ECE 445

Antweight Battlebot Final Report

Team Members

Yuxin Zhang (yuxinz11)

Wenhao Zhang (wenhaoz5)

Xiangyi Kong (xkong13)

TA: Zhuoer Zhang

ECE 445 Senior Design Laboratory, Spring 2026

May 2026

Abstract

This report presents our design, implementation, and verification of a 2 lb antweight battlebot for ECE 445. We divided the robot into four main subsystems: power, control, drive, and weapon. The ESP32-S3 controller receives wireless commands from a host computer, controls the left and right drive motors, actuates the front lifter, and handles safety-related behavior such as startup protection and shutdown during communication loss. The most significant challenge we faced during the project was maintaining power stability during short high-current transients. In our first version, the weapon servo shared part of the supply path with the controller. When the servo moved, it caused about a 1.44 V drop on the ESP32 supply rail. This voltage drop reset the controller and caused the wireless connection to fail.

We solved this issue by separating the servo supply from the logic supply and verifying the revised design through direct measurements. In the final testing stage, the logic rails remained stable, the motor and servo commands worked correctly, the robot stayed still during startup, and the failsafe response worked when communication was lost. The robot also passed the endurance test and the lifting test in the final integrated setup. Overall, this project showed us that a small and low-cost combat robot can still operate reliably when the power distribution, safety behavior, and subsystem interactions are designed and tested carefully.

Contents

List of Figures

List of Tables

1 Introduction

Antweight combat robots have to fit within a strict weight limit while still surviving impact, sudden current demand, and safety-critical operating conditions. From our experience in this project, a robot can appear to work correctly during simple bench testing but still fail during full operation if the power rail drops, the communication link becomes unstable, or one subsystem affects another unexpectedly. Our goal was to build a 2 lb battlebot that could drive in a controlled manner, actuate a front lifter quickly, and remain safe during boot, communication loss, and large transient loads.

Our final robot is organized into four main subsystems: power, control, drive, and weapon. The power subsystem distributes energy from an 11.1 V nominal 3S lithium polymer battery and provides the regulated rails required by the rest of the robot. The control subsystem uses an ESP32-S3 to receive wireless commands and generate the low-level control signals. The drive subsystem uses a dual H-bridge to achieve differential steering, and the weapon subsystem uses a servo-driven front lifter. Figure 1 shows the overall system architecture and the major power and signal connections.

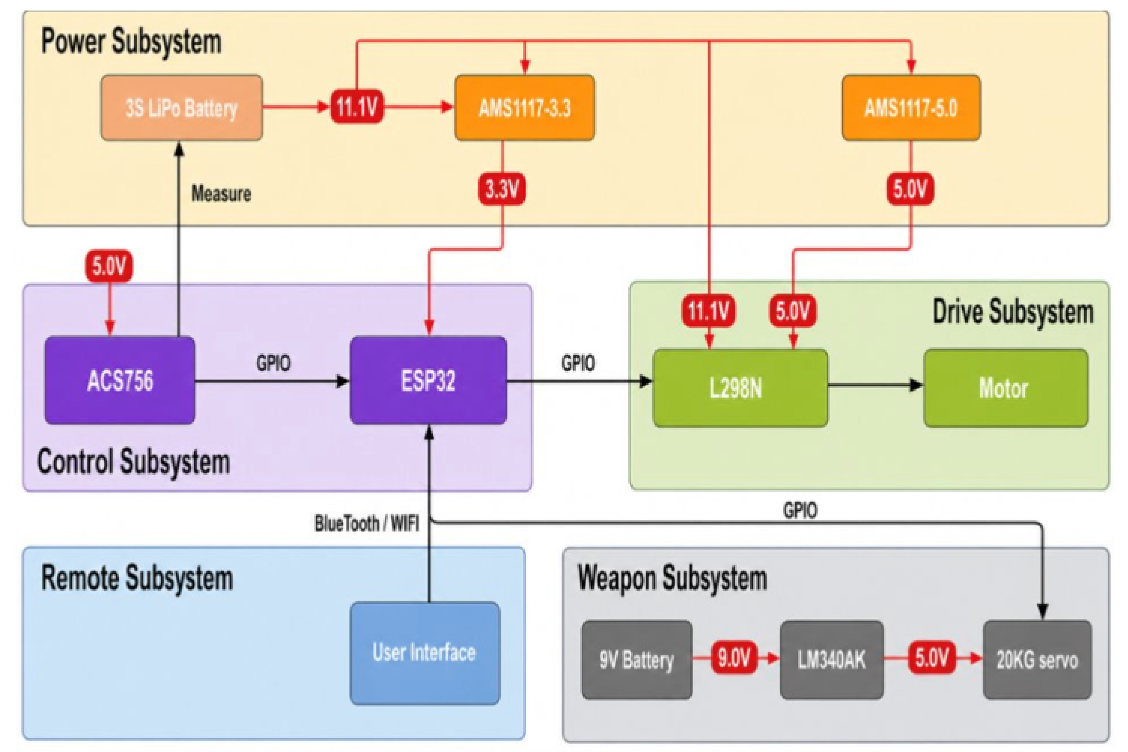

Figure 1: Top-level battlebot block diagram showing the power, control, drive, weapon, and remote subsystems.

We selected the main project functions based on the overall purpose of the robot:

1. Reliable differential drive for forward, reverse, and turning motion.

2. Fast lifter actuation for controlled engagement with an opponent.
3. Wireless operator control with bounded command latency.
4. Safe behavior during boot and communication loss.
5. Stable regulated power during electromechanical transients and safe operation under expected current loading.

The high-level requirements used to guide our final design are summarized in Table 1. A more detailed requirement and verification matrix is included in Appendix A.

Requirement category	Requirement
Mobility	Reach a final calculated top speed of approximately 1.3 m/s on a flat surface.
Control latency	Respond to drive and weapon commands within 100 ms.
Failsafe behavior	Enter a safe state within 250 ms of communication loss and remain motionless at boot until valid commands are received.
Weapon actuation	Move from neutral to maximum extension in 0.5 s or less and lift an approximately 2 lb equivalent load.
Endurance	Operate continuously for at least 3 min on a fully charged battery while driving and performing repeated weapon actuations.
Electrical robustness	Maintain regulated logic rails during motion and actuator activity; tolerate expected current loading without controller reset.

Table 1: High-level project requirements used for design and verification.

2 Design

2.1 Design Procedure

We treated the robot as one integrated system rather than a group of isolated parts. At the beginning, we converted the general project goal into specific subsystem requirements, including stable voltage rails, differential-drive control, actuator torque, command latency, and battery endurance. After building the first prototype, we compared our expected behavior with actual measurements and used those results to revise the design. This process was important because the main failure we encountered was not a steady-state problem. It only appeared during short transient events.

The largest design change happened after integration testing. On paper, the battery had more than enough current capability for the robot. However, the shared supply path allowed servo loading to pull the controller voltage low enough to reset the ESP32. We addressed this problem by isolating the servo supply from the main controller rail. Section 2.3 describes this issue and the final solution in more detail.

2.2 Design Alternatives and Final Architecture

We made several design choices to balance reliability, part availability, and integration effort:

- **Controller:** We selected the ESP32-S3 because it provides wireless communication, PWM generation, and embedded programming support on a single platform.
- **Drive:** We selected the L298N dual H-bridge because it was simple to use and easy to obtain, although a MOSFET-based driver would be more efficient.
- **Weapon:** We chose a front lifter instead of a spinner because it was easier to package, safer to test, and more realistic for the torque available from our actuator.
- **Power** We replaced the original shared regulated path with an isolated weapon supply path after measured transient voltage droop caused controller resets.

Figure 2 summarizes how commands pass through the final hardware and software system.

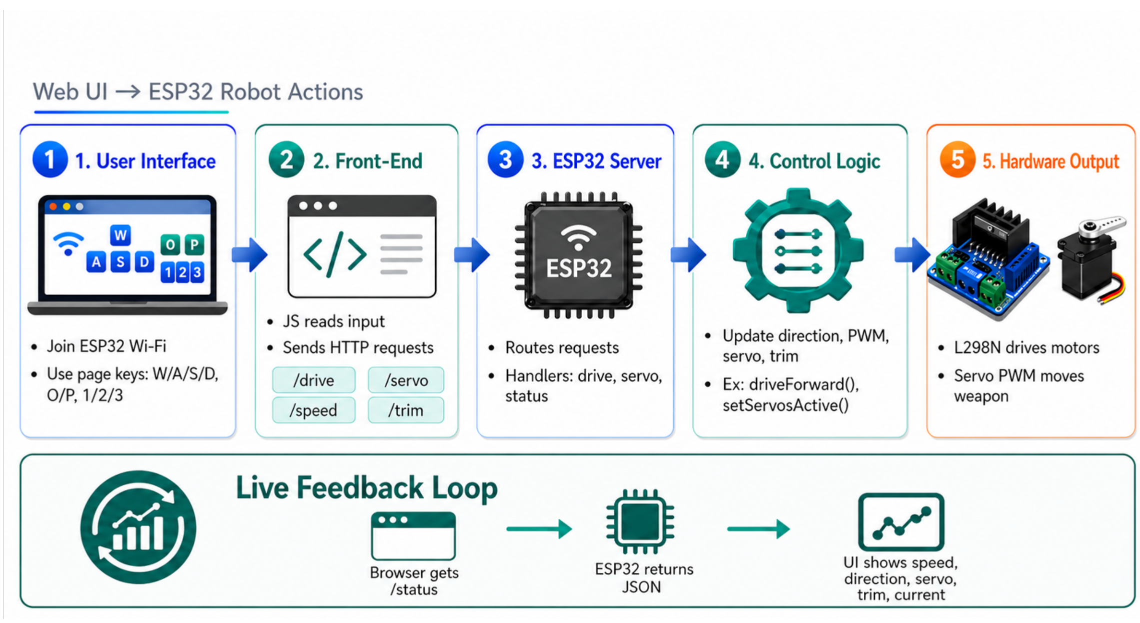

Figure 2: Control and feedback flow from user commands to ESP32 processing, motor outputs, servo outputs, and current feedback.

2.3 Power Subsystem

The power subsystem starts with an 11.1 V nominal 3S lithium polymer battery. The battery directly supplies the motor rail and also feeds regulator stages that generate the logic rails for the ESP32, motor-driver logic, and servo path. For this subsystem, our main design concern was not only the average current capability, but also the ability to avoid short transient voltage drops.

We estimated the worst-case current budget as

$$I_{\text{peak}} = 2I_{\text{motor,peak}} + I_{\text{servo,peak}} + I_{\text{ESP32}}, \tag{1}$$

where the two drive motors are the main load during startup and pushing, while the servo creates short weapon-related current transients. Using our final design assumptions,

$$I_{\text{peak}} \approx 2(1.4\ \text{A}) + 1.8\ \text{A} + 0.5\ \text{A} = 5.1\ \text{A}. \tag{2}$$

The selected battery has a nominal capacity of 2.2 Ah and a 50C discharge rating, so its theoretical continuous current capability is

$$I_{\text{battery,max}} = C_{\text{rating}}Q = 50 \times 2.2\ \text{A} = 110\ \text{A}. \tag{3}$$

The resulting current margin is

$$\text{Current Margin} = \frac{I_{\text{battery,max}}}{I_{\text{peak}}} = \frac{110}{5.1} \approx 21.6. \tag{4}$$

Equations 3 and 4 show that the battery itself was not the main limitation in our design. The actual issue was voltage droop on the controller rail. In early testing, servo loading caused an approximately 1.44 V drop on the ESP32 supply, which was large enough to reset the controller even though the battery could still supply the required current. In the final design, we isolated the weapon power path so that servo transients would not disturb the controller rail.

The final PCB layout used for the integrated electronics is shown in Figure 3. This PCB helped simplify the wiring and integration process. However, in a future revision, we would replace the linear-regulator-centered power path with switching converters and add more bulk decoupling capacitor to improve efficiency and transient performance.

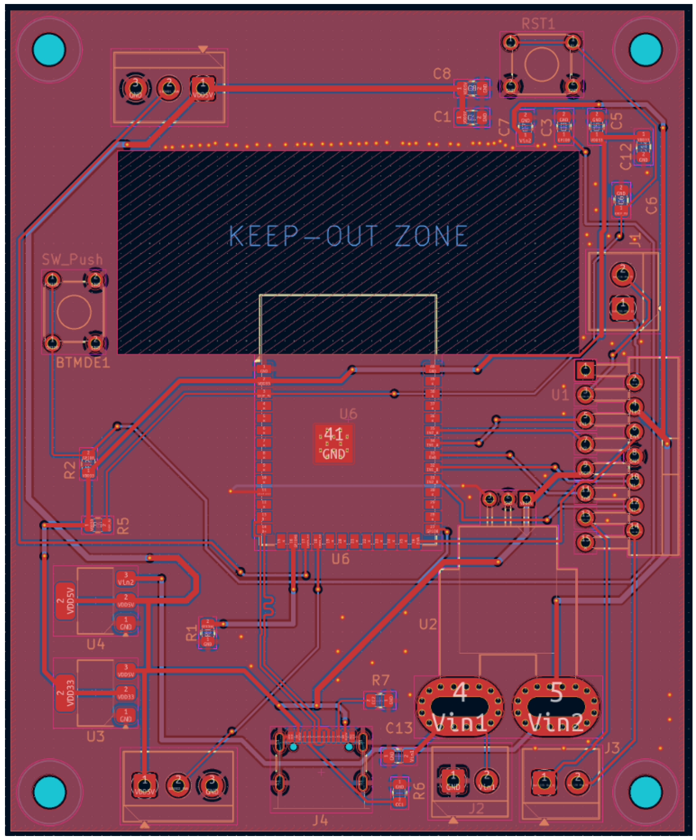

Figure 3: Final PCB layout used to integrate the power, control, and signal-routing hardware.

2.4 Control Subsystem

The control subsystem is based on the ESP32-S3. It receives wireless commands from a host computer, converts those commands into left and right drive outputs, generates servo PWM for the front lifter, and handles the safety behavior. We intentionally kept the firmware straightforward so that we could debug it more easily and understand its failure modes during testing.

We treated three control features as required parts of the system:

1. **Safe boot:** The robot must remain motionless until valid commands are received.
2. **Failsafe timeout:** The robot must disable motion quickly when the wireless link is lost.
3. **User visibility:** The interface should show current-related information and provide trim controls for operation and tuning.

The operator interface used during final testing is shown in Figure 4. The trim control was especially helpful because it allowed us to compensate for drivetrain asymmetry without repeatedly changing the hardware.

Figure 4: Operator interface used for wireless command input, drive tuning, and monitoring.

2.5 Drive Subsystem

The drive subsystem uses an L298N dual H-bridge to convert low-power logic signals into bidirectional motor drive outputs. One channel controls the left motor, and the other channel controls the right motor. This setup allows the robot to move forward, reverse, turn in place, and follow curved paths.

We prioritized controllability over maximum speed because a lifter-based battlebot depends more on steering precision and traction than raw top speed. Based on the final drivetrain calculation using motor speed, wheel geometry, and the integrated mechanical configuration, the robot's top speed was approximately

$$v_{\text{final}} = \frac{s}{t} = \frac{1.0 \text{ m}}{0.8 \text{ s}} = 1.25 \text{ m/s} \approx 1.3 \text{ m/s}. \tag{5}$$

This value was consistent with the final observed straight-line behavior of the robot during testing. For our final report, we therefore used 1.3 m/s as the representative mobility result for the completed drivetrain.

2.6 Weapon Subsystem

The weapon subsystem uses a servo-actuated front lifter. The purpose of the lifter is to get under an opponent and apply upward force in a controlled way instead of relying on a high-energy spinning weapon. We estimated the required output torque using a 2 lb equivalent load:

$$F = mg = (0.907\text{ kg})(9.81\text{ m/s}^2) \approx 8.90\text{ N}, \tag{6}$$

$$T_{\text{ideal}} = Fr = (8.90\text{ N})(0.05982\text{ m}) \approx 0.532\text{ N}\cdot\text{m}. \tag{7}$$

Applying a safety factor of 4 to account for friction, shock loading, misalignment, and battery sag gives

$$T_{\text{required}} = 4T_{\text{ideal}} \approx 2.13\text{ N}\cdot\text{m}. \tag{8}$$

This torque estimate guided our servo choice and confirmed that a 20 kg class actuator was a reasonable option for the final design. In final lifting tests, the mechanism successfully raised an approximately 2 lb equivalent load. Figure 5 shows the loading test we used to confirm the lifting performance.

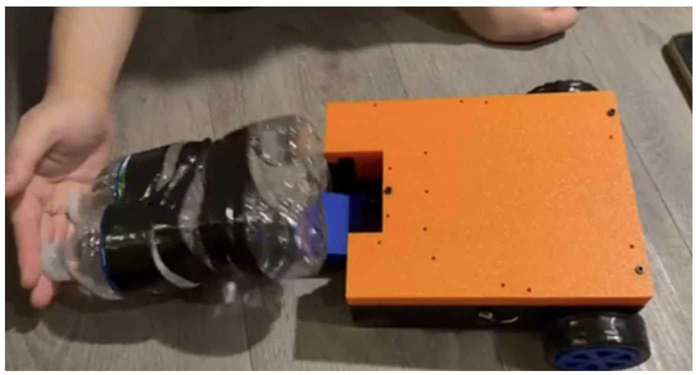

Figure 5: Weapon verification setup used to demonstrate lifting of an approximately 2 lb equivalent load.

2.7 Integrated Mechanical Prototype

Figure 6 shows the assembled robot. We kept the mechanical structure relatively simple so that the integration and debugging process would be manageable. This was a reasonable tradeoff for a senior design project, but it also meant that wiring, grounding, and full-system testing had to be done carefully.

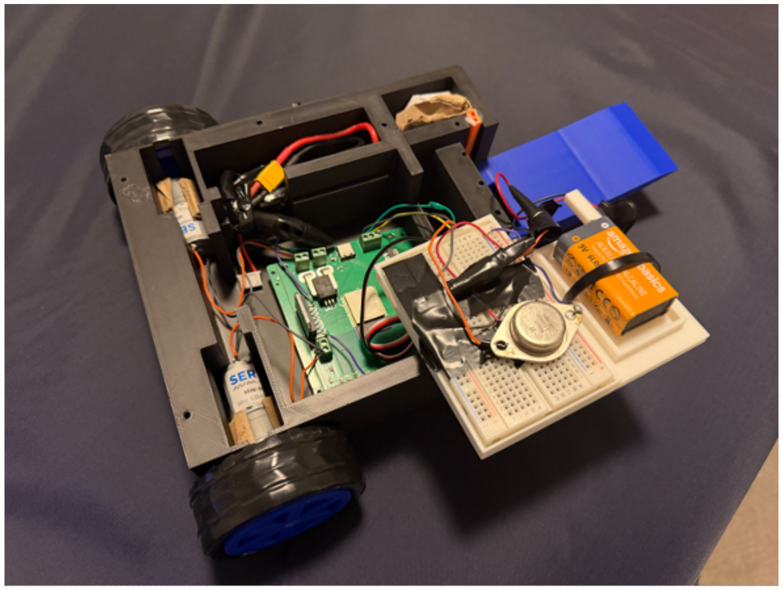

Figure 6: Assembled antweight battlebot prototype used for subsystem and integrated-system testing.

3 Verification

Our verification work focused on proving that the final robot met the project requirements using measurable and repeatable evidence. We used direct voltage and current measurements, timed tests, and fault-condition tests instead of relying only on visual observation. The full requirement and verification matrix is included in Appendix A, and the main results are summarized in this section.

3.1 Power Verification

The 3.3 V controller rail measured 3.315 V at idle and 3.310 V during motion while supplying the ESP32 operating current of approximately 0.5 A. The 5 V logic rail measured 5.080 V at idle and 5.082 V during motion. These measurements showed that the final regulator arrangement kept the logic supplies stable during normal operation. In comparison, the earlier shared-rail weapon design produced an approximately 1.44 V drop on the controller supply during servo activity, which caused ESP32 resets. The revised power architecture directly solved that problem.

3.2 Control Verification

We verified the control subsystem under both normal operating conditions and fault conditions. Safe boot behavior was tested by power cycling the robot and confirming that no unintended motion occurred before valid commands were received. Communication-loss fail-safe behavior was tested by disconnecting the wireless link while the robot was active. In this test, the robot stopped within the required 250 ms window.

We measured command-response latency by comparing a keyboard input event with the resulting motor or servo motion using high-frame-rate video. The measured average delay was approximately 78 ms, which satisfied the requirement that the robot respond within 100 ms.

3.3 Drive Verification

We verified the drive stage by commanding forward, reverse, and turning motion through the two H-bridge channels. During high-speed operation, the system current stayed around 2.6 A to 2.8 A, showing that the driver and power path could support the expected drive demand. The final drivetrain speed calculation gave a top speed of approximately 1.3 m/s, which matched the observed fast straight-line behavior of the robot during final testing.

3.4 Weapon Verification

The isolated weapon supply measured 5.042 V at idle and 5.040 V during motion. This result showed that weapon operation no longer destabilized the control electronics. The servo responded correctly to PWM commands and moved from neutral to maximum extension in approximately 0.31 s, satisfying the 0.5 s actuation requirement. The return-to-default time was approximately 1.08 s. The lifter also raised an approximately 2 lb equivalent load and exceeded the required 10 mm lift height.

Oscilloscope confirmation of stable PWM generation is shown in Figure 7. This measurement supported our conclusion that the control subsystem was generating valid actuation waveforms.

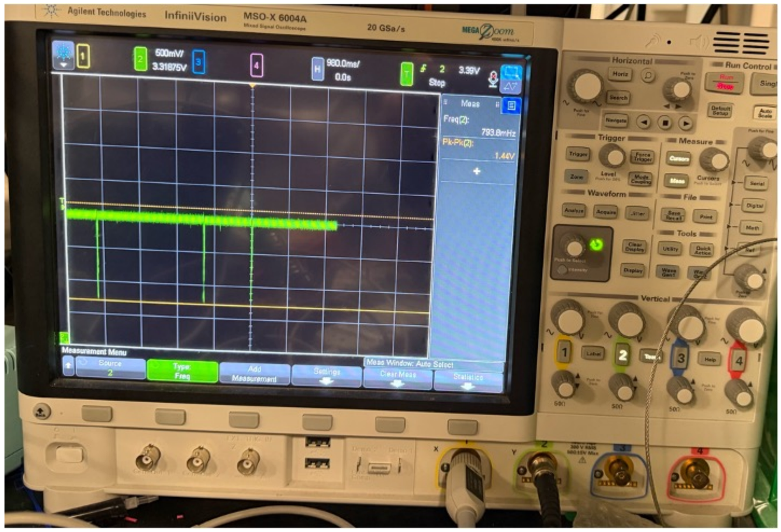

Figure 7: Oscilloscope capture used to confirm valid PWM behavior during subsystem verification.

3.5 System-Level Verification Summary

Table 2 summarizes the main quantitative verification results used in the body of this report.

Metric	Measured value	Status
3.3 V controller rail	3.315 V idle, 3.310 V in motion	Passed
5 V logic rail	5.080 V idle, 5.082 V in motion	Passed
Communication-loss stop time	< 250 ms	Passed
Command-response latency	78 ms average	Passed
Final calculated top speed	1.3 m/s	Passed
System drive current	2.6 A to 2.8 A	Passed
Weapon extension time	0.31 s	Passed
Weapon return time	1.08 s	Informational
Weapon rail	5.042 V idle, 5.040 V in motion	Passed
Endurance test	8.6 min and 26 actuations	Passed

Table 2: Summary of key quantitative verification results.

4 Cost and Schedule

4.1 Parts Cost

Table 3 summarizes the major hardware costs for the final prototype.

Component	Manufacturer	Qty.	Unit cost ($)	Extended cost ($)
ESP32-S3-WROOM-1-N16R8	Espressif Systems	1	7.00	7.00
ACS756 current sensor	Allegro Microsystems	1	9.00	9.00
AMS1117-3.3 regulator	UMW	1	0.50	0.50
AMS1117-5.0 regulator	EVVO	1	0.20	0.20
11.1 V 3S LiPo battery	OVONIC	1	22.00	22.00
L298N motor driver	STMicroelectronics	1	13.00	13.00
508 RPM gear motor	ServoCity	2	15.00	30.00
Servo actuator	DFRobot or final equivalent	2	25.00	50.00
			Estimated materials subtotal	131.70
			Approximate spare/debugging allowance	19.40
			Estimated materials total	151.10

Table 3: Estimated parts cost for the final prototype.

4.2 Labor Cost

ECE 445 estimates labor cost using

$$\text{Labor Cost} = (\text{hourly rate})(\text{hours worked})(2.5). \tag{9}$$

Using an engineering labor rate of \$50/h and 33 h per team member,

$$\text{Individual Labor Cost} = (\$50/\text{h})(33\text{ h})(2.5) = \$4125. \tag{10}$$

Table 4 summarizes the labor estimate for each team member.

Team member	Hours	Rate ($/h)	Multiplier	Labor cost ($)
Yuxin Zhang	33	50	2.5	4125
Wenhao Zhang	33	50	2.5	4125
Xiangyi Kong	33	50	2.5	4125
Total	99	–	–	12375

Table 4: Labor cost estimate for the project team.

The total prototype cost, including materials and labor, was approximately $12,526.10.

4.3 Schedule

Table 5 summarizes the work completed during the semester and shows each team member's main area of responsibility. Yuxin Zhang mainly worked on software and control integration, Wenhao Zhang mainly worked on 3D printing and mechanical assembly, and Xiangyi Kong mainly worked on PCB design and electrical hardware integration.

Week	Major milestone	Yuxin Zhang	Wenhao Zhang	Xiangyi Kong
1	Project planning and requirements	Defined control goals, communication needs, and software requirements	Planned chassis structure and printable mechanical parts	Planned PCB functions, power paths, and required interfaces
2	Initial subsystem design	Set up ESP32 development environment and control logic outline	Modeled and prepared the first 3D-printable body components	Drew initial schematic and selected major PCB components
3	Component integration planning	Worked on wireless command flow and PWM output structure	Printed early chassis parts and adjusted fit between frame elements	Continued PCB layout and power-routing design
4	First prototype bring-up	Implemented drive and weapon firmware functions	Assembled printed parts into the first mechanical prototype	Fabricated and soldered the first PCB revision
5	Subsystem debugging	Tuned drive response, servo behavior, and command handling	Refined printed parts and helped mechanical reassembly	Debugged board connections and verified regulator and signal paths
6	Integration issue analysis	Investigated controller reset behavior during servo actuation	Adjusted mounting and packaging to support rework	Measured voltage droop and traced the problem to the shared supply path
7	Redesign and improvement	Updated firmware after the power redesign and retested control stability	Reprinted or adjusted parts to match the revised integrated layout	Revised the power architecture and isolated the servo supply path
8	Final verification and report preparation	Measured latency, failsafe behavior, and overall software performance	Supported final assembly, driving tests, and lift demonstrations	Verified PCB performance, rail stability, and final electrical results

Table 5: Reconstructed semester schedule by week and team member.

5 Conclusion

In this project, we built a functional antweight battlebot with wireless control, differential drive, a servo-actuated front lifter, and safe operating behavior within a compact 2 lb platform. Final testing showed stable regulated rails, correct command and actuation behavior, practical mobility, successful lifting of an approximately 2 lb equivalent load, and strong endurance performance.

The most important issue we found was not the battery current capability, but the transient voltage behavior in the power distribution network. Although the battery had a large theoretical current margin, the original shared supply path allowed servo loading to create an approximately 1.44 V drop on the ESP32 rail. We identified this issue through measurement, traced it back to the supply architecture, and fixed it by isolating the weapon supply. In the end, the project worked because we revised the design based on the real behavior of the hardware instead of depending only on our original calculations.

For future improvements, we would use switching regulators, add more local decoupling, choose a more efficient MOSFET-based motor driver, and collect more complete quantitative logs during combined driving and weapon operation. These improvements would help with efficiency, thermal behavior, and final documentation quality. A future version could also include more formal pre-layout power simulation, but the biggest lesson from this project is that transient bench measurements at subsystem boundaries are very important in compact battery-powered robots.

5.1 Ethical Considerations

We developed this project with the IEEE Code of Ethics in mind [6]. Safety was the main ethical concern because the robot includes moving mechanisms, a high-current battery, and wireless control. Because of this, we treated safe boot behavior and communication-loss stop behavior as required system features rather than optional improvements. Honest reporting was also important. We included the controller-reset issue caused by servo-induced voltage droop because leaving out a known reliability problem would give an incomplete picture of the project. Responsible charging, storage, and handling of the lithium polymer battery were also part of our team's safety responsibility during testing.

5.2 Broader Impacts

Although this project was focused on educational combat robotics, the engineering lessons apply to many other embedded robotic systems. Small mobile platforms used for inspection, warehousing, or interactive devices also combine sensing, communication, and actuation while operating under transient power demand. This project showed us that low-cost hardware can still support solid system-level engineering when design decisions are supported by measurement and verification. From an environmental point of view, the robot uses a rechargeable lithium battery, which reduces disposable battery waste, but it still requires responsible storage and end-of-life handling.

References

[1] ECE 445 Staff and ECE Editorial Services, "Preparing Your Final Report for ECE 445, Senior Design," University of Illinois Urbana-Champaign, Sept. 2019.

[2] Espressif Systems, "ESP32 Series Datasheet," 2025. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

[3] STMicroelectronics, "L298 Dual Full-Bridge Driver," [Online]. Available: https://www.digikey.com/en/products/detail/stmicroelectronics/L298N/585918

[4] Allegro MicroSystems, "ACS756 Current Sensor Datasheet," [Online]. Available: https://www.allegromicro.com/~/media/Files/Datasheets/ACS756-Datasheet.ashx

[5] DFRobot, "ST3215 Serial Bus Servo, 12 V, 30 kg," [Online]. Available: https://www.dfrobot.com/product-2962.html

[6] IEEE, "IEEE Code of Ethics," 2020. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html

Appendix

A Requirement and Verification Matrix

Requirement	Verification procedure	Quantitative result
Stable 3.3 V controller supply	Measure controller rail at idle and during motion with a multimeter or oscilloscope.	Passed: 3.315 V idle, 3.310 V in motion.
Stable 5 V logic supply	Measure 5 V rail at idle and during motion.	Passed: 5.080 V idle, 5.082 V in motion.
Common ground across major subsystems	Perform continuity check among controller, driver, sensor, and actuator grounds.	Passed.
Independent left and right drive control	Command left and right channels separately and observe motor behavior during bench and floor tests.	Passed.
Valid servo PWM generation	Observe PWM waveform on oscilloscope while commanding the weapon and confirm repeatable servo actuation.	Passed; representative waveform shown in Figure 7.

Requirement	Verification procedure	Quantitative result
Safe boot behavior	Power cycle the system repeatedly and verify no motion before valid commands are received.	Passed.
Failsafe within 250 ms of communication loss	Disconnect the wireless link while the robot is active and measure time to stop.	Passed: stop within required 250 ms window.
Command response within 100 ms	Record command event and physical response using high-frame-rate video, then estimate delay frame by frame.	Passed: 78 ms average delay.
Reach approximately 1.3 m/s top speed	Calculate final drivetrain speed from motor and wheel parameters and confirm consistency with straight-line testing.	Passed: approximately 1.3 m/s.
Continuous drive-current support	Measure current during high-speed operation and repeated starts.	Passed: approximately 2.6 A to 2.8 A system current.
Weapon rail stable under load	Measure isolated weapon supply at idle and during motion.	Passed: 5.042 V idle, 5.040 V in motion.
Weapon extension in 0.5 s or less	Time weapon actuation from neutral to full extension using video review.	Passed: 0.31 s.
Lift approximately 2 lb equivalent load	Apply a water-bottle load equivalent to about 2 lb and observe lift.	Passed.
Minimum lift height greater than 10 mm	Measure vertical displacement of the lifted object during the actuation test.	Passed: exceeded 10 mm.
Endurance greater than 3 min with repeated actuation	Run combined driving and repeated weapon operation on a full battery until the test endpoint is reached.	Passed: 8.6 min, 26 actuations.

Table 7: Detailed requirement and verification matrix.