# Motivation and requirements

Working in the Autonomous Car Project was such an honor and pleasure. Our motivation and mindset were to embrace teamwork as well as enjoying the experience of working with Microcontrollers and entering the exciting field if automation which was totally new to some of us. Building a functioning car before the deadline was of course our main goal but having nice time together while working was also very important. The objective was to build an autonomous driving car using the line tracking sensors and ultrasonic sensors in addition to a live streaming functionality using a camera and a Raspberry Pi. All the system components should work together coherently without interfering or disrupting one another.

# Sketch of approach

Our approach to the project can be summed up in the following topics which started shortly after we received the components of the car :

## Checking that all the required components and parts are available and working

## Building the wooden car prototype and defining the system architecture

## Spliting the whole system into smaller subsystems and establishing the project framework.

## Builing the Hardware and connecting different components (wiring).

## System design using UML diagrams

## Main Code implementation

## Testing phase

## Implementing the Raspberry Pi and camera

## Developing version 2.0

## Code documentation

## Each of these points will be tackled in details in the concept section.

# Concept

## Checking that all the required components and parts are available and working

Once the project box was received we went through the list and checked the presense of all the components stated. Later, the functionality of each component was checked and that proved to be important as the tests showed that one of the line tracking sensors was broken down so it was replaced immediately.

## Building the wooden Car prototype

Once the list of components was checked the process of building the car immediately started. With the help of the already built car model in the lab and by good teamwork the building of the car was carried out quickly and efficiently. Carefully because the wooden pieces were very fragile and step by step, pieces were put in the right place and screwed all together then the sensors and other components were placed in the wooden prototype but not screwed so that adjustments can be easily made at later point of the project if required.

## Spliting the whole system into smaller subsystems and establishing the project framework.

How will we approach the system? That was the next question that needed a quick answer. After discussions it was agreed to split the overall system into smaller tasks and start working on the simplest task then the more complex one then the more complex one and so on. I thought that the system should be built mainly around the ultrasonic sensor; it means that the functionality of the whole system is depending on the wheather the car is going to hit something or not, if not then line tracking sensor and other subsystems starts working and that will be dicussed in details in topic E. The overall framework was to leave a space of freedom for each member of the team to work with his task on his own way and then discuss the progress and do adjustments in our weekly sprint (Agile engineering).

## Builing the Hardware and connecting different components (wiring).

The Hardware was mainly consisted of:

1. Main Adruino board
2. Secondary Adruino board
3. Raspberry Pi and camera
4. 2 DC motors
5. Battery
6. Power supply (for adruino)
7. 3 ultrasonic sensors
8. 2 line tracking sensors

The 5 sensors and the motors were connected to the the adruino board which is considered the brain of the car, while the camera was connected to the raspberry pi, there was also multiple led lights connected to a secondary adruino which was connected to the raspberry pi just as an additional feauture but wasn’t a main part of the system. In the wiring process similar wire colors were used for similar components in order to make the connections clearer, for example green wires were used for connecting the trigger of the ultrasonic sensor and yellow wires for the echo. Later on in the project when adjustments were made on the hardware level this smart wiring concept proved to be very helpful.

## System design using UML diagrams

## Main Code implementation

## Testing phase

The Lab work and testing the car regularly was very important in the project. The testing phase wasn’t just observing what was the car doing correct but instead was focusing on what was the car doing wrong and how can that be fixed as well as trying to understand the overall car behaviour. This mindset helped in eliminating errors in the system and improve the car performance, it also played a key role in the development of version 2.0. Inspection and notes about the tests will be illustrated more in the evaluation section.

## Implementing the Raspberry Pi and camera

## Developing version 2.0

While monitoring our progress in the project and taking notes from the tests we were doing on the car I started working on version 2.0. As mentioned by both professor Rettberg in the sytem design and professor Henkler in the Micro-controller lectures, having only one version of the system is never a good idea because that decreases the system not reliability and makes it not well suited for future updates and improvements. therefore I started implementing a new version of our car, based on our current car version and on the tests we conducted throughout the project. I took the first steps in the new 2.0 version having these questions in mind:

* How can I improve our current version?
* How can I solve the problems we discovered and weren’t able to fix in the first version?
* How can I prepare the system of the new 2.0 version for even more updates and improvements in following versions (version 3.0 and 4.0)?

I decided to take it step by step and I began with designing the new version by doing Activity Diagram for it as taught in the Microcontroller lectures and lab.



I started answering the questions above but not one by one, instead I answered them simultaneously.

That was done by building the whole system from scratch on a more organized basis, going through each part of the system trying to make it better than in the previous, remembering the problems figured out from the tests, leaving room for future updates and improvements and finally adding new features in the system that wasn’t there before.

First, a pseudo code was written to make the overall system more understandable, then developing the main code started step by step.

For the line tracking sensor, I developed it in a sort of function that returns the value ‘1’ if the car is moving in track and returns the value ‘0’ if otherwise. This makes realization and the usage of the line tracking part inside the system much easier. Moreover, an algorithm was implemented for the car to try to find the black line if it cannot detect it anymore, and this wasn’t present in the previous version. Adjusting the line tracking sensors position in the car placing it in a more forward position was also planned in order to decrease the shakiness detected in the original version.

For the Ultrasonic sensor, it was also developed in a similar fashion to the line tracking one as a function that returns either ‘1’ or ‘0’ depending if the car will hit or not. All three u.s. sensors functions were reimplemented in order to improve the car performance, the detection range was adjusted and made closer as well as decreasing the number of variables in the functions, that was done in a bid to increase the car sensitivity and reactivity to objects especially the ones that appear suddenly which was one of the issues faced in the previous version. Last but not least, four escaping algorithms were developed for the car to react to different situations it may face and avoid hitting objects then try to return back to its path instead of moving randomly or just stopping like in the first version.

For the car speed, it was simply set to the maximum which is something that could be improved and updated in following versions. All the car moving directions and even stopping have been implemented in the form of separate functions which facilitates controlling the car and developing complex control algorithms.

One of the new features is an external interrupt button which when pressed by the user enables the ‘manual mode’ and disables the ‘automatic mode’. An interrupt service routine has been therefore implemented as mentioned in the lectures. The function of this button should be then mapped to the GUI to enable the user to control the car remotely which is a future improvement planned.

Another special feature is the connection established between the main Adruino and the Raspberry Pi from the Adruino side. The connection is done using the UART protocol because it is simple, efficient and most importantly full duplex as it allows data transfer in both directions which was learned this semester.

I started working on Version 2.0 before the presentation but it was finished after the presentation so the testing phase is yet to be carried out and completed in future stages.

## Code documentation

After finishing the code implementation documentation was made to the codes of both the main version and version 2.0. The idea behind that was trying to make the code as clean, organized, compressed and as easy to understand as possible.

# Evaluation

1. Line tracking sensor

The decision about where should the the line tracking sensors be placed was one of the trickiest decision done in the project because there were more than one possibility and possibility has a totally different coding approach than the other. The initial idea was to place both sensors in the middle of the car; one exactly in the middle (s1) and the other slightly to the right (s2). The concept behind that was to track the black line by the sensor s1 and then if the line makes a turn the car will detect that by the sensor s2. In this concept, there were four possibilities about the value of the sensors when the car starts running which can be represented in the following truth table

|  |  |  |
| --- | --- | --- |
| Case | S1 | S2 |
| 1. | 0 (detects black) | 0 |
| 2. | 0 | 1 (detects white) |
| 3. | 1 | 0 |
| 4. | 1 | 1 |

Case 1: Car is exactly on the black line

Case 2: Black line starting to turn left

Case 3: Black line starting to turn right

Case 4: Car totally off track (line isn’t detected at all)

The code implementation was almost done but later in the next group meeting it was decided to change the line tracking code and sensors position and placing them on both sides under the car and try to contain the black line with the 2 sensors instead of tracking it because this makes our system simpler, more efficient, and decreases the car shakiness. Also during the tests it was noticed that the brightness of the black line affects the line tracking sensor efficiency; for very dark mediums or very bright mediums the sensors don’t operate properly due to problems with the light reflection which affects the reading of the infrared pulses generated by the line tracking sensor to detect black.

# Summary

At the end of the project the team succeeded in building a fully autonomous moving car with a live streaming possibility as well as an updated version which opens the way to many future improvements.

Although the project’s duration wasn’t long but it was a nice and unique experience which we learned from it a lot and one of the most important things we learned is that systems never work as thought from the first time even if everything seems right, they require a lot of patience and effort to make them really fuction and operate as required. We also learned that Google can’t solve all our problems, we need to use our minds sometimes, that’s why we are engineers.