

**Mikko Kavonius**

**SERVOPAIKOITUSLAITTEISTO MEKATRONIIKAN  
OPETUKSEEN**

**Opinnäytetyö  
KESKI-POHJANMAAN AMMATTIKORKEAKOULU  
Automaatiotekniikka  
Kesäkuu 2012**



## TIIVISTELMÄ OPINNÄYTETYÖSTÄ

<b>Yksikkö</b> Tekniikka ja liiketalous, Kokkola	<b>Aika</b> Kesäkuu 2012	<b>Tekijä/tekijät</b> Mikko Kavonius
<b>Koulutusohjelma</b> Automaatiotekniikan koulutusohjelma		
<b>Työn nimi</b> Servopaikoituslaitteisto mekatroniikan opetukseen		
<b>Työn ohjaaja</b> Mikko Mäki-Petäjä		<b>Sivumäärä</b> 36 + 17
<b>Työelämäohjaaja</b> Mikko Mäki-Petäjä		
<p>Opinnäytetyön aiheena oli suunnitella ja rakentaa mekatroniikan kurssin laboratorioharjoituksissa käytettävä servo-opetuslaitteisto. Laitteiston avulla opiskelija pääsee tutustumaan servotekniikan ohella logiikkaohjelmointiin. Lisäksi opiskelija tekee laitteistolle käyttöliittymän operointipaneelille.</p> <p>Laitteiston runkona käytettiin vanhaa testipenkkiä, jolla on tehty rasituskokeita kuntosalilaitteiden sylintereille. Aiemmasta kokoonpanosta säästettiin tähän laitteistoon runko, mekaaniset osat ja rajakytkimet. Työ piti sisällään laitteiston suunnittelun, laitehankintojen, asennuksen, ohjelmoinnin ja testauksen ohella laboratorioharjoituksen ja käyttöohjeen laatimisen.</p>		
<b>Asiasanat</b> koneturvallisuus, paikoitus, servotekniikka		

**ABSTRACT**

<b>CENTRAL OSTROBOTHNIA UNIVERSITY OF APPLIED SCIENCES</b>	<b>Date</b> June 2012	<b>Author</b> Mikko Kavonius
<b>Degree programme</b> Automation Engineering		
<b>Name of thesis</b> Servo positioning system for teaching mechatronics		
<b>Instructor</b> Mikko Mäki-Petäjä		<b>Pages</b> 36 + 17
<b>Supervisor</b> Mikko Mäki-Petäjä		
<p>The subject of this thesis was to design and build a teaching machine for servo technology to be used in mechatronics laboratory rehearsals. The machine provides a possibility for students to explore servo technology and logic programming. The students also design and program the user interface for the operating panel.</p> <p>An old servo bench which was originally used for tolerance testing for the cylinders of gym equipment was used as a frame for the machine. The frame, mechanic parts and limit switches were maintained and reused in this project. The project consisted of designing, acquisition, assembling, programming and testing the machine. A considerable part of the project was also to create a laboratory rehearsal and a user manual for the machine.</p>		

<b>Key words</b> machine safety, positioning, servo technology
---

**TIIVISTELMÄ**  
**ABSTRACT**  
**SISÄLLYS**

<b>1 JOHDANTO</b>	<b>1</b>
<b>2 SERVOTEKNIikka</b>	<b>2</b>
2.1 Yleistä servotekniikasta	2
2.2 Servovahvistin	3
2.3 Servomoottori	4
2.4 Kiertymänmittausanturit	5
2.4.1 Optinen kiertymänmittausanturi	6
2.4.2 Resolveri	7
2.5 EMI-suodatin	7
<b>3 LOGIIKKAOHJAIMET</b>	<b>9</b>
3.1 Nopea laskuri	10
3.2 Operointipaneeli	10
<b>4 KONETURVALLISUUS</b>	<b>12</b>
4.1 Turvarele	12
4.2 Häätä-seispainike	13
4.3 Valoverho	13
4.4 Mekaaninen rajakytkin	13
<b>5 PROJEKTIN ETENEMINEN</b>	<b>15</b>
5.1 Laitehankinnat	15
5.1.1 Servovahvistin: Yaskawa Junma SJDE-020APA-OY	15
5.1.2 Servomoottori: Yaskawa Junma SJME-02AMB41-OY	16
5.1.3 EMI-suodatin: Block R7A-FIZP105-BE	17
5.1.4 Logiikka: Siemens Simatic S7-1200	17
5.1.5 Operointipaneeli: Siemens Simatic HMI KTP600 Basic color	18
5.1.6 Pulssianturi: Pepperl & Fuchs 58	19
5.1.7 Muut laitteet	19
5.2 Laitteistoon tutustuminen	20
5.3 Häätäpysäytyspiirin suunnittelu	21
5.4 Rakentaminen	22
5.5 Ohjelmointi	25
5.5.1 Ohjelmoinnin ongelmat	26
5.5.2 Laitemäärityt	27
5.5.3 Akselin määritys	28
5.5.4 Ajosekvenssin luominen	29
5.5.5 Operointipaneelin ohjelmointi	31
5.6 Testaaminen	31
5.7 Harjoituksen ja pikaoppaan laatiminen	32
<b>6 POHDINTA</b>	<b>34</b>
<b>LÄHTEET</b>	<b>36</b>

## **LIITTEET**

Liite 1. Laboratorioharjoitus

Liite 2. TIA Portal –pikaopas

Liite 3. Kytkäkaavio: 230 VAC syötöt

Liite 4. Kytkäkaavio: 24 VDC syötöt

Liite 5. Kytkäkaavio: toiminnot

## 1 JOHDANTO

Mekatroniikan ja automaatiotekniikan opintojen eräs olennainen osa-alue on servokäytöt. Servotekniikan hyvä tuntemus on perinteisesti edellyttänyt perehtymistä sähkötekniikan ja sähkökäyttöjen lisäksi myös säätötekniikkaan, sillä servovahvistimien toiminta perustuu viritettäviin PID-säätimiin. Nykyisin markkinoilla on myös itsestään viritettyjä servokoonpanoja, joiden käyttöönotossa ei säätötekniikan tuntemus ole välttämätöntä. Tällaiseen kokoonpanoon tutustuminen soveltuu mainiosti mekatroniikan peruskurssin laboratoriotöihin.

Opinnäytetyöni aihe oli rakentaa ammattikorkeakoulun sähkölaboratorioon opetuslaitteisto, jolla sekä automaatio- että konetekniikan opiskelijat pääsevät helposti tutustumaan servokäyttöjen ja ohjelmoitavien logiikoiden maailmaan. Opinnäytetyö piti sisällään laitteiston suunnittelun ja rakentamisen ohella laboratorioharjoituksen ja käyttöoppaan laatimisen. Lisäksi tein malliohjelman laboratoriotöitä valvovaa opettajaa/ohjaajaa varten. Hyödynsin laitteiston rakentamisessa vanhaa testipenkkiä, jota on aiemmin käytetty kuntosalilaitteiden sylintereiden rasiuskokeisiin. Penkin mekaaninen rakenne koostui runkokehikosta, kuularuuvista, lineaarijohteesta ja kelkasta. Kelkka liikkui lineaarijohdetta pitkin edestakaisin servomoottorin pyörittämän kuularuuvin välityksellä. Testipenkki oli ollut jo vuosikaudet poissa käytöstä.

Vanhan kokoonpanon hyödyntäminen opetuskäytössä oli poissuljettua, sillä sen lisäksi, että tekniikka oli vanhentunutta, oli laite myös rakenteeltaan vaarallinen. Suurin haittatekijä oli kuitenkin dokumentoinnin täydellinen puuttuminen.

Uusi järjestelmä oli tarkoitus rakentaa puhtaasti opetuskäyttöön hyödyntäen nykyaikaista teknologiaa. Vanhasta kokoonpanosta päätettiin säästää vain runko, kuularuuvi ja kelkka, joten kaikki sähköiset osat tuli uudistaa. Vanha ohjauselektroniikka päätettiin korvata ohjelmoitavalla logiikalla. Ohjauksen tuli tapahtua kosketuspaneelilta. Lisäksi järjestelmään haluttiin turvapiiri, joka suojaa sekä laitteistoa että käyttäjää. Järjestelmän tuli tarjota opiskelijoille mahdollisuus harjoitella servomoottorin paikoitus- ja nopeusajoa, servon ohjausta logiikalta, paneelin ohjelmointia, antureiden liittämistä logiikkaan ja turvalaitteiden kytkentää.

## 2 SERVOTEKNIikka

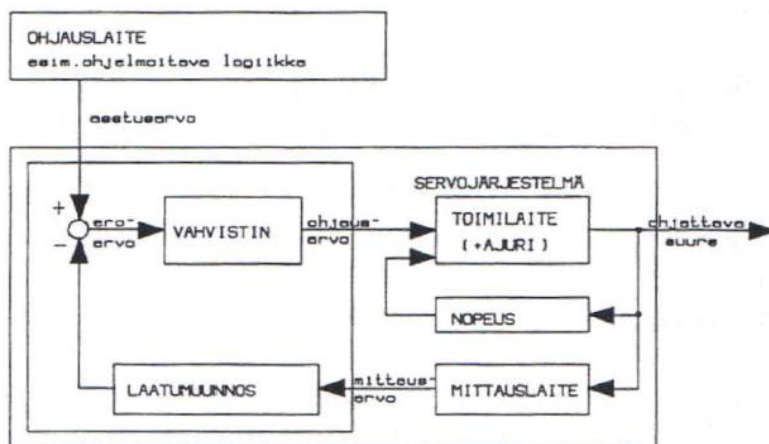
### 2.1 Yleistä servotekniikasta

Servotekniikan kehitys juontaa juurensa 2. maailmansodasta, jolloin säätötekniikan keinoin ryhdyttiin parantamaan sotakaluston nopeutta ja tarkkuutta. Sodan jälkeen servotekniikkaa ryhdyttiin soveltamaan teollisuuden säätöventtiileissä ja kopiotyöstökoneissa. Nykyisin yleisimpiä servotekniikan sovelluksia ovat teollisuusrobotit, manipulaattorit ja CNC-työstökoneet. (Fonselius, Rinkinen & Vilenius 1998, 7.)

Servojärjestelmän tehtävänä on saattaa kappale toimilaitteen, kuten moottorin tai sylinterin, avulla haluttuun asemaan tai nopeuteen. Järjestelmä muodostaa suljetun säätöpiirin, eli se automaattisesti varmistaa ohjearvon saavuttamisen. Tämä perustuu takaisinkytkentään, eli järjestelmä vertaa ohjearvoa todelliseen mitattuun arvoon ja säätää ohjauksen voimakkuutta näiden arvojen erotuksen perusteella. Servojärjestelmät jaetaan säädettävän suureen perusteella

- asemaservoihin
- nopeusservoihin
- momenttiservoihin ja
- voimaservoihin. (Fonseliys ym. 1998, 7.)

Servojärjestelmä koostuu kuvion 1 esittämän periaatteen mukaisesti toimilaitteesta, sitä ohjaavasta vahvistimesta, takaisinkytkentäanturista ja ylemmän tason ohjauslaitteesta, josta järjestelmään syötetään asetusarvo. (Fonselius ym. 1998, 8.)



KUVIO 1. Servojärjestelmän toiminnallinen rakenne (Fonselius ym. 1998, 8.)

## 2.2 Servovahvistin

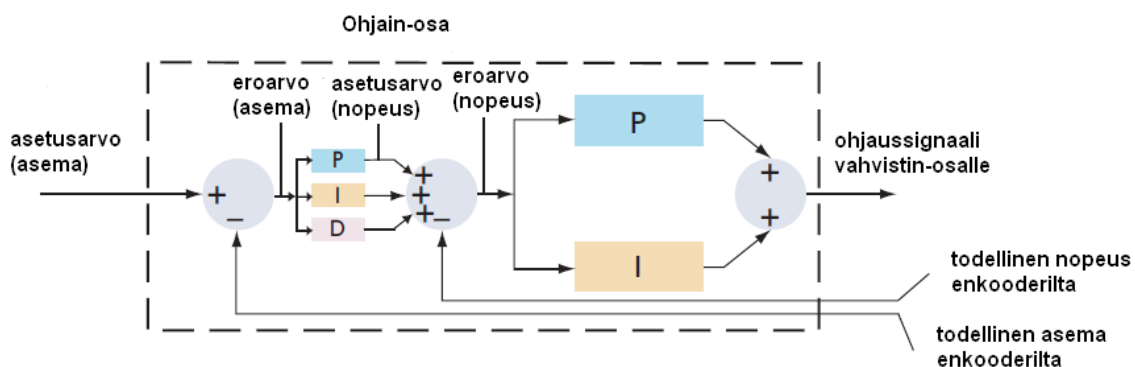
Servojärjestelmän toimilaitteen ohjaus koostuu säätävästä ohjainosasta ja vahvistinosasta. Yleensä sama laite pitää sisällään molemmat, jolloin käytetään nimitystä servovahvistin tai servo-ohjain. Valmistajan mukaan myös muita nimityksiä on käytössä, kuten esimerkiksi servopaketti. Usean akselin sovelluksissa käytetään erillisiä ohjain- ja vahvistinyksiköitä. (SEW Eurodrive. 1998, 29–30.)

Servovahvistin ohjaa toimilaitetta asetusarvon ja mitatun arvon eron perusteella. Vahvistin vähentää asetusarvosta mitatun arvon ja muuntaa tuloksena saadun ohjausarvon toimilaitetta ohjaavaan muotoon. Mitä lähempänä arvot ovat toisiaan, sitä pienempi on ohjaus. Asetusarvo voidaan antaa vahvistimelle vaikkapa pulssijonona, ja toimilaitteen ohjaus voi olla esimerkiksi vahvistimen hakkuripääteasteen tuottamaa kolmivaihevirtaa. (Fonselius ym. 1998, 8.)

Vahvistin päättelee, millaisella ohjaustavalla asetusarvo saavutetaan vakaasti ja mahdollisimman nopeasti. Hitaisissa servojärjestelmissä riittää usein pelkkä eroarvon vahvistus, jolloin puhutaan P-säädöstä. Nopeammissa järjestelmissä vahvistimen piirit ottavat huomioon eroarvon ajallisen käyttäytymisen integrointi- ja derivointitermein, jolloin kyseessä on PI- tai PID-säädin. (Fonselius ym. 2008, 8.)



Kuviossa 2 nähdään tyypillisen asemaservovahvistimen ohjainosan toimintaperiaate, jossa sisempi nopeustakaisinkytkentä muodostaa asematakaisinkytkennän kanssa kaskadisäätöpiirin.



KUVIO 2. Aseman ja nopeuden kaskadisäätö (mukaiillen Lewin, C. 2007)

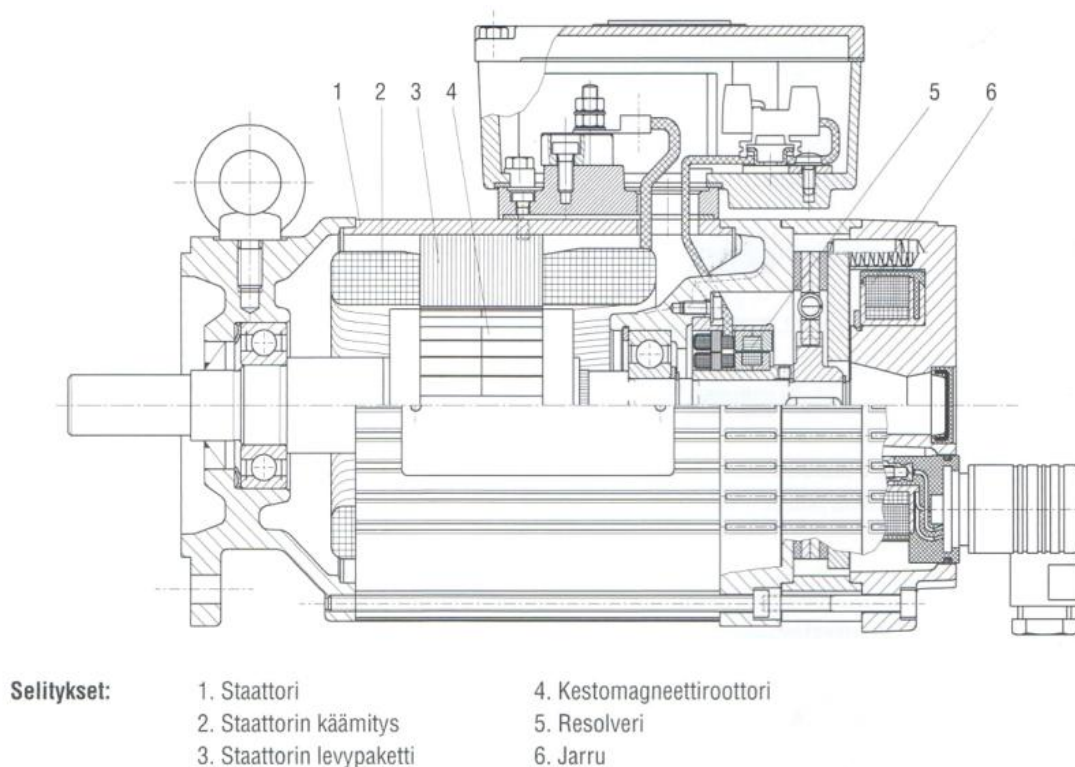
Vahvistimen käyttäjä antaa asetusrvona haluamansa aseman. Enkooderilta saatu todellinen asema vähennetään asetusrvosta ja eroarvo viedään PID-säätimelle. Säätimen vahvistama eroarvo, joka on sitä suurempi, mitä kauempana ollaan halutusta asemasta, viedään sisemmälle säätimelle nopeuden asetusrvona. Tästä taas vähennetään todellinen mitattu nopeus ja näin saadusta eroarvosta muodostetaan lopullinen ohjausarvo, tyypillisesti PI-säätimellä. Vahvistin muodostaa näin kiihdytys- ja jarrutusrampit. (Nelson 1999, 87–90.)

### 2.3 Servomoottori

Sähköinen servojärjestelmä on toteutettavissa millä tahansa sähkömoottorilla, joskin usein käytetään erityistä servomoottoria. Servomoottori on erikoisrakenteinen vaihto- tai tasavirtamoottori. Oleellinen ero tavalliseen moottoriin nähden on hitausmomentti, joka on servomoottorissa huomattavasti pienempi. Tämä mahdollistaa moottorin nopean toiminnan suuren kiihtyvyyden ja hidastuvuuden myötä. (Fonselius ym. 1998, 10.)

Aiemmin servomoottoreina käytettiin DC-moottoreita niiden säätöominaisuuksien vuoksi. Nykyään transistoreiden kehityttyä suurien ja korketaajuisien virtojen säätöön soveltuviksi, ovat AC-servomoottorit yleistyneet. Niiden käyttöä tukevat pieni koko, huoltovapaus ja hiljainen käyntiäänäni. (Halme & Parikka 2005, 6–7.)

Kestomagnetoitu synkronimoottori (KUVIO 3), jota myös harjattomaksi vaihtovirtamoottoriksi kutsutaan, täyttää hyvin servojärjestelmän moottorille asettamat vaatimukset. Moottori koostuu kestmagneettiroottorista ja staattorista, jonka käämeihin tuodaan kolmivaihevirtaa. Moottorin akselille on kytketty suuriresoluutioinen enkooderi tarkkailemaan jatkuvasti roottorin asentoa. Asennon perusteella elektroninen kommutaattori ohjaa staattorin kullekin käämille virtaa siten, että staattorin kenttä on koko ajan  $90^\circ$  sähköisesti roottorin kenttää edellä. Enkooderilta saadaan myös takaisinkytkentä servovahvistimelle. (Fonselius ym. 1998, 145–146.)



KUVIO 3. Kestomagnetoidun synkronimoottorin halkaisukuva (SEW Eurodrive 1998.)

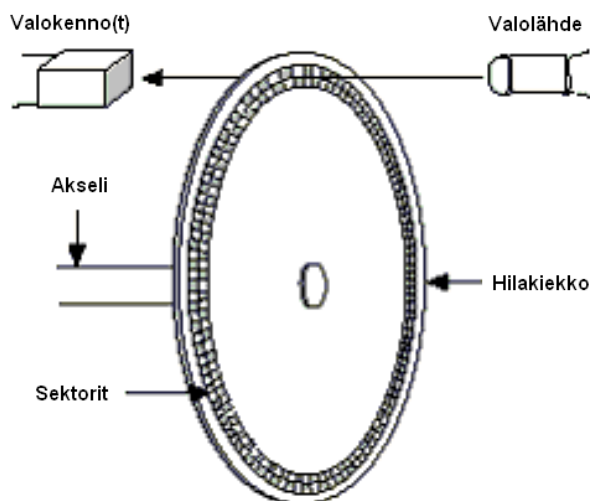
## 2.4 Kiertymänmittausanturit

Ohjattavan kappaleen todellinen asema voidaan mitata suoraan esimerkiksi kappaleeseen kiinnitetyllä lineaaripotentiometrillä tai epäsuorasti moottorin akselin kiertymiskulman perusteella. Molemmille mittaustavoille on saatavilla sekä digitaalisia että analogisia antureita. Mittaustulos voi olla absoluuttinen, jolloin asemasta saadaan todellinen tieto jatkuvasti, tai inkrementaalinen, jolloin mitataan etäisyyttä ns. kotipisteeseen. Inkrementaalinen antu-

ri siis kadottaa paikkatietonsa mekaanisen luiston tai sähkökatkon seurauksena ja se täytyy käyttää kotipisteessään, jotta saataisiin taas realistista tietoa. (Airila 2000, 4:3–16.)

#### 2.4.1 Optinen kiertymänmittausanturi

Digitaaliseen kiertymänmittaukseen käytetään usein optisia antureita. Ne ovat tarkkoja, monipuolisia ja suhteellisen edullisia. Saatavilla on sekä inkrementaalisia eli pulssiantureita että absoluuttisia eli koodiantureita. (Airila 2000, 4:12–13.) Pulssianturi koostuu valolähteestä, valokennosta ja hilakiekosta. Hilakiekkko on jaettu peräkkäisiin valoa läpäiseviin ja läpäisemättömiin sektoreihin. (KUVIO 4). Valokenno vastaanottaa valolähteen tuottamaa valoa, jota hilakiekkko pyöriessään katkoo. Näin muodostuu pulsseja, joiden lukumäärä ilmaisee edetyn matkan ja esiintymistajuuus pyörimisnopeuden. Myös pyörimissuunta saadaan selville, kun käytetään kahta peräkkäistä valokennoa tai vaihtoehtoisesti kahta rinnakkaista valokennoa ja kahta sektoriraitaa siten, että ulostulopulssit ovat 90°:n vaihe-siirrossa toisiinsa nähden. Pulssiantureista saadaan usein myös nollapulssi, jolla määritetään pulssilaskennan nollakohta. Tätä voidaan hyödyntää esimerkiksi kokonaisten kierrosten laskemiseen. (Fonselius ym. 1998, 126–127.)



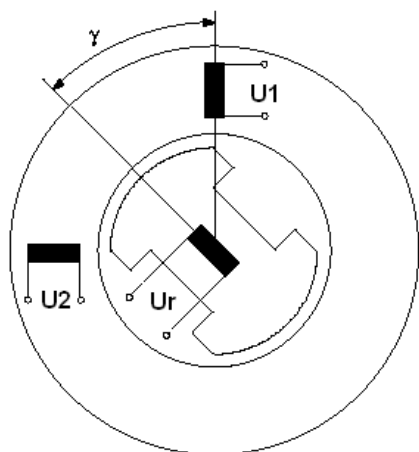
KUVIO 4. Optisen pulssianturin rakenne (mukaillen PC Control Ltd. 2008)

Koodianturi muistuttaa rakenteeltaan pitkälti pulssianturia. Sektoriraitoja on 6–20 ja valokennoja vastaava määrä. Eri raitojen sektorit ovat leveydeltään erisuuruisia, jolloin kukin kiekon asento muodostaa tarkkuuden määräämissä rajoissa oman digitaalisanansa. Koko-

naiset kierrokset täytyy kuitenkin laskea inkrementaalisesti, joten niiden määrä kadotetaan sähkökatkon yhteydessä, vaikka kiertymiskulma säilytetäänkin. (Airila 2000, 4:15.)

## 2.4.2 Resolveri

Resolveri on analoginen kiertymänmittausanturi, joka toimii pyörivän muuntajan periaatteella. Resolverin roottorin käämiin syötetään vaihtojännite, joka vastaa muuntajan ensiöjännitettä. Staattorissa on kaksi käämiä, jotka ovat  $90^\circ$ :n vaihesiirrossa toisiinsa nähden. Roottorin pyöriessä ensiöjännite indusoituu vuoronperään staattorin käämeihin akselin kulman perusteella muodostaen toisiojännitteet, joista voidaan ratkaista akselin tarkka asento yhden kierroksen aikana (KUVIO 5). Resolverin signaalista saadaan lisäksi pyörimisnopeus ja inkrementtianturisignaali. (SEW Eurodrive, 1998.)



KUVIO 5. Resolverin periaatteellinen rakenne (mukaiillen SEW Eurodrive. 1998)

## 2.5 EMI-suodatin

Lyhenteellä EMI (Electromagnetic Interference) tarkoitetaan sähkömagneettisia häiriöitä, joita elektroniikkalaitteet aiheuttavat. Häiriöt etenevät säteilemällä tai johtumalla. Johtuvia häiriöitä ovat yhteismuotoinen häiriö, joka aiheutuu loiskapansseista, ja eromuotoinen häiriö, joka aiheutuu hakkurien kytkimistä. (Berman 2008.)

Elektroniikkalaitteet eivät saa lähettää ympäristöönsä liikaa häiriöitä. Toisaalta niiden tulee kestää tietty määrä häiriöitä. Näiden tavoitteiden saavuttaminen edellyttää usein EMI-suodatusta. EMI-suodatin on keloista ja kondensaattoreista muodostuva piiri, joka suodattaa johtuvat häiriöt. Nykyisin lähes kaikki elektroniikkalaitteet sisältävät EMI-suodatinpiirin, mutta aina tämä ei riitä, vaan tarvitaan lisäsuodatinta. (Berman 2008.)

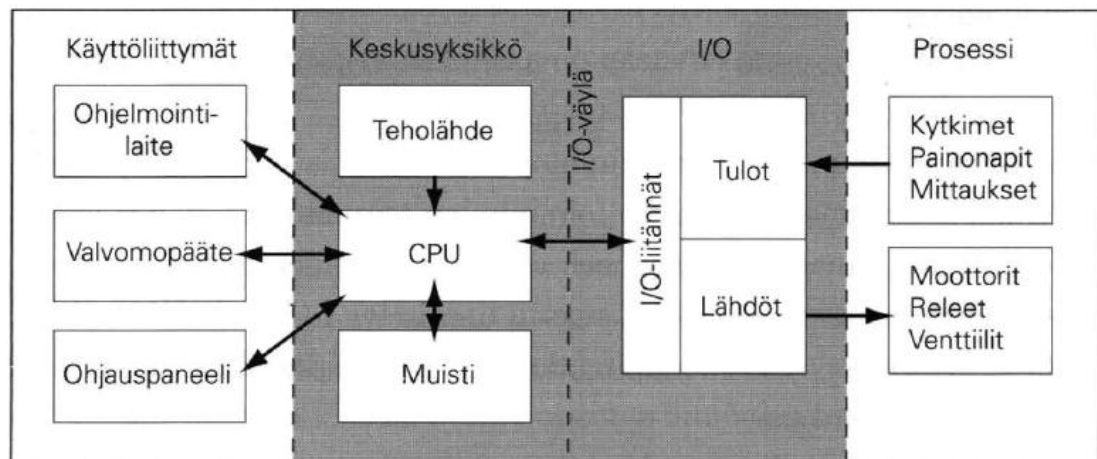
### 3 LOGIikkaOHJAIMET

Automatisoidut ohjausjärjestelmät perustuvat pitkälti työalueen antureilta saatavaan kaksitilaiseen, eli on/off-tyyppiseen tietoon. Myös toimilaitteiden ohjaus on usein kaksitilaista. Tilatietoja yhdistelemällä saadaan aikaan loogisia operaatioita. Kaksitilaiseen tietoon perustuvat ohjausjärjestelmät rakennettiin aiemmin releillä, jolloin loogisten operaatioiden toteutus vaati runsaasti komponentteja ja monimutkaisen johdotuksen. (Keinänen, Kärkäinen, Metso & Putkonen 2000, 241.)

Autoteollisuuden, jossa mekanisoidut tuotantolinjat olivat jo arkipäivää, tarpeet käynnistivät ohjelmoitavien ohjausyksiköiden kehittämisen. Vuonna 1968 General Motors asetti ohjelmoitavalle ohjausyksikölle vaatimukset, joiden pohjalta logiikkaohjaimia ryhdyttiin kehittämään. Vaatimukset olivat seuraavat:

- Laitteen on oltava ohjelmoitavissa, ja se on voitava ohjelmoida uudelleen.
- Laitteen on toimittava moitteettomasti teollisuuden konepajoissa.
- Laitteen on kestävä yhdysvaltalaisen vaatimuksen mukaan 120 V:n vaihtöskösignaaleja, joita annetaan painonapeilta ja rajakytkimiltä.
- Sen lähtöjen tulee kestää sähkömoottoreiden vaatimaa kuormaa sekä käynnistyksissä että jatkuvassa ajossa.
- Hinnan tulee olla kilpailukykyinen jo olemassa oleviin kiinteästi langoitettuihin logiikkalaitteisiin verrattuna. (Keinänen ym. 2000, 241.)

Kehitystyön tuloksena syntyi ohjelmoitava logiikka (Programmable Logic Controller, PLC) eli laite, joka reagoi esimerkiksi antureilta saamaansa tulotietoon ohjelman määräämällä tavalla ja ohjaa lähtöporttejaan tämän mukaan. Logiikan periaatteellinen rakenne on esitetty kuviossa 6. Logiikan käyttäjän laatima ohjelma siirretään logiikan paristolla varmennettuun muistiin, jonka jälkeen se toimii itsenäisenä, yksittäisenä automaatiojärjestelmänä. Nykyajan logiikat pystyvät loogisten operaatioiden ohella usein käsittelemään myös analogisia signaaleja. Usein niissä on sisäänrakennettuna valmiita, kehittyneitä toimintoja, kuten ramppiohjauksia, liikkeenohjauksia ja PID-säätimiä. (Kippo & Tikka 2008, 54–56.)



KUVIO 6. Ohjelmoitavan logiikan järjestelmäarkkitehtuuri (Kippo & Tikka 2008)

Alun perin kantavana ajatuksena oli, että logiikkaohjelmointi ei vaatisi tietokoneohjelmointitaitoja, vaan relekaaviot hallitseva henkilö, joka aiemmin toteutti ohjauksen relekytkennöillä, kykenisi vastedes suoriutumaan logiikan ohjelmoinnista. Tästä syystä logiikoiden kehitystyöhön otettiin olennaiseksi osaksi yksinkertainen käyttöliittymä. (Keinänen ym. 2000, 242.)

### 3.1 Nopea laskuri

Mikäli logiikalla halutaan käsitellä pulssijonoa esimerkiksi pulssianturia tai pulssigeneraattoria käytettäessä, täytyy logiikka olla varustettu erityisellä nopealla laskuriyksiköllä. Logiikan varsinainen ohjelmakierto on aivan liian hidas pulssijonojen käsittelyyn. Nopea laskuri toimii ohjelmakierrosta erillään jopa 2 MHz:n taajuudella, ja sen tulos päivitetään itse ohjelmaan ohjelmakierron mukaisella taajuudella. (Airila 2000, 6:41–42.)

### 3.2 Operointipaneeli

Käyttötarkoituksen mukaan on usein välttämätöntä suoda käyttäjälle mahdollisuus tarkkailla ja ohjata logiikan toimintoja monipuolisemmin kuin parin käyttöpainikkeen ja merkkivalon avulla. Tällöin tarvitaan käyttöliittymää, joka voidaan toteuttaa pc-tietokoneella tai erityisellä operointipaneelilla. Käyttöliittymän sisältävä laite muodostaa ihmisen ja koneen välille rajapinnan, josta käytetään yleisesti lyhennettä HMI (Human

Machine Interface). Operointipaneelit ovat tyypillisesti teollisuudessa käytettyjä kosketusnäyttöjä, joilla voidaan ohjata logiikan muistipaikkoja tai lähtöjä. Niillä voidaan myös esittää logiikan tiloja ja muuttujien arvoja. Operointipaneelien kosketuksen tunnistaminen perustuu pääasiassa paineeseen tai sähköjohtokykyyn (Siemens AG 2011.)



## 4 KONETURVALLISUUS

Koneiden turvallisuutta koskevat asetukset, koneasetus ja käyttöasetus, asettavat suunnitteen samanlaisen vaatimuksen koskien koneiden liikkuvien osien suojaamista: ”Jos liikkuvien osien muoto, voima tai nopeus ovat sellaisia, että koskettamisesta voi aiheutua vaaraa, koskettaminen osien liikkeessä on estettävä joko koneen rakenteilla ja kiinteillä suojuksilla tai varustamalla kone turvalaitteilla, jotka sallivat koskettamisen vain osien ollessa liikkumattomina.” (Siirilä & Kerttula 2009, 58.)

Koneelle, joka voi aiheuttaa ihmiseen kohdistuvan iskun vapaassa tilassa, ilman puristumisvaaraa, on asetettu liikkeen nopeuden raja-arvoksi 200 mm/s. Mikäli puristuminen on mahdollista, raja-arvona on 6 mm/s. Mikäli nämä raja-arvot ylitetään, voi vaaratekijä aiheuttaa tapaturman. (Siirilä & Kerttula 2009, 60.)

### 4.1 Turvarele

Rele on sähköinen kytkin, jonka toimintaa ohjataan erillisellä virtapiirillä. Turvareleellä ohjataan toimilaitteen, kuten servomoottorin, käyttöjännitteen syöttöä, ja ohjaavana virtapiirinä on turvalaite, kuten hätä-seispainike, rajakytkin tai valoverho. Kun turvalaitteessa tapahtuu tilan muutos, ohjaus ja sen myötä jännitteensyöttö, katkevat. Turvarele täytyy kuitata, ennen kuin toimilaitteen ohjaus voi jatkua. Kuittaus voi tapahtua joko käsin tai automaattisesti. (OEM Automatic 2012.)

Turvareleitä käytetään tyypillisesti pienten ja yksinkertaisten järjestelmien ja koneiden turvaratkaisuna. Turvareleen toiminta perustuu kahdennettuihin piireihin, pakkotoimisiin koskettimiin ja itsetestaukseen. Kahdennettu turvapiiri muodostuu samalla ohjaussignaalilla ohjattavista kahdesta erillisestä releestä, joiden kärjet on kytketty sarjaan. Pakkotoimiset koskettimet ovat mekaanisesti toisiinsa kytkettyjä koskettimia, jotka pysyvät toisiinsa nähden samassa asennossa, vaikka yksi kosketin hitsautuisi kiinni. Itsetestaus tutkii releen turvapiirin toimintaa ja ilmoittaa mahdollisesta viasta. (OEM Automatic.)

Nämä menetelmät takaavat, että jännitekatko, sisäinen komponenttivika tai ulkoinen häiriö pysäyttää toiminnan ja estää sen uudelleen käynnistymisen. Markkinoilla on olemassa myös kehittyneempiä releitä, joissa on väyläliitännät ja jotka ovat ohjelmoitavissa esimerkiksi tietokoneella. (OEM Automatic 2012.)

## **4.2 Hätä-seispainike**

Hätä-seispainike on hätäpysäytin, jonka toiminta perustuu pakkotoimisiin avautuviin koskettimiin. Kun pysäyttimeen vaikutetaan, mekaaninen voima pakottaa koskettimet auki, jolloin koneen toiminta pysähtyy. Pysäyttimen on lukkiuduttava tähän asentoon niin luotettavasti, että sen vapautuminen vaatii käsin tehtävän tarkoituksellinen toimenpiteen. Yleensä pysäyttimen painike vapautetaan vetämällä, kiertämällä tai avaimella. Painikkeen tulee olla punainen ja painikkeen taustan keltainen. (Siirilä 2005, 196.)

## **4.3 Valoverho**

Valoverho on valosähköinen turvalaite, joka koostuu lähetin- ja vastaanotinyksiköstä. Lähetinyksikkö lähettää tiheäsäteistä valoa, yleensä infrapunavaloa, jonka vastaanotinyksikkö havaitsee. Mikäli este osuu valoverhon valvomalle alueelle, valonsäde katkeaa ja valoverho lähettää ohjausjärjestelmälle sähköviestin, joka tulkitaan pysähtymiskäskynä. (Siirilä 2005, 326.)

Valoverhon ympäristön taustavalon häiritsevä vaikutus on pyritty minimoimaan valonsäteen moduloinnilla. Tällä tarkoitetaan lähetettävän säteen katkomista alle 10 kHz:n taajuudella. Vastaanotin on määritelty ottamaan vastaan vain lähetintaajuisia valopulsseja. (Fonselius ym. 1994, 32.)

## **4.4 Mekaaninen rajakytkin**

Mekaanisen rajakytkimen toiminta perustuu kytkinelementin sisäisiin koskettimiin yhteydessä olevaan ohjainosaan, johon vaikutetaan mekaanisella liikkeellä. Kosketintoiminnot

voivat olla sulkeutuvia, avautuvia tai vaihtokosketintoimintoja. Koskettimien asennon perusteella ohjaava laite tulkitsee järjestelmän tilan. (Fonselius, Laitinen, Pekkola, Sampo & Välimaa 1994, 21–23.)

Rajakytkintä valittaessa on otettava huomioon suurimmat sallitut jännite- ja virta-arvot, joita ei sovi ylittää. Rajakytkimen yhteydessä suositellaan käytettäväksi vaihtovirtaa, sillä tasavirta aiheuttaa kosketinelementissä pidempikestoisen valokaaren. Rajakytkimet kestävät keskimäärin muutamia miljoonia kytkentäkertoja. (Fonselius ym. 1994, 21–23.)

## 5 PROJEKTIN ETENEMINEN

### 5.1 Laitehankinnat

Työni lähti liikkeelle logiikan valinnalla. Siemensin tuorein logiikkaperhe S7-1200 oli ennestään tuttu laboratoriotöiden parista. Se on yleisesti käytössä prosessiautomaation parissa, alalla, joka on vahvasti edustettuna tällä seudulla. Logiikan pulssilähdöt ja nopeat laskurit mahdollistavat servojärjestelmän pulssiohjauksen, joka on ohjaustapana hyvin selkeä. Siemensin valintaa puolsi myös ohjelmointiympäristö, josta on pyritty tekemään erityisen helppokäyttöinen ja monipuolinen.

Paikallinen alan yritys suositteli Yaskawan servopakettia (moottori ja ohjain) kuultuaan logiikkavalinnastani. Tutustuttuani moottorin esitteisiin heräsi mielenkiinto helposti käyttöön otettavaan ja vieläpä suhteellisen edulliseen moottoriin. Lähetin eri puolille Suomea tarjouspyyntöjä niin moottoreista kuin myös logiikoista melko suurpiirteisillä vaatimuksilla, jolloin sain vertailtavaksi useita hyvin erilaisia kokoonpanoja. Paikallinen tarjous kuitenkin voitti, ratkaisevina tekijöinä hinta ja pulssiohjattavuus.

#### 5.1.1 Servovahvistin: Yaskawa Junma SJDE-02APA-OY

Junma-sarjan servojärjestelmä on erittäin yksinkertainen ja helppo ottaa käyttöön. Servovahvistin (KUVIO 7) ei vaadi vahvistusparametrejä eikä viritystä, vaan vahvistimen algoritmit huolehtivat näistä automaattisesti. Kaapeleiden kytkemisen ohella asetetaan yhdellä ruuvilla sopiva pulssimäärä ja tulotyyppi. Tämän jälkeen järjestelmä on käyttövalmis.



KUVIO 7. Servovahvistin

Vahvistinta ohjataan pulssijonolla, kuten askelmoottorien ohjaimia. Sen käyttöön vaaditaan siis ylemmän tason ohjain, joka syöttää vahvistimelle tarvittavat ohjaussignaalit. Varsinaisen pulssijonon lisäksi vahvistimelle viedään suuntabitti, jonka tila määrittää moottorin pyörimissuunnan. Toinen vaihtoehto on viedä molemmille pyörimissuunnille omat pulssijononsa. Vahvistimessa on sisäänrakennetut kiihdytys- ja jarrutusrampit, joten ylemmän tason ohjaimena voidaan käyttää myös logiikkaa, joka antaa vain pulssien määrän ja taajuuden.

### 5.1.2 Servomoottori: Yaskawa Junma SJME-02AMB41-OY

Järjestelmän toimilaitteena on 200 watin AC-servomoottori (KUVIO 8). Moottorin kiertymää mittaa resolveri. Moottorin kytkeminen vahvistimeen on erittäin yksinkertaista, sillä siinä on vain kaksi kaapelia, joiden liittimet tekevät väärin kytkemisen mahdottomaksi. Moottori on tehoisekseen varsin pienikokoinen.



KUVIO 8. Servomoottori

### 5.1.3 EMI-suodatin: Block R7A-FIZP105-BE

Junman servojärjestelmän mukana toimitettiin erillinen EMI-suodatin, jonka tarpeellisuus tuli hyvin esille laitteistoon tutustuessani. Logiikan ja servojärjestelmän muodostamassa kokoonpanossa servo-ohjaimen hakkurien aiheuttamat häiriöt konkretisoituivat logiikalta tulevan suuntatietobitin epävakautena. Käytännössä tämä aiheutti moottorin epätäsmällisen ja jähmeän toiminnan siinä pyörimissuunnassa, jossa suuntabitti on aktiivisena. EMI-suodattimen lisääminen järjestelmään poisti tämän ongelman.

### 5.1.4 Logiikka: Siemens Simatic S7-1200

Simatic S7-1200 on Siemensin tuorein logiikkatuoteperhe. Logiikoita on tarjolla useampia eri versioita käyttöjännitteen ja I/O-tyyppien mukaan. Sarjan logiikat sisältävät PID-säädön ja liikkeenohjaustoiminnot. Varsinaisesta ohjelmakierrosta riippumattomat nopeat laskuripiirit mahdollistavat pulssiohjauksen sekä pulssiantureiden käytön.

TIA PORTAL -ohjelmointiohjelma tunnistaa sarjan logiikoiden lisäksi myös aiempia logiikkaperheitä, kuten S-400. Se on yhteensopiva myös Simatic-sarjan operointipaneelien kanssa, joten sillä voidaan tehdä varsinaisen logiikkaohjelman lisäksi myös käyttöliittymä. Logiikan ohjelmoinnista on pyritty tekemään mahdollisimman vaivatonta valmiilla kirjastoilla, drag&drop-toiminnoilla sekä kertaalleen tehtyjen toimintojen uudelleenkäytettävyydellä

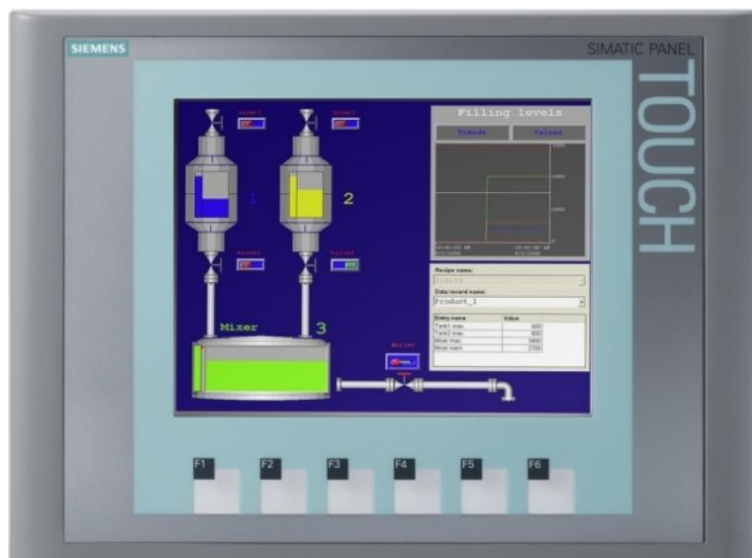
Logiikka on yhteydessä muihin ohjauslaitteisiin, kuten PC-tietokoneeseen, operointipaneeliin ja muihin logiikoihin Siemensin kehittämän Profinet-väylän välityksellä, joka pohjautuu TCP/IP-verkkoon. CPU:n (KUVIO 9) lisäksi hankin samaa tuoteperhettä olevan virtalähteen sekä profinet-kytkimen.



KUVIO 9. Logiikan CPU

### 5.1.5 Operointipaneeli: Simatic HMI KTP600 Basic color

Siemensin 5,7 tuuman operointipaneelissa (KUVIO 10) on kosketusnäytön lisäksi 6 ohjelmoitavaa funktiopainiketta. Kosketusnäyttö on resistiivinen, eli se koostuu kahdesta sähköä johtavasta kalvosta, jotka ovat hieman erillään toisistaan. Kun näyttöä kosketetaan, kalvot painautuvat tältä kohtaa yhteen ja saavat sähkövirran kulkemaan, jolloin laite päättää kosketetun kohdan. Näin ollen paneelia voidaan ohjata millä tahansa esineellä, esimerkiksi kynällä.



KUVIO 10. Operointipaneeli

### 5.1.6 Pulssianturi: Pepperl & Fuchs 58

Junman servokokoonpanossa takaisinkytkentätieto saadaan siniaaltoenkooderilta, joka ohjaimessa muunnetaan digitaaliseen muotoon. Takaisinkytkentäsignaalille ei ole ulostuloa; ainoastaan täysistä kierroksista on tieto vietävissä logiikalle. Opetuslaitteiston ollessa kyseessä koin kuitenkin tärkeäksi saada myös asema- ja nopeustiedon tuotua logiikalle. Tähän tarkoitukseen hankin Pepperl & Fuchsin sarjan 58 pulssianturin (KUVIO 11), jonka kytkin akselistaan kuularuuviin. Anturin tarkkuus on 500 pulssia/kierros, ja ulostulona saadaan kahden kanavan ja nollapulssin lisäksi samat pulssit käänteisinä. Anturin kanavat kytketään logiikan nopeaan laskuriin, jolloin ohjelman kiertoaika ei estä laskentaa.



KUVIO 11. Pulssianturi

### 5.1.7 Muut laitteet

Turvareleeksi valitsin Omronin G9SA-301-mallin, jonka hankin liikkeestä. Turvareleessä on kaksi tulokanavaa, joista voidaan käyttää halutessa vain toista. Lähtöpuolella on kolme sulkeutuvaa ja yksi avautuva kosketinyksikkö.

Laitteistoon sopivan valoverhon löysin koulun varastosta. Verhossa on neljä valokennoa. Käyttöjännite on 15–30 V. Verhon toiminta on NPN-tyyppinen, joten jouduin lisäämään valoverhon ja logiikan väliin kontaktorin.

Kontaktoreiksi valitsin kaksi koululta löytämäni 24 V:n Telemecanique kontaktoria, joissa on kummassakin kolme sulkeutuvaa ja yksi avautuva kosketin. Kontaktoreissa on



diodiyksiköt, jotka luovat kelaan varastoituneelle energialle hallitun purkausreitit. Diodiyksiköissä on lisäksi LED-valot, jotka ilmaisevat kontaktorin tilan.

Johdonsuojakatkaisimet löysin myös koululta. Servomoottorin hetkellinen enimmäisvirta on 3,3 ampeeria, joten servojärjestelmää suojaa neljän ampeerin ja logiikkaa kahden ampeerin katkaisija.

Vanhasta kokoonpanosta otin talteen 4 kpl Omronin D4D-rajakytkimiä. Niissä on sekä avautuva että sulkeutuva kosketinyksikkö. Kaksi kytkintä asensin turvapiiriin ja kaksi ajosekvenssin toimintaa varten.

Liikkeestä hankin pääkytkimen, hätä-seispainikkeen sekä merkkivalolla varustetun painonapin, jota käytin turvareleen kuittauspainikkeena.

## 5.2 Laitteistoon tutustuminen

Koulun laboratoriossa työn alla ollut remontti esti alkuvaiheessa laitteiston rakentamisen sille tarkoitettuun paikkaan, joten päätin tutustua laitteistoon kotona. Kytin logiikan ja servopakettin kotitietokoneeseeni. Liitin logiikan pulssilähtöihin 2 k $\Omega$ :n etuvastukset, jotta sain rajoitettua virran servo-ohjaimelle sopivaksi (7–15 mA). Saatuaani kytkennät valmiiksi ryhdyin tekemään pientä testiohjelmia logiikalle, jolla saisin moottorin pyörimään.

Testaillessani moottoria havaitsin, että se pyöri myötöpäivään aivan kuten pitikin, mutta vastapäivään pyöräminen oli jähmeää ja katkonaista. Aloin epäillä moottorin kuntoa ja vaihdoin logiikasta suuntabitin toiminnan päinvastaiseksi. Havaitsin, ettei moottorin varsinaisella pyörimissuunnalla ole merkitystä, vaan jähmeyttä esiintyy aina kun suuntabitti on ylhäällä, oli pyörimissuunta kumpi tahansa. Seuraavana päivänä sain koululta lainaksi oskilloskoopin, jolla tutkimalla sain selville, että suuntabitti ei pysy vakaasti ylhäällä vaan putoaa vähän väliä hetkellisesti alas. Jokin siis häiritsevi signaalien kulkua. Aikani tutkittuani aiheeseen liittyvää materiaalia tulin siihen tulokseen, että häiriönlähteenä oli servo-ohjaimen hakkuripääteaste, joka katkoo moottorille menevää virtaa kilohertsien taajuudella. Lisäsin verkkojännitteen ja servo-ohjaimen väliin EMI-suodattimen, joka poisti ongelman.

Aikani testailtuani moottorin toimintaa eri ohjelmanpätkillä aloin selvittää, kuinka saan tuotua logiikalle takaisinkytkentätiedon. Hämmästykseni huomasin, ettei se ollut mahdollista. Tällainen asia ei tullut mieleenikään moottoria valitessani. Pidin itsestään selvänä, että koska moottorin enkooderilta tulee takaisinkytkentätieto servo-ohjaimelle, on se myös tuotavissa logiikan hyödynnettäväksi. Koska kyseessä oli opetuslaitteisto, koin tärkeäksi asema- ja nopeustiedon tuomisen käyttäjälle. Ryhdyin harkitsemaan eri vaihtoehtoja asematiedon välittämiseen, päällimmäisenä mielessä optohaarukka ja koodikiekkko. Nykyajan laiterakentelussa on kuitenkin tapana suosia valmiita komponentteja, joten päädyin itsekin hankkimaan valmiin pulssianturin.

### **5.3 Hätäpysäytyspiirin suunnittelu**

Seuraavaksi ryhdyin suunnittelemaan laitteiston turvajärjestelmää. Moottorin tehonsyöttö haluttiin ohjata turvareleen valvoman hätäpysäytyspiirin kautta. Turvareleen valintaan vaikutti saatavuuden ohella se, että olin tähän kyseiseen Omronin malliin ehtinyt pintapuolisesti tutustua opintojeni ohessa. Varsinaisen turvapiirin rakensin kytkemällä sarjaan hätäseispainikkeen, kaksi mekaanista rajakytkintä, jotka toimivat kelkan ajauduttua sen ääriaseentoon, sekä valoverhon. Itse asiassa valoverho ei ole sarjassa muiden turvalaitteiden kanssa, vaan sen ohjaama kontaktori. Kaikki turvalaitteet on kytketty avautuvilla kosketimilla, jolloin myös vikatilanteet, kuten valoverhon käyttöjännitteen katkeaminen, katkaisevat päävirtapiirin syötön.

Rajakytkimet otin talteen kyseisen laitteiston aikaisemmasta kokoonpanosta. Myös valoverho löytyi koululta omasta takaa. Muussa tapauksessa olisi valoverho todennäköisesti jäänyt järjestelmästä pois, sillä kun otetaan huomioon laitteiston tuleva käyttötarkoitus, on turvajärjestelmän osuus kenties hiukan ylimitoitettu. Mitään ongelmaa tästä ei kuitenkaan aiheudu, sillä olkoonkin kyseessä opetuslaitteisto, ei missään tilanteessa ole syytä moottorin ajon aikana päästä käsiksi kelkan toiminta-alueelle. Muutoinkaan turvalaitteiden merkitystä ei voida ylikorostaa.

Saatuani turvajärjestelmän rakennettua totesin sen kyllä katkaisevan jännitteen, mutta ongelmaksi muodostui moottorin hallittu pysäyttäminen. Moottorin tehonsyötöstä vastaa ser-

vo-ohjain, joten kuvittelin ohjaimen jännitteen katkaisemisen riittävän. Ohjain jää kuitenkin toimintaan vielä noin kahdeksi sekunniksi jännitteensyötön katkettua. Moottori pyörii myös tämän ajan, kunnes se pysähtyy kuin seinään. Ongelmani oli siis, kuinka saan moottorin jarrutettua hallitusti pysähdyksiin heti, kun turvapiiri laukeaa, vaikka ohjain jää hetkeksi toimintaan.

Selvisi, että ohjain jarruttaa moottoria 100 ms:n ajan, kun moottorin toiminnan salliva ohjaimen S\_ON-signaali katkeaa. Käytännössä moottori kuitenkin pysähtyy kuin seinään. Laitteiston käyttötarkoituksen vuoksi on oletettavaa, että turvapiiri tullaan laukaisemaan hyvin usein. Tästä syystä halusin moottorille pehmeämmän hätäjarrutuksen, jolloin se täytyy hoitaa logiikan kautta. On huomionarvoista, että tämä rikkoo hätäpysäytyspiirin perimmäisen ajatuksen, sillä piirin toiminnan tulisi olla täysin erillään logiikasta. Laitteistoa säästääkseeni päädyin kuitenkin tähän ratkaisuun.

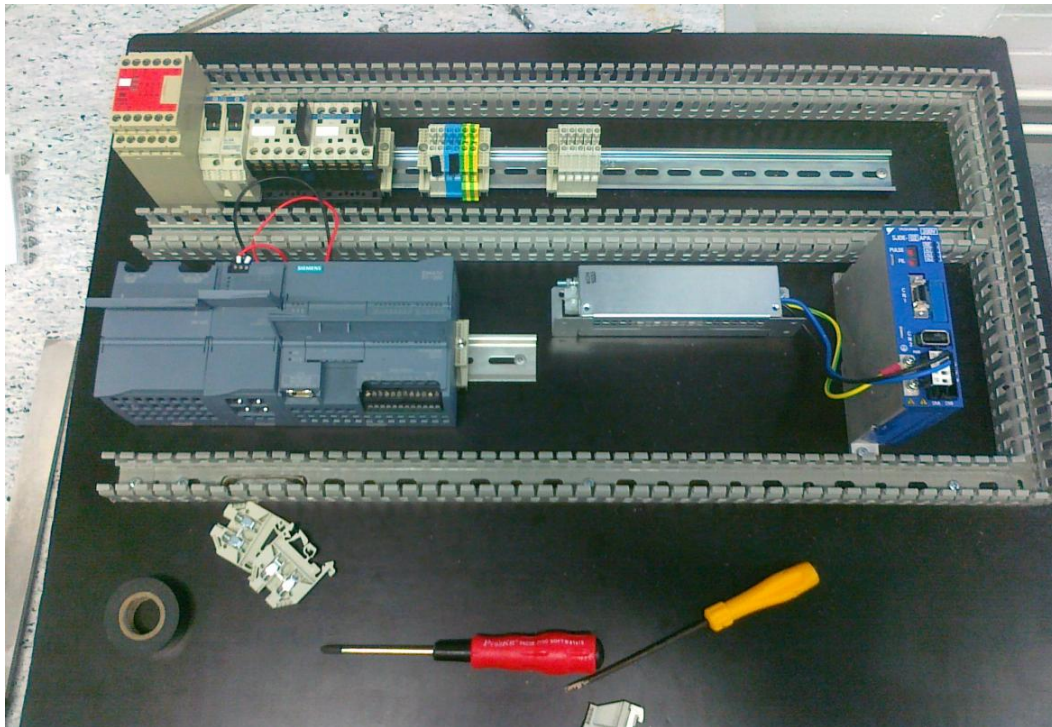
#### **5.4 Rakentaminen**

Varsinainen rakentamisen aloitin purkamalla vanhan testipenkin (KUVIO 12), joka siis tulisi toimimaan uuden laitteiston runkona. Vein vanhat osat huollettavaksi ja maalattavaksi.



KUVIO 12. Penkin vanha kokoonpano

Suunnittelin layoutin sähkökomponenttien sijoittelusta alustalle. Alustana käytin 11 mm:n vanerilevyä. Asensin komponentit alustalle kahteen tasoon: ylemmälle tasolle releet, kontaktorit, johdonsuojat ja riviliittimet ja alemmalle tasolle logiikka, virtalähde, kytkin, EMI-suodatin ja servo-ohjain (KUVIO 13). Jaoin riviliittimet kolmeen ryhmään: ensimmäiseen 230 V:n syötöt ja maadoitukset värillisillä, erottuvilla riviliittimillä, toiseen 24 V:n syötöt ja kolmanteen ohjaussignaalit ja muut ristikytkennät. Kaapelointia suunnitellessa tuli kiinnittää erityistä huomiota servo-ohjaimen I/O-kaapelin ja enkooderikaapelin kulkureitteihin, sillä nämä ovat erittäin alttiita häiriöille.



KUVIO 13. Sähkökomponenttien sijoittelu

Asensin runkokehikon pohjaan kaapelikourut ja kiinnitin kehikon laidoille valoverhoyksiköt. Koulumme laboriomestari teki operointipaneelille telineen, johon tein lisäksi läpiviennit hätä-seispainikkeelle, kuittauspainikkeelle ja pääkytkimelle. Kiinnitin telineen kehikon etureunaan. Asensin paneelin, painikkeet ja kytkimen telineeseen. Kuittauspainikkeessa on merkkivalo, jonka jännitteensyöttö on kytketty turvareleen avautuvien koskettimien kautta. Valo siis ilmoittaa turvareleen launneen. Kytin verkkojännitekaapelin vaihe- ja nollajohtimen pääkytkimen kautta riviliittimille. Suojamaajohdon kiinnitin kehikkoon, josta jatkoin sen maadoituskiskolle. Pääkytkimen alle asensin suojakotelon, jottei kukaan vahingossa pääse koskettamaan jännitteisiä osia.

Seuraavaksi oli vuorossa moottorin kiinnitys. Hankin 2 kpl joustavia kytkimiä, jotka antavat periksi, jos kytkettävät akselit eivät ole tismalleen kohtisuorassa. Toinen kytkimistä piti sorvata moottorin 14 mm:n akselille sopivaksi, sillä paikallisessa liikkeessä oli saatavilla vain 10 mm:n kytkimiä. Entisen moottorin alustaan tehtiin hieman muutostöitä, jotta se saatiin nykyiselle sopivaksi. Porasin alustaan kiinnitysreiät ja tein niihin 4 mm:n kierteet. Kiinnitin moottorin alustaan ja akselin kuularuuviin.

Rajakytkimet asensin kelkan alapuolelle kiskoon, jonka kiinnitin kehikkoon pultilla ja siipimutterilla hienosäädön helpottamiseksi. Asensin kuularuuvin toiseen päähän kierteisen

tapin, jonka kytkin joustokytkimen välityksellä pulssianturin akselille. Pulssianturia varten tehtiin teline, jolla se saatiin sopivan välimatkan päähän kuularuuvista.

Tein 4 kpl ethernet-kaapeleita, joilla kytkin logiikan, paneelin ja PC-tietokoneen Siemensin kytkimeen. Neljännellä kaapelilla saadaan Internet-yhteys tuotua kytkimelle ja sieltä edelleen PC:lle. Tämä edellyttää, että kytkin sallitaan koulun palomuurissa. Loppusilauksena puhdistin lineaarijohteet, rasvasin kelkan ja levitin kuularuuville kuparitahnaa vaimentamaan säröääniä.



KUVIO 14. Laitteisto valmiina

## 5.5 Ohjelmointi

Pyrin suunnittelemaan harjoituksena tehtävän ohjelman sellaiseksi, että se tarjoaisi kattavan yleiskuvan moottorin ohjauksesta, mutta olisi kuitenkin niin yksinkertainen, että sen pystyisi yhden laboratorioharjoituskerran puitteissa tekemään. Ohjelmointiohjelman käyttöönotto-ominaisuus tarjoaa mahdollisuuden ajaa moottoria suoraan ohjelmointiohjelmasta

käsin, kunhan laitemääritykset on tehty. Toimintaa pääsee siis kokeilemaan, vaikkei tekemäänsä ohjelmaa saisikaan syystä tai toisesta toimimaan.

Ajosekvenssi on seuraavanlainen: Aluksi kelkka kotiutetaan, jonka jälkeen ajetaan kohdepisteeseen, joka on alussa 100 mm. Tämän jälkeen palataan kotipisteeseen ja ajetaan uuteen kohdepisteeseen, joka on 100 mm kauempana kuin edellinen. Tätä jatketaan, kunnes saavutaan rajakytkimelle, jolloin asetetaan kohdepisteeksi taas 100 mm ja sekvenssi aloitetaan alusta.

### 5.5.1 Ohjelmoinnin ongelmat

Alun perin projekti piti toteuttaa TIA Portalin V11 Service Pack 2:lla. En saanut kuitenkaan ohjelmaliisenssiä toimimaan, ja myöhemmin selvisikin, ettei Siemens enää julkaise oppilaitoslisenssejä. Tämä oli melkoinen pettymys, sillä jouduin tekemään projektini Service Pack 1:llä tietäen, että tällä laitteistolla tullaan harjoittelemaan vain vanhentuneen tekniikan varassa. Logiikan päivitin versioon 2.2 lataamalla Siemensin verkkosivuilta erityiselle Siemensin SD-muistikortille firmware-päivityksen. Varsinainen päivitys tapahtuu asettamalla muistikortti logiikkaan ja nollaamalla logiikan muisti. Ohjelmointiohjelman vanhempi versio ei kuitenkaan tarjoa tukea kuin päivitykseen 2.1 saakka, mutta onpahan kuitenkin edes logiikka ajan tasalla.

Ohjelmointiohjelman vanhankin version kanssa oli huomattavia ongelmia, joskin sitä sentään pystyi ylipäättään käyttämään. Ongelmallisinta kokoonpanossa oli selkeästi tiedonsiirto, sillä aika ajoin katosi yhteys kaikkiin laitteisiin. Kerran ohjelma oli onnistunut sekoittamaan tietokoneen IP-osoitteen, jonka uudelleenmäärittelyn jälkeen laitteet taas löytyivät. Yleisin ratkaisu tähän ongelmaan oli kuitenkin käynnistää logiikka ja tietokone uudelleen.

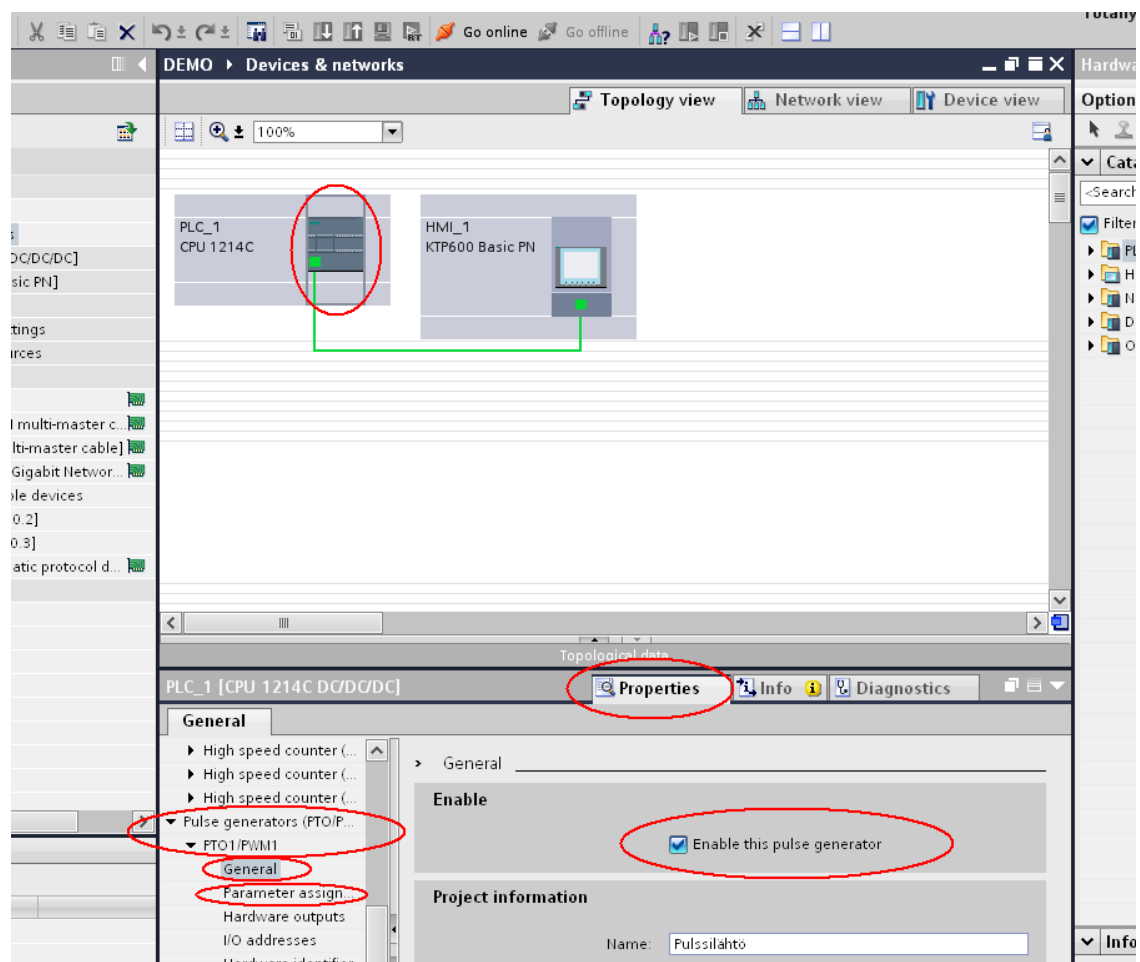
Oli tärkeää myös varmistua logiikan versionumerosta, sillä se on itse määritettävä, vaikka ohjelmointiohjelma sen tunnistaakin myöhemmässä vaiheessa. Ohjelman voi pahimmillaan joutua kopioimaan toimilohko kerrallaan, jotta sen saa siirrettyä toisen version logiikkaan.

Kerran tekemäni ohjelma oli jäänyt tallennettaessa online-tilaan, jolloin se oli siinä myös ladattaessa. Näin ollen ohjelman pystyi lataamaan logiikalle, mutta muutoksia varten se

olisi pitänyt saada asetettua offline-tilaan, joka taas kaatoi TIA Portalin joka kerta. Mikään uudelleenkäynnistäminen ei tätä korjannut, mutta levättyään viikonlopun yli ohjelma suostui taas tottelemaan. TIA Portal tuntuu muutoinkin olevan herkkä kaatumaan, joten projektiin on syytä tallentaa tiheään.

### 5.5.2 Laitemäärittelyt

Projektin luominen alkoi laitteiden, eli tässä tapauksessa logiikan ja paneelin, lisäämisellä. Määritin laitteille IP-osoitteet ja kytkin ne toisiinsa yhdistämällä laitteiden kuvakkeet johdolla (KUVIO 15). Tällä varmistin, että ne ovat samassa aliverkossa.



KUVIO 15. Laitemäärittelyt

Servo-ohjainta ohjataan pulssijonolla, joten pulssigeneraattori täytyi ottaa käyttöön (KUVIO 15). Valittavana oli pulssinleveysmodulaatio (PWM), jolloin käyttäjä määrittelee



pulssisuhteen, tai pulssijono (PTO), jolloin siis PLC määrittelee pulssit. Pulssigeneraattori varaa käyttöönsä yhden nopeista laskureista, joilla on kullakin kiinteästi määritellyt tuloportit. Logiikan tuloista siis kaksi on varattu pulssigeneraattorin laskurille, vaikkei näihin tuloihin mitään kytketäkään.

Pulssianturin käyttöön tarvitaan kaksi nopeaa laskuria laskemaan sijaintia ja nopeutta. Otin laskurit käyttöön ja määrittelin toisen laskemaan pulsseja eli sijaintia ja toisen pulssien taajuutta eli nopeutta. Nopean laskurin ominaisuuksista tulee tarkistaa sen tuloportit, joihin pulssianturi kytketään, sekä osoitevaruus, jonka alkuosoitteeseen tallentuvat lasketut pulssit.

### 5.5.3 Akselin määrittely

Logiikka laskee tarvittavat pulssit määritellyn akselin perusteella. Perusparametreihin määritellään käytettävä pulssigeneraattori ja yksikkö, johon ohjaus perustuu. Valitsin yksiköksi millimetrin. Seuraavaksi siirryin lisäasetuksiin ja valitsin ajon sallivan lähdön, joka on kytketty servo-ohjaimen S\_ON-signaaliin. Annoin valmiustulon olla aina aktiivisena, koska käytin myöhemmin turvareleen signaalia moottorin valmiuslohkossa käytön sallivana signaalina.

Seuraavaksi valitsin pulssimääräksi 1000/kierron, joka täytyy määrittää servo-ohjaimen pulssiasetuksen mukaan. Käytettävässä kuularuuvissa on 4 rinnakkaista 5 mm:n kierrettä, joten valitsin ruuvinnousuksi 20 mm. Suuntasignaali täytyi kääntää, jotta sain kelkan etenemissuunnan oikealle. Äärirajakytkimiä en määritellyt, sillä ne toimivat turvapiirissä logiikasta irrallaan.

Nopeusasetuksissa määritin maksiminopeudeksi 25 000 pulssia/s, joka vastaa tässä kokoonpanossa 500 mm/s. Logiikan pulssigeneraattori kykenee tuottamaan 100 000 pulssia/s, mutta tällaisiin nopeuksiin ei ollut tässä laitteistossa tarvetta. Kiihdytys- ja jarrutusrampin kestoajaksi määrittelin 0.4 sekuntia, joka vastaa  $1200 \text{ mm/s}^2$  kiihtyvyyttä. Servo-ohjaimen hätäpysäytys on kestoltaan 100 ms, jonka halusin hieman pehmeämmäksi, joten asetin hätäpysäytyksen kiihtyvyydeksi  $1500 \text{ mm/s}^2$ , joka kasvatti pysäytysajan noin kolminkertaiseksi.

Seuraavaksi määrittelin referenssi- eli kotipisteen, jossa pulssit nollataan ja laskenta aloitetaan alusta. Pisteeksi valitsi sisemmän vasemmanpuoleisen rajakytkimen, joka sijaitsee logiikan tuloportissa 1.0. Referenssipisteen haulle valitaan taso, joka määrittelee, kuinka tarkasti piste haetaan. Valitsin matalan tason, joka on tässä tapauksessa riittävä. Lopuksi määrittelin kotiutusasetukset. Kotiutussuunnaksi valitsin negatiivisen ja referenssikytkimen sijainniksi alapuolen. Asetin sekä kotipisteen lähestymisen että varsinaisen kotiuttamisen nopeudeksi 400 mm/s.

Sopivat ajonopeudet ja kiihtyvyydet selvitin kokeilemalla. Tavoitteenani oli saada toiminnasta riittävän nopeaa esittelemään servokäytön mahdollisuuksia ja samaan aikaan riittävän hidasta ja pehmeää mahdollistamaan laitteistolle pitkän käyttöiän.

#### **5.5.4 Ajosekvenssin luominen**

Varsinaisen ohjelman laatimisen aloitin tekemällä tagilistan käytettävistä I/O:ista ja muistipaikoista (KUVIO 16). Tämä helpotti ohjelmalohkojen luomista, sillä kaikki tarvittavat tagit sai näin alasvetovalikosta valittavaksi. Retain-valinnalla saadaan kyseinen tagi säilyttämään arvonsa uudelleenkäynnistyksen yhteydessä.

	Name	Address	Retain	Visibl...	Acces...	Corr
1	aja	%M0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	seis	%M0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	nopeus	%MW1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	paikka	%MW3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	Axis_1_Pulse	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	Axis_1_Direction	%Q0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	Axis_1_Drive enabled	%Q0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	Axis_1_Reference point switch	%I1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	turvarele	%I0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	kotiin ajo	%M0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
11	kohteeseen ajo	%M0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12	päätyraja	%I0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
13	alkuhaku	%M5.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
14	Tag_5	%ID1004	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15	Tag_6	%ID1008	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
16	<Add new>		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

KUVIO 16. Tagilista

Tein erillisen käynnistyslohkon, joka suoritetaan vain, kun logiikkaan kytketään virrat. Tässä käynnistyslohkossa nollataan muistibitit ja asetetaan kohdepisteeksi 100 mm. Varsinaisen ajosekvenssin tein omaan päälohkoonsa.

Tein ajosekvenssin liikkeenohjauslohkoilla, joiden käyttö on varsin yksinkertaista. Lohkot tarvitsevat käynnistysbitin, joka asettaa lohkon toimintaan. Bitin asettumisen jälkeen lohko suorittaa toimintonsa alusta loppuun, mikäli toinen lohko ei sitä keskeytä. Vianhakua lohkot tosin saattavat hankaloittaa, sillä niiden sisäistä toimintaa ei pääse tarkkailemaan. Lohkoista saadaan kuitenkin lähtötietoja, kuten esimerkiksi sen, onko liikkeen suoritus kesken vai valmis ja mahdollisia virhekoodeja. Lohkojen toimintojen esittely löytyy liitteenä olevasta pikaoppaasta.

Toteutin ohjelman kombinaatio-ohjauksena. Ohjelmointi poikkesi tottumastani tyylistä, sillä jouduin luomaan liikkeenohjauslohkot ensin ja vasta tämän jälkeen ehdot, joilla lohkoja ohjaavat muistibitit asetetaan aktiivisiksi. Tämä johtui siitä, että halusin käyttää liik-

keenohjauslohkojen lähtötietoja ehtoina ohjauksissani, ja olemattomien lohkojen lähtötietoja ei tietenkään voi käyttää.

### **5.5.5 Operointipaneelin ohjelmointi**

Paneeli ohjelmoidaan samassa ohjelmointiympäristössä kuin logiikka, joten pystyin hyödyntämään aiemmin tekemääni tagilistaa. Halusin myös käyttöliittymästä mahdollisimman pelkistetyn, joten valitsin siihen kaksi painiketta ja neljä I/O-kenttää. Painikkeilla käynnistetään ja pysäytetään moottori. Yksi I/O-kentistä toimii tulona, johon käyttäjä syöttää haluamansa ajonopeuden. Loput kentistä toimivat lähtöinä, jotka ilmoittavat kulloisenkin kohdepisteen, sekä pulssianturilta saadut nopeus- ja asematiedot.

## **5.6 Testaaminen**

Saatuani laitteiston ja ohjelman ajokuntoon ryhdyin testaamaan niiden toimintaa. Aluksi ajo lähti suunnitellusti liikkeelle, mutta hyvin pian ilmeni ongelmia. Mikäli kelkka oli pysähtynyt referenssikytken päälle, lähti moottori ajosekvenssin käynnistyessä hakemaan siitä huolimatta kotipistettä. Kelkka siis lähti peruuttamaan vauhdilla liikealueensa äärirajalle laukaisten rajakytkimellä turvapiirin. Kyseisessä tilanteessa kelkka täytyi palauttaa sallitulle liikealueelle käsin kuularuuvia pyörittämällä tai kelkkaa vetämällä. Ratkaisin ongelman lisäämällä ohjelmaan alkuhakuliikkeen, mikäli referenssikytken on lähtötilanteessa aktiivinen. Liike ajaa kelkkaa 120 mm eteenpäin, jonka selvitin riittävän varmasti ylittämään referenssikytken. Tämän jälkeen ohjelmaa ryhdytään suorittamaan normaalisti.

Jatkaessani testausta havaitsin ohjelman jäävän aika ajoin täydellisesti jumiin. Mitään ei tapahtunut, ennen kuin käynnistin logiikan uudelleen. Syyksi tähän selvisi nopeusmuuttuja, sillä mikäli nopeus jää asettamatta, kun yritetään lähteä ajoon, jää ohjelma ikuisen kiertoon. Ajokäsky kyllä asettaa liikkeenohjauslohkon toimintaan, mutta koska ohjelman etenemisen ehtona on tieto liikkeenohjauslohkon suoritetuksi tulemisesta, ei etenemistä tapahdu. Tämän ongelman ratkaisin asettamalla käynnistymisen ehdoksi, että nopeusmuuttuja on erisuuri kuin nolla.

Itse asiassa ehtona nopeusmuuttujan arvoksi voisi olla erisuuri kuin viisi, sillä tätä pienemmällä nopeuksilla järjestelmä ei toimi. Mikäli tarkoituksenmukaista olisi ajaa pienemmällä nopeudella kuin 5 mm/s, tulisi moottorin ja kuularuuvien välille lisätä alennusvaihteisto.

Yritin pysäyttää kelkan sen myötäsuurtaisen liikkeen ääripäähän eli ts. nollausrajakytkimen päälle. Tämä ei onnistunut, sillä rajakytkin asettaa kotiutuslohkon aktiiviseksi, mikä taas kumoaa pysäytyslohkon. Tämäkin ongelma olisi ollut kierrettävissä lisäämällä ohjelmaan ylimääräinen muistibitti, jota tarvitaan kotiutuskäskyn toteuttamiseksi. Tämä olisi kuitenkin monimutkaistanut ohjelmaa turhan takia, joten annoin asian olla.

Saatuani ohjelman toimimaan halutulla tavalla testasin lopuksi turvapiirin toimintaa. Havaittuani kaikkien komponenttien toimivan halutulla tavalla ryhdyin kartoittamaan mahdollisia vaaranpaikkoja. Havaittiin, että kelkan työalueelle on käytännössä mahdotonta sivullisten joutua vahingossa. Myöskään laitteiston käyttäjä ei pääse horjahtamalla työalueelle kelkan pysähtymättä.

Havaittiin kuitenkin valoverhojen jääneen hiukan turhan korkealle, jolloin niiden alta on mahdollista ujuttaa käsi kelkan työalueelle turvapiirin laukeamatta. Henkilövahinkoa tässä ei kuitenkaan tapahdu, sillä raajan joutuminen tälle alueelle vaatii erityistä tarkoituksellisuutta. Kappaleen jääminen puristuksiin kelkan ja operointiyksikön väliin on kuitenkin mahdollista. Tällaisen tilanteen syntyminen koettiin kuitenkin niin epätodennäköiseksi, että toimenpiteisiin ei ryhdytty, etenkin kun valoverhojen siirtäminen olisi vaatinut kehikon purkamisen.

## **5.7 Harjoituksen ja pikaoppaan laatiminen**

Ainoa työn tilaajalta tullut vaatimus toteutettavalle laboratorioharjoitukselle oli, että sen on oltava suoritettavissa yhden harjoituskerran eli neljän oppitunnin aikana. Pyrin kokoamaan tähän harjoitukseen kuitenkin kaiken olennaisen kyseisen servojärjestelmän perusohjauksesta. Lopputuloksena kehitin kohtuullisen yksinkertaisen ajosekvenssin, jossa on välttämätöntä käyttää paikoitus-, kotiutus- ja pysäytyskäskyä. Opiskelija saa vapauden paikoitustavan valintaan ja ohjelman toteuttamiseen. Myös käyttöliittymän toteuttamisessa

saa käyttää vapautta, kunhan siitä löytyy harjoituksessa esitetyt peruskomponentit, jotka ovat välttämättömiä toteutuksen kannalta.

Pyrin tekemään pikaoppaasta sellaisen, että se auttaa ohjelmoinnissa hyvin alkuun, mutta jättää opiskelijan harteille kuitenkin varsinaisen ohjelman toteuttamisen. Tavoitteeni oli tehdä oppaasta yleisesti kattava eikä kohta kohdalta yksityiskohtaisesti ohjaava. Alkumäärittelyt koin tarpeelliseksi selostaa kohtuullisen tarkkaan ja olen myös oppaassa antanut kaikki tarvittavat arvot, kuten kiihdytykset, jarrutukset jne.

Erikoisempiin tilanteisiin, joihin itse törmäsin ohjelmaa tehdessäni, olen antanut oppaassa vinkkejä. Logiikkaohjelmoinnin perusteet on kuitenkin oltava hallinnassa, sillä varsinaiseen ohjelmointiin en ole ohjeita laatinut, vaan lähinnä esiteltyt ohjelmointiympäristön. Mikäli opiskelija ei hallitse ohjelmoinnin perusteita, ei tämä kuitenkaan estä kokonaan harjoituksen suorittamista, sillä tein harjoituksesta kaksiosaisen.

Ensimmäisessä osassa tehdään tarvittavat asetukset, jotta päästään kokeilemaan moottoria ohjelmointiohjelmasta käsin. Tämä ei vaadi lainkaan itse ohjelman tekoa, ja tarvittavat asetukset on kerrottu pikaoppaassa. Tässä vaiheessa pääsee kuitenkin jo tutustumaan servojärjestelmän toimintaan yleensä ja sen nopeuteen ja tarkkuuteen. Harjoituksen toinen vaihe käsittää varsinaisen ajosekvenssin ja käyttöliittymän luomisen.

Huolimatta ohjelman yksinkertaisuudesta sen toteuttaminen vaatii tiettyjä kikkailuja ja lohkojen keskinäisiin yhteyksiin perehtymistä. Vielä kun ohjelmointiohjelma on melko altis kaatumaan, suosittelen, että harjoituksen läpäisyvaatimukseksi asetetaan ensimmäisen vaiheen suorittaminen.

## 6 POHDINTA

Opinnäytetyön tavoitteena oli tuottaa tuleville opiskelijoille servopaikoitusharjoitus sekä asianomainen laitteisto. Projektiin valitsemani servojärjestelmä osoittautui hyvin mielenkiintoiseksi vaihtoehdoksi automaattisen virityksen ja pulssijono-ohjauksen myötä. Tällaista servokäyttöä ei koulullamme ollut entuudestaan lainkaan.

Servojärjestelmän yksinkertaisuus mahdollisti täydellisen käyttöönottoharjoituksen, jossa opiskelija pääsee laatimaan työkierron aina laitealustuksista saakka. Ainoastaan kaapelointi jää harjoituksen ulkopuolelle. Järjestelmän viritysvapaus suo laajalle opiskelijajoukolle mahdollisuuden hyödyntää harjoitusta, sillä säätötekniikan tuntemusta ei tarvita. Ainoastaan logiikkaohjelmoinnin perusteet tulisi olla hallussa, joskaan tämäkään ei ole välttämätöntä. Jaoin harjoituksen tätä varten kahteen osaan, jossa ensimmäisessä pääsee tutustumaan liikkeenohjauksiin ilman ohjelman luomista.

Laitteiston tarkoitus on tutustuttaa opiskelija servojärjestelmän ohella ohjelmitaviin logiikoihin. Harjoitus jättää opiskelijalle pohdiskeltaviksi servotekniikan ja pulssijono-ohjauksen periaatteet. Laitteiston kokonaisvaltaisen käyttöönoton jälkeen opiskelija pääsee havaitsemaan servokäytölle ominaisen tarkkuuden ja nopeuden.

Itselleni kyseinen viritysvapaa servojärjestelmä oli täysin uusi tuttavuus. Se toi arvokkaan lisän mahdollisuuksiin ratkaista tulevaisuuden haasteita. Tästä huolimatta syvensin projektin aikana ymmärrystäni perinteistä servotekniikkaa kohtaan laajasti.

Mielenkiintoista oli myös lähestymistapa, eli jouduin projektin joka vaiheessa ajattelemaan toisen osapuolen eli tulevaisuudessa harjoitusta tekevän opiskelijan näkökulmasta. Tästä on varmasti hyötyä tulevaisuudessa asiakkaalle toteutettavissa projekteissa, projektia tai asiakasta sen kummemmin määrittelemättä. Toki toisen osapuolen näkökulma oli läsnä laitteiston rakentamisessa ja turvallisuussuunnittelussa, mutta erittäin keskeisessä asemassa se oli harjoituksen ja käyttöoppaan laatimisessa. Mahdollisuudet selvittääkseni minun oli tutustuttava valitsemaani logiikkaan perin pohjin.

Koneturvallisuus muodostui merkittäväksi osa-alueeksi tässä opinnäytetyössä. Halusin paitsi tehdä laitteistosta käyttäjälle turvallisen, myös tutustua käytännössä koneasetuksen ja muiden säädösten ja direktiivien asettamiin vaatimuksiin. Laitteisto ei kuitenkaan täysin täytä näitä vaatimuksia, sillä opetuslaitteiston ollessa kyseessä on pikemminkin tärkeää päästää käyttäjä käsiksi eri komponentteihin kuin suojata häntä niiltä. Vahingoista aiheutuvat tapaturmat olen kuitenkin pyrkinyt eliminoimaan.

Laitteistoa voidaan tulevaisuudessa tarvittaessa laajentaa esimerkiksi moniakselisovellukseksi. Mikäli halutaan säilyttää pulssijono-ohjaus, voidaan laitteistoon lisätä toinen servovahvistin ja -moottori, jolloin näiden keskinäinen toiminta määritellään logiikkaohjelmassa. Mikäli halutaan useampia akseleita on hankittava erillinen servo-ohjain ja tälle yhteensopivat vahvistimet. Muutoinkin laitteiston jatkojalostusmahdollisuudet ovat laajat. Vaikka toiminta on tällä hetkellä puhtaasti opetuksellinen, se tarjoaa hyvän perustan eri hyötykäyttösovelluksille.

Pidän työtä kokonaisuudessaan erittäin onnistuneena. Vaikkakin insinööriytyölle tyypilliset matemaattiset ongelmat ja mitoitukset loistivat poissaolollaan, antoi työ hyvät eväät laitesuunnitteluun. Ennen kaikkea turvallisuuskysymykset ja käyttäjän näkökulma olivat työn opettavaisimpia puolia. Ainoa varsinainen epäonnistuminen oli logiikan valinta, sillä päivitysmahdollisuuksien puuttuessa täytyy opiskelijoiden sietää pikkuvikoja kuhisevaa ohjelmointiohjelmaa. Tämä seikka tuli kuitenkin myös opettajillemme täydellisenä yllätyksenä, sillä kyseinen valmistaja on perinteisesti ollut hyvin vahvasti edustettuna oppilaitoskäytössä. Tästä epäkohdasta huolimatta uskon, että laitteisto tulee tarjoamaan arvokkaan lisän mekatroniikan laboratorioharjoituksiin.



## LÄHTEET

Airila M. 2000. Mekatroniikka. 6. painos. Helsinki: Oy Yliopistokustannus/Otatieto.

Berman M. 2008. All About EMI Filters. TDK-Lambda. Pdf-tiedosto. Saatavissa: [http://us.tdk-lambda.com/lp/ftp/other/all\\_about\\_emi\\_epmag.pdf](http://us.tdk-lambda.com/lp/ftp/other/all_about_emi_epmag.pdf). Luettu 21.3.2012.

Fonselius, J., Laitinen, E., Pekkola, K., Sampo, A. & Välimaa, T. 1994. Koneautomaatio. Anturit. 3. Painos. Helsinki: Painatuskeskus Oy.

Fonselius, J., Rinkinen, J. & Vilenius, M. 1998. Servotekniikka. Helsinki: Oy Edita Ab.

Halme, J. & Parikka, R. 2005. AC-servomoottori – rakenne, vikaantuminen ja havainnointimenetelmät. VTT, tutkimusraportti. Pdf-tiedosto. Saatavissa: [http://virtual.vtt.fi/virtual/proj3/prognos/prognos/pdf/servomoottori\\_rakenne\\_vikaantuminen&havainnointi.pdf](http://virtual.vtt.fi/virtual/proj3/prognos/prognos/pdf/servomoottori_rakenne_vikaantuminen&havainnointi.pdf). Luettu 30.12.2011.

Keinänen, T., Kärkkäinen, P., Metso, T. & Putkonen, K. 2000. Logiikat ja ohjausjärjestelmät. Helsinki: Werner Söderström Oy.

Kippo, A. & Tikka, A. 2008. Automaatiotekniikan perusteet. Helsinki: Oy Edita Ab.

Lewin, C. 2007. Tuning servomotors. PMD, Inc. Www-dokumentti. Saatavissa: [http://www.pmdcorp.com/news/articles/html/Tuning\\_Servomotors.cfm](http://www.pmdcorp.com/news/articles/html/Tuning_Servomotors.cfm). Luettu 28.3.2012.

Nelson, D. 1999. CNC Toolbox. 2. painos. Bristol, WI: Aero Publishing.

OEM Automatic. Yleistä turvareleistä. OEM Finland Oy. Pdf-tiedosto. Saatavissa: [http://opisweb.oemautomatic.se/Navigate/Document.aspx?x&Catalog=productfi\\_kes&DbDocId=847633&Style=BasePacToPdf](http://opisweb.oemautomatic.se/Navigate/Document.aspx?x&Catalog=productfi_kes&DbDocId=847633&Style=BasePacToPdf). Luettu 29.3.2012.

PC Control Ltd. 2008. Incremental Encoders. Www-dokumentti. Saatavissa: [http://www.pc-control.co.uk/incremental\\_encoders.htm](http://www.pc-control.co.uk/incremental_encoders.htm). Luettu 15.2.2012.

SEW Eurodrive. 1998. SEW-vaihdemoottoreiden käyttöratkaisuja.  
Siemens AG. 2011. SIMATIC HMI Panels. Pdf-tiedosto. Saatavissa: [http://www.automation.siemens.com/salesmaterial-as/brochure/en/brochure\\_panels\\_en.pdf](http://www.automation.siemens.com/salesmaterial-as/brochure/en/brochure_panels_en.pdf). Luettu 25.3.2012.

Siirilä, T. 2005. Koneturvallisuus. Ohjausjärjestelmät ja turvalaitteet. Helsinki: Inspecta Koulutus Oy.

Siirilä, T. & Kerttula, T. 2009. Koneturvallisuuden perusteet. Espoo: Opiks-Tiimi Oy.

## Laboratorioharjoitus:

### Pulssiohjatun sähköservojärjestelmän käyttöönotto

#### Tietoa laitteistosta

Oheinen laitteisto tarjoaa mahdollisuuden tutustua Yaskawan parametrintivapaaseen servojärjestelmään, Siemensin S7-1200 tuoteperheeseen ja servon pulssijono-ohjaukseen. Servomoottori pyörittää kuularuuvia, joka kuljettaa kelkkaa liikeradalla. Liikeradan alapuolella sijaitsevista turvakytkimistä kahta sisempää käytetään ohjelmoinnissa. Vasemmanpuoleinen toimii paikoituksen referenssipisteenä, ts. pulssien laskenta aloitetaan nolasta tässä pisteessä.

Laitteistoa käytetään operointipaneelilta. Paneelin vierestä löytyvät pääkytkin, hätä-seis-painike ja turvareleen kuittauspainike. Turvarele täytyy kuitata aina hätäpysäytyspiirin lauettua, jolloin painikkeen merkkivalo syttyy.

Laitteisto pitää sisällään turvareleen valvoman hätäpysäytyspiirin, joka pysäyttää moottorin ja katkaisee servo-ohjaimen ja moottorin käyttöjännitteen. Hätäpysäytyspiiri koostuu hätä-seis-painikkeesta, valoverhosta ja kelkan liikeradan päädyissä sijaitsevista rajakytkimistä. Turvareleeltä on tuotu signaali logiikalle, jota voidaan hyödyntää ajon sallinnassa, mutta itse hätäpysäytyspiiri on täysin riippumaton logiikan toiminnasta.

Servomoottori sisältää luonnollisesti enkooderin, jonka tieto viedään servo-ohjaimelle. Tässä kokoonpanossa enkooderitietoa ei kuitenkaan saada ohjaimelta ulos. Tästä syystä kuularuuvin oikeanpuoleiseen päähän on asennettu pulssianturi, josta saadaan kelkan asema- ja nopeustieto hyödynnettäväksi.

#### Harjoitus

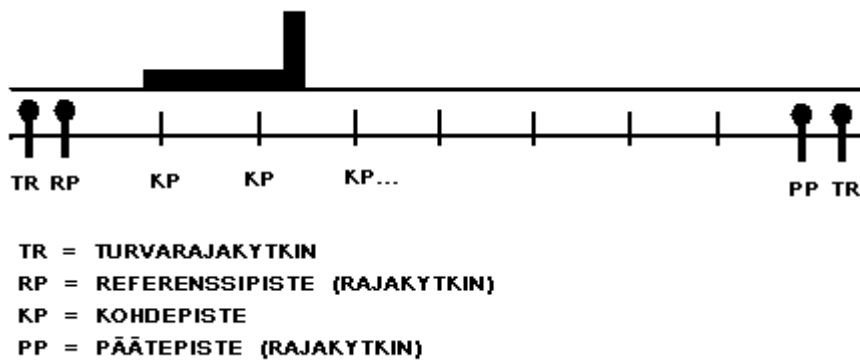
##### Vaihe 1

Tutustu laitteistoon ja pikaoppaaseen ja tee TIA-Portalissa tarvittavat asetukset, jotta saat moottorin toimintavalmiuteen. Kun olet määritellyt pyöritettävän akselin, löydät **Project Tree**-valikosta kyseisen akselin alta **Commissioning**-osion, jossa pääset kokeilemaan

moottoria suoraan ohjelmointiohjelmasta käsin. Kokeile eri liikkeenohjauskäskyjen toimintaa.

## Vaihe 2

Tee logiikalla ajosekvenssi, joka kuljettaa kelkkaa kotipisteestä kohdepisteeseen ja takaisin. Ensimmäisellä kierrolla kohdepiste sijaitsee 100 mm päässä kotipisteestä. Kun kohdepiste on saavutettu, siirretään seuraava kohde 100 mm kauemmaksi. Näin jatketaan, kunnes saavutetaan päätyrajakytkin, jolloin ajosekvenssi aloitetaan alusta.



Ajon sallintasiignaali otetaan turvareleeltä. Ajo tulee voida keskeyttää, jolloin ajetaan kierros loppuun ja pysäytetään moottori. Uudelleenkäynnistyksen yhteydessä ajo aloitetaan aina alusta. Tee operointipaneelille käyttöliittymä, jossa on käyttöpainikkeiden lisäksi ajonopeuden valinta sekä näyttämät nopeudelle ja asemalle, jotka saadaan pulssianturilta.

## Tuotos

Esittele opettajalle/valvojalle valmis ohjelma toiminnassa. Tee seloste, jossa esittelet yleisesti pulssijono-ohjatun servojärjestelmän toimintaperiaatteen. Kerro pulssinleveysmodulaatiosta (millainen pulssijono tuotetaan, kun halutaan liikkua pisteeseen X nopeudella Z). Raportoi harjoituksen etenemisestä ja kerro mahdollisista vastaan tulleista ongelmista. Liitä selosteeseen ohjelmalistaus tulostettuna

# **TIA PORTAL -PIKAOPAS**

Pulssijono-ohjatun servomoottorin käyttöönotto

Mikko Kavonius

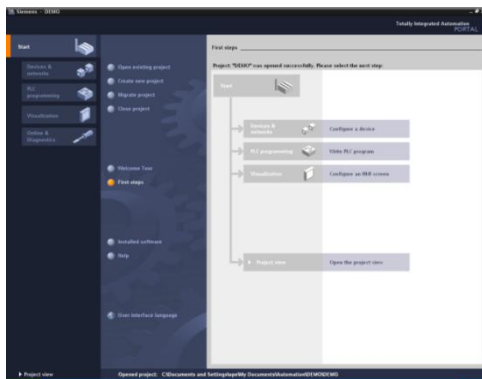
19.3.2012

## Sisältö

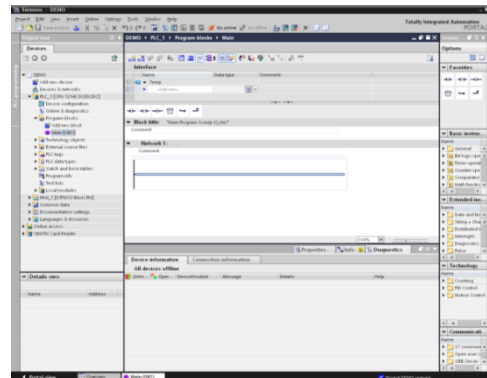
Projektin aloittaminen .....	1
Laitteiden lisääminen & alustaminen.....	1
Akselin määrittely.....	3
Ohjelmointi.....	4
Ohjelmointiympäristö.....	4
Apumuistit & I/O .....	6
Liikkeenohjaus .....	6
Lohkojen kuvaukset.....	6
Vinkkejä.....	9
Suurnopeuslaskurit .....	9
Vinkkejä.....	9
Operointipaneeli.....	10
Projektin suoritusvaiheet.....	10

## Projektin aloittaminen

TIA Portalissa on kaksi eri näkymää: Portaalinäkymä, jossa projekti voidaan luoda wizard-tyyppisesti ja perinteisempi projektinäkö, jossa luodaan varsinaiset ohjelmalohkot ym. Projektinäkössä voi tehdä koko projektin, mutta jos portaalinäkymä tuntuu selkeämmältä, voi sen kautta siirtyä vaiheesta seuraavaan. Ohjelmassa on muitakin Wizard-toimintoja, esim. operointipaneelin liittämisen yhteydessä. On pitkälti makuasia, käyttääkö niitä vai ei. Aluksi on syytä tarkistaa logiikan (PLC) ja operointipaneelin (HMI) IP-osoitteet. **Online & Diagnostics:n** alta löydät kohdan **Accessible Devices**. Tämä hakee verkkoon kytkettyjen **Profinet**-laitteiden IP-osoitteet. Ota osoitteet talteen, sillä joudut määrittämään ne projektiin manuaalisesti.



Kuva 1: Portal View



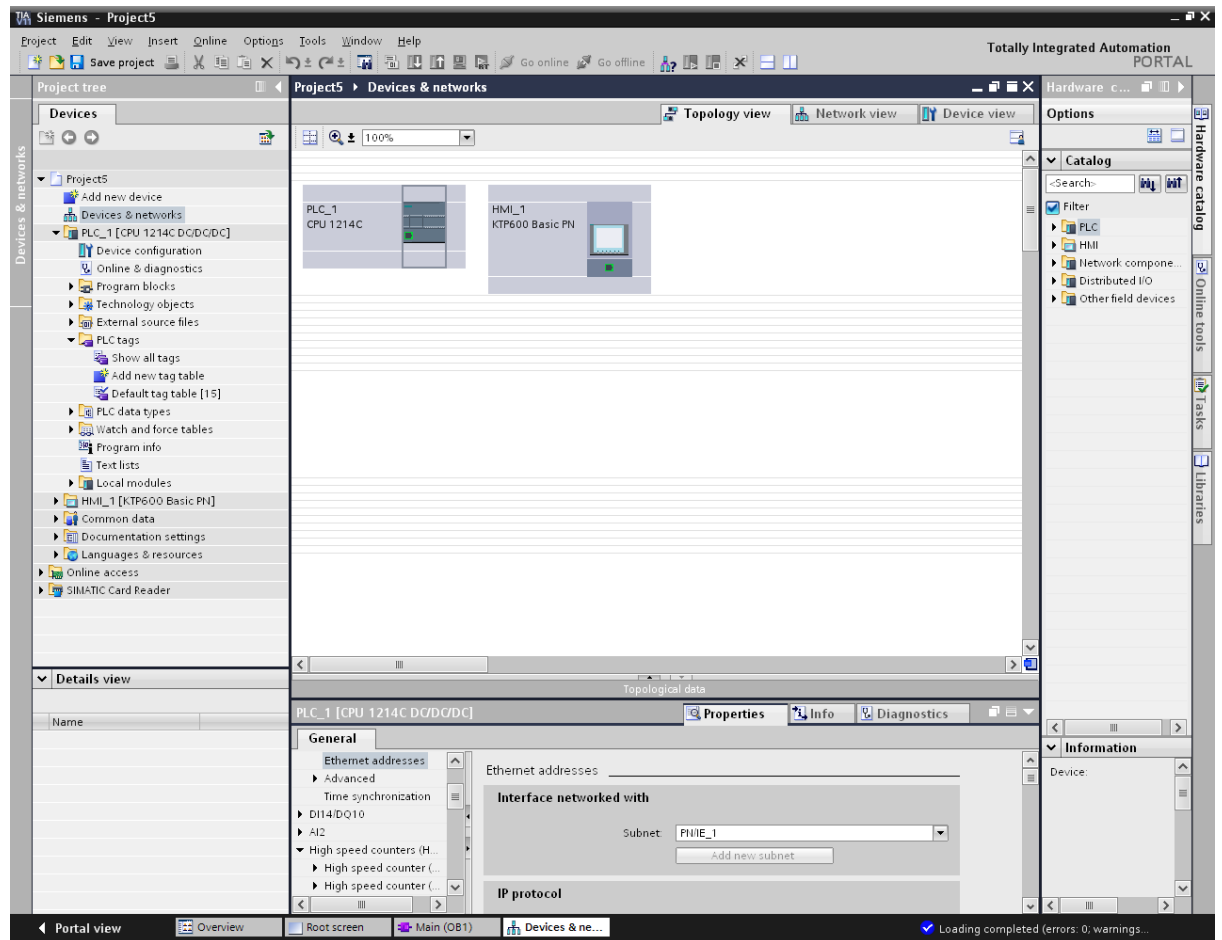
Kuva 2: Project View

## Laitteiden lisääminen & alustaminen

Projektin luominen lähtee liikkeelle logiikan ja paneelin valinnalla. Lisää ensin logiikka, jolloin voit paneelin lisäämisen yhteydessä aukeavassa wizardissa liittää sen valitsemaasi logiikkaan.

PLC = **SIMATIC S7-1200 – CPU – CPU1214C DC/DC/DC – 6ES7 214-1AE30-0XB0 – v2.1**

HMI = **Simatic Basic Panel – 6” Display - KTP600 Basic PN**



Kuva 3: Laitteistomäärittelyt

Määrittele nyt projektiin IP-osoitteet. **Devices & Networks**-osiosta (kuva 3) pääset käsiksi laitekoonpanoon ja laitteiden asetuksiin. Klikkaa PLC:n kuvaa ja valitse alhaalta **Properties**-välilehti. Kohdasta **Profinet interfaces** saat asetettua alussa talteen ottamasi IP-osoitteen.

Pulssilähdön saat käyttöön kohdasta **Pulse generators**. Valitse parametrintikohdasta pulssigeneraattoriksi **PTO** (Pulse train output), jolloin CPU määrittelee pulssien leveyden ja tiheyden. PTO varaa käyttöönsä seuraavan vapaan nopean laskurin (joka varaa kaksi tu-lopporttia) ja kaksi lähtöporttia, joista toinen on pulseille ja toinen suuntabitille.

Pulssianturia varten tarvitset kaksi nopeaa laskuria (**High speed counter**), joista toinen laskee kelkan asemaa ja toinen nopeutta. Ota käyttöön **HSC2** ja valitse laskentatavaksi **Count**. Kohtaan **Operating phase** valitse **AB quadrature phase 1X**. Ota käyttöön **HSC3**,

asetukset kuten edellä, mutta laskentatavaksi **Frequency**. Valitse vielä aikaväliksi 0,1 sekuntia. Aseta vielä operointipaneelille IP-osoite, samalla tavalla kuten CPU:lle.

### Akselin määrittely

Jotta logiikka osaisi syöttää servo-ohjaimelle relevanttia pulssijonoa, on määriteltävä muutama perusasia kokoonpanosta.

TARKISTA ALUKSI ETTÄ SERVO-OHJAIMEN PULSE-VALINTARUUVI ON KÄÄNNETTYNÄ KOHTAAN 8! Tämä kertoo ohjaimelle, että pulsseja tarvitaan 1000/kierros.

**Project tree**:sta löydät **PLC\_1**:n alta kohdan **Technology objects**. Lisää uusi objekti. Valitse **TO\_AXIS\_PTO**. Valitse käyttöön ottamasi pulssigeneraattori ja tarkista, että pulssilähtönä on **Q0.0** ja suuntalähtönä **Q0.1**. Siirry kohtaan **Extended parameters**. Tee seuraavat muutokset:

**Select enable output** = Q0.2

**Distance per motor revolution** = 20 mm

**Invert direction signal**

**Acceleration** = 1200 mm/s<sup>2</sup>

**Deceleration** = 1200 mm/s<sup>2</sup>

**Emergency deceleration** = 1500 mm/s<sup>2</sup>

**Input reference point switch** = I1.0

**Select level** = Low level (määrittelee, kuinka tarkasti nollapiste haetaan)

**Approach velocity** = 400 mm/s

**Referencing velocity** = 400 mm/s

**Technology objects**-valikkoon on nyt ilmestynyt määrittelemäsi **Axis\_1**. Sen alta löytyy kohta **Commissioning**, jolla pääset ajamaan moottoria suoraan. Tätä voidaan hyödyntää haettaessa sopivia nopeus- ja kiihdytysasetuksia. Kokeile liikkeenohjaustoimintoja, saat samalla selville onko yhteys logiikan ja pc:n sekä paneelin välillä kunnossa.

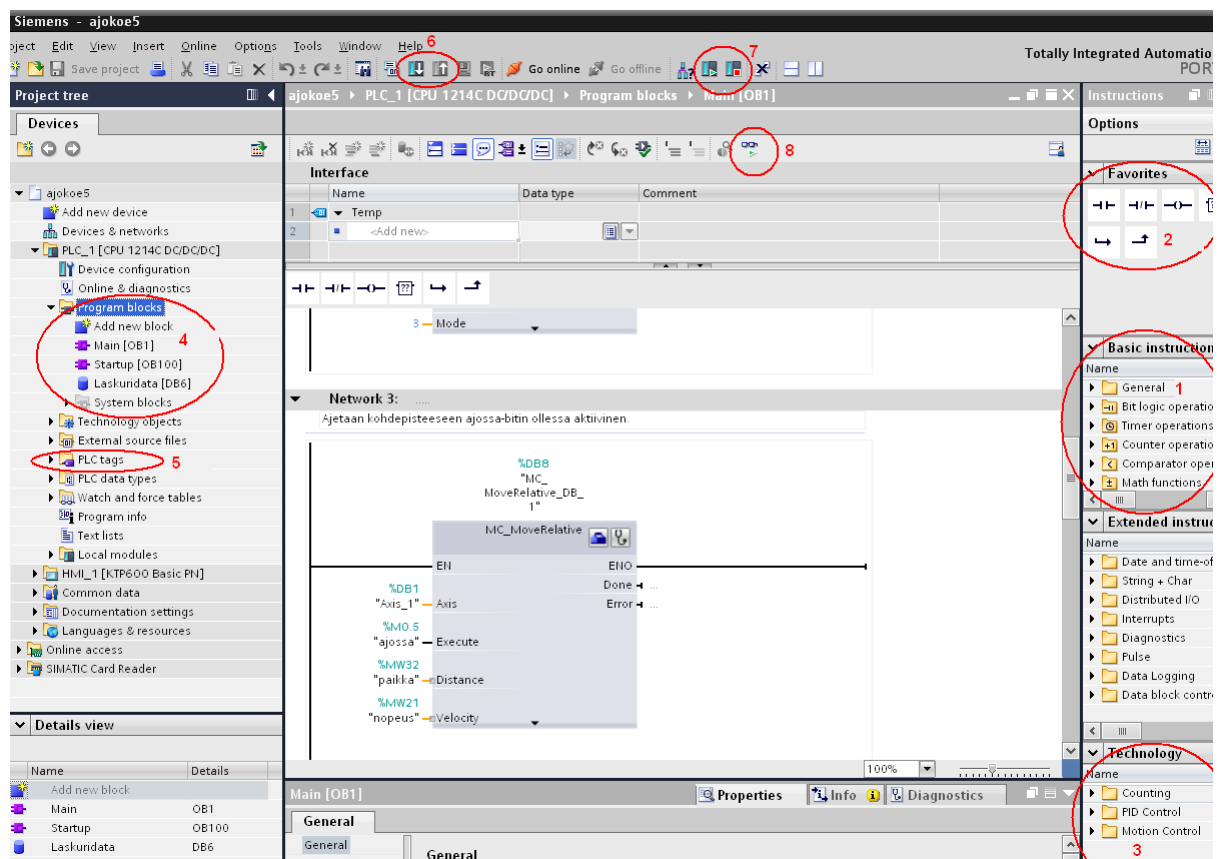


## Ohjelmointi

Nyt voit aloittaa varsinaisen ohjelman laatimisen. Peruslohkojen ohella tarvittavat liikkeenohjauslohkoja ja suurnopeuslaskurin ohjauslohkoa.

## Ohjelmointiympäristö

Ohjelmointi tapahtuu Project view-tilassa. Seuraavassa esiteltynä tärkeimmät kirjastot ja toiminnot:



Kuva 4: PLC-ohjelmointiympäristö

1. Perustyökalut, kuten loogiset operaatiot, ajastimet, laskurit, vertailijat, matemaattiset operaatiot ym.
2. Suosikkeihin voit lisätä usein käyttämiäsi operaatioita, jolloin ne ovat heti käsillä.
3. Teknologyökaluista löydät mm. nopean laskurin ja moottorin ohjaukseen tarvittavat toimilohkot.

4. Täältä voit tehdä sekä selata eri ohjelmalohkoja. Lohkoja on neljää päätyyppiä:

- Organization block (OB)

Organisaatiolohkojen alta löytyy päälohkot (main), joihin perusohjelma luodaan ja käynnistyslohkot (startup), jotka suoritetaan vain kerran käynnistämisen yhteydessä. Käynnistyslohkossa on hyvä nollata muistipaikat ja alustaa muuttujille alkuarvot. Lisäksi organisaatiolohkojen alta löytyy lukuisia eri keskeytysrutiineja.

- Function block (FB)

Toimintolohko, jota voidaan kutsua päälohkosta. Näihin on hyvä tehdä mahdolliset aliohjelmat. FB:llä on oma datalohko, johon tiedot tallentuvat, ja ne ovat toimintolohkon suorittamisen jälkeen saatavissa.

- Function (FC)

Muutoin sama kuin FB, mutta ei sisällä datalohkoa, joten tiedot eivät tallennu.

- Datablock (DB)

Datalohkoihin tallennetaan käyttötietoa, kuten mm. muuttujien arvoja. Datalohkoja on globaaleja sekä paikallisia. Globaalit lohkot ovat käytettävissä missä tahansa ohjelmalohkossa. Paikalliset lohkot liittyvät valittuun FB:hen, eivätkä ole muiden ohjelmalohkojen saatavilla.

5. Täältä saat auki tagilistan, joka sisältää I/O:n, muistibitit, muistisanat, suurnopeuslaskurien osoitteet jne. On suositeltavaa projektin alussa pohtia mitä tageja tulet tarvitsemaan ja tehdä tagilista ensimmäiseksi. Tämä helpottaa ohjelmointia huomattavasti, kun tagit löytyvät nimeltä ja ne voi valita alusvetovalikosta.

6. Ohjelman lataus PLC:lle ja PLC:stä.

7. Logiikan start ja stop.

8. Monitorointi. Tämä aktiivisena näet logiikalle lataamaasi ohjelmaa suorittaessasi näytöllä eri lohkojen kulloisenkin tilan. Helpottaa vianhakua.

## Apumuistit & I/O

Voit käyttää muistibittejä kaksitilaisen tiedon tallentamiseen. Kokonaislukuja voit tallentaa muistisanoihin. Tagilistaa luodessasi lisää sinne I/O:n lisäksi apumuistit ja muista valita tyyppi oikein. Vaihtoehtoisesti voit käyttää datalohkoja tiedon tallentamiseen. Seuraavassa esimerkit apumuistien ja I/O:n tunnuksista:

- Muistibitit = M0.0
- Muistisanat = MW1, älä käytä vierekkäisiä MW:itä, kuten MW1 ja MW2
- Tulot = IO.0
- Lähdöt = Q0.0

## Liikkeenohjaus

Määriteltyä akselia ohjataan **Motion Control**-lohkoilla. Lohkot löytyvät **Project View**-tilassa **Instructions**-välilehden **Technology**-osiosta. Jokaisesta lohkoista löytyvään **Axis**-nastaan viedään ohjattavan akselin nimi, joka määriteltiin **Technology Objects**-osiossa. Jokainen liikkeenohjauslohko varaa itselleen datalohkon tietojen tallennusta varten. Lohkoista saadaan tietoa liikkeensuorituksen vaiheista, kuten esim. **Busy**, joka kertoo liikkeen olevan suorituksessa parhaillaan ja **Done**, joka kertoo liikkeen olevan suoritettu.

## Lohkojen kuvaukset

### MC\_POWER

Vapauttaa/lukitsee moottorin. **Enable**-nastan kytkeminen pois päältä kesken ajon pysäyttää moottorin **StopMode**-tilan mukaisesti.

**Enable:** FALSE – Lukitsee akselin.  
TRUE – Vapauttaa akselin.

**StopMode:** 0 – Häätäpysäytys. Jarruttaa moottoria määritellyn hätäseis-profiilin mukaisesti

1 – Välitön pysäytys. Pysäyttää moottorin heti ilman jarrutusta.

## **MC\_RESET**

Virhetilanteiden kuittaus.

**Execute:** Suoritus nousevalla reunalla

## **MC\_HOME**

Sovittaa paikkakoordinaatit todelliseen fyysiseen sijaintiin. (Käytä mode 3)

**Mode:** 0 – Suora absoluuttinen kotiutus

1 – Suora suhteellinen kotiutus

2 – Käyttäjä ajaa itse akselimääritysten mukaiseen kotipisteeseen

**3 – Automaattinen ajo akselimääritysten mukaiseen kotipisteeseen**

## **MC\_HALT**

Pysäyttää moottorin jarrutusprofiilin mukaisesti ja päättää ajossa olevan komennon.

**Execute:** Suoritus nousevalla reunalla

## **MC\_MOVEABSOLUTE**

Paikoitus määriteltyyn absoluuttiseen pisteeseen.

**Execute:** Suoritus nousevalla reunalla

**Position:** Kohdepiste reaalilukuna

**Velocity:** Ajonopeus reaalilukuna. (Ei välttämättä saavuteta, riippuen kohdepisteen etäisyydestä ja määritellyistä kiihdystys- ja jarrutusajoista)

**MC\_MOVERELATIVE**

Suhteellinen paikoitus määriteltyyn etäisyyteen nykypisteestä.

**Execute:** Suoritus nousevalla reunalla

**Distance:** Kohdepisteen etäisyys reaalilukuna.

**Velocity:** Ajonopeus reaalilukuna. (Ei välttämättä saavuteta, riippuen kohdepisteen etäisyydestä ja määritellyistä kiihdystys- ja jarrutusajoista)

**MC\_MOVEVELOCITY**

Ajo määritellyllä nopeudella.

**Execute:** Suoritus nousevalla reunalla

**Velocity:** Ajonopeus reaalilukuna.

**Direction:** 0 – Pyörimissuunta Velocity-parametrin etumerkin mukaisesti  
 1 – Pyörimissuunta myötäpäivään  
 2 – Pyörimissuunta vastapäivään

**Current:** FALSE – Velocity- ja Direction-parametrien mukainen nopeus ja suunta  
 TRUE – Säilytä nykyinen nopeus ja suunta

**MC\_MOVEJOG**

Ajo määritellyllä nopeudella myötä- ja vastapäivään esim. testiajoihin.

**JogForward:** Ajo myötäpäivään tulon ollessa tosi.

**JogBackward:** Ajo vastapäivään tulon ollessa tosi.

**Velocity:** Ajonopeus reaalitylukuna

### Vinkkejä

Mikäli kelkka lepää referenssikytkimen päällä ajoa aloitettaessa, on se ajettava pois kytkimen päältä ja kotiutettava, jotta saadaan tarkka sijainti.

Voit käyttää liikkeenohjauslohkojen tilatietoja ajosekvenssin ehtoina. Klikkaa ohjattavan lohkon tulonastaa ja valitse haluamasi ohjaava liikkeenohjauslohko, jolloin avautuu uusi valikko, josta saat haluamasi tilatiedon.

Älä salli ajon aloittamista ennen kuin nopeus on syötetty ja turvareleen signaali on aktiivinen. Näin estät ohjelmaa käynnistymästä ”tyhjän päälle”, joka sotkee toiminnan.

### Suurnopeuslaskurit

Samasta paikasta, kuin liikkeenohjauslohkot, löytyy myös lohko **CTRL\_HSC** suurnopeuslaskurin ohjaamiseen. Se löytyy kohdan **Counting** alta. Laskuriinohjauslohkolla voit muuttaa laskurin nykyarvoa, referenssiarvoa, laskentasuuntaa ja taajuusmittauksen jaksonaikaa.

Uusi arvo syötetään nastaan **NEW\_XX** ja se aktivoidaan nastalla **XX** (**XX=CV/RV/PERIOD/DIR**).

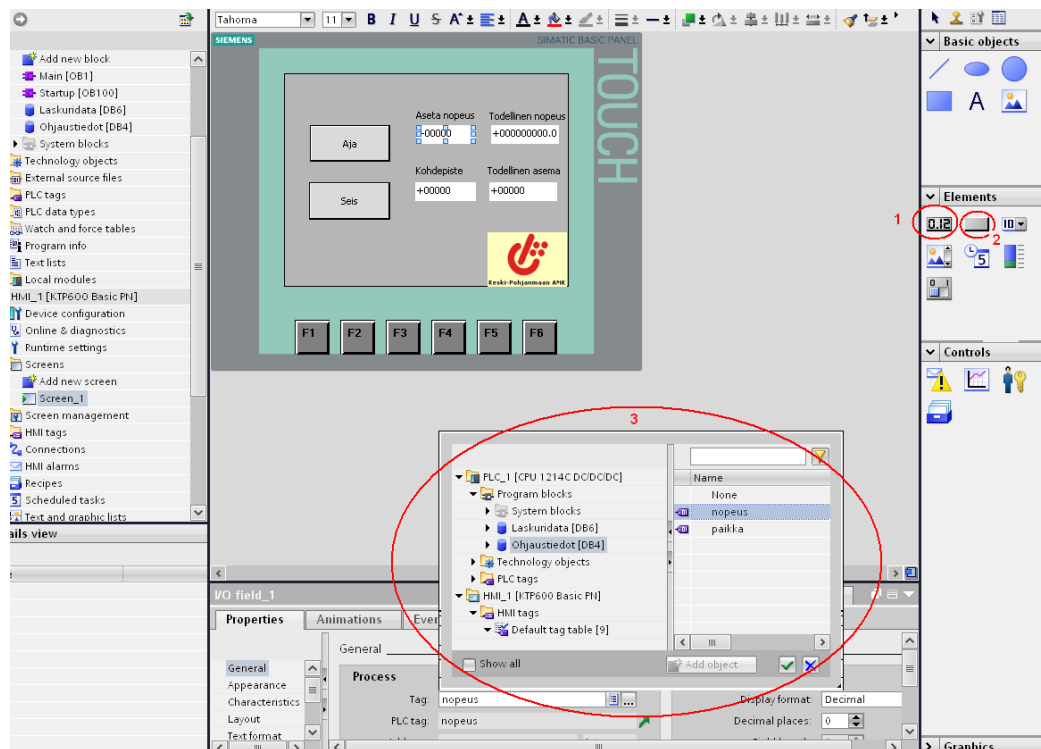
### Vinkkejä

Tässä projektissa täytyy asemalaskuri nollata kotipisteessä.

Suurnopeuslaskurin tiedot pitää viedä datalohkoon, jotta niitä voidaan käsitellä. Laskurit varaavat tietyn osoiteavaruuden, jonka alkuosoitteesta pulssimäärä löytyy. Osoiteavaruuden voit tarkistaa samasta paikasta, jossa alussa loit laskurit (esim. **HSC2**:lla alkuosoite on **%ID1004** ja **HSC3**:lla **%ID1008**) Tee datalohko laskurille, johon teet muuttujat sekä asemalle, että nopeudelle. Vie **MOVE**-käskyllä tiedot laskuriosoitteesta lohkoon.

## Operointipaneeli

Paneelin ohjelmointitilassa voit luoda laitteistolle haluamasi näköisen käyttöliittymän. Paneelille voit tuoda painonappeja, alasvetovalikoita, I/O-kenttiä, tekstiä, kuvia jne. Voit tehdä useita eri ruutuja, joiden välillä siirtyminen voi tapahtua esi. painonapilla tai vaikkapa ohjelmaehtojen seurauksena.



1. I/O-kenttä. Ominaisuuksista valittavissa **Input, Output** tai **Input & Output**. Tällä kentällä saat esitettyä sekä syötettyä esim. kokonaislukumuuttujien tietoa.
2. Painonappi. Esim. muistipaikkojen tai lähtöjen ohjaukseen. Useita eri toimintamalleja. Tässä projektissa käyttökelpoinen esim. **Press-SetBitWhileKeyPressed**.
3. Tagin valinta. Kaikki luomasi tagit löytyvät PLC tags:in alta.

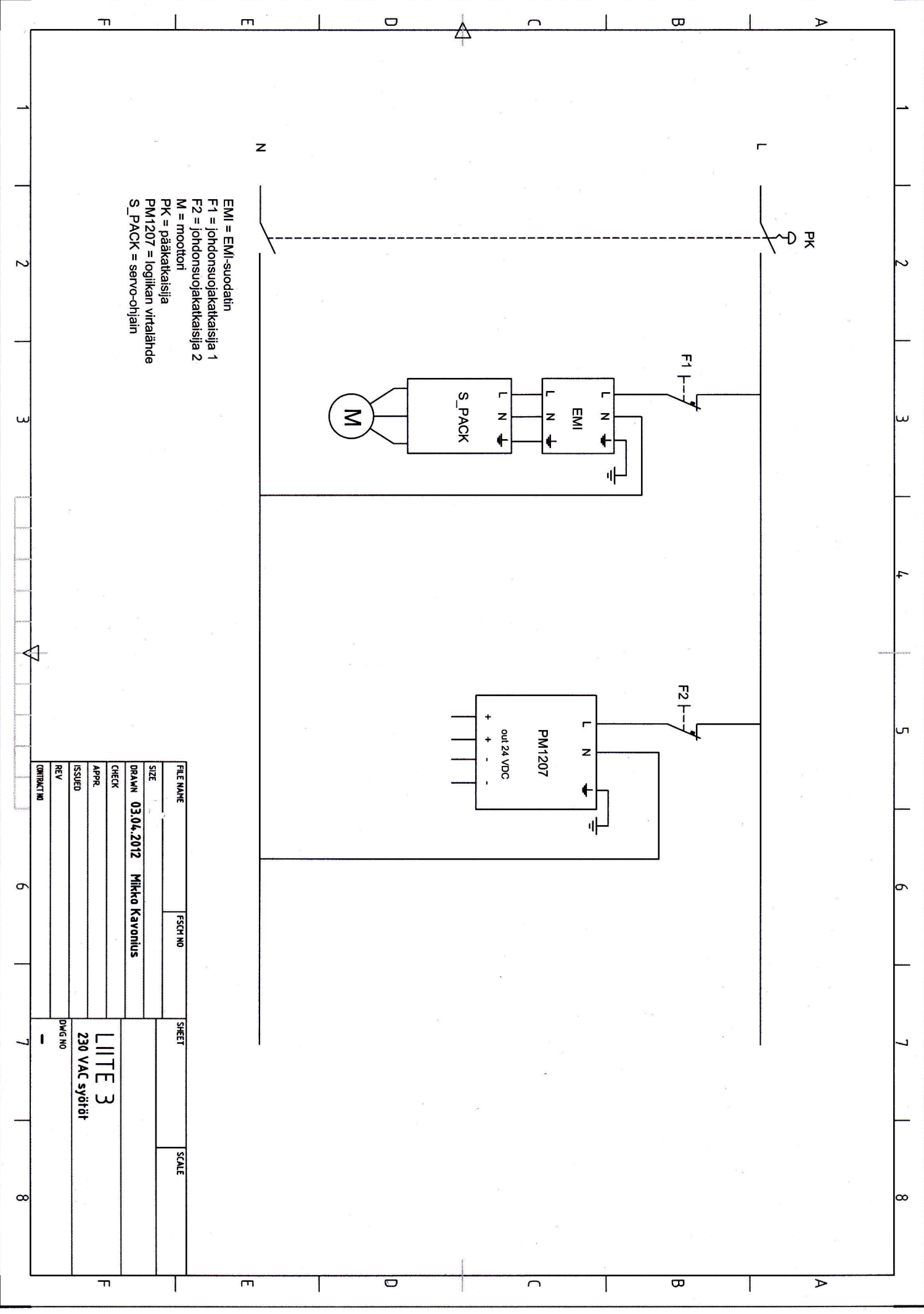
## Projektin suoritusvaiheet

Tässä vielä kerrattuna projektin tekovaiheet. On suositeltavaa tehdä asiat tässä järjestyksessä:

1. PLC:n ja HMI:n määrittely
2. IP-osoitteiden määrittely
3. Pulssigeneraattorin ja suurnopeuslaskurien käyttöönotto

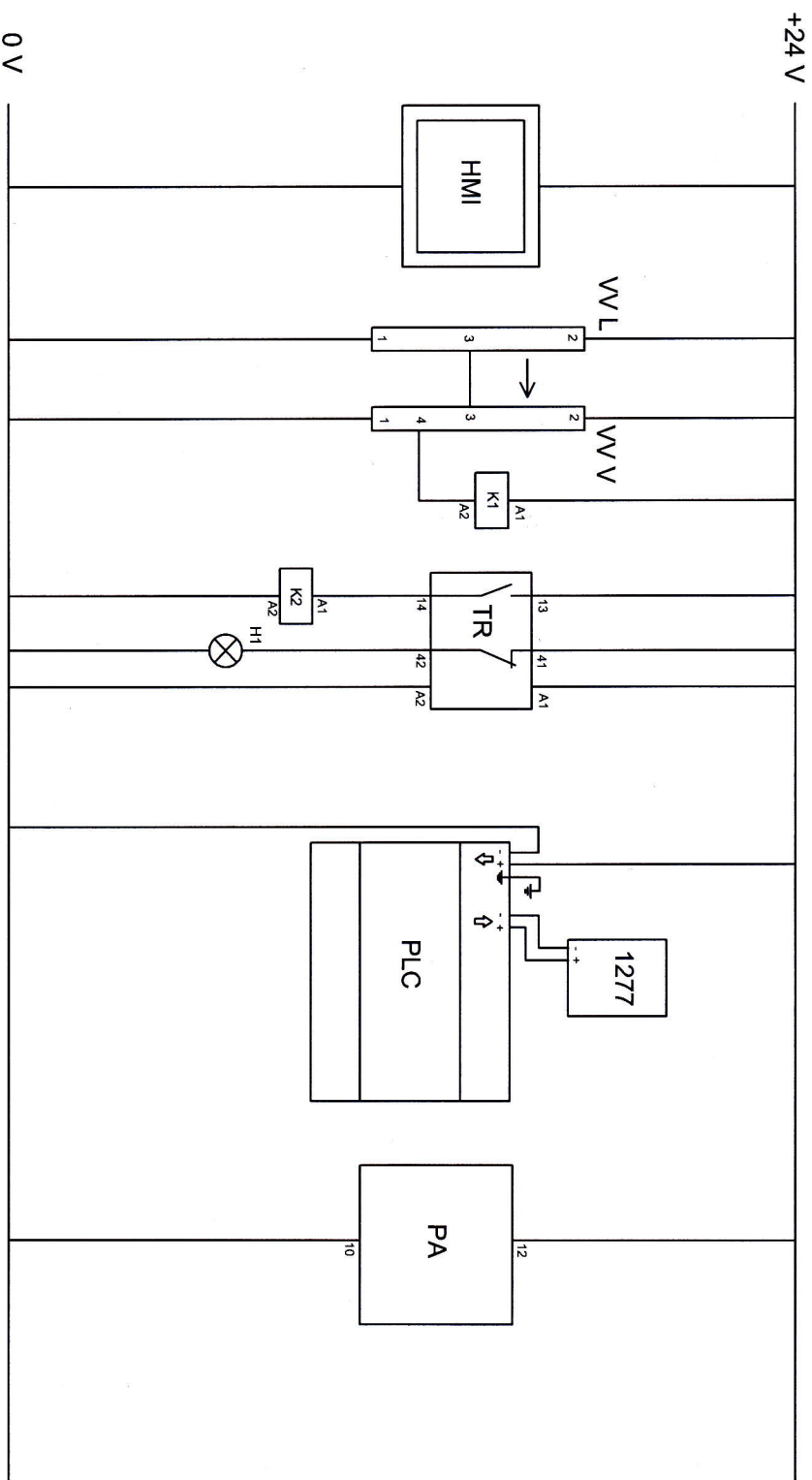
4. Pyöritettävän akselin määrittely
5. Tagilistan teko
6. PLC-ohjelman teko
7. Operointipaneelin muokkaus
8. Ohjelman lataaminen PLC:lle
9. Ohjelman lataaminen HMI:lle





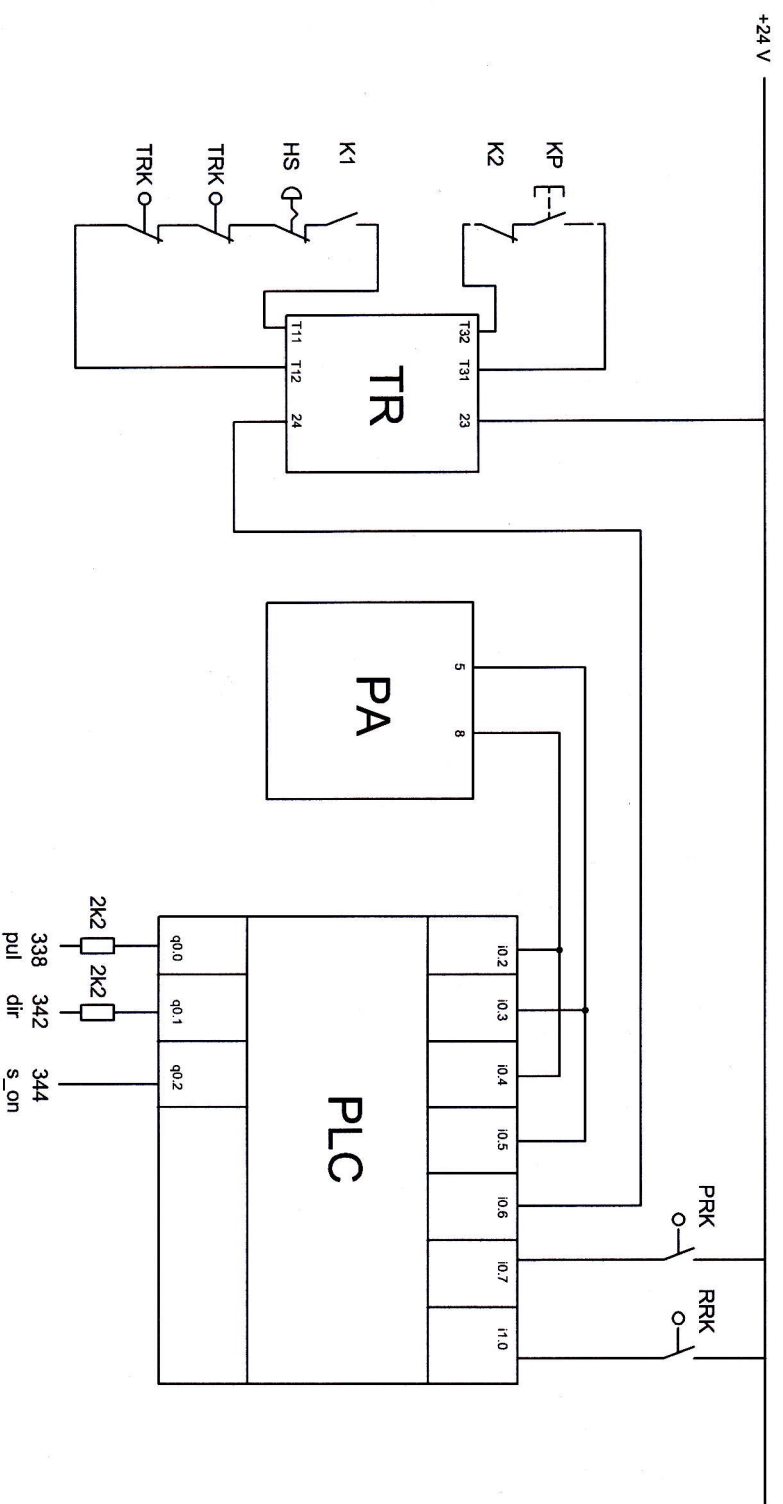
EMI = EMI-suodatin  
 F1 = johdonsuojakatkaisija 1  
 F2 = johdonsuojakatkaisija 2  
 M = moottori  
 PK = pääkatkaisija  
 PM1207 = logiikan viritalähde  
 S\_PACK = servo-ohjain

FILE NAME	FSCH NO	SHEET	SCALE
SIZE			
DRAWN	03.04.2012	Mikko Kavonius	
CHECK			
APPR.			
ISSUED			
REV			
CONTRACT NO			
		LITE 3	
		230 VAC syöttö	
		DWG NO	



HMI = operointipaneeli  
 VV L = valooverho lähetin  
 VV V = valooverho vastaanotin  
 K1 = kontaktori 1  
 K2 = kontaktori 2  
 TR = turvarele  
 H1 = kuittauspainikkeen merkkivalo  
 PLC = logiikka  
 1277 = Profinet-kytkin  
 PA = pulssiarturi

FILE NAME	FSCM NO	SHEET	SCALE
SIZE			
DRAWN	03.04.2012	Mikko Kavonius	
CHECK			
APPR			
ISSUED			
REV			
CONTRACT NO			
		LITE 4	
		24 VDC syötöt	
		DWG NO	
		-	



- HS = hätä-seis-painike
- K1 = kontaktori 1
- K2 = kontaktori 2
- KP = kuitauspainike
- PA = pulssianturi
- PLC = logiikka
- PRK = päätyrajaakytkin
- RRK = referenssiraajaakytkin
- TR = turvarele
- TRK = turvarajaakytkin

FILE NAME	FSCH NO	SHEET	SCALE
SIZE			
DRAWN 03.04.2012 Mikko Kavonius			
CHECK			
APPR.			
ISSUED			
REV			
CONTROL NO			
		<b>LITE 5</b>	
		Pii-kaavio: Toiminnot	
		DWG NO	-