MEAM5100 Final Report

Team 16 - Wenhao Wang, Xiaohang Wang, Binglong Bao

We are the champions!

1. Functionality

High level description of the approach:

We divided our robot into seven systems for presentation. The robot is equipped with three ESP32 boards, and the operating computer is connected to an ESP32C3 board. When all seven systems work together, the robot can be assigned different modes (wall-following, police car pushing, trophy hunting) and manual controls.

Design Choices:

Our game-winning strategy was significantly shaped by the scoring rules from the very beginning of the design phase. In the selection stage, to move the heavy police cars, we chose motors that were stronger yet slower, complemented by the more stable and efficient L298N motor control board and higher voltage 4s Lipo batteries. In terms of coding, knowing that higher multipliers could be achieved with fewer messages sent, we incorporated various functionalities into distinct modes in our code to minimize operational frequency.

Furthermore, to enhance maintainability, we designed our robot modularly. For instance, we completely separated the control of the mobile base movement from the upper layer sensors. The board controlling the movement could be debugged and function independently from the sensor board, allowing parallel progress in our work. This approach helped us become the first team to successfully complete the checkoff with full points. We also segregated the motor power supply from the motor control board power supply, preventing any incidents of board damage during debugging. The code structure was also highly modularized, with functionalities implemented as classes that were then included in the main file.

Finally, we placed a strong emphasis on stability. All circuits were soldered to avoid exposed wire ends. All sensors and soldered boards were designed with interfaces and screw holes for secure mounting on the vehicle, rather than just being taped on. Additionally, we installed a switch to control the power supply to the base, greatly facilitating the debugging process.

Approach for All Functionality:

Vive System: Comprising two sensors located at the front and back of the robot, the Vive system receives external signals to determine the robot's 2D coordinates. Using these two sensors, we can calculate the robot's orientation and central coordinates. By placing the robot at the four corners of the field, we establish a coordinate system, allowing us to pinpoint the robot's position based on sensor data.

TOF System: This system includes a front sensor and two right-side VL53L0X. The front sensor detects the distance from walls in wall-following mode and trophy distance in trophy-hunting mode. The right-side sensors control the angle and distance of the robot from the wall in wall-following mode.

IR Sensor System: Consisting of two IR sensors and two amplifying circuits, this system connects to an ESP32C3 board. A signal from one sensor indicates the general direction is correct but needs slight adjustments, while signals from both sensors indicate a trophy directly ahead and ready to go forward.

Gripper System: The system features a metal gripper and a MG996R servo, controlled by PWM to open and close the gripper.

Mobile Base: Two motors and an L298N motor control board make up this system. Positional PID control is used to achieve desired motor speeds. Controls include 'WASD' for forward/backward and rotational movement, 'QE' for directional shift while moving forward, and 'ZC' for slow rotation.

Control and Communication System: Four ESP32 boards are required for robot control, with real-time communication between them. The first ESP32 WROOM board controls the mobile base and performs PID calculations. The second ESP32 WROOM board processes sensor data and decisions. The third ESP32 C3 board handles initial IR signal processing and Vive data. The fourth ESP32 C3 board, connected to the operator's computer, controls the modes.

Power Supply: The first and second boards are powered by Miady batteries, the claw by a 2S Lipo 7.4V battery, and the motor by a 4S Lipo 14.8V battery. Other sensors and the third board are powered by their respective control boards, and the fourth board is powered by the operator's computer.

Performance in Final Competition and Explanations:

Nearly all systems performed as expected during the Final competition. However, the robot's functionality of returning to our own field after capturing the 'trophy' did not perform well. We identified several reasons for this:

1. The IR system is significantly affected by the environment. During the check-off process, there were no other robots or 'police cars' within the field to serve as interfering targets, and the IR system's navigation function was very stable. But in the Final competition, due to other robots blocking signals and changes in environmental noise and lighting conditions, the IR system did not accurately locate the 'trophy.'

2. In the competition, after successfully capturing the 'trophy,' we needed to return to our half of the field. During this process, we might encounter blockages from opposing robots, which could cause our robot to get stuck and fail to return successfully.

3. The IR system also gives correct detections for reflections. In the final competition field, the presence of reflective tape caused a decrease in the reliability of the IR system's detection. This was also a reason for the failure of the IR system to capture the 'trophy' effectively.

2. Mechanical Design

Parts of Mechanical System:

Motors: Based on our analysis of the game's rules, we concluded that the system's reliability is more important than the speed of the robot. Therefore, we chose a motor with a lower maximum rotation speed but higher torque, which exhibited superior performance in the competition, especially when pushing the police car.

Wheels:
To ensure a stable connection between the wheels and the motor, we selected the wheels based on the motor's size. This decision was influenced by lessons learned in Lab 4. During Lab 4, our goal was to create custom linkage components, but unfortunately, this led to limited reliability in the transmission system. The frequently problematic linkage system presented many challenges in Lab 4.

Layer Structure: To reduce the complexity of assembly and maintenance and to simplify the placement of welding boards, we designed the structure in two layers. The first layer contains the motors and wheels, while the second layer is used for arranging the control boards and other electronic circuit welding boards. This design proved to be very effective in subsequent debugging and maintenance processes. We rarely had to remove the motors and wheels due to maintenance needs.

Steering: Similar to the differential steering method we used in Lab 4, we continued to use this method for the steering function in the Final Project. Its significant advantage is a smaller turning radius and quicker turning process. However, its downside is the inability to steer while moving and the higher torque required for turning. But, considering the high torque of our chosen motors and the lack of need for steering while moving in the Final competition, our steering method performed well throughout the check-off and competition process.

Failure in Mechanical Design:

In fact, during our mechanical structure design process, we did not encounter any significant failures. This was because we had thorough communication and theoretical feasibility analysis before starting the design, aiming to avoid using structures with potential risks as much as possible. Moreover, after the completion of the CAD drawing, we engaged in further discussions and modifications based on the visualized drawings. These processes of communication and discussion ensured that everyone was sufficiently familiar with the mechanical structure, greatly reducing the likelihood of errors.

One failure we encountered in our mechanical design was related to the thickness of the base plate. Since we chose to use ball bearings as our driven wheels, they could not deform like rubber tires to ensure contact between the driving wheel and the ground. The use of a 1/4 inch laser-cutting board as the base plate resulted in a small contact area between the driving wheel and the ground, leading to the risk of the vehicle becoming airborne and severely impacting its mobility.

To address this, we opted for two 1/8 inch laser-cutting boards and used the rastering feature to reduce the thickness where the driven wheels were installed. This solution not only ensured the strength of the base plate but also solved the issue of the vehicle becoming easily airborne. During the subsequent Final competition, the presence of tape on the ground caused some teams' vehicles to tip up. However, because we had already made adjustments in our design process, we were able to avoid this problem to a certain extent.

3. Electrical Design

Intended and actual performance:

Vive(for localization and pushing the police car): The Vive circuit is used for localization finding where the robot is with respect to its environment. In the final project, the base station does the light sweeping, and the photosensor senses initial flash and sweeping light. The timer measures the time between flash and sweep.

- **Intended Performance:** The relative position coordinates (x, y) of the robot are converted and output through program processing which is the intended performance to be realized by the vive circuit. By using two vive circuits at a certain distance from each other, we can obtain the coordinates of the position of two points on the robot, which can be calculated to determine the direction of the robot.
- **Actual Performance:** The positional coordinates of both vives can be obtained successfully, but due to the interference of environmental noise and light, sometimes the obtained coordinates will have abnormal jumps (extremely large or small values), which will affect the positioning and direction determination to a certain extent.

TOF(for wall following): Time-of-Flight (VL53L0X) is used to measure the distance between objects and a sensor by leveraging the time it takes for light to travel to the object and back. In the final project, we plan to use three TOF sensors to accomplish the task of wall following, one on the front of the robot and two on the side, and perform the control through the three TOF return values.

- **Intended Performance:** Thus, the intended performance of the TOF circuit is to obtain the return values of the three individual TOF sensors, which are relatively accurate and do not interfere with each other.
- **Actual Performance:** In fact, with independent XSHUT pin settings and address assignments, all three TOFs can successfully return the correct measured distance values. However, due to the measurement accuracy of the TOF module, when the object is too close (less than or equal to 5cm), the returned measurement data will be inaccurate.

IR(for beacon tracking): The IR phototransistors are used to track signals of a specific frequency emitted by the trophy. We plan to use two IR sensors placed on both sides of the front of the robot to increase the detection range as well as accuracy.

- **Intended Performance:** The intended performance of the IR circuit is to detect the signal frequency as accurately as possible, i.e., when the IR is aligned with the trophy it should show the specific frequency signal it emits, and when it is not aligned it should not detect the signal. we use different color LEDs on the ESP32c3 board to indicate the detection status of the two IRs, and only when both left and right IRs detect the signal will it be regarded as a successful detection and execute the corresponding commands. This is so that when the front of the car is aligned with the trophy, it can stop and move forward so that it can track and get the trophy.
- **Actual Performance:** When actually tested, the IR performed perfectly on the oscilloscope as well as on the test field. On the oscilloscope, the signal was

detected as a perfect square wave with an accurate frequency, and on the test field, the car entered the search mode and turned slowly. When the front end was aligned with the trophy, it would stop and move forward to reach the trophy. However, during the final competition field, we found that the signal amplification gain of the circuit was too strong, resulting in some reflected signals from the field being detected as well, which led to the failure of IR detection.

Things we tried but failed:

Vive

- **Tried but failed:** Our initial idea was to integrate Vive, sendUDP, and espnow communication control functions all in one board, then we found out during actual testing that espnow and sendUDP would conflict because of the wife channel settings, and this problem has not been solved until the end. We can only add a new board to separate these functions.
- **Learned from:** Don't be too ambitious and idealistic at the beginning, and always be ready to plan B to face possible and unexpected bugs!
- **Tried but failed:** To solve the previously mentioned problem of jumping Vive coordinate values, we added a median filter function to the code to get more accurate values. It performed perfectly when we tested it on its own, reading much more stable vive coordinates, but it was only when we fused its other sensor data with the program code that we found the problem. Because of the delay added to the filter function, this affects the implementation of other functions, such as causing a delay in steering when following a wall. Finally, we dropped the filtering function.
- **Learned from:** Think holistically, not just individual features. Sometimes a single feature that performs perfectly ends up having an impact on the entire project

IR

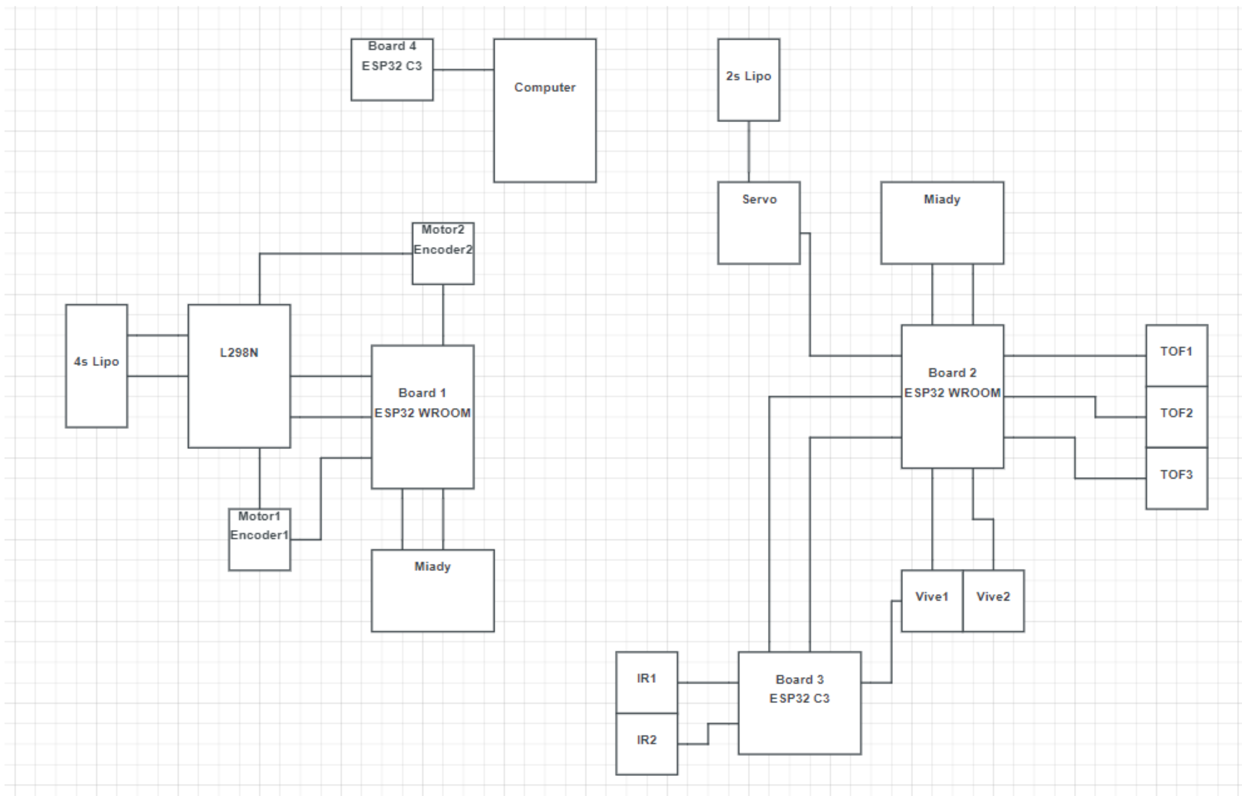
- **Tried but failed:** As mentioned before, our IRs showed a good signal on the oscilloscope and performed well during the test, but when we arrived at the competition venue due to changes in light and reflections, the IRs were rendered ineffective. We had no time to adjust the circuit and failed in the function.
- **Learned from:** Always consider accidental and environmental factors, experimental and actual conditions are often different.

- Include schematics in Appendix.

  The schematics of all circuits are included in the Appendix below.

4. Processor architecture and code architecture

Block diagram:



Software approach:

The robot is controlled by four ESP32 boards, which communicate with each other in real-time.

First ESP32 WROOM Board: This board is responsible for controlling the mobile base and performing PID calculations. Its outputs are the direction pins and PWM speed pins for motor control. The inputs include data from the motor's encoder (to calculate motor speed) and instructions received from the upper-level board. Based on these instructions and the PID calculation results, it selects and outputs the motor's speed.

Type	Name
Class	MotorState
Class	L298N
Class	PID

Function	calculateSpeed(MotorState& motorState)
Function	setupESPNOW()
Function	setupEncoder()
Function	setupMotor()
Function	setupPID()
Function	setup()
Function	loop()
Function	OnDataRecv(const uint8_t *mac_addr, const uint8_t *data, int data_len)

Second ESP32 WROOM Board: This board handles reading and decision-making based on sensor data. It aggregates data from IR, Vive, and TOF sensors. Following the operator's commands, it executes specific modes and issues instructions to both the gripper and the base.

Type	**Name**
Class	TOFSensorManager
Class	Servo
Function	send_message(char where, uint8_t key)
Function	send_message(char where, const char* message)
Function	OnDataRecv(const uint8_t *mac_addr, const uint8_t *data, int data_len)
Function	closeGripper()
Function	openGripper()
Function	setupESPNOW()
Function	isValidKey(char key)
Function	isValidGripper(char key)
Function	setup()
Function	loop()

In this main loop, the robot responds to command inputs for movement and gripper control. It processes received ESPNOW messages to determine movement directions ('w', 'a', 's', 'd', etc.) or gripper actions ('o' for open, 'l' for close). Based on the received command, it either adjusts the servo for the gripper or sends movement commands to the mobile base. Additionally, the loop includes conditional logic for two autonomous modes: wall following (ModeStatus == '1') and trophy searching (ModeStatus == '2'). In these modes, the robot uses sensor data (TOF and IR) to navigate and interact with its environment.

Third ESP32 C3 Board: Primarily dealing with the preliminary processing of IR signals, this board calculates the frequency of IR signals received in its main loop to determine if they signify a trophy or a fake. The results are displayed on LED lights and sent back to the second board via two GPIO pins. This board also reads Vive data (which is shared with both the second and third boards) and transmits UDP.

Type	Name
Class	Vive510
Class	WiFiUDP
Function	UdpSend(int x, int y)
Function	readBeaconFreq(int pin)
Function	detectTrophy(int pin)
Function	detectFake(int pin)
Function	setup()
Function	loop()

In this loop, the primary tasks are sending Vive data over UDP and processing IR signals for object identification. The Vive tracking data is sent at regular intervals, and the Vive tracker's status is continually checked. The loop also handles IR signal processing to detect trophies or fakes based on their specific frequency ranges. Outputs are controlled based on these detections, and visual feedback is provided through LED color changes. The loop also allows for threshold adjustments of the IR signals via serial input.

Fourth ESP32 C3 Board: Connected to the operator's computer, this board is in charge of mode control. It displays sensor information from the second board and allows the operator to send mode commands to the second board.

The communication between the first, second, and fourth boards is facilitated by ESPNOW, while the second and third boards communicate through two GPIO pins.

The libraries we used:

[Adafruit_VL53L0X](https://www.arduino.cc/reference/en/libraries/adafruit_vl53l0x/)

[ESP32Servo by K.H.](https://www.arduino.cc/reference/en/libraries/esp32servo/)

[ESP32Encoder by K.H.](https://www.arduino.cc/reference/en/libraries/esp32encoder/)

[PID by B.B.](https://www.arduino.cc/reference/en/libraries/pid/)

[L298N by A.L.](https://www.arduino.cc/reference/en/libraries/l298n/)

(I modified it as L298N-ESP32, adding `setVelocity` method for more convenient PWM output.)

Things we tried but failed:

UDP and ESPNOW channel confliction

- ESPNOW uses Wi-Fi channels for transmitting information. To send data, Wi-Fi must be set to STA (Station) mode and a specific channel must be designated. However, broadcasting UDP also requires the use of Wi-Fi. When an ESP32 board connects to a router, it automatically synchronizes to the router's set channel, which often doesn't match the channel set for ESPNOW. This mismatch leads to a "peer channel is different from home channel" error, resulting in transmission failures. We tried switching to the router's channel 6, but this did not fully resolve the issue.
- We managed to achieve stable one-way communication while broadcasting UDP, but two-way communication was highly unstable. In some cases, transmissions would succeed, but they often resulted in "out of memory" errors or showed as successful without being received. Ultimately, we decided to add an additional ESP32 C3 board dedicated solely to broadcasting UDP coordinates and used GPIO communication.
- We learned that both wired and wireless communications have their respective advantages and disadvantages. Wired communication involves more complex routing of wires, increasing the risk of issues like loose connections, especially when using Dupont wires instead of printed PCBs. However, its benefit lies in the stability of the software aspect. On the other hand, wireless communication eliminates problems associated with hardware connections but requires more performance and time to be effective.

Upload code:

Code has been submitted.

5. Appendix

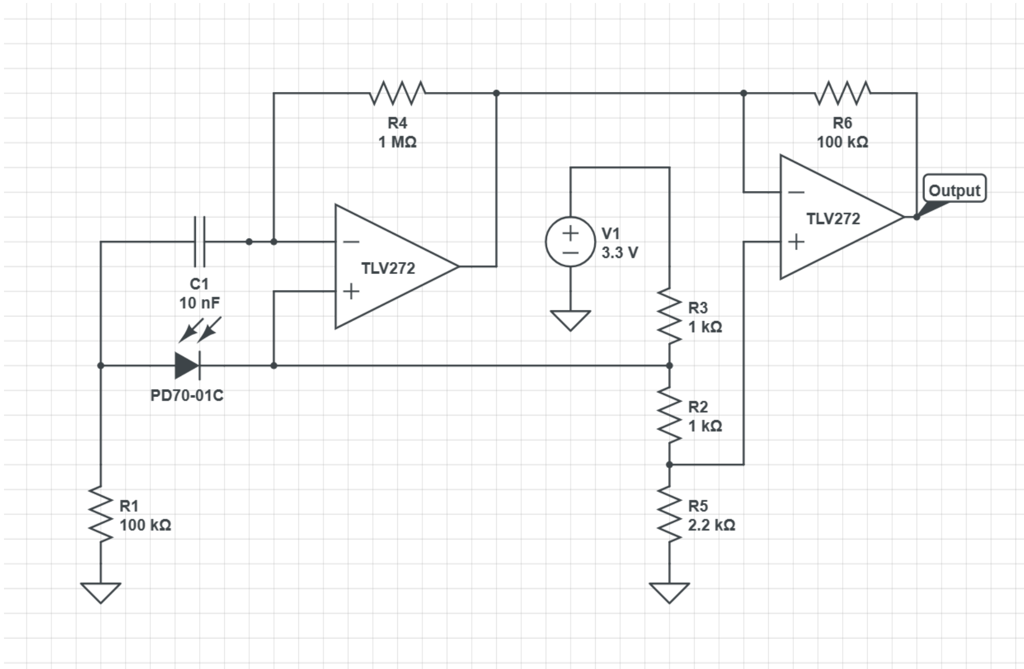
1. **BOM**

Item	Quantities	Price
DC Motors	2	$68.0
Wheels	2	$7.50
ToF sensors	6	$17.0
Claw	1	$18.0
Lipo Battery	1	$17.5
Wheel connectors	2	$3.75

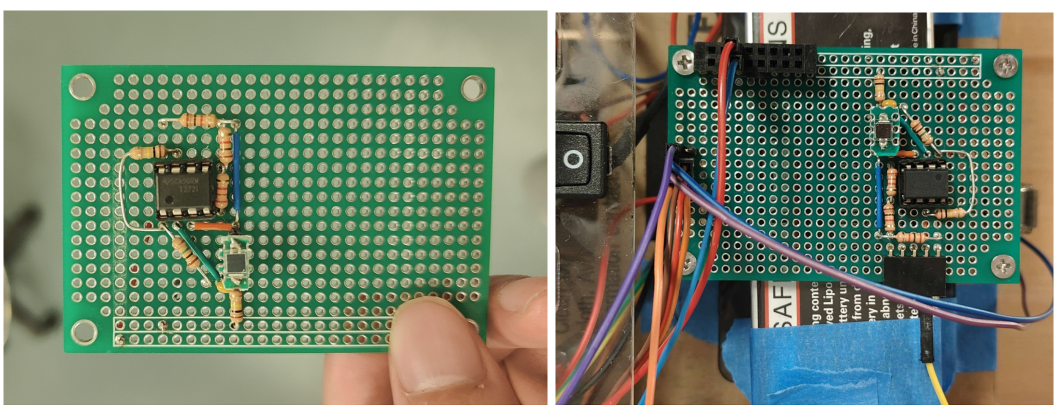
Table of Bill of Materials

The total cost of the robot is $131.5. (not accounting for items received and billed at the TA).

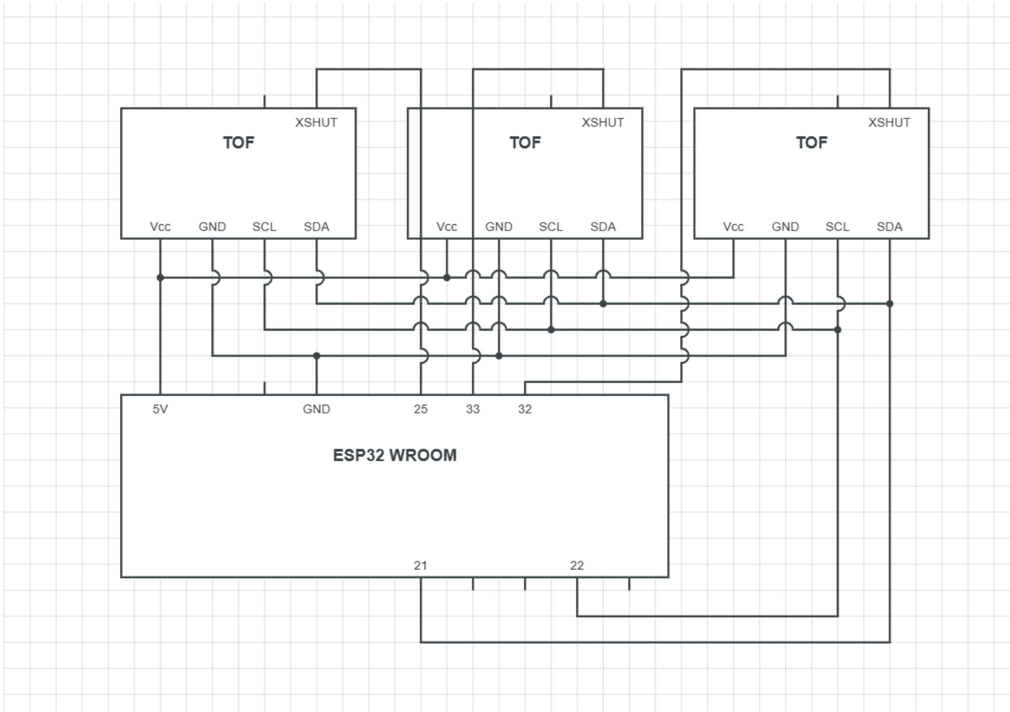
2. **Schematics of all electronic circuits**



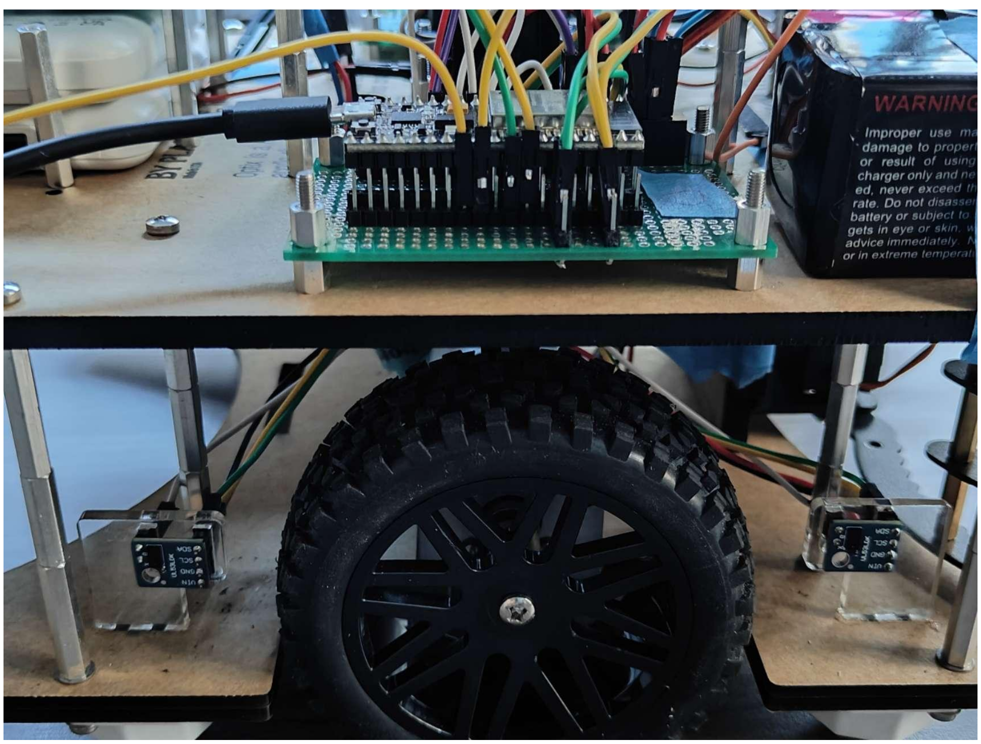
Vive Schematics



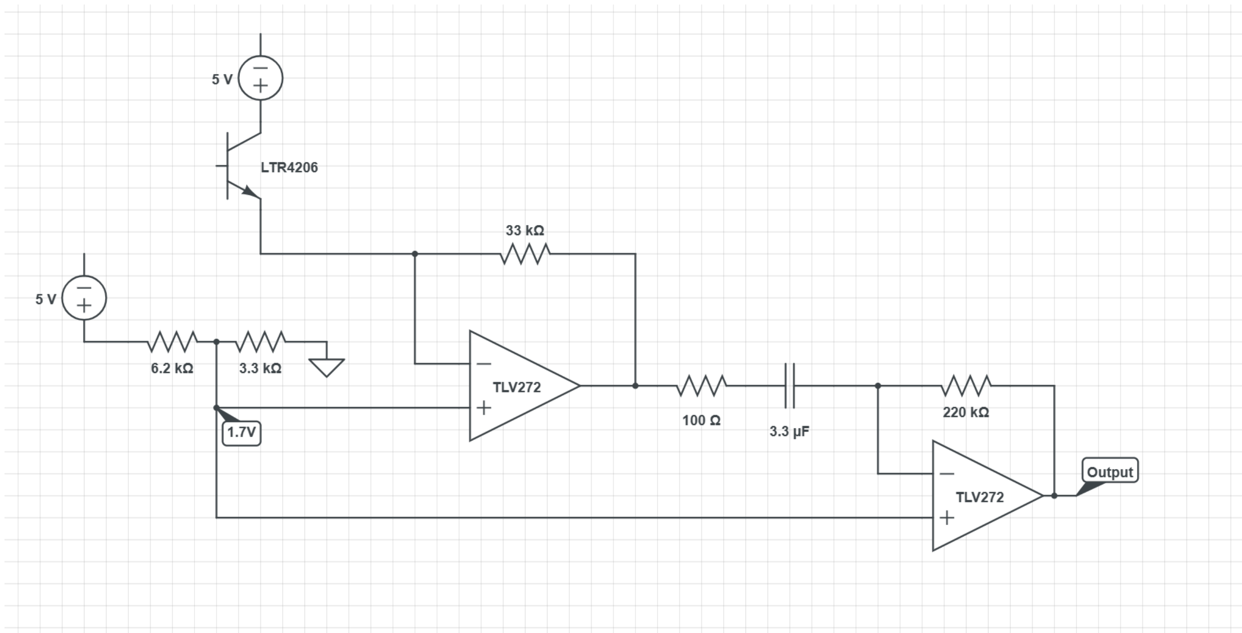
Actual Vive circuit diagram



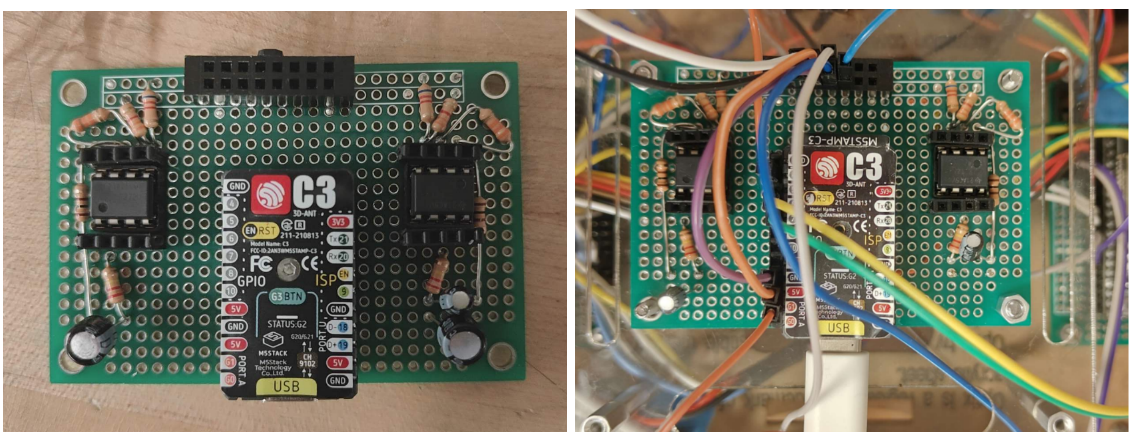
TOF Schematics



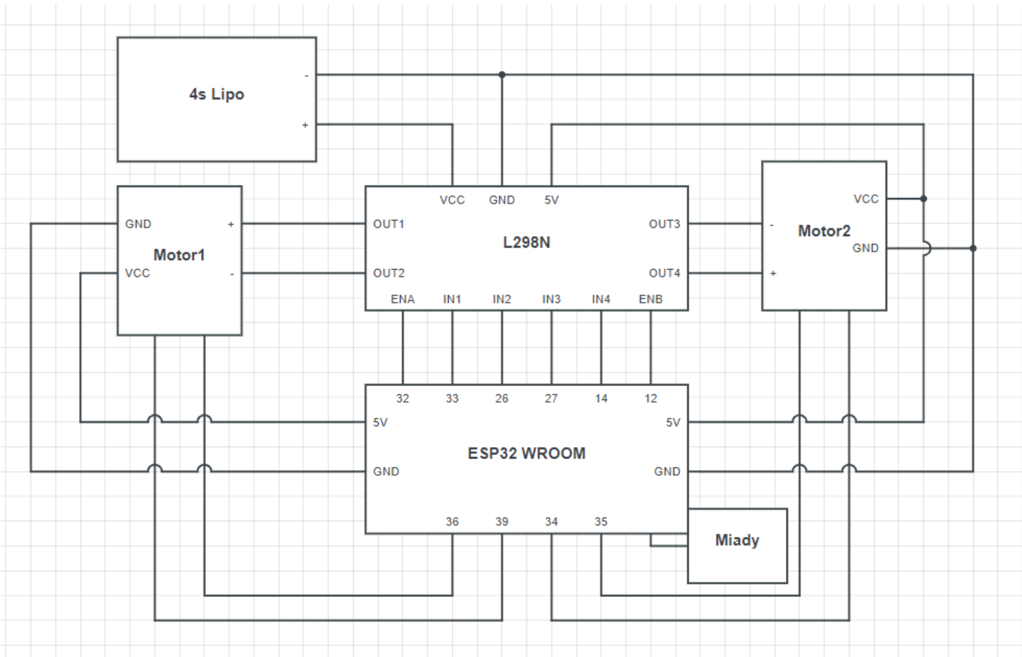
Actual TOF circuit diagram



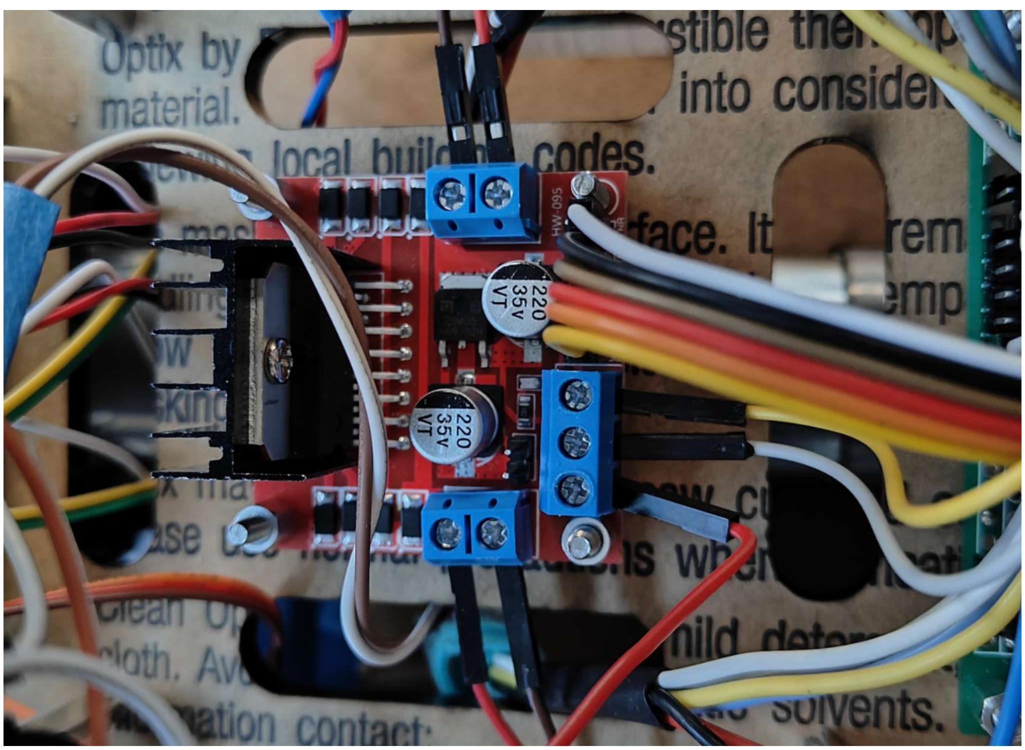
IR Schematics



Actual IR circuit diagram



L298N Schematics



Actual L298N circuit diagram

3. Nice photo of full robot

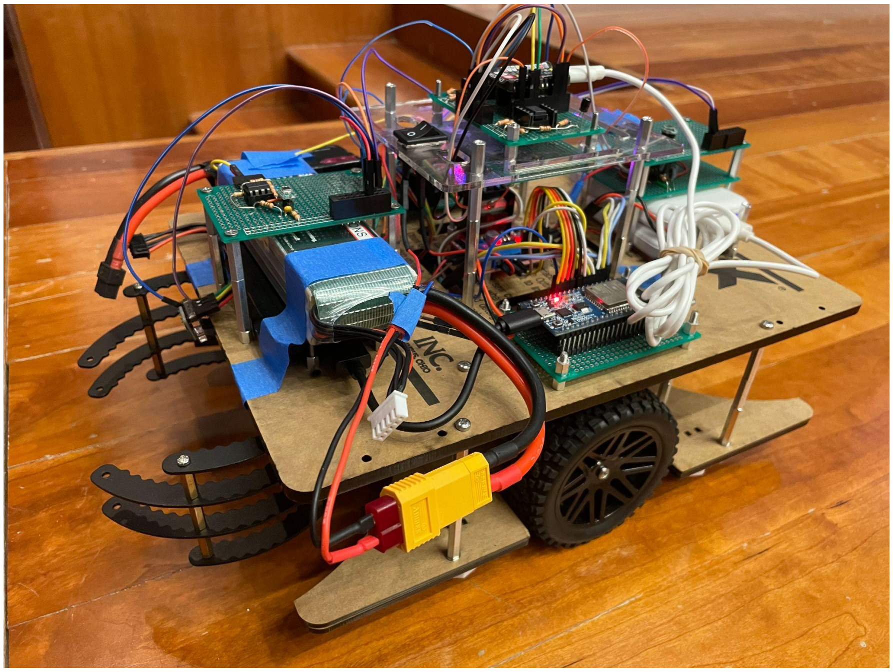
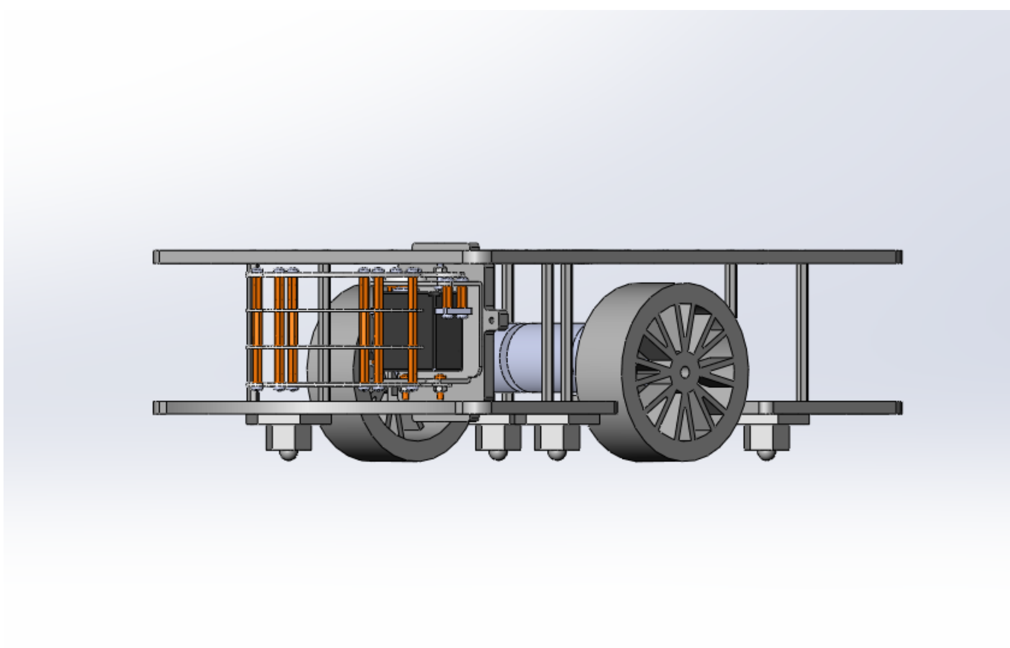
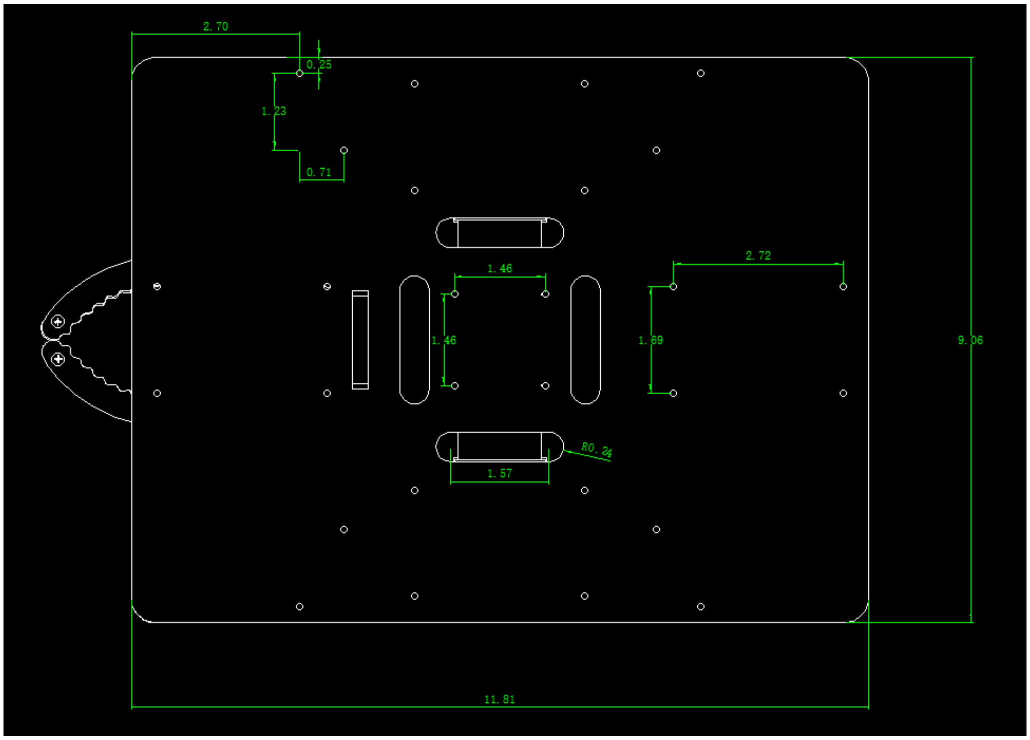


Photo of full robot

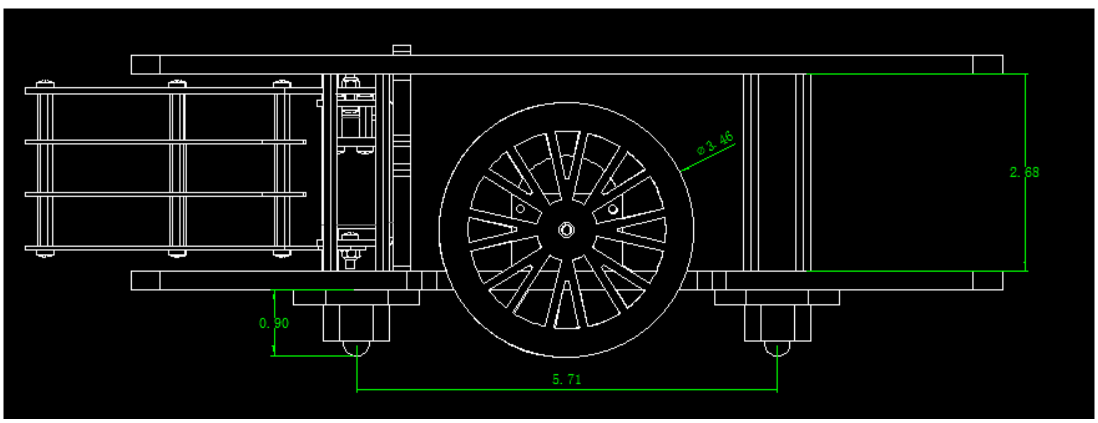
4. CAD drawings (or mechanical drawings) highlighting anything special



Solidworks Drawing of the robot



Robot Top View



Robot Front View

Solidworks model:

https://drive.google.com/drive/folders/15oBofBjnWPR8WpZsgHVasRZK9ttrEHL_

5. **All data sheets for components**
    - **Motor(CQRobot Ocean):** http://www.cqrobot.wiki/index.php/DC_Gearmotor_SKU:_CQR37D
    - **Motor Driver(L298N):** L298N-Motor-Driver-Datasheet.pdf (components101.com)
    - **TOF(VL53L0X):** Time-of-Flight ranging sensor (st.com)
    - **Gripper:** Amazon.com: Mechanical Claw BigClaw Robot Gripper (without Servo) : Toys & Games
    - **Battery(4s Lipo):** Amazon.com: HRB 4S 3300mAh Lipo Battery 60C 14.8V RC Lipo Battery Pack with XT60 Plug for RC Airplane, RC Helicopter, RC Car, RC Truck, RC Boat (EC3/Deans/TR/Tamiya) : Toys & Games
6. **Video Link**
    - **Evaluation of functionality:** https://youtu.be/02jUuSZZPI0
    - **Final competition:** https://youtu.be/HyIkRfzkrz0