Université Cote d'Azur

BIBLIOGRAPHIC REPORT

Years 2024-2025

Scuderia Socca Speed

Students: Di Bona Théophile

Thomas Azzopardi

Supervisor: Pascal Masson

Ecole Polytechnique Universitaire de Nice Sophia-Antipolis, Robotique Systèmes Autonomes
930 route des Colles, Parc de Sophia Antipolis, 06410 BIOT

Table des matières

Introduction

Designing an autonomous vehicle is a significant technical challenge, requiring expertise in robotics, electronics, and computer science. As part of our engineering project, we have chosen to participate in the NXP Cup, a competition where teams must design, program, and optimize a miniature autonomous car capable of following a predefined track.

The NXP Cup is an autonomous vehicle competition designed for students, where teams build, program, and race small-scale autonomous cars on a track. The goal is to complete the track as quickly and accurately as possible, without human intervention. It is organized by NXP Semiconductors and is dedicated to university and engineering students interested in robotics, embedded systems, and autonomous driving technologies.

Key Aspects of the NXP Cup:

1. **Objective**: To design, build, and program a small autonomous car that can navigate in a predefined track autonomously.
2. **Track Design**: The track usually consists of various challenges, including straight paths, curves, and intersections. The car must be able to detect these features and respond accordingly.
3. **Technical Challenges**: Participants face several challenges such as sensor integration, path following, speed control, and obstacle detection.
4. **Evaluation Criteria**: Teams are judged based on their car's ability to complete the track quickly and accurately. Other factors, such as innovation in design and software, may also be considered.

This project presents several issues. First, sensor management is crucial to enable the vehicle to perceive its environment and make real-time decisions. The sensors must provide accurate and reliable data despite variable environmental conditions (lighting, obstacles, sharp turns). Additionally, vehicle control must be robust and responsive, allowing for speed and direction adjustments without compromising system stability. Finally, optimizing image processing algorithms and decision-making systems is essential to ensure smooth and efficient navigation around the track.

Through this bibliographic review, we aim to explore the research and technical solutions that have been developed to address these challenges in the field of autonomous vehicles. This work will help in identifying the best practices to apply in order to maximize the performance of our prototype in the NXP Cup.

1) Specifications

I. NXP Cup Regulations:

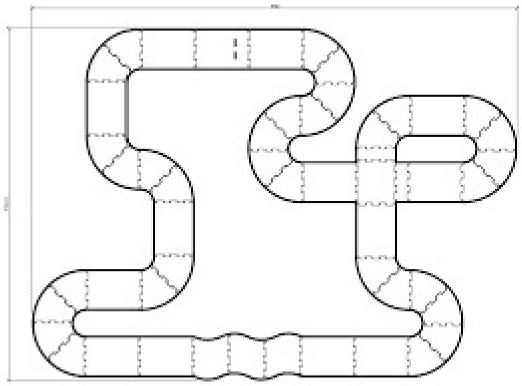


Figure 1 : NXP Cup Track

The NXP Cup consists of two different parts:
The first part is the speed race: Our car must complete an unknown course as quickly as possible. This track includes turns, straight lines, chicanes, and intersections. The track is defined by two black lines on each side. The course varies during different competitions and qualifiers but always maintains a similar length and level of difficulty.
The second part is the lap of honor: After completing the high-speed lap, our car slows down and must stop within 10 cm of an obstacle that will be placed on the track.

II. Project Objectives and Scope:

To participate in the NXP Cup, our car must accomplish the following tasks:

- Move at a speed appropriate to the trajectories taken by the car
- Turn with the highest possible precision
- Navigate the course autonomously
- Stay within the track boundaries
- Detect an obstacle in front of it
- Stop at a specific location

2) Choices and Potential Issues

I. Choices for an optimal car

We will opt for front-wheel drive with only two motors at the front. We'll use a light to illuminate the track if necessary. The battery will be placed at the rear of our vehicle to ensure better stability.

We'll also check the weight of the materials used and choose the lightest one, which appears to be PETG compared to wood (so we'll use 3D printing and design the models ourselves).

This choice is explained by the fact that PETG has a significantly lower density than wood. PETG will also be strong enough to hold the components securely on our car. If we can improve this model, we will add elements to enhance the aerodynamics of our car.

As for the organization, we will adopt a two-layer structure: the chassis provided by NXP, with a 3D-printed layer added on top. The battery will be placed on the chassis, while the microcontroller and the obstacle detector will be located on the added layer of our car.

We are also paying close attention to the dimensions and ensuring compliance with the imposed regulations. As required, we plan to add a button to start the car. Additionally, when creating any added parts, we aim to minimize the extra space they occupy.

After hearing feedback from our colleagues who previously worked on an NXP car project, one significant improvement would be adding suspensions. We hope to incorporate suspensions to enhance the vehicle's stability and potentially improve the image quality. The suspensions would be placed on all four wheels of the vehicle.

Figure 2 : NXP Chassis

II. Potential issue

A potential issue that immediately arises is the stability of our vehicle, along with cable management. Mounting a camera at a certain height requires good stability for better image processing and smoother trajectory. We also need to secure the wheels to ensure that the entire chassis remains stable.

We will need to check the total weight and motor power to avoid either excessive acceleration or, conversely, being too slow (though the latter is unlikely). Additionally, the organization of our car and the distribution of weight during sharp turns could become a problem that we'll need to address.

3) Motors

I. The wheel motors

As for the motors, we don't need very powerful ones. Since the race will consist mainly of turns, we should prioritize motors that respond quickly. For steering the wheels, we have also chosen basic motors with a turning angle that doesn't need to be very high for the curves.

Therefore, we will use XXD A2212, 930KV motors to move our car. We will use two, one for each front wheel of our car.

However, this motor is quite powerful for our project, so we will need a large gearbox and precise control over the resulting acceleration.

We also opted for a brushless motor to further improve reactivity, reduce potential friction compared to a brushed motor, and achieve higher speeds. This motor is powerful enough to move our car and therefore has a high torque.

Finally, to stop our car, we simply plan to cut off the power to the motors. Therefore, we won't need to use H-bridges to reverse direction or to force a sudden stop.

Figure 3 : XXD A2212 motor

Technical Specifications:

- **Speed:** RMP / Volts
- **Power Consumption**: 6-10A
- **Power Input**: 10V
- **Efficiency:** 80%
- **Weight**: 47g

II. The steering motor

We also realized that a good amount of torque would be necessary to turn our car effectively. Therefore, traditional servo motors SG90 with a standard torque might not be sufficient. We decided to opt for a more powerful servo motor, choosing one with 10 times the torque: the Mitoot MG996R.

It also offers high responsiveness, as it can perform a 60-degree rotation in just 0.15 seconds.

Regarding the servo motors, we observed that some require varying levels of current. Therefore, we may need to add a current regulator with this servo motor.

(No information about the current is available online.)

Here are photos of the components mentioned.

Figure 4 : Mitoot MG996R

Technical Specifications:

- **Speed:** 60°/0.17 sec
- **Power Consumption**: 6-10A
- **Power Input**: 6V
- **Couple:** 15 kg/cm
- **Weight**: 55g

4) Battery and design

I. The battery

The battery is also a component that must be carefully chosen, but there are rules. According to the regulations shared by NXP, we are allowed a battery with the following specifications:

- A maximum charge of 6000mAh
- Only one battery allowed
- LiPo battery limited to 3 cells (3S) and 11.1V"

We have chosen a more efficient battery; in this regard, we can regulate it fairly easily, so we opted for the maximum allowable. We decided on the Zeee LiPo battery.

Figure 5 : Zeee Lipo Battery

Technical Specifications:

- **Capacity:** 6000mAh
- **Cells**: 3S
- **Power Output**: 11.1V
- **Dimension :** 138*47*37mm (L x L x H)
- **Weight**: 440g

II. Design

Regarding the design of our car, we will be using a two-tier chassis provided by NXP.

The first level will consist of the motors connected to the battery. The servo motor will be placed between the two levels to enable steering of the wheels.

As for the NXP microcontroller, it will be positioned at the very top. Since we plan to use the NXP chassis, which already has several holes to allow cable routing, we will still need to model 3D parts to ensure everything is stabilized while maintaining optimal cable organization.

Attached is a 3D model illustrating our car design.

5) Wheels

For the wheels, we are not entirely sure what to choose, as their size and other factors greatly influence the performance of the car. Therefore, we will first test the wheels provided by NXP with the chassis, and if they do not meet our needs, we will try other available options.

Our selection criteria for the wheels are as follows:

- **Grip**: The wheels need to adhere well to the ground to ensure stability for the entire structure.
- **Durability**: They must withstand sharp turns (without deflating if there's air).
- **Size**: They should be relatively small; we estimate a diameter of around 5 cm would be suitable.
- **Tread Pattern**: The pattern on the wheels should enhance energy transfer.

Figure 6 : Haptufer 4 Pieces

Based on these criteria, we plan to use rubber wheels with solid plastic hubs. We will conduct several tests to determine the best possible wheels. Through some research, we have found a few options to test.

6) Microcontroller

I. Requirements

Choosing a good microcontroller for the NXP Cup is crucial because it directly impacts the vehicle's performance and responsiveness in a competitive racing environment. A capable microcontroller ensures efficient processing of sensor data and quick execution of control algorithms, which are essential for tasks like navigation, obstacle detection, and speed regulation. Additionally, it facilitates easy integration with various peripherals, allowing for a more versatile and effective autonomous racing system.

II. Selected Solution

For the choice of the microcontroller, we are quite limited. Indeed, we are restricted to NXP microcontrollers. The recommended microcontroller by the brand for the cup is the **RDDRONE-FMUK66.**

Figure 7: Microcontroller RDDRONE-FMUK66 PX4

Technical Specifications :

Processors and Microcontrollers

- **K66_180**:
  - Kinetis® K66-180 MHz
  - Dual High-Speed and Full-speed USBs
  - 2MB Flash Microcontrollers (MCUs) based on Arm® Cortex®-M4 Core

Analog and Mixed Signal

- **Voltage Level Translators**:
  - **NTB0104**: Dual-Supply Translating Transceiver (Auto-Direction Sensing, Three-State)

Interfaces

- **Automotive Ethernet PHYs**:
  - **TJA1100**: IEEE 100BASE-T1 Compliant Automotive Ethernet PHY Transceiver
- **3.3 V / 5 V IO CAN Transceivers**:
  - **TJA1042**: High-Speed CAN Transceiver with Standby Mode

Here are the various advantages offered by this microcontroller:

1. **Sensor Integration**: Integrates navigation sensors (accelerometer, gyroscope), enabling real-time data collection for localization and orientation.
2. **Connection Flexibility**: Multiple interfaces (SPI, I2C, UART) for easily adding external sensors (LIDAR, cameras) for obstacle detection.
3. **Open source**: Based on PX4, providing access to a vast community and resources for developing advanced features.
4. **Processing Capability**: Powerful processors to execute complex algorithms, enhancing the vehicle's responsiveness.
5. **Modularity**: Modular design allows for easy upgrades based on race requirements.

These features make the RDDRONE-FMUK66 a solid choice for developing an autonomous racing car.

However, programming on this microcontroller can be quite complex for a beginner. To avoid losing too much time at the beginning of our project, we will start programming the car using an Arduino board. For this, we have chosen the Arduino Due, as it is the closest in terms of processing speed to the RDDRONE-FMUK66 PX4.

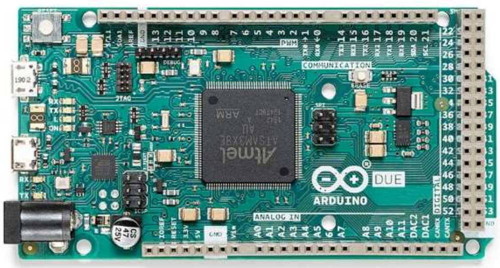


Figure 8: Arduino Due

Technical Specification:

Operating Voltage:

- 3.3V for stable operation.

Input Voltage:

- Recommended input range is 7-12V, with limits of 6-16V for flexibility in power supply.

Digital I/O Pins:

- 54 pins available, with 12 capable of PWM output for motor control and other applications.

Flash Memory:

- 512 KB of flash memory available for user applications, allowing for substantial program storage.

Clock Speed:

- Operates at 84 MHz, providing efficient processing power for real-time applications.

Once the project is functional with the Arduino board, we will switch to the NXP microcontroller.

7) Camera

I. Requirements

To recognize the path as accurately as possible, the choice of camera is crucial.

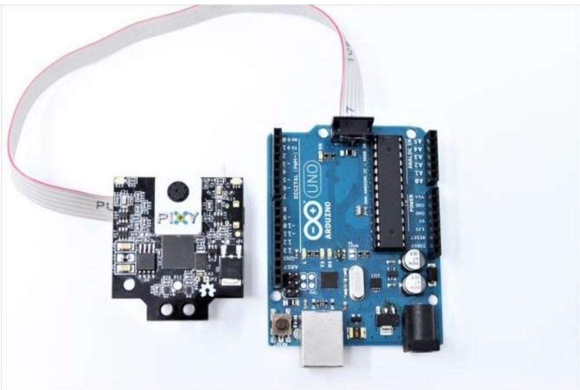
Our goal is to have the fastest car possible. To achieve this, our camera needs to capture as many images per second as possible in order to accurately identify the racetrack and process the images in the shortest time possible.

To find the most suitable one, we studied different types of autonomous cars, particularly those used by participants in previous editions.

As with the choice of the microcontroller, our options for a camera with an internal microprocessor are limited. The only "intelligent" camera allowed is the Pixy Camera because its microprocessor is an NXP.

II. State of the Art

Pixy Camera



Technical Specifications

- **Field of View**: 80 degrees horizontal, 40 degrees vertical
- **Power Consumption**: 140 mA typical
- **Power Input**: USB input (5 V) or unregulated input (6 V to 10 V)
- **RAM**: 264 KB
- **Integrated Light Source**: Approximately 20 lumens
- **Weight**: 10 grams

Figure 9: Pixy Camera connected to an Arduino micro-controller

The Pixy camera is a vision sensor designed for robotics and autonomous vehicles. It uses a powerful image processing algorithm to detect and track objects in real-time. The camera can recognize colors and shapes, allowing it to identify specific targets based on their color signature.

1. Image Capture: The Pixy camera captures images using its built-in lens and sensor.
2. Object Detection: It processes the captured images to detect objects based on predefined color parameters. Users can teach the camera to recognize different colors by placing them in front of the lens.
3. Data Output: Once an object is detected, Pixy outputs the object's position and dimensions to a connected microcontroller via interfaces like SPI, I2C, or UART.

However, after discussing with upper-year students, it seems that the Pixy camera is very sensitive to reflections, which could hinder us depending on the ambient light conditions.

ArduCam MEGA 5MP

Our second option is the ArduCam MEGA camera. This is the camera that was chosen by our classmates in the 5th year.

The Arducam Mega is a versatile camera module designed for embedded vision applications with microcontrollers and single-board computers. It is optimized for compatibility and ease of use, supporting various communication protocols such as I2C, SPI, and UART. This camera is suitable for projects where efficient image processing and analysis are essential, like line following robots, object detection, or IoT systems.

Figure 10: ArduCam Mega

Technical Specifications :

- **Sensor**: Mega 1/4" CMOS color (5 Megapixels)
- **Max Resolution**: 2592 x 1944 pixels
- **Angle**: 75°
- **Focal Length**: 2.8 mm
- **Lens**: M12 screw mount (interchangeable)
- **Power Supply**: 3.3V to 5V DC
- **Consumption**: 55mA (in standby) to 154mA max
- **Supported Formats**: JPEG / RGB / YUV
- **Interface**: SPI

Key Features:

- **Interfaces**: Supports I2C, SPI, and UART, making it compatible with a wide range of development boards, such as Arduino, ESP32, NXP….
- **Frame Rate**: Capable of achieving up to 60 FPS at lower resolutions, making it suitable for high-speed applications where capturing rapid changes is crucial.
- **Compatibility**: Provides SDKs and libraries in C/C++ and Python, ensuring seamless integration with platforms like Arduino and NXP microcontrollers.
- **Global and Rolling Shutter Options**: Available with different sensor types for varied applications, ensuring better performance for motion-heavy scenarios.

III. Selected Solution

The Arducam Mega is ideal for Autonomous Vehicles. Indeed, with its high frame rate and versatile interfaces, it can be used to detect lines and follow paths in autonomous vehicle projects like the NXP Cup.

However, the camera does not have an internal microprocessor. But this is not a major issue since the image processing will be quite basic. Indeed, our only goal is to detect two black lines on a white track. As the camera directly provides us with light intensities, it will be quite easy for us to distinguish between white and black.

Therefore, we will choose the ArduCam. Moreover, if we have any questions or problems, we can reach out to the 5th-year robotics students.

8) Ultrasound Captor

I. Requirements

During the final lap, we need to stop upon detecting a white cube placed on the track. Several types of sensors are available to us for obstacle detection. Since the final lap will be performed at low speed, the sensor's performance is not critically important.

II. State of the art

Captor	HC-SR04	Sharp GP2Y0A41SK0F
Waves type	Ultra-sound	Electromagnetic
Range	2cm-2m	4cm-30cm

III. Selected Solution

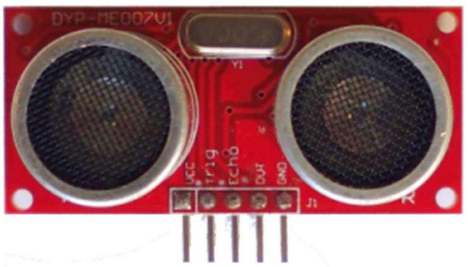


Figure 11: HC-SR04

Technical Specifications

- **Power Supply**: 3,3 or 5 Vcc
- **Consumption**: 15 mA
- **Frequency**: 40 kHz
- **Range** : 2cm – 2m

We will use the HC-SR04 ultrasonic sensor because we want to take measurements at medium distances. Additionally, during previous competitions, the room where the race took place seemed quite bright, which could affect an electromagnetic wave sensor. Furthermore, we do not know the material of the cube or its ability to reflect light. Therefore, opting for an ultrasonic sensor is preferable in our case. Finally, we have already used this sensor in previous classes, so its implementation within our project will be relatively straightforward.

9) Design and Solutions

I. Design

Regarding the design of our car, we will be using a two-tier chassis provided by NXP.

The first level will consist of the motors connected to the battery. The servo motor will be placed between the two levels to enable steering of the wheels.

The microcontroller will be placed on top of our car and will be directly connected to the camera, which is positioned about 15 cm high on the rod located at the front of the vehicle.

As for the NXP microcontroller, it will be positioned at the very top. Since we plan to use the NXP chassis, which already has several holes to allow cable routing, we will still need to model 3D parts to ensure everything is stabilized while maintaining optimal cable organization.

We are drawing inspiration from the designs of previous participants and finalists of the NXP Cup.

We also aim to limit the overall weight of our project, and therefore we have made the following estimates: 229 (chassis) + 410 (battery) + 55 (servo motor) + 47*2 (wheels motors) + ~200 (4 wheels) + 100(plastic pieces) + 80 (microcontroller) + 20 (cables) + 20 (screw) + 10 (camera)

Total ~ 1,218 KG

And for the dimension, we measure: 27cm of length, 13cm wide (+ 3 for the wheels) and we estimated 25cm high. (27 * 16 * 25 L x l x h)

Attached is a 3D model illustrating our car design.

II. Simplified overview

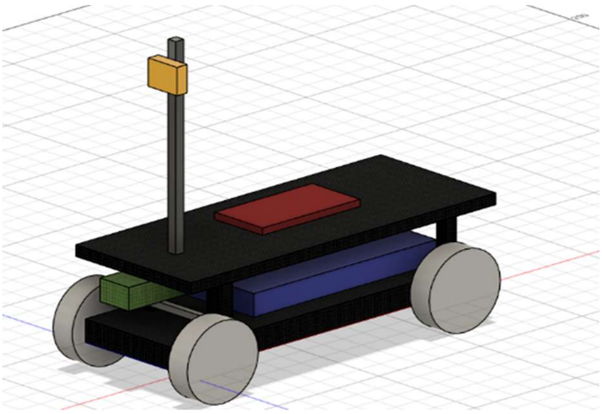


Figure 12: Simplified overview

Color	Piece/Component
Black	Chassis
Grey	Wheels
Blue	Battery
Green	Servomotor
Red	Micro-controller
Yellow	Camera

III. Solutions

We have encountered several issues, and here are the solutions we propose:

-We will ensure greater stability by using custom-designed parts to secure each component and prevent them from falling off during high-speed turns.

-We also plan to add shock absorbers on the sides of our car. Careful control of acceleration and speed will be implemented for the vehicle. Finally, we will optimize the cable management for the connections between each component (for a clearer visual).

-The high RPM of our motor will be reduced by a significant gear system (which we will likely model). Below is a diagram of the system used.

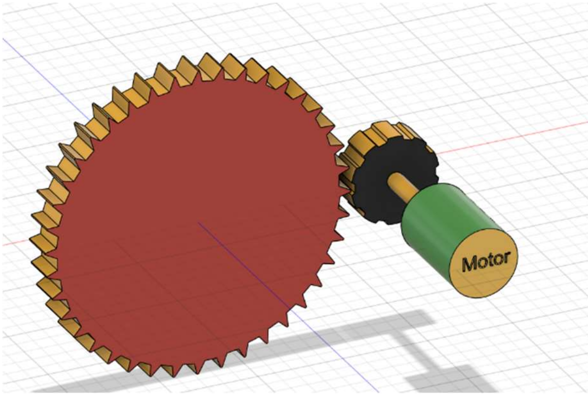


Figure 13: Simplified Reductor

Color	Piece/Component
Red	Big gear connected to the wheels
Green	Motor
Black	Small gear connected to the motor

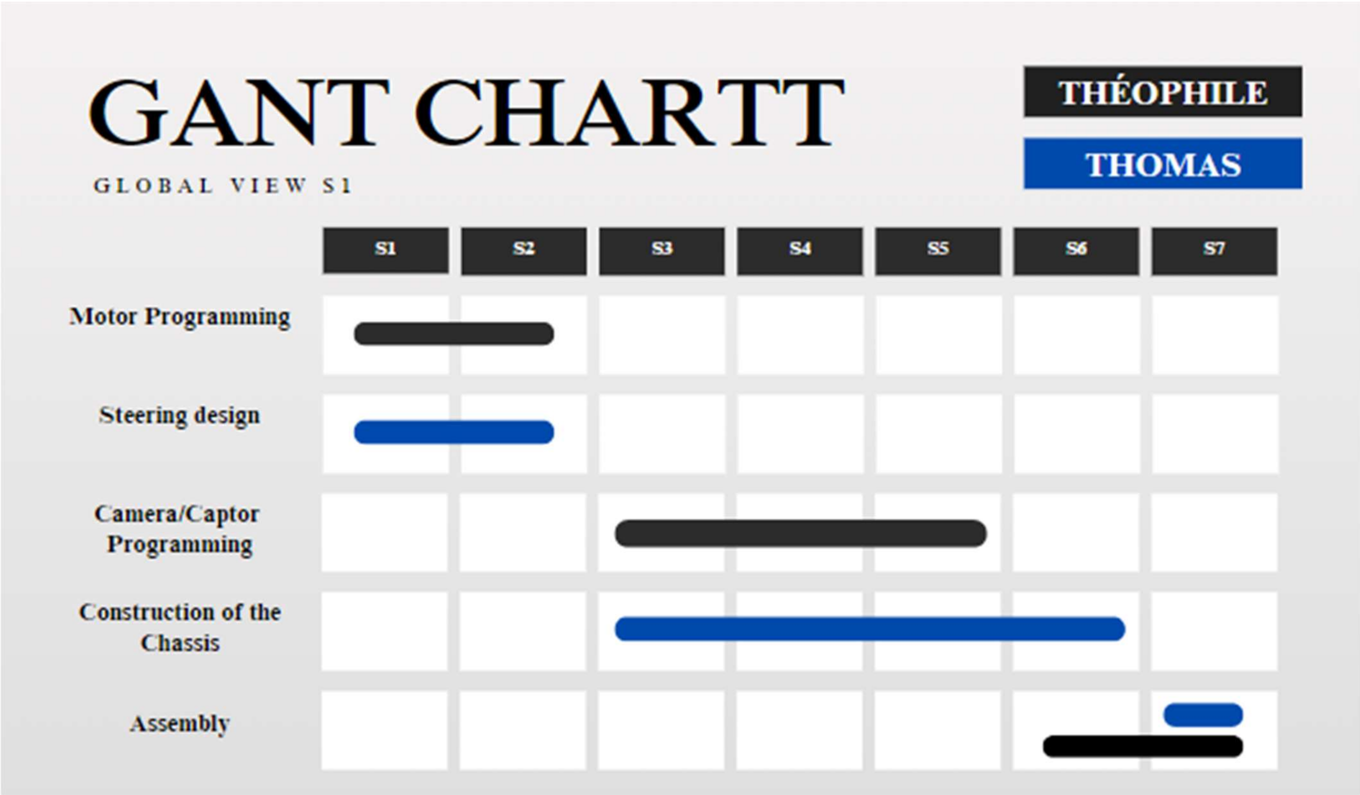
Conclusion

This report outlines our approach and choices for building an efficient autonomous car for the NXP Cup. We selected a front-wheel drive design using powerful brushless motors and a high-torque servo for steering. The RDDRONE-FMUK66 microcontroller will provide advanced processing capabilities, while the ArduCam Mega camera ensures accurate path detection.

With a focus on stability and performance, we have chosen reliable sensors, a lightweight chassis, and a robust gear system. Despite potential challenges, we are confident that our design and iterative development will result in a competitive vehicle that meets all requirements for the NXP Cup.

Organizational and Cost

I. Organizational



II. Cost

PROJECT COST

NXP CUP

ITEM	QUANTITY/HOURS	UNIT PRICE	TOTAL
Engineering Cost Di Bona Théophile	28	23,75€	**665€**
Engineering Cost Azzopardi Thomas	28	23,75€	**665€**
Microcontroller RDDRONE-FMUK66	1	139€	**139€**
Servomotor MITOOT MG996R	1	4.2€	**4,2€**
Chassis* NXP	1	22,40€	**22,40€**
Camera ArduCam MEGA 5MP	1	40€	**40€**
Battery ZEEE	1	37,80€	**37,80€**
Motors XXD A2212	2	5,60€	**11,20€**
Wheels* NXP	4	2,50	**10€**
		TOTAL	**1595,00 €**

References

Sources:

Rules:

https://www.nxp.com/docs/en/supporting-information/NXP-Cup-2024-rules.pdf

NXP CUP 2018 -Day two: Race and awards

https://youtu.be/x5r4GDVfLoM?si=kGlx6Ejort5tdBEr

NXP Cup 2022

https://youtu.be/pwsoolTXwt0?si=1qLYJNi0eGFFyWhD

NXP CUP EMEA Finals 2021

https://youtu.be/Bp2K9iJiKZI?si=Sgdxw1uI5HJ90-U_

The 3D models were created by us

Components:

Motor brushless :
https://fr.aliexpress.com/item/1005006923891145.html?spm=a2g0o.productlist.main.23.319261acKncR5Z&algo_pvid=e8952388-b6bb-493f-8ed9-44685e9de08f&algo_exp_id=e8952388-b6bb-493f-8ed9-44685e9de08f-11&pdp_npi=4%40dis%21EUR%215.76%215.59%21%21%2143.88%2142.62%21%40211b653717277931657631264e3847%2112000038738328729%21sea%21FR%214705888167%21X&curPageLogUid=NqKMQtCFWcQ2&utparam-url=scene%3Asearch%7Cquery_from%3A

Zeee Battery:

https://fr.aliexpress.com/item/1005007464141068.html?src=google

Servo-motor :

https://fr.aliexpress.com/item/4000075435453.html?spm=a2g0o.productlist.main.91.4c473b89GApbNF&algo_pvid=81c0e8cc-dd34-4cca-aa2f-dab9d4ecdafc&algo_exp_id=81c0e8cc-dd34-4cca-aa2f-dab9d4ecdafc-45&pdp_npi=4%40dis%21EUR%214.29%214.29%21%21%214.68%214.68%21%4021039e0c17277135221038833e54cb%2112000036755492895%21sea%21FR%214705888167%21X&curPageLogUid=5u2znqRSya39&utparam-url=scene%3Asearch%7Cquery_from%3A

Microcontroller:

https://www.nxp.com/design/design-center/development-boards-and-designs/px4-robotic-drone-vehicle-flight-management-unit-vmu-fmu-rddrone-fmuk66:RDDRONE-FMUK66

Camera :

https://www.arducam.com/product/mega-5mp-color-rolling-shutter-camera-module-with-m12-lens-noir-for-any-microcontroller/

Ultrasound Captor:

https://www.gotronic.fr/art-module-de-detection-us-hc-sr04-20912.htm