Antweight Battlebot Design Document

Team 37
Deepika Agrawal (deepika7)
Megha Esturi (mesturi2)
Ishanvi Lakhani (ishanvi2)

Fall 2024
TA: Surya Vasanth

Contents

1. Introduction

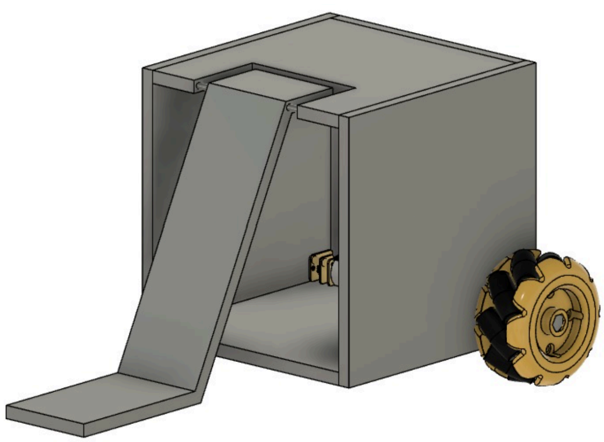
1.1 Problem

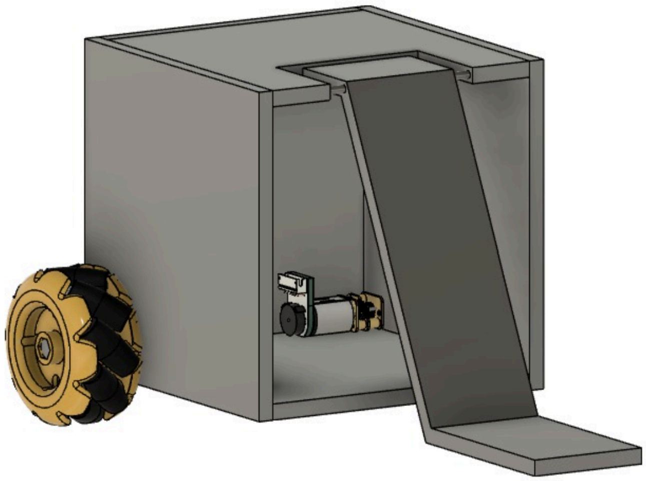
The objective of this project is to design and develop a battlebot that can be remotely controlled from a PC and is capable of competing against other battlebots in a confined arena. The key goal is to build a battlebot that meets competition guidelines, is agile, durable, and equipped with offensive and defensive mechanisms to outperform opponents. Battlebot design must adhere to strict guidelines for agility, durability, and combat functionality. The chassis and weapon must be 3D printed from PET, PETG, ABS, PLA, or PLA+, while other components like motors, electronics, and fasteners can be made from any material but must not enhance the bot's structural integrity or armor. Weighing no more than 2 pounds, the bot will use wheels for continuous movement. Flying, jumping, and hovering are prohibited, and control will be managed via a custom-designed PCB and Wi-Fi. The connection must avoid interference with other robots, and the bot will shut down if Wi-Fi is lost. Powered by 4 AAA batteries, the battery terminals must be safeguarded against shorts, with external lights indicating main power and Wi-Fi status.

1.2 Solution

The battlebot will feature a flipping tool at the front which adds both offensive and defensive capabilities, allowing the bot to flip opponents when they come too close. All functions, including movement via wheels and flipping, will be controlled through PC input for precision. Wi-Fi will enable wireless communication between the bot and the PC, allowing remote control. Designed for agility, the bot will move quickly and strategically to avoid attacks and position itself for offensive maneuvers in order to disable the opponent's battle bot .

1.3 Visual Aid



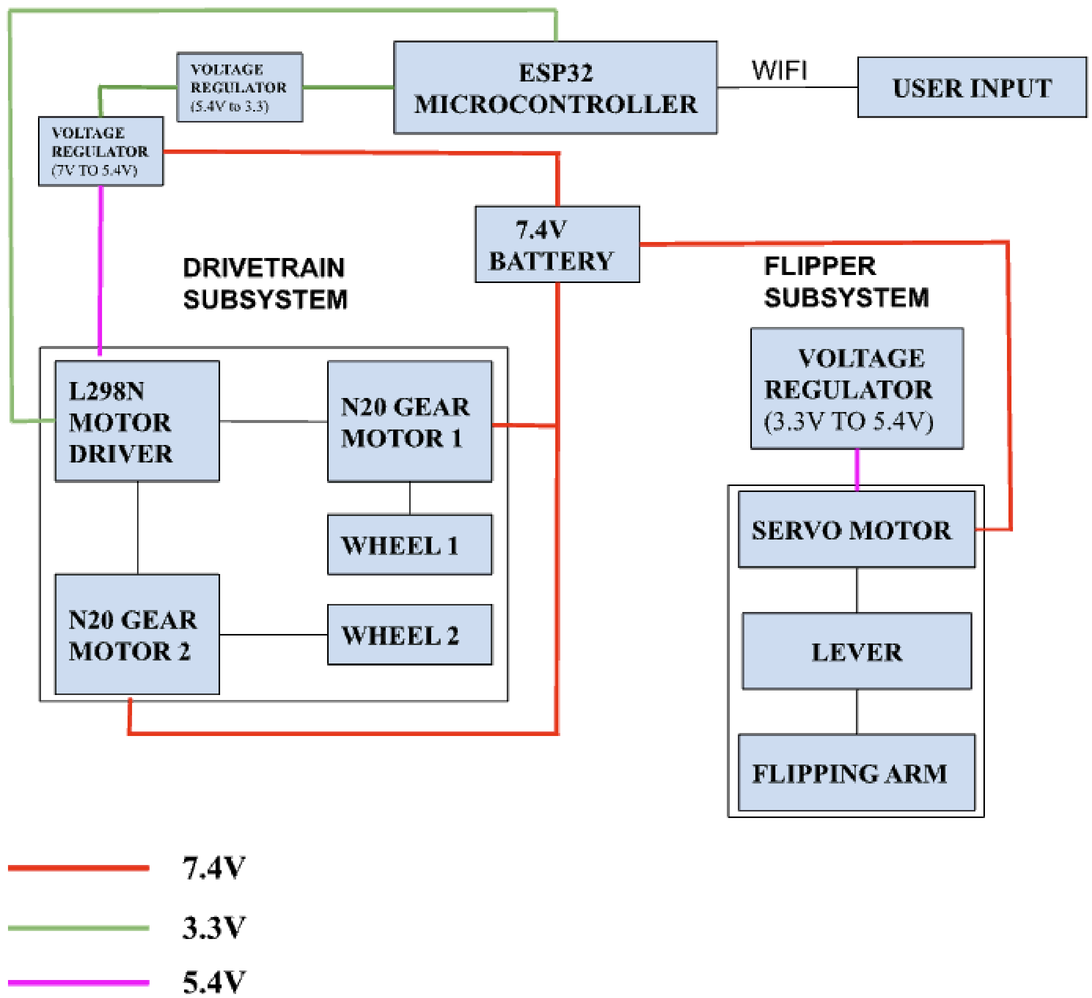


1.4 High-Level Requirements List

1. The total weight of the battle bot, including components such as the chassis, electronics, and weaponry, must not exceed 2 lbs.
2. The battle bot must operate efficiently within a voltage range of 6 volts to ensure consistent mobility and weapon operation throughout the match.
3. The Wi-Fi communication range indoors should be between 30 to 100 meters with the battle bot's response time being 50 to 100 milliseconds.
4. The battle bot shall accelerate at a rate of 6.45 m/s^2, assuming the bot's weight is 2 lbs, the wheel diameter is 48 mm, and the torque is 20 oz-in.

2. Design

2.1 Block Diagram



2.2 Subsystem Overview

2.2.1 Drive Train Subsystem

The maneuverability subsystem controls the movement of the battle bot, consisting of two wheels, each driven by an N20 Micro Gear Motor. These motors are connected to L298N dual-channel motor drivers, which manage direction. The ESP32 microcontroller will output control signals at 3.3V, which will be stepped up to 6V using voltage boosters. These 6V signals will then be input into the L298N motor driver, which communicates with the motors to move the wheels accordingly. This setup enables the bot to move forward, backward, and turn during battles. The motors, operating at 3V to 12V, with an ideal running voltage of 6V, provide RPMs between 100 and 460. The motor drivers, powered by 6V, will control the motors based on the boosted signals from the ESP32. Both motors and motor drivers are connected in parallel to ensure each receives its own power supply. Direction is controlled through user input, which will adjust the signals and change the bot's movement.

Requirements	Verification
The motors and motor drivers must effectively communicate with the microcontroller to relay user commands.	Perform a ping test to measure Wi-Fi latency using the ESP32. Print latency on the connected LED/screen and check if it is between 50 and 100 ms. Preferable latency between 35 ms - 60 ms but up to 10 ms acceptable.
The motors and motor drivers must operate efficiently with a 6V and 100-200 mA supply when connected in parallel.	Use a multimeter to measure the voltage and current supplied to the motors. Verify that the voltage is around 6V and current is within the 100-200 mA range during operation.
The wheels must enable the robot to move at a sufficient speed, achieving an RPM of approximately 100 - 460 .	Manually measure RPM by counting the number of wheel rotations in a fixed period of 10 seconds and multiply by 6 to get the RPM. Ensure the measured RPM falls between 100 - 460 for effective mobility.

2.2.2 Flipping Subsystem

We are using a 180-degree servo motor to control the flipping mechanism of the battle bot's arm. This servo motor is designed to rotate up to 180 degrees, providing precise control over the arm's movement. The motor is connected to a lever arm mechanism. When the user inputs a command through the PC, it is transmitted to the ESP32 microcontroller, which sends a PWM signal to the servo motor.

As the servo rotates 180 degrees to the right, it pulls down the lever arm, which, in turn, lifts the flipping arm into the air. This movement allows the arm to execute a flipping action against the opponent robot. Once the flip is complete, the servo can rotate in the opposite direction, pulling the lever back to its original position. This resets the arm, making it ready for another flip. The motor operates at a voltage of 6V, ensuring that it delivers adequate torque to control the arm's movement efficiently.

Requirements	Verification
Motor Rotation Control: The motor must rotate exactly 180° to compress the spring and stop at the correct position.	Program the ESP-32 to send a PWM signal that commands the motor to rotate 180°. Use an encoder or visual markers to verify that the motor stops after completing one full rotation.

Flipping Force: The arm and spring mechanism should generate enough force to flip the opponent robot.	Place a weight equivalent to the opponent robot on the flipping mechanism and trigger the flip. Verify the mechanism generates sufficient force to flip the weight.

2.2.4 Power Subsystem

The power subsystem will consist of four 1.5V batteries connected in series, providing 6V. This 6V will be stepped down to 3.3V using an external voltage regulator to ensure the ESP32 receives the correct voltage. The ESP32 will output 3.3V control signals, which will then be stepped up to either 5V or 6V, depending on the requirements of the specific components in the system. The total current draw is less than 5000mA, and with a 1200mAh battery, the system is expected to run for about 5 - 8 minutes, depending on usage intensity.

Requirements	Verification
The subsystem must include a 6V battery with at least 1200mAh capacity to power the bot for the duration of the battle.	Measure the battery capacity using a battery tester. Simulate a typical battle load and measure power consumption during a test run and monitor runtime.
The battery must supply at least 1200mAH of current for all subsystems to work together.	Use a multimeter or current meter to measure the continuous current supplied by the voltage regulator while the bot is in operation under full load.
The subsystem must include a kill switch that can be triggered by the PC to immediately cut off power in case of hazard within 1 second.	Perform a live test where the kill switch is manually activated through the PC interface during bot operation. Check if the power is disconnected.

2.2.5 Control Subsystem

The microcontroller we have chosen is the ESP32-WROOM-32 with an integrated 2.4 GHz wi-fi module. Because the wifi is integrated and there is an integrated antennae, we will not need a UART or SPI. Additionally, this microcontroller has low power consumption and allows for multiple tasks to happen simultaneously which we would need for the flipper and wheel control. The microcontroller will be connected to a PC where the commands will be sent to the microcontroller. Then the microcontroller will send commands to the other subsystems.

Requirements	Verification
The ESP32 must support WiFi communication	Use a WiFi benchmark tool or test sketch to

with a throughput of at least 5 Mbps for reliable control of motors and weapons.	send/receive data over WiF and measure the throughput and verify it reaches at least **5 Mbps** consistently during operation.
The latency should be between 50-100 ms for responsive control, with a preferred range of 35-60 ms.	Perform a ping test using the ESP32 WiFi module. Print latency to a connected screen or LED, and log the average latency during operation.
Must provide 6 GPIO pins, including 2 PWM pins for speed control of motors.	Check ESP32 pinout diagram and map at least 6 GPIO pins.

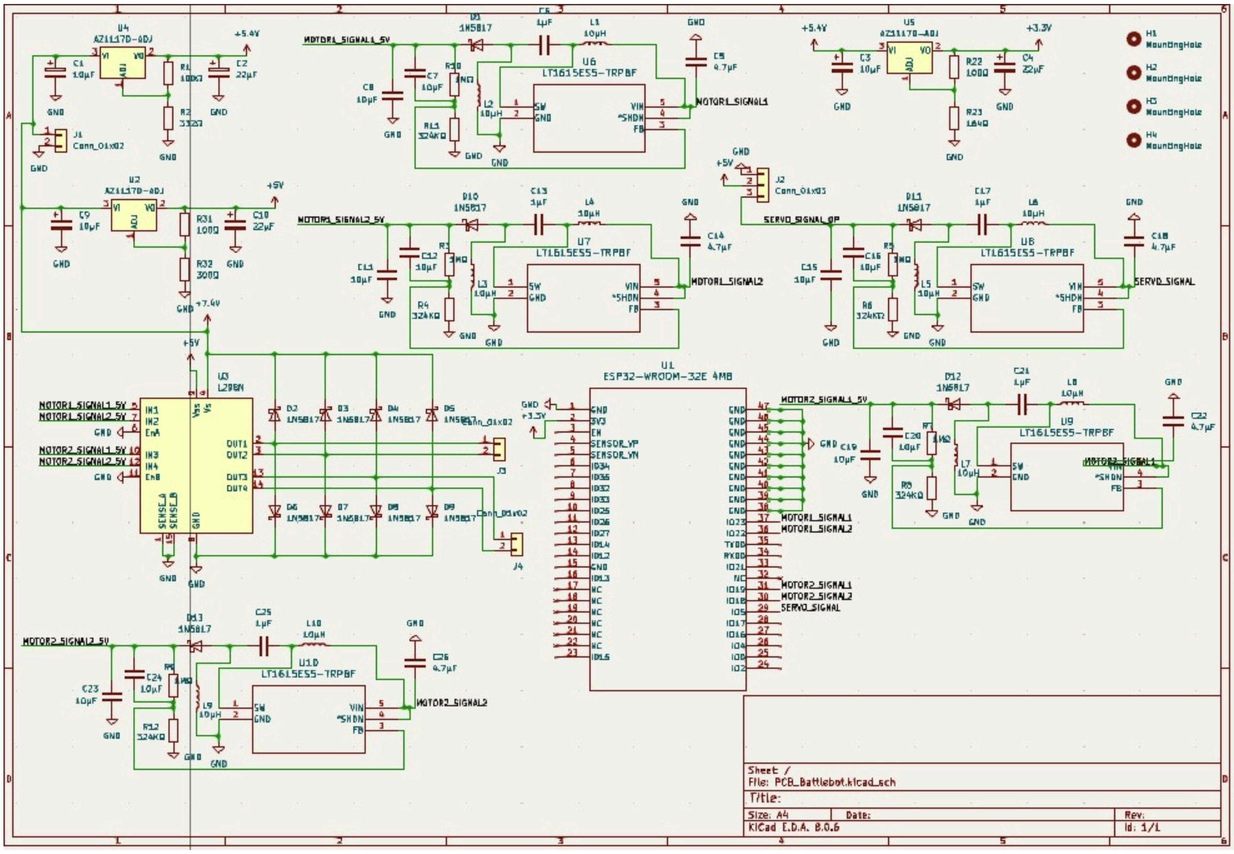
2.2.6 Kill Switch Subsystem

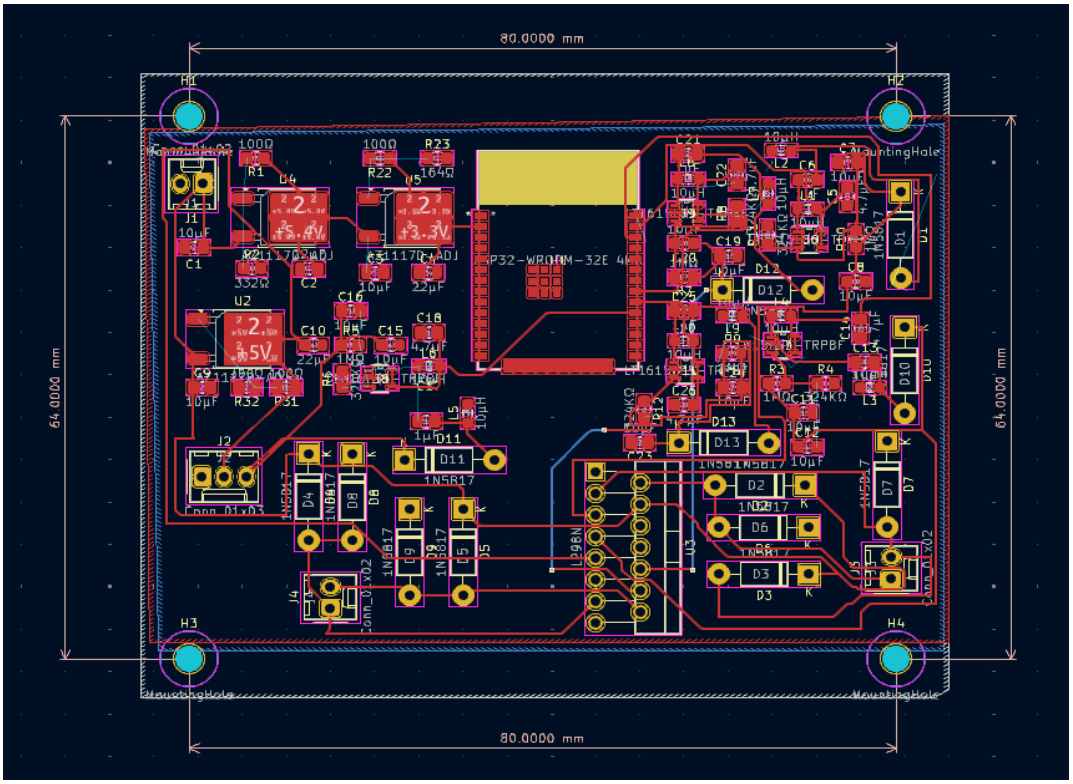
The battlebot includes a kill switch as a critical safety feature, designed to immediately cut off power and halt all movement in case of emergencies or malfunctions. This switch can be manually activated, allowing quick shutdowns when needed. Additionally, the battlebot features an LED indicator that monitors the Wi-Fi connection status. As long as the connection between the bot and the PC remains active, the LED will stay lit. If the connection is lost, the LED turns off, signaling communication failure. When this happens, the microcontroller automatically triggers the kill switch, cutting off power to the entire system to prevent any unintended movements or dangerous actions. This automatic shutdown mechanism ensures that the battlebot is completely disabled when communication is lost, offering an extra layer of protection against malfunctioning or unsafe situations.

Requirements	Verification
The LED indicator must display the Wi-Fi connection status, remaining lit when the connection is active and turning off when the connection is lost.	Test the Wi-Fi connection by turning it off and on during bot operation. Verify that the LED correctly reflects the status—remaining lit when connected and turning off when the Wi-Fi connection is lost.
The microcontroller must automatically trigger the kill switch when Wi-Fi connectivity is lost, cutting power to the entire system.	Simulate a loss of Wi-Fi connection while the bot is operational. Verify that the LED turns off, and the kill switch is automatically triggered by the microcontroller, completely disabling the system and stopping all movement and functions.
The kill switch mechanism must prevent unintended actions after being triggered, ensuring that no power is supplied to the motors or any part of the bot.	After the kill switch is activated, verify that all components remain powered off, and the bot does not move or perform any functions until it is manually reset, ensuring safety and preventing unintended behavior.

2.2.7 PCB Schematic and Routing

Combining all the subsystems above, this is what the schematic looks like, and the PCB routing.





2.3 Tolerance Analysis

In this analysis, we will explore the thermal performance of the key components of the battle, and the possibility of overheating during operation. We will analyze the power dissipation and the heat tolerance and check if we need to worry about cooling any parts or adding heat sinks.

Key Components

1. N20 Gear Motor (2 units)
2. L298N Dual H-Bridge Motor Driver
3. ESP32-WROOM-32E Microcontroller
4. MG996R Servo Motor
5. Voltage Regulator (step down) (AZ1117CD-3.3TRG1)
6. Voltage Regulator (step up) (LM2623MM/NOPB)

First we can start off by calculating the power dissipation by using the formula:

P_dissipated = V x I
P_dissipated → Power in Watts
V → Voltage across components in Volts
I → Current across components in Amps

COMPONENT	VOLTAGE	CURRENT(MAX)	P_dissipated
N20 Gear Motor	6V	2.4A	14.4W (both wheels)
L298N Dual H-Bridge Motor Driver	6V	1.5A	7.5W
ESP32-WROOM-32E Microcontroller	3.3V	0.5A	1.65W
MG996R Servo Motor	6V	5.48A	32.88W
Voltage Regulator (step down) (AZ1117CD-3.3TRG1)	6V	0.011A	0.066W
Voltage Regulator (step up) (LM2623MM/NOPB)	3.3V	0.00011A	0.000363W

The total heat dissipation would be the sum of all the powers:

P_total = 14.4 + 7.5 + 1.65 + 32.88 + 0.066 + 0.000363 = 56.536W

Now we can calculate the temperature rise with the formula:
$\Delta T = P\_dissipated/hA$

An assumption is that the heat transfer coefficient is 10 $W/m^2 \cdot K$. Some rough calculations for the surface area of each component:

N20 Gear Motor (each): 4574.1589 mm^2 = 0.0045741589 m^2 (per unit)
L298N Driver (each): 3870 mm^2 = 0.00387 m^2
ESP32: 1175.04 mm^2 = 0.00117504 m^2
MG996R Servo Motor: 5285.96 mm^2 = 0.00528596 m^2
Voltage Regulator (step down): 9 mm^2 = 9e-6 m^2
Voltage Regulator (step up): 24.79 mm^2 = 2.479e-5 m^2

Now for the Temperature Rise of each Component:

N20 Gear Motor:
Pavg_dissipation = 14.4W/2 = 7.2W

ΔT = (7.2)/(10*0.0045741589) = 157.56 °C

L298N Driver:
P_dissipation = 7.5W
ΔT = (7.5)/(10*0.00387) = 193.38 °C

ESP32:
Pavg_dissipation = 1.65W
ΔT = (1.65)/(10*0.00117504) = 104.25 °C

MG996R Servo Motor:
P_dissipation = 16.5W
ΔT = (16.5)/(10*0.00528596) = 311.39 °C

Voltage Regulator (step down):
P_dissipation = 0.066W
ΔT = (1.098)/(10*9e-6) = 73.33 °C

Voltage Regulator (step up):
P_dissipation = 0.000363W
ΔT = (1.098)/(10*2.479e-5) = 1.46 °C

Summary:

COMPONENT	POWER DISSIPATION (W)	CALCULATED TEMPERATURE RISE (Celcius)	MAX OPERATING TEMP (Celcius)	ACTION
N20 Gear Motors (2 units)	14.4	157.56	60	Cooling Needed
L298N Dual H-Bridge Drivers (2 units)	7.5	193.38	130	Cooling Needed
ESP32-WROOM-32E	1.65	140.51	85	Cooling Needed
MG996R Servo Motor	16.5	311.39	55	Immediate attention needed, overheating issues, might need to check math or find a different motor
Voltage Regulator	0.066	73.33	150	Monitor, but

(step down)				mostly safe
Voltage Regulator (step up)	0.000363	1.46	150	Safe

Looking at these calculations, it seems like there are a lot of overheating issues for many parts. Therefore, we will be adding in heat sinks, airflow management, and possibly consider adding a fan to some of the parts.

3. Cost and Schedule

3.1 Cost Analysis

Component	**Manufacturer**	**Quantity**	**Cost**	**Link**
N20 Gear Motors	servocity	1	$12.99	https://shorturl.at/TmENn
L298N Dual Channel Motor Driver	BOJACK	1	$10.89	https://shorturl.at/FukKl
Mecanum Wheels	SALUTUYA	1	$11.69	https://shorturl.at/FXVYd
Servo Motor	Readytosky	1	$23.88	https://shorturl.at/Lmm7r
Battery	Tenergy	1	$10.79	https://shorturl.at/TymXq
Spring	Uxcell	1	$7.95	https://shorturl.at/N4Gd1
RS-380 Motor DC	Wicocc	1	$17.43	https://shorturl.at/WeK50
ESP32 1965-ESP32-DEVKITC-32E-ND	Espressif Systems	1	$10.50	https://shorturl.at/Ajn0p

3D printing costs	Siebel Center for Design	1.5lbs	$25	

SUM OF ALL COSTS = $131.41

We estimate needing no more than 2-3 hours of assistance from the machine shop. We don't require them to build anything for us since most of our components are being ordered and the rest will be 3D printed. However, we may need their guidance on how to implement the spring flipping mechanism, which should take about 2-3 hours from start to finish.

We anticipate working around 10 hours per week over the course of 5 weeks to complete the battle bot. Given that this project involves expertise in electrical engineering, mechanical engineering, physics, and math, we believe a fair hourly rate is $21. Based on this, each team member would earn a total of 10 hours per week * $21 per hour * 5 weeks = $1,050 by the end of the 5 weeks.

3.2 Schedule

WEEK	TASK AND PERSON
October 7th - October 12th	Design PCB (**Everyone**) Design Review (**Everyone**) Revise Design Document if needed based on feedback (**Everyone**) Review CAD design made by Deepika (**Megha, Ishanvi)** Order all parts (**Ishanvi**)
October 12th - October 16th	Update PCB design based on feedback (**Everyone**) Finalize CAD design (**Everyone**) 3D print chassis (**Deepika**) 3D print axle etc (**Ishanvi & Megha**) Start setting up microcontroller setup (**Megha**) Understand how to connect microcontrollers to motor drivers and motors (**Ishanvi**)
October 18th - October 25th	Update based on feedback and order PCB (**Everyone**) Ensure 3D chassis and all printed parts are stable (**Everyone**) Aim to finish connections with the wheel's and motors and motor drivers (**Ishanvi**) Aim to finish connections with the vertical spinners and motors and motor drivers (**Megha**)

	Aim to finish connections with the vertical spinners and motors and motor drivers (**Megha**) Use verification table to test these parts (**Deepika**) Use voltmeter to test voltage consumption (**Deepika**)
October 28th - 1st November	Update based on feedback and order PCB (**Everyone**) Start working on motors for the flipper arm and ensure it is able to lift 2lbs. (**Ishanvi**) Figure out spring dynamics (**Megha**) integrate with chassis and test through verification table (**Deepika**)
1st November - 9th November	Finalize PCB based on feedback and order PCB (**Everyone**) Test all components work together (**Everyone**)
9th November - 18th November	Extra time - incase we need it to work on any of the parts

4. Ethics and Safety

4.1 Safety

The primary safety concerns we foresee involve the physical safety of individuals and the environment if the robotic car were to malfunction or become uncontrollable. Our defense mechanisms – the spinning blades and the flipping mechanism – can cause injury, so we will ensure that we conduct testing of the robot in a space where nothing is in harm's way. We also have a kill-switch in case of emergencies, that will immediately turn off power to our bot. Our high-speed motors and LiPo batteries also pose a safety risk, for which we will limit motor speed and torque to safe levels, as well as keep electric components enclosed and follow the battery safety guidelines provided for us.

4.2 Ethics

There are a few ethical considerations we have to keep in mind throughout this project. As stated in the IEEE Code of Ethics, there are several areas of consideration when it comes to lab ethics. These include safety, conflict avoidance, honesty, respect, privacy, and support. Our main goal is prioritizing the safety and welfare for all participants and will comply with safety standards to minimize risks. Further, we will use sustainable materials in our design, and disclose any potential risks while building our battlebot. We will treat all team members and competitors with respect, avoiding discrimination, harassment, and injury.

We will also ensure we have the necessary skills and seek help when needed, and make sure that all of our work is our own, and that we are not unfairly plagiarizing others. Finally, our project falls under the IEEE Code of Ethics 1.2 as we are creating a project that integrates technologies that we can demo and compete with [1].

5. References

Include all the websites we used for reference

[1] IEEE. (2016) IEEE Code of Ethics. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html

[2] "A 1 Lb. Spring Powered Flipper: The Complete Journey." *Conn Bots*, connbots.weebly.com/blog/a-1-lb-spring-powered-flipper-the-complete-journey.