Final Report

Project system

By: Velos Company

Version: 1.0

Enschede, 28th May 2021

Team Members

Place, date:	Enschede, 10th February 2021		
Created by:	Project group: Velos		
	Lucas Lalande	495230	495230@student.saxion.nl
	Mahnaz Zarrinfar	496962	496962@student.saxion.nl
	Stelios Diamantopoulos	494002	494002@student.saxion.nl
	Hlib Hryshko	493764	493764@student.saxion.nl
Version	1.0		
Clients	Teachers of Saxion University of Applied Sciences		

Table of Contents

Table of Figures V
Abbreviations VI
1. Introduction 1
1.1 Background 1
1.2 Purpose of the assignment 1
1.3 Methodological approach 2
1.4 Structure 3
2. Project objectives 3
3. **Project activities** 5
3.1 Initiation and planning phase: 5
3.2 Designing Phase 5
3.3 Implementation and Assembling phase: 6
3.4 Testing and Quality Control phase: 6
4. Project boundaries 6
4.1 Product Boundaries 7
4.2 Process Boundaries 8
5. Quality assurance 9
6. Project organization 10
6.1 Organization 10
6.2 Information 12
7. Planning 12
8. Costs and Benefits 13
9. Risk analysis 15
10. Functional Design 15
10.1 Introduction 15
10.2 Analysis of Requirements 15
10.2.1 Geographical Analysis 15
10.2.2 Functional Analysis 17
10.2.3 Technology Analysis 18
10.2.4 Technical Analysis 19
10.3 Selecting a Concept 24
10.3.1 Elaboration of Chosen Concept 26
10.4 Testing Procedure 26

11. Technical Design 32
11.1. Introduction 32
11.2. Mechanical Design 32
11.3. Electrical Design 37
Component List 37
11.4. Electronics 39
11.5. Software 46
12. Result 47
12.1 Introduction 47
12.2 Mechanical Design 47
12.3 Software 50
12.4 Electrical design 53
12.5 Final Design 54
12.6 Final Result 54
13. Reflection on final design 55
14. Conclusion 56
Bibliography 57
Appendix A 58
Appendix B 60
Appendix C 61
Appendix D 65

Table of Figures

Figure 1. V model 2
Figure 2. SMART Objectives 4
Figure 3. MoSCoW Method 7
Figure 4. Cost and Benefits 13
Figure 5. Geographical Analysis 16
Figure 6. Functional Analysis 17
Figure 7. IR sensor-a 19
Figure 8. IR sensor-b 20
Figure 9. Ultrasound Sensor 20
Figure 10. Accelerometer 21
Figure 11. Guroscope 22
Figure 12. ADC Unit 22
Figure 13. PWM Modulation 22
Figure 14. DC Motor 23
Figure 15. Morphologic Design 24
Figure 16. QTR-8A testing setup 27
Figure 17. Front of the car 2D sketch 33
Figure 18 Support front piece ledges 2D sketch 33
Figure 19 Front top 3D View 34
Figure 20 Front 3D Bottom view 34
Figure 21 Front side 3D View 34
Figure 22 Back attach 2D sketch 35
Figure 23 Back attach side 3D view 35
Figure 24 Back attach side 3D view 35
Figure 25 Back attach top 3D view 36
Figure 26 Electrical Design 37
Figure 27 Brushed DC Motor 39
Figure 28 Motor Shield 40
Figure 29 H-Bridge module 40
Figure 30 Servo Motor 41
Figure 31 QTR-8A Sensor Array 42
Figure 32 Ultrasonic Distance Sensor 43
Figure 33 Arduino Uno 43
Figure 34 Gyroscope and Accelerometer 44
Figure 35 LED Strips 45
Figure 36 User Interface PCB 46
Figure 37 Detailed flowchart 47
Figure 38. Top of car shield 48
Figure 39. Front of car shield 48
Figure 40. middle part of car shield 49
Figure 41. side of car shield 49
Figure 42. side of car shield 49

Abbreviations

IPX2: Can resist water that hits the product at a 15° angle or less.

V-model: Verification and Validation model

1. Introduction

Projects usually consist of 4 different phases:

- Initialization
- Planning
- Execution
- Work evaluation

This document is a record of every phase and the steps of every phase.

1.1 Background

The company Velos is a young company of engineers and has been created with the purpose of designing the fastest autonomous racing cars. The engineers, founders and members of the company have a technical background and experience in embedded system design, electronic systems, electrical circuits and software design. The members worked previously on diverse projects more precisely on software design, Printed circuit board designs, electrical circuits, and with diverse microcontrollers. The members all have experience in companies and all participated in the building and design of an intelligent safe. We founded this company because we believe in each other's competences and want to insure a product of the highest quality.

1.2 Purpose of the assignment

This project is part of the study program and initiated by pedagogues in order to teach us the implementation of new electronic parts in order to make an autonomous car. In addition, this project is useful for us to practice using the V-MODEL and the tools we would use in a company (see Figure 1.1), MoSCoW (see Figure 1.3), writing SMART objectives (see Figure 1.2), working with the group and overall writing the project plan and reports.

1.3 Methodological approach

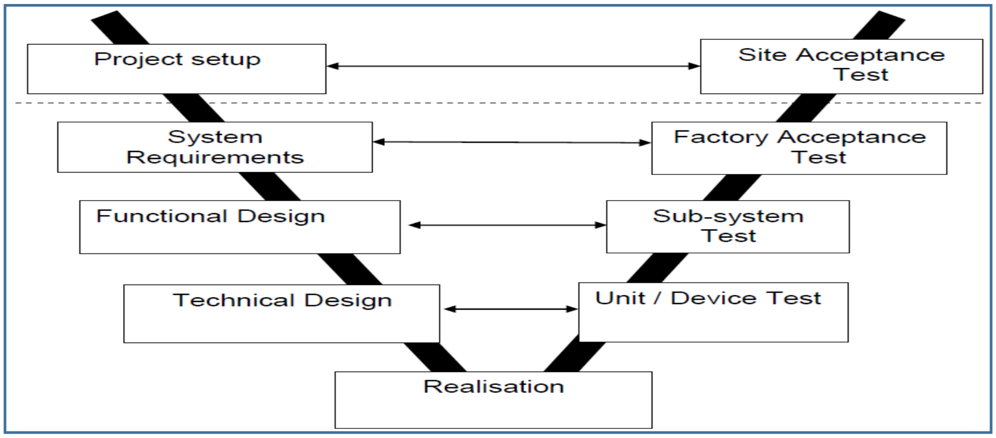


Figure 1. V model

Project Setup

In this step the team makes sure to understand the project outcomes and requirements and what are the anticipated benefits. Also, the team make the decision about every person's responsibilities and roles throughout this project.

System Requirements

In this phase, the focus of the team is on understanding the client's demands and requirements. The requirements phase identifies the functionality, performance levels, and other characteristics which the product must satisfy in order for it to be acceptable to the customer

Functional Design

In this step, the engineers' team analyzes the requirements in detail and divides the whole system into sub-systems. Every sub-system then will be analyzed separately, and basic concept ideas will be generated. After thoroughly examining all the ideas, the best concept will be chosen. Subsequently, the team will design the functional diagrams for each sub-system.

Technical Design

In this phase, the team creates all the needed schematics and simulations in detail. The technical design should be in a way that is possible to make the units in the next step, and the software engineers can start creating the code.

Realization

In this stage, the team will start building the modules and sub-units. The mechanical part will be ordered, and the team will also design and order the PCB. For the software part, the engineers will start writing code for modules.

Unit Testing

This is the first stage of a step-wise testing process. To be sure about the project's functionality, the team plans a testing phase after the developing stage. Each module will be tested separately in this step to make sure they are working correctly and according to the desired plan.

Sub-system Testing

In this stage, different modules will be put together to make a sub-system, e.g. Mechanical, Electronics, Electrical, and Software sub-systems. They will then be tested together to find the faults and errors in the interactions between modules.

Factory Acceptance Test

In this stage, the team will integrate the sub-units and form the final project. Afterward, the engineers will test the complete system and fix the possible errors and faults.

Site Acceptance Test

At this stage, the system will be tested to see if it meets the requirements made by the client and documented in the requirement analysis phase. The client uses this phase to validate the final product.

1.4 Structure

There are two important roles in this project. The first role is the supervisors, who are performing the role of the customers to prepare us for the real world, which we will face after we end our education. The second role is the engineers. 2 of them are from the ACS department and 2 from the EEE department, their task is the creation of the engineering company and the implementation of an autonomous car with some specific options.

2. Project objectives

We come together as a group of four engineers who specialize in applied computer science and electrical engineering on this project. The reason behind this project is for us to utilize our knowledge and skills in designing circuits, electrical and mechanical measurements, and software coding, as well as experiencing teamwork in an organized and structured way.

As presented to us, the teachers' final desired objective is the Design and realization of an Autonomous Electrical Racing Car.

As we work on this project throughout the third and fourth quarter, there are sub-objectives that we have to focus on, design, implement, and deliver based on a pre-specified time basis.

It is important to have the goals of project according to the SMART protocol:

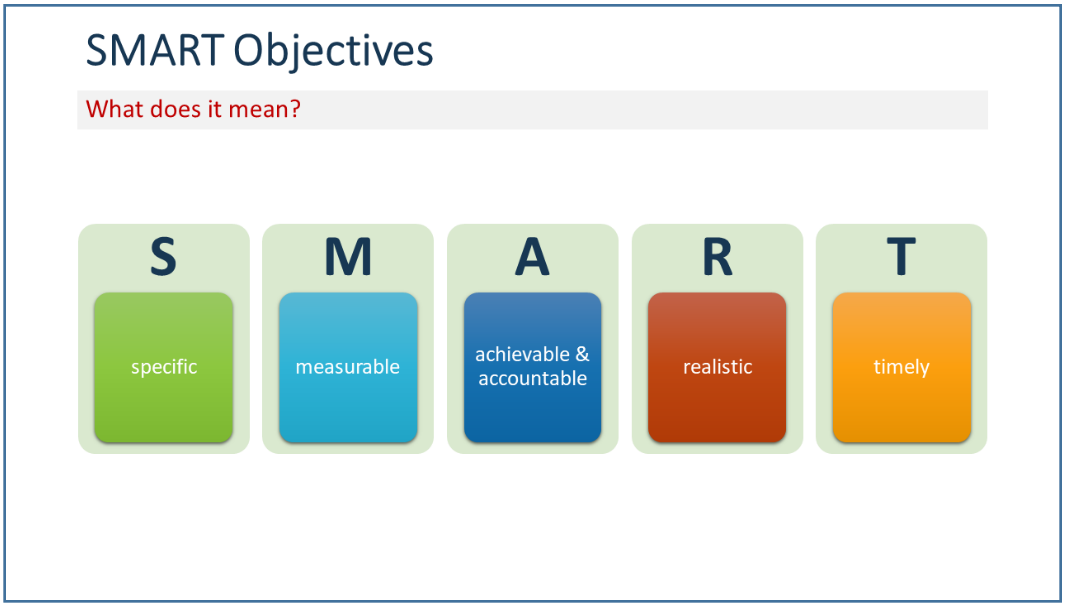


Figure 2. SMART Objectives

The sub-objectives presented to us by teachers are specific and measurable. There is a pre-specified timeline for the project, and there are deadlines for every unit. That is what makes the sub-objectives of this project time based. The objectives also should be attained when the project is finished and presented at the final race. Also based on our sets of skills and knowledge, these objectives are realistic and consequently attainable.

1. ride autonomously
2. has to work for 30 minutes
3. fit in a box of 400mm x 250mm x 200mm
4. must follow the white line (20mm)
5. stop at the end of the track
6. Avoiding obstacles
7. Finding the white line
8. have a body (3D printed separate parts)

This project's final result is an autonomous racing car with all the requirements and a final project plan that documents every step of the way and our methodological design. All the sub-objectives will be achieved with the final product.

3. Project activities

To deliver an excellent final product that stands up to our standards, we need to devise a comprehensive list of activities over a fixed period. A successful project is a combination of four major phases.

3.1 Initiation and planning phase:

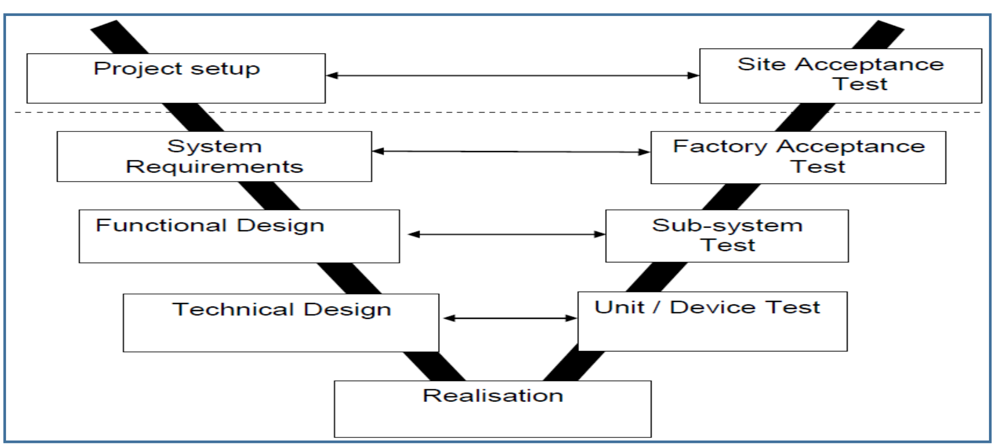
The first step to completing a project successfully is to get to know the project's goal and desired outcome. After the initial introduction to the main idea, our team dives deeper into researching the requirements and deliverables to understand our future car's specifics better.

After researching the objectives and sub-objectives, our team estimates the workload and the time needed for each step of the way. Considering the deadlines, this team will devise a detailed plan by using a Gant Chart. The engineers will also use Scrum as a tool for documenting and managing the work. Finally, a complete project plan is made, using a V-model methodological design that will help to know the course of action throughout this project.

3.2 Designing Phase

After learning and researching every detail of requirements carefully, the team will start with the project's designing phase. It is important to work according to the V-model and in a methodological way. The team will start working from global to detail and begin with more functional designs, and then it is possible to continue to the more technical design. The methodological design is applied in order to:

- minimize the project risks
- be able to improve and guarantee the quality of product
- Reduce the total cost of the project
- Improveme Communication between all Stakeholders



3.3 Implementation and Assembling phase:

Once the team finish the technical design, it is time to move forward to making mechanical units and assembling them. Additionally, the engineers will order our needed electrical components and PCB. Subsequently, the team proceeds with implementing all the mechanical, electronics, and software units together. According to the V-model, this is the realization step.

3.4 Testing and Quality Control phase:

After managing the design steps with V-Model, the team needs to identify problems that threaten to undermine our project's result. To ensure the project's functionality, a testing phase is planned for each unit and sub-unit after the development stage. According to the V-model methodological design, some testing phase stages are Unit testing, Integration Testing, Acceptance and system testing.

In the final step of the testing and quality control phase, the team will thoroughly examine the whole system to ensure that every requirement is met and every option is integrated correctly.

4. Project boundaries

For this project, the engineering team will use the MoSCoW method to organize our objectives and goals. The MoSCoW method is a prioritization technique for managing requirements during development phase. It stands for four categories of importance:

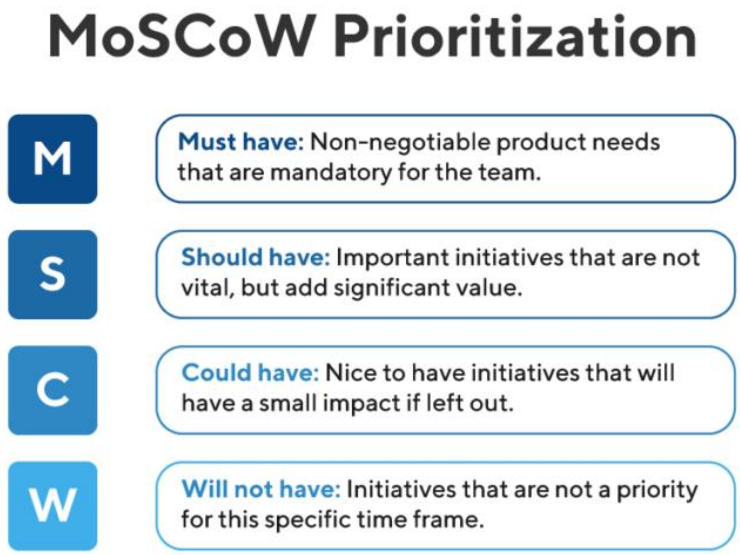


Figure 3. MoSCoW Method

4.1 Product Boundaries

Must have:

- Ride autonomously
- Has to work for 30 minutes
- Fit in a box of 400mm x 250mm x 200mm
- Must follow the white line (20mm)
- Stop at the end of the track
- Avoiding obstacles
- Finding the white line
- Have a body (3d printed separate parts)

Should have:

- 3D printed shell for car
- 3D printed part to seal the top of car
- LED indicator to show the car is on
- Celebratory spin at the end of the track

Could have:

- Car could work in the dark

- IPX2 Water Proof model

Will not have:

- Bluetooth module
- Remote control

4.2 Process Boundaries

Must have:

- Project Plan
- Functional Design
- Technical Design
- Deadlines according to the client
- Scrum according to the deadlines and tasks
- Go- No Go meeting with client

Should have:

- Extra Team meetings
- Agenda of Extra team meetings

Could have:

- Extra meetings with client to report the progress
- Agenda of Extra team meetings with clients

Will not have:

- Any Change in the client's requirements after the Go – No Go meeting

5. Quality assurance

In order to get this project done, the engineers used several tools and techniques to insure the quality of the final task. First, every module that is produced by any third party is submitted to the criticizm of the whole team and needs to be an unanimous agreement.

In addition to that the team have set up a weekly deadline in order to constantly check the quality of the produced work aswell as having time for extra corrections and help from other members of the team.

If one of the member of the team feels or is unnable to complete a task, the task is then passed to another member of the group for help.

Our decided to communicate rationally using different message services in order to check how the assigned tasks are being completed (progression of work).

The team previewed to contact the client bi-monthly to make sure that the product delivered is up to the client's standards.

The engineers are also using a gantt chart for overall task management and sharing.

Every standard from the client is included and there are not any adjustments in that area.

Our company has enough securities to insure the completion of the core requirements. If there is issues and obstacles delaying some modules, the team will make concessions on the extra modules (sofisticated 3d designs, Obstacle recognition ….).

Our Company wants to produce the model for international commerce. The Project will follow the ISO (International Organization of Standardization) for 3 main purposes

- Improve performance
- Better Resource Management
- Achieve quality that is recognized globally

ISO Standards

- ISO/TC 299 –ROBOTICS ''https://www.iso.org/committee/5915511.html''
- ISO 22006:2009 –Quality management ''https://www.iso.org/standard/70397.html''
- ISO 30301:2019 –Information and Documentation ''https://www.iso.org/standard/74292.html''
- ISO 90003:2018 –Software Engineering '' ''https://www.iso.org/standard/74348.html''

6. Project organization

6.1 Organization

To begin with, here is the list of our group members:

Table 1: List of group members

No	Role	Email Address	Telephone Number
1.	Team Leader	495230@student.saxion.nl	+31 68 777 2848
2.	Software Engineer	493764@student.saxion.nl	+38 095 598 81 76
3.	Hardware Engineer	494002@student.saxion.nl	+30 69 401 65375
4.	Administrator and Engineer	496962@student.saxion.nl	+31 61 397 4288

During the project each group member will be able to experience working as company member with different roles, tasks and responsibilities. The roles of our group members are:

Team Leader

The team leader, Software and Hardware Engineer. His responsibilities are:

- Software part reviewing
- Hardware parts

Software Engineer

The chief software engineer. His responsibilities are:

- Software parts
- Functional design
- Technical design

Hardware Engineer

The chief hardware engineer and the team coordinator. His responsibilities are:

- Hardware parts
- Team coordination
- Technical design

Administrator and Engineer

The Documentation and Material Administrator, Hardware and Software Engineer. Her responsibilities are:

- Managing report writing
- Hardware parts
- Software part reviewing
- Technical design
- Functional design

6.2 Information

Table 2: List of Coach(es)/Client(s)

No	Name of Coach(es)/Client(s)	Email	Teaching Roles
1	Robert De Groote	e.degroote@saxion.nl	V-Model and Scrum, Programming
2	Ali Yuksel	a.yuksel@saxion.nl	Sensors, Energy and Actuators
3	Ghayoor Gillani	s.g.a.gillani@saxion.nl	Sensors, Energy and Actuators
4	Ramazan Kirmali	r.kirmali@saxion.nl	Programming

Our group have already planned meetings twice a week through the project to report updates and progress of the project. However, there might be unplanned meetings, triggered by our team coordinator and accepted by the team leader. In addition, the company has a meeting with the stakeholders once a week on Wednesday morning.

7. Planning

The organisation of the projects follows a specific frame with predefined weekly meetings and places of meeting. This section is divided into sub-sections describing very accurately how, when and where the meetings will be held in order to reassure our client.

Where:

As a project group the team has the advantage that 75% of the staff live in the same place so meetings can be held very easily

- Meetings are taking place both in school and online
- Sudden meetings may occur from time to time assuming there is something serious going on (Online due to corona rules)

When:

According to our schedule we are going to have:

- Online meetings every Tuesday (11.00 - 11.45)
- Weekly extra update on Friday (30 minutes)
- Physical meetings on Wednesdays (8.30-10.00)
- Meeting 1-2 days before the submission deadline so everyone can give feedback on the final work we need to submit
- Any unexpected meetings that might occur

How:

The team decided to work in a week by week topic system. That means that the engineers will work every phase of the project for 1-2 weeks depending on the schedule. Every week's subject is listed in a common app that is used to keep track of the work and deadlines

Communication and software:

- Common drive (Provided by Saxion): In that drive the engineers share every Document that is of use for the final project
- Monday.com: Is an application that helps keep track of the work for each week as well as a good representation of the schedule and individual work for the project
- Blackboard: In order to find resources, templates and meet with the clients
- Discord/Microsoft Teams: The engineers' team use discord to do online meetings as well as share information with each other.

8. Costs and Benefits

In order to evaluate the costs and the benefits, to define if this project is going to be valuable financially our engineers gathered data. First step was creating an excel sheet with all the potential components and estimated the price using average price from the internet adding a risk of 20% in order to approximate the concrete finale price

	A	B	C	D	E	F	G
1							
2	Description of component	Reference	Estimated price	Risk = 1.2*average price	Estimated quantity	Final price	Total price
3	Car platfrom	ISH-010	8.5	10.2	1	10.2	
4	Wheel motor	RC 390 Ishima	6	7.2	4	28.8	
5	Servo motor	ISH-010-056	10	12	1	12	
6	Battery Li-ON 7.4V, 1500 maH	ISH-010-020	20	24	2	48	
7	Motor shield	VNH2SP30-E	5	6	5	30	
8	light sensor (12 pack)	IR-01	12	14.4	1	14.4	
9	Sound sensor	SR-04	5	6	1	6	
10	Printed body		5	6	1	6	
11							155.4

Figure 4. Cost and Benefits

In addition to the costs for the car the company must pay the employees. The average time that is required to be spent for this project (data from previous year), is 140 hours/worker. Taking in account all the developpement phases(planning, conception, construction, presentation). The minimum wage in the netherlands is, assuming everyone in our group is 21 years old or older (to facilitate speculations). At a rate of 11 Hours a week, therefore 10,80 euros/ hour(data : Governement of the Netherlands website). The company will need to pay each employee an amount of 10.80*140 of 1512 euros. The price of this project is components + salary (with risks taken in acount). 6048 + 155.4 = 6203,4 euros.

The company therefore needs, for benefits to sell the product that is produced at a price higher than 6203,4 euros. A benefit of 50% on the total costs would cover all the risks and leave with ressources for the future projects of the company a price higher is more benefits but our company is focused on high performances for a reasonable price. Therefore the ideal selling price is 6203,4*1.5 = 9305,6 euros.

The benefits for the company would be therefore 3102,2 euros in addition with the minimal salary paid to each member of the team.

From the perspective of a student we are of course not paid. Hour investment are the hours of work we will spend time in this project and the payment/reward is going to be the grade and credits we will potentially receive from the teachers.

9. Risk analysis

In our risk analysis we took under consideration all the possible facrors that might interfere with our project result. After the identification of minor risks we listed them and gave them a weight value(depending on the impact that will have in the final rtesult).Next our project group gave a value to each risk , this value in combination with the weight give us the total risk. FInnaly after all risks has values we calculated the total risk of our project to be 25.74% a pretty low risk means high chance of success.

10. Functional Design

10.1 Introduction

The first step of Functional Design is to analyze all the project's requirements and subsystems from different aspects. After that, the team of engineers can start designing and generating block diagrams and a flowchart to demonstrate how these subsystems will be put together and how they would perform together.

Subsequently, the engineers will develop various concepts and ideas for each requirement and carefully review them to choose the most competent idea. Finally, the client will be presented with a Functional Design Report and the most promising concept.

10.2 Analysis of Requirements

10.2.1 Geographical Analysis

The system can be divided in accordance with the geographical locations the car will encounter during the race. Using different sensors that will exploit the data from the surrounding environment, the car will react accordingly.
When the car sees the white line:
The car will detect a white line and follow it until the end of the race. It will also stop when it detects the white circle at the end of the race.

When the car detects the ramp:
When the car detects the ramp, the speed should change accordingly for two main reasons: first, not to stay stuck on the ramp not being able to climb because it will not have enough power, and second, not lose speed as we want it to be as fast as possible.

When the car detects an obstacle:
When the car detects an obstacle, it will slow down and try to find a free path to go.

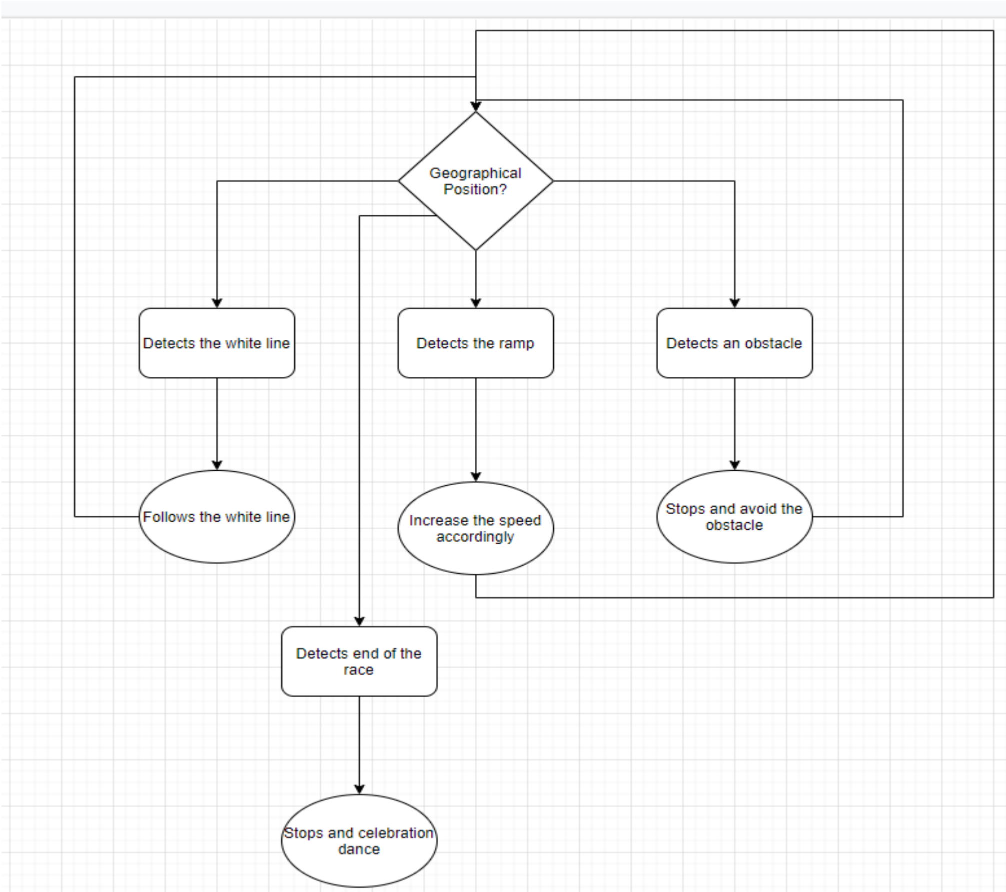


Figure 5. Geographical Analysis

10.2.2 Functional Analysis

This system could be divided into three parts, Input, Processing Unit, and Output. The Input consists of a button and four different sensors: IR Sensor, Sound Sensor, Gyroscope, Accelerometer.

The main processing unit for this autonomous car is the Arduino Uno microcontroller. Additional parts like motor drivers are added to this unit for running the motors.

Finally, this system's output consists of motors responsible for rotating the wheels, alongside four wheels and additional LEDs to display different states of the car.

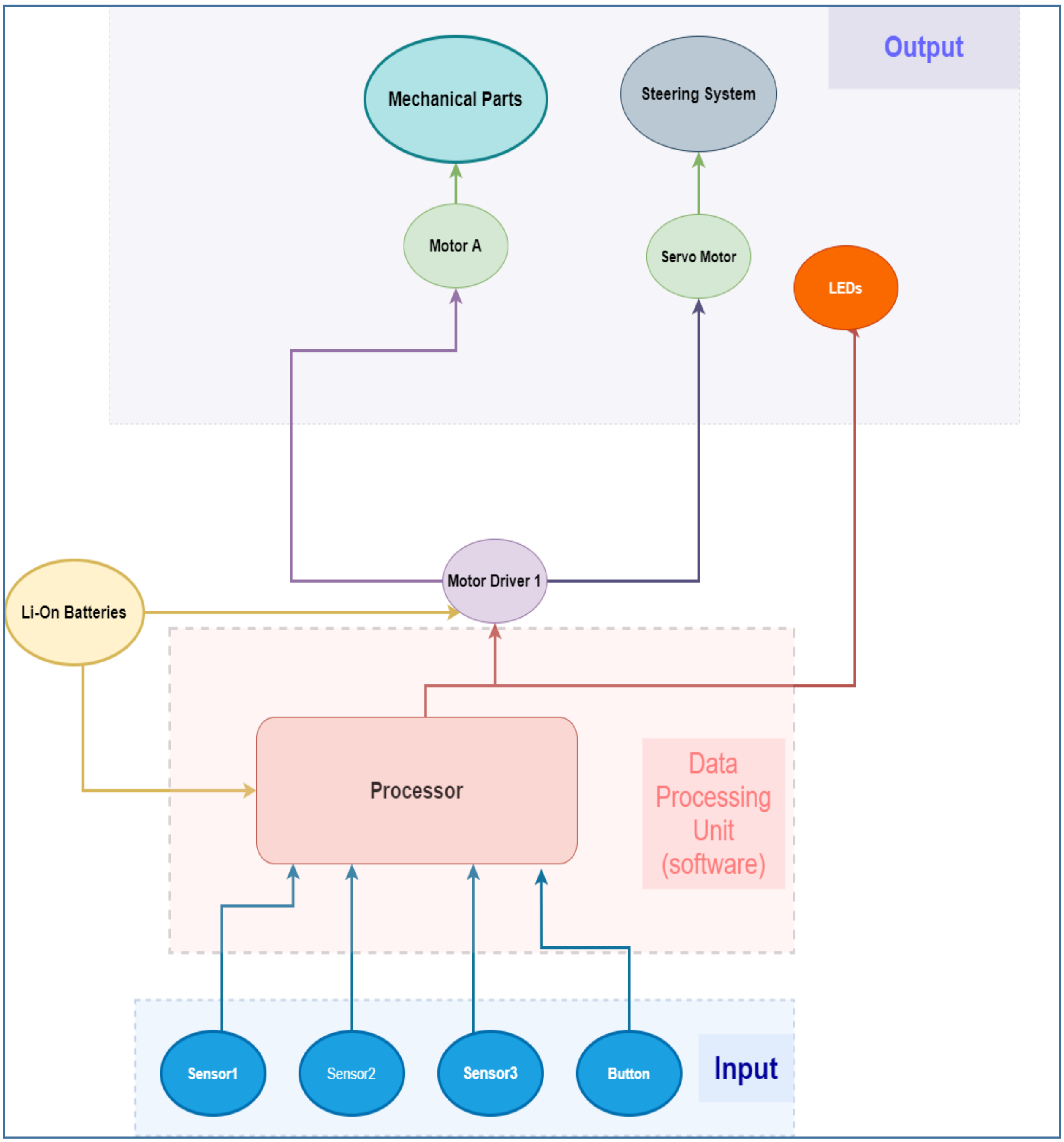


Figure 6. Functional Analysis

10.2.3 Technology Analysis

This autonomous race car consists of various sub-systems, which could be put into four major categories.

- Mechanical
- Electronic
- Electrical
- Software

Mechanical Parts

- Chassis:
  The car will have a chassis that will work as its frame and bears all the other parts.
- Steering System
  The steering system controls the angle of wheels in order to keep the car on the white line.

Electronic

- Servo Motor
  It is a precise motor that allows accurate control of wheels' angle.

- Sound Sensor(detect obstacles)
  A sound sensor is a proximity sensor, meaning it can calculate the distance to an obstacle.

- IR sensor(detect line)
  An IR sensor emits and detects an infrared signal. It can be used to detect black or white by sensing the reflectivity of surfaces. They can also be used to detect obstacles.

- Arduino
  Arduino is a small Microcontroller that is used as the main processing unit for this project.

- Motor driver
  A motor driver is a small Current Amplifier. It is used to take a low-current control signal and turn it into a higher-current signal that can drive a motor

- Brushed DC Motors
  Brushed DC motors are electrical motors that turn electricity into motion. In this project, they are used to turn the wheels and move the car.

- Gyroscope and accelerometer
  A gyroscope sensor can calculate the angular rotation of a vehicle in reference to the earth's surface. This feature is used to detect the ramp.

Electrical

- Li-ion batteries

  A Li-ion battery is a form of an electrochemical cell with a high concentration of stored energy. This is the main power source for the autonomous car.

Software

The software unit consists of code that is compatible with the Arduino microcontroller and controls both inputs and outputs of the microcontroller.

10.2.4 Technical Analysis

Technical Analysis is the analysis of the techniques and concepts we are going to use in order to control every different part of our car to make it work together.
This section includes the algorithms and technologies. The concepts are simplified and illustrated.

Proximity sensor (IR sensors):

A proximity sensor emits an electromagnetic field or a beam of electromagnetic radiation (infrared, for instance), and looks for changes in the field or return signal (Different wavelength signals because of the reflection on different colors).

Reflection on White surface Reflection on Black surface

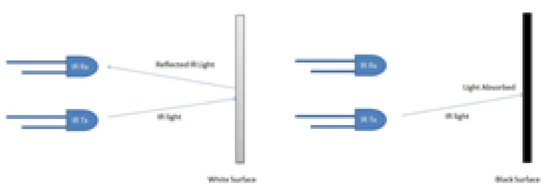


Figure 7. IR sensor-a

In accordance with the number of sensors there will be variation in our algorithm but the concept is the same: We will have two sensors across each side of the line and trying to keep the line within those two sensors and turning according to the data

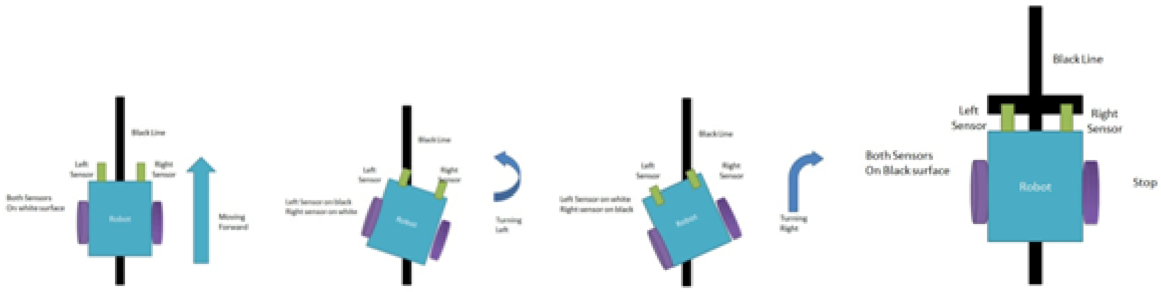


Figure 8. IR sensor-b

Ultrasound sensor:

The sound sensor is a device used to notice the sound. Generally, this device is used to detect the intensity of sound. The accuracy of this sensor can be changed for the ease of usage. Our project will include an ultrasound sensor in order to detect big obstacles on the path of the vehicle. By sending ultrasound waves that will rebound on the object and come back to the sensor that will calculate and convert it to distance, we will be able to calculate the maneuver that we should use to avoid it.

Figure 9. Ultrasound Sensor

Accelerometer:

is a device that measures the vibration, or acceleration of motion of a structure. The force caused by vibration or a change in motion (acceleration) causes the mass to "squeeze" the piezoelectric material which produces an electrical charge that is proportional to the force exerted upon it.

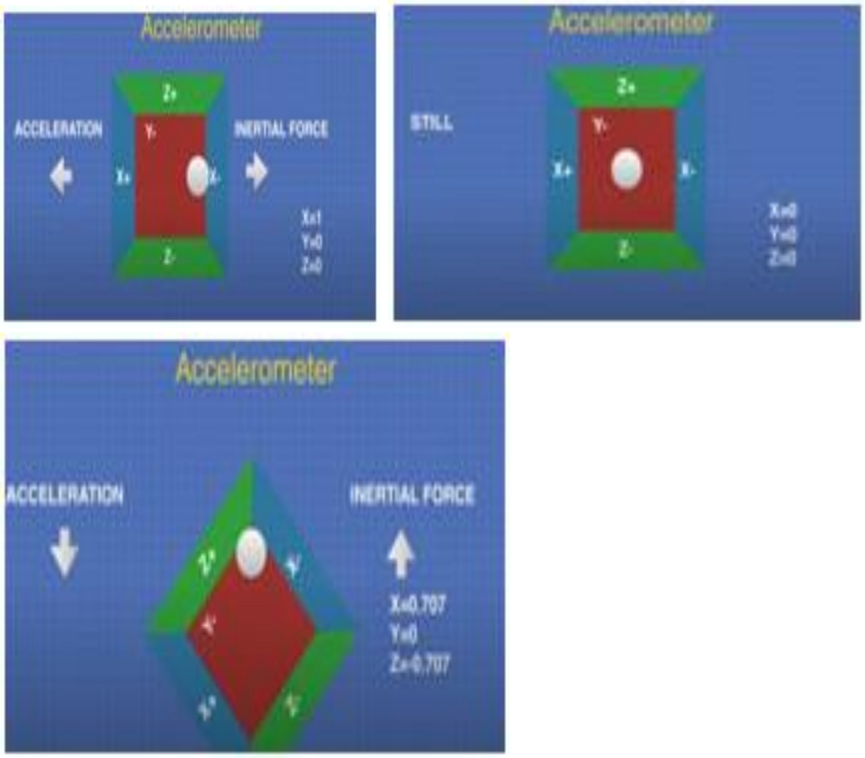


Figure 10. Accelerometer

Micro Electro Mechanical Gyroscope:

MEMS Gyroscope is a device that produces a voltage(16-bit) when an axis is rotated.

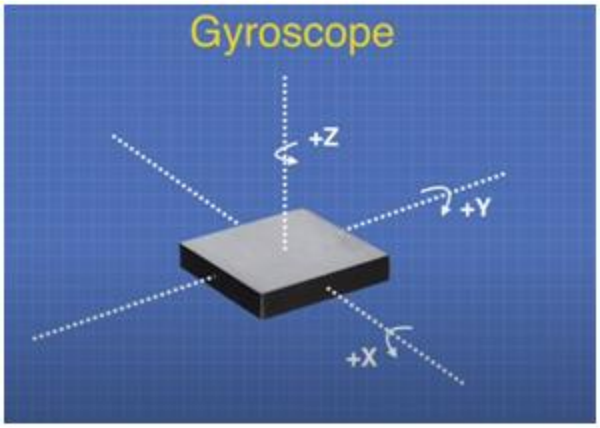


Figure 11. Guroscope

Analog to Digital Conversion (ADC unit):

We will use Analog to digital converter (ADC), converting any analog signal into quantifiable data, which makes it easier to process and store, as well as more accurate and reliable by minimizing errors.

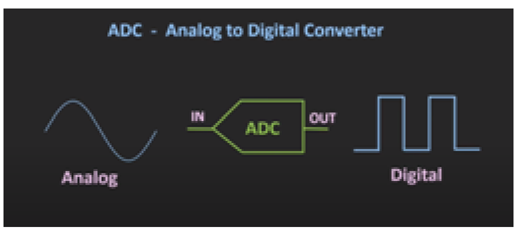


Figure 12. ADC Unit

Pulse Width Modulation PWM:

Pulse width modulation (PWM), or pulse-duration modulation (PDM), is a method of reducing the average power delivered by an electrical signal, by effectively chopping it up into discrete parts.

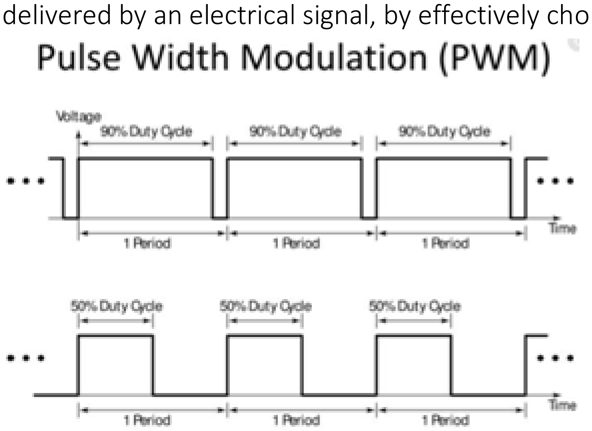


Figure 13. PWM Modulation

H-Bridge:

The H-Bridge is a circuit which can drive a DC motor in forward and reverse. The motor direction is changed by switching the polarity of the voltage in order to turn the motor one way or the other.

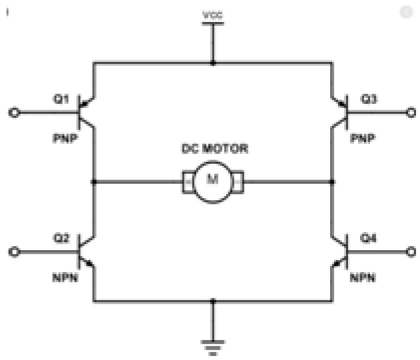


Figure 14. DC Motor

10.3 Selecting a Concept

After defining key parameters with concept-ideas a morphologic diagram [*fig. 3*] can be made.

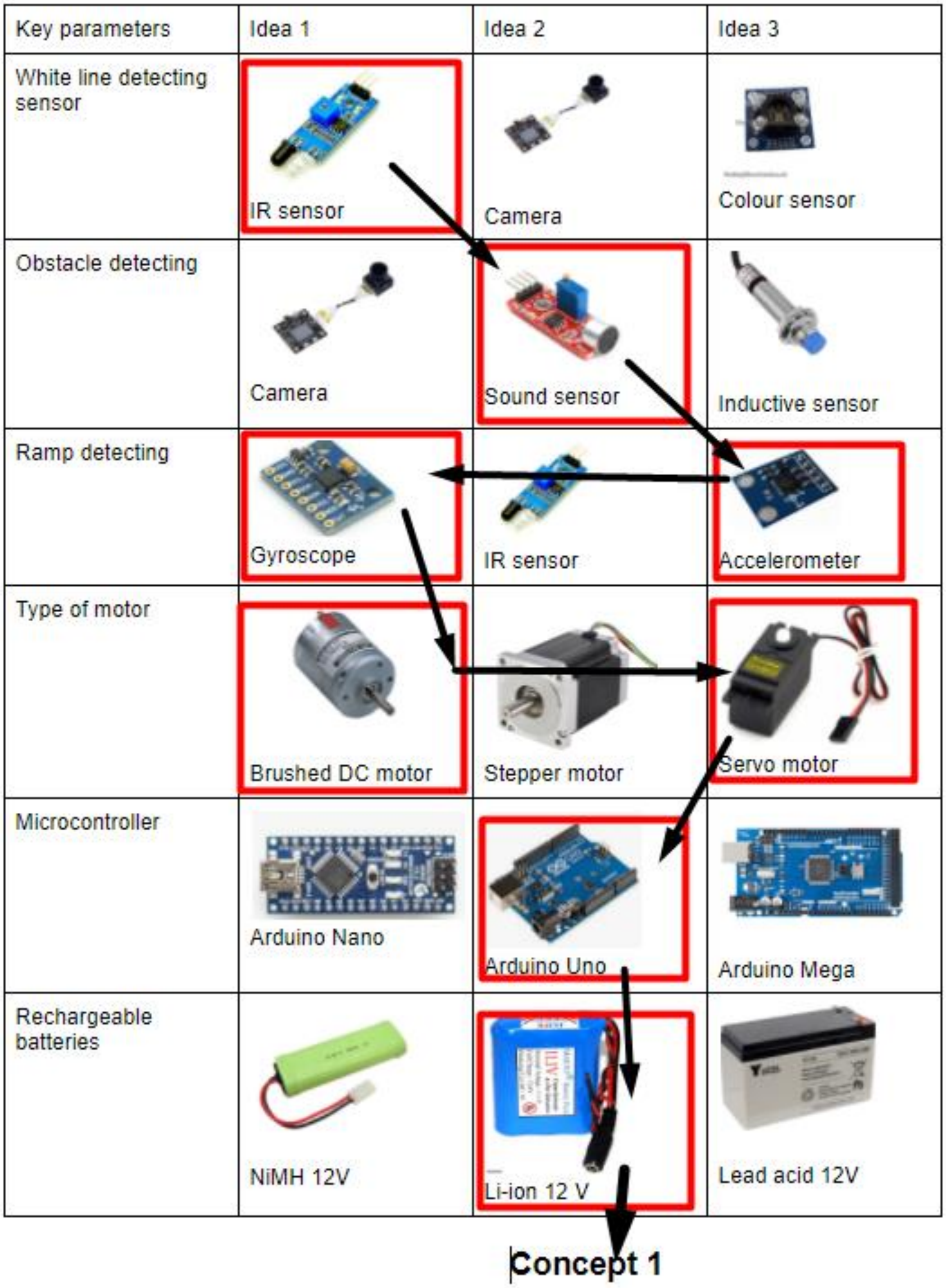


Figure 15. Morphologic Design

Each key parameter has to be evaluated against functional and technical requirements now. A good way to do this is to make each key-parameter a table with concept-ideas and selection criteria. The most promising concept-idea now can be selected.

CONCEPT 1: IR Sensor ++ → Sound Sensor ++ → Gyroscope and Accelerometer ++ → Brushed DC for wheels and Servo motor for steering system ++ → Arduino Uno + → Li-ion battery 12V ++

CONCEPT 2: IR Sensor ++ → Inductive sensor - → Accelerometer ++ → Brushed DC for wheels and Servo motor for steering system ++ → Arduino Mega + → NiMh 12 V battery +

CONCEPT 3: Color Sensor - → Sound Sensor ++ → IR Sensor - → Brushed DC for wheels and Servo motor for steering system ++ → Li-ion battery 12V ++

The company chooses the first concept.

The company chose these components for the project, because the IR sensors are more convenient for software engineers. The sound sensor is one of the best solutions for the obstacle detection. As for the ramp detection, the gyroscope and the accelerometer fit our requirements and will allow us to have more precise data than with any other type of sensors, the main data will be provided by the gyroscope. For the motor type, the team decided to choose the Brushed DC motor for wheels and the Servo motor for the steering system. Choosing the microcontroller was a bit harder, because the Arduino Uno and Arduino Mega are excellent variants, we still are deliberating and will choose accordingly our drivers/ shields since the technology stays the same only the options and pins are different. In the end, the last element to choose was a rechargeable battery type and the company chose the Li-Ion 12V battery since it is the most stable on the market but also the one that is the easiest to manipulate for the power it can deliver.

10.3.1 Elaboration of Chosen Concept

After choosing the most promising concept for this project, a Functional Design is made. This Functional Design consists of two parts:

- System Control and Indication
- Software High-Level Flowchart

The software diagram includes several concepts that can be found in the technical analysis section for more details. The main algorithm consists of 1 main loop continually being updated with new data. The software is designed to continuously process the data received by the sensors and react accordingly. Our main algorithm to follow the path and keep the white line within the two sensors, detect obstacles and detect the ramp. Additionally, lower in the flowchart, we can see the algorithm to detect the end of the track and stop.

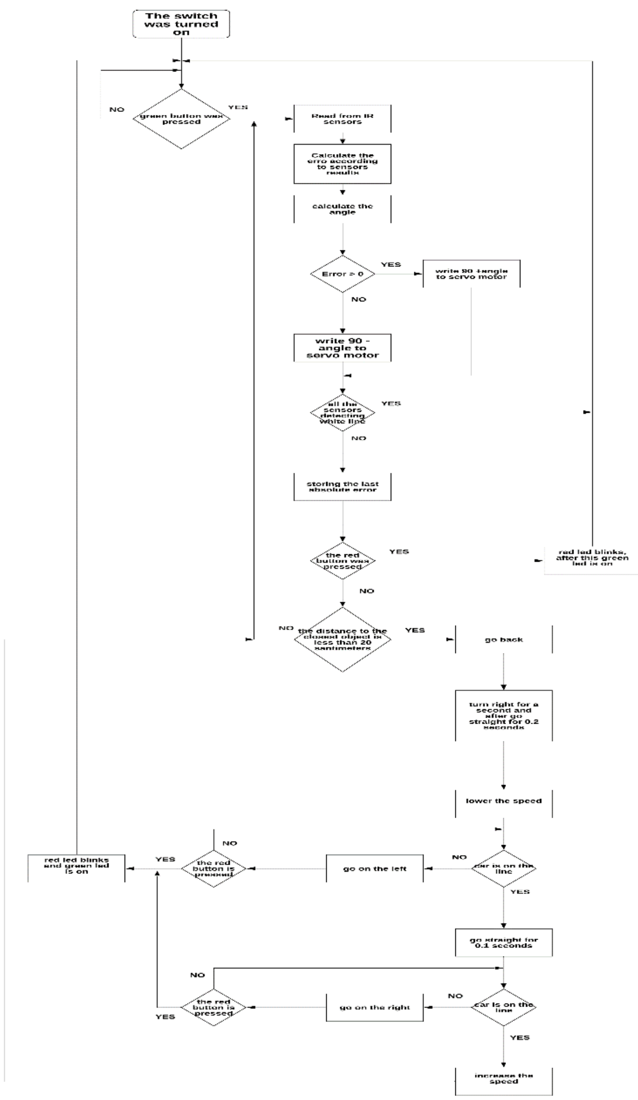


Figure 5. High-Level Flowchart

See Appendix D for the flowchart.

10.4 Testing Procedure

In this phase we are going to test each component independently. Maybe we should calculate the power each component uses for 30 min usage.

Test1: QTR-8A sensor array

QTR-8A sensor array is an IR-sensor module that the racing car will use to detect the white line. This is a key functionality of car and it is important to first make sure the module is working properly by its own.

In order to test the working state of IR module, the hardware engineer used the setup you can see below.

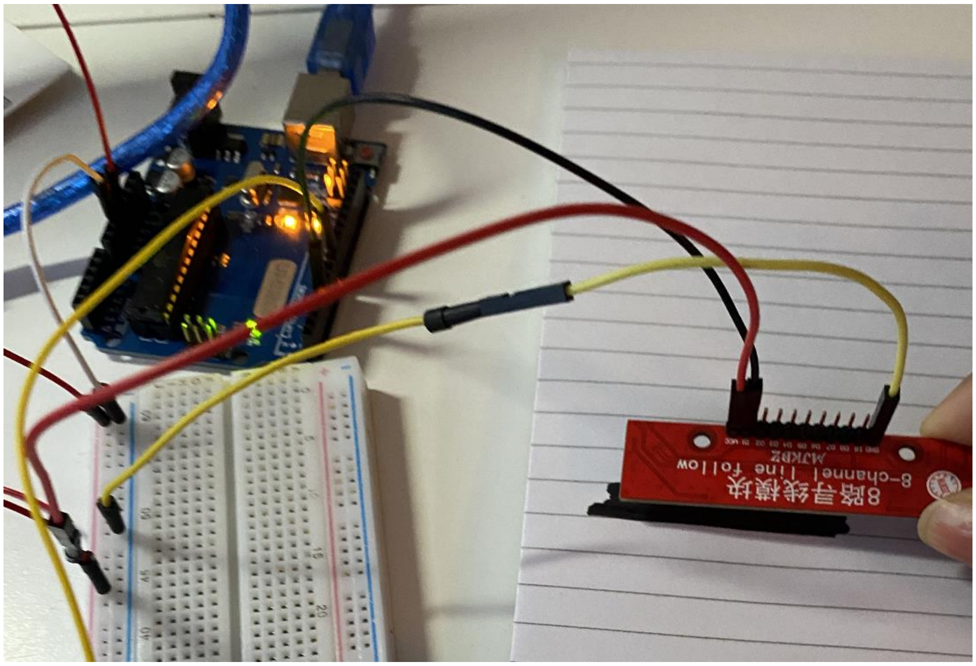


Figure 16. QTR-8A testing setup

The Testing Code

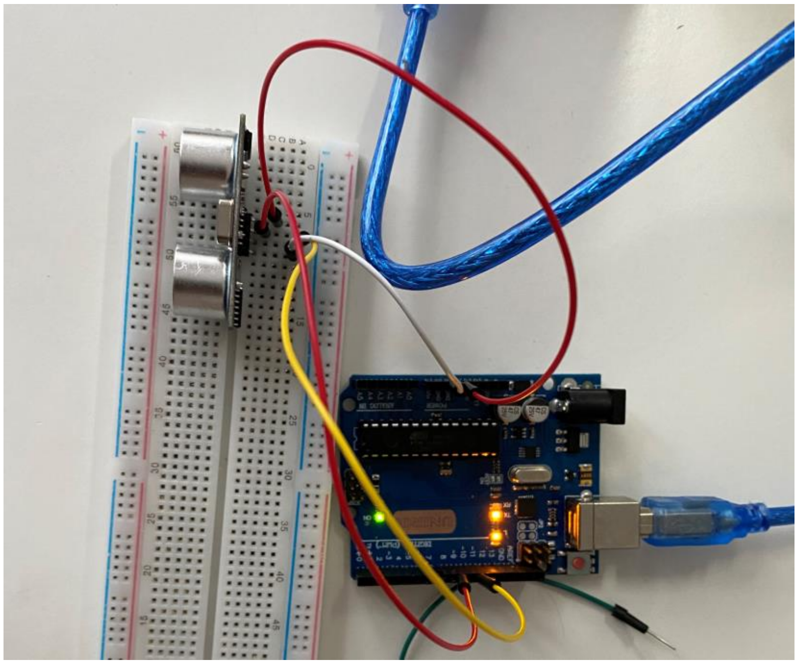
```
int IRSensor = 4; // connect ir sensor to arduino pin
4
void setup()
{
  Serial.begin (9600);
  pinMode (IRSensor, INPUT); // sensor pin INPUT
}
void loop()
{
  int statusSensor = digitalRead (IRSensor);
  Serial.println(statusSensor);
}
```

The result of testing the module with the code above shows that the QTR-8A module in use is working properly and according to the engineers' expectation. As a result, the engineers proceeded with the next step of testing phase.

Test2: Ultrasonic Sound Sensor

The ultrasonic sound sensor will be used in the racing car system to detect any obstacles that is placed on the track. This control data then will be used to avoid the said obstacle. To ensure this functionality, the engineers first make sure that the module alone is working fine.

The module was set up according to the picture below:



The code:

```
const int trigPin = 9;
const int echoPin = 11;
long duration;
int distance;
void setup() {
  pinMode (trigPin, OUTPUT);
  pinMode (echoPin, INPUT);
  Serial.begin(9600);
}
void loop() {
  // put your main code here, to run repeatedly:
```

```
 digitalWrite(trigPin,LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin,HIGH);
  delayMicroseconds(11);
  digitalWrite(trigPin,LOW);
  duration = pulseIn (echoPin, HIGH);
  distance = duration * 0.034/2;
  Serial.print ("Distance: ");
  Serial.println (distance);
}
```

Test3: Servo Motor

Servo Motor consists of a DC Motor, a Gear system, a position sensor, and a control circuit. The DC motors get powered from a battery and run at high speed and low torque. The Gear and shaft assembly connected to the DC motors lower this speed into sufficient speed and higher torque. The position sensor senses the position of the shaft from its definite position and feeds the information to the control circuit. The control circuit accordingly decodes the signals from the position sensor and compares the actual position of the motors with the desired position and accordingly controls the direction of rotation of the DC motor to get the required position. Servo Motor generally requires a DC supply of 4.8V to 6 V.

Controlling a Servo Motor

A servo motor is controlled by controlling its position using Pulse Width Modulation Technique. The width of the pulse applied to the motor is varied and send for a fixed amount of time.

The pulse width determines the angular position of the servo motor. For example, a pulse width of 1 ms causes an angular position of 0 degrees, whereas a pulse width of 2 ms causes an angular width of 180 degrees.

Application Circuit of Servo Motor

From the below application circuit: Each motor has three inputs: VCC, ground, and a periodic square wave signal. The pulse width of the square wave determines the speed and direction of the servo motors. In our case, we just need to change the direction to allow the device to move forward, backward, and turn left and right. If the pulse width is under a certain time frame, the motor will drive in a clockwise direction. If the pulse width exceeds that time frame, the motor will drive in a counterclockwise direction. The middle time frame can be adjusted through a built-in potentiometer inside the motor.

Test4: Gyroscope

Wiring:

5v -> 5v

GND -> GND

SDA -> (any SDA pin of the arduino look up on internet) 20(mega)

SCL-> 21(mega)

The Code:

```
#include<Wire.h>
const int MPU=0x68;
int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;

void setup(){
  Wire.begin();
  Wire.beginTransmission(MPU);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
  Serial.begin(9600);
}
void loop(){
  Wire.beginTransmission(MPU);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU,12,true);
  AcX=Wire.read()<<8|Wire.read();
  AcY=Wire.read()<<8|Wire.read();
  AcZ=Wire.read()<<8|Wire.read();
  GyX=Wire.read()<<8|Wire.read();
  GyY=Wire.read()<<8|Wire.read();
  GyZ=Wire.read()<<8|Wire.read();

  Serial.print("Accelerometer: ");
  Serial.print("X = "); Serial.print(AcX);
  Serial.print(" | Y = "); Serial.print(AcY);
  Serial.print(" | Z = "); Serial.println(AcZ);

  Serial.print("Gyroscope: ");
  Serial.print("X = "); Serial.print(GyX);
  Serial.print(" | Y = "); Serial.print(GyY);
  Serial.print(" | Z = "); Serial.println(GyZ);
  Serial.println(" ");
  delay(333);
}
```

Test5: MONSTER MOTOR SHIELD

The VNH3SP30 Dual Monster Motor Driver Shield is a 2-channel DC motor driver shield that can drive 2 motors at up to 9A sustained.

KEY FEATURES OF VNH3SP30 DUAL MONSTER MOTOR DRIVER SHIELD:

- Drive up to 2 DC motor at 5.5 – 16V
- 30A peak current, 6-9A sustained per channel.
- Two full H-Bridges with speed control via PWM and direction control
- Reverse power protection
- Diagnostic output to detect thermal shutdown and similar faults.

Motor Power Connections

Motor voltage must be between 5.5 – 16V. The manufacturer rates the chip for up to 36V operation but it should be limited to 16V to avoid possible issues that sometimes arise with higher voltages.

The module has reverse power protection via the use of N-Channel MOSFETS on the low side of the drivers. 5V which comes from the Arduino is only used for logic pull-ups on the board for the ENABLE pins. There are two unpopulated thru-hole capacitor locations on the board. These are in parallel with the two 470uF SMD caps that go across the motor

Vcc and Gnd connections. In some high current applications, it may be beneficial to add additional capacitors of 470uF or larger to these locations, but they are typically not needed.

1 x 2 Terminal (Motor Power)

- '+' = Motor Vcc which must be between 5.5 and 16V.
- '-' = Motor Ground.

Motor Connections

The motor connections are via two screw terminals for each motor. Multiple motors can be driven off each connection as long as the total current stays within bounds.

The wiring of which lead of the motor connects to which terminal is somewhat arbitrary and relative to what you consider forward vs reverse motor operation. If the motor goes in the opposite direction that you expect, simply reverse the wiring.

Terminal (Motor 1)

- A1 – Motor 1 '-' positive lead
- B1 – Motor 1 '+' negative lead

Motor Control Pins

The pins A1/A2 and B1/B2 correspond to the INA and INB pins on the driver IC which control the state of the H-Bridge in the device. The basic modes are to rotate CW, rotate CCW or brake.

Arduino to Shield Control Pin Connections

- D4 = A2
- D5 = PWM Motor 1
- D6 = PWM Motor 2
- D7 = A1
- D8 = B1
- D9 = B2

These connections are automatically made when the board is used in a shield configuration. If wiring discretely, these pins can be reassigned as needed.

EN/DIAG Pins

These pins serve a dual purpose and are bi-directional. First they are the Enable pins for the device and are active HIGH. The module has pull-ups on these pins, so if left unconnected or set as inputs, the drivers will always be enabled. If the pins are being used to enable/disable the drivers, they need to be driven HIGH to enable the devices.

The second purpose they serve is to indicate a fault such as a thermal shutdown. If that should occur, the pins are driven LOW by the module. See the truth table in the datasheet below on page 15 for operation of the DIAG function.

Arduino to Shield EN/DIAG Pin Connections

- A0 = EN/DIAG 1
- A1 = EN/DIAG 2

Eq1 = Special designed board with calculated distance points.

Eq2 = Special designed board with calculated angle points.

Test 6: The DC Motor

The motor features a permanent horseshoe magnet (called the stator because it's fixed in place) and an turning coil of wire called an armature (or rotor, because it rotates). The armature, carrying current provided by the battery, is an electromagnet, because a current-carrying wire generates a magnetic field; invisible magnetic field lines are circulating all around the wire of the armature.

The key to producing motion is positioning the electromagnet within the magnetic field of the permanent magnet (its field runs from its north to south poles). The armature experiences a force described by the left hand rule. This interplay of magnetic fields and moving charged particles (the electrons in the current) results in the torque (depicted by the green arrows) that makes the armature spin. Use the Flip Battery button to see what happens when the flow of current is reversed. Take advantage of the Applet Speed slider and Pause button to visualize these forces better.

A single, 180-degree turn is all you would get out of this motor if it weren't for the split-ring commutator — the circular metal device split into halves (shown here in red and blue) that connects the armature to the circuit. Electricity flows from the positive terminal of the battery through the circuit, passes through a copper brush to the commutator, then to the armature. But this flow is reversed midway through every full rotation, thanks to the two gaps in the commutator. This is a clever trick: For the first half of every rotation, current flows into the armature via the blue portion of the commutator, causing current to flow in a specific direction (indicated by the black arrows). For the second half of the rotation, though, electricity enters through the red half of the commutator, causing current to flow into and through the armature in the opposite direction. This constant reversal essentially turns the battery's DC power supply into alternating current, allowing the armature to experience torque in the right direction at the right time to keep it spinning

11. Technical Design

11.1. Introduction

In this phase of the project, the team will work more closely on developing the race car's detailed schematics and low-level flowchart. The hardware and software engineers will use the final design and result of this phase to implement the mechanical and electronics part and develop the software program. The team will design mechanical parts for the Velos car shell, electrical and electronic schematics, and a low-level flowchart for developments of software. They will also provide detailed specifics of the components that will be used in this project alongside with the needed calculations.

11.2. Mechanical Design

For the mechanical design, the motion system is already included in the chassis of the car, therefore we do not need to apply any changes to the steering system which is controlled by the servo motor and the dc motor at the back.

As for the body of the car, it is not provided and the company needs to design it for the client.

The team of engineers came up with a design using the front and back bolt attach, as well as the ledges for extra support. The body is devided in to two main parts since most of the 3D Printers do not have the volume to print a 40 cm long design.

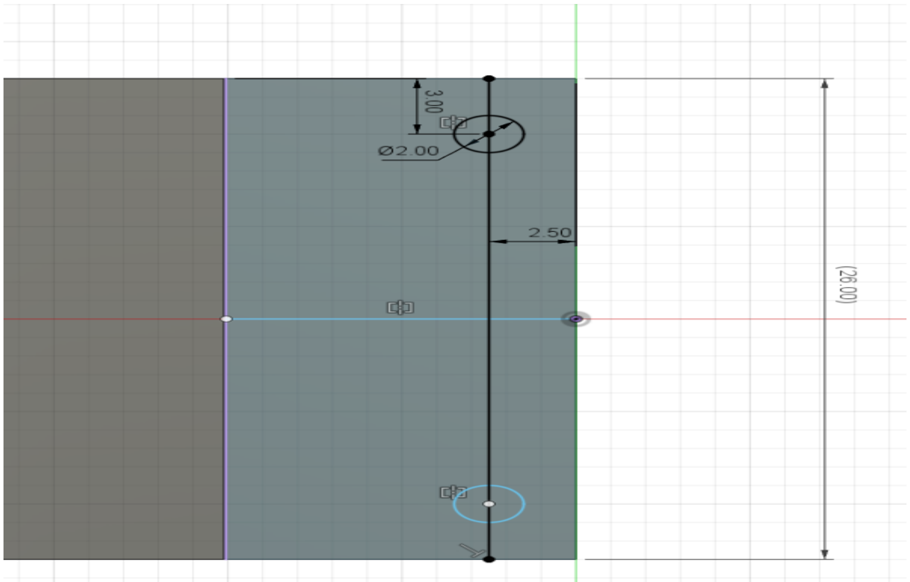


Figure 17. Front of the car 2D sketch

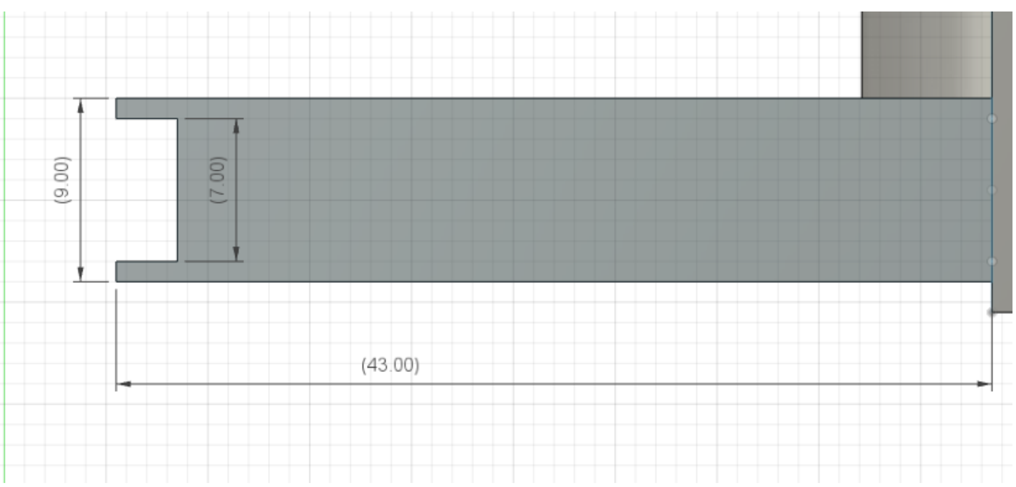


Figure 18 Support front piece ledges 2D sketch

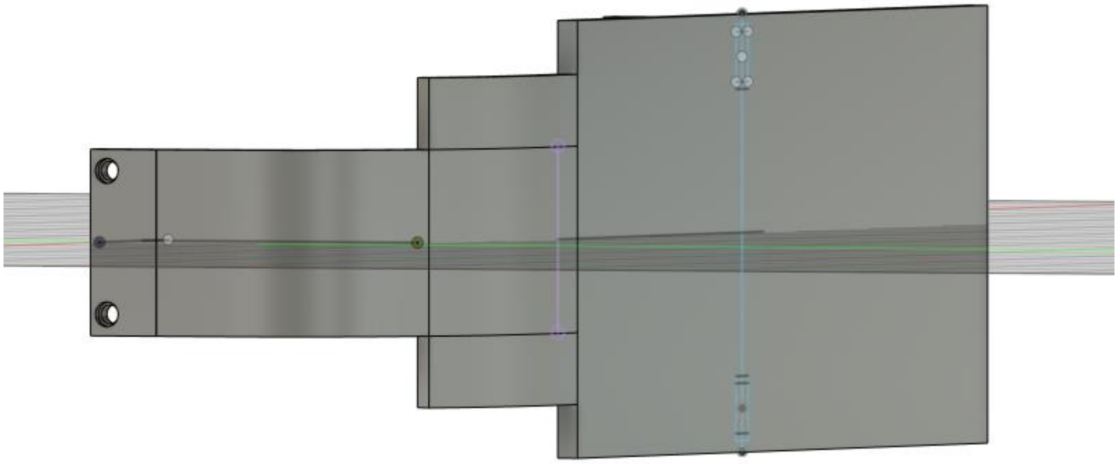


Figure 19 Front top 3D View

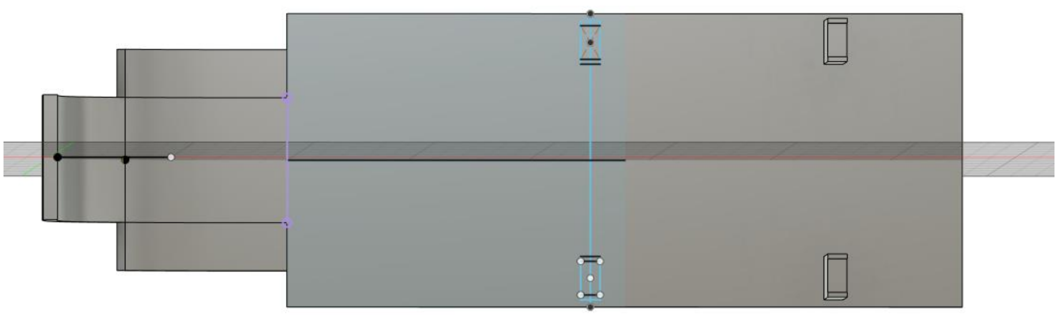


Figure 20 Front 3D Bottom view

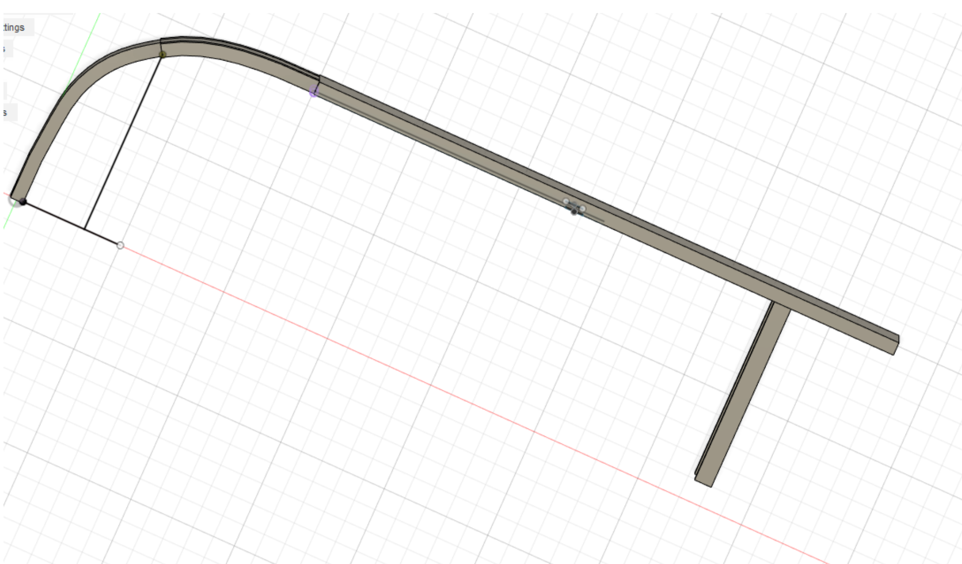


Figure 21 Front side 3D View

This is the prototype for the front part of the shell.
As for the bottom, the same concepts are introduced. The piece is using the holes already on the car to attach at the back, as well as support structures that fit on the ledges.

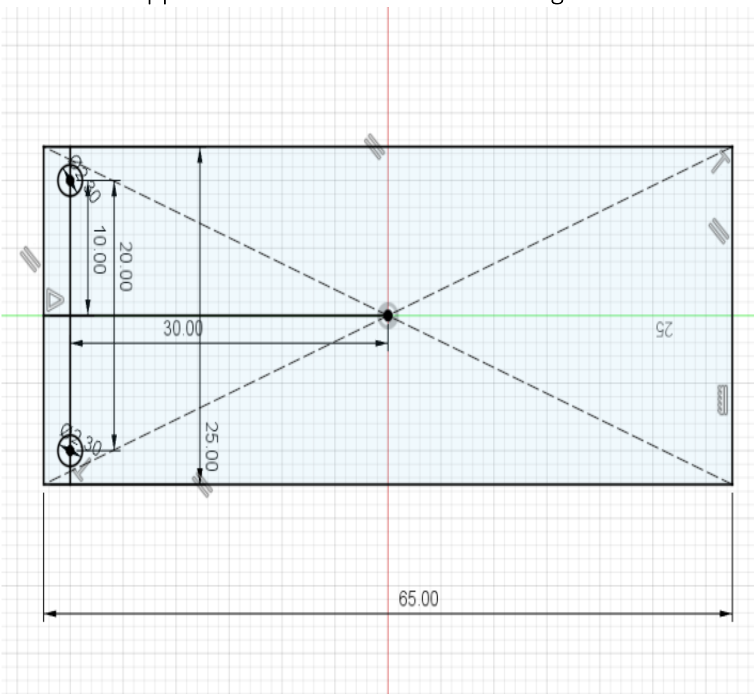


Figure 22 Back attach 2D sketch

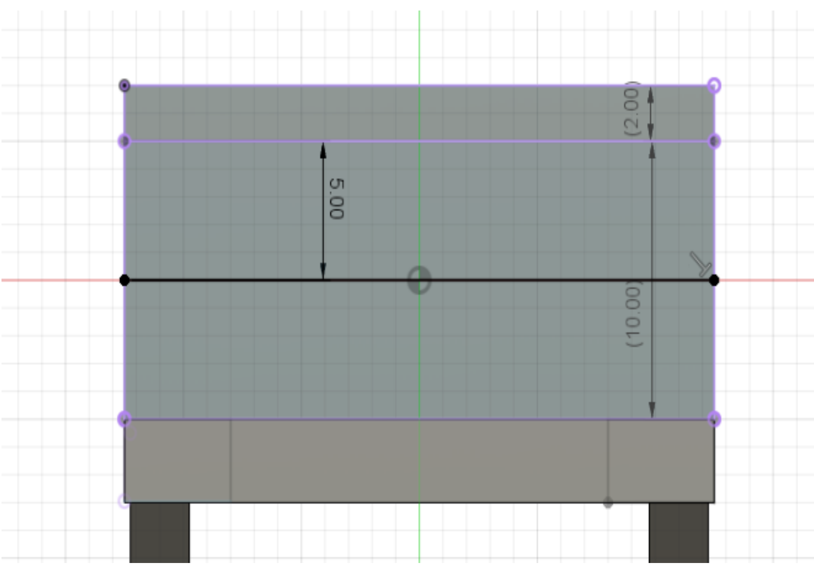


Figure 23 Back attach side 3D view

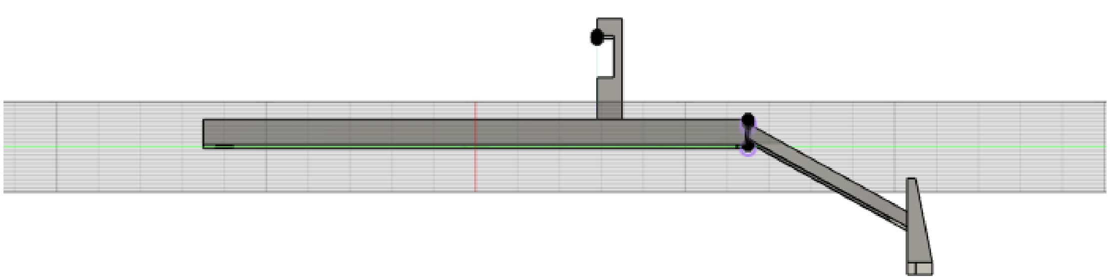


Figure 24 Back attach side 3D view

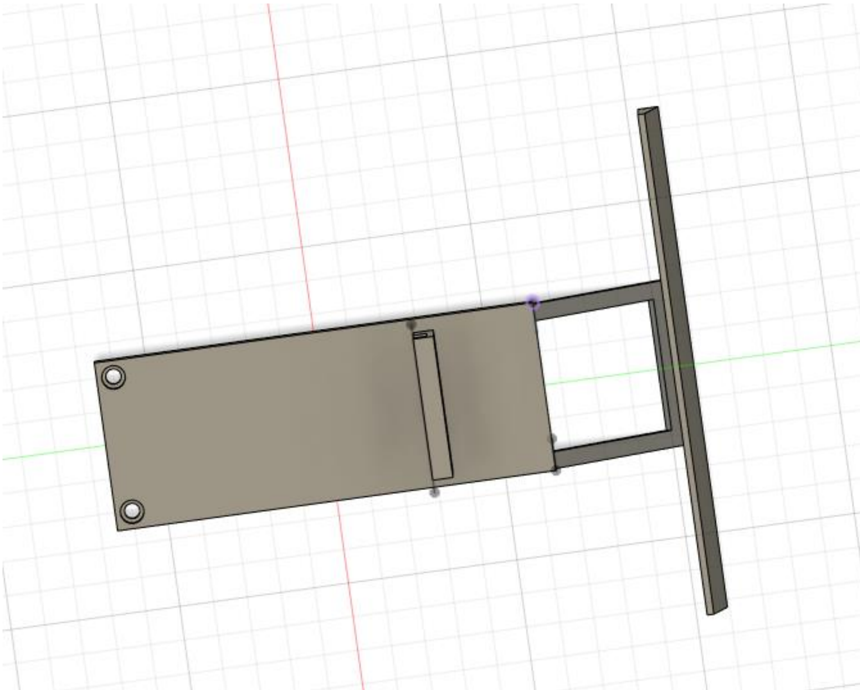


Figure 25 Back attach top 3D view

11.3. Electrical Design

The schematic in this section displays the electrical connections of Velos's autonomous car components.

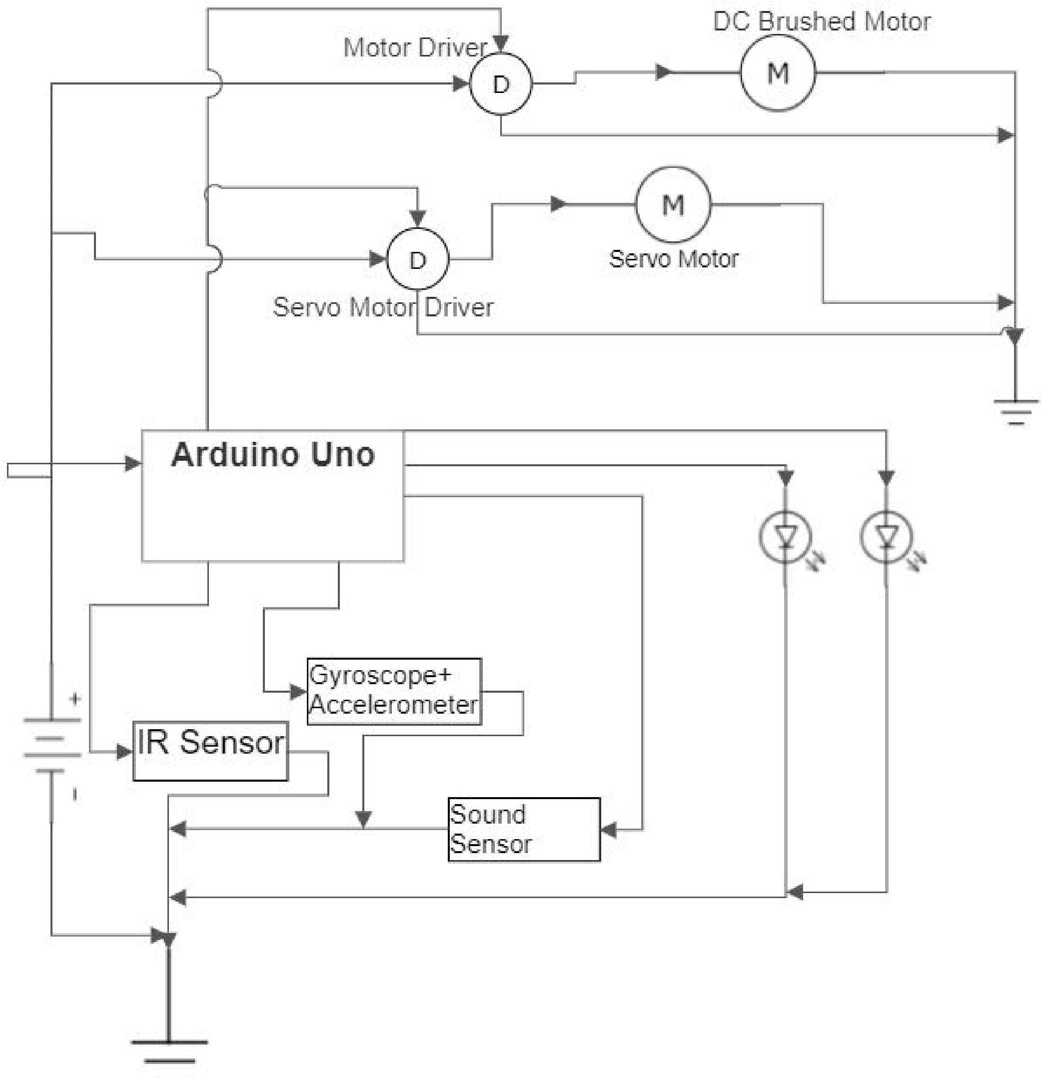


Figure 26 Electrical Design

Component List

In this table a list of all the electronic components that will be ordered and used in Velos autonomous car are listed.

Velos Autonomous Car Components	
1	Ishima RC 390 DC Motor
2	VNH3SP30 Monster Motor Shield
3	TS90M Mini Servo
4	QTR-8A Sensor Array
5	HC-SR04 Ultrasonic Distance Sensor
6	Arduino Uno
7	DEBO SENS 3AXISH Acceleration and Gyroscope
8	VT-2005 LED strip, SMD 3528 60 cool white IP20

11.4. Electronics

In this section an in-depth schematic and technical properties of chosen components are provided in order to provide a deep understanding of how the Velos autonomous car functions. A detailed overview and technical explanation of each module is presented to show the functionality of them.

11.4.1 Brushed DC Motor

Permanent magnet brushed DC motors are the most common. These motors use permanent magnets to produce the stator field. They are generally used in applications needing fractional horsepower. The choice of these component was already made for us by the stakeholders.

Component choice: Ishima RC 390 DC Motor is going to be used for this project.

Figure 27 Brushed DC Motor

Brief Data:

1. Rated voltage: 7.2V

   Operating voltage range: 4.5V-9.6V

2. Maximum efficiency:

   EFF: 60.76%

   Speed: 18000rpm

   Current: 6.33Amp

   Torque: 150g.cm

   Output: 27.692W

   Diameter: 27.5mm

   Length: 46.5mm

   Shaft length: 11mm

   Shaft diameter: 2.3mm

   Weight: 113g/ 3.98oz

11.4.2 Motor Shield

Velocity as a vector has to factors first is the actual speed and secondly the direction of the speed. In order to control the speed of the motor we are going to use a Pulse width modulation (PWM). Now in order to control the direction of the motor clockwise or counter clock wise we are going to use H-Bridge module.

The H-Bridge is a circuit which can drive a DC motor in forward and reverse. The motor direction is changed by switching the polarity of the voltage in order to turn the motor one way or the

Usually it is easier to find a DC motor control module which is a combination of the above. In this project we are going to use VNH3SP30 Monster Motor Shield.

Figure 28 Motor Shield

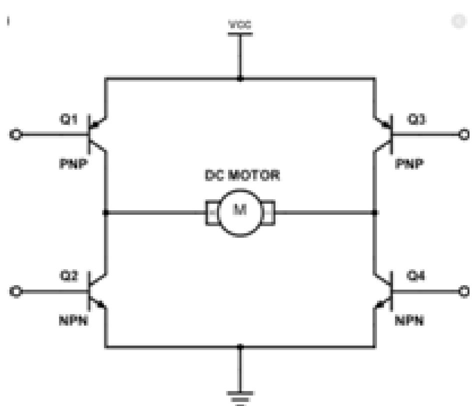


Figure 29 H-Bridge module

Brief Data:

1. maximal Voltage: 16 V

2. maximal Current: 30 A

3. MOSFET max. Resistor: 19 mO

4. max. PWM-Frequency: 20 kHz

5. with Overheat-, Overvoltage- and Undervoltage Protection

6. Size: LWH ca. 60 mm x 53 mm x 12 mm

7. Weight: ca. 17 g

11.4.3 Servo Motor

Velos autonomous car uses a servo motor for the Steering system because it makes it possible to maneuver the tie rod end over 180 degrees and servo motor is designed for this job.

Component choice:TS90M Mini Servo

Figure 30 Servo Motor

Brief Data:

1. Voltage range:4.8-6.0V DC

2. Dimensions(LxWxH):32.4x12.5x30.9mm

3. No-load speed: 0.12 s/60° (4.8V); 0.10 s/60° (6.0V)

4. Torque :1.6kg cm (4.8V)

5. Swivel range: 180°

6. Gear materials: Metal

11.4.4 QTR-8A Sensor Array

In order to detect the white line, we are going to use QTR-ThA Sensor Array. The main principle is that they emit an electromagnetic field and try to detect for changes in the received signal .IN this project the difference in the received values is due to the fact that electromagnetic radiation deflects differently in a white and black surface (Darker objects tend to absorb more radiation than white surfaces) For this project we are going to use 8x IR Lijn Tracking Module. The

QTR sensor is intended to be used as line sensors in line-following, maze-solving, and sumo robots, however, they can be used for as proximity or reflectance sensors. QTR-8A is a reflective sensor type A, which means its output is an analog voltage signal.

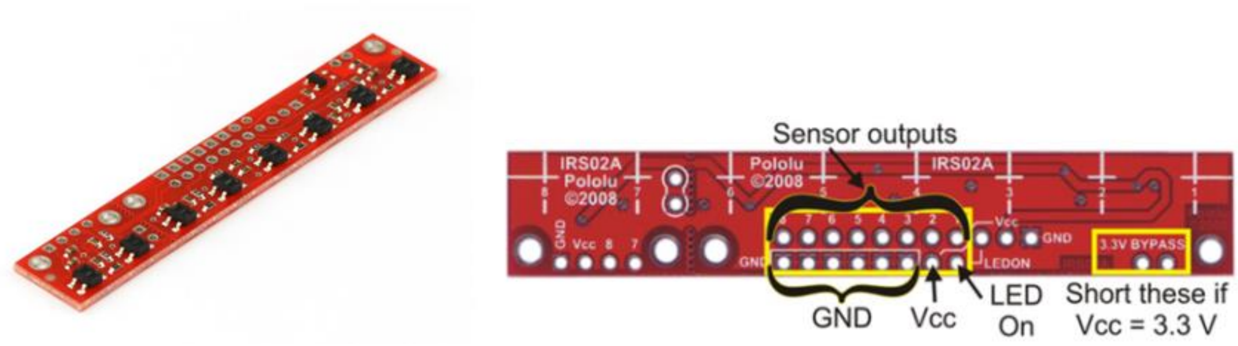


Figure 31 QTR-8A Sensor Array

Brief Data:

1. Dimensions: 2.95" x 0.5"
2. Operating voltage: 3.3-5.0 V
3. Supply current: 100 mA
4. Output format for the QTR-8A: 8 analog voltages ranging from 0 V to supplied voltage
5. Optimal sensing distance: 0.125" (3 mm)
6. Maximum recommended sensing distance for the QTR-8A: 0.25" (6 mm)
7. Weight without header pins: 0.11 oz (3.1 g)

11.4.5 HC-SR04 Ultrasonic Distance Sensor

In order to detect obstacles, this project is going to use Sound sensors which is known as Radar. By sending ultrasound waves that will rebound on the object and come back to the sensor the car will be able to calculate the distance therefore calculate the maneuver to avoid it. The component choice for this car is HC-SR04 Ultrasonic Distance Sensor, The ultrasonic module works at a distance of 3 to 400 cm. The maximum deviation in this regard is only 0.3 millimeters.

Figure 32 Ultrasonic Distance Sensor

Brief Data:

1. Range:3 tp 400 cm
2. Operating voltage 5V DC
3. Frequency: 40 kHz
4. TriggerPulse width: 10 µs

11.4.6 Microprocessor Arduino Uno

For the microprocessor this project is going to use Arduino Uno as it has 5 voltage output which is able to activate all the components on the system. Also it has a 10bit resolution which is high enough for the specific project. Resolution is important for all the subsystems because they are going to communicate information from analog to digital and vise versa (1023 resolution // vin=5/1024= 4.9mV) per unit.

Finally, in terms of computing and storage Arduino is more than enough to guarantee that the project will function smoothly.

Figure 33 Arduino Uno

Brief Data

1. 14 digital I / O interfaces (6 of which can be used as PWM output)
2. 6 analog inputs
3. 16 MHz - Quarzoscillator
4. USB connection
5. Power connection
6. ICSP header
7. Reset
8. ATMega 328
9. Operating voltage: 5V
10. Input voltage: 7 - 12V
11. Input voltage (limit): 6 - 20V
12. Digital inputs / outputs: 14
13. Analogue inputs: 6
14. I DC 40mA (I / O) / 50mA (3.3V)
15. Flash Memory: 32 kByte
16. SRAM: 2 kByte
17. EEPROM: 1 kByte
18. Frequency: 16 MHz

11.4.7 Gyroscope and Accelerometer

This module has more than one uses. Firstly it is used to gain some form of control over the speed of the car while it is on the ramp (Based on the fact that the ramp is only 15 degrees the car will be able to climb on the track) And secondly gyroscope has a 3plie axis accelerometer so we have a real-time data about the acceleration of the car combined with some basic physics this variable will help with the calculation of the maneuver as well.

The Component choice for this car is DEBO SENS 3AXISH Acceleration and Gyroscope.

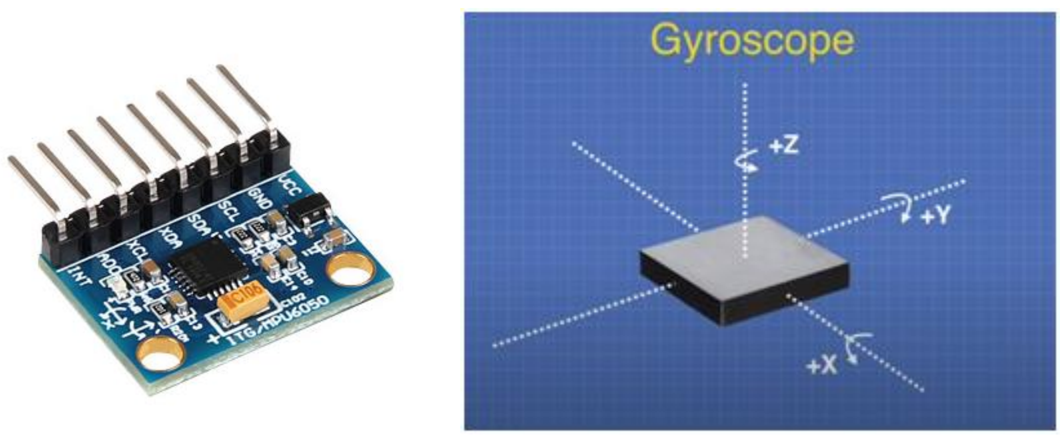


Figure 34 Gyroscope and Accelerometer

Brief Data:

1. Module chip: MPU6050
2. Axes: 3 acceleration and 3 position axes
3. Degrees of freedom: 6 DOF simultaneously
4. Power supply: 3.3 ... 5 V DC
5. Voltage regulation: Low Dropout Regulator
6. Interface: I^2C
7. Built-in chip: 16 bits AD / DA converter
8. Dimensions: 25 x 20 x 7 mm

More specifications

1. Pins: 8 pcs
2. Standard pin pitch: 2.54 mm
3. Accelerometer range: ± 2, ± 4, ± 8, ± 16 g
4. Sensitivity: 16384 LSB / g ± 2g, 8192 LSB / g ± 4g, 4096 LSB / g ± 4g, 2048LSB / g ± 16g
5. Gyroscope range: ± 250, ± 500, ± 1000, ± 2000 ° / s
6. Sensitivity: 131 LSB / dps ± 250 dps, 65.5 LBS / dps ± 500 dps, 32.8 LBS / dps ± 1000 dps, 16.4 LBS / dps ± 2000 dps

11.4.8 LED Strips

In order to work both day and night we need to consider the fact that while there is no light white and black surfaces emit the same kind of radiation so this will confuse the system. In order to avoid that we will use a Led Strip mounted on the bottom of the car. The light emitted by the strip will be enough to make the IR system functional. Component choice is VT-2005 LED strip, SMD 3528 60 cool white IP20.

Figure 35 LED Strips

Brief Data:

1. 12 V LED strip on flexible strap
2. flexible and easy installation
3. low energy consumption
4. Indoor and outdoor variants available
5. visual effects in endless colors and shades
6. Beam angle: 120 °
7. Length: 5 m
8. Width: 8 mm

11.4.9 User Interface PCB

This PCB will be designed in combination with the shield we will place all the components. It will basically be a control system of three buttons, two LED and one switch that will help unfamiliar users to use the project as well as the design team with the testing phase.

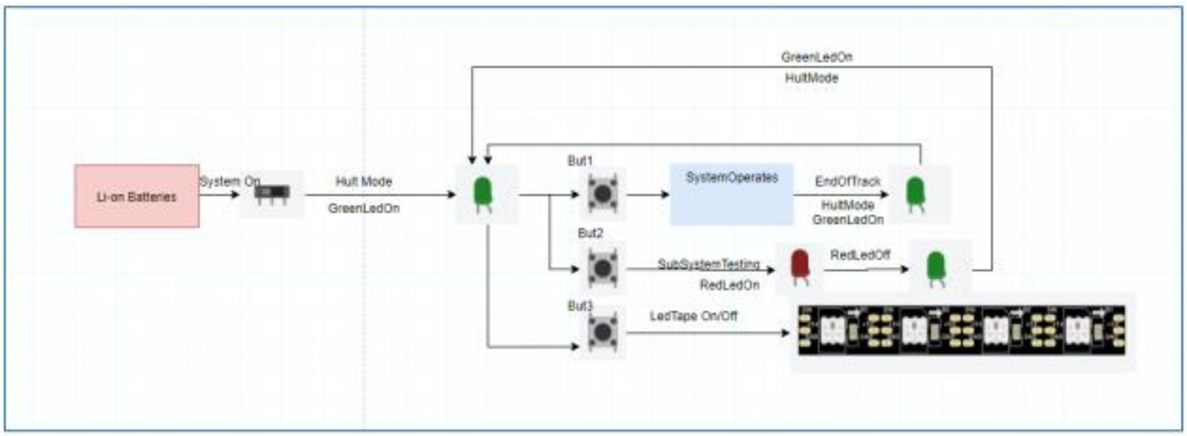


Figure 36 User Interface PCB

11.5. Software

In this part a detailed flowchart of the program for microcontroller, in this car Arduino Uno, is generated to help the software engineer develop the code in the next phase.

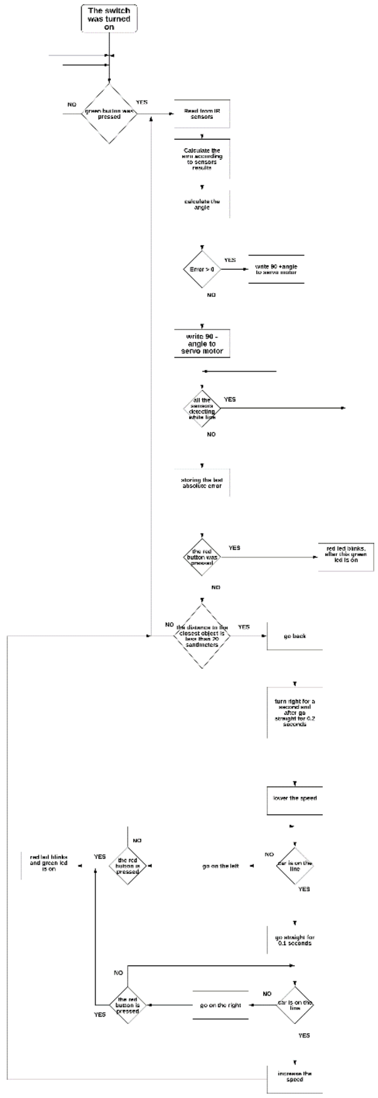


Figure 37 Detailed flowchart

See the flowchart at Appendix D.

12. Result

12.1 Introduction

The results are divided in every category of the final design and are all the product of the activities we defined in previous chapters (project plan, Technical & functional design).

12.2 Mechanical Design

The mechanical design resulted close to the prototype a lot of holder and modifications have been added.

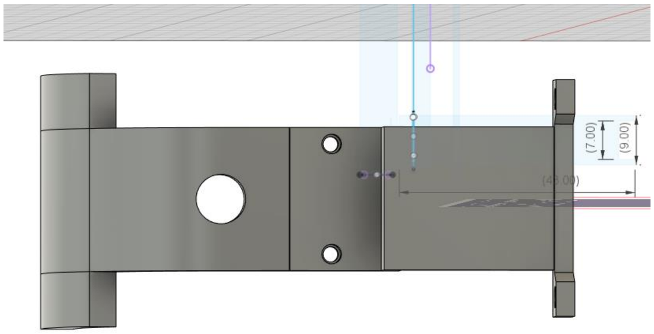


Figure 38. Top of car shield

The front of the car has been updated, There has been included a hole for cables coming from thee IR sensor holder on the bottom of the design .

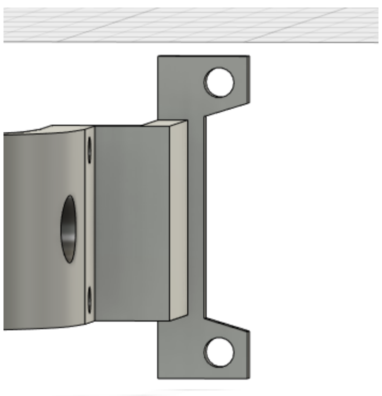


Figure 39. Front of car shield

A holder for the IR sensor . We have measured the optimal operating height for the sensor and tried to keep in account the height of the ramp.

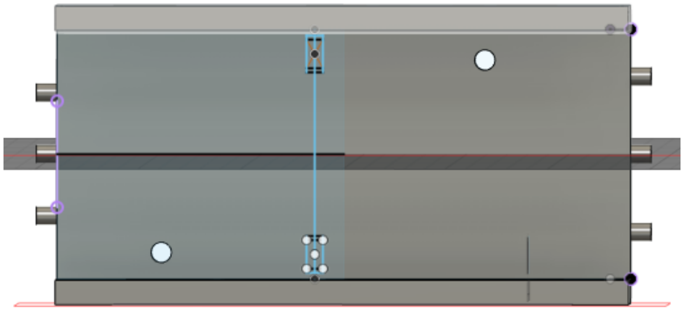


Figure 40. middle part of car shield

The middle part of the body has 2 holes in order to screw the arduino from the bottom and hide all the cables.

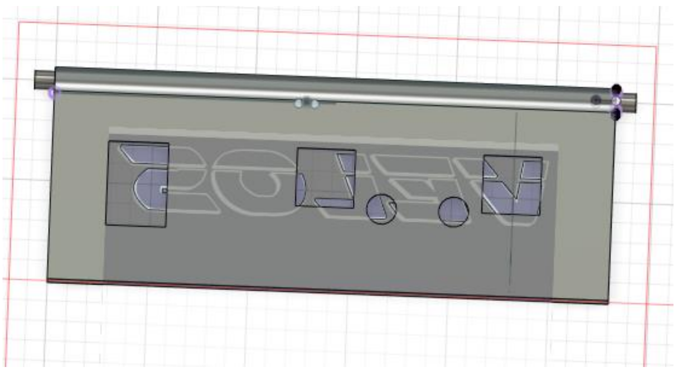


Figure 41. side of car shield

On the side we added holes for the logic circuit and the switch.

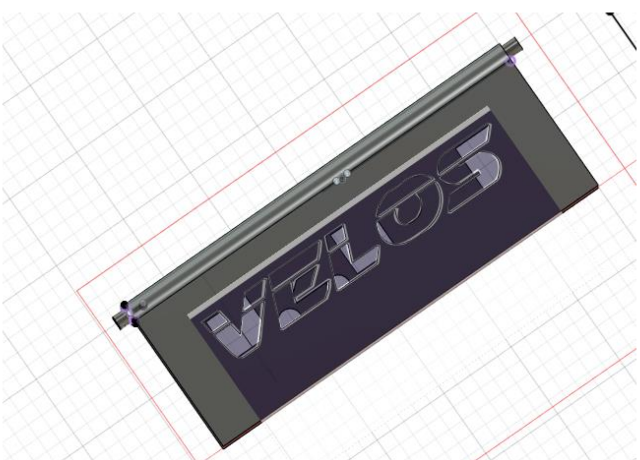
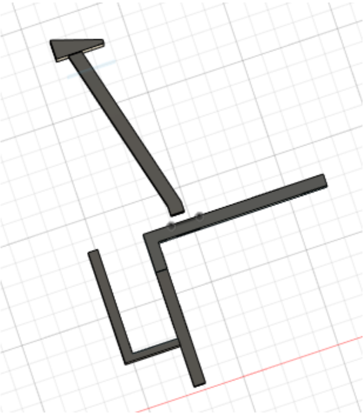


Figure 42. side of car shield

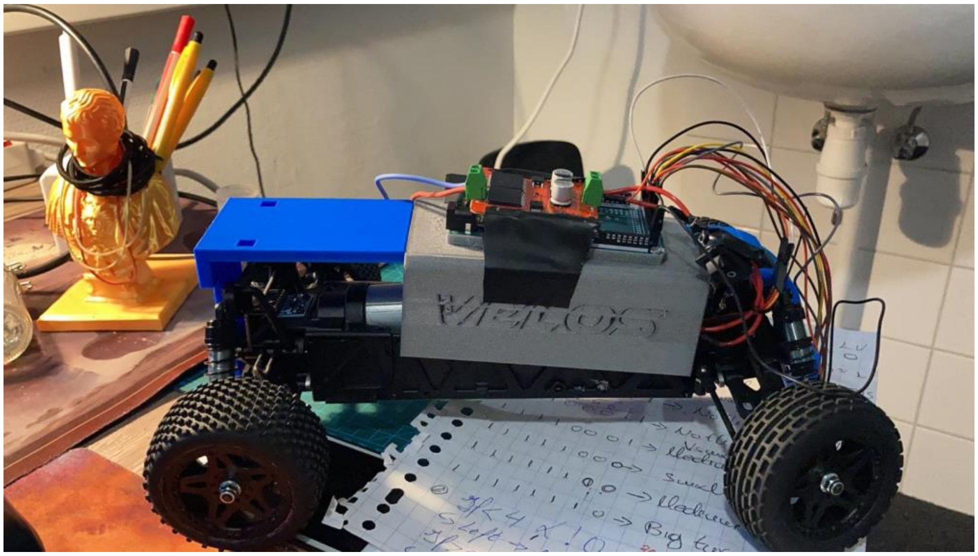
On the other side we display the team name as a hole in our design also to display the electronics inside.



The final back part has a holder for the power supply of the arduino.

12.3 Software

The software team first focused on the testing of the different components. Then using the report made a first testing session with every components required for the line following process.



Final driving algorithm description :

After the first testing run we designed the final algorithm.

The algorithm we chose for the line following car:

Our sensor array gives an 8 bit array of type xxxxxxxx where x oscillate in state between 0 and 1 depending on the color of the surface (0 white 1 Black). In order to take advantage of size of the line which is 3 bits, we decided to use the following formula

S = sum of indexes of Bits with values 0 (important, index starts from 1 -> 8)

Q = quantity of sensors.

It results in the value A = S/Q.

Ex: 11100011 would be read as (4+5+6)/3 = 5

the index for all positions oscillates from 1 to 8, with 0 being black and all sensors = 1 the finishing line.

note that we do 9 -A if the value is > 6 in order to map them from 1 to 4.

Therefore 1 means the line is far on the left and we need to make a bigger turn to correct it we can iterate the logic for other values 4&5 being for ideal line in the middle positions and mirrored from 6 to 8 (line from not far to very far on the right)

Using this average value we can come up with this simple formula that is enough to make the car follow the line

C = Constant angle

V = Constant default speed

v = Constant lower speed v < V

A = Average value

Angle:

f(A) = 90 +- C/A

Speed:

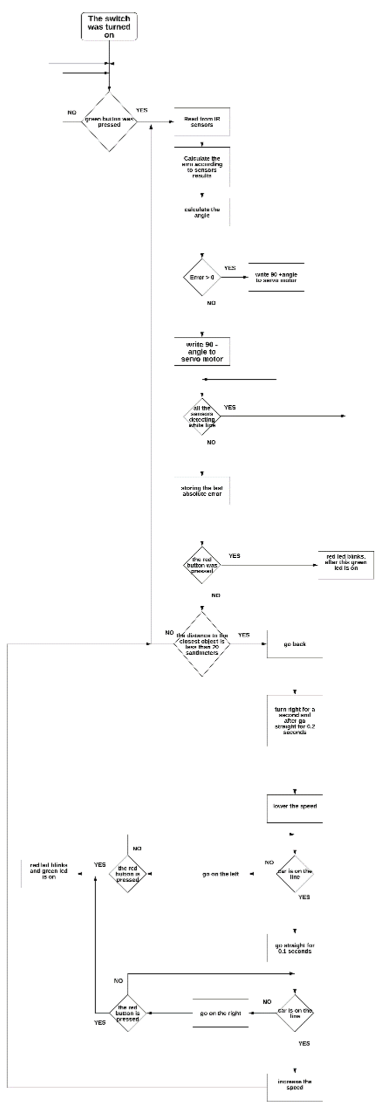
$g(A) = V - v / A$

As for these parameters since the further the line the lower the index, we can see that it will give optimal values of speed and angle if we calibrate it well, in the first function we will have a big turn if the angle is 1 or 2 ,

we can see that the speed accordingly decreases in order to ease the car’s turn.

This example resumes the algorithm’s logic, if we are further than the line we will take a bigger turn and decrease the speed in order to facilitate the rotation, if we are close to the line our speed will be less reduced and closer to the constant, as for the angle he will also tend to the constant.

High level flowchart:



Check the flowchart at the Appendix D.

12.4 Electrical design

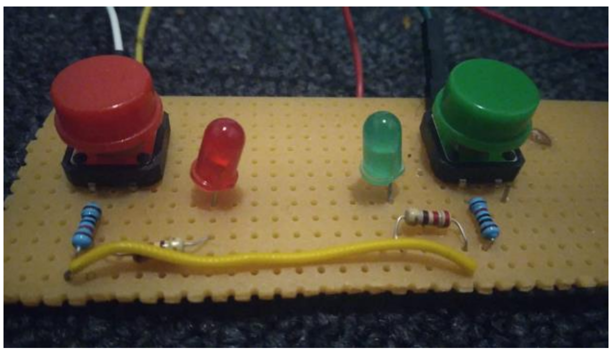
As for the electrical design, we had to create a holder for the batteries

The Mechanical design team 3d printed a holder in order to facilitate the recharge of the batteries, it fits in the case of the given car.

The 2x3.7 v feed the motors, Servo and DC through the motor driver.

We also use a 5v mobile power supply in order to power the logic of the atduino and the sensors. We went for this taking in account the low consumption of the logic units.

As for the rest and the pcb it is as described in the functional design.



12.5 Final Design

As for the final design we mounted every piece in it's designed and accessible place we placed the sensors in the holders the pcb in the side holes, the switch in the side hole of the main body .

12.6 Final Result

Besides the supposedly damaged motor shield we happened to have that cost us 1 weeks trying to make it work and deciding to get a new one, all the components have been tested separately then in unit flawlessly. As for the and the algorithm for the IR sensor it has been worked out and figured out and the testing was a success after the 2nd test with small update on the speed and angle variables.

Same went for the Mechanical design and electrical design.

13. Reflection on final design

The project was carried away not without flaws, the first unexpected one was that we had an issue with the motor drivers we needed to fix and then replace.

Also we needed to design a practical electrical installation so we could take out and recharge the batteries easily, but as well to have everything fitting on the car so we decided to design in 3D a battery holder, we used jumpers and soldered the cables to make everything secure and stable.

For the mechanical design previous versions were tested as the car was evolving so there were no issues on the side of the mechanical design since the measurements were already taken weeks ago, it was necessary only to test the placements for the components and cable management which was done without flaws.

The algorithms for the line recognition worked as intended from the second test . As for the obstacles we tested the recognition with the sound sensors and wrote an algorithm to avoid them. We had an issue with the ramp, since the sound sensor would detect it as an obstacle. We are still trying to solve this issue for the final race.

In the future it would be wise to react faster in the case of a defectuous component, always order backup components for the finalization of the project.

If we wouldn't have been able to find the same we would have had to change plans which would have slowed down the project completion even more.

We spent days on designing the holders for the electrical installation which was necessary to make the car mobile for software testing, in the future we should focus on this first if we have a mobile device, since it is a crucial step, without a mobile electrical installation we cannot get the project further in software testing.

Also during the holidays spread all over may/ april ,we lost rigorosity on the organisation and deadlines, which cost us in less time for coding which was in the end not a problem since the solution was figured out fast.

If we had to extract a lesson from the mistakes we made during this project would be to have extra backup components and plans, but also to analyze the project more deeply from the beginning and follow this fatal step order. Finally, be more rigorous on the project and maybe communicate even more often as a team especially during holidays so some members of the tem do not stay without any tasks to accomplish .

14. Conclusion

The product is overall satisfying, most of the initial components that were supposed to be added have been and the car is working according to the team plan, the 3D body fits perfectly, we also added some extra features such as glowing in the dark and a crash proof front .

Every member of the team participated in a consequent way for the documentation and the concrete work on the product, most of the team members were present at every meeting and also responded within the hour on the group chats. Finally, it never happened that one of the teammates did not manage with its task, this was very helpful since we did not lose any time and it was easy for the team leader to delegate.

Finally I would resume the team as motivated and competent, we did not run in any conflict and always found a solution that was satisfying the majority of the team. The mentality and motivation of the team is what made this project overall without a hitch .

Bibliography

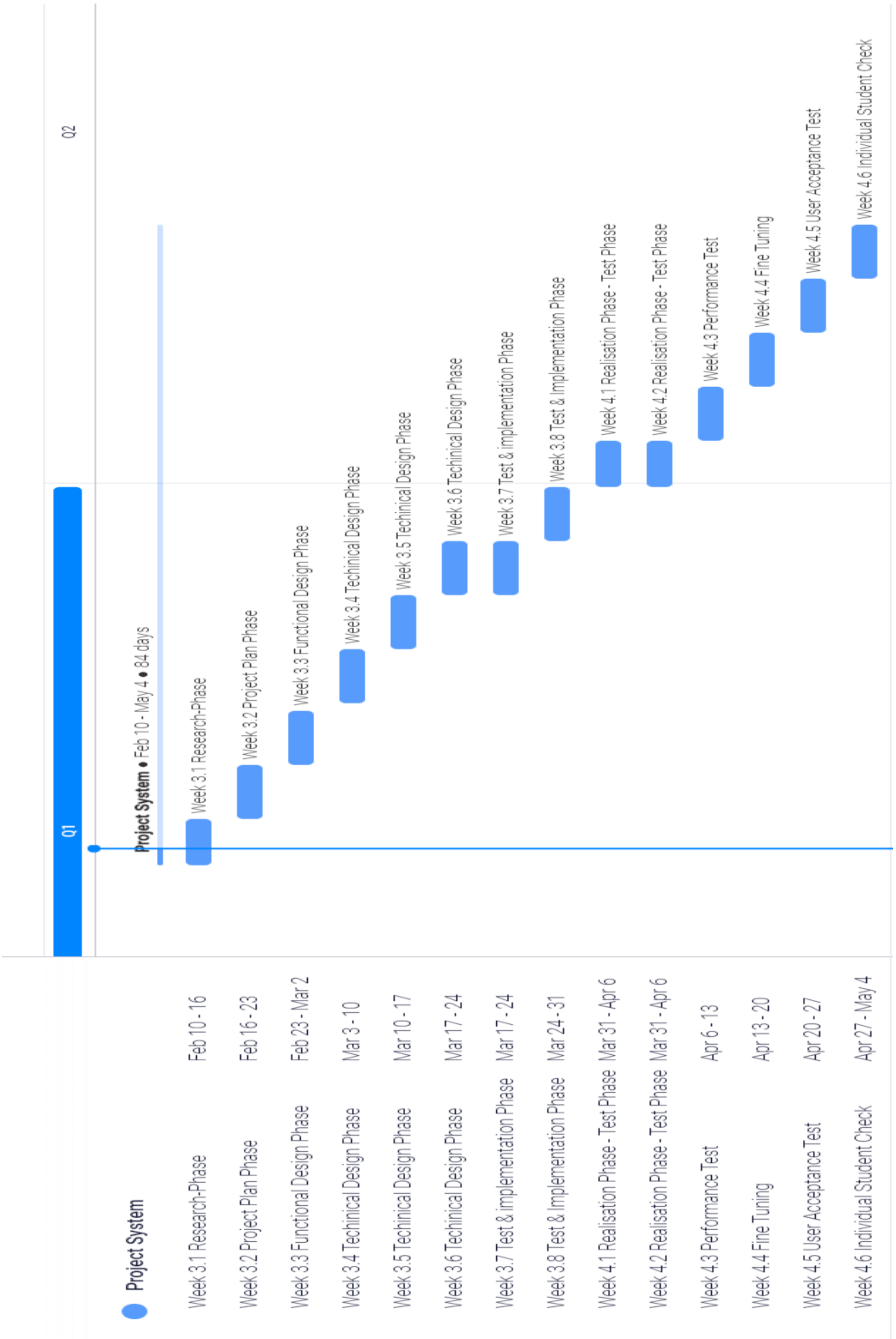
Grit, R. (2010). Project managment. Noordhoff Uitgevers.

Costs and benefits:

- *https://www.government.nl/topics/minimum-wage/amount-of-the-minimum-wage*
- *https://leren.saxion.nl/bbcswebdav/pid-3008862-dt-content-rid-103199221_4/institution/LED/LED_Opleiding_TI/Jaar_1/Project%20System/2021_Project%20System_Introduction.pdf*
- *HTTPS://WWW.ST.COM/RESOURCE/EN/DATASHEET/VNH3SP30-E.PDF*
- https://create.arduino.cc/projecthub/Nicholas_N/how-to-use-the-accelerometer-gyroscope-gy-521-6dfc19

Appendix A

Gantt chart



Project System

		Subite...	Timeline	Person	Status	De
Week 3.1 Research-Phase	1	7	Feb 10 - 16	S +3		
Week 3.2 Project Plan Phase		1	Feb 16 - 23		Working on it	
Week 3.3 Functional Design Phase		4	Feb 23 - Mar 2			
Week 3.4 Techinical Design Phase		10	Mar 3 - 10			
Week 3.5 Techinical Design Phase		10	Mar 10 - 17			
Week 3.6 Techinical Design Phase		8	Mar 17 - 24			
Week 3.7 Test & implementation Phase		5	Mar 17 - 24			
Week 3.8 Test & Implementation Phase		1	Mar 24 - 31			
Week 4.1 Realisation Phase - Test Phase		1	Mar 31 - Apr 6			
Week 4.2 Realisation Phase - Test Phase		1	Mar 31 - Apr 6			
Week 4.3 Performance Test		1	Apr 6 - 13			
Week 4.4 Fine Tuning			Apr 13 - 20			
Week 4.5 User Acceptance Test		1	Apr 20 - 27			
Week 4.6 Individual Student Check			Apr 27 - May 4			

Appendix B

Risk Analysis

		Risk	Value *	Factor **	Weight ***	Total risk
6	**Time factor**		↓choose one↓			
7	1	Estimated duration of project	6+ months	3	4	12
8	2	Does the project have a definite deadline?	Yes	2	4	8
9	3	Is there enough time to complete the project within the time permitted?	More than enough	0	4	0
10	**Complexity of project**		↓choose one↓			
11	4	Number of functional subsectors involved	3+	3	4	12
12	5	Number of functional subsectors that will make use of the results of the project	2-3	1	2	2
13	6	Is is a new project or one that will be adapted?	New project	3	5	15
14	7	To what extend do current authorizations in the organization have to be adjusted?	Medium extend	2	5	10
15	8	Are other projects dependent on this project?	No	0	5	0
16	9	How are users (of the project results) likely to respond to it?	Enthousiastic	0	5	0
17	10	the project depend on coordination between them?	Strongly	3	3	9
18	**The project group**					
19	11	Where do the project workers come from?	Mainly internally	0	4	0
20	12	Where is the project located?	1-3 locations	1	2	2
21	13	How many project members work for more than 80% at peak hours?	1-5	0	5	0
22	14	Balance between subject experts and project experts	Good	0	5	0
23	15	Are users involved in the project?	to a reasonable extend	1	3	3

Project management		↓choose one↓			
16	Does the project management team have any knowledge of the subject?	a reasonable amount	2	3	6
17	Does the project management have any knowledge of how to plan a project?	a reasonable amount	2	3	6
18	How much experience does the project manager have with a project like this?	a lot	0	3	0
19	Does the adviser have much knowledge of the field of the project?	a lot	0	5	0
20	Do the subject experts have much knowledge of the field?	a lot	0	5	0
21	How involved are responsible managers in the project?	very	0	5	0
22	Is there any chance that the project team will change during the project?	little chance	0	5	0
23	Is the project team using existing methods or are they creating their own tools?	some existing methods	2	4	8

Risk analysis continued ….

	Risk	Value *	Factor **	Weight **	Total risk
Project definition		↓maak keuze↓			
24	Are project members sufficiently aware of problems and objectives?	yes, everybody	0	5	0
25	Is the field of result (scope) sufficiently defined?	yes	0	5	0
26	Is there enough distinction between this project and other projects?	reasonable	1	4	4
27	Has enough time been resereved for coordination and decision-making?	reasonable	1	4	4
28	Are the boundaries and preconditions clear?	yes	0	4	0
29	Are the boundaries limiting enough?	moderately	2	5	10
			Total		**111**
			Risk percentage ***		**25.64%**

Appendix C

LogBook

Logbook - project Racing Car									Name: Velos Group
Company: Velos Group									
The locations where specific activities took place are lettercoded.									
The locations and corresponding code can be found in the legenda									
	Week	3.1							total time
Work done	Date	Mon	Tue	Wed	Th	Fri	Sun	Sat	x
Lecture		A							1.5
Problem definition		B		A					2
Project plan		A		B					13
Quality									
Risk assesment		A							2
General planning		A		B					1.5
Personal planning		A							
Meeting		A		B					2.5
Prepare meeting		A							0.5
							Subtotal		23.00

Legenda	
Code	Location
A	Personal residence
B	W2.40 -Saxion
C	W2.29a - Saxion
D	W2.30a - Saxion
E	W2.30b - Saxion
F	OTSWO - Saxion
G	W3.47 - Saxion
H	W2.42 - Saxion
I	W2.43 - Saxion
J	FabLab
# hr	Hours spend

	Week	3.2							total time
Work done	Date	Mon	Tue	Wed	Th	Fri	Sun	Sat	x
Meeting				B			A		5
Project plan		A							6
Risk assesment									
quality									
Lecture		A							6
Logbook		A							1
Rrepare meeting									
							Subtotal		18
							Accumulative total		41

	Week	3.3							total time
Work done	Date	Mon	Tue	Wed	Th	Fri	Sun	Sat	x
Functional design		A		B		A			15
Measurement plan		A		B					2
Measurement plan research						A			2
							subtotal		19
							accumulative total		60

	Week	3.4							total time
Work done	Date	Mon	Tue	Wed	Th	Fri	Sun	Sat	x
Lecture		A							1.5
Prepare meeting		A							1
Meeting				B		A			6
Doing measurements									
Processing measurements									
Tests research									
Physical measurements machine									
Functional design plan		A		B					2
Finalize functional design					A	A			10
							Subtotal		20.5
							Accumulative total		80.5

	Week	3.5							total time
Work done	Date	Mon	Tue	Wed	Th	Fri	Sun	Sat	x
Lecture		A							1.5
Prepare meeting		A							2
improve functional design			A						2
Adapt functional design									
Go/No Go meeting				B					1
Prepare Go/No go			A						2
Technical design research				B	A				2
Technical design discussion		A				A			3
							Subtotal		13.5
							accumulative total		94

	Week	3.6							total time
Work done	Date	Mon	Tue	Wed	Th	Fri	Sun	Sat	x
Lecture		A							1.5
prepare meeting		A							0.5
Mechanical design		A		B		A			10
Set up TAR									
Meeting				B		A			4
3D Design						A	A		6
Finish TD						A	A	A	5
							Subtotal		27
							accumulative total		121

	Week	3.7							total time
Work done	Date	Mon	Tue	Wed	Th	Fri	Sun	Sat	x
Prepare meeting		A							0.5
Meeting				B		A			4
Polish mechanical design						A	A		4
Assist in PCB design			A	B					4
Component selection				B		A			2
Fablab preparation						A	A		3
							Subtotal		17.5
							accumulative total		138.5

Legenda	
Code	Location
A	Personal residence
B	W2.40 -Saxion
C	W2.29a - Saxion
D	W2.30a - Saxion
E	W2.30b - Saxion
F	OTSWO - Saxion
G	W3.47 - Saxion
H	W2.42 - Saxion
I	W2.43 - Saxion
J	FabLab
# hr	Hours spend

	Week	3.8							total time
Work done	Date	Mon	Tue	Wed	Th	Fri	Sun	Sat	x
Prepare meeting		A							0.5
Meeting				B			A		5
Prepare PCB print				B			A		3
Prepare laser cutting									
Laser cutting									
Assist in pcb design		A				A			4
							Subtotal		12.5
							accumulative total		151

	Week	4.1							total time
Work done	Date	Mon	Tue	Wed	Th	Fri	Sun	Sat	x
Compose component list									
Case design tweaks						A	A		4
Testing		A				A			6
Meeting				B		A			4
Testing Reports		A				A			6
							Subtotal		20
							accumulative total		171

	week	4.2							total time
Work done	Date	Mon	Tue	Wed	Th	Fri	Sun	Sat	x

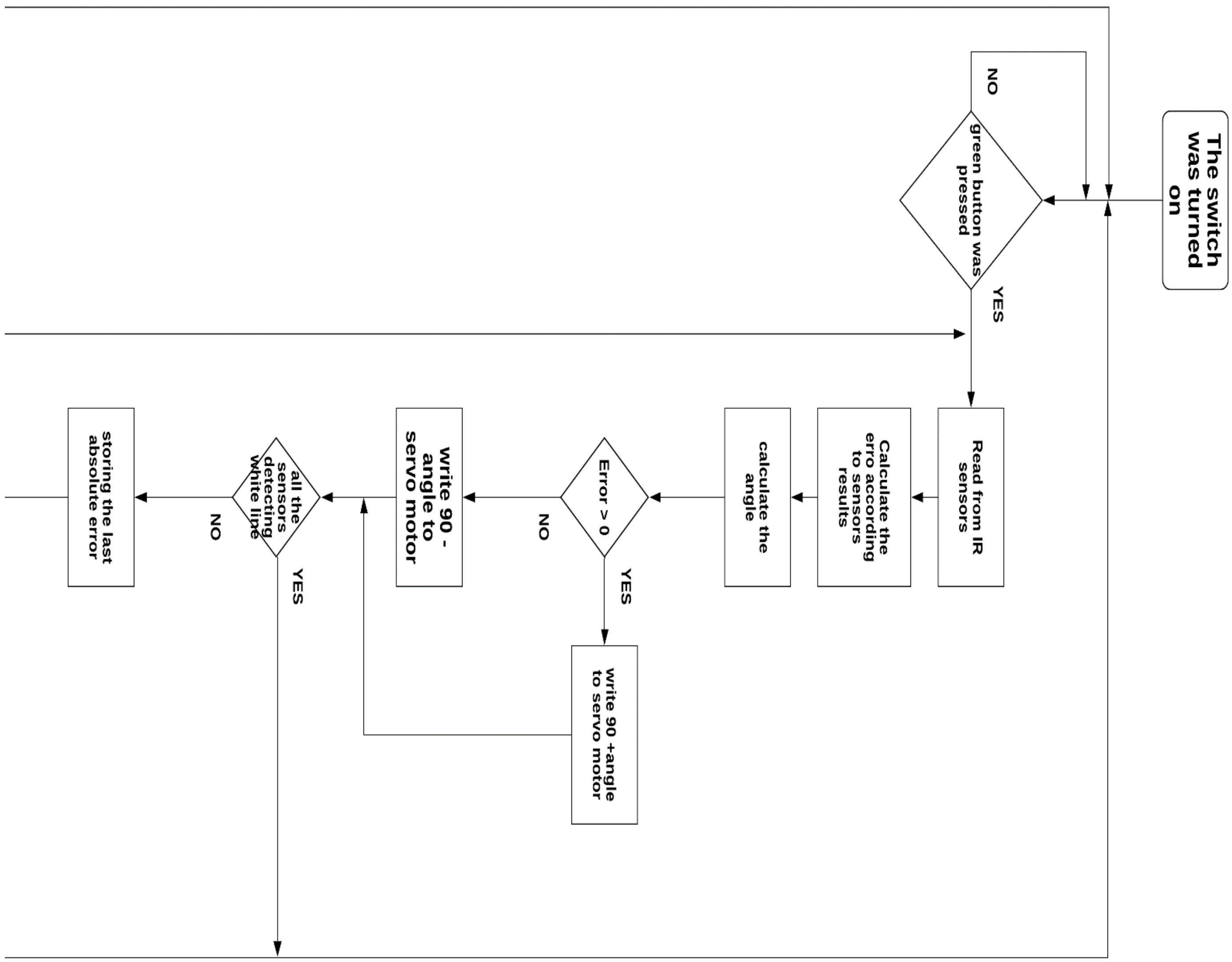
Meeting			B		A			5
Assembly preparations			B			A		3
Mechanical design revison			B		A			2
Finalize report template	A		B					3
Recutting Casing								
Soldering			B					1
Prepare meeting	A							0.5
						Subtotal		14.5
						accumulative total		185.5

	week	4.3						total time
Work done	Date	Mon	Tue	Wed	Th	Fri	Sun	Sat	x

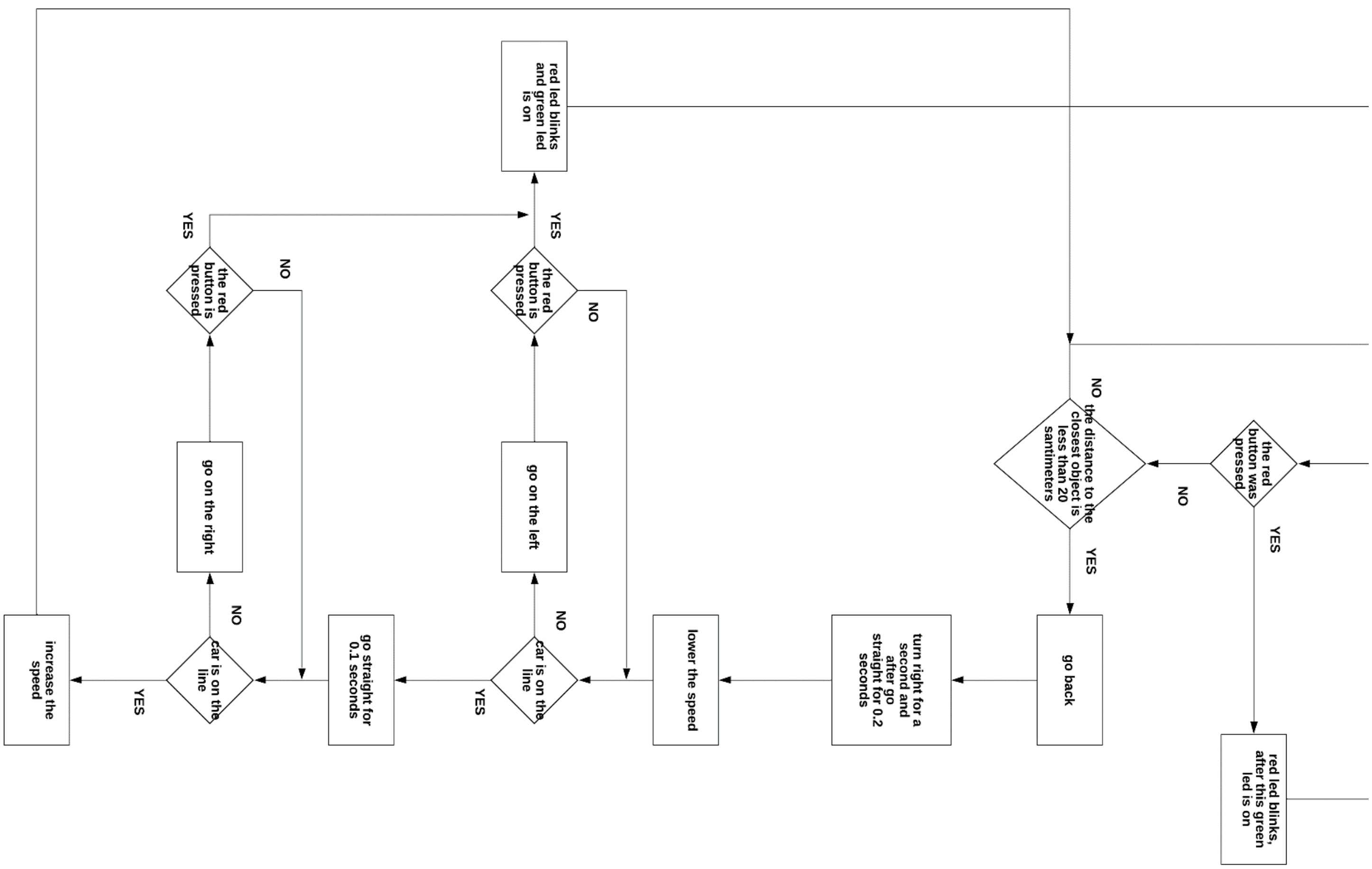
Work done	Mon	Tue	Wed	Th	Fri	Sun	Sat	total time x
Meeting	A		B					5
Assembly preparations			B		A			3
Assembely			B		A	A		5
Preparing Final Report	A				A			4
Prototype preparation			B		A			3
Soldering			B					1
Prepare meeting	A							0.5
						Subtotal		21.5
						accumulative total		207

Work done (week 4.4)	Mon	Tue	Wed	Th	Fri	Sun	Sat	total time x
Meeting			B		A	A		6
Assembly			B		A			5
Making PCB			B					2
Finalize report template		A						1
Finishing Final Report		A		A	A			5
Preparing the Demo			B		A			6
Soldering			B					2
Prepare meeting	A							0.5
						Subtotal		27.5
						accumulative total		234.5

Appendix D


The switch was turned on
green button was pressed
NO
YES
Read from IR sensors
Calculate the erro according to sensors results
calculate the angle
Error > 0
YES
write 90 +angle to servo motor
NO
write 90 - angle to servo motor
all the sensors detecting white line
YES
NO
storing the last absolute error

Appendix D


red led blinks and green led is on
the red button is pressed
YES
NO
the red button is pressed
YES
NO
go on the right
go on the left
NO
car is on the line
YES
increase the speed
go straight for 0.1 seconds
NO
car is on the line
YES
lower the speed
turn right for a second and after go straight for 0.2 seconds
go back
YES
the distance to the closest object is less than 20 santimeters
NO
the red button was pressed
NO
YES
red led blinks, after this green led is on