Final Project

Report submitted in partial fulfillment of the requirements for the course

Introduction to Engineering and Computer Science (CECS1010)

by

Group 09

COLLEGE OF ENGINEERING AND COMPUTER SCIENCE

VINUNIVERSITY

FALL 2022

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Project Background

The final project of the course CECS1010, Introduction to Engineering and Computer Science requires us to create an autonomous car by assembling the components of the car and programming the line and sign detection in Python. Before starting the project, we have learned about how the hardware interacts with software, the basic terms of programming in the course COMP1010, an Introduction to Programming, and some fundamental frameworks of project management, this final project intends to help us apply what we have learned in semester Fall 22. In our team, we have three computer science students, who are responsible for the software parts of the project, one electrical engineering student, and one mechanical engineering student, who account for the hardware parts of the project.

1.2 Project Definition

The target outcome of this final project is an autonomous car that can detect lines and traffic signs so that it can complete the race on CECS day, the day on which CECS students of VinUniversity show the results of their project. The car is assembled by a frame, two motors, one battery, the mainboard, and the camera, which helps the car detect the image and predict the next actions. The project is essential for the development of self-driving automobiles since it will determine if autonomous vehicles are feasible and what problems they will encounter.

1.3 Project Objectives

1. Design and implement a control system for an autonomous vehicle that can navigate a designated test course.
2. Develop and implement computer vision algorithms for object detection and avoidance, lane keeping.
3. Incorporate sensors such as cameras provide real-time feedback to the control system.
4. Work with a multidisciplinary team to complete the project, incorporating electrical engineering, mechanical engineering, and computer science principles.
5. Present the project results to a broader audience, including faculty, peers, and potential employers.
6. Document the design, implementation, and testing process, including results, lessons learned, and recommendations for future work.

1.4 Project Specifications

The project specifications of our team are described in the Table 1.1 below.

Table 1.1: Requirement specification

No	Category	User requirements	Engineering specification
1	Functional requirements		
1.1	Control computer	• The self-driving car can send the image through the camera onboard • Detect the lane, traffic signs, and obstacles. • Follow the instructed command of the users	• Send instructions and receive images through WiFi • Use image segmentation to identify the lane in the pictures. • Use object detection to recognize barriers and stop, left, right, and straight signs. • Determine the car's speed and direction while adhering to traffic signs and obstacles.
1.2	Self-driving car model	• Capture images and send them to the car • Receive the commands from the computers	• Camera that can record the moment and send it to the computers through WIFI • The code of the car doesn't have any bugs so it can smoothly understand the command from the computers
2	Non-Functional requirements		
2.1	Performance		

2.1.1	High level of accuracy	• The self-driving car needs to be able to detect roads and obstacles accurately • The car needs to be able to move accordingly to the detected objects	• The number of inaccurate movements measured • The magnitude of the errors for each inaccuracy movement
2.1.2	Fast, efficient	• The self-driving car should be quick, efficient, and time-saving • It needs to detect, processes the data quickly, and move fast	• The runtime of the program • The interval time between each data transmission • The total time it takes for the car to finish the track
2.2	Reliability		
2.2.1	It needs to work every time	• The car needs to work the same way every time we run it • It should not fail or makes different movements for the same track	• The number of failed runs • How similar each run of the car is when matched together
2.2.2	It needs to be safe to run	• The car should be able to run safely with a low chance of it being broken	• The number of times it got broken compared to the total number of times it doesn't break • How long it takes before the car started to not working
2.3	Durability		

2.3.1	It is a measure of product life	The car needs to have good engines for its performance in duties as long as possible	- It is how long a car performs its duties before its performance degrades (maybe 20 years) - Constructive system and its materials do not exhibit significant deteriorations that imply loss of functionality for which they were designed.
2.3.2	Money and materials saving	It mean reducing maintenance, repair costs, and also environmental impact	- Longer product life spans can contribute to eco-efficiency and sufficiency, thus slowing consumption to progress towards a sustainable level of consumption - It is how much money and materials we spend to maintain the systems or rebuild them.
2.4	Serviceability		
2.4.1	Corrective and preventive maintenance	If there are any problems, they should be repaired and adjusted in a short period.	- If the problems are related to physical equipment, they should be replaced or fixed in less than 10 minutes - If the problems are related to the program, the coder should debug within 15 minutes.

2.5	Confirmance		
2.5.1	The ability of the products	• It is qualified and reliable for users to believe in based on established standards.	• It is usually measured within an acceptable range of tolerance thus a product that meets the acceptable range has a high quality of conformance. • The product or service is measured near the outer boundary that has been established for performance
2.5.2	Services and Process	• The services and processes are reliable and they meet customer needs for the products.	• They track the persistence of the product, service, or process in terms of quality to make sure that it conforms to the standard.
2.6	Aesthetics		
2.6.1	Minimal in appearance but effective in the system	• The cars should be designed to look sporty and luxurious	• Aerodynamic curvature of the body, especially a rounded front • Covered wheels
2.7	Others (Power consumption, cost)		
2.7.1	Power consumption	• the vehicle doesn't require a lot of energy to run	• Low power input enhances the car's • Battery longevity, distance, and stability. it also avoids overheating the battery.

2.7.2	Cost	• The car operates at a low cost	• Using sustainable energy to generate electricity to reduce the cost of operation.
2.7.3	Eco-friendliness	• the car is eco-friendly	• The car operates on electricity and therefore doesn't produce greenhouse gas.

Chapter 2

Project Management

2.1 Project Plan

The project is divided into four main checkpoints.

1. Put all the pieces on the car's frame and drive it with the arrow keys.
2. Has a cover for the car so that it appears to be a car.
3. Automatic lane signs detection and movement
4. Finish the race (going from the starting point to the finish point)

2.2 Contribution of Team Members

There are five CECS students in my team, including:

1. Project Manager: Truong Gia Bao
   Bao is the person who distributed the jobs, set up meetings, and kept track of the progress of work.
2. Software Engineer: Thai Ba Hung
   Hung is responsible for reading and enhancing the algorithms, and codes used to run the car.
3. Electrical Engineer: Le Viet Hai
   Hai's roles are designing and developing electrical systems, and integration of sensors and cameras.

4. Mechanical Engineer: Nguyen Tran Hai Dang
   Dang's responsibilities are Collaborating with other engineers to integrate the mechanical components with the electrical and computer systems, Ensuring the mechanical components are efficient, reliable, and durable.

5. Technical Writer: Le Nguyen Gia Binh
   Binh plays a crucial role in our project. He assists in preparing presentations, proposals, and other promotional materials, ensuring that all project documentation is organized and easy to access.

2.3 Project Execution Monitoring

The figures 2.1, 2.2, 2.3 below shows our team members' responsibilities in detail.

Figure 2.1: RACI Matrix

RACI MATRIX					
AUTONOMOUS VEHICLE					
ROLES	- Project Manager - Truong Gia Bao	- Electrical Engineer - Le Viet Hai	- Mechanical Engineer - Nguyen Tran Hai Dang	- Software Engineer - Le Nguyen Gia Binh	- Software Engineer - Thai Ba Hung
Project deliverables	Project Team				
Phase 1: Hardware and Design					
Model of the car	A	I	R	I	I
Camera	A	R	C	I	I
Battery	A	R	C	I	I
Gears	I	A	R	I	I
Motors, sensors	A	R	C	I	I
Phase 2: Software construction					
Code for running car	C	A	I	C	R
Code for detecting the obtacles	A	C	I	C	R
Code for detecting the line	C	A	I	C	R
Phase 3: Testing					
Assemble the car's components	A	R	C	C	C
Testing and optimize the algorithm of the car	A	C	C	C	R
R	RESPONSIBLE	designates the task as assigned directly to this person (or group of people).			
A	ACOUNTABLE	make sure the responsible person or team knows the expectations of the project and completes work on time			
C	CONSULTED	provide input and feedback on the work being done in a project			
I	IINFORMED	need to know what's going on because it could affect their work, but they're not decision makers in the process.			

Figure 2.2: Task board

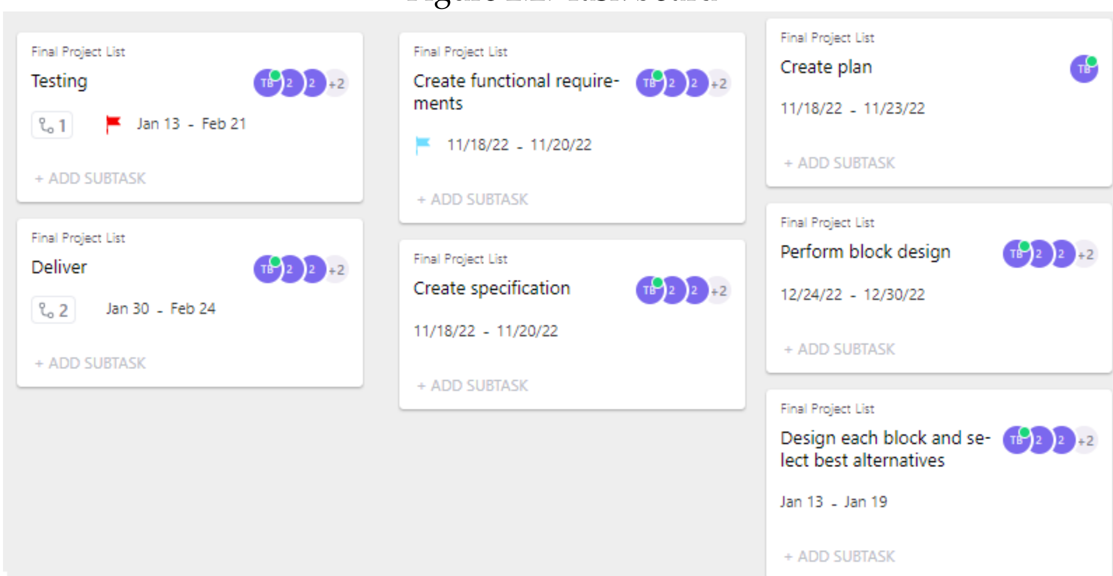
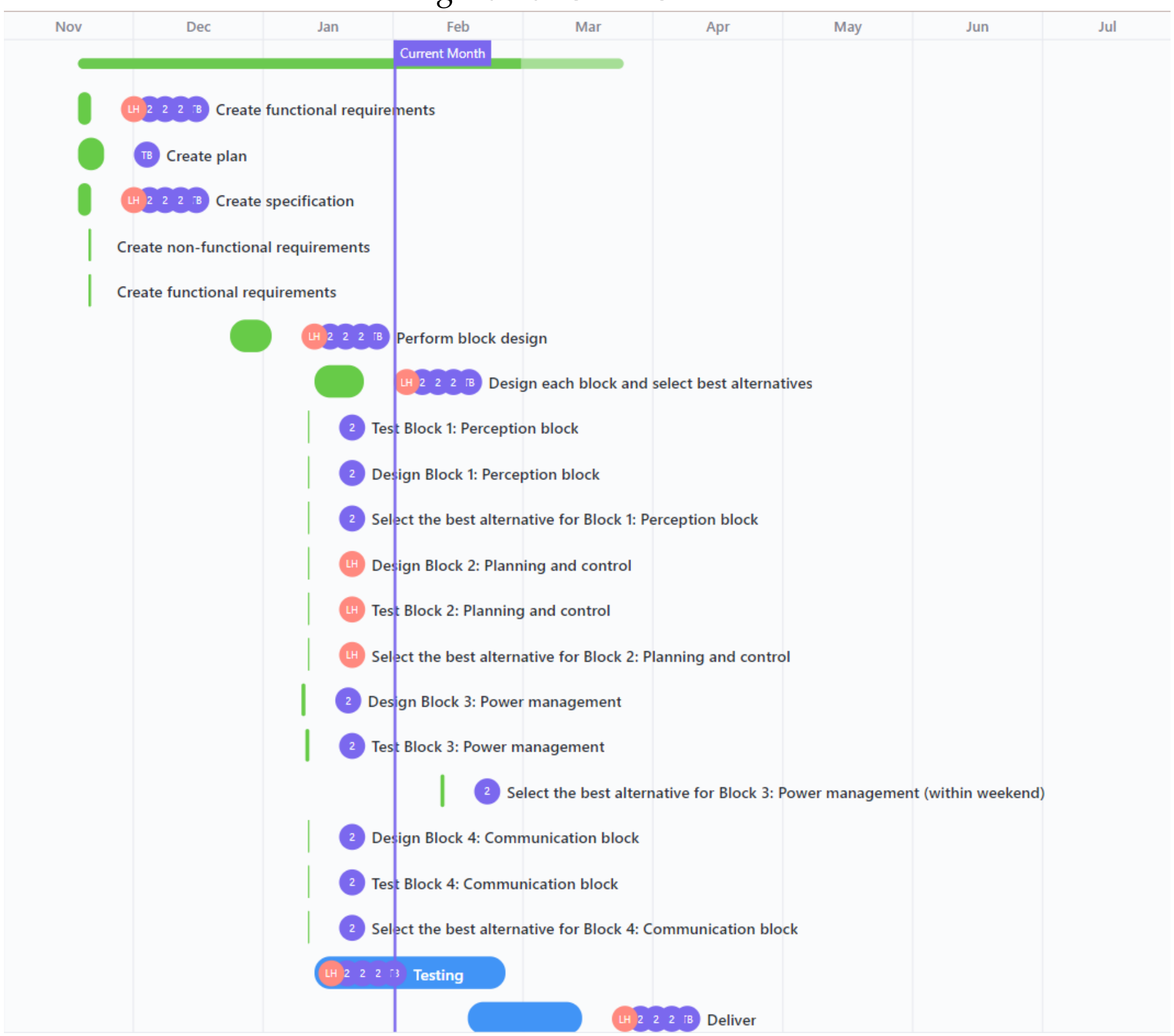


Figure 2.3: Gantt Chart



2.4 Challenges and Decision Making

There are a lot of problems that we encountered in the project. However, after consulting with other teams, Mr. Truong Tuan Vu, the teaching assistant of the course, we have overcome all the challenges and had extremely great experiences while doing the Final project in the course.

- Motor:
    1. The most crucial problem that we faced while doing this project is the unbalanced velocity of two motors. Because of that, our car cannot run straightly; To solve this problem, we carefully read the code and develop the algorithm which helps us balance the motors' velocity of the car.
- The hardware: All the members of my team haven't encountered the hardware of any technical project in high school. Therefore, we faced a lot of problems, including:
    1. Connecting electric wires.
    2. How to use glue to fix 3D components of the car, which is very vulnerable.
    3. Don't know how to use a 3D printer to print the required components for the car.
    4. Assembling all the components of the car.
- The camera: One of the most important aspects of the project is the camera setup since the automobile will use the information from the camera to drive itself. In general, we are facing two main problems:
    1. Choose the right height and location for the camera unless it won't run as we expect.
    2. If the camera runs for a long amount of time, it will be lagging because of the heat.
- The algorithm: The "Brain" of the car, which helps the car automatically detect the line and run autonomously. More information about what we have

done to improve and implement the algorithm can be found in Section 4 4.1.2; However, the problems we faced can be listed as:

1. Read and understand the code, which is quite challenging because we haven't done anything related to computer vision before.
2. Improve and implement the new algorithm for the car. Because the primary algorithm of the car is not optimal and cannot be used to optimally run the car with our components. Therefore, we need to think and try to implement the new algorithm.

- The battery: When we almost finished all of the tasks, the battery doesn't work anymore, and we don't know how to fix it. However, thank Mr. Nguyen Xuan Hieu, a sophomore student who majors in electrical engineering, gave us much useful advice to choose the better substitutive battery for our car.

Chapter 3

System Design

3.1 Block diagram of the system

An autonomous car is a vehicle equipped with various sensors, control units, and actuators that allow it to operate without human input. A block diagram is a graphical representation of the system that provides an overview of its components and how they interact with each other.(figure 3.1)

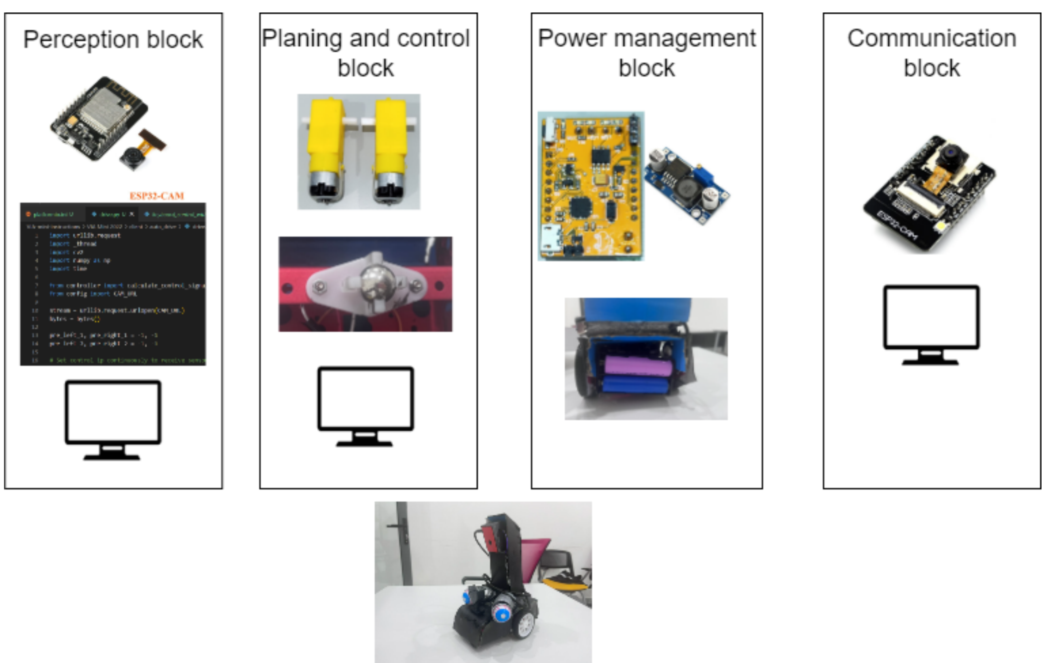


Figure 3.1: Block diagram of the system of an autonomous car

The system consists of several key components, including the perception block, planning and control block, power management block, and communication block. Each of these blocks was designed in detail to meet the project objectives, starting from the images/ information from camera, then the data of environment around the car is processed in processing unit via Wifi and sends commands to the power module to adjust electrical signals to motors of the car.

3.2 Design of each block and select the best alternative

Designing each block of an autonomous car system is an important step that necessitates careful consideration of the available options and selection of the best alternatives. In this step, a detailed design with alternatives for each part of the system is carried out.

3.2.1 Perception Block

The Perception Block of the system includes the ESP32-Cam board (Figure 3.2), a camera module, and the Python programming language for image processing. The camera module captures images of the car's environment, and the ESP32-Cam board sends the images to the Python program running on our computer for processing. The Python program uses image processing algorithms to detect the lane lines and other objects in the car's surroundings.
In detail, the input image is converted to grayscale to simplify the image processing. Then, a Gaussian filter is applied to the image to reduce any noise or unwanted artifacts. Using the Kenny image filter, road markings are detected on the filtered image, allowing for better localization of the vehicle on the road.

Figure 3.2: ESP32-Cam

3.2.2 Planning and Control Block

The Planning and Control Block consists of the ESP32-Cam board, a motor driver, and the car's actuators(Figure 3.4 and Figure 3.5). Based on the images captured by the camera and processed by the ESP32-Cam board, the system calculates the optimal route and sends control signals to the motor driver and actuators to control the car's movements.

In order to calculate the most suitable turning angle for the vehicle, the perspective of the image is transformed from an oblique direction to a direction perpendicular to the road surface using the Birdview transform. This allows us to accurately determine the left and right points on the image, which are then used to calculate the optimal turning angle for the vehicle(Figure 3.3).

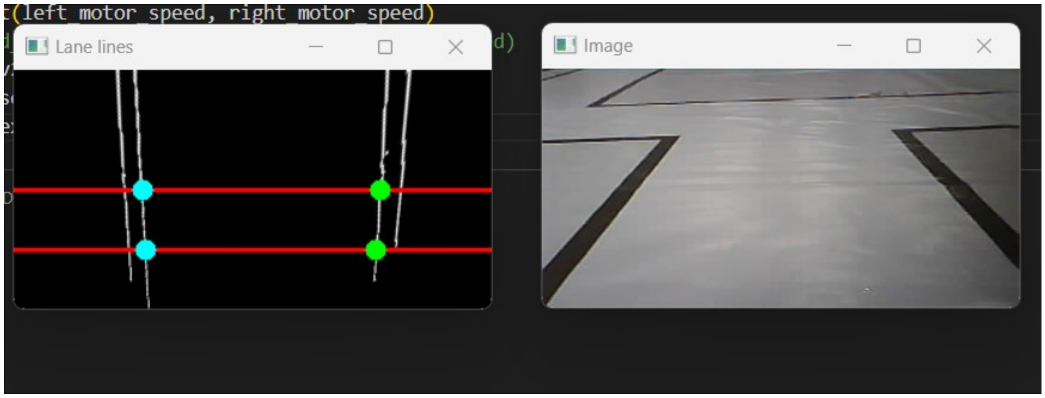


Figure 3.3: Lane_line_detection

Figure 3.4: Actuators_Steering

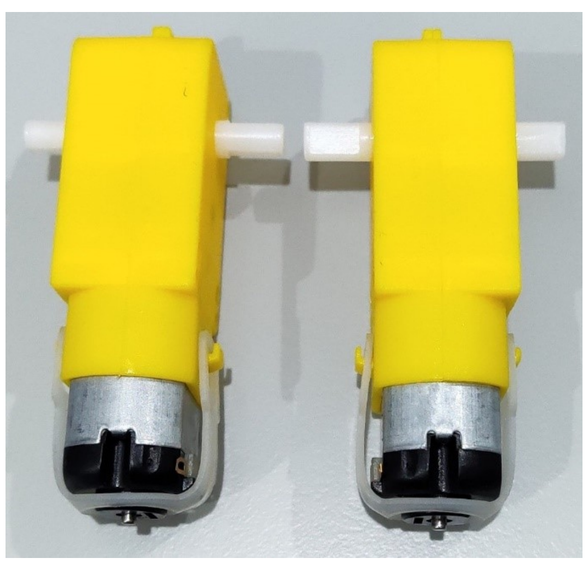


Figure 3.5: Actuators_Motors

3.2.3 Power Management Block

The Power Management Block includes the Power module VIAB (Figure 3.6), Buck converter (Figure 3.7, and the car's battery (Figure 3.8). The Power module VIAB regulates the voltage and current supplied to the ESP32-Cam board and other components, and monitors the battery level to ensure that the car has enough power to operate.

Figure 3.6: Power module VIA B

Figure 3.7: LM2596 DC-DC buck converter

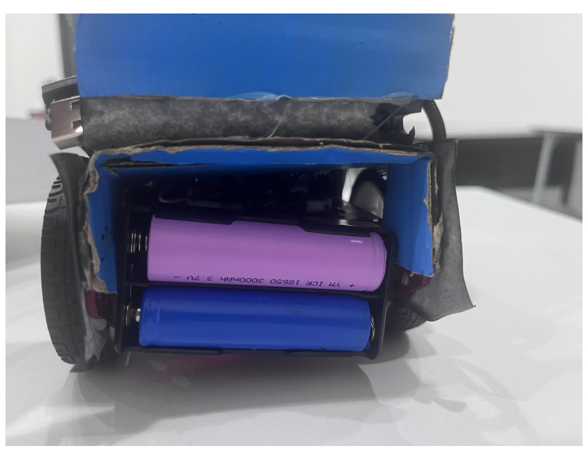


Figure 3.8: Battery with 3,7V

3.2.4 Communication Block

The Communication Block includes the Wi-Fi module on the ESP32-Cam board and a remote control unit (our computer). The Wi-Fi module allows the system to communicate with a remote control unit, which can be used to override the autonomous control if necessary.

3.3 Testing of each block

After designing and implementing each block of the autonomous car system, it was necessary to test each one to ensure they were functioning as expected. This process includes extensive product testing, including testing on the electrical component, the cover, the frame, and the coding under various situations.

This step ensures that the car is working properly: the cover and frame are strong enough to protect the camera and the circuit, the circuit is delivering the correct signal to the correct components, the motors have the same velocity, and the camera works and returns signal to the circuit without delay, and the car can drive autonomously (detect lanes).In this section, we will describe the testing procedures for each block.

3.3.1 Testing of the ESP32-Cam of Perception and Communication blocks

To test the ESP32-Cam block, we first checked if the module was properly connected to the power module VIA B and the remote control unit. Next, we ran a sample code to verify that the camera was working correctly. We also tested the Wi-Fi connection to ensure that the ESP32-Cam module could send the correct data (images) to the remote control unit (our computer).

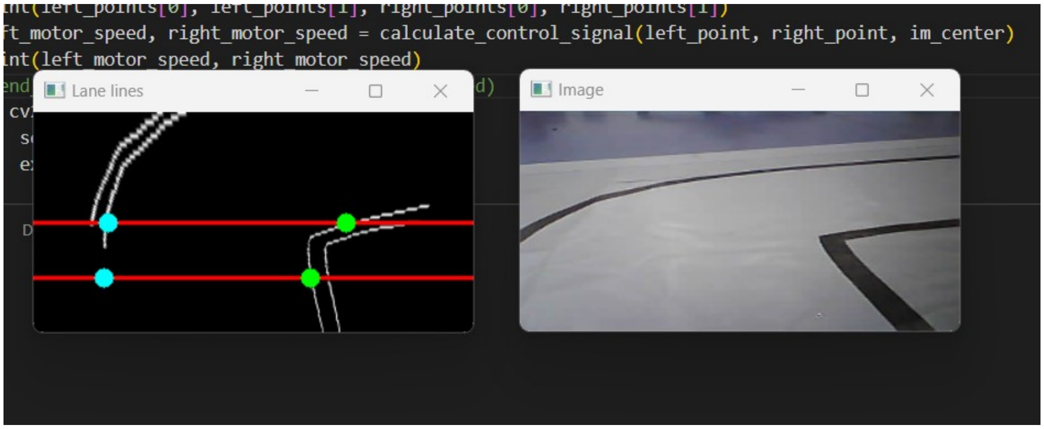


Figure 3.9: Lane_line_detection_2

3.3.2 Testing of the Power Module VIA B of Power management block

To test the Power Module VIA B block, we first checked if the module was properly connected to the ESP32-Cam module and the remote control unit. We used a multimeter to measure the voltage output and made sure it was within the acceptable range (¡5,5 V) (using buck converter to reduce the voltage output to 5,2 V). We also tested the battery charging functionality of the Power Module VIA B by connecting it to a battery and checking if the battery was charging.

3.3.3 Testing of the Line Detection Block

To test the Line Detection block, we used a test image with different road markings to verify the accuracy of the line detection algorithm. We also tested the algorithm's ability to detect the left and right points on the image by using different road widths and angles. We compared the algorithm's results with manual measurements to ensure that it was working correctly.(Figure 3.3 and 3.9)

3.4 Manufacturing and assembly (Implementation)

The manufacturing and assembly (implementation) phase of the autonomous car project involves putting together the hardware and software components that make up the autonomous car system.

3.4.1 Software

- All software was written in Python to facilitate implementation and documentation among team members.
- Making extensive use of Python's Open CV package for processing images from the camera.
- Professors and teaching assistants provided software for the Banhmique board and the ESP32-camera.
- To get a better angle, the camera's coding is changed to the birdview mode.
- The motor's program must be adjusted to match the speed of the two motors.
- Once the software is written, it is uploaded to the ESP32-Cam microcontroller using the PlatformIO. The ESP32-Cam is then connected to the Power module VIA B, which provides power to the motors and other hardware components.

3.4.2 Hardware

- All the hardware is provided by the professor.
- Assemble the various components of the autonomous car such as the ESP32-Cam, Power module VIA B, buck converter, motors, wheels, and other mechanical parts.
- The cover of the car is made by Hardcover and other ornaments, hard enough to protect the mechanical parts of the autonomous car.

3.4.3 Assembling

Following the testing phase, we positioned these parts in the appropriate position, enabling the car to operate at its maximum efficiency:

- The motors need to be securely mounted to the frame of the car using screws and nuts. The wheels are then attached to the motors, ensuring that they are aligned and balanced

- The battery case is connected to a buck converter to step down the voltage to 5,2 V from 7,4 V
- The programming will be loaded onto the Banhmique bard. The new provided firmware is also installed on the ESP-32 camera through PlatformIO.
- Placing the camera at a height of 22cm above the ground and tilting it at an angle of 30° with the vertical axis allows for optimal viewing of the lane and traffic signs, which is ideal for detecting the turn place as it provides a clear view of the road markings and allows the camera to capture a wider field of view, reducing blind spots.
- The Banhmique board and the Camera are linked up, and the computer will connect to the ESP32 Camera through WiFi provided by the Camera. The motors will be wired directly to the Banhmique board and will receive appropriate throttle signals immediately.
- Put the cover and ornaments on the car by glue and tape. Before finalizing the cover, we tested the autonomous car with the cover on to ensure that there are no interferences or obstructions to the cameras or other components.

Chapter 4

System Testing and Analysis

4.1 System testing

4.1.1 Hardware Testing:

1. <u>**Module ESP32 Cam:**</u>
   Specifications:

   - Power Supply: 5VDC
   - SPI Flash: Default 32Mbit
   - RAM: 520KB SRAM +4M PSRAM
   - IO port: 9
   - GPIO: 3.3VDC

   Testing: Upload firmware for ESP32

   - First, we changed the Wifi name and password in config.h file so that the other team could not mistakenly connect to our camera.
   - Then we connected the laptop to ESP32-Cam's Wifi and tested with the manual-driving code. The camera reponsed quickly enough to detect objects.
   - The camera at first attached quite loosely so we had to fastened it with more tapes to have further tests.

2. <u>**Module LM2596:**</u>
   Specifications:

- Input Voltage: 3V-30V
- Ouput Voltage: adjusted 1.5V - 30V
- Wattage: 15W
- Performance: 92
- Size: 45 * 20 * 14 (mm)

Testing:

- We used the electricity meter to measure the Voltage of the module with and without LM2596.
- We found out that the voltage was lowered but still remained effective performance.

3. <u>**DC-DC motors:**</u>
   Specifications:

   - Voltage: 3-12VDC
   - Torque: 800gfcm
   - Weight: 27g
   - Size: 64 * 19 * 22.6 (mm)

   Testing:

   - We also used the manual-driving code to test the two motors.
   - There was no problem with any motor but the speed between them tended to be different a bit. So we had to adjust the throttle in the code so that the car could run properly.

4.1.2 Software Testing

In this step, we tested the code in both simulation and real-world environments by manual and automatic driving.

1. **Manual-driving Testing:**
   The code given in ”keyboard control with cam.py” using the arrow keys to move the car in 4 directions was efficient. However, there was a small difference in the speed of the two motors as the left motor was quite faster than the right one. Therefore, the car tended to head right while moving forward.

2. **Self-driving detection Testing:**
   In the simulation environment, we used the website made by Makerhanoi to create maps, and test the self-driving code. The code given at first moved in the opposite direction at some turns of the maps. Then we recognized a big bug in the code as the variants were at negative values rather than positive. We adjusted some changes and the code worked properly this time in all 3 maps.

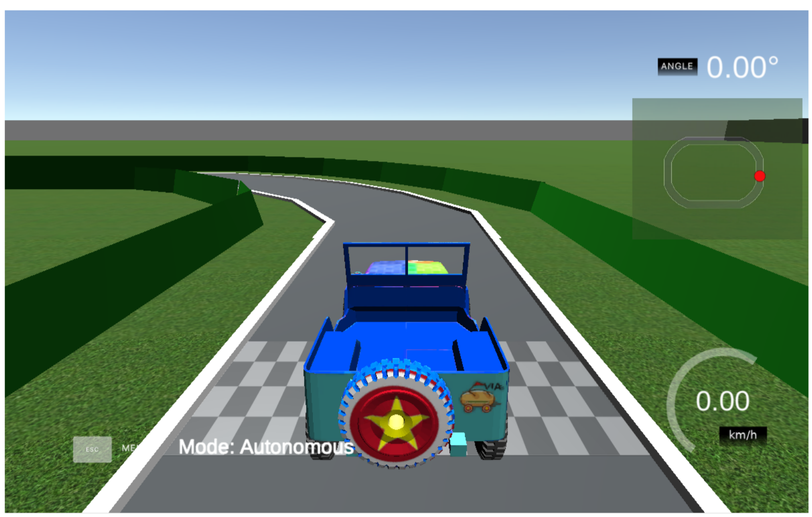


Figure 4.1: Simulation Environment

On the other hand, in the real-world environment, we found some problems:

- Firstly the initial code had only one horizontal line to detect the lane lines; therefore, the car reacted quite slowly at some turns of the real maps. So we drew a second line above the previous one to predict the turning points for the car.

- Secondly the motors only worked when the throttles were bigger than 80 otherwise the speed would remain zero. So we had to calculate the differences between the two motors when turning to maintain the throttles above 80.

After those two big adjustments, the car could finish the rail with limited issues.

4.2 Results, Analysis and Discussion

1. **Results:**

   - Throughout our experiments, we figured out that almost every component in the system was in good condition, except for some noticeable problems when connecting each component together, for instance:
     - The screws were fastened near the wheels so sometimes it caused problems while rolling.
     - We broke the 3D pole one time while testing the car, it took some effort to print a new one and set up the camera in the same position.
   - Manual control of the car works flawlessly in both simulation and real-world environments.
   - In terms of automatic control, while our car followed the line smoothly (both straight and curvy) in the simulation environment, it did not always work properly (e.g., went off the track) in the real world.

2. **Analysis and Discussion:**

   - We noticed the slowness in the camera's response causing many issues in detecting lines. Therefore, we adjusted the code by detecting the current two points using the previous two points.
   - When having a problem while testing, we also printed it on the screen, recorded the bug and discussed the solutions to it. We solved slowly but effectively.

Chapter 5

Conclusion and Recommendation

5.1 Conclusion

The final project of the course CECS1010, Introduction to Engineering and Computer Science, has taught us many useful lessons not only about the autonomous car project but also some of the common project management frameworks, for example, Task table, Gannt chart, Task board, RACI matrix, which will extremely benefit us when we involve ourselves in the real-world project. In the final project, we successfully enhanced and implemented self-driving by using computer vision, a basic lane-detection algorithm. This project has not only improved our technical skills but also provided us with valuable insights into the interaction of software and hardware, and the importance of interdisciplinary. We strongly believe that this project will be the perfect foundation for us to pursue an engineering career.

5.2 Future Recommendation

When it comes to further upgrades for the car of our project, there are many suggestions that can be made to improve our autonomous car.

- We suggest replacing the current plastic camera axis, which was broken at least three times during our final project, with the metal axis (or lightweight material). However, because metal is heavier than plastic, so we need to increase the mass of other components to balance the car.

- The current cover is hard enough to protect the mechanical parts of the car, but it may also add extra weight which can affect the overall performance and fuel efficiency of the car, consider using lightweight materials such as carbon fiber or aluminum in the future.

- The current cover may not be easily removable, which can make maintenance and repairs more difficult, consider designing a modular cover that can be easily removed to facilitate maintenance and repairs
- Improve the accuracy of the line detection algorithm: While the current line detection algorithm may work well in most situations, it may not perform as well in challenging lighting or weather conditions. In our algorithm, based on every camera setup, and the width of the lane, we need to adjust some crucial parameters. Therefore, in the future, it may be beneficial to explore other line detection algorithms or optimize the existing algorithm to improve its accuracy as well as smoothly in every lane without tuning the parameters.
- Implement the car with other algorithms which help the car to detect the traffic signs and obstacles-detection and avoidance. While the current system can detect and follow a lane, it does not have the ability to detect, avoid obstacles and follow traffic signs such as other vehicles or pedestrians. In the future, adding these modules to the system could greatly improve its safety and functionality.
- Improve the system's speed and efficiency: While the current system runs at a respectable pace, it may be feasible to optimize the software and hardware components to improve the system's speed and efficiency.

Bibliography

[1] Https://via.makerviet.org/vi/. https://via.makerviet.org/vi/. (n.d.). Retrieved February 23, 2023, from https://via.makerviet.org/

[2] OpenAI. (2023, February 2). CHATGPT: Optimizing language models for dialogue. OpenAI. Retrieved February 23, 2023, from https://openai.com/blog/chatgpt/

[3] Unity WebGL Player — VIA Simulation - Jeep. (n.d.). https://via-sim.makerviet.org/

[4] Buck converter DC LM2596 3A (GQVT) Shorts. (2019, March 7). Nshopvn.com. https://nshopvn.com/product/mach-giam-ap-dc-lm2596-3a/

[5] Kit RF thu phát Wifi BLE ESP32 Camera ESP32-CAM Ai-Thinker. (n.d.). https://hshop.vn/products/kit-rf-thu-phat-wifi-ble-esp32-cam