase. These updates reflect the continual ethical evaluation needed throughout the development lifecycle [5].

Safety considerations are central to the project design. Electrical safety is addressed through the use of proper battery handling procedures [3] and an accessible hardware kill switch to immediately disable all motor outputs. This design choice prioritizes the safety of the operator and spectators in the event of a control malfunction [1]. Mechanical safety is ensured by enclosing the weapon system within the robot's chassis envelope when inactive and by following competition-mandated weapon spin-down requirements [2]. During development and testing, all laboratory safety protocols will be followed, including the use of protective equipment and controlled testing environments consistent with OSHA safety standards. Collectively, these measures ensure that the project meets the ethical, safety, and professional standards expected of an ECE 445 senior design project.

The project follows a structured safety protocol during development and testing. All battery charging will happen using balance chargers for a LiPo battery. The batteries will never be charged unattended. Before each testing session, wiring, insulation, and connection polarity will be visually inspected. The hardware kill switch will be verified as operational before enabling motor outputs. During weapon testing, the robot will be placed in a contained enclosure while operators and observers maintain a safe distance. All of this adheres to competition rules [2].

During demonstrations, the robot will stay disabled until put within the approved area. Power will only be enabled immediately before testing and disabled right afterwards. These procedures ensure that we, the developers, and observers are protected from mechanical and electrical hazards. Collectively, these documented strategies satisfy all safety requirements for the project's high-current, electrical, and mechanical systems.

References

[1] IEEE, *IEEE Code of Ethics*, Institute of Electrical and Electronics Engineers, 2020. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html

[2] Robot Combat Events (RCE), *Antweight Combat Robot Safety and Weapon Regulations*, 2025. [Online]. Available: https://www.robotcombatevents.com

[3] Texas Instruments, *Designing Motor Driver Power Systems for High Transient Loads*, Application Report SLVA959, 2021.

[4] Espressif Systems, *ESP32-WROOM-32 Technical Reference Manual*, Espressif Systems, Shanghai, China, 2023.

[5] D. Jones and M. Ladd, *Introduction to Combat Robotics*, McGraw-Hill, New York, NY, 2015.