TRIANGLE SIGN DEPLOYER CAR

By

Yuanfeng Niu

Yue Shi

Final Report For ECE 445, Senior Design, Spring 2024

TA: Douglas Yu

1 May 2024

Team No. 51

University of Illinois Urbana-Champaign

Abstract

This report presents the design and testing of a remotely controlled electric vehicle to enhance road safety by deploying a triangle warning sign at the emergency site. Our design aims to eliminate the risk of deployment of signs manually in high-speed traffic areas, minimizing human exposure to traffic hazards. The vehicle can travel and operate effectively within a 30-meter range from the release point. Key features include low power consumption, retractable deployment robotic arm, and remote operation via mobile application, ensuring the vehicle meets operational and safety requirements. Through this innovation, we aim to improve road safety by enabling rapid, safe, and effective warning sign deployment in critical traffic situations.

Contents

1. Introduction

Our team investigates the engineering challenges and solutions related to developing a remotely controlled electric vehicle designed for deploying an emergency warning sign on roads. The vehicle aims to reduce the risk associated with manual sign deployment in high-speed traffic areas, thus minimizing human exposure to traffic hazards. The contents of this report detail the design process, the verification of each system component, the cost analysis, and the ethical considerations we navigated. Finally, discuss potential future enhancements. This device not only mitigates the risks associated with the manual deployment of warning signs but also demonstrates significant advancements in automated traffic management technology.

1.1 Problem

The urgency of efficiently alerting oncoming traffic in the event of a traffic emergency cannot be overstated. Effective communication of hazards ahead is crucial for preventing accidents and ensuring the safety of all road users. The required distance for these warning signs can vary by jurisdiction. In some cases, regulations specify that warning signs be placed up to 100 meters away from the emergency site to provide ample warning to oncoming drivers, thereby significantly reducing the likelihood of further incidents. Given the critical importance of adequate reaction times and stopping distances in high-speed traffic conditions, the Federal Highway Administration provides comprehensive guidelines to ensure safety. These guidelines emphasize the necessity of sufficient stopping sight distances along roadways, factoring in a driver's perception-reaction time of 2.5 seconds, to mitigate the risks associated with increased speeds [1].

However, the manual deployment of these signs poses significant safety risks, particularly in conditions of high traffic volume or on roads with high-speed limits. The act of walking against incoming traffic to position a warning sign is dangerous, exposing individuals to the risk of serious injury or death. Research from the Road Safety Foundation indicates that manual sign placement in high-traffic conditions significantly increases the risk to emergency responders and roadside workers [2].

This situation highlights a critical gap in current traffic management practices, underscoring the need for a safer, more efficient, and practical solution for deploying warning signs without compromising human safety.

1.2 Solution

Our solution is a deployable, remotely controlled electric vehicle (referred to later as “car”) designed specifically for the task of carrying and deploying triangle warning signs at the required distance from a traffic emergency site. This vehicle, released at the emergency site, would be capable of traveling distances ranging from 10 to 100 meters, in accordance with the range suggested in various regulations, to deploy the warning sign accurately and safely. Our design prioritizes low power consumption, ease of storage, a manual control system, and a backup autonomous navigation system. This approach not only mitigates the risks associated with manual sign placement but also enhances the rapid deployment of essential warnings, thereby contributing to road safety and the prevention of further incidents.

2 Design

2.1 Block Diagram

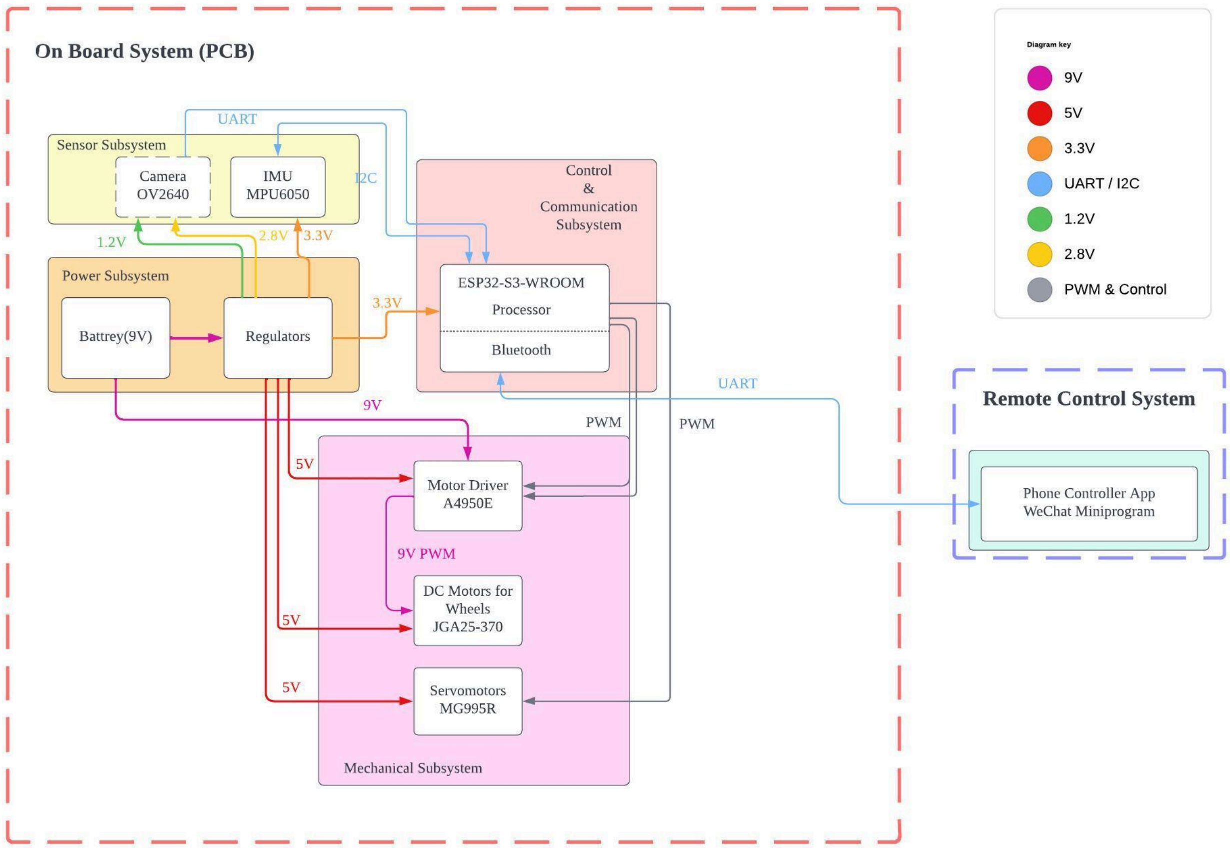

Figure 2.1. Block Diagram of Subsystems

2.2 Subsystems

2.2.1 Subsystem 1: Control

This subsystem functions as the central processing unit of the design, primarily managing data processing and BLE communication. It is built around an ESP32-S3 microchip [3], which coordinates the inputs from the Phone App and the IMU and outputs control signals to the motors and servo. The ESP32-S3 offers dual-mode Bluetooth capabilities, allowing for versatile wireless connectivity options.

The data from the IMU, including displacement vectors computed from the gyrometer and accelerometer readings, is processed through specific microchip pins using the I2C protocol. The processed data—both commands from the Phone App and displacement information from the IMU—are then transmitted through the microchip’s internal Bluetooth antenna, ensuring continuous and real-time communication with the user's phone.

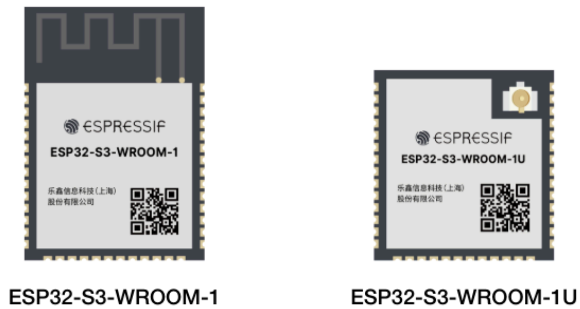

Figure 2.2.1. Top View of ESP32S3 Microchip

2.2.2 Subsystem 2: Communication

This subsystem handles data transmission and receives control inputs from the user. Due to the complexities and potential high costs of operating a fully autonomous vehicle in variable traffic conditions, we primarily utilize human control. The subsystem includes a Bluetooth module integrated into the ESP32 microchip that supports dual-mode BLE communication and can handle multiple connections simultaneously. For Bluetooth communication, we utilize the NimBLE Library, which significantly reduces resource usage and enhances the performance of ESP32 BLE applications. This library also eliminates the need for a soft device, enabling full debugging capabilities and more efficient resource management. For better communication in an open environment, we may consider adding a signal amplifier in future PCB revisions to extend the communication range up to 100 meters.

To calculate the effective transmission distance of the mounted Bluetooth antenna, we use the Friis transmission equation, which takes into account the transmit power, receiver sensitivity, frequency, and path loss:

$$Pr = Pt + Gt + Gr - L \tag{1.1}$$

Where Pr is the received power at the receiver (in dBm), Pt is the transmitted power at the transmitter (in dBm), Gt is the gain of the transmitter antenna (in dB), Gr is the gain of the receiver antenna (in dB), and L is the path loss (in dB).

The equation for path loss:

$$L = 20\log10(d) + 20\log10(f) + 20\log10(4\pi/c) \tag{1.2}$$

Where d is transmission distance, f is operating frequency, and the last term is a constant. Receiver sensitivity can be amplified without consequences, however, transmitter sensitivity is limited to its sensitivity labeled in the datasheet. In the datasheet provided by Espressif Systems(manufacturer), the antenna can send a signal with a maximum of +20 dBm power. This value is the combined effect of Pt, Gt, and $20\log10(f)+20\log10(4\pi/c)$ terms of the Friis equation. The only term left is the distance-related term, which attenuates signals by -40 dB at the receiver end. This results in a -20 dBm power at the receiver end, which is a lot higher than typical receiver sensitivity(~-80 dB). Therefore, transmission of Bluetooth signals at ~100m is theoretically possible.

2.2.3 Subsystem 3: Power

This subsystem is responsible for powering all other subsystems, consisting of a 9-12 V battery and a set of voltage regulators This system requires four different voltages: 3.3 V, 5 V, 1.2 V, and 2.8 V. The 5 V is supported by LM2596-5.0, a highly efficient switch-mode regulator. This choice benefits from its ability to handle larger current loads and maintain stability, especially when paired with a 33 µH inductor which improves its response to changes in load and input current.

By using the 5V output from the LM2596 as a common input for 3.3 V, 1.2 V, and 2.8 V regulators, our design simplifies the power architecture and minimizes the number of power sources. This setup ensures all subsystems receive a consistent supply of power, allowing for easier adjustments and expansions in future development.

Originally, 1.2 V and 2.8 V voltages were to be regulated using XC6206 series regulators to support a camera module. However, with the removal of the camera module from the project, these voltage levels are no longer required.

For the 3.3 V voltage support, which powers the sensor and microprocessor, we continue to rely on the AZ1117- 3.3 regulator. The choice of a 9 V battery as the power source remains unchanged, providing sufficient capacity to drive these regulators efficiently.

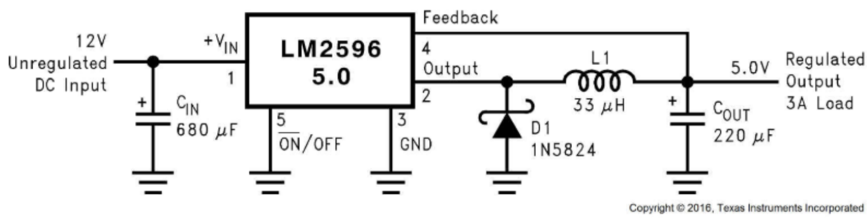

Figure 2.2.3. Circuit Diagram of LM2596-5.0

2.2.4 Subsystem 4: Sensor

This subsystem feeds data to the ESP32 microchip to assist with both human-controlled and autonomous deployment, now solely relying on an Inertial Measurement Unit. The previously planned OV2640 camera module has been removed due to incomplete interfacing work by Chaoyang. Now, the focus is on the MPU6050 series IMU, which measures angular displacements with a gyrometer and acceleration with an accelerometer.

Data from the MPU6050 is transmitted to the ESP32 chip via the I2C protocol. the incoming data requires decoding through driver packages and selective parsing to retain only the relevant gyrometer and accelerometer information. This streamlined approach ensures efficient processing and simplifies the integration into the vehicle's control system.

2.2.5 Subsystem 5: Mechanicals

This subsystem is responsible for all movement mechanisms of the deployer car, incorporating wheel motors, motor drivers, and servos. It features two JGA25-370 motors, each driven by an A4950E motor driver. The A4950E offers several advantages, including robust output performance, integrated protection features such as overcurrent and thermal shutdown, and a high degree of efficiency which is crucial for battery-operated vehicles.

For sign deployment, a single large-torque servo is employed to lift the normal triangle warning sign, controlled via PWM signals from the ESP32's GPIO pins. This setup simplifies the mechanical design while ensuring reliable operation during the sign-lifting process.

The motors, with a gear ratio of 1:9.6, are specifically chosen for their ability to provide significant force at lower speeds, essential for controlled and stable movement. The selected motors operate at an output of 900 RPM at 12 V, achieving an optimal speed of 1.88 m/s and a torque of 1.0 kg · cm when at stall current, which is 1.2 A.

The car chassis for the deployer car is a robot tank measuring 270 mm long, 195 mm wide, and 95 mm high, and weighs 1.0 kg. Its shock absorption feature ensures stable movement across different terrains, protecting the internal components.

$$Torque\ (N{\cdot}m)\ =\ Torque\ (kg{\cdot}cm) \times g \times 0.01\ =\ 1.0\ kg{\cdot}cm \times 9.8\ m/s^2 \times 0.01\ =\ 0.098\ N{\cdot}m$$
(2.1)

$$\mu_s\ =\ 0.1\ (a\ rough\ estimate\ for\ hard\ surface)$$
(2.2)

$$F_{friction}\ =\ F_{normal} \times \mu_s\ =\ m \times g \times \mu s\ =\ 1.0\ kg \times 9.81\ m/s^2 \times 0.1\ =\ 0.981\ N$$
(2.3)

$$F_{wheel} = \tau\ /\ r\ =\ 0.098\ N{\cdot}m\ /\ 0.02\ m\ =\ 4.91\ N$$
(2.4)

According to the calculation, the total force (1.4) available from the motors is significantly greater than the frictional force (1.3), the motors are capable of driving the tank on a hard surface.

Figure 2.2.5. Tank Chassis Diagram

2.2.6 Subsystem 6: Remote User Interface

This subsystem serves as the car's primary control method, favoring human input over autonomous navigation due to the variability of traffic conditions and the limitations of embedding complex algorithms into a standalone microchip. Instead of a custom-developed Android application, control is now managed through a WeChat mini-program. This mini-program communicates with the car's systems via Bluetooth Low Energy (BLE), sending 8-bit hex commands to the microprocessor. Upon receiving these commands, the microprocessor decodes and executes them, directing the car's actions.

3. Design Verification

Our project is a rather complex design with six total subsystems. On top of that, customizing PCB also adds work to the verification process.

A Requirement & Verification table can be found in the appendix, featuring the requirements and their verification methods, and the outcomes in our trials.

3.1 Microchip

The microchip itself, without any circuitry, cannot be debugged. Fearing that a wrong circuitry may burn the chip, we started our testing process by first verifying using an evaluation board.

Finding no specific instructions other than strapping pins in the datasheet, we randomly assigned the remaining pins for connections to the remaining modules.

Interfacing of the ESP32 program, as well as development of its program, is done in Arduino IDE. It provides supporting libraries and examples for ESP32-S3, and also has a serial monitor for debugging visibility. With those, it took us less than a week's work to initialize and interface with all the modules (except for the camera).

3.2 Bluetooth

ESP32 has innate Bluetooth libraries, but they appear quite confusing, as ESP32-S2 and ESP32-S3 use different BlueTooth codecs and both appear in the examples list in Arduino IDE.

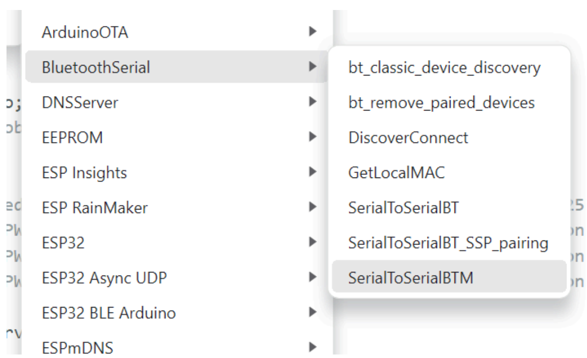

Figure 3.1 Arduino IDE's Example lists. Both BluetoothSerial and ESP32 BLE appear in the examples list, and it turned out that BluetoothSerial is the pack of unamended examples.

The device appears in the device list in our phones after successfully flashing in the program and resetting. It persists in the device list as we move it afar, however we can no longer establish/maintain paired status after around 10m.

3.3 Power

Verification on the regulators is rather simple: observe on a multimeter whether they output voltages as labeled, and whether currents exceed the maxima.

We chose to cascade 3.3V, 1.2V, and 2.8V linear regulators after 5V switching regulators to improve the efficiency of the linear ones, as their efficiency is directly proportional to voltage drop. This 5V switching regulator, therefore, has to have a very high current rating, as it supplies all the parts the above regulators are connected to in addition to motors. LM2596 is such a packaging that fulfilled this need.

It turned out that Digikey sent the wrong package of the 2.8V regulator, and it turned out to be in fact 4.6V. Aside from that one, all other requirements are met.

3.4 Sensors

Sensors are the modules that are tested at the very end (of design phase).

Prior to transplanting the PCB, we tested interfacing of MPU6050 with ESP32 using evaluation kits. With the help of custom libraries, it took less than anticipated time span for us to read the correct data off of the serial monitor in Arduino IDE.

Unfortunately, no evaluation kits of the OV2640 camera were found, and the supporting software was not turned in timely. Therefore we abandoned this part.

3.5 Mechanicals

The mechanical parts were also tested very late as we received our updated version of PCB almost at the last minute.

In the verification process, initially, we connected the motors to the power supply, double-checking and measuring the on-load and stall current of one motor. Upon powering the system, we confirmed that the motor's actual current consumption was within ±5% of the designed value.

Furthermore, we connected the motor to the motor driver and fed the control signal from the App to the motor's encoder, the motor on both sides responded correctly. For the servo motor, we test it by programed PWM signals. The servo's performance was scrutinized, and it successfully reached a maintained 90-degree angle.

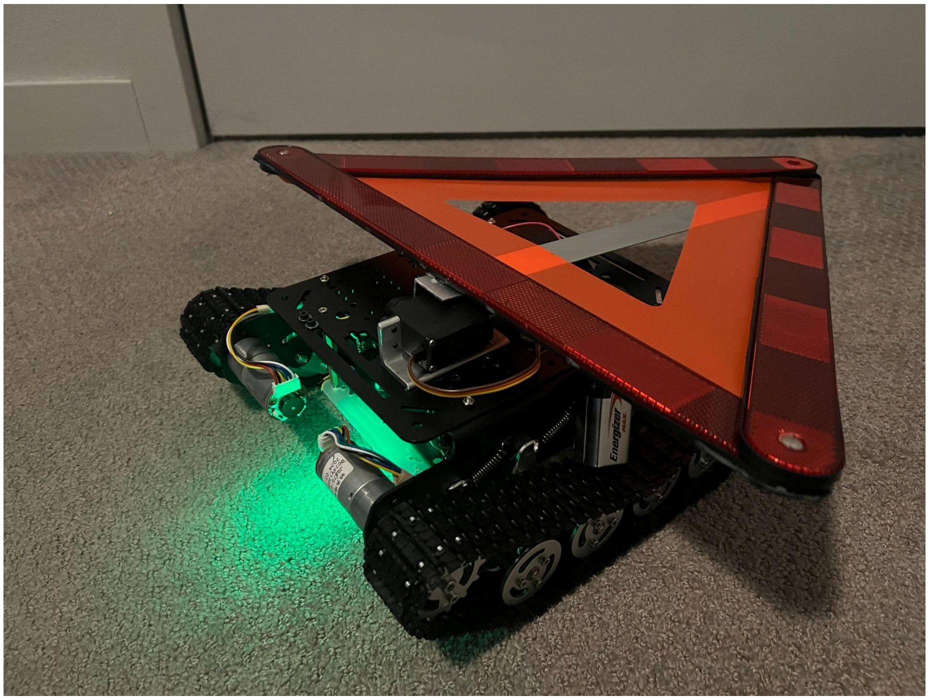

Figure 3.3 Fully assembled car chassis, with motors and a servo.

3.6 Remote User Interface

Initially we proposed developing a standalone Android application for sending instructions. Due to the software specialist's refusal to report progress, we switched to WeChat mini-programs. These are readily available programs, which significantly reduce development difficulty.

Despite being unable to meet the specified GUI, it does meet the core functionality of such an app: it can establish connections with an active ESP32 microchip and can send single-byte instructions as HEX codes.

After repeating some close proximity transception tests at long ranges, it is found that although the bluetooth signal of the ESP32 may appear in the device list, a pairing connection cannot be established. We suspect that this is caused by the low transmitting power of the phone, but that cause is yet to be confirmed.

3.7 PCB

When designing the PCB, we intentionally avoided the use of pin sockets for any evaluation boards or standalone PCBs. To accomplish that, we looked through examples of

Verification of some modules depending on the proper operation of others, therefore we began by partially soldering parts onto the PCB to test high-priority modules first. This includes the power distribution and the ESP32 chip.

As we populated the ESP32 area, we found that a critical error in strapping GPIO wiring had led to the chip being unable to reset, and will not initiate program execution forever. We rewired the pins and ordered another revision.

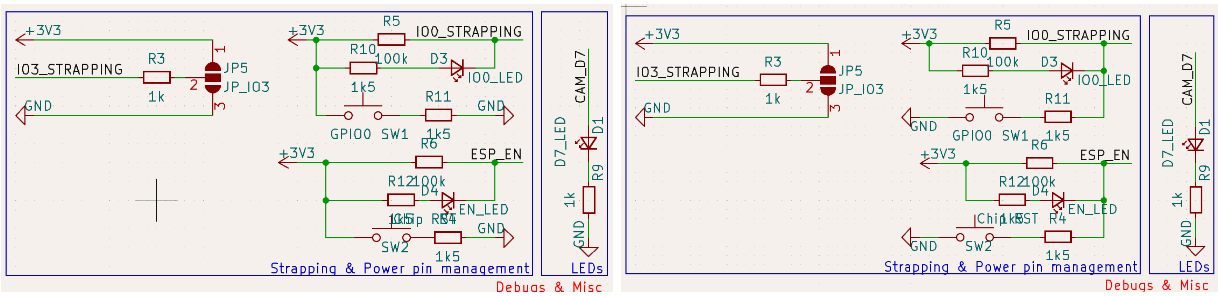

Fig 3.2 ESP32 strapping pin connections in the schematic, before and after spotting the errors.

The second revision was still erroneous, fortunately, a temporary solution was found by applying pin headers to switch masks, and manually pulling up/down the correct nets. The following is the ESP32 region on a populated PCB that incorporated this.

Figure 3.3 Modified PCB. Notice the white pin headers. In practice, these headers are grounded to mimic a button press.

4. Costs

4.1 Parts

Description	Manufacturer	Quantity	Price
ESP32-S3-WROOM-1-N16 / RF TXRX MODULE BT PCB TRACE SMD	Espressif Systems	1	$3.48
AZ1117-3.3 / IC REG LINEAR 3.3V 1A SOT223	Diodes Incorporated	1	$0.38
XC6206P282PR-GTR-ND 150MA SOT89	Torex Semiconductor Ltd	1	$0.24
XC6206P122PR-GTR-ND 60MA SOT89	Torex Semiconductor Ltd	1	$0.28
LM2596S-5.0/NOPB 3A TO-263	Texas instruments	1	$6.96
MPU6050 / IMU ACCEL/GYRO 3-AXIS I2C 24QFN	TDK InvenSense	1	$3.00
JGA25-370 / DC Gearmotor with Encoder	PrUva	2	$35.76
CH340N / USB to TTL Serial Port Converter Module	Generic	1	$3.43
A4950 / DMOS Full-Bridge Motor Driver IC	Allegro MicroSystems	2	$ 5.72
Servomotor 1142 / Servo Motor RC 5V High Torque	Adafruit Industries LLC	1	$19.95
Car Chassis	YCKJ	1	$66.99
			Total Cost = $146.19

4.2 Labor

Cost in this section is approximated with estimated working hours per week times the number of weeks, plus any additional hours.

On average, group members meet and work 6 hours per week. Using the minimum wage in Illinois as an estimate, the total labor cost will be:

Total Cost = 6 Hours/week * 11 Weeks * 14 $/Hour*person * 2 persons = 1,848 $

5. Conclusion

5.1 Accomplishments

By the time of the final demonstration, our team managed to produce a moving toy car with an almost fully populated PCB and a controller application.

1. Successful integration of subsystems, except the camera module and auto-navigation algorithm.
2. Remote operation via mobile app.
3. Efficient power management, running the car on a 9V battery.
4. Fully populated PCB, except the OV2640 camera module.

5.2 Uncertainties

Several factors led to a latency in completing certain objectives in the work contract.

Parts orders were placed on different platforms, including the departmental supply center, Digikey, PCBWay and JLCPCB. Due to high costs, we chose the slowest shipping method on Digikey and PCBWay for the first few batches of orders. That led to them arriving later than anticipated.

ESP32 microchips also turn out to be much more difficult to use, compared to the other two microchip variants available in the departmental supply center. They require complex wiring to control the behavior of system pins, while still requiring UART connection pins, and are in general harder to learn compared to STM32/ATMega series microchips.

We also made quite a few errors in designing the PCB. It's common to make mistakes for first-time PCB designers, and it took us several revisions to fix wiring errors surrounding the ESP32 and MPU6050.

In addition to those above, Chaoyang Yin, originally one of our team members, did not contribute to the project's progression. By the time of the final demonstration, he effectively completed one item specified in the original team labor contract, and more than a month after its original due date. That item was a crucial part of the software development portion of this project, and even after redistributing his work to other members, the progression of the software portion was still significantly hindered. We ended up finding a substitute BT tester program, and cut off the camera module due to simpler BT programs being unable to display video format data.

5.3 Ethical considerations

5.3.1 Public Safety and Responsibility

Central to our project is the imperative to "hold paramount the safety, health, and welfare of the public" as per IEEE's Code of Ethics Section I.1 [4]. Our system is designed with the foremost goal of not endangering human lives or causing unnecessary disruptions in traffic flow. Rigorous testing and validation will be conducted in diverse settings to minimize the hazard of the vehicle's unsafe or unreliable operation. However, it is important to acknowledge that while we strive to mitigate as many risks as possible, not all risks can be completely eliminated. Continuous user caution will be required at all times during operation to ensure the highest level of safety.

5.3.2 Privacy Respect and Data Security

In accordance with ethical standards, our design respects individual privacy and is committed to not collecting or transmitting sensitive information without explicit consent. This commitment to privacy and data security aligns closely with the ACM Code of Ethics, which emphasizes "the importance of privacy and confidentiality in technological advancements." Specifically, the ACM Code of Ethics states, "Design and implement systems that are robustly and usably secure, ensuring the privacy and proper use of personal information obtained in the course of their work" [5]. In line with this principle, our vehicle not only employs strict access controls to protect operational and communication data but also ensures that sensitive sensor data is erased after each operation. This approach minimizes any risk of unauthorized access or misuse of information, adhering to the highest standards of data privacy and security.

5.4 Future work

We considered two directions as possible improvements to continue development:

The mechanical structure used to deploy and retract the sign contains only a servo, and the sign is adhered to the rod on the servo. We can overhaul this to place multiple signs on the ground. This can improve the effectiveness of such cars and open up room for repurposing, which might prove useful in other specialized application scenarios (such as delivering traffic cones for civil engineering services.)

The Bluetooth antenna in this version of ESP32 is low-power-oriented and quickly loses communication over distances. A specialized Bluetooth transceiver module might suit this task better. Some ESP32 variants have customizable antenna mounts, and a narrow-band antenna array can easily outperform the current antenna configuration.

References

[1] Federal Highway Administration, "*Speed Concepts: Informational Guide,*" U.S. Department of Transportation, FHWA-SA-12-004, 2012. Available: https://safety.fhwa.dot.gov/speedmgt/ref_mats/fhwasa12004/. [Accessed: Feb. 22, 2024].

[2] Road Safety Foundation. (2023, December 13). *Road safety performance and investment opportunities.* Road Safety Foundation. https://roadsafetyfoundation.org/road-safety-performance-and-investment-opportunities/

[3] Espressif Systems. (2023). *"ESP32-S3-WROOM-1 & ESP32-S3-WROOM-1U Datasheet."* [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf

[4] IEEE. (2023). *"IEEE Code of Ethics."* [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html

[5] Association for Computing Machinery (ACM). (2023). *"ACM Code of Ethics and Professional Conduct."* [Online]. Available: https://www.acm.org/code-of-ethics

[6] "MPU-6000 Datasheet," TDK Corporation, [Online]. Available: https://product.tdk.com/system/files/dam/doc/product/sensor/mortion-inertial/imu/data_sheet/mpu-6000-datasheet1.pdf.

Appendix A - Requirement and Verification Table

Table 1 System Requirements and Verifications

Subsystem	Requirement	Verification	Status (Y or N or N/A)
Microchip	The chip can flash programs and reset. The chip can flash programs and reset.	Plug in the microUSB-B cable and connect it to a workstation. A new device occupying one COM port should pop up in the device manager. Launch Arduino IDE, and verify that there is a programmable COM port device. Upload the program, and wait until the line "Resetting via RTS pin" shows up.	Y
	The Bluetooth peripheral can be initialized.	Using the serial monitor within the Arduino IDE, wait for the message "BLE initialized!"	Y
	The motors can be initialized.	Using the serial monitor within the Arduino IDE, wait for the message "Motor setup complete!/ Servo setup complete!"	Y
	The sensors can be initialized.	Using the serial monitor within the Arduino IDE, wait for the message "MPU6050 detected!"	Y
Bluetooth	The connection between the phone app and ESP32 is established.	Reset the PCB. The device name "Team51" should appear in the BLE device list. Open the app	Y
	The above connection is maintained at 10m, 25m, 100m.	Find an open ground and walk the respective distance away from the car. Using the phone app, check the connectivity.	10m:Y 25m:Y 100m:N
	The camera feed should be readable at 100m	Inspect the camera feed using the phone app at ~100m away.	N/A
Power	Fixed regulators output Δ5% within base voltage ratings	After soldering all the pins onto the PCB, ensure that JP1 is connected, then connect the battery to the board power input. Probe respective TPs with the multimeter and read out voltage.	1.2V:Y 3.3V:Y 5V:Y 2.8V:N
	Maximum current does not exceed each regulator's label	Change the multimeter type to DCI. Probe the Vin pins and Earth pins of all regulators. Read out current.	Y
Sensors	The camera module interfaces with ESP32 properly.	(Not verifiable since phone app does not support video decoding)	N/A
	The camera feed is consistent.		N/A
	The IMU module interfaces with ESP32 properly.	Probe SDA pin and set oscilloscope to detect I2C envelopes. Read out the data feed on the serial monitor in Arduino IDE. (Note that constant 0s for all accelerations & rotations and a fixed temperature is	Y

		a WRONG form of output, an indicator of a short)	
	Data from the IMU module is within 10% of actual values.	Turn the car chassis around and observe gyroscope data patterns. Borrow the force meter equipment from a Physics student; pull the car with the force meter, calculate acceleration, then compare with data patterns.	Y
Mechanicals	The motors can move, and their maximum current should fall within a safe value.	Connect the motor to the power supply. Ensure all connections are secure and correct according to the motor's wiring diagram. Configure the multimeter to measure current and ensure it's in series with the motor circuit to measure the stall current accurately. Safely secure the rotor or apply a load that prevents the motor from turning. Turn on the power supply to apply the rated voltage to the driver chip and the motor. Read the change of current supply from the power source. Confirm the actual current of the motor is within a tolerance of ±5% of the designed current value.	Y
	The motors can move in accordance with phone app instructions.	With the phone app connected, send “Forward”, “Backward”, “Left”, and “Right” instructions. Motors on both sides should move accordingly.	Y
	The servo motor can reach the designated angle with a tolerance of ±10%.	Execute the program. After establishing BLE connections using the phone app, send a “Deploy” signal. Confirm the actual angle of the motor is within a tolerance of ±10% of the designed value. Send a “Retract” signal. Confirm the angle falling within ±10% of the designed value.	Y
Android App	The phone app has a graphical interface as the design document specified.	Compare the GUI to that suggested in the design document.	N*
	The app can send HEX instructions and receive data from ESP32.	Attempt to connect the Android app to the ESP32 microchip. After establishing connections, send HEX instruction codes as defined in the Arduino program, and verify with the serial monitor.	Y
	The above can be repeated at ~100m.	Repeat at ~100m.	N/A

*Chaoyang turned in a prototype program 2 days before the demo that technically has a customized interface. It deviates from the design document's specified layout a bit, however.

Appendix B - Diagram of the Entire PCB Design

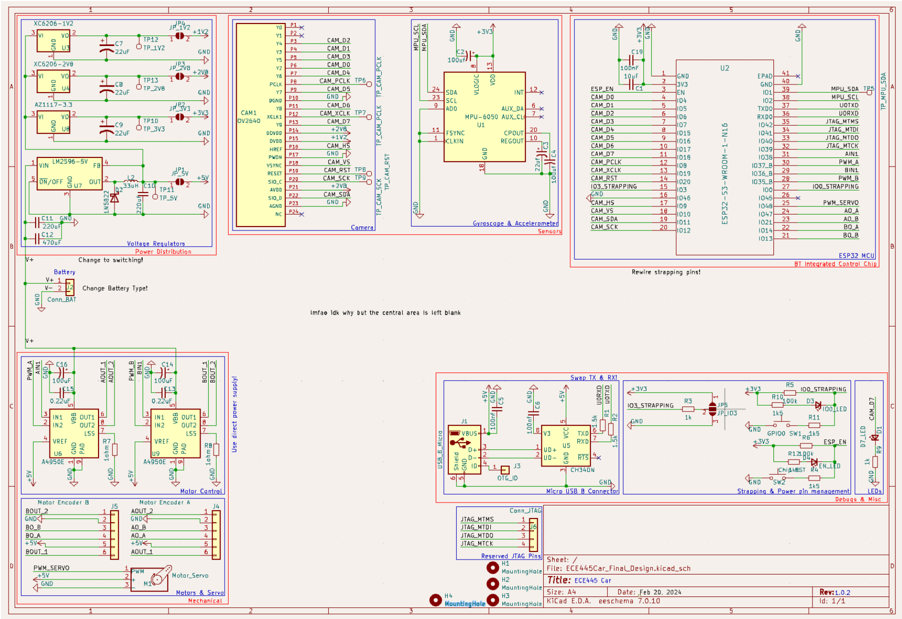

Figure B.1. Schematic of PCB

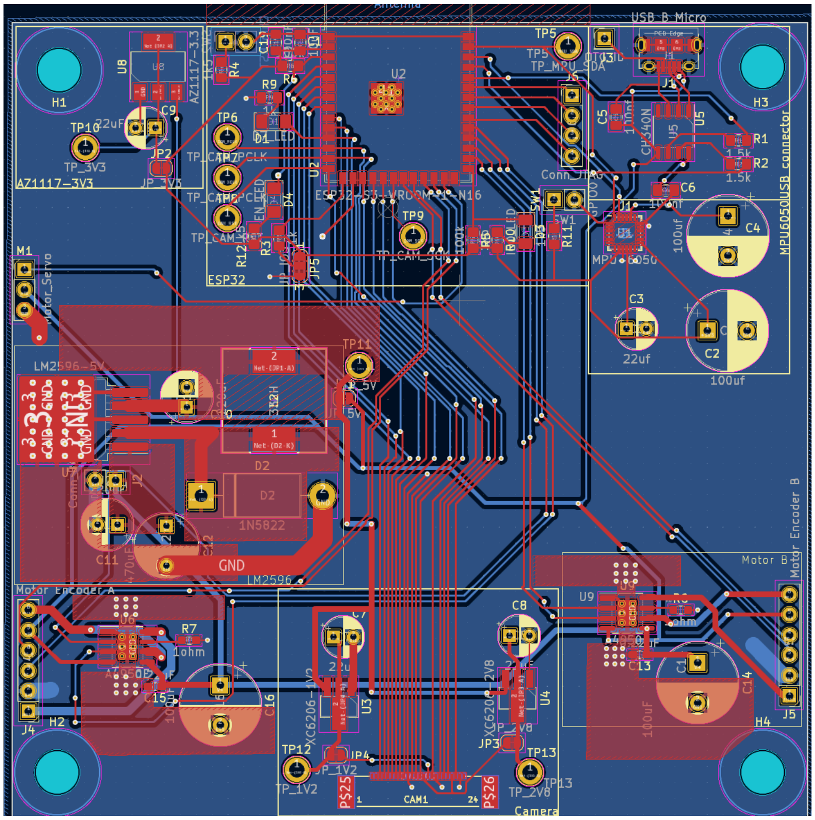

Figure B.1. Layers of PCB