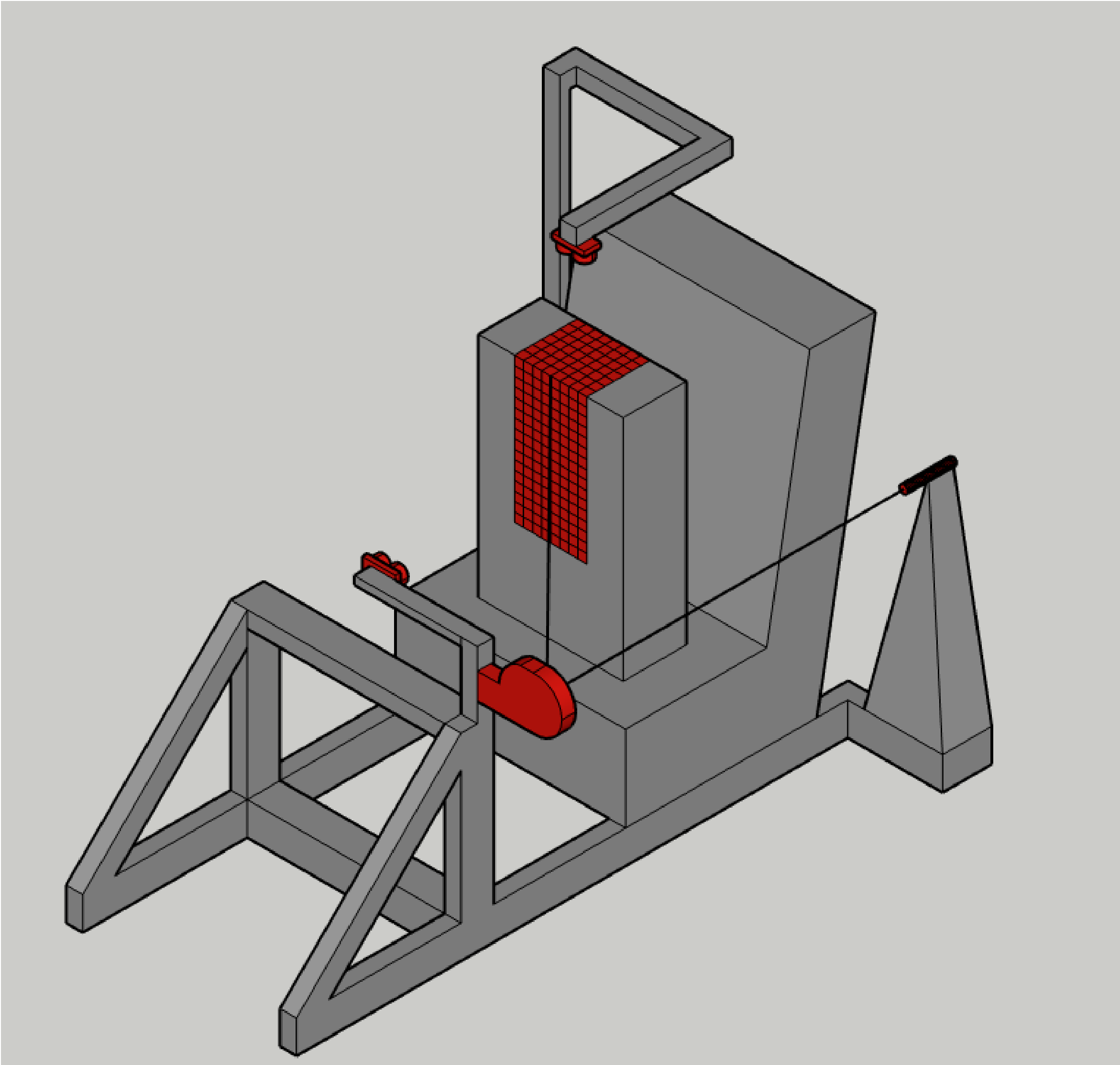
CHALMERS



Automatic rear-view mirror control

A study for how and if it should be implemented in modern vehicles

Bachelor’s thesis in mechatronics engineering

Johan Karlsson and Alexander Mörch

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2020

BACHELOR'S THESIS 2020

Automatic rear-view mirror control

A study for how and if it should be implemented in modern vehicles

ALEXANDER MÖRCH

JOHAN KARLSSON

Department of Electrical Engineering

CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2020

Automatic rear-view mirror control
Bachelor’s Thesis 2020
Department of Electrical Engineering
ALEXANDER MÖRCH
JOHAN KARLSSON

Supervisor: Ola Wennberg
Examiner: Lars Hammarstrand, Electrical Engineering

Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone: +46-(0)31 772 10 0

Cover: A concept of the final test rig developed.

ACKNOWLEDGMENTS

This thesis work covers the topic of automatic rear-view mirror control. It was performed during the spring of 2020, by two students at Chalmers University of Technology. The project spanned 15 HP and was performed at the company AFRY. Both participants studied a 180 HP program within mechatronics and representatives from the electrical institution oversaw the project.

We would like to thank Ola Wennberg for allowing us to perform our thesis work at AFRY, Göran Hult for his council and aid with the MPLAB snap, Lars Hammarstrand for being our examiner, Veronica Olesen for her great work as head of the mechatronics program, and last but not least Rami Karjalainen for his aid in constructing the test rig and all the coffee breaks we shared at AFRY.

ABSTRACT

When using unfamiliar cars, there is a risk that drivers tend to adjust rear-view-mirrors incorrectly, or not at all. This results in a higher risk of an accident. An automatic system that adjusts the rear-view-mirrors for the driver, would ensure that the field of view was always correct and thus decrease that risk.

This project aims to examine the possibility of creating a system that could automatically adjust a car’s rear-view-mirrors, based on the driver’s position. The intent was to create a rig, made of consumer components, that could be used to test said system. A primary goal of the system was to show that an adequate view of the rear of the vehicle was visible after the control is performed.

Our solution was a closed-loop system realised by a microprocessor. Connected to this was two ultrasonic sensors, that were used to measure the position of the driver. This position is then used in trigonometric calculations to determine the wanted angular position of the mirror. The mirror used had built-in potentiometric position feedback, and it was actuated by two electric motors.

All these components were mounted on the test rig. It was constructed to mimic the interior of a car, with a real car seat and side-view-mirror. A laser pointer was placed at the side behind the rig. By aiming the laser straight to the mirror, it was possible to measure where the laser was reflected. This is used to indicate that the wanted field of view has been achieved.

Through a series of tests, the system was evaluated to see how well it performed. Different seat positions and driver heights were chosen. The result showed that the concept works, although not perfect. A series of measurement errors was encountered which had a negative effect on the outcome.

In conclusion, the system works but needs improvements. Although the system can adjust the position of the mirror, it is not accurate enough to replace a human operator. It might, however, be suitable as a fast setting. To increase the durability and accuracy of the system the use of Hall effect sensors or stepper motors should be considered. Furthermore, ultrasonic distance sensors should be switched out for “smart” sensors used in modern cars. Finally, to test the usability of the system it should be implemented in a real vehicle.

TABLE OF CONTENTS

1 INTRODUCTION

Introducing the background for the thesis, why it is interesting to research the possibility of automatic rear-view mirrors and how it is planned to be performed.

1.1 BACKGROUND

In our society, there is a call to decrease energy consumption and to move away from using non-renewable resources. An important part of this is to utilise existing resources more efficiently. As an example, instead of every person owning a car, cars can be shared in carpools. An increasing amount of companies are moving towards the carpooling concept.

A carpool consists of one or more cars that can be shared by multiple people. Instead of buying your own car that sits idle during most of the night and day, you borrow one for a limited amount of time when you need it. When you are not using the car, someone else is. This might, however, present a problem.

It is common for drivers to only adjust the settings of the rear-view-mirrors once. When using an unfamiliar car, the driver might, therefore, fail to adjust the rear-view-mirrors out of habit. This increases the risk of an accident. Some type of system is needed to prevent this.

A system that could adjust the mirrors for the driver would need to know the driver’s position relative to the mirrors, and it would need to be able to adjust the position of said mirrors. Luckily, modern cars are filled with actuators and sensors that can do everything mentioned above.

There are electric motors that control the position of the mirrors, as well as sensors that measure the position of said mirrors. Some cars even have sensors that detect the driver’s eye position and can track where the driver is looking. All prerequisites are there and all that is needed is combine them into one system.

This project aims to create a system that automatically adjusts the position of the rear-view mirrors of a car. The intent is to use components that are available in cars today. The tests will be carried out on a rig that is constructed to mimic a car. By testing the system as true to life as possible, the intent is to evaluate if such a system could be implemented today.

1.2 PURPOSE

The purpose of this thesis is to show how automated adjustment of rear-view mirrors can be done with components already available in modern cars. Such components are advanced eye-tracking sensors and electrically adjustable mirrors with position feedback.

Initial investigation revealed that due to budgetary constraints it would not be possible to use the advanced eye-tracking sensors mentioned above. Instead it was decided to use cheaper ultrasonic sensors that fit into the budget.

An algorithm that determines the desired mirror position for any given driver position will need to be constructed. Then the purpose is to insert this algorithm into a program that controls all the components.

Finally, a test rig will be constructed to evaluate the whole system. The goal is to be able to update the code as needed for proper function.

1.3 LIMITATIONS

- The construction is a prototype and is not, in this stage, intended to be implemented into a real vehicle.
- The system will not be used with a human driver. To get measurements and eliminate the possibility of human movement a box will be used in the tests.
- Eye-tracking sensors will not be used, they are over budget in price.
- The system will not be operating under a time constraint.
- Due to budget constraints an advanced eye-tracking sensor system will not be used. Instead position will be measured with cheaper ultrasonic sensors.
- The system does not have to have high accuracy, tolerance in angle will be allowed.

2 THEORY

This chapter explains the theoretical concepts used in this report. It covers terms such as yaw and pitch, different control concepts, describes the function of electrical motors, and lastly it dives into a description of different types of distance and angle sensors.

2.1 COORDINATE SYSTEM

In this report the words longitude, pitch and yaw are used often. To avoid misinterpretations a definition is needed. The following descriptions will refer to Figure 2.1.

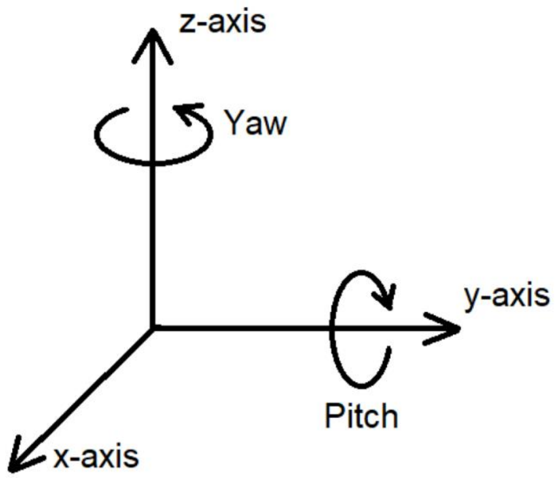


Figure 2.1. Describing the concepts of vertical, longitudinal, and horizontal position, as well as pitch and yaw.

Vertical movement occurs along the z-axis. The vertical position, therefore, refers to displacement along the z-axis in relation to the origin. When the word longitude is used it refers to the direction running along the x-axis. Thus, longitudinal position refers to displacement along the x-axis relative to the origin. By the same logic, horizontal describes movement along the y-axis. The horizontal position being displacement along the y-axis in relation to the origin.

Yaw refers to the motion that spins around the z-axis and is parallel to the yx-plane. Thus, the yaw angle refers to the angular displacement between the y- and x-axis. Similarly, pitch describes the motion that spins around the y-axis and is parallel to the zx-plane. Finally, the pitch angle refers to the angular displacement between the z- and x-axis.

2.2 CONTROL SYSTEMS

In [2] a "system" is defined as: "An arrangement of parts within some boundary which work together to provide some form of output from a specified input or inputs." [2, p. 2]. Then a control system can be defined as a system that is used to maintain a desired result or value [2, p. 3]. There are two main modes of control: open-loop control and closed-loop control.

In an open-loop control system, the output signals have no bearing on the input signals. In other words, the system takes no feedback from the process which it is controlling. Therefore, the system has no way of detecting and correcting errors. If this type of control is used a calibration function must be implemented so that the current position is the right [2, p. 4].

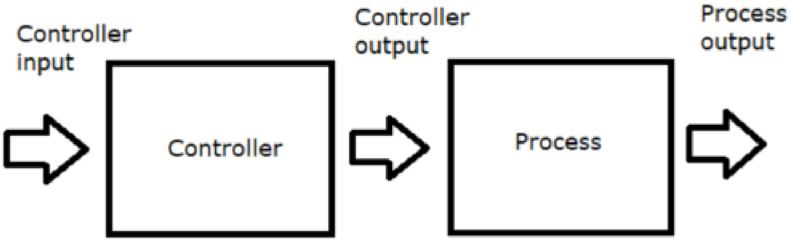


Figure 2.2 Open-loop system

The key in a functioning open-loop system is that it is known how much the input controller changes the output signal. For example, using a stepper motor, it is known digitally which position the motor is supposed to be in as long as no interference has occurred. In other words, an open-loop system enforces restrictions in what kind of hardware that can be used.

A closed-loop system measures the process value and feeds it back into the system, creating a loop. This allows the system to adjust for errors. In general, closed-loop systems are more complex than open-loop systems but in return, they are more accurate [2, pp. 4-5].

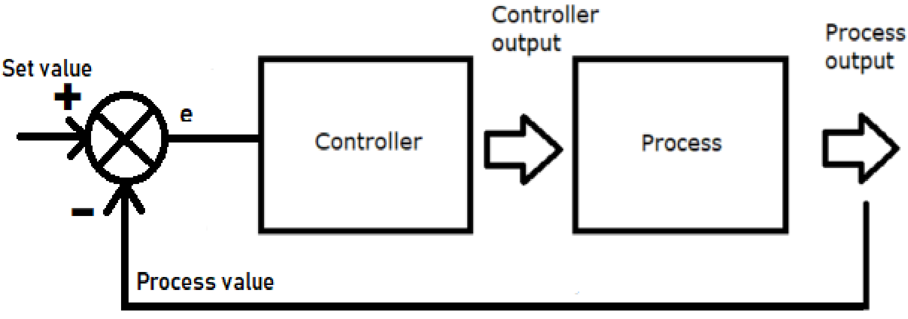


Figure 2.3 Closed-loop system

Take the set-value (the value that the system aims to regulate itself into) and subtract it by the process value. By that an error signal is achieved, if the process value is too high it will be a negative number and if the process value is too low it will be a positive number.

There are different kinds of regulators that uses this error signal differently. The most common regulators are:

- P-regulator
- PI-regulator
- PD-regulator
- PID-regulator
- Two-stage regulator

A P-regulator uses the error signal and amplifies/dampen it to achieve a control signal. The I and D in the 3 regulators after stands for integrate and derivate. They use either the integration of the error or the derivate of the error. These regulators are used for dynamic processes.

The two-stage regulator is simply on or off. Here it is not important how big the error signal is, only if it is positive or negative. Positive means it must be turned on, negative means that it is okay.

The two-stage regulator can be improved upon by setting tolerances, it can be okay that the error is between chosen levels. It can also have a level that triggers a reverse, for example, a water tank. If the level is too low you start water pumps to get water into the tank, then it can be a range where it is okay that the water level is. But it might be a level that is too high, so you want to reverse the pumps. This is still a two-stage regulator, on or off, but it has a reverse function as well.

2.3 ELECTRIC MOTORS

In most cases, the actuation in cars is carried out by direct current motors, also known as DC motors. Following is a description of the conceptual function of two common types of DC motors: brushed DC motors and stepper motors.

The brushed DC motor is the most common version of the DC motor. It consists of three main components: the commutator, the armature, and the brushes.

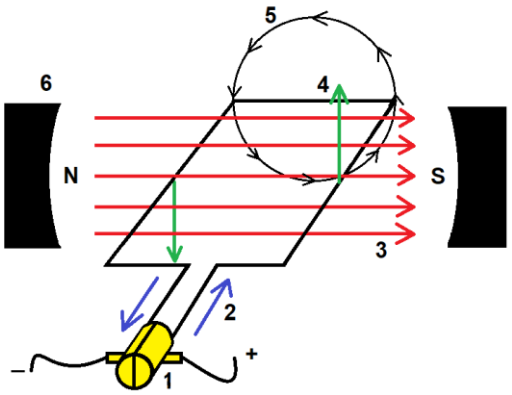


Figure 2.4 Conceptual function of a direct current electric motor. 1: The brushes contacting the commutator. 2: Current moving through the commutator winding. 3: Magnetic field. 4: Resulting force on commutator winding. 5: Rotational direction. 6: Armature supplying the magnetic field. Note the split at the end of the commutator. This is what switches current direction (and thus force direction) after 180 degrees and allows the rotation to be continuous.

The armature provides a static magnetic field, either through a permanent magnet or an electromagnet. Situated in this field is the commutator, which is made up of one or more conducting wire windings that can freely rotate in the field.

The ends of the commutator windings have terminals. Current is supplied to these terminals via energized brushes that drag along the shaft, keeping constant contact. When current is passed through the windings, a force starts acting on them. This force produces rotational motion.

In the simplest case, there is only two terminals and one winding. This means that after 180 degrees of rotation the polarity of the windings are switched due to the brushes switching terminals. Conceptually, this can be thought of like the donkey chasing a carrot that is constantly moving away from it. The commutator is chasing equilibrium within the magnetic field, but the polarity of the windings is switched just before it gets there [3].

The two advantages of a brushed DC motor are low cost and ease of use. The low cost derives from the fact that they are manufactured in high volume which drives down price.

As previously mentioned, a brushed DC motor only requires two leads to operate. The rotational direction can be shifted by changing the polarity of the leads. In addition to this, the rotation speed is directly related to the average voltage over the motor. Then, if voltage pulse width modulation is used, speed control is achieved.

A disadvantage with brushed DC motors is that it cannot be controlled into a position without any position tracking device. It needs feedback to be used in a control system.

Additionally, durability is a problem; constant grinding means the brushes get worn out by time and eventually they need to be replaced. If these brushes are not easy to replace the life cycle of the motor will be short. Short lifecycles waste energy and resources which in turn may have a negative effect on the environment. From a customer perspective it instils a feeling of bad quality, this is undesirable. A possible solution could be stepper motors.

Whereas a brushed DC motor requires a physical connection to actuate the commutator, a stepper motor does not. This makes it of interest from a sustainability perspective. In addition to this, a stepper motor opens the possibility of open-loop control. Here follows a description of the function of a stepper motor.

There are several types of stepper motors. When referring to a stepper motor it is of value to know what type of stepper motor is meant. This project concerns itself with the type known as a hybrid stepper motor. For other types of stepper motors see: [4, pp. 1 - 10] and [3, pp. 315-343].

When thinking of the function of a stepper motor it is helpful to think of a compass. In the same way, the compass needle aligns with the direction of the earth's magnetic field, the commutator of the stepper motor aligns with the magnetic field created by the armature. By altering the orientation of this magnetic field, the orientation of the commutator can be shifted [3].

What sets a stepper motor apart from other motors is that it moves in discrete steps. How many steps vary but a common number is 200 steps per rotation. Instead of being fed a continuous current, the motor is actuated by pulses. Each pulse corresponding to exactly one step. This makes them ideal for being controlled by a microprocessor in an open-loop configuration; see Section 2.2.

2.4 Angular Position Sensors

A change in the angular position is to be converted into a readable signal. There is a myriad of ways in which to achieve this. However, this project concerns itself with Hall effect sensors and potentiometers.

Hall effect sensors detect magnetic fields. When a conducting material is introduced to a magnetic field, the electrons passing through the conductor is pushed to one side of the conductor. This creates a detectable difference in potential that is proportional to the strength of the magnetic field. [5, pp. 119 - 123]. See Figure 2.5 below.

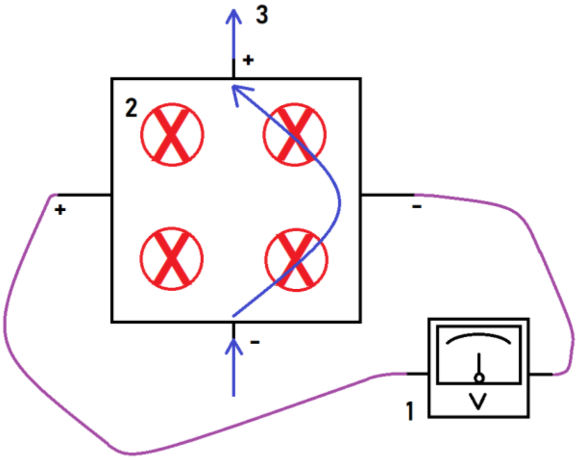


Figure 2.5 The hall effect sensor measures the proximity and/or magnitude of a nearby magnetic field. 1: Voltage gauge measuring the potential difference across the conducting material caused by the magnetic field. 2: Flow lines of the magnetic field passing through the conducting material. 3: Electron flow running through the conducting material.

As stated in [5, p. 357] Hall effect sensors can be used as frictionless angle sensors. Several sensors are connected in a bridge configuration. When a magnet rotates on top of this bridge the difference in potential across the different sensors is converted into a digital signal and interpreted via software as different angles.

The underlying principle of potentiometric sensors is that of the linear relationship between a conductor's length and its resistance. By varying the length of a conductor, it is possible to vary the resistance and therefore the voltage over it. This is used to measure displacement. The potentiometer is mechanically connected to an object whose position is to be measured. Then any movement of the object will result in a change in voltage. This voltage can be measured and used to control the process [5, pp. 336 - 340].

2.5 DISTANCE SENSORS

Information about sensors used to get a position.

- **Camera-based system** – With a camera, using the focus and image processing the distance can be calculated. With an algorithm, face recognition and known position in the space of the camera the coordinates for the eyes can be known. This solution does not require moving parts. It can be difficult to use in dark rooms; this can be worked around using night vision cameras.

  One camera requires several pictures taken and a complicated algorithm, if two cameras are used depth vision can be achieved like human eyes.

  Smart Eye is a company based in Gothenburg [6]. They have developed a system used in vehicles today that detects the eyes, the eyes movement and position. Their goal is safety, alert the driver when the driver is not focused and even more features. Their sensors work with advanced camera algorithms and give the desired information to calculate the angle on the mirror, however, their sensors have a price that was out of budget for this thesis and will therefore not be used here. Note that there are other companies with a similar type of sensors, why Smart Eye is mentioned is that they were contacted to confirm that their sensors give the desired information for the position of the eyes.

- **Ultrasonic** – Ultrasonic sensors use sound to calculate the distance to the nearest object. The logic behind this is producing a short period of sound in a high frequency, wait until the sound has bounced back and measure the time between signal is sent and received. This is the time that the sounds take to travel to the object and back. In other words, take the speed of sound in air, multiplicate with the time measured and divide by two to get the distance to the object. Ultrasonic sensors are comparatively cheap, and they have high accuracy on a short distance. Therefore, a good option for prototypes and easy applications where limitations allow the uncertain location of the object. For example, you get the distance to the nearest object, but you do not know that it is the right object as you are able to do using a smart camera, image processing or LIDAR algorithms.

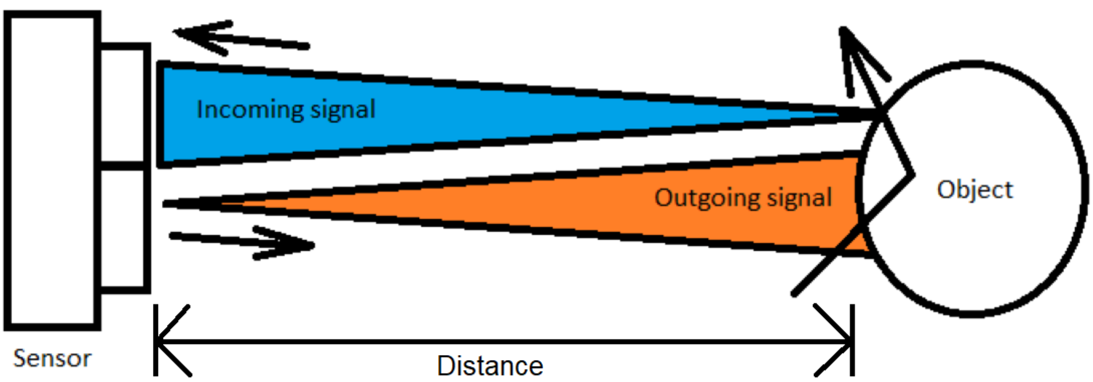


Figure 2.6 Conceptual function of an ultrasonic sensor.

Summarising the pros and cons on the sensors, the best sensor in performance for distance measurement in a stationary environment would be to use a camera-based system with algorithms. Smart eye provides advanced sensor-technology, but it is also a very pricy choice.

Ultrasonic sensors are bad for advanced projects with the disadvantage of no consistent distance in different temperatures and no knowledge on what object is the measured distance. This uncertainty makes it a bad choice for system that wants to find the distance to a specific object such as the eyes. Pros are the price; you can build concept and simulate different distances by using a flat object and move it around. If the goal is to just prove a concept this is a good choice.

3 METHOD

In this chapter is a description of the logic behind the system, an explanation of how the angle is calculated, and how the ultrasonic sensors are implemented.

There is also a description of the code used, for a more detailed view on the code see appendix where a commented version of the code is present. All code is written in C. Some of the code has been partly copied from projects found on the internet and changed into what suits the specific hardware.

3.1 ELECTRIC SYSTEM

This section covers all the specific electrical hardware used in the project. It begins with a figure describing an overview of the system and then precedes to explain the different parts of the system in detail. See Figure 3.1.

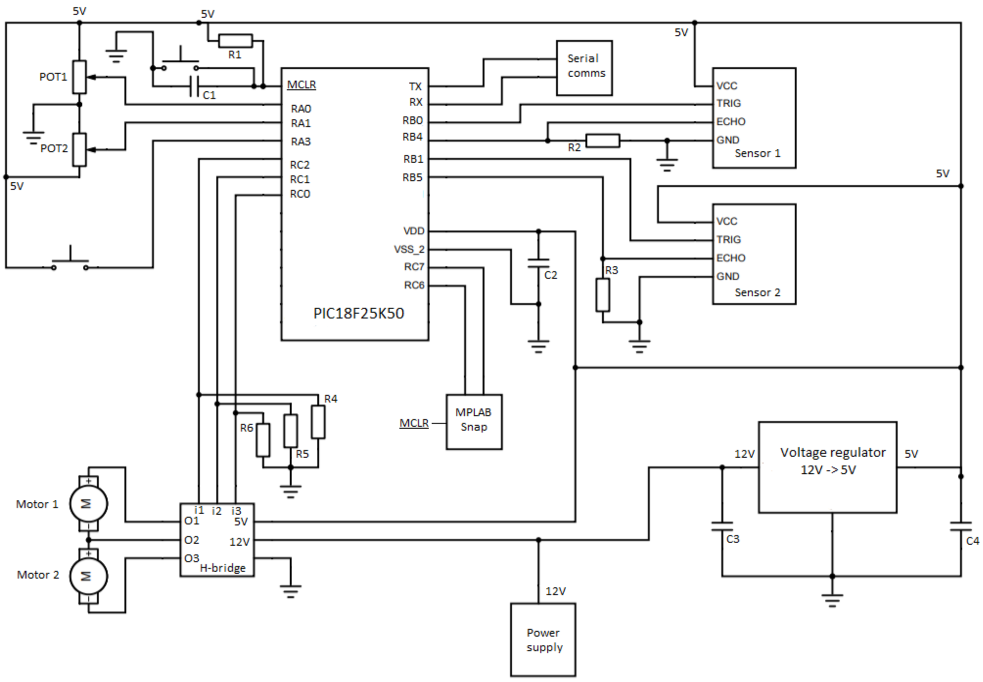


Figure 3.1 Circuit diagram depicting the system

- **ECU** – The control unit used in this project a PIC18F25SK50. It has an ANALOG to DIGITAL function that is needed for the potentiometers in the mirror as well as an internal oscillator with ±1\% accuracy that is used to measure the time for distance calculation with the ultrasonic sensors. Why this was chosen is because it was compatible with MPlab Snap Programmer which is a simple way to program the microprocessor and debug while the processor is in the circuit. The processor is built on an 8-bit architecture and has 28 pins, 25 of which are I/O-ports. Of these 25, 14 have an AD-channel option and the rest is digital [7].

- **H-bridge** – To control a set of 12V DC motors in multiple directions with a 5V microcontroller, a transistor H-bridge is needed. Three 5V signals are supplied to it from the microcontroller and three 12V signals go out to the motors. These signals form a 3-bit number and depending on the value of this number the motors are controlled in different directions. A 3-bit number was chosen instead of 4-bit one since the motors have a shared ground. By the logic of this 3-bit number: 0bxxx, there are three different states that the motors can take. Forwards, backwards and stationary. The lowest bit is used as ground, set this to 0 and you have 0bXX0. By setting one of the bits to 1 you drive one engine forward. Either 0b010 or 0b100. If you want to reverse and engine you need to switch poles, either 0b101 or 0b011. With the microcontroller connected to the H-bridge, you can set the bits to 0 or 1 for each wire on the mirror. The specific H-bridge used in this project is the L293D.
- **Ultrasonic sensors** – The specific ultrasonic sensor used in this project is the HC-SR04. It can measure distances between 2 and 400 [cm] and is powered by 5V.
- **RS-232 To TTL Converter MAX3232IDR** – Enables usage of RS232 serial communication. This is used to log data from the process to a PC.
- **L7805CV voltage regulator** – For this project two voltage levels, 5V and 12V, are required. To avoid using two power supplies an L7805CV voltage regulator was used. It takes input voltages up to 35V and reliably outputs 5V.
- **Mirror assembly** – The mirror assembly used is taken from a 2003 SAAB 9-5. It contains two 12V DC-motors and two 5V potentiometers. In appendix, a circuit diagram for the mirror is attached.
- **Power supply** – The power supply used is a Manson SPS 9250. With a range of 3-15V and a max output current of 25A, it was deemed to more than enough. The components in this project need a maximum supply of 12V at 0.3A.
- **MPlab snap** – The MPLAB Snap is used for uploading and debugging of the code. Thanks to this device it is possible to debug the code in-circuit. When the code is confirmed as ready the PIC microcontroller can be programmed to work without the MPLAB Snap. The MPLAB Snap is compatible with many PIC microprocessors made by Microchip Technology Inc. The choice was made by communication with Hult, Göran. G. Hult is an instructor in Automatic control at Chalmers at the department of electrical engineering.
- **Electronic components** – Components as resistors and capacitors are used for stable current supply and protecting components. Switches are used to give signals.

3.2 BUILDING THE TEST RIG

The test rig has 4 key functions. To be able to read a position that is used as a reference for the angle calculations, changing the position for a new calculation, control the mirror to the calculated mirror angle, and lastly measure the field of view.

For the position reading, two ultrasonic sensors are used. In Figure 3.2 they are marked as 1 and 4. These two sensors give the position of the driver in relation to the mirror. For x-position it is fastened straight ahead of the object in the centre of the chair, for z-position, it is fastened on an aluminium rail above the object. The rail allows it to be above the object no matter where the object is positioned in the x-direction.

To simulate different driver lengths and heights the seat was adjusted to different settings. In the y-direction, a 16 cm high box was placed on top of the measuring area. For the x-direction, the seat's rails were used. They had a range of up to 19 cm.

A mirror is mounted on the rig, this mirror has angle sensors that give feedback to the control system. The distance between the mirror and sensor are measured in y-direction and z-direction. These values are used as constants in the control algorithm. This mirror is represented as nr 5 in Figure 3.2.

For an indication of the field of view, a laser pointer is used. The laser is mounted on a tripod in line with the mirror, slightly behind the rig. The position of the laser is meant to simulate the object/area that the driver wants to see, and the laser point itself is meant to represent the centre of the driver's line of sight. The laser is bounced off the mirror and directed towards the driver's seat. See nr 3 in Figure 3.2.

In the seat, a box is placed with distance markings on it. These markings are used for validation of the system. To simulate that the driver's eyes are halfway down the face a point was marked 4 centimetres down from the top of the box. See nr 2 in Figure 3.2.

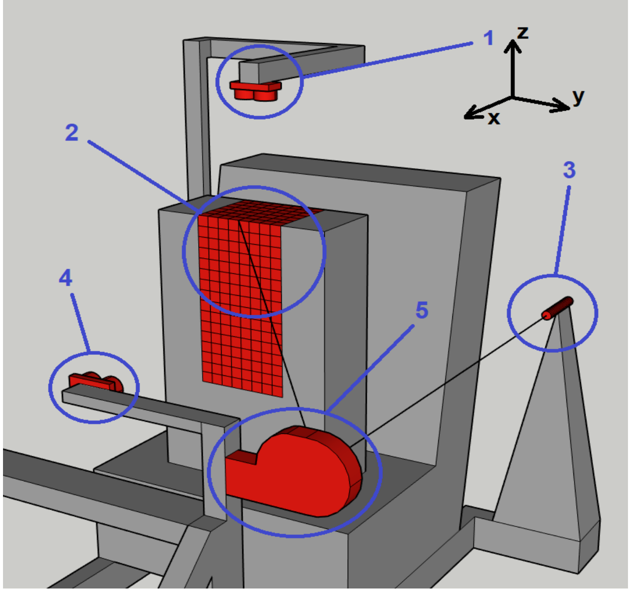


Figure 3.2 Conceptual overview of the system. 1: Vertical distance sensor. 2: Measuring object. 3: Laser pointer that simulates line of sight. 4: Longitudinal distance sensor. 5: Mirror assembly containing a mirror, two actuators and two angle sensors.

3.3 ANGLE CALCULATIONS

This section explains the mathematical formula that gives the desired angle for a given driver position.

First it was realized that the mirror could perform yaw and pitch movements separately. This allowed the problem to be divided into two separate two-dimensional problems. An additional important constraint that made the problem easier to solve was to only look at objects in line with the rotational point of the mirror. This meant that the path of the light could be interpreted as two right angled triangles. This is shown in Figure 3.3 and Figure 3.4.

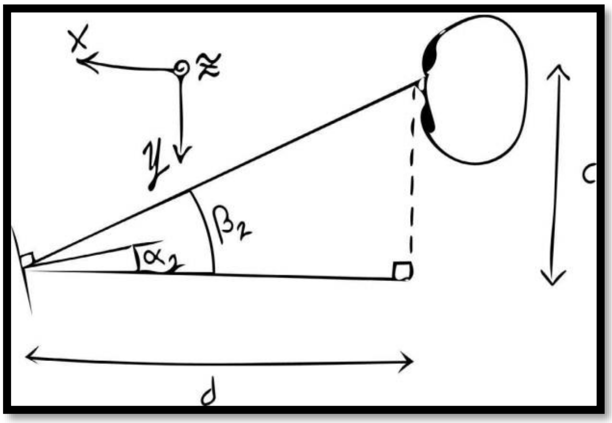


Figure 3.3 Describes the driver as seen from the top.

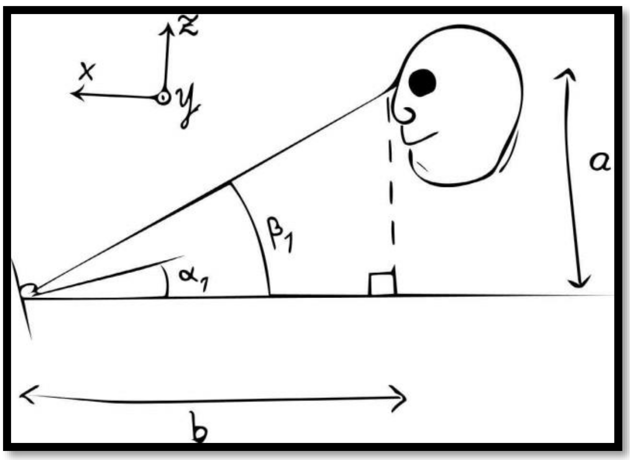


Figure 3.4 Describes the driver as seen from the side.

It is known that the angle with which a ray of light hits a plane mirror, is equal the exit angle. If then the mirror is positioned so that the exiting ray of light forms the horizontal line, then the angle α will be equal to half β. The angle β can be calculated if the distances a, b, c and are known. This describes a mathematical relationship between the driver's position and the desired angle of the mirror.

$$\alpha 1 = \frac{arctan\left(\frac{a}{b}\right)}{2} \tag{1}$$

$$\alpha 2 = \frac{arctan\left(\frac{c}{d}\right)}{2} \tag{2}$$

3.4 DISTANCE MEASURING ALGORITHM

This section aims to describe how the information gained from the ultra-sonic sensors is interpreted and then processed. The sensor sends out an ultrasonic pulse and starts a timer. When the pulse returns to the sensor after being bounced of an object, the timer is stopped. This timer counts with the same frequency as the internal clock. To get the elapsed time that the pulse travelled, the timer count is divided by this frequency. This, in turn, is then multiplied by the speed of sound in centimetres per second. This gives us the total distance travelled by the pulse. This total distance is divided by two and the result is the actual distance to the measured object.

The c-function "distance()" is for the ultrasonic sensors. In this function, a trigger signal is sent out in periods with a delay for 10 microseconds, the timer starts and measures the time until an echo signal is received. Further, it is now known how long time it takes for the sound to travel to the object and back. By using the speed of sound in air, about 34 300 cm/s (20 °C), the distance can now be calculated using (3). Note that this speed is variating depending on temperature, humidity, and air pressure so this will give an approximate end value that has to be tweaked.

$$d = \frac{34300 \times Time}{2} \tag{3}$$

In (3) d = distance in cm, because speed is cm/s. Equation (4), (5), and (6) describes how Time is calculated with the microcontroller. This project uses an 8-bit microcontroller with a 10-bit timer. To accomplish a 10-bit timer with an 8-bit system, 2 registers are being used. TMR1L and TMR1H stand for time module register 1 low respectively high. 8 bits from TMR1L are being used and 2 bits from TMR1H. In (4) Prescaler is a multiplication of the system clock speed, “$TMR1H\!:\!TMR1L$” means that both registers are being used as described with 8 bits from the low register and 2 bits from the high. The time module can be found in the datasheet [7, p. 156]:

$$Time = (TMR1H\!:\!TMR1L) \times \frac{1}{Internal\ clock} * Prescaler \tag{4}$$

$$Internal\ clock = \frac{Fosc}{4} = \frac{16MHz}{4} = 4MHz \tag{5}$$

Fosc = Frequency of the OSCillator. The Internal clock in (4) is described in (5). Put (4) and (5) together for an expression of “Time”:

$$Time = \ (TMR1H\!:\!TMR1L) * \frac{2}{4000000} \tag{6}$$

Using (3) and (6) the following expression is gained:

$$d = \frac{TMR1H\!:\!TMR1L}{116.61} \tag{7}$$

Later when tested, it was found that the constant of 116.61 was too small because the distance measured with the ultra-sonic sensor was too long compared to reality, growing with distance. By measuring a known distance 116.61 was increased until the measured value matched reality. The final value that was settled with was 120.647. Note that this is not the optimal distance measurement, this was a compromise so the project could continue.

3.5 PROGRAM TO CONTROL THE MIRROR FOR A TEST

To log information about the mirror a small program was constructed in the microprocessor. The program begins with initialisation, in these initiations, the bits 0-3 in register A is set as digital inputs. In register C, bit 0-2 are set as digital outputs. In Figure 3.5 it is described how the buttons, 1-4 controls the mirror. In the appendix the code for this program is presented as “Test program”.

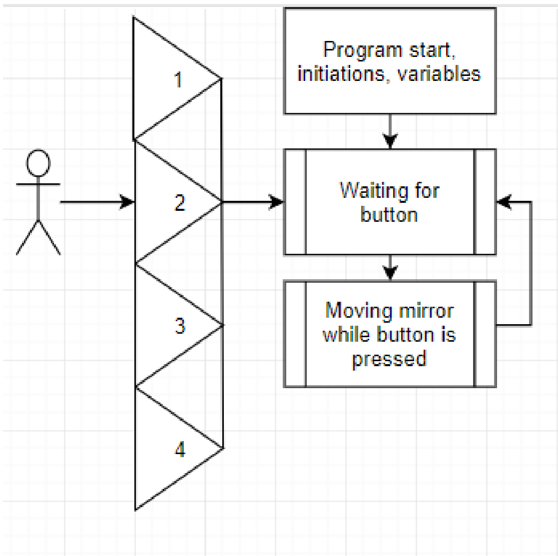


Figure 3.5 Flow chart describing the test program.

The A register is used for control inputs. Using 4 buttons, one for each direction that the mirror can be controlled in, the bits in register C are set for motor-control. The 4 cases are with the least significant bit to the right: 010, 100, 101 and 011.

The first case is controlling the mirror glass inwards, the second is upwards. Then by using the least significant bit as a shared line, the two other cases are the inverse from the first two. Meaning that in the 3rd case the mirror is moving outwards, and in the last case it is moving downwards. If no button is pressed, every bit is set to 0.

Serial communication was intended to be used with the microcontroller. After several attempts to establish a serial communication and failing, an Arduino was used for the logging instead. The code for the Arduino is choosing the analogue PIN in use and set up serial communication with baud rate 9600. Then into a loop where value is being read repeatedly and sent to PC using serial communication. Baud rate can be chosen at different speeds, the important part is that the terminal is set-up to the same speed for a correct data reading.

3.6 THE MAIN PROGRAM

In Figure 3.6 the main program is described. After the start-up, the program runs continuously. It waits for a signal to start the control session and returns to the waiting point after it is performed.

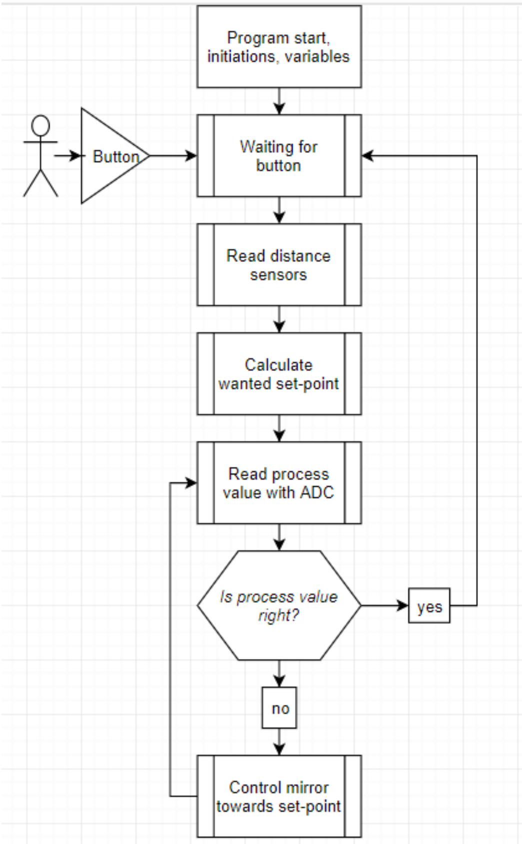


Figure 3.6 Flow chart describing the system algorithm.

The control algorithm starts with reading the sensors. The distance sensors are used to calculate the position of the eyes. This position is used to calculate the wanted angle on the mirror. This value is stored in a global variable and is used for the regulator as a set-value.

Using analogue to digital conversion (ADC) the current value on the mirror is read. If this value is too big compared to the set-value, the mirror is reversed for that axis, the opposite if it is too small. The same procedure is done for both axes individually, one value for mirror up and down, one value for the mirror in and out. This is a two-stage regulator as described in Section 2.2, this was chosen because the motors for the mirror has a constant speed and can only be turned on or off.

For easier reading in the appendix, these are the function names and short description of their usage.

- void **init**(void);
- signed int **AD_omv**(char ADkanal);
- int **distance**(int a);
- void **borverdepot**(void);
- void **motor**(void);

The function init starts up the system with everything needed for it to work. Setting clocks, input/outputs, enable needed functions and declaring variables used.

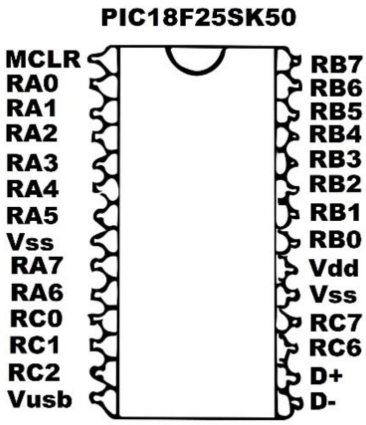


Figure 3.7 Describing the pin layout of the PIC18F25SK50

The AD_omv function is called with a number, the number is which PIN in the PORT-A register that is being used. For example AD_omv(1); does analogue to digital conversion (ADC) on PIN RA1. This is called for PIN RA0 and PIN RA1 which are connected to the potentiometers in the mirror, see Figure 3.7. This gives the process value of the mirror. This function was copied from a previous project at Chalmers and the function is written by Hult, Göran [8, p. 43].

The distance function is described in Section 3.4.

The function borvardepot is a function that uses the distance function to calculate the desired value on the potentiometers in the mirror. Using (1) and (2) described in Section 3.3 Angle calculation, it sets a global variable usable with the ADC for each potentiometer.

The function motor() is the regulator of the program. Using the logic of a two-stage regulator with reverse function as explained in Section 2.2. It uses the global variables that are set by borvardepot.

4 EVALUATION

During this project three test were performed to calibrate and test the system. First a test was conducted to see how well the distance measuring algorithm performed. Based on the results it was further calibrated.

Next up was a test to find the relationship between the measured voltage over the potentiometers and the angle position of the mirror. This was a crucial test since the result dictated how the control signals would be dimensioned. In other words, what voltage the set value needed to be if a certain angle was required.

Finally, a complete system test was performed. A position was set, the system was activated and reacted to the new position, and when it was done the result was analysed.

4.1 DISTANCE MEASUREMENT TEST

Since accurate measurement of distance is required, verification and testing of the distance sensors had to be done. To understand how the distance measuring test was preformed, it is first necessary to understand how distance is calculated. This is described in Section 2.5. In Section 3.4 it is described how the distance algorithm is implemented with available hardware.

These calculations can be expressed as a time divided by a constant. The distance test was meant to test the accuracy of this constant and to test how the sensors reacted to different surfaces.

Different types of objects were placed at known distances in front of the sensors. The measurements were taken, and the system's calculations were compared with the actual distance. The constant in Equation (7) was changed until a correct reading was achieved.

4.2 POTENTIOMETER GAIN MAPPING

A potentiometric angular position sensor converts an angle into a voltage. To be able to control the mirror assembly correctly, it was important to understand the exact relationship between the angular position of the sensors in the mirror assembly and the voltage that they showed. Table 4.1 contains a table from the mirror assembly manual describing this relationship. To evaluate the preciseness of this table a so-called gain mapping test had to be performed.

Rörelsevinkel (°)	kOhm	V
1,5	0,3	0,4
4,5	0,8	1,1
7,5	1,3	1,8
10,5	1,8	2,5
13,5	2,3	3,2
16,5	2,8	3,9
19,5	3,3	4,7
21	3,7	4,9

Table 4.1 Table describing the relationship between position angle, resistance and voltage for the potentiometers.

The test was performed by steering the mirrors between their respective endpoints while logging their corresponding voltages. Analogue voltages were converted into digital values using an analogue to digital conversion. The resulting data set was sent to a computer using serial communication. Then this data was plotted using MATLAB. See Figure 4.2.

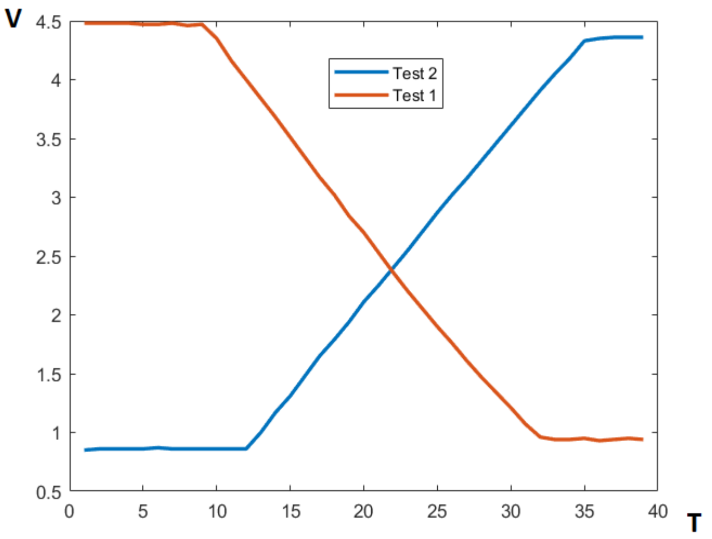


Figure 4.2 Chart depicting the gain mapping of the two potentiometers.

Test 1 in Figure 4.2 is from inner position to outer position while test 2 is from the upper position to the position furthest down. The V-axis is voltage and T is the number of plots. While being linear the maximum and minimum values differ a little between the two potentiometers.

4.3 COMPLETE SYSTEM TEST

To verify the success of a system, it is first necessary to define what constitutes a failure. The goal of this system is to adjust the mirror so that a desirable object/area is in view for the driver. In the ideal case, the system would work flawlessly. However, since this system is meant to be a proof of concept and not a finished product, some tolerances must be allowed. In this case, the tolerance is represented by how far from a centre point of the test object the laser is directed.

The estimated distance between a driver's eyes is within 15 cm, and since the previously mentioned laser represents the centre of a driver's field of view the laser point should not be allowed to deviate more than 7.5 cm from the centre point. This means that at least one eye would have a clear line of sight over the desired field of view.

Each test begins with some adjustment of the seat (and therefore the object), back or forth, up or down. Then the distance sensors are adjusted accordingly. The button is switched, and the measurements begin. For more information on how the test rig works, see Chapter 3.2.

To tweak the results two constants were used. Gain and offset. The calculated angle is multiplied with the gain K and added to an offset M.

5 RESULTS

After repeated tests and calibrations, the system started behaving in a repeatable and reliable way. Figure 5.1 shows the first measurement, done at the lowest point in the forward position. Then the seat was moved to the back position while keeping to the lowest height position. The result of this measurement is shown in Figure 5.2. What is interesting is how similar the position of the laser dot is between the two figures, the closer to each other the better the result. Both are close to the central axis and at the same height, less than 1cm radius from each other and this test is considered a successful result.

In the second test, an additional box was placed on top of the box to simulate a taller driver. Figure 5.3 shows the results of this measurement. Compare the position of the dot between Figure 5.2 and Figure 5.3, what should be interesting is how well the system controls the dot in the z-axis, it should not change in the y-axis. The system handles the z-axis well, it is close to the same height of 4cm from the top in both tests, however, the dot has moved in y-axis which is not wanted. Looking at the result, it was found that the box is slightly tilted which makes the result look worse than it is, the result from this test is considered okay with room for improvement.

For the next test, the seat was then moved to the forward position again while keeping the tall setting of the box and Figure 5.4 shows how well this control worked. To understand how well this position is, compare it both to Figure 5.1 and Figure 5.3. Compared to Figure 5.1 it shows how well it handled the z-axis, here it is seen that the dot is a little bit lower in Figure 5.4 from the top than it was in Figure 5.1. The same problem that was noticed in the earlier test in the y-axis is present here as well. This result is the worst result obtained; it is okay but needs to be improved. Comparing Figure 5.3 with Figure 5.4 it is the y-axis that should be controlled. Here the result was good.

The results showed that the system handled difference in position for the control object in the x-direction well. This meaning, the mirror will have a good sight field in side-ways. For changes in the z-direction, the system struggled more. This meaning that the sight field would at best be considered okay in the height control for the mirror.

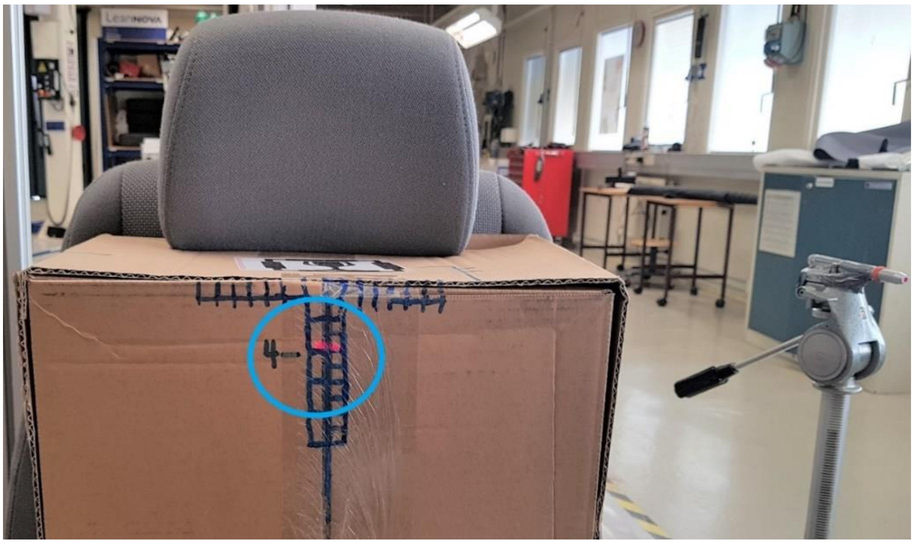


Figure 5.1 Seat moved to the front, without top box

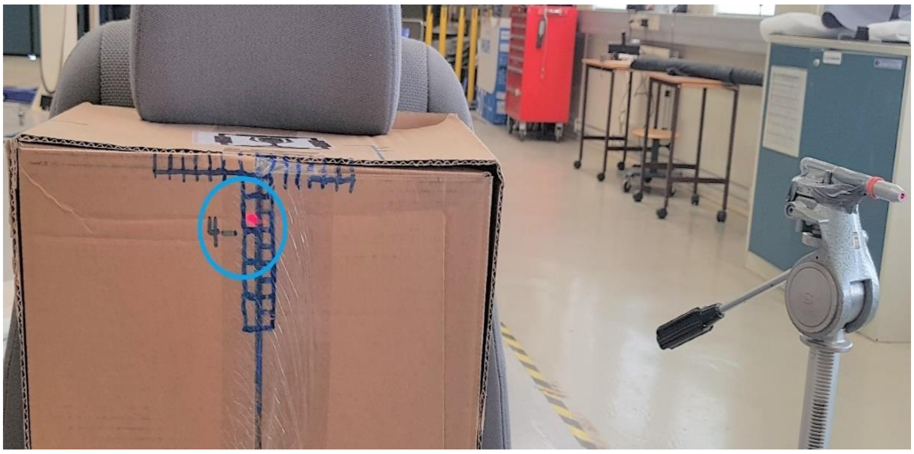


Figure 5.2 Seat moved to the back, without top box

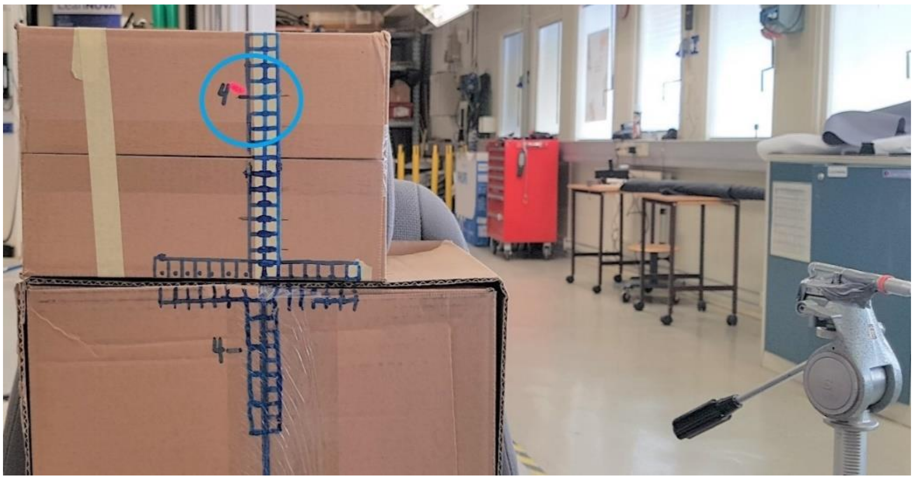


Figure 5.3 Seat moved to the back, with top box

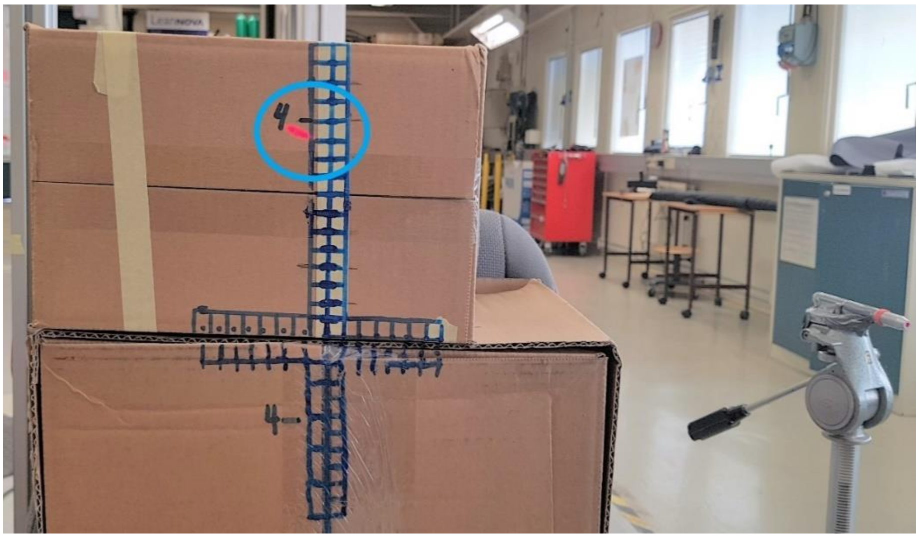


Figure 5.4 Seat moved to the front, with top box

6 CONCLUSION

While there is room for improvement both in hardware and software what this thesis does is proving the concept. It shows that it is possible to implement this kind of system in existing vehicles with existing components. Even if it is not possible to completely replace the manual control of the mirror at the time. Instead, this system can be used as a “fast setting” to give an approximate setting for every mirror. This makes it fast and easy to just tweak the mirrors a little for the wanted view.

The use of potentiometers in the mirror is the obstacle that causes the biggest drawbacks. Feedback is essential to get a good control algorithm. With the use of potentiometers, you get uncertain values from mirror to mirror due to variety of hardware, dirt and oxidation that can cause bad connections and with an ageing factor where the resistance may not be consistent throughout the years.

Is this something a manufacturer is prepared to accept? Is this something a customer is prepared to accept? To change the components in the mirror there must be a big market for it, otherwise, the producers of mirror actuators will not profit from developing an alternative position control for the mirror glass.

Alternatively, if the market is small enough each unit could be specially built for those exclusive vehicles where this can be wanted, for example, police cars which already possess some specially designed equipment. Doing small batches will make the price high and market research must be made if it will be worth it.

In conclusion, it has been interesting to research the possibility of a system for automatic rear-view mirrors. With promising results and positive interest from people spoken to, we hope that this will be further developed and implemented in vehicles. If this prevents any accident the system is successful both in ethical and environmental aspects.

6.1 LEARNINGS

- **Sensors not working on humans** – Through testing, it was found that the ultrasonic sensors do not give accurate readings when measuring surfaces that are not flat. This presents a problem when the purpose is to measure human body parts which tend towards convex shapes
- **Problem with serial-communication** – Attempts to establish serial communication between the microprocessor and computer failed. The reason for this is unclear. Perhaps it was due to the choice of components, or it may have been a software issue. This was never solved, to work around this an Arduino was used for measurement logging instead. The learning being that if one solution does not work it can be worked around to save time.
- **A fault that grows with displacement** – In testing, it was found that the laser dot would deviate from the anticipated spot. This deviation depended on how far the measuring surface was positioned in relation to the sensors. In other words, there was a relationship between object position and laser dot drift from the centre. The hypothesis is that the laser was not correctly aligned with and centred on the mirror.

- **Internal alignment the laser pointer** – A problem faced with the laser pointer was its internal alignment. While the laser beam itself travelled straight the exit angle would change with its rotational orientation.
- **Laser tripod position** – The tripod that the laser was mounted to was not permanently positioned. If it were pushed the position of the laser point would change. This meant that the system had to be recalibrated to fit the new position every time the tripod was moved. The solution to this would be to fasten the tripod to the rig. This means that the laser would only move if the rig moved and the error would be corrected for.
- **Mirror not horizontal due to bad mounting** – It was found that the laser point reflected by the mirror would travel in both the y- and x-direction, even though only one motor was actuated. The hypothesis being that mirrors axes were not sufficiently aligned with the rig.
- **Convex vs flat mirror** – The “flat” part of the mirror had slight convex properties. This presented further problems since the deflection angle would change in a unilinear fashion when the orientation of the mirror changed, due to the parabolic shape of the mirror. This compounds with the issue of the laser not being completely centred.
- **Poor measurements** – Some measurements on the rig was measured with basic tools. This means that there is room for better measurement in further development and better measurement would make it possible to have a more accurate control system.
- **Rig beams/profiles not being properly placed** – The aluminium beams/profiles that make up the rig were not, in some cases, squarely and securely fastened. This presented a problem mainly in the measuring of the y-coordinate. When the sensor was moved the beam would pitch up or down. So not only would the position of the sensors in the y-direction vary with the x-coordinate, due to the pitch, the pitch itself would change between different tests.
- **Ageing hardware** – As seen in Section 4.2 Potentiometer gain mapping, the hardware of the mirror using potentiometer’s for feedback will not be consistent for the whole lifecycle of the system. This problem can be worked around in some ways. One example is using different hardware that does not have this problem with ageing and bad connection, for example, stepper motor or Hall effect sensors. Where the main advantage would be the comparative lack of mechanical wear. Another solution to this problem would be to program in a calibration cycle in the system, what this does is moving the mirror glass from end to end, mapping the voltage and then divide the voltage by the angle in degrees that the mirror glass can be moved. This is a problem that needs to be kept in mind for further development of the system but should not be that big of an issue to solve.

6.3 FURTHER DEVELOPMENT

Some examples of how further development could be performed.

- **Better sensors** – A solution to the issue of the inexact sensors used, would be to use a more refined way of distance measuring. For example, a combination of a camera and face recognition software, such as the Smart Eye system.
- **Implement in a real vehicle** – To know if this is a useful function it must be tested in a real vehicle. User test must be performed etc.
- **Better verification measurements** – Distance measurements used for verification was done with course estimation and high tolerances. More specifically measuring tape was used. This was done to save time. A more exact option would be to have a distance measuring laser that could be aligned with spirit-level gauges.
- **A better measurement object** – Due to the issue of measuring convex surfaces, a cardboard box was used as a measurement object. Due to the nature of cardboard, it proved hard to keep it squared and to add exact distance markings. This, of course, introduced problems when gauging the drift of the laser. In addition to this, it proved problematic to fasten the box to the rig. The solution that was reached was to pin it between the headrest and the seat. In turn, this caused the box to deform.

  The ideal measurement object would be two rigid boards/planes joined at a right angle, creating an L-shaped profile. Both surfaces should have a marked out coordinate system. It should be able to be manipulated in both the y- and x-axis, in discrete steps. In addition to this, there should be a locking mechanism in place so that it could be secured tightly, thus reducing the risk of it changing position unintentionally.
- **Mirror with stepper motor and Hall effect sensor** – Stepper engines and hall effect sensors operate on digital pulses. They also have the advantage of having fewer physical connections. This means that they will not wear out as fast and are more suitable for microcontrollers. This will give the possibility of better control.

  A problem with this might be the supply. Only a few manufacturers supply the whole car industry with actuators and sensors. And the current standard is brushed components [1]. According to industry experts, the whole industry demand would have to switch to provide enough of an economic incentive for the suppliers to move away from brushed components [1].
- **Combining with HUD** – A Head-up display option is available in some cars today. This HUD shows on the windscreen, a different driving position makes a different placement on the windscreen wanted. This can also be automatic and can be implemented in the same control unit.

REFERENCES

[1] Engineers, Interviewee, *Mirror engineers at AFRY and Volvo cars.* [Interview]. 12 Mars 2020.

[2] W.Bolton, Control Systems, 2002-01-30.

[3] B. D. Austin Hughes, Electric Motors and Drives: Fundementals, Types and Applications, 2013-04-05.

[4] P. Acarnley and L. I. o. E. Engineers., Stepping motors: Theory and practice, 4th edition., 2002.

[5] J. Fraden, Handbook of modern sensors: physics, designs, and applications, 2016.

[6] Smart Eye, "Smart Eye," 2020. [Online]. Available: https://smarteye.se/. [Accessed 2020].

[7] Microship, "Datasheet PIC18(L)F2X/45K50," [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/PIC18F2X_45K50-30000684B.pdf.

[8] G. Hult, Kompendium Mekatronikprojekt, Gothenburg: Chalmers, Department of Electrical Engineering, 2019.

APPENDIX

MAIN PROGRAM

```c
#include <xc.h>
#include <pic18f25K50.h>
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
// BEGIN CONFIG
#pragma config LVP=ON
#pragma config CONFIG2L=1, BOREN=OFF
//PWRT Disabled//nPWRTEN OFF
#pragma config FOSC=INTOSCIO //Intern oscilator
#pragma config FCMEN=OFF          //FailSafedisabled
#pragma config DEBUG=OFF, STVREN=ON
#pragma config PCLKEN=ON // Primary oscilator enable
#pragma congig CFGPLLEN = OFF, CPUDIV=NOCLKDIV
// PLL OFF NO DIV CPU
//END CONFIG
#define _XTAL_FREQ 16000000          //System freq
#define E 48.5
#define A2 53.5
#define Kx 0.125      //The gain for potentiometer 1 in the mirror
#define Ky1 0.09      //The gain for potentiometer 2 in the mirror, for the function closer to the front
#define Ky2 0.1093     //The gain further from the front
#define M1 0.31        //offset for the mirror sideways, chosen by testing
#define M2_1 2.385     //offset for the mirror in hight in the function closer to the front, chosen by testing,
#define M2_2 2.5       //offset for the mirror in hight in the function further from the front, chosen by testing

//Function prototypes-------------------------------------------------

signed int AD_omv(char ADkanal);
void init(void);
int distance(int a);
void motor(void);
void borvardepot(void);

//Global variables
float borvarde_x, borvarde_y;
```

```
void main() {
    init();
    LATC=0;
    while (1)  {
        while(!PORTAbits.RA3);
//wait for start signal
        __delay_ms(1500);
//waiting 1,5 seconds to start
        motor();
    }
}
//Functions--------------------------------------------

signed int AD_omv(char ADkanal) {
    ADCON0=(ADCON0 & 0b10000011)|(ADkanal<<2);
//AD-channel
    ADCON0bits.GO=1; //AD-conversion begins
    while(ADCON0bits.GO){} //wait until ready
    return ((signed int)ADRESH<<8)+ADRESL; // AD result
}

void init()  {
    TRISA=0b00000111;
//PORTA bit0, bit1 and bit2 input, rest is output
    TRISB=0b00110000;
//PORTB bit 4-5 input for echo, rest is output
    ANSELA=0b00000011;
//PORTA bit0 and bit1 analog, rest digital
    ANSELB=0b00000000; //PORTB all digital
    ADCON2=0b10100100;
//Rightshifted,AD-clock=Fosc/4, TACQ=8us
    ADCON1=0b00000000; //Vref VDD and VSS
    ADCON0=0x01; //ADON

    TRISC=0;    //PORTC output

    T1CON=0b00010000;
//Enable TMR1,8-bit r/w,Fosc/4


    OSCCON=0b01110010;     //  intern oscilator 16Mhz

    //serieal communication
    TXSTA1=0b00100100;
    RCSTA1=0b10000000;
    BAUDCON1=0b00000000;
    SPBRG1=12;
}

int distance(int a)  {
```

```
        TMR1H = 0;
    //Sets the Initial Value of Timer
        TMR1L = 0;
        int b;
         if(a==1) {                    //Sensor 1
             PORTBbits.RB0=1;     //Trigger high
             __delay_us(10);
             PORTBbits.RB0=0;     //Trigger low
             while(!PORTBbits.RB4);  //Wait for echo
             TMR1ON = 1;             //Count time
             while(PORTBbits.RB4);   //Echo
             TMR1ON = 0;             //Stop timer
        }
        if(a==2) {                   //Sensor 2
             PORTBbits.RB1=1;         //Trigger high
             __delay_us(10);
             PORTBbits.RB1=0;         //Trigger low
             while(!PORTBbits.RB5);  //Wait for echo
      TMR1ON = 1;               //Start timer
      while(PORTBbits.RB5);     //Echo
      TMR1ON = 0;               //Stop timer
    }
        b = (TMR1L | (TMR1H<<8));     //Store both registers
        b = b/120.647; //the value to return in centimeter
        b = b + 1;    //remaining error
        return b;
    }
    void motor()  {
         signed int a, b;
         int i;
         a=AD_omv(0);                 //process-value POT1
         b=AD_omv(1);                 //process-value POT2
         borvardepot();
    //gives setvalue for potentiometers
         LATC=0;
         for(i=0;i<1;i++)  {
             __delay_ms(50);
             while(a<borvarde_x) {
    //if mirror is out from wanted position
                 LATC=0b00000010;              //inwards
                 __delay_ms(10);
    //Delay to give time for movement
                 a=AD_omv(0);
    //process-value for POT1, 3
                 if (a>800)  {
                     break;
                 }
             }
             LATC=0;
             __delay_ms(50);
```

```
        while(a>borvarde_x) {
//if mirror is inside from wanted position
            LATC=0b00000101;                //out
            __delay_ms(10);
//Delay to give time for movement
            a=AD_omv(0);
//process-value for POT1, 3
            if (a<250)    {
                break;
            }
        }
        LATC=0;
        __delay_ms(50);
        while(b>borvarde_y) {
//if mirror is over wanted position
            LATC=0b00000100;                //down
            __delay_ms(10);
//Delay to give time for movement
            b=AD_omv(1);
//process-value for POT2, 7
            if (b<250)
                break;
        }
        LATC=0;
        __delay_ms(50);
        while(b<borvarde_y) {
//if mirror is under wanted position
            LATC=0b00000011;                //up
            __delay_ms(10);
//Delay to give time for movement
            b=AD_omv(1); //process-value for POT2, 7
            if (b>800)
                break;
        }
        LATC=0;
    }

}

void borvardepot()  {
    int i;
    float A1, a=0, aavg=0, b=0, bavg=0;
    a=0;
    aavg=0;
    b=0;
    bavg=0;
    for(i=1;i<101;i++)  {
        while(90<a|a<28) {  //only reasonable values
            a=distance(1);  // sensor 1
        }
```

```
            aavg=aavg+a;
    //increase the total for average
            a=0;
    //resets a for new measurement
            __delay_ms(2);
    //delay to get ready for next measurement
        }
    a=(aavg/100)-3;
    //average, minus 3 because placement of sensor
        for(i=1;i<101;i++) {
            while(55<b|b<8) {
    //only reasonable values
                    b=distance(2);      // sensor 2
            }
            bavg=bavg+b;
    //increase the total for average
            b=0;
    //resets b for new measurement
            __delay_ms(2);
    //delay to get ready for next measurement
        }
        b=(bavg/100);    //average
        A1 = E - b;     // drivers hight

        borvarde_y = ((180/M_PI)*atan(A1/a))/2;
    //the global variable for angle in hight
        borvarde_x = ((180/M_PI)*atan(45/a))/2;
    //the global variable for angle in sideways
        if(a<50) {
            borvarde_y = (borvarde_y*Ky1+M2_1);
    //conversion to Volt
        }
        else {
            borvarde_y = (borvarde_y*Ky2+M2_2);
        }
        borvarde_x = (borvarde_x*Kx+M1);
    //conversion to Volt
        borvarde_x = borvarde_x-(0.001*a); //Linear error
        borvarde_y = borvarde_y+(2/a);
        borvarde_y = borvarde_y*(204.6);
    //conversion to int
        borvarde_x = borvarde_x*(204.6);
    //conversion to int
    }
```

TEST PROGRAM

```
void main()
{
     init();
     while(1)
     {
          //Pinnar: RC0=8, RC1=2, RC2=1
         if (PORTAbits.RA0)
         {
                 //in
                  LATC=0b00000010;
               }
               else if (PORTAbits.RA1)
               {
                  //down
                  LATC=0b00000100;
               }
               else if (PORTAbits.RA2)
               {
                  //out
                  LATC=0b00000101;
               }
               else if(PORTAbits.RA3)
               {
                  //up
                  LATC=0b00000011;
               }
               else
               {
                  LATC=0;
               }

           }
       }
```

MIRROR CIRCUIT