

Opinnäytetyö (AMK)

Tieto- ja viestintäteknikka

2022

Matias Kainulainen

Askelmoottorionhjin ja magneettinen enkooderi

Opinnäytetyö (AMK) | Tiivistelmä

Turun ammattikorkeakoulu

Tieto- ja viestintäteknikka

2022 | 31 sivua

Matias Kainulainen

Askelmoottorihjain ja magneettinen enkooderi

Moottorihjain ja enkooderi ovat askelmoottorin ohjaukseen tarkoitettu yhdistelmä. Yhdistelmä parantaa askelmoottorin tarkkuutta, varmuutta ja tuo loki- sekä sijaintikorjausmahdollisuuden askelmoottorille. Opinnäytetyön tavoitteena oli luoda ajuriohjelman moottorihjaimen ja enkooderin yhdistelmälle kommunikoinnin saavuttamiseksi sekä perehtyä kyseisiin piireihin ja toimintaperiaatteisiin.

Kommunikaation rajapintana toimii SPI-väylä, jota pitkin lähetetään ja luetaan rekistereiden arvoja ohjelmakehyksellä ja ohjataan askelmoottoria tälle sovelletuilla komennoilla. Opinnäytetyön tarkoituksena oli tuoda esiin enkooderilla toimivan moottorihjainjärjestelmän vahvat puolet ja vastata kysymykseen, miten kaikki toimivat yhdessä. Opinnäytetyö toteutettiin yrityksessä käyttämällä kolmannen osapuolen komponentteja pääpiirilevyä lukuun ottamatta.

Työssä saavutettiin ensimmäinen vaihe, jossa enkooderi lukee magneetin sijaintia napojen jännitteenvaihteluun nähden. Opinnäytetyö mahdollistaa laitteen jatkokehityksen, ideana on tehdä toimiva tuote suljettu silmukka systeemillä. Enkooderi ja moottorihjain kommunikoivat keskenään ja enkooderi korjaa askelmoottorin virheitä.

Asiasanat:

Moottorihjain, enkooderi, askelmoottori.

Bachelor's Thesis | Abstract

Turku University of Applied Sciences

Information and Communications Technology

2022 | 31 pages

Matias Kainulainen

Stepper motor controller and encoder

The motor controller and the encoder are a combination for a stepper motor. The combination improves the accuracy and reliability of the stepper motor and creates the possibility of log and position correction to the stepper motor. The main objective of this thesis was to create a driver program for a combination of a motor controller and an encoder to achieve communication and to examine/study these circuits and their operating principles. In particular, the objective was to highlight the strengths of a motor control system operating with an encoder and to answer the question of how everyone works together.

The interface of the communication was an SPI bus, along which the values of the registers are transmitted and read with a program frame and the stepper motor was controlled by commands applied to it. The thesis was carried out on behalf of a company using third-party components, except for the main circuit board.

The main objective of this thesis was achieved; the encoder reads the position of the magnet with respect to the voltage variation. The work carried out in this thesis enables the further development of the device, which would be to create a functional product with a closed loop system. The encoder and motor controller communicate with each other, and the encoder corrects the errors of the motor.

Keywords:

Motor driver, encoder, stepper motor

Sisältö

Lyhenteet tai sanasto	6
1 Johdanto	8
2 Järjestelmä	12
2.1 Elektroniikka	13
2.2 Askelmoottori	13
2.3 Moottorihjain	13
2.3.1 Koko- ja puoliaskleet	14
2.3.2 Mikroaskleet	14
2.3.3 Nopeuden optimointi	15
2.4 Sarjaliitântä rajapinta	16
2.5 Magneettinen enkooderi	17
2.6 Mekaniikka	18
3 Ohjelmisto	20
3.1 Ohjelmointityökalut	20
3.2 Toimintaperiaate	20
4 Toteutus	21
5 Lopuksi	29
Lähteet	30

Kuvat

Kuva 1. Suunnitelma kytkennästä.	12
Kuva 2. Radan mekaniikka.	19
Kuva 3. Piirikorttien ja tietokoneen välinen kommunikaatioon liittyvät johdot.	21

Kuva 4. Datan havainnointi oskilloskoopilla.	22
Kuva 5. Enkooderin kytkentä.	23
Kuva 6. Funktion SPI:n lähetys-, luku- ja virheentarkistuskehystä.	26

Taulukot

Taulukko 1. Moottorin nopeusrekistereiden ohjelmointi. (STMicroelectronics 2015, s. 40.)	15
--	----

Lyhenteet tai sanasto

l_{steps}	Pituus askeleina
n	Askeleiden määrä
t	Aika sekunteina
v_{max}	Enimmäisnopeus
v_{min}	Vähimmäisnopeus
AGC	Automaattinen tasonsäätö (engl. Automatic Gain Control)
AS5147	Magneettisen enkooderi anturipiirin mallinumero
ask/s	Aika-askel
ask/s ²	Aika-askel toiseen
AVR32	Atmelin kehittämä 32-bittinen mikro-ohjainperhe
CS	Ensisijaisen piirilevyn valitsema toissijainen komponentti (engl. Chip Select)
CK	Ensisijaisesta piirilevystä lähetettävä prosessorin kellopulssi (engl. Serial Clock)
IC	Integroitu piiri (engl. Integrated Circuit)
I/O	Sisään- ja ulostulo (engl. Input/Output)
L6470	Moottoriohjaimen mallinumero
LED	Valoa loistava diodi (engl. Light emitting diode)
MCU	Mikro-ohjaus yksikkö (engl. Microcontroller Unit)
MFD	Yrityksessä kehitetty komponentti, jota käytetään opinnäytetyössä pää piirilevynä

MSB	Eniten vaikuttava bitti (engl. Most Significant bit)
MOSI	Pääpiirikortin ulostulo, toissijaisen piirilevyn sisääntulo (engl. Master out slave in)
MISO	Pääpiirikortin sisääntulo, toissijaisen piirilevyn ulostulo (engl. Master in slave out)
ns	Nanosekunti
SPI	Sarjamuotoinen oheislaite rajapinta (engl. Serial Peripheral Interface)
USB	Universaalinen sarjaväylä (engl. Universal Serial Bus)
V	Jännitteen yksikkö voltti
RAM	Keskusmuisti, prosessori tallentaa väliaikaista tietoa ja pääsee nopeasti lukemaan. (engl. Random Access Memory)
Rekisteri	Prosessoriin varattu nopeaa muistialuetta. Sisältää jonkin tietyn määrän bittejä. (ComputerScience.GCSE.GURU n.d)

1 Johdanto

Sähkömoottoreita pystyy ohjaamaan enkooderilla ja enkooderittomalla järjestelmällä. Ilman enkooderia oleva sähkömoottori on halpa, yksinkertainen ja laskentateholtaan vähäinen, mutta sen käyttötarkoitus niin sanotussa sokeassa ajossa on riippuvainen käyttäjästä. Enkooderitonta järjestelmää pystytään kuitenkin ohjailemaan ja tarkkailemaan ulkopuolisilla mittareilla. Roottorin nopeutta konedynaamisten yhtälöiden avulla, jännitettä ja virtaa mittaamalla estimaattorilla eli arvioijalla saadaan arvio sähkömoottorin liikkumisesta, mutta pienimillä nopeuksilla arviot voivat heilua runsaastikkin. (Mughees, N. 2021.)

Askelmoottori on sähkömoottori, jota pystytään ohjaamaan askel kerrallaan sähkömagneettien ansiosta. Se koostuu keskiroottorista ja staattorista, jossa on tietty määrä käämejä. Yhteen käämiin tai käämisarjaan ohjataan jännitettä, jolloin niistä tulee sähkömagneettisia. Sähkömagneetin napaisuus määrittellään virran suunnalla. Roottorin hampaiden on oltava väärin kohdistettu seuraavaan käämiin nähden, joka saa virran. Roottori pyörähtää kohtisuoraan virroissa olevaa käämiä vasten ja saavutetaan yksi askel. (Lewotsky, K. 2014.)

Servomoottoreita on erityyppisiä, hydraulisia, sähköisiä ja pneumaattisia. Servomoottoreita käytetään enkooderin kanssa. Niiden toimintaperiaate perustuu sähkömoottoriin, jonka vääntömomenttia kasvatetaan ulostuloon hammaspyörien avulla moninkertaistaakseen sisääntuloenergian. Servomoottori on suljettu silmukka, suljetulla silmukalla tarkoitetaan itseohjautuvaa komponenttia. Virheenkorjaus on suljetussa silmukassa saavutettu moottorin lähettämällä koordinaatilla ja enkooderilla ohjaamalla. Enkooderi on voitu sijoittaa moottorin kanssa saman rungon sisälle. Suljettu silmukka mahdollistaa moottorin sijainnin löytämisen heti moottorin päälle panemisen jälkeen. Askelmoottorissa sijainti on löydettävä enkooderille liikuttamalla moottoria, koska askelmoottori ei lähetä tietoa enkooderille samalla tavalla kuin servomekanismissa. Servomoottoreille tyypillisiä käyttötarkoituksia ovat ohjaustehostin, työkoneet ja radio-ohjattavat pienoismallit. (Sawicz, D. 2012.)

Askelmoottorityypit jaetaan kolmeen pääkategoriaan: kestopagneetti-, reluktanssi- ja hybridiaskelmoottoreihin. Kestomagneettiaskelmoottorissa on pituussuuntaisesti magnetoitu akseli, joka on magnetoitu plus- ja miinusnavoilla vuorotellen. Kestomagneettiaskelmoottorissa on mahdollista saavuttaa suuri vääntömomentti, mutta se ei pysty toimiaan suurissa nopeuksissa. Reluktanssiaskelmoottori on kolmesta tyypistä yksinkertaisin. Moottorin akseli on hammasrattaan muotoinen raudasta koostuva magneetitön roottori ja staattorikämeihin annetaan vuorotellen virtaa, joka vetää rautaista hammasratasakselia niitä kohti. Staattorikämejä oikeaan tahtiin aktivoimalla saavutetaan pyörivää liikettä akselissa. Reluktanssiaskelmoottorissa on suhteellisen pieni vääntömomentti ja tehon noustessa vääntömomenttikäyrän pudotus on pienempi. Moottori soveltuu keski- ja suurnopeuksiin käyttötarkoituksiin. Hybridiaskelmoottorissa yhdistyy kestopagneetti- ja reluktanssiaskelmoottorien toiminta. Roottorissa on kaksi kestopagneettista hammaspyörää, plus- ja miinusnapaisina. Hammaspyörät ovat lomittain toisiaan vasten, joten staattorilla on aina jompikumpi napa kohdallaan. Roottorin hampaiden lomittainen asento mahdollistaa hybridi-moottorille paremman tarkkuuden edellä mainituista kahdesta moottorista. Staattorikämiä miinus ja plus jännitevaihtelua säädellen tapahtuu ilmiö, jossa plusnapainen liikahtaa miinusnapaistakämiä ja miinusnapainen liikahtaa plusnapaistakämiä kohti. Edellä mainittua tapaa toistaen moottori saadaan liikkumaan tarkasti noin $0,8^\circ$ kerrallaan. (Collins, D. 2018.)

Enkooderiantureiden käytön hyöty ulottuu mittavasti robotiikkaan ja teollisuus-automatiikkaan. On olemassa monia eri periaatteella toimivia enkoodereita, joita käytetään monissa teollisuuden osa-alueissa. (Ganssle, J. 2012.)

Erityyppisiä enkoodereita ovat mekaaniset, optiset, akselilla (engl. on-axis) olevat ja pois akselilta (engl. off-axis) olevat magneettiset enkooderit. Mekaanisella enkooderilla tarkoitetaan esimerkiksi hammaspyörään osuvaa kahta hieman toisistaan erillään olevaa kytkintä. Kun kytkin on pohjassa, lähetetään piirille jokin jännite, ja piiri muuntaa sen digitaaliseksi signaaliksi, esimerkiksi kanttiaalloksi. Mekaanisen enkooderin epäkohta on osien kuluminen, tyypillinen laitteen huoltoväli on 15 000–1 000 000 kierrosta. Mekaanisia enkoodereita käytetään

harvemmin huoltotarpeen takia teollisuudessa, sen sijaan käyttö ulottuu enemmän ihmisten käyttämiin säätimiin. (Ganssle, J. 2012.)

Optiset enkooderit käyttävät valoa säteilevää diodia (LED), jota näyttämällä reikäiseen levyyn valon läpäisemiseksi saadaan tehtyä signaalia. Valodiodi lukee LEDiä ja levyssä olevissa rei'istä diodi saa valoa, ja seinien kohdalla diodi ei näe valoa yhtään. Tällä periaatteella saadaan tehtyä digitaalisignaali 0:ksi tai 1:ksi. Optisen anturin käyttöikä voi olla 300 miljoonan ja ei-arvioidun välillä, sillä muita liikkuvia osia ei ole mitattavan levyn lisäksi. Käytännössä ainoat optisen enkooderin taakkaa kuormittavat ongelmat ovat LEDin palaminen ja ympäristö johon anturi on sijoitettu, pölyn, lian tai veden vuoksi. Magneettinen enkooderi käyttää Hallin ilmiötä (Ganssle, J. 2012.), jossa magneetti tuomalla sähkökenttään elektronien suunta vaihtuu ja saadaan induktiomuutos. (P. Procházka, F. Vaněk 2012.)

Opinnäytetyön tarkoituksena oli kehittää ajuriohjelma magneettisen enkooderin piirille ja moottoriohjaimelle. Magneetilla toimivia enkoodereita on kaksi, joiden toimintaperiaate on sama, mutta magneetin muoto ja piirin sijainti vaihtelevat. Akselilla olevassa enkooderissa magneetti on sijoitettu piirin keskelle ja ainoastaan kaksinapainen magneetti tarvitaan anturin toimivuuden takaamiseksi. Pois akselilla olevalla magneettisella enkooderilla tarkoitetaan anturia, joka lukee magneettilevyä. Ideana on levyn koko mitalle sijoitettujen vuorotellen negatiivisen ja positiivisen magneettien lukeminen. (Ganssle, J. 2012.) Akselilla olevaa magneettista enkooderia tullaan käyttämään opinnäytetyössä.

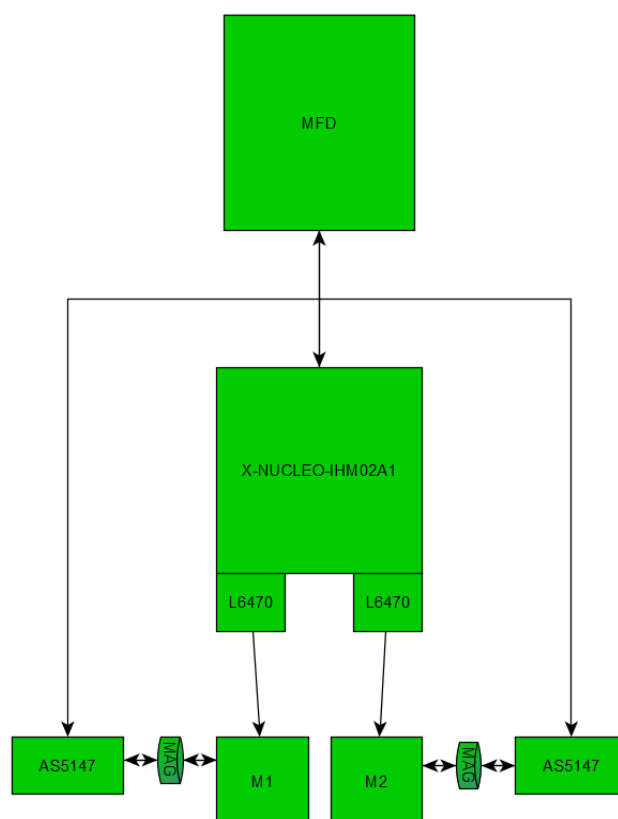
Moottoriohjain ohjaa moottoria ja säätelee moottorille annettavaa virtaa yksilöllisesti. Ohjaimessa voi olla erilaisia suojaominaisuuksia laitteen turvaamiseksi, esimerkiksi ylivirta, lämpötila, matala kommunikaatiojännite, askeleiden häviön tunnistus ja moottorin pysähtymisen ilmoitus (STMicroelectronics n.d.), jolla tarkoitetaan enneaikaista tai odottamatonta pysähtymistä. Ilmoitus johtaa virhe-rekisterin bitin muuttumiseen 0:sta 1:ksi.

Jos moottoriohjainta käytetään ilman enkooderia, voi prosessori hävittää oikean moottorin sijainnin ja tunnettu sijainti on löydettävä joko sensoreilla tai törmää-

mällä ääriasentoon virtapiikin saavuttamiseksi. Virtapiikin tullessa on ohjelmoitavissa moottorin pysäytys, ja sen hetkinen sijainti on oletettavasti tunnettu. Enkooderilla estetään tuntemattoman sijainnin ongelma, kun lakkaamalla voidaan monitoroida moottorin koordinaatistoa ja viedä moottori oikeaan tunnettuun sijaintiin.

2 Järjestelmä

Kehitysympäristö rakennetaan yrityksen sisäiseen ympäristöön. Ympäristöön asennetaan demopiirilevy mallia X-NUCLEO-IHM02A1. Piirikorttiin on integroitu kaksi moottoriohjainta (L6470) ja lisäksi kaksi magneettista enkooderi anturiäpiiriä (AS5147) kytketään sarjamuotoista oheislaite rajapintaa (SPI) väylään. Komponentit kytketään demopiirilevyyn ja kytkentä saa virran yrityksessä kehitetystä komponentista, jota käytetään opinnäytetyössä pää piirilevynä (MFD), joka toimii myös ohjelmointialustana. Ohjelmointi tehdään C-kielillä. Kytkentä tehdään kuvan 1 mukaisella tavalla, komponentit ruuvataan alumiiniseen alustaan ja universaalisella sarjaväylä (USB) tyyppin kaapelilla, jolla otetaan yhteys tietokoneen ja MFD-piirikortin välillä. Piirikortit kommunikoivat SPI-väylän avulla.



Kuva 1. Suunnitelma kytkennästä.

2.1 Elektroniikka

Elektroniikkana käytetään demo piirilevyä, joka soveltuu moottoriohjaimen ja enkooderin suoraaniseen kytkentään. Rajapinta toimii SPI:llä, sillä MFD:ssä on suora ulostulo SPI-väylään. Näin ei tarvitse tehdä suurempia muutoksia piirilevyyn elektroniikan osalta, kun kaikki komponentit käyttävät samaa rajapintaa ja työssä voidaan keskittyä ohjelman luomiseen. Moottoriohjaimen piirilevyn malliksi valittiin X-NUCLEO-IHM02A1, sillä kyseisessä piirilevyssä on valmiina kaksi moottoriohjainta ja sisäänrakennetut oskillaattorit.

2.2 Askelmoottori

Askelmoottori muuntaa tehoa liikkeeksi eri tavoin. Askelmoottoria pystyy ohjaamaan tarkasti, esimerkiksi kuinka pitkään ja nopeasti akselia pyöritetään. Tarkkuus saadaan aikaan siksi, koska askelmoottori liikkuu sähköpulssin mukaisesti. Sähköpulssilla vietävä signaali antaa myös mahdollisuuden helppoon ohjelmointiin. (Oriental motor U.S.A corp n.d.)

2.3 Moottoriohjain

Ohjain liikuttaa moottoria niin paljon kuin sitä käsketään riippumatta siitä, missä kohtaa moottori sillä hetkellä on. Moottoriohjaimeen ohjelmoidaan ajuri, jotta saadaan yhteys SPI-väylään ja mahdollistetaan ohjaus. Jos moottorin sijaintia ei tarkisteta liikuttamisen jälkeen halutun askelmäärän saavuttamiseksi, voi tulla virheaskelia. Askelmoottorit ovat tarkkoja moottoreita, mutta on syytä, jolloin moottori jättää. Moottorin sisällä olevat magneettinavat eivät kosketa moottorin akselia, näin voi tapahtua yli- tai aliaskelia. Kommunikaatio on nopeampaa kuin mekaanisesti moottorin liikkuvuus, moottori jättää ja näin syntyy virheitä. Moottoriin kohdistuu liikaa kuormitusta tai moottori resonoi eli tekee myötävärähtelyä. (Ato 2017.)

Moottoriohjaimessa L6470 on 25 kappaletta ohjelmoitavia rekistereitä ja jokaista rekisteriä pystytään ohjaamaan bittitasolla, jokainen bitti vastaa yhtä tiettyä rakennetta. Esimerkkinä rekisteri askel tilan valinta, rekisterin nimi STEP_SEL. Rekisterin pituus on kolme bittiä. Antaessaan koodiin bittien arvoiksi 000, rekisteri vaihtaa moottoriohjaimen askeltilaksi täysiaskeltilan, vastaavasti, jos bittien arvot ovat 111 vaihdetaan tilaksi 1/128 mikroaskeltila. (STMicroelectronics 2015, s. 1.)

2.3.1 Koko- ja puoliaskeleet

Kokoaskeltilaa voidaan käyttää yksi tai kaksi vaiheisena. Moottori käyttää vähiten tehoa verrattuna muihin tiloihin yksi vaiheisena, sillä yksi käämi saa virtaa kerrallaan. Kaksi vaiheisena kaksi vierekkäin olevaa käämiä saavat täyden virran ja akseli jää kyseisten aktiivisten käämien väliin. Kaksi vaiheisen hyötynä tulee nopeus ja vääntömomentin suurentuminen, mutta virtaa tarvitaan enemmän kahden käämin päälle saantiin. (RS, G. 2015.)

Puoliaskeltilassa yhdistetään kokoaskeleen yksi ja kaksi vaiheiset toimintaperiaatteet. Yhteen käämiin saatetaan virtaa ja akseli pysyy paikallaan, kun seuraavaan käämiin tuodaan virta, liikkuu akseli käämien keskelle, otetaan ensimmäisestä käämistä virran tuonti pois ja akseli siirtyy eteenpäin toisen käämin kohdalle. Edellä mainittua mallia toistaen puoliaskeltila tuo sulavamman liikkeen akseliin, mutta menettää vääntömomenttia noin 15 % verrattuna kaksivaiheiseen kokoaskeltilaan. (RS, G. 2015.)

2.3.2 Mikroaskeleet

Mikroaskeleissa jokaisen käämin virransaantia voidaan säätää yksittäin, minkä vuoksi yksi askel voidaan jakaa 256 yksilölliseksi askeleeksi. Jakaminen poistaa moottorin nykivän askelliikkeen, tekee moottorista hiljaisen ja pienentää tärinää. Kahden käämin siniaaltosignaali toimii yhteen, kun toinen käämi alkaa pienentämään virtaa, 90° kohdalla toinen käämi alkaa nostamaan virtaa porrastaen. (RS, G. 2015.)

2.3.3 Nopeuden optimointi

Moottorin nopeusprofiilit pystytään ohjelmoimaan yksitellen, määrittämällä nopeus, kiihtyvyys, hidastuminen, enimmäis- ja vähimmäisnopeudet. Nopeusprofiilin rekistereitä voidaan tarkastella taulukosta yksi, josta näemme rekisteriarvon, nollaus heksadesimaalin, suurimman ja pienimmän mahdollisen arvon kustakin nopeuteen liittyvästä rekisteristä.

Taulukko 1. Moottorin nopeusrekistereiden ohjelmointi. (STMicroelectronics 2015, s. 40.)

Rekisterin nimi	Kuvaus	Heksadesimaali osoite	Min	Max	Nollaus heksadesimaali
SPEED	Nykyinen nopeus	0x04	0 ask/s	15625 ask/s	0x00000
ACC	Kiihtyvyys	0x05	14,55 ask/s ²	59590 ask/s ²	0x08A
DEC	Hidastus	0x06	14,55 ask/s ²	59590 ask/s ²	0x08A
MAX_SPEED	Enimmäisnopeus	0x07	15,25 ask/s	15610 ask/s	0x041
MIN_SPEED	Vähimmäisnopeus	0x08	0 ask/s	976,3 ask/s	0x000

Nopeus-, enimmäisnopeus- ja vähimmäisnopeusrekisterit ilmoittavat moottorinopeuden aika-askeleina (engl. steps/tick). Rekisterin arvon voi muuntaa aika-askel muotoon kaavalla yksi. Yksi moottorin hetki on ajassa 250 ns tai 0,00000025 sekuntia. Nopeusrekisterin arvo ei voi mennä enimmäisnopeusrekisterin arvon yli. Nopeusrekistin arvo lasketaan seuraavasti: (STMicroelectronics 2015, s. 42.)

$$\left[\frac{l_{step}}{t} \right] = \frac{v \times 2^{-28}}{t_{tick}} \quad (1)$$

Missä $\frac{l_{step}}{t}$ on pituus askeleina jaettuna ajalla, v on nopeus ja t_{tick} on vakio 250 ns. (STMicroelectronics 2015, s. 43.)

Enimmäisnopeusrekisterin muunnoksen aika-askel muotoon seuraavasti: (STMicroelectronics 2015, s. 43.)

$$\left[\frac{l_{steps}}{t} \right] = \frac{v_{max} \times 2^{-18}}{t_{tick}} \quad (2)$$

Vähimmäisnopeuden muunnoksen aika-askel saadaan laskemalla seuraavasti: (STMicroelectronics 2015, s. 43.)

$$\left[\frac{l_{steps}}{t} \right] = \frac{v_{min} \times 2^{-24}}{t_{tick}} \quad (3)$$

Kiihtyvyy- ja hidastusrekisterit voidaan muuttaa samantyyllisellä kaavalla, muuttamalla seuraavassa kaavassa x paikalle joko kiihtyvyyden tai hidastumisen parametriarvon: (STMicroelectronics 2015, s. 42–43.)

$$\left[\frac{l_{steps}}{t^2} \right] = \frac{x \times 2^{-40}}{t_{tick}^2} \quad (4)$$

Edellä mainittujen kaavojen avulla lasketaan lähetettävä heksadesimaali yksinkertaisemmalla muodolla, joka voidaan implementoida koodiin ja kutsua funktiota pelkällä kokonaisluvulla. Muutos tarvitaan, kun muokataan esimerkiksi nopeusprofiilia hitaammaksi tai nopeammaksi.

2.4 Sarjaliitäntä rajapinta

SPI käyttää ensisijaisen piirilevyn valitsema toissijainen komponentti (CS), kellopulssi (CK), pääpiirikortin ulostulo, toissijaisen piirilevyn sisääntulo (MOSI) ja pääpiirikortin sisääntulo, toissijaisen piirilevyn ulostulo (MISO) pinnejä. Kun CS-pinnin signaali on korkealla, laitetta ei ole valittu ja MISO-linja on epäaktiivinen, eli linjassa on suuri impedanssi. Kommunikaatio alkaa, kun CS-pinnin signaali pakotetaan alas. CK-pinniä käytetään datan kommunikaation

synkronoimisessa. Jokainen komento ja databitti siirretään MOSI:lla eniten vaikuttava bitti (MSB) ensimmäisenä. Jos paluu arvoa ei ole mahdollista lähettää, lähetetään takaisin 0:sta koostuvaa tavua. (STMicroelectronics 2015, s. 38.)

X-NUCLEO-IHM02A1 piirilevyssä CS-pinni sijaitsee CN8 pinnivyöhykkeellä kolmannessa pinnissä, CK löytyy pinnivyöhykkeestä CN9 pinnistä neljä, MOSI- ja MISO-pinnit sijaitsevat pinnivyöhykkeessä CN5 pinneissä neljä ja viisi. Lisäksi piirilevy tarvitsee 3,3 voltin johdon ja maajohdon, pinnit ovat pinnivyöhykkeessä CN6 pinneissä neljä ja kuusi.

SPI-rajapinta mahdollistaa monien moottoreiden ja moottoriohjainten kytkemisen yhteen CS-johtoon, data liikkuu jokaisen moottoriohjaimen lävitse ja vuotaa aina seuraavalle, tekemällä kytkennästä suljetun silmukan. Prosessorista lähettäessä jokin heksadesimaaliarvo, tämän jälkeen kaksi kertaa nolla tavu, jos kytkennässä on kiinni kaksi moottoria, palaa lähetetty arvo prosessorille takaisin ja moottoriohjaimet ovat nollattu. Moottoreiden lukumäärätiedolla pystytään lisäämään kehykseen tarvittava määrä lähetettäviä nolla tavuja kaikkien kytkennässä olevien moottoreiden nollaamiseksi. Edellä mainittua periaatetta kutsutaan ketjutukseksi (engl. daisy chain)

2.5 Magneettinen enkooderi

Magneettisia enkoodereiden mittaustyyppisiä on kaksi, ne voivat lähettää absoluuttista tai suhteellista dataa. Suhteellisessa enkooderissa sähkökatkon takia pitää enkooderin löytää tiedetty sijainti uudestaan, koska piiri laskee 0, 1, 2, n niin kauan, että bittinumerot loppuvat ja laskeminen alkaa alusta. Aina kun järjestelmä käynnistetään, alkaa laskeminen nollasta. Absoluuttisessa tyyppissä kulma data on aina saatavilla magneettisesti fyysisestä maailmasta, joten sähkökatkoksenkin tullessa ja magneettia välissä pyörittäen, löytää piiri paikan. (Anderson, M. 2019.)

Magneettisen enkooderin komponenttina käytetään AS5147. AS5147 yhdistää Hall-elementtejä, analogisen käyttöliittymän ja digitaalisen signaalikäsittelyn yhdessä laitteessa. Sisäisen jännitesäätimen ansiosta käyttöjännite voi olla 3,3 V

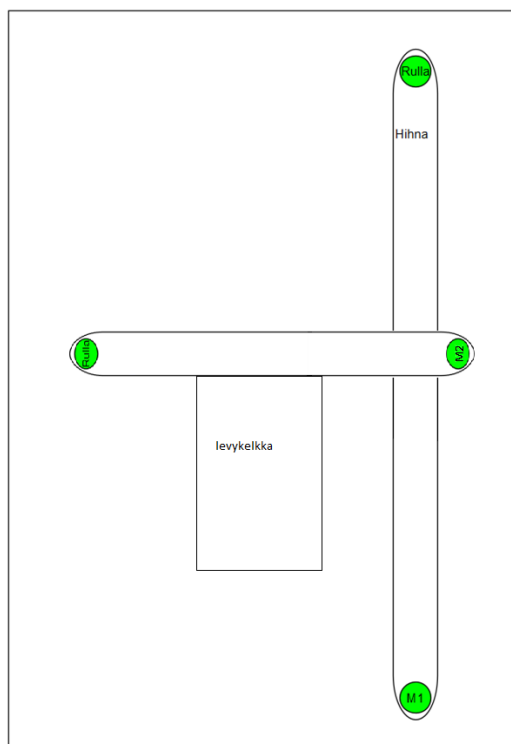
–5 V. Magneettinen enkooderi anturi tarkistaa moottorin sijainnin ja korjaa virheitä, komponentti on ohjelmoitavissa mikro-ohjaus yksikön (MCU) avulla. AS5147 tukee 14-bittistä ydinresoluutiota ja SPI-rajapintaa. (Ams Osram-AG 2021, s. 1)

Absoluuttista enkooderi datatyyppiä on opinnäytetyössä tärkeä käyttää, sillä paikka on löydettävä aina laitteen sammuttamisen jälkeen ilman sijaintihakua. Magneetin alla ei saa olla rautaosaa, sillä magneettikenttä oikosulkee navat ja pienentää magneettikenttää. Magneetin ja rautaosan väliin täytyy laittaa eristettä.

AS5147 tukee itsediagnosoivaa sulautetulla tasolla tapahtuvia virheitä, jos magneettikenttä on liian suuri ja voimakas, voi automaattinen tasonsäätö (AGC) lähettää virhekoodin 0x00, tämä on merkki epälineaarisuusvirheestä. Magneettikentän ollessa liian pieni, lähetetään virhekoodi 0xFF, virhekoodi on merkki mitatun kulman lähtökohinan lisääntymisestä. (Ams Osram-AG 2021, s. 1.) Tyypillisesti magneetin ja Hall sensorin keskellä oleva ilma-aukko on 1 mm–2 mm (Einspieler, R. 2014.)

2.6 Mekaniikka

Mekaniikka koostuu alumiinisesta kehyksestä, jossa on moottorit, moottoreissa on paikat magneeteille ja piirilevyille. Moottorit tulevat kiinni kehykseen, jotka ovat kiinni hihnoissa ja hihnat liikuttavat ratoja. Moottorissa numero kaksi oleva akseli ei ole käytössä. Levykelkka on lyhemmässä x-radassa kiinni kuvan 2 mukaisesti, y-radon hihna liikuttaa koko x-akselin rataa.



Kuva 2. Radan mekaniikka.

M1-moottori on kiinni alustassa ja M2-moottori on M1-moottorin ohjaamassa alustassa. Projektissa ei ohjelmoida M2-moottoriohjaimeen parametreja ja moottori ei ole kytketty käyttöön, moottori-ohjain otetaan huomioon ohjelmoinnissa ja tehdään tarvittavat lähetys- ja vastaanottokehykset oikean dataliikkeen saavuttamiseksi.

3 Ohjelmisto

Ohjelma koostuu kahdesta erillisestä tiedostosta, johon on laitettu moottorihjaimen ja enkooderin kommunikaatioon liittyvät komennot, lähetys- ja vastaanottamiskehykset. Lisäksi molempiin tiedostoihin tehdään omat ostikkotiedostot (engl. header file) näissä tiedostoissa ohjelmoidaan rekisterikirjasto ja makrojen määritelmät, jotta voidaan käyttää useassa C-kielen tiedostoissa funktioiden sisällä ja ohjelman suorittamisen yhteydessä makrot eivät käytä prosessorilta aikaa, koska käännös vaiheessa tehdään makroilla funktioihin vaikuttavat muutokset.

3.1 Ohjelmointityökalut

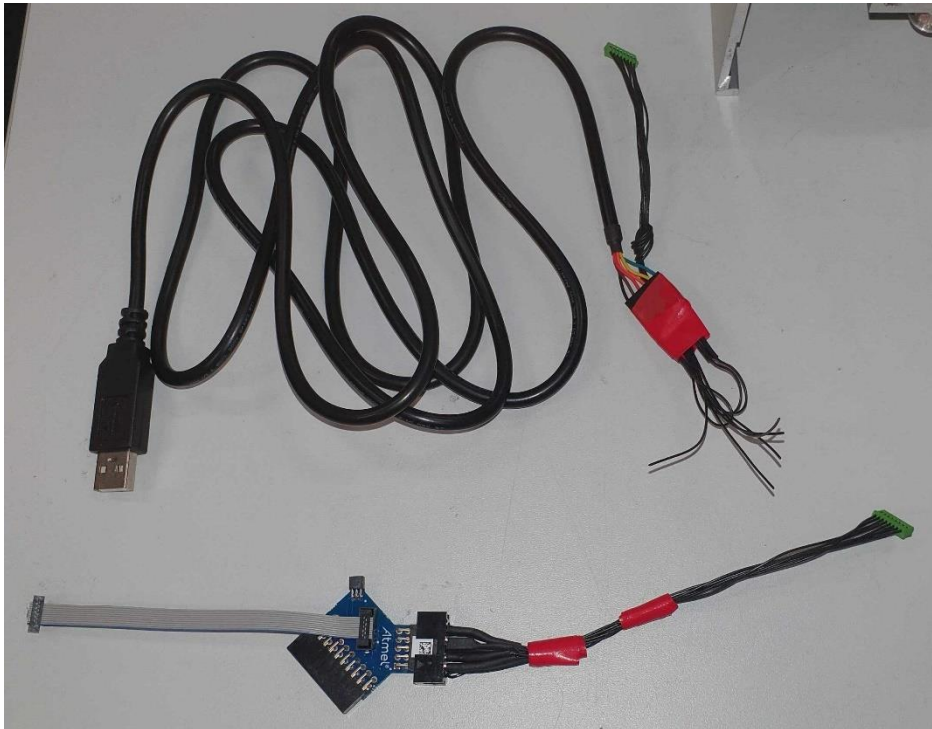
Ohjelmointityökaluja on käytössä terminaaliohjelmat, joissa on USB-luku mahdollista, Linuxilla Minicom ja Windowsilla HyperTerminal, joita käytetään ohjelmiston kehittämiseen. Ohjelmointiympäristönä on Kate Linux, joka on normaali tekstikäsittelyohjelma, mutta tukee lisäosia. Lisäosilla ympäristö tuntee C-kielen kirjoitustyylin ja CTagsia käytetään funktioiden ja muuttujien löytämiseen tiedostojen seasta.

3.2 Toimintaperiaate

Ohjelmointi aloitetaan otsikkotiedostojen tekemisestä. Prosessorin pinnit täytyy ohjelmoida uudelleen, sillä CS-pinni otetaan prosessorin normaalista sisään- ja ulostulo (I/O) pinnistä. Tämä I/O pinni liitetään hyppykaapelilla liittimen pinniin ja koodissa ohjelmoidaan AVR32:lle ymmärrys pinnistä. Alustus sisältää pinnin käyttötarkoituksen ja sen, mikä pinni on kyseessä. Enkooderin ohjelmoiminen on ensimmäinen tehtävä, jotta saadaan SPI-yhteys varmistettua toimivaksi. Komponentin datalehdessä nähdään moottorihjaimen profilointiin tarvittavat komennot ja niillä mahdollistetaan moottorin liikuttaminen muun muassa raja-arvot jännitteille, askeltilan ja nopeuden säädön.

4 Toteutus

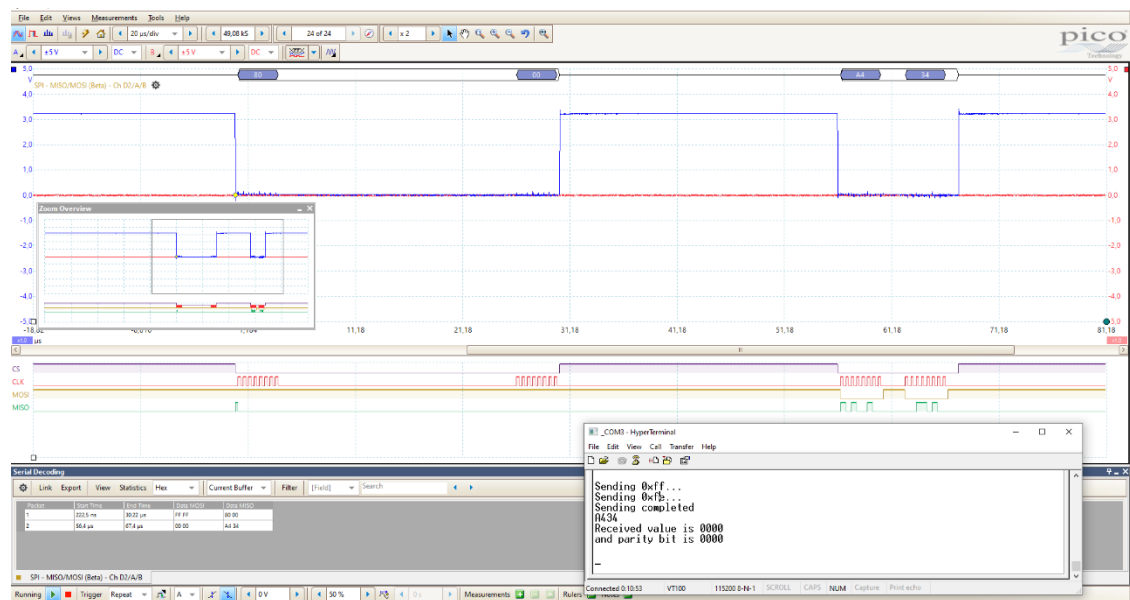
Työ aloitettiin elektroniikan kokoamisella. Piirilevyihin piti tehdä kaksi johtoa, joista toinen liitetään ohjelmoijalaitteen ja piirilevyn väliin. Käynnistysohjelman asentamiseen piirikortille tarvittiin 10 pinnisestä liittimestä kahdeksan pinnin liittimeen, jotta tietokoneesta ja ohjelmoijan ohjelmasta pystytään lukemaan prosessorin malli ja lähettämään käynnistysohjelmalla toimiva ensimmäinen sulautettu versio. Ensimmäisen ohjelmaversion päälle rakennetaan tämä työ, joka kirjoitetaan RAM:lle. Toinen johto tehtiin USB-johdosta, johon lisättiin toiseen päähän 10 pinninen liitin ja sitä käytetään piirikortin väliseen ohjelmointiin. Johdolla mahdollistetaan tietokoneen ja MFD-piirikortin kommunikaatio. Johtojen toteutus voidaan nähdä kuvassa 3.



Kuva 3. Piirikorttien ja tietokoneen välinen kommunikaatioon liittyvät johdot.

Alussa oli tarkoitus saada SPI-yhteys AVR32-prosessorin ja enkooderin välille. Ennen minkäänlaista rekisterin lähetystä, varmistus on saatava perusasioista.

Perusasioihin liittyvät CK lähetys, joka on tasaista kanttiaaltoa, se pitää näkyä liittimestä oskilloskoopilla. Kommunikointi alkaa CS-signaalin ollessa nolla, jolloin piiri menee vastaanottotilaan. CK- ja CS-signaalien ohjaamisesta siirrytään MOSI- ja MISO-signaaleihin, joiden kommunikaatio täytyy toimia CS-signaalin ollessa nolla. Muuta kommunikaatiota MOSI- ja MISO-linjassa ei voi olla, koska piiri ei ymmärrä, jos CS-signaali on yksi. Kytken jälkeen tehtiin päätiedostoon RAM-applikaatio-ohjelma, joka lähettää MFD-piirikortille RAM:sta käskyn SPI:n käyttöönottoon. Ohjelmasta ei saatu tarvittavaa varmistusta siitä, että bitit liikkuvat oikein SPI:n välillä, yhteys olisi muodostettu ja oikeaa raakadataa saadaan enkooderilla. Ongelmaa lähdettiin purkamaan kytkennästä, kuvasta 4 nähdään oskilloskoopin näkymä, josta todettiin, että data liikkuu ja CS-signaali menee oikealla hetkellä alas. CS:n mennessä alas, alkaa MFD lähettämään rekisteripyyntöä enkooderille MOSI-linjalla ja enkooderi vastaa MISO-linjasta rekisterin arvon takaisin heksadesimaalilukuna sen jälkeen, kun CS-signaali on käynyt ylhäällä, odottanut 600 ns ja mennyt takaisin alas. Ohjelmaan lisättiin tulostus terminaaliin, todentaaksemme datan oikean paikkansa pitävyyden myös prosessorin päässä.

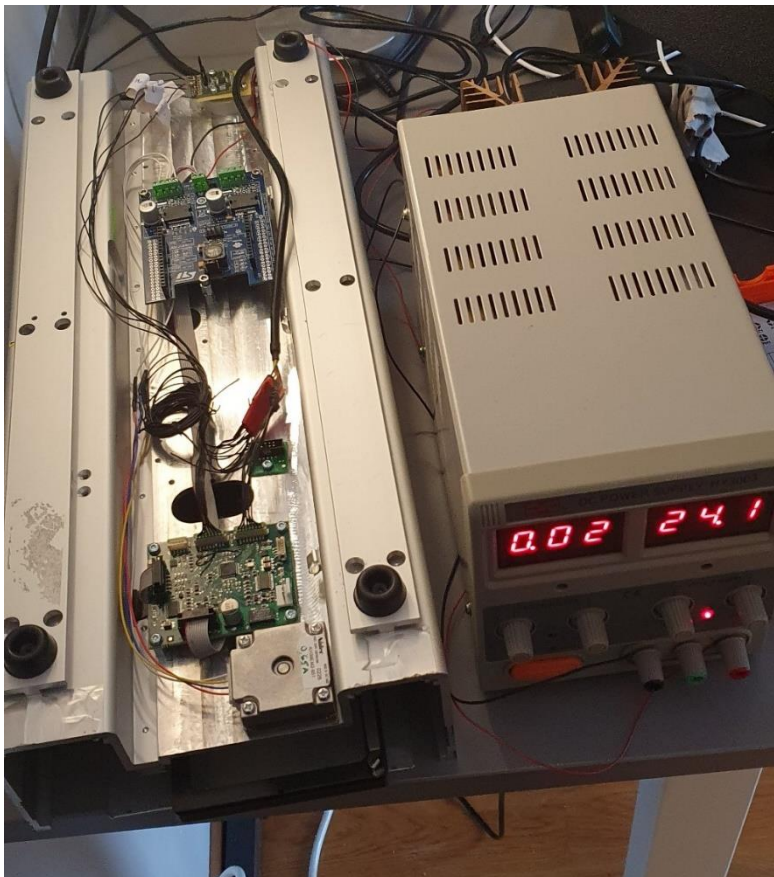


Kuva 4. Datan havainnointi oskilloskoopilla.

Ohjelmaa aloitettiin kirjoittamaan pinnien alustuksella, niitä kytkemällä päälle ja pois saadaan signaali näkymään oskilloskoopissa, näin saatiin rajattua pois kytkennästä johtuvat ongelmat ja koodissa tehty alustus todennettu toimivaksi.

Pinnien tilaa voidaan käyttää manuaalisesti tai SPI:llä ohjattavaksi, kun tila on manuaalinen, saadaan käskyllä muutettua pinnin tilaa nolaksi ja ykköseksi. Tilan ollessa SPI:llä ohjattava, pystyy prosessori lähettämään SPI-rekisterissä olevia komentoja pinniin ja tekemään automaattisesti koodissa halutun käskyn. Esimerkiksi heksadesimaalin lähetyksen bitteinä.

Kuvasta 5 nähdään enkooderin oikeanlainen kytkentä. Enkooderi piirilevyn alla on pieni kolo magneetille, se sijaitsee kuvan yläpäässä keltaisella alustalla, piiri lukee magneetin asentoa ja rekisteri palauttaa kulman arvon. Keltaisessa levyssä on magneetin kokoinen reikä, joka tuo magneetin lähemmäs Hall sensoria parhaan kulmaposition seuraamiseksi.



Kuva 5. Enkooderin kytkentä.

KytKentä ja alustus todettiin toimivaksi ja seuraava vaihe on saada lähetettyä MOSI- ja MISO-kaistalta dataa. Varmistettiin oskilloskoopilla prosessorin kellon lähetys. Prosessorin kello pitää tehdä kaksi kertaa kahdeksan kierrosta ensimmäisellä ja toisella kerralla CS-bitin mennessä nolaksi, koska lähetetään neljä tavua kahdessa osassa kahdeksan bittiä kerrallaan ja MSB tavu ensin, joka sisältää tarkistusbitin (engl. parity bit) ja lähetysbitin. Oskilloskoopissa on ominaisuus digitaalidatan lukemiseen ja datan muuntamiseen heksadesimaaliluvuksi. Ohjelmoitiin MFD lähettämään tiedettyä dataa, jotta oskilloskoopissa näkyisi siirto ja saadaan varmuus MOSI-linjan toiminnasta. Lähetettiin 0xFF kaksi kertaa eli 0xFFFF rekisterin kysely manuaalisesti. MISO-linja palautti epämääräisiä lukuja, joka johtui SPI-linjalle jäädystä datasta, se sekoittui MISO:sta tulleeseen uuteen dataan. Ohjelmaan lisättyä SPI vapautus funktio lähetyksen ja lukemisen väliin, saatiin MISO:n data nolattua joka kerta kyselyn ollessa valmis ja lähetyksen alkaessa. Epämääräinen data rajoittuu tarkistusbitin laskemiseen, joka lasketaan tavussa olevista ykkösien määrästä. Enkooderi AS5147 käyttää päinvastaista järjestelmää tarkistusbitin laskemiseen. Ykkösien määrä ollessa parillinen, tarkistusbitti on 0 ja parittomassa määrässä tarkistusbitti on 1.

Magneetin kulman rekisteriosoite on 0x3FFF, joka on enkooderissa tärkein rekisteri. Muokataan oikeaan muotoon lisäämällä lähetysbitti eli toiseksi ylin bitti ykköseksi. Rekisteri muuttuu luvuksi 0x7FFF, ennen rekisterin lähetystä SPI-väylään ja sitä mukaan enkooderille, pitää lisätä tarkistusbitti. Luku 0x7FFF muutetaan biteiksi, saadaan 0111 1111 1111 1111, ykkösten lukumäärä on pariton ja tarkistusbitti muutetaan 1:ksi ennen lähetystä. Rekisteriosoite muunnosten jälkeen on 0xFFFF, se lähetetään MOSI-linjalla enkooderille. Enkooderi käsittelee kutsun ja lähettää kyseisestä rekisteristä raakaluvun takaisin. Esimerkiksi kulmaluku 0x010E lasketaan asteiksi seuraavasti:

$$\circ = \frac{hex \times 360}{16384} \quad (5)$$

Sijoitettuna saatu raakaluku kaavaan *hex* on magneetin kulma 270°. Lasku voidaan laskea ohjelmassa valmiiksi tai palauttaa ainoastaan raakaluku, jota

käytetään moottorihjaimessa moottorin paikan siirtelyyn. Jokainen rekisterifunktio enkooderissa käyttää kuvassa 6 toteutettua SPI:n lähetys-, luku- ja virheentarkistuskehystä, kuvasta nähdään miten nämä ovat toteutettu. Funktio ottaa yhtenä argumenttina rekisterin arvon, joka halutaan lähettää SPI:llä. Tiedetään, että rekisteriä luetaan ja ensimmäinen vaihe on lisätä viidenteentoista bittiin lähetysbitti ykköseksi. Lukemisbitin jälkeen tiedetään kaikkien bittien olevan kohdallaan ja tarkistusbitti voidaan laskea toisella funktiolla. Tarkistusbitin funktio siirtää ja poistaa ykkösiä niin kauan, että bittien laskeminen onnistuu ja vain tarkistus bitti jää funktion palautukseen. Tarkistusbitti palauttaa siis vain tosi tai epätosi, ykkönen tai nolla. Yhden numeron palautus varmistetaan funktion tyypillä, jotta ei saada turhia bittejä sekoittamaan funktiota ja bittien siirtämistä. 16-bittinen rekisteri täytyy naamioida (engl. masking) 8-bittiseksi luvuksi, koska piiri pystyy käsittelemään vain 8-bittiä kerrallaan. Luvun suurimmat 8 bittiä siirretään kahdeksan bittiä oikealle ja AND-portilla (engl. AND-Gate) naamioidaan turhat nollat pois. Muuttujaan SPI_TDR nimeksi annettu rekisteri ja OR-portilla saadaan CS-signaali menemään alas ja kommunikointi aloitettua. Lähetysten ja lukemisen jälkeen tehdään kahdesta 8-bittisestä numerosta 16-bittinen numero, joka saadaan näkymään terminaalissa ja funktio saa palautettua luvun. Lopussa on tarkistus, joka tarkistaa, ettei data muutu missään vaiheessa.

```

static uint16_t spi_send_read (uint16_t reg)
{
    reg = reg | (1<<14); // reading bit 14 is set to 1
    bool_t parity = __check_parity(reg); // Calculate parity bit
    reg = reg | (parity<<15); // Put parity bit to its place
    // Send register to encoder with SPI MSB byte first
    uint8_t split = (((reg) >> 8) & 0xff);
    spi_wait_tdre_en();
    SPI_TDR = split | (0/*Chipselect down*/ << SPI_TDR_PCS);
    spi_wait_tdre_en();
    split = (reg & 0xff);
    SPI_TDR = split | ((0/*Chipselect down*/ << SPI_TDR_PCS) | (1/*Chipselect up*/ << SPI_TDR_LASTXFER));
    spi_wait_tdre_en();

    // Read from SPI
    spi_clear_rx_en();
    SPI_TDR = 0x00 | (0/*Chipselect down*/ << SPI_TDR_PCS);
    spi_wait_rdrf_en();
    uint8_t byte1 = SPI_RDR; // Read and save hex byte to variable
    spi_wait_tdre_en();
    SPI_TDR = 0x00 | (0/*Chipselect down*/ << SPI_TDR_PCS);
    spi_wait_rdrf_en();
    uint8_t byte2 = SPI_RDR; // Read and save second hex byte to variable
    SPI_CR = (1/*Chipselect up*/ << SPI_CR_LASTXFER);

    uint16_t append = (byte1 << 8) | (byte2); // Make 16 bit hex from two bytes
    bool_t chk_append = __check_parity(append); // Check parity from appended 16 bit hex
    uint16_t raw_w_parity = append | (chk_append << 15);

    /* Check bit 15 PARD Parity bit (even) calculated on the lower 15 bits of the read data frame
    Check bit 14 EF0: No command frame error command occurred, 1: Error occurred */
    if(((append & 0x8000) >> 15) == chk_append && ((append & 0x4000) >> 14)){
        append = (((byte1 & 0x1f)<<8) | byte2);
        stream_push_cstr("Zero position value is ");
        stream_push_hex16_cr(append);
    }

    return raw_w_parity; // Return only raw hex value from the register
}

```

Kuva 6. Funktion SPI:n lähetys-, luku- ja virheentarkistuskehystä.

Moottoriohjaimen L6470 käyttöönnotossa oli kytkennästä johtuvia ongelmia. Rekisterin kyselyn ja virherekisterin palautuksessa huomattiin ongelma oskilloskoopin kanssa ja kytkentä tarkistettiin. Tarkastuksessa ja datalehden lukemisesta kävi ilmi huolimattomuus kytkentää tehdessä. X-NUCLEO-IHM02A1 piirilevyn hyppylangat (engl. jumper) olivat kytketty väärin, ulkoisessa levyn käytössä tarvitaan J1 pinnien 1–2 ja 3–4 väliin hyppylangat, jotka liittävät sisäisen jännitelähteen yhteen piirilevyn virran saantiin.

Oskilloskoopilla MOSI-linjaa tutkimalla saatiin varmuus oikean rekisterin kyselyyn, jota ennen piti tehdä aina kevyt alustus (engl. soft reset). Kun MISO-linjalta saadaan nollaa, tarkoittaa se sitä, että piirillä ei ole kyseisessä rekisterissä lukua palautettavaksi, tämä on niin sanottu oletustila. Tilarekisteriä (engl. status register) kysymällä piiriltä saadaan 16-bittinen heksadesimaaliluku maski, jonka muuttaminen biteiksi paljastaa piirin tilan, tilarekisteri ei voi olla nolla. Kytkennän tarkistamisen ja demo-ohjelman luomisen jälkeen alkoi data virrata linjoissa,

ongelmana vastaan tuli viive. Viiveen ollessa liian lyhyt, vuotaa tilarekisteristä saatu viimeinen bitti seuraavaan CS-signaalin alas menon ei-operaatio (engl. no-operation) eli nolla kellotukseen, jossa ei pitäisi olla minkäänlaista dataa. AVR32:ssa on rekisteri viiveiden säätämiseen, lisäämällä haluttu luku rekisteriin voidaan lisätä viivettä CS-signaalin alas menon ja ensimmäisen kellosignaalin väliin.

Moottoriohjain L6470 on X-NUCLEO-IHM02A1 piirilevyssä kytketty sarjaan, se mahdollistaa yhden CS-johdon käytön kytkennässä, mutta tämä pitää huomioida ohjelmassa. Sarjassa toiselta piiriltä kysyttävä tila saadaan lähettämällä ensin rekisterin arvo tavu ja heti perään ei-operaatio komento, joka on nolla tavu. Kahden tavun verran CS-signaali on nolla ja lähetyksen ollessa valmis, nostetaan CS-signaali ykköseksi. Ei-operaatioiden määrä riippuu kytkentään kytketyistä piirien määrästä. Piiri tietää CS-signaalin mennessä ylös aina tavun jälkeen, että komento loppuu ja alkaa käsittely. Piiri tarvitsee aikaa minimissään 800 ns käsitelläkseen komento, aika annetaan piirille CS-signaalin ollessa ylhäällä. MISO-linjan lukeminen toteutetaan kellottaen kaksi kertaa kaksi tavua nollaa, jonka välissä CS-signaalin bitti laitetaan ykköseksi ja piiri palauttaa MISO-linjaan piirin tilan tavu kerrallaan. Jos lähettää samalla uutta rekisterikomentoa MOSI-linjassa, kun piiri palauttaa arvoa, menee tämä sekaisin ja data on virheellistä. Piiri pystyy käsittelemään yhden komennon kerrallaan, jos MOSI-linjasta on tulossa komento rekisterin kutsumiseen MISO-linjaa ei pystytä käyttämään käsittelyn varauksen vuoksi.

Moottoriohjaimelta kysyttäessä tilarekisteriä, bitit siirtyivät yhdellä vasemmalle ja näin vastaus oli aina väärin. Ongelma testattiin lähettämällä mielivaltaisesti yhden tavun verran 1:stä, 8:aa ja 5:sta, kun bitit tulivat takaisin, ne olivat aina siirtynyt yhdellä. Ratkaisu tähän löytyi SPI-tilasta, piiri aloittaa lukemaan laskevalla reunalla olevan kellosignaalin, mutta se oli konfiguroitu nousevalle reunalle. Kommunikaatiolinjan oikeanlaisen datan varmistumisen johdosta, pystyin aloittamaan siirtämään moottoria siihen tarkoitetun komennon kanssa. Moottorin käynnistämisen yhteydessä parametrit saavat oletusarvot, näitä arvoja ei tarvitse muokata vielä omaan projektiin sopivaksi, mutta voi varmistaa

moottorin toimivuuden. Moottorin liikuttamisen periaate toimii samalla komentokehyksellä kuin tilarekisterin kutsuminen. Lähetetään ensin piirille liikkumisrekisterin arvo heksadesimaali muodossa ja sen perään askeleiden määrä, kuinka paljon halutaan moottoria liikuttaa.

5 Lopuksi

Opinnäytetyön tarkoituksena oli kehittää ajuriohjelma magneettisen enkooderin piirille ja moottoriohjaimelle. Ohjelma luo kommunikaation AVR32-prosessorin ja piirien välille. Lisäksi piirit pystytään ohjelmoimaan yksilöllisesti. Magneettisen enkooderin piiri vastaa moottorin liikuttamiseen ja lähettää tietokoneelle paikka dataa. Opinnäytetyö suoritettiin suunnitellusti. Lopputulos oli haluttu ja vastasi osittain suunnitelmaa. Magneettisen enkooderin datan käsittelyä ja moottorin tarvittavaa paikkakorjausta ei ehditty tekemään. Työn suunnittelu ja ohjelmiston kehitys antaa jatkumahdollisuuden saattaa oikeaksi tuotteeksi monileimalukijan levykelkan rataan.

Moottoriohjaimen ja enkooderin yhdessä toimivuus on suuri harppaus teknologialle, jossa enkooderin käyttöä ei ole vielä olemassa. Varmistaminen fyysisessä maailmassa olevan moottorin paikan ja sen vertaaminen laskennalliseen enkooderin paikkaan, saadaan luotettava käsitys moottorin sijainnista ja tarvittavista korjauksista. Enkooderi mahdollistaa myös lokiin kirjoittamisen, jotta voidaan varmistua, millä tavalla moottori on liikkunut ja pitääkö data paikkansa.

Enkooderin ja moottoriohjaimen toimivuus on varmistettu ja peruskäskyt ovat tehty, lähetys- ja lukukehykset, käyttökomennot ja piirien konfigurointi tehty. Tulevaisuudessa työ jatkuu näiden kahden piirin liittämällä ja kommunikoinnilla. Enkooderi kuuntelee moottoria ja laskee magneetin kulmapositiota, pitää kirjaa siitä, kuinka paljon moottori on liikkunut ja tekee laskennallisesti oikean sijainnin moottorin koordinaateille. Ohjelmaan lisätään abstraktiotasoa, jotka sisältävät muun muassa radan maksimikoordinaatit ja kartan, jonka mukaan levykelkka liikkuu.

Lähteet

STMicroelectronics 2015. The manual: L6470 Fully integrated microstepping motor driver with motion engine and SPI.

ComputerScience.GCSE.GURU n.d. Von Neumann Architecture. Viitattu 9.11.2021 <https://www.computerscience.gcse.guru/theory/von-neumann-architecture>

Mughees, N. 2021. Basic methods for encoderless control of electric motor drivers. Electronics 360. Viitattu 11.4.2022. <https://electronics360.globalspec.com/article/16880/basic-methods-for-encoderless-control-of-electric-motor-drives>

Lewotsky, K. 2014. Tutorial: The basics of stepper motors. Association of advanced automation. Viitattu 21.03.2022 <https://www.automate.org/industry-insights/tutorial-the-basics-of-stepper-motors-part-i>

Sawicz, D. 2012. Hobby servo fundamentals. Princeton. Viitattu 22.03.2022 <http://www.princeton.edu/~mae412/TEXT/NTRAK2002/292-302.pdf>

Collins, D. 2018. Stepper motors: Differences between permanent magnet, variable reluctance and hybrid types. Linear motion tips. Viitattu 14.03.2022 <https://www.linearmotiontips.com/stepper-motors-differences-between-permanent-magnet-variable-reluctance-and-hybrid-types/>

Ganssle, J. 2012. A Designer's Guide to Encoders. Digikey Electronics. Viitattu 24.2.2022 <https://www.digikey.fi/en/articles/a-designers-guide-to-encoders>

Procházka, P. & Vaněk, F. 2012, 3.4 Hall effect (HE) sensors. ScienceDirect. Viitattu 24.02.2022 <https://www.sciencedirect.com/topics/chemistry/hall-effect>

STMicroelectronics n.d. Product overview/Description. Viitattu 24.2.2022 <https://www.st.com/en/motor-drivers/l6470.html>

RS, G. 2015. Stepper motors and drives, what is full step, half step and microstepping. Design spark. Viitattu 28.2.2022 <https://www.rs-online.com/designspark/stepper-motors-and-drives-what-is-full-step-half-step-and-microstepping>

Ams Osram-AG 2021, The manual: AS5147 14-bit on-axis magnetic rotary position sensor with 11-bit binary incremental pulse count.

Einspieler, R. 2014. Accurate contactless position sensors improve performance, reduce cost of electric motors. Fierce electronics. Viitattu 5.4.2022 <https://www.fierceelectronics.com/components/accurate-contactless-position-sensors-improve-performance-reduce-cost-electric-motors>

Anderson, M. 2019. Absolute vs incremental encoders. Real pars. Viitattu 11.3.2022 <https://realpars.com/absolute-vs-incremental-encoder/>

Ato 2017. How to prevent stepper motor losing steps. Viitattu 8.11.2021 <https://www.ato.com/how-to-prevent-stepper-motor-losing-step>

Oriental motor U.S.A corp n.d. Basics of stepper motor. Viitattu 9.11.2021 <https://www.orientalmotor.com/stepper-motors/technology/stepper-motor-basics.html>