

ANT-WEIGHT BATTLEBOT – DC HAMMER

By

Carson Sprague

Ian Purkis

Gage Gathman

Project Proposal for ECE 445, Senior Design, Spring 2026

TA: Haocheng Yang

01 May 2026

Project No. 37

Contents

1. Introduction	1
2. Design.....	2
.....	2
2.1 User Interface Subsystem	2
2.1.1 Microcontroller	2
2.1.2 Kill Switch	3
2.2 Sensor Subsystem	3
2.2.1 Accelerometer.....	3
2.2.2 Ultrasonic Sensor	3
2.2.3 Limit Switches	4
2.2.4 Battery Voltage Sensor	4
2.2.5 TTL Chips	4
2.2.6 Gate Drivers	4
2.3 Hammer/Wedge Subsystem	4
2.3.1 Hammer	4
2.3.2 Wedge	6
2.3.3 High Torque Motor	6
2.4 Power Subsystem.....	6
2.4.1 Battery.....	6
2.4.2 Voltage Regulator – 3.3 V	6
2.4.2 Voltage Regulator – 5.0 V	7
2.5 Software Subsystems.....	7
2.5.1 Bluetooth	7
2.5.2 Input Decoder	7
2.5.3 Drive Inversion	7
2.5.4 Software Motor Control.....	7
2.6 Drive Subsystem.....	8
2.6.1 H-Bridge NMOSFETs.....	8
2.6.2 Drive Motors	8

2.7 Tolerance Analysis – Critical Subsystem	8
3. Ethics, Safety, and Societal Impact	8
3.1 Ethics	9
3.2 Safety	9
3.3 Societal Impact.....	9
References	10

1. Introduction

Our battle Bot, “DC Hammer”, is an optimized smashing machine. Our primary design philosophy centers around efficiency; fulfilling competition weight requirements, automating attack swings, and even enabling flip detection and controls inversion all demand efficient use of resources whether those are digital, analog, or even mechanical. Our goal in designing “DC Hammer” was to develop a well-rounded battle Bot that could effectively deploy offensive and defensive measures against a wide range of diverse opponents. Many battle Bot designs fall short on balance, typically leaning on a single feature to propel the bot to victory. With emphasis on both offense and defense, and a robust/sustainable design, “DC Hammer” will match-up against any opponent and be able to continue competing match after match. The modular subsystems of our bot will allow for a clear roadmap of development. Creating a functioning bot is our priority, with tuneups and additions as the development progresses. These modular subsystems will allow for flexibility with our weapons systems in the future as we build on top of the base bot. In this document we will go over the high-level motivations and expected product of our designs, as well as cataloguing the various components we will use to realize this design. Additionally, we will cover our expectations for the various subsystems in the bot and how we will approach realizing this finished product from an ethics and safety perspective.

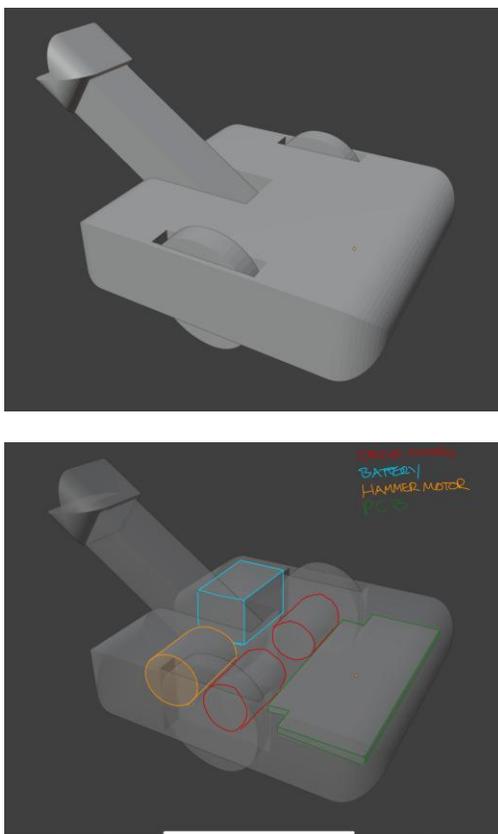


Figure 1 and 2 – Visual Aid (CAD Drawing) of Battle-Bot

2. Design

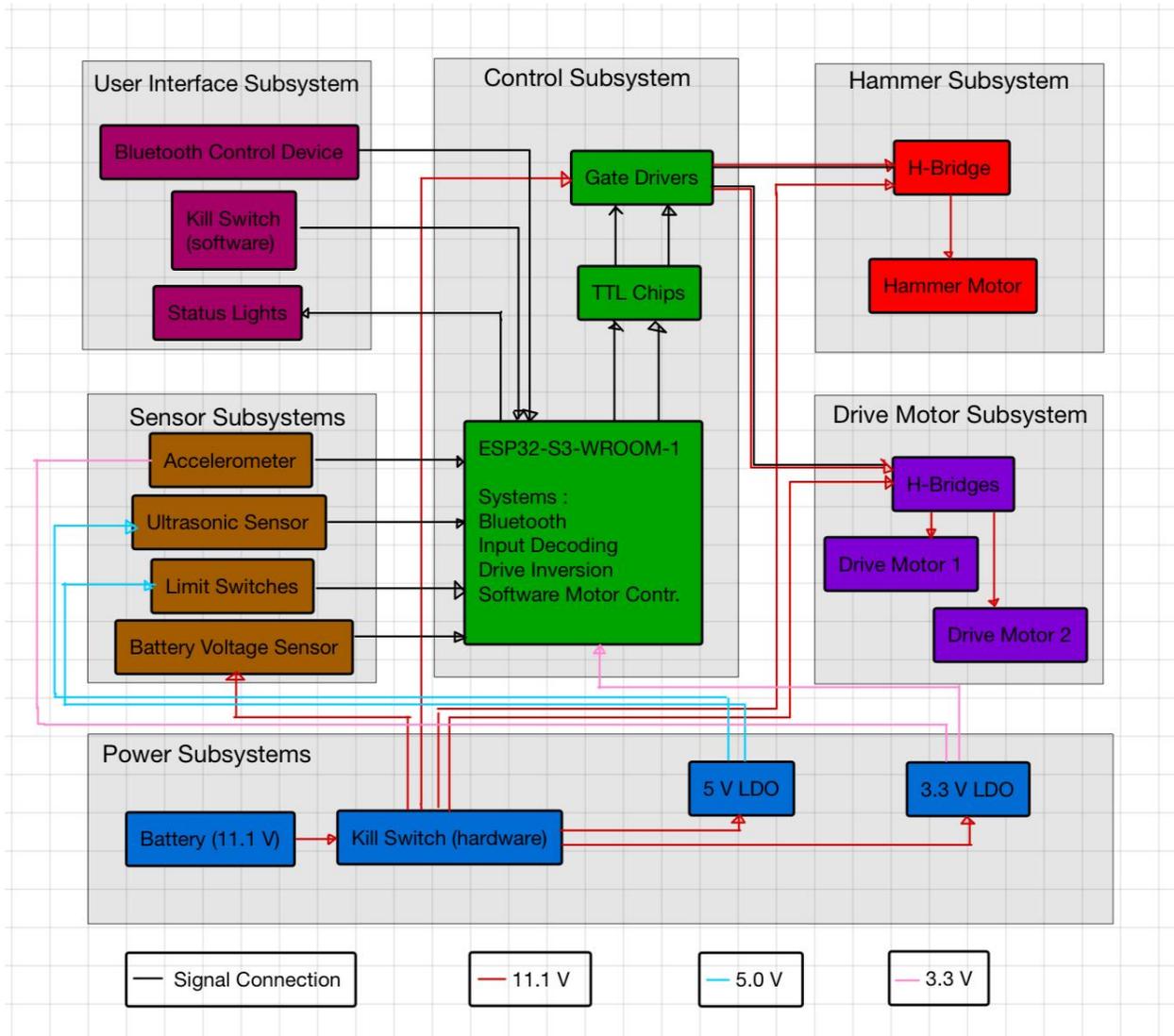


Figure 3 - Block Diagram

2.1 User Interface Subsystem

The user interface subsystem is responsible for providing a seamless interface for the user to control and monitor the battle bot. Users control the bot via a Bluetooth keyboard and can monitor the bot's status via a status LED. A kill switch is located on the bot, as required by competition specifications.

2.1.1 Microcontroller

The microcontroller (ESP32-S3-WROOM-1) has built-in Bluetooth support, supporting Bluetooth 5 and BLE. We will leverage the official Espressif Bluetooth Main API [2] for setting up the connection and reading input from the connected Bluetooth device. The MCU will run a loop that will process keyboard input. Along with taking input, the MCU will blink an LED depending on the bot status via one of its

many GPIO pins. Status codes have yet to be formally defined, but a rich enough set should be provided to promote a good user experience.

Specifically, as the MCU for our battle bot, this will interface with the ultrasonic sensor, the accelerometer, the limit switches, the battery voltage sensor, and the TTL chips on the hardware side, and will also handle the Bluetooth IO connection. This serves as the heart of our robot, to this extent it is the most important single item. Thus, it is critical it remains powered, and for which it requires a 3.0-3.6 V power supply, capable of 500 mA of current draw to function.

Requirements

- 1. The latency from keypress on the keyboard to motor control should be under 50ms. Keeping a low response time is critical to provide a clean user experience and a competitive advantage.*

2.1.2 Kill Switch

A physical kill switch is required per competition guidelines, but it is important nonetheless to promote the safety and wellbeing of those nearby. We will also provide a kill switch remotely via the keyboard to further promote safety.

Requirements

- 1. The kill switch should be easily accessible, if necessary, but difficult/impossible for a rival battle bot to trigger it. The user should be able to trip the kill switch in under 3 seconds (without harming themselves).*

2.2 Sensor Subsystem

The sensor subsystem collects data about the current orientation of the battle bot and position relative to the other battle bot. Namely, an accelerometer is used to keep track of the vertical acceleration of the bot, and an ultrasonic sensor is used to track when a bot is in front of us in striking range. We aim to make our control system robust to flips by automatically adjusting based on orientation.

2.2.1 Accelerometer

We use an MC3416 accelerometer as it should satisfactorily provide us with vertical acceleration data. By using an accelerometer and keeping track of the vertical acceleration, we can dynamically update our motor control system based on our orientation. This provides a seamless control interface for the user. The MC3416 uses the I2C protocol to transmit data to the MCU (ESP32-S3) via two GPIO pins and requires 3.3 volts. Pullup resistors are also required on output pins [4].

2.2.2 Ultrasonic Sensor

We use an Adafruit 4700 ultrasonic sensor not only because it should be satisfactory in terms of getting the distance to the nearest object in front of us, but also because it is functional with both 5v and 3.3v [1], which allows us more flexibility in our power subsystem design. This sensor, like most other ultrasonic sensors, transmits data via two GPIO pins, sending a trig and echo signal. We can compute the distance to the nearest object directly in front of us in software.

2.2.3 Limit Switches

Two simple switches will be used to pull down pin inputs to our MCU, to indicate an end of motion for our attack arm. These will prevent the over-currenting of our hammer motor and provide simple feedback. When operating a brushed DC motor, its speed will change with voltage, and the battery voltage will vary with the state of charge. These constitute critical components, as failure of them to detect a stop of the motor may cause component failure. This is due to the high torque of the hammer motor that may result in damage to our chassis.

2.2.4 Battery Voltage Sensor

This simple subsystem will handle battery voltage monitoring. It will function to stop our LiPo batteries from being over-discharged. The voltage will be taken into our MCU and be used as a key to turn off our motors in the case of a low battery. Failure of this component is not critical for our robot, but will be costly, as ruined batteries may never re-charge properly. This will be a passive system, with only a voltage divider and diode protection for our MCU.

2.2.5 TTL Chips

To take our digital logic out of the MCU and limit the number of pins required, we will do some post processing using TTL chips. Using a given number of pins to control the direction of the drive motors, we will be able to control the H-bridges. This component thus takes the MCU outputs and “switches” the path on the H-bridge, more conveniently than using BJTs or other components, still it remains a 3.3 V output which will have to be stepped up by the gate drivers.

2.2.6 Gate Drivers

We have chosen DGD0211CWT-7 as gate drivers. These chips will be able to take our final PWM signals from the TTL chips at 3.3 V and apply the battery voltage ~ 11 V to the gates of the H-bridge MOSFETs. These devices serve a critical role in the control circuit, bridging the gap between the “low” and “high” voltage segments.

2.3 Hammer/Wedge Subsystem

The primary offensive tool is the hammer/wedge “attack arm”. This arm is driven by a high torque motor and activated by a proximity-detecting ultrasonic sensor for faster than human reaction time when opponents enter attack range. The attack swings will also have a user input trigger coming from a key on the laptop controlling the bot: likely the space bar. Additionally, the arm will have two default or resting positions to allow for a variety of methods for offense.

2.3.1 Hammer

The first attack mode, and namesake of the bot, is the hammer. The resting position will be high above the bot at an obtuse angle relative to the arena surface to allow for the greatest work done, or energy delivered in the resulting swing (the angle will be approximately 115° or 2 rad).

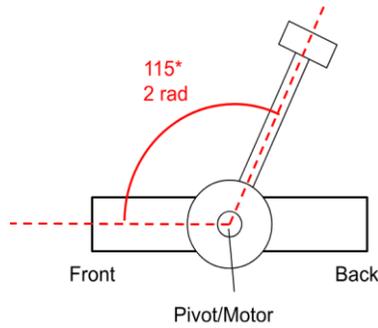


Figure 4 – Hammer Arm Action Diagram

Figure is not a visual representation of robot appearance, simply to visualize the position of the hammer relative to the rest of the bot.

A potential addition to the hammer arm that we'd like to explore is the addition of a kinetic payload to the hammer head. In this implementation of the hammer head, we'd effectively have a second impact for every hammer swing and maximize the energy transfer of the hammer impact by taking advantage of the net forces experienced by the payload during the motion. We would hollow out the inside of the hammer head to be a tube-shaped shaft, housing a steel bearing ball. The ends of the hammer head would have circular openings that allow for the bearing ball to extend partially outside of the hammer dimensions when forced toward the opening. The material, steel, of the bearing ball and more focused point of impact should allow for a more devastating transfer of energy upon impact, especially against other battle bots.

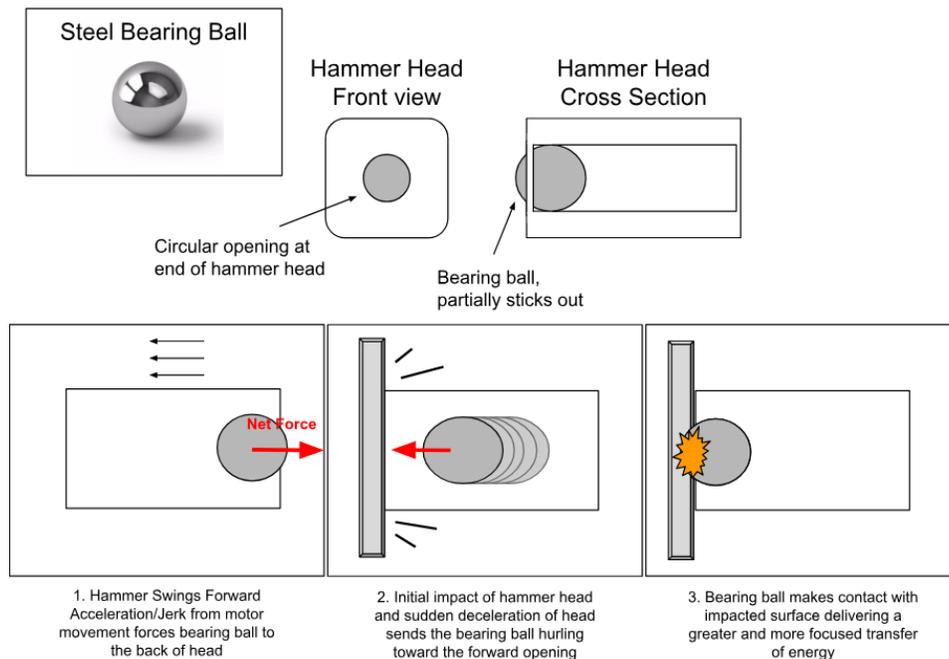


Figure 5 – Hammer Ball Bearing Mechanism

2.3.2 Wedge

The secondary attack mode will have a default resting position about -5° below the central plane of the robot (approximately -0.1 radians). Additionally, the shape of the hammer head will include a wedge on either end of the head. The goal of this mode is to get the wedge underneath the chassis/drive trains or end effectors of other robots and then execute a forceful upward swing to attempt to disable opposing bots by flipping them.

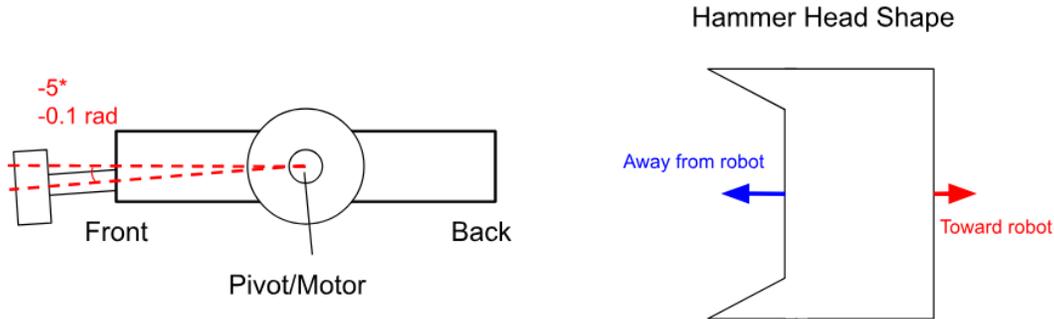


Figure 6 – Wedge Diagram

2.3.3 High Torque Motor

We have selected a 12 V, 45 RPM, brushed DC geared motor, with a rated stall torque of 50 kg-cm. This motor will conduct any flipping or smashing rotations of the hammer and should be adequate for any roughly 1 kg battle bots it should face. For this purpose, this motor requires 0.68 to 2.19 A of current, rated and stall respectively. In combination with the drive motors, this will compromise the majority of the current required by our robot.

2.4 Power Subsystem

To give life to our battle bot, a power system is needed to supply necessary power to all components, including drive motors, hammer motor, MCU, sensors, etc. A variety of components are used to ensure a safe and robust system to deliver power where necessary.

2.4.1 Battery

The main battery is to provide requisite power for the entire system. For this purpose, a 3s (11.1v) LiPo battery of sufficient capacity is required. This battery will connect directly to the kill switch, through which there will be connections to all major components, particularly the voltage regulators and to the motors. This is a critical component due to the fact that it must provide sufficient current and voltage to three 12v DC motors which represent the majority of our battle bot's power consumption. Particularly, this battery must be able to store sufficient energy to power these motors for 3-minute bouts.

2.4.2 Voltage Regulator – 3.3 V

Multiple components on our battle bot will require a 3.3 V power supply. Principally, the MCU, an ESP32-S3 chip, will require a rated 3.3 V and 0.5 A of power to operate with a minimum and maximum operating voltage of 3.0 V and 3.6 V respectively. To this constraint, we have chosen the AZ1117C-3.3, whose datasheet indicates a $\pm 1\%$ accuracy, minimal regulation, and a minimum of 1 amp of load.

Regarding the decision between using a buck converter and a low drop-out voltage regulator, we chose the latter due to its size and weight, although it may be less efficient it makes up for that in weight savings.

2.4.2 Voltage Regulator – 5.0 V

Currently two sensor systems will operate on this 5 V power supply, our Ultrasonic Sensor and limit switches. The combination of these three will not draw more than 20 mA. The AZ117C-5.0 possesses similar qualities to the AZ117C-3.3, with 1% accuracy, minimal regulation, and at least 1 amp of load capability. This component's functionality is a requisite for the control systems of the weapon motor.

2.5 Software Subsystems

2.5.1 Bluetooth

For this project to be functional, we will need a way to communicate with a control device. For this purpose, we have selected a Bluetooth connection for its simplicity and widespread availability. The Bluetooth handler will interface between our onboard ESP32-S3 chip, on which it will run, and the remote-control device, principally a computer. This is a fundamental design component to compete; the battle bot must be able to move under control. This block will send the decoded signals to the Input Decoder.

2.5.2 Input Decoder

This block will serve as the interface between the Bluetooth connection, and the drive inversion module which eventually controls the output to the motors. It will need to be simple and reliable in the sense that it will need to constantly decode the incoming inputs and determine what messages to send downstream to the motors, whether that be a left/right/backwards/forwards, speed, attack, or mode change input, and alter the outputs based on all these as it will need to function in either hammer or flipper mode. This module will additionally need to decode the input of the ultrasonic sensor, to determine whether or not to swing the hammer. All of these functionalities need to be handled quickly. As response time will be imperative, any slowdowns will impair robot performance.

2.5.3 Drive Inversion

This block will take inputs from the input decoder and accelerometer, to determine the orientation of the battle bot and from there, determine whether robot controls will need to be flipped or not. If the system cannot poll the accelerometer frequently enough to switch the controls when flipped, or if it causes too much delay when processing, then it will be considered failing.

2.5.4 Software Motor Control

This block is the final module of software; it will take the highly processed inputs of the previous software blocks and be the final output layer. This block will use the built-in PWMMC functionality of the ESP32-S3, for its speed control and various other utilities. This block will effectively determine the drivability of the robot, which if unsteady or changeable will drastically affect the efficacy of the battle bot to damage opponents effectively.

2.6 Drive Subsystem

2.6.1 H-Bridge NMOSFETs

For our motor to operate properly we will need high-power MOSFETs for our H-Bridge. The chosen devices are Onsemi's FDC655BN, which are 30 V, 6.3 A rated MOSFETs. These should be overrated for our application to ensure safety in our circuit. If one of these devices were to fail, it would not only constitute a robot failure but may be critical to battery safety. However, since these devices are overrated, they should not be a limiting element in our design. Like any other critical path electrical component, failure will be caused by loss of voltage, or improper operation.

2.6.2 Drive Motors

Our main two drive motors are 12 V, 460 RPM, brushed DC geared motors. These will propel our battle bot, which is a critical feature for obvious reasons. Their current requirements are 100-1600 mA for the no-load to stall current operating range. This helps us inform how we must compute our battery size. These will interface with the H-bridge modules and the ground. These motors will need to be durable, as this robot will likely experience high-g forces being tossed around.

2.7 Tolerance Analysis – Critical Subsystem

For this device, we have gone with the LDO voltage regulator. This presents the largest risk to our system. Generally considered the simplest form of voltage supply, they are small and cheap. This makes them excellent for our application. However, they are not without their downsides. The efficiency of an LDO regulator decreases with an increasing voltage difference between the input voltage and the output voltage. This efficiency is not a concern for our overall circuit power requirements but is instead a large concern for our thermal requirements. Since these devices can be as low or lower than 30% efficient, a 3.3 V, 1 amp device, may be putting off 7-8 watts of waste heat. This means component temperature rises and thus component failure.

To address this concern, the team has begun looking into solutions to disperse heat or divide load. Two avenues have arisen. The first is simple, using a 5 V LDO in front of a 3.3 V LDO. This would reduce the heat generated by a single component significantly and thus increase our theoretical reliability. However, this introduces possible concerns with noise in our system if these components do not interact well, or alternatively, if one fails, then the entire circuit stops working. An alternative approach is to parallel multiple LDOs with ballast resistors; this would allow the division of heat generation between multiple components, thus increasing reliability of our board.

3. Ethics, Safety, and Societal Impact

Despite the proposed system being a small-scale combat robot/battle bot designed for a controlled competition environment, it still involves mechanisms with intent to do harm. Thus, it's important and necessary to describe our responsibility as engineers to uphold the highest standards regarding ethics, safety, and impact.

3.1 Ethics

The ethical responsibility in this context revolves around containment, preventing misuse, and playing by the rules described in the competition handbook. We want to emphasize that this system is designed strictly for the end of year battle-bot competition, which takes place in an enclosed and supervised testing environment, and is not at all intended to harm people, animals, property, etc. As referenced in the IEEE Code of Ethics I.1 [3], we hold the safety, health, and welfare of the public as the upmost priority. We commit ourselves to creating a safe and contained system and will take measures along the way to ensure that the power of our battle-bot is never in the wrong hands.

3.2 Safety

This battle-bot includes rotating motors and potentially high-speed mechanical components, which would introduce injury risk if improperly handled. Furthermore, all components are powered by electricity, which could pose a threat if safe procedures are not followed during development. We commit ourselves to firstly follow the competition handbook and ensure that our bot meets competition regulations and adheres to the safety standards as defined within them. Secondly, but equally as importantly, we commit ourselves to following lab and general electrical safety standards during development. We understand that, as defined in the IEEE Code of Ethics I.6 [3], if we encounter tasks beyond our training or qualifications, we will reach out to someone who is qualified. Examples include using the machine shop or using high-voltage components. Thirdly and finally, we commit ourselves to implementing control safety. Should one of our systems go out, such as a loss of Bluetooth signal, we will implement fail-safe behavior to completely shut down our system should we ever lose control.

3.3 Societal Impact

Although this is an educational toy battle-bot, the broader societal implications contain the development and/or normalization of autonomous weapon systems. This project is limited and does not pursue autonomous targeting nor serious offensive capabilities, but it is important for us to remain aware of the broader implications during development. We as young engineers use this project to contribute to our hands-on education, safe electronics design, and control systems development. This battle-bot and the skills acquired during development are transferable to a number of constructive applications, such as search and rescue robotics. We hope that by the completion of this project, we've inspired others to use and develop more robots to contribute to the good of society, rather than take away from it.

References

- [1] Adafruit Industries, “Ultrasonic Distance Sensor – 3V or 5V – HC-SR04 compatible – RCWL-1601,” 2019. Available:
https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/307/4007_Web.pdf.
(Accessed: Feb. 9, 2026).
- [2] Espressif Systems, “Bluetooth® Main API,” *ESP-IDF Programming Guide*. Available:
https://docs.espressif.com/projects/esp-idf/en/stable/esp32s3/api-reference/bluetooth/esp_bt_main.html#api-reference. (Accessed: Feb. 9, 2026).
- [3] IEEE, “IEEE Code of Ethics,” 2020. Available:
<https://www.ieee.org/about/corporate/governance/p7-8>. (Accessed: Feb. 12, 2026).
- [4] Memsic Semiconductor Co., Ltd., “MC3416 3-Axis Accelerometer,” 2021. Available:
<https://www.digikey.com/en/products/detail/memsic-inc/MC3416/15292804>. (Accessed: Feb. 9, 2026).