Project Proposal

Introduction

Project Title: Fun Name for Robot
Team Number: 37
Team Members: Megha Esturi, Ishanvi Lakhani, Deepika Agrawal

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 follow strict material and mobility specifications like the following:

- Only the following plastic materials are allowed for the chassis and fighting tool: PET, PETG, ABS, PLA, or PLA+.
- Chassis and fighting tools must be 3D printed.
- Motors, electronics, axles, fasteners, and adhesives can be made from any material.
- These components cannot be used to enhance the robot's structural integrity, armor, or fighting tool.
- Maximum total weight of the robot is limited to 2 pounds.
- Approved locomotion for this design includes rolling mechanisms, specifically wheels, which will provide continuous motion driven by motors.
- Methods such as flying, jumping, or hovering are strictly prohibited.
- The battlebot will be controlled wirelessly via Wi-Fi.
- A custom-designed PCB will house the microcontroller that facilitates communication between the robot and a PC.
- The Wi-Fi connection must be programmed to prevent interference with other participants' robots.
- In case of lost Wi-Fi connectivity, the robot will automatically enter shutdown mode, stopping all movement and deactivating the fighting tool.
- The battlebot will use Li-po batteries for power.
- Batteries must be sealed and non-spillable, with a maximum voltage of 16V.
- Battery terminals must be protected from shorts to prevent fires.
- The robot must have an external light indicating when the main power is on and a secondary light to signal an active Wi-Fi connection.

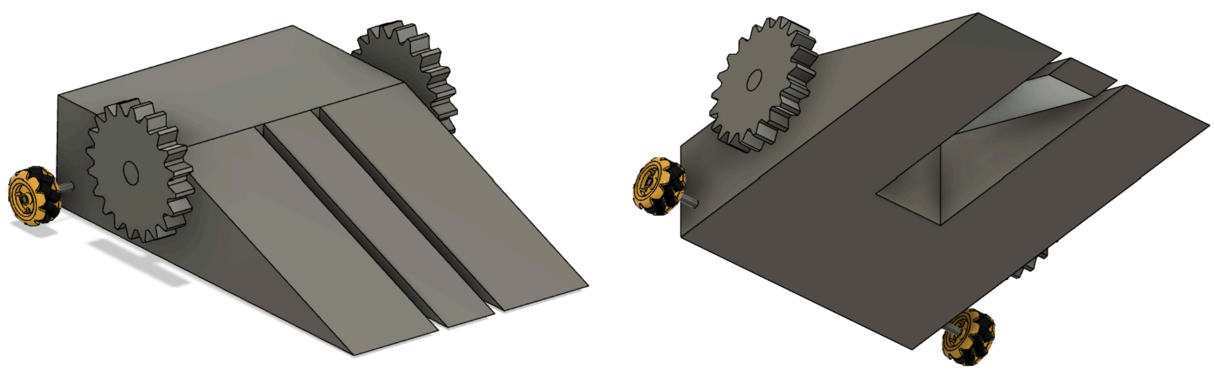
SOLUTION :

To solve this problem:

- The battlebot will feature two vertically rotating circular blades, one on each side. These blades will serve as the primary offensive tool to damage or destabilize opponents.
- The rotating blades will function as a high-speed cutting or destabilizing mechanism, making it difficult for other bots to approach without being damaged.
- A flipping tool is incorporated at the front of the bot. This tool is designed to flip opposing battlebots if they get too close, providing both offensive and defensive capabilities
- The movement of the battlebot (wheels), spinning of the blades, and flipping of the slate will all be controlled via PC input, ensuring precision control over every aspect of the bot's functionality.
- Wi-Fi will be used as the communication protocol between the bot and the PC, allowing the user to control the battlebot remotely and wirelessly.
- The battlebot will use wheels for movement, ensuring controlled mobility and the ability to maneuver around opponents in the arena.
- The bot will be designed for agility, enabling quick movements to avoid attacks and position itself strategically for offensive actions.

VISUAL AID :

The following is a side view of what our battlebot is going to look like.

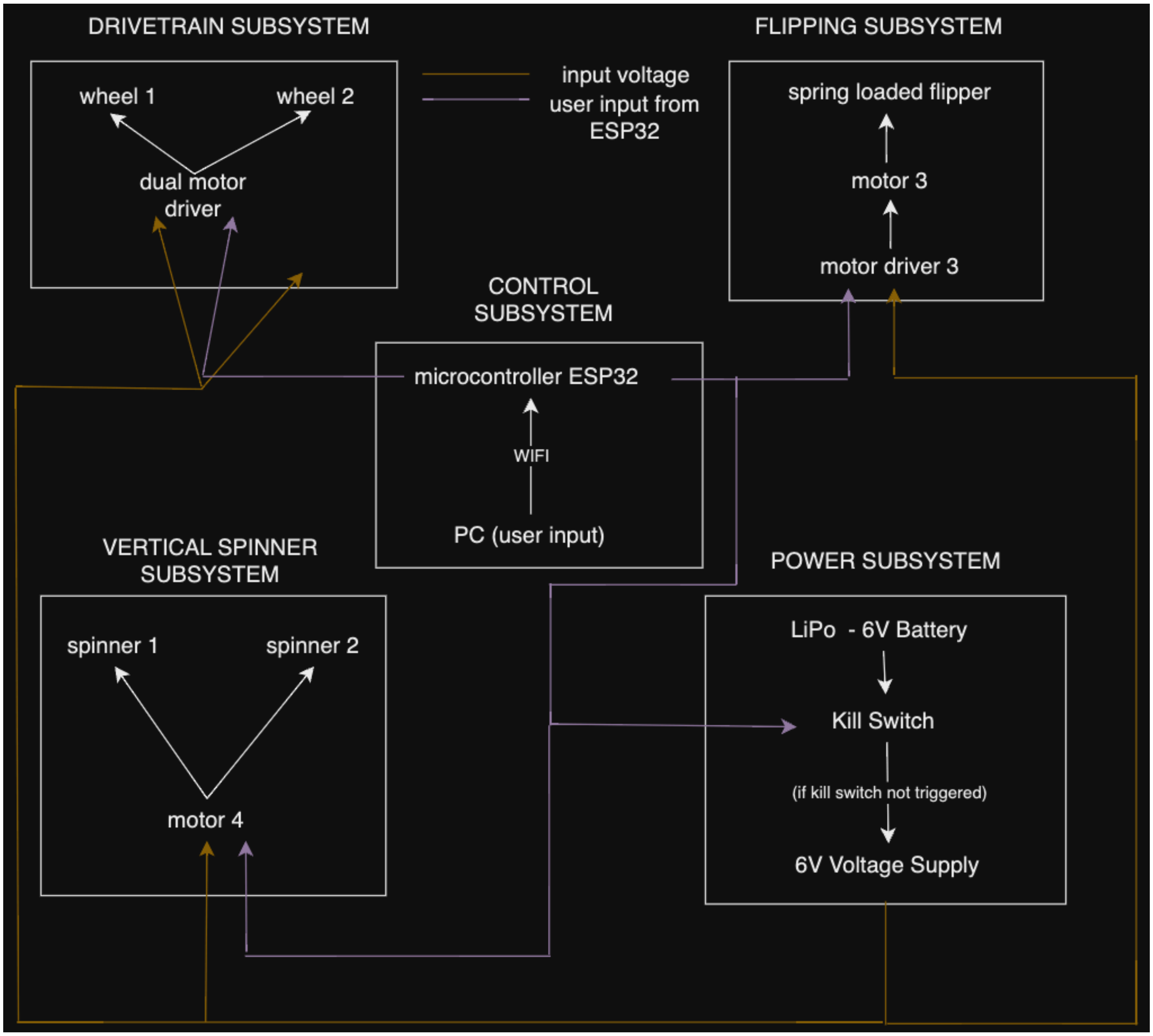


High-Level Requirements List

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 12 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.

Design

1. Block Diagram

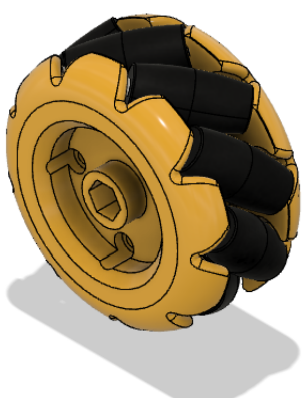


2. Subsystem Overview

DRIVETRAIN 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 speed and direction based on signals from the microcontroller. This setup enables the bot to move forward, backward, and turn during battles. The motors operate at 3V to 12V, with an ideal running voltage of 6V, providing RPMs between 30 and 1000. The motor drivers, with an operating voltage of 5V to 35V, will also be powered by 6V, regulated from a 12V source. Both motors and motor drivers are connected in parallel to ensure each receives its own power supply. Speed and direction are controlled through pulse width modulation (PWM), utilizing the built-in PWM capabilities of the ESP32 microcontroller, which allows precise control through user inputs.

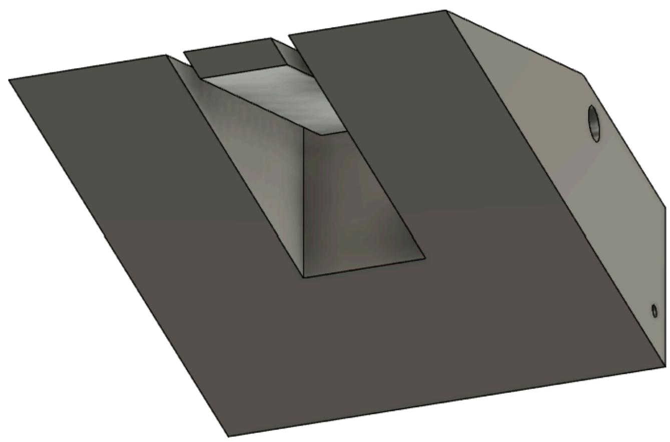
- Requirement 1 : The motor and motor drivers wheels must be able to communicate with the microcontroller effectively and relay the instructions given by the user.
- Requirement 2 : The motor and motor drivers should be able to work effectively with a supply of 6V and 100 - 200 mA when they are connected in parallel.
- Requirement 3 : The wheels should be able to allow the robot to move fast enough, i.e. rpm must be high approx. 150 - 300.



FLIPPING SYSTEM

We will have an arm that sits on top of the bot, and can flip upwards to flip the opponent robot. We will be using a [Digital 30KG Servo 360 Degree High Torque Metal Gear Servo Motor] that can stop precisely after 360 degrees, typically operates at a voltage range of 4.8V to 7.2V, and weighs approximately 112.8311 grams (0.24875 lbs) . The motor can be controlled directly with a PWM signal from a microcontroller (the ESP-32 we plan to implement has one), which allows precise control over when and how much the motor rotates. We can program the servo to rotate the required 360°, compressing the spring, and then stop. When the flip is triggered, the spring would release, performing the flipping action. The spring we intend to use is a compression spring, specifically the [uxcell Steel Coil Extended Compressed Spring](). Once the action is complete, the motor could reset to its original position and be ready for the next trigger.

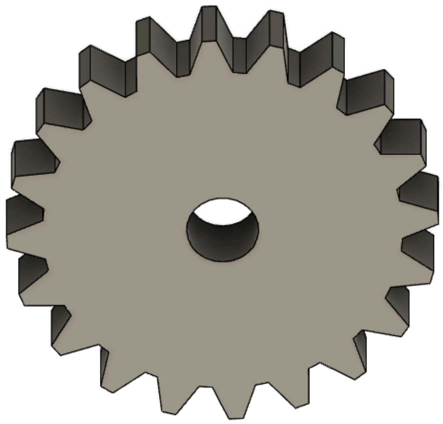
- Requirement 1 : The motor must rotate exactly 360° to compress the spring and stop at the correct position.
- Requirement 2 : The spring should release upon reaching full compression to flip the opponent robot.
- Requirement 3 : The arm and spring mechanism should generate enough force to flip the opponent robot.



VERTICAL SPINNER SUBSYSTEM

This subsystem consists of two rotating blades made from PLA+ plastic, positioned on opposite sides of the bot. The blades are connected by an axle and powered by a single motor, which is controlled by a motor driver. This system serves as the primary offensive tool, using rotational force to damage or destabilize opponents. The speed and operation of the spinners are controlled by the microcontroller, allowing the user to engage or disengage them strategically during the battle.

- Requirement 1 : The motor and motor driver must be able to communicate with the microcontroller effectively with a maximum of a few ms of delay.
- Requirement 2 : The motor and motor drivers should be able to work effectively with a supply of 6V and 300mA to 1A when they are connected in parallel.
- Requirement 3 : The vertical spinners should move fast enough so the defense mechanism is strong, i.e. rpm must be high approx. 1500 - 2000.



POWER SUBSYSTEM

The power subsystem will consist of a 6V NiMh battery. The ESP32 has an inbuilt voltage regulator, so it will get fed 6V and will drop down to 3.3V internally. The other subsystems will be powered by the 6V battery directly because they can handle it. The total current that is drawn is less than 5000mA, so we are using a 2000mAh battery which lasts almost half an hour.

- Requirement 1: The subsystem must include a 6V NiMh battery with at least 2000mAh capacity to power the bot for the duration of the battle.
- Requirement 2: The battery must supply at least 5A of continuous current.
- Requirements 3: 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.

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.

- Requirement 1: 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.
- Requirement 2: The microcontroller must automatically trigger the kill switch when Wi-Fi connectivity is lost, cutting power to the entire system.
- Requirement 3: 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.

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, spinners, 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.

- Requirement 1: The ESP32 must support WiFi communication with a throughput of at least 5 Mbps for reliable control of motors and weapons..
- Requirement 2: The latency should be between 50-100 ms for responsive control, with a preferred range of 35-60 ms.
- Requirement 3: Must provide 6 GPIO pins, including 2 PWM pins for speed control of motors.

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 (2 units)
3. ESP32-WROOM-32E Microcontroller
4. RS380 Brushed High Speed DC Motor
5. Digital 30KG Servo 360 Degree

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	1.6A	9.6W
L298N Dual H-Bridge Motor Driver	6V	2A	12W (per driver)

ESP32-WROOM-32E Microcontroller	3.3V	0.5A	1.65W
RS380 Brushed High Speed DC Motor	6V	0.5A	3W
Digital 30KG Servo 360 Degree	6V	0.183A (at 6V)	1.098W

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

$P\_total = 9.6 + 24 + 1.65 + 3 + 1.098 = 39.348W$

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 (per unit)
ESP32: 1175.04 mm^2 = 0.00117504 m^2
RS380 Motor: 1296 mm^2 = 0.001296 m^2
Digital Servo: 4100.5 mm^2 = 0.0041005 m^2

Now for the Temperature Rise of each Component:

N20 Gear Motor:
$Pavg\_dissipation = 9.6W/2 = 4.8W$
$\Delta T = (4.8*2)/(10*0.0045741589) = 104.25$ °C

L298N Driver:
$P\_dissipation = 24W$
$\Delta T = (24)/(10*0.00387) = 620.93$ °C

ESP32:
$Pavg\_dissipation = 1.65W$
$\Delta T = (1.65)/(10*0.00117504) = 104.25$ °C

RS380 Motor:
$P\_dissipation = 3W$
$\Delta T = (3)/(10*0.001296) = 231.48$ °C

Digital Servo:
P_dissipation = 1.098W
$\Delta T = (1.098)/(10*0.0041005) = 26.83$ °C

Summary:

COMPONENT	**POWER DISSIPATION**	**CALCULATED TEMPERATURE RISE (Celcius)**	**MAX OPERATING TEMP (Celcius)**	**ACTION**
N20 Gear Motors (2 units)	9.6W	104.25	60	Cooling Needed
L298N Dual H-Bridge Drivers (2 units)	24W	620.93	130	Cooling Needed
ESP32-WROOM-32E	1.65W	140.66	85	Cooling Needed
RS380 Brushed DC Motor	3W	231.48	40	Immediate action needed to cool down
Digital 30KG Servo	1.098W	26.83	70	Cooling is not needed but safe to monitor

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.

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.

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.

Citations and References

ESP_WROOM_LINK
ESP_WROOM_32E Datasheet
ESP_WROOM_32 INFO

DC Gear Motor → Wheels
N20 Gear Motors
N20 Gear Motor Datasheet
Dual H-Bridge Motor Driver Datasheet
Brushed High Speed DC Motor

Servo Motor MG995 360 Degree Continuous Rotation - ProtoSupplies → Flipping Mechanism
Servo Motor Vs DC Motor: What are the Main Differences?
Compression Springs vs. Tension Springs: A Guide to Choosing the Right One - Chaoyi Spring
A Deep Dive into Spring Types: The Mechanics of Compression Springs