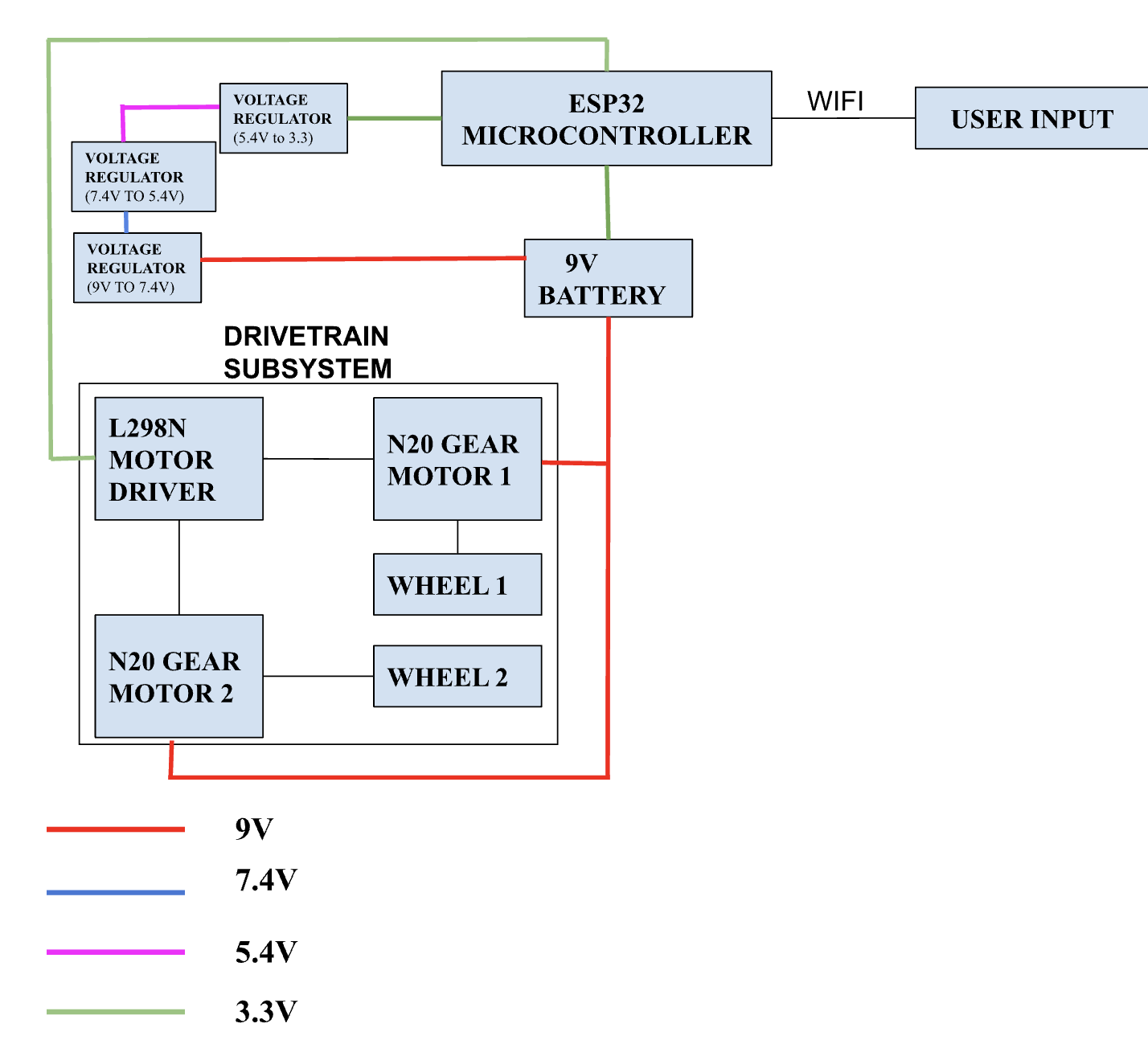
## **Team 37 – Antweight Battlebot**

Deepika Agrawal, Megha Esturi, Ishanvi Lakhani

## **Block Diagram**



## **Requirements and Verification**

### **DriveTrain Subsystem**

| Requirements | Verification |
| --- | --- |
| The motors and motor drivers must operate efficiently with a 9V battery. | Measure the voltage supplied to the motor driver and the motors. Verify that the voltage is between 6 - 8.5V range during operation. |
| The wheels must enable the robot to move at a sufficient speed, achieving an RPM of approximately 50 - 460 . | Calculate the wheel's circumference using its diameter, then find the theoretical speed by multiplying the circumference by the motor's RPM and converting to meters per second. Finally, divide the desired travel distance by the speed to estimate the time required. |

2. To determine the actual RPM of our motors, we conducted a test where the battlebot traveled approximately 10 meters, and we recorded the time taken. Using the wheel diameter of 48 mm (0.048 meters), we calculated the wheel's circumference and applied the following formulas to derive the actual RPM:

Circumference= π × Diameter= π × 0.048m ≈ 0.1508m

Speed= Time / Distance​

Rotations per Second = Circumference / Speed​

RPM = Rotations per Second × 60

By applying these calculations to our recorded data, we compiled the following table comparing the theoretical and actual RPM values

| Weight Added | Distance (meters) | Time (sec) | RPM (Theoretical) | RPM (Actual) |
| --- | --- | --- | --- | --- |
| None | 10 | 22.1 | 450 | 180 |
| 1.8lbs | 10 | 49.74 | 450 | 80 |

### **Defense Subsystem**

| **Requirements** | **Verification** |
| --- | --- |
| Destabilizing Force: The should generate enough force to destabilize the opponent robot. | Place a weight equivalent to the opponent robot on one side of the battle bot. Make the battle bot run into the opponent and see if it gets destabilized or flipped. |

### **Power Subsystem**

| Requirements | Verification |
| --- | --- |
| The subsystem must include a 9V battery to power the bot for about 3-5 minutes which is what we expect the duration of the battle to be. | Test the battlebot until it gets to 6V (voltage at which circuit is not useful) by keeping the power on and making the bot run around for the duration of the time and make sure it doesn’t power down. |

### We tried running the battlebot and measured the voltage being supplied by the battery and we noticed the following pattern. We set the critical voltage to 6V because that is the ideal voltage at which our motors need to function well, therefore we want the battery to supply at least 6V for as long as possible. We expect the battlebot competition to be about 3-5 minutes so we need to ensure it can be powered from that long. Given the graph above, we can say that it would work well during the battlebot competition.

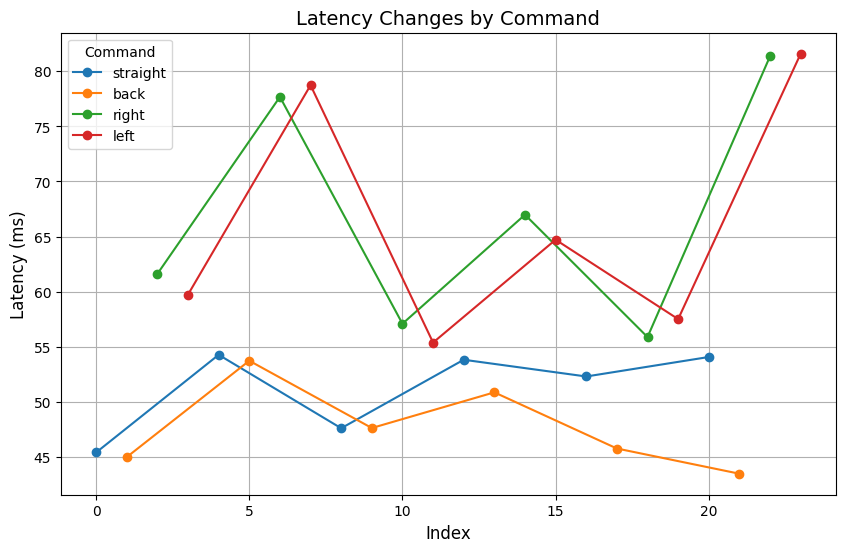
### **Control Subsystem**

| **Requirements** | **Verification** |
| --- | --- |
| The latency should be between 50-100 ms for responsive control. | Measure the time between a button being pressed and the command being relayed. Test multiple times with multiple commands to get average latency across operation. |
| The battlebot should stop operating if WIFI connection is lost as a safety precaution (kill switch) | Turn off wifi, and observe if the battle bot stops operating |
| The battlebot should be able to receive commands and execute them up to a distance of 8.48ft | Put the battlebot at incremental distances up to 10ft from the PC and check the operational condition. |

1. We printed the latency per command on the serial monitor and used the following values. Using them, we calculated the average latency.

| **Command Sent** | **Latency (milliseconds)** |
| --- | --- |
| Straight | 45.39 |
| Back | 44.97 |
| Right | 61.61 |
| Left | 59.73 |
| Straight | 54.26 |
| Back | 53.72 |
| Right | 77.68 |
| Left | 78.74 |
| Straight | 47.59 |
| Back | 57.08 |
| Right | 55.37 |
| Left | 53.81 |
| Straight | 50.85 |
| Back | 66.98 |
| Right | 64.71 |
| Left | 52.29 |
| Straight | 45.75 |
| Back | 55.87 |
| Right | 57.50 |
| Left | 54.06 |
| Straight | 43.46 |
| Back | 42.98 |
| Right | 81.60 |
| Left | 81.37 |

The average latency of this is approximately **58ms** which meets the requirements.



We also observed a pattern where we saw the right and left signals take a little longer to transmit than the straight and back signals. However they are still in the 50 - 100ms range, therefore we were able to verify that this requirement is met.

1. We verified this by switching off mobile hotspot and noticed that the battlebot abruptly stopped. We also added an immediate stop button to ensure if we wanted to stop the movement without stopping the hotspot we can do that.
2. To ensure that the bot will be able to communicate everywhere within the arena, we measured the arena. The arena is approx 6ft lengthwise and breadthwise. Therefore, the maximum distance we would need the signal to reach is the hypotenuse which is 6 multiplied by square root of 2 which is **8.48ft.**

To ensure this, we held the bot and walked far away from it and kept pressing the button until it lost connection. We were able to walk down the entire hallway in the ECEB without it losing connection, therefore we are confident that it will be able to communicate in the arena. Therefore, we were able to verify that this requirement is met.