Invertible Control Antweight Battle Bot

By

Ben Goldman

Jack Moran

Final Report for ECE 445, Senior Design, Spring 2026

TA: Haocheng Bill Yang

May 6, 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 (IMU) and an Espressif 32-bit Microcontroller (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 that delivers powerful blows to opponents. The 3D-printed chassis is designed with defense in mind, featuring few edges, corners, and other vulnerabilities that could be exploited by weapons, preventing them from damaging or grabbing the robot. At the center, a custom Printable Circuit Board (PCB) consolidates 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 addressed the technical challenge of designing and building a high-impact antweight combat robot that maintained operational control while inverted. This report outlines a dual-sided 2 lb robot that uses an IMU to drive a feedback loop to automatically remap drive logic to the controller, ensuring steering in any orientation. The following sections detail the battle-ready systems, including hardware and software subsystems, used to construct a highly capable and powerful antweight combat robot.

1.1 Problem

In the high-stakes environment of antweight combat robotics, all robots must weigh under 2 lb 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 self-righting system often leads to a knockout, as the opponent can take advantage of the vulnerability.

1.2 Solution

Our solution is an invertible control ant-weight combat robot designed for maximum offensive output and intuitive operation. The physical architecture of the device features 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 relies on two recessed wheels that protrude very slightly from both the top and bottom of the chassis. This allows for a "tank-like" differential drive system. The main innovation lies in the integration of a 6-axis IMU 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 of the robot are automatically inverted. The inversion of the control logic ensures that with the press of the forward input from the laptop, the robot always moves in the direction the weapon is facing. By handling the orientation logic in software, the implementation achieves a seamless transition that allows for continuous combat effectiveness.

1.3 High-Level Requirements List

At the beginning of the project, we outlined three high-level requirements that would be used to determine success for our project. The following three requirements have been satisfied in full:

1. The robot must adhere to all of the antweight class specifications, specifically maintaining a total mass less than or equal to 2 lb 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.

1.4 Subsystem Description Overview

The system architecture of the combat robot is divided into six subsystems, as shown in Figure 1. This diagram serves as the map for power distribution and signal routing described in the following sections.

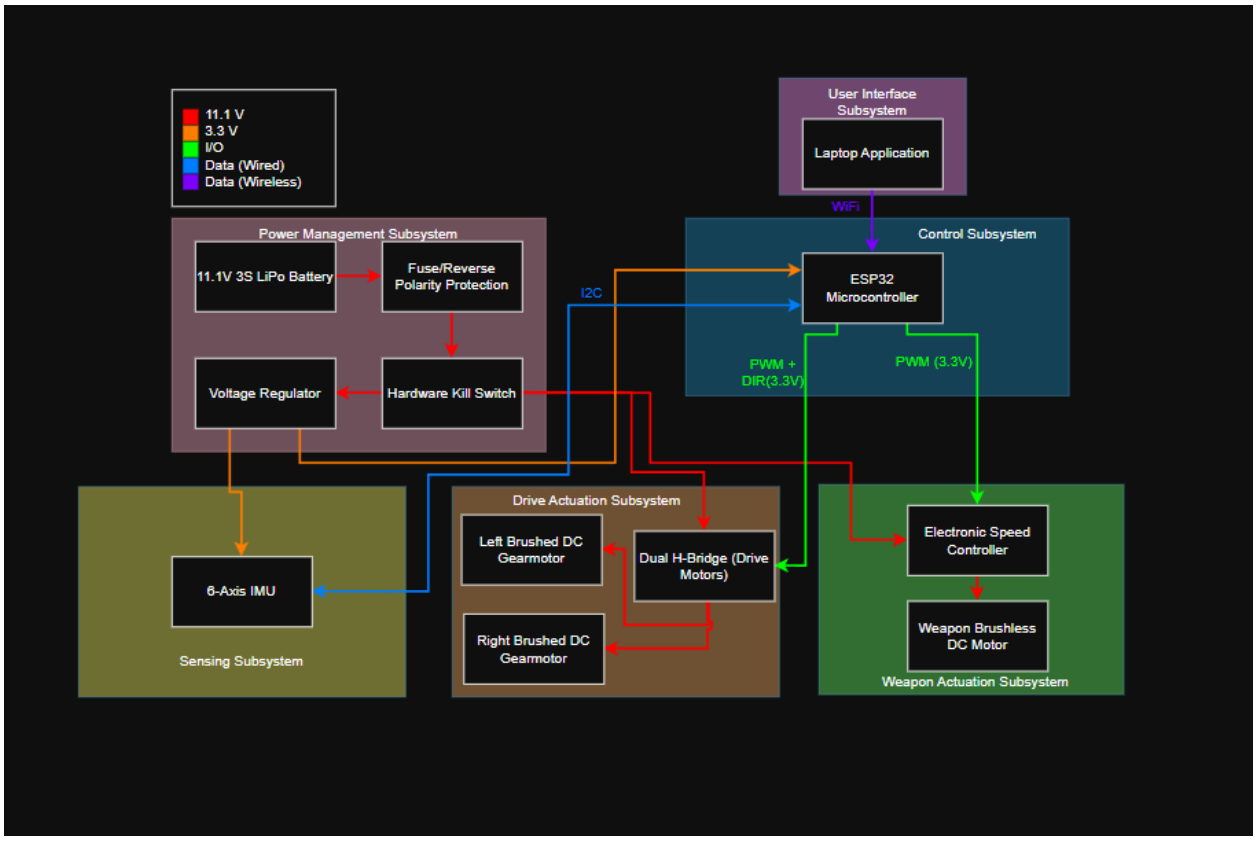

Figure 1: Block Diagram of the Invertible Control Antweight Robot. This diagram illustrates the separation of the robot into six primary subsystems, including the power management, control, sensing, drive actuation, weapon actuation, and user interface subsystems.

1.4.1 Power Management Subsystem

The Power Management subsystem provides electrical power to all components of the robot. An 11.1 V 3S Lithium Polymer Battery (LiPo) supplies high-current loads to the drive motors and the weapon motor, while the voltage regulator reduces the voltage to 3.3 V 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.

1.4.2 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 Wireless Fidelity Communication Standard (WiFi) and Inter-Integrated Circuit Communication Protocol (I2C), respectively. It then generates Pulse-Width Modulation (PWM) and directional signals for the dual H-bridge and a PWM signal for the Electronic Speed Controller (ESC). Therefore, this subsystem works in tandem with all other subsystems of the robot as the ESP32 acts as the brain of the operation. This subsystem forms the core logic of the robot and coordinates all of the sensing and communication functions.

1.4.3 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 supports the control subsystem by supplying the real-time orientation data required for remapping the drive signals. This subsystem relies on the 3.3 V supplied by the power management subsystem.

1.4.4 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.

1.4.5 Weapon Actuation Subsystem

The weapon actuation subsystem consists of an ESC, a motor driver, 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.

1.4.6 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. Design

2.1 Equations & Simulations

A primary risk in this design is voltage instability on the 3.3 V 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 3.3 V 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.

The transient voltage drop on the 3.3 V rail is approximated by Equation (1) shown below.

$$\Delta V \approx \Delta I \cdot \Delta t \,/\, C \; + \; \Delta I \cdot ESR \tag{1}$$

ΔI is the transient current step, Δt is the regulator response time, C is the bulk capacitance, and ESR is the equivalent series resistance of the capacitor network.

The minimum allowed voltage during a transient is 3.10 V. This gives us an allowed drop of up to 0.20 V from the 3.30 V stability. We can test the hypothetical drop with different elements. We will assume a ΔI of 0.50 A for the logic rail and a Δt of 300 us, to stay conservative for the response time. We can see these results below in Table 1.

Table 1: 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

2.2 Design Alternatives

While working on the weapon actuation subsystem, a significant design inconsistency emerged when transitioning from the off-the-shelf ESC to our triple half H-bridge motor driver using the DRV8313PWPR. We originally planned on using a standard high-kV brushless DC motor. However, when driven by the DRV8313PWPR, which was controlled by the ESP32, the motor failed to continuously rotate. Instead, the motor often stalled and emitted an audible hum. We were able to determine that this issue was a result of the low internal resistance. When a voltage step was applied, there would be an instantaneous current spike that exceeded the maximum current rating of the DRV8313PWPR. As a result, the chip's internal overcurrent protection circuitry would detect the event and assert an internal hardware interrupt. This automatically forced the bridges to a high impedance state to prevent damage, which stopped the motor.

To resolve this hardware incompatibility, we replaced the standard brushless DC motor with a 2208 Gimbal Motor rated at 90 KV. This design alternative proved highly successful due to the motor's significantly higher internal winding resistance. This helped to act as a natural current limit, keeping the maximum phase current below the threshold of the chip. To address the issue of the motor's lower starting torque against the rotational inertia of the weapon bar, we made an adjustment in the software. We built in a spin-up feature to quickly change the step delay until it reached its final operating speed. Using a ramp rate of 5 µs per step to a final target delay of 1,500 µs proved highly successful. This design alternative, in replacing the motor when transitioning from ESC to a triple half H-bridge, was a crucial adaptation that successfully reconciled our hardware constraints with the mechanical demands of the weapon system.

2.3 Design Justification

The system was designed modularly to isolate power, control, sensing, and actuation functions. This structure simplifies the debugging process and ensures that failure in one subsystem does not propagate to the whole system. Design decisions were driven primarily by electrical stability, response time, and performance under combat conditions. Shown in Figure 2 is the schematic used to create our PCB for the robot, which includes all subsystems, excluding the user interface subsystem.

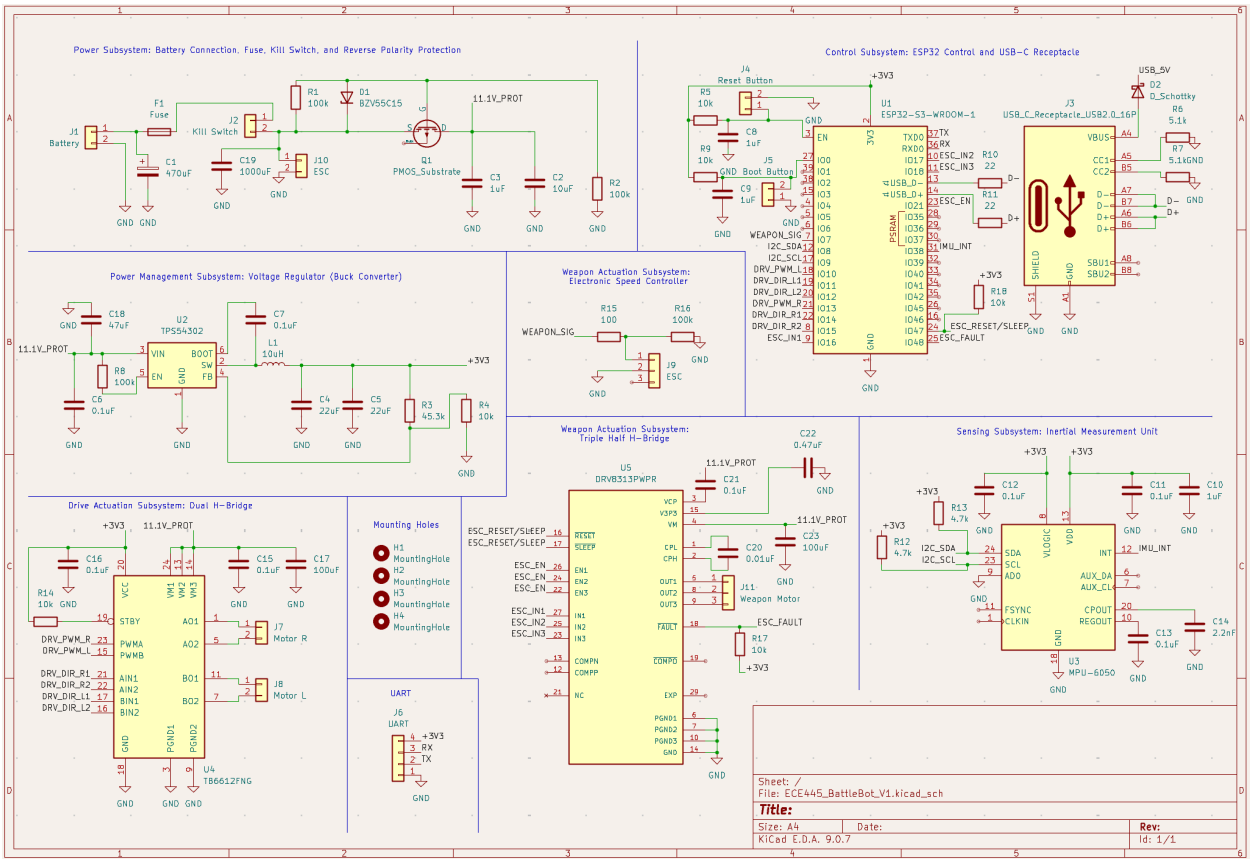

Figure 2: Schematic used to create the PCB. This diagram illustrates the separation of the electronics into five sections for the power management, control, sensing, drive actuation, and weapon actuation subsystems.

2.3.1 Power Subsystem

As shown in Section 2.1, we used Equation (1) in the construction of the power subsystem. The transient voltage drop on the 3.3 V rail is a critical constraint due to current spikes from the motor operation. Using a worst-case transient of ΔI = 0.50 A and Δt = 300 us, the analysis in Table 1 shows that capacitance values below 1000 uF result in voltage drops exceeding 0.20 V. Specifically, a 470 uF bulk capacitance has an estimated drop of about 0.329 V, which risks damage to the ESP32.

Based on this analysis, a design value of 1000 uF of capacitance was chosen. This results in a predicted drop of about 0.16 V, satisfying the requirement. This selection was further validated during testing of a transient load, where we observed no brownout and successful operation. Also, the TPS54302 buck regulator was selected based on its current capacity, and this ensured that the assumed response time used in the analysis remains under real operating conditions. The schematic for this subsystem is shown in Figure 2.

2.3.2 Control Subsystem

The control subsystem, as shown in Figure 2, satisfies a maximum inversion response time of 300 ms. Testing results show measured inversion response times at 58-74 ms, well beneath this range. A software-based inversion approach was chosen because it provides a deterministic response time that is also bounded by the IMU sampling rate. This guarantees a worst-case response time of about one control cycle plus our detection latency. We also evaluated how a manual inversion compensation would work out. This showed us that there would be too much human latency for a typical reaction time and that our software-based approach provides a more consistent response. Therefore, the selected approach satisfies both performance and reliability requirements.

2.3.3 Sensing Subsystem

The sensing subsystem, as shown by its schematic in Figure 2, reliably distinguishes between upright and inverted states. This is defined by the requirement that $|a_z| \geq 0.8$ g. Testing confirmed that the measured values were consistently within ± 1.0 g ± 0.1 g when stationary. The MPU-6050 IMU was also selected based on its ability to provide continuous acceleration data to the ESP32, its sufficient resolution to detect inversion thresholds, and low amounts of noise relative to the ± 0.8 g threshold.

We rejected alternative methods of sensing, including tilt switches and limit switches, because they do not provide continuous measurement and are falsely triggered more often during impact. The IMU-based approach ensures that inversion detection remains reliable even under dynamic motion, as confirmed by consistent threshold crossing during testing.

2.3.4 Drive Actuation Subsystem

The drive subsystem produces predictable torque and direction with PWM control. The relationship between the input PWM duty cycle and the motor voltage follows Equation (2) linearly.

$$V_{avg} = (Duty\ Cycle\ \%) \cdot V_{in} \tag{2}$$

Testing confirmed that the measured motor terminal voltage remained within 10 % of the expected values across all duty cycles tested. This perfectly satisfied the subsystem requirements. A differential

drive configuration was selected because it minimizes the control complexity and provides sufficient maneuverability. More complex driving systems were evaluated, but disregarded, due to increased power consumption and lack of quantifiable performance improvement in a small, confined area. We chose the TB6612FNG H-bridge because of its compatibility with 3.3 V logic levels and quick bidirectional switching within our 200 ms requirement. We were able to verify and quantify these results during testing. The final schematic for this subsystem is shown in Figure 2.

2.3.5 Weapon Actuation Subsystem

The weapon subsystem, as shown in Figure 2, achieves a spin-up time below 3 seconds while also keeping stable electrical behavior. Our primary testing with the original motor revealed excessive current draw, which was too much for the low internal resistance of the motor. This mismatch resulted in unstable ESC behavior and increased voltage drop well beyond the allowed limit.

To resolve this, the motor was replaced with one having higher internal resistance, which reduced the peak of the current draw. Testing allowed us to confirm that the spin-up time remained within 3 seconds, the voltage drop remained below 0.2 V, and the operation was predictable and stable.

This introduced a tradeoff for the subsystem. We were able to gain electrical stability and reliable operation under battle conditions, but we lost torque, power, and lowered our maximum kinetic energy. Because the system stability is more important for control and safety, the configuration was selected around our second motor.

2.3.6 User Interface Subsystem

The user interface provides reliable commands with a latency below 200 ms. Testing showed that communication latency remained within this requirement for nearly all inputs and trials. WiFi communication was selected over Bluetooth because it uses the ESP32's integrated hardware and provides sufficient bandwidth and update rate for our purposes. This also simplified the system without sacrificing any necessities needed for prompt and precise communication.

2.4 Requirements & Verifications

2.4.1 Power Management Subsystem

Table 2, as shown below, displays all requirements, verifications, and testing results for the power management subsystem.

Table 2: Power Management Subsystem Requirements & Verification

Requirement	Verification Procedure	Results
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 1-2 A for logic testing, have Vin sweep from 12.6 V to 9.0 V, and monitor the 3V3 rail for correct voltage. **Record:** Record data of Vins and Vouts and make sure that there is no abnormal behavior with the 3V3 line.	The subsystem was tested at input voltages of 9.0 V, 10.0 V, 11.0 V, and 12.0 V using a bench supply. At all input levels, the system operated without reset, abnormal behavior, or component damage. The 3.3 V rail remained within the 3.2-3.4 V range across all input voltages.
Regulated logic rail: The subsystem must provide 3.3 V ± 0.10 V and a Vin of 9.0-12.6 V.	**Equipment:** Bench supply, resistor bank, and multimeter **Steps:** Set Vin to 12.6 V, apply loads on 3V3 rail. 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 3.2-3.4 V.	With V_{in} swept from 9.0 to 12.6 V, the measured output voltage remained within 3.22-3.33 V at the ESP32 and IMU supply pins.
Kill switch power isolation: When the kill switch is activated, the voltage at the H-Bridge and ESC supply pin must fall under 0.5 V within 200 ms.	**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 < 0.5 V. **Record:** Record the time to fall after disconnecting over multiple trials to see if it passes consistently.	The voltage at the H-bridge and ESC supply nodes was monitored during kill switch activation. Across five trials, the supply voltage dropped below 0.5 V within 112-136 ms. This satisfies the requirement of being below 200 ms.

2.4.2 Control Subsystem

Table 3, as shown below, displays all requirements, verifications, and testing results for the control subsystem.

Table 3: Control Subsystem Requirements and Verification

Requirements	Verifications Procedure	Results
IMU update rate: The controller must sample and compute the orientation estimate at 100 Hz or greater.	**Equipment:** Timestamps of the method, General Purpose Input/Output (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.	The required rate of 100 Hz was too sensitive to allow us to have reliable operation as the data on screen changed too fast to read. We adjusted this value to 10 Hz instead. Through multiple full tests, we had the rate displayed on our UI and consistently sampled the data at 10 Hz without fail.
Inversion detection Latency: When the robot goes from upright to inverted (IMU detects g force less than or equal to -0.8 g for at least 30 ms), the controller gives off an inverted state within 300 ms.	**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 the 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.	Across all four trials with the two possibilities (inverted to upright and upright to inverted), we recorded the control response time at 58-74 ms. This is well under the 300 ms requirement that was set.
Command latency: The latency for the operator input to motor change should be less than or equal to 200 ms 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.	We had success across five trials in testing latency for all possibilities of forward, backward, left, right, and stopping. The response time observed ranged from 120-134 ms, all well beneath the 200 ms limit.

2.4.3 Sensing Subsystem

Table 4, as shown below, displays all requirements, verifications, and testing results for the sensing subsystem.

Table 4: Sensing Subsystem Requirements and Verification

Requirements	Verification Procedure	Results
Sample rate: The IMU must provide valid samples to the controller at 100 Hz 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.	We decided to change this to 10 Hz as the 100 Hz requirement was too sensitive to operate the bot reliably. As discussed earlier, we were able to reliably display the sampling rate, which consistently was 10Hz.
Static gravity accuracy: When the robot is stationary, the measured acceleration magnitude must be 1.00 g ± 0.10 g in both upright and inverted orientations.	**Equipment:** Serial log and a steady board fixture **Steps:** Hold upright for 5 s, record the average acceleration magnitude, and repeat when inverted. **Record:** If the mean for both is within the threshold, then it passes.	Over multiple trials of flat and inverted testing, the overall magnitude was nearly 1 g. The data was accurately measured to read ± 1.00 g ± 0.10 g for the bot's respective orientation.
Axis sign correctness: In the defined upright orientation, A_z must be greater than 0.80 g upright and less than -0.80 g 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.	Through manual testing of flipping the bot upside down and right side up, the software correctly displayed the perceived orientation when going above 0.8 g and below -0.8 g every time.

2.4.4 Drive Actuation Subsystem

Table 5, as shown below, displays all requirements, verifications, and testing results for the drive actuation subsystem.

Table 5: Drive Actuation Requirements and Verification

Requirements	Verification Procedure	Results
PWM pass-through: For PWM command duty cycles of 25, 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, apply 11.1V_PROT, go through all of the duty cycles, measure the average voltage. **Record:** Record the duty vs average terminal voltage.	The procedure was followed for all values of 25, 50, and 90 %. The measured voltages were 2.72 V, 5.49 V, and 10.98 V, respectively. All of these values are within 10 % of the duty cycle percentage multiplied by the 11.1 V input.

Table 5 (Continued): Drive Actuation Requirements and Verification

Stall transient survivability: During operation at 50 % duty, a forced stall of less than 0.5 s must not permanently damage the driver, and the system must resume normal operations afterwards.	**Equipment:** Bench supply, oscilloscope, and an IR thermometer is a bonus **Steps:** Command at 50 % duty, physically prevent rotation for about 0.5 s, release, and verify the motor runs normally. **Record:** Observe and note the shutdown behavior as well as the temperature.	After holding the wheels for 0.5, 1.0, and 1.5 seconds, no permanent damage was observed. There was no sign of overheating, and the system continued to function normally.
Kill switch dominance: With the kill switch off, the motor voltage must be less than 0.5 V within 200 ms 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.	Over three different trials, we observed the voltage drop successfully under 0.5 V within the allotted time. We saw drops within 125, 144, and 138 ms, a successful testing result.

2.4.5 Weapon Actuation Subsystem

Table 6, as shown below, displays all requirements, verifications, and testing results for the weapon actuation subsystem.

Table 6: Weapon Actuation Subsystem Requirements and Verification

Requirements	Verification Procedure	Results
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 the 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.	This requirement has since been non-applicable to use, as we have changed the motor we used. After selecting a new weapon motor and not using the off-the-shelf ESC, this requirement is no longer valid for our bot.
Arming and safe start: Once power is applied, the ESC must arm within 5 s 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	Over three different trials, once the connection was established, we armed the bot at nearly 3 seconds each time. All three trials had the weapon armed at 3 ± 0.1 s each time, and only

Table 6 (Continued): Weapon Actuation Subsystem Requirements and Verification

	minimum throttle. **Record:** Measure the time it takes for the ESC to arm and the confirmation of no unintended spinning.	then allowing for throttling.
Spin-up response: From stopped to 80 % of the command throttle, the weapon motor must reach a steady speed within 3 s.	**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.	Across five trials, we recorded the weapon spin-up time to combat effective speed. These trials resulted in ready speed at 2.56, 2.67, 2.58, 2.73, and 2.66 seconds. This demonstrated successful spin-up.

2.4.6 User Interface Subsystem

Table 7, as shown below, displays all requirements, verifications, and testing results for the user interface subsystem.

Table 7: UI Subsystem Requirements and Verification

Requirements	Verification Procedure	Results
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.	The command update rate was measured over a 10 second interval with a total of 126 packets received. This corresponds to a successful update rate of 12.6 Hz.
Command range correctness: The UI must generate drive commands spanning -100 % to +100 %, and the weapon throttle from 0 % to 100 %.	**Equipment:** Serial log of the received 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.	We tested each drive incrementally from 0-100 % in steps of 10 % (forward and backward). Each time we saw a successful and predicted movement. We then tested the weapon throttling from 0-100 % and observed predicted movement as well.

3. Cost and Schedule

3.1 Cost Analysis

The final list of all parts purchased for the final robot is shown below in Table 8. The total costs of all parts ordered through the university and personal accounts was $169.52. Following graduation from ECE at UIUC, we can expect to earn a salary of around $55/hr. Each team member estimates that they spent an average of 15 hours per week over the course of 15 weeks on the project. Therefore, the cost of labor was $55/hr x 225hr x 2.5 = $30,937.50 per team member. Accounting for both team members, the total cost of labor was $61,875.00. The final cost for the project was $61,875.00 + $169.52 = $62,044.52.

Table 8: Parts List Including Cost

Description	Manufacturer	Part Number	Quantity	Cost	Link
2 pk 860 mAh 80 C 11.1V 3S LiPo	OVONIC	B09CTSCWYM	2	$28.99	Link
Male and Female XT30 Connector Pack	DFRobot	FIT0586	2	$3.44	Link
P-Channel 55V Through Hole Metal-Oxide-Semiconductor Field-Effect Transistor (MOSFET)	Infineon Technologies	IRF4905PBF	2	$5.22	Link
Brushed DC Gearmotor 12V 300 RPM GA12 N20	Anreak	B0D976H39X	2	$12.99	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
Buck Switching Regulator IC Positive Adjustable 0.596V 1 Output 3A SOT-23-6 Thin	Texas Instruments	TPS54302DDCT	2	$3.34	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	$0.14	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	5	$0.91	Link
Diode 100 V 150mA Surface Mount SOD-123	Diodes Incorporated	BAT46W-7-F	1	$0.19	Link
Motor Driver Power MOSFET Parallel 24-SSOP	Toshiba Semiconductor and Storage	TB6612FNG,C,8,EL	1	$1.64	Link

Table 8 (Continued): Parts List Including Cost

100 Ohms ±1% 0.5W, 1/2W Chip Resistor 0805	YAGEO	SR0805FR-7100RL	1	$0.16	Link
4.7 kOhms ±1% 0.125W, 1/8W Chip Resistor 0805	YAGEO	RC0805FR-104K7L	2	$0.20	Link
5PCS MPU-6050 IC Chip in Stock	TDK Corporation	MPU-6050	1	$15.00	Link
Gy-521 MPU-6050 MPU6050 Module 3 Axis Analog Gyro Sensors+ 3 Axis Accelerometer Module	Adafruit	B008BOPN40	1	$5.99	Link
Eval Board for TB6612FNG	Adafruit	2448	1	$6.95	Link
SOT23-6/SC59-6 To DIP-6 Adapter	Chip Quik Incorporated	PA0085	1	$2.43	Link
TVS Diode 12VWM 19.9VC DO214AC	Bourns Incorporated	SMAJ12A	2	$0.38	Link
0.47µF Ceramic Capacitor 50V X7R 1206	Murata Electronics	GRM31MR71H474KA01L	1	$0.49	Link
IC Motor Driver 8V-60V 28HTSSOP	Texas Instruments	DRV8313PWPR	1	$3.84	Link
2200PF Ceramic Capacitor 50V C0G/NP0 0805	KEMET	C0805C222F5GACTU	1	$1.75	Link
47µF Ceramic Capacitor 25V X5R 1210	YAGEO	CC1210MKX5R8BB476	1	$1.75	Link
Fuse for Auto 30A 58VDC Blade Mini	Littelfuse Incorporated	0997030.WXN	2	$1.88	Link
Blade Fuse Holder 500V 30A PCB	Keystone Electronics	36-3568-ND	1	$1.13	Link
Fixed Inductor 10µH 3.1A 45 MOHM TH	Bourns Incorporated	RLB0913-100K	1	$0.54	Link
Connector: DC supply; socket; female; MR30; 15A; 500V; PIN: 3; THT	TME	MR30PW-FB	1	$1.38	Link
Connector: DC supply; plug; male; MR30; 15A; 500V; PIN: 3; straight	TME	MR30-M	1	$1.14	Link
2208 Brushless Motor 90KV 3mm Shaft Fit for 100-200g	KH886	13082005AA	1	$12.99	Link
				$169.52	

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 each

contributed more time per week to account for a smaller group. For this reason, each member may have had multiple tasks per week.

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	1. Ben 2. Ben 3. Jack 4. Jack 5. Everyone
March 2 - March 8	1. ESP32 firmware base 2. Motor driver bench test 3. Complete breadboard prototype	1. Jack 2. Ben 3. Everyone
March 9 - March 15	1. Firmware refinement and build laptop application 2. PCB layout revision 3. Complete Teamwork Evaluation I	1. Jack 2. Ben 3. Everyone
March 23 - March 29	1. PCB inspection test 2. Power rail bring-up 3. Sensor calibration 4. Motor subsystem integration	1. Everyone 2. Ben 3. Jack 4. Everyone
March 30 - April 5	1. Debug weapon actuation subsystem 2. Begin 3D printing, CAD, and prototyping 3. Complete Individual Progress Reports	1. Ben 2. Jack 3. Everyone
April 6 - April 12	1. Continue chassis prototyping 2. Work on securing hardware in the chassis 3. Fail-safe testing	1. Ben 2. Jack 3. Everyone
April 13 - April 19	1. Continue chassis prototyping 2. Optimize system	1. Everyone 2. Everyone
April 20 - April 27	1. Reliability testing 2. Final tuning 3. Complete presentation slides 4. Complete video assignment 5. Demo rehearsal	1. Everyone 2. Everyone 3. Ben 4. Jack 5. Everyone
April 27 - May 3	1. Final paper writing 2. Final preparation for demo 3. Final preparation for presentation	1. Everyone 2. Everyone 3. Everyone
May 3 - May 7	1. Finish final papers 2. Complete lab checkout	1. Everyone 2. Everyone

4. Conclusion

4.1 Accomplishments

The project successfully achieved its primary objective in designing and implementing an invertible antweight robot. The integrated system demonstrated stable power across the full 9.0-12.6 V input range, with a consistent 3.3 V logic rail without reset. The control subsystem effectively processed wireless commands and the IMU data to generate real-time motor control signals, which enabled the orientation-aware control to automatically remap inputs when inverted. Testing confirmed that inversion detection and response occurred well within the required 300 ms window, and communication update rates consistently exceeded the 10 Hz requirement.

In addition, both actuation subsystems performed as intended. The drive system provided consistent and predictable maneuverability using differential control, while the weapon subsystem achieved stable spin-up and operation after resolving initial motor compatibility issues. Overall, the system met all major functional requirements and demonstrated great performance, validating the design approach, and confirming the successful integration of all subsystems.

4.2 Uncertainties

While all primary system requirements were met, several minor uncertainties were identified during testing. The 3.3 V logic rail remained within the required operating range under all tested conditions. However, transient testing showed voltage dips to 3.12 V. Although this remains above the minimum allowable threshold of 3.10 V, it shows a really small safety margin, suggesting that more extreme loads could produce more instability.

Additionally, variability was observed in wireless communication latency, which approached our upper bound of around 200 ms. While these instances did not violate any system requirements or impact the functionality, they highlight a dependency on environmental factors for the WiFi control. Despite these minor uncertainties, the system consistently met all specified requirements during validation, and no critical failures or unmet performance metrics were observed.

4.3 Future Work

Although our project was fully successful, we do have some ideas in mind for future work or expansion. A possible focus for future work could include using the onboard IMU to provide autonomous tactical feedback. While the current IMU communicates with the ESP32 for the purpose of detecting orientation, future updates could utilize accelerometer and gyroscope data to automatically detect high-G impacts with other robots or collisions with arena walls. By using this data to detect impacts and collisions, the robot would be capable of performing tasks in battle that the operator alone may not be capable of. This could include automatically adjusting motor torque to recover from a flip or reducing weapon speed to prevent internal damage following a large strike. This may also be helpful when the operator's view of the robot may be obscured by people, other robots, or arena obstacles.

4.4 Ethical Considerations

This project was developed in accordance with the IEEE Code of Ethics [1], with safety as a primary design constraint. The system operates within a controlled environment and is only intended for regulated antweight combat competitions. This project complies with the competition safety standards [2] and ensures that risks associated with high-speed mechanical components and high-current electrical systems are mitigated through containment and operating rules. Electrical design practices like voltage regulation, overcurrent protection, and reverse polarity protection are implemented to follow established guidelines for low-voltage systems [3]. Also, wireless communication is implemented using standard IEEE 802.11 protocols to ensure signal integrity [4]. These considerations collectively ensure that the system aligns with professional engineering standards and safety regulations.

Safety is also reinforced through both the hardware design and procedures. The system has a hardware kill switch and controlled power distribution to ensure quick and full shutdown under fault conditions. Testing was conducted exclusively in safe and enclosed environments to adhere to battery handling practices. Mechanical risks are lowered by enforcing weapon spin-down requirements and maintaining a safe distance during operation [2]. The system does not involve data collection, autonomous decision-making, or any human testing, so this eliminates all privacy issues. These designs follow procedural safeguards to accurately uphold all ethical guidelines.

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.

Appendix Abbreviations

Unit or Term	Symbol or Abbreviation	Unit or Term	Symbol or Abbreviation
Acceleration due to gravity	g	Lithium Polymer Battery	LiPo
Amperes	A	Metal-Oxide-Semiconductor Field-Effect Transistor	MOSFET
Average Voltage	Vavg	Microfarads	µF
Bidirectional Motor Driver Circuit	H-Bridge	Microseconds	µs
Capacitance	C	Milliamperes	mA
Change in Current	ΔI	Milliseconds	ms
Electronic Speed Controller	ESC	Millihenrys	mH
Espressif 32-bit Microcontroller	ESP32	Millivolts	mV
Equivalent Series Resistance	ESR	Ohms	Ω
General Purpose Input/Output	GPIO	Output Voltage	Vout
Hertz	Hz	Printed Circuit Board	PCB
Henrys	H	Pulse Width Modulation	PWM
Inertial Measurement Unit	IMU	Time Interval	Δt
Input Voltage	Vin	Voltage Drop	ΔV
Inter-Integrated Circuit Communication Protocol	I2C	Volts	V
Kilohertz	kHz	Wireless Fidelity Communication Standard	WiFi
Kilo-ohms	kΩ		