ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

Combative Hardened Ultra Tumbler (C.H.U.T.)

Antweight Battlebot

Team #47

ABHINAV GARG (ag90@illinois.edu)
RAHUL KRISHNAMOORTHY
(rahulr9@illinois.edu)
SHOBHIT SINHA (ss194@illinois.edu)

TA: Ye Xiaodong

May 5, 2026

Abstract

Combative Hardened Ultra Tumbler (C.H.U.T.) is a 2 lb antweight battlebot platform that combines a custom printed circuit board (PCB), an ESP32-C3 wireless controller, independent brushed drivetrain channels, a front-mounted brushless weapon driven through an electronic speed controller (ESC), and a printed chassis. The final system demonstrated stable wireless command reception at 50 Hz over a 15 ft line-of-sight link, 75 ms end-to-end command latency, motor-output shutdown within 250 ms of communication loss, and a regulated 3.3 V logic rail that remained within 5% tolerance during simultaneous drivetrain and weapon loading. The drivetrain and control PCB reached functional integrated operation, while the weapon subsystem was completed by replacing the original custom brushless-driver path with an external ESC after configuration and tuning failures in the integrated MCF8316A path.

Contents

1 Introduction

The antweight battlebot competition is a constrained systems-design problem. The robot must remain below the 2.00 lb weight limit, use approved printed thermoplastics, operate wirelessly, and survive impacts while carrying a lithium-polymer (LiPo) battery and high-current motors. C.H.U.T. addresses that problem with a compact printed chassis, an ESP32-C3-based control board, two independently driven brushed drivetrain motors, and a front vertical spinner weapon.

The main engineering challenge was not any single subsystem in isolation. The project required the electrical, mechanical, firmware, and safety decisions to work together inside a small package. Motor current transients can cause voltage droop or electromagnetic interference, mechanical shock can damage connectors and solder joints, and wireless control failure can make the robot unsafe if motor outputs are not disabled quickly. The final design therefore emphasized a stable logic rail, clear separation between control and high-current power paths, accessible debugging points, and firmware failsafe behavior.

Figure 1 shows the high-level system architecture. A laptop-based operator interface sends wireless commands to the onboard ESP32-C3. The custom PCB distributes battery power, regulates the logic rail, drives the brushed wheel motors through H-bridges, and provides the command signal used by the final external ESC weapon path. The final implementation differs from the first design document in one important way: the original MCF8316A integrated brushless motor controller remained on the PCB, but the final working weapon subsystem used an external ESC because the custom brushless-driver configuration could not be stabilized within the remaining schedule.

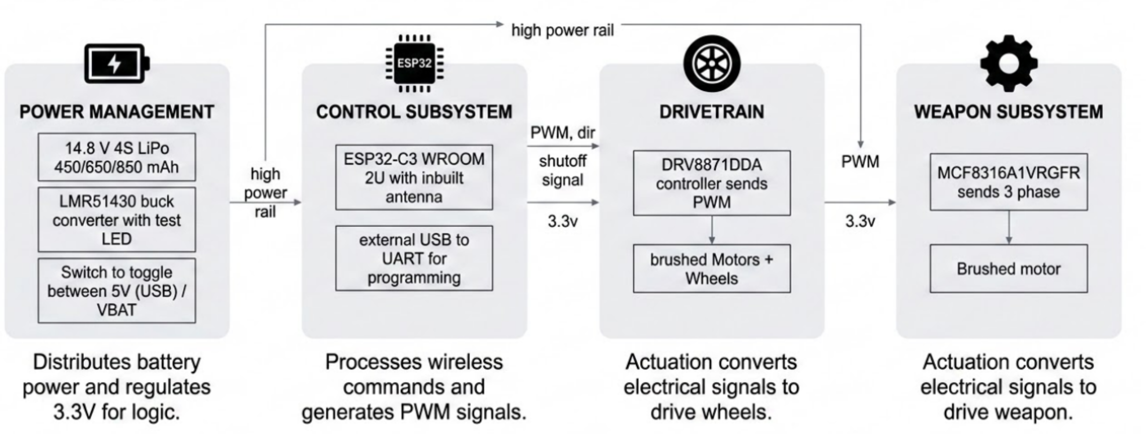

Figure 1: High-level block diagram of the C.H.U.T. remote system, onboard control electronics, drivetrain, weapon, and power interfaces.

1.1 High-Level Requirements

Table 1 lists the high-level requirements used to judge the final system. These requirements emphasize the behaviors most important to a combat robot: weight compliance, responsive wireless control, failsafe shutdown, drivetrain mobility, weapon operation, and logic-rail stability under combined electrical loading.

Table 1: High-level requirements for C.H.U.T.

ID	Requirement
HLR-1	The fully assembled robot shall have a total mass less than or equal to 2.00 lb and comply with applicable material and electrical competition rules.
HLR-2	The robot shall respond to a valid wireless command within 100 ms and disable all motor outputs within 250 ms of communication loss.
HLR-3	The drivetrain shall provide stable remote-controlled motion with a target speed of approximately 2 m/s.
HLR-4	The weapon subsystem shall achieve at least 2000 rpm and operate without causing logic brownout or motor-control shutdown during the demonstrated load condition.
HLR-5	The regulated 3.3 V logic rail shall remain within 3.0–3.6 V during simultaneous drivetrain and weapon loading.

1.2 Subsystem Overview

The system was partitioned into four main subsystems. The power subsystem accepts the nominal 14.8 V LiPo battery input, includes transient suppression and bulk decoupling, and produces the 3.3 V logic rail. The control subsystem uses the ESP32-C3-WROOM-02 module for Bluetooth Low Energy (BLE) communication, command parsing, pulse-width modulation (PWM) generation, and failsafe logic [1]. The drivetrain subsystem uses two DRV8871 H-bridge motor drivers for independent left and right brushed motor control [2]. The mechanical and weapon subsystem packages the PCB, battery, drive motors, weapon motor, skids, and wheel guards into a printed chassis, while the final weapon actuation uses a commercial ESC after the integrated brushless-driver approach was abandoned.

The final robot also has broader safety and societal implications because it intentionally stores and transfers kinetic energy. Although this project is a competition robot rather than a consumer product, it still requires disciplined operation: a restrained test setup, controlled activation procedure, attended LiPo charging, and honest reporting of unverified limits. These concerns are addressed explicitly in the conclusion.

2 Design

2.1 Design Procedure and Alternatives

The final design was reached through staged integration rather than a single linear implementation. The proposal treated the battlebot as a 2 lb printed robot with a two-wheel drivetrain, a front drum or vertical-spinner weapon, a 4S LiPo battery, and a custom control PCB. During the semester, three design changes became important.

First, the mechanical concept shifted away from an early self-righting goal. The front vertical spinner and compact chassis made self-righting geometry difficult to package without increasing mass, height, or print complexity. The final chassis instead prioritizes low height, rear-biased drive wheels, replaceable front skids, and wheel protection.

Second, the weapon control architecture changed. The original PCB included an MCF8316A sensorless brushless direct current (BLDC) driver [3]. That path was attractive because it would keep the weapon commutation on the custom board. In testing, however, the driver was difficult to configure repeatably over I2C, and the motor did not enter stable operation. The final design therefore retained the custom PCB for power, control, and drivetrain functions but moved weapon commutation to a dedicated external ESC. This reduced integration risk by converting the weapon command interface to standard servo-style PWM.

Third, the control interface evolved from early Wi-Fi and keyboard tests to a BLE Xbox-controller workflow. The final host script reads the controller with Python, maps joystick and trigger values into signed motor commands, and sends periodic BLE updates to the ESP32-C3. This split allowed rapid tuning of dead zones, motor inversion, turn scaling, and weapon command limiting without reflashing the embedded firmware after every operator-interface change.

2.2 Electrical Design

Figure 2 shows the routed custom PCB. The board is organized around four functional regions: the ESP32-C3 controller, the LMR51430 buck regulator and power-entry circuitry, two DRV8871 brushed drivetrain channels, and the original MCF8316A brushless weapon-driver region. The layout uses large copper areas near the motor drivers for current return and heat spreading. It also physically separates the nominal 14.8 V motor-power region from the 3.3 V logic region to reduce coupling from motor transients into the microcontroller supply.

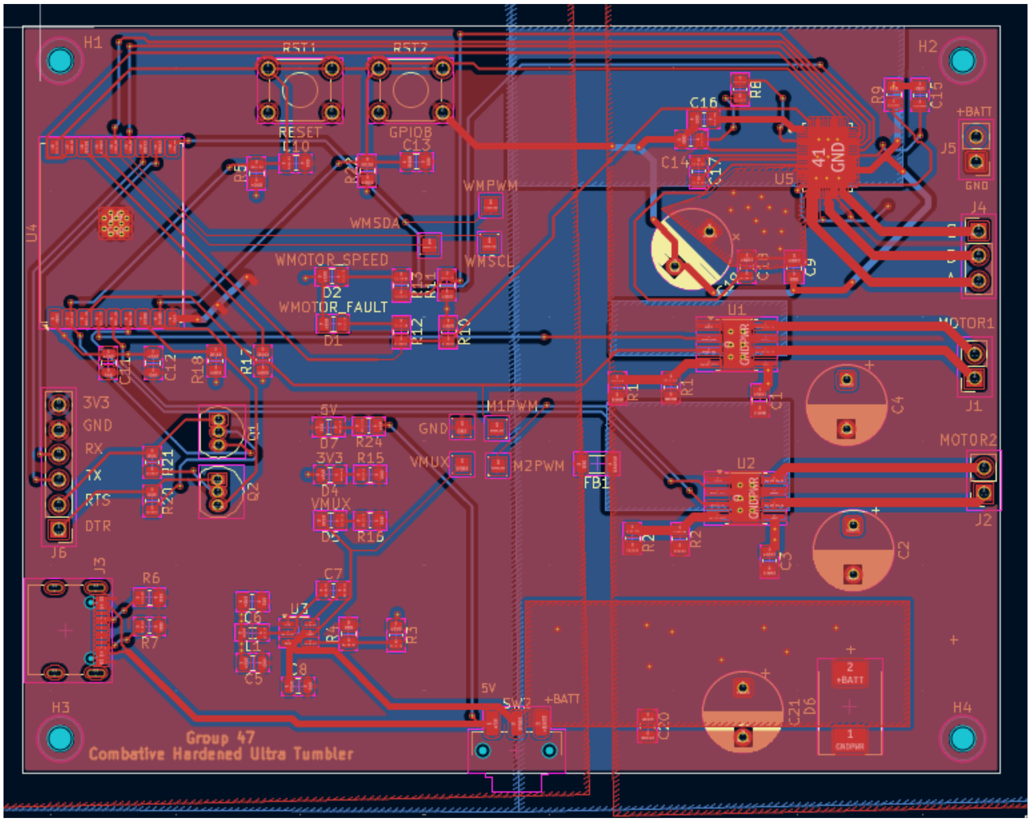

Figure 2: Custom PCB routing with ESP32-C3 control, power regulation, drivetrain H-bridges, and original integrated weapon-driver region.

2.2.1 Power Subsystem

The power subsystem supplies the high-current motor path directly from the LiPo battery and generates a regulated logic rail for the microcontroller. The design uses an LMR51430 synchronous buck regulator to step the battery or bring-up input down to 3.3 V [4]. A source-select switch allows the board to be powered from USB during bring-up or from the battery during robot operation. Indicator light-emitting diodes (LEDs) on the input and 3.3 V rails provide immediate debugging feedback.

The battery input includes bulk capacitance, ceramic decoupling, and a transient-voltage-suppression (TVS) diode. These parts address different failure modes. Bulk capacitance reduces low-frequency rail sag during motor current steps, ceramic capacitors reduce higher-frequency switching noise, and the TVS diode clamps larger transient voltage spikes from motor back-electromotive-force events. The board also exposes test points so the 3.3 V rail and input rail can be measured during verification.

2.2.2 Control and Communication Subsystem

The ESP32-C3-WROOM-02 module was selected because it integrates the 2.4 GHz radio, antenna, microcontroller, and peripheral support in a compact module [1]. The final firmware exposes a BLE service named `oakremote-bot`. Commands use a simple text format such as `m1=-180,m2=175,wm=70`, where `m1` and `m2` are signed drivetrain commands and `wm` is the weapon throttle command.

The drivetrain PWM channels run at 20 kHz with 8-bit resolution. The firmware uses sign-magnitude control: one driver input receives PWM while the opposite input is held low for forward motion, and the roles reverse for reverse motion. The final firmware also explicitly drives the motor-control pins low at boot before arming the ESC. This change was added because floating ESP32 pins during the ESC arming delay could otherwise cause wheel twitching.

The control software includes two failsafe mechanisms. On BLE disconnect, the ESP32 immediately stops all motors and restarts advertising. While connected, a command watchdog stops all motors if no command is received within 500 ms. The Python controller sends a heartbeat every 200 ms, which provides a 2.5x timing margin relative to the firmware timeout.

2.2.3 Drivetrain Subsystem

The drivetrain uses two brushed drive motors and independent DRV8871 H-bridge drivers. This architecture was selected instead of brushless drive motors because brushed motors provide simple bidirectional control, predictable low-speed behavior, and a lower firmware burden. Each DRV8871 channel includes local decoupling and hardware current limiting. The final presentation recorded the current-limit design target as approximately 2 A per channel using 32 kΩ current-limit resistors.

The drivetrain speed target was evaluated with the wheel-speed relation

$$v = \frac{\pi D \cdot \mathrm{RPM}_{wheel}}{60}, \tag{1}$$

where v is linear speed, D is wheel diameter, and RPM_{wheel} is wheel rotational speed. Solving Equation (1) for a 2 m/s target gives

$$D_{min} = \frac{60v}{\pi \mathrm{RPM}_{wheel}}. \tag{2}$$

Using the conservative design-document value of 1068 rpm under battery sag gives $D_{min} = 35.8$ mm. With a 5% printed-wheel tolerance, the nominal wheel diameter must exceed 37.7 mm. The final wheel-selection work stayed above that bound, giving margin for the 2 m/s design target.

2.2.4 Weapon Subsystem

The weapon subsystem uses a front-mounted brushless motor and spinner assembly. The initial integrated-driver approach used the MCF8316A on the PCB, which offered a clean custom-board architecture and possible feedback through the fault and frequency-generator pins. The difficulty was practical bring-up. The team observed inconsistent startup and configuration behavior, and the remaining schedule made continued low-level driver tuning a larger risk than using a known ESC.

The final design uses a commercial ESC. The ESP32 sends a servo-style PWM signal with a 1000–2000 μs pulse range, and the host software limits the weapon command window to the stable range found during testing. The final operating point used approximately 35% command for the electrical load tests, where the weapon path drew about 3 A steady-state and up to 4 A during ramp-up. This command limiting reduced current draw and startup instability while still allowing a functional weapon demonstration.

The motor-speed feasibility check used the first-order BLDC relationship

$$\mathrm{RPM} \approx K_V V_{eff}, \tag{3}$$

where K_V is the motor speed constant and V_{eff} is the effective applied voltage. For a 2000 rpm requirement at a conservative 10 V effective bus, the required value is only 200 KV. The selected 2207-class weapon motor has much more speed headroom than this; therefore, the practical limitations were command stability, mounting, and safety rather than theoretical no-load speed.

2.3 Mechanical Design

The mechanical subsystem packages the PCB, battery, drivetrain, weapon motor, and wiring into a printed PLA/PLA+ chassis. The final geometry is a low rectangular body with rear-biased wheels and a front vertical spinner. The rear drive placement offsets the mass of the front weapon and improves stability during acceleration and weapon spin-up. Figure 3 shows the final CAD assembly, including the chassis, lid, skids, wheel protectors, and weapon supports.

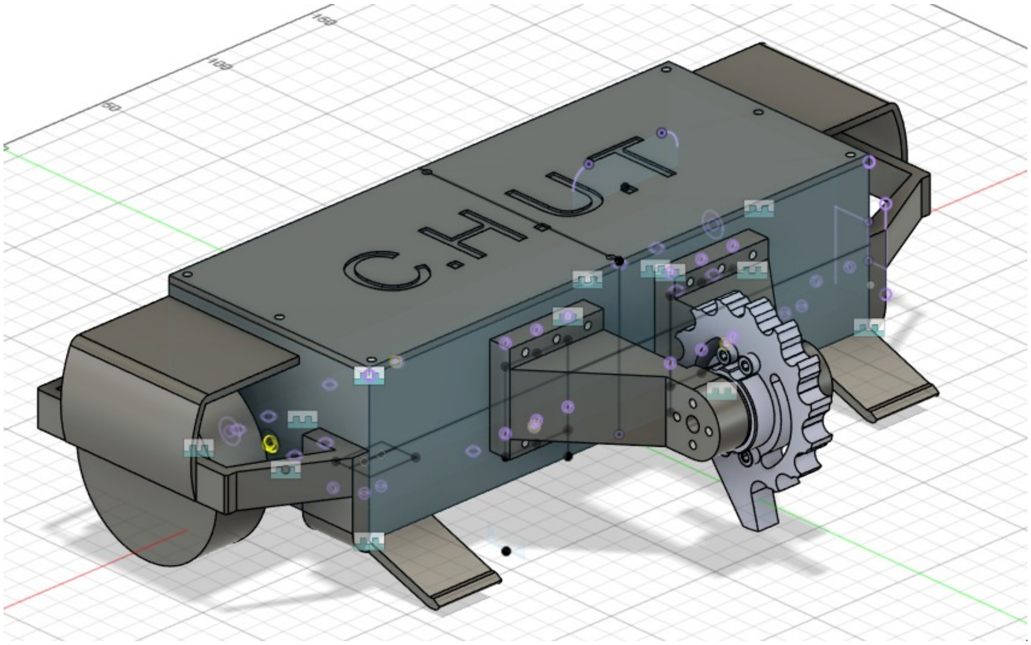

Figure 3: Final CAD assembly showing the rectangular chassis, printed lid, rear-biased wheels, front skids, wheel protection, and front-mounted weapon assembly.

The chassis design followed an iterate-print-test workflow. Early printed bodies verified the main internal packaging. Subsequent revisions added skids, revised front motor arms, a more robust wheel-to-shaft interface, and wheel protectors. One important mechanical failure occurred when the first shaft/coupler concept broke. That failure directly justified a stronger wheel-to-motor-shaft mechanism and later front-mount redesign. The final assembly is shown in Figure 4.

Figure 4: Final assembled C.H.U.T. battlebot with printed chassis, lid, skids, wheel protectors, rear wheels, and front spinner assembly.

2.4 Firmware and Operator Interface

The final operator interface uses an Xbox controller connected to a laptop. The Python host program applies dead zones and exponential response curves to the forward and turn axes, mixes them into left and right motor commands, maps those commands into per-motor PWM ranges, and sends BLE messages to the ESP32. The right trigger controls the weapon command, while buttons provide drive stop, weapon off, and program quit functions.

The firmware design favors deterministic safety over complexity. It clamps all drive commands to the valid 8-bit signed range, clamps weapon commands to a nonnegative range, returns a status string after valid commands, and sends an error for unrecognized messages. It also stops all outputs on disconnect and on watchdog timeout. This approach made the final behavior easier to test because the same command path controlled both normal driving and safety shutdown.

3 Verification

3.1 Verification Strategy

Verification focused on the failure modes most likely to make the robot unsafe or nonfunctional: wireless latency, loss-of-link behavior, power-rail stability, motor current under combined loading, drivetrain operation, weapon operation, and integrated mechanical packaging. Full requirement details are in Appendix A. This section summarizes the principal measurements and observations from final testing.

3.2 Wireless Control and Failsafe Results

The final wireless tests verified the control requirement. The BLE command link maintained stable command reception at 50 Hz over a 15 ft line-of-sight link. End-to-end command latency was measured at 75 ms, which satisfies the 100 ms requirement. Motor outputs disabled within 250 ms of communication loss, satisfying the safety portion of HLR-2. These results validate both the host-side heartbeat and the ESP32 firmware timeout path.

Table 2: Wireless control verification results.

Metric	Requirement	Measured result	Status
End-to-end response latency	< 100 ms	75 ms	Pass
Link range	Stable command reception at demo distance	Stable at 15 ft line of sight	Pass
Command update rate	At least 50 Hz target	50 Hz stable reception	Pass
Communication-loss shutdown	≤ 250 ms	Disabled within 250 ms	Pass

3.3 Electrical Stability Results

The power subsystem was verified under simultaneous drivetrain and weapon loading. The regulated 3.3 V rail remained within 5% tolerance during the combined-load condition, satisfying the 3.0–3.6 V high-level requirement. This result is significant because brownout was one of the main system-level risks: a controller reset during weapon operation or aggressive driving could cause loss of control.

Table 3 summarizes the final current measurements. The drivetrain was tested at a nominal 14.8 V battery input. The drive motors drew approximately 0.6 A steady-state and approximately 0.8 A during ramp-up in the recorded condition. The weapon motor path drew approximately 3 A steady-state and up to 4 A during ramp-up when command-limited to roughly 35%. These values show that the final demo configuration did not overload the regulator or induce a logic brownout.

Table 3: Preserved electrical loading results from final testing.

Subsystem and condition	Steady current	Ramp current	Interpretation
Drivetrain at 14.8 V	0.6 A	0.8 A	Low enough that the drive path did not dominate final electrical stress.
Weapon at roughly 35% command	3 A	Up to 4 A	Dominant load in the combined test condition; stable with command limiting.
3.3 V logic rail under combined load	Within 5% tolerance	Within 5% tolerance	No observed logic brownout in the recorded combined-load test.

Using $P = VI$, the drivetrain consumed about 8.9 W at steady state and 11.8 W during ramp-up in the recorded condition. The weapon consumed about 44.4–59.2 W over the 3–4 A range at 14.8 V. These calculations explain why the weapon subsystem drove the power-integrity risk and why command limiting was a reasonable final integration choice.

3.4 Drivetrain and Integrated Motion

The final drivetrain, PCB, and BLE control path were functionally integrated. The robot accepted operator commands, mapped them to independent left and right motor commands, and demonstrated controlled movement in the integrated test configuration shown in Figure 5. The final software included motor inversion and per-motor PWM tuning because physical motor orientation and real floor behavior did not match the idealized first mapping.

Figure 5: Integrated floor-testing configuration used during final drivetrain and system verification.

The 2 m/s drivetrain target is supported by the design calculation in Equations (1) and (2), the selected wheel diameter, and the final integrated drive tests. During floor testing, the robot accepted left/right differential commands over BLE and maintained stable motion without resetting the controller or browning out the logic rail.

3.5 Weapon Verification

The final weapon path operated through an external ESC rather than the integrated MCF8316A driver. The external ESC resolved the main practical issue with commutation and made the weapon commandable from the ESP32. Final testing showed stable electrical operation at roughly 35% command, with 3 A steady-state and up to 4 A ramp current. The weapon remained functional during the integrated testing condition while the 3.3 V logic rail stayed within tolerance.

The 2000 rpm weapon-speed requirement is also supported by the speed-margin calculation in Equation (3). At a conservative 10 V effective bus, the required motor constant is only 200 KV, which is far below the selected 2207-class motor rating. In practice, the limiting factors were ESC startup behavior, safe command limiting, and mechanical mounting rather than motor speed headroom.

3.6 Mechanical Verification

Mechanical verification consisted of CAD fit checks, printed prototype iterations, physical assembly, and integrated testing with the final chassis. The final mechanical design successfully packaged the PCB, battery, wiring, drive motors, skids, wheel protectors, and weapon mount in a coherent printed assembly. The project also recorded a non-routine shaft/coupler failure and subsequent redesign, which improved the final mounting strategy.

A scale measurement recorded a pair of newly printed orange protective parts at approximately 0.14 lb, and the final robot was assembled from printed PLA/PLA+ parts consistent with the competition material rules. The final chassis packaged the PCB, battery, wiring, drive motors, skids, wheel protectors, and weapon mount while remaining serviceable for testing and debugging.

4 Cost and Schedule

4.1 Parts Cost

Table 4 summarizes the final parts cost estimate. Retail cost is the estimated replacement cost. Actual cost reflects items that were purchased directly or counted as lab/course stock. Passive components and printed chassis material are grouped because many of those items came from common lab inventory.

Table 4: Estimated parts cost for the final robot.

Item	Manufacturer / source	Qty.	Retail cost ($)	Actual cost ($)
Repeat Mini Brushed Mk2 drive motor	Repeat Robotics	2	40.00	40.00
2207 weapon motor / hub motor	Repeat Robotics / RC-class motor	1	50.00	50.00
External weapon ESC	RC ESC	1	22.00	22.00
4S 650 mAh LiPo battery	Tattu	1	24.00	24.00
ESP32-C3-WROOM-02 module	Espressif	1	3.28	3.28
DRV8871 H-bridge driver	Texas Instruments	2	5.46	5.46
LMR51430 buck regulator	Texas Instruments	1	1.58	1.58
MCF8316A BLDC driver on PCB	Texas Instruments	1	3.91	3.91
USB-to-UART adapter	DSD TECH or equivalent	1	13.99	13.99
Custom PCB fabrication	PCB vendor	1	20.00	20.00
Connectors, switch, TVS, passives	Mixed / lab stock	1	15.00	0.00
PLA/PLA+ printed chassis material	Lab stock	1	15.00	0.00
Fasteners, bearings, wheel hardware	Mixed	1	20.00	20.00
Total			234.22	204.22

4.2 Labor Cost

The course-recommended labor estimate is

$$C_{labor} = (\text{hourly rate})(\text{hours worked})(2.5). \tag{4}$$

Table 5 uses an ideal engineering labor rate of $45/h and estimated actual hours from the semester-long design, fabrication, integration, and documentation effort.

Table 5: Estimated labor cost using the ECE 445 labor multiplier.

Team member	Hours	Rate ($/h)	Labor cost ($)
Abhinav Garg	95	45	10,687.50
Rahul Krishnamoorthy	95	45	10,687.50
Shobhit Sinha	100	45	11,250.00
Total	290		32,625.00

Combining the retail parts estimate with labor gives a total project cost of $32,859.22. Using actual parts cost gives $32,829.22. These values are not proposed commercial manufacturing costs; they are course-style engineering development estimates.

4.3 Schedule

Table 6 summarizes the work completed through the semester. A more detailed week-by-week schedule with team-member responsibilities is included in Appendix B.

Table 6: Schedule summary and division of work.

Period	Work completed	Primary owners
January 2026	Chose battlebot concept, studied rules and reference designs, identified major subsystems.	All
Early February 2026	Selected preliminary motor, battery, control, and weapon architecture.	All; Rahul on power, Abhinav on control, Shobhit on packaging
Feb. 23–Mar. 4	Completed schematic and initial PCB design around ESP32, buck regulator, DRV8871, and MCF8316A.	Rahul and Abhinav, with Shobhit integration review
Mar. 5–Mar. 27	Routed PCB, organized BOM, prepared fabrication files, and began chassis CAD.	Rahul and Abhinav on PCB; Shobhit on CAD
Mar. 31–Apr. 16	Printed first chassis revisions, fit drive motors, iterated skids, selected wheels, and reached first full assembly.	Shobhit lead, all integration support
Apr. 21–Apr. 30	Built BLE controller workflow, tested drive path, debugged weapon driver, moved weapon to external ESC, added wheel protection.	Abhinav on firmware/control, Rahul on electrical debug, Shobhit on mechanical revisions
May 1–May 3	Weighed printed parts, captured final assembly, and performed final electrical and wireless tests.	All

5 Conclusion

5.1 Accomplishments

C.H.U.T. reached a functional integrated battlebot platform with a custom PCB, ESP32-C3 wireless command path, independent brushed drivetrain control, printed chassis, and externally controlled brushless weapon subsystem. The strongest final results were the verified control and power behaviors: 75 ms end-to-end command latency, stable 50 Hz command reception at 15 ft, motor shutdown within 250 ms of communication loss, and a 3.3 V rail that stayed within 5% tolerance under simultaneous drivetrain and weapon loading.

The project also produced meaningful engineering revisions rather than simply following the first architecture. The team abandoned self-righting geometry when it conflicted with the spinner layout, redesigned mechanical parts after a shaft/coupler failure, and replaced the original integrated MCF8316A weapon-driver path with an external ESC when driver configuration became a schedule risk. Those changes improved the probability of a working final robot while preserving the custom-board control and drivetrain core.

5.2 Future Work

The final tests showed strong wireless-control and electrical-stability performance, and the integrated platform successfully combined the PCB, firmware, drivetrain, weapon command path, and printed chassis. Future validation should extend that test set with repeated match-length drive runs, marked-distance speed trials, and optical-tachometer weapon-speed measurements while logging rail voltage.

Future hardware revisions should simplify the weapon-control architecture. A later PCB could remove the unused integrated MCF8316A path or replace it with a brushless-control solution that the team has already validated on a bench setup. The design should also add motor-speed feedback, improve packaging density, retain the wheel protectors, and capture oscilloscope traces directly in the notebook for all high-current transient tests.

5.3 Ethical and Broader Impacts

This project was guided by the IEEE Code of Ethics, especially the obligation to prioritize public safety, report limitations honestly, and avoid behavior that could harm others [5]. The robot contains a high-speed rotating weapon and a high-discharge LiPo battery, so testing must use a restrained setup, clear standoff distance, supervised battery handling, and an accessible power-disconnect procedure. Wireless control must fail safe, and the final firmware directly addresses that requirement by disabling motor outputs on disconnect and watchdog timeout.

The broader impact of the project is educational rather than commercial. It demonstrates compact embedded control, power electronics, mechanical packaging, and safety tradeoffs in a visible robotics platform. The same constraints that make the robot interesting also make honest documentation important: the team should not claim unmeasured performance as verified, and any future operation should stay inside the tested electrical, mechanical, and safety limits.

References

[1] Espressif Systems, “ESP32-C3-WROOM-02 and ESP32-C3-WROOM-02U Datasheet,” https://www.espressif.com/sites/default/files/documentation/esp32-c3-wroom-02_datasheet_en.pdf, accessed: May 5, 2026.

[2] Texas Instruments, “DRV8871 3.6-A Brushed DC Motor Driver With Internal Current Sense (PWM Control),” https://www.ti.com/product/DRV8871, accessed: May 5, 2026.

[3] ——, “MCF8316A Sensorless Field Oriented Control Integrated FET BLDC Driver,” https://www.ti.com/product/MCF8316A, accessed: May 5, 2026.

[4] ——, “LMR51430 SIMPLE SWITCHER Power Converter 4.5-V to 36-V, 3-A Synchronous Buck Converter,” https://www.ti.com/product/LMR51430, accessed: May 5, 2026.

[5] IEEE, “IEEE Code of Ethics,” https://www.ieee.org/about/corporate/governance/p7-8.html, accessed: May 5, 2026.

A Requirement and Verification Table

Table 7: Final requirement and verification table.

ID	Requirement	Verification procedure	Final result
HLR-1	Robot mass shall be less than or equal to 2.00 lb and use approved printed thermoplastic materials.	Weigh printed assemblies and inspect printed material selection.	Printed protective parts measured approximately 0.14 lb; final chassis used PLA/PLA+ printed parts consistent with the material rules.
HLR-2	Wireless command response shall be less than 100 ms.	Send timestamped controller commands over BLE and measure end-to-end response latency from host command to ESP32 acknowledgement or motor-output change.	Pass. Measured latency was 75 ms.
HLR-3	Motor outputs shall disable within 250 ms of communication loss.	Remove or interrupt the wireless link and measure time until drive and weapon outputs are commanded off.	Pass. Outputs disabled within 250 ms.
HLR-4	Drivetrain shall support stable operator-controlled motion with target speed near 2 m/s.	Drive the integrated robot under wireless control; compare selected wheel diameter against the wheel-speed design calculation.	Integrated BLE drive control functioned, and the selected wheel diameter satisfies the design calculation for the 2 m/s target.

ID	Requirement	Verification procedure	Final result
HLR-5	Weapon shall reach at least 2000 rpm without causing brownout or motor-control shutdown in the demonstrated load condition.	Secure the robot, command weapon spin-up, and monitor the 3.3 V rail and current draw.	Weapon operated through the external ESC at roughly 35% command, drawing about 3 A steady-state and up to 4 A during ramp-up without logic brownout.
PS-1	3.3 V logic rail shall remain within 3.0–3.6 V under simultaneous drivetrain and weapon loading.	Measure the 3.3 V rail with a DMM or oscilloscope while operating both drivetrain and weapon.	Pass. Rail remained within 5% tolerance during combined loading.
PS-2	Drivetrain current shall remain within driver and wiring limits during the demonstrated drive condition.	Measure battery-side or motor-path current during steady operation and ramp-up.	Pass for recorded condition. Drive motors drew 0.6 A steady-state and 0.8 A during ramp-up at 14.8 V.
PS-3	Weapon current shall remain within ESC and wiring limits during the command-limited demonstrated condition.	Measure weapon current during steady operation and ramp-up at the final command limit.	Pass for recorded condition. Weapon drew about 3 A steady-state and up to 4 A during ramp-up at roughly 35% command.
CTRL-1	ESP32-C3 shall boot, run firmware, and accept wireless commands.	Program firmware over UART/USB path, connect over BLE, send valid `m1`, `m2`, and `wm` commands, and observe status response.	Pass. BLE controller and ESP32 firmware operated in the final configuration.
CTRL-2	Firmware shall clamp invalid motor commands and stop outputs on explicit stop command.	Send out-of-range and stop commands; verify output clamping and zero-output state.	Pass by firmware implementation and integration testing.

ID	Requirement	Verification procedure	Final result
CTRL-3	Host controller shall send periodic heartbeat commands faster than the firmware timeout.	Inspect host control timing and run connected idle test.	Pass. Host heartbeat interval was 200 ms versus 500 ms firmware timeout.
MECH-1	Chassis shall package PCB, battery, drive motors, weapon motor, skids, lid, wheel protectors, and wiring without preventing assembly.	Assemble the final robot and inspect fit, clearance, and serviceability.	Pass. Final robot was assembled and photographed.
MECH-2	Printed weapon and drivetrain mounts shall survive integration testing without immediate structural failure.	Run drivetrain and weapon in secured/integrated test setup and inspect mounts afterward.	Pass for final integration state. Earlier shaft/coupler failure was redesigned.

B Detailed Schedule

Table 8: Detailed schedule by period and team member.

Period	Task	Owner(s)	Output
January 2026	Study battlebot rules, reference robots, and feasible chassis concepts.	All	Selected compact two-wheel battlebot direction with front weapon.
Feb. 1–13	Define problem, solution, high-level requirements, and first subsystem split.	All	Proposal architecture with power, control, drivetrain, weapon, and mechanical blocks.
Feb. 14–24	Select ESP32-C3 controller, battery class, brushed drivetrain approach, and brushless weapon path.	Abhinav, Rahul, Shobhit	Initial component set and control-interface assumptions.
Feb. 23–Mar. 4	Capture schematic and review programming, reset, and debug access.	Rahul and Abhinav; Shobhit review	Full schematic with ESP32-C3, buck regulator, DRV8871 channels, and MCF8316A path.
Mar. 5–12	Route PCB, organize footprints, check board manufacturability, and order parts.	Rahul and Abhinav	Routed PCB and BOM.
Mar. 13–19	Spring break / limited project work.	All	No major hardware changes.
Mar. 21–31	Begin board assembly and first chassis CAD/print preparation.	Rahul and Abhinav on PCB; Shobhit on CAD	Assembled control board in progress and first chassis print.
Apr. 1–6	Fit drive motors into chassis, revise skid geometry, and print updated chassis/skids.	Shobhit lead; all support	First real fit checks and revised printed parts.
Apr. 7–16	Debug weapon shaft/coupler failure, choose wheels, print holders, and complete first full assembly.	Shobhit lead; Abhinav/Rahul integration support	Redesigned coupling and first full physical assembly.

Period	Task	Owner(s)	Output
Apr. 17–24	Build BLE controller flow, verify drive command path, revise weapon motor arms, and assess brushless-driver issue.	Abhinav on firmware; Rahul on electronics; Shobhit on mounts	Functional drive-control path and decision pressure on weapon driver.
Apr. 25–30	Move weapon to external ESC, test electrical loading, tune controller ranges, and finalize wheel protectors.	All	Final command-limited weapon path and final CAD assembly.
May 1–3	Weigh printed parts, capture final robot, and record final control and electrical verification results.	All	Final assembled robot, current measurements, wireless latency, and shutdown timing.
May 4–6	Complete final documentation and report.	All	Final report and submission package.

C Additional Design Figures

Figure 6 shows the full schematic screenshot used during final report preparation. It is included in the appendix because the full schematic is too dense for the main design narrative but is useful for reviewing subsystem boundaries and signal names.

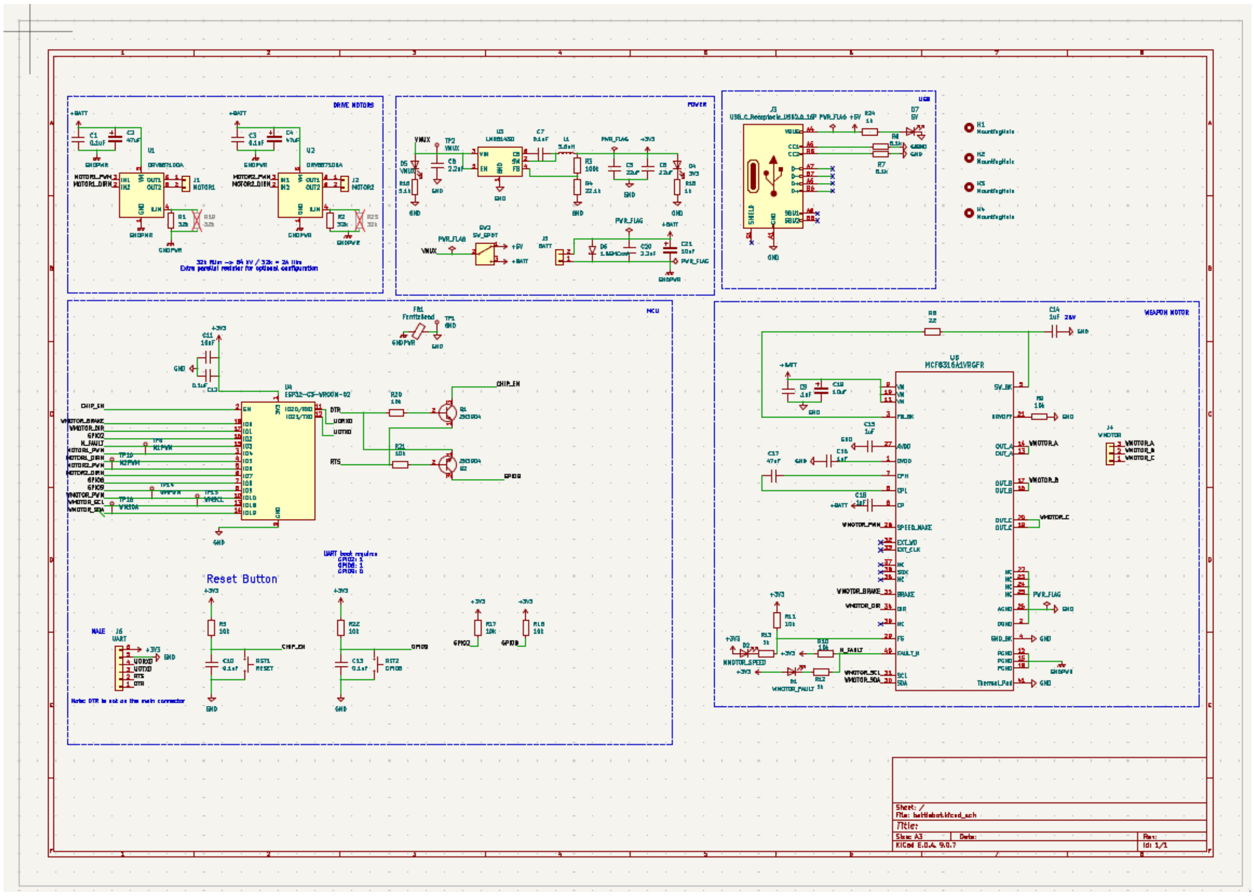

Figure 6: Full custom PCB schematic showing drivetrain drivers, power regulation, USB/UART support, ESP32-C3 controller, and original MCF8316A weapon-driver region.

Figure 7 shows the routed PCB at full appendix width for layout review.

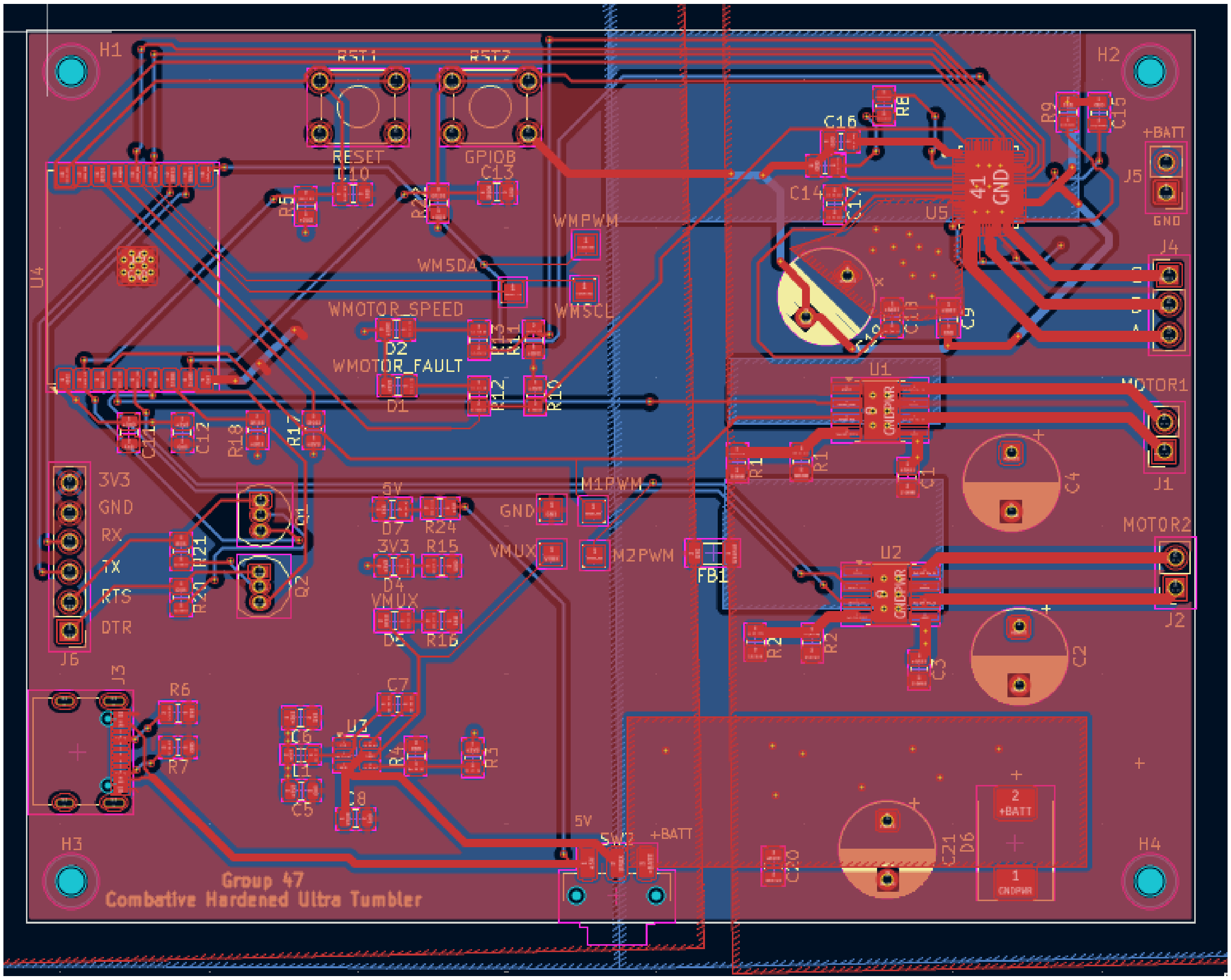

Figure 7: Routed PCB layout at full appendix width.