



Open your mind. LUT.
Lappeenranta University of Technology

**DIESELMOOTTORIN ANALOGISEN KIERROSNOPEUS-
MITTAUKSEN DIGITALISOINTI**
Digitizing an analog tachometer of a diesel engine

Janne Rossi

TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto
Teknillinen tiedekunta
Sähkötekniikan koulutusohjelma

Janne Rossi

Dieselmoottorin analogisen kierrosnopeusmittauksen digitalisointi

2011

Kandidaatintyö.

33 sivua, 24 kuvaa, 7 taulukkoa ja 2 liitettä.

Tarkastaja: Professori Pertti Silventoinen
TKT Mikko Kuisma

Hakusanat: Kierrosnopeuden mittaaminen, induktioanturi, askelmoottori, H -silta

Työssä suunnitellaan kierrosnopeusmittauksen toteutus vanhaan Ford -teollisuusdieseliin käyttäen moottorin alkuperäistä kierrosnopeusanturia. Anturi kunnostetaan vaihtamalla vanha, palanut käämi uuteen digitaaliseen järjestelmään soveltuvaan käämiin. Sen toiminta halutulla kierrosnopeusalueella varmistetaan mittauksin ja tämän perusteella suunnitellaan kytkentä sen liittämiseksi kierrosnopeuden laskevaan mikrokontrolleriin. Kytkennän toimivuutta testataan simuloimalla ennen prototyypin rakentamista.

Erilaisia vaihtoehtoja analogisen näytön toteuttamiseksi tutkitaan ja niistä valitaan yksi venekäyttöön soveltuva, joka toteutetaan järjestelmän näyttöratkaisuksi. Järjestelmälle suunnitellaan piirilevy, jolle prototyyppi kasataan. Mikrokontrollerille koodataan C - ohjelmointikielellä ohjelma, joka laskee dieselmoottorin kierrosnopeuden anturipulssien perusteella ja ohjaa näyttöä.

ABSTRACT

Lappeenranta University of Technology
Faculty of Technology
Degree Program in Electrical Engineering

Janne Rossi
Digitizing an analog tachometer of a diesel engine

2011

Bachelor's Thesis.
33 pages, 24 pictures, 7 tables and 2 appendices.

Examiner: Professor Pertti Silventoinen
TKT Mikko Kuisma

Keywords: Tachometer, inductive sensor, stepper motor, H -bridge

In this paper a digital tachometer system is designed for Ford industrial diesel engine to replace old analog system. The original inductive sensor is modernized to fulfill the sensor needs of a digital system. Proper operation of the sensor in the rpm range of the diesel engine is confirmed with measurements. A circuit to connect the sensor to a microcontroller is designed and its operation is tested with simulations.

Different approaches for an analog display are examined, and one fit for marine use is produced as the display of the system. Circuit board is designed for the whole system and the prototype is assembled and tested. A program for the microcontroller is written in C - programming language to calculate the engine speed of the diesel engine and to control the display.

SISÄLLYSLUETTELO

1.	JOHDANTO	7
2.	KIERROSNOPEUDEN MITTAMINEN SÄHKÖISESTI.....	8
2.1	Analoginen toteutus	8
2.2	Digitaalinen toteutus	8
3.	ANTUROINTI.....	9
3.1	Alkuperäisen anturin kunnostus	12
3.2	Anturille suoritettut mittaukset	15
3.3	Anturisignaalin käsittely	18
4.	NÄYTTÖ	21
4.1	Askelmoottori	21
4.2	H -silta.....	26
5.	PIIRILEVY.....	26
6.	DATAN PROSESSOINTI	28
6.1	Ohjelmakoodi.....	29
7.	YHTEENVETO JA JOHTOPÄÄTÖKSET.....	31
	LÄHTEET.....	33
LIITE I	Mikrokontrollerin ohjelmakoodin vuokaavio	
LIITE II	Mikrokontrollerin ohjelmakoodi	

KÄYTETYT MERKINNÄT JA LYHENTEET

ADC	Analog to Digital Conversion
BJT	Bipolar Junction Transistor
ESK	Elektroniikan suunnittelukeskus
LCD	Nestekidenäyttö, Liquid Crystal Display
MOSFET	Metal-Oxide-Semiconductor Field-Effect-Transistor
PWM	Pulssinleveysmodulaatio, Pulse Width Modulation
TI	Texas Instruments
<i>buffer</i>	koodissa käytetty puskurimuuttuja
<i>C</i>	kapasitanssi, kondensaattori
<i>D</i>	diode
<i>drive_or_count</i>	koodissa käytetty lippu-muuttuja
<i>e</i>	sähkömotorinen voima
<i>f</i>	taajuus
<i>flop</i>	koodissa käytetty viimeisen askelen tallentava muuttuja
<i>forward()</i>	koodissa toteutettu funktio, ajaa askelmootoria eteenpäin
<i>I</i>	virta
<i>k</i>	esijakaja
<i>M</i>	MOSFET (M1, M2, jne.)
<i>m</i>	ajastimen alkuarvo
<i>N</i>	kierrosmäärä, käämissä
<i>P</i>	teho
<i>Q</i>	transistori
<i>R</i>	resistanssi, vastus
<i>reverse()</i>	koodissa toteutettu funktio, ajaa askelmootoria taaksepäin
<i>T</i>	jaksonaika
<i>t</i>	aika
<i>U</i>	jännite, integroitu piiri
Φ	magneettivuo
ω	kulmanopeus

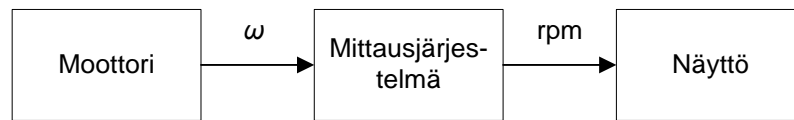
Alaindeksit

CC	käyttöjännite
clk	kellotaajuus

d	diodi (bias -jännite)
des	haluttu (desired) jaksonaika
f	fault, vika, oikosulkua kuvaava
ka	keskiarvo
max	maksimi
min	minimi
moottori	dieselmoottori
nokka	nokka-akseli
o	ulostulo, lähdön jännite
pulse	anturipulssi, pulssi
sekv	sekvenssin aika
zener	zener (jännite, virta)

1. JOHDANTO

Polttomoottorista saatava vääntömomentti on verrannollinen moottorin kierrosnopeuteen. Pääsääntöisesti suurin vääntö saadaan kierroslukualueen viimeisellä kolmanneksella, kun taas polttoainetalouden kannalta olisi edullista ajaa moottoria pienimmillä mahdollisilla kierrosnopeuksilla. Tästä johtuen on löydettävä sopiva kierrosnopeus, jolla vääntö juuri riittää sovelluksen tarpeisiin, mutta samalla kierrosnopeus on mahdollisimman alhainen. Käytännössä tämä tarkoittaa sitä, että käyttäjälle on välitettävä tietoa moottorin kierrosnopeudesta, jotta moottoria voidaan ajaa sopivalla kierrosnopeudella. Kierrosnopeusmittauksen yleistetty lohkokaavio on esitetty kuvassa 1.1.



Kuva 1.1 Kierrosnopeusmittauksen yleistetty lohkokaavio.

Kierrosnopeuden mittaus voidaan toteuttaa joko mekaanisesti tai sähköisesti. Mekaanisen toteutuksen ongelmiksi muodostuvat kalliit valmistuskustannukset ja mekaanisen kuluminen mukanaan tuoma huollon tarve. Sähköisessä järjestelmässä sen sijaan voidaan anturintiratkaisusta riippuen välttyä mekaaniselta kulumiselta ja näin ollen lisätä järjestelmän käyttövarmuutta.

Työn tavoitteena on suunnitella ja toteuttaa digitaalinen kierrosnopeuden mittausjärjestelmä Ford:n 2700-sarjan teollisuudieselmoottorille venekäyttöön alkuperäistä anturia käyttäen. Lisäksi loppukäyttäjän toiveena oli ”analogisen näköinen” viisariosoitukseen perustuva mittari kierrosalueelle 500 – 2500 rpm, jolla kierrosnopeus esitetään käyttäjälle. Digitaalisen järjestelmän etuna on helpompi kalibrointi ja suurempi häiriönsietokyky. Viisariosoitus puolestaan on ajoneuvokäytössä havainnollisempi kuin numeronäyttö.

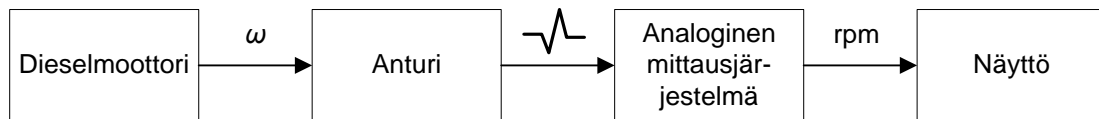
Työssä kunnostetaan dieselmoottorin alkuperäinen induktiivinen kierrosnopeusanturi ja kunnostetun anturin toiminta varmistetaan mittaamalla. Itse kierrosnopeuden laskenta toteutetaan mikrokontrollerilla, jonka tulokytkeä mitoitetaan anturin mittaustulosten perusteella. Kierrosnopeustiedon esittämiseksi käyttäjälle suunnitellaan kytkentä jolla voidaan ohjata askelmoottoria joka kääntää mittariston neulaa. Kytkentöjen toimivuutta simuloidaan OrCAD -simulointiympäristössä ja järjestelmälle suunnitellaan piirilevy. Prototyyppi kasataan ja sen toiminta testataan, mutta kotelointia ei toteuteta tämän työn piirissä.

2. KIERROSNÖPEUDEN MITTAMINEN SÄHKÖISESTI

Kierrosnopeuden mittaamiseen sähköisesti on monia tapoja, mutta toteutukset voidaan karkeasti jakaa analogisiin ja digitaalisiin mittausjärjestelmiin.

2.1 Analoginen toteutus

Ennen digitaalijärjestelmien yleistymistä analoginen järjestelmä oli ainoa vaihtoehto sähköisen kierrosnopeusmittauksen toteuttamisessa. Vaihtoehtoisina tapoina on käytännössä joko jännitteen tai virran mittaus ja tuloksen skaalaus kierrosluvuksi järjestelmän mukaan. Tämä käytännössä tarkoittaa sitä, että mittausjärjestelmän kalibrointi on hyvin voimakkaasti riippuvainen järjestelmän fyysisestä toteutuksesta. Esimerkiksi pidemmän kaapeloinnin lisääminen anturille voi vaikuttaa merkittävästi mittaustarkkuuteen. Myös anturin toteutus voi vaihdella erilaisista sähkömekaanisista toteutuksista induktiiviseen anturiin. Analogisen kierrosnopeusmittauksen yksinkertaistettu lohkokkaavio on esitetty kuvassa 2.1.

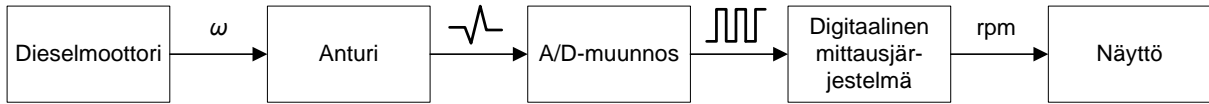


Kuva 2.1 Analogisen kierrosnopeusmittauksen yksinkertaistettu lohkokkaavio.

Työn aiheena olevan dieselmoottorin vanha kierrosnopeusmittaus on oletettavasti perustunut tasasuunnatun anturijännitteen mittaamiseen. Tällaisessa toteutuksessa mittarin kalibroiminen on varsin hankalaa, sillä induktiivisen anturin tuottama jännite ei kasva lineaarisesti, joten kierrosnopeuden mittaaminen on hankalaa jos halutaan lineaarisella asteikolla toimiva mittari. Vanhoissa järjestelmissä tosin usein mittarin asteikko on epälineaarinen juuri tämän vuoksi.

2.2 Digitaalinen toteutus

Digitaalisen kierrosnopeusmittauksen perustana on useimmiten akselin asentotiedon ja ajan yhdistäminen. Esimerkiksi yhdellä asentoanturilla voidaan laskea akselin kierrokset ennalta määritellyssä ajanjaksossa ja tämän tiedon perusteella laskea kierrosnopeus. Digitaalisen kierrosnopeusmittauksen periaate on esitetty lohkokkaaviona kuvassa 2.2.



Kuva 2.2 Digitaalisen kierrosnopeusmittauksen periaatteellinen lohkokkaavio.

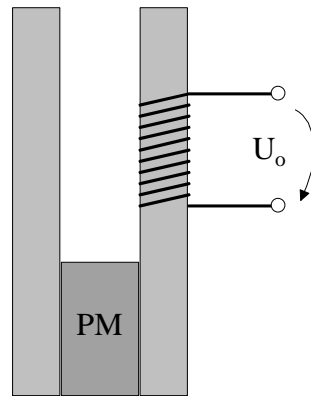
Valmiita asentoantureita on saatavilla laaja valikoima erilaisiin mittaustapoihin perustuen. Useimmin käytettyjä ovat induktio- ja Hall -anturit. Myös erilaisia enkooderitoteutuksia käytetään asentotiedon selvittämiseen. Nämä voivat perustua mekaaniseen kosketukseen, optisiin antureihin, induktioon tai Hall -ilmiöön. Enkooderi voi tarjota yksityiskohtaisempaa tietoa liikkeestä kuin yksi asentoanturi, ja niitä käytetäänkin lähinnä sovelluksissa joissa tarvitaan liikkeen tarkkaa hallintaa. Tällaisia käyttökohteita ovat esimerkiksi robotit, työstökoneet ja hissit. Yhteinen piirre näille antureille kuitenkin on se, että ne tuottavat pulssimaista signaalia, jota voidaan havainnoida digitaalisella järjestelmällä.

Digitaalisessa järjestelmässä kierrosnopeuden laskeminen on hyvin yksinkertainen operaatio. Anturipulssien määrä ajanjaksossa lasketaan ja sen perusteella lasketaan kierrosnopeus.

Myös näyttölaitteen implementointi on digitaalisessa järjestelmässä yksinkertaisempaa, kuin analogisessa, sillä voidaan hyödyntää esimerkiksi LCD- tai 7 -segmenttinäyttöjä. Sopivalla ohjauspiirillä ei tarvita montaa lähtöä kontrollerilta tarvittavaa nelinumeroista näyttöä varten. Joissain järjestelmissä, kuten ajoneuvosovelluksissa perinteinen analoginen viisarinäyttö on havainnollisempi, ja myös sen toteuttamiseen on tarjolla valmiita toteutuksia. Itse rakennettavana toteutuksena voisi olla esimerkiksi PWM -ohjattu jännitemittaukseen perustuva mittari tai DC-moottorilla ohjattava mekaaninen mittari, jota on alun perin pyöritetty vaijerilla. Myös askelmoottoreihin perustuvia toteutuksia on markkinoilla valmiina ratkaisuina.

3. ANTUROINTI

Anturin tehtävänä on toimia välittäjänä mittausjärjestelmän ja mitattavan ilmiön välillä. Alkuperäisessä analogisessa järjestelmässä induktiivisen anturin tuottama jännite mitattiin ja sen perusteella näytettiin käyttäjälle kierrosnopeustieto. Käytetty anturi on mekaaniselta rakenteeltaan yksinkertainen, se koostuu kestopagneetista, rautasydäimestä ja kelasta. Anturin rakenne on esitetty kuvassa 3.1 ja itse anturi on esitetty kuvassa 3.2.



Kuva 3.1 Kierrosnopeusanturin rakenne yksinkertaistettuna.



Kuva 3.2 Anturi ja sille herätteen antava sakarapyörä juuri dieselmoottorista purettuna.

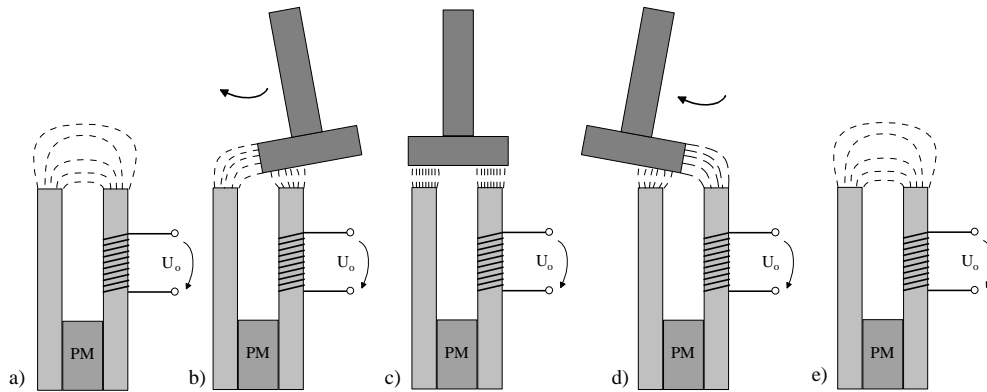
Kuvassa esitetyn anturille herätteen antavan sakarapyörän sakarat pienentävät huomattavasti U-sydämen ilmapäliä kulkiessaan sen ohi. Tällöin kelan läpi kulkeva magneettivuoto muuttuu. Vuon muutokset indusoivat kelan päiden välille jännitteen, joka on verrannollinen magneettivuon muutosnopeuteen ja käämikierrosten määrään, yhtälön 3.1 esittämällä tavalla.

$$U_o = -e_{smv} = N \cdot \frac{d\Phi}{dt} . \quad (3.1)$$

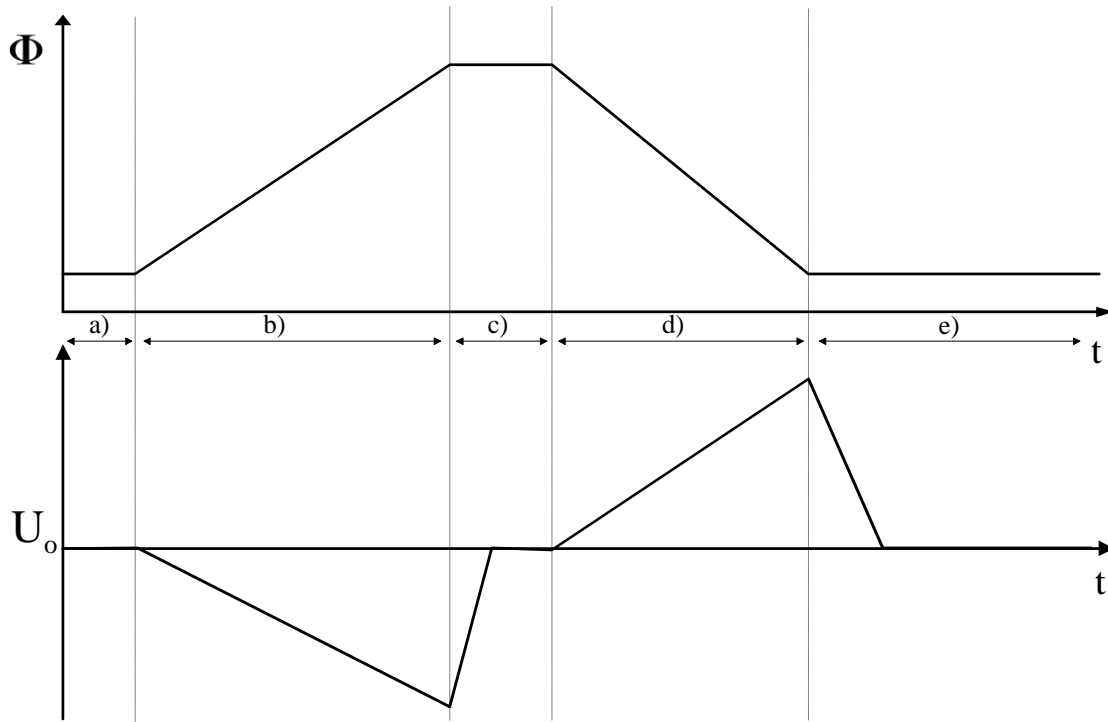
N on käämikierros määrä, Φ magneettivuoto ja e_{smv} käämiin indusoitunut magnetomotorinen voima. (Pyrhönen & Nerg 2004, 7.3) (Yeadon & Yeadon 2001, 1.8)

Kuva 3.3 havainnollistaa anturin toimintaa sakarapyörän pienentäessä ilmapäliä, eli laukaistessa sen. Suurin osa magneettivuosta kuluu ilmapäliissä ja rautaosien aiheuttamat häviöt voidaan olettaa häviävän pieniksi. Kun ilmapäli pienenee sakaran ohittaessa anturin rautasydämet, käämin lävistävä magneettivuoto kasvaa merkittävästi häviötä tuottavan

ilmavälin pienetessä, kuva 3.3 b). Vuon lisääntyminen aiheuttaa negatiivisen jännitteen indusoitumisen käämiin. Vuon ollessa hetkellisesti vakio, kuva 3.3 c), jännite käämin päiden välillä pienenee. Sakaran alkaessa etäännyä, pienenevä magneettivuoto indusoi käämiin positiivisen jännitteen. Vuon tasaantuessa vakioarvoon, jännite putoaa jälleen nol-
laan, kuva 3.3 d). Kuvassa 3.4 on esitetty yksinkertaistetusti magneettivuon ja jännitteen muutokset anturin toiminnan eri vaiheiden aikana.



Kuva 3.3 Anturin toimintaperiaate



Kuva 3.4 Anturin magneettivuon ja jännitteen muutoksen yhden toimintajakson aikana.

Vanha kierrosnopeuden mittausjärjestelmä perustui jännitteen mittaukseen. Tarkkaa tietoa vanhasta järjestelmästä ei ole, sillä anturi oli ainoa tallessa oleva osa, mutta yhtenä toteutusvaihtoehtona on tasasuunnata anturin tuottama jännite ja suodattaa se kondensaattorin avulla. Näin päästään huomattavasti tasaisempaan jännitteeseen, kuin suoraan

anturilta mitattaessa. Lisäksi tällaisella toteutuksella päästään eroon negatiivisesta jännitepulssista, joka ilmenee aina anturia laukaistaessa. Tämä on vain yksi mahdollinen toteutus, eikä vanhan järjestelmän rakenteeseen tarkemman tiedon puuttuessa oteta tämän enempää kantaa.

3.1 Alkuperäisen anturin kunnostus

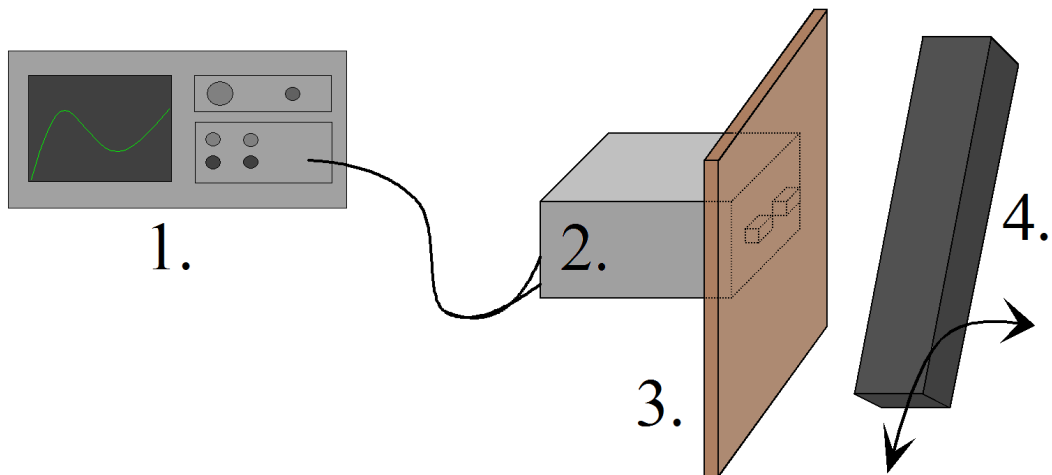
Koska anturia käytetään uudessa järjestelmässä asentotiedon mittaamiseen, poikkeaa käyttö oleellisesti alkuperäisestä toiminnasta. Vanhan käämin tarkoituksena oli tuottaa kohtuullisen suuri jännite, jotta kierrosnopeus voitaisiin mitata luotettavasti. Kun jännite on korkeampi, pienet muutokset kelan induktanssissa, kuten esimerkiksi tuotantotavasta johtuvat vaihtelut, eivät vaikuta merkittävästi mittaustarkkuuteen. Uudelle järjestelmälle sen sijaan riittää että anturi tulisi antaa jännitepulssin aina kun sakarapyörä sen laukaisee. Korkea jännite ei ole tarpeen, mutta jännitepulssien tulisi olla selvästi havaittavissa. Käytännössä tämä tarkoittaa sitä, että käämikierroksien määrää voidaan vähentää.

Uutta käämiä suunniteltaessa todettiin, että laskennallinen analyysi joudutaan jättämään käytännössä kokonaan pois, sillä anturin magneettipiirin materiaalien tunnistaminen ja niiden ominaisuuksien määrittäminen on vaikeaa. Esimerkiksi sydänmateriaalin tarkempaa analyysiä ei ole mahdollista tehdä, sillä se vaatisi näytepalojen ottamista ja koska kyseessä on yksittäiskappale, ei tämä tule kyseeseen.

Vaikka magneettipiirin rautaosien osalta olisi mahdollista käyttää yleistettyjä arvoja, on vanhan kestopagneetin tuottamaa magneettivuontiehyttä vaikea mitata. Koko magneetti-piirin vuontiheyttä yritettiin mitata Hall -anturin avulla anturin sydämien päästä, mutta mittausjärjestelmän tarkkuus ei riittänyt anturin pienen vuontiheyden luotettavaan mittaamiseen. Tämän vuoksi anturin käämiä uusittaessa jouduttiin turvautumaan kokeelliseen lähestymiseen käämin kierrosmäärän määrittämisessä.

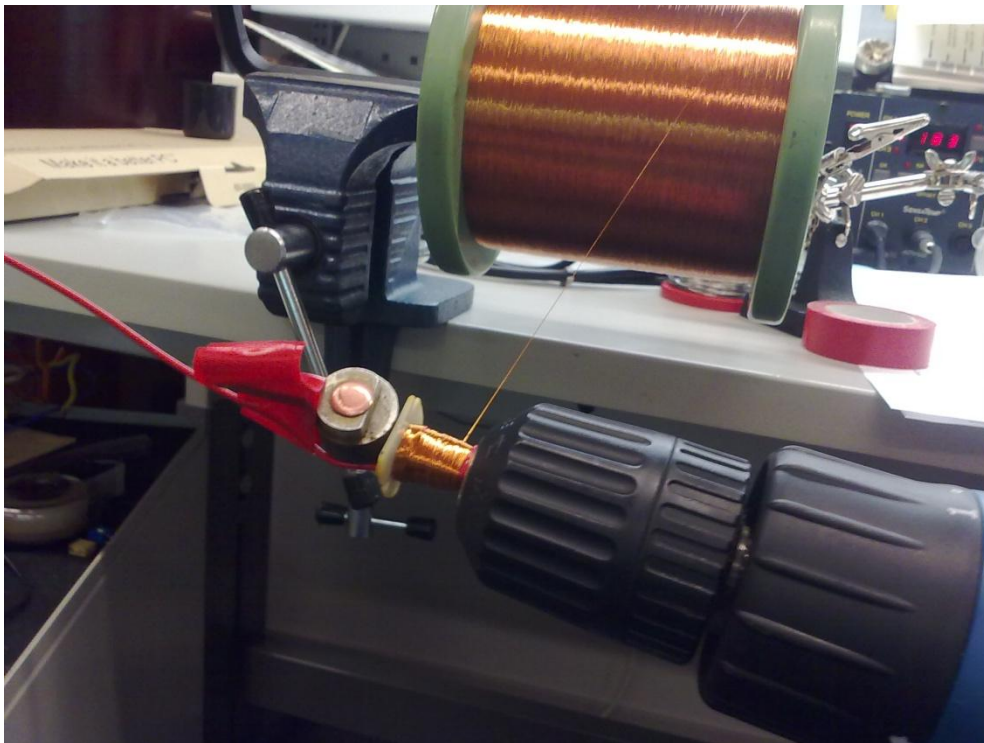
Vanha käämi purettiin ja sen kierrosmäärä laskettiin, jotta saadaan vertailukohta uuden käämin kierrosmäärälle. Kierroksia oli noin 8600.

Uuden käämin käämiminen aloitettiin siten, että anturiin käämittiin ensin 100 kierrosta, jonka jälkeen anturin toimintaa kokeiltiin valurautakappaleen avulla. Anturin eteen asetettiin noin 1 mm:n paksuinen aaltopahvi, ja tämän jälkeen anturin ilmaväliä suljettiin valurauta kappaleen avulla pahvin suojatessa anturia kolhuilta. Mittauksen periaate on esitetty kuvassa 3.5. Kuvassa mittausvälineet on numeroitu siten, että oskilloskooppi on merkitty numerolla 1, anturi numerolla 2, pahvi numerolla 3 ja valurautakappale numerolla 4.



Kuva 3.5 Käämikiertosuunnan vaikutuksen määrittämiseksi tehdyn mittauksen periaate

Tämä mittaustapa on varsin epätarkka, mutta tässä tapauksessa se todettiin riittäväksi. 100 käämikiertoksella saatiin 25 – 80 mV:n jännitepulssi, riippuen anturin ja rautakappaleen välissä olevan suojapahvin paksuudesta. Tällä perusteella kierrosmäärää kasvatettiin 1500 kierrokseen, jolloin päästiin noin 1V:n jännitepulssiin. Kuvasta 3.6 voidaan nähdä kuinka anturi käämittiin akkuporakoneita apuna käyttäen.

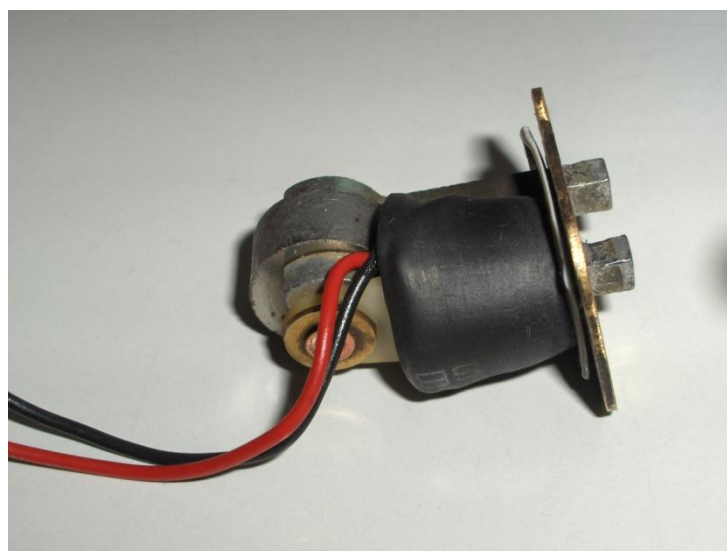


Kuva 3.6 Käämiä kierretään akkuporakoneen avulla.

Lopuksi käämin päälle pujotettiin kuminen kutistesukka, joka kuumentamisen jälkeen sitoo käämin tiukasti kasaan. Kuvassa 3.7 on anturin magneettipiiri vanhalla käämillä ja kuvassa 3.8 uudella käämillä.

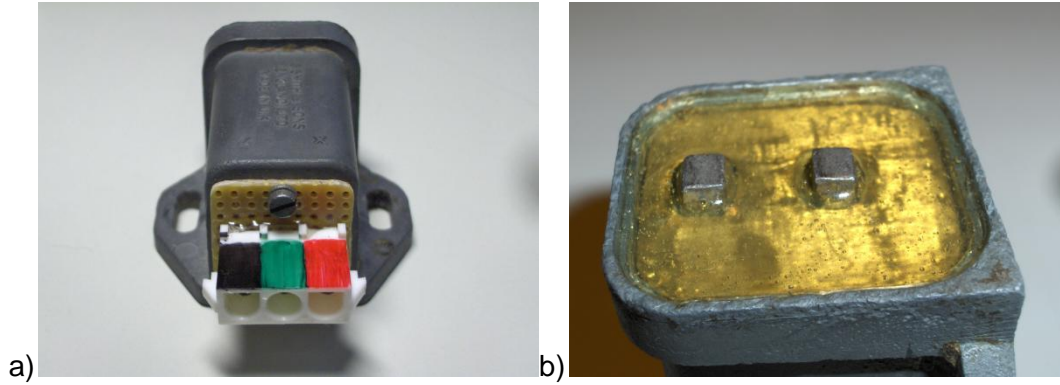


Kuva 3.7 Anturi vanhalla käämillä



Kuva 3.8 Anturi uudella käämillä

Lopuksi anturi valettiin epoksihartsiin ja siihen lisättiin 3-napainen liitin, jonka navoista kaksi liittyy suoraan anturin käämiin ja yksi anturin runkoon. Kunnostettu anturi on esitetty kuvassa 3.9.



Kuva 3.9 Kunnostetun anturin a) liitin ja b) epoksihartsin valettu anturin pää

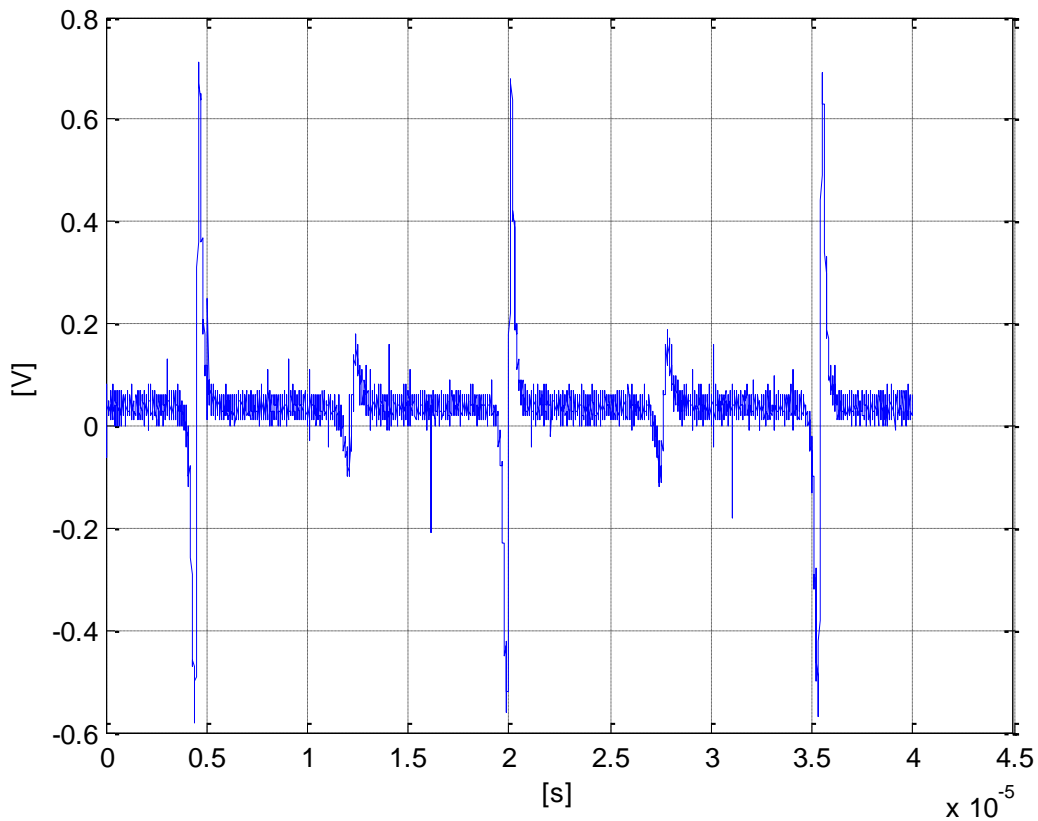
3.2 Anturille suoritettut mittaukset

Työn kohteena olevan dieselmoottorin maksimikierto nopeudeksi on ilmoitettu 2500 rpm. Anturin laukaiseva sakarapyörä kiinnittyy moottorin nokka-akselin päähän, jolloin, nokka-akselin pyöriessä puolet hitaammin kuin itse moottori, tulee sen kierrosnopeudeksi maksimikiirroksilla 1250 rpm. Koska sakarapyörässä on 6 sakaraa, tuottaa anturi tuolloin 7500 pulssia minuutissa, joka vastaa 125 Hz:n taajuutta yhtälön 3.2 mukaan.

$$f_{pulse} = \frac{n_{nokka} \cdot 6 ppr}{60s} = \frac{\frac{n_{moottori}}{2} \cdot 6 ppr}{60s} = \frac{\frac{2500rpm}{2} \cdot 6 ppr}{60s} = 125Hz. \quad (3.2)$$

Mittarin minimikiirroksiksi sen sijaan haluttiin 500 rpm, jota vastaava laukaisutaajuus on 25 Hz. Näin ollen anturin toimintaa tulee tarkastella ensisijaisesti taajuusalueella 25 – 125 Hz.

Anturin kunnostusvaiheessa suoritettiin joitakin mittauksia laukaisemalla anturi valuraudalla, kuten edellä kuvattiin. Myös akkuporakoneeseen kiinnitettyä valurauta- ja rakenneteräskappaletta käytettiin suurempien kierrosnopeuksien mittaamiseen. Porakoneen kierrosnopeus on maksimissaan 900 rpm ja anturin tuottaessa kaksi pulssia kierroksen aikana, pulssinopeus vastaa 600 rpm itse dieselmoottorilla. Tämä on jotakuinkin minimikierto nopeus joka halutaan mitata. Valurautakappaleella tehdyn mittauksen anturikäänin jännite on esitetty kuvassa 3.10.



Kuva 3.10 Porakoneella suoritetussa mittauksessa anturikäämiin indusoitunut jännite, anturia laukaisevana materiaalina valurauta.

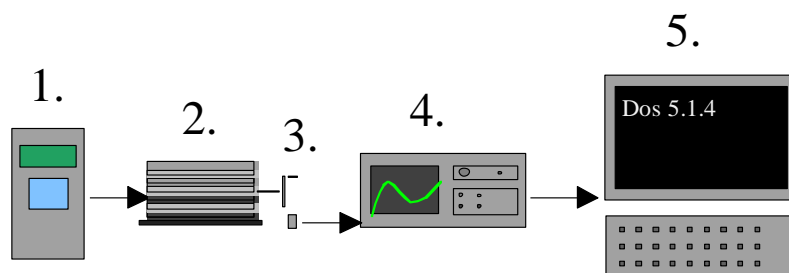
Muuttumaton ilmaväli on anturin toiminnan kannalta tärkeä, jotta anturin tuottamien pulssien jännite pysyy vakiona. Tämä ei toteutunut valurautakappaleen kohdalla sen ollessa hieman epäkesko. Kuvasta voidaan selvästi havaita että joka toisella laukaisulla ilmaväli on suurempi, ja näin ollen indusoitunut jännite on pienempi. Vaikka ilmavälin vaihtelu ei ollut kuin millimetrien luokkaa, on sen vaikutus jännitteeseen varsin suuri.

Taulukko 3.1 Porakoneella suoritetussa mittauksessa saadut keskimääräiset pulssin maksimijännitteet eri materiaaleilla.

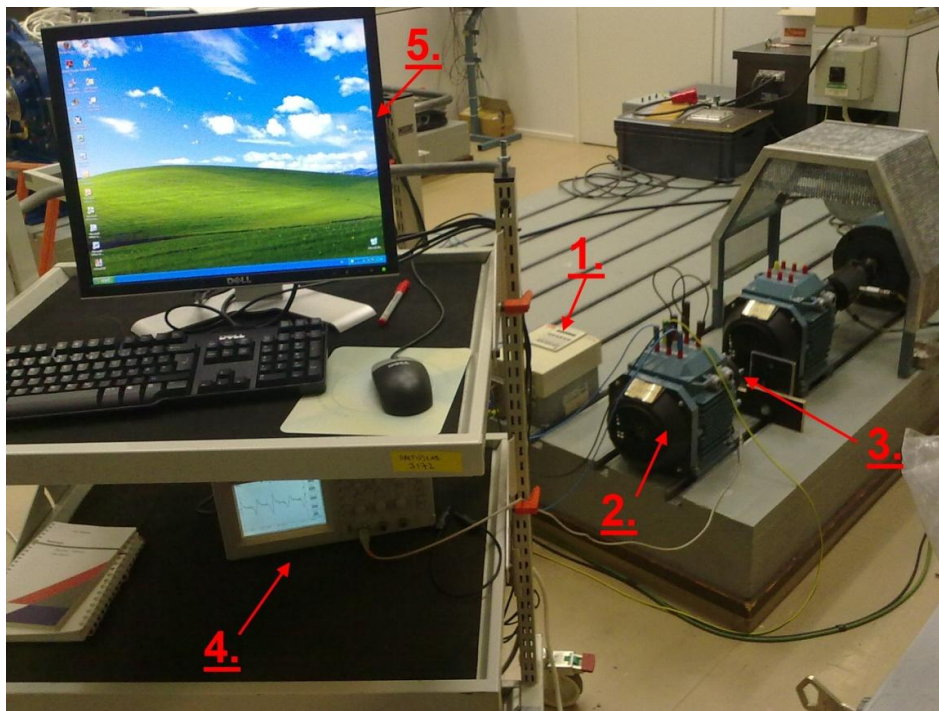
Kierrosnopeus porakone/dieselmoottori [rpm]	Jännite, valurauta [V]	Jännite, rakenneteräs [V]
900 / 600	0,8	0,3

Kun anturi saatiin kasaan, testattiin sitä aina 165 Hz:n pulssitaajuuteen asti. Tämä vastaa noin 3300 rpm:n kierrosnopeutta dieselmoottorilla. Mittaus toteutettiin asentamalla ABB:n

M2AA 112M -moottoriin laippa, johon kiinnitettiin ruuvien avulla neljä anturia laukaisevaa kappaletta. Kappaleina toimivat eri mittauksissa päällekkäin pinotut mutterit, hiiliteräksestä sorvatut holkit ja rautaisesta L-profiilitangosta katkaistut palat. Mutterit ja L-profiili ovat mukana lähinnä kontrolliryhminä, sillä oletettavasti hiiliteräs on magneettisilta ominaisuuksiltaan näistä paras, ja myös lähimpänä valurautaista sakarapyörää. Oikosulkumoottorin ohjaamiseen käytettiin ABB:n ACS 200 -taajuusmuuttajaa ja anturin jännitepulsseja mitattiin Tektronix TDS 210 -oskilloskoopilla. Mittausdata tallennettiin tietokoneelle. Mittausjärjestelyjen vuokaavio on esitetty kuvassa 3.11 ja valokuva mittauksen yleisjärjestelyistä kuvassa 3.12. Taajuudenmuuttaja on molemmissa kuvissa merkitty numerolla 1, oikosulkumoottori numerolla 2, anturi ja laukaiseva kappale numerolla 3, oskilloskooppi numerolla 4 ja tietokone numerolla 5.



Kuva 3.11 Mittausjärjestelyjen vuokaavio.



Kuva 3.12 Mittauksen yleisjärjestelyt.

Mittauksia suoritettiin 10, 20, 30 ja 40 Hz:n syöttötaajuuksilla. Näitä vastaavat pulssitaajuuudet ja dieselmootorin kierrosnopeudet, sekä suurin ja pienin mitattu positiivinen jännitepulsssi, on esitetty taulukossa 3.2.

Taulukko 3.2 Mittaustulokset oikosulkumoottorin avulla suoritetusta anturimittauksesta

Kappale	Syöttötaajuus [Hz]	Pulssitaajuus n. [Hz]	Moottorin kierrosnopeus n. [rpm]	Suurin pulssi [V]	Pienin pulssi [V]
Rautaholkki	10	42	833	1,34	0,78
Mutteri	10	42	833	0,86	0,52
Kulmarauta	10	42	833	0,68	0,3
Rautaholkki	20	77	1538	2,2	0,84
Mutteri	20	77	1538	1,16	0,62
Kulmarauta	20	77	1538	1,14	0,52
Rautaholkki	30	125	2500	2,64	0,96
Mutteri	30	125	2500	1,5	0,98
Kulmarauta	30	125	2500	1,58	0,68
Rautaholkki	40	167	3333	3,04	1,12
Mutteri	40	167	3333	1,8	1
Kulmarauta	40	167	3333	1,88	0,88

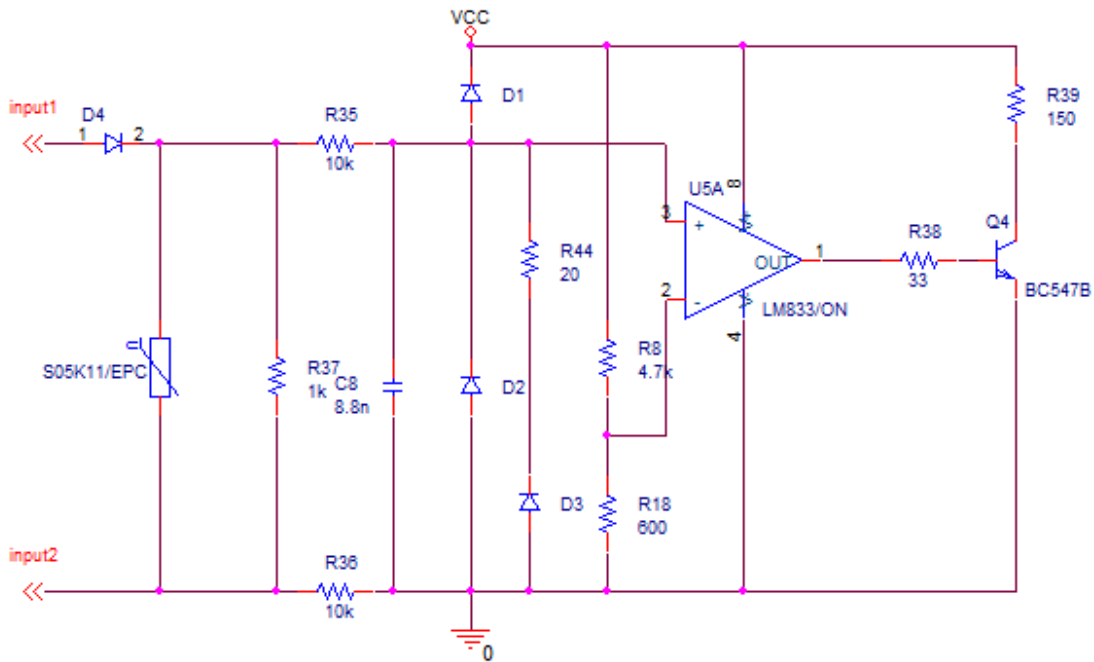
Mittausten perusteella voidaan todeta, että anturin tuottamaan jännitteen suuruuteen vaikuttaa huomattavasti anturin ilmväliä sulkeva materiaali. Tämä on hyvin ymmärrettävää, sillä olettaessa että kaikilla materiaaleilla magneettiin pituus ja poikkipinta-ala pysyy vakiona, on ainoa vaihteleva suure materiaalin permeabiliteetti. Mitä suurempi permeabiliteetti materiaalilla on, sitä vähemmän se tuottaa häviöitä magneettiin ja sitä suurempi vuon vaihtelu saadaan aikaan suljetun ja avoimen magneettiin välille. Näin ollen myös jännite suurenee.

Mittaustulosten perusteella päädyttiin vielä uusimaan anturin käämi, sillä suurillakaan nopeuksilla ei ylitetty 5 V:n rajaa. Lisäksi häiriöiden vaikutus jää korkeammalla jännitteellä pienemmäksi, edellä kuvatuissa mittauksissa taajuudenmuuttajan aiheuttamat häiriöt vaikuttivat myös osaltaan mittaustulokseen. Dieselmoottorin voidaan olettaa olevan ympäristönä vähemmän häiriöinen kuin oikosukumoottori ja sitä ajava taajuudenmuuttaja, mutta varmuudenvuoksi anturin käämiin lisättiin kierroksia niin paljon, kuin sen fyysinen koko salli. Uuden käämin toiminnan varmistamiseksi suoritettiin samanlainen yksinkertainen mittaustulos, kuin anturin käämikierroksien vaikutusta tutkittaessa ennen ensimmäisen käämin käämimistä. Tällöin anturilta saatiin noin 4-5 V:n jännite. Anturisignaalin korkeammat jännitepulssit pienentävät häiriöiden merkitystä signaalin digitalisoinnissa.

3.3 Anturisignaalin käsittely

Jotta analogista anturisignaalia voidaan hyödyntää digitaalisessa järjestelmässä, täytyy se muuntaa digitaaliseen muotoon. Koska kyseessä on asentoanturi, joka ilmaisee vain onko sakarapyörän sakara kohdalla vai ei, riittää 1-bittinen A/D -muunnos. Tämä toteutetaan järjestelmään suunnitellulla tulokytkenällä siten että anturipulssin aikana tulokytkenän lähtö on looginen 0 ja muutoin looginen 1. Tulokytkenän tarkoituksena on myös suojata

järjestelmän mikrokontrolleria mahdollisilta ylijännitteiltä ja suodattaa häiriöistä tulosignaalia. Tulokytken piirikaavio on esitetty kuvassa 3.13.



Kuva 3.13 Tulokytken piirikaavio

Suuremmilta ylijännitteiltä suojaavana komponenttina toimii varistori heti anturitulojen, *input1* ja *input2*, välissä. Tulevia virtoja voidaan rajoittaa 10kΩ:n vastuksilla *R35* ja *R36*, sekä korkeataajuisia häiriöitä suodattaa lisäämällä kondensaattori *C8*, jonka kapasitanssi on 8.8nF. Komponenttien muodostamalla yksinkertaisella RC -alipäästösuoitimella saadaan vähennettyä tulosignaalin korkeataajuisia häiriöitä. Rajataajuus on asetettu 18 kHz:n yhtälön 3.4 mukaan.

$$f = \frac{1}{2\pi RC} = \frac{1}{2\pi \cdot 10k\Omega \cdot 8,8nF} \approx 18kHz. \quad (3.4)$$

Lisäksi operaatiovahvistinkytkennän ei-invertoivan tulon jännite on rajoitettu diodeilla *D1* ja *D2*. Maksimijännite ei-invertoivassa tulossa on diodin *D1* bias-jännitteen V_d ja V_{CC} käyttöjännitteen summa yhtälön 3.5 mukaan.

$$V_{\max} = V_{CC} + V_d = 5V + 0,7V = 5,7V. \quad (3.5)$$

Minimijännite ei-invertoivassa tulossa on vastaavasti diodin *D2* bias-jännitteen V_d suuruinen, mutta negatiivinen kuten yhtälössä 3.6.

$$V_{\min} = -V_d = -0,7V. \quad (3.6)$$

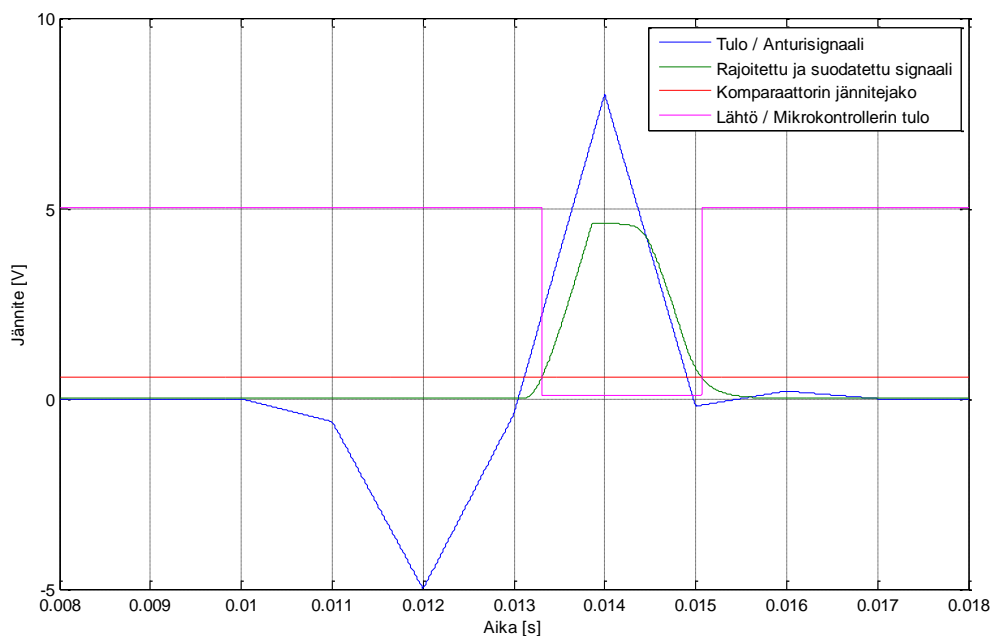
Näiden tavallisten diodien lisäksi piirissä on vielä 4,7V:n zener -diodi $D3$ (kuvassa simulatiomalli väärällä piirikaaviosymbolilla), jonka tarkoituksena on pitää jännite käyttöjännitetason alapuolella. Operaatiovahvistimen oikean toiminnan kannalta on tärkeää että jännite ei pääse korkeammaksi kuin käyttöjännite. Zener -diodin virtaa rajoittava vastus $R44$, on mitoitettu siten, että sen virta on 48 mA korkeimmalla mahdollisella jännitteellä joka kato-dille voi tulla diodin $D1$ johtaessa. Koska operaatiovahvistimen tulon virta on pieni, voidaan se olettaa nolaksi zener -diodin etuvastusta mitoitettaessa ja käyttää ainoastaan diodille ilmoitettua optimaalisen toimintapisteen virtaa. Vastuksen arvo voidaan laskea Ohmin lain mukaan yhtälöstä 3.7.

$$R = \frac{V_{\max} - V_{\text{zener}}}{I_{\text{zener}}} = \frac{5,7V - 4,7V}{48mA} = 20,8\Omega \approx 20\Omega. \quad (3.7)$$

Yhtälössä R on virranrajoitusvastuksen arvo, V_{\max} on operaatiovahvistimen ei-invertoivan tulon maksimijännite, V_{zener} on diodin zener -jännite ja I_{zener} on valmistajan diodille ilmoitama mitoitusvirta.

Operaatiovahvistin on kytketty tulopiirissä komparaattoriksi, ja invertoivaan tuloon kytketty jännitteenjako määrittää jännitetason jonka ylittäessään anturisignaali muuttaa komparaattorin lähdön tilaksi 1. Kun anturisignaali on pienempi kuin jännitejaon määrittämä 560 mV, on komparaattorin lähtö taas tilassa 0.

BJT –transistoria käytetään ohjaamaan kytkennän lähtöä, jotta saadaan mahdollisimman nopea muutos tilasta toiseen. Lepotilassa ylösvetovastus $R39$ nostaa lähdön tilaan 1, ja se vedetään alas transistorilla anturipulssin saapuessa. OrCAD -piirisimulaattorilla simuloidun A/D-piirin toiminta on esitetty kuvassa 3.14.



Kuva 3.14 A/D-piirin toiminta OrCAD:ssa simuloituna.

Simulaation mukaan piiri toimii moitteetta, mutta kytkentää testatessa ilmeni että mikrokontrollerintulo värähtelee huomattavan paljon tilojen vaihtuessa, ja voi aiheuttaa useita keskeytyksiä yhdellä anturipulssilla. Tämän vuoksi lisättiin kondensaattori kytkennän lähdön ja maan välille vakauttamaan tilojen muutoshetkiä.

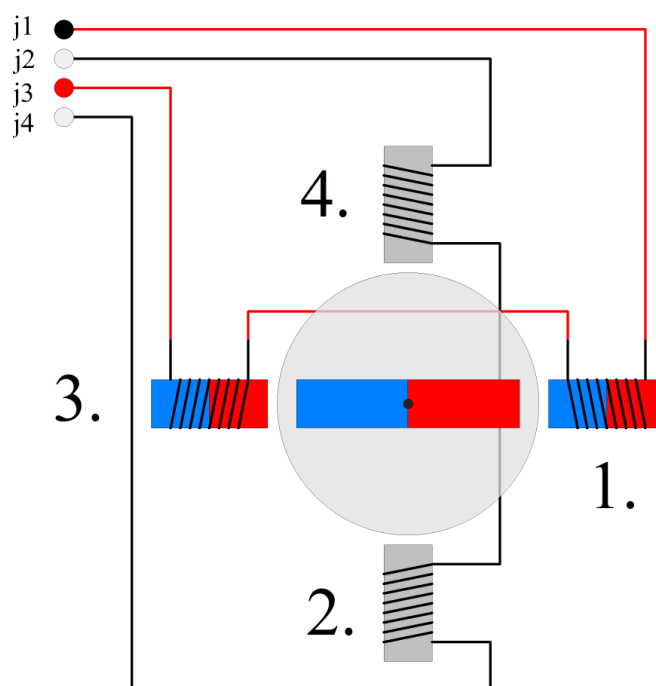
4. NÄYTTÖ

Kuten työn alussa mainittiin, loppukäyttäjän toiveena oli analoginen, viisarillinen, näyttölaitte työn aiheena olevaan kierrosnopeusmittausjärjestelmään. Nykyaikaisissa ajoneuvosovelluksissa analogiset viisarinäytöt on usein toteutettu askelmoottorin avulla. Askelmoottorin etuna näissä sovelluksissa on suuri pitovääntömomenti, jonka ansiosta mittari ei värähtele vaikka se kokisi kuoppaisen tien, tai kilparadan mutkien, aiheuttamia kiihtyvyyksiä. Työssä päädyttiin käyttämään askelmoottorinäyttöä juuri tämän ominaisuuden vuoksi, sillä venekäytössä näytön tulisi pysyä vakaana kovassakin aallokossa.

4.1 Askelmoottori

Askelmoottori on harjaton sähkömoottorityyppi, jota voidaan ajaa askel kerrallaan ilman takaisinkytkentää. Kun askelmoottorin maksimivääntömomenttia ei ylitetä, asentotiedoksi riittää otettujen askelten määrä. Askelmoottorin asentovirhe ei ole kumuloituvaa. (Condit & Jones 2008)

Yksinkertaistettu kestomagneettiaskelmoottorin käämitys koostuu kahdesta käämistä, joiden jännitteen napaisuutta kääntämällä oikealla sekvenssillä roottori saadaan pyörimään siirtyvien magneettikenttien mukaan. Yksinkertaistettu kuva bipolaarisen kestomagneettiaskelmoottorin käämityksistä on esitetty kuvassa 4.1. (Acarnley 2007, 12) (Yeadon 2001, 5.47)



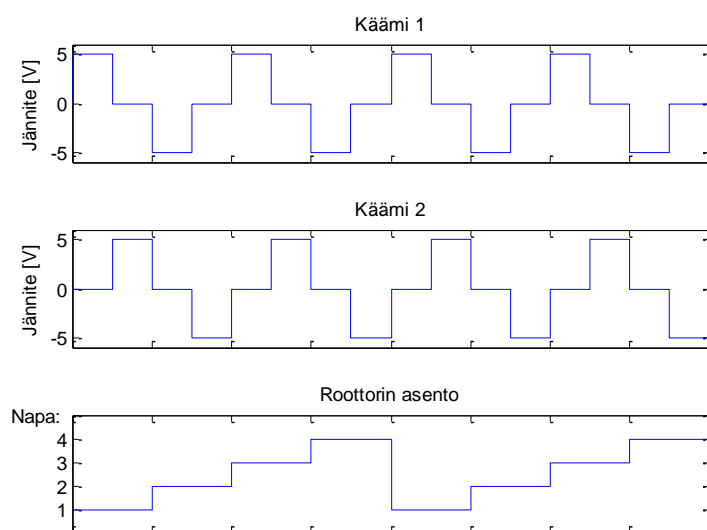
Kuva 4.1 Bi-polaarisen askelmoottorin periaatekuva

Vaikka kuvassa näkyy ainoastaan neljä napaa staattorissa, voi niitä olla reaalisessa toteutuksessa huomattavasti enemmän suuremman vääntömomentin ja pienemmän askelkulman saavuttamiseksi. Lisäksi pienemmän askelkulman saavuttamiseksi, napojen päät, sekä roottorin pinta voivat olla hammastettuja. Molemmat tekniikat ovat käytössä lähes kaikissa moderneissa askelmootoreissa. (Kenjo 1984, 4)

Kuvan esittämän peruserätyksen kytkettyä bipolaarista askelmoottoria voidaan ajaa pääasiallisesti kolmella eri sekvenssillä. Nämä ovat aaltoaskellus (*eng. wave drive* tai *one-phase-on*), koko- ja puoliaskellus, sekä mikroaskellus.

Aaltoaskelluksessa ainoastaan yhteen moottorin käämiin kerrallaan kytketään jännite ja muut pidetään maadoitettuna. Tämä ajotapa on kaikkein yksinkertaisin toteuttaa, eikä se aseta suuria vaatimuksia sitä ajavalle piirille. Moottorin täyttä vääntömomenttia ei tällä ajotavalla saada, mutta varsinkin harrastelijakäytössä nopea ja yksinkertainen implementointi on etuna. Aaltoaskelluksen käämien jännitteet on esitetty kuvassa 4.2 ja looginen toiminta

taulukossa 4.1. Roottorin asentoa kuvaavat napojen numerot vastaavat kuvan 4.1 numerointia.



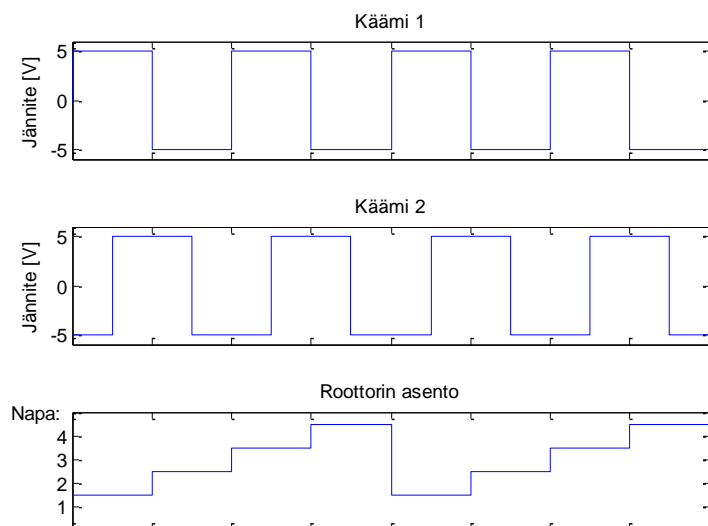
Kuva 4.2 Aaltoaskelluksen (*eng. wave drive*) moottorikäämien jännitteet ja roottorin asento.

Taulukko 4.1 Aaltoaskelluksen looginen taulukoesitys

Moottorin johdin	Askel 1	Askel 2	Askel 3	Askel 4
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

Kuten roottorin asennon kuvaajasta voidaan nähdä, kääntyy roottori neljänneskierroksen jokaisella askeleella. Jos käytettävän askelmoottorin askelkulma on pieni, voidaan aaltoaskelluksellakin päästä kohtuullisen tasaiseen ajoon, mutta kuten edellä mainittiin, ei vääntömomenttia saada kuitenkaan yhtä paljon kuin kokoaskelluksella. Aaltoaskelluksen maksimi vääntömomentti on noin puolet moottorin maksimivääntömomentista ja se heikenee askeltaajuuden kasvaessa.

Kokoaskelluksessa molempien käämien lävitse kulkee jatkuvasti virta, jolloin saadaan enemmän vääntömomenttia ja tällä askelluksella saavutetaan moottorin maksimivääntömomentti. Kokoaskelluksen käämien jännitteet on esitetty kuvassa 4.3 ja looginen esitys toiminnasta taulukossa 4.2.



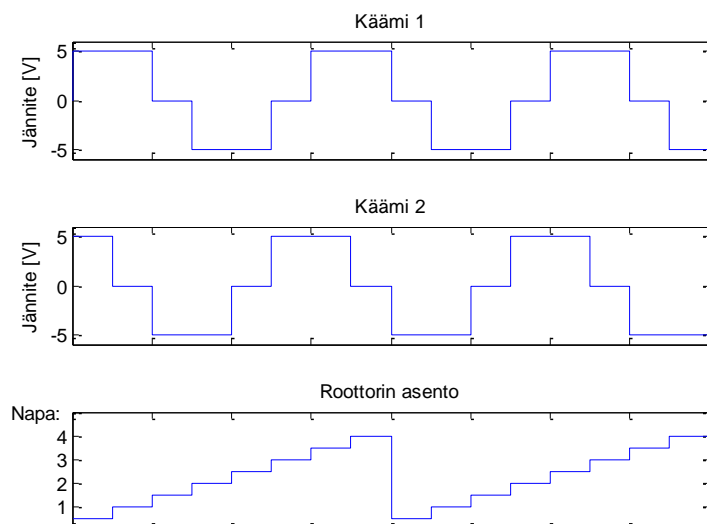
Kuva 4.3 Kokoaskelluksen käämien jännitteet ja roottorin asento.

Taulukko 4.2 Kokoaskelluksen looginen taulukoesitys

Moottorin johdin	Askel 1	Askel 2	Askel 3	Askel 4
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

Roottorin asennon kuvaajan perusteella voidaan todeta kokoaskelluksen tarjoavan saman resoluution roottorin asennon hallintaan kuin aaltoaskellus. Erona aaltoaskellukseen on, että roottori siirtyy aina kahden navan välille, eikä suoraan niiden kohdalle.

Puoliaskelluksella saadaan suurempi resoluutio, mutta vääntömomentti on joka toisella puoliaskeleella vain puolet askelmoottorin nimellisestä vääntömomentista. Tämä täytyy ottaa huomioon moottoria mitoitettaessa. Puoliaskelluksen käämien jännitteet on esitetty kuvassa 4.4 ja toiminnan looginen esitys taulukossa 4.3.



Kuva 4.4 Puoliaskelluksen käämien jännitteet ja roottorin asento.

Taulukko 4.3 Puoliaskelluksen looginen taulukkoesitys

Moottorin johdin	Askel 1	Askel 2	Askel 3	Askel 4	Askel 5	Askel 6	Askel 7	Askel 8
1	1	1	1	0	0	0	0	0
2	0	0	1	1	1	0	0	0
3	0	0	0	0	1	1	1	0
4	1	0	0	0	0	0	1	1

Kuten roottorin asennon kuvaajasta nähdään, on puoliaskellus kolmesta edellä esitetystä askellustavasta kiistatta sulavin ja tarkin tapa ajaa bi-polaarista askelmoottoria. Moottorista saatava vääntömomentti ei vastaa kokoaskellusta, mutta tarkkuutta vaativissa sovelluksissa puoliaskellus on varteenotettava vaihtoehto.

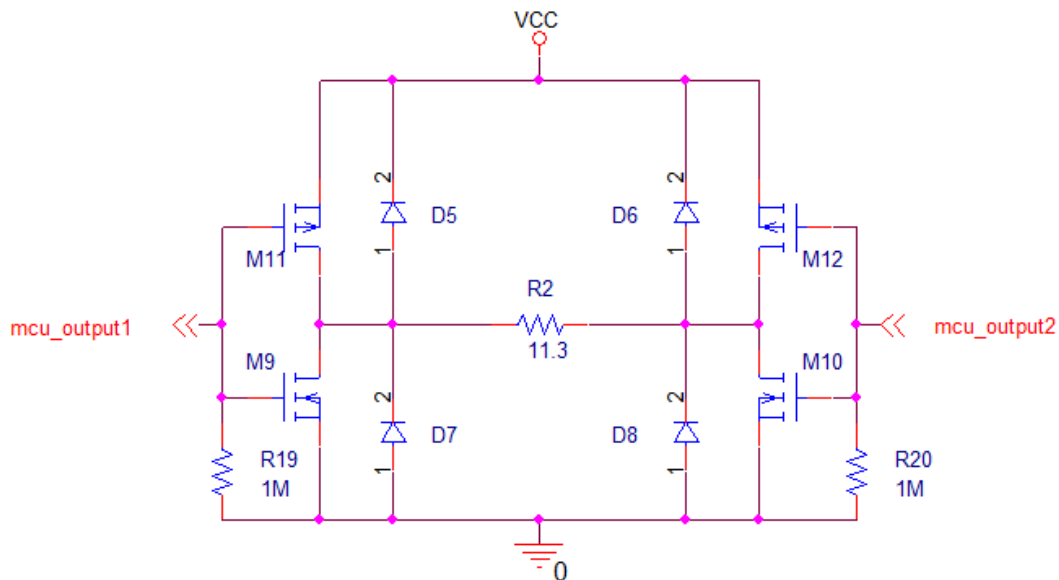
Lisätarkkuutta ja pehmeyttä askelmoottorin ohjaukseen voidaan lisätä tuottamalla puolitai kokoaskelluksen ohjaus PWM:n avulla, jolloin askelmoottoria voidaan ohjata lähes portaattomasti. Tätä ohjausmetodia kutsutaan mikroaskellukseksi, tai sini-kosini-askellukseksi, eikä siihen perehdytä tässä työssä, sillä puoliaskelluksen tarkkuus riittää sovellukseen varsin mainiosti.

Työssä käytettävä askelmoottori on Mitsumi M42SP-4 bipolaarinen askelmoottori 3,75°:n askeleella ja 44,1 mNm:n pitovääntömomentilla. Koska näyttösovelluksessa askelmoottoria kuormittaa ainoastaan osoittimena toimiva mittarin neula, ei sen tuottama vääntömomentti ylitä puoliaskelluksellakaan, ja näin ollen otettujen askelten määrä riittää näytön asentotiedoksi.

4.2 H-silta

Koska työssä päätettiin käyttää kahta H -siltaa ajamaan askelmoottorin käämejä, on ajotapa vapaasti ohjelmoijan valittavissa, sillä H -sillan avulla toteuttaa voidaan kolme ensimmäistä edellä mainituista ajotavoista.

Työssä käytettävän askelmoottorin ohjaukseen päätettiin käyttää kahta H -siltaa, koska niiden avulla moottorin askellustapa voidaan määrittää ohjelmallisesti, jolloin vaihtoehtoina ovat aalto-, koko- ja puoliaskellus. Kun siltojen kytkiminä käytetään logiikkatason MOS-FET:jä, päästään yksinkertaisimpaan toteutukseen, koska kytkimiä voidaan ajaa suoraan mikrokontrollerin lähdöillä. Työtä varten suunnitellun H -sillan piirikaavio on esitelty kuvassa 4.5.



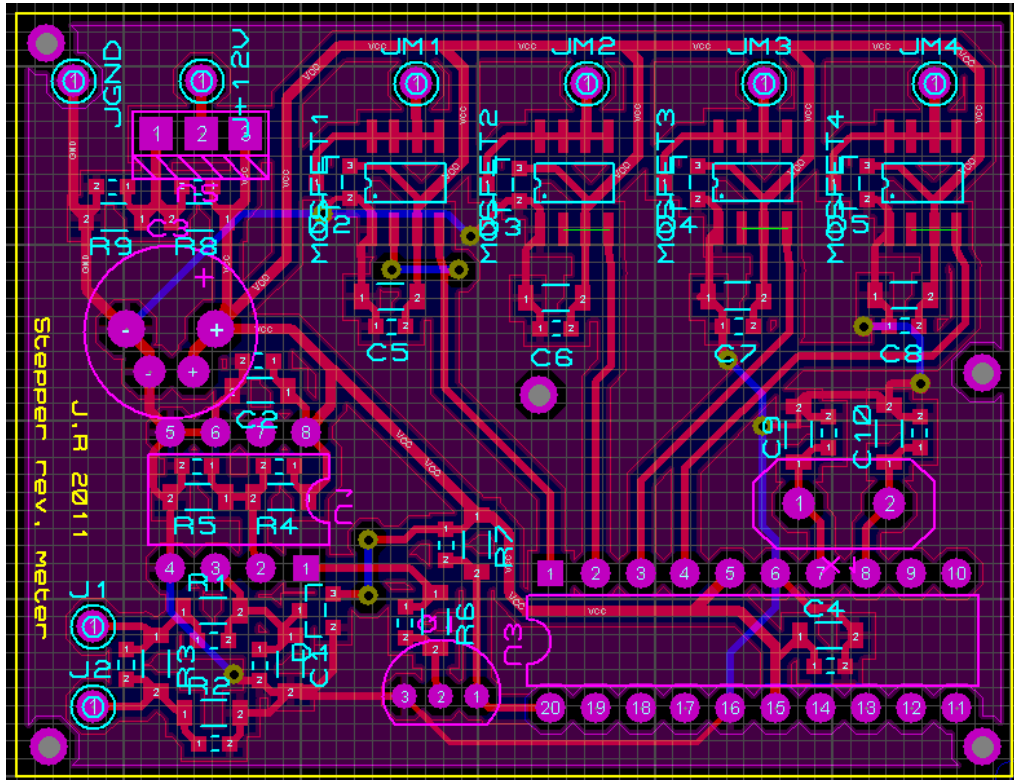
Kuva 4.5 Askelmoottorin ohjaukseen käytetyn H-sillan piirikaavio.

Vastus R2 kuvaa H -sillan kuormana olevaa moottorikäämitystä. Järjestelmän maksimitaajuuksien ja -jännitteiden ollessa pieniä (<200Hz ja 5V), ei käämin tarkempaa mallinnusta nähty tarpeelliseksi.

5. PIIRILEVY

Järjestelmän piirilevyn suunnitteluun käytettiin Proteus 7 -piirisuunnitteluohjelmistoa. Tällä ohjelmalla piirretään ensin piirikaavio ja sen perusteella ohjelma luo tietokannan komponenttien kytkennöistä. Tässä vaiheessa järjestelmälle suunniteltiin jännitteensyöttösimulaatioista saatujen virta-arvioiden perusteella. Mikrokontrollerille lisättiin myös ulkoinen kellokide varmuuden vuoksi, mikäli sisäinen kellon stabiilius ei riittäisi. Tämän jälkeen

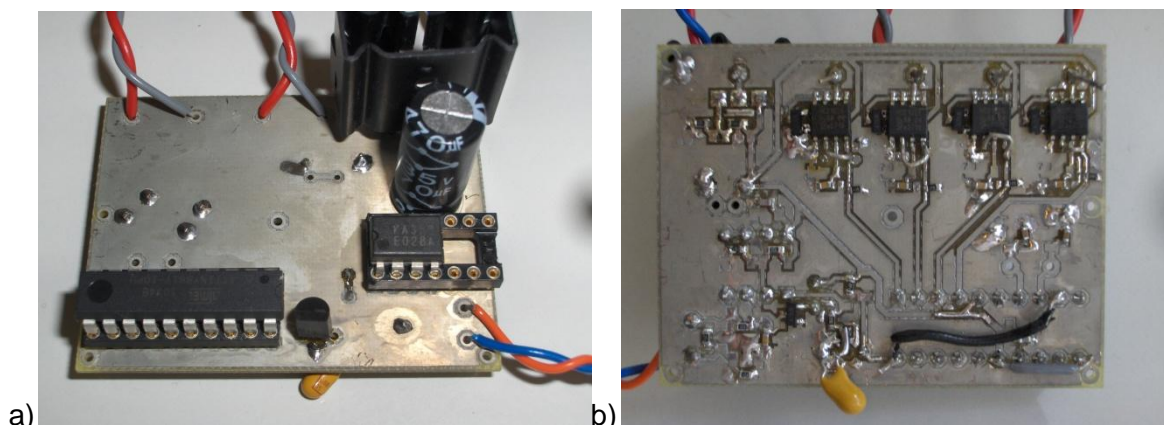
komponentit sijoitettiin levylle ja piirrettiin tarvittavat vedot. Proteuksesta löytyy myös automaattinen reititys ominaisuus, mutta vedot päätettiin piirtää käsin virheiden välttämiseksi. Valmis piirilevy layout on esitetty kuvassa 5.1.



Kuva 5.1 Piirilevyn layout korjaamattomana.

Lappeenrannan teknillisen yliopiston Elektroniikan suunnittelukeskus (ESK) jyrssi kaksipuolisen levyn piirrosten valmistuttua. Läpivientejä ei levyyn saatu tehtyä ESK:lla läpivientiholkkien huonon varastotilanteen vuoksi, joten jouduttiin turvautumaan levyn läpi juotettuihin johtimiin läpivienneissä.

Levystä huomattiin joitakin suunnitteluvirheitä heti kun se saapui ESK:lta, mutta kaikki näistä virheistä pystyttiin korjaamaan prototyypistä ilman ongelmia. Prototyypin piirilevy korjauksineen on esitetty kuvissa 5.2 a) ja b).



Kuva 5.2 Piirilevy kalustettuna ja korjattuna, a) päältä ja b) alta.

Kuten kuvasta 5.2 a) ja b) voidaan nähdä, vaati prototyyppi paljon korjauksia, ja sen visuaalinen ilme ei ole vielä kovin houkutteleva. Yhteenvedossa on pohdittu mahdollisen kehitysversion hyödyllisyyttä myös järjestelmän ulkoisen ilmeen kannalta.

6. DATAN PROSESSOINTI

Anturin mittaama asentotieto on sinällään varsin hyödytön käyttäjälle ja siitä pitäisi tuottaa hyödyllisemmässä muodossa olevaa informaatiota. Uudessa digitaalisessa järjestelmässä tämä datan prosessointi toteutetaan mikrokontrollerilla ja tässä luvussa esitellään työhön liittyvän kontrollerin suorittaman ohjelman rakenne ja tarvittavat oheiskytkennät.

Kontrolleritarjonnan kirjavuuden vuoksi mahdollisia liityntärajapintoja anturin ja kontrollerin välillä on monia. Riippuen anturin itsensä ominaisuuksista, voidaan liityntä tehdä käyttäen ADC -kanavaa tai esimerkiksi ulkoiseen keskeytykseen kykenevää tuloa. Ohjelmoinnin yksinkertaistamiseksi päädyttiin suunnittelemaan tulopiiri, joka kytkeytyy tuloon, joka voi tuottaa ulkoisen keskeytyksen. Tulokytkeä pitää tulon jännitteen loogisessa tilassa 1, kunnes se saa jännitepulssin sensorilta ja tulo maadoitetaan loogiseen 0:an. Kun jännitepulssi on tasoittunut, siirtyy tulo taas tilaan 1.

Alun perin tarkoituksena oli käyttää Texas Instruments:n MSP430G2452 -kontrolleria, mutta toimituksen viivästyessä päädyttiin käyttämään Atmel:n Attiny861 -kontrolleria, joka on ominaisuuksiltaan lähestulkoon identtinen TI:n kontrollerin kanssa. Kontrollerin vaihto aiheutti kuitenkin muutoksia piirilevyyn kontrollerin jalkojen järjestyksen muuttuessa. Myös ohjelmakoodi oli jo kirjoitettu MSP430 -kontrollerille ja vaihdoksen vuoksi se jouduttiin vielä kääntämään Attiny861:lle sopivaksi.

6.1 Ohjelmakoodi

Mikrokontrollerin suorittaman ohjelman tarkoituksena on laskea anturilta saatavat pulssit tunnetussa ajan jaksossa, sen perusteella laskea dieselmoottorin kierrosnopeus ja ajaa mittarin neula askelmoottorin avulla oikeaan kohtaan mittaritaululla. Mikrokontrollerin suorittaman ohjelman vuokaavio on esitetty kuvassa liitteessä I ja ohjelmakoodi kokonaisuudessaan liitteessä II.

Pulssien laskeminen toteutetaan siten, että anturi on kytketty mikrokontrollerin tuloon, joka voi tuottaa ulkoisen keskeytyksen. Tällaisia tuloja Attiny861:ssä ovat tulot INTO ja INT1.(Atmel 2010, 51) Anturipulssin saapuessa ulkoinen keskeytys asettaa puskurimuuttujan *buffer* arvoksi 1. Puskurin arvo tarkastetaan 3 ms välein, ja sitä verrataan edelliseen puskurin arvoon. Mikäli arvo poikkeaa edellisestä ja on 1, korotetaan laskurina toimivan muuttujan arvoa.

Näytteenottoteoreeman (*Nyquist -teoreema*) mukaan näytteistystaajuuden tulee olla vähintään kaksikertainen näytteistettävän signaalin taajuuteen nähden, jotta laskostumista ei tapahdu.(Proakis & Manolakis 1997, 271) Puskurin tarkistus on ajoitettu käyttämällä 8-bittistä Timer/Counter0 -ajastinta ja edellä kuvatulla ajastimen kalibroinnilla voidaan havaita 167 Hz:n pulssitaajuuksia ilman laskostumista.

Kierrosnopeuden laskeminen tasaisin väliajoin on toteutettu käyttämällä kontrollerin sisäistä 10-bittistä ajastinta Timer/Counter1:ä. Ajastin on alustettu siten, että se vuotaa yli 0,2 s päästä sen käynnistämisestä ja aiheuttaa keskeytyksen jonka aikana lasketaan kierrosnopeus. Pulssien laskemisen ja näyttöä käyttävän askelmoottorin ajamista kontrolloidaan muuttujan *drive_or_count* avulla. Muuttujan arvon ollessa 1, kytketään ajastin keskeytykset pois päältä, kunnes näyttö on ajettu oikeaan arvoonsa. Tämän jälkeen ajastimet alustetaan ja keskeytykset sallitaan. Ajastin keskeytyksen ilmetessä muutetaan *drive_or_count* -muuttujan arvoksi 0, alustetaan Timer/Counter1 ja seuraavan 0,2 sekunnin ajalta lasketaan saapuneiden anturipulssien lukumäärä.

Koska kierrosnopeuden laskenta ja askelmoottorin ajo tapahtuu edellä kuvatuissa sykleissä, voi nopeissa kiihdytyksissä ja järjestelmän käynnistyksen yhteydessä ilmetä jättämää ja katkoja mittarin neulan liikkeessä. Kierrosnopeuden tasoittuessa näyttämä vakiintuu, eikä mittarin askelmainen eteneminen häiritse käyttöä.

Kontrollerissa käytetään sen omaa sisäistä 8 MHz:n kelloa skaalattuna 1MHz:n, joten 10- ja 8-bittiset ajastimet tarvitsevat esiskaalauksen, jotta halutut viiveet voidaan toteuttaa. Esiskaalauksen jakajat molemmille ajastimille on esitetty taulukossa 6.1.

Taulukko 6.1 Ajastimien esiskaalauksen mahdolliset jakajat

Timer/Counter0	Timer/Counter1
1	1
8	2
64	4
256	8
1024	16
	32
	64
	128
	256
	512
	1024
	2048
	4096
	8192
	16384

Ajastimen Timer/Counter0 esiskaalauksetoimeksi valittiin 256 ja Timer/Counter1:lle 2048. Esiskaalatun järjestelmäkellon jaksonaika saadaan yhtälön 6.1 avulla, jossa T on jaksonaika, f_{clk} on järjestelmän kelloaajuus ja k on esiskaalauksen jakaja. (Ahola 2011)

$$T = \frac{1}{\left(\frac{f_{clk}}{k}\right)} = \frac{1}{\left(\frac{1MHz}{2048}\right)} = 2ms . \quad (6.1)$$

Ajastimen laskentasekvenssin pituus saadaan yhtälön 6.2 avulla, jossa t_{sekv} on kokonaisen laskentasekvenssin aika, n laskurin bittimäärä ja T esiskaalatun kellon jaksonaika. (Ahola 2011)

$$t_{sekv} = (2^n - 1) \cdot T = (2^{10} - 1) \cdot 2ms \approx 2,1s . \quad (6.2)$$

Ajastimen Timer/Counter1 lähtöarvoksi asetetaan 0,2 sekuntia vastaava arvo, joka saadaan yhtälön 6.3 avulla. Yhtälössä m on ajastimen alkuarvo, t_{des} on haluttu keskeytysaika ja t_{sekv} kokonaisen laskentasekvenssin aika.

$$m = (2^n - 1) - \frac{t_{des}}{T} = (2^{10} - 1) - \frac{0,2s}{2ms} \approx 925 . \quad (6.3)$$

Timer/Counter0:n lähtöarvo on laskettu vastaavalla tavalla. Ajastimien kalibrointiin tarvittavat tiedot on koottu taulukkoon 6.2.

Taulukko 6.2 Ajastimien asetustiedot

Ajastin	Timer/Counter0	Timer/Counter1
Bittisyys	8	10
Esiskaalaus	256	2048
Lähtöarvo	243 = 0xF3	925 = 0x09D
Ylivuotoaika (lähtöarvosta)	3 ms	200ms

Kummankin ajastimen ylivuotoaika tarkastettiin muuttamalla koodia siten, että aina keskeytyksen ilmetessä kontrollerin lähdön PB5 tila muuttuu. Tämän muutoksen taajuus mitattiin oskilloskoopin avulla. Ajastimien ylivuotojen väli on sama kuin tilamuutoksen jaksonaika. Ajastimen Timer/Counter0 lähtöarvo hienosäädettiin tämän mittauksen perusteella arvoon 242 (0xF2). Timer/Counter1:n ylivuoto tapahtui juuri 200ms:n välein, kuten se suunniteltiin.

Näytön askelmoottorin ajosekvenssit noudattavat taulukossa 4.3 esitettyä puoliaskelluksen sekvenssiä. Koodiin koottiin kaksi funktiota, *forward()* ja *reverse()*, jotka toteuttavat puoliaskellusta eteen- ja taaksepäin. Viimeksi otettu askel tallennetaan muuttujaan *flop*, jonka perusteella seuraava askel otetaan. Askelmoottorin asento-ohjeeksi riittää skaalattu laskurin arvo. Skaalaus tapahtuu kertomalla laskurin arvo neljällä. Tällä saadaan pulssi-taajuuskaistalla 0...130 Hz noin 180°:n liike mittarin neulalle.

7. YHTEENVETO JA JOHTOPÄÄTÖKSET

Vanha kierrosnopeusanturi kunnostettiin uusimalla sen käämitys ja mittaustulosten perusteella päädyttiin vielä lisäämään siihen kierroksia, jotta voidaan varmistaa tarpeeksi korkea signaalitaso mittausjärjestelmälle. Anturidatan prosessointiin valittiin sopiva TI MSP430 -sarjan mikrokontrolleri, mutta toimitusvaikeuksien vuoksi jouduttiin vaihtamaan Atmel:n vastaavaan. Järjestelmän tulon suojaus- ja suodatuskytkentä suunniteltiin mikrokontrollerin asettamilla vaatimuksilla. Askelmoottorin ohjausta varten suunniteltiin sopivilla logiikkatason MOFET:illa toteutetut kaksi H -siltaa.

Työn perusteella voidaan todeta että vanha kierrosnopeusmittausjärjestelmä voidaan kunnostaa monella eri tavalla. Yhtenä varteenotettavana vaihtoehtona on mikrokontrolle-

riohjattu askelmoottori, jonka suuren pitovääntömomentit ansiosta näyttö on vakaa epätaisisessakin käytössä. Mikäli halutaan katkoton mittarin osoittimen liike, vaaditaan korkeampaa kellotaajuutta, kuin työn toteutuksessa käytetty 1 MHz. Suurempaa kellotaajuutta käytettäessä voitaisiin luopua sekventiaalisesta näytön ajosta, ja näin saataisiin jatkuva liike.

Mikrokontrollerin vaihdoksesta, suunnittelijan kokemattomuudesta ja kiireisestä aikataulusta johtuen prototyypissä ilmeni varsin monenlaisia vikoja, komponenttien epäideaalisuudesta johtuvia ongelmia ja jopa suoranaisia suunnitteluvirheitä, mutta nämä voitiin korjata jälkikäteen ja mikäli toinen kehitysversio tuotetaan, on niiden korjaaminen kytkennän ja piirilevyn suunnitelmasta varsin helppoa.

Prototyypin perusteella voidaan kuitenkin todeta konseptin toimiva, jatkokehityksellä päästäisiin varmasti parempaan lopputulokseen. Mahdollisena jatkokehityksenä järjestelmälle voisi olla askelmoottorin käämien virranrajoitus mittarin ollessa paikallaan, komponenttien sijoittelun parantaminen piirilevyllä sekä koteloinnin toteuttaminen.

LÄHTEET

Acarney, P. 2007. Stepping Motors, a guide to theory and practice. Fourth edition. London: The Institution of Engineering and Technology, 159 s. ISBN 978-0-85296-417-0

Ahola, J. 2011. Mikroprosessorit A, Opintojakson luentokalvot, [verkkodokumentti]. [viitattu 22.4.2011]. Saatavissa: <https://noppa.lut.fi/noppa/opintojakso/bl40a1810/luennot>

Atmel ATtiny261/461/861 Datasheet. [www-tuotedokumentti]. Yrityksen www-sivut: Atmel 2010. [viitattu 22.4.2011].

Saatavissa: http://www.atmel.com/dyn/resources/prod_documents/doc2588.pdf

Condit, R. & Jones, D. W. 2008. Stepping Motors Fundamentals, [verkkodokumentti]. Microchip Technology Inc., University of Iowa. [viitattu 7.5.2011].

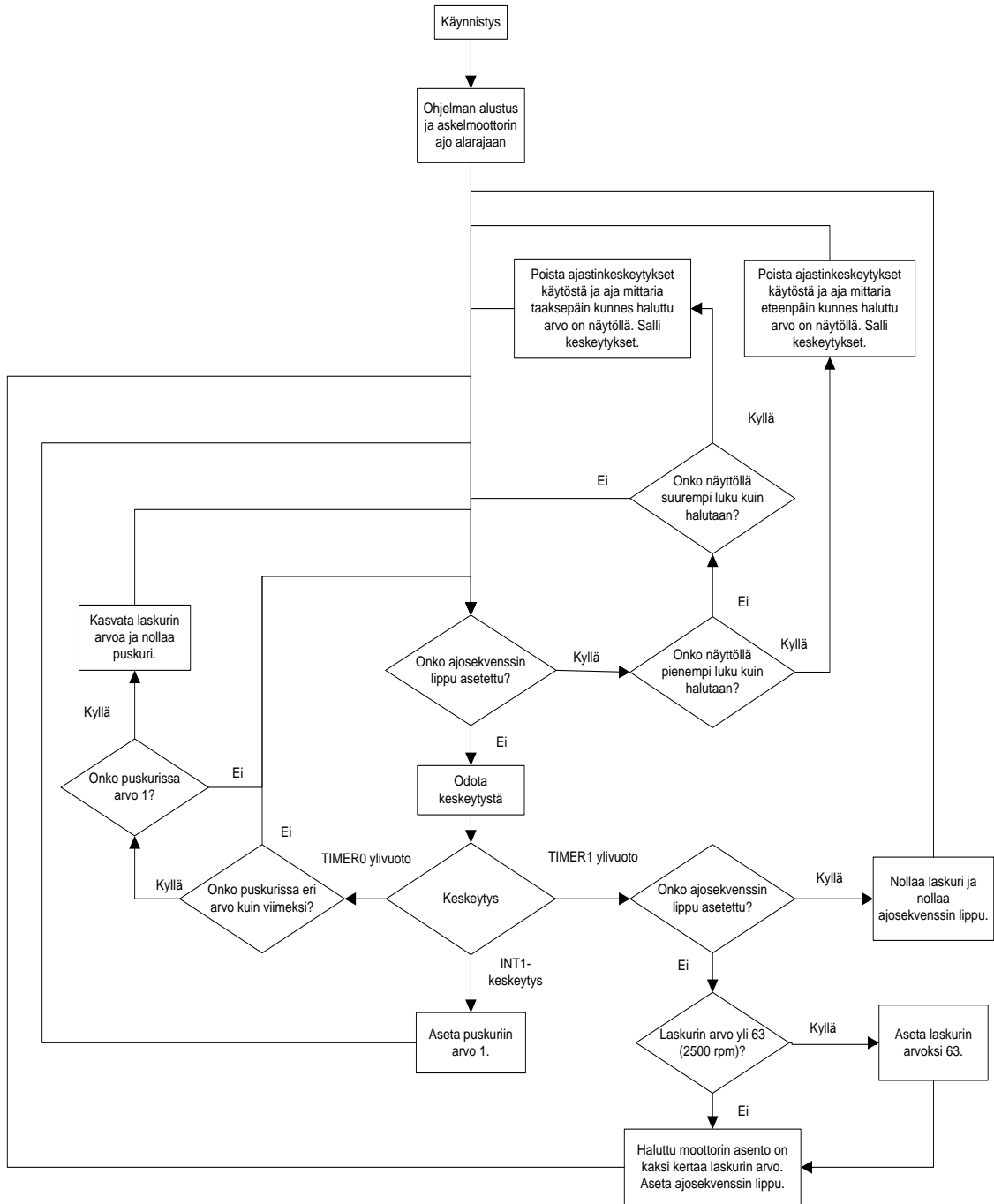
Saatavissa: <http://ww1.microchip.com/downloads/en/AppNotes/00907a.pdf>

Kenjo, T. 1984. Stepper motors and their microprocessor controls. New York: Oxford University Press, 244 s. ISBN 0-19-859326-0.

Proakis, J. G. & Manolakis, D. G. 1997. Digital Signal Processing: Principles, algorithms, and applications. Third edition. New Jersey: Prentice-Hall, Inc, 968 s. ISBN 0-13-394338-9.

Pyrhönen, J. & Nerg, J. 2004. Sähkömagnetismi, Opetusmoniste. LTKK Sähkötekniikan osasto, 301 s. ISBN 951-764-625-9.

Yeadon, W. H. & Yeadon A. W. 2001. Handbook of small electric motors. New York, McGraw-Hill, 1040 s. ISBN 0-07-072332-X



LIITE II

MIKROKONTROLLERIN OHJELMAKOODI

```
// *****Definitions*****
#define ADJ_DELAY 500 // Delay in µs for motor driving sequence

// *****Library inclusions*****
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>

// *****Global variable declarations*****
volatile int drive_or_count = 0; // Drive/counting sequence flag
volatile uint8_t previous = 0; // Previous pulse buffer value
volatile uint8_t buffer = 0; // Pulse buffer
volatile uint8_t counter = 0; // Counter variable
volatile int8_t flop = -1; // Rotor position tracker
volatile int16_t curr_pos = 0; // Motor position tracker
volatile int16_t des_pos = -150; // Desired position

// *****Function declarations*****
void forward(void);
void reverse(void);

// *****Main program*****
int main (void)
{
    // *****Initialize PORTS*****
    DDRB |= ((1 << PB0) | (1 << PB1) | (1 << PB2) | (1 << PB3) | (1 << PB5) |
    (1 << PB6));
    // Set PB0, PB1, PB2 and PB3 as output and PB5 and PB6 as output for de-
    bugging purposes

    PORTB = 0x00; // Disable internal pull-ups for PORTB
    DDRA &= ~(1 << INT1) | (1 << PA0); // Set INT1 and PA0 as inputs
    PORTA |= ((1 << PA0) | (1 << INT1));
    // Disable internal pull-ups for INT1 and PA0

    // *****Initialize external interrupt INT1*****
    MCUCR |= (1 << ISC01);
    MCUCR &= ~(1 << ISC00);
    // Set external interrupt configuration to interrupt on falling edge

    GIMSK |= (1 << INT1); // Enable external interrupts on pin INT1

    // *****Initialize TIMER1*****
    TCCR1A |= 0x00; // Default settings for Timer/Counter1
    TCCR1B |= ((1 << CS13) | (1 << CS12));
    TCCR1B &= ~(1 << CS11) | (1 << CS10); // Set prescaler to 2048
    TCCR1C |= 0x00; // Default settings
    TCCR1D |= 0x00; // Default settings (normal operation)
    TCNT1 |= 0x9D;
    // Set TIMER1 value to 0x9D for 0,2 sec delay (calibrated with oscillos-
    cope)

    TC1H &= ~(1 << TC19) | (1 << TC18);
    // Two highest bits of TIMER1 set to 0

    TIMSK |= (1 << TOIE1); // Enable overflow interrupt for TIMER1

    // *****Initialize TIMER0*****
    TCCR0B |= (1 << TSM); // Halt Timer/Counter0
    TCCR0B |= (1 << CS02); // Set prescaler to 256
    TCCR0B &= ~(1 << CS01) | (1 << CS00);
    TCCR0A &= ~(1 << TCW0); // Set timer width to 8-bits
    TCNT0L = 0xF4;
```

```

// Set TIMER0 value to (theoretical 0xF3) 0xF4 for 0,003 sec delay (cali-
brated with oscilloscope)

TIMSK |= (1 << TOIE0);          // Enable overflow interrupt for TIMER0

// *****Initialize display and start TIMER0*****
PORTB |= (1 << 0) | (1 << 3);

while(des_pos<curr_pos){        // Drive motor to the beginning
    reverse();
}
des_pos=0;
curr_pos=0;
TCCR0B &= ~(1 << TSM);         // Start TIMER0
sei();

// *****Main loop*****
while(1){
    if(drive_or_count != 0){    // If drive cycle:
        TIMSK &= ~((1 << TOIE1) | (1 << TOIE0));
        // Disable overflow interrupt for timer1

        if(des_pos>(curr_pos)){
            // If current position is less than desired position:
            while(des_pos>curr_pos){
                forward();      // Drive forward
            }

            TCNT1 |= 0xF0;      // Set TIMER1 to overflow
            TC1H |= ((1 << TC19) | (1 << TC18));
            // Two highest bits of TIMER1 set to 1
        }

        if(des_pos<(curr_pos)){
            // If current position is more than desired position:
            while(des_pos<curr_pos){
                reverse();      // Drive backwards
            }
            TCNT1 |= 0xFB;
            // Set TIMER1 to overflow in lms (abort drive se-
            // quence)

            TC1H |= ((1 << TC19) | (1 << TC18));
            // Two highest bits of TIMER1 set to 1
        }
        TIMSK |= ((1 << TOIE1) | (1 << TOIE0));
        //Enable overflow interrupt for TIMER1 and TIMER0
    }
}

// *****Interrupt service routines*****

// TIMER1 overflow interrupt
ISR(TIMER1_OVF_vect){
    if(drive_or_count==0){      // If counting sequence ended:
        des_pos = counter*4;    // Set desired position
        drive_or_count=1;      // Begin drive sequence
    }
    else{                       // If drive sequence ended:
        counter = 0;           // Reset counter variable
        drive_or_count=0;      // Start counting pulses
    }
    TCNT1 |= 0x9D;
    // Set Timer1 value to 0x9D for 0,2 sec delay

    TC1H &= ~((1 << TC19) | (1 << TC18));
    // Two highest bits of TIMER1 set to 0
}

```

```

}

// TIMER0 overflow interrupt
ISR(TIMER0_OVF_vect){
    if(buffer != previous){
        // If pulse buffer is different than previous value:
        if((buffer == (1 << 0))){ // If pulse buffer value is 1:
            counter = counter + 1; // Increment counter
            buffer = 0; // Reset pulse buffer
        }
        previous = buffer; // Set previous value
    }
    TCNT0L = 0xF4;
    // Set TIMER0 value to (theoretical 0xF3) 0xF4 for 0,003 sec delay
}

// External interrupt INT1
SIGNAL(INT1_vect){
    buffer = 1; // Set pulse buffer as 1
}

// *****Motor driving sequences*****

// Forward sequence
void forward(void){
    flop=flop+1; // Keep track of rotor position
    if(flop>7){
        flop=0;
    }

    if(flop==0){
        _delay_us(ADJ_DELAY);
        PORTB &= ~(1 << 3); // Off
    }
    if(flop==1){
        _delay_us(ADJ_DELAY);
        PORTB |= (1 << 1); // On
    }
    if(flop==2){
        _delay_us(ADJ_DELAY);
        PORTB &= ~(1 << 0); // Off
    }
    if(flop==3){
        _delay_us(ADJ_DELAY);
        PORTB |= (1 << 2); // On
    }
    if(flop==4){
        _delay_us(ADJ_DELAY);
        PORTB &= ~(1 << 1); // Off
    }
    if(flop==5){
        _delay_us(ADJ_DELAY);
        PORTB |= (1 << 3); // On
    }
    if(flop==6){
        _delay_us(ADJ_DELAY);
        PORTB &= ~(1 << 2); // Off
    }
    if(flop==7){
        _delay_us(ADJ_DELAY);
        PORTB |= (1 << 0); // On
    }
    curr_pos=curr_pos+1; // Update current position
    return;
}

// Reverse sequence

```

```

void reverse(void) {
    flop=flop-1; // Keep track of rotor position
    if(flop<0){
        flop=7;
    }

    if(flop==7){
        _delay_us(ADJ_DELAY);
        PORTB &= ~(1 << 0); // Off
    }
    if(flop==6){
        _delay_us(ADJ_DELAY);
        PORTB |= (1 << 2); // On
    }
    if(flop==5){
        _delay_us(ADJ_DELAY);
        PORTB &= ~(1 << 3); // Off
    }
    if(flop==4){
        _delay_us(ADJ_DELAY);
        PORTB |= (1 << 1); // On
    }
    if(flop==3){
        _delay_us(ADJ_DELAY);
        PORTB &= ~(1 << 2); // Off
    }
    if(flop==2){
        _delay_us(ADJ_DELAY);
        PORTB |= (1 << 0); // On
    }
    if(flop==1){
        _delay_us(ADJ_DELAY);
        PORTB &= ~(1 << 1); // Off
    }
    if(flop==0){
        _delay_us(ADJ_DELAY);
        PORTB |= (1 << 3); // On
    }
    curr_pos=curr_pos-1; // Update current position
    return;
}

```