

Mikko Laine

Antennimaston suuntausjärjestelmä

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Automaatiotekniikka

Insinöörityö

30.3.2018

Tekijä Otsikko	Mikko Laine Antennimaston suuntausjärjestelmä
Sivumäärä Aika	30 sivua + 2 liitettä 30.3.2018
Tutkinto	insinööri (AMK)
Tutkinto-ohjelma	Sähkö- ja automaatiotekniikka
Ammatillinen pääaine	Automaatiotekniikka
Ohjaajat	Pauli Gäddala, toimitusjohtaja Timo Tuominen, lehtori
<p>Opinnäytetyö tehtiin GadSat-nimiselle sähkö- ja teleurakoitsijalle Tervakoskella. Työn tavoitteena oli suunnitella ja toteuttaa antennimaston suuntimiseen ohjausjärjestelmä Siemensin TIA Portalilla Siemensin G120C-sarjan taajuusmuuttajalle, Profibus-kenttäväylän avulla. Elektronisen kompassin kanssa käytettiin Arduino-mikro-ohjainta.</p> <p>Opinnäytetyön kirjallisen osan alkupuoli esittelee antennimastoa, johon työ liittyy, sekä käsittelee laitteita ja ohjelmia, joita työssä käytettiin.</p> <p>Työn loppupuoli keskittyy käsittelemään asioita, joita työn ensimmäisessä puoliskossa esitellyillä työkaluilla tehtiin.</p> <p>Lopputuloksena saavutettiin tavoiteltu ja toimiva järjestelmä käyttöliittymineen antennimaston suuntausta varten.</p>	
Avainsanat	TIA Portal, s7-1200, Siemens G120C, Profibus, Arduino, automaatio

Author Title	Mikko Laine Guidance System for Antenna Tower
Number of Pages Date	30 pages + 2 appendices 30 March 2018
Degree	Bachelor of Engineering
Degree Programme	Electrical and Automation Engineering
Professional Major	Automation Technology
Instructors	Pauli Gäddala, CEO Timo Tuominen, Senior Lecturer
<p>The study was carried out for the electrical and telecommunications contractor GadSat in Tervakoski. The aim of the study was to design and implement the antenna guidance control system on Siemens's TIA Portal to the Siemens G120C Series frequency inverter using the Profibus fieldbus. An electronic compass was used with the Arduino microcontroller.</p> <p>The first part of the thesis presents the antenna tower to which the work relates. It also deals with the equipment and programs that were used in the work.</p> <p>The end part of the thesis focuses on dealing with the tools presented in the first part of the thesis.</p> <p>Result of this study is the fully functioning control system and graphical user interface for the antenna tower.</p>	
Keywords	TIA Portal, s7-1200, Siemens G120C, Profibus, Arduino, automation

Sisällys

Lyhenteet

1	Johdanto	1
2	Antennimasto	1
3	Käytetyt laitteet ja ohjelmat	3
3.1	Siemens 1212C AC/DC/RLY	3
3.2	Siemens G120C	4
3.3	Beijer Electronics T4A	5
3.4	Arduino	5
3.5	Siemens Ethernet-kytkin	8
3.6	KCI moottori	8
3.7	Siemensin TIA Portal	9
3.8	iX Developer SP1	10
3.9	Arduino IDE	11
4	Tiedonsiirto ja väylät	11
4.1	Profibus	12
4.2	Ethernet	13
4.3	I2C-väylä	14
4.4	SPI-väylä	14
5	Kokoonpano ja testaus	14
6	Ohjelmointi	16
6.1	Taajuusmuuttaja	16
6.1.1	Taajuusmuuttajan parametrisointi	16
6.1.2	Taajuusmuuttajan ohjaus	17
6.2	Käyttöliittymä	21
6.3	Elektroninen kompassi	26
6.3.1	Arduinon ohjelmointi	28
7	Yhteenveto	29

Liitteet

Liite 1. Arduinon ohjelmakoodi

Lyhenteet

HMI Käyttöliittymä. Human-machine interface.

I2C Inter-Integrated Circuit. Yksinkertainen kaksisuuntainen tiedonsiirtoväylä.

PLC Ohjelmoitava logiikka, jota käytetään automaatioprosessien ohjauksessa.

Profibus Avoin kenttäväyläjärjestelmä.

SPI Serial Peripheral Interface bus. Sarjamuotoinen oheislaiteväylä.

Telegram Sähköinen sanoma.

1 Johdanto

Opinnäytetyö tehtiin yritykselle nimeltä GadSat, joka on tele- ja sähköasennusurakoitsija Tervakoskelta. Työn tarkoituksena laatia antennimaston suuntimiseen automaattinen ohjausjärjestelmä, jonka avulla voidaan mastoa kääntää haluttuun suuntaan. Vuodenajasta johtuen ei antennimastoa voitu vielä pystyttää, joten työssä keskityttiin maston koneiston suunnitteluun, rakentamiseen ja testaamiseen.

Tässä työssä toteutettiin moottorin ohjaus Siemens G120C -sarjan taajuusmuuttajalla. Taajuusmuuttajaa ohjataan Siemensin 1200-sarjan logiikalla, joka on yhteydessä taajuusmuuttajaan Profibus-väylän kautta. Taajuusmuuttaja toimii väylässä orjana, logiikka taas vastaavasti isäntänä.

Elektroninen kompassi kytkettiin Arduino-mikro-ohjaimen IO-liitäntään, josta sen data koodattiin logiikalle sopivaan muotoon. Logiikan sekä Arduinon välillä kommunikoinnissa käytettiin TCP/IP Ethernet -protokollaa.

Antennimaston mielekästä ohjausta varten tarvittiin käyttöliittymä. Tämä toteutettiin yhdistämällä Beijer Electronicsin T4A-kosketusnäyttöpaneeli Ethernet-liitännällä logiikkaan. Käyttöliittymä suunniteltiin Beijer Electronicsin iX Developer -suunnittelutyökalulla.

Logiikan ohjelma sekä taajuusmuuttajan ohjaus suunniteltiin Siemensin TIA Portalilla. TIA Portal on Siemensin integroitu automaatio suunnittelutyökalu. Arduinon konfigurointiin käytettiin Arduino IDE -ohjelmaa.

2 Antennimasto

Antennimasto on suunniteltu rakennettavaksi pellonreunalle, josta on pitkälti laakeata maastoa hyvän kuuluvuuden saavuttamiseksi. Mastoon on suunniteltu asennettavaksi radio- sekä teleantenneja. Masto rakentuu kahdeksasta viiden metrin pätkästä, joista koostuu yhteensä 40 metriä korkea masto.

Tällä hetkellä mastoa varten on kaivettu neljä kuoppaa, johon valetaan kevään aikana betonia ja kiinnitetään harukset sekä maston alusta. Talven pakkasista sekä syksyn saateista johtuen ei betonia voitu valaa ennen tulevaa kevättä.



Kuva 1. Maston runkopalkit.

Kuvassa 1. näkyvät antennimaston runkopalkit, joiden parreleveys on 0,5 m ja pituus 5 m. Painoa palkeilla on noin 2000 kiloa.



Kuva 2. Maston kääntömekanismin osia.

Kuvassa 2. on muutamia maston osia, kuten vetoratas, joka yhdistetään ketjulla moottoriin, sekä maston jalustaan asennettavat laakeroidut ylä- ja alaohjaimet sekä akseli.

3 Käytetyt laitteet ja ohjelmat

3.1 Siemens 1212C AC/DC/RLY



Kuva 3. Siemensin logiikka.

Työssä käytettiin Siemensin 1200-sarjaan kuuluvaa 1212C AC/DC/RLY CPU -yksikköä. Yksikössä on integroituna kahdeksan digitaalista sisääntuloa, kuusi digitaalista ulostuloa ja kaksi analogista sisääntuloa. CPU-yksikössä on yksi Ethernet-portti. Laitteessa on myös integroitu 24 V:n DC-jännitelähde.

Työssä käytettyyn logiikkaan liitettiin Siemensin CM 1243-5 Profibus master -liitännän tarjoava lisämoduli, kuvassa 3, joka kykenee 12 MBps:n siirtonopeuteen. Profibusin avulla saadaan yhteys taajuusmuuttajaan.

3.2 Siemens G120C



Kuva 4. Siemensin taajuusmuuttaja.

Kuvassa 4. on työssä käytetty taajuusmuuttaja, joka on Siemensin valmistama G120C DP. Taajuusmuuttajalla pystyy ohjaamaan moottoreita teholtaan 4 kilowattiin asti. Kotelointiluokka on IP 21 ja malli on suunniteltu kotelon pohjaan asennettavaksi. Ohjelmointi

tapahtuu USB-portin kautta tietokoneella tai Profibus-väylään kytketyn logiikan kautta. Myös ohjelmointi erikseen ostettavan näyttöpaneelin avulla onnistuu. Taajuusmuuttajassa on myös integroitu Profibus-liitäntä, jolla saadaan yhteys ohjaavaan logiikkaan.

3.3 Beijer Electronics T4A



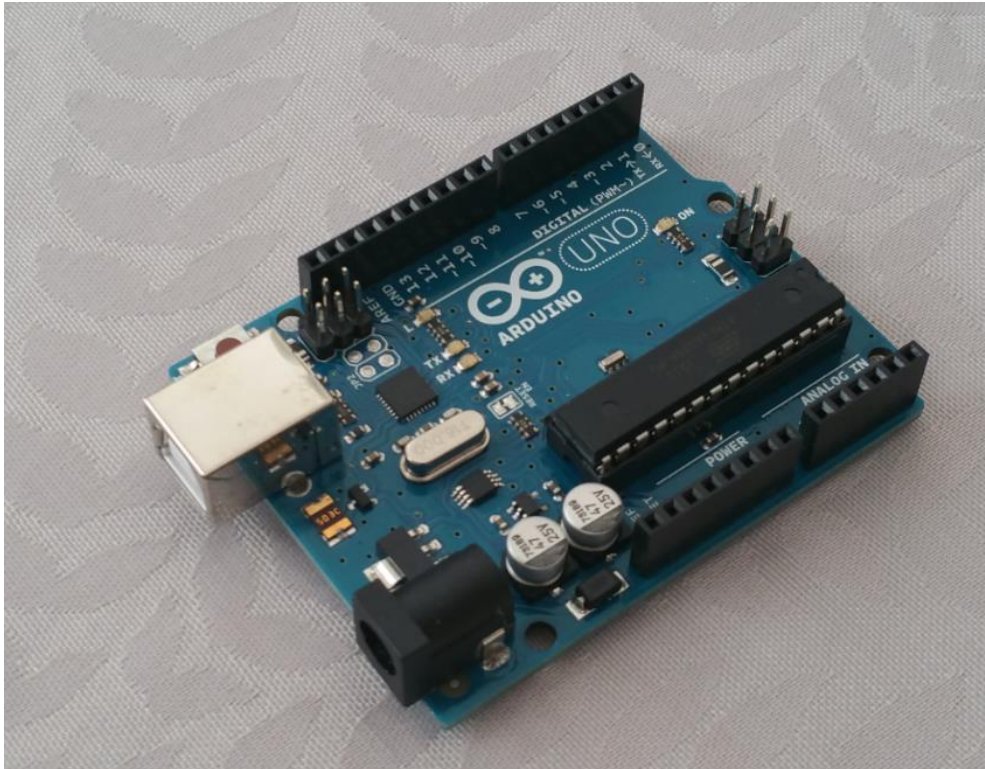
Kuva 5. Beijer Electronicsin kosketuspaneeli.

Työssä käytettiin Beijer Electronicsin valmistamaa T4A-kosketuspaneelia (kuva 5.), joka on 4,3 tuumainen kosketusnäyttö, joka toimii 480 x 272 -kuvapisteen tarkkuudella. Laite on edestä IP65-suojattu ja takaa IP20. Laitteessa toimii taustalla Windows Embedded Compact -käyttöjärjestelmä, joka on kuitenkin piilotettu käyttäjältä. Ohjelmointi ja kommunikointi tapahtuu Ethernetin kautta.

3.4 Arduino

Arduino on avoimeen laitteistoon perustuva mikro-ohjainalusta sekä ohjelmointiympäristö. Laitteisto pohjautuu 32-bittiseen Atmel AVR -mikro-ohjaimen, jonka pinneihin voi kytkeä esimerkiksi antureita, moottoreita tai LED-valoja. Arduinoa ohjelmoidaan

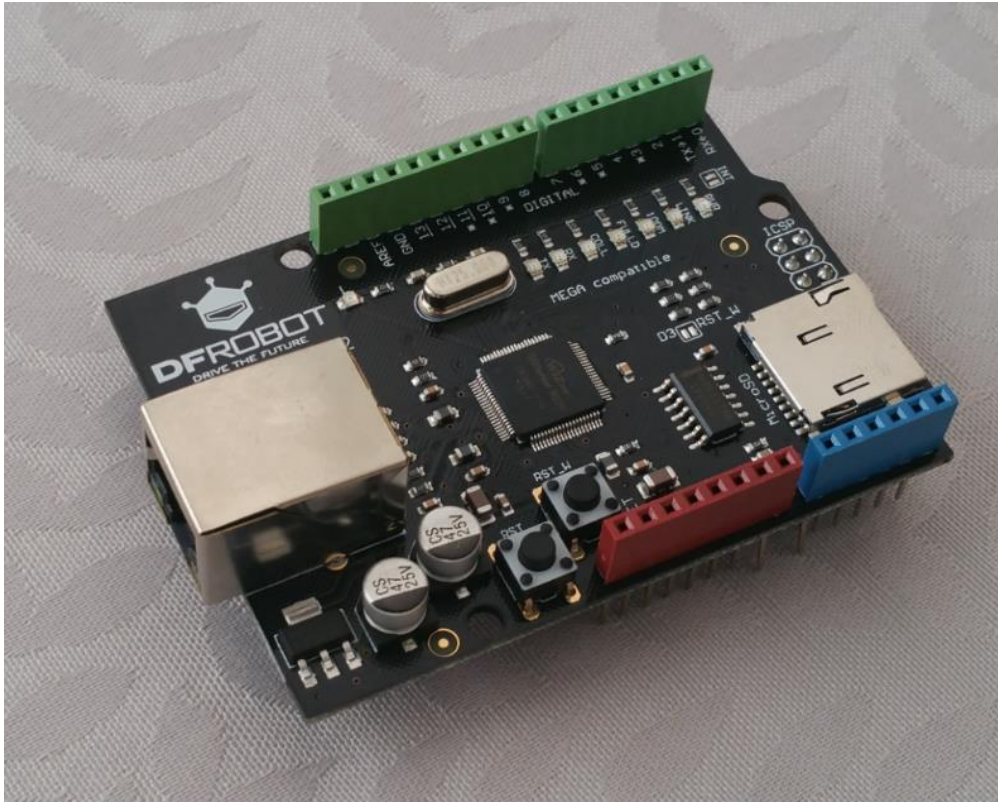
C++:aan perustuvalla Arduino-ohjelmointikielellä. Arduinoja on eri malleja, jotka eroavat toisistaan esimerkiksi keskusmuistin ja Flash-muistin sekä digitaalisten ja analogisten pinnien määrissä. Alkuperäisissä laitteissa ohjelmointi tehtiin sarjaportin kautta. Nykyisin käytävissä on usein USB tai Bluetooth. Ohjelmointiin käytetään Arduino IDE -ohjelmaa, joka on saatavilla yleisimmille käyttöjärjestelmille. Arduinolle on saatavissa myös lisälaitteita, joita kutsutaan shieldeiksi, jotka liitetään suoritinkortin päälle. Lisälaitteiden avulla Arduinossa voi käyttää mm. Ethernetiä, sensorikortteja ja kosketusnäyttöjä.



Kuva 6. Arduino UNO.

Kuvassa 6. on työssä käytetty Arduino Uno -mallinen mikro-ohjain, johon liitettiin Ethernet-lisämoduuli (kuva 7.) logiikan kanssa kommunikointia varten.

Arduino Ethernet shieldin avulla voi helposti liittää Arduinon internetiin. Tämä moduuli mahdollistaa Arduinon lähettää ja vastaanottaa tietoja kaikkialta maailmasta, internetyhteyden avulla. Ethernet shield perustuu W51000-piiriin, jossa on sisäinen 16K-puskuri sekä 10/100Mb-yhteysnopeus. Ethernet shield sisältää myös micro SD -korttipaikan, joka mahdollistaa suurempien datamäärien tallentamisen.



Kuva 7. Arduino Ethernet shield.

3.5 Siemens Ethernet-kytkin

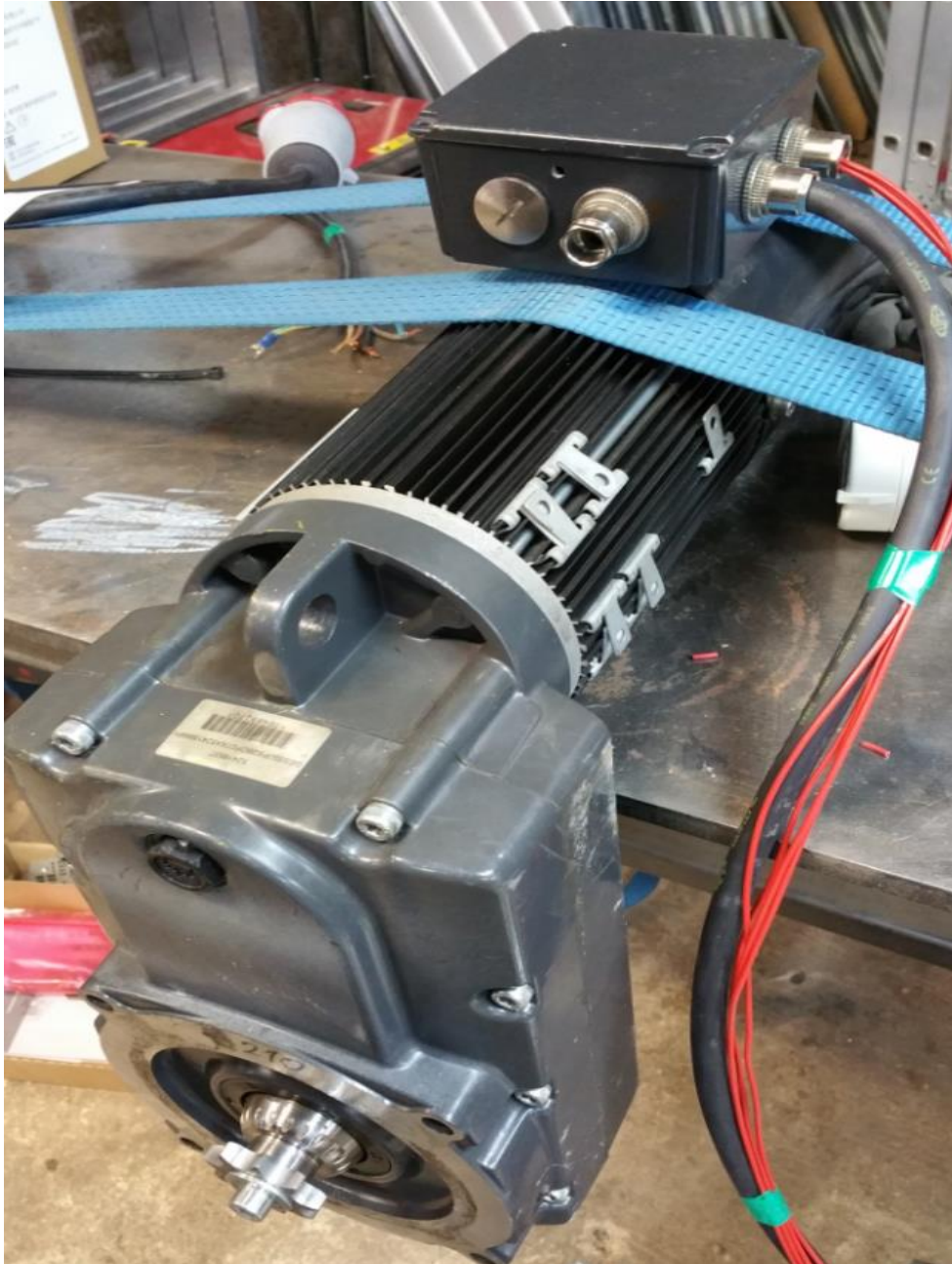


Kuva 8. Siemensin kytkin

Kuvassa 8. näkyvä Siemensin valmistama Ethernet-kytkin on tässä työssä hyödyllinen, koska logiikassa on vain yksi integroitu Ethernet-portti ja tarve on saada useampi laite samaan verkkoon.

3.6 KCI moottori

Moottorina toimii Konecranesin siltanosturin moottori, johon kuuluu vaihteisto, jonka avulla pyörimisnopeus saatiin riittävän alhaiseksi, sillä haluttu nopeus maston suuntaukseen oli yksi kierros per minuutti, ettei mastoon kohdistuisi äkkinäisiä liikkeitä.



Kuva 9. Siltanosturin moottori.

Kuvassa 9. moottori on kiinni testipöydässä, jossa sitä voitiin käyttää turvallisesti. Moottori painaa vaihteistoineen noin 50 kiloa.

3.7 Siemensin TIA Portal

Siemensin TIA (Totally Integrated Automation) Portal -ohjelmointityökalu nopeuttaa, helpottaa ja tehostaa suunnittelua. Kilpailukyky paranee, koska samaan työkaluun on

yhdistetty logiikkojen, käyttöliittymien ja turvaratkaisujen sekä taajuusmuuttajien ohjelmointi. TIA-portaalin käyttöliittymä on helposti omaksuttava ja vuorovaikutteinen. Tämä mahdollistaa automaation konfiguroinnin, diagnostiikan sekä ylläpidon helposti yhdellä ohjelmalla. TIA Portal yhdistää logiikkaohjelmoinnin (SIMATIC STEP 7) ja käyttöliittymäsuunnittelun (SIMATIC WinCC) sekä uusimpana myös taajuusmuuttajat (SINAMICS StartDrive).

Kolme ominaisuutta tekee portaalista ainutlaatuisen: käyttäjäystävällisyys, tehokkuus ja luotettavuus. TIA Portalilla on mahdollista tehdä suunnittelu- ja tuotantoprosessit koko tuotantoketjulle. Tämä varmistaa tehokkaamman työskentelyn sekä paremman tuottavuuden ja kehittyneen kilpailukyvyn. [1.]

Työssä käytetty versio TIA Portalista oli v13 SP1.

SIMATIC STEP 7 on osa Totally Integrated Automation (TIA) Portal -ohjelmistokehystä. Keskeisenä suunnitteluohjelmistona se on perustana kaikkien SIMATIC-ohjelmoitavien logiikoiden ohjelmoinnissa. STEP 7 on tärkeä osa suunnittelua sen kaikissa vaiheissa.

Yksi integroitu suunnitteluohjelmisto kaikille SIMATIC-logiikoille säästää aikaa. Uudistetut työkalut parantavat merkittävästi helppokäyttöisyyttä. Valokuvanomainen HW-konfiguraatio selkeyttää järjestelmän rakennetta. Älykäs drag/drop-toiminnallisuus STEP 7- ja WinCC-editoreiden välillä nopeuttaa suunnittelua ja vähentää virheitä. Olemassa olevien sovellusten hyödyntäminen merkitsee investointivarmuutta. [2.]

Työssä käytetty versio oli STEP 7 Professional v13 SP1.

3.8 iX Developer SP1

iX Developer on Beijer Electronicsin ohjelmisto käyttöliittymien tekemiseen. Ohjelmisto tukee yleisimpiä automaatiovalmistajien logiikoita.

iX HMI -ohjelmisto antaa ainutlaatuiset työkalut kommunikoida automaation kanssa. Se yhdistää huippuluokan vektorigrafiikkaa ja älykkäämpiä toimintoja, jotka tarjoavat intuitiivisen toiminnan paikallisesti tai etäkäytettynä. Lähes rajattomat yhteydet automaation laitteisiin mahdollistaa kattava yhteensopivien kommunikointiajuriin kirjasto.

iX HMI -ohjelmistoista löytyy kaikki olennaiset ominaisuudet ja toiminnot, joita tarvitaan käyttöliittymään, kuten tietojen tallentamisen, hälytyskäsittelyn ja tiedonkeruun. Valmiit sisäänrakennetut toimintokokonaisuudet sivupohjat, vektoripohjaiset symbolit ja kuvagrafiikka voidaan helposti asemoida ja ottaa käyttöliittymään nopeasti ja helposti, mikä tehostaa työskentelyä ja säästää suunnittelun kustannuksia.

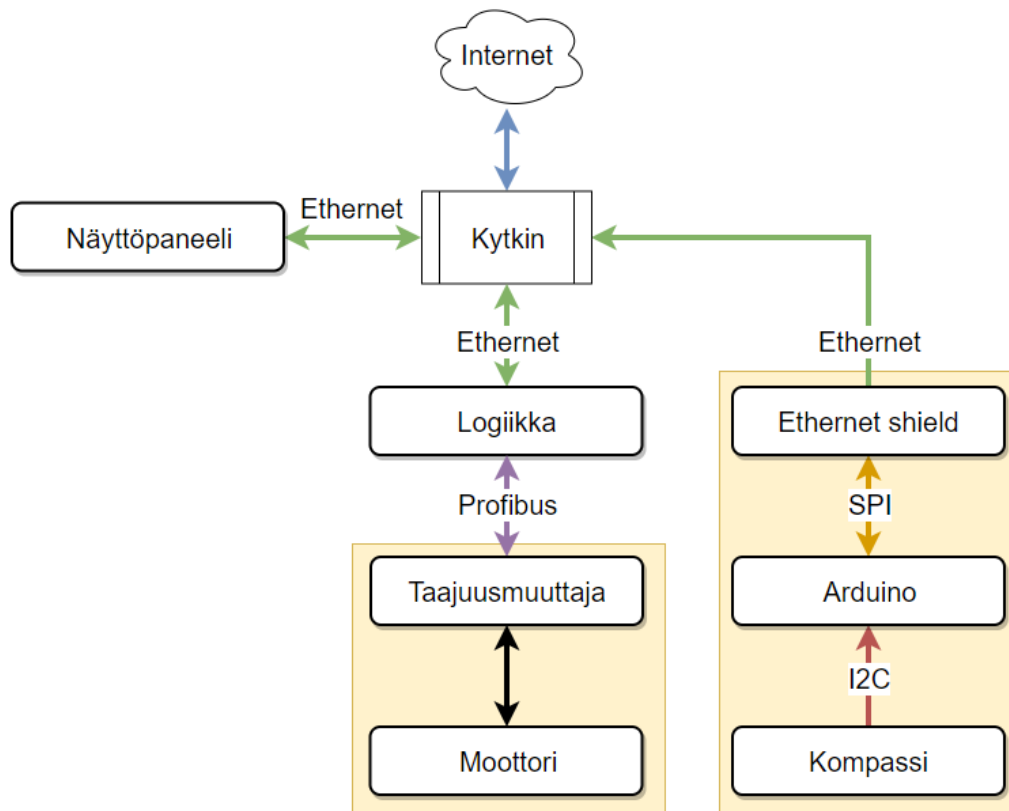
iX HMI -ohjelmistoilla on tuki .NET-tekniikalle, joka tarjoaa vaihtoehtoja kehittyneille kehittäjille suunnitella erikoistunutta toiminnallisuutta. C# -komentosarjat tai .NET-komponentteja ovat kattavasti käytettävissä toiminnallisuuden laajennuksiin. Tiedonvaihtoon on saatavilla FTP, OPC UA ja web--yhteydet. [3.]

3.9 Arduino IDE

Arduino IDE on avoimeen lähdekoodiin perustuva ohjelmisto, jonka avulla on mahdollista kirjoittaa sekä lukea koodia Arduino-piirikortista. Ohjelmisto toimii Windows- Mac OS X- sekä Linux-käyttöjärjestelmillä. Ohjelmointiympäristö on kirjoitettu Javalla, ja se perustuu muihin avoimen lähdekoodin ohjelmiin. Ohjelma toimii kaikkien Arduino-piirikorttien kanssa.

4 Tiedonsiirto ja väylät

Kuvassa 10. näkyy, että työssä ohjelmoitavan logiikan, käyttöpaneelin sekä Arduinon välillä käytettiin Ethernet-verkkoa, taajuusmuuttajan ja ohjelmoitavan logiikan välillä käytettiin Profibus-väylää. Arduinon kytketty Ethernet-moduuli kommunikoi Arduinon kanssa SPI-väylän avulla, kompassin ja Arduinon välillä taas käytetään I2C-väylää.

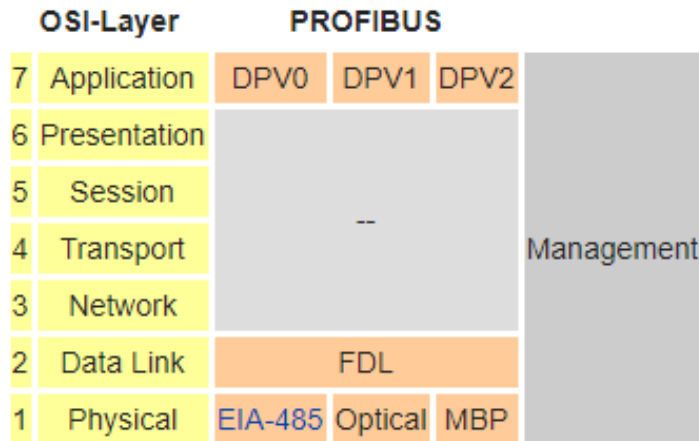


Kuva 10. Insinööriyössä käytetyt laitteet väylineen.

4.1 Profibus

Profibus eli Process Field Bus on avoin toimittajasta riippumaton kenttäväylästandardi automaatioteknologiassa. Profibusista löytyy kaksi eri versiota:

- Profibus DP (Decentralised Peripherals) on luotu hajautettujen kenttälaitteiden, kuten taajuusmuuttajien, liittämiseen automaatiojärjestelmään hyvin nopealla vasteajalla. Yleisin Profibus-versio ja tässäkin työssä käytetty.
- Profibus PA eli (Process Automation) on laajennettu versio Profibus DP:stä. Se tarjoaa turvallisen tiedon- ja tehonsiirron prosessiautomaation kenttälaitteille. [4.]



Kuva 11. Profibusin OSI-malli. [5]

Profibus-väylän rakenne perustuu OSI-malliin (kuva 11.): Se käyttää mallin kerroksia yksi, kaksi ja seitsemän. Ensimmäiselle kerrokselle sijoittuu Profibus-väylän siirto- ja liitinteknologia. Toinen kerros kattaa fieldbus data linkin. Seitsemännelle kerrokselle sijoituvat erilaiset viesteistä huolehtivat sovellukset. Profibus-väylä ei käytä kerroksia 3--6.

4.2 Ethernet

Ethernet on laajin asennettu lähiverkkotekniikka (LAN). Ethernet on TCP/IP-pinoyhteyskerros-protokolla, jossa kuvataan, miten verkkolaitteet voivat muodostaa tietoja lähetettäväksi samaan verkkosegmenttiin muille verkkolaitteille ja miten nämä tiedot voidaan liittää verkkoyhteyteen. Se koskettaa sekä tasoa 1 (fyysistä kerrosta) että tasoa 2 (data-yhteyskerros) OSI-verkkoprotokollamallissa. Ethernet määrittää kaksi lähetysyksikköä, paketin ja kehysen. Kehys ei sisällä vain lähetettävän datan "hyötykuormaa", vaan myös käsittelee tietoja, jotka tunnistavat sekä lähettäjän ja vastaanottajan fyysiset "Media Access Control" (MAC)-osoitteet, VLAN-koodauksen ja palvelun laatuun liittyvät tiedot, sekä virheenkorjaustiedot havaitsemaan ongelmia lähetyksessä. Jokainen kehys kääritään pakettiin, johon kiinnitetään useita tavuja tietoja, joita käytetään yhteyden muodostamisessa ja merkinnöissä, mistä kehys alkaa.

Ethernetin kaistanvarausmenetelmä on CSMA/CD (Carrier Sense Multiple Access with Collision Detection). Se on kilpavarausmenetelmä: jos mikään muu laite ei lähetä (Carrier Sense), kaikilla laitteilla on oikeus aloittaa lähetys (Multiple Access). Jos kaksi tai useampi laite alkaa lähettää samanaikaisesti, ne havaitsevat törmäyksen (Collision

Detection) ja keskeyttävät lähetyksen. Törmäykseen osallistuneet laitteet odottavat satunnaisen ajan, jonka jälkeen ne yrittävät uudelleen lähettämistä.

Eri verkkolaitteet tunnistetaan MAC-tason (Media Access Control) osoitteilla, joka on 48-bittinen yksilöllinen osoite jokaiselle verkkolaitteelle. [6.]

4.3 I2C-väylä

I2C-kommunikointiväylää käytetään monissa elektroniikkalaitteissa, koska se on helppo sisällyttää useimpiin elektroniikkalaitteisiin, jotka tarvitsevat isäntä-orjatyypin tiedonsiirron. Helppous tulee siitä, koska kommunikointiin tarvitaan ainoastaan kaksi johdinta - joita jatkamalla voi järjestelmä tukea jopa 1 008 laitetta kerrallaan.

4.4 SPI-väylä

Serial Peripheral Interface eli SPI-väylä on synkronoitu sarjaväylä, missä tieto liikkuu bitteinä kahden tai useamman laitteen välillä. Synkronoitu väylä tarkoittaa tässä tapauksessa sitä, että bittien ajoitukset ovat synkronoituja, eli väylän toimintaan liittyy olennaisesti kellosignaali, joka "tikittää" lähettävän laitteen tahdissa. SPI-väylä on synkronisuuden lisäksi kaksisuuntainen, eli lähettävä laite yhtä aikaa lukee ja kirjoittaa tietoa väylälle. [7.]

5 Kokoonpano ja testaus

Antennimaston juurelle on aikanaan tulossa oma laitekoppi, joka tulee sisältämään ohjaus- ja kommunikointilaitteita, joten toistaiseksi jouduttiin tyytymään vanerilevyyn, johon koottiin kaikki tässä työssä tarpeelliset komponentit.

Työn fyysinen vaihe aloitettiin tekemällä demokokoonpano, joka koostui vanerilevystä ja DIN-kiskon pätkistä. Siihen koottiin ajan myötä riviliittimiä, taajuusmuuttaja, logiikka sekä tarvittavat kontaktorit moottorin lämpösuojalle sekä jarrulle. Moottori vaihteistoineen kiinnitettiin pöytään, jotta moottoria voitiin ajaa turvallisesti.

Laitteiston testaaminen ja kasaaminen tehtiin osissa, alkuun oli saatavilla vain Siemensin logiikka, johon kirjoitettiin koodia valmiiksi, jota simuloitiin, että ohjelma saatiin toimimaan halutulla tavalla. Kompassia eikä taajuusmuuttajaa ollut vielä olemassa, joten testi-simuloinnit tehtiin logiikalle IO-porttien kautta.

Taajuusmuuttajan saavuttua täytyi se ensin kytkeä moottoriin, jossa oli 400 V:n sähköinen jarru. Moottorin mukana ei tullut mitään tietoja, koska se oli hankittu käytettynä. Koska moottoritietoja ei ollut, testattiin jarrua 230 V:n jännitteellä, mutta se ei irrottanut jarrua kokonaan, joten moottori ylikuumeni jatkuvasti. Erilaisten testailujen jälkeen havaittiin, että jarru toimii 400 V jännitteellä. Kun moottorin ongelmat oli selvitetty, siirryttiin taajuusmuuttajan ohjelmointiin. Aluksi työssä käytettiin Siemensin 2,2 kW:n G120C-taajuusmuuttajaa, mutta moottorin käytössä ilmeni ongelmia vajaan minuutin ajon jälkeen. Muutamien virtamittausten jälkeen havaittiin, että kyseisen taajuusmuuttajan virtarajat tulivat vastaan ja taajuusmuuttaja oli liian pienitehoinen kyseiselle moottorille. Päädyttiin tilaamaan Siemensin 4 kW:n G120C-taajuusmuuttajan, jonka avulla moottori saatiinkin pyörimään halutulla tavalla. Moottoria voitiin nyt ajaa logiikkaohjelmaa simuloimalla. Moottorin toimiessa halutulla tavalla siirryttiin käyttöliittymän pariin.

Käyttöliittymää tehtäessä sitä simuloitiin virtuaalisesti tietokoneella. Siitä testattiin kompassineulan liikkuvuus, sekä numerokenttiin kirjoittaminen ja kirjoittamisen esto. Simulointien jälkeen käyttöliittymän toimivuus testattiin demokokoonpanossa. Hienosäätöjen jälkeen oli vuoro tilata Arduino sekä kompassi.

Arduino testattiin ja koodattiin kahdessa osassa, ensin kirjoitettiin kompassille koodi, jota testattiin useamman kerran eri variaatioilla. Kompassin toimiessa siirryttiin datan kirjoittamiseen logiikalle, ja tämä ei ollutkaan aivan helposti tehty. Koodiin piti etsiä useita pätkiä, jotta kirjoittaminen saatiin toimimaan ilman virheilmoituksia. Kun logiikalle saatiin hyväksytysti kirjoitettua dataa, voitiin kaksi toimivaa koodia yhdistää. Koodin vihdoinkin toimiessa lisättiin Arduino demokokoonpanoon ja testattiin kokoonpanon toimivuus. Koska antennimasto ei saanut pyöriä ympyrää, täytyi asteiden olla rajoitettuna 0--359. Moottoria testattiin kompassin eri suuntimilla siltä varalta, että moottori lähtisi pyörimään väärään suuntaan.



Kuva 12. Demolaitteisto toimintakuntoisena.

Kuvassa 12. on demolaitteisto koottuna ja toimintakuntoisena.

6 Ohjelmointi

6.1 Taajuusmuuttaja

Tässä luvussa kerrotaan, kuinka taajuusmuuttajaa ohjelmoitiin, sekä kuinka logiikan ja taajuusmuuttajan välinen tiedonsiirto toteutettiin.

6.1.1 Taajuusmuuttajan parametrisointi

Taajuusmuuttaja täytyy ohjelmoida ennen käyttöä, jotta se toimisi halutulla tavalla. Ohjelmointi tapahtui USB-portin välityksellä käyttäen Siemensin TIA -portalia. Ohjelma on tehty hyvin yksinkertaiseksi, eikä kaikkia parametreja tarvitse käydä läpi vaan riittää, että syötetään moottorin tiedot sekä käytettävä kommunikointiprotokolla. Ohjelma itse laskee moottorille teoreettiset toiminta-arvot. On myös mahdollista tehdä moottorin optimointi,

jossa taajuusmuuttaja pyörittää moottoria ja sen avulla selvittää parhaimmat asetukset moottorin toimintaan.

Motor
Specification of motor type and motor data

Motor configuration
Enter motor data

Select motor type
[1] Induction motor

Select the connection type for your motor and 87 Hz operation:
Star Motor 87 Hz operation

Motor data

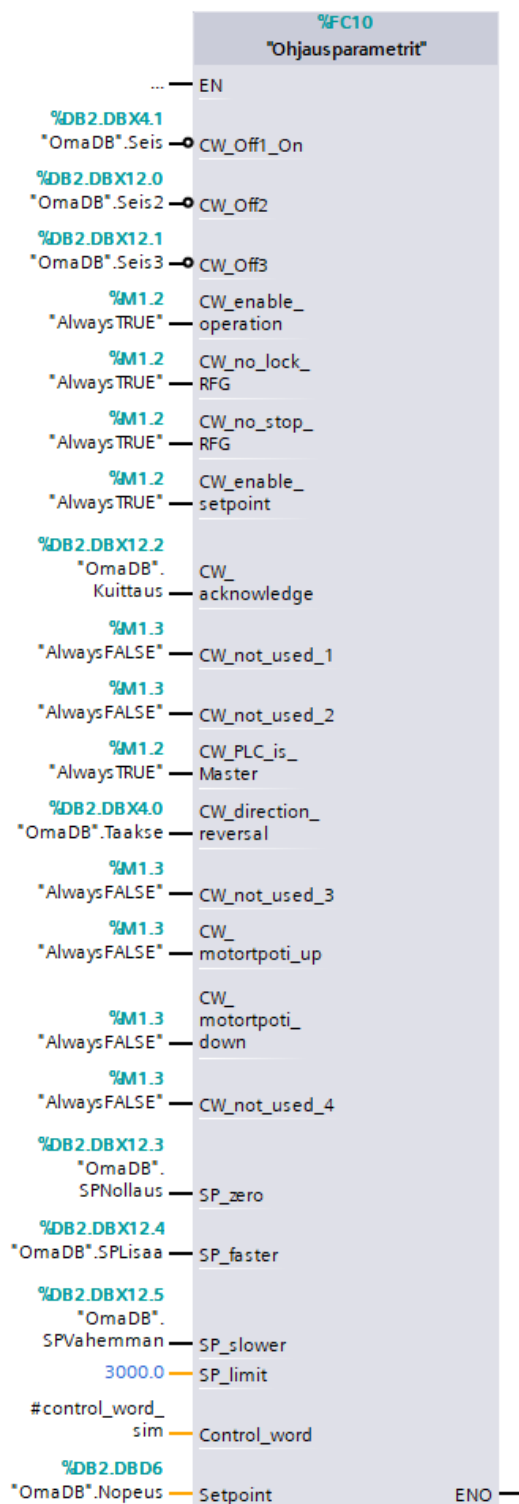
Parameter	Parameter text	Value	Unit
p304[0]	Rated motor voltage	400	Vrms
p305[0]	Rated motor current	4.10	Arms
p307[0]	Rated motor power	3.20	kW
p308[0]	Rated motor power factor	0.750	
p310[0]	Rated motor frequency	50.00	Hz
p311[0]	Rated motor speed	1450.0	rpm
p335[0]	Motor cooling type	[0] Non-v...	<input type="text"/>

Kuva 13. Moottoritietojen syöttö taajuusmuuttajalle.

Kuvassa 13. näkyvään taulukkoon syötetään moottorin tiedot kuten jännite, virta sekä teho. Myös moottorin tyylistä täytyi valita, onko kyseessä oikosulkumoottori vai kesto-
magneettimoottori sekä onko moottorin käämit kytketty tähteen vai kolmioon.

6.1.2 Taajuusmuuttajan ohjaus

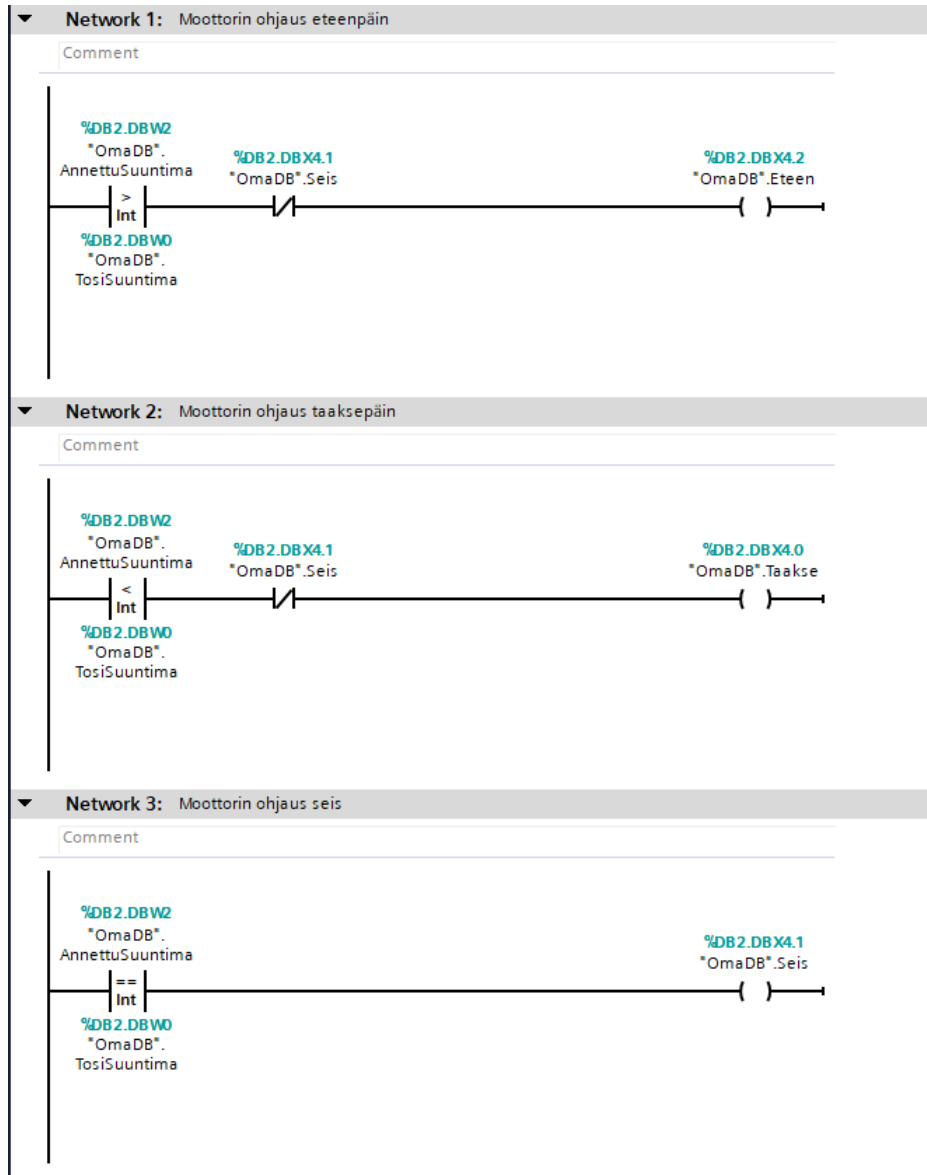
Siemensin G120C-sarjan taajuusmuuttajiin löytyy Siemensin internetsivuilta TIA Portaliin ladattava ohjelmaloikka, jolla voidaan helposti ohjata taajuusmuuttajan peruskomentoja, kuten eteen, taakse ja seis. Tämän ohjelmaloikkon avulla moottorinohjauksen lisäys projektiin kävikin helposti.



Kuva 14. TIA-portalin ohjelmalohko moottorinohjaukselle.

Kuvassa 14. on funktioblokki, jonka avulla moottoria voitiin ohjata haluttuun suuntaan ja oikealla nopeudella.

Logiikalle luotiin funktioblokki, joka ohjaa taajuusmuuttajaa kompassilta saatujen tietojen mukaisesti. Annettua suuntimaa verrataan kompassilta saatuun ja sen mukaisesti moottoria ajetaan joko eteen- tai taaksepäin.



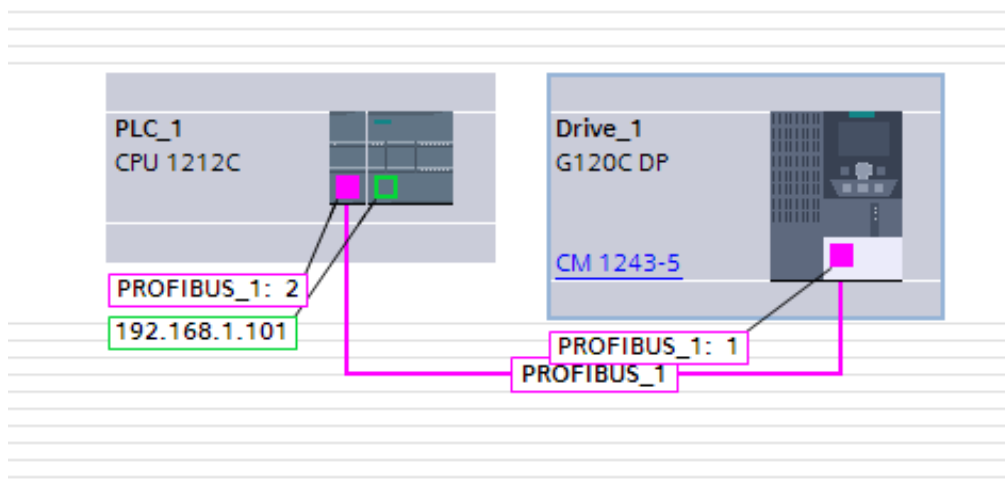
Kuva 15. Moottorin ohjaus.

Kuvassa 15. on relekaavio moottorin ohjauksesta, jossa matemaattisen vertailun jälkeen yksi lähdöstä kytkeytyy päälle ja moottori saa ohjauksenkäskyn joko pysähtyä tai liikkua eteen- tai taaksepäin.

	Name	Data type	Start value	Retain	Accessible f...	Visible in ...	Setpoint
1	▼ Input			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	RD_HWHD	HW_SUBMODULE	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	WR_HWHD	HW_SUBMODULE	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	control_word	Word	16#047E	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	setpoint	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	ref_speed_p2000	Real	1500.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	ref_current_p2002	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	ref_torque_p2003	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	▼ Output			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10	status_word	Word	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11	actual_speed	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12	actual_current	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13	actual_torque	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14	actual_alarm	Word	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
15	actual_fault	Word	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16	RETVAL_RD	Word	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17	RETVAL_WR	Word	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
18	InOut			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
19	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Kuva 16. Taajuusmuuttajan datablokki.

Kuvassa 16. on datablokki, joka toimii datan välivarastona, jonne data tallennetaan ennen sen käyttämistä ohjelmassa. Tällaiselle blokille syntyy tarve, koska FC (Functions) -tyyppiset blokit eivät säilö mitään dataa välimuistiin, vaan toimivat ainoastaan ohjelman suorittavana osana. Tässä datablokissa Input-lehden alla olevat muuttujat lähetetään taajuusmuuttajalle, toimien ohjauskomentoina. Output-lehden alla olevat muuttujat luetaan taajuusmuuttajalta ja ne toimivat tilatietona.



Kuva 17. Profibus-kytkentä.

Kuvassa 17. pinkillä värillä näkyy logiikan sekä taajuusmuuttajan välinen Profibus-väylä. Väylän aliverkkonimi on PROFIBUS_1 ja väylä toimii 1,5 MBps:n nopeudella. Myös

Profibus-osoitteet näkyvät kuvassa. Isäntänä väylässä toimivan logiikan Profibus-osoite on 2 ja orjana toimivan taajuusmuuttajan osoite on 1.

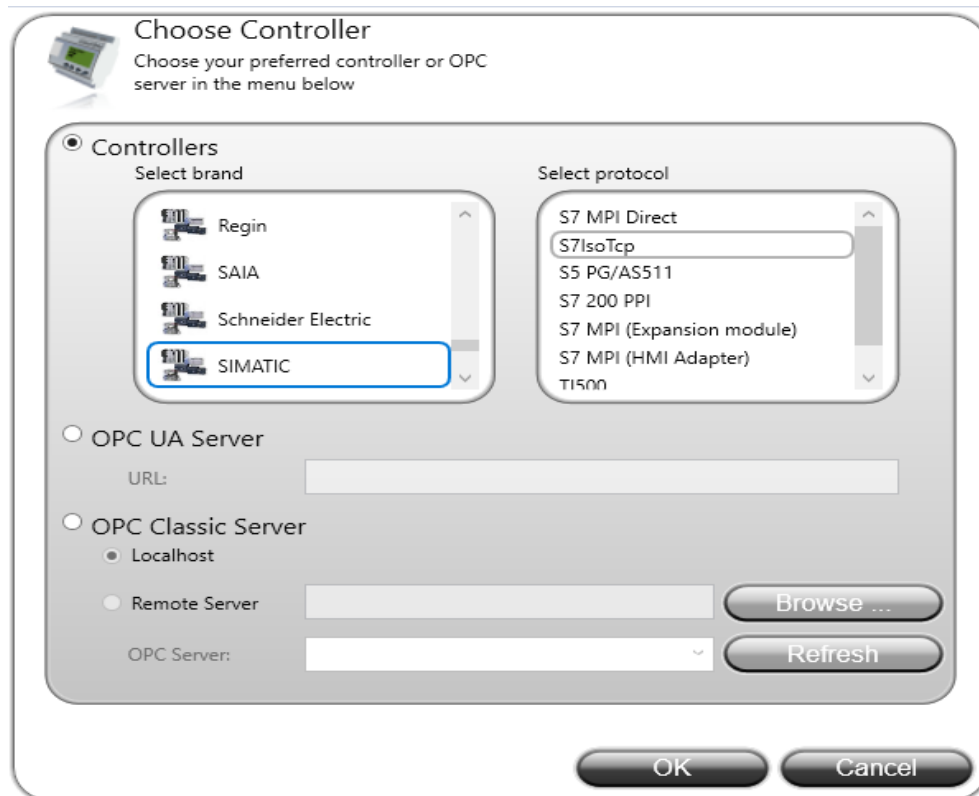
Drive object	Link	Telegram	Length	Extension	Partner	Partner data area
Actual value	➤	SIEMENS Telegramm 352	6 words	0 words →	MS PLC_1	I 256...267
Setpoint	➤	SIEMENS Telegramm 352	6 words	0 words ←	MS PLC_1	Q 256...267
<Add telegram>						

Kuva 18. Profibus-sanoma.

Taajuusmuuttajan kommunikointiin käytettiin Siemensin 352 sanomaa (kuva 18.), joka koostuu 6 lähtevästä sanasta, sekä 6 vastaanotetusta sanasta.

6.2 Käyttöliittymä

Käyttöliittymä oli tarkoitus luoda Siemensin valmistamalla paneelilla, mutta koska työntilaajalta löytyi jo valmiiksi Beijer Electronicsin kosketuspaneelleja, käytettiin niitä projektissa. Ohjelmointityökalu oli alkuun vieras, mutta lyhyen kokeilun jälkeen se tuntui jo su-lavalta käyttää. Kommunikointia varten täytyi logiikka ohjelmaan lisätä datablokki, johon lisättiin paneelissa olevat muuttajat. Myös logiikan IP-osoite täytyi tietää kommunikointia varten.



Kuva 19. Ohjelmointityökalussa on suoraan tuki Siemensin logiikoille.

Kommunikoinnin asettaminen käyttöliittymän sekä logiikan välillä kävi yksinkertaisesti valitsemalla logiikan valmistaja, sekä protokolla kommunikointia varten (kuva 19.). Tuki löytyy melkein kaikille tunnetuille logiikkavalmistajille, sekä lisää ajureita voi hankkia Beijerin sivustolta.

Tag				Controllers	
Name	Data Type	Access Right	Data Type	S7_1200	
Tosisuuntima	DEFAULT	Read	INT16	DB2.DBW0	
Annetusuuuntima	DEFAULT	ReadWrite	INT16	DB2.DBW2	
Taakse	DEFAULT	Write	BOOL	DB2.DBX4.0	
Seis	DEFAULT	Write	BOOL	DB2.DBX4.1	
Eteen	DEFAULT	Write	BOOL	DB2.DBX4.2	
I Nopeus	DEFAULT	ReadWrite	INT16	DB2.DBW6	

Kuva 20. Käyttöliittymän muuttujalista.

Kuvassa 20. käyttöliittymälle luotiin muuttujat, jotka linkitettiin logiikan tietokantaan. Logiikalle luotiin oma tietokanta näitä muuttujia varten. Logiikan puoleisen tietokannan

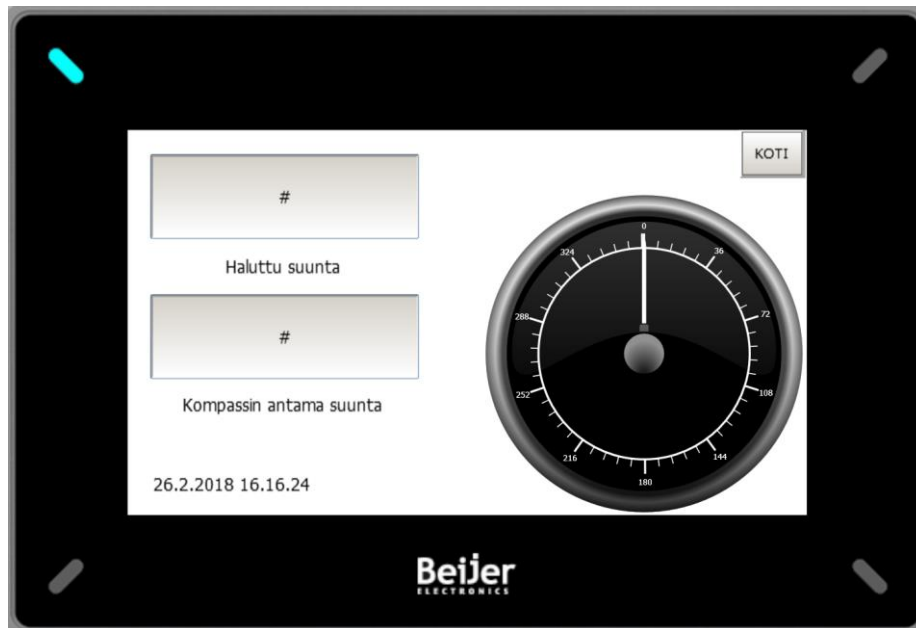
salaus sekä optimointi täytyi poistaa, jotta muuttujia voitiin ohjata toisesta laitteesta käsin. Kuvassa 20. näkyy, että muuttuja Tosisuuntima on muotoa Read, joka tarkoittaa, ettei muuttujan arvoa voida muokata näyttöpäätteestä.

OmaDB				
	Name	Data type	Offset	Start value
1	Static			
2	TosiSuuntima	Int	0.0	0
3	AnnettuSuuntima	Int	2.0	0
4	Taakse	Bool	4.0	false
5	Seis	Bool	4.1	false
6	Eteen	Bool	4.2	false
7	Nopeus	Int	6.0	0
8	<Add new>			

Kuva 21. Logiikan tietokanta käyttöliittymälle.

Logiikalle luotu käyttöliittymän tietokanta (kuva 21.), joka on yhtenäinen kuvassa 20. olevaan tietokantaan. Muuttujan Offset-arvo tarkoittaa muuttujan alkutavun osoitetta.

Näyttöpäätteen koosta johtuen ei käyttöliittymään (kuva 22.) mahtunut paljoa tietoa, joten siitä tehtiin hyvin yksinkertainen. "Haluttu suunta" -kenttään kirjoitetaan luku, jonne antennimaston tahdotaan kääntyvän, luku on rajoitettu 0--359 asteen välille. Vastavuoroisesti kompassi antaa lukeman antennimaston suunnasta "Kompassin antama suunta" -kenttään. Myös graafinen kompassi liikkuu antennimaston suunnan mukaisesti.



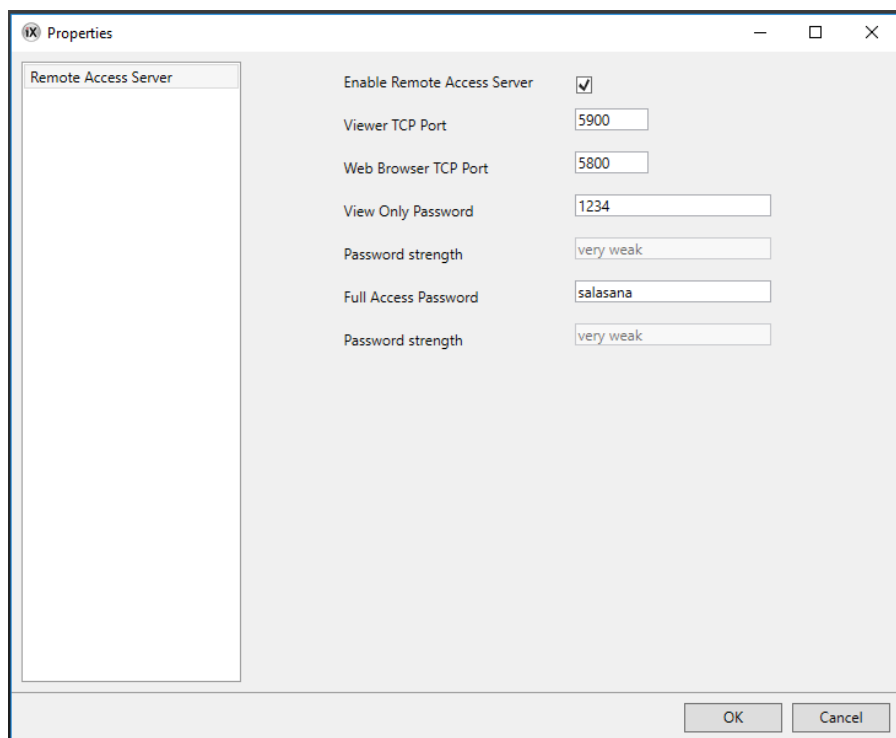
Kuva 22. Käyttöliittymän perusnäkö.

Asiakkaan pyynnöstä käyttöliittymään (kuva 23.) tehtiin myös välilehti moottorin pakko-käyttöä varten. Siitä oli myös apua laitteiston testaamisessa. Tulevaisuudessa jos moottorin sekä maston välitykset tulevat muuttumaan, on moottorille helppo asettaa uusi pyörimisnopeus vastaamaan haluttua nopeutta.



Kuva 23. Käyttöliittymän testausnäkö.

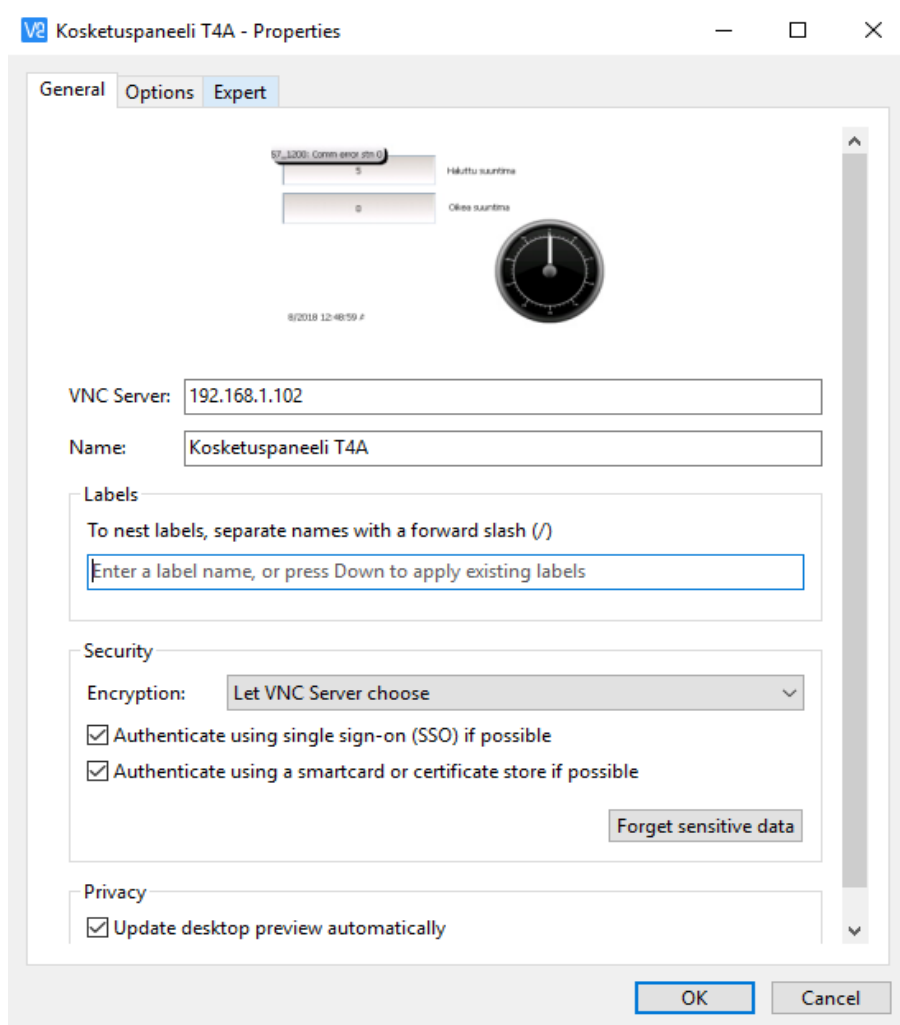
Antennimaston ohjaukselle tahdottiin etäkäyttö mahdollisuus, esimerkiksi tabletista tai älypuhelimesta. Koska näyttöpaneeliin saa asennettua etäkäyttöpalvelimen, saatiin siihen yhteys käyttämällä VNC Viewer -nimistä etäkäyttöohjelmaa. Ohjelmalla saadaan täysi toimivuus antennimaston ohjaamiseen. VNC Viewer tukee Androidia, joten sen voi ladata Google Play Storesta omalle tabletille tai älypuhelimelle.



Kuva 24. Etäkäyttöpalvelimen määrittämät.

Etäkäyttöpalvelin (kuva 24.) on hyvin yksinkertainen asettaa päälle, tarvitsee vain asettaa TCP-portit, sekä salasana katsojille sekä käyttäjille.

VNC Vieweriin luotiin New Connection, josta päästiin kuvassa 25. olevaan näkymään. Tarpeellisia tietoja olivat vain kosketusnäytön IP-osoite, sillä salaustapa oli asetettu serverin, eli kosketusnäytön puolelta. Asetusten määrittämisen jälkeen yhteyttä ottaessa kyettiin kuvassa 24. asetettua salasanaa.



Kuva 25. VNC Viewerin määrittelyt.

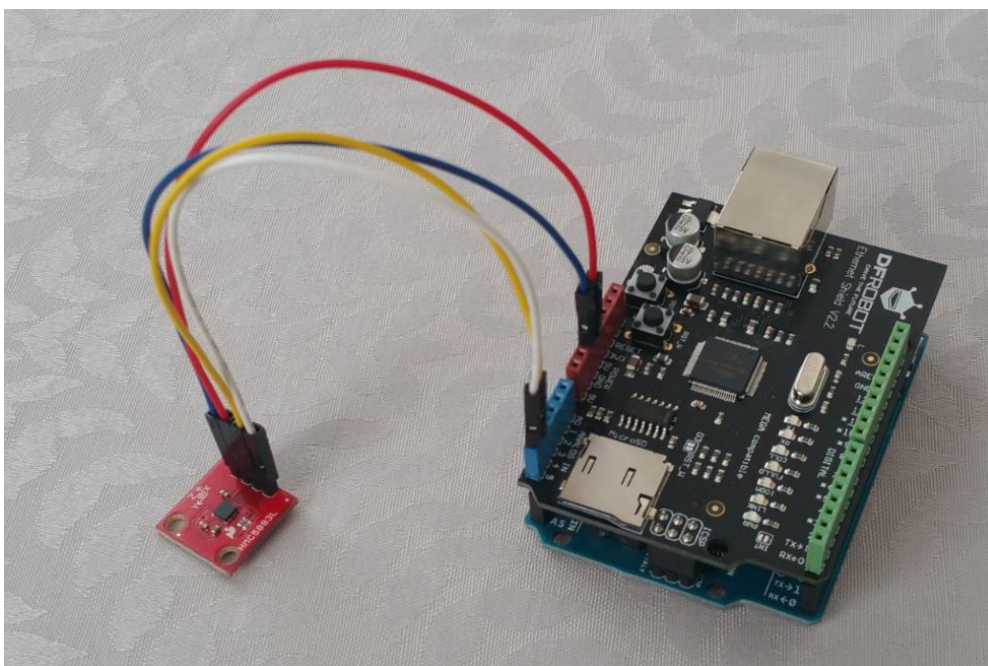
6.3 Elektroninen kompassi

Kompassi (kuva 26.) on pelkkä mikropiiri, joka lähettää analogista dataa. Ennen logiikalle menoa data täytyy koodata sille sopivaan muotoon, ja siihen tarvittiin Arduino-mikrotietokonetta. Kompassina toimi HMC5883L-mikropiiri, joka kytkettiin Arduinoon käyttämällä I2C-väylää. Ethernet-moduuli, joka mahdollistaa tiedonsiirron logiikalle, kytkettiin Arduinon SPI-väylään.



Kuva 26. Kompassi.

Arduinon sekä logiikan välinen tiedonsiirto tehtiin aluksi IO:n kautta binäärikoodina siten, että yksi IO-kanava vaihtaa bittiä seuraavaan, toinen kanava kertoo, onko bitti 1 vai 0 ja kolmas kanava kertoo, kun bittisana alkaa alusta. Myöhemmässä vaiheessa kommunikointi protokollaksi valittiin Ethernet.



Kuva 27. Kompassi kytkettynä Arduinoon.

Kuvassa 27. kompassi kytkettiin Arduinon pinneihin A4 ja A5, jotka toimivat I2C-väylän data- ja clock-kanavina. Virta otettiin 5 V:n pinnistä.

6.3.1 Arduinon ohjelmointi

Ensimmäisenä täytyi etsiä kompassille kuuluva kirjasto internetistä, jonka löydyttyä oli mahdollista testata kompassin toimivuus. Nopealla haulilla löytyi Kirjasto, joka sisälsi hyvin kommentoidun esimerkkiohjelman. Koodiin tarvitsi vain muuttaa oman alueen magneettikentän eranto, jonka jälkeen kompassi olikin jo käyttövalmis.

Kun kompassi oli saatu koodattua toimivaksi, tarvittiin vielä ohjelma datansiirrolle, jotta kompassin data siirtyisi logiikalle. Suoraan valmista koodia ei löytynyt, vaikka kuinka etsittiin. Lähin koodiesimerkki jonka löysin, oli Siemensin S7-tietokannan lukemiseen, joten koodi täytyi muokata kirjoittamaan tietokantaan. Suurin muutos koodin oli tavujärjestyksen muuttaminen, sillä Arduino käyttää little endian -tavujärjestystä, kun taas Siemens S7 1200 käyttää big endian -tavujärjestystä. Tavujärjestyshän tietokoneessa tarkoittaa sitä, missä järjestyksessä tietokoneen suoritin käsittelee suurempia kuin yhden tavun pituisia kokonaisuuksia.

```
void Reverse2(void *ptr)
{
    byte *pb;
    byte tmp;
    pb=(byte*) (ptr);
    // Swap byte 2 with byte 1
    tmp=*(pb+1);
    *(pb+1)=*pb;
    *pb=tmp;
}
```

Esimerkkikoodi 1. Tavujärjestyksen muuttaminen.

Esimerkkikoodissa 1. nähdään kyseinen tavujärjestyksen muutos, jossa kaksi tavuisessa sanassa vaihdetaan ensimmäinen sekä toisen tavun paikkaa.

7 Yhteenveto

Opinnäytetyön aihe oli melko laaja, ja se sisälsi paljon jo opitun sekä uusien asioiden soveltamista. Työn aikana opin paljon uutta sulautetuista järjestelmistä sekä niiden ohjelmoinnista. Työn tavoitteina oli laatia antennimaston suuntimiseen automaattinen ohjausjärjestelmä, jonka avulla mastoa voidaan kääntää haluttuun suuntaan. Työssä suunniteltiin, rakennettiin ja testattiin kääntökoneistoa pienissä paloissa antennimastoa varten. Koneistoa varten saatiin luotua käyttöliittymä, sekä moottorinohjaukseen tarvittava logiikkaohjelma. Tavoitteet, jotka asetettiin työn alussa, saavutettiin täysin. Jollei opinnäytetyötä olisi tehty, asiakas olisi joutunut hankkimaan valmiin ratkaisun, joka olisi ollut suljettu tietojärjestelmä. Asiakas sai tästä työstä avoimen sekä helposti muokattavan kokonaisuuden. Testaamatta jäi tietenkin se, kuinka kokonaisuus tulee toimimaan aikaan oikean maston kanssa.

Lähteet

- 1 Tia portal. Verkkoaineisto. Siemens Suomi. <http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/tia_portal.php>. Luettu 16.2.2018.
- 2 Step 7. Verkkoaineisto. Siemens Suomi. <http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/automaatiotekniikka/ohjelmoitavat_logikat_simatic/ohjelmistot/tia_portal_step7.htm>. Luettu 16.2.2018.
- 3 Beijer kosketusnäytöt. Verkkoaineisto. UTU Automation. <<https://www.utu.eu/automaatio/kayttoliittymat/beijer-electronics-hmi/ohjelmistot>>. Luettu 20.2.2018.
- 4 Profibus. Verkkoaineisto. Siemens suomi. <http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/automaatiotekniikka/teollinen_tiedonsiirto_esim_profinet/profibus.htm>. Luettu 25.2.2018.
- 5 Profibus. Verkkoaineisto. Automation Networks. <<https://automation-networks.com/glossary/profibus>>. Luettu 28.2.2018.
- 6 Ethernet. Verkkoaineisto. TechTarget. <<http://searchnetworking.techtarget.com/definition/Ethernet>>. Luettu 27.2.2018.
- 7 SPI-sarjaliikenne. Verkkoaineisto. Hutasu. <<https://www.hutasu.net/elektronikka/sulautettu-elektronikka/spi-sarjaliikenne/>>. Luettu 4.3.2018.

Arduinon ohjelmakoodi

```
#define S7WIRED
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_HMC5883_U.h>
#include <SPI.h>
#include <Ethernet.h>
#include <Settimino.h>

Adafruit_HMC5883_Unified mag = Adafruit_HMC5883_Unified(12345); // Kompassille
uniikki ID
byte mac[] = { 0x90, 0xA2, 0xDA, 0x0F, 0x08, 0xE1 }; // Arduinon MAC
IPAddress Local(192,168,1,105); // Arduinon Osoite
IPAddress PLC(192,168,1,101); // PLC Osoite
byte Buffer[1024];
S7Client Client(_S7WIRED); // Luodaan TCP Client
//SETUP
void setup() {
  Serial.begin(9600); // Määritetään yhteysnopeus
  //Ethernet Shieldin määrittäykset
  Ethernet.begin(mac, Local);
  delay(2000);
}
//Yhteys PLClle
bool Connect()
{
  int Result=Client.ConnectTo(PLC,
                              0, // PLCn räkki
                              0); // PLCn paikka räkissä
}
//Datapuskurin tyhjennys
void Dump(void *Buffer, int Length)
{
  int i, cnt=0;
  pbyte buf;
  if (Buffer!=NULL)
    buf = pbyte(Buffer);
  else
    buf = pbyte(&PDU.DATA[0]);
  for (i=0; i<Length; i++)
  {
    cnt++;
    if (cnt==16)
    {
      cnt=0;
    }
  }
}
// Tavujärjestyksen muuntaminen Little to Big Endian
void Reverse2(void *ptr)
{
  byte *pb;
  byte tmp;
  pb=(byte*) (ptr);
  // Swap byte 2 with byte 1
  tmp=* (pb+1);
  * (pb+1)=*pb;
  *pb=tmp;
}
// PÄÄOHJELMA
void loop()
{
```

```
// Kompassin ohjelma
sensors_event_t event;
mag.getEvent(&event);
float heading = atan2(event.magnetic.y, event.magnetic.x); // Pidä kompassia
Z-akseli ylöspäin
float declinationAngle = 0.22; // Magneettikentän eranto radi-
aaneina:http://www.magnetic-declination.com/
heading += declinationAngle;
// Korjaus käänteisille arvoille.
if(heading < 0)
    heading += 2*PI;
if(heading > 2*PI)
    heading -= 2*PI;
float headingDegrees = heading * 180/M_PI; // Radiaanit asteiksi
int angle=(int)headingDegrees; // Liukuluku kokonaisluvuksi
int Size, Result;
void *Target;
byte Bytes;
// Yhteys PLClle
while (!Client.Connected)
{
    if (!Connect())
        delay(500);
}
Reverse2(&angle); // Suuntiman tavujärjestyksen muuttaminen
Result=Client.WriteArea(S7AreaDB, // Komento Databaseen kirjoittamiseen
                        2, // Databasen numero
                        0, // Muuttujan numero
                        sizeof(int), // Muuttujan koko
                        &angle); // Muuttuja joka kirjoitetaan

if (Result==0)
{
    Dump(Target, Size);
}
delay(500);
}
```