# ANTWEIGHT BATTLEBOT BLADE BLADE

By

Patrick Mugg

Jack Tipping

Sam Paone

Design document for ECE 445, Senior Design, Fall 2025

TA: Gayatri Chandran

29 October 2025

Project no. 16

## Abstract

Ant weight battlebot competitions feature miniature combat robots weighing under 2 lbs. Due to their small size, these bots require careful engineering design including custom chassis, weapon mechanisms, and control systems. Competitors battle in enclosed arenas, aiming to disable or flip opponents. Antweight events emphasize creativity, precision, and design trade-offs, making them a popular entry point for hobbyists interested in robotics and mechanical design. Our ECE 445 senior design project features an antweight battlebot with a front-mounted spinning blade, serving as both its primary weapon and defining visual element. The blade's rotational speed allows it to damage opponents' armor or destabilize them on impact. Mounted low to the ground, it doubles as a defensive barrier, deflecting frontal attacks while maintaining aggressive forward momentum. The blade may also lift to throw opponents off balance or flip them over.

# Contents

# 1. Introduction
The goal is to design an antweight-class battlebot which satisfies the competition constraints while maintaining robustness and offensive capability towards an opponent. The robot must adhere to rules governing maximum weight, dimensions, materials, and weapon safety, while remaining structurally robust and operationally reliable under competitive conditions. This report outlines the system architecture, subsystem-level design decisions, and verification methodology. Primary design goals include stable locomotion, structural integrity, and active engagement features such as a spinning saw weapon and real-time current sensing. Subsystems are divided into mechanical structure, drive system, weapon system, power management, and sensing/control. Design verification will be incremental, beginning with individual subsystem functionality and progressing to full system integration, similar to an approach taken in system-on-chip design validation.  Throughout development, the design must meet course-imposed constraints such as cost, power consumption, run time and weight to qualify as a legal antweight competitor.

## 1.1 Visual Aid
Some explanation on why the image may appear simple: All internals are expected to be hidden by the cube itself and therefore the expectation is our battle bot looks like a floating cube. Attached going outward is a saw which can be raised vertically by an internal servo. In this image, the front of the square is not open, but later in physical design you will see the piece where the saw comes out is etched out allowing for vertical movement.

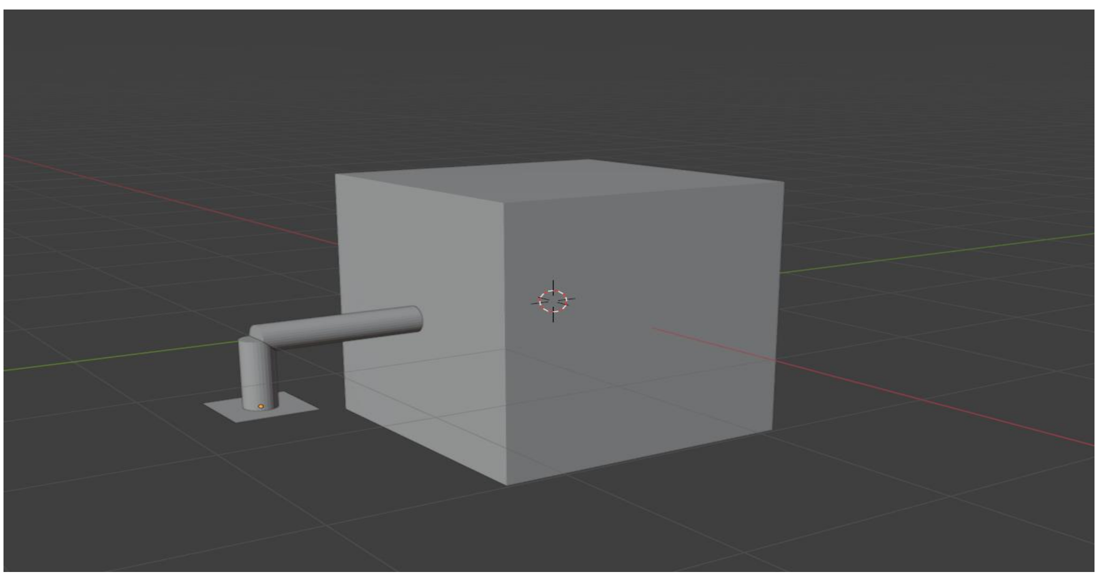

Figure 1 Visual Aid, initial representation

## 1.2 High Level Requirements List

1. The battlebot will be controlled by a laptop via Bluetooth. The user will be able to control the microcontroller to output PWM signals to adjust the saw blade spin speed and lift angle, as well as the DC signals to wheel motors for acceleration and steering.

2. The battlebot will have 2 attack mechanisms, one to spin the saw blade and one to lift it. The lift system shall support a torque load equivalent to 2 pounds of force applied at a 10-inch moment arm from the servo shaft.

3. The battle-bot chassis should handle a blunt force of 40N applied to the top-center region without failure of internal electronics, particularly the PCB housed at this location. This provides protection against opponents employing vertical hammer-style attacks.

# 2 Design

## 2.1 Block Diagram

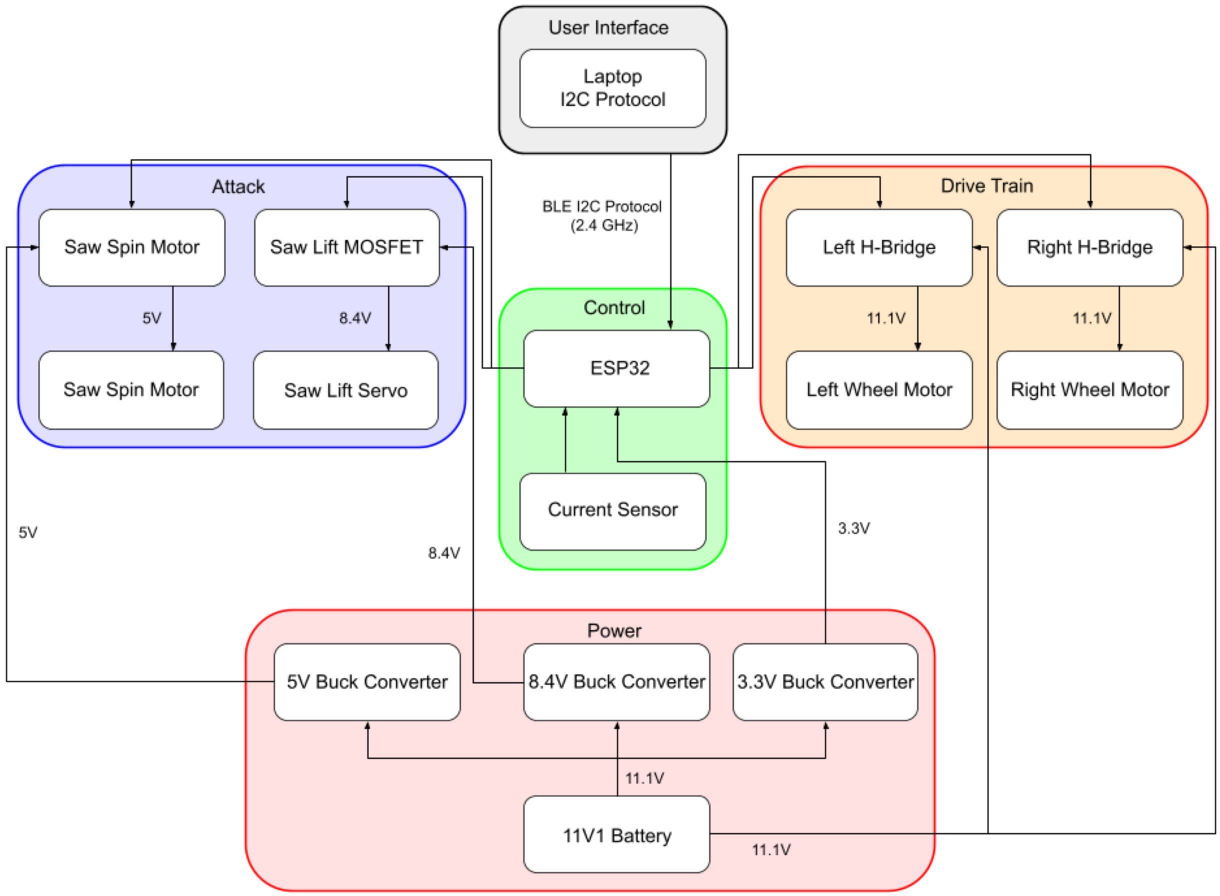


**Figure 2. Block diagram of battle-bot design. Separate subsystems handle power delivery, drive train, attach mechanism and a user interface for remote control and communication.**

## 2.2 Physical Design

Below is an internal layout diagram illustrating the placement of components within the bot. Figure 1 shows the exterior design, while Figure 2 offers a perspective of the interior. The chassis must accommodate the saw blade and its motor, the lift servo, two rear wheels with their respective motors, two freely rotating front wheels, the PCB, the battery, and all interconnecting wiring. To support these components, the design includes a 3D-printed linkage between the servo and the blade motor, as well as dedicated platforms for the servo, PCB, and battery. Additionally, structural housing is provided for the rear wheel assemblies.

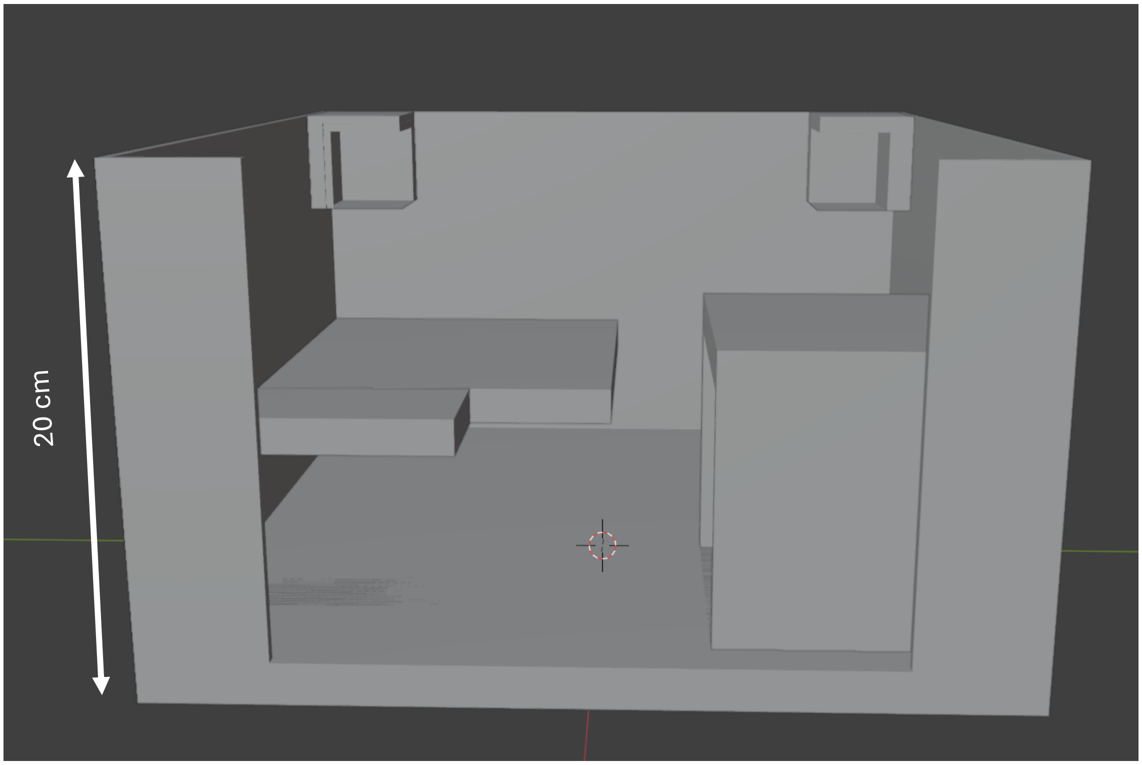


**Figure 3. 3D model of the battle bot chassis designed in Blender.**

To ensure all critical components are housed efficiently within the battlebot's chassis, we compiled a detailed list of the physical dimensions and placement considerations for each subsystem element. The PCB has also been designed in a manner to allow for ease of wiring by having internal components near the middle of the PCB around the microcontroller and external components such as motors connected closer to the edges via general connectors.

| COMPONENT | DIMENSIONS (CM) | COMMENTS |
|---|---|---|
| **OUTER CHASSIS** | 20 × 20 × 12 | Wall thickness: 0.25 cm |
| **PCB** | 10 × 10 × 2 | Placed for ease of wiring |
| **BATTERY** | 6 × 2.5 × 3 | |
| **WHEEL MOTORS (×2)** | 1 × 1 × 2.5 | Rear-mounted, driven wheels |
| **FRONT WHEELS (×2)** | | No drive |
| **LIFT SERVO** | 4 × 4 × 2 | Enclosed in 7 × 6 × 6 housing |
| **SAW BLADE MOTOR** | — | Connected to servo via 3D-printed linkage |

**Table 1. Component dimensions and placement within the chassis**

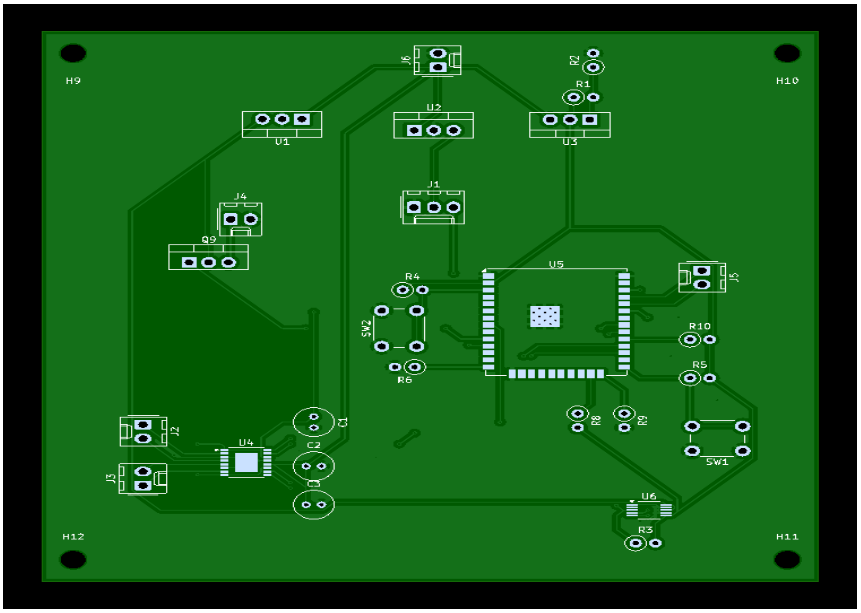


**Figure 4. PCB printed in round 3**

## 2.3 Subsystems

### 2.3.1 Power Subsystem + Current Sensor

The basics of our power system from the outside consist of a battery and voltage regulators. The regulators generate from an 11.1V input three different stepped-down voltages: 3.3V, 5V, and 8V. We generate these voltages to match the requirements of different components. Specifically, the ESP32 microcontroller runs at 3.3V, the saw blade motor runs at 5V, the servo runs at 8V, and the H-bridges receive 11.1V from the battery to give to the wheel motors and 3.3V signals input from the ESP32 to control wheel direction.

Cutting power regardless of the Bluetooth connection is going to be done using the current sensor. The current sensor communicates with the microcontroller digitally with an I2C protocol. It senses current with a shunt resistor and sends a 16-bit value to the microcontroller measuring the current. The microcontroller will be programmed to read this value and disable power to the system that's drawing too much current. This will likely be the saw blade motor if it were to stall.

### 2.3.2 Controller Subsystem

We will use an ESP32-WROOM-32UE microcontroller. The primary benefit is that it has integrated WIFI and Bluetooth. This will allow us to add custom telemetry to our laptop to control our bot remotely, such as controlling saw blade motor speed, raising our blade to flip the opponent bot, or steering the bot. The ESP32 has plenty of peripheral support. There are many PWM outputs, so we can directly drive multiple items at different pulses. There are also ADC inputs that would allow us to detect battery life or sensor information. It provides everything needed, is compact, and doesn't use much power.

Also included in the controller subsystem is the circuitry required to program the ESP32. We will be manually flashing the microcontroller, requiring that some pins are pulled high or low while booting, and may need to switch during programming. We can implement this simply with pull-up and pull-down resistors at the required pins and buttons to switch them. Lastly, a general connector connects the transmitter and receiver pins of the microcontroller to our laptop for programming.

### 2.3.3 Drive Subsystem

We have 4 wheels total, two of which are powered, requiring two H-bridges total to allow for forward and backward motions. Additionally, the H-bridges are controlled by separate PWM signals to allow for steering. The two wheels up front will be omni-directional and unpowered, which will essentially be a ball free to rotate around in a socket. This will assist in steering and keeping the battle bot up to glide around the floor.

### 2.3.4 Attack Subsystem

The attack subsystem consists of the saw motor, servo, and 3D printed components. First, the servo will be sat in a casing near the center of the bot, facing the side. There will be a 3d printed beam that connects to the servo on one end, extending toward the front of the bot. At the other end there will be a cylindrical casing to hold the saw motor upside-down. This allows the saw blade to be lifted through the rotation of the servo at a distance. This allows for flipping the opponent and allowing for different attack angles with the saw.  Reference physical design description above. The attack subsystem is powered as mentioned in power systems.

### 2.3.5 User Interface Subsystem

The user interface subsystem consists of the laptop keyboard used to control the bot remotely. The laptop will connect to the ESP32 via a Bluetooth I2C protocol, so specific key presses can control microcontroller signals to peripherals on the PCB. We will configure this so that W accelerates the bot forward, S accelerates the bot backwards, A turns the bot to the left, and D turns the bot to the right. This will be done by applying DC signals to the wheels' H-bridges through the microcontroller.

To control the attack mechanisms, we will have other keys to switch the saw motor and servo between different presets. The motor can be switched to full power by pressing 'O' or turned off by pressing 'I', while the servo can be switched to the high position by pressing 'L' or to the low position by pressing 'K'. In these cases, the microcontroller will be programmed to send these components a specific PWM signal based on the key last pressed.

## 2.4 Tolerance Analysis

Our two concerns for tolerance are total weight and power consumption. Power consumption is important because we must last 2 minutes to survive the fight, while the 2lb weight limit of the bot is a rule of the antweight competition. We decided to first resolve power, as it decides battery size and therefore battery weight. Below is a table of power estimations, and for the given battery size it shows that with an estimated total power 193W and battery of 850mAh our bot will last a comfortable 2.93 minutes. Having room for error is a must because there are components that can unexpectedly draw excess current. In particular, the motors can each draw more currents while stalled.

| | |
|---|---|
| BV = 11.1v | 1 (LiPo Battery Voltage) |
| BC = 850mAh | 2 (Battery Capacity) |
| MCV = 3.3 V | 3 (Micro Controller Voltage/pin and /instance) |
| MCA = 0.5 A | 4 (Micro Controller Amps/pin and /instance) |
| MCP = MCV * MCA = 1.65W | 5 (Micro Controller Power/pin and /instance) |
| CSA = 2 mA | 6 (Current Sensor Amps/instance) |
| CSP = MCV * CSA = 6.6e-3 | 7 (Current Sensor Power/instance) |
| WA = 2 A | 8 (Wheel Motors Amps/wheel) |
| WP = BV * WA * 2 = 44.4W | 9 (Wheel Motors Power/total) |
| HR = 200 mOhms | 10 (HBridge Resistance/transistor) |
| HP = $4(HR)(WA)^2$ = 80W | 11 (HBridge Power/total) |
| SWR = 200 mOhms | 12 (Saw Resistance/instance) |
| SWP = $(BV)^2 \div HR$ = 16W | 13 (Saw Power/total) |
| SV = 8.4 V | 14 (Servo Voltage/instance) |
| SP = SV * WA = 16.8W | 15 (Servo Power/total) |
| MCP = V * MCA = 1.65W | 16 (Micro Controller Power/pin) |
| BA = 3A | 17 (Buck Converter Amps) |
| BP = BA * SP + BA * MCV = 25.2W | 18 (Buck Converter Power/total) |
| Sum(Powers) = 193W | 19 (Total Power) |
| E = 11.1V *.85Ah = 9.435Wh | 20 (Battery Energy) |
| T = E÷Sum(Powers) = .0489 h= 2.93 minutes (Good) | 21 (Time of Operation Max) |

Now that the battery size has been resolved, we can focus on weight. Having two functionalities to our weapon means more weight space must be allocated to add these objects. With some initial estimates:

| | |
|---|---|
| WW = 138g | 1 (Wheel Motor Weight) |
| SW = 25g | 2 (Saw Weight) |
| Current Sensor = 5g | 3 (Current Sensor Weight) |
| BW = 80g | 4 (Battery Weight) |
| CV=LWH−(L−2t)(W−2t)(H−t) = 332 $cm^3$ | 5 (Chassis Volume (cm cubed) L = 20, W = 20 and H=12 t = .25) |
| BV = $\pi r^2 * h$ = 39.27 $cm^3$ | 6 (Blade Volume r = 5cm and h = .5cm) |
| CW = p(BV+CV) = 445.6g | 7 (Chassis Weight p = 1.2g/$cm^3$) |
| PW = 40g (approximation by online reference) | 8 (PCB Weight) |
| SRW = 66g | 11 (Servo weight) |
| RC = 907-(WW + SW + BW + CW + PW + SRW)= 107g | 10 (Room For Comfort 2lbs = 907g) |

With 107g of leeway for other components such as the soldered components or additional 3d printed material, our goal should be obtainable, but final product weight will be close. Ideally, we want the bot to be just under 2lbs to make it as difficult to lift as possible, but to not go over. Because the tolerance is low for weight, we have some backup plans in place where we can cut holes into the chassis retaining resiliency against our opponents' attacks while still having the chassis necessary to host our system and protect it. These approaches allow for increased tolerance for weight and allow our design to be more obtainable.

# 3. Design Verification

Reference Appendix A (R&V) for quick, organized summary of below information.

## 3.1 Physical Design

First steps of verification for physical design is going to be checking if our design first can fit. This entails just making sure motors, PCB, and all other components can fit in their allotted space within the design. While many of these pieces can be measured before printing, there still are variances depending on 3D printing output. Once this piece has been confirmed, we can move on to verifying the strength of chassis. The chassis should be able to withstand an impact from the opposing bot. If the opposing bot is 2lbs and accelerates in one second 3 meters then we can assume a force of around 205N. By replicating this we can model an object which matches 205N colliding. If the chassis survives, it passes verification, and if not, we make the chassis thicker. Another issue is getting flipped, the plan is to have the chassis nearly be in contact with the floor and have all components on the inside so it can't be flipped as easily.

## 3.2 Subsystems

### 3.2.1 Power Subsystem + Current Sensor

To verify the power subsystem, the approach is going to be very incremental. Before testing the system, each component will need to be tested, verifying with a multimeter accurate voltage outputs and current draws. For example, with the battery we expect 11.1V, ensure that it is read and guarantee the battery isn't faulty before using it. Then we move on the voltage regulators, ensuring that with 11.1V input the regulator can output the expected 3.3V, 5V, and 8.4V. Continue this testing approach (satisfying dependencies like a makefile), and eventually once the final target is satisfied in terms of voltage output, we have verified the output of our system is as expected.

After this, we set up breadboard circuits to test if the current sensor can cut power when the motors draw more than 4A of current (each motor rated for 2A). This can just be done with an arbitrary source on a breadboard and an identical circuit to the PCB. Then, we must verify the sustainability of our system by testing maximum power consumption, how long the battery can sustain the design at full operation mode (expected being at least 2 minutes).

| Component | Requirements | Verification | Verification Status |
|---|---|---|---|
| Battery/Current Sensor | - Needs to power the robot for 2 minutes<br>- Needs to provide at least 11 volts<br>- Needs to be protected from motor malfunction resulting in high current draw by the current sensor to prevent increased consumption<br>Protected from shorts via robust PCB connection | - Tolerance analysis ensures sizing is correct before testing<br>- Run robot subsystems at full capacity (maxed out saw, servo moving, and wheels moving) and ensure that that is able to run the full 2 minutes. Calculates for the worst case.<br>- Probe using voltmeter in breadboard that battery isn't faulty and provides advertised 11.1 volts. | Verified |

| | | | |
|---|---|---|---|
| | | Ensure current sensor cuts connection when current is raised over 50A (Approximately the amount of current draw that would make the bot die in 1 minute). | |
| Voltage Regulators | - Needs to be protected from motor malfunction resulting in high current draw by the current sensor to prevent damaging of transistors and subcomponents<br>- Protected from shorts via robust PCB connection<br>Needs to provide outputs 3.3, 5, and 8.4 volts from 11.1 volt input | Probe using voltmeter on a breadboard that when buck converter is given 11.1 volts it's 3 outputs are the voltages described in requirements. | Verified |

### 3.2.2 Controller Subsystem and User Interface

By connecting the power system to our microcontroller, we can begin to test our Bluetooth functionality. With given software we should be able to toggle GPIO pins between high and low, upon toggling systems should turn on an off. Assuming the other systems are verified, the only verification necessary here is to ensure the Bluetooth connection is correctly toggling the GPIO outputs between 3.3 volts and 0V with the user interface (the laptop keyboard). This can simply be tested with a multimeter. We must also verify that the microcontroller outputs the correct PWM signals to control the saw blade and servo. We can test this similarly using an oscilloscope.

| Component | Requirements | Verification | Verification Status |
|---|---|---|---|
| Controller GPIO | Controller GPIO needs to be able to output given Bluetooth output 3.3 volts and a voltage less than 2 volts | Use Voltmeter to measure pin voltage before connecting to rest of design. | Verified |

### 3.2.3 Drive Subsystem

Once the microcontroller is verified and the power system is verified, we should at this point have all the means to adjust the drive system. Adjusting the GPIO pins between high and low in different variations, we should be able to qualitatively see the robot moving forward, backwards, to the right, and to the left through. This is tested through GPIO pins being HIGH or LOW, toggling transistors in an H-bridge to allow for current flow through the wheel motors in each direction. The system can be verified when we observe that the wheels can individually accelerate forward and backward based on microcontroller output. Once working, try forcing the motors to be given more than 4A of current and ensure the current sensor cuts the battery to the system in that result.

| Component | Requirements | Verification | Verification Status |
|---|---|---|---|
| H-Bridge | - H-Bridge needs to be able to drive both motors in conjunction with the above components.<br>- H-Bridge needs to be able to provide max current draw of 2A to motors to ensure motors are not expected to be fried | - Connecting to ESP microcontroller from before (which at this point has been verified) drive the motor check following box for microcontroller requirements.<br>- Use voltmeter and ensure current draw no greater than 2A at max operating mode of 3.3volts at gate and 11.1volts into the motor itself | Verified |

### 3.2.4 Attack Subsystem

Once again, this piece is dependent on power system being verified and microcontroller. After these are verified connect to servo and motor. Qualitatively, evaluate that the saw can reach outside the plastic frame given the rod connected to the servo. Then because power system is verified servo and saw should receive necessary power at this point to operate, once again qualitatively evaluate the saw is able to lift and spin. Then on the quantitative said, ensure the range of motion extends the entire opening (around 11 cm as height is 12 cm and the slit is 1 cm which implies 11 cm). After that do more qualitative evaluation of saw rotation (rotates at a speed which can cause damage, difficult to measure exact RPM). Test if saw can lift 2lbs as well, since one aspect of our attack subsystem is to lift or at least make one object of 2lbs off balance without damaging the saw itself. This will be tested by turning off the saw, lowering the bot, moving the bot forward and then lifting the servo. Ideally this sequence of events will be capable of lifting the 2lbs to some extent.

| Component | Requirements | Verification | Verification Status |
|---|---|---|---|
| Saw/Servo | Must be able to fit outside chassis and move full 11 cm range of motion. | Test this by connecting saw to motor and to servo, when running servo, ensure with input to controller gpio we are able to move the 11cm given the mounting point in the chassis. | Verified |

### 3.2.5 Overall System

| Component | Requirements | Verification | Verification Status |
|---|---|---|---|
| System | - Ensure Robot when connected to PCB and not breadboard exhibits all requirements mentioned above and fully autonomous. | - Practice results of 2lb force listed above crashing into bot while moving<br>- Practice moving the saw and trying to damage something. Try hitting the daw with 2lb of force (exact newtons listed above) to see if stays operational after words. | Not Verified |

# 4. Costs

## 4.1 Parts

**Table X Parts Costs**

| Part | Manufacturer | Retail Cost ($) | Bulk Purchase Cost ($) | Actual Cost ($) |
|---|---|---|---|---|
| The Yeah Racing XT30 Connectors | Hobby Town | $3.49 | $3.49 | $3.49 |
| ESP32-Wroom | Espressif | $5.28 | $5.28 | $5.28 |
| Lite Flite Wheels | Dave Brown Products | $9.59 | $9.59 | $9.59 |
| 11.1 Volt Lipo Battery | RCBattery | $14.99 | $14.99 | $14.99 |
| HPS-2018 High Torque Servo | Hiwonder | $14.99 | $14.99 | $14.99 |
| Brushless Motor | EMAX USA | $26.99 | $26.99 | $26.99 |
| INA237 Current Sensor | Adafruit Industries | $9.95 | $9.95 | $9.95 |
| Mini 3-12VDC Gear Motor | Aretronics | $6.75 | $12.75 | $12.75 |
| **Total** | | **$92.03** | **$98.78** | **$98.78** |

## 4.2 Labor

We each spend roughly 8 hours a week for this class. Looking at a recent Illini success report, the average ECE major earns roughly $100,000 salary per year. This correlates to about$50 an hour. Labor = $50 * 2.5 * 8 * 16 = $16,000 For our 3 person group this value reaches $48,000.

# 5. Conclusion

## 5.1 Uncertainties

Most uncertainties by this point have been resolved, but we are yet to receive our 3rd iteration of our PCB and therefore may encounter more upon that iteration received. Work has been done to test and verify the above circuitry on breadboards, but there could be schematic issues when translating for our PCB order so overall an extreme uncertainty. Also, as the PCB won't be received until the 4th one is ordered, we may be locked into that certain PCB design and must compensate in other ways.

## 5.2 Ethical considerations

### 5.2.1 IEEE Code of Ethics – Safety and Well-Being

Following the IEE Code of ethics, our team will ensure that we place an emphasis on the health and safety of all our team members, competition officials, and spectators throughout the entire process. The assembly and testing of our bot will take place in extremely controlled environments, such as an enclosed lab or an official competition arena. Since our project contains moving mechanical parts as well as a spinning weapon, we want to have strict safety measures, so there are no accidental injuries. Our robot will not be energized unless it is secured on a safe surface and under proper supervision. We will utilize a physical guard attached to the blade during testing if we determine it is sharp enough to cause harm, as well as a kill-switch will be installed to disconnect the battery in case of emergency. Another priority will be our battery. We want to only use LiPo batteries that are properly rated for our current draw. As well as storing them in fire-resistant containers when we are not using them. The charging will be done using lab-approved power supplies under our supervision. These practices will help ensure the well-being of our team, and the surrounding ones too.

### 5.2.2 IEEE Code of Ethics – Security and Responsible Use

Our robot is controlled using Bluetooth between our laptop interface and the ESP32 microcontroller. Wireless communication allows for the risk of unauthorized control or interference. To comply with the standards of responsible technology use from IEEE, we will securely pair devices using unique identifiers and authentication protocols. We will avoid unencrypted and open pairing mode. The software that we write will have security in mind. The ESP32 will ignore commands from unrecognized addresses, and the connection status will be monitored on our laptop interface. During the competition, connections will be tested to prevent unintended cross-communication with other robots. Prioritizing data integrity and responsible use will mitigate the chance of interference that could cause harm to people or equipment.

### 5.2.3 IEEE Code of Ethics – Fairness and Compliance

The ACM code puts an emphasis on adherence to rules and fairness. We will build our bot to fully comply with the regulations of the competition. We will ensure there are no hidden features or modifications that give us an unfair advantage in the competition. The materials we choose to use will be properly documented and transparent to the organizers. We are big fans of supporting integrity and fair competition.

### 5.2.4 Anticipated Safety Concerns and Mitigation Procedures

Our project has several safety concerns that we will carefully address to ensure we operate our bot safely during development, testing, and in competition. The first area involves battery hazards. Overcharging, short-circuiting, or potentially puncturing a LiPo battery can lead to harm through a fire or overheating. To prevent this, we will use an overcharge protector and store batteries in fireproof safe bags when not in use. There will be a voltage monitor that will automatically cut off the power if we are operating below safe operating limits. In addition, we will inspect batteries before and after use to check for swelling or damage which are a sign of potentially unsafe conditions.

Another concern we have are mechanical hazards. Given the rotating saw blade and lifting mechanism, there is always the potential for error. During assembly, transportation, or testing, we want to mitigate all risks. To achieve this, we will have a removable guard over the blade when it is not in competition. During operation, we will maintain a safe distance around the bot and always have a hardware and software kill switch for any emergency stops. The blade will only be activated inside any approved test or competition areas. Finally, the batteries will be disconnected with the blade removed when we are transporting the bot.

Another hazard we must properly deal with are electrical issues. All exposed wires we have will be insulated to prevent accidental contact or disconnection. We will test all solder joints for continuity before powering up. To prevent overvoltage damages, we will ensure we properly supply our components with the correct voltage amounts.

We also recognize the potential for wireless interference. Our bot relies on Bluetooth communication, so this will be a major concern for the safety and simply the function of our bot. Another nearby device could attempt to pair or interfere with our connection. To avoid this, we will use fixed pairing modes along with unique device identifiers. We will conduct all testing on dedicated communication channels.

# References


[1] Institute of Electrical and Electronics Engineers. "P7.8: IEEE Code of Ethics." *IEEE*, https://www.ieee.org/about/corporate/governance/p7-8.html. Accessed 13 Oct. 2025.

[2] Association for Computing Machinery. *ACM Code of Ethics and Professional Conduct*. Association for Computing Machinery, 2018, https://www.acm.org/code-of-ethics

[3] Leiton GmbH. "PCB Weight Calculation." *Leiton Tools*, Leiton GmbH, https://www.leiton.de/leiton-tools-weight-calculation.html?Projectname=&LayerCount=2&Quantity=100&SizeX=100&SizeY=100&CopperThickness=35&CopperCapacityUtilization=75&CarrierWidth%5BFR4%5D=1%2C50&CarrierWidth%5BAlu%5D=0&ShippingCosts=3%2C00&PCBThickness=1%2C57&weight=3%2C29&Area=1&WeightPerPiece=39%2C17&Volume=1%2C57&ShippingPrice=12%2C34#WeightCalculation

[4] Unionfab. "PLA Density: A Comprehensive Guide." *Unionfab Blog*, 23 Apr. 2024, https://www.unionfab.com/blog/2024/04/pla-density