MEAM 5100: Design of Mechatronic Systems

Team 35: Final Report

Team Members: Sahil Mahendra Mangaonkar, Aamir Khambaty, Jalaj Shukla, Rohan Shende

GRAND THEFT AUTONOMOUS

(SECOND POSITION)

Note:

Although we are a team of 4, 2 of our team members, Sahil & Aamir faced injuries and medical emergencies during the semester and final project. We shall be considered as a team of 3 as per permission from Dr. Mark Yim.

Functionality

Aim: To design and manufacture a robot vehicle that can perform the following functions-

1. Detect and reach objects autonomously on the playing field
2. Transmit an ESP-NOW packet every time it receives communication
3. Autonomously perform wall-following
4. Uses Vive to transmit X-Y coordinates to the user once per second via UDP broadcast
5. Autonomously locate and push the police car

We divided the requirements into three parts:

1. Perform wall following considering it as a reference to move around the arena.
2. Detecting, reaching, and grabbing real and fake trophies
3. Detecting and pushing a police car

1. Perform wall following:

The main objective here was for the bot to move parallel to the walls and take a 90-degree turn at the four corners. To achieve this, we had to choose between two sensors, an ultrasonic sensor and force-sensing resistors/flex sensors. We chose to go with the ultrasonic sensors (HC-SR04) due to their reliability and ease of availability. We initially placed one sensor at the center of all four sides, but later realized that the bot was moving too far from the wall before correcting itself. To tackle this issue, we decided to place two sensors on the left and the right side of the bot, one at the front end and one at the back end. Using this, we could find out the orientation of the bot with the wall and then use feedback control to correct it to move parallel to the wall. It basically helped us to align the bot parallel to the wall by reading the values between the two sensors on the same side. We placed one sensor at the front, and one at the rear to determine when to take turns at the corners. When the sensor at the front measured a set distance that was close enough to the wall, we turned the bot 90 degrees, so during turning, one wheel rotated clockwise and the other rotated counterclockwise. We also added an if-else statement in the code, which would make sure that the bot would not crash into the wall, but instead would turn 90 degrees. The sensor mounted in front of the bot made sure that the distance from the wall is always approximately 20 cm. We used 2 sensors on each side and were able to detect the alignment of the bot with respect to the wall, i.e. if the bot is parallel to the wall then both the sensors facing the wall would give the same sensor reading and the difference between them would be negligible. Hence we provided this error to be of 2cms. Hence it ensured that the bot was always aligned and was parallel to the wall. Using this logic, we were able to implement wall following successfully and were able to complete one lap around the track within 50 seconds. While integrating everything for the final competition, the wall following performed well and it can be seen from the competition video and the submitted videos. We have a video of testing this functionality where it accurately follows the wall 3 times by self-aligning with the side walls along with it performing exceptionally well in the competition as well.

2. Detecting, reaching, and grabbing real and fake trophies (beacon detection):

The main objective for this task was to detect the signals from the real and fake trophies and move the bot toward the trophies to push them. To detect the signal from the beacon, we used two IR phototransistors placed beside each other at a distance of 55 mm as shown in Figure 1, and used an op-amp circuit to amplify the signal. A 3D-printed holder was used to hold the 2 phototransistors at the center of the front face of the bot. The holder made sure that the phototransistors pointed straight towards the front and also made sure that they were shielded from all other sides, minimizing interference of ambient lighting. We adjusted the height of our bot such that the phototransistors are mounted at the same height as that of the IR LEDs in the trophy, to maximize the chances of signal detection. Our logic behind using two phototransistors was that, when the signal from the trophies is detected by both the phototransistors meaning that the bot is aligned to the trophy, then only the bot will move forward. In the beginning, the bot turns on the spot till it detects a signal from the trophy, then it moves in the direction of the signal received. If at some point the signal is lost, the bot will again turn on its spot check for the signal, and then move in a straight line toward the signal. Using this logic, we were able to track the trophy and the fake trophy and push it.

Future work: We had planned to make a gripping mechanism using a single servo motor, which would grip the trophy and bring it back to the doubling zone in our half. Although we had planned and designed parts for this, we did not get enough time to implement it. We also planned on identifying the location coordinates of our and the enemy doubling zones, orienting our bot using the two vive sensors after grabbing the trophy or the fake, and then moving straight toward the coordinates of the doubling zones. While integrating everything for the final competition, the Beacon detection performed significantly well detecting the signals and it was moving towards the trophies. We have a video of testing this functionality individually where it rotates on its own axis, detects & aligns itself in front of the trophies, and pushes it as well as multiple clips from the competition showcasing it.

3. Detecting and pushing the police car:

The main objective of this task was to locate the police car, direct the robot to it, and push the police car. The Bot has 2 Vive Circuits to determine the x, and y positions. We used two Vive Circuits to determine the slope of the bot(Angle of Inclination). This slope is calculated using the *atan* function in the Arduino IDE. It receives the Police Car coordinates using the UDP Handle Function. Once all the coordinates are known the bot will now determine its orientation w.r.t the Police Car and then determine the direction in which it must proceed to reach the Police Car. Using position coordinates and the Front Ultrasonic Sensor it will determine if it has come near the Police Car and it will try to push it.

The police car detection and pushing were visible in the competition rounds. As our team got checked off for this task in the competition it can be seen in the video from the competition that the robot detects the police car understands its coordinates and moves towards it and it pushes it with quite a force as it was seen in the video that our bot was not only capable of pushing just the car but also was able to push other obstacles as well.

Mechanical Design

- Describe intended and actual mechanical performance (wheels, steering, motors, etc.)
- Include things that you tried but failed (and thus learned from)
- Include measured drawings (e.g., CAD) in Appendix

Design approach:

We approached the mechanical design keeping in mind the three functionalities. We wanted our bot to be as compact as possible so that it is easier to steer, but also be heavy enough to push the police car. We studied different types of steering and shortlisted the mecanum and differential steering options. Considering the budget restrictions and simplicity of code and design, we went forward with the differential steering option, in which both the rear wheels would be powered using motors, and there would be a castor wheel at the front to act as a support. Another reason why we chose differential steering was that we had used the same in lab 4, and it worked pretty well. When taking a turn, the wheels would rotate in opposite directions which would ensure that the bot is turning at its spot.

In lab 4, we used hot glue to mount the motors to the laser cut base, and the perf boards were mounted using standoffs. We realized that the use of hot glue was not ideal as it adds a layer of 1-2mm thickness, and also forms an uneven joint. We decided to only use screws, nuts, and bolts for mounting. We cut holes in the acrylic base according to the mounting positions of the sensors. We also 3D printed holders for the ultrasonic sensors

Wheels and motors:

Initially, we used the Yellow T-motors provided for Lab 4, after which we realized that the motors had less torque, and we would need motors with higher torque considering the size and the number of components of the new bot. Hence, we chose 12V DC High Torque Worm gear motors operating at 100 RPM (torque was more important to us than the speed of the motor). The motors were mounted on the base using L-shaped brackets. Adapters were mounted on the motor shafts using screws which would hold the wheels.

Wheels are the only point of contact of the bot with the arena, and choosing the correct wheels was essential in order to optimize the transfer of torque from the motors. Having a good grip on the arena was necessary to push the police car. We tried out different wheels that would fit the Yellow T-motors in lab 4, but they did not have sufficient contact area and grip, hence used to slip on the arena. That is why we chose wheels with a wider diameter and more width, which were also padded with sponge from the inside allowing them to compress and provide a better grip. The wheels were mounted on the adaptors connected to the motor shafts, with the help of screws.

Chassis:

We designed our chassis to be made out of 1/8th inch laser-cut acrylic. We chose 1/8th inch over 1/4th inch because 1/4th inch added unnecessary weight. As mentioned before, we aimed at mounting all our sensors and perf boards using fasteners instead of hot glue, for which we made holes and slots in the acrylic base. We chose transparent acrylic to increase visibility. Our bot was designed to have 2 bases, a lower base that held the motors, ultrasonic sensors, IR circuit, and the battery, and an upper base that held all the perfboards. We aimed to push the trophies instead of grabbing them, and for this, we designed the front end of the lower board to have arm-like protrusions which will make sure the trophies do not slide out while pushing. The bases were separated using standoffs, and so were the perfboards. We also placed the components such that they were symmetrically divided along the lateral axis of the bot, to keep the center of mass near the lateral axis and between the rear wheels and the castor wheel. Since we first planned on using only one ultrasonic sensor at the side, our bot was well within the dimensional limits. But later we had to add another ultrasonic sensor on both sides at the back, which caused the bot to increase in length and thus we had to reduce the length of the arm-like protrusions we made at the front.

Failures in design:

We intended our robot to have a gripper mechanism in the front to pick up and drop the trophy, but due to time constraints we could not implement it. During the competition, we observed that our bot would collide with the trophies, and the sidewalls and the front part would climb over them. This caused the bot to stop. To avoid this we can make a cowcatcher type of attachment which would prevent obstacles from going below the bot. Though temporarily we attached various cardboard components to minimize the bot climbing the wall. Another aspect that we realized was that as we cleared the initial rounds of the competitions they began to get damaged, the screws and the sensor mounts were harmed. Hence we did not account for its ruggedness and did not provide any external casing or protection to the PCB and the connecting power cables, which at times could get pulled off by interference of other teams' robots during the competition.

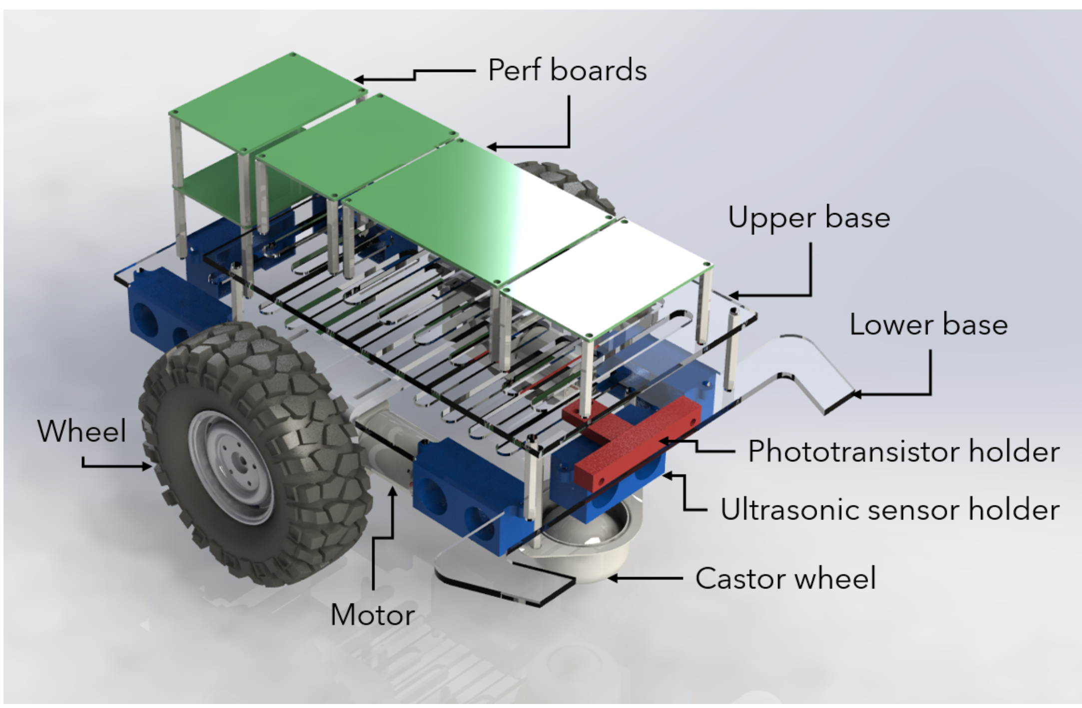

Figure 1: CAD of the completely assembled bot with all the peripherals

Electrical Design

We started with testing each individual circuit on a breadboard. Once we confirmed the working of each circuit, we went ahead with finalizing our circuits on breadboards. We used jumpers and molex connectors to connect all the individual boards together. We added an SPDT switch in series with the battery to disconnect the power whenever needed. We also added a slider switch to disconnect the 5V supply from ESP32-S2 while we upload the programs. 2 of our team members had good experience with hardware design. Hence, all our circuits worked really well.

Vive:

We used the vive circuit to determine the coordinates of our robot. Starting with the 2 vive circuits provided in the class, we tried to modify the circuit to give us the best results. We tested the circuit on breadboard and then soldered it on perfboard. We provided 3.3V supply from an LDO. The circuit had 3 pins, Vcc, GND & output. It has 3 stages namely sensing stage along with the high pass filter, first gain stage & comparator like stage. The Vive circuit was a little noisy sometimes and gave us wrong readings occasionally.

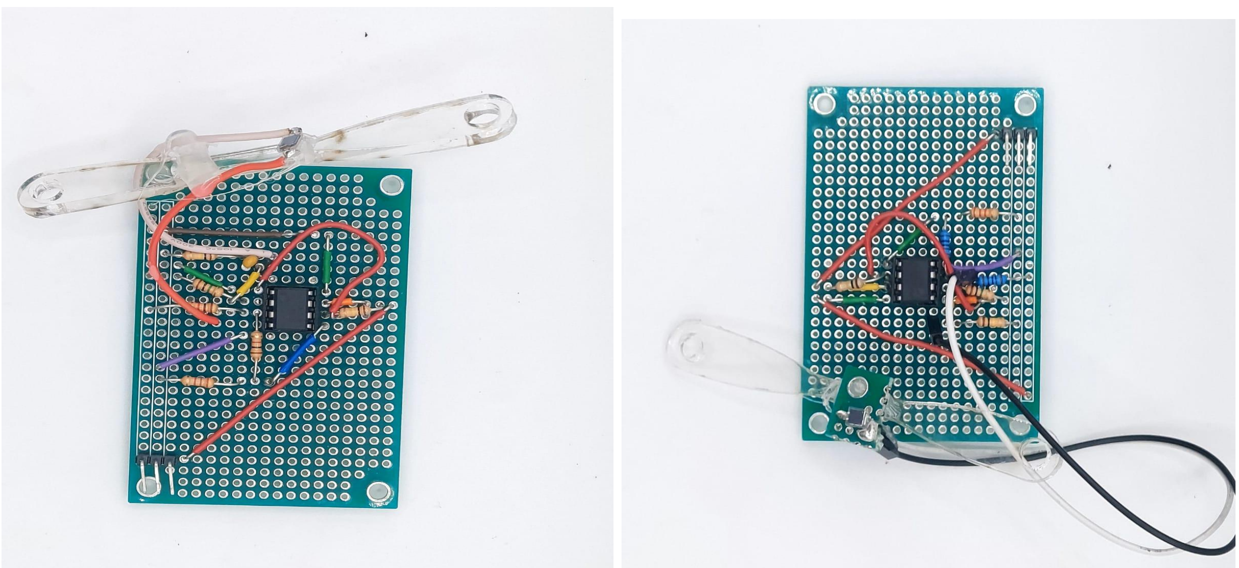

Fig 2: Vive circuit

Beacon Detection:

We used a beacon detection circuit consisting of a phototransistor along with a 3 stage signal conditioning circuit using opamp. As shown in class, the first stage is the photodetection stage that generates an output signal. Capacitor in series with a resistor after stage 1 acts as an AC coupling. Stage 2 amplifies the AC coupled signal from stage 1. The output of stage 2 even though a square wave, it’s still distorted. Hence we used a comparator circuit at stage 3 with a threshold of just more than 50% (~1.8V) which gave us a perfect squarewave of the frequency required. We tested the circuit on breadboard and then soldered it on perfboard. We provided 3.3V supply from an LDO. The circuit had 3 pins, Vcc, GND & output. We could detect the signal from 2.5meters away.

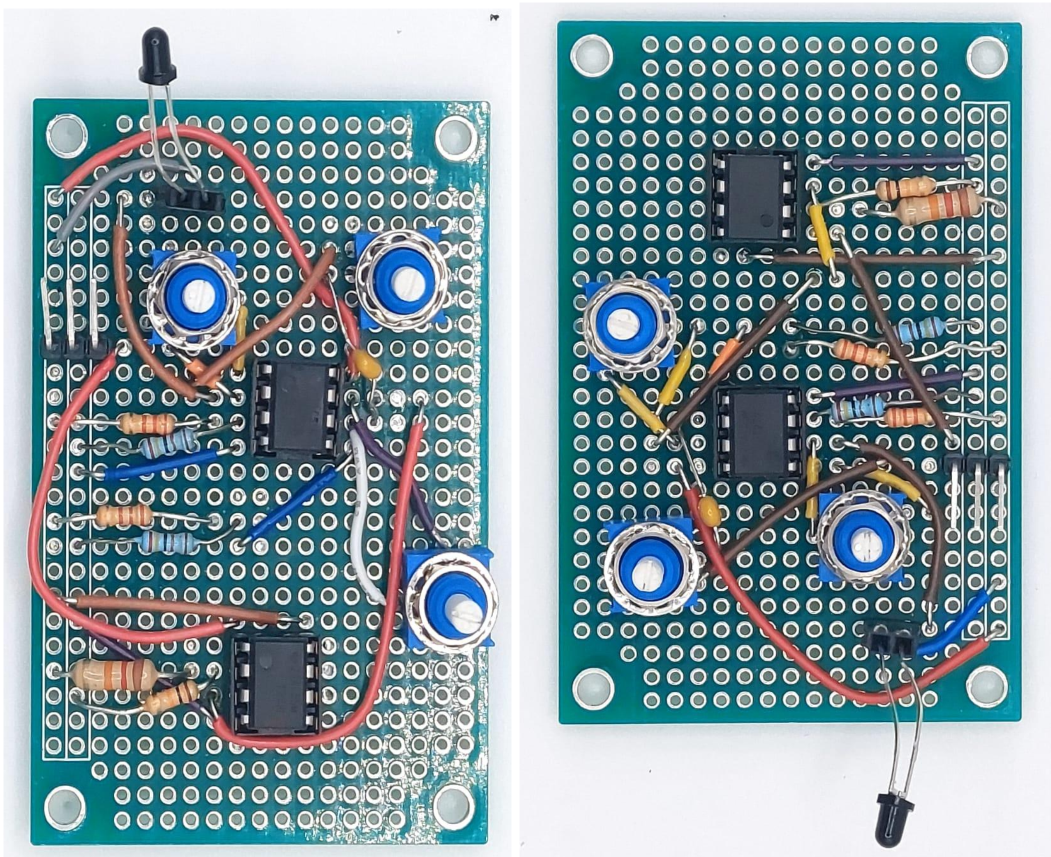

Fig 3: Beacon detection circuit

<u>Sensors for wall following:</u>

We planned to use 6 ultrasonic sensors. 1 each on front and back. 2 on either side. We ended up using 3 of them for simplicity. Front & 2 on the left side. Using the 2 sensors on the left side, we aligned the robot across the left wall of the arena & used the front ultrasonic sensor to decide when to take turns. Ultrasonic sensor HC SR04 works on 5V and its trigger & echo pulse have 5V amplitude. As esp32 works on 3.3V logic, we used level shifters. We added 4 level shifters, although we worked with 2 of them after coming down to 3 ultrasonic sensors instead of 6.

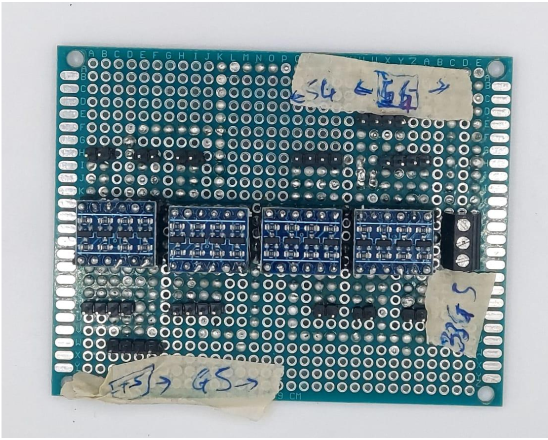

Fig 4: Circuit for ultrasonic sensors along with level shifters

Power circuit:

We needed 3 power supplies 3.3V for vive & IR, 5V for ultrasonic sensors, ESP32 & motor driver, & 14V for motors. We used a 3 cell 11.1V LiPo battery as our power supply. A SPDT switch was added in series to disconnect the supply. We used buck converter LM2596 for 5V, LDO MC33269T for 3.3V & boost converter XI6009E1 for 14V. We also added a slider switch to disconnect the 5V supply from the ESP32 for uploading code.

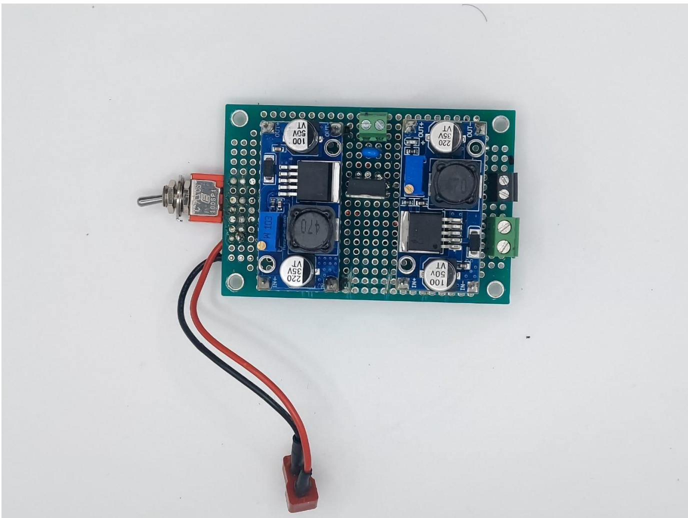

Fig 5: Power Circuit

MCU board and motor driver:

This board contains the ESP32 S2, motor driver, connectors for Vive, IR frequency detector and ultrasonic sensors. As we changed our motors from yellow TT motors to DC Gear motors, we needed to change our motor driver. This motor provided us with better torque as well as speed.

Motor driver TB67H450FNG needs a 5v reference along with 14V supply that we need to turn on the motors. ESP32 was powered using a 5V regulated supply. This board could supply enough power for motors to achieve high torque along with 100 rpm speed.

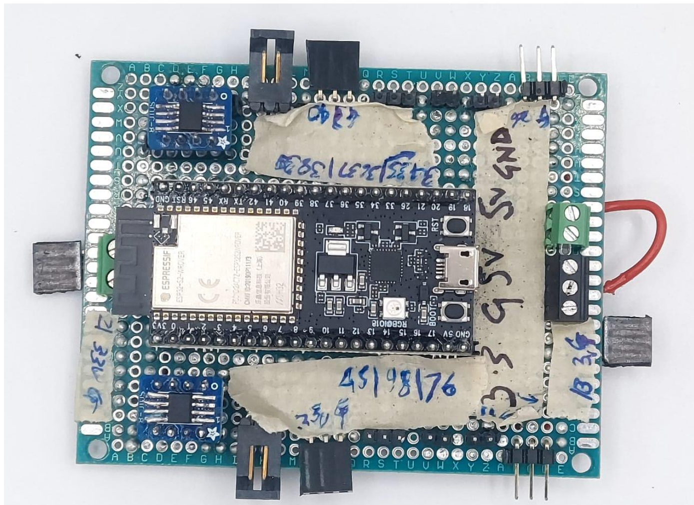

Fig 6: MCU board and motor drive circuit

<u>Failures:</u>

1. We got a lot of noise from the vive circuit which led to inconsistent & wrong readings. We modified the circuit referring to both of the schematics provided in the class, to get consistent & appropriate readings.
2. While trying the IR frequency detection circuit we observed noise and it wasn’t working for a very long distance. By tuning the potentiometer values, we could reduce noise and achieve a range of 2.5meters.
3. Initially, we didn’t add a switch for 5V supply. So when we used to plug in a USB cable, the motor used to start at a low speed using the 5V supply from the USB. We added a switch in order to prevent the 5v supply being fed to the peripherals.
4. We further reduced the noise by soldering all the circuits on perf board.
5. The motor has a brake feature i.e. both of the sides of the motors get high voltage. But the motor driver started burning due to the high amount of current flow. We have observed similar behavior with the ULN2803 darlington pair. We avoided this condition and applied 0 voltage on both sides to stop the motor.
6. We tried using PID running the motor in a straight line by matching their speed. But each motor individually wasn’t giving a constant reading of RPM even when a constant PWM duty cycle was provided. This can be an issue with the motors or the encoders. We fixed this issue by aligning the robot across the wall using 2 ultrasonic sensors.

Processor architecture and code architecture

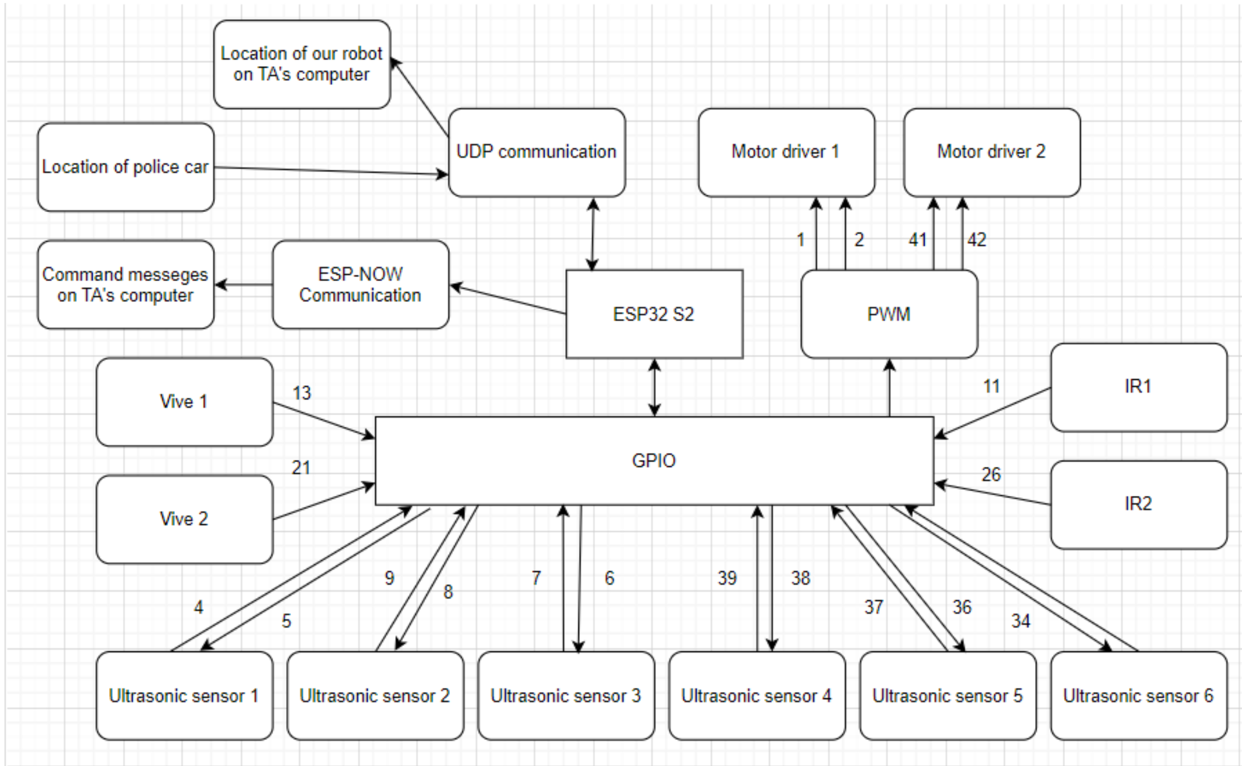

Fig 7: Block diagram of processor & communication architecture

Software Approach:

Due to the large number of devices present in the Lab and our experience from Lab 4 we decided to not use any control mechanism using any WebSever and instead used a completely Autonomous approach to drive our Bot.

The Autonomous control of the Bot was based on the following principle:

Each Task had been assigned a certain priority with a Task Priority Number. Higher the Priority of the task higher was the priority number. The Tasks and their priority were devised as follows:

Task Name	Priority Number
Police Car Tracking	2
Trophy Detection	1
Autonomous Wall Following	0

The default priority was 0 i.e. Wall Following.
When the system is switched on, it begins with the default priority number of 0. This means that it will begin execution of the Wall Following Action(performed only once).

Wall Following Behaviour:

1. Begin Wall Following Behaviour
2. Go straight along the wall if no obstacle is detected in front of the front ultrasonic Sensor.
3. Actively Correct the alignment of the robot while it travels straight.
4. If it detects a wall in front of it then it should turn Right and increment the Turn Flag to indicate the number of turns performed.
5. If the Turn Flag reaches a value of 3, this means that it has performed 3 Right turns indicating the wall following is complete. It will now then change the Task Priority to 1 i.e. Trophy Detection.
6. Also if during any step of the Wall Following Behaviour, if both se receives an IR Frequency of either 23 Hz or 550 Hz on both of its IR Sensors, then it will immediately suspend the Wall Following Task, Stop the Bot and change the Task Priority t0 1, i.e. Trophy Detection.

Trophy Detection:

1. Once the Task Priority is 1, the robot will try to look for the real or fake trophy using the IR Detection Circuit.
2. The robot will slowly circle around its point and scan for any of the two IR Frequencies.
3. If the robot IR Detectors are detecting the same frequency it means that the robot is facing either of the trophies.
4. The robot will now go straight in the direction of the Trophy.
5. With the help of the front ultrasonic sensor it will then determine if the robot is close to the trophy and it will start pushing it.
6. The robot will then update the Task Priority to 2, i.e Police Car Tracking.

Police Car Detection:

1. Once the Task Priority is two, the robot will start looking for the Police Car.
2. The Bot has 2 Vive Circuits to determine x,y position. We used two Vive Circuits to determine the slope of the robot (Angle of Inclination).
3. This slope is calculated using the *atan* function in the Arduino IDE.
4. The bot then broadcasts its x,y position to the Staff using UDP Broadcast.
5. It also receives the Police Car co-ordinates using the UDP Handle Function.
6. Once all the coordinates are known the bot will now determine its orientation w.r.t the Police Car and then determine the direction in which it must proceed to reach the Police Car.
7. Using position coordinates and the Front Ultrasonic Sensor it will determine if it has come near the Police Car and it will try to push it.
8. Once this Task is completed the Task Priority will go back to 1 i.e. Trophy Detection.

Failures and Learning Outcomes:

1. We were unable to fine tune our PID Model and thus had to do away with it. I believe this is due to the fact that we did not have a robust RPM Detection Mechanism in software coupled with poorly tuned PID Parameters.
2. We were also unable to perform adequate pushing behavior. Often our front ultrasonic sensor would not work at such close proximities, confusing the robot and making it behave in an unexpected manner.

Appendix

Bill of materials:

Sr. no.	Item	Quantity	Source
1	DC Geared Motor	2	Amazon
2	Motor bracket	2	Amazon
3	Motor shaft adapter	2	Amazon
4	Wheels	2	Amazon
5	Castor wheel	1	GM Lab
6	3S Lipo battery	2	Amazon
7	HC-SR04 Ultrasonic sensor	6	Amazon
8	Buck converter	1	Amazon
9	Boost converter	1	Amazon
10	Phototransistor	2	GM Lab
11	Lower base and upper base (1/8th inch acrylic)	1	RPL
12	ESP32 S2	1	GM Lab
13	TB67H450FNG Motor driver	1	GM Lab

Circuit Schematics:

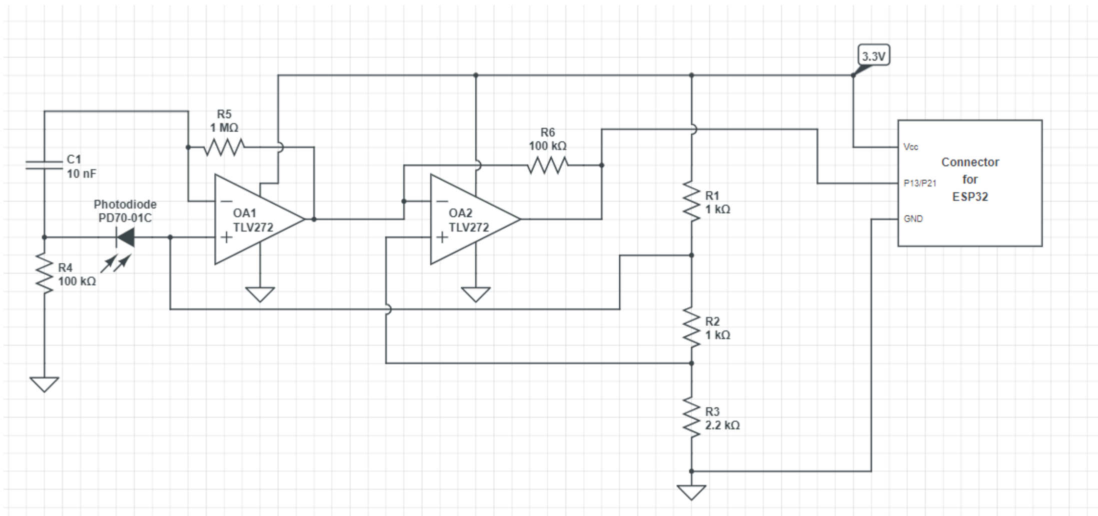

Fig 8: Vive circuit

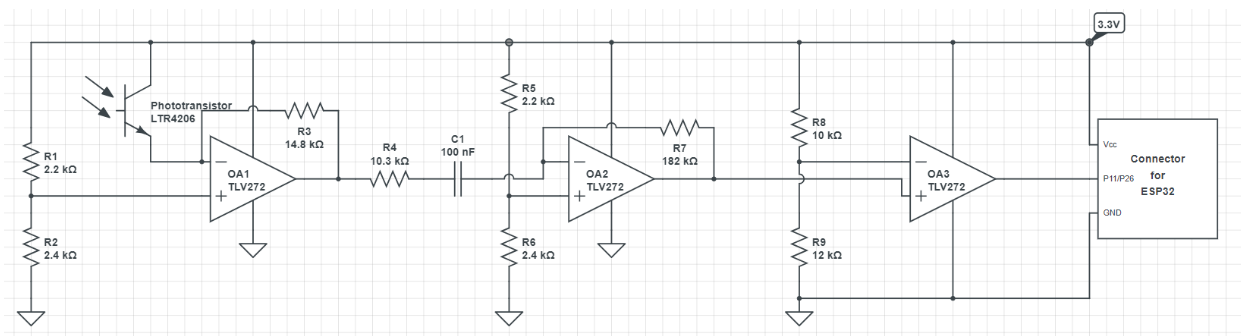

Fig 9: Circuit for Trophy detection

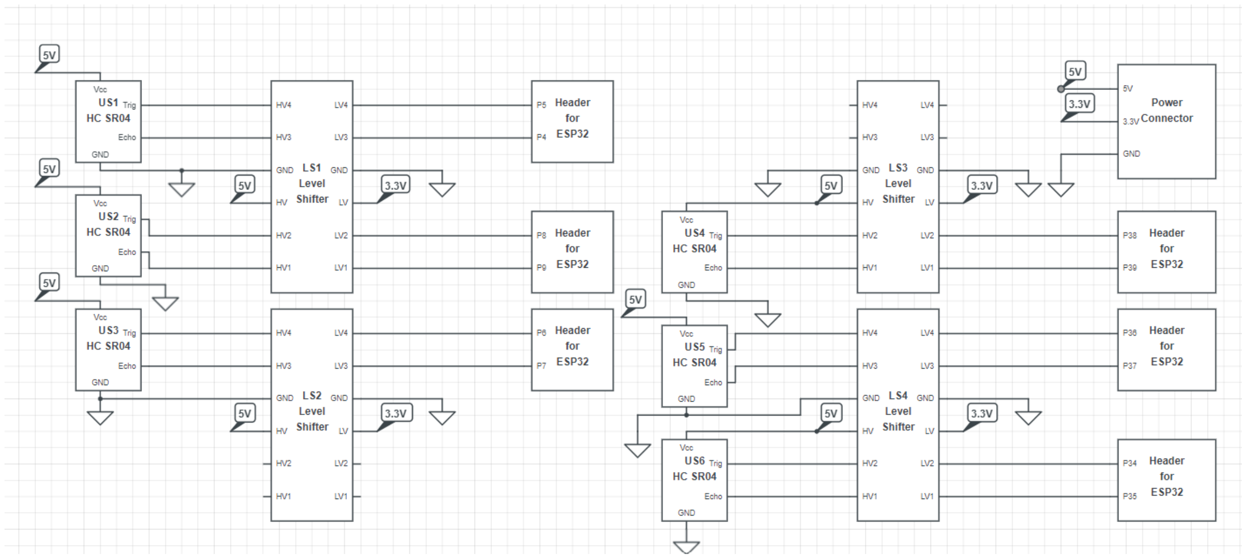

Fig 10: Circuit for ultrasonic sensors along with level shifters

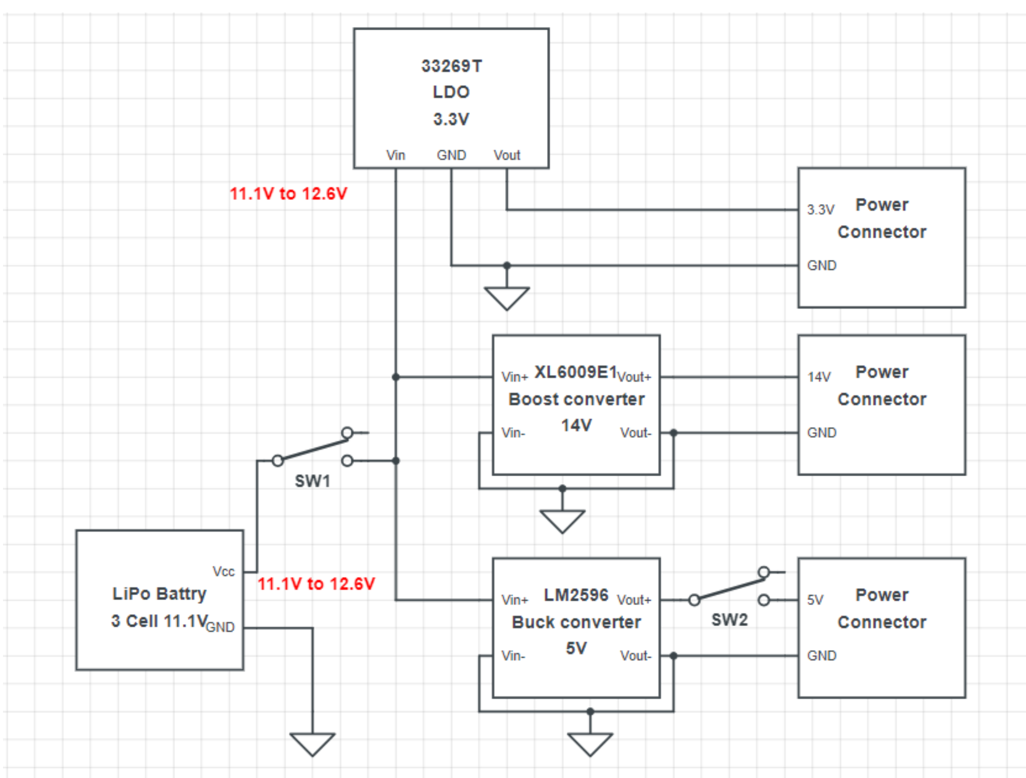

Fig 11: Circuit for Power regulation

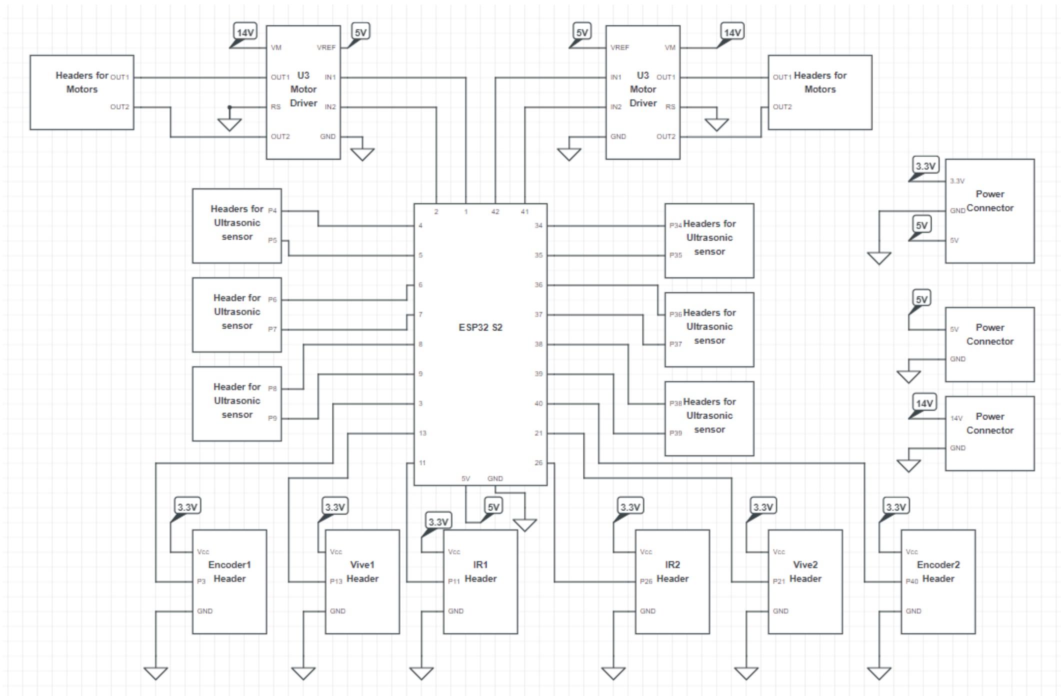

Fig 12: Circuit with ESP32 & Motor driver

Pictures of the robot:

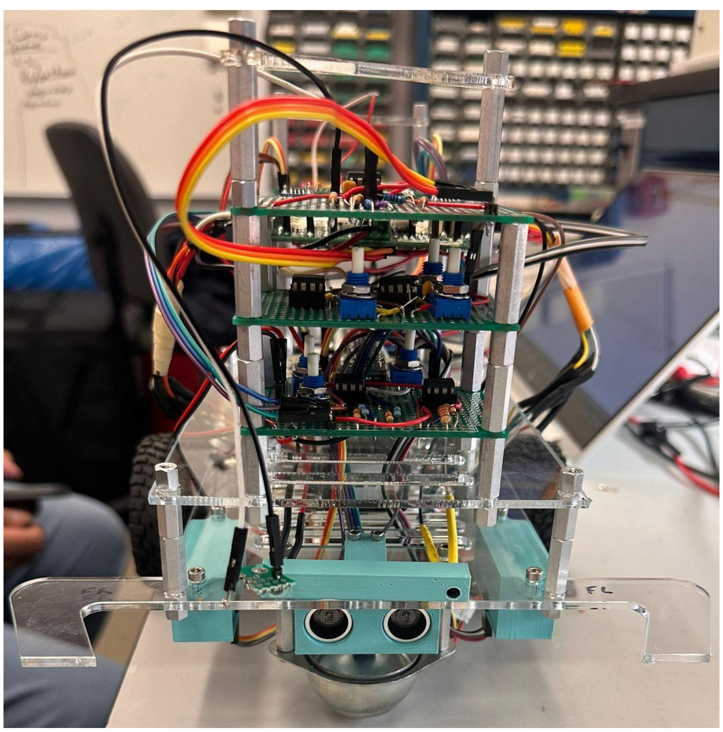

Fig 13: First look front view

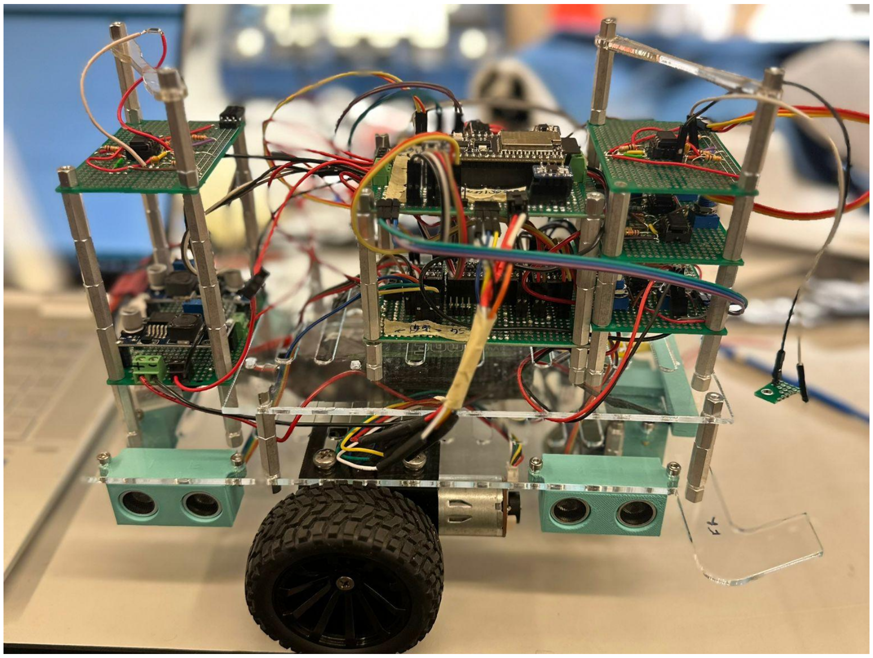

Fig 14: First look side view

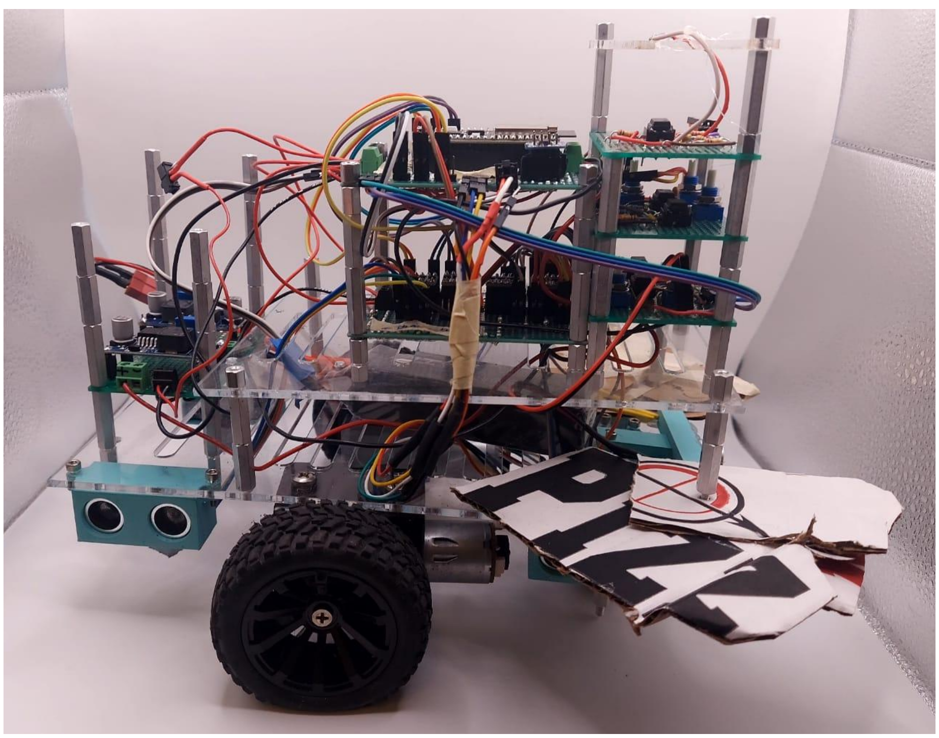

Fig 15: After competition look side view

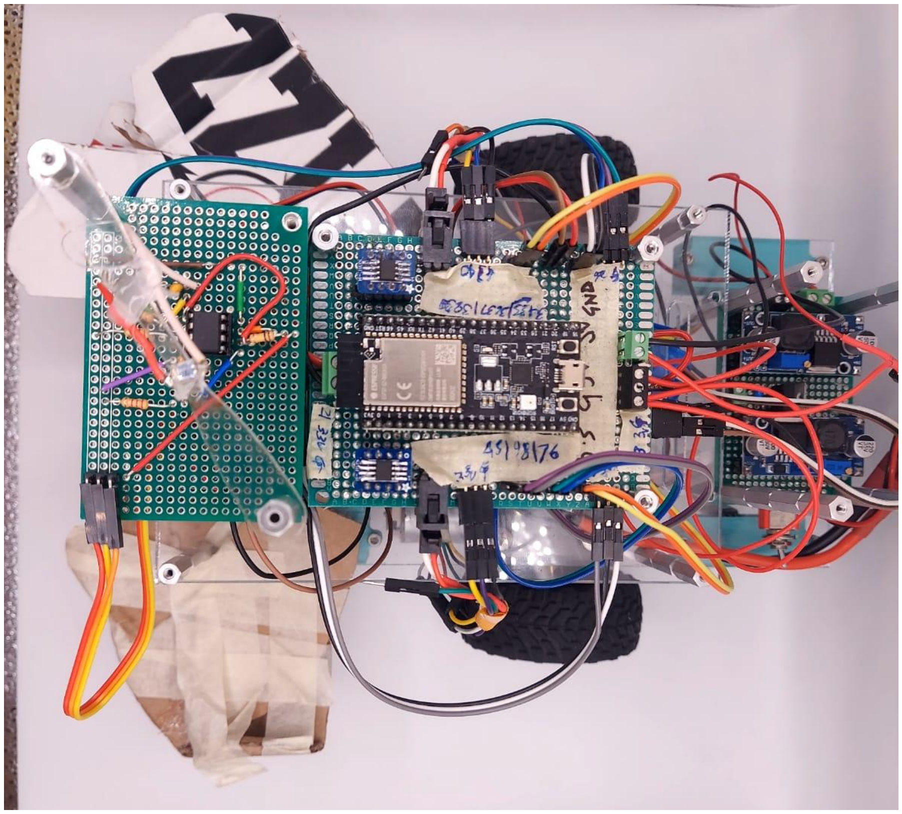

Fig 16: After competition top view

CAD Drawings:

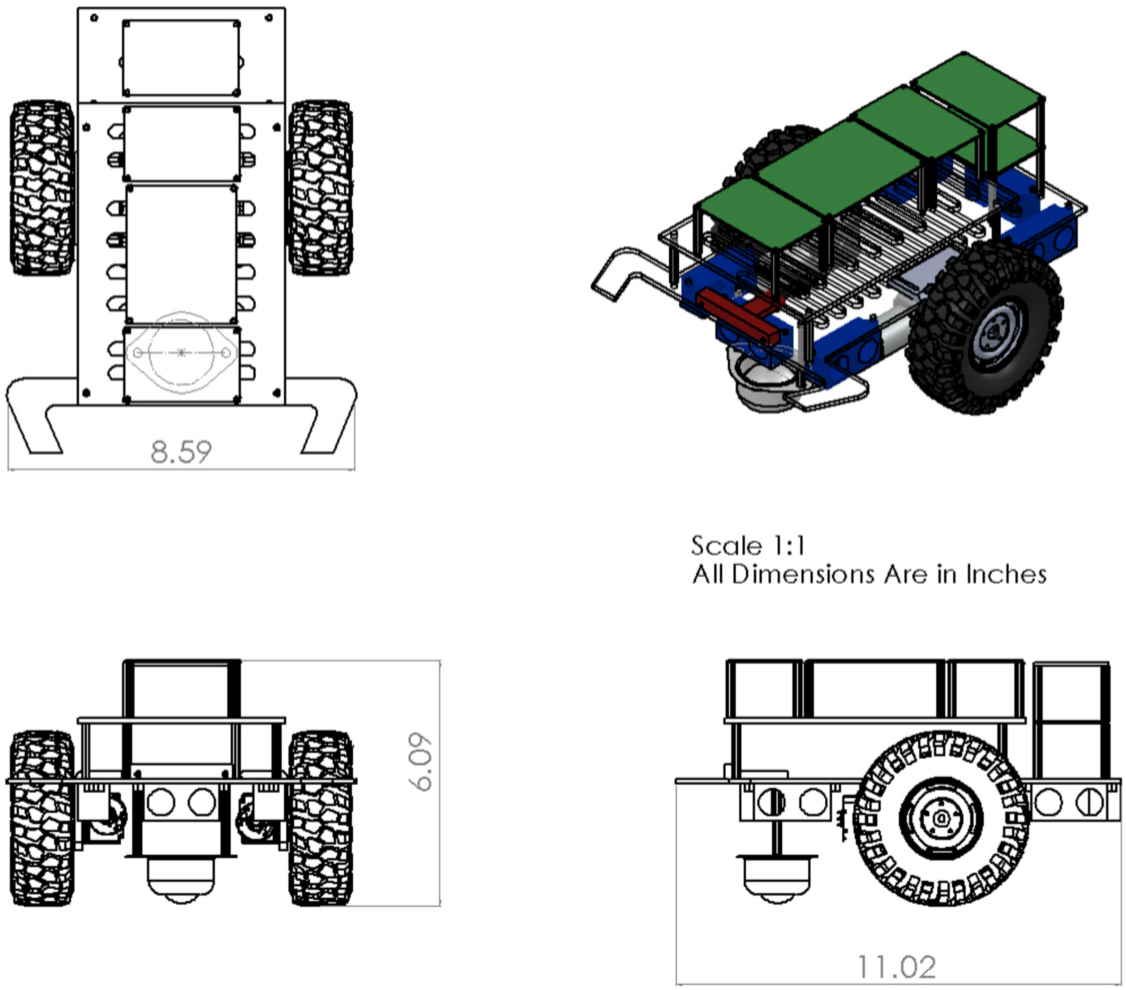

Fig. 17: Dimensions of the bot

Datasheets:

1. DC Gear Motor
2. HC-SR04 Ultrasonic sensor
3. TB67H450FNG Motor Driver
4. LM2596 Buck converter
5. XI6009E1 Boost converter
6. MC33269T 3.3V LDO

Videos:

1. Wall following individual functionality
2. Beacon detection individual functionality
3. Wall following in competition
4. Beacon tracking in competition
5. Police car in competition

Retrospective (non-graded)

Sahil: The most important thing I learnt from this class is integrating hardware, software and mechanical aspects to make a robot that can perform designated tasks accurately. That's the best part of the class where I used everything from the class and labs to achieve something. I can read and understand datasheets, use the right components, compare & choose from various approaches to complete a task. I had the most trouble with lab 2, detection of IR signals, especially at a long distance (i.e. Lab 2.4). If we could allot more time for the final project then we could have done better.

Jalaj: I learned the art of patience and problem identification. I also was able to form steps for systematic problem-solving. Rather than learning something technical, I was even able to learn the soft skill of thinking in a particular way which aided me in identifying and defining a problem and scrutinizing the possible outcome of solutions. Apart from this, I realized that there are multiple ways to solve the same problem and hence it forced me and trained me to be a lateral thinker. The competition was the best part where I not only integrated the bot but also developed good friends and learned teamwork and people management skills. The time vested for completion of the project was a lot and that can be reduced by employing different trustworthy peripherals.

Aamir: Integrating Hardware both Electrical and Mechanical all controlled by software is one of the hardest challenges. This course not only teaches each aspect of these individually but also provides a great insight into the various challenges one faces while consolidating these systems. I learnt so many new things not just about the field of Mechanical Engineering but also the various new techniques of solving the same problem statement which has helped widen my thinking spectrum.

Rohan: This course has been a rollercoaster of emotions throughout the semester. The most important thing I learned in this course is the use of electrical components, sensors and MCUs, along with coding in C. There was a perfect mix of mechanical and electrical engineering, with a bit of controls as well. Being a mechanical engineer, I was looking for a course that would introduce me to electrical engineering and coding, and this one turned out to be exactly what I was looking for. I enjoyed working on lab 3 and 4 the most, since there was design and fabrication included, which I was familiar with. I had some trouble getting checked off for lab 2. One thing I feel should be changed is the time allotted for the final project. If one of the functionalities like wall following or beacon detection could be pushed to lab 4, teams will have a better chance of completing the project on time. Overall, I enjoyed this course a lot and I would like to thank Prof. Mark for taking the efforts throughout the course, and especially to accommodate our team during the final competition.

Acknowledgments:

We would like to express my very great appreciation to Dr Mark Yim for his excellent teaching and kind consideration in our condition. His willingness to give his time and accommodate students generously is very much appreciated. We would also like to thank all the teaching staff for all of their support & feedback.

Fig 18: Team picture with Dr. Mark Yim after being first runner up at the competition