Functional Design (FD)

Project: Autonomous Electric Car

Company: BlitzDrive

Place, date: Enschede, 11st November 2020

Drawn up by Aritra Biswas

Dung Phan

Mathieu Gayler

Jimmy Tithphit

Nghia Hoang

Version: 0.1 (final version will have the number 1.0 and will be only on the title page)

Place, date:	Saxion Hogeschool, Enschede 15-03-2024		
Drawn by:	BlitzDrive		
Chapters 3, 4	Aritra Biswas	545606	545606@student.saxion.nl
Chapters 1, 4	Mathieu Geiller	553016	553016@student.saxion.nl
Chapters 2, 3	Nghia Hoang	553429	553429@student.saxion.nl
Chapters 3, 4	Jimmy Tithphit	518275	518275@student.saxion.nl
Chapters 2, 3	Dung Phan	546821	546821@student.saxion.nl
Version Number:	[0.1]: The first draft for functional design of our project.		
	[0.2]:		
Clients:	Saxion Hogeschool		

Table of Contents

I. List of figures

II. Abbreviations

DSP	Digital Signal Processing
FFT	Fast Fourier Transform
TL;DR	Too long; didn't read

1. Introduction

In this project, we need to make a car that can follow the main requirements that were provided by our clients. The desired result is to explain and show our functional design and how everything works on a functional level, by explaining our choices for components and high-level flowcharts to show what we are going to be doing for the main requirements.

The goal and objective of this document are to provide an overview of the technical aspects in the form of a functional design. In this document, we aim to provide our reasons for choosing certain concepts and ways of proceeding with our project compared to others. We explain our chosen concept and aim to give a small insight as to why it is better compared to other concepts.

2. Analysis of requirements

To design and realisation an Autonomous Electric Racing Car, there are sub-objectives that BlitzDrive has to achieve, our team has grouped the sub-objectives into 4 main requirements to clarify and track the team's process.

MOTION:

Ride autonomously

Our car has to be able to drive around without any human assistance, such as using a controller to control the car, using our hands to push it or control it. To ride autonomously means it has to meet all the requirements underneath to achieve that.

Avoid obstacles and should have to go up a ramp

To avoid obstacles, the car needs to dodge the object in front which is a block by turning and turning back to try the detect line again.

When we detect an inclination then add more power to the motors to get over the ramp and maybe brake a bit as we do not want it to go too fast as it might go off track or if the end is close then go over the end.

DETECTION:

Must follow the white line (20mm)

It has to follow the white line for the track (Figure 1.1). To do this, we are going to use IR sensors that will be placed underneath the car to help detect the white line.

Figure 1.1 The Track

Finding the white line

When dodging the obstacle, our car first turns right and then turns back left, while turning left, it has to catch the white line again and stop turning left. The same thing applies to when we have to turn the car when the white line curves, it has to keep on turning right or left until the line is detected again and stop turning to be able to follow it.

Stop at the end of the track

When arriving at the end of the track, the car has to stop by itself without any assistance, to do this, we can reprogram the wheels to brake when the line is not detected, so this correlates back to the detect line requirement

BUILD:

Fit in a box of 400mm x 250mm x 200mm

While building our car, we must respect the dimensions of the box that we are using to store our car.

Have a body (3D printed separate parts)

We must design a body and how we will place the components and our PCB and breadboard on it.

Maximum weight is 3 kg

We must research the components that are suitable for this requirement.

POWER CONSUMPTION

Has to work for 30 minutes

Our car has to run nonstop for 30 minutes, depending on what batteries we have, the amount of batteries will depend. Besides, some perspectives could also affect the battery such as the velocity, temperature, and the quality of itself.

3. Concept Principles

After the discussion, we vote on which concepts/components are suited to the scope of the project requirement. We ended up using “Infrared Sensors “to detect the line. “Ultrasonic Sensor” for obstacle detection, “Arduino Uno” for the microcontroller, “lithium-ion Battery” for our power source, “Wheels” for manoeuvring on the track, and “C++” as our coding language.

[+] = Compatible

[++] = Very Compatible

[-] = Bad

[- -] = Very Bad

Microcontroller	Processing Power	Memory	Price	Power Consumption	Compatibility
Arduino UNO	+	+	++	+	+
Raspberry Pi	++	+	-	- -	+
Arduino Nano	+	+	++	+	-

Inclination Detection	Accuracy	Sensitivity	Range	Price	Response time
Tilt Sensor	-	+	+	++	++
Gyroscope	++	- -	++	- -	+
Sensitive Tilt Sensor	+	-	+	+	+

Obstacle Detection Sensor	Detection Range	Accuracy	Response Time	Field of View	Price
Ultrasonic Sensor	+	-	+	+	++
Camera Sensor	+	++	-	+	- -
LiDar Sensor	+	+	-	+	-

Line Detection Sensor	Accuracy	Detection Range	Response Time	Resolution	Price
IR Sensor	+	-	+	-	++
LiDar Sensor	++	+	+	+	- -
Camera Sensor	++	+	-	+	- -

Movement and Propulsion	Power Rating	Torque	Compatibility	Efficiency	Price
Brushed Motor	++	+	+	++	+
Brushless Motor	+	+	+	+	+

Battery or Power Source	Power Rating	Capacity	Size and Weight	Dimensions	Efficiency	Price
Nickel Battery	-	-	+	+	+	++
Lithium-ion Battery	++	++	++	++	++	- -
Alkaline Battery	+	+	- -	- -	-	-

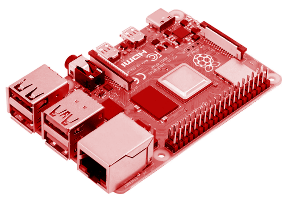
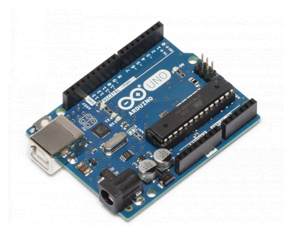
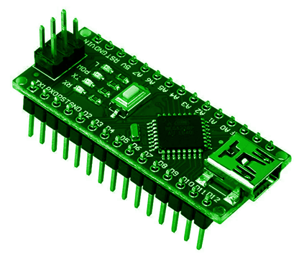
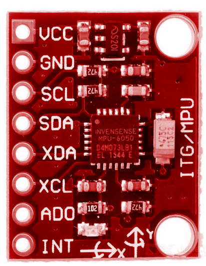
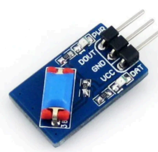
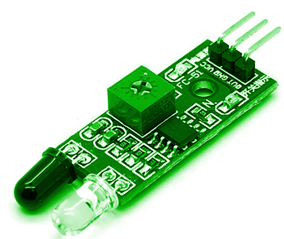
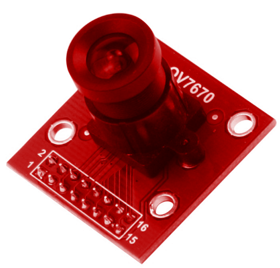
3.1 Description of alternative concepts

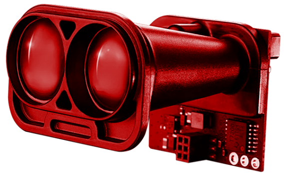
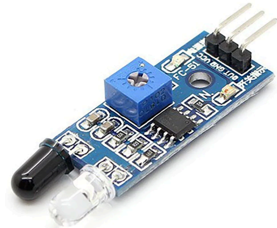
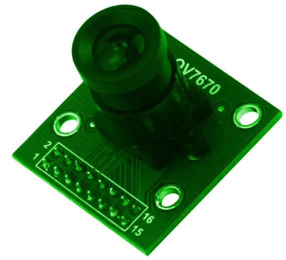
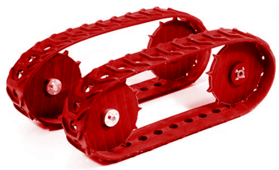
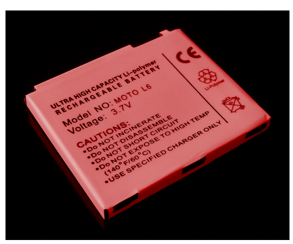
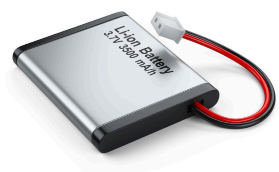
We also have alternative concepts by using camera sensors combined with a few ultrasonic sensors. Within this concept, we'll be using Raspberry pie for our microcontroller. 24V battery for our power source, a Specialised wheels set that can move sideways to manoeuvre on the track and C++ as our coding language. There are certain reasons why we didn't choose the following components in the concepts

- Microcontroller: The Raspberry Pi is clearly the strongest out of the 3 with superior power processing and memory, but with our limited budget, we couldn't afford it. While for the Arduino NANO, its processor is simply inefficient enough for our project.

- Ramp Detection: The Gyroscope has great accuracy measurement and range, but it is also expensive and insensitive, while the sensitive tilt sensor is worse in every aspect compared to our choice.
- Obstacle Detection: For obstacle detection, the camera sensor has good stats, but not affordability, while the LiDar sensor also poses little to none outstanding stats, but also costs more than the Ultrasonic Sensor as well.
- Line Detection: Both the LiDar sensor and the Camera sensor pose better stats than our choice, but we cannot afford both of them because of our budget.
- Movement and propulsion: The brushless motor has worse qualities in every aspect compared to the Thrust Eco, that is the reason why we don't choose it.
- Battery or Power Source: The alkaline battery poses the worst qualities out of the 3, and the Lithium-Ion battery is more expensive compared to other options, so we're going to stick with the Nickel battery.

3.2 Comparison of concept principles

MicroController			
Ramp Detection			
Obstacle Detection			

Detect the line			
Wheels			
Motors			
Battery			

	Speed	Weight	Accuracy	Power Usage	Budget
	10%	15%	35%	20%	20%
Concept 1	++	- -	++	--	- -
Concept 2	-	++	+	+	+
Concept 3	+	+	+	++	++

3.3 Choice of most promising concept principle

Probably there is no single best solution, without any disadvantages. So an argumentation to support your choice is required.

- Microcontroller: We chose “Arduino Uno” as our microcontroller choice because of the ease of use and the cheaper price of this device compared to “Raspberry Pi”. Since our budget was limited to 50 EUR, the Arduino Uno was the most ideal option. Even though Raspberry pi is better in terms of memory, it was over-budget and filled with features requiring higher technical expertise such as Linux knowledge.
- Sensors: We chose “Infrared Sensor” for line detection because it’s budget friendly and how resourceful it is. Another sensor we chose was “Ultrasonic Sensor” for obstacle detection since it can still detect objects even in harsh environments such as dust and smoke. For the ramp detection, we use a “Tilt Sensor” to measure the inclination of the car.
- Power Source: We chose “Nickel Battery” for our power source because they have a higher energy density compared to other rechargeable batteries, allowing them to store more energy per unit volume. This attribute makes them suitable for powering portable electronic devices. Besides, Nickel batteries are less harmful to the environment than traditional disposable batteries.
- Wheels: We will use the wheels that are given to us since they can manoeuvre on the track smoother, which can be beneficial as we could get past the obstacle more easily. Moreover, we don't have to spend extra budget on new wheels, which can be saved for other components
- Motor: We will be using the motor provided by the client, since it is easily modified and has all of the required technical specifications needed for our car, which could help us to have more time to focus on other components that we could face problems later on.
- Coding Language: We will be using “C++” to code since we have background using it on our previous project.

4. Elaboration of chosen principle to functional design

4.1 Overview of functional design

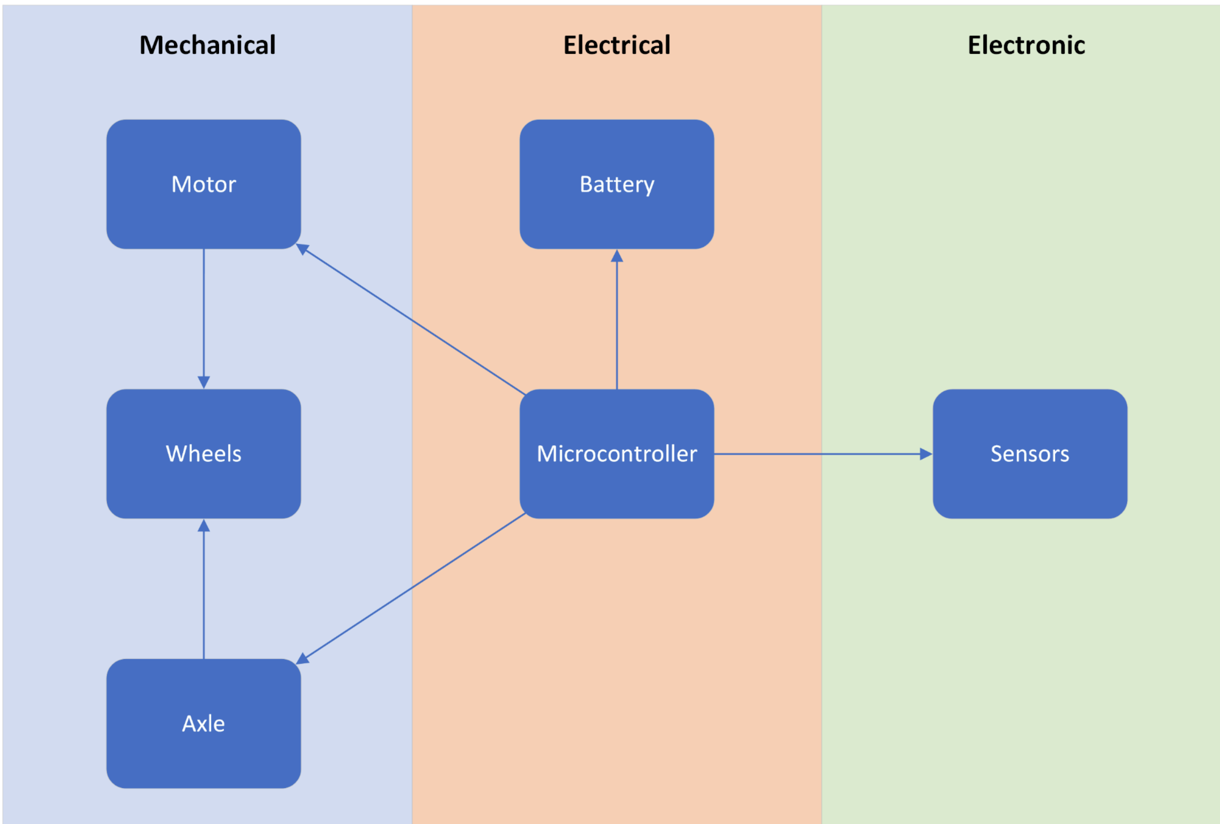


Figure 4.1 Block Diagram for our Project System

As given in the block diagram (Figure 4.1) for the overall project, the functional design implemented in the overall project can be divided into different aspects of the project. The aspects are software, mechanical, electrical and electronic. Figure 4.2 illustrates the High-level flowchart.

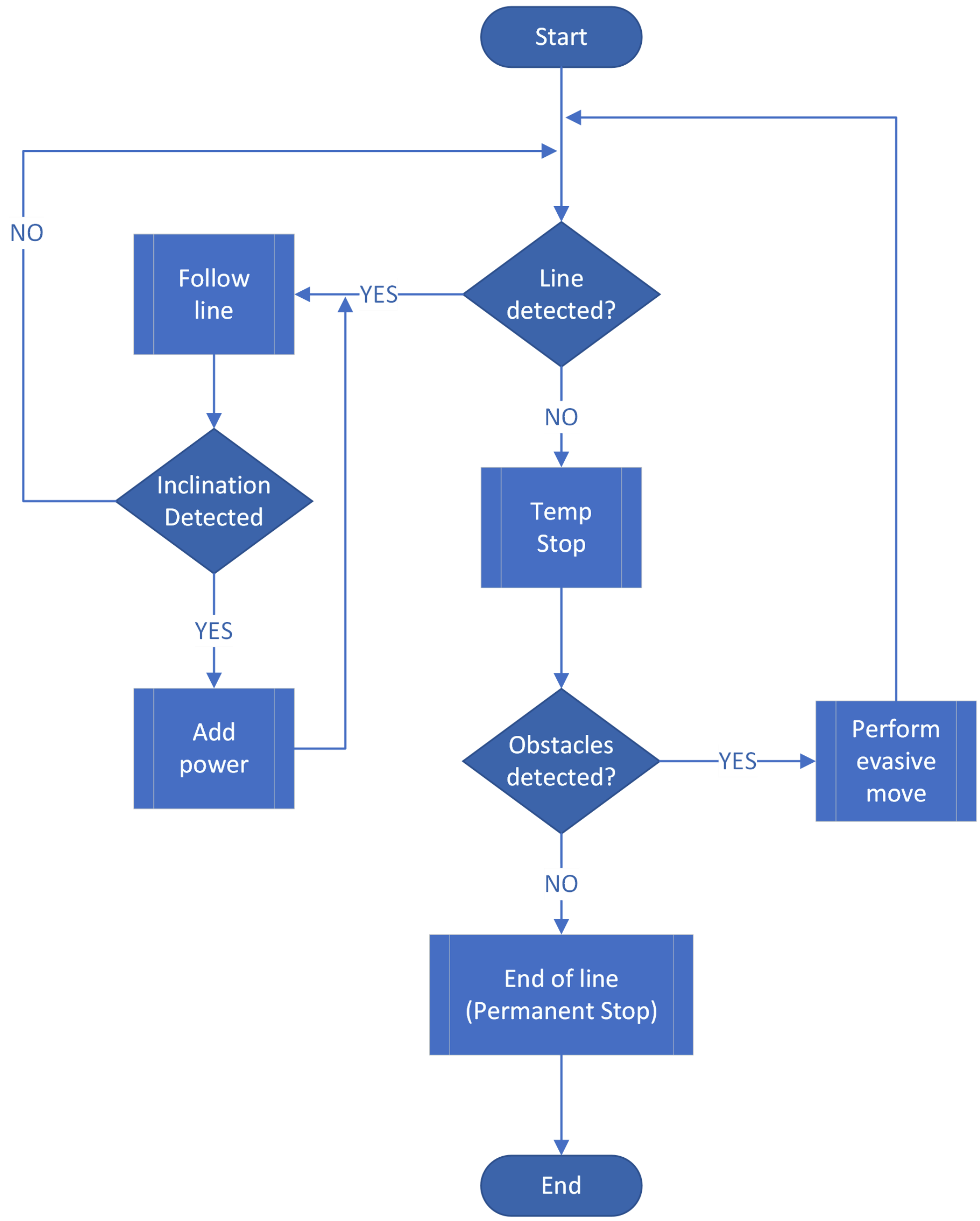


Figure 4.2 The High-level Flowchart

Safety Philosophy

For the safety philosophy of this project, backup of ideas, components or people is kept in mind. If in the duration of building the electric car, components are incompatible or damaged, a backup plan using other available components is kept. This “Plan B” is the culmination of all ideas that didn't make it to the final decision for the project.

Standards and Measures

- **Accuracy:** To accurately measure the white line given to follow and also detect obstacles. The measures for this are + or - 10 cm away from the centre of the white line and not less than 5 cm away from an obstacle, the car should stop.
- **Efficiency:** The car, according to system requirements, has to meet the requirement of running the car for 30 minutes.
    - Hence battery requirements have to be 3000 mAh if operating current is within 2000 mA

$$\text{Battery Life} = \frac{\text{Battery Capacity (mAh)}}{\text{Load Current (mA)}}$$

Figure 4.3 Formula of Battery

4.2 Elaboration of functional design

4.2.1 Mechanical

Frames

The chassis of the electric car will be the same as given. A 3D printed cover over the electric car will be present. This is because to keep the components and wiring of the car protected and also to give the upper section of the car strength and a frame. The weight of the 3D printed section will not exceed 500 grams.

Mechanisms

The car must have enough suspension to handle bumps or small objects less than 5 cm high. It must have a centre of mass near the bottom centre of the chassis to handle the centrifugal force of the turn radius.

Furthermore there are many manually added mechanisms that are added by software which control the mechanical components. This is used for

- The **deviation from the white line** to avoid obstacles. The approximate deviation from the white line should not exceed more than 20 cm in order to have appropriate time for the car to detect the white line after avoidance manoeuvre.
- **Increase in torque** to overcome the ramp/inclination. The calculation for those requirements are:

$$Torque = Force \ X \ Perpendicular \ Radius \ Vector$$

Where Force is the weight of the car 3kg force and perpendicular radius vector of approx 0.1 m.

Hence torque requirements comes to 0.3 Newton metre that should be provided from the motor.

Drives

The car is electric and has a DC motor (Figure 3.3), hence the transmission of the car is automatic. Also a manual transmission system is not required since the top speed of the car cannot exceed 20 km/h.

Figure 4.4 DC motor for high torque requirements. Suitable for racing electric cars.

4.2.1 Electrical

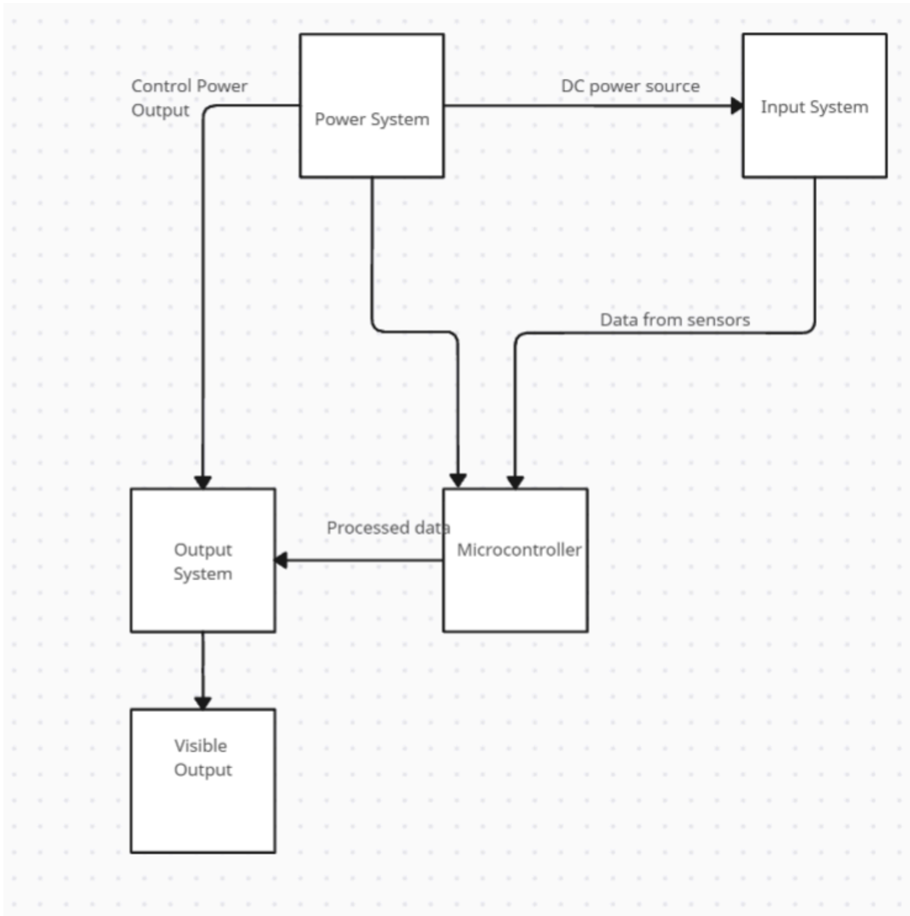


Figure 4.5 Block Diagram of the Combination of Electrical System

The above diagram (Figure 4.2) shows the overall design of the system and all the subsystems working together to make the electrical work of the car.

The processes/subprocesses are:

1. The input system is powered by the power system. The power system consists of the battery which powers all components.
2. The input system is a system that takes input from hardware sources and converts it to digital data.
3. The microcontroller controls the processing of data and the flow of data in a certain direction. Eg. Input system to Output System.
4. The power system combined with the microcontrollers make sure correct power output is given at the right time.

4.3.1 Electronic.

Processing Platform

For the controlling of the car, a microcontroller would be used as it would enable autonomous running and looping of code to run continuously as opposed to it connected to a PC where a constant connection to the PC would be required. Many microcontrollers like Arduino Nano, Arduino Uno or Raspberry Pi can be used for this. For this project, Arduino Uno is the first and best choice.

I/O and other info

The microcontroller would be running at a baud rate of 9600 clock cycles per second but can be increased to 115200 if deemed necessary or for smooth functioning and coordination between components. The data from input and output will be handled by the microprocessor itself since all components will be connected to the microcontroller directly or indirectly.

Measurement Systems

The measurement for accuracy in the car would be how much deviation from the white line the car has taken. This will be judged by the IR sensor embedded in the car. The deviation can be measured from the output data from the sensor and a few lines of code. In a similar way the location and awareness of the surrounding can be measured the same way. The ultrasound sensor can be used for confirmation of the objects in the surrounding.

4.3.2 Software

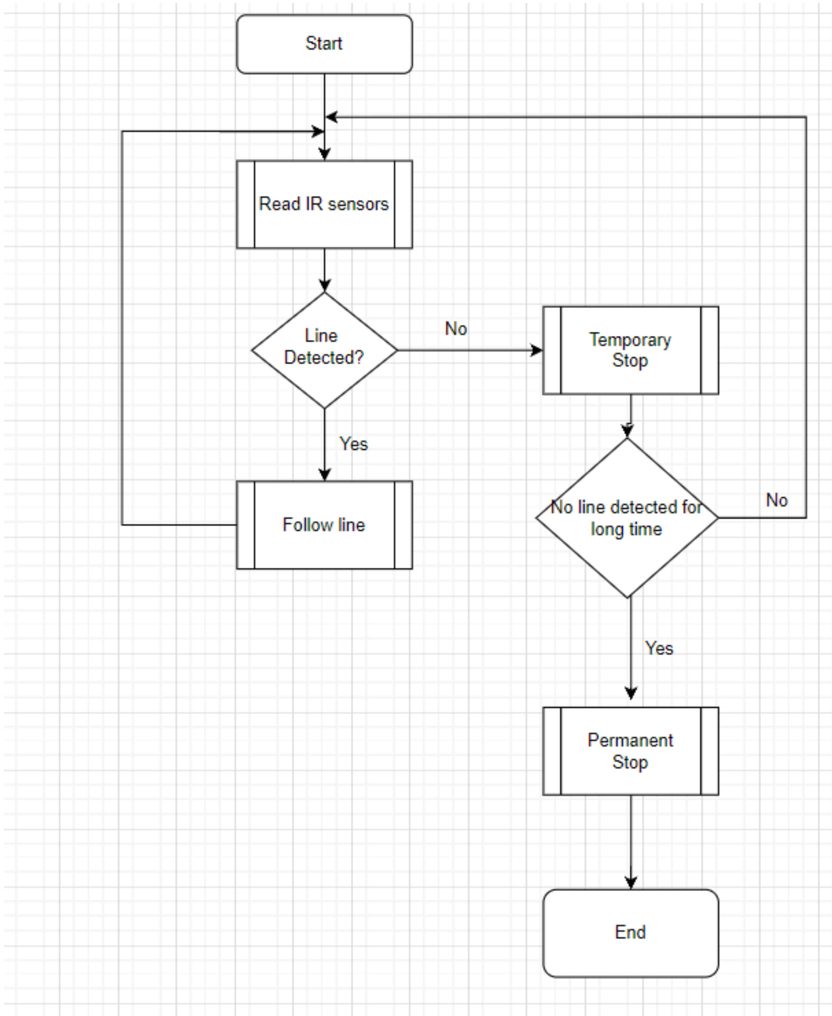


Figure 4.6 White Line Detection

Fig 4.6 illustrates the flowchart above gives a functional overview of the line detection system of our project and how code will execute the process. It describes a loop where constant input is read from the IR sensor. If the line is detected, the car will follow, otherwise the car will evaluate whether the line has reached a permanent end or a simply a temporary end.

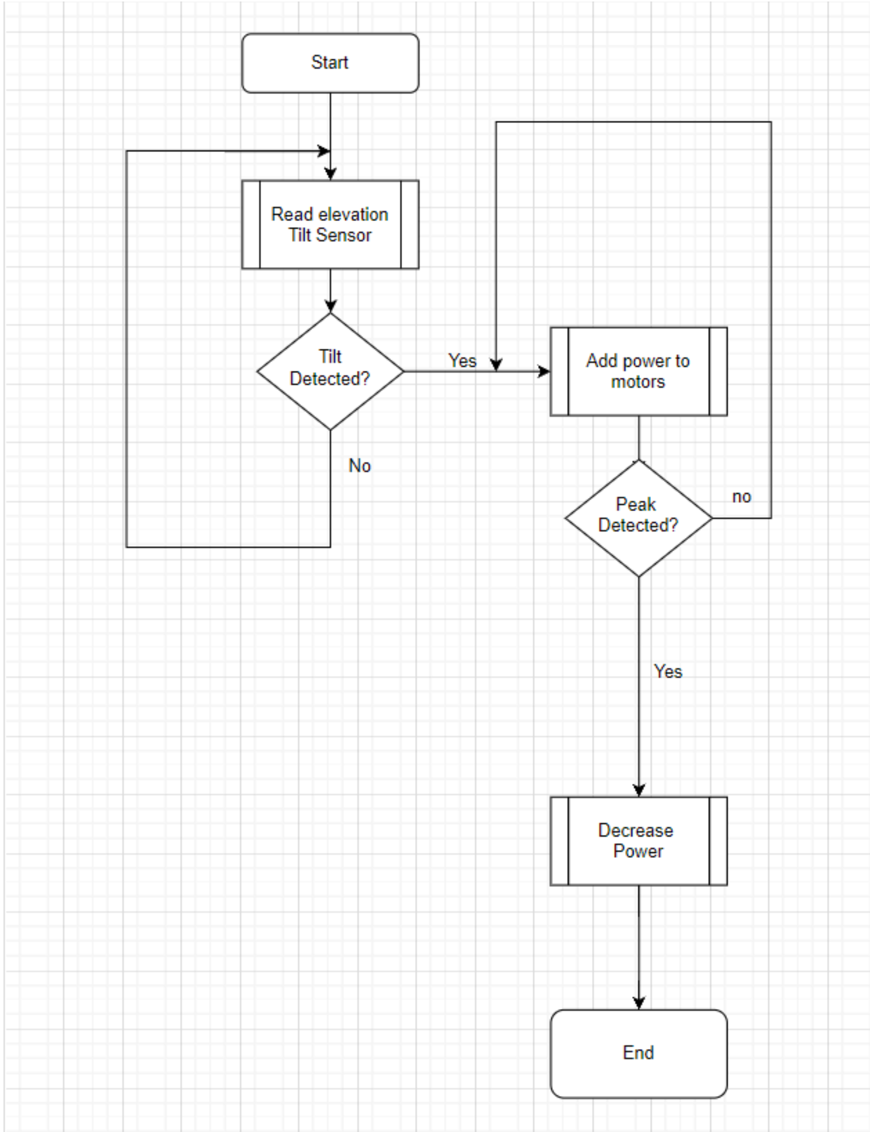


Figure 4.7 Ramp Detection

Figure 4.7 demonstrates the flow of a Microcontroller receiving the signal from the Tilt Sensor. When the inclination is detected, the Arduino Uno will accelerate the motor to overcome the ramp. Next, the Tilt Sensor will check whether the car is already at the peak of the ramp or not. If yes, the car will decelerate the motor, and otherwise.

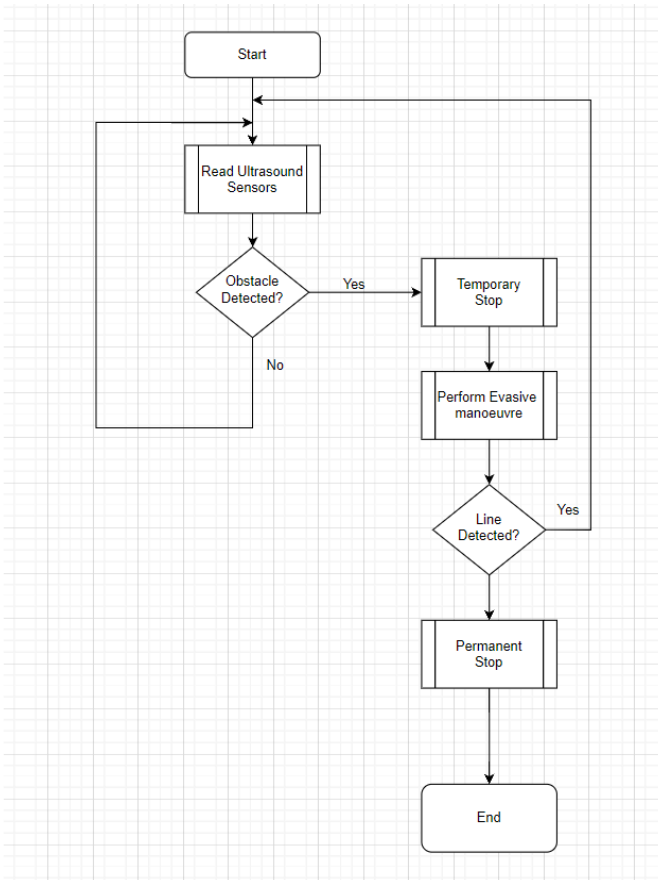


Figure 4.8 Obstacle detection

Figure 4.8 illustrates the process of detecting obstacles. The Ultrasound sensor will be read to check if the obstacle has been detected, if it has not, then it will go in a loop until the obstacle is found. If the obstacle is detected, then the car will be put to a temporary stop, and then perform an evasive manoeuvre to avoid the obstacle. It will constantly check if the line is detected while manoeuvring to make sure that it will still follow the line afterwards.

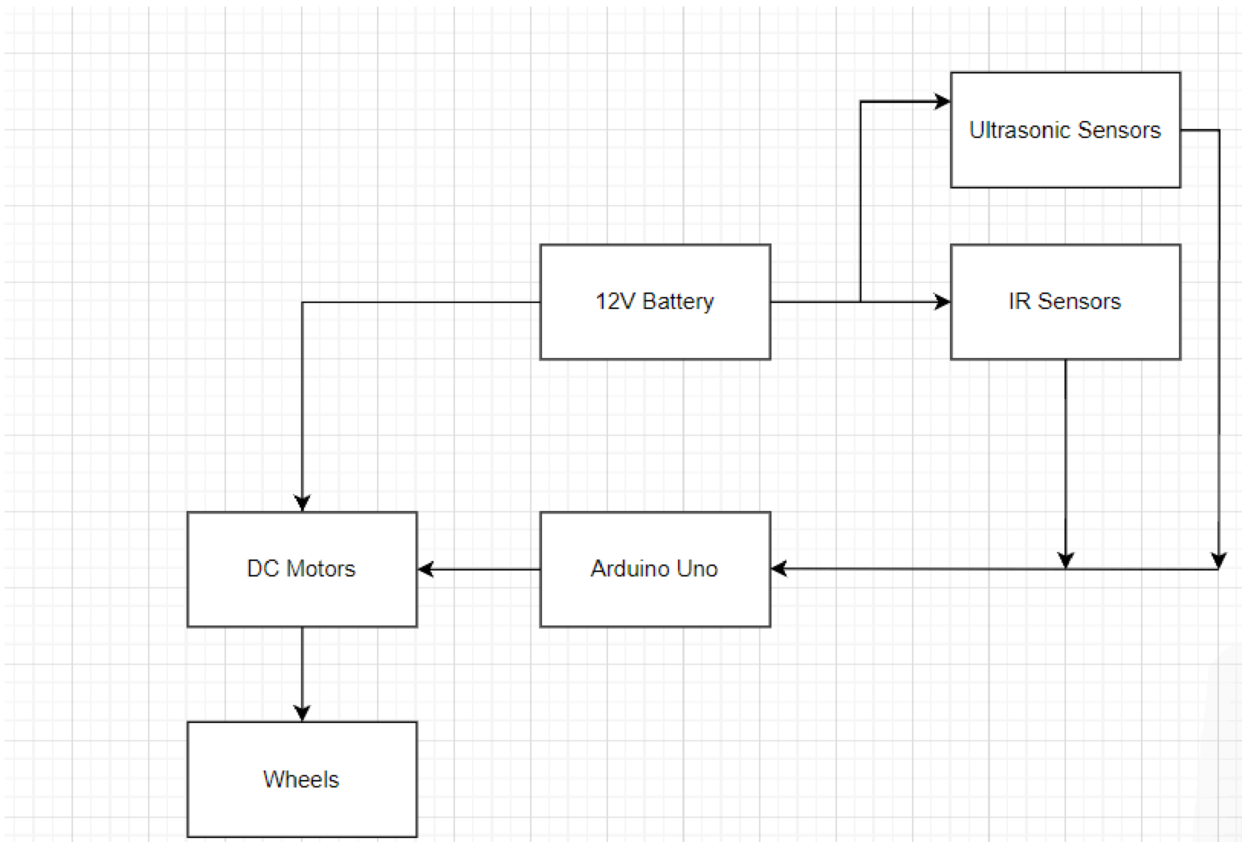
Programming Language

For the microcontroller in the car, the coding language to be used is CPP or its variation of INO for Arduino-related projects. This along with the Arduino Integrated Development Environment for the most optimum coding environment for this project. CPP is a low-level language with high control over memory and hence is the best language for this.

References

Number	Author(s)	Title
[1]	Saxion School	2324_Module_Description-Project System.pdf
[2]	Dreamvu	All You Need To Know About Obstacle Detection Sensor
[3]	Arduino Modules	KY-020 Tilt Switch Module - ArduinoModulesInfo
[4]	Paul McWhorter	9-Axis IMU LESSON 26: Understanding PID Control systems with Arduino (youtube.com)
[5]	Muhammad Ansar	How to Make Line Follower with Obstacle Avoiding Robot using Arduino and L298 Motor Driver (youtube.com)
[6]	Muhammad Ansar	Line Follower Obstacle Avoiding Robot Using Arduino And L298 Motor Driver – MArobotic
[7]	Haase Industries	Are PID line followers ACTUALLY faster? - YouTube
[8]	Maker 101	Line Following / Tracking Robot Car (Arduino and QTR-8 Sensor) (youtube.com)
[9]	Zinbal	30. What is IR Sensor | How IR Sensor works | Working principle of IR Sensor. - YouTube
[10]	mybotic	How to Use TCRT5000 IR Sensor Module With Arduino UNO : 7 Steps (with Pictures) - Instructables
[11]	Arduino Project Hub	Using IR Sensor (TCRT 5000) With Arduino | Arduino Project Hub
[12]	Robu	IR sensor Working Principle and Applications | Robu.in
[13]	CircuitMix	What is an IR Proximity Sensor? - CircuitMix
[14]	Rebecca Skantar	Obstacle Detection Methods
[15]	Circuit Basics	How to Use Obstacle Avoidance and Tracking Sensors on an Arduino - Ultimate Guide to the Arduino #36

Appendix A:


Ultrasonic Sensors
12V Battery
IR Sensors
DC Motors
Arduino Uno
Wheels