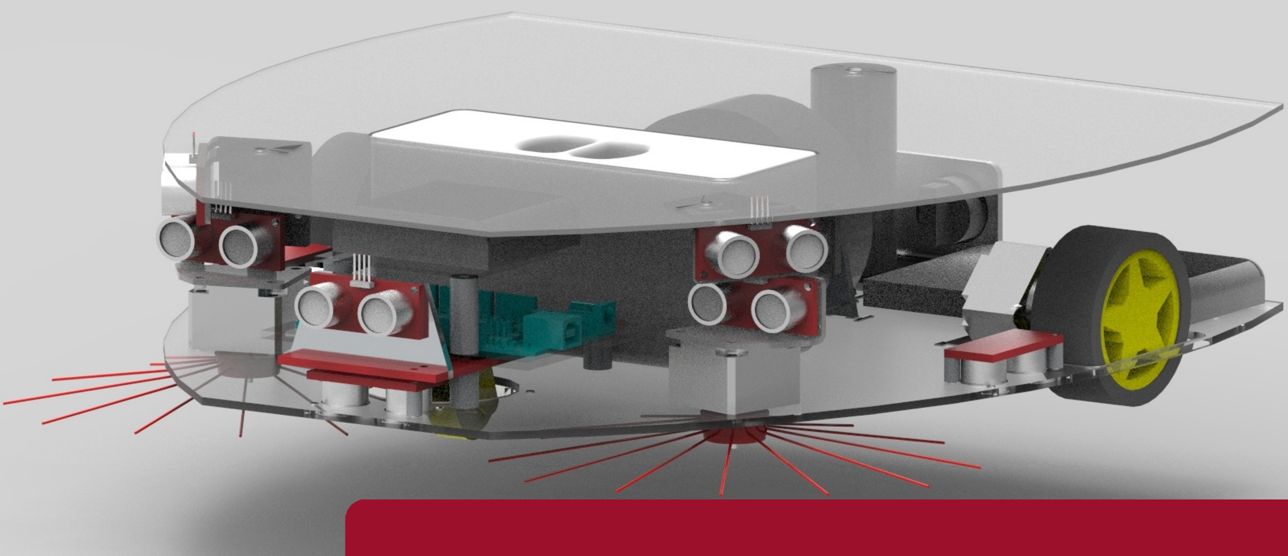
DEGREE PROJECT IN ELECTRONICS AND COMPUTER ENGINEERING,
FIRST CYCLE, 15 CREDITS
STOCKHOLM, SWEDEN 2019

Robot Vacuum cleaner

JOEL BERGMAN AND JONAS LIND



KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ENGINEERING SCIENCES

Robot Vacuum cleaner

Thesis

JOEL BERGMAN AND JONAS LIND

Kandidate Thesis at ITM
Supervisor: Nihad Subasic
Examiner: Nihad Subasic

TRITA-ITM-EX 2019:28

Abstract

Although a robot vacuum cleaner is a well-known product, development is still interesting. Better working sensors and more sophisticated algorithms and sensors are used in new cleaners.
The purpose with this thesis was to learn more about different sensors, algorithms, and designs of robot vacuum cleaners, to try ideas and to find improvements to implement on the demonstrator.
The initial work was to make a market investigation to find customer needs and expectations. Also making a rough design and layout for the mechanical and electrical system.
The budget for the demonstrator was 1000 SEK. Using laser cut plastic sheets, 3D printed parts, and receiving motors for free, costs did not exceed budget.
Literature and theses in the area of interest were studied to find answers to some of the research questions. The theses studied subjects such as driving pattern and designs to find back to charge station. Some ideas for driving patterns were implemented on a demonstrator. The development method used was iteration of finding useful information, testing components, codes and also the complete demonstrator.
The components used were DC motor, stepper motors, ultrasonic sensors, Arduino mega micro controller, switches and AA batteries. The different components required different voltage and the stepper motors used driver cards.

Keywords: Arduino, DC motors, Stepper motors, Ultrasonic sensors, Robot vacuum cleaner.

Referat

Robotdammsugare

Även om robot dammsugare är en välkänd produkt är produktutveckling fortfarande intressant. Bättre sensorer och mer sofistikerade algoritmer och sensorer används i dammsugare.
Syftet med denna avhandling var att lära sig mer om olika dammsugare, algoritmer och konstruktioner av robot dammsugare, för att prova idéer och möjliga hitta förbättringar att implementera på demonstranten.
Det första arbetet var att göra en marknadsundersökning för att hitta kundens behov och förväntningar. Även att göra en grov design och layout för det mekaniska och elektriska systemet.
Budgeteringen för demonstranten var 1000 kr. Kostnaderna översteg inte budgeten eftersom vi använde oss av laser skurna plastplattor, 3D-printade delar och erhöll motorer utan kostnad.
Sex olika avhandlingar studerades för att hitta svar på några av frågorna. Avhandlingarna studerade ämnen som körmönster och mönster för att hitta tillbaka till laddstationen. Vissa idéer för körmönster implementerades på demonstranten.
Den använda utvecklingsmetoden var iteration av att hitta användbar information, testa komponenter, koder och även den fullständiga demonstranten.
De komponenter som användes var likströmsmotor, stegmotorer, ultraljudssensorer, Arduino mega-mikrokontroller, strömbrytare och AA-batterier. De olika komponenterna krävde olika spänningar och stegmotorn använder ett specifikt drivkort.

Nyckelord: Arduino, likströmsmotorer, stegmotorer, ultraljudssensorer, Robotdammsugare

Acknowledgements

We would like to thank the following persons for all the help and good advices:

Nihad Subasic, Examiner at mechatronics KTH
Staffan Qvarnström, Teacher at mechatronics KTH
Seshagopalan Thorapalli Muralidharan,Assistant at mechatronics KTH
Lars Hässler, Prototype center, KTH
Tomas Östberg Labs and Workshops supervisor
Fredrik Lundell, Professor at fluid mexhanics KTH
Sebastian Lind, Student at Mälardalens Tekniska gymnasium

Also of course all of our colleagues at MF133X, Degree project in mechatronics, undergraduate level and other teachers and assistants that we have met during our education.

Contents

List of Figures

Nomenclature

Abbreviation

PCB	Printed Circuit Board
DC	Direct Current
AC	Alternating Current
PWM	Pulse Width Modulation
EMF	Electromotive Force
PPCR	Path Planning Coverage Region

Chapter 1

Introduction

1.1 Background

Robot vacuum cleaners are well-known products. Still there are continuously new products introduced on the market, products with new or improved functionality. Robot vacuum cleaners are mainly used in domestic areas for removing particles from indoor floors.
The aim of this thesis was to learn more about the design and its requirement, hopefully be able to improve some functions and finally have a functional prototype for testing.

1.2 Purpose

- How does a prototype made of cheap parts and controlled by an Arduino compare to vacuum cleaners on the market?
- How should sensor and code be designed to get a functional vacuum cleaner?
  - Position, type and number of sensors?
  - Driving pattern?
- How can a function that returns the cleaner to a battery charge station, when the battery charge level is low, be designed?

1.3 Scope

The main purpose with this thesis was to design a robot vacuum cleaner that can perform tasks that a regular vacuum cleaner, on the market, can do. During the design phase, ideas and different designs were tested. The most interesting ones were added on the physical prototype. Criteria for interesting designs were low cost, long drive time, design shape and dimensions. The budget for the prototype was limited to 1000 SEK. Therefore, parts with low cost were used, this did in some way limit the design of the prototype, for example no tooling for included parts were made. The robot was equipped with sensors to detect its surroundings and the dimensions were chosen so it can manage normal obstacles and clean in a domestic area.

1.4 Method

The first step of this project was to investigate the requirements for a robot vacuum cleaner. This step provided information about specifications that customers expect from the product. The result was an important input to the design work. The necessary parts included and dimensions could be determined.
A preliminary design was developed, and a first prototype manufactured. Depending on the function of the prototype the design process continued and improved the product. Finally, through numerous iterations, a functional prototype was developed.

Chapter 2

Theory

2.1 DC Motor

Electrical motors are common motors for powering a machine. They are often light, cheap and silent. Things to consider when choosing a motor are the values of revolutions/minute, power and start up torque. In some applications the revolutions/minute is too high on an engine and the torque is too low, in these cases mechanical gears are used. If the start up torque is to low it might result in that the engine does not start at all due to external loads such as gears and connected machines.

Electrical motors come in a variety of types. For motors powered by batteries at common type is direct current (DC) engines.

An advantage with DC engines is that rotation speed can be adjusted within a large span without losing efficiency. A disadvantage can be that they are more expensive than alternating current (AC) engines.

The motor factor, k, is a way to consider the increased torque necessary due to start up, impacts and irregularities from the driven machine. For electrical motors which drive fans the value of k is 1.4 [Dahlvig, 1999].

For electrical engines:

$$k < \frac{Mstart}{M} \tag{2.1}$$

The major part of the torque increase is because of the torque necessary to accelerate the motor.

In electrical motors there are different events. First is the electrical response which is in milliseconds. The mechanics are slower, within seconds it reacts. Finally it is the event when the temperature rises in the engine, this happens within minutes.

The temperature rise in the electrical engine follows a transient pattern.

The torque necessary for the engine can be described as follows [Johansson, 2013]:

$$M_m = M_{last} + M_{acc} \; [Nm] \tag{2.2}$$

where

$M_m = torque\ from\ electric\ engine\ [Nm]$
$M_{last} = torque\ necessary\ for\ the\ load\ [Nm]$
$M_{acc} = torque\ necessary\ to\ accelerate\ the\ system\ [Nm]$

where:

$$M_{acc} = J\frac{d\omega}{dt} \; [Nm] \tag{2.3}$$

At constant speed $M_{acc} = 0 \; [Nm]$

where the moment of inertia is [Apazidis, 2012]:

$$J = \sum_i r_i^2 m_i \; [Nm] \tag{2.4}$$

where r_i^2 is the distance from rotation axle to the mass m_i

M_{start} and M_{max} (maximum torque for the engine) is described in data sheets from engine supplier.

The components included in an electric motor are the rotor which consists of windings and the the stator that generates a magnetic field.

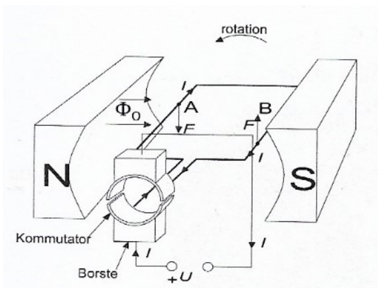


Figure 2.1. Sketch of the inside of a DC motor [Johansson, 2013]

When a current is flowing in the winding a magnetic field is induced. This field will, in the magnetic field from the stator, generate a force on the rotor that makes it rotate. The current direction is switching to maintain the force direction when the winding passes through the stator magnetic field and a rotation occurs. When the speed reaches a certain value the stator magnetic field will start to induce a opposite current flow in the rotor winding and the system will be in equilibrium. This engine state is refereed to as idling, U from the external source is the same as the emf induced. For a common DC engine the emf will be a square wave. Measured at connections of the rotor the emf (E) value according to the equation below. When idling, without load ($M_{axle} = 0 \; and \; I_A = 0$), $\omega = \omega_0$.

$$E = K_2\Phi\omega \; [V] \tag{2.5}$$

Measuring outside at the engine connections the anchor voltage U_A is

$$U_A = R_A I_A + K_2\Phi\omega \; [V] \tag{2.6}$$

where
R_A = resistance in the rotor windings [Ω]
I_A = current in the circuit [A]
$K_2\Phi$ = engine constant

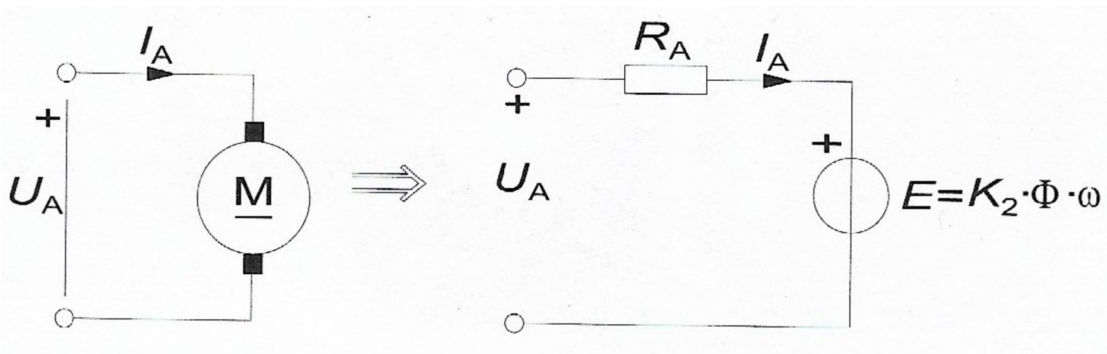


Figure 2.2. Motor circuit [Johansson, 2013]

Following equations are adaptions of the one above.

$$P_{axle} = I_A K_2\Phi\omega \; [W](Effect \; on \; output \; axle) \tag{2.7}$$

$$M_{axle} = I_A K_2\Phi \; [Nm](Torque \; in \; output \; axle) \tag{2.8}$$

2.2 Fan

There are different kind of fans, radial, axial and cross flow fans.

Radial fans are often used in ventilation installations and in vacuum cleaners. A rotating wheel transports air from the centre to the outer edge using centrifugal force. The bending of the vanes results in different function of the fans.Vanes bending backwards are self-cleaning and used to transport gases with dust. This fan type can easily be regulated by changing rotation speed.

Axial fans move the gas in an axial direction. They have lower efficiency than radial fans but can move large volumes of gas. The capacity can be adjusted with the rotation speed.

Cross flow fans suck air in compresses it and pushes it out in desired direction. Often used in ventilations when low noise level is a requirement.

Radial fans use centrifugal force. The difference in velocity between the inner radius Ri and the outer radius Ry can be transformed to a change in pressure.

Centrifugal force on a particle is described with the equations below.

$$F = \frac{mv^2}{r} = m\, r\, \omega^2 \tag{2.9}$$

$$F = m\, a \tag{2.10}$$

combined

$$a = \frac{v^2}{r} = r\, \omega^2 \tag{2.11}$$

$$F = \frac{mv^2}{r} = m\, r\, \omega^2 \tag{2.12}$$

where

F is the force directed towards the rotation centre [N]
m is the mass of the particle [kg]
v is the tangental velocity $[m/s]$
ω is angular velocity $[rad/s]$
r is radius [m]
a is radial acceleration $[m/s^2]$

The average acceleration between the inner radius Ri and the outer radius Ry over the distance Ry – Ri results in a change in velocity which results in pressure (p [Pa]) change. According to Bernoulli's equation for certain conditions. v is the radial velocity m/s.

$$p_1 - p_2 = \frac{v_2^2}{2} - \frac{v_1^2}{2} = \frac{1}{2}(v_2^2 - v_1^2) \tag{2.13}$$

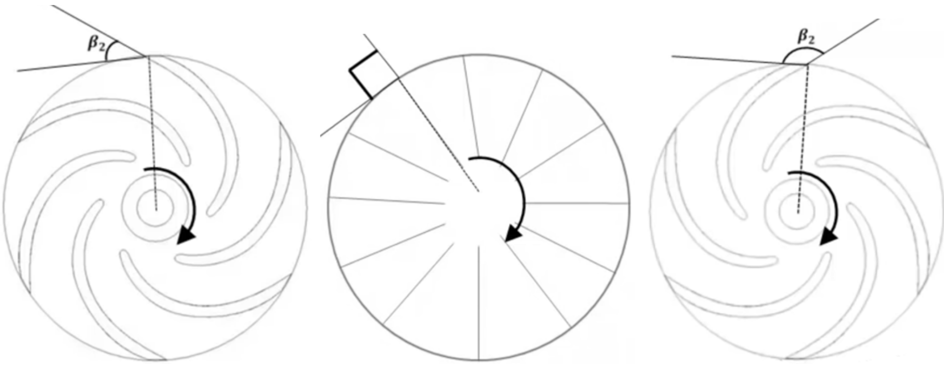


Figure 2.3. Radial fan [youtube, 2019]

It is common in literature that the description of the performance is measured in meters and called head, h_p, of the pump. The total height the pump can move fluid.

P = Power [W]
β = Blade angle [rad]
Q = Flow [m^3/s]
ρ = Density [kg/m^3]
U = Blade tangential velocity [m/s]
V = Fluid velocity [m/s]
b = Fan width at exit [m]
g = Gravity [m/s^2J]

For the pump, flow, head, and power necessary for driving the pump can be described by the equations below.

$$Q \alpha r \omega \tag{2.14}$$

$$h_p \alpha r^2 \omega^2 \tag{2.15}$$

$$P = \rho g h_p Q \tag{2.16}$$

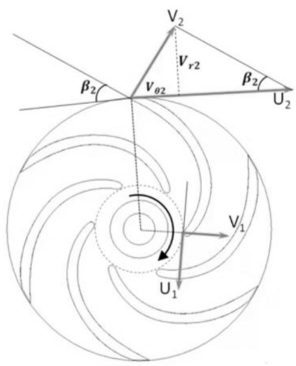


Figure 2.4. Energy Velocity and Flow [youtube, 2019]

The power needed to increase the velocity of the fluid can be established by the following expression. Since the projection of $V_{\theta 1} = 0$ the expression can be simplified.

$$P = \rho Q(U_2 V_{\theta 2} - U_1 V_{\theta 1}) \tag{2.17}$$

$$h_p = \frac{1}{g}(U_2 V_{\theta 2}) \tag{2.18}$$

$$Q = 2\pi r_2 b_2 V_{r2} \tag{2.19}$$

The equations above can be combined into (2.16) which estimates the head for the pump.

$$h_p = \frac{U_2}{g} - \frac{U_2 Q}{2\pi r_2 b_2 g tan\beta} \tag{2.20}$$

Pump performance curves $h_p(Q)$ can be described by the equation below but are normally determined experimentally.

$$p_2 - p_1 = h_p \rho g - \frac{\rho(V_2^2 - V_1^2)}{2} \tag{2.21}$$

Head can be transformed into pressure difference.

$$p_1 - p_2 = h_p \rho g \tag{2.22}$$

2.3 Stepper Motor

A stepper motor is a DC motor but instead of rotating with a certain speed a stepper motor moves in steps. Stepper motors have different number of steps for one revolution, the range of steps can be from 4 to 400 for one revolution [Bill, 2015]. The resolution is often expressed in degrees per step. For example with a 200 step motor you have a resolution of 1.8 degrees per step.

A motor of this kind can have any number of coils, when the coils are connected in groups they are called phases. The coils surrounds a piece of metal (shown in red in Figure 2.5) which is magnetic. When current flows in a phase, the inner metal ring moves one step. This is because the coils becomes magnetic and attract the inner metal piece.

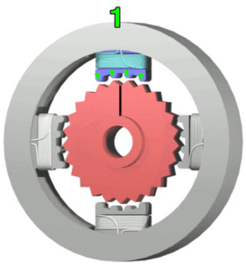


Figure 2.5. Inside of a stepper motor [Bill, 2015]

The stepper motor can be energized in two different ways with help of bipolar and uni-polar drivers. Bipolar drivers let current flow in both directions, this is done with an H-bridge circuit. Because of the change of direction, the phases can change polarity. Bipolar drivers makes it possible for the stepper motor to use all its coils for turning the motor.

Unipolar drivers let the current flow in the same way in all of the phases. The coils have a permanent polarity in each phase. This is done with a couple of transistors as a driver. With unipolar drivers the stepper motor can only use half of its coils at the time, which gives the motor less torque.

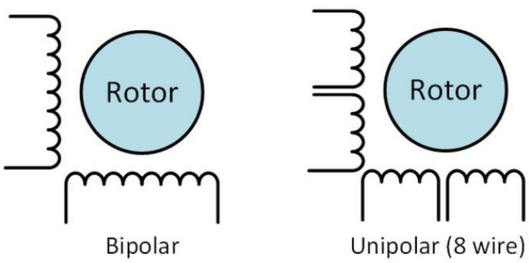


Figure 2.6. The differens between Bipolar and Unipolar [Bill, 2015]

The advantages with a stepper motor is that it can move precisely and has good speed control. This is possible because the motor turn in steps, which are always the same. Regular DC motors have to come up to high speed to have high torque. But with a stepper motor it is possible to have low speed and high torque, this makes it a god motor when low speed and high torque is required.
One of the limitations for a stepper motor, is that when the motor has no load it draws the most current, this makes the motor heat up. A regular DC motor can have high rpm, but with at stepper motor you have limited rpm, this is because the motor only can go in steps and that makes it slower.

2.4 Ultra Sonic Distance Sensor

The human ear can detect frequencies in a range from about 20 to 20 000 Hz [Arthout and Freedman, 2016]. Everything above 20 000 Hz is called ultrasonic and everything below 20 Hz is called infrasonic.
An ultrasonic sensor uses a frequency of approximately 40 000 Hz. The sensor has one transmitter that emits a sound wave and one receiver that detects the wave that have bounced back from an object. The transmitter and the receiver together is a type of a transducer. A transducer converts physical energy to electrical and vice versa.

The transmitter is like a speaker, it has a membrane that generates a sound wave. The electrical energy makes a membrane move, which generates a wave. The receiver works in the same way, but opposite. The energy from a sound wave makes a membrane move which creates an electric signal.

The sound wave has a certain scattering angle, this angle depends on the size of the membrane. Figure 2.7 shows a transmitter, “T” on the sensor and a receiver, ”R” on the sensor.

Figure 2.7. Ultrasonic Sensor [Elektrokit, 2019]

With help of the software, the distance can be calculated. This is done by taking the time from when the transmitter transmit to when the receiver gets a sound-wave back. With a temperature of 20 degrees the velocity of sound is 340 m/s [Arthout and Freedman, 2016]. The distance can be calculated by:

$$L = \frac{v \cdot t}{2} \tag{2.23}$$

Where t is the time and v is the velocity. The formula is divided by 2 because the sound-wave goes to the obstacle and back. Ultrasonic sensors have different applications. A transmitter can only make one type of sound wave, which is going to have a certain scattering angle and certain range. This angle and range, limits an ultrasonic sensor to a specific application.

2.5 Battery and Charging

Batteries are probably the best solutions for electrification of a moving object.They can be recharged during activity by solar panels and during inactivity by a battery charger. Things to consider choosing batteries is the maximum continuous output current and the maximum voltage it can provide. Also, the weight of the batteries can be a factor to consider.

The maximum continuous output current can be increased by parallel connection of batteries. The maximum output voltage can be increased by serial connection of batteries.

Common battery types in a lot of household machines are AAA, AA and A. Theses battery types can be rechargeable and have a continuous output current of 5000 mA for each battery cell.

Another common battery type is led accumulators, usually used in cars and bikes. They are also rechargeable and have often a higher continuous output current. A disadvantage is that they are heavier than the types above.

2.6 Robot Home Procedure

When the robot vacuum cleaner is done cleaning or has low charge on the battery it must find its way back to the recharge station. This is an advanced assignment for the cleaner. The robot can be far away from the charge station or it can have a lot of walls and obstacles in the way.

2.6.1 Using sound

Using a transmitter and microphones is a solution to get the robot to the station. The best way would be to put the microphones on the robot and the transmitter on the recharge station. The transmitter would transmit a low frequency sound that only the robot microphones can pick up.

There is a bachelor thesis on this subject [Enkoson and Zhou, 2017]. In this thesis Enoksson and Zhou designed a robot that could follow sound. This thesis has a couple of restrictions, one test in the thesis was that the sound source and the microphones had a maximum distance of 125 cm from each other. With a cleaner the distance would be much greater. It is unsure if this method would work for the robot cleaner, the result of the thesis is that it did not work in all cases. It worked best when the sound come from an angle between 60 to 120 degrees. The robot cleaner would need a reliable solution that works in all angels.

2.6.2 Using GPS

Another way to get the robot cleaner to the charging station, is by using GPS. If the robot vacuum cleaner uses a GPS, it can locate where it is in the house and locate where the re-charger is. The thesis, POKI [Olsson and Kringberg, 2016], a robot that collects golf balls on a field. POKI uses a GPS so it can find the way back to the recharge station and so it stays within the field. For the POKI robot the GPS works fine, the robot gets its GPS position with an accuracy of 5 - 8 meter. This works when you have big field to cover. But in a small apartment, 5 – 8 m is a lot. With a better GPS, the position accuracy could be smaller than 1 meter.

2.6.3 Using IR sensors

A possible way for the robot vacuum cleaner to find a way back to a recharge station would be using Infrared light. The thesis “Can cheap robotic vacuum cleaners be made more efficient” [Kiritsis, 2018] mentions a robot vacuum cleaner, the Roomba 605 that uses this method. On the vacuum cleaner there is an infrared receiver and on the charge stations there is a infrared transmitter. The Roomba 605 follows the IR signal back to the charge station. The IR method have some restrictions, one of them is that the IR transmitter and receiver have to be in sight of each other. This means that the robot vacuum cleaner must find the way back to the room where the recharge stations is. When the robot cleaner is in that room, the IR signal from the charging station will be easy to find.

2.6.4 Tracking the robot movements

The cheapest solution would be tracking the robot movements. This method requires a large amount of memory, because it saves every movement made. When it is time for the robot to go back to the charge station it is going make the same trip as the robot has done to get to the current position. This means that it is going to take the same time to go back to its charging station, as it took to get to the final position. This method is time consuming and very inefficient.

A more advanced way would be to save the movements the robot makes, and then calculate the shortest way to get back to its station. This method demands an advanced algorithm and probably a better micro computer to make its calculations.

2.7 Research Question, Efficiency Of Driving Patterns

In the thesis Can cheap robotic vacuum cleaners be more efficient, written by Lea Mets Kiritsis [Kiritsis, 2018], a computer simulation was done to investigate this. It was done by comparing two different vacuum cleaners with different driving patterns.
The simple vacuum cleaner had a random driving pattern and changed direction when it came to an obstacle. The new direction was considered random. The other vacuum cleaner had a sophisticated driving pattern moving in straight paths across the area. Even when an obstacle was in the way it found its way around it and continued.
The result was that the vacuum cleaner with the sophisticated navigational system was a lot better and the difference increased for larger areas. The room areas were 10x10 m and 20x20 m and the number of obstacles were none or one. For a room 10x10 m and without obstacles the improvement was a reduction in steps by 82% in average.
The writers conclusion was that there is a save potential by improving the drive

pattern and recommend additional psychical tests.

In the thesis Husbie 2.0 [Svensson and Granström, 2017] a study of different drive patterns area coverage efficiency was made. The different drive patterns that were analyzed was Minimizing Square, Parallel Movement and Random bounce. The test was conducted with a psychical prototype. Time is a fixed value and the area covered is measured. The areas are squares and there are no obstacles except the walls.
Minimizing Square method is established by when the front sensor detects an obstacle the vacuum cleaner measures to the sides to determine where to turn. It turns towards the largest measurement 90 degrees. When three turns have been made the front sensor detects at a different value.
Parallel Movement pattern is parallel lines between walls. It is similar as the Minimizing Square but the robot turns 180 degrees when the front sensor is activated and there is no adjustment of the detection length for the front sensor.
Random Bounce is described as when the front sensor detects an obstacle a random rotation angle is done and the robot continues in the new direction.
The result from these tests showed that with short driving time the Parallel patterns showed best result but with longer driving times there was no significant difference between the different methods. These writers also recommend additional tests.

The thesis Improving robotic vacuum cleaners [Gylling and Elmarsso, 2018] minimizing time needed for complete dust removal was investigated. The investigation was done to see if the time to dust an empty room can be reduced by providing a map where the dust is normally located to the vacuum cleaner.
Three different cases are investigated to shorten the cleaning time .Clean along the walls first, clean the closest known dusty place, and a dust map algorithm that is better than regular known patterns.
There were a lot of different tests conducted in this thesis. The writers' conclusion was that a dust map shortens the cleaning time since it can reduce the amount of time used to clean the same area several times. Having obstacles in the room increases the time and power consumption since the vacuum cleaner must do more turns. The downside for a dust map is that it requires accurate sensors that are expensive. The recommendation was to use a efficient PPCR algorithm to get at optimized first sweep.

Chapter 3

Demonstrator

3.1 Layout

3.1.1 Overall layout

The goal was to keep the dimensions of the vacuum cleaner within 350 x 350 x 100 mm (width x width x hight). The dimensions set was from the market investigation, see Appendix G. Brushes and other parts below the chassis bottom and the bottom needed to have sufficient ground clearance. A goal of at least 15 mm ground clearance was set.
The overall dimensions requirement could be kept. The parts that are below the chassis bottom plate are covered by the wheels. The front wheel should move backwards to cover the brushes and the nozzle from the dust collector shortened. There are no technical obstacles to do these things but they were not implemented on the demonstrator.

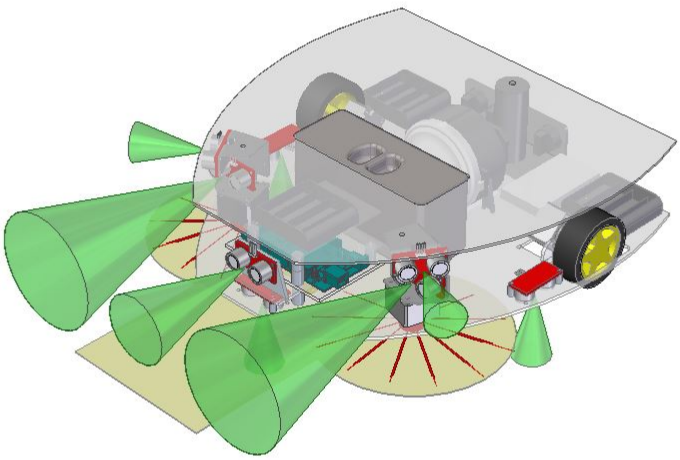


Figure 3.1. Overall layout [Bergman and Lind, 2019a]

The green cones, in Figure 3.1, represent the detection length and the scattering angle of the Ultrasonic sensors. The yellow circle represent the sweep area of the brushes and the yellow square the width of the suctions channel.
The brushes has an elliptical shape to reach as much forward and sideways as possible without increasing the resistance against obstacles more than necessary. Two brushes sweeps the floor for particles and gather them in the center of the cleaner. In the centre the particles are sucked up into a dust container and stopped there by a HEPA filter. The clean air then passes the filter and goes through the fan and out in the rear of the cleaner.

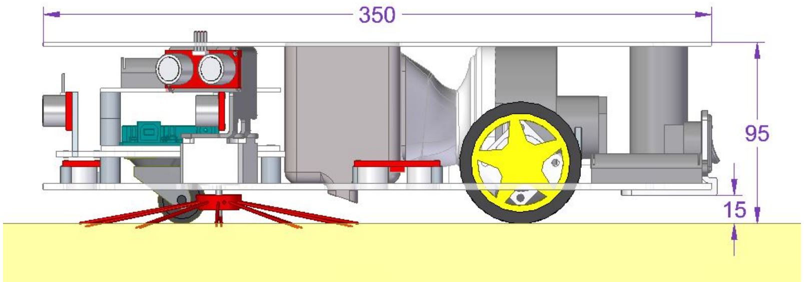


Figure 3.2. Overall side view [Bergman and Lind, 2019a]

3.1.2 Sensor layout and shape

The front 3 horizontal sensors detect obstacles in front of the cleaner. To cover the whole width of the cleaner it was necessary to have more then one sensor. The sensors pointing to the sides are for detecting distance to closes obstacle on the side. The three sensors with direction downwards are to detect obstacles like stairs or other high edges.
The shape of the cleaner seen from above was a result of the idea to turn the vacuum cleaner only by stopping one wheel. The shape is created by a radius with its center in the stopped wheel.

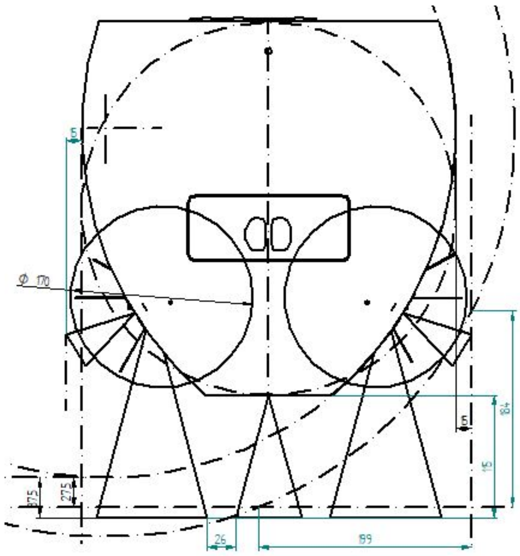


Figure 3.3. Sensor layout and shape [Bergman and Lind, 2019a]

3.2 Hardware

3.2.1 DC motor

The DC motor used has a voltage range between 2.4 and 4.5 V. At nominal voltage 3.6 V and without load the speed is 20200 *revolutions/minute*. See Appendix A. Revolutions/minute is directly proportional to the Voltage, see equation (2.5). When pumping air the load is low and it is assumed to be the same as the *revolutions/minute* for no load.
At 4.8 V it will be approximately 26900 *revolutions/minute* according to what follows:

$$E = K_2 \Phi \omega \; [V] \tag{3.1}$$

$$3.6 = K_2 \Phi 20200 [V] \tag{3.2}$$

$$4.8 = K_2 \Phi 26933 [V] \tag{3.3}$$

The speed will be reduced during use since the voltage decreases.

3.2.2 Fan

Assuming an isentropic process (Ideal thermodynamic process) the pressure difference in the fan can be calculated using equations (2.13) and 2.17).

$$p_1 - p_2 = \rho U_2 V_{\theta 2} \tag{3.4}$$

The velocities can be calculated by using the velocity triangles shown in figure 2.14 and the fact that the flow into the fan must be equal to the flow out from the fan.

Parameters for the fan in the demonstrator are:
Blades are backward curved.
At blade entrance β_1= 25 [°]
At the blade exit β_2= 50 [°]
Outer diameter $D_2 = 0.075$ $[m]$
Inner diametersD_1= 0.033 $[m]$
Width of blade at entrance b_1= 0.01 $[m]$
Width of blade at exit b_2= 0.005 $[m]$
Rotational speed is n = 20200 $[revolutions/minut]$
Angular velocity is $\omega = 2115$ $[rad/s]$
Air density $\rho = 1.22$ $[kg/m^3]$

$$U_1 = \frac{\omega D_1}{2} \tag{3.5}$$

$$V_1 = U_1 tan(\beta_1) \tag{3.6}$$

$$V_1 D_1 b_1 = V_{r2} D_2 b_2 \tag{3.7}$$

$$V_{\theta 2} = \frac{V_{r2}}{tan(\beta_2)} \tag{3.8}$$

$$U_2 = \frac{\omega D_2}{2} \tag{3.9}$$

With values the results are:
$U_1 = 35$ $[m/s]$
$V_1 = 16$ $[m/s]$
$V_{r2} = 14$ $[m/s]$
$V_{\theta 2} = 12$ $[m/s]$
$U_2 = 79$ $[m/s]$

$p_1 - p_2 = 1157$ $[Pa]$

The nozzle to the dust container has a rectangular shape. To ensure that this nozzle is not a restriction for the flow more than the circular inlet to the fan the hydraulic diameters are compared. For the inlet to the fan the hydraulic diameter is the same as the diameter which is 35 mm.
For the rectangular inlet nozzle the hydraulic diameter can be calculated according to the equation below:

$$d_h = \frac{2ab}{a+b} \tag{3.10}$$

a and b are the length of the sides in the rectangle. For the nozzle a = 11 mm and b = 142 mm and the resulting d_h = 59 mm. This means that the nozzle too the dust container is not a restriction for the flow more than the fan housing.

3.2.3 Stepper Motor and Driver Card

There are four stepper motors in the robot vacuum cleaner. Two of the stepper motors are used to power the wheels and the others two are used to power the brushes. The stepper motors that controls the brooms are going three times faster than the motors that controls the wheel.

The stepper motors are of model TS3214N61 and they need 12,5 V and 0,25 A to work. Because of the age of the model, the manufacturer does not have the specifications, but they have a similar motor. The specifications for that motor can be found in Appendix C.

To calculate how much weight every stepper motor can push forward, following equations are needed:

$$F_{torque} = ma + F_{dynamic} + F_{air} \tag{3.11}$$

Where F_{torque} is the force from the stepper motor:

$$F_{torque} = M_{tot}/r \tag{3.12}$$

$F_{dynamic}$ is the wheel friction.

$$F_{dynamic} = NC_r = mgC_r \tag{3.13}$$

where $C_r = 0.015$ [Siöland, 1986].

F_{air} is the air resistance and because of the low speed the air resistance is zero.

From the equation 2.2 M_{tot} can be calculated.

Where:
M_{acc} is shown in the equation 2.3.
J and M_{last} is shown in Appendix C as rotor inertia and holding torque:
$J = 7,5\ gcm^2 = 0.000075\ Nm^2$
$M_{last} = 580\ gcm = 0.0580\ Nm$

$\frac{d\omega}{dt}$ can be calculated by:

Assuming a velocity for the robot cleaner: $v = 0,02\ m/s$ and that it reaches its speed within one step.
With the know radius of the wheel $(r = 0,03)\ m$, the girth can be calculated to:

$0,1885\ m$.
With the girth and the velocity it can be calculated that it takes $9,425\ s$ per lap. Which means 6,3660 rounds per minute and 6,3660 rpm is $0,666\ rad/s$.

M_{acc} and M_{tot} can now be calculated:
$M_{acc} = 0,00004995\ Nm$ and $M_{tot} = 0.05805\ Nm$.

From the equations 3.11, 3.12 and 3.13 the mass can now be calculated:

$$m = M_{tot}/r(a + gC_r) \tag{3.14}$$

Where the:
$a = 0.4\ m/s^2$
$g = 10\ m/s^2$

With the values that has been calculated the maximum mass that one stepper motor can transport is: $m = 3.52\ kg$.
The holding torque is define when the engine is stationary. If acceleration and speed are applied the delivered torque from the engine will be reduced.

Every stepper motor needs a driver card. The driver card that are used for our stepper motors is of model A4988. In the figure 3.4 it is shown how to connect the driver card to make it work correctly.

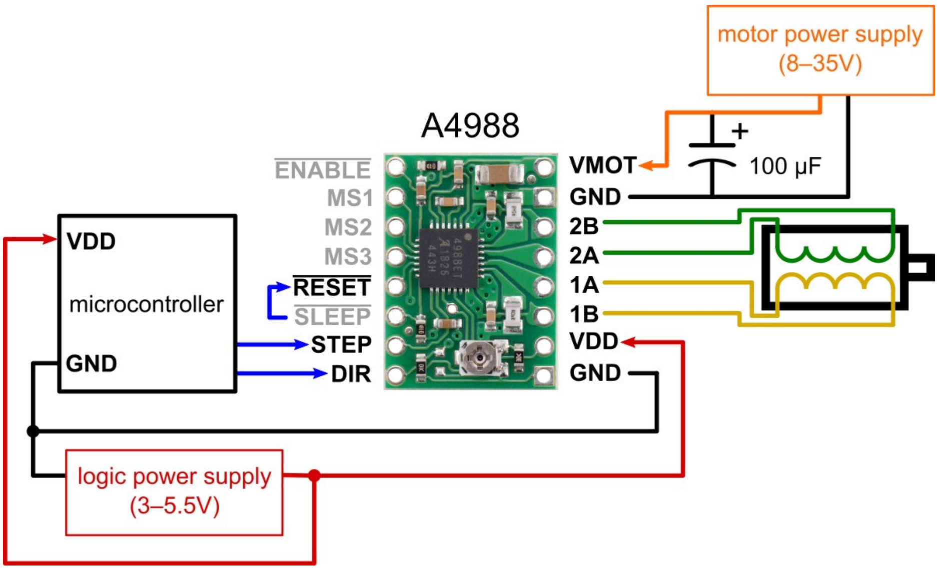


Figure 3.4. Wiring diagram for driver card A4988 [Pololu, 2019]

The driver is sensitive to high spikes of voltage, this is why a capacitor needs to be connected between the 15 voltage batteries and the driver card. The capacitor delivers a steady voltage without spikes of high voltage. The capacitor needs to be at least 47 micro Farad, the robot vacuum cleaner uses four stepper motors. To be on the safe side the electric circuit in the robot uses a capacitor of 1000 micro Farad. In appendix D the driver card specification is shown.

3.2.4 Ultra Sonic Distance Sensor

The robot vacuum cleaner has a total of eight ultra sonic sensors. Five of the ultra sonic sensors that controls if there are any obstacles or walls in the way. The other three sensors control if there are any stairs. The sensors are of the same model, HC – Sr04.

Model HC – Sr04 has a range from 2 cm to 400 cm and the total measuring angle is 30 degrees. A measuring test was executed, where the sensors spreading angle needed to be confirmed.

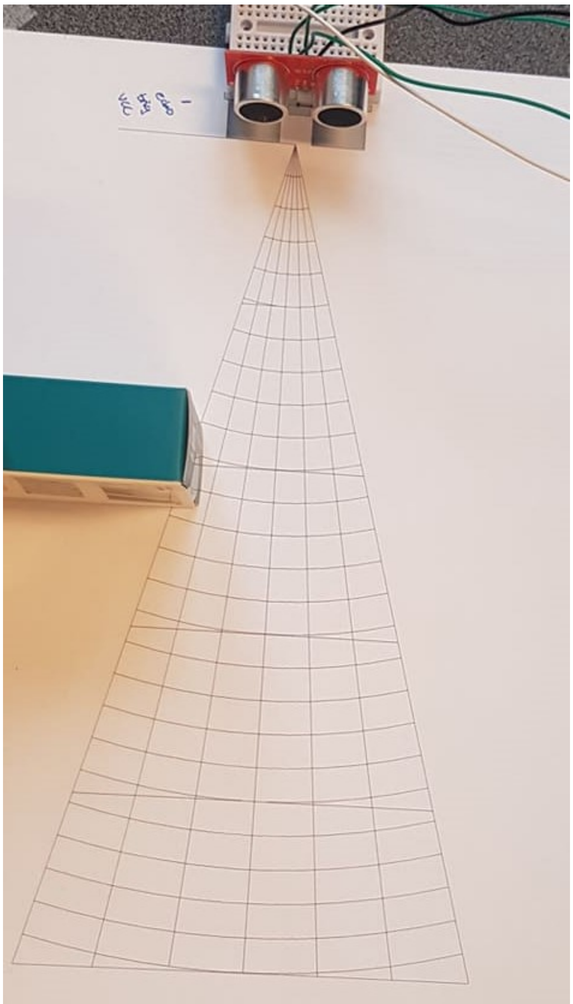


Figure 3.5. Picture of how the test was executed [Bergman and Lind, 2019c]

The test showed that sensors of the same Model, but different price gave different results. The cheaper one, got a spreading angle at 30 degrees if the obstacle was tangent to a radius created from the sensor, if the obstacle was perpendicular it had a spreading angle at 15 degrees. The more expensiver one got a spreading angle at 30 degrees also if the obstacle was perpendicular to the sensor.

That is why the more expensive sensor controls if there are any obstacles in the way, and the sensor that is cheaper controls if there is any stairs. The data-sheet for the sensor can be found in appendix F.

3.3 Electronics

In the Hardware chapter there are descriptions for electrical requirements and connections for each component. This chapter contains an overview of the complete electrical installation.
A summary of the requirement will show that the stepper motor requires both 12 V and 5 V and consumes a current of maximum 0.25 A, 12 V is for driving and 5 V is for control of the stepper motor driver. The Arduino require 5 V, the DC engine for the fan require 2.4 - 4.5 V and at no load it consumes 2.9 A.
When trying to power the Arduino with only one battery cell, 4.8 V, it did not work properly. Most likely because there is a power drop from the batteries when components are connected. Instead the Arduino is connected to two parallel connected cells which results in 9.6 V. This is possible because the Arduino has a built in power protection. 5 V DC is powered to components from the Arduino.
The DC fan motor is connected to one battery cell of 4.8 V and 2.5 Ah.
To enable testing different functions three separate switches was installed, one for the DC motor to the fan, one for the Arduino and one for the powering of the stepper motors. Two inputs were installed in the rear panel to enable charging.

During test of the stepper engines and Arduino an average current consumption was 0.75 A. If an assumed power consumption for the DC fan motor is added of 3 A, the total current consumption is 3.75 A. This will result in a possible drive time of 40 minutes.
Below a breadboard diagram designed in a program called Fritzing. A more conventional type of diagram is in Appendix B.1.

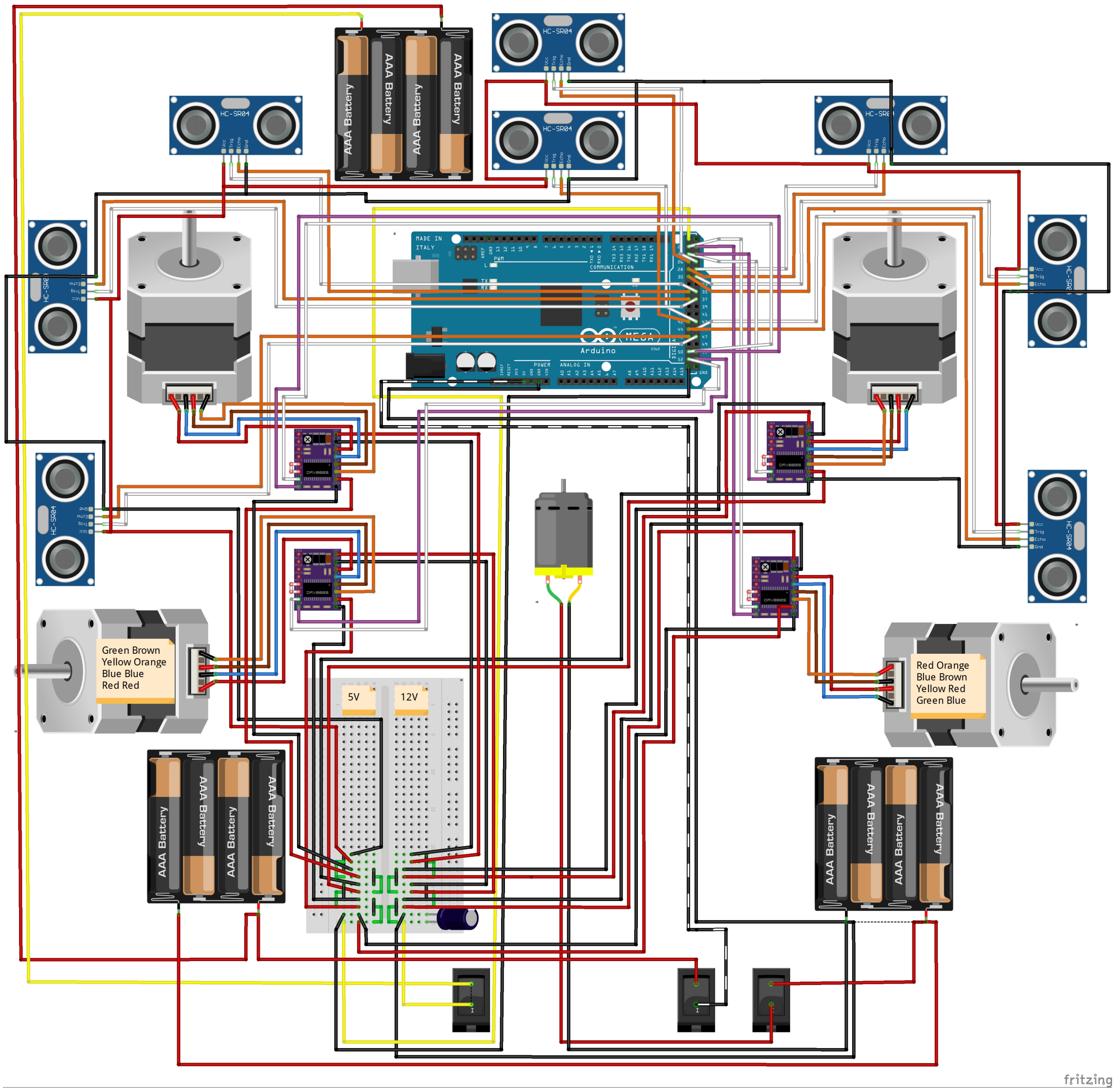


Figure 3.6. Electrical circuit diagram [Lind, 2019]

3.4 Software

The robot vacuum cleaner is controlled by a microcomputer called Arduino Mega. The Arduino Mega uses the programming langued C++. The Arduino mega only has one processor, this limits how the software will be formed.
Both the stepper motor and the ultrasound sensor need the program to stop a few milli seconds for the parts to work correctly. This means that the stepper motor and

the sensor can not run together. Therefore, the program needs to run the stepper motors and periodically turn on the sensors to detect if there is any obstacles or stairs in its way. In figure 3.7 the flow chart show what happens if the robot vacuum cleaner detects an obstacle or stairs.

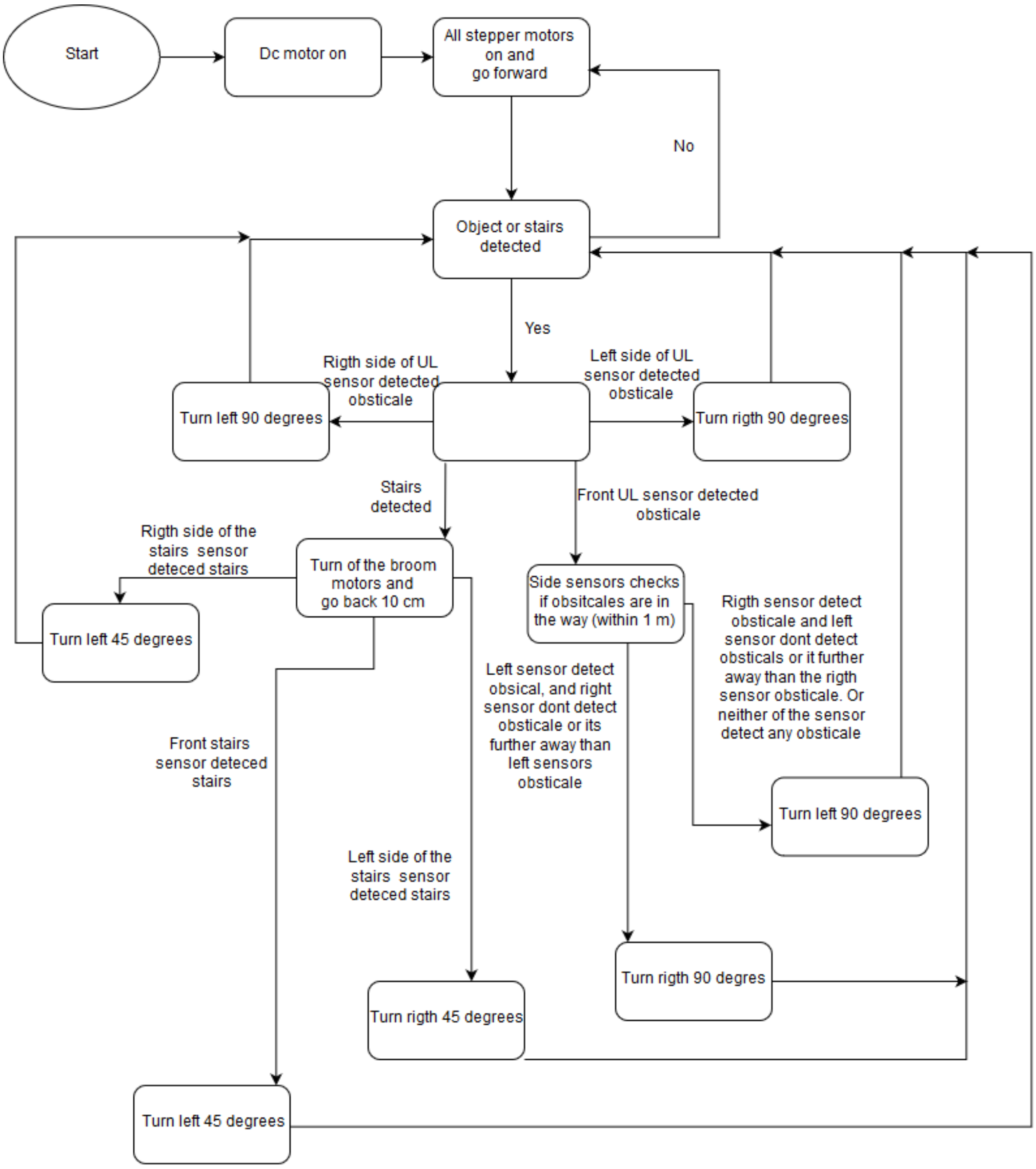


Figure 3.7. Flow chart of the program [Bergman, 2019]

For the entire code to the program look at appendix E.

Chapter 4

Result

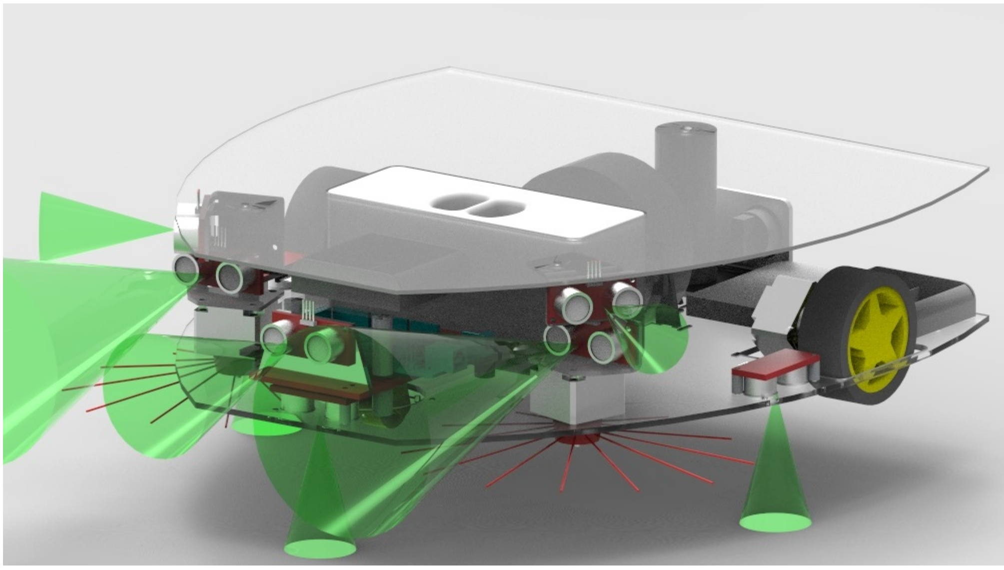


Figure 4.1. Picture of the Robot Vacuum cleaner [Bergman and Lind, 2019b]

In chapter 1.2 the research questions are stated. In sub-chapters below the results are described.

4.1 Comparing demonstrator and cleaners on the market

Comparing driving time all cleaners on the market have longer drive time. The demonstrator has about 40 minutes in theory and the cleaners on the market has 45 – 120 minutes. One of the market cleaners reduces the suction capability to increase the drive time. To increase the drive time additional battery capacity, have to be added. The most current consuming part is the DC fan.

The dimension of the cleaner were chosen after the market investigation and are common dimensions. The shape is different, see description why in chapter Driving pattern. The demonstrator height is 95 mm, it might be a bit high but was a result of the size of the DC fan.
The pressure difference created is in theory 1157 [Pa] at 3.7 V. At 4.8 V it will be higher. This is in parity with the cleaners that stated a value for this.
The number of sensors are less on the demonstrator but the number of different driving patterns are more for the market cleaners.
Implementing more driving patterns is only a matter of developing several algorithms and test them.
The weight of the demonstrator is 2600 g, comparing with other cleaners this is not a high or a low weight.
Maximum volume of the dust container is on the demonstrator 0.4 liters, nor this value is high or low comparing with cleaners on the market.

4.2 Design of sensors and code

To be able to detect obstacles over the full width and close to the cleaner, to reduce measuring time, three sensors have to point forward.
In front of each wheel there is a sensor which detects that the cleaner does not pass over a high edge, for example a stair. To measure the distance to an obstacle on one of the sides of the Arduino two sensors are needed.
When driving forward only the front three sensors and the wheel sensors are active when using the driving pattern that is random, see description in next chapter. The front and wheel sensors are check every 25 mm. The reason for this is that if they are checked to often there will be too much disturbance to the forward driving motions. There is only one processor in the Arduino and the code is executed in sequence so every action hive to be in the loop.
When using the s-shaped driving pattern also the side sensors have to be activated when driving along the first wall.
A total of eight Ultrasonic sensors are required to achieve the functions required. The chosen type of Ultrasonic sensor works for this application.

In chapter 2.7 the investigation about driving patterns are described. Three different theses were analyzed. In general, a S-shaped driving pattern (Parallel Movement pattern) had better performance than a random pattern. One of the theses describes that the differences between S-shape and random pattern reduces with longer driving time.
An idea tested with the demonstrator was to shape it so when it turns one wheel stops rotating and the cleaner turns. This will most probably reduce the necessary driving time to clean a surface since the cleaner does not have to go backwards for each turn. The downside of this shape and driving patterns is that the corners will have an uncleaned triangle with the base and height about 200 mm.

4.3 How can a function that returns the cleaner to a battery charge station, be designed

This thesis presents four solutions on how to the get the robot to its charging station. Where three of the methods has been tested and documented in other theses. The methods that had been tested are using IR sensor, GPS and sound.

In the thesis where they used sound to get the robot home, the test environment was not a special real and they only did test when the maximum distance between the sound source and the microphone was 125 cm. Therefore, this method is not suitable for our vacuum cleaner.

The best way to get the robot vacuum cleaner home is to combine two of the methods. In the thesis where they used the IR sensor to get the robot home. The robot needs to find the infrared light, so it could track the light to find its way to the charging station. But to find the infrared light could be difficult, the robot needs to be in the same room and have a free sight to its charging station.

For the vacuum cleaner to finds its way to the right room and come to reasonable distance within the charging station, another method would need to be used.

There is two possible ways, first one is using GPS. With a good GPS the robot can come within a few meters to its charging station, and then the infrared light can take over. The other method would be that the robot tracks its way back to the room where the charging station is, and then drives random to find the infrared light.

Chapter 5

Disscusion

All of the robot's functions have been tested, the obstacle and stair sensors worked, in the test that was made the robot vacuum cleaner avoided all the obstacles and stairs.

The biggest problem with the prototype is that the robot goes very slow. This is because of the weight of the robot vacuum cleaner, 2.6 kg. With this kind of weight, the stepper motor needs high torque to make it move forward.
When the stepper motors have high torque, it goes slow and when it has low torque, it goes fast. Therefore, the robot goes slow. We could solve this problem by exchanging stepper motors to bigger ones. If we had more time, we would do the exchange.

On the other hand, maybe if the robot did go faster the robot could not detect the obstacle or stairs as good as it does now. Probably the robot cleaner could go a bit faster than it does now but at the same time sustain how it detects obstacle and stairs.

Because of the robot vacuum cleaner weight, the bottom plate that are made of acrylic plastic and are only three mm thick is very fragile. If we had more time, it would be smart to change the materiel or have at thicker bottom plate.

Because of the time limit we could not make a program that did the recommended driving pattern. Instead we made a program that was semi random. For instance this program could compare the distance between two sensors and then decide what turn it should make. This program worked fine but could be improved a lot more to make the robot cleaner be more efficient.

Chapter 6

Recommendations for Further work

The robot vacuum cleaner that was made in this thesis has its limits and are missing a couple of functions that can make the robot much more attractive on today's market.

A key function would be that the robot can find its way back to the recharge station. If further work would be done, designing a system out of our theory how to get the robot vacuum clean to its station would be recommended.

Constructing a better program for the robot would also be the next step for a greater cleaning ability. This can be done by either improve the program that the robot runs on, or trying to make a more advanced algorithm so it can clean with the recommended pattern.

To get a better and safer circuit two PCB would needed to be constructed to replace the unstable breadboards connections.

Bibliography

Centrifugal pump basics. URL https://www.youtube.com/watch?v=v_k3u9Qxo6c. Video, Acessed: 2019-03-29.

Nicholas Apazidis. *Mekanik II*. Studentlitteratur AB, 1:3 edition, 2012.

H.D Young Arthout and R.A Freedman. *University Physics with Modern Physics*. PEARSON, 14 edition, 2016.

Joel Bergman. Program drawio, google, 2019. URL: https://www.draw.io/.

Joel Bergman and Jonas Lind. Program solid edge st10, simens, 2019a. URL https://solidedge.siemens.com/en/. Acessed: 2019-03-29.

Joel Bergman and Jonas Lind. Program keyshot 6.1, 2019b. URL https://www.keyshot.com/. Acessed: 2019-03-29.

Joel Bergman and Jonas Lind. Photo, 2019c.

Earl Bill. *All about stepper Motors*. Adafruit learning system, 09 2015. Info & Fig.

Eric f Curd, Jorma Railio, Jan Gustavsson, Jaap Hogeling, Mamdouh El Haj Assad, Jan Emilsen, Sante Mazzacane, and Ralf Wiksten. Industrial ventilation design guidebook, 2019. DOI: 10.1016/B978-012289676-7/50012-6.

Gunnar Dahlvig. *Konstruktionselement och maskinbyggnad*. Liber, 3:4 edition, 1999.

Elektrokit. Avståndsmätare ultraljud hc-sr04 2 – 400cm, 2019. URL https://www.electrokit.com/produkt/avstandsmatare-ultraljud-hc-sr04-2-400cm/. Accesed: 2019-04-23, Fig.

Karl Enkoson and Bohan Zhou. Sound following robot. Bachelor Thesis, Kungliga Tekniska Högskolan, OAI: DiVA.org:kth-226665, 2017.

Per Foreby. Att skriva en rapport med latex, 09 2006. URL http://www.ddg.lth.se/perf/handledning/handledning.pdf. Acessed: 2019-05-02.

Andreas Gylling and Emil Elmarsso. Improving robotic vacuum cleaners. Bachelor Thesis, Kungliga Tekniska Högskolan, OAI: DiVA.org:kth-229393, 2018.

Hans Havtun. *Applied Thermodynamics*. Studentlitteratur AB, 2014. Edition 1:2, Info & Fig.

Jernkontoret. Fans, 2019. URL `http://www.energihandbok.se/flaktar/`. Acessed: 2019-03-29, Info & Fig.

Hans Johansson. *Elektroteknik*. Institutionen för Maskinkonstruktion, 2013. Info & Fig.

Lea Mets Kiritsis. Can cheap robotic vacuum cleaners be made more efficient? Bachelor Thesis, Kungliga Tekniska Högskolan, OAI: DiVA.org:kth-230228, 2018.

Jonas Lind. Program fritzing, 2019. URL `http://fritzing.org/home/`. Acessed: 2019-03-29, Fig.

Marcus Olsson and Fredrika Kringberg. Poki, robotic ball collector on golf ranges. Bachelor Thesis, Kungliga Tekniska Högskolan, OAI: DiVA.org:kth-192008, 2016.

Pololu. A4988 stepper motor driver carrier, 2019. URL `https://www.pololu.com/product/1182/resources`. Acessed: 2019-05-02, Fig.

Burnett Roderick. Understanding how ultrasonic sensors work. URL `https://www.maxbotix.com/articles/how-ultrasonic-sensors-work.htm`. Acessed: 2019-03-29.

Tomas Siöland. Rullmotstånd hos dragna hjul med lågprofildäck, 1986. Institutionen För Lantbruksteknik, Uppsala, ISBN: 91-576-2620-0.

Erik Svensson and Daria Granström. Husbie 2.0. Bachelor Thesis, Kungliga Tekniska Högskolan, OAI: DiVA.org:kth-226649, 2017.

Virtual university youtube. Centrifugal pump working, 2019. URL `https://www.youtube.com/watch?v=IiE8skW8btE`. Video, Acessed: 2019-03-29, Info& Fig.

Appendix A

Dc Motor Datasheet

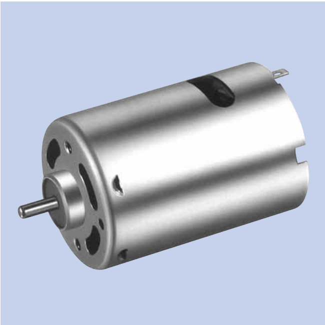
RS-540RH/SH

OUTPUT: APPROX 2.0W~65W

Carbon-brush motors

Typical Applications

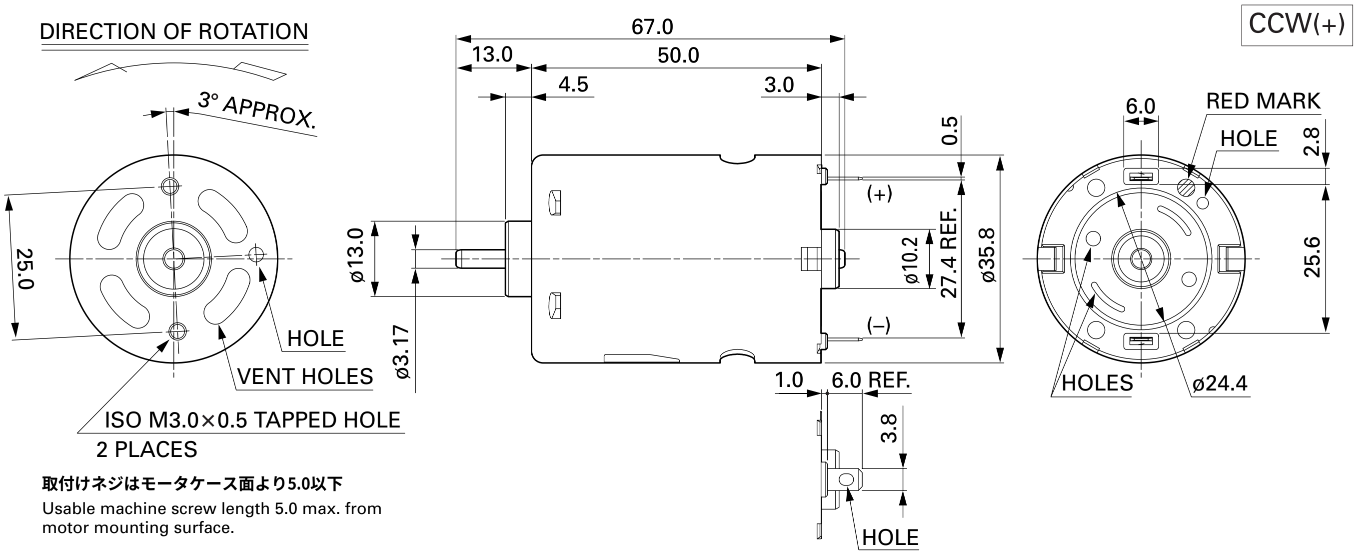
Household Appliances : Vacuum Cleaner
Cordless Power Tools : Air Compressor / Drill & Screwdriver / Cordless Garden Tool
Toys and Models : Radio Control Model



MODEL	VOLTAGE		NO LOAD		AT MAXIMUM EFFICIENCY					STALL		
	OPERATING RANGE	NOMINAL (COMMUTATOR POSITION)	SPEED	CURRENT	SPEED	CURRENT	TORQUE		OUTPUT	TORQUE		CURRENT
			r/min	A	r/min	A	mN·m	g·cm	W	mN·m	g·cm	A
RS-540RH-7516	2.4 ~ 4.5	3.6V CONST. (CCW+)	20200	2.90	16310	12.2	15.1	154	25.8	78.5	800	51.0
RS-540RH-5045	4.5 ~ 9.6	6V CONST. (CCW)	11600	0.82	9660	4.09	14.4	147	14.6	86.3	880	20.4
RS-540RH-6035	3.6 ~ 7.2	3.6V CONST. (CCW+)	9500	1.15	7700	4.91	12.3	125	9.88	64.7	660	21.0
RS-540SH-5045	4.5 ~ 12.0	12V CONST. (CCW+)	17500	0.95	15080	5.93	31.8	324	50.1	230	2350	37.0
RS-540SH-7520	3.6 ~ 7.2	4.8V CONST. (CCW+)	15400	2.00	13010	10.9	26.0	265	35.3	167	1700	59.0

The terminal position against the tapped holes varies depending on CW+/NEUTRAL.

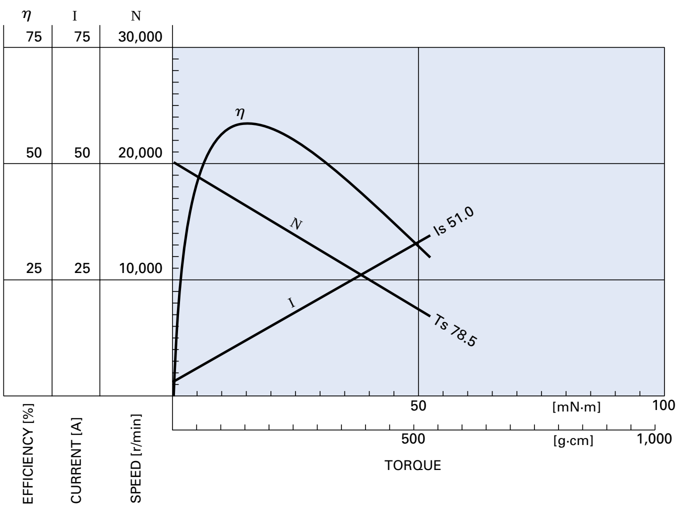
UNIT: MILLIMETERS



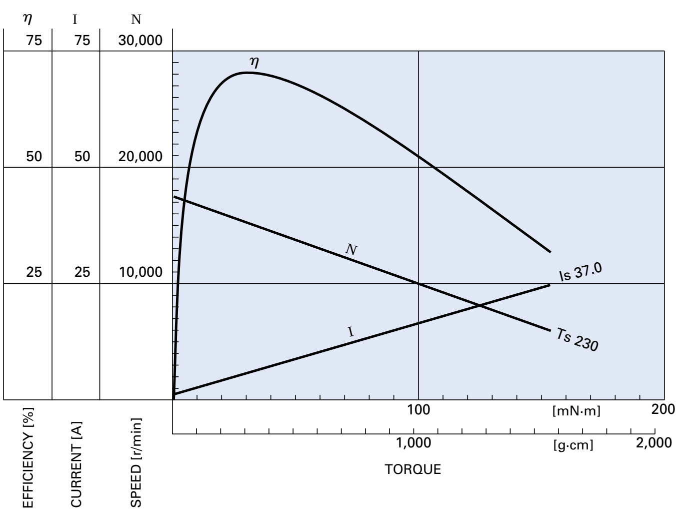
取付けネジはモータケース面より5.0以下
Usable machine screw length 5.0 max. from motor mounting surface.

WEIGHT: 160g (APPROX)

RS-540RH-7516 3.6V



RS-540SH-5045 12.0V



Appendix B

Circuit Diagram

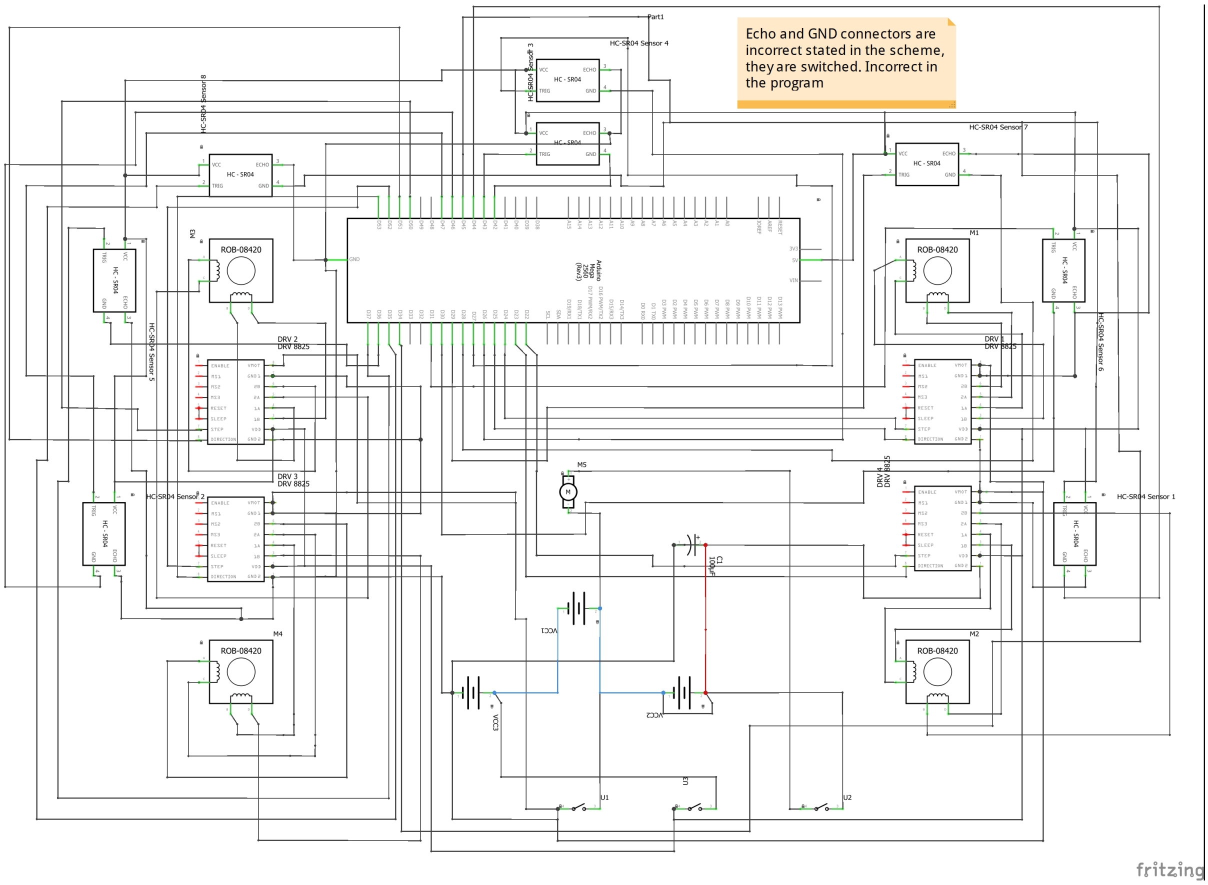


Figure B.1. Electrical circuit diagram [Lind, 2019]

Appendix C

Stepper Motors Datasheet

ステップモータ仕様書

STEP MOTOR SPECIFICATIONS

形式 MODEL	TS3214N28

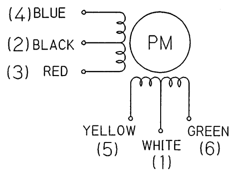
電気的特性 ELECTRICAL CHRACTERISTICS		
基本ステップ角度 FUNDAMENTAL STEP ANGLE	1.8°	
励磁方式 DRIVING METHOD	UNIPOLAR 4-2PHASE ON	
定格電圧 RATED VOLTAGE	DC 12 V	
定格電流 RATED CURRENT	DC 0.19 A	
巻線抵坑 WINDING RESISTANCE	63 Ω ±10%	3
巻線インダクタンス WINDING INDUCTANCE	27 mH ±20%	2
絶縁階級 INSULATION CLASS	CLASS B (W/O READWIRE)	
絶縁抵坑 INSULATION RESISTANCE	100MΩ MIN. DC 500 V	
絶縁耐圧 DIELECTRIC STRENGTH	500 V AC 1MIN.	

機械的特性 MECHANICAL CHARACTERISTICS	
外形図 OUT LINE	032140047F30
回転子イナーシャ ROTOR INERTIA	7.5 g-cm²
軸振れ SHAFT RUNOUT	0.1 mm T.I.R.MAX
ラジアルプレイ RADIAL PLAY	0.02 mm MAX. 500 gr LOAD
スラストプレイ THRUST PLAY	0.02 mm MAX. 900 gr LOAD
重量 WEIGHT	140 gr

モータ性能 MOTOR PERFORMANCE		
静止角度誤差 POSITIONAL ACCURACY	1.8° ± 5%	1
隣接角度誤差 STEP POSITION ACCURACY	1.8° ± 5%	1
ホールディングトルク HOLDING TORQUE	580 gr-cm MIN. AT 2 PHASE ON 0.19 A/PHASE	
ディテントトルク DETENT TORQUE	36 gr-cm MIN.	
最大自起動周波数 MAX.STARTING PULSE RATE	— PPS MIN.	
最大応答周波数 MAX.SLEWING PULSE RATE	— PPS MIN.	
プルアウトトルク PULL-OUT TORQUE	— gr-cm MIN. at — PPS	
平均故障時間間隔 M.T.B.F	1000000 h	4
温度上昇値 TEMPERATURE RISE	80 degMAX. AT 2PHASE ON DC 12 V	

環境条件 ENVIRONMENT CONDITION	
使用温度範囲 OPERATING TEMP.RANGE	- 5℃~ + 50℃
使用相対湿度範囲 OPRATING HUMIDITY RANGE	5 % RH~ 95 % RH
保存温度範囲 STORAGE TEMP.RANGE	- 40℃~ + 70℃
保存相対湿度範囲 STORAGE HUMIDITY RANGE	5 % RH~ 95 % RH

結線図 SCHEMATIC



下記励磁順序にて取付け面側から見て出力軸が CW 回転
SWITCHING SEQUENCE FOR CW ROTATION VIEWED FROM MOUNTING END

STEP	BLUE	YELLOW	RED	GREEN	BLACK WHITE
0	ON	ON			+V
1		ON	ON		+V
2			ON	ON	+V
3	ON			ON	+V
4	ON	ON			+V

備考. REMARKS

1. MEASURMENTS MADE AT RATED CURRENT IN EACH PHASE
2. AT 1kHz 1.0V RMS
3. AT 25°C
4. M.T.B.F.=1/λp=1/(Failure Rate)

λp: Failure Rate

$$\lambda p=\left(\frac{t^2}{\alpha_\beta{}^3}+\frac{1}{\alpha_\omega}\right)\times 10^6$$

t : 5000h

α_β : 59600h (AT25°C)

α_ω : 1.2×10²h (AT25°C)

The calculation is based on the MIL-HDBK-217E

5. FRONT AND REAR END BELL:ALUMINUM.
6. SHAFT AND THREAD:STAINLESS STEEL(SUS303)
7. SCREWS: STEEL
8. THE LOAD ON THE MOTORAXIS IS 1kg IN Z DIRECTION.
9. FLAMMABILITY CLAAS 94HB

REVISIONS		
No.	DESCRIPTION	DATE / SIGN

DS'D M.Kubota	DATE 97.11.14	MODEL NO. TS3214N28	TITLE STEP MOTOR SPECIFICATIONS
CH'D T.Miyazawa	SCALE 1/1	3RD ANGLE PROJECTION	
APP'D	DWG NO. 032140044S30		SHEET 1

TSC 2019-04-24 PC

JIS A3(297X420)

TAMAGAWA SEIKI CO.,LTD.

Appendix D

Stepper Card Datasheet

A4988

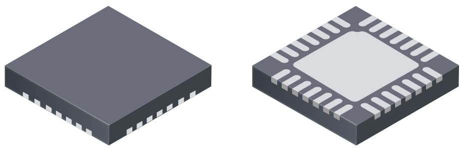
DMOS Microstepping Driver with Translator And Overcurrent Protection

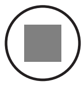
Features and Benefits

- Low $R_{DS(ON)}$ outputs
- Automatic current decay mode detection/selection
- Mixed and Slow current decay modes
- Synchronous rectification for low power dissipation
- Internal UVLO
- Crossover-current protection
- 3.3 and 5 V compatible logic supply
- Thermal shutdown circuitry
- Short-to-ground protection
- Shorted load protection
- Five selectable step modes: full, $^1/_2$, $^1/_4$, $^1/_8$, and $^1/_{16}$

Package:

28-contact QFN
with exposed thermal pad
5 mm × 5 mm × 0.90 mm
(ET package)





Approximate size

Description

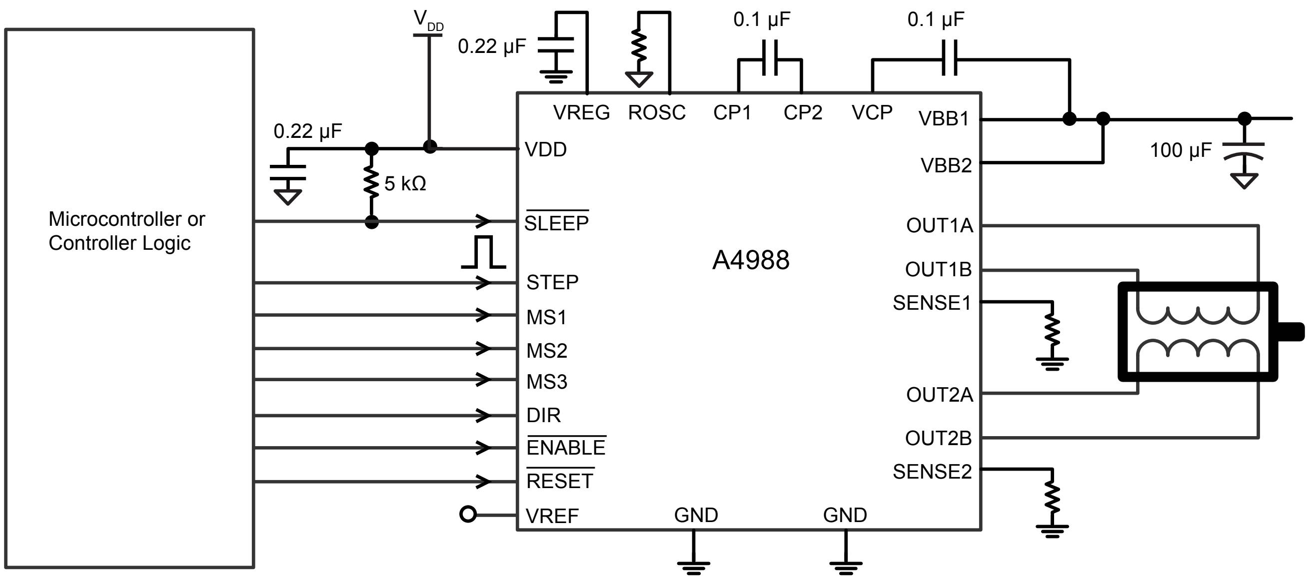
The A4988 is a complete microstepping motor driver with built-in translator for easy operation. It is designed to operate bipolar stepper motors in full-, half-, quarter-, eighth-, and sixteenth-step modes, with an output drive capacity of up to 35 V and ±2 A. The A4988 includes a fixed off-time current regulator which has the ability to operate in Slow or Mixed decay modes.

The translator is the key to the easy implementation of the A4988. Simply inputting one pulse on the STEP input drives the motor one microstep. There are no phase sequence tables, high frequency control lines, or complex interfaces to program. The A4988 interface is an ideal fit for applications where a complex microprocessor is unavailable or is overburdened.

During stepping operation, the chopping control in the A4988 automatically selects the current decay mode, Slow or Mixed. In Mixed decay mode, the device is set initially to a fast decay for a proportion of the fixed off-time, then to a slow decay for the remainder of the off-time. Mixed decay current control results in reduced audible motor noise, increased step accuracy, and reduced power dissipation.

Continued on the next page…

Typical Application Diagram



Description (continued)

Internal synchronous rectification control circuitry is provided to improve power dissipation during PWM operation. Internal circuit protection includes: thermal shutdown with hysteresis, undervoltage lockout (UVLO), and crossover-current protection. Special power-on sequencing is not required.

The A4988 is supplied in a surface mount QFN package (ES), 5 mm × 5 mm, with a nominal overall package height of 0.90 mm and an exposed pad for enhanced thermal dissipation. It is lead (Pb) free (suffix –T), with 100% matte tin plated leadframes.

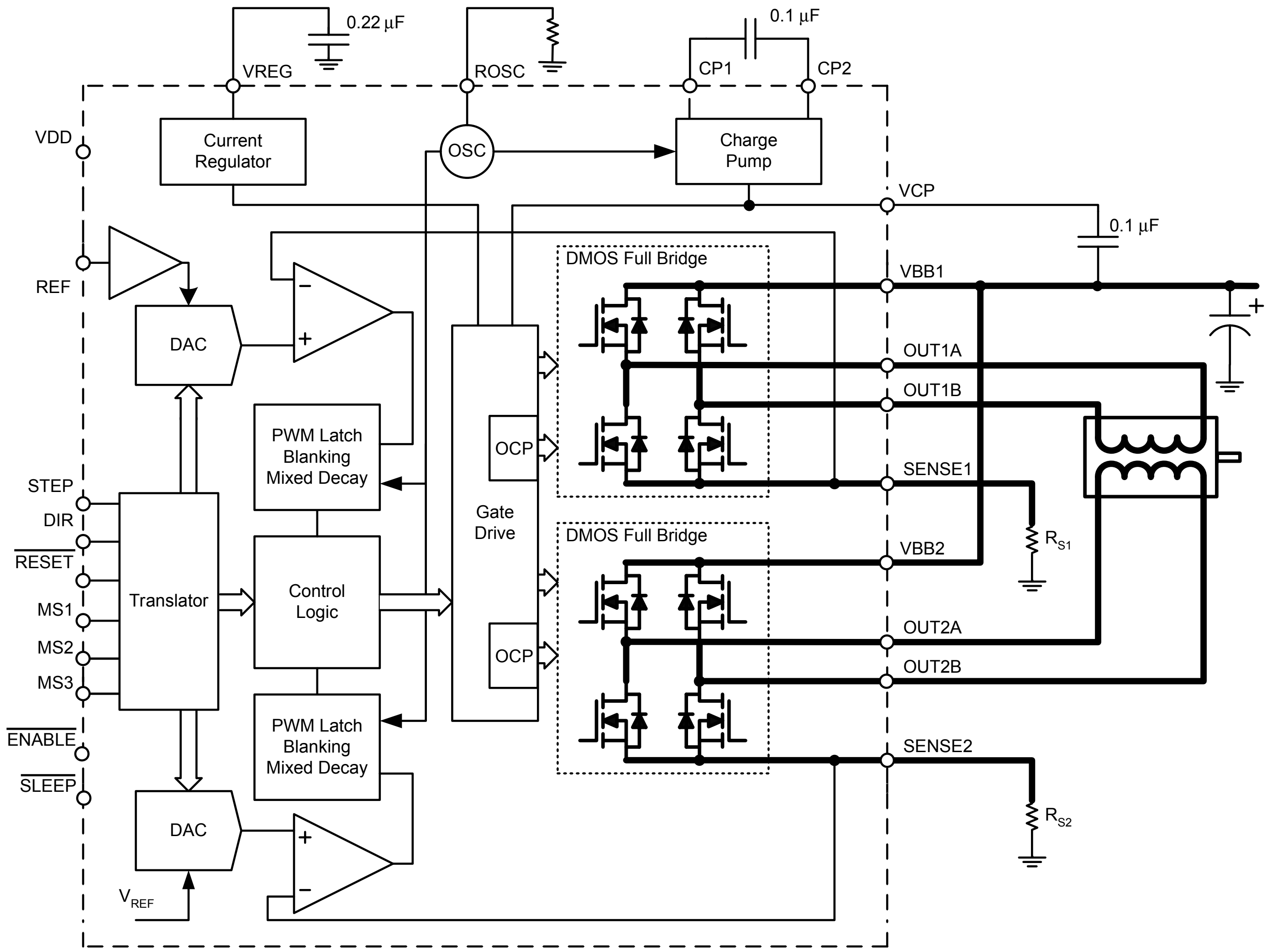
Selection Guide

Part Number	Package	Packing
A4988SETTR-T	28-contact QFN with exposed thermal pad	1500 pieces per 7-in. reel

Absolute Maximum Ratings

Characteristic	Symbol	Notes	Rating	Units
Load Supply Voltage	V_{BB}		35	V
Output Current	I_{OUT}		±2	A
Logic Input Voltage	V_{IN}		–0.3 to 5.5	V
Logic Supply Voltage	V_{DD}		–0.3 to 5.5	V
Motor Outputs Voltage			–2.0 to 37	V
Sense Voltage	V_{SENSE}		–0.5 to 0.5	V
Reference Voltage	V_{REF}		5.5	V
Operating Ambient Temperature	T_A	Range S	–20 to 85	ºC
Maximum Junction	T_J(max)		150	ºC
Storage Temperature	T_{stg}		–55 to 150	ºC

Functional Block Diagram



ELECTRICAL CHARACTERISTICS[1] at T_A = 25°C, V_{BB} = 35 V (unless otherwise noted)

Characteristics	Symbol	Test Conditions	Min.	Typ.[2]	Max.	Units
Output Drivers						
Load Supply Voltage Range	V_{BB}	Operating	8	–	35	V
Logic Supply Voltage Range	V_{DD}	Operating	3.0	–	5.5	V
Output On Resistance	R_{DSON}	Source Driver, I_{OUT} = –1.5 A	–	320	430	mΩ
		Sink Driver, I_{OUT} = 1.5 A	–	320	430	mΩ
Body Diode Forward Voltage	V_F	Source Diode, I_F = –1.5 A	–	–	1.2	V
		Sink Diode, I_F = 1.5 A	–	–	1.2	V
Motor Supply Current	I_{BB}	f_{PWM} < 50 kHz	–	–	4	mA
		Operating, outputs disabled	–	–	2	mA
Logic Supply Current	I_{DD}	f_{PWM} < 50 kHz	–	–	8	mA
		Outputs off	–	–	5	mA
Control Logic						
Logic Input Voltage	$V_{IN(1)}$		V_{DD}×0.7	–	–	V
	$V_{IN(0)}$		–	–	V_{DD}×0.3	V
Logic Input Current	$I_{IN(1)}$	V_{IN} = V_{DD}×0.7	–20	<1.0	20	μA
	$I_{IN(0)}$	V_{IN} = V_{DD}×0.3	–20	<1.0	20	μA
Microstep Select	R_{MS1}	MS1 pin	–	100	–	kΩ
	R_{MS2}	MS2 pin	–	50	–	kΩ
	R_{MS3}	MS3 pin	–	100	–	kΩ
Logic Input Hysteresis	$V_{HYS(IN)}$	As a % of V_{DD}	5	11	19	%
Blank Time	t_{BLANK}		0.7	1	1.3	μs
Fixed Off-Time	t_{OFF}	OSC = VDD or GND	20	30	40	μs
		R_{OSC} = 25 kΩ	23	30	37	μs
Reference Input Voltage Range	V_{REF}		0	–	4	V
Reference Input Current	I_{REF}		–3	0	3	μA
Current Trip-Level Error[3]	err_I	V_{REF} = 2 V, $\%I_{TripMAX}$ = 38.27%	–	–	±15	%
		V_{REF} = 2 V, $\%I_{TripMAX}$ = 70.71%	–	–	±5	%
		V_{REF} = 2 V, $\%I_{TripMAX}$ = 100.00%	–	–	±5	%
Crossover Dead Time	t_{DT}		100	475	800	ns
Protection						
Overcurrent Protection Threshold[4]	I_{OCPST}		2.1	–	–	A
Thermal Shutdown Temperature	T_{TSD}		–	165	–	°C
Thermal Shutdown Hysteresis	T_{TSDHYS}		–	15	–	°C
VDD Undervoltage Lockout	V_{DDUVLO}	V_{DD} rising	2.7	2.8	2.9	V
VDD Undervoltage Hysteresis	$V_{DDUVLOHYS}$		–	90	–	mV

[1]For input and output current specifications, negative current is defined as coming out of (sourcing) the specified device pin.

[2]Typical data are for initial design estimations only, and assume optimum manufacturing and application conditions. Performance may vary for individual units, within the specified maximum and minimum limits.

[3]$V_{ERR} = [(V_{REF}/8) - V_{SENSE}] / (V_{REF}/8)$.

[4]Overcurrent protection (OCP) is tested at T_A = 25°C in a restricted range and guaranteed by characterization.

Appendix E

Code

```
/**
 * CODE BY JOEL BERGMA AND JONAS LIND
 * All right reserved
 *
 */
 //Globala Variabler
 unsigned long t1;
 unsigned long t2;
 unsigned long pulse_width;
 const int hastighet = 30;
 const int motorDelay = 3;
 int Gi = 0;
 bool GV = true;

//Deklaration
///////////////////STEG MOTOR VARIABLER///////////////////////////
const int stepPinVD = 52;
const int dirPinVD = 53;
const int stepPinHD = 24;
const int dirPinHD = 25;
const int stepPinVB = 50;
const int dirPinVB = 51;
const int stepPinHB = 22;
const int dirPinHB = 23;

///////////////////SENSOR VARIABLER//////////////////////////////////
////////Front////////////
const int sensEchoFr = 26;
const int sensTrigFr = 27;
////////Right///////////
const int sensEchoHsFr = 28;
const int sensTrigHsFr = 29;

const int sensEchoHsSi = 30;
const int sensTrigHsSi = 31;
////////Left//////////
const int sensEchoVsFr = 34;
const int sensTrigVsFr = 35;

const int sensEchoVsSi = 36;
const int sensTrigVsSi = 37;
///////Down///////////
const int sensEchoFrNr = 42;
const int sensTrigFrNr = 43;

const int sensEchoHsNr = 44;
const int sensTrigHsNr = 45;

const int sensEchoVsNr = 46;
const int sensTrigVsNr = 47;

///////////////////LÄNGD VARIABLER//////////////////////////////////
// Anything over 400 cm (23200 us pulse) is "out of range"
const unsigned int MAX_DISTUPP =580;
const unsigned int MAX_DISTNER =580;
const unsigned int MAX_SIDA =580;
const unsigned int MAX_FRAMSIDOR =1050;


///////////// SVÄNG 90 GRADER FRAM FUNKTION /////////////////
```

```
//För att damsugaren ska svänga 90 grader, behöver ena hjulet röra sig 475
mm.
//Detta motsvarar 505 steg
int vanSvangFram(){
  int i = 0;
  while(i < 505){
  digitalWrite(dirPinHD, LOW);
  digitalWrite(stepPinHD,HIGH);
  delayMicroseconds(motorDelay);
  digitalWrite(stepPinHD,LOW);
  int cnt = 0;
  while (cnt < 6){
  digitalWrite(stepPinHB,HIGH);
  digitalWrite(stepPinVB,HIGH);
  delayMicroseconds(motorDelay/6);
  digitalWrite(stepPinHB,LOW);
  digitalWrite(stepPinVB,LOW);
  cnt = cnt+1;
  delay(hastighet/6);                  // Kommenterar bort och ersätter med
en av sensorerna. på 8 steg har alla gåtts genom
   }
  i = i +1;
  }
}

int hogSvangFram(){
  int i = 0;
  while(i < 505){
  digitalWrite(dirPinVD, LOW);
  digitalWrite(stepPinVD,HIGH);
  delayMicroseconds(motorDelay);
  digitalWrite(stepPinVD,LOW);
  int cnt = 0;
  while (cnt < 6){
  digitalWrite(stepPinHB,HIGH);
  digitalWrite(stepPinVB,HIGH);
  delayMicroseconds(motorDelay/6);
  digitalWrite(stepPinHB,LOW);
  digitalWrite(stepPinVB,LOW);
  cnt = cnt+1;
  delay(hastighet/6);                  // Kommenterar bort och ersätter med
en av sensorerna. på 8 steg har alla gåtts genom
 }
  i = i +1;
  }
}


///////////// SVÄNG 90 GRADER BAK FUNKTION /////////////////
//För att damsugaren ska svänga 90 grader, behöver ena hjulet röra sig 475
mm.
//Detta motsvarar 505 steg
int hogSvangBak(){
  int i = 0;
  while(i < 505){
  digitalWrite(dirPinHD, HIGH);
  digitalWrite(stepPinHD,HIGH);
  delayMicroseconds(motorDelay);
  digitalWrite(stepPinHD,LOW);
  delayMicroseconds(motorDelay);
  delay(hastighet);
```

```
  i = i +1;
  }
}

int vanSvangBak(){
  int i = 0;
  while(i < 505){
  digitalWrite(dirPinVD, HIGH);
  digitalWrite(stepPinVD,HIGH);
  delayMicroseconds(motorDelay);
  digitalWrite(stepPinVD,LOW);
  delayMicroseconds(motorDelay);
  delay(hastighet);
  i = i +1;
  }
}



////////Kollar sensor Sidorna///////////////////////
int sensSidornaKoll(int inputEcho, int inputTrig ){
  // const int tempSensor = input
  // Measure how long the echo pin was held high (pulse width)
  // Note: the micros() counter will overflow after ~70 min
  int x = 1;
  digitalWrite(inputTrig, HIGH);            // Skickar ut en signal
0.000010 s
  delayMicroseconds(10);
  digitalWrite(inputTrig, LOW);

  while (digitalRead(inputEcho) == 0);      // Signalen har inte kommit
tillbaka tid tagning startar
    t1 = micros();
  while (x == 1){                           // Signalen har kommit tillbaka
    x = digitalRead(inputEcho);             // Slår över till 0 då signalen
är tillbaka
    t2 = micros();
    if(t2-t1 > 5800){                        // Gränsen satt strax över för
entydigt svar
      x = 0;                                // Om tiden går längre än
mätavståndet avbryts den.
    }
  }
  pulse_width = t2 - t1;
  return pulse_width;
  delay(10);
}



/////////////KOLLAR SENSORER FRAMFÅT FUNKTION/////////////////////////
int sensUppKoll(int inputEcho, int inputTrig ){
  // const int tempSensor = input
  // Measure how long the echo pin was held high (pulse width)
  // Note: the micros() counter will overflow after ~70 min
  int x = 1;
  digitalWrite(inputTrig, HIGH);            // Skickar ut en signal
0.000010 s
  delayMicroseconds(10);
  digitalWrite(inputTrig, LOW);
```

```
  while (digitalRead(inputEcho) == 0);      // Signalen har inte kommit
tillbaka tid tagning startar
    t1 = micros();
  while (x == 1){                           // Signalen har kommit tillbaka
    x = digitalRead(inputEcho);             // Slår över till 0 då signalen
är tillbaka
    t2 = micros();
    if(t2-t1 > 600){                        // Gränsen satt strax över för
entydigt svar
      x = 0;                                // Om tiden går längre än
mätavståndet avbryts den.
    }
  }
  pulse_width = t2 - t1;
  if ( pulse_width > MAX_DISTUPP ) {
    pinMode(4, OUTPUT);
    digitalWrite(4, LOW);
    return true;
  } else {
    pinMode(4, OUTPUT);
    digitalWrite(4, HIGH);
    return false;
  }
  // Wait at least 60ms before next measurement
  // Att mäta 17 cm tar tur och retur 0.1 s. för 800 cm rekommenderas delay
60 ms. Därför kan detta minskas till
  // 40 / 800 = 1 / 20 => 60 / 20 = 3 ms, Testar 10 ms eftersom ljudet ändå
tar 100 ms.
  delay(10);
}

//////////////////////Kolla sensorna sidor fram////////////////////
int sensFramSidaKoll(int inputEcho, int inputTrig ){
  // const int tempSensor = input
  // Measure how long the echo pin was held high (pulse width)
  // Note: the micros() counter will overflow after ~70 min
  int x = 1;
  digitalWrite(inputTrig, HIGH);            // Skickar ut en signal
0.000010 s
  delayMicroseconds(10);
  digitalWrite(inputTrig, LOW);

  while (digitalRead(inputEcho) == 0);      // Signalen har inte kommit
tillbaka tid tagning startar
    t1 = micros();
  while (x == 1){                           // Signalen har kommit tillbaka
    x = digitalRead(inputEcho);             // Slår över till 0 då signalen
är tillbaka
    t2 = micros();
    if(t2-t1 > 1100){                        // Gränsen satt strax över för
entydigt svar
      x = 0;                                // Om tiden går längre än
mätavståndet avbryts den.
    }
  }
  pulse_width = t2 - t1;
  if ( pulse_width > MAX_FRAMSIDOR ) {
    pinMode(4, OUTPUT);
    digitalWrite(4, LOW);
    return true;
  } else {
```

```
    pinMode(4, OUTPUT);
    digitalWrite(4, HIGH);
    return false;
  }
  // Wait at least 60ms before next measurement
  // Att mäta 17 cm tar tur och retur 0.1 s. för 800 cm rekommenderas delay
60 ms. Därför kan detta minskas till
  // 40 / 800 = 1 / 20 => 60 / 20 = 3 ms, Testar 10 ms eftersom ljudet ändå
tar 100 ms.
  delay(10);
}

//////////////KOLLAR SENSORER NER FUNKTION////////////////////////// P.S.S
int sensNerKoll(int inputEcho, int inputTrig ){
  //const int tempSensor = input
  // Measure how long the echo pin was held high (pulse width)
  // Note: the micros() counter will overflow after ~70 min

  int x = 1;
  t1 = micros();
  digitalWrite(inputTrig, HIGH);
  delayMicroseconds(10);
  digitalWrite(inputTrig, LOW);

  while ( digitalRead(inputEcho) == 0 );
    t1 = micros();
  while ( x == 1){
    x = digitalRead(inputEcho);
    t2 = micros();
    if(t2-t1 > 600){
      x = 0;
    }
  }
  pulse_width = t2 - t1;
  if ( pulse_width < MAX_DISTNER ) {
    pinMode(4, OUTPUT);
    digitalWrite(4, LOW);
    return true;
  } else {
    pinMode(4, OUTPUT);
    digitalWrite(4, HIGH);
    return false;
  }

  delay(10);
}
/////////////////////Kollarsensorer/////////////////////
int checkSense(){
  // Check av Sensorer UT Anledning till att de flimrar är för att de
hoppar ut funktionen till nästa. 8 ggr / varv
  if (Gi == 0){
   tittaUppFrNerVsHs();     // Kollar sensorn som är uppe och riktad i
körriktningen, returnerar false om detektering
  }
  if (Gi == 25){
   tittaUppVsHsNerFr();         //Kollar sensorn som är på undersidan, inte
inkopplad ännu
  }
  if (Gi == 75){
    Gi = -1;
  }
```

```
}

/////////////// Funktion upp fram sida och ner framhjul /////////////////
int tittaUppVsHsNerFr(){
  bool x = true;
  bool y = true;
  bool z = true;
  x = sensNerKoll(sensEchoFrNr, sensTrigFrNr);
  y = sensFramSidaKoll(sensEchoHsFr, sensTrigHsFr);
  z = sensFramSidaKoll(sensEchoVsFr, sensTrigVsFr);

  //Om y = hinder höger sida, sväng då vänster
  if(y == false){
    vanSvangFram();
  }
  //Om z = hinder vänster sida, sväng då höger
  if(z == false){
    hogSvangFram();
  }
  //Om x = trappa fram ......
  if(x == false){
    kBak();
    int LVHS = sensSidornaKoll(sensEchoHsSi, sensTrigHsSi);
    int LVVS = sensSidornaKoll(sensEchoVsSi, sensTrigVsSi);

    if(LVHS <= LVVS){
      vanSvangFram();
    }
    else{
      hogSvangFram();
    }
  }

}
///////////////// Funktion titta upp center och ner framför sidohjul
//////////////////
int tittaUppFrNerVsHs(){
  bool x = true;
  bool y = true;
  bool z = true;
  x = sensUppKoll(sensEchoFr, sensTrigFr);
  y = sensNerKoll(sensEchoVsNr, sensTrigVsNr);
  z = sensNerKoll(sensEchoHsNr, sensTrigHsNr);

  if(x == false){
    int LVHS = sensSidornaKoll(sensEchoHsSi, sensTrigHsSi);
    int LVVS = sensSidornaKoll(sensEchoVsSi, sensTrigVsSi);

   if(LVHS < LVVS){
      vanSvangFram();
    }
   if(LVHS > LVVS){
      hogSvangFram();
   }
   if( LVHS == LVVS){
    hogSvangFram();
   }

  }
  if(y == false){
```

```
    hogSvangBak();
  }
  if(z == false){
    vanSvangBak();
  }

}
///////////////Köra rakt fram////////////////
void kRaktfram(){

   digitalWrite(dirPinVD, LOW);
   digitalWrite(dirPinHD, LOW);
   digitalWrite(stepPinVD,HIGH);
   digitalWrite(stepPinHD,HIGH);
   delayMicroseconds(motorDelay);
   digitalWrite(stepPinVD,LOW);
   digitalWrite(stepPinHD,LOW);
   int cnt = 0;
   while (cnt < 6){
   digitalWrite(stepPinHB,HIGH);
   digitalWrite(stepPinVB,HIGH);
   delayMicroseconds(motorDelay/6);
   digitalWrite(stepPinHB,LOW);
   digitalWrite(stepPinVB,LOW);
   cnt = cnt+1;
   delay(hastighet/6);                  // Kommenterar bort och ersätter
med en av sensorerna. på 8 steg har alla gåtts genom
   }




}
//////////////////////// Bakar damsugaren ett halvt
varv////////////////////////////////////////
void kBak() {
  int i = 0;
  while(i <= 150){
   digitalWrite(dirPinVD, HIGH);
   digitalWrite(dirPinHD, HIGH);
   digitalWrite(stepPinVD,HIGH);
   digitalWrite(stepPinHD,HIGH);
   delayMicroseconds(motorDelay);
   digitalWrite(stepPinVD,LOW);
   digitalWrite(stepPinHD,LOW);
   delayMicroseconds(motorDelay);
   delay(hastighet);        // Kommenterar bort och ersätter med en av
sensorerna. på 8 steg har alla gåtts genom
   i = i+1;
  }
}
void setup() {
  /////////// TILL STEGMOTOR STYRNING PIN DEFINITION/////////////////////
  pinMode(stepPinVD,OUTPUT);
  pinMode(dirPinVD,OUTPUT);
  pinMode(stepPinHD,OUTPUT);
  pinMode(dirPinHD,OUTPUT);
  pinMode(stepPinVB,OUTPUT);
  pinMode(dirPinVB,OUTPUT);
  pinMode(stepPinHB,OUTPUT);
  pinMode(dirPinHB,OUTPUT);
```

```
  //////////////TILL SENSOR PIN DEFINITION//////////////////////////
  // The Trigger pin will tell the sensor to range find
  pinMode(sensTrigFr, OUTPUT);
  digitalWrite(sensTrigFr, LOW);
  pinMode(sensTrigHsFr, OUTPUT);
  digitalWrite(sensTrigHsFr, LOW);
  pinMode(sensTrigHsSi, OUTPUT);
  digitalWrite(sensTrigHsSi, LOW);
  pinMode(sensTrigVsFr, OUTPUT);
  digitalWrite(sensTrigVsFr, LOW);
  pinMode(sensTrigVsSi, OUTPUT);
  digitalWrite(sensTrigVsSi, LOW);
  pinMode(sensTrigFrNr, OUTPUT);
  digitalWrite(sensTrigFrNr, LOW);
  pinMode(sensTrigHsNr, OUTPUT);
  digitalWrite(sensTrigHsNr, LOW);
  pinMode(sensTrigVsNr, OUTPUT);
  digitalWrite(sensTrigVsNr, LOW);
}
////////////// HUVUDPROGRAMMET //////////////////////////
//Denna loop körs hela tiden, som en while slinga
void loop() {

while(Gi <= 75){
 if (Gi != 0 or Gi != 25){
  kRaktfram();          //Gör att damsugaren kör rakt fram
 }
 // Check av Sensorer UT Anledning till att de flimrar är för att de hoppar
ut funktionen till nästa. 8 ggr / varv
  if (Gi == 0){
   tittaUppFrNerVsHs();     // Kollar sensorn som är uppe och riktad i
körriktningen, returnerar false om detektering
  }
  if (Gi == 25){
   tittaUppVsHsNerFr();         //Kollar sensorn som är på undersidan, inte
inkopplad ännu
  }
  if (Gi == 75){
    Gi = -1;
  }
 //checkSense();
 Gi = Gi+1;
}
}
```

Appendix F

Ultra Sonic Sensor Datasheet

Ultrasonic Ranging Module HC - SR04

Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:
(1) Using IO trigger for at least 10us high level signal,
(2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
(3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.
Test distance = (high level time×velocity of sound (340M/S) / 2,

Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

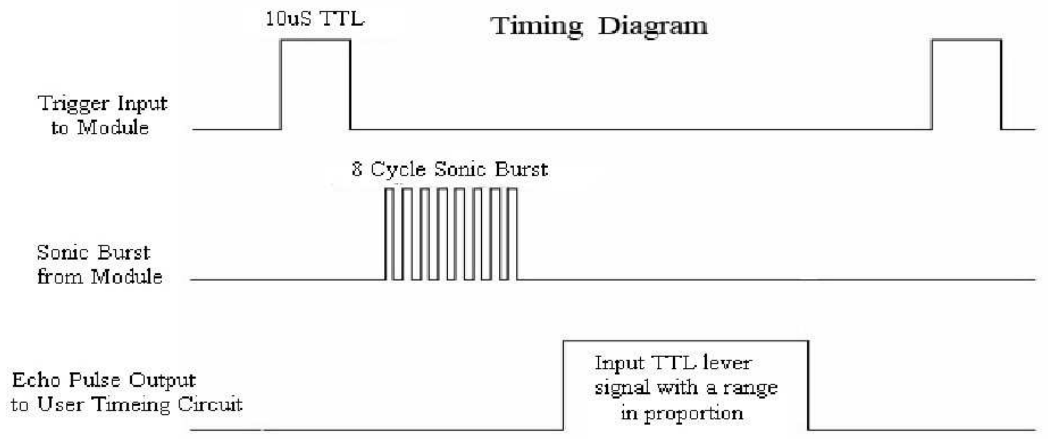
Electric Parameter

Working Voltage	**DC 5 V**
Working Current	**15mA**
Working Frequency	**40Hz**
Max Range	**4m**
Min Range	**2cm**
MeasuringAngle	**15 degree**
Trigger Input Signal	**10uS TTL pulse**
Echo Output Signal	**Input TTL lever signal and the range in proportion**
Dimension	**45*20*15mm**

Vcc Trig Echo GND

Timing diagram

The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion .You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: uS / 58 = centimeters or uS / 148 =inch; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



Attention:

- The module is not suggested to connect directly to electric, if connected electric, the GND terminal should be connected the module first, otherwise, it will affect the normal work of the module.
- When tested objects, the range of area is not less than 0.5 square meters and the plane requests as smooth as possible, otherwise ,it will affect the results of measuring.

www.Elecfreaks.com

Appendix G

Market analysis

Marknadsundersökning

Prestanda: Menuett 999:- Jula

- Timerstyrd
- 4 städmönster, slump (väljer mellan programmen), spiral, kant, S-form
- Anpassad sugstyrka
- Borstar utbytbara
- 400 ml behållare, hepa f5 filter
- 14.4 V 2 Ah NiMH

Drifttid: 90 min

Storlek: h = 85 mm, diameter 340 mm, 3,4 kg

Sensorer: 5 krocksensorer som stannar innan hinder, 5 trappscensorer

Övrigt: Schemastyrning, 230 V ladd station, 60 dB, Klarar tröskel 10 mm


Prestanda: Dyson 360 9190:-

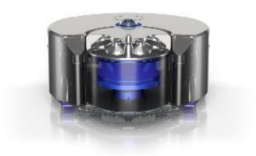
- 330 ml behållare, hepa f5 filter
- Sugeffekt 20AW

Drifttid: 45 och 75 minuter

Storlek: 120 x 242 x 230

Sensorer:

Övrigt: ladd station, IR-sensorer, APP, larvband för exakthet, nylonborstar



Prestanda: Cleanmate s800 1990:- Claes Ohlson

- Slumpvis navigering, kant, plats,
- Schemaläggning
- Fjärrkontroll
- 500 ml behållare, hepa
- 14.4 V 2 Ah NiMH
- Borstar

Drifttid: 90 min , laddtid 4 h

Storlek: h = 88 mm, diameter 340 mm, 3 kg

Sensorer: krocksensorer , trappscensorer

Övrigt: 60 dB, UV ljus ultimat rengöring, moppning, magnetslinga för begränsning

Prestanda: Cleanmate s800 1990:- Claes Ohlson

- Slumpvis, spiral, längs vägg. Z form
- Schemaläggning
- Fjärrkontroll
- 300 ml behållare, hepa , borstar
- 7.4 V 2.2 Ah NiMH Laddtid 4 h
- Borstar

Drifttid: 90 min ,

Storlek: 81 x 255 x 243 1.2 kg

Sensorer: trappscensorer

Övrigt:

Prestanda:
Ex1: Sugkraft 1500 pa, Dammkapacitet: 0.45L
Ex2: Dammkapacitet: 0,30L
Ex3: Dammkapacitet: 0.5

Driftstid:
Ex1: Batteri kapacitet: 2000mha, litium, Drifttid: (1.5-2h),
Ex2: Batteri, litiumjon, drifttid: 120min, 12.8 V
Ex3: Batteri kapacitet: 2600mah, litium, 120min, 14.6V

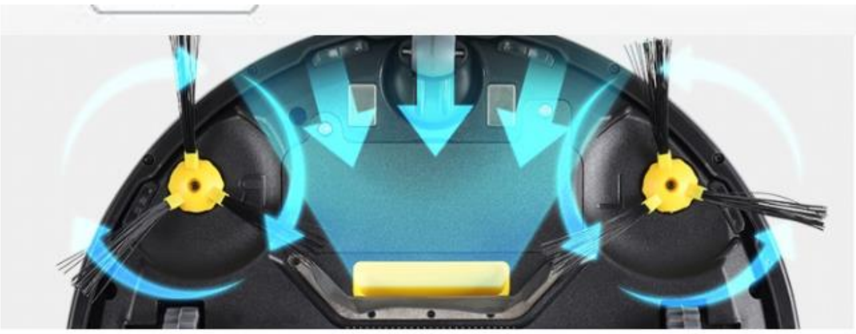
Storlek och vikt:
Ex1: D: 34cm, H: 9cm, vikt 4.1kg
Ex2: D: 33cm, H: 9cm, 1.73kg
Ex3: D: 35cm, H: 8.5cm, 3.5 kg

Sensorer:
Ex1: Damsensor, antistöt detektor, antifall detektor,
Ex2: Damm full-indikator, dammsensor, trappsensor, 6 st sensorer för att undvika hinder
Ex3: ir sensorer, 5st avsatts sensorer.

Kostnad:
Ex1: ca 2500kr

Ex2: 2889 kr
Ex3: 5490kr

Insug:
Ex1:



Ex:3

Ex:4

TRITA 2019:28