The Autonomous Car

- Investigating the Concept -

Project Report P2

B311a

Aalborg University Esbjerg
Department of Electronics & Computer Engineering
Niels Bohrs Vej 8
DK-6700 Esbjerg

This report is written in $\LaTeX$.

AALBORG UNIVERSITY
STUDENT REPORT

Department of Electronics & Computer Engineering
Niels Bohrs Vej 8
DK-6700 Esbjerg
http://esbjerg.aau.dk

Title:
The Autonomous Car

Theme:
Analogue Instrumentation

Project Period:
Spring Semester 2014

Project Group:
B311a

Participants:
Brian Hansen
Jacob Lynge
Martin Marcell Gregersen
Kristian Brogaard Kristensen
Rolf Magnus Roos

Supervisors:
Akbar Hussain
Torben Rosenørn

Copies: 4

Page Numbers: 82

Date of Completion:
May 20, 2014

Abstract:

The purpose of this report is to figure out how object avoidance and motor control for an autonomous car could be implemented intelligently. These topics are being analysed while keeping in mind that they have to fit within the framework of an autonomous car. Finally we have evaluated the public opinion for such system, and found there is a definite need.

In order to evaluate our solution we have built a proof-of-concept vehicle which had to comply with a set of requirements which were defined in the delimitation. We reduced the objective down to being able to drive around and avoid objects in a controlled environment. Finally we will also evaluate how the real world environment will affect our system as well as how it will scale when being applied to a full-size car, which will increase the demands of the system.

Our effort yielded the result that, while the technology is definitely mature enough for creating a vehicle such that we visualise it, our choice of hardware was flawed. We have however found other technologies that should be able to solve the issue and if we were to continue our endeavour we would know where to begin.

Contents

Preface

In this section we would like to thank a fellow student, Christian Reimer Hansen for lending us his Sparkfun Redboard after our own Arduino Uno malfunctioned. This kindness ensured that we could complete our solution for P2.

Aalborg University Esbjerg, May 20, 2014

Brian Hansen
<bhans13@student.aau.dk>

Jacob Lynge
<jlynge13@student.aau.dk>

Martin Marcell Gregersen
<mmgr13@student.aau.dk>

Kristian Brogaard Kristensen
<kbkr13@student.aau.dk>

Rolf Magnus Roos
<rroos13@student.aau.dk>

Chapter 1

Introduction

The transportation today is relying heavily on cars, this leads to congestion during rush hour in densely populated areas and is also relying on the awareness and skill of the human driver. The unaware, intoxicated or reckless driver is the main cause of accidents in traffic. We believe, that some of the accidents can be avoided by making the cars more or even fully autonomous[1]. Further, as a result of this automation, the traffic could become denser due to the reduced reaction time, thereby utilising the road infrastructure better.

If we look into other branches of transportation like; trains, aeroplanes and ships, they are already partly controlled by automation. A perfect example is from the civil aviation, where commercial aeroplanes are highly depended on the integrated autopilot and are used during every flight. The autopilot is so advanced that it could, and sometimes does handle the aeroplane, from take-off to flight at cruising altitude and the landing of the aeroplane afterwards. Although it is common practise for the pilots to steer the aeroplane at crucial moments like take-off and landings, millions of people worldwide put their trust in automated systems like this every day.

The same goes for public transportation by train and in particular the metro infrastructure in some major cities, are controlled and driven by automated systems that does not have a human operator. They are monitored from a central location and human interaction only occur during malfunctions like mechanical/electrical issues, unwanted objects on the tracks and other unforeseeable events. The third means of transportation that is highly depended on automation are ships, which have many things in common with commercial flights. Unlike trains that are fixed to the tracks and cannot deviate, aeroplanes and ships rely on advanced GPS along with RADAR, to be aware of their positions and avoid obstacles in the way. They navigate on invisible highways or corridors in the air or at sea, that are based on international agreements across countries to ensure high level of safety. Although the air and sea is relative traffic free, compared to roads, bottlenecks is a reality for example at airports and harbours around the world.

Since it is possible to control these big machines at the highest safety and with a low rate of accidents, why have these automated systems not been integrated to the most common means of transportation, the car?

[1]By autonomous is ment a fully automated system.

1.1 About the Group

This project is made by five university students from Aalborg University Esbjerg, as our P2. This is the third project and were written during the second semester. The lectures this semester are ”Calculus“, ”Elementary Electronics“, ”Digital Design & Sensors“ and ”3D CAD Modelling and Product Development“.

1.2 Initiating Problem

An autonomous car needs many different sub-systems, among others are object avoidance and motor control, how can this be done smart?

Chapter 2

Problem Analysis

2.1 Introduction

In the following section of the report we will be analysing our initiation problem: ″An autonomous car needs many different sub-systems, among others are object avoidance and motor control, how can this be done smart?" As we will be investigating both the technology needed to solve the problem but also try to figure out what kind of implications it can have for society.

As our problem will be part of a bigger system that is an autonomous car, at times throughout we will be dealing with the autonomous car as a concept. Instead of the systems we are working on. This can for example be seen in the demand analysis, as the need for our solution will be solely dependent on the need for an autonomous car.

2.2 Finding the Problem

In order to start off our project, we tried to figure out what problems the car had, by making a 6W-diagram (See Appendix A) with the core problem "Problems the car has" to make things more simple and get a better overview of how to proceed. This diagram however was very substantial and did not help narrowing our project, but it helped us get a few things listed we did not think of before. A common vector was noticed namely accidents, and it was thereby decided to make a new 6W-digram with the core problem "Car accidents" (See Appendix B). From this it can be deduced that the main problem of interest is the human driver. To proceed from here the human factor from accidents were taken out of the equation and that left us with a driverless car. If an autonomous car could be the solution to fewer car accidents, it first has to be investigated further, bringing up the next step in the analysis; analysing the autonomous car by using a breakdown diagram to deconstruct its components.

Problem Breakdown

The autonomous car has a lot of different systems and subroutines to make it move and to avoid it crashing into things as it moves along. The systems can be split into three subcategories namely movement, awareness and reasoning. Movement covers the cars ability to move and change direction, awareness is used as an abstract term meaning awareness of its surroundings and by reasoning is meant the systems ability to figure out how to react in a given situation to get the best outcome.

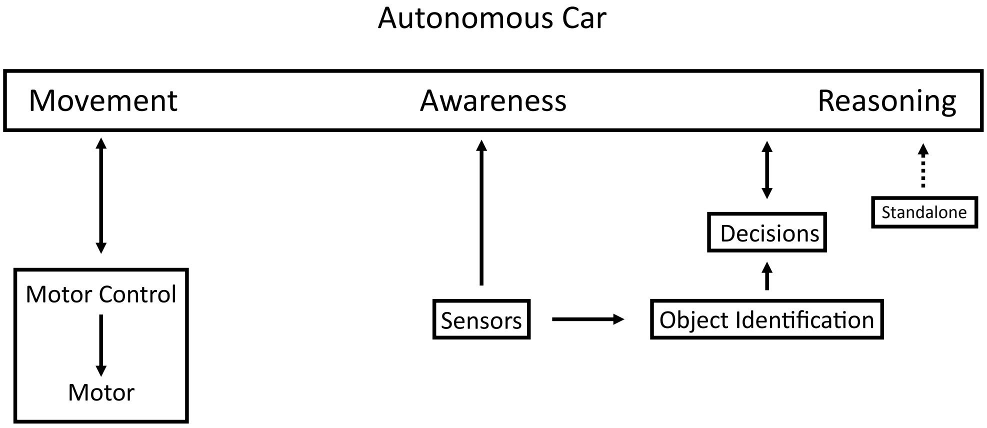

Figure 2.1: Breakdown-diagram

As seen from the breakdown diagram Figure 2.1 there is a lot of different hardware and software necessary for the autonomous car to function. The simplest task of the autonomous car is making it move, the speed sensor, motor and motor control handles this. The tricky part is making it aware and avoid obstacles in the way like children, dogs, etc.

2.3 Demand Analysis

One part of this analysis is based on accident statistics and the assumption that the numbers of accidents, can be brought down by applying more safety equipment into cars. The other part of this analysis is based on what the industry is developing and researching.

Statistics covering Danish car accidents

Year	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011
Cars Below 2001 kg*	49183	49030	49280	50149	49705	49686	51229	51501	51155	50727	52067
Accidents Total	4378	4557	4502	3973	3398	3312	3424	3093	2628	2173	1990
Fatalities	248	254	244	191	175	146	177	200	169	137	110
Major Trauma	1705	1772	1672	1468	1210	1119	1215	1121	968	826	794
Minor Trauma	2425	2531	2586	2314	2013	2047	2032	1772	1491	1210	1086

Table 2.1: Summation from the tables in Appendix C

**Unit: million person kilometres*

By looking at Table 2.1 it can be seen that there is a small increase in the amount of person kilometres over a ten year period and that there have been a decline in accident in the same period. Although the table shows decrease in accidence there are still room for improvements to lower the amount of accidents and thereby the cost for society, one way to lower the amount of accidents is to improve safety in the cars and to use more automation in the cars.

We believe that to get the amount of accidents down further will require one or more of the following; better education of drivers, a change of how human behave in the traffic, change of rules and legislation(more strict), improved infrastructure or safer cars. It is not realistic to get zero accidents in the traffic however by applying a combination of these methods we can approach it. The change of infrastructure, driver education and legislation is done by politicians, and as such it is part of a democratic process that could take a long time. The development of safety equipment for cars and implementation of it, is the only thing that car manufacturers have a direct influence over, and it is used as a competitive parameter in the same way as the price of the car. From a technological perspective the development of safer cars is challenging and building these is the industry's only way of influencing the statistics positively.

Intel conducted a survey in the US anno 2014, to find out what the general opinion were on future technology, like smart cites, autonomous cars and transportation. In this survey it was shown that up to 44 % would use automated transportation systems, some of the reasons people mentioned were better road navigation, fewer traffic jams and increased comfort((1) & (2)).

The concept of an autonomous car and the development of them are pursued by companies, like Google(3), Mercedes-Benz(4), Nissan(5) and Toyota(6) who all have prototypes of autonomous cars. The main reason for these companies to invest in research of such systems is most likely to get the competitive edge over other companies. The companies are not investing so heavily in this kind of technology if they do not see some kind of benefit from the technology, i.e. they need to get their product sold at some point or get the prestige and PR benefits.

There are a demand for safer and more automated cars, based on these findings. The survey conducted by Intel, showed that a decent percentage of the population would give up some of the control of the cars to get better comfort, less traffic jams, faster and smother point to point travel. The car manufacturers' development and research into autonomous cars, are done to get a competitive edge, and to get the benefits of PR and the prestige that the result of this research yields. The car manufacturer have an interest in getting their product sold at some point, but even without getting the autonomous car on the roads in the near future, they benefits from the PR it can create and they have the possibility to adapt parts of the technology to their car production to stay in the forefront. The wish to drive the accidents statistic down also raises a indirect demand in the form of demand for safer cars.

2.4 Stakeholders

The moment a product is published; there are people interested and opinionated about it. It is very crucial besides a demand for it, that the product has a positive influence on the world. The impact different parties can have on the autonomous car is analysed in this section.

2.4.1 Actors

An actor is a part of the interested parties, and they have a possibility of affecting the development of the product. A few examples of actors can be investors, politicians, manufacturers and their competitors.

Government

As the government is the main legislative body, their opinion of the autonomous car will have a very direct impact on which kind of rules and taxes a fully implemented autonomous car would operate with. As we see the possibility of severely bringing down the number of traffic accidents, and ensuring the car will be operating 100 % within the traffic laws, the car should be very attractive from a governmental point of view. There is also the ever present environmental concerns which could also be improved via an autonomous car as it will also overcome the effect that people's mood affect fuel consumption.

Insurance Companies

The effect a successful autonomous car would have upon insurance is really interesting as who has liability for an autonomous vehicle. It will hold some challenges in the future, but the reduction in accidents that should be seen from autonomous cars would make insurance policies for these cars very attractive. That is of course, if the regulations on who is responsible for accident with the car is still with the owner of said car.

Investors

A project such as this will require some serious funding, as is the case in all development of solutions to real-world problems. There are several places such funding potentially could be found:

- Car manufacturers
- Investment funds
- Technology companies

It is of course natural to expect some kind of requirements that is to be met when dealing with investors, car manufacturers for example, might have special clauses limiting the sales of the system to their company. While investment funds will likely be more focused on the profit return period. Technology companies is a wide term, but meaning any company that profits from or has an interest in appearing on the cutting edge of technological possibility. The kind of influence this kind of investor will have is difficult to foresee.

2.4.2 Technology Carriers

To be a technology carrier you have to possess six things:

- Interest
- Power
- Organisation
- Information
- Access
- Knowledge

By having an interest is meant a company that is willing to choose and develop the technology. To possess the power means that you have the social, economic and political power to implement the product. Organisation is to be able to organise workflow within a company/organisation when handling complex projects. Information is to know that the technology is existing. Access is to have the technology available, and knowledge is to have the know-how of the technology and how to use it, or if that fails, the ability to acquire said knowledge.

If a company do not possess all these things, then they cannot constitute a technology carrier on their own. Companies can form groups and thereby constitute a technology carrier together, with each of them bringing some of the six points to the table.

Car Manufacturers

The autonomous car system could be adapted for any car technology available to car manufacturers. They will also be certifying the mechanics servicing the cars when they need repairs and software updates. As car manufacturers in the future will most likely want to be part of the market for autonomous cars, they will have to either create their own system or acquire a proprietary solution. For this reason the manufacturers can be potential investors.

Technology Companies

Apart from the car manufacturers which has an obvious interest in the autonomous car, there are the more general “technology companies”, companies making their business being on the cutting edge, working to accumulate patents across many different sectors, an obvious example would be Google. These companies will both have the knowledge, finance, power, organization, interest and access to realise this technology.

2.4.3 Interested Parties

Interested parties can be a person or a group effected by the implementation of the product that is to be made. It can also be a group or a person affecting the implementation of the product. This can for example be environmental organisations or consumers.

Health and Safety

The health and safety issues is about work place health and safety, and since the autonomous car idea does not change the work routine for the workers, it is therefore irrelevant for our project.

General Population/Consumers

The autonomous car, as a fully implemented system, should make people far safer in the sense that; it has a drastically quickened reaction time, it is not subject to factors as fatigue, lack of awareness nor driving under the influence. Furthermore it could help people without a license or with disabilities to move around more easily as well.

According to Intel(1) the public interest in an autonomous car is high, 44 % of Americans say they would like to live in a city with driverless cars driving them around. Moreover almost 60 % would like to disclose travel patterns to a database if it meant emergency personnel would be able to respond quicker due to them knowing the traffic status. 54 % responded they would let an intelligent system calculate their itinerary if it meant a smoother traffic flow overall not for just themselves. If the status quo of traffic is known, more precisely that is, optimisation of infrastructure is easier and congestion could be reduced dramatically.

On the other hand, the public might have an inherent distrust or even disliking of an autonomous car since it is first off asking people to put their trust in a gadget which they might not even understand how works, and which, if it has any kind of errors or faults, could cost them their life. Finally there is also people who considers driving as being close to the ultimate freedom, and have the car drive for you, might not appeal in any way to these individuals.

Liability

This is an exceedingly interesting area when dealing with an autonomous vehicle, because today, we are used to the liability being with the person who did something wrong. However, if the car is driven by a completely autonomous system, it seems unlikely that the person sitting in the car, without influence over the manoeuvring, would accept responsibility for the accident.

It seems that it is possible, that the liability will be placed with the manufacturer of the vehicle, or at least, some speculate that this will be the way it is handled, if you compare to other technologies(7). This is a very new area so how this will be handled in a future with autonomous vehicle being a common sight on the roads is still unknown.

2.4.4 Legislation

When creating a new product it is very important to make sure it follows the legislation in the country/union the product is to be released in, and stay informed on upcoming laws that might affect the product.

There will be legislations and standards, on how the autonomous car has to be build and what demands it must fulfil, to be brought on to the roads. Because it is a prototype that is being developed the legislation has been disregarded in this project, but if the prototype is going to be developed into a fully functioning autonomous car, these legislations have to be followed for the manufacturing, development and EOL(**E**nd-**O**f-**L**ife) of the car. Electronics that are specifically build for the car is legislated under the ELV(**E**nd of **L**ife **V**ehicles) directives, and electronic equipment such as car radios sold separately at another dealer is subjugated to the RoHS. For correct waste management the manufacturer must comply with the WEEE directive. If an electronic device is manufactured to go into a car, it must comply with the E-mark and they can get their product CE-mark certified if they fulfil the requirements. For a complete technical approval of a car, vehicle systems and separate parts in EU, the manufacture must comply with the EC Directive. If vehicle systems or other components are to be fitted on an already approved car in EU, it must comply with the UN Regulations. For a complete technical approval of a car, vehicle systems and separate parts in the EU, manufacturers must comply with the EC Directive. If vehicle systems or other components are to be fitted on an already approved car in EU, it must comply with the UN Regulations. For more details about CE-mark, EC, UN, E-mark, WEEE, RoHS and ELV (See Appendix D).

2.5 Technologies

In this section, we will be looking at all the equipment and technology, which we will be using in this project and to build the proof-of-concept vehicle. This overview include sensors, electric motors and Hall Effect sensors, to get a better understanding of how they work and how to utilise their full potential.

2.5.1 Distance Sensing Technologies

In order to enable an autonomous car to avoid obstacles it will need to be able to locate and measure distances to objects in its path. This can be achieved through a variety of means, in the following section we will go through a selection of the possible technologies that can be utilized in order to achieve this. Finally, we will compile a table rating each sensor in a 1-5 scale in categories that is relevant.

The grades given in the tables are based on a subjective evaluation of four categories, they are roughly defines in the following:

Range: Is given as an approximation of "easily" adjustable ranges, meaning without a lot of extra work, where a 1 would mean extreme close ranges. And a 5 is possibility for ranges way beyond the need for a car, in the kilometre scale.

Directedness: In essence 5 is a point measurement where a 1 would be a very broad fan, approaching 180°, this can of course be both a good and a bad thing depending on the situation.

Environmental influence: Is how much weather effects and other background noise can confuse and/or disable the sensor, where 1 means it should work in all conditions and a 5 means there are conditions which are both often occurring and critical for the sensors functionality.

Complexity: Is a subjective evaluation of how complex the sensor would be to build and implement. Factors affecting this is both the price of components, the amount of documentation easily available and how much extra will have to be done if for example you need to sort away uninteresting reading caused by a wide detection area. 1 would be a well-documented, easily build system where a 5 would be one that requires special components, complex programming and hard to get documentation.

Ultrasonic

The principle in the ultrasonic is fairly simple, the sensor emits ultrasonic waves, after which it awaits an echo, where the time interval can be measured. The time difference from emitting to receiving a signal again can then be used to calculate a distance to the object bouncing back the wave.

Range	Directness	Environmental influence	Complexity
3	2	5	2

Table 2.2: Ultrasonic sensor

The ultrasonic-sensor has a major drawback because it has a hard time detecting "soft" objects that can absorb sound waves. Furthermore, it can also get confused by sound waves that bounce off of objects in an indirect fashion, so that it travels from the emitter, to the first object, bounces, hits a second objects and bounces back towards the sensors(8).

Sound waves does have another drawback and that has to do with the physics of sound waves. Sound waves are the movement of particles in any medium could be air as well as steel, this causes some problems when utilised in distance measurements. If the wind blow perpendicular to the road the ultrasonic wave transmitted, could potentially be blown off to the side of the road and the echo might not reach the sensor. This poses a problem when you try to rely on this sensor type for distance measurement especially when that measurement is crucial to the system. As mentioned sound waves is movement of the particles in a medium, this means if the partials in said medium moves faster i.e. the material is warm, the sound travels faster. This also has to be taken in to account because the distance measurements might not be the same from day to day or even from hour to hour, depending on the temperature and the humidity of the air(9).

Infrared

The infrared distance sensor works with the same principle as the ultrasonic sensor, it uses infrared waves instead of sound waves however. If used in places where there is chance of infrared coming from other sources than the sensor, it is needed to modulate the infrared emitter so that the sensor is only looking for infrared light pulsing at a given frequency, this ensures that the sensor does not get confused by for example, the sun.

Range*	Directness	Environmental influence	Complexity
2 || 3	5	3	2

Table 2.3: Infrared sensor

*The main challenge with an IR-sensor will be creating one operating in a wide enough minimum and maximum range in order to remain a feasible option.

Finally the infrared sensor will have to be created so that ambient light will not confuse the sensor. This is due to the fact that the sun also emits infrared light, and this can confuse the sensor. Finally, objects of different colours and materials, can have different reflection indexes, which can alter the reflection of the infrared light, this can also give faulty readings(8).

RADAR

An acronym for ”**RA**dio **D**etection **A**nd **R**anging“ which describes the basic functionality fairly well. It actually work with roughly the same principles as the Ultrasonic sensors, instead of emitting ultrasonic waves, it emits radio waves.

Range	Directness	Environmental influence	Complexity
5	3	4	4

Table 2.4: RADAR sensor

The RADAR system is quiet an efficient system, and is used in many applications. It is a bit complex owing to the need for radio wave modulation, transmitting and receiving, furthermore some algorithms to sort the interesting waves from non-interesting can be difficult to implement(10).

In the case of implementing RADAR on cars, it will be necessary to pick the right frequency of the RADAR as to ensure that it will not start detecting rain droplets(11). But if modulated with the right frequencies it possible to adapt the RADAR to detect almost anything, as long as it is not specifically modelled and/or coated to decrease its RADAR signature(12).

LIDAR

LIDAR is based on the same principles as both RADAR and ultrasonic sensors. However LIDAR, uses a laser as the medium to measure distances. The sensor then measures the time it takes light from travelling out from the laser, to the object in the path and back to the sensor receiving the light signal, the time taken for this can then be used to calculate the distance to the object.

Range*	Directness	Environmental influence	Complexity
5	5	4	4

Table 2.5: LIDAR sensor

*The distance and effect from the environment or weather is difficult to precisely specify as it is wholly dependent upon the wavelength of the operation laser.

As the LIDAR system is very much still based on light, akin to the infrared sensor, it can be subject to the same faults, it is however much more directed and powerful, so generally the effect is reduced and at close range it should measure most materials(13).

Optical(w/Camera)

Using cameras to calculate distance to object can be done in a variety of ways, covering every way is out of the scope of this section, however, one way it can be done is using two cameras as a stereo camera, in this way you can get two images, of the same scene, and with a known distance between each camera it is possible to use triangulation to calculate the distance to an object. This will also require some form of image recognition software in order to identify different types of objects, and figure out which ones is of interest for the rest of the system.

Range	Directness	Environmental influence	Complexity
4	5	5	5

Table 2.6: Optical(w/Camera)

When implementing cameras to calculate distances to objects, it holds the clear advantage that it is possible to make the distance sensor able to also identify the type of objects as well as distance to said object. This makes it a really powerful device in the autonomous car setting.

However, it also holds severe drawbacks, as human sight can have severely reduced view range in adverse weather, likewise the camera range finder will have the same disabilities.

2.5.2 Speed Sensing Technologies

A vital part of a normal car is its tachometer that rely on a speed sensor to make it function and provide important information to the driver. In many modern cars, the readings from the speed sensor is what makes the cars cruise control function by comparing it to the engines revolutions to obtain the selected speed setting. In an autonomous car a speed sensor will be as important for the system to know the speed of the car and thereby comply to speed regulations and keep safe distances to nearby objects. There are different types of sensors for measuring the current speed of a car and they are presented in this section.

Hall Effect Sensors

The most common speed sensor in the industry and cars worldwide is based on the Hall Effect and the sensor is named thereafter. It is a simple sensor that reacts to the presence of a magnetic field by sensing the change in the magnetic field and converting the readings into a signal. The sensor is basically a transducer that outputs the read signal as a change of the voltage from either high to low or opposite, depending on the direction of the magnetic field passing it. This enable the possibility for the receiver of the signal to know what direction the rotation goes. The sensor is usually placed near a rotating metal object like a gear, shaft or flywheel See Figure 2.2 and they are all connected to an engine to make them rotate.

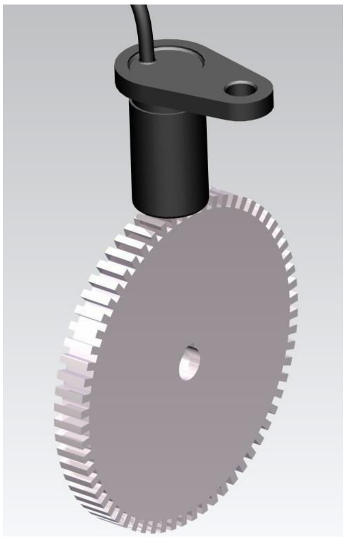

Figure 2.2: Illustration of a large industrial Hall Effect sensor reading a gear.(14)

The sensor illustrated here in Figure 2.2 is made for large industrial use only in machinery and in cars. For making the Hall Effect sensor a useful tool for accurate speed readings some factors have to be known such as number of teeth and diameter of the gear.

Furthermore the total distance travelled by, for instance a car, after one full revolution of the gear taking into account the effect from a gearbox. After these factors is known, the output from the sensor can be taking into account and a precise speed can then be calculated.

The output from a Hall Effect sensor is analogue, in the sense of changing voltage. The way the sensor does this is by changing the voltage going through acting like a variable resistor, this results in an altering output voltage that by a big enough margin can be distinguished but without breaking the circuit.

For a better illustrative understanding of the variation in the magnetic field from a gear (See Figure 2.3). The different colours represents the intensity in the magnetic field where red is strongest and dark blue is the weakest area in relation to the hall sensors coil. From this information it can be seen that the highest magnetic force is concentrated around the gear tooth closest to the hall sensor. This results in a low resistance inside the hall sensor that thereby reduces voltage drop across the resistor depending on the intensity of the magnetic force and the opposite happens when the hall sensor is in between gear teeth resulting in a high resistance and a high voltage drop.

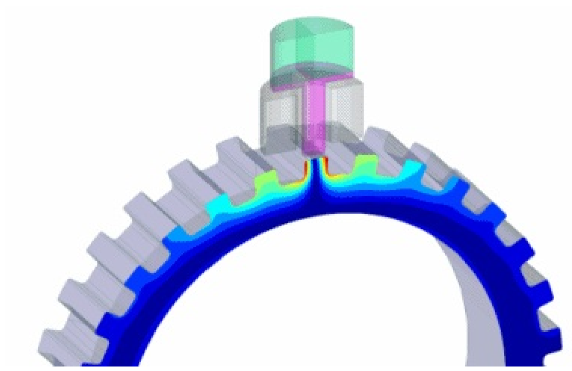

Figure 2.3: Illustration of the magnetic field from a gear(15)

The analogue output from the sensor in terms of different voltage that can be directly interpreted by an ADC that converts the signal to a digital one.

Hall Effect Latch

Some Hall Effect sensors has been designed so that they can easily be applied in digital systems, here the Hall Effect sensor is extended with a Schmitt trigger which will switch at a present value. In effect this means that the output from the sensor is reduced to either a magnet being close or not. The threshold for how many *gauss* is equivalent to a magnet being close is different from component to component.

In the illustration below (See Figure 2.4) where “Output” is marked, the result of the Schmitt trigger can be seen. The output signal from the Hall Effect Latch is changing based on a magnet being present or not.

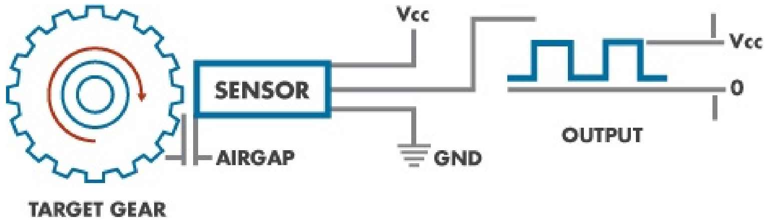

Figure 2.4: Illustration of the output signal from a Hall Effect Latch(16)

LED and Optical Receivers

An optical receiver is simply a photo resistor that is sensitive to photons and reacts in the form of change in its resistance. The light source can come from anything like the sun, a bulb or an LED, but since they emit different levels of light intensity the variation of the resistance can vary by a huge margin. For example when a normal LED from household appliances lights up, the intensity is very low compared to the light from the sun. This relative low difference in light from the LED, when it is on or off, can still be distinguished by measuring the different resistance in the photo resistor.

This technology is already used in a computer mouse, or at least in earlier models to determine the direction of the mouse's movements. For sensing the speed of a shaft or a gear can be done by using an optical receiver connected to an LED and a simple counter.

The disadvantage of using this technology in the environment around a car is the sensor is error prone in the sense that the sensor can be obscured. Either by oil, dirt or other debris from its surroundings and the humidity level around causing the sensor to be blocked by mist. For the LED component the same conditions is in play for preventing the correct amount of light to reach the sensor. LEDs are also known to lose their full intensity over time making them unreliable as a precise light source for the photo resistor, causing faulty speed readings(17).

2.5.3 Motor Control

Cars and autonomous cars need motor control to be able to change speed and start or stop the motor, the control in new cars are electro mechanical, where a mechanical input is computed in a microcontroller which then outputs to an electro mechanical unit on the motor.

The control of a motors speed(**R**evolutions **P**er **M**inutes) and direction(clockwise or anticlockwise), can be done in different ways depending on the motor and the type of control. The first distinction is based on the type of motor, mechanical i.e. some kind of engine based on combustion this could be turbines, diesel engine, gasoline engine or gas engines. The other main type of motor is electrical motors, this type is powered by electricity and are subdivide into two main types, the AC motor powered by alternating current and the DC motor powered by direct current. In this section the combustion engines are only mentioned as another motor option. It will be the electrical motors and the controlling of this type of we are looking into, this choice is made because it is perfectly suitable for a proof-of-concept

AC Motors

The **A**lternating **C**urrent motors are made in many different sizes and for various speeds, the rating plate on the motor is made by the manufacturer and states what kind of motor it is and the motor's capabilities, an example could be; the speed it is designed for, number of phases it need for powering it one or three, the amount of amperes it consumes and the frequency at which it runs or rather what frequency the rating plate speed is valid for. There is also voltage information as well as power consumption info i.e. the amount of watts the motor was designed for. If the motor is a three phased motor the rating plate, will also indicate under what kind of set up these data is valid.

Control Methods for AC Motors:

Start/stop
Control of the motors revolutions could be by simply starting and stopping the motor in a sequential pattern, this will result in a speed between maximum and standstill similar to PWM.

Drive or Converter
Another way of regulating the speed of an AC motor is by altering the frequency it is supplied with through a converter, by altering the frequency between that of the rating plate and zero the motor's speed can change. Although the frequency can be increased above the the rating plate specification, to get an even higher speed than stated on the rating plate, this option will require a closer look into the data sheet of the motor or a confirmation from the manufacturer about what maximum speed the motor can run at, the limiting factors are the bearings or problems with vibrations if the speed gets too high(18).

DC Motors

The **D**irect **C**urrent motor should be supplied by the voltage on the rating plate, example a 12 V motor is supplied with +12 V and 0 V lines. Like the AC motor the DC motor have a rating plate stating some of its characteristic. The DC motor can be obtained in different sizes and for different speeds and loads.

Control Methods for DC Motors:

Start/stop
As with the AC motor, the DC motor can be controlled by just starting and stopping it in a sequential methods, but unlike the AC motor it can not get fed by alternating frequencies.

PWM
The DC motors speed can be controlled by a method called **P**ulse **W**idth **M**odulation, this method is similar to start/stop control, but it gives a more dynamic speed alteration, the controller for this method starts and stops the current supply to the motor and in that sense it is similar to start/stop control. The controller also controls the length of the periods for starting and stopping, by doing this it can make the DC voltage pulse between zero and the set voltage supply((18) & (19)).

Comparison of AC and DC Motors

For many applications the choice of motor, is based on price and what kind of power that are available. The main differences are that above approximately 1/2 HP, the AC motor are cheapest but they have a more expensive control system if they should have variable speed. The position control that can be obtained in AC motors are not as good as the control obtained in DC motors, the control of speed in DC motors are cheaper. The DC motor have better position control capabilities than the AC motor. Most AC motors are build for supply voltage in the range of 110 V to 400 V, whereas the DC motor are most frequently used in application where they are fed with much lower voltage than the AC motor. Depending on application there is also the option to have a DC or AC motor custom made to suite specific needs, but that will increase the price. The cheapest solution is to look at the need in the application and then find a suitable motor that can be bought off the shelf((20) & (18)).

In an electrical car, the choice would be AC motors, although the batteries deliver DC voltage, the conversion is effective and weight power-ratio in AC motors are better than DC motors(21).

2.5.4 Actuator

To give the automated car the capability to steer around obstacles, and not just drive and halt, it needs some kind of actuator and a feedback system to the control system.

There have been two dominating mechanical ways to steer an ordinary car until now, the two methods are the recirculating-ball steering and the rack-and-pinion steering(22), both systems can be fitted with power steering in the form of a hydraulic amplification to assist in turning of the front wheels. These methods are still the dominating ways to steer cars, and these system are based on many mechanical parts. In recent years, there have been a development in EPS(**E**lectric **P**owered **S**teering) or drive-by-wire. Examples of cars with that kind of technology are GM's "Hy-wire" which is a concept car and BMW's "528i xDrive"((23) & (24)). The main difference between EPS and drive-by-wire are that in the EPS system there is still a mechanical connection between the steering wheel and the wheels. Whereas the drive-by-wire is a translation of the movements of the steering wheel to electrical signal, which goes to microcontroller, that then signals an actuator that turns the wheels of the car. Both drive-by-wire and EPS uses a form of electrical linear actuator or an electric motor and a gear.

Comparison between the systems:

The advantages of hydraulic power steering is that the system gives the best feedback to the driver about the movements of the front wheels. The disadvantage is that it requires a mechanical connection to the front wheels and that it needs a hydraulic pump powered by the car's engine.

The advantages of EPS is that the hydraulic pump is absent in cars and this leads to a better mileages because the engine does not need to power a hydraulic system. The disadvantage of EPS is that some of the driver's feedback when turning is smaller and thereby leads to the sense of not turning enough.

The advantages of drive-by-wire is that it frees space in the engine compartment of the car, because there is no need for a steering shaft, with physical connection to the front wheels. Like in EPS the system works without a hydraulic pump, that way it uses less energy and frees up even more space. The disadvantage in drive-by-wire is the driver might miss the feedback from turning the steering wheel.

In an automated car, the mechanics of the steering is not that interesting from a technological point, because there is already plenty of solutions for that on the marked, but the electrical control of the actuator through a microcontroller are interesting from an electro technical point of view.

The drive-by-wire uses an actuator directly and would be the preferred technology, but as the legislation is currently, it can only be used in prototyping and as a proof-of-concept. The reason is that there is a legislative requirement of a physical connection between steering wheel and front wheel. In the case of hydraulic power steering it could be retro fitted with an actuator to the steering shaft. Another possibility to achieve this could be the use of electromagnetic valves to control the hydraulic system. This method would probably be the easiest and the method requiring the least space to implement. The electro mechanical power steering uses an actuator already, because of this it would be possible to utilize this by altering the control mechanism of this actuator in a way so it is the microcontroller that controls the actuator.

2.6 Knowledge

2.6.1 Environmental Factors

In order to choose the right sensors to implement within an autonomous car it will be necessary to think about the kind of things that is present on the roads where our car will be operating, ignoring the kind of objects that are present could result in a wrongly picked sensor which will leave the car unaware of an object which could in turn cause accidents.

The first thing to consider is the weather; this can potentially obscure the view or confuse sensors as particles is aerosolised. This can range from sunny, cloudy, rainy, snowy, dusty etc. Some sensor has the potential to detect these particles, and this can in turn result in the sensor being confused and have their sensor distances reduced if not completely nullified.

Next are the objects we can find on, or near, the roads. There are the obvious hard objects such as cars, signposts, trees, barriers, etc. and then there is the other challenge of more soft objects such as humans, animals etc.

Added to this is the wide variety of materials and colours these objects can be made of. This adds up to, that the sensor or sensors that are chosen for keeping the autonomous car updated upon the roads conditions and the location of obstacles in relation to the car, will have to be able to sense a wide variety of object types to ensure that the car is not left blind to certain object types.

2.6.2 Car Environment

The environment the car is exposed to can at times be rough and implementation of additional sensors and systems has to be evaluated and assessed to ensure the ruggedness and safety of the additions. The same goes along with the environment the car in itself, poses. Cars with traditional piston engines creates vibration and heat, electronics systems in the car and more specifically the engine compartment is exposed to these factors plus the cold and potential moisture when the car is parked.

Adding more sensors could pose as a problem because the potential of failure rises, more points where things can go wrong. If a car has to act on its own, redundancy has to be introduced to ensure if things go wrong and a sensor returns a faulty signal, the system can adapt and manage using the input from the other sensors until the problem is rectified.

Cars today already have many safety systems incorporated in them; ABS, ESP and RADAR guided cruise control. All of these systems have one thing in common they act as a stand-alone system meaning they sense what is required and act accordingly they do not need any other system to specify when and how to act. The stand-alone strategy is smart, this means specially manufactured parts can handle one job and handle it well instead of one system managing all the tasks. When envisioning the autonomous car a good strategy would be to leave these stand-alone systems in the car and simple telling the autonomous system of their presents. This means the autonomous system does not have to waste computing time dealing with applying the breaks too much so the wheels will block, the ABS will mitigate this.

2.6.3 Breaking Distance

When developing an automated breaking system for a car, it is crucial to know all relevant variables when it comes to the cars breaking distance, like types of tyres and the condition of the road. If a normal equation is used for calculating the breaking distance needed for a full stop, it would become error prone if the conditions are not optimal and such assumptions is not safe enough for an autonomous car. Furthermore, a system must as a minimum, have the same reaction time as an average human driver when emergency breaking is needed. However, this variable can be hard to determine since people are individuals and there reaction time varies, depending on the conditions they are in when driving a car. The normal average reaction time for humans is 1,5 seconds and it is this number that is used as a constant in the example below although it can be as low as 0,3 seconds in some cases(25). In Appendix E is the breaking distance at different speeds given under various road conditions and tyres to illustrate that optimal safety distance to other cars can vary.

It is clear that the breaking distance needed for a full stop, rapidly increases as the road conditions gets worse and clearly shows that if an automated system were responsible for breaking, it has to know all relevant variables. An automated system, besides knowing the road conditions, has to know what types of tyres are mounted on the car as it also has a huge difference of the cars breaking capabilities. In Appendix E under the 130 km/h column the breaking distance is 111 m with normal tires under perfect road conditions.

If the human reaction time is subtracted in metres, it will show the effectiveness of an automated breaking system over human interaction as the calculation below will show:

$$130\,km/h \approx 36\,m/s$$

So

$$36\,m/s \cdot 1,5\,s \approx 54\,m$$

It is clear that $54\,m$ travelled caused by a 1,5 second reaction delay at $130\,km/h$ is something that can be drastically improved upon if the reaction delay can be brought down by, for example, a computer system. Their response time lies in the millisecond range depending on the complexity of the system like types of distance sensors and processor speed. So the system would theoretically be able to bring the car to a full stop in $56\,m$, an 49 % improvement.

2.6.4 Passenger Comfort

In an ordinary car the driver has control of the acceleration, this gives the driver control over the situation and gives the possibility to adjust acceleration to what the driver feels comfortable. In the autonomous car the driver does not have the same control over the situation, therefore comfort of the passenger becomes an issue, in form of comfortable acceleration and negative acceleration. According to(26) the comfortable exposure levels where obtained, in Table 2.7, they have been categorised and processed to give an overview of the limits and the legislation requirements for emergency breaking have been added.

Δ 100 km/h	**Comfortable Starting**	**Comfortable Breaking**	**Emergency Breaking**
g-Force	0,11 g - 0,15 g	0,12 g - 0,14 g	0,59 g
Δ Time	19 - 26 s	20 - 24 s	4,79 s
Acceleration m/s^2	1,37 m/s^2	-1,37 m/s^2	-5,8 m/s^2
Breaking Distance m	N/A	286 m	71 m

Table 2.7: Comfortable starting & breaking

As it can be seen in Table 2.7, it should be acceptable to set the acceleration and negative acceleration limits for the autonomous car to 0,14g ≈ 20 seconds for 0 - 100 km/h. In case of emergency breaking the system have to comply with the legislation. The requirements to the breaking capabilities of a car; it should be able to break with an acceleration of at least -5,8 m/s^2 on dry plain level roads with good friction coefficient((27) & (28)).

Based on this requirement the emergency breaking in Table 2.7 has been calculated, the calculation is in Appendix F.1. The resulting breaking distance at 30 km/h are 7 m, and the g-force equals 0,59, this is based on gravitational force of 9,816 m/s^2 (Denmark). If the speed where raised to 100 km/h the resulting breaking distance would be 71 m.

The difference between the calculated values of breaking distances and the distances in Appendix E are a result of an added safety margin for the values. This safety margin is added because the distances is meant as a guideline for the driver of a car, so they can keep the right distance to the car in front. Where as the calculated values are requirements for the cars breaking system.

2.7 Challenges Man vs. Machine

Who is superior, human or machine? The old battle between human reasoning and acting, according to changing circumstances, versus the machines reliability to give the same output every time, if it is fed with the same input. This battle have been resurfacing from time to time when new technology is implemented or being tested. This is especially true when the topic is automation; one reason for this is that humans can be reluctant to just give over the control to a machine, the objection raised is that humans can take reasonable decisions when unexpected events occur or the driver gets a premonition about a dangerous situation emerging. Whereas the machine can only do what it is programmed to do, this means that it cannot react to unexpected situations. The thought that some things are done better by machines than by humans is not new, and humans accept it in many places where it improves comfort or safety. It is only when we feel the technology threatens the way we live or our standard of living, that we doubt the technology and decides not to use it.

Embedded controllers are still at a state where they only do what they are programmed to do, by this is meant that they do not learn from experiences and alter their programming according to these experiences. To work around this problem, the development and programming of microcontrollers consider this when they design and implement a system. One way of doing this is by programming the microcontroller in a way that ensures it acts as intended or that it fails to a safe side when the unexpected occur. This is what it is programmed to do in these circumstances and should not be confused with AI(Artificial Intelligences). Although there are research in AI, and simulating the human brain with neural networks, the state of this technology and size of such systems, makes them unrealistic in embedded systems like microcontrollers in cars of today.

Reliability and failure, in case of failure of an ordinary car, the human driver can react and thereby reduce the consequences. Two examples could be the loss of electrical power in the car or the engine stalls unintentionally, in both circumstances the human driver, can use the cars inertia and safely navigate the car in to the side and stop it. If the microcontroller should have the same capabilities it would need to be programmed to have that kind of reaction and for the power failure it would need some sort of backup on the electrical power for the controller, actuators and sensors. The controller, sensors and actuator could be powered by a battery backup, that takes over and delivers power in case of failure in the ordinary system. The main point in this is to get high enough security and safety, therefore autonomous car system would require redundancy in many of its systems.

2.7.1 SWOT Analysis

We made a SWOT-analysis to get a better overview of pros and cons of human vs machine as drivers of cars. Based on this analysis we made a comparison of strengths and weaknesses humans and machines have in 2.7.2-Comparison

Strengths	Weakness
Reliable No lack of concentration Cannot get bored Constant awareness (always running)	Do not learn from mistakes No premonition No Reasoning Cannot adjust for the unforeseen
Opportunities	**Threats**
Always yield same outcome No lack of concentration	The unexpected occur Accountability not decided

Table 2.8: SWOT: Machine

Strengths	Weakness
Premonition The unexpected occur Reasoning Accountability	Semi reliable Do not always yield same output/reaction
Opportunities	**Threats**
Adjust for the unforeseen Can learn from mistakes	Periodic lack of concentration Easily bored No constant awareness Easily distracted

Table 2.9: SWOT: Human

2.7.2 Comparison

Machine

Pro: They are reliable, under the condition they are built for, the microcontroller never needs rest, it can be constant aware and process its input to yield an output. Under the same condition, the output will be the same every time.

Con: They cannot reason, they can only do what they are programmed to do, this means no capability to take decisions when the unexpected occur, the event that the microcontroller's programming does not cover. Accountability and liability falls back on the manufacturer or the user of the system, depending on the maintenance and the agreements between user and manufacturer. No premonition capabilities and difficult to implement.

Human

Pro: A person is able to adapt, to cope with changes in its surrounding and to take decisions both based on what have been learned and by using reasoning. By reasoning is meant the ability to use what have been learned in new ways to adapt to changes. Humans are accountable and liable for their actions. Premonition, making adjustment of speed according to the situation possible.

Con: The lack of constant awareness, humans can easily get bored when the task is monotonous, with the result that the awareness of what we do is slipping, that is like an autopilot without the sensory input to its decision making. Humans takes chances sometimes, which could lead to unintended incidents.

2.8 State of Autonomous Cars

As stated in 1-Introduction trains, aeroplanes and ships rely on some sort of autopilot system, the implementation and use vary depending on application. Even though all of these systems are in use currently, no such system is used in cars around the world. Many entities worldwide have tried to implement automation in the car industry, mainly the bigger car firms but Google have also been developing, and are currently running at least ten autonomous cars(As of 2012)(29). It might seem weird these systems are implemented in many different places; however the usual place for implementation usually has far simpler interaction with the autonomous vehicle and its surroundings than a public road. Keep in mind the subway taking you to work, or the aeroplane flying intercontinental routes does not have to negotiate obstacles or pedestrians, cars do.

The idea of making a fully self-driving car is not new; it has been tried loads of times during the course of the past eight decades. However, it is only in the last decade that things really have been gathering speed, with the price of electronics and microchips decreasing and computational power increasing.

As an example; some new car have an automated system that controls breaking when a dangerous situation is emerging. If the driver is unaware for some reason the automated system would take over and start slowing the car down, to avoid an accident.

The current state of the autonomous cars have two challenges one the mechanical part where all the manufacturing and the designing of the systems take place, and two the legislation, where all the laws are agreed upon.

The recipe for designing and making driverless cars is non-existent because no company have been able to manufacture a fully functioning driverless car. However almost all of the driverless cars in development today do have few parts in common; they share a LIDAR system, used to sense objects around the car and a satellite navigation system. Other systems are also employed like computer vision and RADAR. Computer vision looks for obstacles in front of the car as it is driving along. Stationary sensors sometimes, uses RADAR for sensing of object on the cars sides(30). Moreover, for precise travelled distance measurements a tachometer is fixed directly to a wheel of the car.

Another thing to consider when evaluating the autonomous technology is the potential of interference and unintended access, like software and hardware hacking. When more and more electronics and microcontrollers are implemented in cars worldwide, the potential of the systems being hacked rises. In current developments, of driverless cars, this risk is minute because of the small number of vehicles on the road. However if the technology do become available to the public the risk of the system being hacked rises, this has to be taken into account since normal cars with a driver is not as susceptible to this problem.

Legislation, for the self-driving car is a tough nut to crack because where we before were able to make humans responsible for an accident, with the self-driving car who are we going to blame? In the United States, four of the fifty states have passed laws allowing driverless cars on public roads. As far as accidents go Google themselves, say, their cars have only been impeded in two accidents during the course of 300.000 km, they also say the two accidents occurred while the car was driven manually.

2.9 Conclusion of Problem Analysis

Within the 2-Problem Analysis we have tackled the Autonomous car and by extension, our initiating problem, we have investigated it from the technology that is needed to produce it, the environment within it will be acting, to the effect it will have upon society. With all this research done we can now decide how we should continue.

In the Sections 2.2-Finding the Problem and 2.3-Demand Analysis we found that; the initial problem should be processed further, that the autonomous car could help reduce injuries due to accident and that there could be a demand for an autonomous car. This means that we have found there is a need for the solution we have chosen to work with.

In the Sections 2.4-Stakeholders and 2.7-Challenges Man vs. Machine we found that the implementation of the autonomous car would require some changes in legislation and a solution to the issues about who is liable under different circumstances. This of course means that the solution we are working on is not something we can bring to the roads right now, what we are doing can however, be utilised to help show that it is possible; how it is possible and maybe even help with laying the foundation for defining the needed changes in legislation. Finally we learned from the Intel survey(1) that there is a public interest in the project, if it can help bring down traffic jams and queues due to congestion.

In Section 2.5-Technologies we investigated possible speed sensors, distance sensors, motor controls and actuators that are available technology which can be used to build the sub-systems for the autonomous car that we are interested in. In Section 2.8-State of Autonomous Cars we analysed the autonomous car technology and where it is today. We found that there already is research into autonomous cars and there are prototypes on the roads for testing, furthermore we learned that the companies uses new cars that they modify. This way they can focus on the automation part of the autonomous car, during their research. For us this means, that we have several large corporations already investigating the technology, but as none have a marketable product on the market, we will all be in the process of testing different technologies and investigating their viability within an autonomous car.

Finally we investigated the factors that will be relevant for the system implementation, this was done in Section 2.6-Knowledge, this gives us an idea as to the ruggedness the system must possess, the kind of objects the sensors will need to be able to identify and finally, in case of emergency breaking, a baseline for the distance needed as well as a rough idea of the kind of improvement an autonomous car should have on this. Furthermore it helps us to choose more wisely which components, sensors and requirements our system has to fulfil, it also assists us in analysing the situations our system has to be able to cope with.

This leads to the conclusion that from a technological point of view the autonomous car can be built, one challenge would be to give the system reasoning skills and precognition, that are comparable to a human drivers skill and to get the systems to interact and act in a reliable fashion(2.7-Challenges Man vs. Machine). The implementation within society also require changes to legislation and a solution to the liability issues. We also have a good idea of the factors that will be relevant towards the systems that is mounted on the car, which of course will also effect the specifications. All this summed up means large scale implementation is years out in the future; we can however conclude that our problem has relevance for society, and that our project might help at least show that it is both possible and maybe even viable way to handle personnel transportation in the future.
With the previous chapter and conclusion in mind, we envision the requirements as follows:

1: We assume a newer car as it is build today is the framework for the systems(thus the mechanics and standard security systems of today are implemented)

2: Sensors: These needs to be able to perceive a wide variety of objects, and one sensor will not be enough, at least a camera sensor, LIDAR and RADAR sensor will be needed to cover all circumstances. These will be providing input to the microcontroller about the surroundings of the car.

3: GPS and a detailed navigation system, i.e. display, and detailed map. To ensure route planning and knowledge of the road is available for the microcontroller. This also need connections to camera vision for comparison.

4: Speed sensors for the input and feedback to the microcontroller, about the car's speed, and the motor's speed, as well as direction. The Hall sensor will be best applied here due to ruggedness and reliability.

5: An actuator, for the steering of the front wheel on the car. There have to be a feedback to the microcontroller about, the angle of the wheels, so it knows how much it is turning and in what direction or if the car is driving straight. The feedback are needed to give the microcontroller the ability to adjust its output to the actuator in accordance with the feedback and other sensory input.

6: An actuator for activating and releasing of the breaks on the car. It needs to provide feedback to the microcontroller.

7: A motor control for managing the starting and stopping of the motor and to change the speed and direction. The implementation of this system will vary a lot depending on the car-framework being worked upon.

8: A microcontroller to handle all the sensory input, process it and give output that correspond to the input and the decision the microcontroller is taking based on its programming.

9: All the systems mentioned above must also be analysed to ensure that they are rugged and reliable in the environment that the car demands that they work within 2.6.2-Car Environment.

10: Must also have implemented fail-safe systems in case of; complete system failure, or failure of selected sub-systems in different combinations.

Chapter 3

Problem Delimitation

The autonomous car might be part of the transportation infrastructure of tomorrow, but as we have seen, there are quite a few challenges that has to be overcome before this will be the method used to drive vehicles on the road. Our project, with the time allocated, can unfortunately not contain all the things needed in order to create a fully functional autonomous car, we will have to focus our efforts on certain parts of the challenge.

3.1 Time Frame

We have a good idea of how much is needed to create a fully functional system in order to make a car able to manoeuvre around on its own, but we have not gone into detail on every needed part, as just analysing these would be a complete report of its own. This means we have to be realistic with our time and so we must limit our scope in order to get a manageable project and one that will be interesting within the scope of the autonomous car.

The project was started the 1st of February 2014 and has to be concluded on the 21st of May 2014. Furthermore the official time allocation for the project is 405 hours per student. This means that we have to utilise our time efficiently so that the limited time is spent wisely. This will be done by limiting the project to a part where we think we can both be able to learn something and furthermore show it is viable to create a system for self-driving cars.

3.2 Conclusion of Delimitation

With the aforementioned time constraints in mind, we will have to limit the scope of the project. What we will do, is create a proof-of-concept for a self-driving vehicle. Referring back to our initiating problem, we feel confident we could combine those systems onto the frame of a toy car.

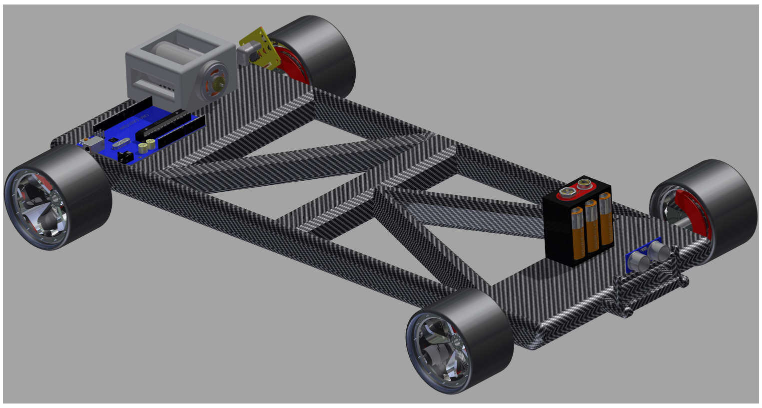

Figure 3.1: Illustration of our vehicle

Figure 3.1 shows a rough sketch of the proof-of-concept we envision, it will be built on a toy car. The "brain" of the car will be an Arduino Uno, a microcontroller development board, which will be interacting with a number of shields[1] which is add-on electronic boards which can be connected to the Arduino, they have a specific function ranging across a wide area.

In our product we will need a distance sensor connected to the Arduino; we will be using an ultrasonic sensor for the prototype as it is fairly simple and should have a relatively impressive sensory range. We will also acquire a shield with motor control functions so that we can control the speed of the motor in the toy car, it will also be able to interface with servos for steering that is present on the toy car. We would like to get a Hall Effect sensor which we can use for measuring rotation on the electric motor used for propulsion so that we can measure the speed of the car as it moves around. We will also write a program which combines all the functionality of the sensors and allows the vehicle to move around.

Finally we need to set some minimum requirements to our proof-of-concept so that we can evaluate whether or not our project was a success when we are done with creating the solution.

- Must be able to detect obstacles
- Must be able to change direction if obstacle is detected
- Must be able to control motors
- Must be able to reverse
- Must be able to turn left
- Must be able to turn right
- Must be able to move forward
- Must be able to measure speed

[1]Premade boards which fit on top of an Arduino.

This is quite reduced compared to the full system implementation, we do however feel, that it is an interesting challenge, and one that can help show whether or not this is a viable solution for a self-driving vehicle.

Chapter 4

Solution

Following the previous chapters we have now come to the solution part where we will create a proof-of-concept vehicle that should show our solution to the problem, and get us closer to the objective of answering our initial question as to how the challenges can be handled smartly.

The solution we will be working on, is based on the specifications in Section 3.2-Conclusion of Delimitation we will assemble our chosen hardware on the frame of a toy car. For the solution the following equipment was used:

- Toy car (With servo and drive motor)
- Arduino Uno (rev. 3)
- TinkerKit Hall Effect sensor
- Ultrasonic sensor (HC-SR04)
- Arduino Motor shield (rev. 3)
- Battery pack and assorted wires
- Computer with Arduino software installed

After this we will be creating a piece of software that enables the Arduino to handle the motor and steering based on input from the sensors. We also plan on doing some tests in the electronics laboratory both to test signals that the ADC read as well as visualizing our PWM signal. Furthermore, as the motors on the toy car are of unknown specification and product number we will need to do some testing to figure out how much and how little power we can feed them.

To start off the process we have created a Use-Case, this will help us have a clear goal as to what our proof-of-concept need to be able to achieve in order to be considered a success. Furthermore we have made use of UML(**U**nified **M**odelling **L**anguage) and flow charts to present an easy overview of our code.

4.1 Use-Case

In order to be able to validate that our solution functions as we want, we have created the following Use-Case it will be used to compare our solution and evaluate whether or not our goals has been reached.

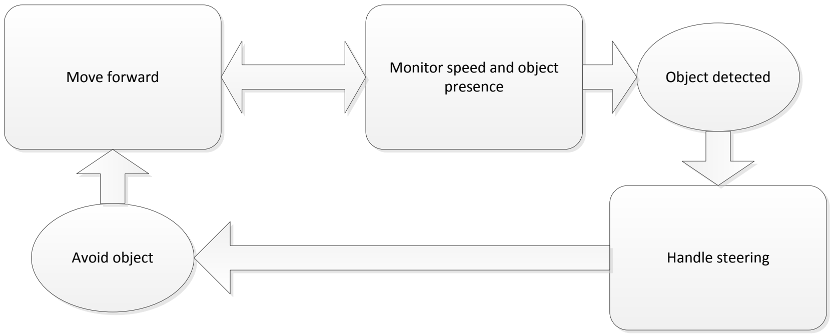

Figure 4.1: Use-Case for program

In Figure 4.1 above, is the Use-Case we created to illustrate what we want our solution to fulfil. The toy car should be able to move forward and detect objects in its way, avoid the object by steering past it. When the path is clear of objects the steering would turn the wheels back to a neutral position.

4.2 Arduino Uno

The choice of microcontroller was based on the criteria that we needed something that is relatively easy to work with, well documented and did not require expensive hardware in order to develop with. Due to these factors the Arduino Uno board was an easy choice. The board is based on the Atmel microcontroller ATmega328(31), furthermore it has a USB port which makes for easy interfacing with today's computers.

The microcontroller should also provide us with all the features that we will need to implement the proof-of-concept, it has 14 I/O pins, of which 6 can be used for PWM(See Section 4.2.1), it has 10-*bit* ADCs(See Section 4.2.2) and can operate across a wide range of voltages.

Finally it is important to note that the microcontroller cannot supply current above 40 mA, so it cannot be used to power anything that needs more than this amount.

Programming of the Arduino board will be where the primary challenge will lie. The manufacturer of the Arduino made a program where you can write code and program the Arduino, they have made their own language for the Arduino programming, which is based on C/C++ but with some modifications(32). This "language" is however merely a set of pre-defined C/C++ functions which are called.

It is also possible to program the board without use of the standard Arduino IDE program and instead use specific AVR compilers, such as: AVR C, AVR-GCC(for C++) and other compilers(32).

4.2.1 Pulse Width Modulation

PWM is a technique to change a digital signal(voltage) fully **off** or fully **on** to a voltage that will result in an effective voltage between 0 and V_{cc}. By doing this it is possible to change the speed of a DC motor, so that the motor runs between zero rpm and maximum rpm according to the voltages it is supplied with.

A $5\,V$ DC motor will run at its maximum speed when it is supplied with $5\,V$, if the supply are done using PWM the average voltage to the DC motor, or the load can be given all average values between $0\,V$ and $5\,V$.

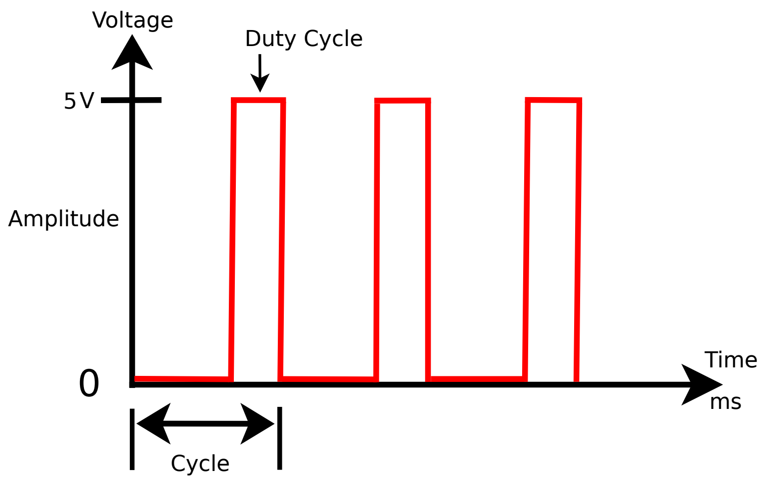

Figure 4.2: PWM diagram

Figure 4.2, shows an example of PWM, here the amplitude is set to $5\,V$, and the term cycle or period refer to frequency, this means if the frequency is $50\,Hz$ then there are 50 periods or cycles each second. The value of the duty cycle is an expression for how many percent of the cycle time the pulse are high((33) & (34)).

The duty cycle in Figure 4.2 shows that the duty cycle are 1/3 of the cycle length , this translate to a duty cycle of 33 %, this number is obtained by looking at the cycle or period and determine how large a fraction of this period the duty cycle is occupying.

To obtain the average voltage to the load in one period, the average amplitude for that period is calculated. In the example from Figure 4.2. The amplitude are $5\,V$, 33 % of the time and $0\,V$ for the rest of the period that gives an average value of $5V \cdot 33\,\% \approx 1,66V$. In analogue technique this are equivalent to feeding the load with a continues voltage of $1{,}66\,V$, whereas in digital technique using PWM the load is fed $5\,V$, 33 % of the time and $0\,V$ the rest of the time, giving an average value of $1{,}66\,V$. A duty cycle of 0 % equals no feed to the load(turned off) and a duty cycle of 100 % equals feed to the load as being on all the time in the cycle.

By using PWM there are some things to consider, the load can have some requirements that set limits on frequency range, and the time of the pulse length. If the load is a DC motor there will be a requirement of a minimum duty cycle size to overcome the initial inertia of the motor and to counter the effect of "Back-EMF"(Back **E**lectro**m**otive **F**orce). Below this minimum requirements it is best to shut off the motor, the reason is that from a duty cycle between 0 and the minimum required duty cycle, the motor only heats up but it will not start turning. From the minimum duty cycle size for turning the motor to 100 % duty cycle the motors rpm will be corresponding to the fed average voltage((35) & (36) & (37)).

The PWM technique will be used in the proof-of-concept, because the technique yields a good solution to controlling the speed of electric motors, by using it on both the propulsion and servo motors we can easily alter the speed of the motors and thereby both the speed of the car and the turning of our front wheels.

4.2.2 Analogue to Digital Converter

The main function of an ADC is to convert the analogue input signal, represented as a voltage, to a digital output. The analogue input in this setup comes from various sensors within the system and they all communicate with the Arduino Uno that reacts accordingly. The microcontroller on the Arduino Uno, the ATmega328 has a build-in ADC which it can measure analogue inputs(38). The sensor shields in the system, outputs analogue DC voltages according to various readings from their surroundings and they are all processed by the ADC. These analogue signals are translated into a series of binary values representing the analogue readings so it can be read by the microcontroller and thereby controls the correct running sequence by the running code.

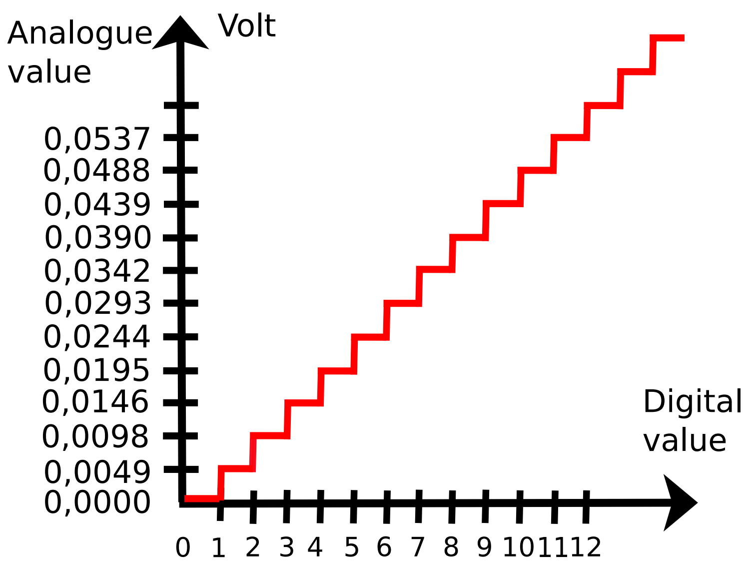

Figure 4.3: Resolution steps

In Figure 4.3 it is visualised how the transformation from analogue inputs to digital values are handled by the ADC, it shows only the first 12 steps, but the Arduinos 10-*bit* ADC has 1024 steps covering digital values from 0 to 1023. From an initial analogue value with corresponding digital value, the digital value will increase or decrease with 1 for each change in the analogue value of 0,0049 V depending on the sign (+ or −) of the change in analogue value, within the limits of 0 - 1023. The value 0,0049 V is a rounded off value, the calculation for the figure and table can be seen in Appendix F.4-ADC Step-size and Resolution.

Decimal Number	Voltage
0	$0.0 \geq V$
1 - 1022	$0.0 < V < 5.0$
1023	$5.0 \leq V$

Table 4.1: Binary voltage representation from the ADC

The ADC has a 10-*bit* resolution that enables it to convert a 0 V to V_{cc} max into a binary representation between the numbers 0-1023. It takes eight analogue inputs through an 8-channel multiplexer, meaning that the ADC can handle eight sensors sequentially(38).

4.2.3 Motor Control Unit

We choose the Arduino Motor Shield(rev. 3) option since it is the easiest way to get the desired results. The motor shield is based on the double H-bridge L298. The L298 also incorporates voltage measurements across the H-bridge itself, enabling somewhat accurate speed approximations, by feedback to the controller. The L298 chip itself is limited to 2 A per channel and 4 A in total across both bridges at 12 V(39).

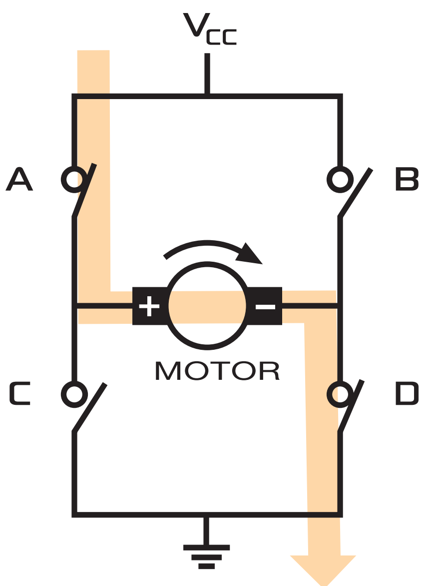

Figure 4.4: H-bridge(40)

As seen from the Figure 4.4 the current is flowing from A to D through a motor turning in a clockwise direction. Moreover, the motor would spin in an anticlockwise direction if the current were flowing from B to C through the motor. This is used to dictate the direction and the speed of the drive however breaking of the motor is a possibility using the H-bridge, if C and D was set and A and B not set both poles of the motor would be grounded thereby removing the potential generated from the motor in turn increasing the resistance of the motor.

4.2.4 Distance Sensing

The HC-SR04 ultrasonic sensor, is what we are going to use for our proof-of-concept vehicle. The sensor needs 5 V to operate properly according to the data sheet and it has a maximum consumption of 15 mA. It has four pins; one for power, trigger, echo and ground so it needs two I/O pins from the Arduino Uno. According to the components data sheet it can sense distances between 3-400 cm and can send out an ultrasonic burst of 8 pulses at 40 kHz(41).
The HC-SR04 is going to be mounted on the front of the proof-of-concept vehicle.

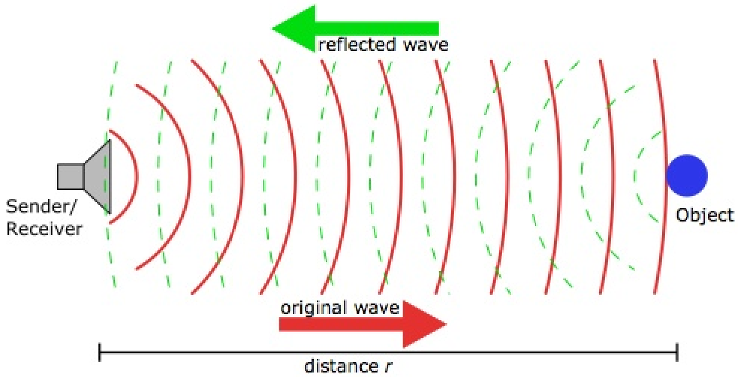

Figure 4.5: Sound wave propagation(42)

Figure 4.5 shows how ultrasonic sensors function, it emits ultrasonic sound waves from its transceiver, and then waits for the echo. Since we know the speed of sound in air, we can calculate how far the ultrasonic sound has travelled based on the time difference between transmission and reception of the echo.

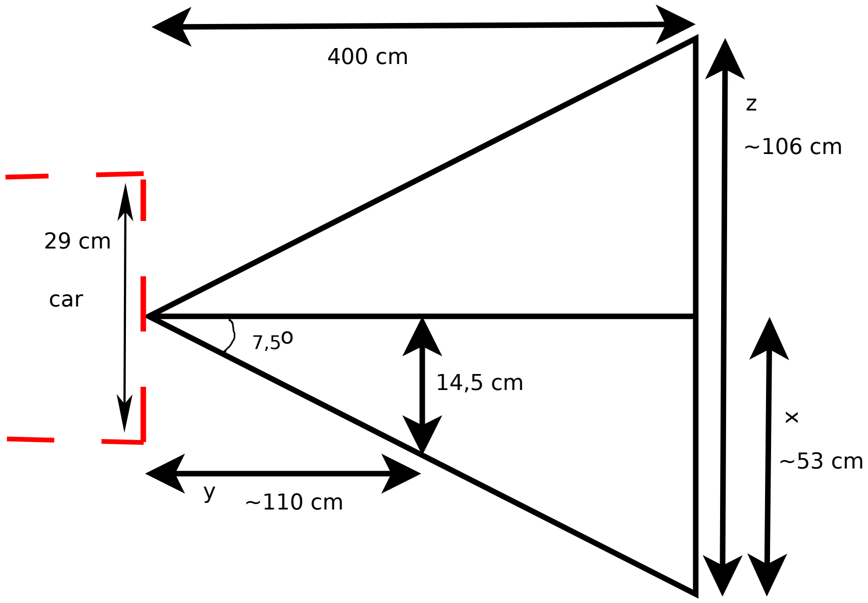

Figure 4.6: Ultrasonic Sensor blind spot

The view angel, which is 15° according to the data sheet, of the HC-SR04 distance sensor are illustrated on Figure 4.6, the figure shows the area from the sensor to 400 *cm*, it can be seen that in the area from the sensor and out to 110 *cm* there is a large blind spot, due to the car's width of 29 *cm*, where objects will not be detected. The blind spot could be made smaller by having the sensor on a servo that turned the sensor from side to side, or by having multiple sensors, in this solution there are only one sensor and it is centre mount on the front of the car. The calculations to Figure 4.6, can be found in Appendix F.2-Blind Spot Calculation.

The calculations also made it clear that, sound waves as a measuring tool for long distances at high speed, have a disadvantages, because the time to get an echo is to long. At 80 km/h where the breaking distance is 184 m. the echo time is 1 s. This calculation however do not take the cars travelled distance in that time into account.

4.2.5 Hall Effect Sensor

The ThinkerKit hall module has a sensitivity of 1.3mV/Gauss and changes output voltage from 0 to 5 volts accordingly. The ADC of the Arduino interprets this as a value from 0 to 1023(10-*bits*). The module itself has three pins one for power input, one for ground and one for the output signal to the ADC. TinkerKit supplys a library with the module, it incorporates only one function for the Hall Effect sensor.

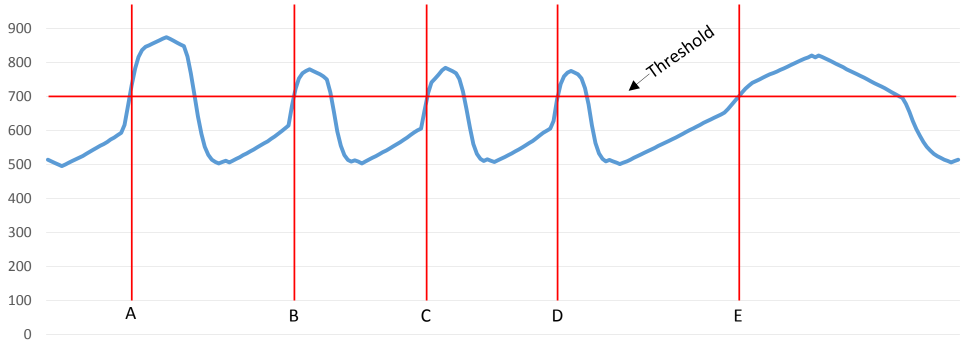

Figure 4.7: Data plot from ADC reading that shows the output signal

Looking at Figure 4.7 above the output(See Section 4.3.6-Speed Test) from the Hall Effect Sensor is displayed by the blue line in the coordinate-system. To calculate the distance travelled over time and there by the velocity one of the ways of achieving this would be to measure the amount of time between the same point of two different waves, as illustrated by the letter A through E lines and the "Threshold" line. Two magnets were placed on the wheel on opposite sides, which means the physical distance between the two magnets is the wheel circumference divided by two. Knowing the amount of time used to travel a known distance one can calculate the velocity. D is the distance travelled, in this case half the circumference, t is the amount of time used to travel the distance.

$$\frac{D}{t} = v$$

Another way of calculating the velocity is to measure the time the magnet is above the threshold the challenge here is the difficulty in getting a precise length of the magnet, because it is only the length of magnet where the flux density is higher than the set threshold. The good thing about this method in particular is the speed. Since this method only uses one magnet the maximum amount of time it has to wait to get a reading is half a revolution, where the other method mentioned above in worst case has to wait one full revolution if it just missed the magnet.

4.3 Experimentation Documentation

As a part of our solution, laboratory testing was essential so that the proper documentation could be acquired for the different items used. Several things were tested and some of them were already on the toy car like the DC motor and the servomotor including its potentiometer. Other items were acquired, an Arduino Uno, Arduino motor shield, Tinkerkit Hall Effect sensor and a HC-SR04 ultrasonic sensor and in total all of these things is the hardware part of the proof-of-concept solution on the autonomous car. For testing a variable power supply, multimeter, oscilloscope and a laptop were used.

4.3.1 The DC Motor

The toy car used as a part of the solution came with a pre-installed DC motor, but no information could be found on the motor plate so several experiments were conducted. From about 6 V the DC motor began rotating and could barely move the toy car so this was concluded to be the minimum amount of power needed for any reaction. When 9 V was applied, the toy car moved at an acceptable acceleration and the variable power supply was set to deliver up to 2 A.

4.3.2 The Servo Motor

Pre-installed on the toy car was a servomotor for steering purposes that had an attached potentiometer for position measurements. Like on the DC motor no information or specification could be found on the motor plate or the steering housing at all so a series of tests were conducted. The servo motor was also applied 9 V for acceptable movement of the front wheels since 6 V were not enough for making the front wheels turn at standstill, only when the toy car was in motion 6 V could barely do the job, so this was determined to also be 9 V.

4.3.3 The Potentiometer

The sole purpose of the potentiometer is to be able to measure the direction and the position of the front wheels to a degree to that it would be possible to control the toy car. This is done by the gear housing pushing the centre pin left or right on the potentiometer thereby altering its resistance.

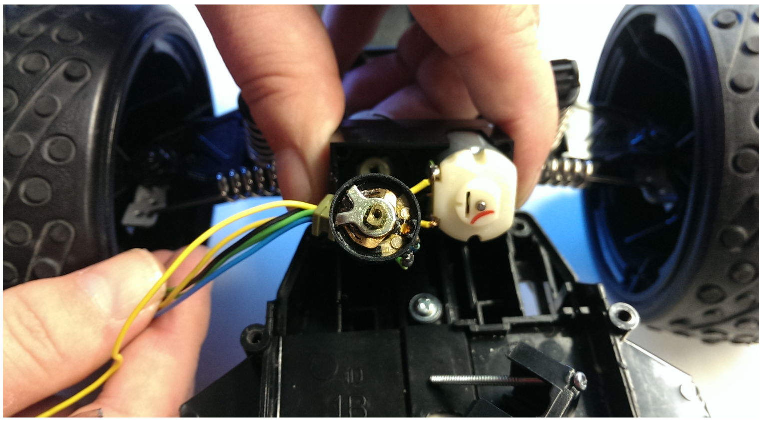

Figure 4.8: Toy car potentiometer and servo motor

In Figure 4.8 above the potentiometer is seen to the left where three pins are connected, two green wires, one on each side and a blue wire in the middle and it functions like any other potentiometer where the two green wires are the minimum and maximum values and the blue wire is relative to them.

Position	Resistance
Right Turn Max	3.33 $K\Omega$
Centre	2.36 $K\Omega$
Left Turn Max	0.88 $K\Omega$

Table 4.2: Potentiometer resistance values

Table 4.2 shows the resistance values of the potentiometer hooked up with the servo motor. This is where the ADC in the ATMega328 will measure the position of the wheels based on a voltage drop across the resistance of the potentiometer. The ADC then outputs a value between 0 – 1023, depending on the position of the potentiometer, to the microcontroller which then sends it to the Arduino serial monitor software on a computer for a visual inspection on the changes. The values can be seen in Figure 4.11 and these values are essential for the software that has to run the toy car autonomously since the confirmation on that the wheels are in the correct position is vital. Normally when the code is running the output values from the potentiometer to the ADC are being directly fed to the program that then bases its running sequences on this and does not send it to the Arduinos serial monitor.

4.3.4 Combined Motor Test

After separate testing on both motors, six AA 1.5 V batteries were installed in series for a combined voltage of 9 V to test if the theory applied for the setup. A simple self-written motor steering program was uploaded to the Arduino Uno and with the two motors connected to the Arduino motor shield the first test was set in motion. The toy car's movement was steered by entering one or more of six possible values, "Q,W,E,A,S,D" into the serial monitor console and they each representing a direction in which the car had to move. It soon became clear that when both motors were given an instruction to move, a shortage of power was apparent and worsened under the start-up of the main DC motor.

4.3.5 PWM Test

To control the DC motor and the servo motor a PWM signal is used because it is a perfect way to control a speed by applying the correct voltage. In the self-written test software the percentage of the PWM signal and the duration was set to test at which Duty-Cycle the different motors reacted, this was done by connecting a variable voltage supply and an oscilloscope. A PWM signal at 90 % or more with 9 V was found to be needed for a sufficient acceleration and the signal strength and duration was checked by the attached oscilloscope. See Figure 4.8 for the PWM output were it is clear that a PWM signal is used for powering the motors.

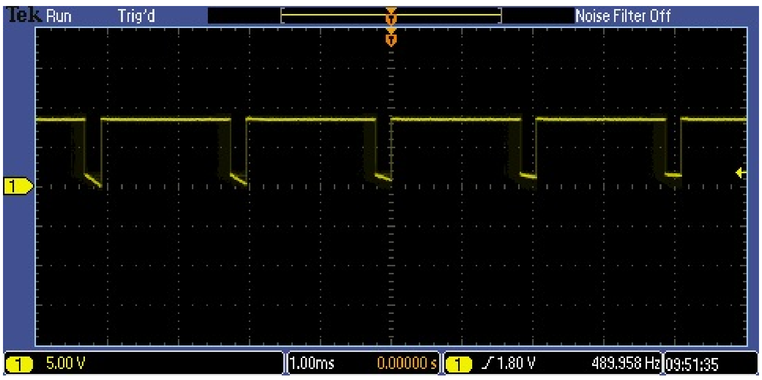

Figure 4.9: Screen shot from oscilloscope of 90 % PWM signal at 9V send to the DC motor

4.3.6 Speed Test

The Tinkerkit Hall Effect sensor is used for the speed test and has been installed on the back of the toy car. The two neodymium magnets that came with the snesor has been fitted to the rear left wheel rim, these will be used for the speed measurements, by counting the numbers of revolution in a given time-frame. As seen in Figure 4.10 the Hall Effect sensor had to be mounted in a close proximity to the magnet before any significant output could be distinguished. Between 0.1 and 0.8 *cm* the signal readout were strong enough to be useful.

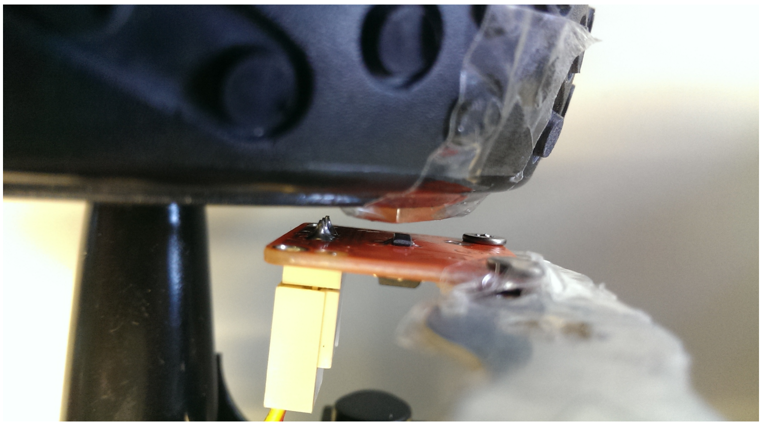

Figure 4.10: Picture of the Hall Effect sensor as the magnet passes by on the rotating wheel

A readable digital signal output was the main goal of the test since the finished code would rely on these values for calculating the speed. The self-written test code were able to use the Arduino Uno's ATmega328 ADC to create an integer value from the passing magnets and output it to the attached laptop in a serial monitor console and the result can be seen in Figure 4.11.

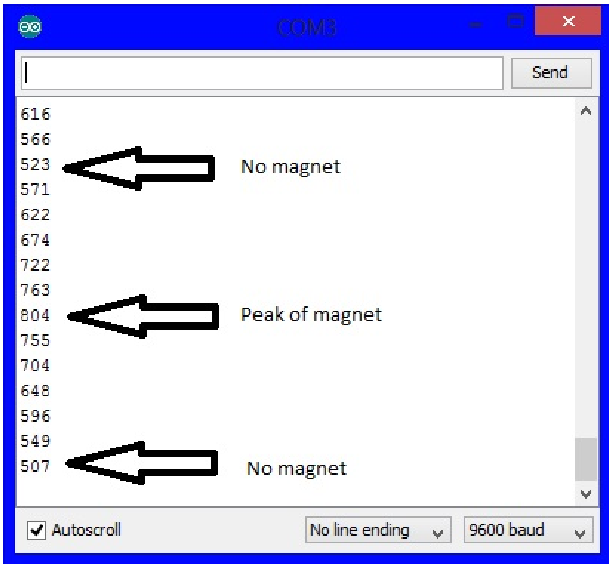

Figure 4.11: Data output from ADC*

**The program used to read values from the ADC and output them on to the console was written* *based on us being interrested in the peak values, so it ignored values which was within +/-5 steps, that is why you only see one value between the peaks. in reallity there would be a significant amount of time between peaks.*

The output values from the Hall Effect sensor were plotted with data from the serial monitor console to get a visual inspection of the signal over a known distance. The result can be seen in Figure 4.12 below, were it becomes clear that the wheel is out of balance but the signal is still useful for its purpose.

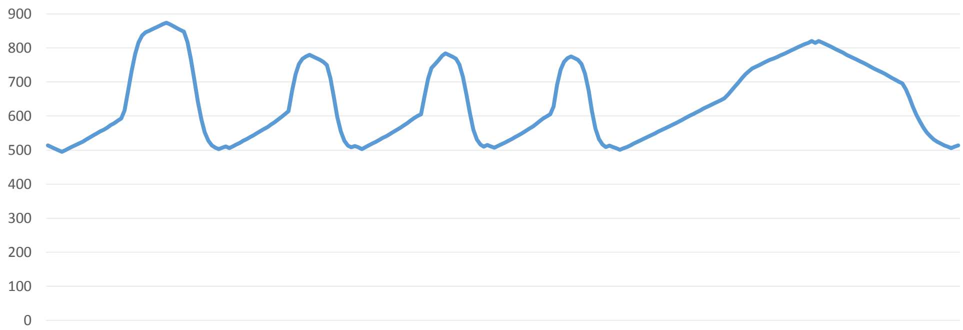

Figure 4.12: Data plot from ADC reading that shows the output signal

4.3.7 Distance Test

A series of reliability test were conducted on the HC-SR04 ultrasonic sensor on range precision and the influence from different types of materials that could disturb the accuracy of the ultrasonic echo. The first test checked if the specified range interval was reliable (from 3-400 cm) and this was proven correct on a vertical hard surface. When the proximity of 3 cm and below were tested, the accuracy dropped with an error percent of over 3000 % when one or both sensors were obscured. This shows that the test car must avoid coming closer than 3 cm to any object, otherwise the system will assume that the path is clear for up to 3000 cm and based on that information, begin a forward acceleration with an eminent accident to follow. Later versions of the program mitigated this problem.

The second series of test were based on the ultrasonic sensors abilities to get accurate readings when facing different types of objects made from various materials. The result when testing the different types of materials and the angle of them in relation to the ultrasonic sensor are showed in Table 4.3 below where 1 is best and 5 is very poor.

Material/Angle	90 °	45 °
Book	1	3
Clothe	2	3
Blackboard Sponge	4	5
Wall	1	3

Table 4.3: Distance sensor test result from materials at different angels

4.3.8 Test Incident

During one of the final testing days in the laboratory, a malfunction occurred on the Arduino Uno board. The problem occurred during a test run of sample code, which after a few runs and some minor code changes the Arduino Uno entered an infinite program loop. After a soft reset of the program, it did not run the installed program as usual, it simply did not respond. Several re-flashing of the Arduino Uno was attempted, but the same communication error kept appearing and switching of computers, cables, drivers and software did not help the problem. An overheating might have occurred so it was left to cool off for a while and in the meantime, a troubleshooting session was conduced. A possible answer was it being stuck in some internal process; this was tested by erasing and reinstalling the communication chip(ATmega16U) however this did not help.

Another possibility was that too much voltage or amps might have been delivered to Arduino Uno through the motor shields power connection. This may have damaged some of the circuits, chips or internal fuses on the Arduino Uno. In the datasheet of the motor shield, it was found that if an external power supply was applied directly to the board, a jumper could be cut to break the power line between the motor control and the Arduino Uno. This recommendation is valid if the Arduino Uno had to be supplied through the motor shield, which it did in this case, but it is unclear if this had anything to do with the problem, and if it were responsible, why did it first occur after several weeks of testing? But none of the online solutions did solve the error and other methods was all in vain as we did not get the Arduino Uno board to function again and was deemed useless for the rest of the project.

We found out that a fellow classmate had a spare Arduino Uno that could be borrowed for the remaining project period. This kindness got us back on track and enabled the last test sessions to be completed without further delays. The cause of the problem were never discovered and it did not appear again, leaving only theories and speculations behind(43).

We should stress that we changed the Arduino Uno to a RedBoard. This is the same microcontroller, the ATmega328, as the Arduino because of this equality we will continue to refer to the board as an Arduino Uno.

4.3.9 Complete System Test

The final test session was conducted with all the different components attached to the toy car and the final system code installed on the microcontroller. A variable power supply was connected to the motor control shield, since powering the whole system with batteries will take up too much space and be impractical(two packs of six AA batteries in parallel). Different current limit settings were tried on the variable power supply while the system was at maximum load and it was clear that a recommended setting should be on $9\,V$ at $1{,}7\,A$. The maximum set load of the system is during start-up of both motors while both the distance and Hall Effect sensors are active.

As the power issue was resolved, the second part of the test session was to confirm that the code See Appendix H would run as predicted and it could fulfil the scenario from Figure 4.1. The proof-of-concept vehicle was proven to function accordingly to the requirements, it was able to avoid objects in its path and change direction if it came to close.

Two issues did appear during the test session and the first was that the proof-of-concept vehicle had a limited operating radius due to its wired power connection that is stationary. The second thing was that the distance sensor only could register objects in front of it, making it blind when reversing. For example, if it came in close proximity to an object and it had to reverse for clearing enough space to turn around it. During this, the vehicle occasionally bumped into other objects; this is of course an early state proof-of-concept vehicle so issues like these is expected.

4.4 Programming

The Arduino and all the hardware connected to it is not very useful without some code running on the Arduino itself. In this section, we will be looking at the techniques used to write the code and run through the source-code's most important functions. A run through of how the programming of the ATmega328 chip works will also be conducted.

4.4.1 Arduino IDE

The Arduino IDE is a graphical front-end for programming the Arduino, it is a Java program and requires Java runtime to work, that makes it cross platform usable, i.e. it works on Mac, Windows and Linux. The IDE uses AVR-GCC and related tools for compiling and debugging. The actual programming is made in a modified C/C++ language, but it can handle both C and C++. The IDE is also used for transferring the program to the Arduino after compilation and to communicate with the Arduino if one wants to have readouts from the microcontroller.

The ATmega328's architecture does not support multiple threads, which means it is not possible to run functions in parallel. The **`main`** function in normal C/C++ programming is called **`loop`** in the Arduino language the difference is the **`loop`** function is called continually after end execution. This sequential execution manner poses problems when it comes to speed and efficiency, because the program could get stuck in a time consuming function without the ability to do something else. These time consuming functions could for instance be; velocity measurements with the Hall Effect sensor where the function has to wait for the passing of two magnets. With multithreading the programmer would have the ability to run the velocity function in parallel with other function thereby have the ability to do something else while the velocity functions was executing.

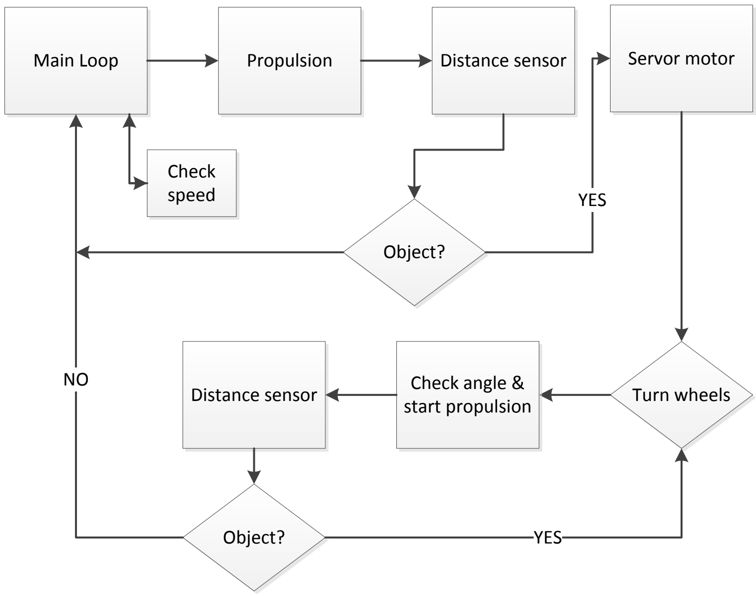

Figure 4.13: Flow chart of program

In the above Figure 4.13 it is clear the flow we envision for our program, it will contain a main loop where the program flow is handled. The default flow will be "check for object, drive forward if no object". When this breaks, the servos will be activated for steering, this part of the program will need to be able to call the propulsion code as well, so that it can drive forward while checking for object and ensuring that after the path is clear, the front wheels will be straight again, thus returning to the default flow.

UML Diagram

UML is a way to create a graphical overview for a program you are designing. There are a couple of different types of UML diagrams, we have used one of them already and that is the Use-case diagram, and the next one we are going to use is UML class diagram, as we are not implementing classes, we have used it to get an overview of the functions and variables contained within our program.

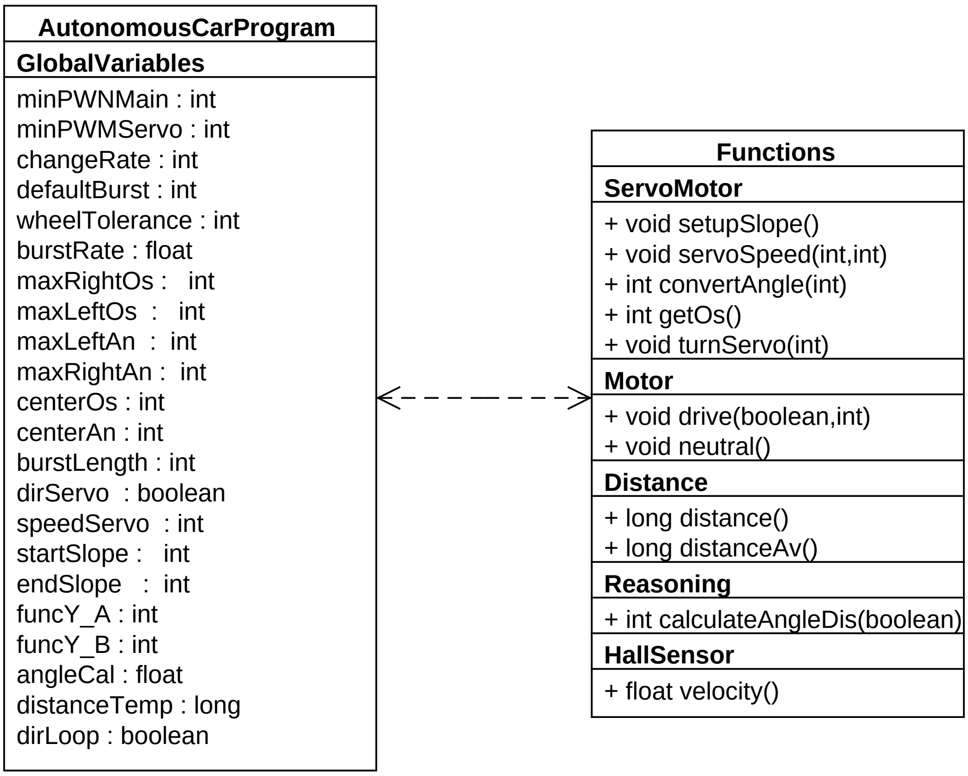

Figure 4.14: UML diagram of program

Figure 4.14 is a graphical overview of our program, we have gathered the global variables on the left side for easy referencing and to the right is the ServoMotor functions, the Motor functions, the Distance functions, HallSensor and then the Reasoning functions. The diagram was created as a side project while we were working on the code in order to provide a clear overview of what variables and functions our program implements.

4.4.2 Source-Code Explanation

Reasoning Description

`int calculateAngleDis(boolean right)`(See Appendix H, Line 212-233)

The function is used to calculate the angle of the wheels when coming closer to an object, and then determine to whether to make a sharper left or right turn. The Boolean **`right`** argument is used to determine if it has to turn to the right or left, the first **`if`** statement determines what angle the wheel has to be at while turning to the right and when getting close to an object. It also has an **`if`** and **`else if`** statement that take care of the overflow that can occur when the distance becoming too short and causing the angle to become negative. This function is invoked when needed to see how much the servo has to turn to the right or to the left.

Motor Description

To be able to control the toy cars DC motor for propulsion only two functions have to be used; one for setting the direction and another for controlling the amount of voltage applied.

`void drive(boolean directionMotor, int speedMotor)`(See Appendix H, Line 160)
The motor part of the Arduino code is for controlling the toy cars propulsion DC motor referred to as drive in the code. When the **`drive()`** function is called, it first disables the "brakes" (Software determined by grounding both motor pins) so that the motor can be activated.

`boolean directionMotor;`(See Appendix H, Line 162)
The **`directionMotor`** variable is a Boolean and as the name imply it is responsible for setting the right direction of the toy car by entering one of two states, **`HIGH`**=forward or **`LOW`**=reverse, this is done through the **`digitalWrite`** function.

`int speedMotor;`(See Appendix H, Line 163)
The **`speedMotor`** variable is an integer and is called right after the motors direction is set. This is done through the **`analogWrite`** function that applies the necessary voltage by PWM.

`void neutral()`(See Appendix H, Line 166-169)
When the motor has to reach full stop the **`neutral()`** function is called that first cut the voltage flow to the motor and thereafter applying the brakes.

Distance Description

`long distance()`(See Appendix H, Line 190-195)
The **`distance()`** part is responsible for keeping track of incoming objects in the toy cars path and returning a distance in centimetres. These functions are called and controlled by the **`loop`** function and is therefore only active when needed.

This is done by applying a voltage to the distance sensors **`triggerPin`** for two microseconds, which results in a series of sound burst being emitted by the sensor. Thereafter the distance sensors **`echoPin`** is set to read the incoming signal and calculate the time for the sound wave to travel from the car to the nearest object and back. This information is processed by a simple mathematical formula for sound travelling at normal atmospheric pressure resulting in a distance.

ServoMotor Description

Controlling the servo on our toy car is quite challenging, as it needs to both handle output for a motor, but furthermore it is also necessary to be able to measure the relative location of the motor, and if necessary also activate the motor again for adjustments to ensure that the right angle has been reached with the wheels. Due to this complexity our servo handling is done in several separate functions within the program.

The function furthermore implements a series of variables, these are used to contain different information needed to both ensure that the servo will not try to turn further in any direction that the steering allows as well as parse information easily between the functions.

`void setupSlope()`(See Appendix H, Line 55-62)
This function is used to setup the initial conditions for the later servo functions. This way has been chosen so that some global variables can easily be altered while the rest of the program will simply adjust depending on those values. The function is invoked only once, during the setup sequence of the program.

The mathematical idea of the approach is that as we get nearer an object, within a margin that we setup in the **`endSlope`** and **`startSlope`** variables, we will turn the wheels sharper, we have chosen to use 45 steps, but this can also be altered easily. This is done in the **`maxLeftAn`** and **`maxRightAn`** variables. We created two graphs with this information with distance plotted vs. angle. The first calculations yields a rate of change for the graphs, this will be the same for both graphs, this will be the same for both graphs, but with opposite signs.

Next we need to calculate the y-intercept values for the linear functions, this is done in line 59 and 61 of Appendix H. The reason we use two functions is because we want one for right and one for left turns, as their slopes will be equal, with opposite signs, we only need to calculate different b-values.

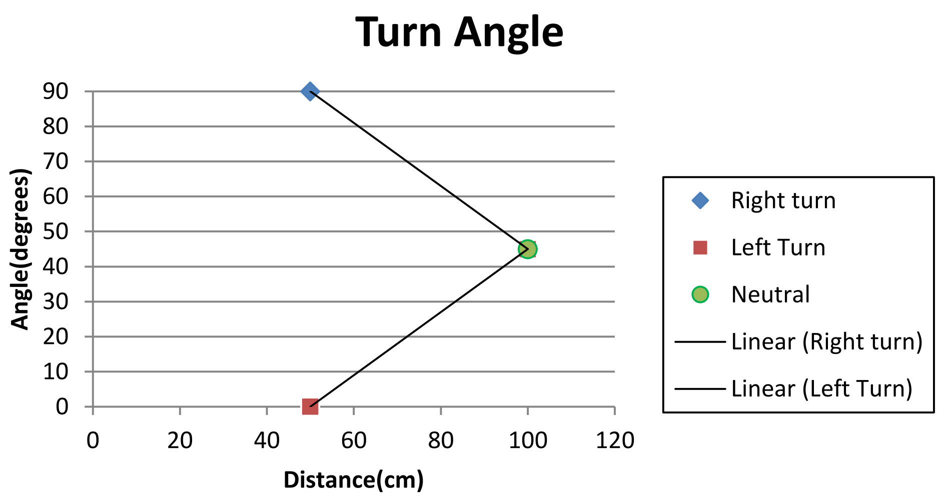

Figure 4.15: Turn angle

Figure 4.15 shows our plot of the two equations (turn left and turn right) plotted vs. the desired "angle". It is worth noting that "45 °" is what our program interprets as being straight.

`void servoSpeed(int minimum, int change)` (See Appendix H, Line 65-86)
This function is invoked in order to calculate the needed PWM to send to the servo,

this will be higher if the servo needs to change a large angle, this will cause the servo to turn quicker. This is done to a minimum of 30 *ms* PWM burst to the servo, if the resulting PWM value becomes greater than the max value of 255, the function will increase the time the burst is send to ensure the wheels get to the right position. The change parameter is used to calculate the PWM value.

The minimum parameter is used to ensure that the PWM will have a desired minimum value as electric motors will have a minimum required duty cycle(See Section 4.2.1-Pulse Width Modulation).

`int convertAngle(int Angle)`(See Appendix H, Line 88-99)
This function is used to calculate the angle of the wheels, furthermore it also ensures that the program will not ask the microcontroller to turn the wheels further than the steering hardware allows. It is worth noting that it is not equivalent to an actual angles, it is 45 steps to either side. This function returns the ADC value that will approximate the position.

`Int getOs()`(See Appendix H, Line 101-110)
This function is used to measure upon the potentiometer attached to the servo, this is done using the ADC, it is done ten times in order to reduce the effect from small fluctuations and an average is calculated from this. The 20 millisecond delay is done in order to allow the system to stabilise before measuring, in case the servo has been moving in recent time.

`void turnServo(int Angle)`(See Appendix H, Line 112-134)
This function is the one that the main loop is invoking, and need only one parameter in order to do its job, the Angle parameter, this will be used to turn the wheels in the desired angle. This function invokes the earlier mentioned functions in succession to ensure that all necessary variables are set to the correct values and then handles the output to the servo motor.

HallSensor Description

`float velocity()`(See Appendix H, Line 244-273)
This function called when we need to know the speed of which we travelling. It uses two **`while(1)`** loops and two **`if`** statements. The first **`if`** statement compares a **`hs.read()`** to a threshold value based on test experience as seen in Figure 4.7, the **`hs.read()`** uses the ADC to read a value from the hall sensor to find out if the magnetic field is above the threshold. Then we set the timer and make a delay, just a short while to get away from the first magnet.

The next **`while(1)`** is where the calculations of the speed is done by measuring the time between passing magnets, this could be a dangerous thing if it moves very slow, since it would cause the program to halt while waiting.

4.5 Conclusion of Solution

The main conclusion for the solution is that the system code worked as planned. The use of an ultrasonic sensor for distance measuring can function in small scale in a controlled environment. The use of a Hall Effect sensor for speed measuring is a precise way and if more neodymium magnets were used as measuring points the speed-readings would have been quicker. The decision of measuring the speed of the proof-of-concept vehicle along with the other functions resulted in an unexpected delay in the system code.

The sensible power consumption for the whole proof-of-concept vehicle was found to be $1{,}7\,A$ at $9\,V$ under maximum load, this was based on hardware limitation. Furthermore, the idea of powering the system from batteries was discarded due to impractical issues so a variable power supply is used.

Chapter 5

Discussion

The solution we ended up with is a small-scale system, which helped us figure out how an autonomous car could be implemented on a full-scale.

Feature	Achieved
Must be able to detect objects	✓
Must be able to change direction	✓
Must be able to control motors	✓
Must be able go forward/backwards	✓
Must be able to turn left/right	✓
Must be able to measure speed	✓

Table 5.1: Showing the goals from the 3.2-Conclusion of Delimitation

Looking at Table 5.1 we can see that we actually achieved all our goals from our 3.2-Conclusion of Delimitation, this was done with sensor packs, motor shield and our Arduino Uno Board.

When we had to program the Arduino Uno, we used the Arduino IDE to program the board. The IDE is missing a clear marked line number, for easier error correcting. When looking past this flaw in the IDE, it functioned acceptably and communicated well with the Arduino Uno board, and that was really what was required of it.

We had to be able to control the motor on the toy car, this was done by using an Arduino motor shield. The motor shield actually functioned acceptably, and was easy to use for what we needed it to do. We used it to feed power to our proof-of-concept vehicle, the pin layout was clearly marked, all in all it was a very satisfying choice of motor shield for our solution.

It is possible to get sensors, which are better suited to an autonomous car than the ultrasonic sensor. The LIDAR sensor works with light and the RADAR sensor works with radio waves, both technologies work much faster than ultrasonic sensors thereby making them better suited for this application. Another technology is a photo optical sensor, which also could be fast, but the drawback is that it is more processor intensive than LIDAR or RADAR. (Pros and Cons See 2.5.1-Distance Sensing Technologies). The choice of only installing one sensor results in blind spots, this would be the case with all sensor types, so the solution to that are to implement multiple sensors. This approach raises another challenge because the sensors could interfere with each other if they are of the same type, this can be overcome but it is something to be aware of. In a fully integrated system the best solution would be to have multiple sensors of different kinds because they supplement each other, and it makes the system more robust through redundancy.

The Arduino Motor Shield(43), recommends supplying the shield by external voltage source, not through the Arduino. The shield can supply the Arduino with power but if the voltage supplied to the shield exceeds 9 V, it is recommended to cut the V_{in} connection on the backside of the shield, so there are no connection between the Arduino and the shields voltage source. It is then necessary to supply the Arduino with a dedicated power source. This recommendation is to ensure that the power through the Arduino stays within the safe operating range of the Arduino. Since all equipment is borrowed, it is not an option to cut connections on the motor shield. Because of that, it is necessary to stay within the limit of 9 V to the motor shield, it should then be safe to supply the Arduino through the shield.

During the testing and implementation of the solution, some issues was discovered with the initial setup concerning the power usage and the use of batteries. Initially it was planned to use six standard AA batteries to reach 9 V. During testing it were discovered that the power consumption was greater than what could be supplied by the batteries for extended periods. Because the batteries could not deliver enough current. To get this issue resolved, it was discussed; to use more batteries, to use a combination of batteries in series and parallel or to use other kinds of batteries. Changing the battery setup would increase the weight of the prototype, because of this and the late state in the project it was decided to use a power supply instead. Thus making sure that the power supplied is within the limits of the hardware. The choice of using a power supply, have the drawback that the prototype have to be connected with wires to the power supply, but it makes it possible to have full control of volt and amperes because the power supply have an adjustable maximum setting of these.

We had to be able to turn left and right so we could avoid an object, this was done by using the servomotor. The servo was a bigger time consumer during solution implementation than expected, because of its inconsistencies. The inconsistencies was cause by a very unstable servo assembly. We used a lot of time getting the measurements, and decided to make a margin +/− the targeted value when turning the wheel, so the wheels would not fluctuate from the right to the left all the time looking for a precise value. Because we have this instability in the servo assembly, we need to have this +/− margin of the targeted value when turning the wheels. However, this would not be a viable option to have on a car, since it needs to be as precise as possible. We did not buy a servomotor for the proof-of-concept vehicle, we used the integrated one in the toy car. It was lacking a rating plate with information concerning what it needed to run, so we had to conduct some test to find the important information.

For further development of the prototype, it will be necessary to address the issue with the power because the car does not have the required free movement if it has to be connected to a power supply outside the car.

When looking at a car it needs to be able to move forward and backwards for this, we used the integrated motor on our toy car to propel the proof-of-concept vehicle along. Although it worked fine, there were just a slight problem with it. The motor was lacking a rating plate, since this was not present we had no chance of knowing how many volts it needed to run sufficiently. We then proceed to testing this and then found the minimum requirements was 6 V and recommended was 9 V.

When a car is driven it is essential to know the speed, and for this we used a TinkerKit Hall Effect sensor, this sensor would give us an output that we used to calculate the speed(See 4.2.5-Hall Effect Sensor for information on calculations). The TinkerKit hall sensor was delivered with two neodymium magnets for use, this was not optimal since we could be caught waiting for another magnet to pass the Hall Effect sensor. We used the sensor to show how it could be possible to measure speed, but on a full-scale you would use the system mentioned in 2.5.2-Speed Sensing Technologies.

A flaw our program at this stage contains is that our initial reason for having speed measurements, to make the distance reading more accurate while moving at faster speeds, is not implemented. There are two reasons for this; first of all, due to our accident with the Arduino we lost important trial and implementation time in the laboratory. Secondly, as both our distance sensor and the speed sensor are very time consuming we are uncertain if it would have been a good idea to implement our distance sensor in a way that made it dependent upon our speed sensor.

What we did achieve was a function that can use the Hall Effect sensor to accurately read the speed of the vehicle and we know what we would need in order to make the speed reading fast. Mounting a Hall Effect Sensor directly to a flywheel in the motor or differential thus ensuring the time between two referencing points would be reduced.

One of the main goals when implementing an autonomous car, is insuring that it can avoid objects. We chose an ultrasonic sensor, as we concluded in the 3.2-Conclusion of Delimitation this would be easy to work with and fulfil the point of detecting objects. We chose the HC-SR04 as the ultrasonic sensor for our proof-of-concept vehicle. Several test were conducted of this product and saw that while it could detect objects on a distance of 400 cm, but when objects got closer than 3 cm, it would give an unreliable reading to the nearest object. You cannot have an error reading like this on a full-scale autonomous car, as it would cause it to crash if this were the case. We also had an issue with the wave propagation delay of the ultrasonic sensor, since the echo time would be to large. It would not work that well outside, because of rain and wind as stated in 2.5.1-Distance Sensing Technologies.

One thing that throughout the report has been identified as a key danger with our proof-of-concept is the danger of ending up in a process that takes a very long time and by extension keeps the controlling microcontroller unaware of potentially changing or even dangerous circumstances. This was for example obvious in our speed sensing function where if the car is travelling fairly slowly we can get stuck waiting for the magnets to pass and this results in the car being unable to monitor the distance sensor.

The problem was also seen in the distance sensing program where waiting for the sound waves to propagate out to an object and back again could in a full-scale system become a long time where no other parts of the program can act. We accepted this inherent flaw in our system as it was just a prototype but in an implementation this challenge will have to be handled.

The fix is obvious but also complicates the system further, parallel computing, in one form or the other the ability to execute crucial tasks simultaneously is required. There are several ways to achieve this, we discussed the possibility of implementing dedicated microprocessors to each sensor, which would then be monitored constantly and then relay the latest measurements to a central processor upon request. This would mean sensors which takes a relatively long time to yield a result, would not hold up the run sequence.

Another possibility would be acquiring processing units with several cores implemented in the hardware so that the program could be split in to threads that is executed simultaneously, this would require some careful planning and timing of the program to ensure that the program functions quicker and tasks does not get neither missed nor interrupted at the wrong moments.

Most likely the problem would best be solved with a combination of both ideas as we see it, but unfortunately we did not have time to investigate further what this direction of thinking could have done for our solution.

Chapter 6

Conclusion

The main conclusion of this report is that the autonomous car can be built by the technology available today. In the sense of creating the system in a “smart” way, this is doable by choosing the right components for the job. Several distance sensors are required for creating a complete picture of the near surroundings to the autonomous car and of different types. As stated later in this chapter an ultrasonic sensor is not a viable option.

In the 3.2-Conclusion of Delimitation a set of requirements for the proof-of-concept solutions were listed, and it was possible to fulfil them all. Furthermore, the Use-Case in 4.1-Use-Case also set up a minimum requirement from which the toy car should act within and this was also obtained.

Like stated in 5-Discussion, parallel computing must be available in the control system since several critical measurements are taking place at the same time. Furthermore, the use of an ultrasonic sensor for distance measurements to incoming objects in the autonomous car’s path is not a viable solution, It is simply too slow and unreliable for this purpose. A RADAR, LIDAR or camera image recognition system, or a combination there off, is recommended for a reliable system similar to what Google has developed(44) as stated in 2.8-State of Autonomous Cars.

In regard of steering an autonomous car, the use of a servo motor or an actuator is a perfect way of controlling it around objects as proven in 4.3-Experimentation Documentation. The only drawback is that a faster way of measuring the wheels position has to be found, since the servo takes too long to get to its designated destination.

When it comes to a DC motor control in a fully electrical car the control of this should not be a hurdle and it has been proven that the use of a PWM signal is a reliable way of controlling the throttle.

The use of a Hall Effect sensor for measuring the speed is still believed to be the optimal option over other components although more reference points, i.e. gear teeth on a flywheel, are needed for a fast and reliable system like what is implemented in cars already.

Chapter 7

Perspective

The technology required for the autonomous car is gradually being implemented, but the fully automated car is still somewhat a thing of the future. The use of technologies like lane assistance and anti-collision are already in use in some cars. But the fully autonomous car where all subsystems required are built in and integrated to be a full automated system are not available to the consumer yet.

7.1 The Next Step

As we are now more enlightened about the general principal of autonomous cars and knowing that our choices was not the best for an implementation. It is therefore definitely worth considering what the next step would be if we were to continue our investigation of the autonomous car and how the sub-systems that we have focused on can be done smart.

As we learned, the ultrasonic sensor is seriously flawed in some key areas that makes it exceedingly impractical for solving the problem that we needed it to, object distance detection. It was good for our proof-of-concept but when moving on to larger vehicles, longer distances and faster speeds it is unsuitable.

In continued work the main focus should be getting a better choice of sensor, and we are of the firm believe, that more than one would be necessary. LIDAR and RADAR are both great choices for sensors, as they are using the electromagnetic spectrum these waves move *a lot* faster than sound waves. As such they detect at larger distances at a fraction of the time, thus ensuring that we do not get a large lag while waiting for distance measurements to different objects in an ever road changing environment. It would also be necessary to implement more than one sensor, so that it would be possible to not only measure a distance to an object but also a relative location.

Another hurdle that will have to be defeated as well is that while knowing the distance to an object you do not want to hit is crucial, we have so far not dealt with any kind of identification of objects. Identifying objects will however also be necessary as it will have a severe impact on the way that you want to react, furthermore it could also allow the car to react better to things such as roadworks, bicycles, pedestrians, etc. and even read the different signal such things would use as well.

Finally, our proof-of-concept was focused on avoiding objects and controlling the motor, this will of course not make an autonomous car very useable, so implementing some navigation system would also have to be done. Using the GPS system already in wide use would of course be an optimal choice for this as the system is already well tested, so it would only be a matter of allowing the software to communicate with the autonomous cars' system.

All this would of course also change the approach to the program thus it would have to be altered so that it could base intelligent decisions on the input from the better sensors. Some sort of parallel task execution would also be required in order for it to function fast enough so that the car would react fast enough to be a safer choice than driving the car yourself. The technology to solve that is however available so it is simply a matter of choosing the parts and making the programming function.

7.2 Possibilities for Society

The impact and perspective such system would have on society could be denser traffic with less accidents on the road, thereby utilising the infrastructure in a better way than today. The cars would be safer and have intelligent control systems, leading to a decrease in numbers of accidents. The decrease in accidents and especially the drop in severe accidents will have a positive impact on society in the form of less expenses in relation to car accidents.

The autonomous car will have sensors that make the control system aware of weather conditions and its surroundings, so it can navigate and behave according to the circumstances, this will increase safety, because conditions like fog, rain, and black ice etc. can be taken into account when the car drives by itself.

If the car only need to know its location and a destination, to handle the driving by itself without other interaction from the driver, it will be possible to change the interior of the car in a way to increase comfort considerably. That way the transportation can be more relaxing and the time in the car can be used for office work or recreational activities.

Because the former human driver will be a passenger in the fully autonomous car, it could be possible to be transported in these cars without a driver license. Further more making it possible for some disabled people to get increased freedom and mobility, because they can use an autonomous car that drives itself, to free them of the need of a personal assistant or helper to get around in a car. Another possibility is that intoxicated people can be transported by their autonomous car.

By making the experience of transportation more relaxing for the passenger, some of the former drivers bad behaviour, like taking chances and road rage, could be minimised and the traffic on the road will be more pleasant for all, because the former bullies now have the role as passenger.

Another possibility will be to utilise the concept of car-pools better, because the car can drive itself short distances so there are no need for a driver, that are tied up to travel around to pick up the rest of the people for that car-pool. The car can park by itself so the passenger can get of nearby the destination and the car will then go and find a car park, that is not necessarily close by, this way the infrastructure will also be utilised better. The car could schedule and drive itself to service at a car-workshop, in periods where it is known that the owner do not require the car.

Transportation of goods in lorries could change because the rest hour requirements could see a change, since the driver is made a passenger or he could be absent so the lorry would drive itself to its destination. That way the lorry only needs handling by humans at starting point and at arrival to destination. This could lead to savings in the transportation business, due to the decrease in the need of lorry drivers.

Further, in the future it would be possible to implement some form of communication between the autonomous cars, thereby the utilisation of the infrastructure could be even better because the cars can plan ahead based on information from other nearby cars.

The implication and possibilities mentioned above depend on how the population receive the technology, and how legislation, insurance and liability issues ends up being resolved.

References

[1] IntelPR. *The Vote Is In: Citizens Support 'Smart Cities' with Driverless Cars, Public Service Drones and Surroundings that Sense Activities*. [Online]. Available from: http://newsroom.intel.com/community/intel_newsroom/blog/2014/02/10/the-vote-is-in-citizens-support-smart-cities-with-driverless-cars-public-service-dro utm_source=feedburner&utm_medium=feed&utm_campaign=Feed:+IntelNewsroom+%28Intel+Newsroom%29, February 2014. Accessed 13th March 2014.

[2] Wade Roush. *Intel Survey: Americans Skeptical, Yet Hopeful, About Smart Cities*. [Online]. Available from: http://www.xconomy.com/san-francisco/2014/02/19/intel-survey-americans-skeptical-yet-hopeful-about-smart-cities/, February 2014. Accessed 13th March 2014.

[3] WIRED.com. *The Ethics of Saving Lives With Autonomous Cars Are Far Murkier Than You Think*. [Online]. Available from: http://www.wired.com/opinion/2013/07/the-surprising-ethics-of-robot-cars/, July 2013. Accessed 04th March 2014.

[4] Mercedes-Benz. *Autonomous long distance drive research vehicle S500 intelligent drive*. [Online]. Available from: http://www5.mercedes-benz.com/en/innovation/autonomous-long-distance-drive-research-vehicle-s-500-intelligent-drive/. Accessed 04th March 2014.

[5] Forbes. *Nissan's Autonomous Car: A Test Drive*. [Online]. Available from: http://www.forbes.com/sites/michaelkanellos/2013/10/09/nissans-autonomous-car-a-test-drive/, September 2013. Accessed 04th March 2014.

[6] Tom Simonite. *Toyota Unveils an Autonomous Car, but Says It'll Keep Drivers in Control*. [Online]. Available from: http://www.technologyreview.com/news/509616/toyota-unveils-an-autonomous-car-but-says-itll-keep-drivers-in-control/, January 2013. Accessed 04th March 2014.

[7] Jeffrey K. Gurney. *Sue My Car Not Me: Products Liability and Accidents Involving Autonomous Vehicles*. [Online]. Available from: http://works.bepress.com/jeffrey_gurney/1/, 2014. Accessed 3rd April 2014.

[8] Eric. *Infrared vs. Ultrasonic - What You Should Know*. [Online]. Available from: http://www.societyofrobots.com/member_tutorials/book/export/html/71. Accessed 03rd March 2014.

[9] sfu.ca. *SOUND PROPAGATION*. [Online]. Available from: http://www.sfu.ca/sonic-studio/handbook/Sound_Propagation.html. Accessed 13th May 2014.

[10] naic.edu. *How does the Arecibo 430 MHz radar make measurements in the ionosphere?* [Online]. Available from: http://www.naic.edu/~isradar/is/aboutis/radar.html. Accessed 03rd March 2014.

[11] Bureau of Meteorology Australia. *How Radar Works*. [Online]. Available from: http://www.bom.gov.au/australia/radar/about/what_is_radar.shtml, 2014. Accessed 10th March 2014.

[12] fas.org. *MTCR Handbook*. [Online]. Available from: http://www.fas.org/nuke/control/mtcr/text/mtcr_handbook_item17.pdf. Accessed 10th March 2014.

[13] Robert A. Fowler. *LIDAR vs RADAR? Some Insight*. [Online]. Available from: http://www.lasermap.com/laserM/en/doc05.htm, September 2000. Accessed 10th March 2014.

[14] Alibaba.com. *Picture of Hall Effect sensor reading a gear tooth*. [Online]. Available from:http://de1008556171.trustpass.alibaba.com/productshowimg/126850556-103461279/Hall_Effect_Gear_Tooth_Sensor_CYGTS104U_for_large_gear_modulus_.html. Accessed 25th February 2014.

[15] INTEGRATED Engineering Software. *Picture of the magnetic field on a gear with tooth*. [Online]. Available from: http://www.integratedsoft.com/Applications/Sensors-VR, 2014. Accessed 25th February 2014.

[16] sesoronix.com. *Picture of the output signal form Hall Effect sensor*. [Online]. Available from: http://www.sensoronix.com/00img/p01.gif, 2014. Accessed 25th February 2014.

[17] Resistor Guide. *Photo Resistor in General*. [Online]. Available from: http://www.resistorguide.com/photoresistor/, 2014. Accessed 27th February 2014.

[18] Poul Erik Petersen. *Elektroteknik 3 Elektriske Maskiner*. Bogfondens forlag, A/S, 4. ed., 2 pr. edition, 2006. Chapter 7: Hastighedsregulering af motorer.

[19] Poul Erik Petersen. *Elektroteknik 3 Elektriske Maskiner*. Bogfondens forlag, A/S, 4. ed., 2 pr. edition, 2006. Chapter 5: Stepmotorer. SR-motorer.

[20] Dendritics. *Comparison of DC Brushed Motors vs. AC Induction Motors for Electric Vehicle Drive*. [Online]. Available from: http://heaa.info/wp-content/uploads/library/DC_vs_AC_motors.pdf, August 2010. Accessed 4th April 2014.

[21] Niels Frees. *Elbilen – simpel men alligevel kompliceret*. [Online]. Available from: http://www.ligeudadlandevejen.nu/elbilen-%E2%80%93-simpel-men-alligevel-kompliceret. Accessed 18th May 2014.

[22] Karim Nice. *How Car Steering Works*. [Online]. Available from: http://auto.howstuffworks.com/steering.htm. Accessed 04th March 2014.

[23] Tom Harris. *How GMs Hy-wire Works*. [Online]. Available from: http://auto.howstuffworks.com/hy-wire3.htm. Accessed 04th March 2014.

[24] Car and Driver. *Are We Losing Touch? A Comprehensive Comparison Test of Electric and Hydraulic Steering Assist* . [Online]. Available from: http://www.caranddriver.com/features/electric-vs-hydraulic-steering-a-comprehensive-comparison-test-feature, December 2012. Accessed 04th March 2014.

[25] Marc Green. *Driver Reaction Time*. [Online]. Available from: http://www.visualexpert.com/Resources/reactiontime.html, 2014. Accessed 27th March 2014.

[26] l. l. Hoberock. *A Survey of Longitudinal Acceleration Comfort Studies in Grund Transportation Vehicles* [Page 7, Page 15]. [Online]. Available from: https://repositories.lib.utexas.edu/bitstream/handle/2152/20856/cats_rr_40.pdf?sequence=2, July 1976. Accessed 13th March 2014.

[27] Trafik Styrelsen. *Vejledning om syn af køretøjer* [Section 5, Page 36, Subsection 5.03.021]. [Online]. Available from: http://www.trafikstyrelsen.dk/~/media/Dokumenter/01%20Syn%20og%20koeretoejer/03%20Regler%20om%20koeretoejer/01%20Regler%20om%20koeretoejer/Vejledning_om_syn_af_koeretoejer_g%C3%A6ldende.ashx, January 2014. Accessed 13th March 2014.

[28] Trafik Styrelsen. *Vejledning om syn af køretøjer* [Section 5, Page 25, Subsection 5.03.021]. [Online]. Available from: http://www.trafikstyrelsen.dk/~/media/Dokumenter/01%20Syn%20og%20koeretoejer/03%20Regler%20om%20koeretoejer/01%20Regler%20om%20koeretoejer/Vejledning_om_syn_af_koeretoejer_g%C3%A6ldende.ashx, January 2014. Accessed 13th March 2014.

[29] WIRED.com. *Exclusive: Google Expands Its Autonomous Fleet With Hybrid Lexus RX450h*. [Online]. Available from: http://www.wired.com/autopia/2012/04/google-autonomous-lexus-rx450h/, March 2014. Accessed 03rd March 2014.

[30] AUTONOMOS LABS. *MadeInGermany*. [Online]. Available from: http://www.autonomos.inf.fu-berlin.de/made-in-germany. Accessed 03rd March 2014.

[31] Arduino. *Arduino Uno*. [Online]. Available from: http://arduino.cc/en/Main/arduinoBoardUno. Accessed 27th March 2014.

[32] Andrew Chakley. *The Absolute Beginner's Guide to Arduino*. [Online]. Available from: http://www.rfid-specialisten.dk/default.asp, January 2013. Accessed 27th March 2014.

[33] National Instruments Corporation. *What is a Pulse Width Modulation (PWM) Signal and What is it Used For?* [Online]. Available from: http://digital.ni.com/public.nsf/allkb/294E67623752656686256DB800508989, August 2003. Accessed 26th March 2014.

[34] newbiehack.com. *Introduction to PWM for the AVR (Atmel) Microcontrollers*. [Online]. Available from: https://www.newbiehack.com/MicrocontrollerIntroToPWM.aspx. Accessed 26th March 2014.

[35] Joliet Technologies. *DC DRIVE FUNDAMENTALS*. [Online]. Available from: http://www.joliettech.com/dc_drive_fundamentals.htm. Accessed 26th March 2014.

[36] NMB. *Selection of Brush DC Motors - By Rated Load / Rotation Speed*. [Online]. Available from: http://www.nmbtc.com/brush-dc-motors/engineering/select-load.html. Accessed 26th March 2014.

[37] Northwestern University. *General Theory on Motors*. [Online]. Available from: http://mechatronics.mech.northwestern.edu/design_ref/actuators/motor_theory.html. Accessed 26th March 2014.

[38] Atmel. *ATmega48PA/88PA/168PA/328P Datasheet*[Page 250]. [Online]. Available from: http://www.atmel.com/Images/doc8161.pdf, October 2009. Accessed 26th March 2014.

[39] STMicroelectronics. *L298 Datasheet*. [Online]. Available from: https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf, 2000. Accessed 30rd March 2014.

[40] ROHA Marketing. *Controlling DC Brush Motors with H-bridge Driver ICs*. [Online]. Available from: http://www.rohm.com/documents/11308/12928/100260.H-BRDG_WP_Jan09.pdf. Accessed 30rd March 2014.

[41] ELEC Freaks. *Ultrasonic Ranging Module HC - SR04*. [Online]. Available from: http://www.electroschematics.com/wp-content/uploads/2013/07/HCSR04-datasheet-version-1.pdf. Accessed 27th March 2014.

[42] sensorwiki.org. *Picture of Wave*. [Online]. Available from: http://www.sensorwiki.org/lib/exe/fetch.php/sensors/ultrasound_echo_ranging.jpg?w=&h=&cache=cache. Accessed 24th April 2014.

[43] Arduino. *Arduino Motor Shield*. [Online]. Available from: http://arduino.cc/en/Main/ArduinoMotorShieldR3. Accessed 8th May 2014.

[44] Chris Urmson. *The latest chapter for the self-driving car: mastering city street driving*. [Online]. Available from: http://googleblog.blogspot.dk/2014/04/the-latest-chapter-for-self-driving-car.html, April 2014. Accessed 11th May 2014.

[45] Dansk Standard. *CE-mærkning*. [Online]. Available from: http://www.ds.dk/da/raadgivning/ce_maerkning. Accessed 10th March 2014.

[46] ShenZhen BEL Technology. *CE Certification*. [Online]. Available from: http://www.belemc.com/en/page/default.asp?pageID=26. Accessed 10th March 2014.

[47] ShenZhen BEL Technology. *EU E / e mark certification*. [Online]. Available from: http://www.belemc.com/en/page/default.asp?pageID=23. Accessed 10th March 2014.

[48] Eurofins China. *e-Mark & E-Mark* . [Online]. Available from: http://product-testing.eurofins.cn/en/certification/e-mark.aspx. Accessed 19th March 2014.

[49] Elite Electronic Engineering. *An Overview of European Union EMC Type Approval Regulations for Motor Vehicles* [Page 5]. [Online]. Available from: http://www.elitetest.com/assets/files/Elite%209-Step%20Guide%20to%20European%

20Vehicle%20Type%20Approval%20E-Marking%20%2812-2013%29.pdf. Accessed 20th March 2014.

[50] Vehicle Certification Agency. *Type Approval for Cars*. [Online]. Available from: http://www.dft.gov.uk/vca/vehicletype/type-approval-for-ca.asp. Accessed 19th May 2014.

[51] European Commission. *End of Life Vehicles* . [Online]. Available from: http://ec.europa.eu/environment/waste/elv_index.htm, March 2014. Accessed 1st April 2014.

[52] Gode Artikler. *Hvad sker der med din bil, når den skrottes?* [Online]. Available from: http://godeartikler.dk/biler/hvad-sker-der-med-din-bil-naar-den-skrottes.html. Accessed 10th March 2014.

[53] Bilpriser.dk. *Skrot øsen*. [Online]. Available from: http://www.bilpriser.dk/guide.do?guideid=15. Accessed 10th March 2014.

[54] Rådet for Sikker Trafik. *Online speed test and breaking distance*. [Online]. Available from: http://bremselaengde.sikkertrafik.dk/bremse-test.html?v=2. Accessed 27th March 2014.

[55] Richard R. Gerhold. *Gravity Zones*. [Online]. Available from: https://dendritics.com/scales/gravity-zones.asp. Accessed 1st April 2014.

Appendix A

Brainstorm 6W

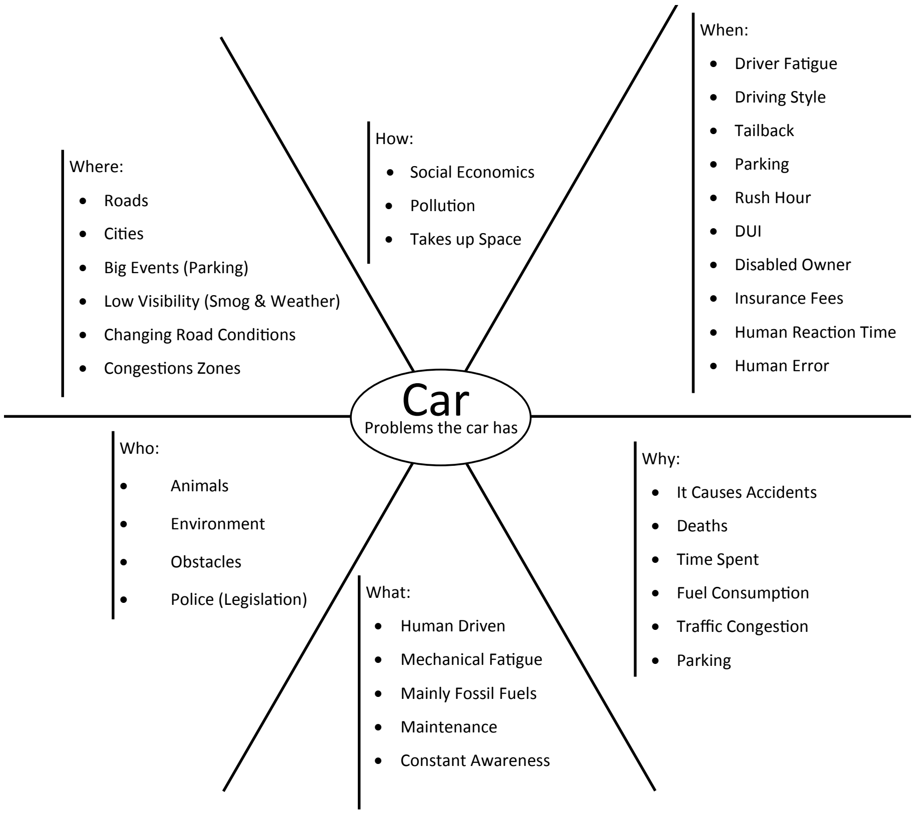

Figure A.1: 6W-diagram

The above 6W-diagram was our first brainstorm session on our initiating problem, our hope was to narrow down what an autonomous could help alleviate, starting with the problems we saw the regular car having. We quickly realised that we were having an extremely extensive 6W-diagram with little focus, so we had to remake our 6W-diagram with a core-problem which was more to the point. The diagram above did help us identify the problem, car accidents.

Appendix B

6W-diagram

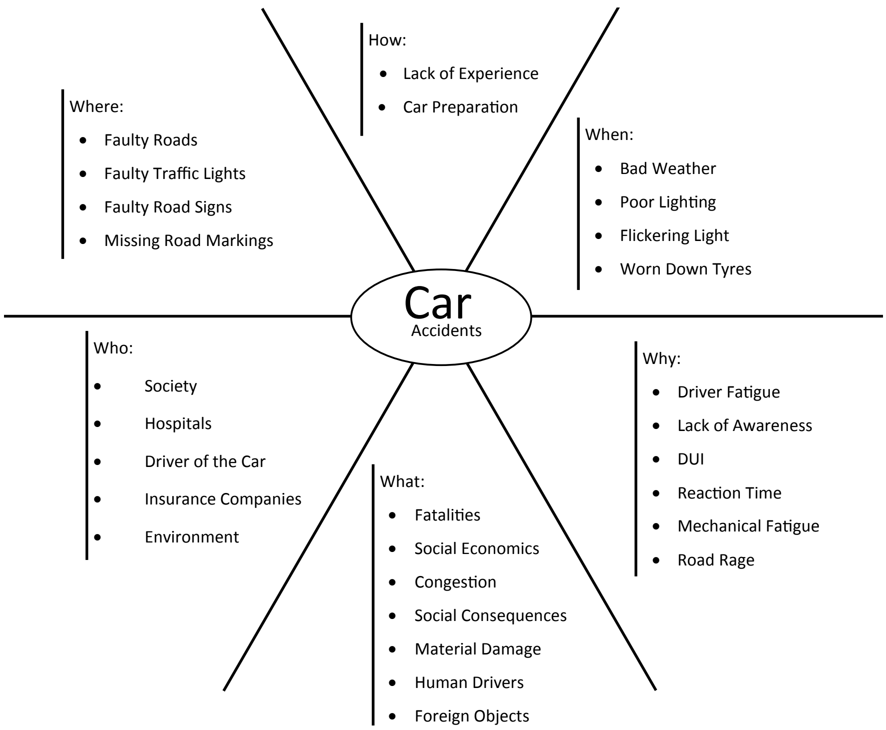

Figure B.1: 6W-diagram

Looking at the "why" part it is clear that most of the things listed is related to the human driver inside the car and taking said driver out of the equation entirely could decrease the number of accidents. For a human driver, driving many hours can be monotones and level of awareness drops and the reaction time increases and an accident is imminent. The same goes for driver fatigue; driving long hours can be tiring causing people to lose focus.

Not all the accidents happening on the road is due to human error some also boils down to bad maintenance of the vehicle, for example, worn breaks increases breaking distance this is a problem in cases where an emergency stop is needed to stop in time.

A lot of things is listed in the 6W-diagram we have chosen to look deeper into the why part since this the things we can improve upon. However, that is the case, many of the things listed is a result of the human driver take for example poor lighting an autonomous car would have no problem negotiating badly lit roads since its sensors would not need light to function. The same goes with a person's lack of experience some people are not interested in driving and because of this have no intention of becoming good drivers, a driver-less car would not have this problem since it is programmed and not taught.

Appendix C

Statistics

Tables acquired from Danmarks Statistik

Tilskadekomne og dræbte i færdselsuheld efter uheldsart, transportmiddel, personskade og tid

	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011
Alle uheld											
Almindelig personbil											
Personskade i alt	4 178	4 375	4 328	3 839	3 219	3 157	3 301	2 997	2 545	2 090	1 939
Dræbte	238	245	236	186	168	138	167	196	164	134	106
Alvorligt tilskadekomne	1 620	1 695	1 603	1 412	1 147	1 076	1 176	1 087	936	799	770
Lettere tilskadekomne	2 320	2 435	2 489	2 241	1 904	1 943	1 958	1 714	1 445	1 157	1 063
Taxi											
Personskade i alt	68	43	33	45	30	38	22	19	19	23	7
Dræbte	4	1	0	0	1	0	1	0	0	1	0
Alvorligt tilskadekomne	24	13	10	16	7	8	4	4	7	5	2
Lettere tilskadekomne	40	29	23	29	22	30	17	15	12	17	5
Varebil 0-2.000 kg											
Personskade i alt	132	139	141	89	149	117	101	77	64	60	44
Dræbte	6	8	8	5	6	8	9	4	5	2	4
Alvorligt tilskadekomne	61	64	59	40	56	35	35	30	25	22	22
Lettere tilskadekomne	65	67	74	44	87	74	57	43	34	36	18

Statistikken omfatter kun de personskader, der er kommet til politiets kendskab. For at belyse det såkaldte mørketal har Danmarks Statistik siden 1996 gennemført en undersøgelse, hvor data om skadestuebesøg inddrages. Undersøgelsen viser, at det samlede antal personskader ved færdselsuheld er langt større end det antal, der registreres af politiet. Dækningsgraden vedr. dræbte er dog næsten 100 pct. Fra 2001 er resultaterne offentliggjort i tabellen MOERKE.

27-2-2014 Danmarks Statistik , © www.statistikbanken.dk/UHELD8

Persontransport efter transportmiddel og tid

	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	
Personbiler og varebiler under 2.001 kg.	50 186	49 183	49 030	49 280	50 149	49 705	49 686	51 229	51 501	51 155	5(

Unit: million person kilometres

En revision af busstatistikken pågår. Nuværende tal er behæftet med stor usikkerhed. Vejdirektoratet har i august 2011 revideret en ræ vejtransportstatistikkens tal. For yderligere oplysninger om metodegrundlaget henvises til Vejdirektoratets hjemmeside, www.vd.dk. Kil Vejdirektoratet (vejtrafik) og Danmarks Statistik (tog, skib, fly)

27-2-2014 Danmarks Statistik , © www.statistikbanken.dk/PKM1

Persontransport efter transportmiddel og tid

	2005	2006	2007	2008	2009	2010	2011
Personbiler og varebiler under 2.001 kg.	49 705	49 686	51 229	51 501	51 155	50 727	52 067

Unit: million person kilometres

En revision af busstatistikken pågår. Nuværende tal er behæftet med stor usikkerhed. Vejdirektoratet har i august 2011 revideret en række af vejtransportstatistikkens tal. For yderligere oplysninger om metodegrundlaget henvises til Vejdirektoratets hjemmeside, www.vd.dk. Kilde: Vejdirektoratet (vejtrafik) og Danmarks Statistik (tog, skib, fly)

27-2-2014 Danmarks Statistik , © www.statistikbanken.dk/PKM1

Appendix D

Directives, Certificates & Regulations

D.1 CE-Marking

If a company put the CE-mark on their products, it is a statement that the product fulfils the safety demands from the relevant directive. When a company have produced a product it is their responsibility that the product fulfil the requirements for a CE-mark. The CE-mark is created to make a more free flowing trade market and its directives is valid for all EU's countries. If there is a CE-mark on a product, it means it has been produced in correlation with the European demands for health, safety and environment. Some of the products that goes into an autonomous car must comply with the following CE-mark directive:

Low-voltage Electrical products 72/23/EEC, #93/68/EEC

To get a product CE-marked they have to compile a technical document of how the product is developed and what materials is used, a risk assessment of the product basically if it is safe to use or not, and a declaration of conformity of the product.((45) & (46))

D.2 E-Marking

When a company manufactures electronics with the purpose of installing it into European registered cars their product have to undergo E-mark testing. This is because of the ECE(Economic Commission for Europe), they made an EU directive that came into effect in October 2002 (EU directive 72/245/EEC). When a product has gone through testing and received the E-Mark, it is thereafter allowed into the European car market. The E-mark is some basic demands for a product, but ECE member-country can also have more strict regulations on this topic(47).

> *"...cars, motorcycles and a variety of components and systems must be certified to meet the basic requirements of traffic safety and environmental protection."*
>
> Eurofins China(48)

E-mark numbering: There will be numbering on the E-mark such as E18 referring to

an E-mark from Denmark.
E-mark Logo: There are two different E-mark logos, there is a rectangular one and a circular one. There is a rectangular one it has to do with car air pumps, seat heating, Electrical Cars, Car stereo etc. whereas the round version of the mark has to do with headlights, seat belts, windscreens etc(47).

It is quite a long process for the E-mark certification it may take as long as a year. The first thing a company should do is contact an EMC(**E**lectro**m**agnetic **C**ompatibility) laboratory, and notify them of early production and request a Notified Body[1] to show up at test day. The company have to talk with the Notified Body about conformity of production, certification services and request of test witnessing. After this is done the company have to complete a formal application with the Notified Body. The Notified Body and the Witness should oversee the test, if all is good only a final review of results and contracts is needed for approval(49).

D.3 EC Directive & UN Regulations

The purpose of these directives and regulations is to ensure that a car and all parts of it comply with different standards on performance, safety and design. Before approval, a third party must be contacted and similar guidelines must be followed as in (D.2-E-Marking, fourth paragraph)The EC Directive is valid in all EU member states and are used to technical approve a car that have not yet been registered within EU, and it is also valid for vehicle systems and other car parts. The UN Regulations are similar to the EC Directive except it does not type approve whole cars only vehicle systems and car parts(50).

D.4 WEEE Directive and RoHS Certificate

The WEEE and RoHS directives are directives made by the European Union. Both of the directives are concerning electronic equipment in some way. The WEEE (Waste Electrical and Electronic Equipment) directive is concerning what happens to the electronics at EOL of a product, some electronics cannot be reused and it can just be recycled along with the car. The RoHS is concerning the electronics in the car. RoHS stands for "Restriction of Hazardous Substances", and it prohibits the use of the following six heavy metals; mercury, lead, polybrominated diphernyl ether, hexavalent chromium, cadmium and poly bromiated biphenyl in electronics devices.

D.5 End of Life Vehicle

Every year thousands of cars are taken out of commission this amounts to many tons of garbage, this garbage needs to be dealt with in an environmentally friendly way. In the European Union the Directive 2000/53/EC – the "ELV Directive", is dealing with how the dismantling of cars are handled, and how the disposing of the fluids in the car must be handled so it does minimal harm to the environment. The same directive also deals with the reuse/recycling of components in a car(51).

[1] A Notified Body is a third-party company authorised to grant certification on behalf of the EU.

In Denmark when a car is too severely damaged for it to be repaired, it is mandatory by legislation to be recycled. When a car is recycled in Denmark, it is taken to a scrapyard, and the owner is compensated from the government. A scrap dealer is responsible for disposal of everything in an environmental way and takes out everything that can be used as spare parts. The scrap metal is re-melted and all fluids is drained for re-usage and up towards 80 % is reused of the car that is being scrapped((52) & (53) & (51)).

Appendix E

Breaking Diagram

S is summer tyres and W is winter tyres

	40 *km/h*	50 *km/h*	60 *km/h*	70 *km/h*	80 *km/h*	90 *km/h*	100 *km/h*	110 *km/h*	120 *km/h*	130 *km/h*
S:Dry	18*m*	25*m*	33*m*	41*m*	51*m*	61*m*	72*m*	84*m*	97*m*	111*m*
S:Wet	19*m*	26*m*	35*m*	44*m*	54*m*	66*m*	78*m*	91*m*	105*m*	121*m*
S:Snow	42*m*	62*m*	86*m*	114*m*	146*m*	181*m*	221*m*	264*m*	311*m*	362*m*
S:Ice	59*m*	88*m*	124*m*	165*m*	212*m*	265*m*	325*m*	390*m*	461*m*	538*m*
W:Dry	19*m*	27*m*	35*m*	45*m*	55*m*	67*m*	79*m*	93*m*	107*m*	123*m*
W:Wet	20*m*	28*m*	37*m*	46*m*	57*m*	70*m*	83*m*	97*m*	113*m*	129*m*
W:Snow	29*m*	42*m*	58*m*	75*m*	95*m*	117*m*	141*m*	168*m*	197*m*	228*m*
W:Ice	38*m*	56*m*	77*m*	102*m*	130*m*	161*m*	196*m*	234*m*	275*m*	320*m*

Figure E.1: Emergence breaking distance with summer & winter tyres under various road conditions(54).

Appendix F

Calculations

F.1 Breaking Distance

Constants used:
The car's breaking system are allowed a reaction time of up to 0,3 s (27).
30 km/h, 50 km/h, 80 km/h, 100 km/h speed selections of the car.
$9,816\,m/s^2$ gravitational pull in Denmark(55).
0,14 g-force from Table 2.7 found to be a comfortable acceleration(26).
$5,8\,m/s^2$ requirements to cars breaking capabilities(27).

Formulas used:

Symbol	Meaning	Unit
a	Acceleration	m/s^2
b	Breaking distance	m
g	Gravitational pull	$9,816\,m/s^2$
t	Time	s
V	Velocity	km/h
v	Velocity	m/s
y	Reaction Time	s

Table F.1: Symbols & units

Converting km/h to m/s

$$\frac{V \cdot 1000}{3600} = v$$

Calculate time from speed in m/s and acceleration in m/s^2

$$\frac{v}{a} = t$$

Converting acceleration to g-force

$$\frac{a}{g} = \text{g-force}$$

Converting g-force to acceleration

$$\text{g-force} \cdot g = a$$

Calculate breaking distance (Formula for even acceleration)

$$(t + y) \cdot \frac{1}{2} \cdot v = b$$

Converting a g-force of 0,14 to acceleration

$$0,14 \cdot 9,816\,m/s^2 \approx 1,37\,m/s^2$$

Converting 5,8 m/s^2 to g-force

$$\frac{5,8\,m/s^2}{9,816\,m/s^2} \approx 0,59$$

Converting 30 km/h, 50 km/h, 80 km/h, 100 km/h to equivalent speed as m/s

$$\frac{30\,km/h \cdot 1000\,m/km}{3600\,s/h} \approx 8,33\,m/s \qquad \frac{50\,km/h \cdot 1000\,m/km}{3600\,s/h} \approx 13,89\,m/s$$

$$\frac{80\,km/h \cdot 1000\,m/km}{3600\,s/h} \approx 22,22\,m/s \qquad \frac{100\,km/h \cdot 1000\,m/km}{3600\,s/h} \approx 27,78\,m/s$$

Calculating emergency breaking time at 30 km/h, 50 km/h, 80 km/h, 100 km/h

$$\frac{8,33\,m/s}{5,8\,m/s^2} \approx 1,44\,s \qquad \frac{13,89\,m/s}{5,8\,m/s^2} \approx 2,39\,s$$

$$\frac{22,22\,m/s}{5,8\,m/s^2} \approx 3,83\,s \qquad \frac{27,78\,m/s}{5,8\,m/s^2} \approx 4,79\,s$$

Calculating emergency breaking distance at 30 km/h, 50 km/h, 80 km/h, 100 km/h

$$(1,44\,s + 0,3\,s) \cdot \frac{1}{2} \cdot 8,33\,m/s \approx 7\,m \qquad (2,39\,s + 0,3\,s) \cdot \frac{1}{2} \cdot 13,89\,m/s \approx 19\,m$$

$$(3,83\,s + 0,3\,s) \cdot \frac{1}{2} \cdot 22,22\,m/s \approx 46\,m \qquad (4,79\,s + 0,3\,s) \cdot \frac{1}{2} \cdot 27,78\,m/s \approx 71\,m$$

Calculating comfortable breaking time at 30 km/h, 50 km/h, 80 km/h, 100 km/h

$$\frac{8,33\,m/s}{1,37\,m/s^2} \approx 6,08\,s \qquad \frac{13,89\,m/s}{1,37\,m/s^2} \approx 10,14\,s$$

$$\frac{22,22\,m/s}{1,37\,m/s^2} \approx 16,22\,s \qquad \frac{27,78\,m/s}{1,37\,m/s^2} \approx 20,28\,s$$

Calculating comfortable breaking distance at 30 km/h, 50 km/h, 80 km/h, 100 km/h

$$(6,08\,s + 0,3\,s) \cdot \frac{1}{2} \cdot 8,33\,m/s \approx 27\,m \qquad (10,14\,s + 0,3\,s) \cdot \frac{1}{2} \cdot 13,89\,m/s \approx 73\,m$$

$$(16,22\,s + 0,3\,s) \cdot \frac{1}{2} \cdot 22,22\,m/s \approx 184\,m \qquad (20,28\,s + 0,3\,s) \cdot \frac{1}{2} \cdot 27,78\,m/s \approx 286\,m$$

F.2 Blind Spot Calculation

Calculating the view angle for the distance sensor mounted on the car.

The car have a width of 29 cm. The sensor need at least a distance of 3 cm to get an echo, view angel are 15 ° and range are 400 cm.

The half and total width the sensor covers in 400 cm distance are
$x = half$ width, $z = 2 \cdot x = total$ width.

$$x = \tan\left(\frac{15^\circ}{2}\right) \cdot 400\,cm \approx 53\,cm \Rightarrow z = 2 \cdot 53\,cm \approx 106\,cm$$

The distance from the car needed to cover the width of the car with the sensor.
$y = distance$ to cover the width of the car.

$$y = \frac{\frac{29\,cm}{2}}{\tan\left(\frac{15^\circ}{2}\right)} \approx 110\,cm$$

F.3 Distance Sensor Time

The use of the HC-SR04 distance sensor are based on the data provided in the user documentation: When $\Delta\,time$ between trigger and echo is measured in μs the distance, in cm, to an object can be found by dividing $\Delta\,time$ with 58(41).

$$range\,cm = \frac{\Delta\,time\,\mu s}{58} \implies range\,m \cdot 100\,\frac{cm}{m} \cdot 58 = \Delta\,time\,\mu s$$

This is the formula used to calculate the $\Delta\,time$ in [μs] needed to receive an echo from comfortable breaking distance range at, 30 km/t, 50 km/t, 80 km/t, 100 km/t. To get the $\Delta time$ in s divide the found $\Delta time$ [μs] with 1000000 $\frac{\mu s}{s}$ The modified formula:

$$\frac{range\,m \cdot 100\,\frac{cm}{m} \cdot 58}{1000000\,\frac{\mu s}{s}} = \Delta\,time\,s$$

The comfortable breaking distance for these speeds are calculated in Appendix F.1-Breaking Distance. (The cars movment during this time is neglected in these calculation).

$$\frac{27\,m \cdot 100\,\frac{cm}{m} \cdot 58}{1000000\,\frac{\mu s}{s}} \approx 0,16\,s \qquad \frac{73\,m \cdot 100\,\frac{cm}{m} \cdot 58}{1000000\,\frac{\mu s}{s}} \approx 0,42\,s$$

$$\frac{184\,m \cdot 100\,\frac{cm}{m} \cdot 58}{1000000\,\frac{\mu s}{s}} \approx 1,07\,s \qquad \frac{286\,m \cdot 100\,\frac{cm}{m} \cdot 58}{1000000\,\frac{\mu s}{s}} \approx 1,66\,s$$

F.4 ADC Step-size and Resolution

The ADC converts an analogue signal to a digital signal, in this case the ADC are 10-*bit* which gives a resolution of 1024 steps, and the span in this case are 0-5 V. The analogue value are transformed to a digital signal between 0-1023. By dividing 5 V with 1024 the step size can be obtained.

$$stepsize = \frac{5\,V}{1024} = 0,004882812\,\frac{V}{step} \approx 0,0049\,\frac{V}{step}$$

Based on the result above it is possible to generate a table of 1024 steps with cosponsoring analogue and digital values by starting from 0 and increment by steep size for each increment in step. Below there are a table of the first 12 steps. For the table it are the rounded off values that are used.

Step	**Analogue value**	**Digital value[dec]**
12	0,0537 V	11
11	0,0488 V	10
10	0,0439 V	9
9	0,0390 V	8
8	0,0342 V	7
7	0,0293 V	6
6	0,0244 V	5
5	0,0195 V	4
4	0,0146 V	3
3	0,0098 V	2
2	0,0049 V	1
1	0,0000 V	0

Table F.2: Step size, resolution, ADC

Appendix G

Pin Layout

Arduino	I/O	Purpose	Wire	Note
Analogue:				
A0	Input	Curent sensing channel A on motor shield		Unused
A1	Input	Curent sensing channel B on motor shield		Unused
A2	Input	Signal from potentiometer (steering)	Blue	
A3	Input	Signal from Hall Effect sensor	Red	
Digital:				
D2	Output	Signal to distance sensor (trigger)	Yellow	
D3	Output	PWM signal to motor shield channel A	Shield	
D4	Input	Signal from distance sensor (echo)	Blue	
D8	Output	Break signal to motor shield channel B	Shield	
D9	Output	Break signal to motor shield channel A	Shield	
D11	Output	PWM signal to motor shield channel B	Shield	
D12	Output	Direction channel A on motor shield	Shield	
D13	Output	Direction channel B on motor shield	Shield	

Table G.1: Pin layout

Appendix H

Code

```
/*#########################################################################################
                              START OF SERVOMOTOR PART
#########################################################################################*/

/* Motor Control Constants Begin */
const int
PWM_A    = 3,
DIR_A    = 12,
BRAKE_A = 9,

PWM_B    = 11,
DIR_B    = 13,
BRAKE_B = 8;

SERVO_OS = A2;        //Servo Oscillation

const int
minPWMMain  = 170,   //Minimum PWM of main motor
minPWMServo = 150,   //Minimum PWM of servo motor
changeRate = 2,      //PWM change rate slope (See servoSpeed())
defaultBurst = 30,   //Default servo burst duration (See servoSpeed())

wheelTolerance = 15;     //Accuracy of front wheels

const float
burstRate = 0.45;    //Burst servo PWM slope

int
//maxLeftOs is bigger than maxRightOs
maxLeftOs = 573,     //Max Left Oscillation
maxRightOs = 233,    //Max Right Oscillation
maxLeftAn = 90,      //Max Left Angle
maxRightAn = 0,      //Max Right Angle
centerOs = (maxLeftOs + maxRightOs)/2, //Centre Oscillation
centerAn = (maxLeftAn + maxRightAn)/2; //Centre Angle

int burstLength;         //Global variable containing burst duration

boolean dirServo;        //Servo Direction (HIGH == Left || LOW == Right)
int speedServo;          //Servo PWM speed (0-255)

const int
endSlope = 50,           //End turning at endSlope (in cm)
```

```
startSlope = 100;         //Start turning at startSlope (in cm)

int
funcY_A,                  //Function A y-intercept
funcY_B;                  //Function B y-intercept

float
angleCal;                 //Angle calculation slope

/* Motor Control Constants End.... */

void setupSlope() {
  //Calculate turn slope of both functions
  angleCal = (maxRightAn - centerAn)/(endSlope - startSlope);
  // Calculate y-intercept of function A
  funcY_A = ((endSlope*centerAn) - (maxRightAn*startSlope))/(endSlope -
      startSlope);
  // Calculate y-intercept of function B
  funcY_B = ((endSlope*centerAn) - (maxLeftAn*startSlope))/(endSlope -
      startSlope);
}


void servoSpeed(int minium,int change) {
  int t1; //Set variable
  burstLength = defaultBurst; //Default duration
  //Set direction of servo
  if (change < 0)  {
    dirServo = 0;
  } else {
    dirServo = 1;
  }
  //Set speed of servo based of change needed
  //If max speed reached set to 255 == "Full Duty Cycle"
  t1 = abs(change*changeRate) + minium;
  //Increase PWM duration
  if (t1 > 256) {
    //Calculate burst duration (PWM speed overflow)
    burstLength = ((t1-255)*burstRate)+defaultBurst;
    speedServo = 255; //Set PWM to "Full Duty Cycle"
  } else {
    speedServo = t1; //Set Servo PWM speed
  }

}

int convertAngle(int Angle) {
  //Check for Angle overflow and set max allowed angle
  if (Angle > maxLeftAn) {
    Angle = maxLeftAn;
  }
  if (Angle < maxRightAn) {
    Angle = maxRightAn;
  }
  //Return calculated oscillation from angle
  return (((maxLeftOs - maxRightOs)/maxLeftAn)*Angle) + maxRightOs;

}

```

```
int getOs() {
  int temp = 0; //Set Variable
  int getOsAcc = 10;
  delay(20); //Delay 20 milliseconds
  //Measure angle, for precision repeat ten times (Average)
  for (int i = 0; i < getOsAcc; i++) {
    temp += analogRead(SERVO_OS);
  }
  return temp/getOsAcc;
}

void turnServo(int Angle) {
  int turn1;
  int cAngle = convertAngle(Angle); //Get converted angle
  int Os1 = getOs();      //Get current oscillation
  int loopCount=0;        //Set control counter
  while(abs((cAngle - Os1)) > wheelTolerance){
    //Get difference between current angle and angle to set
    turn1 = cAngle - Os1;
    servoSpeed(minPWMServo,turn1);
    digitalWrite(DIR_B,dirServo);     //Set direction
    digitalWrite(BRAKE_B,LOW);        //Release breaks
    analogWrite(PWM_B,speedServo);    //Move servo
    delay(burstLength); //Delay using burstLength set in servoSpeed()
    digitalWrite(PWM_B,LOW);          //Engage breaks on servo
    digitalWrite(BRAKE_B,HIGH);
    Os1 = getOs();            //Recheck oscillation
    if (loopCount > 5) {      //If run 5 times break
      break;
    }
    loopCount++;
    printStuff(Angle); //Print servo information to serial port
  }
}

void printStuff(int test) { //Servo information print function
  for (int i = 0; i < 10; i++) {
    Serial.println();
  }
  Serial.println(test);
  Serial.println(convertAngle(test));
  Serial.println(getOs());
  Serial.println(dirServo);
  Serial.println(speedServo);
  Serial.println(burstLength);
  Serial.println("---------------");
}

/*#####################################################################################
                              END OF SERVOMOTOR PART
#####################################################################################*/


/*#####################################################################################
                              START OF MOTOR PART

 1) Pins defined in SERVOPART
#####################################################################################*/
```

```
void drive(boolean directionMotor,int speedMotor) { //Move Main Motor
  digitalWrite(BRAKE_A,LOW);
  digitalWrite(DIR_A, directionMotor); //HIGH == FORWARD && LOW == REVERSE
  analogWrite(PWM_A, speedMotor);
}

void neutral() { //Stop Main Motor
  digitalWrite(PWM_A,LOW);
  digitalWrite(BRAKE_A,HIGH);
}

 /*#########################################################################################
                    END OF MOTOR PART
#########################################################################################*/


 /*#########################################################################################
                    START OF DISTANCE PART
#########################################################################################*/

/* Distance Constants Begin */

const int
echoPin = 4, //Echo pin
triggerPin = 2; //Trigger pin

/* Distance Constants End.... */

/* Keep in mind distance measurements are very SLOW */

long distance() {
  digitalWrite(triggerPin,HIGH);      //Trigger output of Ultrasonic Sensor
  delayMicroseconds(2);               //Keep output HIGH for 2 microseconds
  digitalWrite(triggerPin,LOW);       //Set output LOW
  return pulseIn(echoPin,HIGH)/29/2;  //Calculate distance to object
}

/*#########################################################################################
                    END OF DISTANCE PART
#########################################################################################*/


/*#########################################################################################
                    START OF REASONING PART
#########################################################################################*/

/* Reasoning Constants Begin */

long distanceTemp;

/* Reasoning Constants End.... */

int calculateAngleDis(boolean right) {
  int temp;

  if (right) { //Calculate oscillation using distance to object (Going Right)
    temp = (distanceTemp*angleCal) + funcY_A;
    if (temp > centerAn) {
        temp = centerAn;
```

```
    }
  } else { //Calculate oscillation using distance to object (Going Left)
    temp = (distanceTemp*(-angleCal)) + funcY_B;
    if (temp < centerAn) {
      temp = centerAn;
    }
  }
  if (temp < maxRightAn) {//Correct Angle Overflow (Right)
    temp = maxRightAn;
  } else if (temp > maxLeftAn) {//Correct Angle Overflow (Left)
    temp = maxLeftAn;
  }
  Serial.println(temp);
  return temp;
}

/*######################################################################################
                      END OF REASONING PART
######################################################################################*/


/*######################################################################################
                      START OF HALLSENSOR PART
######################################################################################*/

float velocity(){
  float test;
  long milliSec = millis(); // set timer
  while(1){
    if(hs.read() >= 550 || hs.read() <= 450){ // north or south pole is close
      milliSec = millis(); // reset timer
      /*
      Movement during 50ms(1m/s): 1*0,020 = 0,02 m (2 cm; enough for magnet?)
      Movement during 50ms(2,778m/s same as 10 km/h):2,778*0,020 = 0,056 m (~
          quarter rounds)
      */
      delay(20); //Delay short while to get magnet away from sensor, get
          better time.
      while(1){
        if(hs.read() >= 750 || hs.read() <= 250){
          millisec = millis() - milliSec;
          float speedCmSec = ((8.5/(millisec/1000))/100)*3.6;
          return (speedCmSec); // 8,5 is the distance between magnets in cm.
        }
        if(millis()-milliSec >= 100){ //Time-out in case of too long wait,
            assumes standing still
          return(0.00);
        }
        if(millis()-milliSec >= 21){ // Time-out if reading magnet as soon as
            20 ms.
          return(0.00); // delay is over, assumes standing still next to magnet
        }
      }
    }
    if(millis()-milliSec >= 100){
      return(0.00);
    }
  }
}
```

```

/* #####################################################################################
                        END OF HALLSENSOR PART
##################################################################################### */

boolean doWeRule() {
    return 1;
}

void setup() { //Set-up function

  Serial.begin(9600);
  // Configure the A output
  pinMode(BRAKE_A, OUTPUT);  // Brake pin on channel A
  pinMode(DIR_A, OUTPUT);    // Direction pin on channel A
  pinMode(PWM_A, OUTPUT);    // PWM pin on channel A, servo

  // Configure the B output
  pinMode(BRAKE_B, OUTPUT);  // Brake pin on channel B, servo
  pinMode(DIR_B, OUTPUT);    // Direction pin on channel B, servo
  pinMode(PWM_B, OUTPUT);    // PWM pin on channel B, servo

  digitalWrite(BRAKE_A,HIGH);  // Set brake on channel A, to ON
  digitalWrite(BRAKE_B,HIGH);  // Set brake on channel B, to ON

  //Enable UltraSonic Sensor Pins
  pinMode(echoPin,INPUT);
  pinMode(triggerPin,OUTPUT);

  setupSlope(); //Setup slopes
}

/* LOOP Constants Begin */

boolean dirLoop = 1;

/* LOOP Constants End.... */

void loop() {

  boolean loopBool = 0; //Control variable

  float speed = velocity();

  while (distanceTemp > startSlope) { //Move vehicle forward
    distanceTemp = distance();
    drive(1,255);
    turnServo(45);
  }

  //Turn servo according to calculateAngleDis()
  while (distanceTemp < startSlope && distanceTemp > endSlope) {
    drive(1,255);
    turnServo(calculateAngleDis(dirLoop));
    distanceTemp = distance();
  }

  //Stop vehicle select reverse until endSlope(Distance) to object
  if (distanceTemp < endSlope) {
```

```
    neutral();
    delay(50);
    while (distanceTemp < endSlope) {
      drive(0,240);
      if(!loopBool) {
        //Turn front wheels opposite
        if (dirLoop) {
          turnServo(maxRightAn);
          dirLoop = 0;
          loopBool = 1 ;
        } else {
          turnServo(maxLeftAn);
          dirLoop = 1;
          loopBool = 1;
        }
      }
      distanceTemp = distance();
    }
    distanceTemp = distance();
    neutral();
    delay(50);
    }
    doWeRule();
}
```